Learning Subjectively Interesting Data Representations

Bo Kang

Promotoren: prof. dr. T. De Bie, dr. J. Lijffijt
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. K. De Bosschere
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2019 - 2020

ISBN 978-94-6355-314-8
NUR 984
Wettelijk depot: D/2019/10.500/122

Examination Committee

Prof. dr. ir. Gert De Cooman, *chair*
Department of Electronics and information systems,
Faculty of Engineering and Architecture
Ghent University

Prof. dr. ir. Thomas Demeester, *secretary*
Department of Information technology,
Faculty of Engineering and Architecture
Ghent University

Prof. dr. Tijl De Bie, *supervisor*
Department of Electronics and Information Systems,
Faculty of Engineering and Architecture
Ghent University

Dr. Jefrey Lijffijt, *supervisor*
Department of Electronics and Information Systems,
Faculty of Engineering and Architecture
Ghent University

Dr. Raúl Santos-Rodríguez
Department of Engineering Mathematics,
Faculty of Engineering
University of Bristol

Prof. dr. Remco Chang
Department of Computer Science,
School of Engineering
Tufts University

Prof. dr. ir. Sofie Van Hoecke
Department of Electronics and Information Systems,
Faculty of Engineering and Architecture
Ghent University

Wir müssen wissen, Wir werden wissen!

David Hilbert

Table of Contents

Acknowledgements

I want to express my sincere gratitude to my supervisors Tijl De Bie and Jefrey Lijffijt, for supervising my thesis, for mentoring me how to become an academic, and for their patience and generosity. I would like to thank Tijl for taking me by the hand and teaching me the Lean Research principles, for his amazing flexibility at work that allows us to walk-in and discuss research problems at any time, and for all those long-hour discussions. I thank Jef for his ability to always ask the right question, for keeping high scientific rigor in our research projects, for his excellent scientific writing skills, which play an important role in the acceptance of our research papers, and for writing the Dutch summary of this thesis.

I want to especially thank the colleagues with whom I collaborated on the papers related to this thesis. Without them, this thesis would not have happened. In addition to my supervisors, these are Raúl Santos-Rodríguez, Darío García García, Kai Puolamäki, Emilia Oikarinen, and Wouter Duivesteijn. Particularly, I want to thank Raúl for all the discussions, for the late nights we were working together toward deadlines, and for all the fun we have had in the last four years. I would also like to thank the members of my examination committee for their insightful questions and valuable comments.

I am deeply grateful to the colleagues who have encouraged, inspired, and supported me along the journey of pursuing my Ph.D. I want to thank Mario Boley and Tamás Horváth for encouraging me to pursue my Ph.D., and for recommending me to Tijl's group. I want to thank Remco Chang for showing me how to effectively involving other people in a thought process toward a research idea, for sharing his invaluable experience of how to corporate with other people to succeed in a paper project, and for the long walks in the beautiful Boston autumn. I thank the fun and inspiring working experiences with the current and former members of the AIDA group and the support staff of IDLab. Especially, I want to thank Xi Chen for her inspiring persistence, and for challenging my thoughts, opinions, showing me a different perspective on things.

I gratefully acknowledge that the research leading to the results presented in this thesis has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the FWO (project no. G091017N, G0F9816N), from the European Union's Horizon 2020 research and innovation programme and the

FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501, from the EPSRC (EP/M000060/1), from the EPSRC (SPHERE EP/R005273/1), and from the Academy of Finland (decision 288814), and Tekes (Revolution of Knowledge Work project).

Thanks to my friends: Cheng Fang, Ziduan Fang, Shan Huang, Jun Liu, Zhaoyi Liu, Jialu Liu, Qiao Wei, Ke Ma, Lotte van den Berg, Jefrey Lijffijt, Jiake Yu, Xia Zhao, and to my little sister Ziying Liao, for all the great memories we shared. Especially, I thank Jiake Yu for her support, encouragement, and accompany over the past years. I want to mention Shan Huang, Ziduan Fang, Cheng Fang, and Jun Liu, for all the fun trips we did and the long-hour chats we had together. I would also like to mention my roommate Xia Zhao for his untamed ambition and endeavor for publishing in top-tier venues, and for his amazing working ethos.

最后，我要谢谢妈妈和爸爸，谢谢你们鼓励我踏上出国求学的路，谢谢你们这些年来对我的支持。你们辛苦了！

Ghent, December 17, 2019
Bo Kang

Samenvatting

Achtergrond. Het 'leren van representaties' is een werkwijze in de machine learning waarbij het doel is om informatie uit data te vatten, door de data-objecten als vectoren in een (typisch) continue ruimte te plaatsen, bijvoorbeeld een Euclidische ruimte. Zo'n representatie geeft een eenvormige kijk op verschillende soorten data (tabellen, grafen, beelden) en breidt daarmee de toepassingsmogelijkheden van traditionele machine-learning en data-analysemethodes uit. Als gevolg heeft het leren van representaties voor een revolutie gezorgd in vele subgebieden van de computerwetenschap, waarvan we hieronder twee voorbeelden geven.

In de exploratieve data-analyse is de hoofdtaak om van data te leren door middel van exploratie. In deze taak coöpereren data scientists vaak met computeralgoritmes. Mensen zijn ongeëvenaard in het opmerken van interessante visuele patronen, terwijl computers uitblinken in het manipuleren van hoog-dimensionale data en zwakker zijn in het identificeren van echt relevante patronen. Daarom gebruiken data scientists vaak dimensionaliteitsreductiemethodes (DR-methodes), een specifieke vorm van het leren van representaties, om hoog-dimensionale data in laag-dimensionale representaties om te zetten. Door deze laag-dimensionale representaties te visualiseren kunnen data scientists de complexe informatieruimte navigeren die verborgen zit in hoog-dimensionale data.

Netwerkanalysetaken (zoals het voorspellen van verbindingen en het classificeren en clusteren van knopen) vereisen dat de eigenschappen van knopen in het netwerk worden uitgeschreven. Deze eigenschappen zijn typisch vooraf met de hand bepaald en hebben een combinatorische aard om te berekenen uit een netwerk. Het extraheren van deze eigenschappen is daardoor vaak computationeel kostbaar. Recentelijk toont een nieuwe categorie methodes voor het leren van representaties, zogenoemde 'netwerkembeddings', de mogelijkheid aan om automatisch de eigenschappen van knopen in een netwerk te leren in de vorm van hoog-dimensionale Euclidische vectoren. Deze hoog-dimensionale vectoren zijn makkelijk te berekenen en tegelijkertijd in staat rijke informatie uit het netwerk te vangen. Netwerkembeddings hebben daardoor de prestaties van voorgenoemde taken in netwerkanalyse substantieel verbeterd.

Tekortkomingen van bestaande methoden. Desondanks de succesvolle toepassingen van het leren van representaties observeerden wij dat bestaande methodes twee beperkingen hebben. Ten eerste, laag-dimensionale representaties (zoals een datavisualisatie) zijn niet in staat de volledige structuur in de data te vatten. Daarom is het wenselijk om data te exploreren door middel van complementaire laag-dimensionale representaties. Bestaande DR-methodes zijn echter

typisch 'statisch' in de zin dat wanneer ze herhaaldelijk toegepast worden, ze vrijwel dezelfde weergave of weergaves geven, die niet noodzakelijk complementair zijn aan elkaar. Een ander probleem van huidige representatiemethodes is dat deze met Euclidische ruimtes werken. Ondanks dat Euclidische ruimtes veel wenselijke eigenlijkhappen hebben (zoals continuïteit), is hun expressieve kracht om de complexe structuur in de data te representeren fundamenteel beperkt.

Bijdragen. Om deze twee beperkingen aan te pakken stellen we een nieuw raamwerk voor om representaties te leren. Het raamwerk beschrijft hoe representaties gevonden kunnen worden door zowel data als voorkennis over de data als invoer nemen en algoritmes te gebruiken om representaties te construeren die deze voorkennis complementeren met een maximum aan informatie over de data. Door gebruik te maken van het raamwerk kunnen we dus de eerste beperking aanpakken: bereken iteratief een laag-dimensionele visuele representatie die de kennis bevat in eerdere visualisaties complementeert. Voor de tweede beperking kunnen we juist de structuur die moeilijk te representeren is in een Euclidische ruimte als voorkennis opnemen en de een representatie gebruiken om de complementaire informatie the vatten. Door de voorkennis en representatie te combineren verkrijgen we een beter model van de data. We refereren naar een representatie gevonden via ons raamwerk als een Subjectief-Interessante Data Representatie (SIDR) en naar het raamwerk als het SIDR-raamwerk. De term 'subjectief' komt van het feit dat het raamwerk representaties geeft die contrasteren met voorkennis. De term 'interessante' is omdat de informatie die verrassend is ten opzichte van wat er van tevoren geweten is, typisch is wat echt interessant is.

In dit proefschrift introduceren we eerst het SIDR-raamwerk. Vervolgens presenteren we vijf uitvoeringen van het raamwerk, waarbij iedere uitvoering overeenkomt met een sleutelpublicatie. Een uitvoering van het SIDR-raamwerk benadrukt een aspect van het raamwerk en laat zien hoe het SIDR-raamwerk gebruikt kan worden om de voorgenoemde twee beperkingen aan te pakken. We bespreken deze uitvoeringen kort in de rest van deze samenvatting. Allereerst introduceren we de methodes die subjectief-interessante representaties voor verschillende datatypes vinden, namelijk reëelwaardige tabellen alsook netwerken. In de rest bediscussiëren we uitvoeringen die gericht zijn op specifieke toepassingen zoals mens-computer interactie en interpreteerbaarheid.

Lineaire dimensionaliteitsreductiemethodes worden veel gebruikt in exploratieve data-analyse. Door data naar een laag-dimensionale ruimte te projecteren kunnen data scientists de datarepresentatie weergeven en inzichten opdoen. Hoogdimensionale data bevat echter typisch een complexe structuur en een enkele laagdimensionele representatie is niet in staat de volledige structuur te vatten. Complementaire representaties zijn dus gewenst voor efficiënte data-exploratie. Bestaande lineair DR-methodes zijn allemaal statisch in de zin dat, afgezien van stochasticiteit in de optimalisatieprocedure, ze vrijwel of exact dezelfde representatie produceren wanneer ze herhaaldelijk toegepast worden. Om een complementaire representatie te verkrijgen kan men naar verschillende combinaties van lineaire projecties kijken. Er is echter geen garantie dat een van de geconstrueerde dimensies maximaal complementair is aan de voorkennis van een analist. Voor

dit doel stellen we een methode voor het leren van lineaire representaties voor, genaamd Subjectief-Interessante ComponentenAnalyse (SICA). SICA vindt laagdimensionale representaties die de voorkennis van een analist complementeren. Uit uitgebreide empirische studies die we hebben gedaan blijkt de effectiviteit van SICA in het vinden van complementaire lineaire representaties.

Non-lineaire dimensionaliteitsreductiemethodes genieten de voorkeur boven lineaire wanneer het doel is om meer complexe structuren in de data te vatten. Zij lijden echter ook aan de eerder besproken beperkingen, zoals de lineaire DR-methodes doen. Om dit aan te pakken hebben we ook een methode voor het leren van non-lineaire representaties ontwikkeld die een uitvoering is van het SIDR-raamwerk. Omdat de resulterende methode complementaire representaties vindt door middel van het generaliseren van t-Stochastic Neighbor Embedding (t-SNE) refereren we ernaar als conditionele t-SNE (ct-SNE in het kort).

Er zijn ook structuren in de data die niet volledig gevat kunnen worden in Euclidische vectoren. Netwerkdata valt in deze categorie door de complexe structurele eigenschappen zoals multipartietheid, bepaalde graadverdelingen en assortativiteit. Om de netwerkdata beter te modelleren modeleren we eerst de complexe structuurinformatie als voorkennis. Vervolgens gebruiken we het SIDR-raamwerk om complementaire informatie te leren, opgeslagen als Euclidische vectoren. Het combineren van voorkennis en vectorrepresentaties resulteert in een beter model van de data. We noemen deze methode Conditionele NetwerkEmbedding (CNE). CNE presteert consistent beter dan de bestaande methodes, op taken zoals verbindingen voorspellen en classificatie van knopen.

De voorgaande uitvoeringen maken aannames over de voorkennis die een analist kan hebben. Het kan echter de voorkeur genieten om de data scientist direct op te nemen in het proces van het leren van representaties, om iteratief meer inzichtelijke representaties te ontdekken. Om dit te bereiken hebben we een methode ontwikkeld voor Subjectief-Interessante Data Exploratie (SIDE). SIDE geeft een visuele (lineaire) representatie weer aan de gebruiker en staat de gebruiker toe om interessante clusters te specificeren die zij in de representatie heeft gevonden. SIDE neemt aan dat de data scientist deze clusterstructuur assimileert en modelleert de positie van de punten in de clusters als voorkennis. Daarna worden complementaire representaties gezocht die met de voorkennis contrasteren. Meerdere casusstudies tonen aan dat SIDE nuttig is voor het iteratief en interactief ontdekken van subjectief-interessante structuren uit data.

Recentelijk is de aandacht voor methodes voor interpreteerbare machine learning (ML) sterk toegenomen, doordat een groeiend aantal aan reglementen interpreteerbaarheid vereisen van ML-methodes. De voorgaande uitvoeringen van het SIDR-raamwerk richten zich niet expliciet op deze toepassing. Het naïef interpreteren van subjectief-interessant datarepresentaties vereist representaties direct interpreteerbare assen te hebben of de representaties op post-hoc te analyseren. In onze laatste uitvoering, Subjectief-Interessante Subgroep ontDekking (SISD), brengen we interpreteerbaarheid naar het SIDR-raamwerk door middel van het zoeken naar subjectief-interessante representaties die tegelijk informatief en beschrijvend zijn. Empirische studies tonen aan dat SISD inderdaad representaties kan vinden waar-

van de interessantheid goed verklaard kan worden.

Samenvattend, dit proefschrift presenteert het SIDR-raamwerk dat beschrijft hoe datarepresentaties gevonden kunnen worden die voorkennis complementeren. We tonen vijf uitvoeringen die het gebruik van het SIDR-raamwerk op verschillende manieren belichten. Uitgebreide empirische studies tonen de capaciteit aan van het SIDR-raamwerk om effectief datarepresentaties te vinden die inderdaad subjectief interessant zijn.

Perspectieven. De resultaten gepresenteerd in dit proefschrift hebben mogelijk impact in drie richtingen. Ten eerste, nieuwe methoden voor het leren van subjectieve representaties kunnen afgeleid worden van het SIDR-raamwerk. Ten tweede, de gepresenteerde uitvoeringen van het SIDR-raamwerk kunnen direct toegepast worden in exploratieve data-analyse. Data scientists kunnen de geïntroduceerde DR-methodes zoals SICA, ct-SNE en SIDE gebruiken om data te exploreren via subjectief-interessante visualisaties. De patroonontdekkingsmethode SISD kan ook toegepast worden om goed-uitgelegde subgroeppatronen te vinden die subjectief-interessante informatie over de data onthullen. Ten laatste, CNE toont superieure prestaties in netwerkanalysetaken zoals verbindingen voorspellen en knopen classificeren. We verwachten dat het integreren van CNE in bestaande netwerkembedding-gebaseerde machine-learning processen een positieve impact heeft of de prestaties.

Summary

Background. Representation learning is a recently widely popularized machine learning approach that aims to capture the information in data by embedding the data objects as vectors in a (typically) continuous space (e.g., Euclidean space). Such a representation provides a unified view of the different types of data (e.g., tabular, graph, image) and thus further extends the application scope of traditional machine learning and data analytics methods. As a result, representation learning revolutionized many subfields in computer science, for which we provide two examples below.

In exploratory data analysis, the main task is to learn from the data via exploration. In this task, human operators often work interactively with computer algorithms. Humans are unmatched in spotting interesting visual patterns, while computers excel in manipulating high-dimensional data and are weaker at identifying truly relevant patterns. Thus human operators often use dimensionality reduction methods (DR), a type of representation learning, to transform high-dimensional data into low-dimensional representations. By visualizing those low-dimensional representations, the human operators are able to navigate the complex information space hidden within high-dimensional data.

In network analysis, tasks such as link prediction, node classification, and clustering rely on the features extracted from the network data. The types of features are typically manually engineered and often have combinatorial nature. Hence extracting these features is often computationally expensive. Recently, a new category of representation learning methods called 'network embeddings' demonstrate the ability to automatically learn node features of a network in the form of high-dimensional Euclidean vectors. These high-dimensional vectors are easy to compute and at the same time, able to capture rich information from the network. As a result, network embeddings substantially improved the performance of the aforementioned tasks in the network analysis.

Shortcomings in the state-of-art. Despite the successful applications of representation learning approaches, we observed they have two limitations. First, since a low-dimensional representation (e.g., a data visualization) is not able to capture the complete structure in the data, it is desirable to explore the data via complementary low-dimensional representations. However, the existing DR methods are typically 'static' in the sense that when applied repeatedly, they provide the same view or views that do not necessarily complement each other. Another issue of current representation methods is the data are typically embedded in Euclidean space. Despite Euclidean space having many desirable properties (e.g., being con-

tinuous), its expressive power to represent the complex structure in the data is limited.

Contributions. We propose a representation learning framework to alleviate these two limitations. The framework delineates how to take data plus certain prior knowledge about the data as input, and finds a representation of the data that complements the prior and conveys the most information about the data. Using the framework, we can thus address the first limitation by iteratively computing the low-dimensional visual representations of the data that complement the knowledge conveyed by the previous visualizations. For the second limitation, we can encode the structure that is difficult to represent in the Euclidean space as prior knowledge, and let the representation capture complementary information. Combining both the prior and the representation, we obtain a better model of the data. We refer to a representation found by the instantiations of our framework as *Subjectively Interesting Data Representation* (SIDR) and to the framework as the *SIDR framework*. The term 'subjective' comes from the fact that the SIDR framework outputs representations that contrast with a priors. The term 'interesting' is because the information that is surprising to what is known a priori is typically what is truly interesting.

In this thesis, we first introduce the SIDR framework. Then, we present five instantiations of the framework, each corresponding to a key publication. Each instantiation of the SIDR framework emphasizes one aspect of the framework and demonstrates the ability of the SIDR framework to address the two limitations. We briefly discuss these instantiations in the remainder of this summary. First, we introduce the methods that find subjectively interesting representations for different data types, namely real-valued tabular data as well as networks. In the remainder, we discuss the instantiations that focus on more application perspectives such as human-computer interaction and interpretability.

Linear dimensionality-reduction methods are common tools in exploratory data analysis. By projecting the data onto a low-dimensional space, the human operator can visualize the data representation and gain insights. However, high-dimensional data typically contains a complex structure, and a single low-dimensional representation is not able to capture the full structure. Thus complementary representations are desired for efficient data exploration. Existing linear DR methods are all static in the sense that, besides the randomness in the optimization procedure, they produce the same representation if applied repeatedly. To obtain a complementary representation, one may indeed look at different combinations of the linear projections. However, there is no guarantee that the constructed dimensions maximally complement the prior knowledge of an analyst. To this end, we proposed a linear representation learning method called Subjectively Interesting Component Analysis (SICA). SICA finds low-dimensional representations that complement the prior knowledge of an analyst. We conducted extensive empirical studies and showed the effectiveness of SICA in finding complementary linear representations.

Non-linear dimensionality-reduction methods are preferred to the linear ones when the goal is to capture more complex structure in the data. However, they also suffer from the limitations discussed above, as the linear DR methods do. To

address this, we also developed a non-linear representation-learning method that instantiates the SIDR framework. Because the resulting method finds complementary representations by generalizing t-distributed Stochastic Neighbor Embedding (t-SNE), we refer it to as conditional t-SNE (ct-SNE in short).

There are also structures in the data that can not be fully captured using Euclidean vectors. Network data falls into this category due to its complex structural properties such as (approximate) multipartiteness, certain degree distributions, assortativity. To better model the network data, we first encode complex structure information as prior knowledge. Then we use the SIDR framework to derive a method that models complementary information, stored as Euclidean vectors. Combining both the prior and vector representations results in a better model of the data. We call the method Conditional Network Embedding (CNE). CNE consistently outperforms the baseline methods on downstream tasks such as link prediction and node classification.

The previous instantiations make assumptions about the prior knowledge an analyst may have. However, it may be more desirable to directly involve the human analyst in the representation learning process to iteratively discover more insightful representations. To achieve this, we developed a method for Subjectively Interesting Data Exploration (SIDE). SIDE shows a visual (linear) representation of a dataset to the user and allows the user to specify the interesting clusters she found in the data. SIDE assumes the user assimilates this clustering structure and encode the position of the points in the clusters as prior. Then the complementary representations that contrast with the prior are sought. Multiple case studies show SIDE is useful for iteratively and interactively discovering subjectively interesting structure from data.

Recently, interpretable machine-learning (ML) methods have gained lots of attention because of a growing number of regulations requiring interpretability of the ML methods. The previous instantiations of the SIDR framework are not explicitly focusing on this perspective. Naively, interpreting the subjectively interesting data representations requires either the representation to be sparse or to be analyzed in a post-hoc manner. In our final instantiation, Subjectively Interesting Subgroup Discovery (SISD), we bring interpretability to the SIDR framework by jointly searching subjectively interesting representations that are both informative and descriptive. Empirical studies show SISD can indeed find representations whose interestingness is well explained.

To sum up, this thesis presents the SIDR framework that formalizes how to find data representations that complement the prior knowledge of a user. We show five instantiations that highlight the usage of the SIDR framework in different perspectives. Extensive empirical studies demonstrate the capability of the SIDR framework to effectively find data representations that are indeed subjectively interesting.

Perspectives. Results presented in this thesis have potential impact in three directions. First, new subjective representation learning methods can be derived from the SIDR framework. Second, presented instantiations of SIDR framework can be directly applied in exploratory data analysis. Analysts can use the introduced DR

methods such as SICA, ct-SNE, and SIDE to explore data via subjectively interesting visualizations. Pattern discovery method SISD can also be applied to find well-explained subgroup patterns that reveal subjectively interesting information about the data. Finally, CNE demonstrates superior performance in downstream tasks (e.g., link prediction, node classification) in graph learning. Integrating CNE into existing network-embedding based machine learning pipelines is expected to have positive impact on the performance.

Acronyms

AAE	Adversarial Auto Encoder
CCA	Canonical Component Analysis
CNE	Conditional Network Embedding
CORAND	Constrained Randomization
ct-SNE	Conditional t-distributed Stochastic Neighbor Embedding
DL	Description Length
DR	Dimensionality Reduction
EMM	Exceptional Model Mining
FORSIED	Formalizing Subjective Interestingness In Exploratory Data Mining
IC	Information Content
IM	Interestingness Measure
k-NN	k-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
NE	Network Embedding
PCA	Principal Component Analysis
PCP	Parallel Coordinate Plot
PP	Projection Pursuit
SD	Subgroup Discovery
SICA	Subjectively Interesting Component Analysis
SIDE	Subjectively Interesting Data Exploration
SIDR	Subjectively Interesting Data Representation

SISD	Subjectively Interesting Subgroup Discovery
SVD	Singular Value Decomposition
SVM	Support Vector Machine
t-SNE	t-distributed Stochastic Neighbor Embedding

1 Introduction

1.1 Motivation

Machine learning and data analytics methods heavily rely on the availability of good data representations (or features). For example, manual feature engineering is widely adopted in traditional prediction and classification models to boost these models' performance. The recent development of deep neural networks automated the process of identifying and extracting good data representations, which in return, popularized deep neural networks. Word and network embedding methods learn representations from textual and network data thus extend the application scope of traditional machine learning methods. In data analytics, dimensionality reduction methods are often applied to construct lower-dimensional representations of data. Visualizing low-dimensional representation allows human analysts to gain intuition about the data efficiently. The common practices of automatically discovering the representations from raw data are collectively identified as "representation learning" by the machine learning community.

Despite the success of representation learning in both academia and industry, it currently has two limitations. First, low-dimensional data representation is typically insufficient to capture all structure in the data, and the most salient structure is often already known. Still, it is not obvious how to extract the remaining information in a similarly effective manner. Another limitation is the original data can be fundamentally more expressive than data representations that are typically formulated as vectors in the Euclidean space. This potentially causes the complex

structure in the data not being fully captured by the representations.

To overcome these limitations, we propose a representation learning framework that delineates how to take a prior about the data as an extra input and finds a representation that complements this prior. By discounting the known salient structure in the data, the framework enables complementary structure to be captured in the representation, providing new insights. Second, a structure that is difficult for Euclidean vectors to represent can also be encoded as a prior in the proposed framework. This allows the Euclidean representation to model complementary information in the data. By combining the prior and the Euclidean representation, methods derived from the proposed framework yields a better model of the data. We term the representation constructed by the methods derived from our framework as *subjectively interesting data representation* (SIDR). We use the term 'subjective' because the representation complements a certain prior. We use the term 'interesting' because the information that is surprising to what is known a priori is typically what is truly interesting [1].

1.2 Contributions

The main contribution of this dissertation is a mathematically principled framework for learning subjectively interesting representations, to which we refer as the SIDR framework for brevity. Based on this framework, we developed five instantiations. Each instantiation focuses on a different perspective of the framework: linear representations (Chapter 1), non-linear representations (Chapter 2), network representations (Chapter 3), representation learning with human in the loop (Chapter 4), and interpretable representations (Chapter 5). In this section, we formally define the SIDR framework and summarize the five instantiations. These instantiations are discussed in greater detail in the later chapters.

1.2.1 SIDR framework

The SIDR framework consists of three components: (1) a prior distribution that encodes the known information about data. (2) A conditional distribution that describes the relationship between data and its representations. (3) An optimization strategy for finding the most informative data representation that complements the prior. SIDR quantifies the informativeness of a representation in terms of probability, which converts the problem of searching for the most informative complementary representation into a mathematical optimization problem. This further allows the SIDR framework to address the two limitations of current representation learning methods in a mathematically principled manner.

To formally describe the framework, we first need to introduce a few concepts. Following the definition in [2, 3], we assume the data is drawn from a set of possi-

ble values, termed the data space. Their formal definitions are as follows.

Definition 1 (Data space and data)**.** *Given a set $\mathbf{\Omega}$ called the data space, the data is an element $\hat{\boldsymbol{D}} \in \mathbf{\Omega}$ with corresponding random variable denoted as $\boldsymbol{D}$.*

In this dissertation, we specify the data $\hat{\boldsymbol{D}}$ to be a set of data objects. For real-valued tabular data [4, 5, 6, 7], $\hat{\boldsymbol{D}}$ consists of $|\hat{\boldsymbol{D}}|$ real vectors with dimensionality d. We further overload the notation $\hat{\boldsymbol{D}}$ as a $|\hat{\boldsymbol{D}}| \times d$ real-matrix by stacking the vectors row-wise. This leads to a data space $\mathbf{\Omega} = \mathbb{R}^{|\hat{\boldsymbol{D}}| \times d}$. For a network with a node set $\hat{\mathbf{V}}$ and a set of links $\hat{\mathbf{E}}$ [8], the corresponding data $\hat{\boldsymbol{D}}$ consists of $|\hat{\mathbf{V}}|$ data objects ($|\hat{\boldsymbol{D}}| = |\hat{\mathbf{V}}|$), where each data object is a set of neighboring nodes (defined by $\hat{\mathbf{E}}$) of the corresponding node. Thus, the data space is the set of all possible links between the nodes, namely $\mathbf{\Omega} = 2^{\binom{\hat{\mathbf{V}}}{2}}$.

Example. To better understand the concepts in the SIDR framework, we introduce a running example of exploring data via subjectively interesting linear representations [4]. In this example, we would like to explore the extended Yale Face Database B [9, 10]. The dataset contains 1684 frontal images (32×32 gray-level) of human subjects under various illumination conditions. Thus, we have real-valued tabular data which belongs to the data space $\mathbf{\Omega} = \mathbb{R}^{1684 \times 1024}$. As the first exploration step, we performed Principal Component Analysis (PCA) [11] on the data. We observed the top principal components given by PCA (namely the Eigenfaces) are influenced substantially by the variation in the illumination conditions. Knowing that the variation of the illumination conditions is the dominant structure in the data, we would like to explore other structure further via low-dimensional representations of the data.

In SIDR, data representation is defined as a set of Euclidean vectors that convey information about certain aspects of the data.

Definition 2 (Data representation)**.** *A representation $\hat{\boldsymbol{R}}$ of data $\hat{\boldsymbol{D}}$ consists of $|\hat{\boldsymbol{D}}|$ real-valued k-dimensional vectors, namely $\hat{\boldsymbol{R}} \in \mathbb{R}^{|\hat{\boldsymbol{D}}| \times k}$. Given data space $\mathbf{\Omega}$, a representation $\hat{\boldsymbol{R}}$ of data $\hat{\boldsymbol{D}}$ corresponds to a subset $\mathbf{\Omega}_{\hat{\boldsymbol{R}}}$ of the data space such that $\mathbf{\Omega}_{\hat{\boldsymbol{R}}} \subseteq \mathbf{\Omega}$ and $\hat{\boldsymbol{D}} \in \mathbf{\Omega}_{\hat{\boldsymbol{R}}}$.*

Example cont. In our running example, we explore the data via its linear representations. The term "linear" means the representation is obtained by projecting the data onto an orthonormal matrix $\boldsymbol{W} \in \mathbb{R}^{d \times k}$, i.e., $\hat{\boldsymbol{R}} = \hat{\boldsymbol{D}}\boldsymbol{W}$. Thus representation of the image data is a set of k-dimensional vectors. Since only a restricted set of images $\mathbf{\Omega}_{\hat{\boldsymbol{R}}} \subseteq \mathbf{\Omega}$ has the representation $\hat{\boldsymbol{R}}$ after projecting onto the matrix $\boldsymbol{W}$, the representation $\hat{\boldsymbol{R}}$ corresponds to a subset of the data space $\mathbf{\Omega}_{\hat{\boldsymbol{R}}}$.

With the concepts introduced above, we can now define the **first component** of SIDR.

Definition 3 (Prior distribution)**.** *A prior distribution $p_{\boldsymbol{D}}$ is a probability distribution over the data space $p_{\boldsymbol{D}} : \mathbf{\Omega} \to \mathbb{R}$ that encodes certain prior knowledge about*

the data. [1]

Here we briefly discuss how to encode prior knowledge and subsequently compute the prior distribution. For real-valued tabular data, the prior knowledge can be quantified using statistics of the data $f : \mathbf{\Omega} \to \mathbb{R}^m$, where m is the dimensionality of the output of measure function f. To derive the prior distribution, we can pose the statistics as constraints on the expectations of the prior distribution, with a form: $\mathbb{E}_{\boldsymbol{D}\sim p_{\boldsymbol{D}}}[f(\boldsymbol{D})] = \hat{\mathbf{m}}$, where $\hat{\mathbf{m}}$ is a vector of the statistics measured on the observed data. However, such constraints do not determine the distribution $p_{\boldsymbol{D}}$ fully, so we determine $p_{\boldsymbol{D}}$ as the distribution with maximum entropy (MaxEnt) from all distributions satisfying the constraints. The resulting MaxEnt distribution is the distribution with least injected information other than the information of the constraints. Finding this distribution is a convex optimization problem which can often be solved efficiently [3].

Example cont. In our facial image exploration example, the illumination conditions are known to be one of the dominant factors in the data. Thus, each data object is labeled with one of the illumination conditions. This knowledge can be translated by declaring that images with the same illumination condition are similar to each other. The similarities can be further formalized using a graph with the edges M connecting all the pairs of nodes (data objects) that share the same illumination condition. In this way, the prior knowledge regarding similarity between the data objects can be measured as the average pairwise Euclidean distance of connected nodes in the graph: $\mathbb{E}\left[\frac{1}{|M|}\sum_{(i,j)\in M}||\boldsymbol{D}_{i,:} - \boldsymbol{D}_{j,:}||^2\right] = \hat{m}$, where $\hat{m}$ is a constant scalar obtained by computing the same pairwise similarity measure on the observed data. Maximizing the entropy of the distribution under the pairwise similarity prior results in a matrix normal distribution [4].

Other priors derived from real-valued tabular data are the magnitude of the scale of the data [4], the mean and the variance statistics of full or subsets of data [6, 7]. For network data, the prior knowledge can be the degree of each node (i.e., the number of directly connected nodes) and the density of the links within each community structure of the network [8]. Apart from deriving the prior distribution using the MaxEnt approach, the prior distribution can also be directly postulated based on the assumptions made on the data [5].

As the **second component** of SIDR, the conditional distribution of representation is defined as follows:

Definition 4 (Conditional distribution)**.** *The conditional distribution that describes the relation between data and its representation is referred to as* $p_{\boldsymbol{R}|\boldsymbol{D}}$.

Using Bayes' rule, we can further derive the distribution of the data $\boldsymbol{D}$ condi-

[1] For real tabular data, $p_{\boldsymbol{D}}$ is a density function. For network data, $p_{\boldsymbol{D}}$ is a discrete probability distribution.

tioning on $\boldsymbol{R}$:

$$p_{\boldsymbol{D}|\boldsymbol{R}} = \frac{p_{\boldsymbol{R}|\boldsymbol{D}} p_{\boldsymbol{D}}}{p_{\boldsymbol{R}}}.$$

where $p_{\boldsymbol{R}}$ is the marginal distribution and can be obtained by marginalizing the joint distribution $p_{\boldsymbol{R},\boldsymbol{D}} = p_{\boldsymbol{R}|\boldsymbol{D}} p_{\boldsymbol{D}}$ over $\boldsymbol{D}$.

The form of the conditional distribution varies for different instantiations. In the running example, the linear representation random variable $\boldsymbol{R}$ is uniquely determined by the data random variable $\boldsymbol{D}$ and projection matrix $\boldsymbol{W}$, namely $\boldsymbol{R} = \boldsymbol{D}\boldsymbol{W}$ and $p_{\boldsymbol{R}|\boldsymbol{D}} = 1$. Applying Bayes' rule, the distribution of the data given the representation reads: $p_{\boldsymbol{D}|\boldsymbol{R}} = \frac{p_{\boldsymbol{D}}}{p_{\boldsymbol{R}}}$, where $p_{\boldsymbol{R}}$ is the marginal density function obtained by projecting the prior distribution onto matrix $\boldsymbol{W}$. This formulation of the conditional distribution gives zero probability to the datasets in the data space that cannot have representation $\boldsymbol{R}$ via projection matrix $\boldsymbol{W}$, namely $p_{\boldsymbol{D}} = 0$ for $\boldsymbol{D} \notin \boldsymbol{\Omega}_{\boldsymbol{R}}$.

Besides the proposed conditional distribution for linear representations [4, 6, 7], this dissertation also introduces other forms of conditional distributions for non-linear representations of tabular data [5] and network representations [8]. We further motivate the choice of conditionings in the later chapters.

Building on the previous two components, we now define the **third component** of SIDR, namely the strategy of finding a representation that both complements the prior and is maximally informative about the data.

Definition 5 (Search strategy)**.** *Given dataset $\hat{\boldsymbol{D}}$ and posterior probability $p_{\boldsymbol{D}|\boldsymbol{R}}$, searching for subjectively interesting representations is to find a representation $\hat{\boldsymbol{R}}$ with that maximizes the conditional probability of the data:*

$$\underset{\hat{\boldsymbol{R}}}{\textit{argmax}} \; p_{\boldsymbol{D}|\boldsymbol{R}}(\hat{\boldsymbol{D}}|\hat{\boldsymbol{R}}).$$

Example cont. In our face image data example, since the representations are given projection matrix $\boldsymbol{W}$, finding the complementary linear representation is equivalent to maximizing the probability $p_{\boldsymbol{D}|\boldsymbol{R}}(\hat{\boldsymbol{D}}|\hat{\boldsymbol{R}}) = \frac{p_{\boldsymbol{D}}(\hat{\boldsymbol{D}})}{p_{\boldsymbol{R}}(\hat{\boldsymbol{R}})}$ over projection matrix $\boldsymbol{W}$. Note this is also equivalent to finding a linear representation $\hat{\boldsymbol{R}}$ that has the smallest probability under the marginalized prior distribution (i.e., $p_{\boldsymbol{R}}(\hat{\boldsymbol{R}})$). Thus the resulting representation is maximally informative (measured by the negative log probability $-\log p_{\boldsymbol{R}}(\hat{\boldsymbol{R}})$ [12]) and also complements the prior. This is exactly what we aimed for.

1.2.2 SIDR instantiations

We study five instantiations of the SIDR framework with each instantiation focusing on a different perspective of the framework. First, we introduce the methods that finds subjectively interesting representations for different data types, namely

real-valued tabular data as well as networks. In the remainder, we discuss the instantiations that focus on more applicative perspectives such as human-computer interaction and interpretability.

Chapter 2 Linear representation To explore high-dimensional data, dimensionality reduction (DR) methods are typically used to obtain a low-dimensional (2D/3D) data representation. By visualizing the representation, human analysts can efficiently explore and find structure in the data. However, existing linear DR methods yield a single static representation, which in most cases is insufficient to capture all structure in the data. Furthermore, since different human analysts have different prior knowledge and interests, they are unlikely to have an equal interest in the same low-dimensional representation. One may indeed construct high-dimensional representations, hoping to discover more structure, but there is no guarantee that the constructed dimensions complement the prior knowledge of an analyst. To address these issues, we present Subjectively Interesting Component Analysis (SICA) [4]. SICA instantiates the SIDR framework by searching for linear representations that reveal the complementary information (with respect to the prior knowledge) about the data. SICA is evaluated in both qualitative and quantitative experiments against several synthetic and real-world datasets. The experiments suggest that SICA enables users to find low-dimensional linear representations while discounting prior information.

Chapter 3 Non-linear representation Comparing to Linear DR methods which create low-dimensional data representations via linear projection, non-linear DR methods are more powerful in the sense that they can capture complex non-linear structure. Despite the popularity of the non-linear DR methods, they all yield a single static representation, which is insufficient to capture all structure in the data. In addition, the salient structure in the representation is often already known. To effectively explore the data, some new structure needs to be captured in the subsequent low-dimensional representations. Thus, we present Conditional t-distributed Stochastic Neighbor Embedding (ct-SNE) [5], a conditioned version t-distributed Stochastic Neighbor Embedding (t-SNE) [13]. ct-SNE encodes prior information from the lower-dimensional representation in the form of labels, namely the same-labeled data objects are expected to be more similar and vice-versa. By discounting the prior information, the low-dimensional representation focuses on reflecting the proximities that complements the prior. Extensive case studies on both synthetic and (large) real-world datasets show ct-SNE effectively factors out prior knowledge and allows the complementary structure to be captured in the low-dimensional embedding, providing new insights.

Chapter 4 Network representation Network embedding (NE) methods map nodes into corresponding Euclidean vector representations hence enabling a variety of machine learning methods to be applied on network data. This also explains the exploding popularity of NE methods. However, a problem with existing NE methods is that networks are fundamentally more expressive than the representations in Euclidean spaces. For example, network structural properties such as (approximate) multipartiteness, certain degree distributions, assortativity are difficult to express using Euclidean vectors. To address such limitation, we propose Conditional Network Embeddings (CNE) [5] that optimize network representations with respect to certain prior knowledge about the network. Conditioning on structural information as prior, a Euclidean representation does not need to represent the structural information but focuses on encoding the complementary information. As a result, the combination of prior and representation better captures the information conveyed by the network. Comparing to the state-of-art methods on a wide range of networks, CNE shows superior performance in link prediction and multi-label classification tasks. This proves CNE's capability of better representing network data.

Chapter 5 Learning representation with human in the loop SICA and ct-SNE both find low-dimensional representations that complement the prior information. Thus, in theory, they can be applied to find subjectively interesting low-dimensional representations iteratively. However, both methods lack a mechanism that accumulates the knowledge learned by a user along with the iterations, which prevent the user to learn from the data in a progressive manner. We developed a DR method with such mechanism and termed it Subjectively Interesting Data Exploration (SIDE) [6]. When exploring data using SIDE, the user can specify the interesting clusters she found in the data. SIDE assumes users assimilate this clustered structure and encode the position of the points in the clusters as prior. Then SIDE finds representation that complements the prior. Finally, the new representation is presented to the user, and a new iteration starts. This process can be iterated until the user runs out of time, or only noises are left in the representation. Two case studies, one controlled study on synthetic data and another on census data show SIDE is useful for iteratively and interactively discovering subjectively interesting structure from data.

Chapter 6 Interpretable representations The previous instantiations yields representations that are not easy to interpret. Namely, understanding these representations requires either the representations to be sparse or to be analyzed in a post-hoc manner, e.g., referring to external information. To compute interpretable representations, we introduce Subjectively Interesting Subgroup Discovery (SISD) [7]. SISD jointly search representations of a subset of the data objects that have

the most informative characteristics compared to the prior and also have concise descriptions. Empirical studies on four datasets show SISD can indeed find representations that are interesting and at the same time well explained by their descriptions.

1.3 Publications

This dissertation consists of key works published in the following conferences, journal articles, and preprints.

- Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *SICA: Subjectively Interesting Component Analysis*. Data Mining and Knowledge Discovery, 32(4):949–987, 2018

  This journal paper is a substantial extension of the following conference paper:

  - Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *Subjectively Interesting Component Analysis: Data Projections That Contrast with Prior Expectations*. In Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD), pages 1615–1624, 2016

- Bo Kang, Darío García García, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *Conditional t-SNE: Complementary t-SNE embeddings through factoring out prior information*. arXiv:1905.10086, 2019

- Bo Kang, Jefrey Lijffijt, and Tijl De Bie. *Conditional Network Embeddings*. In International Conference on Learning Representations (ICLR), 2019

- Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. *A Constrained Randomization Approach to Interactive Visual Data Exploration with Subjective Feedback*. IEEE Transactions on Knowledge and Data Engineering, 2019

  This journal paper is a substantial extension of the following conference paper:

  - Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. *A Tool for Subjective and Interactive Visual Data Exploration*. In The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), pages 3–7. Springer International Publishing, 2016

- Jefrey Lijffijt, Bo Kang, Wouter Duivesteijn, Kai Puolamäki, Emilia Oikarinen, and Tijl De Bie. *Subjectively Interesting Subgroup Discovery on Real-valued Targets*. arXiv:1710.04521, 2018

  This preprint is a substantial extension of the following conference paper:

  - Jefrey Lijffijt, Bo Kang, Wouter Duivesteijn, Kai Puolamäki, Emilia Oikarinen, and Tijl De Bie. *Subjectively interesting subgroup discovery on real-valued targets*. In Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE), 2018

References

[1] David J Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT Press Cambridge, MA, 2001.

[2] Jefrey Lijffijt, Panagiotis Papapetrou, Niko Vuokko, and Kai Puolamäki. *The smallest set of constraints that explains the data: a randomization approach*. Technical report, Aalto University School of Science and Technology, 2010.

[3] Tijl De Bie. *Subjective interestingness in exploratory data mining*. In International Symposium on Intelligent Data Analysis, pages 19–31. Springer, 2013.

[4] Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *SICA: Subjectively Interesting Component Analysis*. Data Mining and Knowledge Discovery, 32(4):949–987, 2018.

[5] Bo Kang, Darío García García, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *Conditional t-SNE: Complementary t-SNE embeddings through factoring out prior information*. arXiv:1905.10086, 2019.

[6] Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. *A Constrained Randomization Approach to Interactive Visual Data Exploration with Subjective Feedback*. IEEE Transactions on Knowledge and Data Engineering, 2019.

[7] Jefrey Lijffijt, Bo Kang, Wouter Duivesteijn, Kai Puolamäki, Emilia Oikarinen, and Tijl De Bie. *Subjectively interesting subgroup discovery on real-valued targets*. In Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE), 2018.

[8] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. *Conditional Network Embeddings*. In International Conference on Learning Representations (ICLR), 2019.

[9] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. *From few to many: Illumination cone models for face recognition under variable lighting and pose*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):643–660, 2001.

[10] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. *Acquiring linear subspaces for face recognition under variable lighting*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5):684–698, 2005.

[11] Karl Pearson. *LIII. On lines and planes of closest fit to systems of points in space*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11):559–572, 1901.

[12] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, 2nd edition, 2005.

[13] Laurens van der Maaten and Geoffrey Hinton. *Visualizing data using t-SNE*. Journal of machine learning research, 9(Nov):2579–2605, 2008.

[14] Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. *Subjectively Interesting Component Analysis: Data Projections That Contrast with Prior Expectations*. In Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD), pages 1615–1624, 2016.

[15] Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. *A Tool for Subjective and Interactive Visual Data Exploration*. In The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), pages 3–7. Springer International Publishing, 2016.

[16] Jefrey Lijffijt, Bo Kang, Wouter Duivesteijn, Kai Puolamäki, Emilia Oikarinen, and Tijl De Bie. *Subjectively Interesting Subgroup Discovery on Real-valued Targets*. arXiv:1710.04521, 2018.

2

Linear Representations

SICA: Subjectively Interesting Component Analysis

Abstract The information in high-dimensional datasets is often too complex for human users to perceive directly. Hence, it may be helpful to use dimensionality reduction methods to construct lower dimensional representations that can be visualized. The natural question that arises is *how do we construct a most informative low dimensional representation?* We study this question from an information-theoretic perspective and introduce a new method for linear dimensionality reduction. The obtained model that quantifies the informativeness also allows us to flexibly account for prior knowledge a user may have about the data. This enables us to provide representations that are *subjectively interesting*. We title the method Subjectively Interesting Component Analysis (SICA) and expect it is mainly useful for iterative data mining.

SICA is based on a model of a user's belief state about the data. This belief state is used to search for surprising views. The initial state is chosen by the user (it may be empty up to the data format) and is updated automatically as the analysis progresses. We study several types of prior beliefs: if a user only knows the scale of the data, SICA yields the same cost function as Principal Component Analysis (PCA), while if a user expects the data to have outliers, we obtain a variant that we term t-PCA. Finally, scientifically more interesting variants are obtained when a user has more complicated beliefs, such as knowledge about similarities between data points. The experiments suggest that SICA enables users to find subjectively

more interesting representations.

2.1 Introduction

The amount of information in high dimensional data makes it impossible to interpret such data directly. However, the data can be analyzed in a controlled manner, by revealing particular perspectives of data (lower dimensional data representations), one at a time. This is often done by means of projecting the data from the original feature space into a lower-dimensional subspace. Hence, such lower dimensional representations of a dataset are also called *data projections*, which are computed by a dimensionality reduction (DR) method.

DR methods are widely used for a number of purposes. The most prominent are data compression, feature construction, regularization in prediction problems, and exploratory data analysis. The most widely known DR technique, Principal Component Analysis (PCA) [32] is used for each of these purposes [2], since it is computationally efficient, and more importantly, because large variance is often associated with structure, while noise often has smaller variance.

Other DR methods include linear methods such as Multidimensional Scaling [25], Independent Component Analysis [17] and Canonical Correlations Analysis [16], and non-linear techniques such as ISOMAP [35], Locality Preserving Projections [15], and Laplacian-regularized models [39]. The aforementioned methods all have objective score functions whose optimization yields the lower-dimensional representation, and they do not involve human users directly. Hence, we argue that these methods may well be suitable for, e.g., compression or regularization, but not optimal for providing most insight.

In exploratory data analysis, data is often visualized along the dimensions given by a DR method. Humans are unmatched in spotting visual patterns but inefficient at crunching numbers. Hence, visualizing high dimensional data in human perceivable yet computer-generated 2D/3D space can efficiently help users to understand different perspectives of the data [33]. However, since different human operators have different prior knowledge and interests, they are unlikely to have equal interest in the same aspect of data. For instance, PCA might be applied to obtain an impression about the spread of data. But for many users, the structure in the data with largest variance may not be relevant at all.

To address this issue, Projection Pursuit (PP) [11] was proposed, which finds data projections according to a certain interestingness measure (IM), designed with specific goals. With the ability to choose between different IMs, PP balances the computational efficiency and its applicability. However, because there are many analysis tasks and users, very many IMs are required, and this has led to an explosion in the number of IMs. Hence, unlike DR used for a specific analysis task or a predictive model, it seems to be conceptually challenging to define a generic

quality metric for DR in the tasks of exploratory data analysis. This is precisely the focus of this chapter.

In this chapter we present Subjectively Interesting Component Analaysis (SICA), a dimensionality reduction method that finds *subjectively interesting* data projections. That is, projections that are aimed to be interesting to a particular user. In order to do so, SICA relies on quantifying how interesting a data projection is to the user. This quantification is based on information theory and follows the principles of FORSIED [7]. Here we discuss the central idea of FORSIED and more detail will follow in Section 2.2.

FORSIED is a data mining framework for quantifying *subjective interestingness of patterns*. The central idea is that a user's belief state about the dataset is modelled as a Maximum Entropy (MaxEnt) probability distribution over the space of possible datasets. This probability distribution is called the *background distribution* and is updated as the analysis progresses, based on user interaction and the patterns in the data provided to the user. One can quantify the probability that a given pattern is present in data that is randomly drawn from the background distribution. Clearly, the smaller this probability, the more surprising the pattern is, and the more information it conveys to the user. More specifically, in FORSIED, the self-information of the pattern, defined as minus the logarithm of that probability, is then proposed as a suitable measure of how informative it is given the belief state.

In this chapter, we define a pattern syntax called *projection patterns* for data projections that is compatible with FORSIED. By following FORSIED's principles, we can quantify the probability of a projection given the user's belief state. The lower the probability, the more surprising and interesting the pattern is, since surprising information about the data is typically what is truly interesting [14]. Because this surprisal is evaluated with respect to the belief state, SICA can evaluate the subjective interestingness of projection patterns with respect to a particular user.

Contributions. We introduce SICA, a dimensionality reduction method that tracks a user's belief about the data and presents subjectively interesting data projections to the user. To achieve this,

- we define *projection patterns*, a pattern syntax for data projections (§2.2);
- we derive a measure that quantifies the *subjective interestingness* of projection patterns (§2.2);
- we propose a method that finds the most subjectively interesting projections in terms of an optimization problem (§2.2);
- we define three types of prior beliefs a user may have knowledge about (§2.3);
- we demonstrate that with different prior belief types, SICA is able to (approximately/exactly) find the subjectively most interesting patterns. In particular, for

some prior belief types, the subjective interestingness can be efficiently optimized by solving an eigenvalue problem (§2.3);

– we present three case studies and investigate the practical advantages and drawbacks of our method, which show that it can be meaningful to account for available prior knowledge about the data (§2.4).

This chapter is an integrated and extended version of papers by De Bie et al. [9] and Kang et al. [21].

2.2 Subjectively Interesting Component Analysis

SICA allows one to find data projections that reveal unexpected variation in the data. In this section, we introduce the ingredients needed to achieve this. Namely, we (a) define an interestingness measure (IM) that quantifies the amount of information a projection conveys to a particular user, (b) following to the IM, find interesting data projections for the user. In Section 2.3, we then develop SICA further for various types of prior beliefs.

2.2.1 Notation

We use upper case bold face letters to denote matrices, lower case bold face letters for vectors, and normal lower case letters for scalars. We denote a d-dimensional real-valued dataset as $\hat{\mathbf{X}} \triangleq (\hat{\mathbf{x}}_1', \hat{\mathbf{x}}_2', \ldots, \hat{\mathbf{x}}_n')' \in \mathbb{R}^{n \times d}$, and the corresponding random variable as $\mathbf{X}$. We will refer to $\mathbb{R}^{n \times d}$, the space the data is known to belong to, as the *data space*. Dimensionality reduction methods search weight vectors $\mathbf{w} \in \mathbb{R}^d$ of unit norm (i.e. $\mathbf{w}'\mathbf{w} = 1$) onto which the data is projected by computing $\hat{\mathbf{X}}\mathbf{w}$. If k vectors are sought, they will be stored as columns of a matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$. We will denote the projections of a data set $\hat{\mathbf{X}}$ onto the column vectors of $\mathbf{W}$ as $\hat{\mathbf{\Pi}}_{\mathbf{W}} \in \mathbb{R}^{n \times k}$, or formally: $\hat{\mathbf{\Pi}}_{\mathbf{W}} \triangleq \hat{\mathbf{X}}\mathbf{W}$, and analogously for the random variable counterpart $\mathbf{\Pi}_{\mathbf{W}} \triangleq \mathbf{X}\mathbf{W}$. We will write $\mathbf{I}$ to denote the identity matrix of appropriate dimensions, and $\mathbf{1}_{n \times d}$ (or $\mathbf{1}$ for short if the dimensions are clear from the context) to denote a n-by-d matrix with all elements $\mathbf{1}_{ij} = 1$. We define the matrix interval with lower bound $\mathbf{B}$ and upper bound $\mathbf{C}$ denoted by $\mathbf{A}_{n \times m} \in [\mathbf{B}_{n \times m}, \mathbf{C}_{n \times m}]$, which indicates $a_{i,j} \in [b_{i,j}, c_{i,j}]$ for every $i, j = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, m\}$.

2.2.2 Subjective interestingness measure for projections

We now derive an IM for SICA following the framework for subjective interestingness measures (FORSIED) [7, 8]. FORSIED is a data mining framework that specifies on an abstract level how to model a user's belief state about a given dataset, and how to quantify the informativeness of patterns with respect to a particular

user. It works as follows: in order to measure the subjective interestingness of projections, SICA needs to maintain a model of the user's belief state. In addition, SICA should be able to describe data projections in a pattern syntax compatible with FORSIED. We discuss both these issues in turn below.

Modeling the user's belief state

We formalize a user's belief state as a probability distribution over the data space [7]:

Definition 6 (Background distribution)**.** *The* background distribution *is a distribution over the data space $\mathbb{R}^{n\times d}$ that represents the user's belief state: the probability it assigns to any measurable subset of $\mathbb{R}^{n\times d}$ corresponds to the probability that the user would ascribe to the data $\hat{\mathbf{X}}$ belonging to that subset. The background distribution can be represented by a probability density function $p_{\mathbf{X}} : \mathbb{R}^{n\times d} \to \mathbb{R}^{+}$.*

For brevity, and slightly abusively, we will often refer to the density function $p_{\mathbf{X}}$ as the background distribution.

Of course, the background distribution is typically not known to the data mining system. Thus, it has to be inferred from limited information provided by the user. De Bie [8] proposed an intuitive while mathematically rigorous language a user can employ to express certain beliefs about the data. The language assumes that important characteristics of the data can be quantified by means of statistics $f : \mathbb{R}^{n\times d} \to \mathbb{R}$. Using such statistics, the user can express their beliefs by declaring which value they expect f to have when evaluated on the data. Mathematically, this then becomes a constraint on the background distribution $p_{\mathbf{X}}$.

Definition 7 (Prior belief constraints)**.** *When the user expresses a prior belief by declaring that they expect a specified statistic f to be equal to a specified value $\hat{m} \in \mathbb{R}$, they are declaring that their background distribution $p_{\mathbf{X}}$ satisfies the following prior belief constraint:*

$$\mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[f(\mathbf{X})\right] = \hat{m}. \tag{2.1}$$

Except in degenerate cases, such constraints will not uniquely determine $p_{\mathbf{X}}$, such that an additional criterion is required to decide which one to use. Amongst those satisfying the prior belief constraints, the distribution with the maximum entropy (MaxEnt) is an attractive choice, given its unbiasedness and robustness. Further, as the resulting distribution belongs to the *exponential family*, its inference is well understood and often computationally tractable.

Formally, a user's background distribution can thus be obtained by solving the

following constrained entropy maximization problem:

$$\begin{aligned} \underset{p_{\mathbf{X}}(\mathbf{X})\geq 0}{\text{argmax}} \quad & -\int p_{\mathbf{X}}(\mathbf{X}) \log\left(p_{\mathbf{X}}(\mathbf{X})\right) d\mathbf{X} \\ \text{s.t.} \quad & \int p_{\mathbf{X}}(\mathbf{X}) f_i(\mathbf{X}) d\mathbf{X} = \hat{m}_i, \quad \forall i, \\ & \int p_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} = 1. \end{aligned} \tag{2.2}$$

As we will show in Section 2.3, by solving optimization problem (2.2) with different types of statistics f_i, one can model a wide variety of prior beliefs, and hence obtain very different types of background distributions.

Projection patterns: a pattern syntax for data projections

In FORSIED[1], a *pattern* is defined as any information that restricts the set of possible values the data may have. For example, if the user is shown a scatter plot of the projections in $\hat{\mathbf{\Pi}}_{\mathbf{W}}$, the user will from then on know that $\hat{\mathbf{X}}\mathbf{W}$ is equal to $\hat{\mathbf{\Pi}}_{\mathbf{W}}$ (up to the resolution of the plot), which clearly constrains the set of possible values of the data to a subset of $\mathbb{R}^{n\times d}$.

One could thus be tempted to define a *projection pattern* as a statement of the kind $\hat{\mathbf{X}}\mathbf{W} = \hat{\mathbf{\Pi}}_{\mathbf{W}}$. This would tell the user that the projections of the data $\hat{\mathbf{X}}$ onto the columns of $\mathbf{W}$ are found to be equal to the columns of $\hat{\mathbf{\Pi}}_{\mathbf{W}}$.

However, real-valued data projections are often conveyed visually to a user, and in any case with finite accuracy, e.g. by means of a scatter plot. Because human eyes as well as the visualization devices (e.g., monitor, projector, and paper) have finite resolution, the precise value of the projected data can only be determined up to a certain resolution-dependent uncertainty $2\Delta\mathbf{1} \in \mathbb{R}^{n\times k}$. With these considerations[2], we formally define the syntax of a projection pattern as follows:

Definition 8 (Projection pattern)**.** *Let $\mathbf{W} \in \mathbb{R}^{d\times k}$ be a projection matrix, and let $\hat{\mathbf{\Pi}}_{\mathbf{W}}$ be the value of the projections of the data $\hat{\mathbf{X}} \in \mathbb{R}^{n\times d}$ onto the columns of $\mathbf{W}$. Then a* projection pattern *is a statement of the form:*

$$\hat{\mathbf{X}}\mathbf{W} \in \left[\hat{\mathbf{\Pi}}_{\mathbf{W}} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_{\mathbf{W}} + \Delta\mathbf{1}\right]. \tag{2.3}$$

[1] As well as in the only other framework for interactive data mining, CORAND [27]. By a framework for interactive data mining we mean a generic method that can be used to design specific data mining methods that take into account results previously shown to the user or other prior knowledge about the data. Such a framework would specify certain aspects of the method while other aspects are left open and only a guideline is provided on how to fill in that part. E.g., FORSIED specifies to define the background model as a MaxEnt distribution and the objective to maximize is the Subjective Interestingness. CORAND mandates another objective score (to maximize the p-value of the data) and the form of the background distribution is left open; it may be anything. As far as we know, there are no other works published with a similar spirit.

[2] To simplify our notation, we assume the resolution parameter being the same through all dimensions. It is indeed an interesting direction to further develop SICA for the resolution varying in different dimensions.

Thus, the projection pattern specifies, up to an accuracy of 2Δ, the value of the projections of the data onto the columns of the projection matrix $\mathbf{W}$.

Subjective interestingness of projections

Relying on the background distribution, we can now quantify the *subjective interestingness* of a projection pattern:

Definition 9 (Subjective interestingness of projection pattern)**.** *The* subjective interestingness *(SI) of a projection pattern is defined as the negative log probability of the pattern under the background distribution.*[3] *For a projection pattern with projection matrix* $\mathbf{W}$ *and observed projections* $\hat{\mathbf{\Pi}}_\mathbf{W}$*, it is equal to:*

$$SI(\mathbf{W}, \hat{\mathbf{\Pi}}_\mathbf{W}) = -\log\left(Pr\left(\mathbf{XW} \in \left[\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}\right]\right)\right). \tag{2.4}$$

The probability of a pattern can be computed by integrating the background distribution over all $\mathbf{X}$ for which $\mathbf{XW} \in \left[\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}\right]$:

$$\Pr\left(\mathbf{XW} \in \left[\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}\right]\right) = \int_{\mathbf{X}:\mathbf{XW}\in\left[\hat{\mathbf{\Pi}}_\mathbf{W}-\Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W}+\Delta\mathbf{1}\right]} p_\mathbf{X}(\mathbf{X})d\mathbf{X}. \tag{2.5}$$

This can be expressed more conveniently in terms of the marginal density function $p_{\mathbf{\Pi}_\mathbf{X}}$ for the projection $\mathbf{\Pi}_\mathbf{W} \triangleq \mathbf{XW}$ of the data:

$$\Pr\left(\mathbf{XW} \in \left[\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}\right]\right) = \int_{\hat{\mathbf{\Pi}}_\mathbf{W}-\Delta\mathbf{1}}^{\hat{\mathbf{\Pi}}_\mathbf{W}+\Delta\mathbf{1}} p_{\mathbf{\Pi}_\mathbf{W}}(\mathbf{\Pi}_\mathbf{W})d\mathbf{\Pi}_\mathbf{W}. \tag{2.6}$$

For sufficiently small Δ, we approximate the integral in Equation (2.6) as the value of the integrand in the middle of the integration domain times the integration domain's volume:[4]

$$\Pr\left(\mathbf{XW} \in \left[\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}\right]\right) \approx p_{\mathbf{\Pi}_\mathbf{W}}(\hat{\mathbf{\Pi}}_\mathbf{W})(2\Delta)^{nk}. \tag{2.7}$$

Then Definition 9 can be reformulated into:

$$\mathrm{SI}(\mathbf{W}, \hat{\mathbf{\Pi}}_\mathbf{W}) \approx -\log\left(p_{\mathbf{\Pi}_\mathbf{W}}(\hat{\mathbf{\Pi}}_\mathbf{W})\right) - nk\log\left(2\Delta\right). \tag{2.8}$$

[3] In FORSIED, the subjective interestingness of a pattern is generally defined by a trade off between the information content (i.e., negative probability) of the pattern and the descriptional complexity (i.e., the amount of effort needed to assimilate the pattern). Here we assume all projections of the same dataset have the same descriptional complexity. As a result, the descriptional complexity can be ignored from the definition of SI.

[4] The tightness of this approximation for the cases in Section. 2.3.1 and 2.3.2 will be investigated in detail in Appendices A and B.

Thus, to compute the interestingness of a projection pattern it is sufficient to know the marginal density function $p_{\mathbf{\Pi_W}}$. We will compute this marginal density function in Section 2.3 for a number of background distributions.

2.2.3 Searching subjectively interesting projection patterns

Searching for subjectively interesting projection patterns amounts to finding a set of weight vectors $\mathbf{W} \in \mathbb{R}^{d \times k}$ that yield projections with the largest SI value. The resulting weight vectors $\mathbf{W}$ linearly transform the original d features of the data $\hat{\mathbf{X}}$ into k features. Similar to the definition of the (principal) components in PCA, we refer to those k transformed features as the *subjectively interesting components* (SICs) of the data $\hat{\mathbf{X}}$.

The projection matrix $\mathbf{W}$ that corresponds to the SICs of data $\hat{\mathbf{X}}$ under background distribution $p_{\mathbf{X}}$ can thus be obtained by finding the $\mathbf{W}$ maximizing $\text{SI}(\mathbf{W}, \hat{\mathbf{\Pi}}_{\mathbf{W}})$. As $\hat{\mathbf{X}}\mathbf{W}$ must represent projections, all weight vectors in the columns $\mathbf{W}$ must have unit norm. Additionally, to ensure non-redundancy of the different projections, we will require the weight vectors to be orthogonal, such that $\mathbf{W}'\mathbf{W} = \mathbf{I}$. Substituting $\hat{\mathbf{X}}\mathbf{W}$ for $\hat{\mathbf{\Pi}}_{\mathbf{W}}$ to make the dependencies on the data $\hat{\mathbf{X}}$ and the projection matrix $\mathbf{W}$ explicit, the optimization problem to be solved is thus:

$$\begin{aligned} \underset{\mathbf{W} \in \mathbb{R}^{d \times k}}{\text{argmax}} \quad & -\log\left(p_{\mathbf{\Pi_W}}\left(\hat{\mathbf{X}}\mathbf{W}\right)\right), \\ \text{s.t.} \quad & \mathbf{W}'\mathbf{W} = \mathbf{I}. \end{aligned} \tag{2.9}$$

Note that this problem is independent of the resolution parameter Δ. In other words, as soon as Δ is small enough for Equation 2.7 to hold to a sufficient approximation, its precise value is irrelevant to the problem.

It is this problem that we will be solving in Section 2.3 for a number of different types of background distributions.

2.3 SICA with different types of prior beliefs

In this section, we develop SICA further for three different types of prior beliefs. Each is discussed in a separate subsection. In Section 2.3.4, we discuss how SICA can in principle be used for other prior belief types as well, while also highlighting the difficulties in tackling other prior belief types that may limit the applicability of SICA in practice.

2.3.1 Scale of the data as prior belief

When the user only has a prior belief about the average variance of a dataset, SICA will aim to find projections with large variances. As we will show here, SICA with such prior is equivalent to PCA.

Prior belief With a given dataset, the user might have certain prior knowledge about the scale of a dataset. She might believe that the average scale of the data points, quantified by their squared norms, is some constant $\sigma^2 d$ and have no other knowledge. This can be formalized in a constraint of the form of Equation (2.1):

$$\mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[\frac{1}{n}\operatorname{Tr}\left(\mathbf{X}\mathbf{X}'\right)\right] = \sigma^2 d. \tag{2.10}$$

The corresponding statistic f of prior (2.10) is $f(\mathbf{X}) = \frac{1}{n}\operatorname{Tr}\left(\mathbf{X}\mathbf{X}'\right)$.

Background distribution Inserting (2.10) into (2.2), we obtain the following MaxEnt problem:

$$\begin{aligned} \underset{p_{\mathbf{X}}(\mathbf{X})\geq 0}{\operatorname{argmax}} \quad & -\int p_{\mathbf{X}}(\mathbf{X})\log\left(p_{\mathbf{X}}(\mathbf{X})\right)d\mathbf{X} \\ \text{s.t.} \quad & \int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{n}\operatorname{Tr}\left(\mathbf{X}\mathbf{X}'\right)d\mathbf{X} = \sigma^2 d, \\ & \int p_{\mathbf{X}}(\mathbf{X})d\mathbf{X} = 1. \end{aligned} \tag{2.11}$$

The optimal background distribution $p_{\mathbf{X}}$ is a product distribution of identical multivariate Normal distributions with mean $\mathbf{0}$ and covariance matrix $\sigma^2\mathbf{I}$. This is summarized in the following theorem:

Theorem 1. *Given prior belief (2.10), the MaxEnt background distribution is*

$$p_{\mathbf{X}}(\mathbf{X}) = \prod_{i=1}^{n} p_{\mathbf{x}}(\mathbf{x}_i), \tag{2.12}$$

where $p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}}\exp\left(-\frac{\mathbf{x}'\mathbf{x}}{2\sigma^2}\right)$ *is multivariate Normal density function with mean zero and covariance matrix* $\sigma^2\mathbf{I}$*.*

Proof. Through application of the Lagrange multiplier method, we find the Lagrangian for Problem (2.11):

$$\begin{aligned} \mathcal{L}\left(p_{\mathbf{X}},\lambda,\mu\right) = & -\int p_{\mathbf{X}}\left(\mathbf{X}\right)\log\left(p_{\mathbf{X}}(\mathbf{X})\right)d\mathbf{X} - \lambda\left(\int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{n}\operatorname{Tr}(\mathbf{X}\mathbf{X}')d\mathbf{X} - \sigma^2 d\right) \\ & + \mu\left(\int p_{\mathbf{X}}(\mathbf{X})d\mathbf{X} - 1\right). \end{aligned} \tag{2.13}$$

Then, finding the function $p_{\mathbf{X}}$ that maximize this functional amounts to solving a *Euler-Lagrange equation* with Lagrangian $\mathcal{L}$ in form (2.13):

$$\frac{\partial\mathcal{L}}{\partial p_{\mathbf{X}}} - \frac{d}{d\mathbf{X}}\frac{\partial\mathcal{L}}{\partial p'_{\mathbf{X}}} = \frac{\partial\mathcal{L}}{\partial p_{\mathbf{X}}} + 0 = 0. \tag{2.14}$$

Hence, we compute the functional derivative of the Lagrangian with respect to $p_{\mathbf{X}}$ at $\mathbf{X}$:

$$\frac{\partial}{\partial p_{\mathbf{X}}(\mathbf{X})}\mathcal{L} = -1 - \log\left(p_{\mathbf{X}}(\mathbf{X})\right) + \frac{\lambda}{n}\operatorname{Tr}(\mathbf{X}\mathbf{X}') + \mu. \tag{2.15}$$

By equating the partial derivative to zero, we obtain an expression of $p_{\mathbf{X}}$ parametrized by λ and μ:

$$\begin{aligned} p_{\mathbf{X}}(\mathbf{X}) &= \exp\left(\mu - 1 + \frac{\lambda}{n}\operatorname{Tr}(\mathbf{X}\mathbf{X}')\right) \\ &= \exp(\mu-1)\exp\left(\frac{\lambda}{n}\sum_{i=1}^{n}\mathbf{x}_i'\mathbf{x}_i\right) \\ &= \prod_{i=1}^{n}\frac{1}{Z}\exp\left(\frac{\lambda}{n}\mathbf{x}_i'\mathbf{x}_i\right), \end{aligned} \tag{2.16}$$

where $Z = \exp^{-1}\left(\frac{\mu-1}{n}\right)$. In order to find optimal solutions for λ and μ, observe that $p_{\mathbf{X}}(\mathbf{X})$ in Equation (2.16) is the product of n identical multivariate Normal distributions, one for each data point $\mathbf{x}_i$, with zero mean and -$\frac{n}{2\lambda}\mathbf{I}$ as covariance matrix. As the expected two-norm squared of a multivariate Normal random vector with zero mean is equal to the trace of its covariance matrix, and as the expectation of the average two-norm squared of the identically distributed data points is constrained to be $\sigma^2 d$, this means that $\sigma^2 d = -\frac{dn}{2\lambda}$, such that $\lambda = -\frac{n}{2\sigma^2}$.

Therefore, the MaxEnt background distribution is an independent multivariate normal distribution, where each independent random variable has zero mean, and covariance matrix $\sigma^2\mathbf{I}$, i.e., $\mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$. □

Subjectively interesting patterns Now we can search for subjectively interesting patterns by solving problem (2.9). This requires to first compute distribution $p_{\mathbf{\Pi_W}}$ as the marginal of the background distribution (2.16).

Given a normal random vector $\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$, a projection onto weight vectors $\mathbf{W}$ with $\mathbf{W}'\mathbf{W} = \mathbf{I}$ is also normal: $\mathbf{x}'\mathbf{W} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$. Thus, the marginal density function distribution $p_{\mathbf{\Pi_W}}$ for the projection $\mathbf{\Pi_W} = \mathbf{X}\mathbf{W}$ is given by:

$$\begin{aligned} p_{\mathbf{\Pi_W}}(\mathbf{X}\mathbf{W}) &= \prod_{i=1}^{n}\prod_{j=1}^{k}\frac{1}{\sqrt{(2\pi\sigma^2)}}\exp\left(-\frac{(\mathbf{w}_j'\mathbf{x}_i)^2}{2\sigma^2}\right) \\ &= \prod_{i=1}^{n}\frac{1}{\sqrt{(2\pi\sigma^2)^k}}\exp\left(-\frac{\mathbf{x}_i'\mathbf{W}\mathbf{W}'\mathbf{x}_i}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^{nk}}}\exp\left(-\frac{1}{2\sigma^2}\operatorname{Tr}\left[\mathbf{W}'\mathbf{X}'\mathbf{X}\mathbf{W}\right]\right). \end{aligned} \tag{2.17}$$

Given density function (2.17), we can now use (2.9) to find projection patterns ($\mathbf{X}\mathbf{W} \in [\hat{\mathbf{\Pi}}_{\mathbf{W}} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_{\mathbf{W}} + \Delta\mathbf{1}]$) that are subjectively interesting. This is only

true if the approximation (2.7) is good enough. In Appendix. A, we show this is indeed the case. Thus, substituting the marginal distribution (2.17) into the objective function of problem (2.9) gives:

$$-\log\left(p_{\mathbf{\Pi_W}}(\hat{\mathbf{\Pi}}_{\mathbf{W}})\right) = \frac{nk}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\operatorname{Tr}\left[\mathbf{W}'\hat{\mathbf{X}}'\hat{\mathbf{X}}\mathbf{W}\right]. \tag{2.18}$$

Ignoring the first constant term and constant factor $\frac{1}{2\sigma^2}$, the optimization problem (2.9) is equivalent to:

$$\begin{aligned} \max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad & \operatorname{Tr}\left[\mathbf{W}'\hat{\mathbf{X}}'\hat{\mathbf{X}}\mathbf{W}\right] \\ \text{s.t.} \quad & \mathbf{W}'\mathbf{W} = \mathbf{I}. \end{aligned} \tag{2.19}$$

This is equivalent (up to rotation) to the problem of finding the k dominant principal component of $\mathbf{X}$ in classical PCA[5].

2.3.2 t-PCA: magnitude of spread as prior belief

In contrast to believing the data has a certain scale, a user might expect that the data has certain magnitude of spread. In this subsection, we show that with such prior expectation, SICA yields an alternative result, that turns out to be more robust against outliers.

Prior belief Denote γ to be the parameter that expresses the user's belief about the magnitude of spread of the data. The user's expectation about the magnitude of spread to be some value a is then defined by:

$$\mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n}\log(1+\frac{1}{\gamma}\mathbf{x}_i'\mathbf{x}_i)\right] = a. \tag{2.20}$$

If the user is expecting outliers in the data, she may specify γ to be small. This will up-weight the outliers (who have relatively large 2-norms) such that they contribute more to the expectation. In contrast, by setting a larger γ, the expectation is focused more on the bulk of the points.

[5]In this chapter, by performing PCA, we mean the data $\mathbf{X}$ is first centered ($\mathbf{X}_c = \mathbf{X} - \frac{1}{n}\mathbf{1}_{n\times 1}\mathbf{1}_{n\times 1}'\mathbf{X}$), then the eigenvectors of matrix $\mathbf{X}'\mathbf{X}$ are computed and sorted in descending order according to the absolute value of the eigenvalues. After sorting, the eigenvectors of $\mathbf{X}'\mathbf{X}$ with largest absolute eigenvalues correspond to the top principal components.

Background distribution with the prior belief (2.20) we need to solve the following optimization problem to obtain the MaxEnt background distribution:

$$\begin{aligned} \underset{p_{\mathbf{X}}(\mathbf{X})\geq 0}{\text{argmax}} \quad & -\int p_{\mathbf{X}}(\mathbf{X}) \log\left(p_{\mathbf{X}}(\mathbf{X})\right) d\mathbf{X} \\ \text{s.t.} \quad & \int p_{\mathbf{X}}(\mathbf{X}) \cdot \frac{1}{n} \sum_{i=1}^{n} \log(1 + \frac{1}{\gamma} \mathbf{x}_i' \mathbf{x}_i) d\mathbf{X} = a, \\ & \int p_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} = 1. \end{aligned} \tag{2.21}$$

Relying on the result by Zografos [40], we find that the optimal solution is a product of independent multivariate standard t-distributions. Here, we denote a digamma function as φ, and introduce the function $\kappa(\nu) = \varphi(\frac{\nu+d}{2}) - \varphi(\frac{\nu}{2})$, where d is the dimension of data $\hat{\mathbf{X}}$. In the sequel, the value $\nu = \kappa^{-1}(a)$ will be used, i.e., ν depends on the expected magnitude of spread a. The background distribution with prior belief (2.20) is then defined as:

Theorem 2. *Given prior belief (2.20), the MaxEnt background distribution is*

$$p_{\mathbf{X}}(\mathbf{X}) = \prod_{i=1}^{n} p(\mathbf{x}_i) \tag{2.22}$$

where $p(\mathbf{x})$ is the density function of a multivariate standard t-distribution with form:

$$p(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{(\pi\rho)^{d/2}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1 + \frac{1}{\rho}\mathbf{x}'\mathbf{x}\right)^{\frac{\nu+d}{2}}}. \tag{2.23}$$

where $\rho = \gamma\nu$, the correlation matrix is a d-by-d identity matrix $\mathbf{I}$.

Proof. We restate the Theorem 2.1 and the derivation of equation 2.12 from the paper by Zografos [40]. From this the proof immediately follows.

Theorem 2.1 in [40] sates that for MaxEnt problem:

$$\begin{aligned} \underset{p_{\mathbf{x}}(\mathbf{x})\geq 0}{\text{argmax}} \quad & -\int p_{\mathbf{x}}(\mathbf{x}) \log\left(p_{\mathbf{x}}(\mathbf{x})\right) d\mathbf{x} \\ \text{s.t.} \quad & \int p_{\mathbf{x}}(\mathbf{x}) \log\left(1 + (\mathbf{x}-\mu)'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mu)\right) d\mathbf{x} = \varphi(m) - \varphi(m - \frac{d}{2}) \\ & \int p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = 1. \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^d$,$m > (d+2)/2$, $\mathbb{E}(\mathbf{x}) = \mu$, $\text{Cov}(\mathbf{x}) = 1/(2m-d-2)\boldsymbol{\Sigma}$. The solution of this problem is a special case of *Pearson's Type VII* multivariate

distribution with density:

$$p(\mathbf{x}) = \frac{\Gamma(m)}{\pi^{d/2}\Gamma(m-d/2)}|\mathbf{\Sigma}|^{-1/2}[1+(\mathbf{x}-\mu)'\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)]^{-m}.$$

That is, the multivariate t-distribution with ν degrees of freedom, $\nu = \kappa^{-1}(a) > 0$, $\mu = \mathbf{0}$, and $\mathbf{\Sigma} = \gamma\mathbf{I}$ can be obtained from Pearson's Type VII distribution using transformation $\mathbf{z} = \sqrt{\nu}\mathbf{x} + (1-\sqrt{\nu})\mu$ and taking $m = (\nu+d)/2$:

$$\begin{aligned} p(\mathbf{z}) &= \frac{\Gamma((\nu+d)/2)}{(\pi\nu\gamma)^{d/2}\Gamma(\nu/2)}[1+\frac{1}{\nu\gamma}\mathbf{z}'\mathbf{z}]^{-(\nu+d)/2} \\ &= \frac{\Gamma((\nu+d)/2)}{(\pi\rho)^{d/2}\Gamma(\nu/2)}[1+\frac{1}{\rho}\mathbf{z}'\mathbf{z}]^{-(\nu+d)/2} \end{aligned}$$

By setting $\rho = \gamma\nu$, only one parameter needs to be tuned. □

Remark 1. *Note that for $\rho, \nu \to \infty$, $\frac{\rho}{\nu} \to \sigma^2$ this density function tends to the multivariate Normal density function with mean $\mathbf{0}$ and covariance $\sigma^2\mathbf{I}$. For $\rho = \nu = 1$ it is a multivariate standard Cauchy distribution, which is so heavy-tailed that its mean is undefined and its second moment is infinitely large. Thus, this type of prior belief can model the expectation of outliers to varying degrees.*

Given the reliance on a multivariate t-distribution as the background distribution, we will refer to this model as *t-PCA*.

Subjectively interesting patterns According to Kotz and Nadarajah [24], the marginals of a t-distribution with given correlation matrix are again a t-distribution with the same number of degrees of freedom. Each marginal is obtained by selecting the relevant part of the correlation matrix. This means that the marginal density function for projection $\mathbf{\Pi}_\mathbf{W} = \mathbf{XW}$ onto k weight vectors $\mathbf{W}$ with $\mathbf{W}'\mathbf{W} = \mathbf{I}$ is

$$p_{\mathbf{\Pi}_\mathbf{W}}(\mathbf{\Pi}_\mathbf{W}) = \prod_{i=1}^{n} \frac{\Gamma\left(\frac{\nu+k}{2}\right)}{(\pi\rho)^{k/2}\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\left(1+\frac{1}{\rho}\mathbf{x}_i'\mathbf{W}\mathbf{W}'\mathbf{x}_i\right)^{\frac{\nu+k}{2}}}. \tag{2.24}$$

Given density function (2.24), we can now use (2.9) to find projection patterns ($\mathbf{XW} \in [\hat{\mathbf{\Pi}}_\mathbf{W} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_\mathbf{W} + \Delta\mathbf{1}]$) that are subjectively interesting. This is only true if the approximation (2.7) is good enough. In Appendix. B, we show this is indeed the case. Thus, substituting the marginal distribution (2.24) into the objective function of problem (2.9) by gives:

$$-\log\left(p_{\mathbf{\Pi}_\mathbf{W}}(\hat{\mathbf{\Pi}}_\mathbf{W})\right) = \frac{\nu+k}{2}\sum_{i=1}^{n}\log\left(1+\frac{1}{\rho}\hat{\mathbf{x}}_i'\mathbf{W}\mathbf{W}'\hat{\mathbf{x}}_i\right) + \text{a constant.} \tag{2.25}$$

Ignoring some constant factors and terms, searching for the subjectively most interesting pattern is thus equivalent to solve:

$$\begin{aligned} \max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad & \sum_{i=1}^{n} \log(\rho + \hat{\mathbf{x}}_i'\mathbf{W}\mathbf{W}'\hat{\mathbf{x}}_i) \\ \text{s.t.} \quad & \mathbf{W}'\mathbf{W} = \mathbf{I}. \end{aligned} \tag{2.26}$$

Remark 2. *By varying ρ, SICA interpolates between maximizing the arithmetic mean, like PCA does, and maximizing the geometric mean of the square of the data projections, which is more robust against outliers. To be precise, for $\rho = 0$ the objective function (2.26) is monotonically related to the geometric mean of the squared norm of data projections $\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2$:*

$$\exp\left[\frac{1}{n}\sum_{i=1}^{n}\log(\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2)\right] = \left(\prod_{i=1}^{n}(\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2)\right)^{\frac{1}{n}}. \tag{2.27}$$

On the other hand, for $\rho \to \infty$, the objective function (2.26) is monotonically related to arithmetic mean,

$$\begin{aligned} &\lim_{\rho\to\infty}\frac{\rho}{n}\sum_{i=1}^{n}\log\left(\rho + \|\hat{\mathbf{x}}_i'\mathbf{W}\|^2\right) - \rho\log(\rho) \\ &\quad= \lim_{\rho\to\infty}\frac{1}{n}\sum_{i=1}^{n}\rho\log\left(1 + \frac{\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2}{\rho}\right) + \rho\log(\rho) - \rho\log(\rho) \\ &\quad= \lim_{\rho\to\infty}\frac{1}{n}\sum_{i=1}^{n}\log\left[\left(1 + \frac{\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2}{\rho}\right)^{\rho}\right] \\ &\quad= \frac{1}{n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i'\mathbf{W}\|^2, \end{aligned}$$

That is, for sufficiently large ρ the objective function is equivalent to the arithmetic mean, up to factor ρ and additive constant $-\rho\log(\rho)$.

To get some insight into the computational complexity of problem (2.26), let us consider the one dimensional case where we search for weight vector $\mathbf{w} \in \mathbb{R}^{d\times 1}$. Clearly, the larger $\mathbf{w}'\mathbf{w}$, the larger the objective, so the constraint can be relaxed to $\mathbf{w}'\mathbf{w} \leq 1$. Hence the feasible set of $\mathbf{w}$ is convex. Denote $s_i = \text{sign}\,(\hat{\mathbf{x}}_i\mathbf{w})$ as the sign of the scale value $\hat{\mathbf{x}}_i\mathbf{w}$. For $\rho = 0$, the objective can be re-written as $\sum_{i=1}^{n}\log\left((\hat{\mathbf{x}}_i'\mathbf{w})^2\right) = \sum_{i=1}^{n}\log\det\begin{pmatrix} s_i\hat{\mathbf{x}}_i'\mathbf{w} & 0 \\ 0 & s_i\hat{\mathbf{x}}_i'\mathbf{w}\end{pmatrix}$, which is a sum of log determinant functions of the parameters $\mathbf{w}$. Hence the objective function is concave. Based on this observation, a possible solution strategy is to enumerate all possible sign vector $\mathbf{s} = \text{sign}(\hat{\mathbf{X}}\mathbf{w}_i)$, and first find an optimal $\mathbf{w}$ for each of

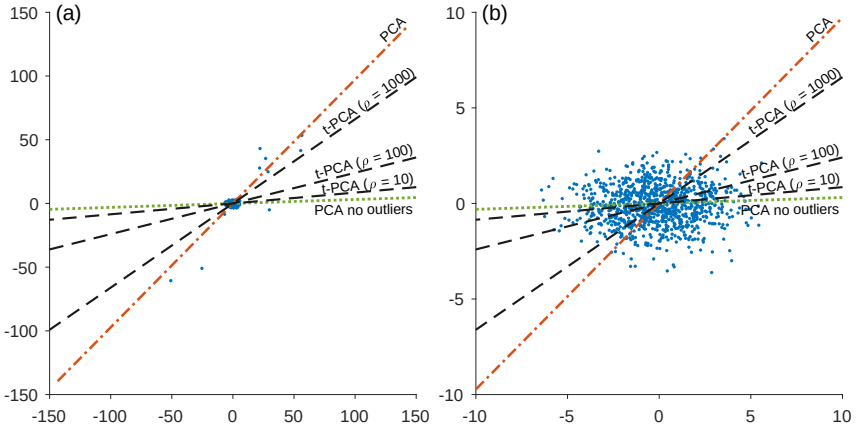

Figure 2.1: Synthetic data (§2.3.2) visualized with weight vectors of PCA (red dash-dotted line), SICA (black dashed lines, ρ = 10, 100, 1000), and PCA fitted excluding the outliers (green dotted line). (a) data visualized including outliers. (b) data visualized excluding outliers.

those convex problems. The global optimal solution can then be found over all enumerations. Although this is not a proof of the complexity of the problem, and the existence of an efficient algorithm cannot be ruled out, it shows that at least a naive algorithm needs an exponential time in $\mathcal{O}((n-1)^{d-1})$.

We solve the problem (2.26) by approximation. Observe that the orthonormality constraint posed on $\mathbf{W}$ leads to problem (2.26) being a Stiefel manifold [30] optimization problem. This can be addressed fairly efficiently with a standard tool box. We use the Manopt toolbox [4] to obtain an approximate solution.

Remark 3. *For the parameter ρ in constraint (2.20), a user can set it freely according to her prior belief. Namely, if the user feels confident about the average squared norm of the data points, a large ρ should be used, but if the user feels confident only about the* order of magnitude *of the norms of the data points, a small ρ should be used. The next example illustrates the effect of different choices for ρ.*

Example. As an illustrative example, we compare PCA and SICA on synthetic data. We generated a dataset consisting of two populations with different covariance structures: 1000 data points sampled from $\mathcal{N}\left(\mathbf{0}, \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}\right)$, and 10 'outliers' from $\mathcal{N}\left(\mathbf{0}, \begin{pmatrix} 16 & 12 \\ 12 & 13 \end{pmatrix}\right)$, i.e., $\hat{\mathbf{X}} \in \mathbb{R}^{1010\times 2}$. After sampling, the data is centered. Figure 2.1a shows the first components resulting from PCA, SICA, and PCA had there been no outliers. The PCA result is determined primarily by the outliers. The right plot (Figure 2.1b) shows the components on top of a scatter

plot without the 10 outliers, illustrating that SICA is hardly affected by outliers. That is, the lower ρ the more the user's belief allows for the existence of outliers, hence SICA shows the projection with fewer outliers as additional information. By varying the ρ parameter ($\rho = 10, 100, 1000$), the resulted projection interpolates between PCA and PCA on data with outliers removed.

2.3.3 Pairwise data point similarities as prior beliefs

In SICA, users may specify not only global characteristics of the data, such as the expected magnitude of spread, but they can also express expectations about local characteristics, such as similarities between data points.

Prior belief. Assume the user believes that a data point is similar to another point or group of points. She may then want to discover other structure within the data, in addition to the known similarities. Generally speaking, the most interesting/-surprising information would be a pattern that *contrasts* with the known similarities. For example, consider a user interested in social network analysis, and more specifically, interested in finding social groups that share certain properties. Suppose the user has already studied the network structure to some degree, and now it would be more interesting for her to learn about other properties shared by different social groups; other as in properties not aligned with the network structure.

SICA allows the user to encode their beliefs as follows. The data points are represented as nodes in a graph $G = (\mathbf{X}, E)$, and the user can connect all pairs of points that she expects to be similar with an edge. In this way, the user's prior belief regarding similarities among data points can be measured as the average pairwise Euclidean distance of connected nodes in graph G:

$$\mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[\frac{1}{|E|}\sum_{(i,j)\in E}||\mathbf{x}_i - \mathbf{x}_j||^2\right] = b, \tag{2.28}$$

where b is some constant. Constraint (2.28) on its own still has ambiguity, as a small b can be due to a belief that connected data points in G are close together, but also due to a belief that the scale of the data is simply small. Thus, to forestall the second interpretation, another constraint needs to be imposed which fixes the expected scale of the data:

$$\mathbb{E}_{\mathbf{X}\sim p_{\mathbf{X}}}\left[\frac{1}{n}\sum_{i=1}^{n}||\mathbf{x}_i||^2\right] = c. \tag{2.29}$$

Background distribution. To obtain the background distribution, the following

MaxEnt problem needs to be solved:

$$\begin{aligned} \underset{p_{\mathbf{X}}(\mathbf{X})\geq 0}{\text{argmax}} \quad & -\int p_{\mathbf{X}}\left(\mathbf{X}\right)\log\left(p_{\mathbf{X}}(\mathbf{X})\right)d\mathbf{X} \\ \text{s.t.} \quad & \int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{|E|}\sum_{(i,j)\in E}||\mathbf{x}_i-\mathbf{x}_j||^2 d\mathbf{X}=b, \\ & \int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{n}\sum_{i=1}^{n}||\mathbf{x}_i||^2 d\mathbf{X}=c, \\ & \int p_{\mathbf{X}}(\mathbf{X})d\mathbf{X}=1. \end{aligned} \tag{2.30}$$

Denote $\mathbf{I}$ as identity matrix and $\mathbf{L}$ as the Laplacian of the graph G defined as $\mathbf{L}=\mathbf{D}-\mathbf{A}$, with $\mathbf{A}$ the adjacency matrix of graph and $\mathbf{D}$ the diagonal matrix with the degrees of nodes on its diagonal. We now show that the solution of problem (2.30) is a matrix normal distribution $\mathcal{MN}_{n\times d}\left(\mathbf{M},\mathbf{\Sigma},\mathbf{\Phi}\right)$, specifically:

Theorem 3. *The optimal solution of problem (2.30) is given by a matrix normal distribution:*

$$\mathbf{X}\sim\mathcal{MN}_{n\times d}\left(\mathbf{0},\left(2\left[\frac{\lambda_1}{|E|}\mathbf{L}+\frac{\lambda_2}{n}\mathbf{I}_n\right]\right)^{-1},\mathbf{I}_d\right), \tag{2.31}$$

namely,

$$p_{\mathbf{X}}(\mathbf{X})=\frac{1}{Z}\exp\left\{\mathrm{Tr}\left(-\mathbf{X}'\left[\frac{\lambda_1}{|E|}\mathbf{L}+\frac{\lambda_2}{n}\mathbf{I}_n\right]\mathbf{X}\right)\right\}, \tag{2.32}$$

with partition function $Z=(2\pi)^{\frac{nd}{2}}\left|2\left[\frac{\lambda_1}{|E|}\mathbf{L}+\frac{\lambda_2}{n}\mathbf{I}_n\right]\right|^{\frac{d}{2}}$.

The proof, provided below, makes clear that the values of λ_1 and λ_2 depend on the values of b and c in the constraints, and can be found by solving a very simple convex optimization problem:

Proof. The Lagrangian for (2.30) is:

$$\begin{aligned} \mathcal{L}(p_{\mathbf{X}},\lambda,\mu)=&-\int p_{\mathbf{X}}\left(\mathbf{X}\right)\log\left(p_{\mathbf{X}}(\mathbf{X})\right)d\mathbf{X} \\ &-\lambda_1\left(\int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{|E|}\sum_{(i,j)\in E}||\mathbf{x}_i-\mathbf{x}_j||^2 d\mathbf{X}-b\right) \\ &-\lambda_2\left(\int p_{\mathbf{X}}(\mathbf{X})\cdot\frac{1}{n}\sum_{i=1}^{n}||\mathbf{x}_i||^2 d\mathbf{X}-c\right)-\mu\left(\int p_{\mathbf{X}}(\mathbf{X})d\mathbf{X}-1\right), \end{aligned} \tag{2.33}$$

whose partial derivative with respect to $p_{\mathbf{X}}$ at $\mathbf{X}$ reads:

$$\frac{\partial}{\partial p_{\mathbf{X}}(\mathbf{X})}\mathcal{L} = -1 - \log\left(p_{\mathbf{X}}(\mathbf{X})\right) - \frac{\lambda_1}{|E|}\sum_{(i,j)\in E}||\mathbf{x}_i - \mathbf{x}_j||^2 - \frac{\lambda_2}{n}\sum_{i=1}^{n}||\mathbf{x}_i||^2 - \mu. \tag{2.34}$$

Equating this partial derivative to zero yields:

$$\begin{aligned} p_{\mathbf{X}}(\mathbf{X}) &= \exp(-1-\mu)\cdot\exp\left\{-\frac{\lambda_1}{|E|}\sum_{(i,j)\in E}||\mathbf{x}_i - \mathbf{x}_j||^2 - \frac{\lambda_2}{n}\sum_{i=1}^{n}||\mathbf{x}_i||^2\right\} \\ &= \frac{1}{Z}\exp\left\{\mathrm{Tr}\left(-\mathbf{X}'\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\mathbf{X}\right)\right\}. \end{aligned} \tag{2.35}$$

Observe that (2.35) is a matrix normal distribution [13] with partition function Z and parameters $\mathbf{M} = 0, \mathbf{\Sigma} = \left(2\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\right)^{-1}$, and $\mathbf{\Phi} = \mathbf{I}_d$. Hence, the matrix-valued random variable $\mathbf{X} \in \mathbb{R}^{n\times d}$ belongs to:

$$\mathbf{X} \sim \mathcal{MN}_{n\times d}\left(\mathbf{0}, \left(2\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\right)^{-1}, \mathbf{I}_d\right), \tag{2.36}$$

with the partition function

$$Z = (2\pi)^{\frac{nd}{2}}\left|2\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\right|^{\frac{d}{2}}. \tag{2.37}$$

□

Remark 4. *To compute the multipliers λ_1 and λ_2, we substitute the distribution (2.35) back into the Lagrangian (2.33) and solve λ_1 and λ_2 that minimizes the following Lagrange dual function using, e.g., gradient based methods:*

$$\mathcal{L}(\lambda) = \frac{d}{2}\log((2\pi)^n|\mathbf{\Sigma}|) + \lambda_1 b + \lambda_2 c,$$

where $\mathbf{\Sigma} = \left(\frac{2\lambda_1}{|E|}\mathbf{L} + \frac{2\lambda_2}{n}\mathbf{I}_n\right)^{-1}$. Since $\mathbf{L}$ is a real symmetric matrix, we can simultaneously diagonalize $\mathbf{L}$ and $\mathbf{I}_n$. Denote the eigenvalues of the matrix $\mathbf{L}$ to be $\sigma_1, \sigma_2, \ldots, \sigma_n$. Then the determinant of the covariance matrix reads:

$$|\mathbf{\Sigma}| = \prod_{i=1}^{n}\left(\frac{2\lambda_1\sigma_i}{|E|} + \frac{2\lambda_2}{n}\right)^{-1}.$$

Thus the Lagrange dual function can be further simplified as:

$$\mathcal{L}(\lambda) = -\frac{d}{2}\sum_{i=1}^{n}\log\left(\frac{2\lambda_1\sigma_i}{|E|} + \frac{2\lambda_2}{n}\right) + \frac{nd}{2}\log(2\pi) + \lambda_1 b + \lambda_2 c.$$

Hence, computing the multipliers requires to first compute the eigenvalues of $\mathbf{L}$ *(*$\mathcal{O}(n^3)$*), then the evaluation of each gradient step has complexity* $\mathcal{O}(n)$*.*

Remark 5. *In order to determine suitable values for* b *and* c *in the prior belief constraints, SICA may assume that the user already has a good understanding of the point-wise similarity (Equation 2.28) and scale (Equation 2.29) of the data points (or, that the user is not interested in these). Given this assumption,* b *and* c *can simply be set equal to the empirical value of these statistics as measured in the data. If the user wishes, she could of course specify values herself that differ from these. More realistically though, she may be able to specify a* range *of values for the point-wise similarity and scale. The background distribution should then be found as the MaxEnt distribution subject to two* box constraints*, i.e., four inequality constraints: a lower and a upper bound for pairwise similarity as well as for the scale measure. Theorem 3 still applies unaltered though: while the four inequality constraints lead to four Lagrange multipliers, only two may be non-zero at the optimum (one for each box constraint), as for each box constraint only either the upper or the lower bound constraint can be tight.*

Subjectively interesting patterns As the projection $\mathbf{\Pi_W}$ is a linear transformation of matrix random variable $\mathbf{X}$, and $\mathbf{W}$ is of rank $k \leq n$ (full column rank), then $\mathbf{\Pi_W} \sim \mathcal{MN}_{n\times k}\left(\mathbf{0}, \left(2\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\right)^{-1}, \mathbf{I}_k\right)$ [13]. So the marginal $p_{\mathbf{\Pi_W}}$ of background distribution (2.31) reads:

$$p_{\mathbf{\Pi_W}}(\mathbf{\Pi_W}) = \frac{1}{Z}\exp\left\{\mathrm{Tr}\left(-\mathbf{\Pi}'_{\mathbf{W}}\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\mathbf{\Pi_W}\right)\right\}. \tag{2.38}$$

Substituting the marginal distribution (2.38) into the objective function of problem (2.9), and $\hat{\mathbf{X}}\mathbf{W}$ for $\hat{\mathbf{\Pi}}_{\mathbf{W}}$, yields:

$$-\log\left(p_{\mathbf{\Pi_W}}\left(\hat{\mathbf{X}}\mathbf{W}\right)\right) = \mathrm{Tr}\left(\mathbf{W}'\hat{\mathbf{X}}'\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\hat{\mathbf{X}}\mathbf{W}\right) + \log(Z). \tag{2.39}$$

Since the second term of (2.39) is constant, it can be safely left out. Thus the optimization problem (2.9) is equivalent to:

$$\begin{aligned}\max_{\mathbf{W}\in\mathbb{R}^{d\times k}} \quad & \mathrm{Tr}\left(\mathbf{W}'\hat{\mathbf{X}}'\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\hat{\mathbf{X}}\mathbf{W}\right) \\ \text{s.t.} \quad & \mathbf{W}'\mathbf{W} = \mathbf{I}.\end{aligned} \tag{2.40}$$

The solution to this problem consists of a matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$ whose k column vectors are the eigenvectors that corresponding to the top-k eigenvalues of the matrix $\hat{\mathbf{X}}'\left[\frac{\lambda_1}{|E|}\mathbf{L} + \frac{\lambda_2}{n}\mathbf{I}_n\right]\hat{\mathbf{X}} \in \mathbb{R}^{d\times d}$ [23].

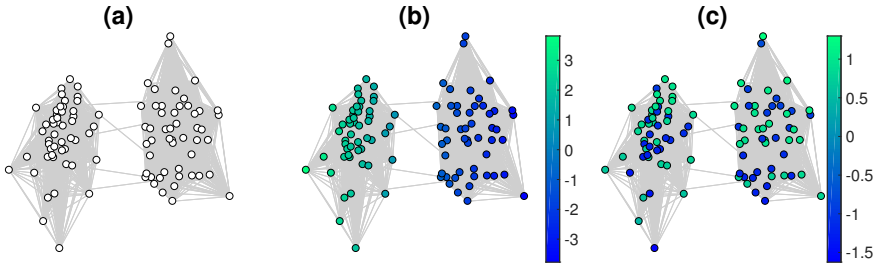

Figure 2.2: Communities data (§2.3.3), (a) the actual network, (b) nodes colored according to their projected values using the first PCA component (c) similar to (b), but for the first SICA component (our method). The x-axis corresponds to the first feature in the data, while the position of points on the y-axis is picked at random. The PCA projection picks up the variance across the clusters, while the SICA projection highlights the variance within the clusters.

The computational complexity of finding an optimal projection $\mathbf{W}$ consists of two parts: (1) solving a convex optimization problem to obtain the background distribution. This can be achieved by applying, e.g., a steepest descent method, which uses at most $\mathcal{O}(\varepsilon^{-2})$ steps (until the norm of the gradient is $\leq \varepsilon$) [28]. For each step, the complexity is $\mathcal{O}(n)$ with n being the size of data. (2) Given the background distribution, we find an optimal projection, the complexity of which is dominated by eigenvalue decomposition ($\mathcal{O}(n^3)$). Hence, the overall complexity of SICA with graph prior is $\mathcal{O}(\frac{n}{\varepsilon^2} + n^3)$.

Example. We synthesized a dataset with 100 users, where each user is described by 10 attributes, i.e., $\hat{\mathbf{X}} \in \mathbb{R}^{100\times 10}$. The first attribute is generated from a bimodal Gaussian distribution such that the first attribute clearly separates the users into two groups. We assume that people within each community are fully connected. To have a more interesting simulation, we also insert a few connections between the communities. The second attribute value is uniformly drawn from $\{-1, +1\}$ which could resemble, e.g., people's sentiment towards a certain topic. The remaining eight attributes are standard Gaussian noise. After sampling, we centered the data.

We assume the user has studied the observed connection between all data points. Hence, the graph-encoded prior expectation is chosen as the actual network structure; i.e., the resulting prior graph consists of the two cliques and a few edges in-between, see Figure 2.2a.

We compare the primary projections given by PCA and SICA, see Figures 2.2b and 2.2c. For both the PCA and SICA projections, we colored the data points according to their projected values, i.e., $\hat{\mathbf{X}}\mathbf{w}$, where $\mathbf{w}$ correspond to the first component of PCA/SICA. In Figure 2.2b, we see that the PCA projection gives one cluster a higher score (green vertices) than the other (blue vertices). Clearly, PCA

	Feature 1	Feature 2	···
PCA 1st component	-0.998	0.015	···
SICA 1st component	0.186	0.957	···

Table 2.1: Communities data (§2.3.3), weights of first component for PCA and SICA.

picks up the structure of the two communities defined by the first attribute. In contrast, SICA assigns both high and low scores within each cluster (Figure 2.2c). That is, it highlights variance *within* the clusters. This is to be expected, because the community structure is very similar to the graph structure, with which we assume the user knows already.

Table 2.1 lists the weight vectors of the projections. As expected, PCA gives a large weight to the first feature, which has higher variance. However, SICA's first component is dominated by the second feature. Hence, by incorporating the community structure as prior expectation, SICA finds an alternative structure corresponding to the second feature.

2.3.4 Discussion: potential and limitations of SICA

Potential of SICA. The three instantiations of SICA discussed in this section are illustrative of SICA's potential to take into account prior beliefs of the data analyst, and to find projections that are interesting with respect to it. The three steps that need to be followed to instantiate SICA are always the same: (1) Express the prior belief in the form of constraints on the expected value of certain specified statistics—i.e. in form of Equation (2.1)—and solve the MaxEnt problem (2.9) to obtain the background distribution. (2) Compute the marginal density function of the background distribution for the data projection onto a projection matrix $\mathbf{W}$. And (3), come up a good optimization strategy. *In principle*, any data analyst able to express their prior beliefs in the required form can thus benefit from this approach.

Limitations of SICA. Yet, each of these steps also implies some important limitations of SICA that should be the subjects of further work. The result of the first step will always be an exponential family distribution, and hence have an analytical form. However, expressing prior belief types as required will often be beyond the capabilities of a data analyst. Also the second step may require considerable mathematical expertise. Indeed, it may not be possible to express the marginal distribution in an analytical form such that it may need to be approximated. And even when it can be expressed analytically, deriving it mathematically may be non-trivial. Finally, thanks to the orthonormality assumption of the projection matrix, general purpose (Stiefel) manifold optimization solvers are in principle applicable, but doing this does not provide any optimality guarantees.

SICA in practice. For these reasons, SICA as a framework is not directly suitable for use by practitioners. Instead, it can be used by researchers to develop specific instantiations of sufficiently broad applicability, which can then be made available to practitioners. Probably the most powerful example of this is the third instantiation (Section 2.3.3). Indeed, it is a very generic prior belief type for which an efficient algorithm exists, and which is relatively easy to be used.

2.4 Experiments

In this section, we present several case studies which demonstrate how SICA may help users to explore various types of real world data. For every case, we specify some background knowledge a user might have, and encode that knowledge using previously defined expressions. The encoded beliefs are then provided to SICA in the form of the background distribution. Third, we analyze the projections computed by SICA and evaluate whether they are indeed interesting with respect to the assumed user's prior. Finally, we summarize the runtime of all experiments presented in this section.

Note that the purpose of our experiments is not to investigate superiority of SICA over existing methods for dimensionality reduction. Instead, we aim to investigate whether and to which extent SICA's results usefully depend on the various prior beliefs, in highlighting information that is complementary to them. Where the answer to this question is positive, SICA is the method of choice—of course, assuming the prior beliefs are well-specified.

2.4.1 t-PCA on real-world data

Setup. We evaluate the use of SICA with a spread prior (t-PCA) on two datasets. The Shuttle[6] data describes radiator positions (seven position classes: 'Rad Flow', 'Fpv Close', 'Fpv Open', 'High', 'Bypass', 'Bpv Close') in a NASA space shuttle and consists of 58000 data points and 9 integer attributes, i.e., $\hat{\mathbf{X}} \in \mathbb{R}^{58000\times 9}$. The 20 Newsgroups[7] data describes four newsgroups (four classes) and has 16242 points and 100 integer attributes, i.e., $\hat{\mathbf{X}} \in \mathbb{R}^{16242\times 100}$. Both datasets are centered such that each attribute has zero mean.

Both of the datasets contain complex structures. Particularly, the shuttle dataset contains highly imbalanced cluster structure: one of the classes forms 80% of the population. For both datasets, we assume the user has a prior belief only about the order of magnitude of the data, i.e., the user would not be surprised by the

[6] `https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)`, retrieved November 18, 2016.

[7] `http://cs.nyu.edu/~roweis/data.html`, retrieved November 18, 2016.

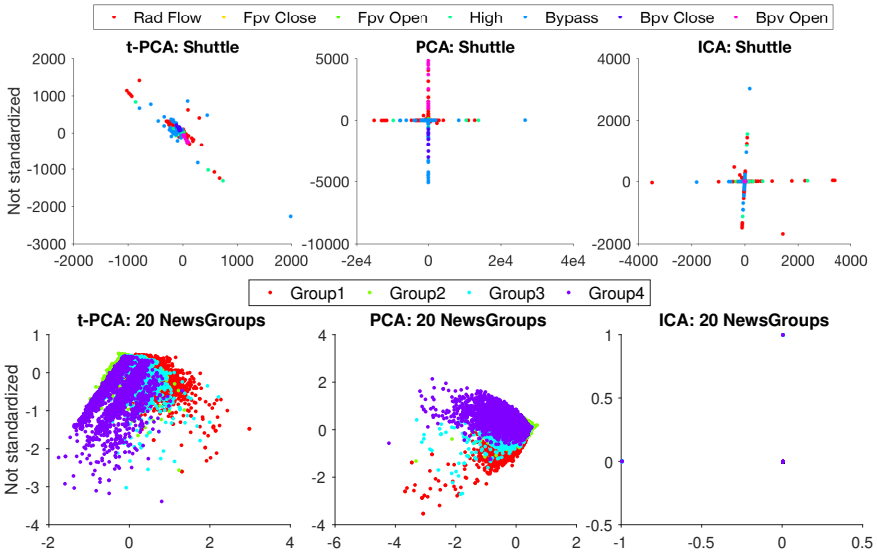

Figure 2.3: Real world data case study for t-PCA (§2.4.1). The top 2 projections found by t-PCA (left), PCA (middle), and FastICA (right). Top row: Shuttle; bottom row: 20 NewsGroups. For the Shuttle dataset, the PCA and FastICA projections show highest variances as well as the most independent dimensions. SICA projection exhibits other, smaller-scale variation. For the 20 NewsGroup dataset, SICA's result is qualitatively similar to PCA's result but with slightly lower variance. The FastICA's result is qualitatively different.

presence of outliers. This can be encoded using the spread prior with a small ρ, e.g., $\rho = 10^{-5} \cdot (\frac{1}{|\mathbf{X}|} \sum_{i=1}^{|\mathbf{X}|} ||x_i||_2)^{\frac{1}{2}}$.

Results. We compared the results of SICA, PCA, and FastICA[8] [18]. FastICA is a popular PP method that implements ICA. We used FastICA with default parameters. The classes for each dataset are plotted in different colors.

Figure 2.3 shows the results of SICA with this prior belief model, for PCA, and for FastICA. For the Shuttle dataset, PCA and FastICA give visually similar results: the highest-variance as well as the most independent dimensions appear to be affected by relatively few data points with large projection values along them. Especially for PCA, the resulting scatter plot has axes with very large scales. Hence the data points that correspond to small scale structure are more likely to be plotted on top of each other, making them harder to discern. SICA, in accounting for order of magnitude variations in the norms of data points, is less biased towards these distant data points. As a result, it prefers lower-variance projections which exhibit other, smaller-scale variation, which therefore provide information that complements the user's expectations.

[8]In the experiment we used the FastICA package for MATLAB. The package can be downloaded from `https://research.ics.aalto.fi/ica/fastica/`

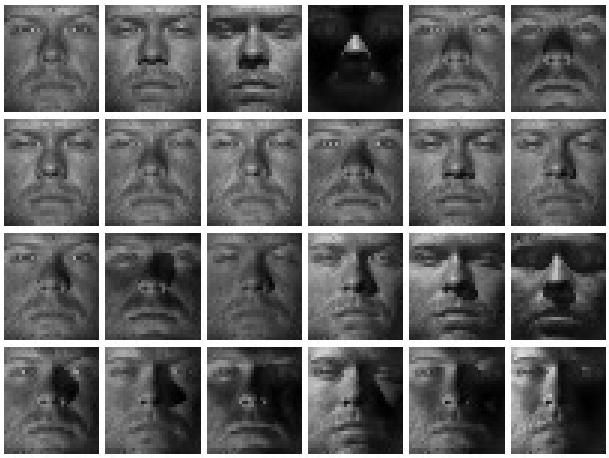

Figure 2.4: Faces dataset (§2.4.2), subject one, first 24 lighting conditions. The data set contains 31 human subjects where each of them has face image taken under 64 lighting conditions. Each face image contains 32×32 *pixels.*

For the 20 Newsgroup dataset, SICA's result is qualitatively similar to PCA's result, although the variance of the SICA projection is slightly lower, arguably in favor of making the more fine-grained variation in the data more apparent. FastICA's result, however, is qualitatively different. It puts all weight on a single binary attribute, such that its top components project all data points onto just three points.

2.4.2 Images and lighting, with a graph prior

Setup. We now apply SICA to explore image data. The Extended Yale Face Database B[9] contains frontal images of 38 human subjects under 64 illumination conditions, for example, see Figure 2.4. We ignored the images of seven subjects whose illumination conditions are not fully specified. The input dataset then contains 1684 data points, each of which is described by 1024 real valued features , i.e., $\hat{\mathbf{X}} \in \mathbb{R}^{1684 \times 1024}$. The data is then centered to have a zero mean. The task of decomposing images in order to account for a number of pre-specified factors has been addressed in the past (e.g., using a N-mode SVD; Vasilescu and Terzopoulos 36). Here we want to explore how SICA weight vectors change according to the prior belief of a specific user.

Let us assume that the user already knows there are lighting conditions and is

[9]This data is available as a preprocessed Matlab file at `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`. The original dataset is described in Georghiades et al. [12], Lee et al. [26].

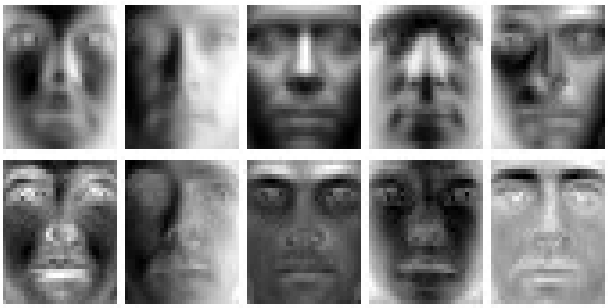

Figure 2.5: Faces data case study (§2.4.2), top five Eigenfaces for PCA (top) and SICA (bottom). The Eigenfaces from PCA are influenced substantially by the variation in lighting conditions, while the Eigenfaces from SICA mainly highlight local facial structures

not interested in them. We can encode such knowledge by declaring that images (data points) with the same lighting condition are similar to each other. This can be expressed in a point-wise similarity prior. We construct a graph where each image is a node and two nodes are connected by an edge if the corresponding images have the same lighting conditions. The resulting prior graph consists of 64 cliques, one for each lighting condition.

Results. We compare the weight vectors of the subjectively interesting components (SICs) given by SICA and top principal components (PCs) given by PCA, namely the Eigenfaces from the two methods. We expect PCA to find a mixture of illumination and facial features, while SICA should find mainly facial structure. Note that illumination conditions vary similarly across the human subjects, while facial structures are *subject specific*. The principal Eigenfaces from PCA and SICA are presented in Figure 2.5. We observe that the Eigenfaces given by PCA are influenced substantially by the variation in lighting conditions. These conditions vary from back-to-front, right-to-left, top-to-down, down-to-top and left-top-to-right-bottom. Because the images of each subject contain every lighting condition, it appears indeed more difficult to separate the subjects based only on the top PCA components. On the other hand, the Eigenfaces from SICA highlight local facial structures, like the eye area (first, third and fifth faces), and the mouth and nose (first, third and fifth faces). Note though that the first and second SICA Eigenfaces also still pick up some lighting variation, which is confirmed by the similarity between the top two SICA and PCA components (left upper corner of Figure 6). The absolute value of the inner product between the first SICA and PCA components is 0.91 and the value of the second components is 0.93. Note also that the similarities of most other SICA and PCA components are considerably smaller, confirming that SICA components are indeed truly different from the PCA components.

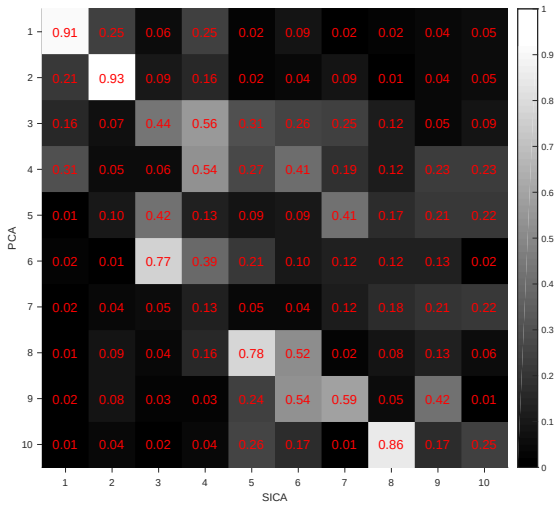

Figure 2.6: Face data case study (§2.4.2), Similarity (absolute value of inner product) between PCA and SICA top 10 components. The similarity between the top two SICA and PCA components confirms that SICA top two components still pick up some lighting variation. The less significant similarity between the other SICA and PCA components indicates SICA components are indeed truly different from the PCA components.

If SICA succeeds in providing insights that contrasts with the prior beliefs about the lighting conditions, the projection of an image onto the top SICs can be expected to separate the subjects better than the projection onto an equal number of top PCs. To verify this, we computed the 10-fold cross-validation loss (with respect to the subjects as labels) of a k-Nearest Neighbors (k-NN) classifier on the projected features with respect to the top PCs and SICs. A projection that separates the subjects well will have low classification loss. We applied k-NN on the SICA/PCA projections with number of components ranging from 1 to 50. Since our goal here is to evaluate whether top SICs are more likely to correspond to facial structure than top PCs, rather than achieve best classification accuracy, we fix $k = 3$. Figure 2.7a shows that indeed top SICs (orange line) give a better separation than top PCs (purple line). In addition, we performed the same experiment

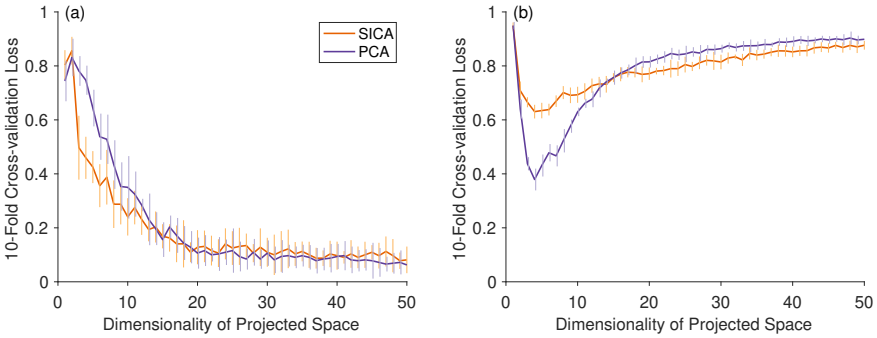

Figure 2.7: Faces data case study (§2.4.2), (a) average 10-fold cross-validation loss (error bars gives one standard deviation, the smaller loss the better) for 3-NN subject classification on the projected data. Top SICs gives better separation of subjects than top PCs. (b) average 10-fold cross-validation loss (error bars gives one standard deviation, the smaller loss the better) for 3-NN lighting condition classification. Top PCs gives better separation of lighting conditions than SICs.

using an SVM (rather than 3-NN) with 10-fold cross validation on the projected features to perform classification. We measured the average losses over 10 folds while varying the number of projected features form 1 to 50. The result (Figure 2.8a) shows SICA is more accurate than PCA when the number of features is small. PCA then catches up when the number of the dimensions increases.

Conversely, as SICA with the stated prior beliefs should result in a projection that highlights information *complementary* to lighting conditions, one can expect that the top SICs perform worse in separating the different lighting conditions than the top PCs. To evaluate this, instead of classifying subjects, we used k-NN to classify different illumination conditions, using the same PCs and SICs as before. That is, where we told SICA explicitly we were not interested in light variation. Figure 2.7b shows that PCA indeed gives better 3-NN classification accuracy than SICA. The result (Figure. 2.8b) obtained by SVM confirms this with another classifier.

2.4.3 Spatial socio-economy, with a graph prior

Now we use SICA to analyze a socio-economic dataset. The German socio-economic data [3] was compiled from the database of the German Federal Statistical Office. The dataset consists of socio-economic records of 412 administrative districts in Germany. The data features used in this case study fall into two groups: election vote counts and age demographics. We additionally coded for each district the geographic coordinates of the district center and which districts share a border with each other.

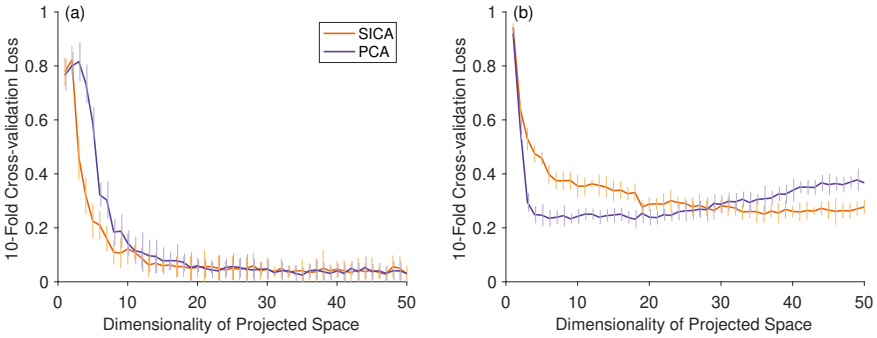

Figure 2.8: Faces data case study (§2.4.2), (a) average 10-fold cross-validation loss (error bars gives one standard deviation, the smaller loss the better) for SVM subject classification on the projected data. Top SICs gives better separation of subjects than top PCs. (b) average 10-fold cross-validation loss (error bars gives one standard deviation, the smaller loss the better) for SVM lighting condition classification. Top PCs gives better separation of lighting conditions than SICs.

Vote attribute group

Setup. Let us assume a user is interested in exploring the voting behavior of different districts in Germany. The (real-valued) data attributes about the 2009 German elections cover the percentage of votes on the five largest political parties[10]: CDU/CSU, SPD, FDP, GREEN, and LEFT. Thus, we have a dataset $\hat{\mathbf{X}} \in \mathbb{R}^{412\times 5}$. We centered the data attribute-wise by subtracting the mean from each data point.

Let us assume also that the user already knows the East-West divide has still a large influence. Hence, she may believe the voting behavior of the districts in the east are similar to each other, and the same goes for the west. This prior belief can also be encoded as point-wise similarities. By treating each district as a graph node, we can translate our knowledge into prior expectations, by connecting similar districts with edges. This results in a graph with two cliques: one clique consists of all districts in East Germany, the other clique contains the rest.

Results. The projection onto the first PC (Figure 2.9a) shows smooth variation across the map. Districts in western Germany and Bavaria (south) receive high scores (red circles) and districts in East Germany (Brandenburg and Thuringa) have low scores (dark blue circles). Table 2.2 additionally shows the weight vectors of the top PC and SIC. The PC is dominated by the difference between CDU/CSU and Left. This is expected, because this indeed is the primary division in the elections; East Germany votes more Left, while in Bavaria, CSU is very popular.

[10] `https://en.wikipedia.org/wiki/List_of_political_parties_in_Germany`, retrieved November 18, 2016.

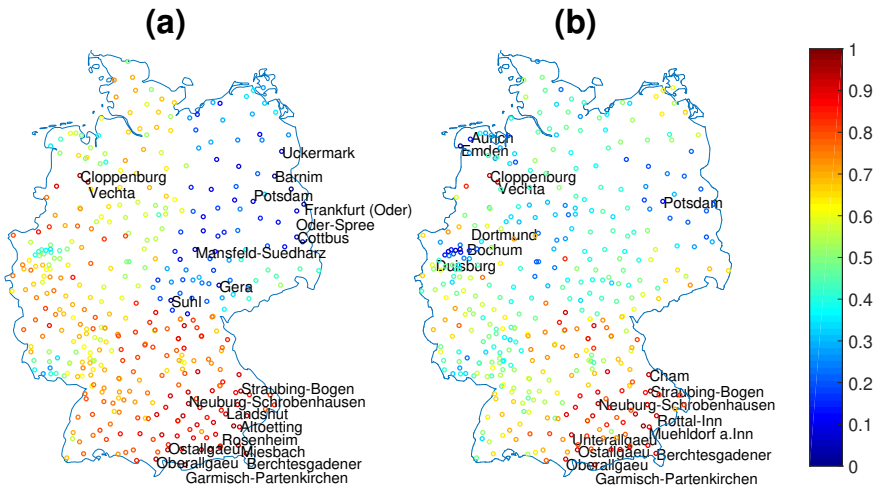

Figure 2.9: German socio-economics data vote attributes (§2.4.3). (a) The geographic scatter plot of districts with each district colored according to its projective value onto top PC. The top 10 districts with most positive and most negative projective values are labeled. The top PC assigns low scores to the districts in East Germany, while it gives rest districts relatively high scores. (b) The same geographic scatter plot for the top SIC. Although SICA still shows considerable global variation (in this case between the north and the south), it also highlights the variations that are more local.

However, SICA highlights a different pattern; the competition between CDU/CSU and SPD is more local. Although there is still considerable global variation (in this case between the south and the north), we also observe that the Ruhr area (Dortmund and around) is similar to East Germany in that the social-democrats are preferred over the Christian parties. Arguably, the districts where this happens are those with a large fraction of working class, like the Ruhr area. Perhaps they vote more on parties that put more emphasis on interests of the less-wealthy part of the population.

To investigate this in a more quantitative manner, we applied an SVM to classify the eastern versus non-eastern districts using the vote data projected onto the top SICA and PCA components. We measured the 10-fold cross-validation losses for the projected data's dimensionality ranging from 1 to 5. Figure 2.10a shows that the first two PCA components lead to a smaller loss than SICA. This indicates that the two top PCs indeed reflect more to the eastern and non-eastern division. The similarity matrix (Figure 2.10b) of the PCs and SICs also shows that the first and second components of the two methods are different. Notice that the third SIC (third column) is similar to the first and second PCs. This explains why when the dimensionality of projected space increased to three, the classification loss of SICA drops to the same as PCA.

	CDU/CSU	SPD	FDP	GREEN	Left
PCA 1st	0.53	-0.13	0.22	0.13	-0.80
SICA 1st	0.72	-0.65	0.10	-0.09	-0.19

Table 2.2: German socio-economics data vote attributes (§2.4.3), weights given by top PCA and SICA component.

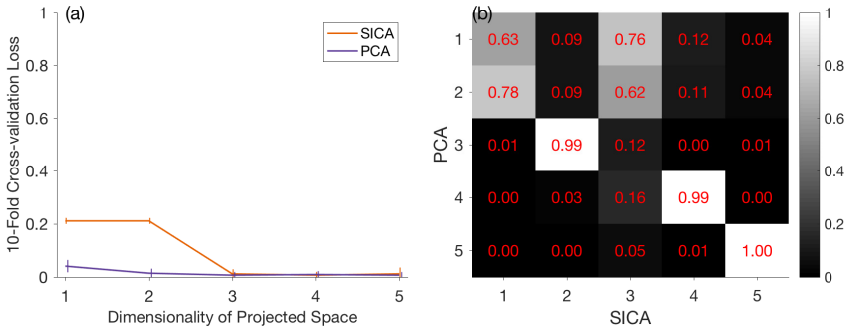

Figure 2.10: German socio-economics data vote attributes (§2.4.3). (a) average 10-fold cross-validation loss (error bars gives one standard deviation) for eastern and non-eastern districts classification on the projected data. The top two PCs lead to a smaller loss than the top two SICs. (b) Similarity (absolute value of inner product) between PCA and SICA components. The first and second components of the two methods are different. The third SIC is similar to the first and second PCs

Demographic attribute group

Setup. Next, we assume that the user is interested in exploring the age demographics of different districts. The demographic attribute group describes the age distribution of the population (in fractions) for every district, over five categories: *Elderly* (age > 64), *Old* (between 45 and 64), *Middle Aged* (between 25 and 44), *Young* (between 18 and 24), and *Children* (age < 18), represented by a positive real-valued vector of length 5. Thus, we have a data set $\hat{\mathbf{X}} \in \mathbb{R}^{412\times 5}$. We then centered the data attribute-wise.

We assume again the user understands the influence of the historical east-west divide. We are interested in finding patterns orthogonal to that division. The population density is lower in East Germany than the rest of country. According to Wikipedia[11]: "About 1.7 million people have left the new federal states since the fall of the Berlin Wall, or 12% of the population. A disproportionately high number of them were women under 35". Also the Berlin-Institute for Population and Development[12] reports: "the birth rate in East Germany dropped down to 0.77 af-

[11] `https://en.wikipedia.org/wiki/New_states_of_Germany#Demographic_development`, retrieved November 18, 2016.

[12] `http://www.berlin-institut.org/fileadmin/user_upload/Studien/`

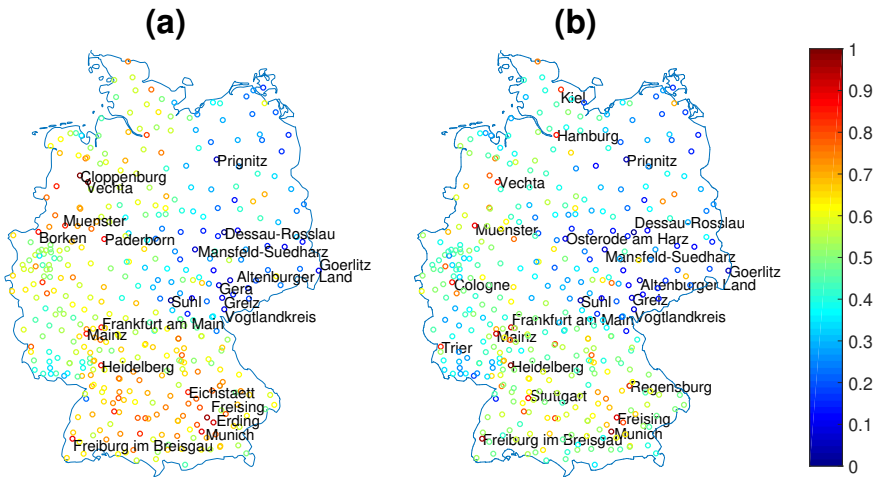

Figure 2.11: German socio-economics data demographic attributes (§2.4.3). (a) The geographic scatter plot of districts with each district colored according to its projective value onto first PC. The top 10 districts with most positive and most negative projective values are labeled. The PC again highlights the difference between East and West Germany. (b) The same geographic scatter plot against first SICA component. The top SIC assigns large negative scores to East Germany, while it also highlights the large cities.

	Elderly	Old	Mid-Age	Young	Child
PCA	-0.61	-0.42	0.43	0.09	0.51
SICA	-0.62	-0.32	0.69	0.19	0.06

Table 2.3: German socio-economics data age demographics (§2.4.3), weights given by first PCA and SICA component.

ter unification, and raised to 1.30 nowadays compare to 1.37 in the West". Given this (in Germany common sense) knowledge, SICA should be able to offer new insights. Hence, we assume again that the demographics of the districts in East Germany are similar, and the remaining districts are also similar. Formalizing such belief as point wise similarities results in a graph with two cliques: one consists of all districts in East Germany, the other contains the rest.

Results. Projection on the top PC (Figure 2.11a) confirms the user's prior expectations. There is a substantial difference between East and West Germany. In the visualization, high projection values (red color) appear mostly in East Germany, while low values (blue color) appear mostly in the rest of Germany. If we look at the weights of the top PC (Table 2.3), we find that the projection is based on large negative weights to people above 44 (Old and Elder), and large positive weights

`Kurzfassung_demografische_lage_englisch.pdf`

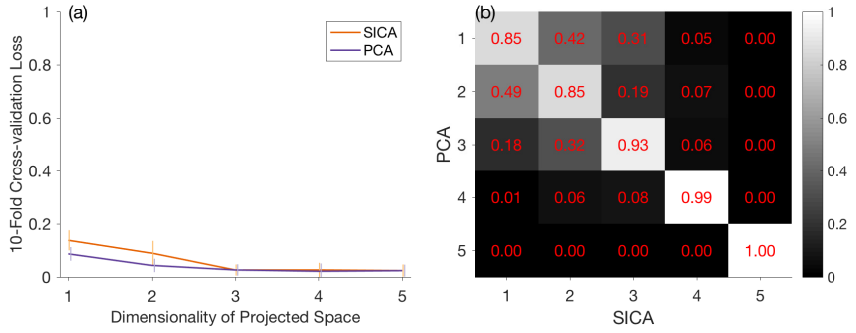

Figure 2.12: German socio-economics data age demographics (§2.4.3). (a) average 10-fold cross-validation loss (error bars gives one standard deviation) for eastern and non-eastern districts classification on the projected data. The top two PCA components result in a slightly smaller loss than SICA. (b) Similarity (absolute value of inner product) between PCA and SICA components. The first and second components of the two methods are very similar.

to the younger population (age < 45). This confirms that indeed the demographic status of East Germany deviates.

SICA results in a different projection (2.11b), even though the difference is more subtle than in the analysis of the voting behavior. Although SICA also assigns large negative scores to East Germany, presumably because there are relatively many elderly there, SICA also highlights the large cities, e.g., Munich, Cologne, Frankfurt, Hamburg, Kiel, Trier. In addition to showing a smooth geographic East-West trend, SICA also seems to highlight districts whose demographic status deviates from its surrounding districts. Indeed, from the weight vector (Table 2.3) we see that these districts are found by considering the number of middle aged people against the number of elderly. We know that many middle-aged ($24 - 44$) working people live in large cities, and, according to the report from Berlin-Institute for Population and Development, "large cities generally have fewer children, since they offer families too little room for development". Indeed, we find that families live in the neighboring districts, highlighting a perhaps less-expected local contrast.

Also, to further investigate this more quantitatively, we applied an SVM to classify the eastern versus non-eastern districts using the projected demographic attributes. Figure 2.12a shows that the top two PCA components result in a slightly smaller loss than SICA. This indicates that the top PCs and SICs both correspond to the eastern and non-eastern division. The similarity matrix (Figure 2.12b) of PCA and SICA components also shows the first and second components of the two methods are very similar. However, according to the visualization, the best (positively) scored districts by SICA (Figure 2.11b) highlight large cities more

	Synthetic outlier	Shuttle	20News Group	Synthetic community	Socio-eco. (age)	Socio-eco. (vote)	Face image
SICA	0.12	1.75	8.07	0.03	0.06	0.04	2.26
PCA	< 0.01	0.08	0.25	0.01	< 0.01	< 0.01	0.56

Table 2.4: Runtime (in seconds) of SICA and PCA for all experiments (§2.4.4). Each measurement is averaged over ten runs. We used a machine with Intel Quad Core 2.7 GHz CPU and 16 GB 1600 MHz DDR3 RAM.

than the PCA result (Figure 2.11a). Also the highlighted cities stand out more from their surrounding area.

2.4.4 Runtime

Table 2.4 summarizes the runtime of PCA and SICA in all experiments presented in this chapter. In all these cases, SICA takes more time to compute the projections. For the first three columns (t-PCA cases), we used the solver offered by Manopt to perform gradient descent over the Stiefel manifold. We tried ten random starts in all three cases and picked the projection that gives the best objective. The ten random starts already give stable local optima in all three cases. Note that t-PCA scales gracefully when the data size increases from Synthetic dataset (1010×2) to Shuttle (58000×9) and then 20NewsGroup (16242×100).

The other four experiments are about SICA with graph prior. Again, SICA scales well from the Synthetic data (100×10) to the socio-economical dataset (both 412×5) and then the Face image dataset (1684×1024). However, although both SICA and PCA are based on eigenvalue decomposition, SICA spend more time than PCA. One reason is that in order to construct the Laplacian matrix in (2.40) SICA needs to loop through the data as well as find the best multipliers. Note that the current experiments are based on a quick implementation—a more careful implementation may improve the run time of SICA.

2.5 Related work

SICA is linear, unsupervised, and subjective. Dimensionality reduction (DR) methods, as indicated by the name, aim to find lower dimensional representation of high dimensional data. Here "dimension" refers to the number of features that are used to describe the data. Finding a lower dimensional representation further boils down to either select a subset of the original features or transform the feature space to another (low-dimensional) space. Here we mainly discuss the line of work for feature transformation (extraction), since they are more closely related to our work.

Supervised v.s. Unsupervised. DR methods are often designed with a certain

goal: to have lower dimensional representations with some specific properties. For example Principal Component Analysis (PCA) [32, 20] is often used for computing a presentation of dataset where the data variance is preserved, whereas Canonical Component Analysis (CCA) [16] aims to find pairs of directions in two feature spaces where the corresponding two datasets are highly correlated. While PCA and CCA achieve their goals in an unsupervised manner, Linear Discriminant Analysis (LDA) [10], on the other hand, extracts discriminative features according to the given class labels with a supervised flavor. The new features provided by DR methods can not only be used for later classification or prediction, but also to explore the structures in the data, e.g., Self Organizing Map (SOM) [22] for exploratory data analysis. In order to meet different analysis goals under a unified framework, Projection Pursuit (PP) [11] was proposed to locate different projections according to some predefined "interestingness index". Different from the previous works, we seek for data projections that are interesting particularly to the user. Therefore, SICA aims to propose a generic interestingness measure that does not explicitly depend on the context of the data or on the specific analytic tasks.

Linear v.s. Non-linear. Orthogonally, when approaching these goals, DR methods further assume the relationship between the original data and its lower dimensional representation to be either linear or non-linear. The aforementioned methods (PCA, CCA and LDA) compute new data representations via linear transformation. Additionally, classical Multidimensional Scaling [25] also finds a linear transformation that preservers the distances between the data points. We refer the reader to the survey by Cunningham and Ghahramani [6] and the references therein for a comprehensive review of linear DR techniques. However, in reality, high dimensional data often obeys certain constraints; data then lies on a low-dimensional (non-linear) manifold embedded in the original feature space. Non-linear dimensionality reduction methods like SOM approximate such a manifold by a set of linked nodes. Building upon Multidimensional Scaling, ISOMAP [35] seeks to preserve the intrinsic geometry of the data by first encoding neighborhood relations as a weighted graph. This inspired later spectral methods [37, 29] as well as different manifold learning approaches [1, 15, 39] that try to solve a eigenproblem in order to discover the intrinsic manifold structure of the data, using an eigendecomposition to preserve the local properties of the data. Note that with a graph prior, SICA computes linear projections in a spectral-method-like manner (§2.3.3). However, the previously mentioned non-linear DR methods are interested in the eigenvectors corresponding to the smallest k eigenvalues of the Laplacian, as they provide insights into the local structure of the underlying graph, while SICA identifies mappings that *target* non-smoothness with respect to the user's beliefs about the data, while maximizing the variance of the data in the resulting subspace. Interestingly, the resulting optimization problem is not simply the opposite of existing approaches.

Objective v.s. Subjective. The aforementioned methods are mainly "objective" in the sense there that user is not explicitly considered. A notable exception is the work on User Intent Modeling for Information Discovery [34], where indeed an explicit relevance model is built to help a user find information relevant to her search. Their tool also computes a 2D embedding of the search results, accounting for their user and session specific relevance. However, they do not introduce a new theoretically well-motived method to find a low-dimensional subspace that accounts for background knowledge or intent. That is also not the focus of their work, which is rather the identification of relevant results. Some other techniques have been proposed in exploratory data analysis that take into account the user knowledge to determine interesting projections. For instance, Brown et al. [5] suggests an interactive process in which the user provides feedback by moving incorrectly-positioned data points to locations that reflect their understanding. In a similar manner, Paurat and Gärtner [31] make use semi-supervised least squares projections but allowing the user to select and rearrange some of the embedded data points. In the work by Iwata et al. [19], the authors use active learning to select candidate data points for the user to relocate so that they can achieve their desired visualization. All of these methods, guided by the user, interactively present different aspects of the data. Finally the work by Weinberger and Saul [38] require the practitioner to provide auxiliary information, e.g. a similarity graph, that identify target neighbours for each data point, that is then used to constraint their optimization problem. This prior knowledge is the structure that one wants to preserve, as opposed to SICA. To our best knowledge, SICA is the first subjective DR method which finds lower-dimensional data representations that are as interesting as possible for a particular user. Hence, SICA adds another layer to the family of dimensionality reduction methods.

2.6 Conclusion

In exploratory data analysis, structures in the data often have different value for different tasks and data analysts. To address this, the Projection Pursuit literature has introduced numerous *projection indices* that quantify the interestingness of a projection in various ways. However, it still seems to be conceptually challenging to define a generic quality metric for the tasks of exploratory data analysis. As an attempt in this direction, we present SICA, a new linear dimensionality reduction approach that explicitly embraces the subjective nature of interestingness. In this chapter, we show how the modeling of a user's belief state can be used to drive a subjective interestingness measure for DR. Such interestingness measure is then used to search for subjectively interesting projections of data. Results from several case study show that it can be meaningful to account for available prior knowledge about the data.

Avenues for further work include incorporating multiple prior expectations simultaneously (e.g., define multiple (disjoint) groups of similar nodes using graph prior), to enable more flexible iterative analysis. This involves solving a MaxEnt optimization problem subject to multiple constraints. We also plan to study how to improve the interpretability of the projections, e.g., finding projections with sparse weight vectors. In terms of visualization, an interesting future direction is to investigate how the SICA result will be affected by removing the assumption of the resolution being the same through all dimensions. Although that is already possible, one question is how a user could conveniently input these expectations into the system. Another open question is to what extend SICA can be applied to nonlinear dimensionality reduction. Finally, alternative types of prior expectations are also worth examining.

Acknowledgments

We thank the anonymous reviewers for their constructive and insightful comments. We are grateful to Petteri Kaski for discussions about the complexity of t-PCA.

Appendices

A Probability approximation based on distribution (2.17)

We want to show that given marginal density function (2.17) the probability $\Pr(\mathbf{XW} \in [\hat{\mathbf{\Pi}}_{\mathbf{W}} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_{\mathbf{W}} + \Delta\mathbf{1}])$ can be approximated well by using the form $p_{\mathbf{\Pi}_{\mathbf{W}}}(\mathbf{XW}) \cdot 2\Delta$ for sufficiently small Δ. As random variable $\mathbf{XW}$ in distribution (2.17) consists of elements that are all independent to each other, it is sufficient to show the approximation quality for one dimensional normal distribution $\mathbf{x} \sim \mathcal{N}(0, \sigma^2)$:

Proposition 1. *For one dimensional normal random variable $\mathbf{x} \sim \mathcal{N}(0, \sigma^2)$, the approximation of probability $Pr(\mathbf{x} \in [\mathbf{x} - \Delta, \mathbf{x} + \Delta])$ by $p_{\mathbf{x}}(\mathbf{x}) \cdot 2\Delta$ has a bounded log approximation ratio:*

$$\left| \log \left(\frac{Pr(\mathbf{x} \in [\mathbf{x} - \Delta, \mathbf{x} + \Delta])}{p_{\mathbf{x}}(\mathbf{x}) \cdot 2\Delta} \right) \right| \leq \begin{cases} \frac{\Delta(2|\mathbf{x}| + \Delta)}{2\sigma^2} & : |\mathbf{x}| \geq \Delta, \\ \frac{3\Delta^2}{2\sigma^2} & : |\mathbf{x}| \leq \Delta. \end{cases}$$

Thus, for given σ and $\mathbf{x}$, if Δ is sufficiently small and $\mathbf{x}\Delta$ tends to 0, the upper bound of the log approximation ratio tends to zero. Namely, the approximation is tight.

Proof. Let us first consider the case where $\mathbf{x} - \Delta > 0$. Because of the symmetry of the normal distribution, the result also applies for the case where $\mathbf{x} + \Delta < 0$.We have:

- Estimation of the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}$.
- Upper bound on the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}-\Delta)^2/(2\sigma^2)}$.
- Lower bound on the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}$.

Then the log approximation ratio can be computed as:

1. for upper bound we have

$$\left|\log\left(\frac{2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}-\Delta)^2/(2\sigma^2)}}{2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \left|\log\left(\frac{e^{-(\mathbf{x}-\Delta)^2/(2\sigma^2)}}{e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \frac{\Delta(2\mathbf{x}-\Delta)}{2\sigma^2}$$

2. and for lower bound we have

$$\left|\log\left(\frac{2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}}{2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \left|\log\left(\frac{e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}}{e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \frac{\Delta(2\mathbf{x}+\Delta)}{2\sigma^2}$$

Since $\mathbf{x}, \Delta > 0$, the absolute log approximation ratio of lower bound is always smaller than the ratio achieved by the upper bound, we have for $\mathbf{x} - \Delta > 0$:

$$\left|\log\left(\frac{p_{\mathbf{x}}\left(\mathbf{x} \in (\mathbf{x}-\Delta, \mathbf{x}+\Delta)\right)}{p_{\mathbf{x}}(\mathbf{x}) \cdot 2\Delta}\right)\right| \leq \frac{\Delta(2\mathbf{x}+\Delta)}{2\sigma^2} \tag{41}$$

Given σ and $\mathbf{x}$, if Δ is sufficiently small such that $\mathbf{x}\Delta$ close to 0, then the approximation at $\mathbf{x}$ ($|\mathbf{x}| \geq \Delta$) is tight.

Remark 6. *In general, for $|\mathbf{x}| \geq \Delta$, the right hand side in inequality (41) can be replaced by $\frac{\Delta(2|\mathbf{x}|+\Delta)}{2\sigma^2}$*

Let us now consider the case where $-\Delta \leq \mathbf{x} \leq \Delta$. Without losing generality, we assume $p(\mathbf{x}-\Delta) > p(\mathbf{x}+\Delta)$. This leads to:

- Estimation of the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}$.
- Upper bound on the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}$.
- Lower bound on the probability: $2\Delta \cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}$.

Then the log approximation ratio can be computed as:

1. for upper bound we have

$$\left|\log\left(\frac{2\Delta\cdot 1/\sqrt{2\pi\sigma^2}}{2\Delta\cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \left|\log\left(\frac{1}{e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| \tag{42}$$

$$= \frac{\mathbf{x}^2}{2\sigma^2} \tag{43}$$

$$\leq \frac{\Delta^2}{2\sigma^2}, \tag{44}$$

2. and for lower bound we have

$$\left|\log\left(\frac{2\Delta\cdot 1/\sqrt{2\pi\sigma^2}e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}}{2\Delta\cdot 1/\sqrt{2\pi\sigma^2}e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| = \left|\log\left(\frac{e^{-(\mathbf{x}+\Delta)^2/(2\sigma^2)}}{e^{-\mathbf{x}^2/(2\sigma^2)}}\right)\right| \tag{45}$$

$$= \frac{\mathbf{x}\Delta}{\sigma^2} + \frac{\Delta^2}{2\sigma^2} \tag{46}$$

$$\leq \frac{3\Delta^2}{2\sigma^2} \tag{47}$$

Thus, we have for $-\Delta \leq \mathbf{x} \leq \Delta$:

$$\left|\log\left(\frac{\Pr\left(\mathbf{x}\in[\mathbf{x}-\Delta,\mathbf{x}+\Delta]\right)}{p_{\mathbf{x}}(\mathbf{x})\cdot 2\Delta}\right)\right| \leq \frac{3\Delta^2}{2\sigma^2} \tag{48}$$

Given σ, if Δ is sufficiently small, then the approximation at $\mathbf{x}$ ($|\mathbf{x}| \leq \Delta$) is tight. □

B Probability approximation based on distribution (2.24)

We want to show that given marginal density function (2.24) the probability $\Pr(\mathbf{XW} \in [\hat{\mathbf{\Pi}}_{\mathbf{W}} - \Delta\mathbf{1}, \hat{\mathbf{\Pi}}_{\mathbf{W}} + \Delta\mathbf{1}])$ can be approximated well by using the form $p_{\mathbf{\Pi}_{\mathbf{W}}}(\mathbf{XW}) \cdot 2\Delta$ for sufficiently small Δ. As random variable $\mathbf{XW}$ in distribution (2.24) consists of elements that are all independent to each other, it is sufficient to show the approximation quality for a one dimensional t-distribution with degree of freedom ν:

Proposition 2. *The one dimensional r.v.* $\mathbf{x}$ *follows a* t*-distribution with density function*

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}.$$

Hence, the approximation of probability $Pr\left(\mathbf{x}\in[\mathbf{x}-\Delta,\mathbf{x}+\Delta]\right)$ *by* $p_{\mathbf{x}}(\mathbf{x})\cdot 2\Delta$*, has a bounded log approximation ratio:*

$$\left|\log\left(\frac{Pr\left(\mathbf{x}\in[\mathbf{x}-\Delta,\mathbf{x}+\Delta]\right)}{p_{\mathbf{x}}(\mathbf{x})\cdot 2\Delta}\right)\right| \leq \begin{cases} \frac{\Delta(2|\mathbf{x}|+\Delta)}{\mathbf{x}^2+\nu} & : |\mathbf{x}| \geq \Delta, \\ \frac{\Delta^2}{\nu}\max\left(\frac{\nu+1}{2}, 4\right) & : |\mathbf{x}| \leq \Delta. \end{cases}$$

Thus, for given σ, ν ($\nu > 0$), and $\mathbf{x}$, if Δ is sufficiently small and $\mathbf{x}\Delta$ tends to 0, the upper bound of the log approximation ratio tends to zero. Namely, the approximation is tight.

Proof. Let us first consider the case where $\mathbf{x} - \Delta > 0$. Because of the symmetry of the t-distribution, the result also applies for the case where $\mathbf{x} + \Delta < 0$. Let $1/\mathbf{Z}_\nu = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}$, we have:

- Estimation of the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}$.
- Upper bound on the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}-\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}$.
- Lower bound on the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}+\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}$.

Then the log approximation ratio can be computed as:

1. for upper bound we have

$$\begin{aligned}\left|\log\left(\frac{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}-\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}}{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}}\right)\right| &= \left|\log\left(\frac{\nu + \mathbf{x}^2 - 2\mathbf{x}\Delta + \Delta^2}{\mathbf{x}^2 + \nu}\right)\right| \\ &= \left|\log\left(1 + \frac{\Delta^2 - 2\mathbf{x}\Delta}{\mathbf{x}^2 + \nu}\right)\right| \\ &\leq \frac{\Delta(\Delta - 2\mathbf{x})}{\mathbf{x}^2 + \nu}\end{aligned}$$

2. and for lower bound we have

$$\begin{aligned}\left|\log\left(\frac{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}+\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}}{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}}\right)\right| &= \left|\log\left(\frac{\nu + \mathbf{x}^2 + 2\mathbf{x}\Delta + \Delta^2}{\mathbf{x}^2 + \nu}\right)\right| \\ &= \left|\log\left(1 + \frac{\Delta^2 + 2\mathbf{x}\Delta}{\mathbf{x}^2 + \nu}\right)\right| \\ &\leq \frac{\Delta(\Delta + 2\mathbf{x})}{\mathbf{x}^2 + \nu}\end{aligned}$$

By the assumption $\mathbf{x} > \Delta$, we have $\frac{\Delta(\Delta-2\mathbf{x})}{\mathbf{x}^2+\nu} \leq \frac{\Delta(\Delta+2\mathbf{x})}{\mathbf{x}^2+\nu}$, that is

$$\left|\log\left(\frac{\Pr\left(\mathbf{x} \in [\mathbf{x} - \Delta, \mathbf{x} + \Delta]\right)}{p_{\mathbf{x}}(\mathbf{x}) \cdot 2\Delta}\right)\right| \leq \frac{\Delta(\Delta + 2\mathbf{x})}{\mathbf{x}^2 + \nu}. \tag{49}$$

For given σ and $\mathbf{x}$, if Δ is sufficiently small such that $\mathbf{x}\Delta$ close to 0, then the bound $\frac{\Delta(\Delta+2\mathbf{x})}{\mathbf{x}^2+\nu}$ is close to zero. Namely, the approximation at $\mathbf{x}$ ($|\mathbf{x}| \geq \Delta$) is tight.

Remark 7. *In general, for $|\mathbf{x}| \geq \Delta$, the right hand side in inequality (49) can be replaced by $\frac{\Delta(2|\mathbf{x}|+\Delta)}{\mathbf{x}^2+\nu}$*

Let us now consider the case where $-\Delta < \mathbf{x} < \Delta$. Without losing generality, we assume $p(\mathbf{x} - \Delta) > p(\mathbf{x} + \Delta)$. This leads to:

- Estimation of the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}$.
- Upper bound on the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}$
- Lower bound on the probability: $2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}+\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}$.

Then the log approximation ratio can be computed as:

1. for upper bound we have

$$\begin{aligned}\left|\log\left(\frac{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}}{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}}\right)\right| &= \left|\log\left((1 + \frac{\mathbf{x}^2}{\nu})^{\frac{\nu+1}{2}}\right)\right| \\ &\leq \frac{\nu+1}{2}\log(1 + \frac{\Delta^2}{\nu}) \\ &\leq \frac{\nu+1}{2} \cdot \frac{\Delta^2}{\nu}\end{aligned}$$

2. and for lower bound we have

$$\begin{aligned}\left|\log\left(\frac{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{(\mathbf{x}+\Delta)^2}{\nu})^{-\frac{\nu+1}{2}}}{2\Delta \cdot \frac{1}{\mathbf{Z}_\nu}(1 + \frac{\mathbf{x}^2}{\nu})^{-\frac{\nu+1}{2}}}\right)\right| &= \left|\log\left(\frac{\nu + \mathbf{x}^2 + 2\mathbf{x}\Delta + \Delta^2}{\mathbf{x}^2 + \nu}\right)\right| \\ &= \left|\log\left(1 + \frac{\Delta^2 + 2\mathbf{x}\Delta}{\mathbf{x}^2 + \nu}\right)\right| \\ &\leq \left|\log\left(\frac{\nu + 4\Delta^2}{\nu}\right)\right| \\ &\leq \frac{4\Delta^2}{\nu}\end{aligned}$$

For given σ and ν, if Δ is sufficiently small, then the bound $\frac{\Delta^2}{\nu}\max\left(\frac{\nu+1}{2}, 4\right)$ is close to zero. Namely, the approximation at $\mathbf{x}$ ($|\mathbf{x}| \leq \Delta$) is tight. □

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 27–35, New York, NY, USA, 2013. ACM.

[4] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014. URL `http://www.manopt.org`.

[5] Eli T. Brown, Jingjing Liu, Carla E. Brodley, and Remco Chang. Disfunction: Learning distance functions interactively. In *IEEE VAST*, pages 83–92, Seattle, WA, USA, 2012. IEEE. ISBN 978-1-4673-4752-5.

[6] John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015.

[7] Tijl De Bie. An information theoretic framework for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 564–572, New York, NY, USA, 2011. ACM.

[8] Tijl De Bie. Subjective interestingness in exploratory data mining. In *International Symposium on Intelligent Data Analysis*, pages 19–31, Berlin, Heidelberg, 2013. Springer.

[9] Tijl De Bie, Jefrey Lijffijt, Raul Santos-Rodriguez, and Bo Kang. Informative data projections: A framework and two examples. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. i6doc.com, 2016.

[10] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[11] Jerome H Friedman and John W Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(9):881–890, 1974.

[12] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[13] Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. CRC Press, 1999.

[14] David J Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT press, 2001.

[15] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, pages 153–160, 2004.

[16] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28 (3/4):321–377, 1936.

[17] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons, 2004.

[18] Aapo Hyvärinen et al. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[19] Tomoharu Iwata, Neil Houlsby, and Zoubin Ghahramani. Active learning for interactive visualization. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 342–350, 2013.

[20] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[21] Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. Subjectively interesting component analysis: Data projections that contrast with prior expectations. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1615–1624, 2016.

[22] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.

[23] Effrosini Kokiopoulou, Jie Chen, and Yousef Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011.

[24] Samuel Kotz and Saralees Nadarajah. *Multivariate t-distributions and their applications*. Cambridge University Press, 2004.

[25] Joseph B Kruskal and Myron Wish. *Multidimensional scaling*. Sage, 1978.

[26] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[27] Jefrey Lijffijt, Panagiotis Papapetrou, and Kai Puolamäki. A statistical significance testing approach to mining the most informative set of patterns. *Data Mining and Knowledge Discovery*, 28(1):238–263, 2014.

[28] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

[29] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[30] Onishchik. Stiefel manifold. Encyclopedia of Mathematics. `http://www.encyclopediaofmath.org/index.php?title=Stiefel_manifold&oldid=12028`, 2011. Oneline; Accessed on 2017-06-21.

[31] D Paurat and T Gärtner. Invis: A tool for interactive visual data analysis. In *ECML-PKDD*, pages 672–676, 2013.

[32] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[33] Kai Puolamaki, Panagiotis Papapetrou, and Jefrey Lijffijt. Visually controllable data mining methods. In *IEEE International Conference on Data Mining Workshops*, pages 409–417. IEEE, 2010.

[34] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. Interactive intent modeling: Information discovery beyond search. *Communications of the ACM*, 58(1):86–92, 2015.

[35] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319, 2000.

[36] M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision*, pages 447–460, Berlin, Heidelberg, 2002. Springer.

[37] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[38] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[39] Kilian Q Weinberger, Fei Sha, Qihui Zhu, and Lawrence K Saul. Graph laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, pages 1489–1496, 2006.

[40] Konstantinos Zografos. On maximum entropy characterization of pearson's type ii and vii multivariate distributions. *Journal of Multivariate Analysis*, 71 (1):67–75, 1999.

3

Non-linear Representations

Conditional t-SNE: Complementary t-SNE embeddings through factoring out prior information

Abstract Dimensionality reduction and manifold learning methods such as t-Distributed Stochastic Neighbor Embedding (t-SNE) are routinely used to map high-dimensional data into a 2-dimensional space to visualize and explore the data. However, two dimensions are typically insufficient to capture all structure in the data, the salient structure is often already known, and it is not obvious how to extract the remaining information in a similarly effective manner. To fill this gap, we introduce *conditional t-SNE* (ct-SNE), a generalization of t-SNE that discounts prior information from the embedding in the form of labels. To achieve this, we propose a conditioned version of the t-SNE objective, obtaining a single, integrated, and elegant method. ct-SNE has one extra parameter over t-SNE; we investigate its effects and show how to efficiently optimize the objective. Factoring out prior knowledge allows complementary structure to be captured in the embedding, providing new insights. Qualitative and quantitative empirical results on synthetic and (large) real data show ct-SNE is effective and achieves its goal.

3.1 Introduction

Dimensionality reduction (DR) methods can be used to create low-dimensional (typically 2-dimensional; 2-d) representations that are straightforward to visualize and subsequently explore the dominant structure of high-dimensional datasets. Non-linear DR methods are particularly powerful because they can capture complex structure even when it is spread over many dimensions. This explains the huge popularity of methods such as t-SNE [13], LargeVis [23], and UMAP [16].

However, DR methods yield a single static embedding and the most prominent structure present in the data may already be known to the analyst. One may indeed construct higher-dimensional embeddings, hoping to uncover more structure, but there is no guarantee that any of the constructed dimensions is fully complementary to the prior knowledge of an analyst. Besides, the salient structure that is already known may be spread across all attributes, hence we cannot just remove the associated attributes and generally speaking it is not obvious how to visualize the remaining structure. The question arises: can we actively filter or discount prior knowledge from the embedding?

To this end, we introduce *conditional t-SNE* (ct-SNE), a generalization of t-SNE that accounts for prior information about the data. By discounting the prior information, the embedding may focus on capturing complementary information. More concretely, it does not aim to construct an embedding that reflects all the proximities in the original data (the objective of t-SNE), but it should reflect all pairwise proximities *conditioned* on whether we expect that pair to be close or not.

ct-SNE enables at least three new ways to obtain insight into data:

- The prior knowledge may indeed be available beforehand, in which case we can straight away focus the analysis on an embedding that is more useful.
- Such prior knowledge may be gained during analysis, leading to an iterative data analysis process.
- If we observe some specific structure X in an embedding and then factor out specific information Y, then if X remains present in the embedding, we learn that X is Y complementary to Y.

Note we use the term *prior knowledge*, even when this knowledge is not available a priori, but gained during the analysis. This reflects the knowledge is available prior to the embedding step.

Example. To demonstrate the idea behind ct-SNE more concretely, consider a ten-dimensional dataset with 1,000 data points. In dimension 1–4 the data points fall into five clusters (following a multi-variate Gaussian with small variance), similarly for dimensions 5–6 the points fall randomly into four clusters. Dimensions

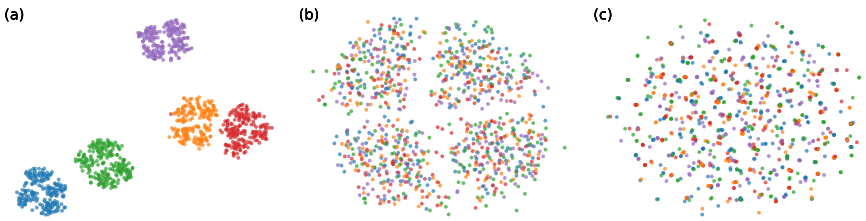

Figure 3.1: Visualization of 2-d embeddings of synthetic data (see 'example' below).

7–10 contain Gaussian noise with larger variance. Figure 3.1a gives the t-SNE embedding. It shows five large clusters, where some can be somewhat clearly split further into smaller clusters. The large clusters correspond to those defined in dimension 1–4. Figure 3.1b is the ct-SNE embedding where we have input the five colored clusters as prior knowledge. This figure shows four clusters that are complementary to the five clusters observed in 3.1a. We see they are complementary because there is no correlation between the colors and the clusters in Figure 3.1b. These four clusters are indeed those defined in dimensions 5–6. Finally, Figure 3.1c shows that after combining the labels, ct-SNE yields an embedding capturing only on the remaining noise.

The implementation of ct-SNE and code for the experiments on public data are available at `https://bitbucket.org/ghentdatascience/ct-sne`.

Contributions. This chapter makes the following contributions:

- ct-SNE, a new DR method that searches for an embedding such that a distribution defined in terms of distances in the input space (as in t-SNE) is well-approximated by a distribution defined in terms of distances in the embedding space *after conditioning on the prior knowledge* (Sec. 3.2.2);

- A Barnes-Hut-Tree based optimization method to efficiently find an embedding (Sec. 3.2.3);

- We illustrate that the concept of conditioning embeddings on prior information can be applied to other popular non-linear DR methods (mentioned in Sec. 3.2, with details in Appendix B);

- Extensive qualitative and quantitative experiments on synthetic and real world datasets show ct-SNE effectively removes the known factors, enables deeper visual analysis of high-dimensional data, and that ct-SNE scales sufficiently to handle hundreds of thousands of points (Sec. 3.3).

3.2 Method

In this section, we first briefly recap the idea behind t-SNE and introduce the basic notation. Then, we derive ct-SNE and describe a Barnes-Hut based strategy to optimize the ct-SNE objective. Due to space limitations, we discuss in Appendix B how the idea of factoring out prior information can be applied to many other existing non-linear DR methods such as LargeVis and UMAP.

3.2.1 Background: t-SNE

In t-SNE, the input data set $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is taken to define a probability distribution for a categorical random variable e, of which the value domain is indexed by all pairs (i, j) of indices $i, j \in [1..n]$ with $i \neq j$. This distribution is determined by specifying probabilities $0 \leq p_{ij} \leq 1$ such that $\sum_{i \neq j} p_{ij} = 1$, equal to the probability that $e = (i, j)$. For brevity, below we will speak of *the distribution* $\boldsymbol{p}$ when we mean the categorical distribution with parameters p_{ij}.

More specifically, in t-SNE, the distribution $\boldsymbol{p}$ is defined as follows:

$$p_{ij} \triangleq P_{\boldsymbol{p}}(e = (i, j)) = \frac{\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2 / 2\sigma^2\right)}. \tag{3.1}$$

The goal of t-SNE is to find another embedding $\boldsymbol{Y} \in \mathbb{R}^{n \times d'}$, from which another categorical probability distribution is derived, specified by the values q_{ij} defined as follows:

$$q_{ij} \triangleq P_{\boldsymbol{q}}(e = (i, j)) = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}}. \tag{3.2}$$

An embedding $\boldsymbol{Y}$ is deemed better if the distance between these two categorical distributions is smaller, as quantified by the KL-divergence: $KL(\boldsymbol{p} \| \boldsymbol{q}) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$.

3.2.2 Conditional t-SNE

Let us now assume that each data point $\boldsymbol{x}_i$ has a label l_i associated, with $l_i \in [0..L]$ for all $i \in [1..n]$. Moreover, let us assume that it is known a priori that same-labeled data points are more likely to be nearby in $\boldsymbol{X}$. Our goal is to ensure that the embedding $\boldsymbol{Y}$ does not reflect that information again. This can be achieved by minimizing the KL-divergence between the distributions $\boldsymbol{p}$ and $\boldsymbol{r}$ (rather than $\boldsymbol{q}$), where $\boldsymbol{r}$ is the distribution derived from the embedding $\boldsymbol{Y}$ but *conditioned on the prior knowledge*.

We formalize this using the following notation. The indicator variable $\delta_{ij} = 1$ if $l_i = l_j$ and $\delta_{ij} = 0$ if $l_i \neq l_j$, and the label matrix $\boldsymbol{\Delta}$ is defined by $[\boldsymbol{\Delta}]_{ij} = \delta_{ij}$.

The probability that the random variable e is equal to (i,j), *conditioned on* the label matrix $\mathbf{\Delta}$ (i.e. the prior information) is denoted as:

$$r_{ij} \triangleq P_{\boldsymbol{q}}(e=(i,j)|\mathbf{\Delta}) = \frac{P(\mathbf{\Delta}|e=(i,j))P_{\boldsymbol{q}}(e=(i,j))}{P_{\boldsymbol{q}}(\mathbf{\Delta})}.$$

In ct-SNE, this is the probability distribution that needs to be similar to $\boldsymbol{p}$ for the embedding to be a good one. Note that if we ensure that $P(\mathbf{\Delta}|e=(i,j))$ is larger when $\delta_{ij}=1$ than when $\delta_{ij}=0$, it will be less important for the embedding to ensure that $q_{ij}=P_{\boldsymbol{q}}(e=(i,j))$ is large for same-labeled data points, even if p_{ij} is large. I.e., *for same-labeled data points*, it is less important to be embedded nearby even if they are nearby in the input representation. This is precisely the goal of ct-SNE.

To compute $P_{\boldsymbol{q}}(e=(i,j)|\mathbf{\Delta})$, we now investigate its different factors. First, $P_{\boldsymbol{q}}(e=(i,j))=q_{ij}$ is simply computed as in Eq. (3.2). Second, we need to determine a suitable form for $P(\mathbf{\Delta}|e=(i,j))$, based on the above intuition. To do this, we assume that δ_{ij} is the sufficient statistic for $P(\mathbf{\Delta}|e=(i,j))$, i.e. $P(\mathbf{\Delta}|e=(i,j))=\alpha^{\delta_{ij}}\beta^{1-\delta_{ij}}$, where α and β can be regarded as the confidence of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ being randomly picked to have the same or different labels. Let us further denote the class size of the l'th class as n_l. Then, for this distribution to be normalized, it must hold that:

$$\begin{aligned}
1=&\sum_{\mathbf{\Delta}} P(\mathbf{\Delta}|e=(i,j)),\\
=&\alpha\left(\sum_l \frac{(n-2)!}{(n_l-2)!\prod_{f\neq l} n_f!}\right)+\beta\left(\frac{n!}{\prod_l n_l!}-\sum_l \frac{(n-2)!}{(n_l-2)!\prod_{f\neq l} n_f!}\right),\\
=&\frac{n!}{\prod_l n_l!}\left(\alpha\frac{\sum_l n_l(n_l-1)}{n(n-1)}+\beta\left(1-\frac{\sum_l n_l(n_l-1)}{n(n-1)}\right)\right).
\end{aligned}$$

This yields a relation between α and β. It also suggests a ballpark figure for α. Indeed, one would typically set $\alpha>\beta$. For $\alpha=\beta$ (i.e. the lower bound for α), they would both be equal to $\alpha=\beta=\frac{\prod_l n_l!}{n!}$, i.e. one divided by the number of possible distinct label assignments (this is of course entirely logical). Thus, in tuning α, one could take multiples of this minimal value.

We can now also compute the marginal probability $P_{\boldsymbol{q}}(\mathbf{\Delta})$ as follows:

$$\begin{aligned}
P_{\boldsymbol{q}}(\mathbf{\Delta})=&\sum_{i\neq j} P(\mathbf{\Delta}|e=(i,j))P_{\boldsymbol{q}}(e=(i,j))=\sum_{i\neq j} q_{ij}\alpha^{\delta_{ij}}\beta^{1-\delta_{ij}},\\
=&\alpha\sum_{i\neq j:\delta_{ij}=1} q_{ij}+\beta\sum_{i\neq j:\delta_{ij}=0} q_{ij}.
\end{aligned}$$

Given all this, one can then compute the required conditional distribution as fol-

lows:

$$r_{ij} \triangleq P_{\boldsymbol{q}}(e=(i,j)|\boldsymbol{\Delta}) = \frac{P(\boldsymbol{\Delta}|e=(i,j))P_{\boldsymbol{q}}(e=(i,j))}{P_{\boldsymbol{q}}(\boldsymbol{\Delta})}, \tag{3.3}$$
$$= \begin{cases} \frac{\alpha q_{ij}}{\alpha \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta \sum_{i\neq j:\delta_{ij}=0} q_{ij}} & \text{if} \quad \delta_{ij}=1, \\ \frac{\beta q_{ij}}{\alpha \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta \sum_{i\neq j:\delta_{ij}=0} q_{ij}} & \text{if} \quad \delta_{ij}=0. \end{cases}$$

It is numerically better to express this in terms of new variables $\alpha' \triangleq \alpha \frac{n!}{\prod_l n_l!}$ and $\beta' \triangleq \beta \frac{n!}{\prod_l n_l!}$:

$$r_{ij} = \begin{cases} \frac{\alpha' q_{ij}}{\alpha' \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta' \sum_{i\neq j:\delta_{ij}=0} q_{ij}} & \text{if} \quad \delta_{ij}=1, \\ \frac{\beta' q_{ij}}{\alpha' \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta' \sum_{i\neq j:\delta_{ij}=0} q_{ij}} & \text{if} \quad \delta_{ij}=0, \end{cases}$$

where the relation between α' and β' is:

$$1 = \alpha' \frac{\sum_l n_l(n_l-1)}{n(n-1)} + \beta' \left(1 - \frac{\sum_l n_l(n_l-1)}{n(n-1)}\right). \tag{3.4}$$

This has the advantage of avoiding the large factorials and resulting numerical problems. The lower bound for α' to be considered is now 1 (in which case also $\beta'=1$).

Finally, computing the KL-divergence with $\boldsymbol{p}$, yields the ct-SNE objective function to be minimized:

$$\begin{aligned} KL(\boldsymbol{p}\|\boldsymbol{r}) =& \sum_{i\neq j} p_{ij} \log\left(\frac{p_{ij}}{r_{ij}}\right), \\ =& KL(\boldsymbol{p}\|\boldsymbol{q}) + \sum_{i\neq j} p_{ij} \log\left(\frac{\alpha' \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta' \sum_{i\neq j:\delta_{ij}=0} q_{ij}}{\alpha'^{\delta_{ij}} \beta'^{1-\delta_{ij}}}\right), \\ =& KL(\boldsymbol{p}\|\boldsymbol{q}) + \log\left(\alpha' \sum_{i\neq j:\delta_{ij}=1} q_{ij} + \beta' \sum_{i\neq j:\delta_{ij}=0} q_{ij}\right) \\ & - \sum_{i\neq j:\delta_{ij}=1} p_{ij} \log(\alpha') - \sum_{i\neq j:\delta_{ij}=0} p_{ij} \log(\beta'). \end{aligned} \tag{3.5}$$

Note that the last two terms are constant w.r.t. q_{ij}. Moreover, it is clear that for $\alpha'=\beta'=1$, this reduces to standard t-SNE. For $\alpha' > 1 > \beta'$ (and related as per the Eq. (3.4)), the minimization of this KL-divergence will try to minimize q_{ij} when $\delta_{ij}=1$ more strongly (as it is multiplied with the larger number α') than when $\delta_{ij}=0$ (when it is multiplied with the smaller number β').

3.2.3 Optimization

The objective function (Eq. (3.5)) is non-convex w.r.t the embedding $\boldsymbol{Y}$. Even so, we find that optimizing the objective function using gradient descent with random restarts works well in practice. The gradient of the objective function w.r.t. the embedding of a point $\boldsymbol{y}_i$ reads: [1]

$$\begin{aligned}\nabla_{\boldsymbol{y}_i} KL(\boldsymbol{p}\|\boldsymbol{r}) &= 4\left(F_{\text{attr}} + F_{\text{rep}}\right), \\ &= 4\sum_j \left(p_{ij}q_{ij}Z(\boldsymbol{y}_i - \boldsymbol{y}_j) - \frac{\delta_{ij}\alpha' + (1-\delta_{ij})\beta'}{O}\cdot q_{ij}^2 Z(\boldsymbol{y}_i - \boldsymbol{y}_j)\right).\end{aligned}$$

where $Z = \sum_{k\neq l}(1+\|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}$ and $O = \alpha' \sum_{i\neq j:\delta_{kl}=1} q_{kl} + \beta' \sum_{i\neq j:\delta_{kl}=0} q_{kl}$. The gradient can be decomposed in attraction and repelling forces between points in the embedding space. Thus the underlying problem of ct-SNE, just like many other force-based embedding methods, is related to the classic n-body problem in physics[2], which has also been studied in the recent machine learning literature [8, 21]. The general goal of the n-body problem is to find a constellation of n objects such that equilibrium is achieved according to a certain measure (e.g., forces, energy). In the problem setting of ct-SNE, both the pairwise distances between points and the label information affect the attraction and repelling forces. Particularly, the label information strengthens the repelling force (assume $\alpha' > 1 > \beta' > 0$) between two points if they have the same label and weakens the repelling force if two points have different labels. This is desirable behavior because we do not want to reflect the known label information in the resulted embeddings.

Evaluating the gradient has complexity $\mathcal{O}(n^2)$, which makes the computation (both time and memory cost) infeasible when n is large (e.g., $n > 100k$). As an approximation of the gradient computation, we adapt the tree-based approximation strategy described in van der Maaten [27]. To efficiently model the proximity in high-dimensional space (Eq. (3.1)) we use a vantage-point tree-based algorithm (which exploits the fast diminishing property of the Gaussian distribution). To approximate the low-dimensional proximity (Eq. (3.3)) we modify the Barnes-Hut algorithm to incorporate the label information. The basic idea of the Barnes-Hut algorithm is to organize the points in the embedding space using a kd-tree (which for 2-d embeddings is equivalent to a quad tree). Each node of the tree corresponds to a cell (dissection) in the embedding space. If a target point $\boldsymbol{y}_i$ is far away from all the points in a given cell, then the interaction between the target point and the points within the cell can be summarized by the interaction between $\boldsymbol{y}_i$ and the cell's center of mass $\boldsymbol{y}_{\text{cell}}$ that is computed while constructing the kd-tree. More specifically, the summarization happens when $r_{\text{cell}}/\|\boldsymbol{y}_i - \boldsymbol{y}_{\text{cell}}\|^2 < \theta$,

[1] A detailed derivation of the gradient computation can be found in Appendix A.

[2] https://en.wikipedia.org/wiki/N-body_problem#Other_n-body_problems

where r_{cell} is the radius of the cell, while θ controls the strength of summarization, i.e. the approximation strength. The summarized repelling force in t-SNE reads $F_{\text{rep}} = -n_{\text{cell}} q_{i,\text{cell}}^2 Z(\boldsymbol{y}_i - \boldsymbol{y}_{cell})$, where n_{cell} is the number of data points in that cell.

In the ct-SNE approximation, we had to overcome an additional complication though: we also need to summarize the label information for the points in a cell when the summarization happens. This can be done by maintaining a histogram in each cell, and counting the numbers of data points with different labels that fall into that cell. Then the repelling force of a target point $\boldsymbol{y}_i$ can be weighted proportional to the number of points that have same (different) label(s) within the cell. Namely:

$$F_{\text{rep}}^{\text{approx.}} = -\frac{\alpha' n_{\text{cell},l=l_i} + \beta'(n_{\text{cell}} - n_{\text{cell},l=l_i})}{O} q_{i,\text{cell}}^2 Z(\boldsymbol{y}_i - \boldsymbol{y}_{cell}),$$

where $n_{\text{cell},l=l_i}$ is the number of data points in a cell that has the same label as point $\boldsymbol{y}_i$.

As both tree-based approximation schemes have complexity $\mathcal{O}(n \log n)$, counting the label will add an extra multiplicative constant L, equal to the number of label values in the prior information. Thus the final complexity of approximated ct-SNE is $\mathcal{O}(L \cdot n \log n)$.

3.3 Experiments

The experiments investigate 4 questions: **Q1** Does ct-SNE work as expected in finding complementary structure? **Q2** How should α (or equivalently, β) be chosen? **Q3** Could ct-SNE's goal be achieved also by using (a combination of) other methods? **Q4** How well does ct-SNE scale? Two case studies addressing **Q1** are presented in Sections 3.3.1–3.3.3. Two more case studies addressing **Q1** as well as the experiments addressing **Q2–Q4** are summarized in Sec. 3.3.4, and described in detail in Appendix C.

3.3.1 Datasets used, and experimental settings

The first dataset is a **Synthetic dataset** consisting of 1000 ten-dimensional data points, as explained in Section 3.1. The second is a **Facebook dataset** consisting of 128-dimensional embedding of a de-identified random sample of $500k$ Facebook users in the US. This embedding is generated based purely on the list of pages and groups that the users follow, as part of an effort to improve the quality of several recommendation systems at Facebook.

To study **Q1**, both qualitative and quantitative experiments were performed on the synthetic dataset. On the Facebook dataset we only conducted a qualitative evaluation (given the lack of ground truth).

Qualitative experiment. We qualitatively evaluate the effectiveness of ct-SNE through visualizations. More specifically, we compare the t-SNE visualization of a dataset with the ct-SNE visualization that has taken into account certain prior information that is visually identifiable from the t-SNE embedding. Thus by inspecting the presence of the prior information in the ct-SNE embedding and comparing to the t-SNE embedding, we can evaluate whether the prior information is removed. Conversely, we test whether information present in the ct-SNE embedding could have been identified from the t-SNE embedding to verify whether it indeed contains complementary information.

To select the prior information, we first visualize the t-SNE embedding and manually select points that are clustered in the visualization. Then we perform a *feature ranking* procedure to identify the features that separate the selected points from the rest. This is done by fitting a linear classifier (logistic regression) on the selected cluster against all other data points. By inspecting the weights of the classifier, we can identify the feature that contributes the most to the classifier. Repeating this *feature ranking* procedure for other clusters, we aim to find a feature that correlates with the majority of the clusters in the t-SNE visualization. This feature is then treated as prior information and provided as input to ct-SNE. In the reported experiments, the most prominent feature was always categorical, so all points with the same value were treated as being in a cluster to define the prior. We apply exact ct-SNE on Synthetic and approximated ct-SNE ($\theta = 0.5$) on the Facebook dataset.

We also evaluated whether ct-SNE can continuously provide new insights, by repeatedly applying the cluster selection and feature ranking procedure on ct-SNE embeddings.

Quantitative experiment. In this experiment, we quantify the presence of certain prior information in a ct-SNE embedding that also take the same prior information as input. For example, if ct-SNE encodes the prior information using labels, the strong presence of certain prior information is equivalent to the high homogeneity of the encoded labels in the embedding, i.e., points that are close to each other in the embedding often have the same label. To quantify such homogeneity, we developed a measure termed *normalized Laplacian score* defined as follows. Given an embedding $\boldsymbol{Y}$ and parameter k, we denote $\boldsymbol{A}_k$ as the adjacency matrix of the k-nearest graph computed from the embedding. Then, the Laplacian matrix of the kNN graph has the form $\boldsymbol{L}_k = \boldsymbol{A}_k - \boldsymbol{D}_k$ where $\boldsymbol{D}_k = \text{diag}(\text{sum}(\boldsymbol{A}_k, 1))$. We further normalize the Laplacian matrix ($\boldsymbol{D}_k^{-1/2}\boldsymbol{L}_k\boldsymbol{D}_k^{-1/2}$) to obtain a score that is insensitive to the node degrees. Given a label vector $\boldsymbol{f}$ with L values where each label l has n_l points, and denote the one-hot encoding for each label l as $\boldsymbol{f}_l$, then the normalized Laplacian score can be computed as:

$$\sum_{l \in [0..L]} \frac{n_l}{n} \boldsymbol{f}_l' \boldsymbol{D}_k^{-1/2} \boldsymbol{L}_k \boldsymbol{D}_k^{-1/2} \boldsymbol{f}_l. \tag{3.6}$$

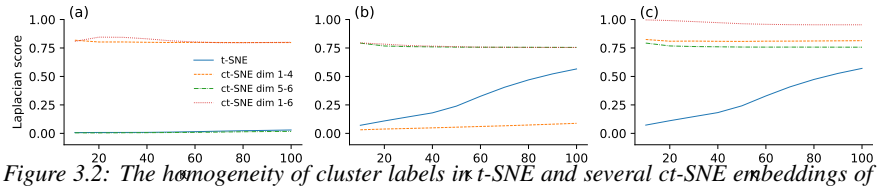

Figure 3.2: The homogeneity of cluster labels in t-SNE and several ct-SNE embeddings of the synthetic dataset for k (a parameter of the Laplacian score) ranging from 10 *to* 100*, for the three label sets: (a) $\boldsymbol{f}_{1-4}$, (b) $\boldsymbol{f}_{5-6}$, and (c) $\boldsymbol{f}_{1-6}$. Colored lines give the scores for different embeddings: t-SNE (blue), ct-SNE with prior $\boldsymbol{f}_{1-4}$ (orange), ct-SNE with prior $\boldsymbol{f}_{5-6}$ (green), and ct-SNE with prior $\boldsymbol{f}_{1-6}$ (red).*

This score is essentially the pairwise difference (in terms of labels) between the data points that are connected according to the kNN graph. If a label is locally consistent (homogeneous) in an embedding, the feature difference among the kNN graph neighborhood is small, which results in a small Laplacian score. Conversely, a less homogeneous label over the kNN graph would have a large Laplacian score. Thus, if ct-SNE removes certain prior information from its embedding, then the embedding should have a large Laplacian score on the labels that encode the prior information.

3.3.2 Case study: Synthetic dataset

Qualitative experiment. The t-SNE visualization of the synthetic dataset shows five large clusters (Fig. 3.1a). Feature ranking (Sec. 3.3.1) shows these clusters correspond to the clustering in dimensions 1-4 of the data. Taking the cluster labels in dimensions 1-4 ($\boldsymbol{f}_{1-4}$) as prior, ct-SNE gives a different visualization (Fig. 3.1b). The feature ranking further shows the ct-SNE embedding indeed reveals the clusters in the dimension 5-6 of the data. We further combine the labels $\boldsymbol{f}_{1-4}$ and $\boldsymbol{f}_{5-6}$ by assigning a new label to each combinations of the label in $\boldsymbol{f}_{1-4}$ and $\boldsymbol{f}_{5-6}$, denoted as $\boldsymbol{f}_{1-6}$. ct-SNE with $\boldsymbol{f}_{1-6}$ yields an embedding based only on the remaining noise (Fig. 3.1c).

Quantitative experiment. We computed the normalized Laplacian scores (Eq. (3.6)) of the t-SNE and several ct-SNE embeddings. Subfigures in Fig. 3.2a–c give the Laplacian score for three label sets: $\boldsymbol{f}_{1-4}$, $\boldsymbol{f}_{5-6}$, and $\boldsymbol{f}_{1-6}$. Fig. 3.2a shows that labels $\boldsymbol{f}_{1-4}$ are less homogeneous (higher Laplacian score) in the ct-SNE embeddings with prior $\boldsymbol{f}_{1-4}$ and $\boldsymbol{f}_{1-6}$ than in the t-SNE embedding, indicating that ct-SNE effectively discounted the prior from the embeddings. Both the t-SNE embedding and ct-SNE with prior $\boldsymbol{f}_{5-6}$ clearly pick up the cluster in $\boldsymbol{f}_{1-4}$, as indicated by the very low Laplacian score. Similarly, Figures 3.2b,c show that ct-SNE removes the prior information effectively for labels $\boldsymbol{f}_{5-6}$ and $\boldsymbol{f}_{1-6}$, respectively, given the associated priors.

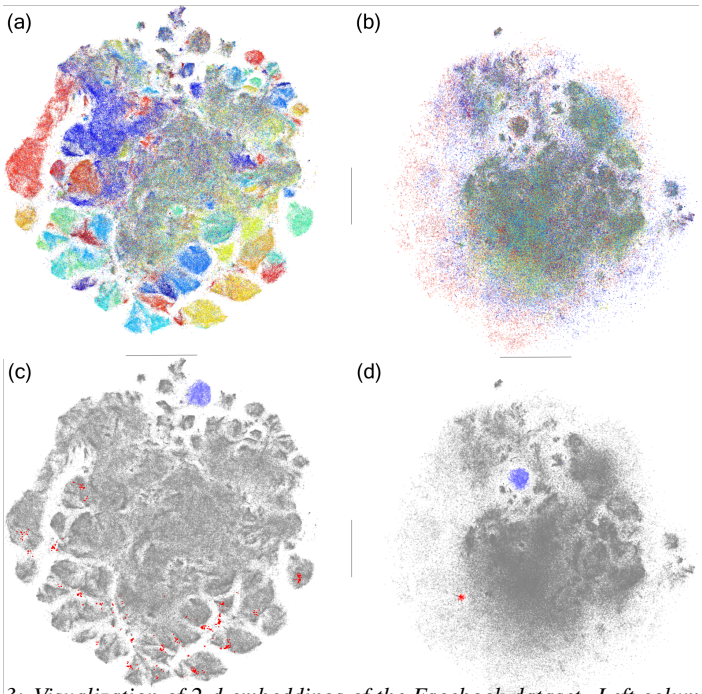

Figure 3.3: Visualization of 2-d embeddings of the Facebook dataset. Left column: t-SNE embedding, right column: ct-SNE embedding with region as prior. The two rows show identical embeddings but with different cluster markings (colors). See Section 3.3.3 for further info.

3.3.3 Case study: Facebook dataset

Qualitative experiment. Applying t-SNE on the Facebook dataset gives a visualization with many visually salient clusters (Fig. 3.3a). Computing the feature ranking for classification of selected clusters shows that the geography (i.e., the states) contributes to the embedding the most. This is further confirmed by coloring the data points according to the geographical region in the visualization as shown in Fig. 3.3a: most of the clusters are indeed quite homogeneous with respect to geography.

To understand the effect of an embedding like this in a downstream recommendation system, an analyst would want to know what type of user interests the embedding is capturing. For this, the regional clusters are not very informative. To alleviate that we can encode the region as prior for ct-SNE so that other interesting structures can emerge in the visualization. Using the same coloring scheme, ct-SNE shows a cluster with large mass that consists of users from different states (Fig. 3.3b). There are also a few small clusters with mixed color scattered on the periphery of the visualization. The visualization indicates that geographical infor-

mation is mostly removed in the ct-SNE embedding. This is further confirmed by selecting clusters (highlighted in red color) in ct-SNE embedding (Fig. 3.3d) and highlighting the same set of points in the t-SNE embedding (Fig. 3.3c). The cluster highlighted in the ct-SNE embedding spreads over the t-SNE embedding, indicating these users are not geographically similar. Indeed, feature ranking (Sec. 3.3.1) indicates that the selected group of users (Fig. 3.3d) share an interest in horse riding: they tend to follow several pages related to that topic. Interestingly, we noticed that some of the clusters in the ct-SNE embedding are also clustered in the t-SNE embedding. These clusters are indeed not homogeneous in terms of the geographical regions. For example, the cluster highlighted in blue in the ct-SNE embedding (Fig. 3.3d) also exists in the t-SNE embedding (Fig. 3.3c). Using feature ranking as above we found that these clusters are not homogeneous in terms of geography, but in terms of users' interest in Indian culture. While these clusters can thus also be seen in the t-SNE embedding, ct-SNE removes the irrelevant (region) cluster structure, such that those other clusters become more salient and easy to observe.

3.3.4 Summary of additional experimental findings

Two other case studies (App. C–C) on the UCI adult dataset [4] and a DBLP citation network dataset [24] confirm the ability of ct-SNE visualizations to reveal insightful clusters after conditioning on prior information that dominates the t-SNE visualizations (**Q1**). In Appendix C we also analyzed the sensitivity of the ct-SNE embedding with respect to the hyperparameter α' (or β') (**Q2**). By varying the hyperparameter, we found ct-SNE yields low-dimensional embeddings that better approximate the original data than t-SNE (i.e., smaller KL-divergence). The analysis also shows that using a small β' (e.g., $\beta' = 0.01$) is a good rule of thumb when using ct-SNE for visualization. To answer **Q3**, we compared ct-SNE to two non-trivial baselines that remove the known factors from the high-dimensional data using either an adversarial auto-encoder (AAE [15]) or canonical correlation analysis (CCA [10]) and then apply t-SNE for visualization (App. C). We show that these baselines are either difficult to tune (AAE-based baseline) or have limited applicability (CCA-based baseline), while ct-SNE has essentially only one parameter to tune, and does not suffer from the limitations of the CCA baseline. Finally, we conducted a runtime experiment (App. C) showing that the approximated ct-SNE can efficiently embed large, high-dimensional data, without substantial quality loss (**Q4**).

3.4 Related work

Many dimensionality reduction methods have been proposed in the literature. Arguably, n-body problem based methods such as MDS [26], Isomap [25], t-SNE

[13], LargeVis [23], and UMAP [16] appear to be the most popular ones. These methods typically have three components: (1) a proximity measure in the input space, (2) a proximity measure in the embedding space, (3) a loss function comparing the proximity between data points in the embedding space with the proximity in the input space. ct-SNE belongs to this class of DR methods. It accepts both high-dimensional data and priors about the data as inputs, and searches for low-dimensional embeddings while discounting structure in the input data specified as prior knowledge.

As a core component of ct-SNE is the prior information specified by the user, it can be considered an interactive DR method. Closely related to ct-SNE, there is a group of interactive DR methods that adjust the algorithms according to a user's inputs [e.g., 12, 20, 5, 1, 2, 17]. These methods contrast with ct-SNE in that the user feedback must be obeyed in the output embedding, while for ct-SNE the prior knowledge defined by the user guides what is irrelevant to the user.[3]

3.5 Conclusion

We introduce conditional t-SNE to efficiently discover *new* insights from high-dimensional data. ct-SNE finds the lower dimensional representation of the data in a non-linear fashion while removing the known factors. Extensive case studies on both synthetic and real-world datasets demonstrate that ct-SNE can effectively remove known factors from low-dimensional representations, allowing new structure to emerge and providing new insights to the analyst. A tree-based optimization method allows ct-SNE to scale to a high dimensional dataset with hundreds of thousands of data points.

Acknowledgement

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the FWO (project no. G091017N, G0F9816N), from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501, and from the EPSRC (SPHERE EP/R005273/1). We thank Laurens van der Maaten for helpful discussions.

[3] For an extended discussion about the related work, please refer to Appendix D.

Appendices

A Detailed derivation of the gradient of the ct-SNE objective function

Here we derive in detail the gradient of the ct-SNE objective function. Denote the euclidean distance between points as $d_{ij} \triangleq \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$. The derivative of d_{ij} with respect to embedding $\boldsymbol{y}_i$ reads:

$$\nabla_{\boldsymbol{y}_i} d_{ij} = \frac{\boldsymbol{y}_i - \boldsymbol{y}_j}{d_{ij}}.$$

Denote the cost (KL-divergence) by C:

$$\begin{aligned} C &= KL(\boldsymbol{p}\|\boldsymbol{r}) \\ &= C_1 + C_2 - \sum_{k\neq l:\delta_{kl}=1} p_{kl}\log(\alpha') - \sum_{i\neq j:\delta=0} p_{kl}\log(\beta'), \end{aligned}$$

where

$$C_1 = KL(\boldsymbol{p}\|\boldsymbol{q}),$$

and

$$C_2 = \log\left(\alpha' \sum_{k\neq l:\delta_{kl}=1} q_{kl} + \beta' \sum_{k\neq l:\delta_{kl}=0} q_{kl}\right).$$

Following the derivation from t-SNE paper, the derivative of C_1 with respect to $\boldsymbol{y}_i$ reads:

$$\nabla_{\boldsymbol{y}_i} C_1 = 4\sum_j (p_{ij} - q_{ij})(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1}(\boldsymbol{y}_i - \boldsymbol{y}_j).$$

To compute the derivative of C_2 with respect to $\boldsymbol{y}_i$, we first have:

$$\nabla_{\boldsymbol{y}_i} C_2 = 2\sum_j \frac{\partial C_2}{\partial d_{ij}} \cdot \frac{\boldsymbol{y}_i - \boldsymbol{y}_j}{d_{ij}}.$$

Denote $O = \alpha' \sum_{i \neq j : \delta_{kl}=1} q_{kl} + \beta' \sum_{i \neq j : \delta_{kl}=0} q_{kl}$The derivative of C_2 with respect to d_{ij} can be computed as:

$$
\begin{aligned}
\frac{\partial C_2}{\partial d_{ij}} &= \frac{1}{O} \frac{\partial}{\partial d_{ij}} \left(\alpha' \sum_{k \neq l : \delta_{kl}=1} q_{kl} + \beta' \sum_{k \neq l : \delta_{kl}=0} q_{kl} \right) \\
&= \frac{1}{O} \left(\alpha' \sum_{k \neq l : \delta_{kl}=1} \frac{\partial q_{kl}}{\partial d_{ij}} + \beta' \sum_{k \neq l : \delta_{kl}=0} \frac{\partial q_{kl}}{\partial d_{ij}} \right) \\
&= \frac{1}{O} \Bigg(\alpha' \Big(-2\delta_{ij} q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} + 2 \sum_{k \neq l : \delta_{kl}=1} q_{kl} q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} \Big) \\
&+ \beta' \Big(-2(1 - \delta_{ij}) q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} + 2 \sum_{k \neq l : \delta_{kl}=0} q_{kl} q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} \Big) \Bigg) \\
&= \frac{1}{O} \Bigg(2\alpha' \Big(-\delta_{ij} + \sum_{k \neq l : \delta_{kl}=1} q_{kl} \Big) q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} \\
&+ 2\beta' \Big(-(1 - \delta_{ij}) + \sum_{k \neq l : \delta_{kl}=0} q_{kl} \Big) q_{ij} (1 + d_{ij}^2)^{-1} d_{ij} \Bigg) \\
&= 2 \left(1 - \frac{\delta_{ij}\alpha' + (1 - \delta_{ij})\beta'}{O} \right) \cdot q_{ij} (1 + d_{ij}^2)^{-1} d_{ij}.
\end{aligned}
$$

Thus we have derivative of C_2 with respect to $\boldsymbol{y}_i$

$$
\nabla_{\boldsymbol{y}_i} C_2 = 4 \sum_j \left(1 - \frac{\delta_{ij}\alpha' + (1 - \delta_{ij})\beta'}{O} \right) \cdot q_{ij} (1 + d_{ij}^2)^{-1} \cdot (\boldsymbol{y}_i - \boldsymbol{y}_j).
$$

Finally, we have derivative:

$$
\begin{aligned}
\nabla_{\boldsymbol{y}_i} C &= \nabla_{\boldsymbol{y}_i} C_1 + \nabla_{\boldsymbol{y}_i} C_2 \\
&= 4 \sum_j \left(p_{ij} - \frac{\delta_{ij}\alpha' + (1 - \delta_{ij})\beta'}{O} \cdot q_{ij} \right) \cdot (1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)^{-1} (\boldsymbol{y}_i - \boldsymbol{y}_j).
\end{aligned}
$$

B On generalizing the idea of ct-SNE

The idea of removing known factors from low-dimensional representations can be generalized to other n-body problem based DR methods. Oftentimes, the gradient of the n-body problem based methods can be viewed as a summation of attraction forces and repelling forces. Removing a known factor thus amounts to re-weighting the attracting and repelling forces such that points that have the same label repel each other and points with different labels attract each other. For example, LargeVis [23] differs from t-SNE by modeling input space proximity using

random KNN graph. Thus we can use the same conditioning idea as in ct-SNE to remove the known factors in LargeVis. However, for Uniform Manifold Approximation and Projection (UMAP) [16], conditioning is not readily applicable. In contrast to t-SNE, UMAP uses fuzzy sets to model the proximity in both input space and embedding space. Then the cross entropy between two fuzzy sets serves as loss function to compare the modeled proximity between input space and the embedding space. In the UMAP setting, it is not straightforward to condition the lower dimensional proximity model on the prior. But we can still directly reweight the repelling forces: for data points with the same label, the pushing effect is strengthened by α; for samples with different labels, the pushing effect is weakened by multiplying with β, with assumption $\alpha > 1 > \beta > 0$. However, without proper conditioning, parameter α and β loose their probabilistic interpretation and along with it their one-to-one correspondence (as in ct-SNE), thus both parameters α and β need to be set.

C Extended experiments

Datasets

In this section, we introduce two additional datasets:

UCI Adult dataset. We sampled 1000 data points from the UCI adult dataset [4] with six attributes: the three numeric attributes *age*, *education level*, and *work hours per week*, and the three binary attributes *ethnicity* (white/other), *gender*, and *income (>50k)*.

DBLP dataset. We extracted all papers from 20 venues[4] in four areas (ML/DM/DB/IR) of computer science from the DBLP citation network dataset [24]. We sampled half of the papers and constructed a network ($122,962$ nodes[5]) based on paper-author, paper-topic, paper-venue relations. Finally, we embedded the network into a 64 dimensional euclidean space using node2vec [9] with walk length 80, window size 10. In our experiment, both p and q are set to 1. Under this setting, node2vec is equivalent to DeepWalk [18].

Case study: UCI Adult dataset

Qualitative experiment. Fig. 4a shows t-SNE gives an embedding that consists of clusters grouped according to combinations of three attributes: *gender*, *ethnicity* and *income (>50k)*. By incorporating the attribute *gender* as prior, the ct-SNE embedding (Fig. 4b) contains clusters with a mixture of *male* and *female* points,

[4]These venues are: NIPS, ICLR, ICML, AAAI, IJCAI, KDD, ECML-PKDD, ICDM, SDM, WSDM, PAKDD, VLDB, SIGMOD, ICDT, ICDE, PODS, SIGIR, WWW, CIKM, ECIR.

[5]The network consists of $43,346$ paper nodes, $63,446$ author nodes, $16,150$ topic nodes and 20 venue nodes.

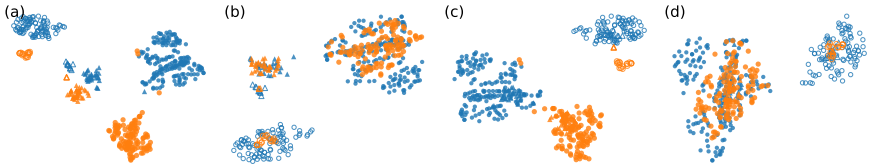

Figure 4: Visualization of 2-d embeddings of the UCI Adult dataset. Points are visually encoded according to their attributes. gender*:* female *(orange color),* male *(blue color);* ethnicity*:* white *(circle),* other *(triangle);* income (>50k)*:* true *(unfilled marker),* false *(filled marker). (a) t-SNE embedding shows clusters that are grouped according to the combinations of all three attributes. (b) With attribute* gender *as prior, ct-SNE embedding shows four clusters each has a mixture of points with different genders, indicating the* gender *information is removed. (c) With attribute* ethnicity *as prior, ct-SNE embedding also shows four clusters but each has a mixture of points with different ethnicities. (d) Incorporating the combination of attributes* gender *and* ethnicity *as prior, the resulted ct-SNE embedding shows two clusters that are correlated with the remaining attribute:* income (>50k).

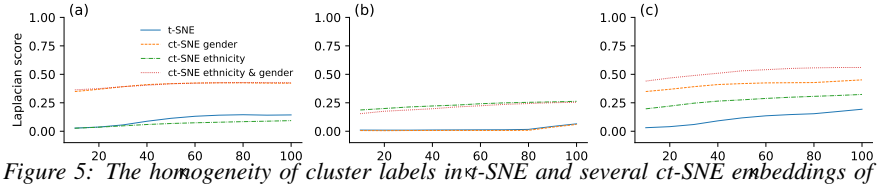

Figure 5: The homogeneity of cluster labels in t-SNE and several ct-SNE embeddings of the UCI Adult dataset for k (a parameter of the Laplacian score) ranging from 10 *to* 100 *with step size* 10*. Colored lines correspond to scores for different embeddings: t-SNE (blue), ct-SNE with prior* gender *(orange), ct-SNE with prior* ethnicity *(green), and ct-SNE with prior* ethnicity & gender *(red). Subfigures give homogeneity scores for various labels: (a)* gender *(b)* ethnicity *(c)* gender & ethnicity*. (a) The attribute* gender *has lower homogeneity (high Laplacian score) in the ct-SNE embedding with* gender *or* ethnicity & gender *as prior than in t-SNE embedding and ct-SNE embedding with* ethnicity *as prior. (b) The attribute* ethnicity *has lower homogeneity in the ct-SNE embedding with* ethnicity *or* ethnicity & gender *as priors than in the t-SNE embedding and ct-SNE with* gender *as prior) embeddings. (c) The attribute* ethnicity & gender *has high homogeneity in the t-SNE embedding only.*

indicating the *gender* information is removed. Instead, by incorporating the attribute *ethnicity* the ct-SNE embedding (Fig. 4c) contains clusters with a mixture of ethnicities. Finally, incorporating the combination of attributes *gender* and *ethnicity* as prior, the ct-SNE embedding contains data points grouped according to *income* (Fig. 4d).

Quantitative experiment. We analyzed the homogeneities (Laplacian scores) of attributes *gender*, *ethnicity* and *income (>50k)* measured on both t-SNE and ct-SNE embeddings. Fig. 5a shows ct-SNE with prior *gender* removes the *gender* factor from the resulted embedding, while ct-SNE with prior *ethnicity* makes the *gender* factor in the resulted embedding clearer. Similarly, Figure. 5b,c show

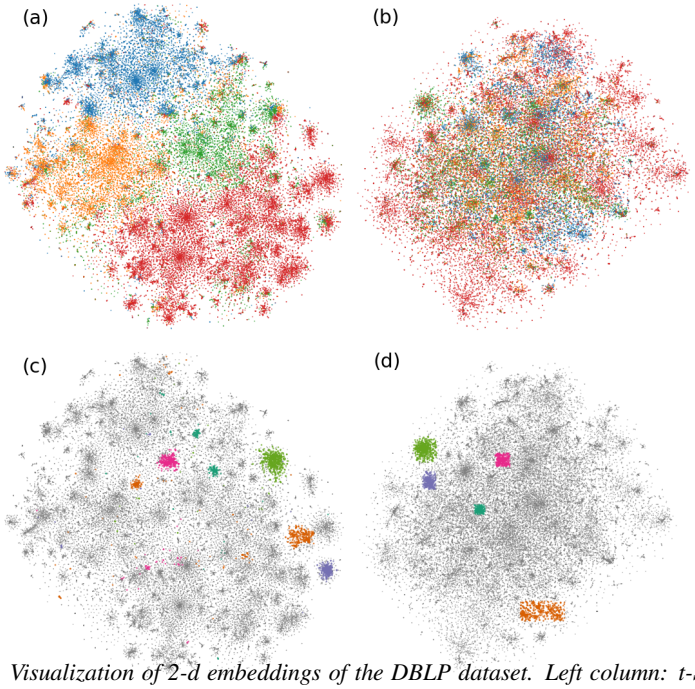

Figure 6: Visualization of 2-d embeddings of the DBLP dataset. Left column: t-SNE embedding, right column: ct-SNE embedding with area as prior. The rows contains different cluster markings. (a) t-SNE embedding shows a clustering according to four areas in computer science (red - machine learning, green - data mining, blue - data base, orange - information retrieval). (b) ct-SNE embedding shows a different clustering, with area information removed. (d) Newly emerged visual clusters (magenta - topic 'privacy', dark green - topic 'data stream', orange - topic 'computer vision') in ct-SNE embedding spread over in the t-SNE embedding (c). (d) Clusters (grass green - topic 'clustering', purple - topic 'active leraning') stood-out in the ct-SNE embedding also exists in the t-SNE embedding (c). These are a few out of many clusters that we found to exhibit a much more informative, interest-centric structure than the t-SNE projection.

ct-SNE removes the prior information effectively for labels *ethnicity* and *ethnicity&gender* respectively, given the associated priors.

Case study: DBLP dataset

Qualitative experiment. Applying t-SNE on the DBLP dataset gives a visualization with many visual clusters (Fig. 6a). Feature ranking for classification of the selected clusters shows the topics that contribute the most to the visualization. Moreover, we used mpld3[6] (an interactive visualization library) to inspect (i.e., hovering over data points and check tooltips) the metadata of t-SNE plot. Upon

[6] `https://mpld3.github.io`

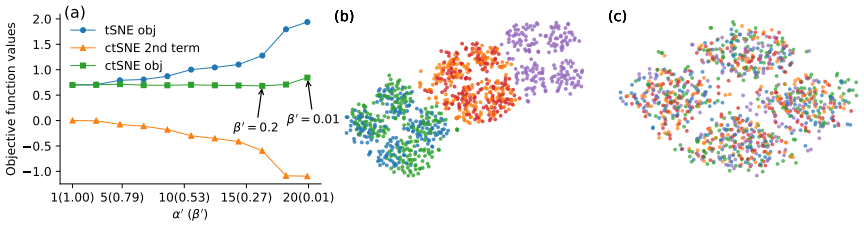

Figure 7: Visualizing the effect of different β's (α's) have on the ct-SNE embeddings. The embeddings are computed on the synthetic dataset with the prior information to be the cluster labels in dimensions 1-4*. (a) The values of ct-SNE objective (green), t-SNE objective (blue), and ct-SNE prior term (orange) against different β's. ct-SNE achieves smaller KL-divergence than t-SNE. (b) ct-SNE embedding with $\beta' = 0.2$ has smallest KL-divergences but is not the best visualization. (c) ct-SNE embedding with $\beta' = 0.01$ gives a better visualization.*

inspection, the visualization appears to be globally divided according the four areas. This is further confirmed by coloring the data points according to the four areas: most of the clusters are indeed quite homogeneous with respect areas

Knowing from the t-SNE visualization the papers are indeed divided according to areas, the area structure in the visualization is not very informative anymore. Thus we can encode the area as prior for ct-SNE so that other interesting structures can emerge. Using the same color scheme, ct-SNE shows a visualization that has many clusters with mixed colors (Fig. 6b). This indicates the area information is mostly removed in the ct-SNE embedding. This is further confirmed by selecting clusters in ct-SNE embedding (Fig. 6d) and highlight the same set of points in the t-SNE embedding (Fig. 6c). The clusters highlighted in the ct-SNE visualization often consists of clusters (topics) from different areas (i.e., t-SNE clusters with different colors) that spread over the t-SNE visualization. Indeed, feature ranking indicates that papers in the selected ct-SNE cluster have similar topics in e.g., 'privacy', 'data steam', 'computer vision'. Finally, we noticed that some clusters in ct-SNE (Fig. 6d) embedding also exist in the t-SNE embedding (Fig. 6c). Using feature ranking as above we found these clusters are not homogeneous in terms of area of study, but in terms of topics (e.g., 'clustering', 'active learning'), indicating a tightly connected research community behind the topic. Thus, by removing the irrelevant area structure using ct-SNE, clusters that persists in both visualizations become more salient and easier to observe.

Parameters sensitivity

To understand the effect of the parameter α' (or equivalently, β') on ct-SNE embeddings (**Q3**), we study ct-SNE embeddings on the synthetic dataset with the prior fixed to be the cluster labels in dimensions 1–4. First, we try to understand the relation between the ct-SNE objective and the parameter α' (or equivalently,

β'). We evaluated the ct-SNE objective (Eq. 3.5) on the ct-SNE embeddings obtained by ranging β' (and α' correspondingly) from 0.01 (strong prior removal effect) to 1.0 (no prior remove effect, equivalent to t-SNE) with step size 0.1. We also evaluated the t-SNE objective (first term in Eq. 3.5) and the second term in Eq. 3.5 (the only term that depends on the prior, subsequently referred to as the *prior term*) for the ct-SNE embeddings associated with various β's.

Fig. 7a visualizes the values of the ct-SNE objective, t-SNE objective, and ct-SNE prior term against different β's. Observe that by using a prior, the ct-SNE embedding achieves a better approximation to the higher dimensional data. That is, ct-SNE achieves a lower KL-divergence (lowest at $\beta' = 0.3$) than t-SNE does ($\beta' = 1$). This is because the prior term in the ct-SNE objective can be negative. Although the t-SNE objective increases when β' decreases, it is compensated by the negative value contributed by the prior term. Indeed, by factoring out certain prior from the lower dimensional embedding, the necessity of the embedding to represent the prior is alleviated, enabling ct-SNE to have more freedom to approximate the high-dimensional proximities.

Interestingly, we observe that the embedding with smallest KL-divergence does not necessarily give better visualization (e.g., clear separation of the clusters). We visualize the ct-SNE embedding that achieves smallest KL-divergence ($\beta' = 0.3$, Fig. 7b) and compare it with the ct-SNE embedding that has strongest prior removal effect but larger KL-divergence ($\beta' = 0.01$, Fig. 7c). Although the embedding with stronger prior removal effect has larger objective value, it gives a clearer clustering than in the embedding with smaller KL-divergence ($\beta' = 0.3$). As a result, the clusters in dimensions 5–6 are easier to identify. Hence, we propose as rule of thumb when using ct-SNE for visualization to use small β' (e.g., $\beta' = 0.01$).

Baseline comparisons

In this section, we compare ct-SNE with two non-trivial baselines. The basic idea is to first remove the known factor from the dataset, and perform t-SNE to produce lower dimensional representations. Here we use a non-linear and a linear method to remove the known factors: adversarial auto-encoder (AAE) and canonical correlation analysis (CCA). The implementation of the baselines and code for comparison experiments are also available at `https://bitbucket.org/ghentdatascience/ct-sne`.

Baseline: AAE and t-SNE. Adversarial auto-encoder (AAE) [15] can be used to learn a latent representation that prevents the discriminator from predicting certain attributes [14]. In order to remove prior information from the low-dimensional representation of a dataset using AAE, we can configure the discriminator to predict the prior attributes, and using the auto-encoder to adversarially remove the prior from the latent representation of the dataset.

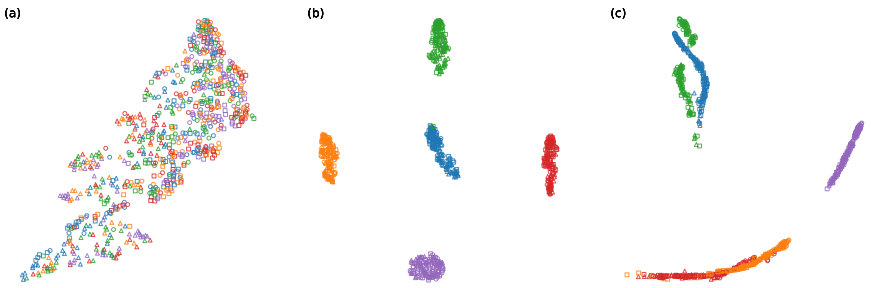

Figure 8: Visualization of 2-d embeddings obtained by applying the AAE based approach on the synthetic dataset. The data points are colored according to the cluster label in dimensions 1-4*. The data points are also plotted using different markers based on the cluster labels in dimensions* 5-6*. (a) The AAE based approach successfully removed the clustering information in dimensions* 1-4*, but failed to reveal the clusters in dimensions* 5-6 *(b) AAE successfully removed the clustering information in dimensions* 5-6 *and also reveals the clusters in dimensions* 1-4 *(c) AAE failed to remove the clustering information in dimensions* 1-6*.*

We adopt the AAE configuration described by Edwards and Storkey [6]. AAE is in general difficult to tune: it has 8 hyperparameters (4 network structure parameters, 2 weights in the objective, and 2 learning rates) and a few design choices about the network architecture (e.g., the number of layers in each subnetwork and activation functions). We tried different parameter settings and managed to remove the clustering label information in dimensions 1–4 (Fig. 8a) and 5–6 (Fig. 8b) from the data. In Figure 8a, the AAE approach manages to remove the prior information, but it fails to pick up the complementary structure in the data (clusters in dimensions 5–6). It also fails to remove the prior information (cluster labels in dimension 1–6) in Figure 8c. Comparing to this baseline, ct-SNE practically has only one parameter (β') to tune, which often can be set to a small positive number (e.g., 0.01).

Baseline: CCA and t-SNE. Canonical correlation analysis [10] aims to find a linear transformation for two random variables such that the correlation between transformed variables is maximized. To remove the prior information from data using CCA, one approach is to first find the (at most) $d - 2$ subspace (d is the dimensionality of the data) in which the transformed data and the prior information (one hot encoding of the labels) have the largest correlation. Then the data is whitened by projecting it onto the null space (at least 2-d) of the subspace found in the first step. By doing so, the whitened data is less correlated to the known factor.

Another variant of the CCA-based approach is directly projecting the data onto the 2-dimensional subspace found by CCA in which the transformed data and labels has smallest correlation. To be consistent, we also apply t-SNE to the trans-

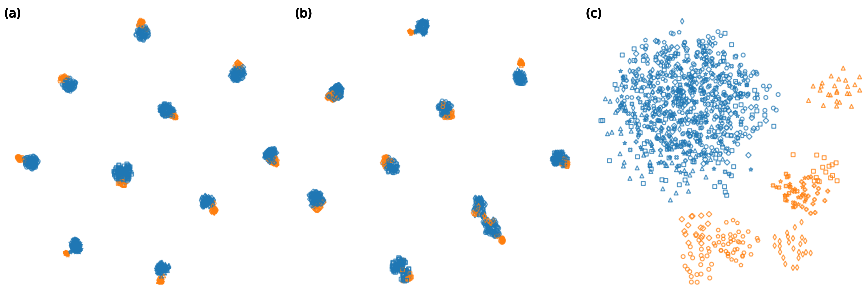

Figure 9: Visualization of 2-d embeddings obtained by applying CCA-based approaches and ct-SNE on a synthetic 5 *dimensional dataset. (a) Projecting data onto the null space of CCA top components and then apply t-SNE gives an embedding that picks up the* 10 *large clusters (plotted with different markers) but failed to pick up the structure of two small clusters (colored differently) within each large cluster. (b) Projecting the data onto CCA components with least correlation and then apply t-SNE also fails to pick up the two-cluster structure within the large clusters. (c) ct-SNE removes the* 10 *cluster information in the embedding and shows clearly the two cluster structure within each larger cluster.*

formed data.

Our experimental results show the CCA-based approaches can easily remove label information that is orthogonal to other attributes in the data. For example, in the UCI Adult dataset, the gender information is orthogonal to the ethnicity and income, which can be easily removed using the CCA approach. However, the CCA-based approach performs poorly when the known factor is correlated with other attributes. Moreover, the CCA-based approaches also have the limitation that the number of the projection vectors is upper-bounded by the dimensionality of the data. If the number of unique values of an attribute exceeds the dimensionality of the data, the CCA projection would not be able to remove the label info entirely from the data. To illustrate our points, we synthesized a 5-dimensional dataset with 1,000 data points. The data points are grouped into 10 clusters each corresponding to a multi-variate Gaussian with random location and small variance. Additionally, each cluster is separated into two small clusters (one contains 20% points of the cluster, and another includes the rest) along one randomly chosen axis. Figure 9a,b shows both the CCA approaches pick up only the 10 large clusters (differentiated using marker shape) but failed to pick up the structure of two small clusters (plotted in different colors) within each large cluster. On the other hand, ct-SNE removes the 10 cluster information in the embedding and shows each large cluster can be further separated in to two smaller clusters.

Thus, the CCA-based baselines perform poorly when the known factor is correlated with other attributes. Moreover, the number of the projection vectors in CCA-based baselines is upper-bounded by the dimensionality of the data. Meanwhile, ct-SNE does not have these limitations.

name	size	dim.	exact	apprx. ($\theta = 0.5$)
Synthetic	1,000	10	0.06	0.01
UCI Adult	1,000	6	0.07	0.01
DBLP	43,346	64	503.97	0.45
Synthetic	500,000	128	100,278	9.1

Table 1: Average runtime (in seconds) of exact and approximated ct-SNE in computing one gradient update step. To measure the runtime of ct-SNE on a dataset with similar size as the Facebook dataset, we scaled the Synthetic dataset up to $500,000$ *data points with* 128 *dimensions.*

Runtime

We measure the runtime of the exact ct-SNE and the approximated version ($\theta = 0.5$) on a PC with a quad-core 2.3GHz Inter Core i5 and a 2133MHz LPDDR3 RAM. By default, the maximum number of iterations of ct-SNE gradient update is 1,000. For larger datasets and prior attributes that have many values, more iterations are required to achieve a convergence. For example, the synthetic dataset (1,000 samples and 10 dimensions) requires fewer than 1,000 iterations to converge while the Facebook dataset (500,000 examples and 128 dimensions) requires 3,000 iterations to converge. Table. 1 shows that approximated ct-SNE is efficient and applicable to large data with high dimensionality, while exact ct-SNE is not.

D Extended related work

Many dimensionality reduction methods have been proposed in the literature. Arguably, n-body problem based methods[7] such as MDS [26], Isomap [25], t-SNE [13], LargeVis [23], and UMAP [16] appear to be the most popular ones. These methods typically have three components: (1) a proximity measure in the input space, (2) a proximity measure in the embedding space, (3) a loss function comparing the proximity between data points in the embedding space with the proximity in the input space. When minimizing the loss over the embedding space, the data points (i.e., the n bodies) have pairwise interactions and the embedding of all points needs to be updated simultaneously. Since the optimization problem is not convex, local minima are typically accepted as output. ct-SNE belongs to this class of DR methods. It accepts both high-dimensional data and priors about the data as inputs, and searches for low-dimensional embeddings while discounting structure in the input data specified as prior knowledge. Closely related, in the multi-maps t-SNE work [28] factors that are mutually exclusive are captured by multiple t-SNE embeddings at once. Comparing to multi-map t-SNE, ct-SNE allows users to disentangle information in a targeted (subjective) manner, by specifying which information they would like to have factored out.

[7] In Section 3.2.3 we provide more information on the n-body problem

As a core component of ct-SNE is the prior information specified by the user, it can be considered an interactive DR method. Existing papers on *interactive* DR can be categorized into two groups. The first group aim to improve the explainability and computation efficiency of existing DR methods via novel visualizations and interactions. iPCA [11] allows users to easily explore the PCA components and thus achieve better understanding of the linear projections of the data onto different PCA components. Cavallo and Demiralp [3] helps the user to understand low-dimensional representations by applying perturbations to probe the connection between input attributed space and embedding space. Similarly, Faust et al. [7] introduce a method based on perturbations to visualize the effect of a specific input attribute on the embedding, while Stahnke et al. [22] introduce 'probing' as a means to understand the meaning of point set selections within the embedding. Steerable t-SNE [19] aims to make t-SNE more scalable by quickly providing a sketch of an embedding which is then refined only upon the user's interests.

The second group of interactive DR methods adjust the algorithms according to a users' inputs. SICA [12] and SIDE [20] explicitly model the user's belief state and find linear projections that contrast to it. These two methods are linear DR methods thus cannot present non-linear structures in the low-dimensional representations. Work by Dıaz et al. [5] allows users to define their own metric in the input space, after which the low-dimensional representation reflects the adjusted importance of the attributes. This method puts the burden on the user for direct manipulation of the input space metric. Many variants of existing DR methods have been introduced where user feedback entails editing of the embedding, and such manually embedded points are used as constraints to guide the dimensionality reduction [e.g., 1, 2, 17]. These methods contrast with ct-SNE in that the user feedback must be obeyed in the output embedding, while for ct-SNE the prior knowledge defined by the user guides what is irrelevant to the user.

References

[1] B Alipanahi and A Ghodsi. Guided locally linear embedding. *PRL*, 32(7): 1029–1035, 2011.

[2] E Barshan, A Ghodsi, Z Azimifar, and M Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *PR*, 44(7):1357–1371, 2011.

[3] M Cavallo and Ç Demiralp. A visual interaction framework for dimensionality reduction based data exploration. In *CHI*, page 635, 2018.

[4] D Dheeru and E Karra Taniskidou. UCI machine learning repository, 2017.

[5] I Dıaz, A A Cuadrado, D Pérez, F J Garcıa, and M Verleysen. Interactive dimensionality reduction for visual analytics. In *ESANN*, pages 183–188, 2014.

[6] H Edwards and A Storkey. Censoring representations with an adversary. *arXiv:1511.05897*, 2015.

[7] R Faust, D Glickenstein, and C Scheidegger. Dimreader: Axis lines that explain non-linear projections. *TVCG*, 25(1):481–490, 2019.

[8] A G Gray and A W Moore. N-body'problems in statistical learning. In *NeurIPS*, pages 521–527, 2001.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[10] H Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4): 321–377, 1936.

[11] D H Jeong, C Ziemkiewicz, B Fisher, W Ribarsky, and R Chang. ipca: An interactive system for pca-based visual analytics. In *CGF*, volume 28, pages 767–774, 2009.

[12] B Kang, J Lijffijt, R Santos-Rodríguez, and T De Bie. Subjectively interesting component analysis: data projections that contrast with prior expectations. In *KDD*, pages 1615–1624, 2016.

[13] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[14] D Madras, E Creager, T Pitassi, and R Zemel. Learning adversarially fair and transferable representations. *arXiv:1802.06309*, 2018.

[15] A Makhzani, J Shlens, N Jaitly, I Goodfellow, and B Frey. Adversarial autoencoders. *arXiv:1511.05644*, 2015.

[16] L McInnes and J Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

[17] D Paurat and T Gärtner. Invis: A tool for interactive visual data analysis. In *ECML-PKDD*, pages 672–676, 2013.

[18] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[19] N Pezzotti, B PF Lelieveldt, L van der Maaten, T Höllt, E Eisemann, and A Vilanova. Approximated and user steerable tsne for progressive visual analytics. *TVCG*, 23(7):1739–1752, 2017.

[20] K Puolamäki, E Oikarinen, B Kang, J Lijffijt, and T De Bie. Interactive visual data exploration with subjective feedback: An information-theoretic approach. In *ICDE*, pages 1208–1211, 2018.

[21] P Ram, D Lee, W March, and A G Gray. Linear-time algorithms for pairwise statistical problems. In *NeurIPS*, pages 1527–1535, 2009.

[22] J Stahnke, M Dörk, B Müller, and A Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *TVCG*, 22(1):629–638, 2016.

[23] J Tang, J Liu, M Zhang, and Q Mei. Visualizing large-scale and high-dimensional data. In *WWW*, pages 287–297, 2016.

[24] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.

[25] J B Tenenbaum, V De Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[26] W S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[27] L van der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[28] L van der Maaten and G Hinton. Visualizing non-metric similarities in multiple maps. *MLJ*, 87(1):33–55, 2012.

4

Network Representations

Conditional Network Embeddings

Abstract Network Embeddings (NEs) map the nodes of a given network into d-dimensional Euclidean space $\mathbb{R}^d$. Ideally, this mapping is such that 'similar' nodes are mapped onto nearby points, such that the NE can be used for purposes such as link prediction (if 'similar' means being 'more likely to be connected' or 'having similar neighborhoods') or classification (if 'similar' means 'being more likely to have the same label'). In recent years various methods for NE have been introduced, all following a similar strategy: defining a notion of similarity between nodes, a distance measure in the embedding space, and a loss function that penalizes large distances for similar nodes and small distances for dissimilar nodes.

A difficulty faced by existing methods is that certain networks are fundamentally hard to embed due to their structural properties: (approximate) multipartiteness, certain degree distributions, assortativity, etc. To overcome this, we introduce a conceptual innovation to the NE literature and propose to create *Conditional Network Embeddings* (CNEs); embeddings that maximally add information with respect to given structural properties (e.g. node degrees, block densities, etc.). We use a simple Bayesian approach to achieve this, and propose a block stochastic gradient descent algorithm for fitting it efficiently. We demonstrate that CNEs are superior for link prediction and multi-label classification when compared to state-of-the-art methods, and this without adding significant mathematical or computational complexity. Finally, we illustrate the potential of CNE for network

visualization.

4.1 Introduction

Network Embeddings (NEs) map nodes into d-dimensional Euclidean space $\mathbb{R}^d$ such that an ordinary distance measure allows for meaningful comparisons between nodes. Embeddings directly enable the use of a variety of machine learning methods (classification, clustering, etc.) on networks, explaining their exploding popularity. NE approaches typically have three components [10]: (1) A measure of similarity between nodes. E.g. nodes can be deemed more similar if they are adjacent, have strongly overlapping neighborhoods, or are otherwise close to each other (link and path-based measures) [9, 18, 20], or if they have similar functional properties (structural measures) [19]. (2) A metric in the embedding space. (3) A loss function comparing similarity between node pairs in the network with the proximity of their embeddings. A good NE is then one for which the average loss is small.

Limitations of existing NE approaches A problem with all NE approaches is that networks are fundamentally more expressive than embeddings in Euclidean spaces. Consider for example a bipartite network $G = (V, U, E)$ with V, U two disjoint sets of nodes and $E \subseteq V \times U$ the set of links. It is in general impossible to find an embedding in $\mathbb{R}^d$ such that $v \in V$ and $u \in U$ are close for all $(v, u) \in E$, while all pairs $v, v' \in V$ are far from each other, as well as all pairs $u, u' \in U$. To a lesser extent, this problem will persist in approximately bipartite networks, or more generally (approximately) k-partite networks such as networks derived from stochastic block models.[1] This shows that first-order similarity (i.e. adjacency) in networks cannot be modeled well using a NE. Similar difficulties exist for second-order proximity (i.e. neighborhood overlap) and other node similarity notions. A more subtle example is a network with a power law degree distribution. A first-order similarity NE will tend to embed high degree nodes towards the center (to be close to lots of other nodes), while the low degree nodes will be on the periphery. Yet, this effect reduces the embedding's degrees of freedom for representing similarity independent of node degree.

CNE: the idea To address these limitations of NEs, we propose a principled probabilistic approach—dubbed *Conditional Network Embedding (CNE)*—that allows optimizing embeddings w.r.t. certain prior knowledge about the network, for-

[1] For example multi-relational data can be represented as a k-partite network, where the schema specifies between which types of objects links may exist. Another example is a heterogeneous information network, where no schema is provided but links are more or less common depending on the (specified) types of the nodes.

malized as a prior distribution over the links. This prior knowledge may be derived from the network itself such that no external information is required.

A combined representation of a prior based on structural information and a Euclidean embedding makes it possible to overcome the problems highlighted in the examples above. For example, nodes in different blocks of an approximately k-partite network need not be particularly distant from each other if they are a priori known to belong to the same block (and hence are unlikely or impossible to be connected a priori). Similarly, high degree nodes need not be embedded near the center of the point cloud if they are known to have high degree, as it is then known that they are connected to many other nodes. The embedding can thus focus on encoding which nodes in particular it is connected to.

CNE is also potentially useful for network visualization, with the ability to filter out certain information by using it as a prior. For example, suppose the nodes in a network represent people working in a company with a matrix-structure (vertical being units or departments, horizontal contents such as projects) and links represent whether they interact a lot. If we know the vertical structure, we can construct an embedding where the prior is the vertical structure. The information that the embedding will try to capture corresponds to the horizontal structure. The embedding can then be used in downstream analysis, e.g., to discover clusters that correspond to teams in the horizontal structure.

Contributions and outline Our contributions can be summarized as follows:

- This chapter introduces the *concept of NE conditional on certain prior knowledge* about the network.

- Section 4.2 presents *CNE ('Conditional Network Embedding')*, which realizes this idea by using Bayes rule to combine a prior distribution for the network with a probabilistic model for the Euclidean embedding conditioned on the network. This yields the posterior probability for the network conditioned on the embedding, which can be maximized to yield a maximum likelihood embedding. Section 4.2.2 describes *a scalable algorithm* based on block stochastic gradient descent.

- Section 4.3 reports on *extensive experiments*, comparing with state-of-the-art baselines on link prediction and multi-label classification, on commonly used benchmark networks. These experiments show that CNE's link prediction accuracy is consistently superior. For multi-label classification CNE is consistently best on the Macro-F_1 score and best or second best on the Micro-F_1 score. These results are achieved with *considerably lower-dimensional embeddings* than the baselines. A case study also demonstrates the usefulness of CNE in *exploratory data analysis* of networks.

- Section 4.4 gives a brief overview of *related work*, before *concluding* the chapter in Section 4.5.
- All code, including code for repeating the experiments, and links to the datasets are available at: `https://bitbucket.org/ghentdatascience/cne`.

4.2 Methods

Section 4.2.1 introduces the probabilistic model used by CNE, and Section 4.2.2 describes an algorithm for optimizing it to find an optimal CNE. Before doing that, let us introduce some notation. An undirected network is denoted $G = (V, E)$ where V is a set of $n = |V|$ nodes and $E \subseteq \binom{V}{2}$ is the set of links (also known as edges). A link is denoted by an unordered node pair $\{i, j\} \in E$. Let $\hat{\mathbf{A}}$ denote the network's adjacency matrix, with element $\hat{a}_{ij} = 1$ for $\{i, j\} \in E$ and $\hat{a}_{ij} = 0$ otherwise. The goal of NE (and thus of CNE) is to find a mapping $f : V \to \mathbb{R}^d$ from nodes to d-dimensional real vectors. The resulting embedding is denoted $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)' \in \mathbb{R}^{n \times d}$.

4.2.1 The Conditional Network Embedding model

The newly proposed method CNE aims to find an embedding $\mathbf{X}$ that is maximally informative about the given network G, formalized as a Maximum Likelihood (ML) estimation problem:

$$\underset{\mathbf{X}}{\operatorname{argmax}} \; P(G|\mathbf{X}). \tag{4.1}$$

Innovative about CNE is that we do not postulate the likelihood function $P(G|\mathbf{X})$ directly, as is common in ML estimation. Instead, we use a generic approach to derive prior distributions for the network $P(G)$, and we postulate the density function for the data conditional on the network $p(\mathbf{X}|G)$. This allows one to introduce any prior knowledge about the network into the formulation, through a simple application of Bayes rule[2]: $P(G|\mathbf{X}) = \frac{p(\mathbf{X}|G)P(G)}{p(\mathbf{X})}$. The consequence is that the embedding will not need to represent any information that is already represented by the prior $P(G)$.

Section 4.2.1 describes how a broad class of prior information types can be modeled for use by CNE. Section 4.2.1 describes a possible conditional distribution (albeit an improper one), the one we used for the particular CNE method in this chapter. Section 4.2.1 describes the posterior distribution.

[2]Note that this approach is uncommon: despite the usage of Bayes rule, it is not Maximum A Posteriori (MAP) estimation as the chosen embedding $\mathbf{X}$ is the one maximizing the likelihood of the network.

The prior distribution for the network

We wish to be able to model a broad class of prior knowledge types in the form of a manageable prior probability distribution $P(G)$ for the network. Let us first focus on three common types of prior knowledge: knowledge about the overall network density, knowledge about the individual node degrees, and knowledge about the edge density within or between particular subsets of the nodes (e.g. for multipartite networks). Each of these can be expressed as sets of constraints on the expectations of the sum of various subsets $S \subseteq \binom{V}{2}$ of elements from the adjacency matrix: $\mathbb{E}\left\{\sum_{\{i,j\}\in S} a_{ij}\right\} = \sum_{\{i,j\}\in S} \hat{a}_{ij}$, where the expectation is taken w.r.t. the sought prior distribution $P(G)$. In the 1st case, $S = \binom{V}{2}$; in the 2nd case, $S = \{(i,j)|j \in V, j \neq i\}$ for information on the degree of node i; and in the 3rd case $S = \{(i,j)|i \in A, j \in B, i \neq j\}$ for specified sets $A, B \in V$.

Such constraints do not determine $P(G)$ fully, so we determine $P(G)$ as the distribution with maximum entropy from all distributions satisfying all these constraints. Adriaens et al. [1], van Leeuwen et al. [22] showed that finding this distribution is a convex optimization problem that can be solved efficiently, particularly for sparse networks. They also showed that the resulting distribution is a product of independent Bernoulli distributions, one for each element of the adjacency matrix:

$$P(G) = \prod_{\{i,j\}\in\binom{V}{2}} P_{ij}^{\hat{a}_{ij}}(1-P_{ij})^{1-\hat{a}_{ij}}, \tag{4.2}$$

where $P_{ij} \in [0,1]$ is the probability that $\{i,j\}$ is linked in the network under this distribution. They showed that all these P_{ij} can be expressed in terms of a limited number of parameters, namely the unique Lagrange multipliers for the prior knowledge constraints in the maximum entropy problem. In practice, the number of such unique Lagrange multipliers is far smaller than n.

The three cases discussed above are merely examples of how constraints on the expectation of subsets of the elements of the adjacency matrix can be useful in practice. For example, if nodes are ordered in some way (e.g. according to time), it could be used to express the fact that nodes are connected only to nodes that are not too distant in that ordering. Moreover, the above results continue to hold for constraints that are on *weighted* linear combinations of elements of the adjacency matrix. This makes it possible to express other kinds of prior knowledge, e.g. on the relation between connectedness and distance in a node order (if provided), or on the network's (degree) assortativity. A detailed discussion and empirical analysis of such alternatives is deferred to further work.

The distribution of the data conditioned on the network

We now move on to postulating the conditional density $P(\mathbf{X}|G)$. Clearly, any rotation or translation of an embedding should be considered equally good, as we are only interested in distances between pairs of nodes in the embedding. Thus, the pairwise distances between points, denoted as $d_{ij} \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|_2$ for points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$, must form a set of sufficient statistics.

The density should also reflect the fact that connected node pairs tend to be embedded to nearby points, while disconnected node pairs tend to be embedded to more distant points. Let us focus initially on the marginal density of d_{ij} conditioned on G. The proposed model assumes that given $\hat{a}_{ij}$ (i.e. knowledge of whether $\{i,j\} \in E$ or not), d_{ij} is conditionally independent of the rest of the adjacency matrix. More specifically, we model the conditional distribution for the distances d_{ij} given $\{i,j\} \in E$ as half-normal $\mathcal{N}_+$ [11] with spread parameter $\sigma_1 > 0$:[3]

$$p\left(d_{ij}|\{i,j\} \in E\right) = \mathcal{N}_+\left(d_{ij}|\sigma_1^2\right), \tag{4.3}$$

and the distribution of distances d_{kl} with $\{k,l\} \notin E$ as half-normal with spread parameter $\sigma_2 > \sigma_1$:

$$p\left(d_{kl}|\{k,l\} \notin E\right) = \mathcal{N}_+\left(d_{kl}|\sigma_2^2\right). \tag{4.4}$$

The choice of $0 < \sigma_1 < \sigma_2$ will ensure the embedding reflects the neighborhood proximity of the network. Indeed, the differences between the embedded nodes that are not connected in the network are expected to be larger than the differences between the embedding of connected nodes. Without losing generality (as it merely fixes the scale), we set $\sigma_1 = 1$ through out this chapter.

It is clear that the distances d_{ij} cannot be independent of each other (e.g. the triangle inequality entails a restriction of the range of d_{ij} given the values of d_{ik} and d_{jk} for some k). Nevertheless, akin to Naive Bayes, we still model the joint distribution of all distances (and thus of the embedding $\mathbf{X}$ up to a rotation/translation) as the product of the marginal densities for all pairwise distances:

$$p(\mathbf{X}|G) = \prod_{\{i,j\} \in E} \mathcal{N}_+\left(d_{ij}|\sigma_1^2\right) \cdot \prod_{\{k,l\} \notin E} \mathcal{N}_+\left(d_{kl}|\sigma_2^2\right). \tag{4.5}$$

This is an improper density function, due to the constraints imposed by Euclidean geometry. Indeed, certain combinations of pairwise distances should be assigned a probability 0 as they are geometrically impossible. As a result, $p(\mathbf{X}|G)$ is also not properly normalized. Yet, even though $p(\mathbf{X}|G)$ is improper, it can still be used to derive a properly normalized posterior for G as detailed next.

[3] A half-normal distribution, with density denoted here as $\mathcal{N}_+(\cdot|\sigma^2)$, is a zero-mean normal distribution with standard deviation σ, conditioned on the random variable being positive. Of course the standard deviation of the conditioned normal distribution is not equal to σ, so we refer to σ more loosely as its spread parameter.

The posterior of the network conditioned on the embedding

The (also improper) marginal density $p(\mathbf{X})$ can now be computed as:

$$p(\mathbf{X}) = \sum_G p(\mathbf{X}|G)P(G) = \sum_G \prod_{\{i,j\}\in E} \mathcal{N}_+\left(d_{ij}|\sigma_1^2\right) P_{ij} \cdot \prod_{\{k,l\}\notin E} \mathcal{N}_+\left(d_{kl}|\sigma_2^2\right)(1-P_{kl}),$$
$$= \prod_{i,j} \left[\mathcal{N}_+\left(d_{ij}|\sigma_1^2\right) P_{ij} + \mathcal{N}_+\left(d_{ij}|\sigma_2^2\right)(1-P_{ij})\right].$$

We now have all ingredients to compute the posterior of the network conditioned on the embedding by a simple application of Bayes' rule:

$$P(G|\mathbf{X}) = \frac{p(\mathbf{X}|G)\cdot P(G)}{p(\mathbf{X})} = \prod_{\{i,j\}\in E} \frac{\mathcal{N}_+\left(d_{ij}|\sigma_1^2\right) P_{ij}}{\mathcal{N}_+\left(d_{ij}|\sigma_1^2\right) P_{ij} + \mathcal{N}_+\left(d_{ij}|\sigma_2^2\right)(1-P_{ij})}$$
$$\cdot \prod_{\{k,l\}\notin E} \frac{\mathcal{N}_+\left(d_{kl}|\sigma_2^2\right)(1-P_{kl})}{\mathcal{N}_+\left(d_{kl}|\sigma_1^2\right) P_{kl} + \mathcal{N}_+\left(d_{kl}|\sigma_2^2\right)(1-P_{kl})}. \tag{4.6}$$

This is the likelihood function to be maximized in order to get the ML embedding. Note that, although it was derived using the improper density function $p(\mathbf{X}|G)$, thanks to the normalization with the (equally improper) $p(\mathbf{X})$, this is indeed a properly normalized distribution.

4.2.2 Finding the most informative embedding

Maximizing the likelihood function $P(G|\mathbf{X})$ is a non-convex optimization problem. We propose to solve it using a block stochastic gradient descent approach, explained below. The gradient of the likelihood function (Eq. 4.6) with respect to the embedding $\mathbf{x}_i$ of node i is:[4]

$$\nabla_{\mathbf{x}_i} \log\left(P(G|\mathbf{X})\right) = 2 \sum_{j:\{i,j\}\in E} (\mathbf{x}_i - \mathbf{x}_j) P(a_{ij}=0|\mathbf{X}) \left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)$$
$$+ 2 \sum_{j:\{i,j\}\notin E} (\mathbf{x}_i - \mathbf{x}_j) P(a_{ij}=1|\mathbf{X}) \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right). \tag{4.7}$$

As $\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right) < 0$, the first summation pulls the embedding of node i towards embeddings of the nodes it is connected to in G. Moreover, if the current prediction of the link $P(a_{ij}=1|\mathbf{X})$ is small (i.e., if $P(a_{ij}=0|\mathbf{X})$ is large), the pulling effect will be larger. Similarly, the second summation pushes $\mathbf{x}_i$ away from the embeddings of unconnected nodes, and more strongly so if the current prediction

[4] We refer the reader to the supplementary material for detailed derivations.

of a link between these two unconnected nodes $P(a_{ij} = 1|\mathbf{X})$ is larger. The magnitudes of the gradient terms are also affected by parameter σ_2 and prior $P(G)$: a large σ_2 gives stronger push and pulling effect. In our quantitative experiments we always set $\sigma_2 = 2$.

Computing this gradient w.r.t. a particular node's embedding requires computing the pairwise differences between n proposed d-dim embedding vectors, with time complexity $\mathcal{O}(n^2 d)$ and space complexity $\mathcal{O}(nd)$. This is computationally demanding for mainstream hardware even for networks of sizes of the order $n = 1000$ and dimensionalities of the order $d = 10$, and prohibitive beyond that. To address this issue, we approximate both summations in the objective by sampling $k < n/2$ terms from each. This amounts to uniformly sampling k nodes from the set of connected nodes (where $a_{ij} = 1$), and k from the set of unconnected nodes (where $a_{ij} = 0$).[5] This reduces the time complexity to $\mathcal{O}(ndk)$.

Note that each of the terms is bound in norm by the diameter of the embedding, as the other factors are bound by 1 for $\sigma_1 = 1, \sigma_1 < \sigma_2$. If the diameter were bounded, a simple application of Hoeffding's inequality would demonstrate that this average is sharply concentrated around its expectation, and is thus a suitable approximation. Although there is no prior bound that holds with guarantee on the diameter of the embedding, this does shed some light on why this approach works well in practice. The choice of k will in practice be motivated by computational constraints. In our experiments we set it equal or similar to the largest degree, such that the first term is computed exactly.

4.3 Experiments

We first evaluate the network representation obtained by CNE on downstream tasks typically used for evaluating NE methods: link prediction for links and multi-label classification for nodes. Then, we illustrate how to use CNE to visually explore multi-relational data.

4.3.1 experiment setup

For the quantitative evaluations, we compare CNE against a panel of state-of-the-art baselines for NE: Deepwalk [18], LINE [20], node2vec [9], metapath2vec++ [7], and struc2vec [19]. Table 4.1 lists the networks used in the experiments. A brief discussion of the methods and the networks is given in the supplement.

For all methods we used their default parameter settings reported in the original papers and with d $= 128$. For node2vec, the hyperparameters p and q are tuned

[5] If a node i has a degree smaller than k, we sample more non-connected neighbors to make sure that $2k$ points are used for the approximation of the gradient – and conversely if a node has a degree larger than $n - k$.

Data	Type	#Nodes	#Links	#Labels
Facebook [12]	Friendship	4,039	88,234	–
arXiv ASTRO-PH [12]	Co-authorship	18,722	198,110	–
Gowalla [6]	Friendship	196,591	950,327	–
StudentDB [8]	Relational/k-partite	403	3,429	–
BlogCatalog [23]	Bloggers	10,312	333,983	39
Protein-Protein Int. [4]	Biological	3,890	76,584	50
Wikipedia [14]	Word co-occurrence	4,777	184,812	40

Table 4.1: Networks used in experiments.

over a grid $p, q \in \{0.25, 0.05, 1, 2, 4\}$ using 10-fold cross validation. We repeat our experiments for 10 times with different random seeds. The final scores are averaged over the 10 repetitions.

4.3.2 Link prediction

In link prediction, we randomly remove 50% of the links of the network while keeping it connected. The remaining network is thus used for training the embedding, while the removed links (positive links, labeled 1) are used as a part of the test set. Then, the test set is topped up by an equal number of negative links (labeled 0) randomly drawn from the original network. In each repetition of the experiment, the node indices are shuffled so as to obtain different train-test splits.

We compare CNE with other methods based on the area under the ROC curve (AUC). The methods are evaluated against all datasets mentioned in the previous section. CNE typically works well with small dimensionality d and sample size k. In this experiment we set $d = 8$ and $k = 50$. Only for the two largest networks (arXiv and Gowalla), we increase the dimensionality to d $= 16$ to reduce underfitting. To calculate AUC, we first compute the posterior $P(a_{ij} = 1|\mathbf{X}_{\text{train}})$ of the test links based on the embedding $\mathbf{X}_{\text{train}}$ learned on the training network. Then the AUC score is computed by comparing the posterior probability of the test links and their true labels.

In this task we first compare CNE against four simple baselines [9]: Common Neighbors ($|\mathbf{N}(i) \cap \mathbf{N}(j)|$), Jaccard Similarity ($\frac{|\mathbf{N}(i) \cap \mathbf{N}(j)|}{|\mathbf{N}(i) \cup \mathbf{N}(j)|}$), Adamic-Adar Score ($\sum_{t \in \mathbf{N}(i) \cap \mathbf{N}(j)} \frac{1}{\log |\mathbf{N}(t)|}$), and Preferential Attachment ($|\mathbf{N}(i)| \cdot |\mathbf{N}(j)|$). These baselines are neighborhood based node similarity measures. We first compute pairwise similarity on the training network. Then from the computed similarities we obtain scores for testing links as the similarity between the two ending nodes. Those scores are then used to compute the AUC against the true labels.

For the NE baselines, we perform link prediction using logistic regression based on the link representation derived from the node embedding $\mathbf{X}_{\text{train}}$. The link representation is computed by applying the Hadamard operator (element wise

Algorithm	Facebook	PPI	arXiv	BlogCat.	Wikiped.	studentdb	Gowalla
Common Neigh.	0.9735	0.7693	0.9422	0.9215	0.8392	0.4160	0.7769
Jaccard Sim.	0.9705	0.7580	0.9422	0.7844	0.5048	0.4160	0.7519
Adamic Adar	0.9751	0.7719	0.9427	0.9268	0.8634	0.4160	0.7719
Prefer. Attach.	0.8295	0.8892	0.8640	0.9519	0.9130	0.9106	0.5626
Deepwalk	0.9798	0.6365	0.9207	0.6077	0.5563	0.7644	0.7156
LINE	0.9525	0.7462	0.9771	0.7563	0.7077	0.8562	0.8173
node2vec	0.9881	0.6802	0.9721	0.7332	0.6720	0.8261	0.7984
metapath2vec++	0.7408	0.8516	0.8258	0.9125	0.8334	0.9244	0.7769
struc2vec	0.6909	0.7752	0.7182	0.8631	0.8062	0.6290	TimeOut
CNE (uniform)	0.9905	0.8908	0.9865	0.9190	0.8417	0.9300	0.9738
CNE (degree)	**0.9909**	**0.9115**	**0.9882**	**0.9636**	**0.9158**	**0.9439**	**0.9818**
CNE (block)	NA	NA	NA	NA	NA	**0.9830**	NA

Table 4.2: The AUC scores for link prediction. TimeOut means aborted after 24 hours.

multiplication) on the node representation $\mathbf{x}_i$ and $\mathbf{x}_j$, which is reported to give good results [9]. Then the AUC score is computed by comparing the link probability (from logistic regression) of the test links with their true labels.

Results The link prediction results are shown in Table 4.2. Even with a uniform prior (i.e. prior knowledge only on the overall density), CNE performs better than all baselines on 5 of the 7 networks. With a degree prior, however, CNE outperforms all baselines on all networks. We attribute this to the fact that the degree prior encodes information which is hard to encode using a metric embedding alone. For the multi-relational dataset studentdb, metapath2vec++, which is designed for heterogeneous data, outperforms other baselines but not CNE (regardless of the prior information). Moreover, CNE is capable of encoding the knowledge of the block structure of this multi-relational network as a prior, with each block corresponding to one node type. Doing this improves the AUC further by 3.91% versus CNE with degree prior (from 94.39% to 98.30%; i.e., a 70% reduction in error).

In terms of runtime, over the seven datasets CNE is fastest in two cases, 12% slower than the fastest (metapath2vec++) in one case, and takes approximately twice as long in the four other cases (also metapath2vec++). Detailed runtime results can be found in the supplementary material.

4.3.3 Multi-label classification

We performed multi-label classification on the following networks: BlogCatalog, PPI, and Wikipedia. Detailed results are given in the supplement, while Table 4.3 contains an excerpt of the results. All baselines are evaluated in a standard logistic regression (LR) setup [18].

When using logistic regression also on the CNE embeddings, CNE performs on-par, but not particularly well (row CNE-LR). This should not be a surprise though, as potentially relevant information encoded by the prior (the degrees) will

Algorithm	BlogCatalog		PPI		Wikipedia	
	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1
Deepwalk	0.2544	0.3950	0.1795	0.2248	0.1872	0.4661
LINE	0.1495	0.2947	0.1547	0.2047	0.1721	**0.5193**
node2vec	0.2364	0.3880	0.1844	0.2353	0.1985	0.4746
metapath2vec++	0.0351	0.1684	0.0337	0.0726	0.1031	0.3942
struc2vec	0.0493	0.1653	0.0669	0.0971	0.1124	0.4019
CNE-LR (degree)	0.1833	0.3376	0.1484	0.1952	0.1370	0.4339
CNE-LP (block+degree)	**0.2935**	**0.4002**	**0.2639**	**0.2519**	**0.3374**	0.4839

Table 4.3: The F_1 scores for multi-label classification.

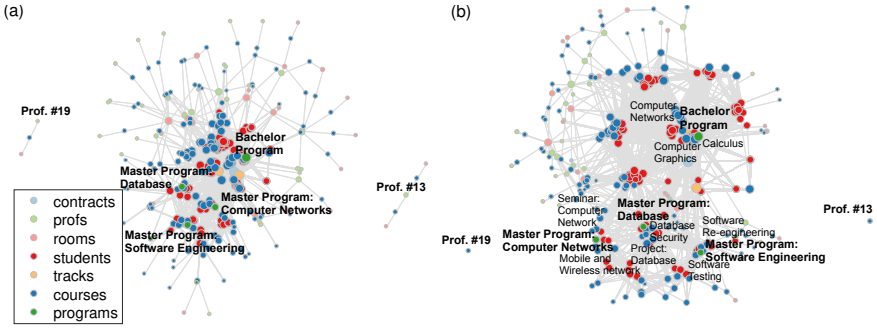

Figure 4.1: (a) 2-d embedding with uniform prior. (b) 2-d embedding with degree prior.

not be reflected in the embedding. However, multi-label classification can easily be cast as a link prediction problem, by adding to the network a node for each label, with a link to each node to which the label applies. Predicting a label for a node then amounts to predicting a link to that label node. To evaluate this strategy, we train an embedding on the original network plus half the label links, while the other half of the label links is held out for testing.

For the baselines, this link prediction setup does not lead to consistent improvements (see supplement), but for CNE it does (row CNE-LP, where LP stands for Link Prediction, in Table 4.3). On Micro-F_1 it is best or once close second best (after LINE with LR, see Table 4.3), and on Macro-F_1 it greatly outperforms any other method, suggesting improved performance mainly on the less frequent labels.

4.3.4 Visual exploration of multi-relational data

Here we qualitatively evaluate CNE's ability to facilitate visual exploration of multi-relational data, and how a suitable choice of the prior can help with this. To this end, we use CNE to embed the studentdb dataset directly into 2-dimensional space. As a larger σ_2 in general appears to give better visual separation between

node clusters, we set $\sigma_2 = 15$.

For comparison, we first apply CNE with uniform prior (overall network density). The resulting embedding (Fig. 4.1a) clearly separates bachelor student/-courses/program nodes (upper) from the master's nodes (lower). Also observe that the embedding is strongly affected by the node degrees (coded as marker size = log degree): high degree nodes flock together in the center. E.g., these are students who interact with many other smaller degree nodes (courses/programs). Although there are no direct links between program nodes (green) and course nodes (blue), the students (red) that connect them are pulling courses towards the corresponding program and pushing away other courses.

Next, we encode the individual node degrees as prior. As in this case the degree information is known, the embedding in addition shows the courses grouped around different programs, e.g.: "Bachelor Program" is close to course "Calculus"; "Master Program Computer Network" is close to course "Seminar Computer Network"; "Master Program Database" is close to course "Database Security"; "Master Program Software Engineering" is close to courses "Software Testing".

Thus, although this last evaluation remains qualitative and preliminary, it confirms that CNE with a suitable prior can create embeddings that clearly convey information in addition to the given prior.

4.4 Related Work

NE methods typically have three components [10]: (1) A similarity measure between nodes, (2) A metric in embedding space, (3) A loss function comparing proximity between nodes in embedding space with the similarity in the network. Early NE methods such as Laplacian Eigenmaps [3], Graph factorization [2], GraRep [5], and HOPE [15] optimize mean-squared-error loss between Euclidean distance or inner product based proximity and link based (adjacency matrix) similarity in the network. Recently, a few NE methods define node similarity based on paths. Those paths are generated using either the adjacency matrix [LINE, 20] or random walks (Deepwalk, Perozzi et al. 18, node2vec, Grover and Leskovec 9, methapath2vec++, Dong et al. 7, and struc2vec Ribeiro et al. 19). Path based embedding methods typically use inner products as proximity measure in the embedding space and optimize a cross-entropy loss. The recent struc2vec method [19] uses a node similarity measure that explicitly builds on structural network properties. CNE, unlike the aforementioned methods, unifies the proximity in embeddings space and node similarity using a probabilistic measure. This allows CNE to find a more informative ML embedding.

The question of how to visualize networks on digital screens has been studied for a long time. Recently there has been an uplift in methods to embed networks in a 'small' number of dimensions, where small means small as compared to the

number of nodes, yet typically much larger than two. These methods enable most machine learning methods to readily apply to tasks on networks, such as node classification or network partitioning. Popular methods include node2vec [9], where for example the default output dimensionality is 128. It is not designed for direct use in visualization, and typically one would fit a higher-dimensional embedding and then apply dimensionality reduction, such as PCA [16] or t-SNE [13] to visualize the data. CNE finds meaningful 2-d embeddings that can be visualized directly. Besides, CNE gives a visualization that conveys maximum information in addition to prior knowledge about the network.

4.5 Conclusions

The literature on NE has so far considered embeddings as tools that are used on their own. Yet, Euclidean embeddings are unable to accurately reflect certain kinds of network topologies, such that this approach is inevitably limited. We proposed the notion of Conditional Network Embeddings (CNEs), which seeks an embedding of a network that maximally adds information with respect to certain given prior knowledge about the network. This prior knowledge can encode information about the network that cannot be represented well by means of an embedding.

We implemented this conceptually novel idea in a new algorithm based on a simple probabilistic model for the joint of the data and the network, which scales similarly to state-of-the-art NE approaches. The empirical evaluation of this algorithm confirms our intuition that the combination of structural prior knowledge and a Euclidean embedding is extremely powerful. This is confirmed empirically for both the tasks of link prediction and multi-label classification, where CNE outperforms a range of state-of-the-art baselines on a wide range of networks.

In our future work we intend to investigate other models implementing the idea of conditional NEs, alternative and more scalable optimization strategies, as well as the use of other types of structural information as prior knowledge on the network.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the FWO (project no. G091017N, G0F9816N), and from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501.

Appendices

A Derivation of the gradient

Denote the Euclidean distance between two points as $d_{ij} \triangleq ||\mathbf{x}_i - \mathbf{x}_j||_2$. The derivative of d_{ij} with respect to embedding $\mathbf{x}_i$ of node i reads:

$$\nabla_{\mathbf{x}_i} d_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{d_{ij}}$$

Then the derivative of the log posterior with respect to $\mathbf{x}_i$ is given by:

$$\begin{aligned}
\nabla_{\mathbf{x}_i} \log\left(P(G|\mathbf{X})\right) &= \sum_{j:\{i,j\}\in E} \left(\frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} + \frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ji}} \right) \nabla_{\mathbf{x}_i} d_{ij} \\
&+ \sum_{j:\{i,j\}\notin E} \left(\frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} + \frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ji}} \right) \nabla_{\mathbf{x}_i} d_{ij} \\
&= 2 \sum_{j:\{i,j\}\in E} \frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} \frac{\mathbf{x}_i - \mathbf{x}_j}{d_{ij}} + 2 \sum_{j:\{i,j\}\notin E} \frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} \frac{\mathbf{x}_i - \mathbf{x}_j}{d_{ij}}
\end{aligned}$$

Using shorthand notation $\mathcal{N}_{ij,\sigma_1} = \mathcal{N}_+\left(d_{ij}|\sigma_1^2\right)$ and $\mathcal{N}_{ij,\sigma_2} = \mathcal{N}_+\left(d_{ij}|\sigma_2^2\right)$, we can compute the partial derivative $\frac{\partial \log(P(G|\mathbf{X}))}{\partial d_{ij}}$ for $\{i,j\} \in E$ as:

$$\begin{aligned}
\frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} &= \frac{\partial}{\partial d_{ij}} \sum_{\{i,j\}\in E} \log\left(\mathcal{N}_{ij,\sigma_1} P_{ij}\right) - \log\left(\mathcal{N}_{ij,\sigma_1} P_{ij} + \mathcal{N}_{ij,\sigma_2}\left(1 - P_{ij}\right)\right) \\
&= \frac{\mathcal{N}_{ij,\sigma_1} P_{ij} \cdot \frac{-d_{ij}}{\sigma_1^2}}{\mathcal{N}_{ij,\sigma_1} P_{ij}} - \frac{\mathcal{N}_{ij,\sigma_1} P_{ij} \cdot \frac{-d_{ij}}{\sigma_1^2} + \mathcal{N}_{ij,\sigma_2}\left(1 - P_{ij}\right) \cdot \frac{-d_{ij}}{\sigma_2^2}}{\mathcal{N}_{ij,\sigma_1} P_{ij} + \mathcal{N}_{ij,\sigma_2}\left(1 - P_{ij}\right)} \\
&= -\frac{d_{ij}}{\sigma_1^2} + P(a_{ij} = 1|\mathbf{X}) \frac{d_{ij}}{\sigma_1^2} + P(a_{ij} = 0|\mathbf{X}) \frac{d_{ij}}{\sigma_2^2}
\end{aligned}$$

Similarly, the partial derivative $\frac{\partial \log(P(G|\mathbf{X}))}{\partial d_{ij}}$ for $\{i,j\} \notin E$ reads:

$$\frac{\partial \log\left(P(G|\mathbf{X})\right)}{\partial d_{ij}} = -\frac{d_{ij}}{\sigma_2^2} + P(a_{ij} = 1|\mathbf{X}) \frac{d_{ij}}{\sigma_1^2} + P(a_{ij} = 0|\mathbf{X}) \frac{d_{ij}}{\sigma_2^2}.$$

The partial derivatives $\frac{\partial \mathcal{N}_{mn,\sigma} P_{mn}}{\partial d_{ij}}$ are nonzero only when $m = i$ and $n = j$, which gives the final gradient:

$$\begin{aligned}
\nabla_{\mathbf{x}_i} \log\left(P(G|\mathbf{X})\right) &= 2 \sum_{j:\{i,j\}\in E} (\mathbf{x}_i - \mathbf{x}_j) P(a_{ij} = 0|\mathbf{X}) \left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2} \right) \\
&+ 2 \sum_{j:\{i,j\}\notin E} (\mathbf{x}_i - \mathbf{x}_j) P(a_{ij} = 1|\mathbf{X}) \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right) \qquad (8)
\end{aligned}$$

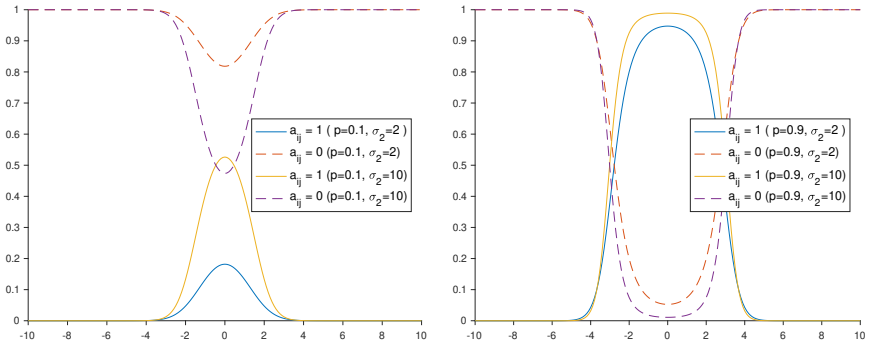

Figure 2: The posterior distribution $P(a_{ij} = 1|\mathbf{X})$ and $P(a_{ij} = 0|\mathbf{X})$ with different prior probability P_{ij} and σ_2

B Deriving the log probability of posterior $P(G|X)$

$$
\begin{aligned}
\log P(G|\mathbf{X}) &= \log\left(\prod_{\{i,j\}\in E}\frac{\mathcal{N}_{ij,\sigma_1}P_{ij}}{\mathcal{N}_{ij,\sigma_1}P_{ij}+\mathcal{N}_{ij,\sigma_2}(1-P_{ij})}\cdot\prod_{\{k,l\}\notin E}\frac{\mathcal{N}_{kl,\sigma_2}(1-P_{kl})}{\mathcal{N}_{kl,\sigma_1}P_{kl}+\mathcal{N}_{kl,\sigma_2}(1-P_{kl})}\right)\\
&= \log\left(\prod_{\{i,j\}\in E}\frac{1}{1+\frac{\mathcal{N}_{ij,\sigma_2}(1-P_{ij})}{\mathcal{N}_{ij,\sigma_1}P_{ij}}}\cdot\prod_{\{k,l\}\notin E}\frac{1}{1+\frac{\mathcal{N}_{kl,\sigma_1}P_{kl}}{\mathcal{N}_{kl,\sigma_2}(1-P_{kl})}}\right)\\
&= -\sum_{\{i,j\}\in E}\log\left(1+\frac{(2\pi\sigma_2^2)^{-1/2}\exp\left(-d_{ij}^2/(2\sigma_2^2)\right)(1-P_{ij})}{(2\pi\sigma_1^2)^{-1/2}\exp\left(-d_{ij}^2/(2\sigma_1^2)\right)P_{ij}}\right)\\
&\quad-\sum_{\{k,l\}\notin E}\log\left(1+\frac{(2\pi\sigma_1^2)^{-1/2}\exp\left(-d_{kl}^2/(2\sigma_1^2)\right)P_{kl}}{(2\pi\sigma_2^2)^{-1/2}\exp\left(-d_{kl}^2/(2\sigma_2^2)\right)(1-P_{kl})}\right)\\
&= -\sum_{\{i,j\}\in E}\log\left(1+\frac{\sigma_1}{\sigma_2}\frac{1-P_{ij}}{P_{ij}}\exp\left(\left(\frac{1}{\sigma_1^2}-\frac{1}{\sigma_2^2}\right)\frac{d_{ij}^2}{2}\right)\right)\\
&\quad-\sum_{\{k,l\}\notin E}\log\left(1+\frac{\sigma_2}{\sigma_1}\frac{P_{kl}}{1-P_{kl}}\exp\left(\left(\frac{1}{\sigma_2^2}-\frac{1}{\sigma_1^2}\right)\frac{d_{kl}^2}{2}\right)\right) \qquad (9)
\end{aligned}
$$

C Effects of the σ_1 and σ_2 parameters

CNE seeks the embedding $\mathbf{X}$ that maximizes the likelihood $P(G|\mathbf{X})$ for given G. To understand the effect of parameter σ_1 and σ_2 we plot the posterior $P(a_{ij} = 1|\mathbf{X})$ as well as $P(a_{ij} = 0|\mathbf{X})$ in Figure 2. The plot shows a large σ_2 corresponds to more extreme minima of the objective function (Fig2a), thus results in stronger push and pulling effect in the optimization. Large link probability in the network prior further strengthen the pushing and pulling effects (Fig 2b). The flat area in Figure 2b ($\sigma_2 = 10$) allows connected nodes to keep some small distance from each other, and larger σ_2 also allows larger corrections to the prior probabilities (both Fig 2a and Fig 2b), but also makes the optimization problem harder.

D Baseline methods used in experiments

We used the following baselines in the experiments:

- Deepwalk [18]: This embedding algorithm learns embedding based on the similarities between nodes. The proximities are measured by random walks. The transition probability of walking from one node to all its neighbors are the same and are based on one-hop connectivity.
- LINE [20]: Instead of random walks, this algorithm defines similarity between nodes based on first and second order adjacencies of the given network.
- node2vec [9]: This is again based on random walks. In addition to its predecessors, it offers two parameters p, q that interpolates the importance of BFS and DFS like random walk in the learning.
- metapath2vec++ [7]: This approach is developed for heterogeneous NE, namely, the nodes belong to different node types. methapath2vec++ performs random walks by hopping from a node form one type to a node from another type. It also utilizes the node type information in the softmax based objective function.
- struc2vec [19]: The method first measures the structural information by computing pairwise similarity between nodes using a range of neighborhood sizes. This results in a multilayer weighted graph where the edge weights on the same layer are derived from the node similarity computed on one neighborhood size. Then the embedding is constructed by a random walk strategy that navigates the multilayer graph.

E Networks used in the experiments

We used the following commonly used benchmark networks in the experiments:

- Facebook [12]: In this network, nodes are the users and links represent the friendships between the users. The network has 4,039 nodes and 88,234 links.
- arXiv ASTRO-PH [12]: In this network nodes represent authors of papers submitted to arXiv. The links represents the collaborations: two authors are connected if they co-authored at least one paper. The network has 18,722 nodes and 198,110 links.
- studentdb [8]: This is a snapshot of the student database from the University of Antwerp's Computer Science department. There are 403 nodes that

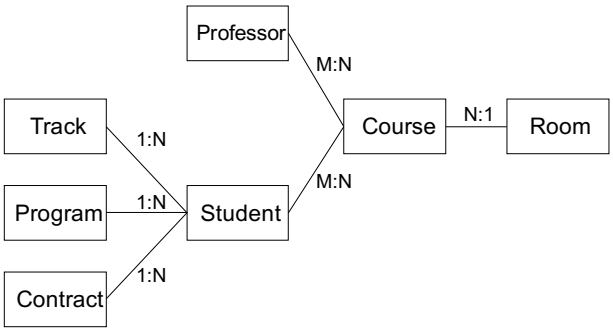

Figure 3: The entity relationship diagram of the studentdb dataset.

belong to one of the following node types including: course, student, professor, program, track, contract, and room. There 3429 links that are the binary relationships between the nodes: student-in-track, student-in-program, student-in-contract, student-take-course, professor-teach-course, course-in-room. The database schema is given in Figure 3.

- Gowalla [6]: This is a undirected location-based friendship network. The network has 196,591 nodes, 950,327 links.

- BlogCatalog [23]: This social network contains nodes representing bloggers and links representing their relations with other bloggers. The labels are the bloggers' interests inferred from the meta data. The network has 10,312 nodes, 333,983 links, and 39 labels (used for multi-label classifications).

- Protein-Protein Interactions (PPI) [4]: A subnetwork of the PPI network for Homo Sapiens. The subnetwork has 3,890 nodes, 76,584 links, and 50 labels.

- Wikipedia [14]: This network contains nodes representing words and links representing the co-occurrence of words in Wikipedia pages. The labels represents the inferred Part-of-Speech tags [21]. The network has 4,777 nodes, 184,812 links, and 40 different labels.

F Detailed results for multi-label classification

In the multi-label classification setting, each node is assigned one or more labels. For training, 50% of the nodes and all their labels are used for training. The labels of the remaining nodes need to be predicted. We train CNE and baselines based on the full network. Then 50% of the nodes are randomly selected to train a L2 regularized logistic regression classifier. The regularization strength parameter of

the classifier is trained with 10-fold cross-validation (CV) on the training data. We report the Macro-F_1 and Micro-F_1 based on the predictions. For the logistic regression classifier [sklearn, 17] we require every fold to have at least one positive and one negative label and we removed the labels that occur fewer than 10 times (number of folds in CV) in the data.

The detailed results of this approach based on logistic regression are shown in the upper half of Table 4. For CNE (written as CNE-LR to emphasize logistic regression was used for classifying), the embeddings are obtained with $d = 32$ and $k = 150$ (without optimizing). Somewhat surprisingly, CNE still performs in line with the state-of-the-art graph embedded methods, although without improving on them (on BlogCatalog, CNE performs third out of five methods, in PPI and Wikipedia it performs fourth out of five). This is surprising, given the fact that CNE yields embeddings that, by design, do not reflect certain information about the nodes that may be useful in classifying (here, their degree).

Multi-label classification can however be cast as a link prediction problem—a task we know CNE performs well at. To do this, we insert a node into the network corresponding to each of labels, and link the original nodes to the label nodes if they have that label. We can then employ link prediction, exactly as in the link prediction case (training on the full network, but with only 50% of the edges between original nodes and label nodes, and the other half for testing), to do multi-label classification. For CNE, besides a degree prior, we can encode a 'block' prior which encodes the average connectivity between original nodes–original nodes, original nodes–labels, and labels–labels (which is zero, as labels are not connected to each other). Note that this approach means that also neighborhood-based link prediction methods can be used for multi-label classification.

The detailed results of this link prediction approach to multi-label classification are shown in the lower half of Table 4. CNE-LP (block+degree) (with LP to indicate it is based on link prediction) consistently outperforms all baselines on Macro-F_1, while on Micro-F_1 it is best on two datasets (BlogCatalog and PPI), and close second-best on one (Wikipedia). We note that while the benefit of this link prediction approach to multi-label classification is clear (and unsurprising) for CNE, there is no consistent benefit to other methods. This shows that the superior performance of CNE-LP for multi-label classification is not (or at least not exclusively) thanks to the link prediction approach, but at least in part also thanks to a more informative embedding when considered in combination with the prior.

G Runtime experiment

We compare the runtime (in second) of CNE with other baselines in this section. We use the parameters settings in link prediction task for all methods. Namely, for CNE, we set $d = 8$ (For arXiv $k = 16$ to reduce underfitting) and $k = 50$. We

Algorithm	BlogCatalog		PPI		Wikipedia	
	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1	Macro-F_1	Micro-F_1
Multi-label classification using logistic regression (standard approach):						
Deepwalk	0.2544	0.3950	0.1795	0.2248	0.1872	0.4661
LINE	0.1495	0.2947	0.1547	0.2047	0.1721	**0.5193**
node2vec	0.2364	0.3880	0.1844	0.2353	0.1985	0.4746
metapath2vec++	0.0351	0.1684	0.0337	0.0726	0.1031	0.3942
struc2vec	0.0493	0.1653	0.0669	0.0971	0.1124	0.4019
CNE-LR (degree)	0.1833	0.3376	0.1484	0.1952	0.1370	0.4339
Multi-label classification through link prediction where labels are nodes:						
Common Neigh.	0.2115	0.2931	0.1792	0.1831	0.1212	0.3332
Jaccard Sim.	0.2157	0.1915	0.1799	0.1642	0.0552	0.0486
Adamic Adar	0.2301	0.3198	0.1698	0.1825	0.1035	0.3264
Preferential Attach.	0.2460	0.2084	0.2504	0.0953	0.2890	0.4454
Deepwalk	0.2372	0.2407	0.1848	0.1648	0.0876	0.0440
LINE	0.1599	0.2457	0.1052	0.1100	0.0976	0.2954
node2vec	0.2490	0.3462	0.2081	0.2069	0.1640	0.3057
metapath2vec++	0.0633	0.1415	0.0571	0.0542	0.2021	0.3673
struc2vec	0.0644	0.1100	0.0631	0.0757	0.0905	0.3485
CNE-LP (degree)	0.2839	0.3929	0.2139	0.2303	0.1825	0.4407
CNE-LP (block+degree)	**0.2935**	**0.4002**	**0.2639**	**0.2519**	**0.3374**	0.4839

Table 4: The F_1 scores for multi-label classification.

Algorithm	Facebook	PPI	arXiv	BlogCat.	Wikiped.	studentdb	Gowalla
Deepwalk	120.78	116.09	714.68	344.72	138.89	8.34	5717.67
LINE	253.20	203.92	649.98	218.20	232.11	180.35	10988.71
node2vec	86.61	64.96	291.42	1054.73	288.32	6.04	5593.52
metapath2vec++	130.78	39.59	274.60	332.19	78.14	3.50	333.29
struc2vec	2692.96	1105.41	54218.82	1356.67	1691.79	9245.23	TimeOut
CNE (uniform)	86.89	75.15	728.74	227.11	92.35	7.25	642.14
CNE (degree)	77.80	70.35	579.85	204.48	87.69	6.80	670.26
CNE (block)	NA	NA	NA	NA	NA	10.68	NA

Table 5: The runtime (in seconds) of embedding methods. TimeOut means aborted after 24 hours.

set stopping criterion of CNE $||\nabla_{\mathbf{X}}||_\infty < 10^{-2}$ or maxIter < 250 (whichever is met first). These stopping criteria yield embeddings with the same performance in link prediction tasks as reported in the chapter. For other methods, we use the default setting as reported in their original paper. The hyper-parameters p, q of node2vec are tuned using cross validation. This experiment is performed with single process/thread on a desktop with CPU 2,7 GHz Intel Core i5 and RAM 16 GB 1600 MHz DDR3. Table 5 summarizes the runtime of all methods against all datasets we used in the chapter. Over the seven datasets CNE is fastest in two cases, 12% slower than the fastest in one case (metapath2vec++), and approximately twice slower in the four other cases (also metapath2vec++).

References

[1] Florian Adriaens, Jefrey Lijffijt, and Tijl De Bie. Subjectively interesting connecting trees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 53–69. Springer, 2017.

[2] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48. ACM, 2013.

[3] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[4] Bobby-Joe Breitkreutz, Chris Stark, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, Michael Livstone, Rose Oughtred, Daniel H Lackner, Jürg Bähler, Valerie Wood, et al. The biogrid interaction database: 2008 update. *Nucleic acids research*, 36:D637–D640, 2007.

[5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[6] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2017.

[8] Bart Goethals, Wim Le Page, and Michael Mampaey. Mining interesting sets and rules in relational databases. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 997–1001, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-639-7. doi: 10.1145/1774088.1774299. URL `http://doi.acm.org/10.1145/1774088.1774299`.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[10] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[11] FC Leone, LS Nelson, and RB Nottingham. The folded normal distribution. *Technometrics*, 3(4):543–550, 1961.

[12] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, 2015.

[13] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[14] Matt Mahoney. Large text compression benchmark. *URL: http://www. mattmahoney. net/text/text. html*, 2011.

[15] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

[16] K Peason. On lines and planes of closest fit to systems of point in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[17] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[18] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[19] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM, 2017.

[20] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[21] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In

Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 173–180. Association for Computational Linguistics, 2003.

[22] Matthijs van Leeuwen, Tijl De Bie, Eirini Spyropoulou, and Cédric Mesnage. Subjective interestingness of subgraph patterns. *Machine Learning*, 105(1): 41–75, 2016.

[23] Reza Zafarani and Huan Liu. Social computing data repository at asu, 2009.

5

Representation Learning with Human in the Loop

A Constrained Randomization Approach to Interactive Visual Data Exploration with Subjective Feedback

Abstract Data visualization and iterative/interactive data mining are growing rapidly in attention, both in research as well as in industry. However, while there are plethora of advanced data mining methods and lots of works in the field of visualisation, integrated methods that combine advanced visualization and/or interaction with data mining techniques in a principled way are rare. We present a framework based on *constrained randomization* which lets users explore high-dimensional data via 'subjectively informative' two-dimensional data visualizations. The user is presented with 'interesting' projections, allowing users to express their observations using visual interactions that update a background model representing the user's belief state. This background model is then considered by a projection-finding algorithm employing data randomization to compute a new 'interesting' projection. By providing users with information that contrasts with the background model, we maximize the chance that the user encounters striking new information present in the data. This process can be iterated until the user runs out of time or until the difference between the randomized and the real data is insignificant. We present two case studies, one controlled study on synthetic data and another on census data, using the proof-of-concept tool SIDE that demonstrates the presented

framework.

5.1 Introduction

Data visualization and iterative/interactive data mining are both mature, actively researched topics of great practical importance. However, while progress in both fields is abundant, methods that combine them in a principled manner are rare.

Yet, methods that combine state-of-the-art data mining with visualization and interaction are highly desirable as they could exploit the strengths of both human data analysts and of computer algorithms. Humans are unmatched in spotting interesting patterns in low-dimensional visual representations, but poor at reading high-dimensional data, while computers excel in manipulating high-dimensional data and are weaker at identifying patterns that are truly relevant to the user. A symbiosis of human analysts and well-designed computer systems thus promises to provide the most efficient way of navigating the complex information space hidden within high-dimensional data. This idea has been advocated within the visual analytics field already a long time ago [39, 22, 32].

Contributions. In this chapter we introduce a generically applicable method based on constrained randomizations for finding interesting projections of data, given some prior knowledge about that data. We present use cases of interactive visual exploration of high-dimensional data with the aid of a proof-of-concept tool [20] that demonstrates the presented framework. The method's aim is to aid users in discovering structure in the data that the user was previously unaware of.

Overview of the method. The underlying idea is that the analysis process is iterative, and during each iteration there are three steps (Fig. 5.1).

Step 1. The user is presented with an 'interesting' projection of the data, visualized as a scatter plot. Here, interestingness is formalized with respect to the initial belief state and the scatter plot shows *projections of the data to which the data and the background model differ most*.

Step 2. The user investigates this scatter plot, and may observe structure in the

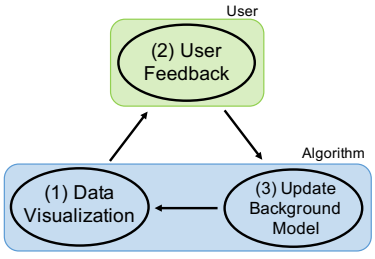

Figure 5.1: The three steps of SIDE's operation cycle.

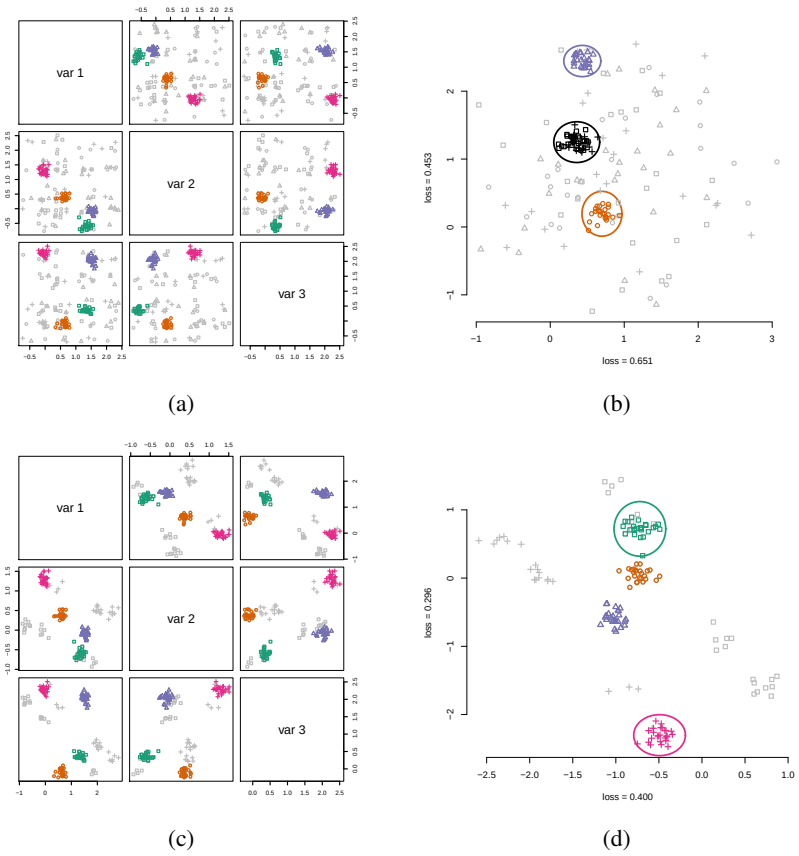

Figure 5.2: (a) Pairwise scatter plots of a 3-dimensional toy dataset that contains four clusters (indicated by different glyphs/colors). The initial random background model is shown with gray glyphs. (b) Two-dimensional projection to a direction where the data and the background model differ most. The user marks three clusters visible in the scatterplot as shown by ellipsoids. Two of the clusters (blue triangles and orange circles) correspond to the actual clusters of the toy data, but the third cluster (black) is a combination of two clusters (green boxes and cyan crosses). (c) The information of the three clusters has been absorbed into the background model which now shows more structure. (d) The next projection shows the largest difference between the updated background model and the data, which now clearly highlights the difference between the green (box) and cyan (cross) clusters, formerly presented in Fig. 2b to be one (black) cluster. The points in the orange (circle) and violet (triangle) clusters are exactly on top of the respective background distribution points. After marking these cluster with ellipsoids the user has completely understood the structure of the data and after updating the background model matches the data.

data that contrasts with, or add to, their beliefs about the data. We will refer to observed structures or features as *patterns*. The user then indicates what patterns

the user has seen.

Step 3. The background model is updated according to the user feedback given above, in order to reflect the newly assimilated information.

Next iteration. Then, the most interesting projection with respect to this updated background model can be computed, and the cyclic process iterates until the user runs out of time or finds that background model (and thus the user's belief state) explains everything the user is currently interested in.

Central objective. Our main goal is to support serendipity, i.e., the discovery of new knowledge 'by chance'. However, instead of user randomly guessing feature combinations that may yield interesting visualizations, we employ an algorithm that provides projection vectors that provide maximally contrasting information against an evolving background model. The central idea is that this increases the chances of finding truly interesting patterns in the data.

Example. Consider the 3-dimensional dataset of four clusters shown in Fig. 5.2. The raw data and the initial background model are shown in Fig. 5.2a. The clusters are shown with colored glyphs and the background model that reflects the user's initial beliefs is shown with gray markers. Initially, the background model is totally random (no beliefs).

Step 1 is that the user is presented with an initial scatter plot as shown in Fig. 5.2b. In step 2, the user marks clusters, as shown also in Fig. 5.2b. Step 3 is that the background model is updated based on this feedback, which results in a new background distribution (Fig. 5.2c). In the next iteration, the process repeats itself; steps 1 and 2 of the second iteration are shown in Fig. 5.2d.

To illustrate the stepwise process, this example was constructed such that the cluster structure of the data is obvious in any pairwise scatter plot. However, the objective is that the user can efficiently explore the data, also if the data has very high dimensionality. In that case, it is beneficial that an algorithm computes meaningful axes (i.e., *interesting projections*) to use for visualization. In Section 5.3 we present more extensive walkthrough examples on both synthetic and real data.

Formalization of the background model. To compute interesting projections, a crucial challenge is the formalization of the background model. To allow the process to be iterative, the formalization has to allow for the model to be updated after a user has given feedback on the visualization. There exist two frameworks for iterative data mining: FORSIED [6, 7] and a framework that we will refer to as CORAND [14, 24], for COnstrained RANDomization.

In both cases, the background model is a probability distribution over datasets and the user beliefs are modelled as a set of *constraints* on that distribution. The CORAND approach is to specify a randomization procedure that, when applied to the data, does not affect how plausible the user would deem it to be. That is, the user's beliefs should be satisfied, and otherwise the data should be shuffled as much as possible.

Given an appropriate randomization scheme, we can then find interesting remaining structure that is not yet known to the user by contrasting the real data with the randomized data. A most interesting projection can be computed by defining an optimization problem over the difference between the real data and the randomized data. Here, the optimization criterion is chosen as the maximal L_1-distance over the empirical cumulative distributions.

New beliefs can be incorporated in the background model by adding corresponding constraints to the randomization procedure, ensuring that the patterns observed by the user are present also in the subsequent randomized data. Hence, subsequent projection will again be informative because the randomized and the real data will be equivalent with respect to the statistics already known to the user.
Outline of this chapter As discussed in Section 5.2, three challenges had to be addressed to use the CORAND approach: (1) defining intuitive pattern types (constraints) that can be observed and specified based on a scatter plot of a two-dimensional projection of the data; (2) defining a suitable randomization scheme, that can be constrained to take account of such patterns; and (3) a way to identify the most interesting projections given the background model. The evaluation with respect to usefulness as well as computational properties of the resulting system is presented in Section 5.3. Experiments were conducted both on synthetic data and on a census dataset. Finally, related work and conclusions are discussed in Sections 5.4 and 5.5, respectively.

NB. This manuscript is an expanded and integrated version of two conference papers [33, 20]: [33] introduced the algorithmic problem, while [20] presented the proof-of-concept tool and interface. Besides the integration and changes throughout, the main differences are this new introduction and the introduction of a stopping criterion (Secs. 5.2.4, 5.3.5).

5.2 Methods

We will use the notational convention that upper case bold face symbols ($\mathbf{X}$) represent matrices, lower case bold face symbols ($\mathbf{x}$) represent column vectors, and lower case standard face symbols (x) represent scalars. We assume that our dataset consists of n d-dimensional data vectors $\mathbf{x}_i$. The dataset is represented by a real matrix $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_n^T \end{pmatrix}^T \in \mathbb{R}^{n \times d}$. More generally, we will denote the transpose of the ith row of any matrix $\mathbf{A}$ as $\mathbf{a}_i$ (i.e., $\mathbf{a}_i$ is a column vector). Finally, we will use the shorthand notation $[n] = \{1, \ldots, n\}$.

5.2.1 Projection tile patterns in two flavours

In the interaction step, the users declare that they have become aware of (and thus are no longer interested in seeing) the value of the projections of a set of points

onto a specific subspace of the data space. We call such information a *projection tile pattern* for reasons that will become clear later. A projection tile parametrizes a set of constraints to the randomization.

Formally, a projection tile pattern, denoted τ, is defined by a k-dimensional (with $k \leq d$) subspace of $\mathbb{R}^d$, and a subset of data points $\mathcal{I}_\tau \subseteq [n]$. We will formalize the k-dimensional subspace as the column space of an orthonormal matrix $\mathbf{W}_\tau \in \mathbb{R}^{d \times k}$ with $\mathbf{W}_\tau^T \mathbf{W}_\tau = \mathbf{I}$, and can thus denote the projection tile as $\tau = (\mathbf{W}_\tau, \mathcal{I}_\tau)$. We provide two ways in which the user can define the projection vectors $\mathbf{W}_\tau$ for a projection tile τ.

2D tiles. The first approach simply chooses $\mathbf{W}_\tau$ as the two weight vectors defining the projection within which the data vectors belonging to $\mathcal{I}_\tau$ were marked. This approach allows the user to simply specify that he or she knows the positions of that set of data points within this 2D projection. The user makes no further assumptions—they assimilate solely what they see without drawing conclusions not supported by direct evidence.

Clustering tiles. It is possible that after inspecting a cluster, the user concludes that these points are clustered *not just within the two dimensions shown* in the scatter plot, and wishes for the system to model immediately also other dimensions in which the selected point set forms a cohesive cluster. This would lead to the system not considering other projections that highlight this cluster as particularly informative. To allow the user to express such belief, the second approach takes $\mathbf{W}_\tau$ to additionally include a basis for other dimensions along which these data points are strongly clustered. This is achieved as follows.

Let $\mathbf{X}(\mathcal{I}_\tau, :)$ represent a matrix containing the rows indexed by elements from $\mathcal{I}_\tau$ from $\mathbf{X}$. Let $\mathbf{W} \in \mathbb{R}^{d \times 2}$ contain the two weight vectors onto which the data was projected for the current scatter plot. In addition to $\mathbf{W}$, we want to find any other dimensions along which these data vectors are clustered. These dimensions can be found as those along which the variance of these data points is not much larger than the variance of the projection $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}$.

To find these dimensions, we first project the data onto the subspace orthogonal to $\mathbf{W}$. Let us represent this subspace by a matrix with orthonormal columns, further denoted as $\mathbf{W}^\perp$. Thus, $\mathbf{W}^{\perp T}\mathbf{W}^\perp = \mathbf{I}$ and $\mathbf{W}^T\mathbf{W}^\perp = \mathbf{0}$. Then, Principal Component Analysis (PCA) is applied to the resulting matrix $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}^\perp$. The principal directions corresponding to a variance smaller than a threshold are then selected and stored as columns in a matrix $\mathbf{V}$. In other words, the variance of each of the columns of $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}^\perp\mathbf{V}$ is below the threshold.

The matrix $\mathbf{W}_\tau$ associated to the projection tile pattern is then taken to be:

$$\mathbf{W}_\tau = \begin{pmatrix} \mathbf{W} & \mathbf{W}^\perp\mathbf{V} \end{pmatrix}.$$

The threshold on the variance used could be a tunable parameter, but was set here to twice the average of the variance of the two dimensions of $\mathbf{X}(\mathcal{I}_\tau, :)\mathbf{W}$.

5.2.2 The randomization procedure

Here we describe the approach to randomizing the data. The randomized data should represent a sample from an implicitly defined background model that represents the user's belief state about the data. Initially, our approach assumes the user merely has an idea about the overall scale of the data. However, throughout the interactive exploration, the patterns in the data described by the projection tiles will be maintained in the randomization.

Initial randomization. The proposed randomization procedure is parametrized by n orthogonal rotation matrices $\mathbf{U}_i \in \mathbb{R}^{d\times d}$, where $i \in [n]$, and the matrices satisfy $(\mathbf{U}_i)^T = (\mathbf{U}_i)^{-1}$. We further assume that we have a bijective mapping $f : [n] \times [d] \mapsto [n] \times [d]$ that can be used to permute the indices of the data matrix. The randomization proceeds in three steps:

Random rotation of the rows: Each data vector $\mathbf{x}_i$ is rotated by multiplication with its corresponding random rotation matrix $\mathbf{U}_i$, leading to a randomised matrix $\mathbf{Y}$ with rows $\mathbf{y}_i^T$ that are defined by:

$$\forall i : \ \mathbf{y}_i = \mathbf{U}_i \mathbf{x}_i.$$

Global permutation: The matrix $\mathbf{Y}$ is further randomized by randomly permuting all its elements, leading to the matrix $\mathbf{Z}$ defined as:

$$\forall i, j : \ \mathbf{Z}_{i,j} = \mathbf{Y}_{f(i,j)}.$$

Inverse rotation of the rows: Each randomised data vector in $\mathbf{Z}$ is rotated with the inverse rotation applied in step 1, leading to the fully randomised matrix $\mathbf{X}^*$ with rows $\mathbf{x}_i^*$ defined as follows in terms of the rows $\mathbf{z}_i^T$ of $\mathbf{Z}$:

$$\forall i : \ \mathbf{x}_i^* = {\mathbf{U}_i}^T \mathbf{z}_i.$$

The random rotations $\mathbf{U}_i$ and the permutation f are sampled uniformly at random from all possible rotation matrices and permutations, respectively.

Intuitively, this randomization scheme preserves the scale of the data points. Indeed, the random rotations leave their lengths unchanged, and the global permutation subsequently shuffles the values of the d components of the rotated data points. Note that without the permutation step, the two rotation steps would undo each other such that $\mathbf{X}^* = \mathbf{X}$. Thus, it is the combined effect that results in a randomization of the dataset.

The random rotations may seem superfluous: the global permutation randomizes the data so dramatically that the added effect of the rotations is relatively unimportant. However, their role is to make it possible to formalize the growing understanding of the user as simple constraints on this randomization procedure, as discussed next.

Accounting for one projection tile. Once the user has assimilated the information in a projection tile $\tau = (\mathbf{W}_\tau, \mathcal{I}_\tau)$, the randomization scheme should incorporate this information by ensuring that it is present also in all randomized versions of the data. This ensures that the randomized data is a sample from a distribution representing the user's belief state about the data. This is achieved by imposing the following *constraints* on the parameters defining the randomization:

Rotation matrix constraints: For each $i \in \mathcal{I}_\tau$, the component of $\mathbf{x}_i$ that is within the column space of $\mathbf{W}_\tau$ must be mapped onto the first k dimensions of $\mathbf{y}_i = \mathbf{U}_i\mathbf{x}_i$ by the rotation matrix $\mathbf{U}_i$. This can be achieved by ensuring that:

$$\forall i \in \mathcal{I}_\tau : \ \mathbf{W}_\tau^T \mathbf{U}_i = (\mathbf{I} \ \ \mathbf{0}) \, . \tag{5.1}$$

This explains the name *projection tile*: the information to be preserved in the randomization is concentrated in a 'tile' (i.e., the intersection of a set of rows and a set of columns) in the intermediate matrix $\mathbf{Y}$ created during the randomization procedure.

Permutation constraints. The permutation should not affect any matrix cells with row indices $i \in \mathcal{I}_\tau$ and columns indices $j \in [k]$:

$$\forall i \in \mathcal{I}_\tau, j \in [k] : \ f(i,j) = (i,j). \tag{5.2}$$

Proposition 3. *Using the above constraints on the rotation matrices* $\mathbf{U}_i$ *and the permutation* f*, it holds that:*

$$\forall i \in \mathcal{I}_\tau, \mathbf{x}_i^T \mathbf{W}_\tau = \mathbf{x}_i^{*T} \mathbf{W}_\tau. \tag{5.3}$$

Thus, the values of the projections of the points in the projection tile remain unaltered by the constrained randomization. Hence, the randomization keeps the user's beliefs intact. We omit the proof as the more general Proposition 4 is provided with proof further below.

Accounting for multiple projection tiles. Throughout subsequent iterations, additional projection tile patterns will be specified by the user. A set of tiles τ_i for which $\mathcal{I}_{\tau_i} \cap \mathcal{I}_{\tau_j} = \emptyset$ if $i \neq j$ is straightforwardly combined by applying the relevant constraints on the rotation matrices to the respective rows. When the sets of data points affected by the projection tiles overlap though, the constraints on the rotation matrices need to be combined. The aim of such a combined constraint should be to preserve the values of the projections onto the projection directions for *each* of the projection tiles a data vector was part of.

The combined effect of a set of tiles will thus be that the constraint on the rotation matrix $\mathbf{U}_i$ will vary per data vector, and depends on the set of projections $\mathbf{W}_\tau$ for which $i \in \mathcal{I}_\tau$. More specifically, we propose to use the following constraint on the rotation matrices:

Rotation matrix constraints. Let $\mathbf{W}_i \in \mathbb{R}^{d \times d_i}$ denote a matrix of which the columns are an orthonormal basis for space spanned by the union of the columns of the matrices $\mathbf{W}_\tau$ for τ with $i \in \mathcal{I}_\tau$. Thus, for any i and $\tau : i \in \mathcal{I}_\tau$, it holds that $\mathbf{W}_\tau = \mathbf{W}_i \mathbf{v}_\tau$ for some $\mathbf{v}_\tau \in \mathbb{R}^{d_i}$. Then, for each data vector i, the rotation matrix $\mathbf{U}_i$ must satisfy:

$$\forall i \in \mathcal{I}_\tau : \ \mathbf{W}_i^T \mathbf{U}_i = (\mathbf{I} \ \mathbf{0}) \, . \tag{5.4}$$

Permutation constraints. Then the permutation should not affect any matrix cells in row i and columns $[d_i]$:

$$\forall i \in [n], j \in [d_i] : \ f(i,j) = (i,j).$$

Proposition 4. *Using the above constraints on the rotation matrices $\mathbf{U}_i$ and the permutation f, it holds that:*

$$\forall \tau, \forall i \in \mathcal{I}_\tau, \mathbf{x}_i^T \mathbf{W}_\tau = \mathbf{x}_i^{*T} \mathbf{W}_\tau.$$

Proof. We first show that $\mathbf{x}_i^{*T} \mathbf{W}_i = \mathbf{x}_i^T \mathbf{W}_i$:

$$\begin{aligned} \mathbf{x}_i^{*T} \mathbf{W}_i = \mathbf{z}_i^T \mathbf{U}_i^T \mathbf{W}_i = \mathbf{z}_i^T \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} \\ = \mathbf{z}_i (1 : d_i)^T = \mathbf{y}_i (1 : d_i)^T = \mathbf{y}_i^T \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} = \mathbf{x}_i^T \mathbf{W}_i. \end{aligned}$$

The result now follows from the fact that $\mathbf{W}_\tau = \mathbf{W}_i \mathbf{v}_\tau$ for some $\mathbf{v}_\tau \in \mathbb{R}^{d_i}$. □

Technical implementation of the randomization. To ensure the randomization can be carried out efficiently throughout the process, note that the matrix $\mathbf{W}_i$ for the $i \in \mathcal{I}_\tau$ for a new projection tile τ can be updated by computing an orthonormal basis for $(\mathbf{W}_i \ \ \mathbf{W})$. Such a basis can be found efficiently as the columns of $\mathbf{W}_i$ in addition to the columns of an orthonormal basis of $\mathbf{W} - \mathbf{W}_i^T \mathbf{W}_i \mathbf{W}$ (the components of $\mathbf{W}$ orthogonal to $\mathbf{W}_i$), the latter of which can be computed using the QR-decomposition.

Additionally, note that the tiles define an equivalence relation over the row indices, in which i and j are equivalent if they were included in the same set of projection tiles so far. Within each equivalence class, the matrix $\mathbf{W}_i$ will be constant, such that it suffices to compute it only once, tracking which points belong to which equivalence class.

5.2.3 Visualization: Finding the most interesting two-dimensional projection

Given the dataset $\mathbf{X}$ and the randomized dataset $\mathbf{X}^*$, it is now possible to quantify the extent to which the empirical distribution of a projection $\mathbf{X}\mathbf{w}$ and $\mathbf{X}^*\mathbf{w}$

onto a weight vector $\mathbf{w}$ differ. There are various ways in which this difference could be quantified. We investigated a number of possibilities and found that the L_1-distance between the cumulative distribution functions works well in practice. Thus, with $F_{\mathbf{x}}$ the empirical cumulative distribution function for the set of values in $\mathbf{x}$, the optimal projection is found by solving:

$$\max_{\mathbf{w}} \|F_{\mathbf{Xw}} - F_{\mathbf{X}^*\mathbf{w}}\|_1 .$$

The second dimension of the scatter plot can be sought by optimizing the same objective while requiring it to be orthogonal to the first dimension.

We are unaware of any special structure of this optimization problem that makes solving it particularly efficient. Yet, using the standard quasi-Newton solver in R [34] with random initialization and default settings (the general-purpose `optim` function with `method="BFGS"`), or the *numericjs* library for Javascript [27], already yields satisfactory results, as shown in the experiments below.

5.2.4 Significance of a projection and stopping criterion

Although it has not been written down before, it is conceptually straightforward in CORAND to assess the statistical significance of any pattern of interest (here projection), because it is always possible to compute the empirical p-value of a pattern under the background model.

This works as follows. Denote the score function of a pattern as $f(\mathbf{X}, \mathbf{X}^*)$, e.g., the optimized statistic is

$$f(\mathbf{X}, \mathbf{X}^*) = \max_{\mathbf{w}} \|F_{\mathbf{Xw}} - F_{\mathbf{X}^*\mathbf{w}}\|_1 .$$

This statistic hinges by definition on a comparison between the real data $\mathbf{X}$ and the randomized data $\mathbf{X}^*$. An important question is: *how surprising is this statistic?*

We can take the viewpoint that we are comparing a certain randomized dataset $\mathbf{X}^*$, which has no structure except for the constraints that we have defined so far, to another dataset $\mathbf{X}$. The question that we need to consider is, does the real data $\mathbf{X}$ still have interesting structure with respect to the pattern syntax? Essentially, we are asking whether $f(\mathbf{X}, \mathbf{X}^*)$ is surprising given the background model. Equivalently, if $\mathbf{X}$ would *not* contain interesting structure anymore, we expect $f(\mathbf{X}, \mathbf{X}^*)$ to be 'similar' to $f(\mathbf{X}^{*'}, \mathbf{X}^*)$, where $\mathbf{X}^{*'}$ is another randomized dataset from the same constraints.

This latter statement about similarity can be made quantified in an empirical p-value $\hat{p}$ [29, 24], where we compare $f(\mathbf{X}, \mathbf{X}^*)$ against $f(\mathbf{X}_1^{*'}, \mathbf{X}^*), \ldots, f(\mathbf{X}_N^{*'}, \mathbf{X}^*)$ with $\mathbf{X}_i^{*'}$ being a randomized version of $\mathbf{X}^*$, employing still the same constraints. A rationale why $\mathbf{X}_i^{*'}$ should derived from $\mathbf{X}^*$ and not from $\mathbf{X}$ can be found in [13].

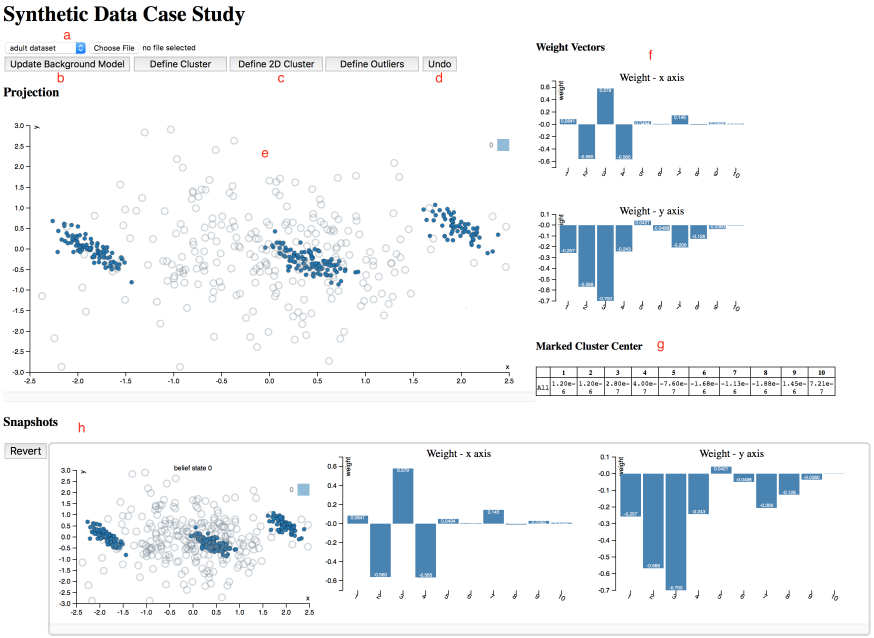

Figure 5.3: Layout of our web app SIDE*, with the data visualization and interaction area (a–e), projection meta information (f, g), and timeline (h).*

In full, given N randomizations to compare with, the empirical p-value is

$$\hat{p} = \frac{1 + \sum_{i=1}^{N} \mathbf{1}\left(f(\mathbf{X}_i^{*\prime}, \mathbf{X}^*) \geq f(\mathbf{X}, \mathbf{X}^*)\right)}{N+1}.$$

The two-dimensional scatterplot is based on two orthogonal projections that each have a different value $\|F_{\mathbf{Xw}} - F_{\mathbf{X}^*\mathbf{w}}\|_1$. These can be compared against the the series $f(\mathbf{X}_i^{*\prime}, \mathbf{X}^*)$ to obtain an empirical p-value for either axis. If the p-value for an axis is above a threshold that the user finds acceptable, e.g., 0.05, the values should not be studied. Since constraints can only be added, meaning the model will be closer to the data, the p-values should be roughly monotonic and the analysis can be terminated when the threshold is reached. See Section 5.3 for an example.

5.2.5 The risk of false positive observations

One may have the concern that even with the use of a stopping criterion, showing a user projections that hopefully contain meaningful structure can lead to—or even increase the chance to—the observation of patterns that are not real. There are three important aspects to consider here:

Figure	Cluster	Dim_1	Dim_2	Dim_3	Dim_4	Dim_5	Dim_6	Dim_7	Dim_8	Dim_9	Dim_10
5.4b	top (1)	0.250	0.467	-0.334	0.347	-0.00263	-0.0331	-0.0201	-0.0506	-0.00254	-0.0610
	mid (2)	-0.774	-1.45	1.03	-1.07	0.0815	0.103	0.0623	0.157	0.00787	0.189
	bottom (3)	0.348	0.0525	0.401	-0.329	0.0859	-0.0694	-0.0212	-0.0307	0.0557	-0.152

Table 5.1: Mean vectors of user marked clusters for the Synthetic data (Section 5.3.2).

1. The proposed approach makes no claims about causality. For example, the data may be biased, contain errors, there may be missing variables that could explain observed correlations and patterns. The projections may highlight information that is spurious in the sense that it pertains to the data collection process rather than the reality the data was intended to capture. However, this should be considered a positive feature, because learning about such artefacts in the data can be greatly beneficial. During interpretation of the patterns, one should always be cautious and aim to explain the observed patterns, instead of taking them at face value.

2. The patterning (i.e., arrangement) of the points in the visualizations shown to a user correspond to projections, which is simply a weighted combination of the original features. As such, only structure that is present in the data can be shown.

3. The prototype implementation introduced in the next section shows besides the data also the randomized version of the data that the projection is aimed to contrast with. In our experience, it is straightforward to visually observe whether the structure shown in the visualization has substantial magnitude as compared to the randomized data. As such, the stopping criterion can be used to make the system even more robust against the analysis of noise, but it is usually easy to see when the projections no longer pick up any significant structure, even without the stopping criterion. See for example Figure 5.7.

5.3 Experiments

We present two case studies to illustrate the framework and its utility. We first introduce a proof-of-concept tool and discuss how this tool implements the concepts presented in Section 5.2. A description of how the tool may be used in practice is interweaved with the subsequent case studies. Finally, we present an evaluation of the runtime performance and the stopping criterion.

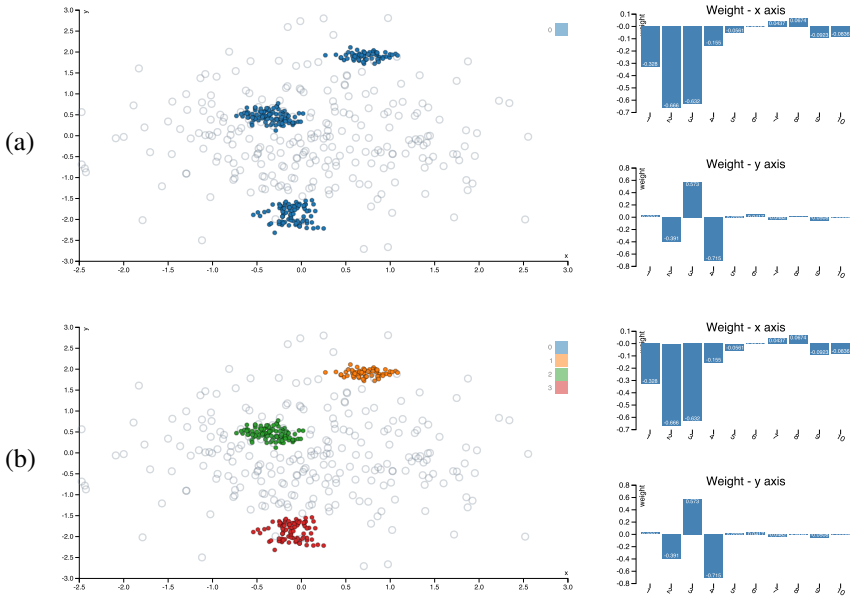

Figure 5.4: Example of the first visualization given by SIDE *on the synthetic data (Section 5.3.2). Solid dots represent actual data vectors, whereas open circles represent vectors from the randomized data. Row (a) shows the first visualization, while (b) shows the same visualization with the three clusters marked by us. Right of the scatter plot are bar charts that represent the weight vectors that constitute the projection vectors.*

5.3.1 Proof-of-concept tool SIDE

The case studies are completed with the a JavaScript version of our tool, which is available freely online, along with the used data for reproducibility.[1][20]

The full interface of SIDE is shown in Figure 5.3. SIDE was designed according to the three principles for 'visually controllable data mining' [32], which essentially state that both the model and the interactions should be transparent to users, and that the analysis method should be fast enough such that the user does not lose its trail of thought.

The main component is the interactive scatter plot (Figure 5.3e). The scatter plot visualizes the projected data (solid dots) and the randomized data (open gray circles) in the current 2D projection. By drawing a polygon, the user can select data points to define a *projection tile pattern*. Once a set of points is selected, the user can press either of the three feedback buttons (5.3c), to indicate these points form a cluster or to define them as outliers.

[1] `http://www.interesting-patterns.net/forsied/side/`

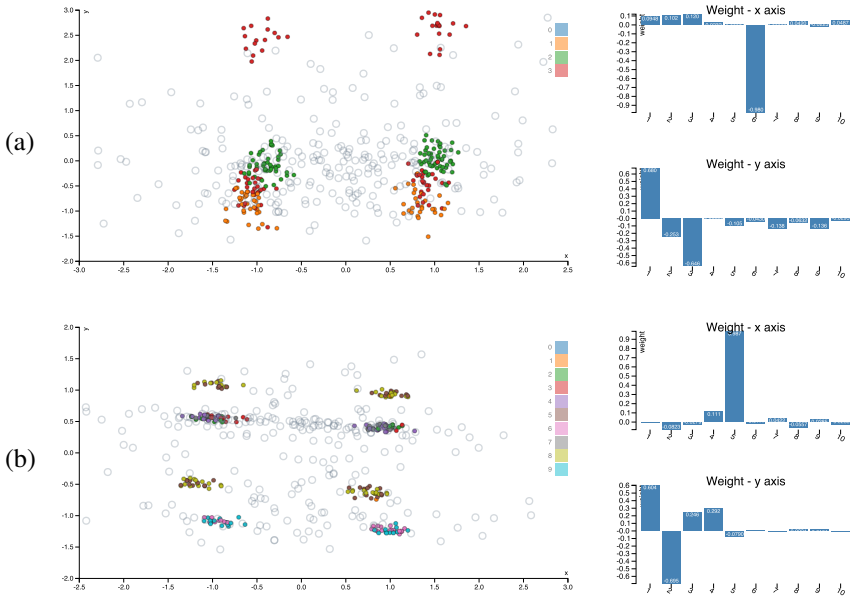

Figure 5.5: Continuation of the visualizations given by SIDE *on the synthetic data (Section 5.3.2). Rows (a) and (b) show the second and third visualization.*

If the user thinks the points are clustered only in the shown projection, they click 'Define 2D Cluster', while 'Define Cluster' indicates they expect that these points will be clustered in other dimensions as well. 'Define Outliers' fully fixes the location of the selected points in the background model to their actual values, such that those points do not affect the projections anymore.

To identify the defined clusters, those data points are given the same color, and their statistics are shown in a table (Figure 5.3g). The user can define multiple clusters in a single projection, and they can also *undo* (Figure 5.3d) the feedback. Once a user finishes exploring the current projection, they can press 'Update Background Model' (Figure 5.3b). Then, the background model is updated with the provided feedback and a new scatter plot is computed and presented to the user in an iterative fashion.

A few extra features are provided to assist the data exploration process: to gain an understanding of a projection, the weight vectors associated with the projection axes are plotted in bar charts (Figure 5.3f). Below those, a table (Figure 5.3g) lists the mean vectors of each colored point set (cluster). The exploration history is maintained by taking snapshots of the background model when updated, together with the associated data projection (scatter plot) and weight vectors (bar charts). This history in reverse chronological order is shown in Figure 5.3h.

The tool also allows a user to revert back to a certain snapshot, to restart from that time point. This allows the user to discover different aspects of a dataset more consistently. Finally, custom datasets can be loaded for analysis from the drop-down menu (Figure 5.3a). Currently our tool only works with CSV files and it automatically sub-samples the custom dataset so that the interactive experience is not compromised. By default, two datasets are preloaded so that users can get familiar with the tool. Notice that, since the tool runs locally in your browser and there are no server-side computations, you can safely analyse data that you cannot share or transmit elsewhere.

5.3.2 Synthetic data

In the first case study, we generated a synthetic dataset that consists of 1000 ten-dimensional data vectors of which dimensions 1–4 can be clustered into five clusters, dimensions 5–6 into four clusters *involving different subsets of data points*, and of which dimensions 7–10 are Gaussian noise. All dimensions have equal variance.

In Figure 5.4a we observe the initial visualization from SIDE. The blue dots are data points while the open circles correspond to a randomized version of the data. The randomized data points are shown in order to ground any observed patterns in the visualization because they show what we should be expecting given the background knowledge encoded thus far. As this is the initial visualization, the only encoded knowledge is the overall scale of the data.

Next to the visualization we find two bar charts that visualize the projection vectors corresponding to the x- and y-axis. We observe the x-axis has loadings mostly on dimensions 2 and 3 and to a lesser extent 1 and 4. The other loadings (dimensions 7–10) are so small they likely correspond to noise that is by chance slightly correlated to the cluster structure in dimensions 1–4. The y-axis is loaded onto dimensions 2–4.

The distribution of the projected data points clearly contrasts with the randomized data, indicating that probably the visualization is showing meaningful structure. Because the data is 10-dimensional while the scatter plot is 2-dimensional, we cannot be sure just from the visualization where in the original space the observed clusters are located. Hence, we mark the three clusters, as shown in Figure 5.4b.

Table 5.1 shows the mean vectors for each of the three clusters. Because this is synthetic data, the dimensions are meaningless, but normally it should be possible to understand what the clusters mean and how they differ from each other based on careful inspection of these numbers. Future use of the tool will have to show whether these mean statistics are sufficient, or whether additional information (e.g., variances) could be helpful or necessary.

Once we understand the meaning of the clusters, we ask for a new visualization

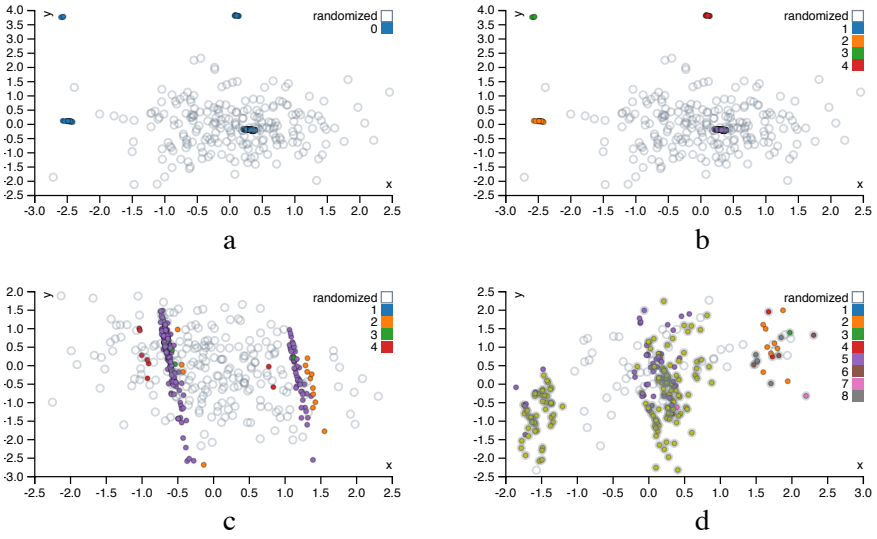

Figure 5.6: Projections of UCI Adult dataset: (a) projection in the $1st$ *iteration, (b) clusters marked by user in the* $1st$ *iteration, (c) projection in the* $2nd$ *iteration, and (d) projection in the* $3rd$ *iteration*

('Update background model' in the full interface shown in Figure 5.3), which is then based on a model that incorporates the marked structure.

The subsequent most interesting projection is given in Figure 5.5a. The x-axis corresponds almost purely to dimension 6, which together with dimension 5 contains the orthogonal cluster structure. The y-axis again corresponds to a subspace of dimensions 1–4, highlighting that indeed the red cluster actually consists of two parts.

If we mark the four clusters shown in Figure 5.5a, our model will contain eight clusters: the red cluster breaks into four parts, the green and orange into two each. In Figure 5.5b we recover the remaining structure in the data; the x-axis (dimension 5) divides each of the already defined clusters into two, and on the y-axis, there is again a subspace of dimensions 1–4, which splits the brown-yellow cluster into two, while the others are unaffected.

We designed this example to illustrate the feedback that a user can give using the constrained randomizations. Additionally, it shows how the methods succeeds in finding interesting projections given previously identified patterns. Thirdly, it also demonstrates how the user interactions meaningfully affect subsequent visualizations.

Figure	axis	Age	Edu.	h/w	EG_AsPl	EG_Bl.	EG_Oth.	EG_Whi.	Gender	Income
5.6a	X	-0.039	-0.001	0.001	**0.312**	**-0.530**	**-0.193**	**0.763**	0.017	0.008
	Y	0.004	-0.004	-0.002	**0.816**	**-0.141**	**0.465**	**-0.313**	-0.011	0.002
5.6c	X	0.081	-0.028	-0.022	-0.259	-0.233	-0.104	-0.380	**-0.846**	-0.001
	Y	-0.590	0.541	0.143	-0.233	-0.380	-0.026	-0.293	0.232	0.000
5.6d	X	0.119	-0.149	0.047	0.102	0.191	0.104	**-0.556**	0.0581	**-0.769**
	Y	**-0.382**	**-0.626**	**-0.406**	0.346	0.317	-0.0287	0.111	-0.248	0.059

Table 5.2: Projection weight vectors for the UCI Adult data (Section 5.3.3).

Figure	Cluster	Age	Edu.	h/w	EG_AsPl	EG_Bl.	EG_Oth.	EG_Whi.	Gender	Income
5.6b	top left	35.0	8.67	34.7	0.00	0.00	**1.00**	0.00	0.667	0.333
	bott. left	37.2	9.43	40.3	0.00	**1.00**	0.00	0.00	0.286	0.071
	top right	35.6	1.3	51.1	**1.00**	0.00	0.00	0.00	0.750	0.250
	bott. right	38.4	10.2	41.6	0.00	0.00	0.00	**1.00**	0.762	0.275
5.6c	left	39.0	10.2	43.3	0.0377	0.0252	0.0126	0.925	**1.00**	0.321
	right	36.0	9.95	37.9	0.0339	0.169	0.0169	0.780	**0.00**	0.102
5.6d	left	**42.5**	**11.6**	**46.3**	0.00	0.00	0.00	**1.00**	1.00	**1.00**

Table 5.3: Mean vectors of user marked clusters for the UCI Adult data (Section 5.3.3).

5.3.3 UCI Adult data

In this case study, we demonstrate the utility of our method by exploring a real world dataset. The data is compiled from the UCI Adult dataset[2]. To ensure the real time interactivity, we sub-sampled 218 data points and selected six features: "Age" $(17-90)$, "Education" $(1-16)$, "HoursPerWeek" $(1-99)$, "Ethnic Group" (White, AsianPacIslander, Black, Other), "Gender" (Female, Male), "Income" $(\geq 50k)$. Among the selected features, "Ethnic Group" is a categorical feature with five categories, "Gender" and "Income" are binary features, the rest are all numeric. To make our method applicable to this dataset, we further binarized the "Ethnic Group" feature (yielding four binary features), and the final dataset consists of 218 points and 9 features.

We assume the user uses clustering tiles throughout the exploration. Each of the patterns discovered during the exploration process corresponds to a certain demographic clustering pattern. To illustrate how the constrained randomizations help the user rapidly gain an understanding of the data, we discuss the first three iterations of the exploration process. The first projection (Figure 5.6a) visually consists of four clusters. The user notes that the weight vectors corresponding to the axes of the plot assign large weights to the "Ethnic Group" attributes (Table 5.2, 1st row). As mentioned, we assume the user marks these points as part of the same cluster. After marking (Figure 5.6b), the tool informs the user of the mean vectors of the points within each clustering tile. The 1st row of Table 5.3 shows that each cluster completely represents one out of four ethnic groups, which may corroborate with the user's understanding.

Taking the user's feedback into consideration, a new projection is generated.

[2] `https://archive.ics.uci.edu/ml/datasets/Adult`

		rand.	$k \in \{2, 4, 8, 16\}$	
n	d	(s)	optim. (s)	#tries $\Delta < 1\%$
64	16	0.1	$\{1.0, 1.2, 0.9, 1.2\}$	$\{10, 10, 9, 8\}$
64	32	0.5	$\{1.8, 2.1, 2.4, 2.5\}$	$\{10, 8, 10, 10\}$
64	64	2.5	$\{5.6, 3.5, 4.6, 4.5\}$	$\{10, 9, 10, 8\}$
64	128	11.5	$\{8.9, 10.1, 11.4, 10.2\}$	$\{10, 10, 8, 9\}$
128	16	0.2	$\{2.0, 1.7, 2.4, 2.0\}$	$\{10, 1, 6, 8\}$
128	32	0.8	$\{2.6, 3.5, 4.0, 4.8\}$	$\{9, 10, 10, 10\}$
128	64	5.1	$\{6.7, 5.3, 8.3, 9.6\}$	$\{8, 10, 10, 9\}$
128	128	24.5	$\{13.8, 17.4, 15.2, 20.4\}$	$\{10, 9, 10, 7\}$
256	16	0.4	$\{4.3, 2.6, 3.3, 4.7\}$	$\{10, 8, 10, 9\}$
256	32	1.8	$\{6.3, 8.2, 7.9, 8.8\}$	$\{8, 9, 10, 10\}$
256	64	9.2	$\{12.4, 10.1, 19.2, 16.3\}$	$\{10, 10, 10, 9\}$
256	128	39.9	$\{33.5, 36.3, 30.6, 35.6\}$	$\{10, 9, 8, 9\}$
512	16	0.5	$\{6.7, 6.3, 6.1, 7.5\}$	$\{10, 9, 10, 10\}$
512	32	2.4	$\{16.6, 19.6, 20.2, 17.5\}$	$\{9, 9, 10, 10\}$
512	64	13.6	$\{34.9, 23.5, 22.3, 41.0\}$	$\{10, 10, 8, 7\}$
512	128	68.0	$\{74.5, 68.1, 72.3, 62.8\}$	$\{10, 1, 9, 9\}$

Table 5.4: Median wall clock running times, for randomization and optimization over ten iterations of finding 2D-projections using L_1 loss. Also shown is the number of iterations in which the L_1 norm first component ended up within 1% of the result with the largest L_1 norm (out of 10 tries). A high number indicates the solution quality is stable, even though the actual projections may vary.

The new scatter plot (Figure 5.6c) shows two large clusters, each consisting of some points from the previous four-cluster structure (points from these four clusters are colored differently). Thus, the new scatter plot elucidates structure not shown in the previous one. Indeed, the weight vectors (2nd row of Table 5.2) show that the clusters are separated mainly according to the "Gender" attribute. After marking the two clusters separately, the mean vector of each cluster (2nd row of Table 5.3) confirms this: the cluster on the left represents male group, and the female group is on the right. Notice that these clusters also yield other meaningful information, because the projection vectors not only correspond to gender (Table 5.2, 2nd row). We find in the table of cluster means (Table 5.3, 2nd row) that the genders are skewed over age, ethnicity, and income.

The projection in the third iteration (Figure 5.6d) consists of three clusters, separated only along the x-axis. Interestingly, the corresponding weight vector (3rd row of Table 5.2) has strongly negative weights for the attributes "Income"

and "Ethnic Group - White". This indicates the left cluster mainly represents the people with high income and whose ethnic group is also "White". This cluster has relatively low y-value; i.e., they are also generally older and more highly educated. These observations are corroborated by the cluster mean (Table 5.3, 3rd row).

For this case study, we also measured the performance of SIDE in three components: loading data, fit background model then compute new projection, update visualizations. We repeated the experiment (with two iterations each) ten times on a desktop with 2.7 GHz Intel Core i5 processor and recorded the wall clock time. On average, loading Adult dataset takes 11ms, fitting the background model then computing the new projection takes 7.0s, updating the visualization takes 41ms.

This case study illustrates how the proposed constrained randomization methods facilitates human data exploration by iteratively presenting an informative projection, considering what the user has already learned about the data.

5.3.4 Performance on synthetic data

Ideally any interactive data exploration tool should work in close to real time. This section contains an empirical analysis of an (unoptimized) R implementation of the method, as a function of the size, dimensionality, and complexity of the data. Note that limits on screen resolution as well as on human visual perception render it useless to display more than of the order of a few hundred data vectors, such that larger datasets can be down-sampled without noticeably affecting the content of the visualizations.

We evaluated the scalability on synthetic data with $d \in \{16, 32, 64, 128\}$ dimensions and $n \in \{64, 128, 256, 512\}$ data points scattered around $k \in \{2, 4, 8, 16\}$ randomly drawn cluster centroids (Table 5.4). The randomization is done here with the initial background model. The most costly part in randomization is usually the multiplication of orthogonal matrices, indeed, the running time of the randomization scales roughly as nd^x, where x is between 2 and 3. The results suggests that the running time of the optimization is roughly proportional to the size of the data matrix nd and that the complexity of data k has here only a minimal effect in the running time of the optimization.

Furthermore, in 90% of the tests, the L_1 loss on the first axis is within 1% of the best L_1 norm out of ten restarts. The optimization algorithm is therefore quite stable, and in practical applications it may well be be sufficient to run the optimization algorithm only once. These results have been obtained with unoptimized and single-threaded R implementation on a laptop having 1.7 GHz Intel Core i7 processor.[3] The performance could probably be significantly boosted by, e.g., carefully optimizing the code and the implementation. Yet, even with this unoptimized code, response times are already of the order of 1 second to 1 minute.

[3]The R implementation used to produce Table 5.4 is available also via the demo page (footnote 1).

Iteration	$f_x(\mathbf{X}, \mathbf{X}^*)$	$f_y(\mathbf{X}, \mathbf{X}^*)$	$\hat{p}_x$	$\hat{p}_y$
1	0.127	0.093	0.01	0.01
2	0.084	0.078	0.01	0.01
3	0.080	0.044	0.01	0.01
4	0.028	0.026	0.17	0.14
5	0.000	0.000	1.00	1.00

Table 5.5: Test statistic and empirical p-value for both projections (x and y axes) in a test run of the synthetic data.

5.3.5 Stopping criterion

Finally, we tested whether the stopping criterion presented in Section 5.2.4 can indeed quantify whether the current projection is different from the structure level present due to random noise. We evaluated this in a controlled setting, i.e., using the synthetic data described in Section 5.3.2, which consists of 1000 ten-dimensional data vectors of which dimensions 1–4 can be clustered into five clusters, dimensions 5–6 into four clusters *involving different subsets of data points*, and of which dimensions 7–10 are Gaussian noise.

Since the data essentially contains cluster structure at three levels (in dimensions 1–6) and noise (dimensions 7–10 are purely random, 1–6 also contain some noise), we expect that in the fourth iteration the background model does not yet contain all the exact values of the data, but it contains the cluster structure, assuming the user has properly marked that. Then, because the constraints contain all real structure, the projection is based purely on random differences between the real data and the randomized data.

In experiments, we find that not in every run the results are the same, due to the nondeterministic randomization and optimization procedures. For example, it is not rare that the background model already contains the exact values of all data points after three iterations. However, if the run goes indeed as described above, where the first three iterations show the various clusterings in the data, then the empirical p-values align perfectly with our expectation: the p-values should be high after three iterations, and equal to one after four iterations. In the other cases, the p-values are equal to one already after three iterations.

The test statistic of the projections and the empirical p-value for five iterations in one test run are given in Table 5.5. We observe that in the first three iterations, $\hat{p} \leq 0.01$ for both axes. As expected, in the fourth iteration (shown in Figure 5.7) the projections do not correspond to substantial structure anymore, and $\hat{p} > 0.05$ for both axes. In the fifth iteration, the data is completely fixed and hence we find $\hat{p} = 1$.

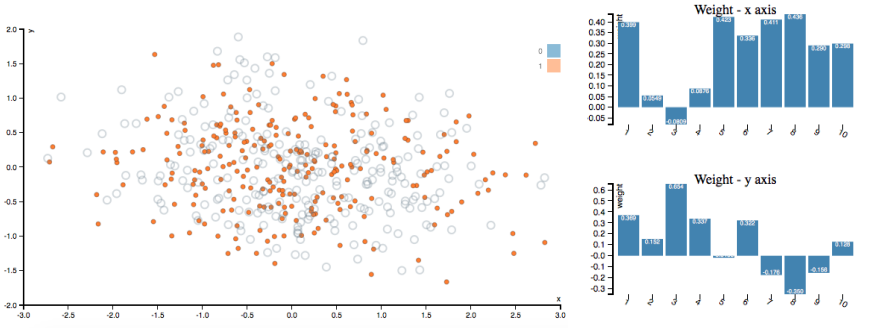

Figure 5.7: Projection of the synthetic data, fourth iteration in the empirical p-value test run. The empirical p-values for the axes are 0.17 and 0.14, indicating the amount of structure shown is comparable to what is expected in random noise. Notice also that the distribution of the randomized data is very similar to that of the real data and that the projection vectors are not similarly sparse as in the previous iterations (Figures 5.4 and 5.5), both signalling that the background model captures all meaningful structure present in the data.

5.4 Related work

Visualization pipeline. The pipeline of visualizing high-dimensional data is recognized to have three stages [26]:

Data transformation is the act of changing the data into a desired representation. In this stage methods such as dimensionality reduction (DR), clustering, and feature extraction are used. As we aim to find informative projections in lower dimension, we focus on the discussion of DR methods. Dimensionality reduction for exploratory data analysis has been studied for decades. Early research into visual exploration of data led to approaches such as multidimensional scaling [23, 40] and projection pursuit [11, 18]. Most recent research on this topic (also referred to as manifold learning) is still inspired by the aim of multi-dimensional scaling; find a low-dimensional embedding of points such that their distances in the high-dimensional space are well represented. In contrast to Principal Component Analysis [31], one usually does not treat all distances equal. Rather, the idea is to preserve small distances well, while large distances are irrelevant, as long as they remain large; examples are Local Linear and (t-)Stochastic Neighbor Embedding [16, 35, 41]. Even that is typically not possible to achieve perfectly, and a trade-off between precision and recall arises [44]. Recent works are mostly spectral methods along this line.

Visual mapping aims to encode the information in data space (the outcome of data transformation) into visual representations. For different types of the input data, the applicable encoding varies [26, 21]. Our approach takes multivariate real-valued data as input and visualizes the 2D projections of the data using scatterplots.

While simple 2D scatter plots allow to track the information learned by user, it would be possible to simultaneously visualize multiple pairwise relationships. For example, Scatterplot Matrix (SPLOM) [10] and Parallel Coordinate Plot (PCP) [38] show pairwise relationships between multiple data data attributes at once. Based on radial coordinates, visual encodings such as Star Coordinate Plot [19] and Radviz [17] are also used for simultaneous multivariate data visualization.

View transformation renders the visual encodings on the screen. Visualization of large number of data points usually has limitations such as high computational cost, visual cluttering (hence occlusions). To address these issues, continuous scatterplots [1] and continuous PCPs [15] as well as splatting scatterplots [28] and splatting PCPs [49] have been introduced. Such techniques are not yet used in proof-of-concept tool SIDE but may be useful if users need to anaylze datasets with very many data points.

User Interaction. Orthogonal to the data visualization pipeline, data visualization methods and systems can also be categorized by the amount of user interaction involved. We adopt the categorization proposed by Liu et al. [26]:

Computation-centric approaches have minimum interactivity, where a user only needs to set the initial parameters. The previously introduced dimensionality reduction methods all belong to this category.

Interactive exploration approaches fix data transformation models but allow users to explore the models with interactive visual mappings, e.g., navigate, query, and filter. For example, SAMP-Viz [48] and the work by Liu et al. [25] first compute a few data representatives using clustering methods. A user can navigate through these representatives and study the corresponding visualizations. Voyager [46] takes user selected data attributes as input and recommends either the visualizations that contains the selected attributes or representative visualizations that reveal the relationships between other attributes. Although the described recommendation mechanism is rather naive (visualizations are ordered by the types and names of the corresponding attributes). For each visualization, the authors propose a rule of thumb for choosing the visual encodings based on cognitive considerations. SeeDB [43] takes a user-specified database query and a reference query as input. For both queries, SeeDB evaluates all possible aggregate views that defined by a triplet: a group-by attribute, a measure attribute, and an aggregation function. Based on the deviation between the aggregative views of user-specified query and the corresponding one of the reference query, SeeDB visualizes the top k views that have largest deviation in bar charts.

Model manipulation techniques maintain a model that reflects a user's interaction in order to provide the user new insights. The existing methods (e.g., [12, 3, 9]) usually assume the user have a specific hypothesis in mind. Through interactions, these methods aim to help the user efficiently confirm or reject the hypothesis. On the other hand, we model user's belief about the data, and update the

model after a user has studied a new visualization. Our approach exposes as much new information as possible to the user, thus increasing the user's serendipity of gaining new insights about the data.

In order to reflect a user's interaction in the model, it is important to acknowledge the cognitive aspect of how humans identify [37, 45, 47] and assimilate [5] visual patterns. As our first attempt, SIDE assumes a user can visually identify the clusters in 2D scatterplots and internalize the position of the points in the clusters. One important line of future work is to investigate alternative assumptions about what a human operator can learn from a scatterplot.

Iterative data mining and machine learning. There are two general frameworks for iterative data mining: FORSIED [6, 7] is based on modeling the belief state of the user as an evolving probability distribution in order to formalize subjective interestingness of patterns. This distribution is chosen as the Maximum Entropy distribution subject to the user beliefs as constraints, at that moment in time. Given a pattern syntax, one then aims to find the pattern that provides the most information, quantified as the 'subjective information content' of the pattern.

The other framework, which we here named CORAND [14, 24], is similar, but the evolving distribution does not necessarily have an explicit form. Instead, it relies on sampling, or put differently, on randomization of the data, given the user beliefs as constraints. Both these frameworks are *general* in the sense that it has been shown they can be applied in various data mining settings; local pattern mining, clustering, dimensionality reduction, etc.

The main difference is that in FORSIED, the background model is expressed analytically, while in CORAND it is defined implicitly. This leads to differences in how they are deployed and when they are effective. From a research and development perspective, randomization schemes are easier to propose, or at least they require little mathematical skills. Explicit models have the advantage that they often enable faster search of the best pattern, and the models may be more transparent. Also, randomization schemes are computationally demanding when many randomizations are required. Yet, in cases like the current chapter, a single randomization suffices, and the approach scales very well. For both frameworks, it is ultimately the pattern syntax that determines their relative tractability.

Besides FORSIED and CORAND, many special-purpose methods have been developed for active learning, a form of iterative mining or learning, in diverse settings: classification, ranking, and more, as well as explicit models for user preferences. However, since these approaches are not targeted at data exploration, we do not review them here. Finally, several special-purpose methods have been developed for visual iterative data exploration in specific contexts, for example for itemset mining and subgroup discovery [2, 8, 42, 30], information retrieval [36], and network analysis [4].

Visually controllable data mining. This work was motivated by and can be con-

sidered an instance of *visually controllable data mining* [32], where the objective is to implement advanced data analysis method so that they are understandable and efficiently controllable by the user. Our proposed method satisfies the properties of a visually controllable data mining method (see [32], Section II B): (VC1) the data and model space are presented visually, (VC2) there are intuitive visual interactions that allow the user to modify the model space, and (VC3) the method is fast enough to allow for visual interaction.

5.5 Conclusions

In order to improve the efficiency and efficacy of data exploration, there is a growing need for generic and principled methods that integrate advanced visualization with data mining techniques to facilitate effective visual data analysis by human users. Our aim with this chapter was to present a principled framework based on constrained randomization to address this problem: the user is initially presented with an 'interesting' projection of the data and then employs data randomization with constraints to allow users to flexibly express their interests or beliefs. These constraints expressed by the user are then taken into account by a projection-finding algorithm to compute a new 'interesting' projection, a process that can be iterated until the user runs out of time or finds that constraints explain everything the user needs to know about the data. By continuously providing a user with information that contrasts with the constructed background model, we maximize the chance of the user to encounter striking and truly new information presented in the data.

In our example, the user can associate two types of constraints on a chosen subset of data points: the appearance of the points in the particular projection or the fact that the points can be nearby also in other projections. We also provided case examples on two datasets, one controlled experiment on synthetic data and another on real census data. We found that in these preliminary experiments the framework performs as expected; it manages to find interesting projections. Yet, interestingness can be case specific and relies on the definition of an appropriate interestingness measure, here the L_1 norm was employed. More research into this choice is warranted. Nonetheless, we think this approach is useful in constructing new tools and methods for interactive visually controllable data mining in variety of settings.

Also, a fundamental problem with linear projections is that they may not capture all types of structure in the data. It would be possible to work in a kernel space to overcome this or study non-linear manifold learning. However, the definition of clusters in the visualization does not readily map back to the original data space. Hence, it is not obvious then how to track the user's gained knowledge in a background model. Thus, this remains an open research question.

We have been actively working to put SIDE into practical use. One interesting application is a data analysis task called "*gating*". Gating is an analysis technique applied by biologists to flow cytometry data, where cells are data points and each point is described by a few intensity readings corresponding to emissions of different fluorescent dyes. The goal of gating is to extract clusters ('gates') based on cell's fluorescence intensities so that the cell types of a given sample can be differentiated. This is ongoing work.

SIDE is a prototype with several limitations. From a fundamental perspective, we assume a user can visually recognize the clusters in 2D scatterplots and internalize the position of the points in the clusters. This may misguide users if they give feedback and progress through a series of visualizations without making the effort to truly understand the defined clusters. They may not learn much, but more importantly because the intent is to provide new information continuously, there is almost no redundancy between the visualizations so information that is a combination of two or more previous visualizations is also never shown.

In further work we intend to investigate the use of the FORSIED framework to also formalize an analytical background model [6, 7], as well as its use for computing the most informative data projections. Additionally, alternative pattern syntaxes (constraints) will be investigated. Another future research direction is the integration of the constrained randomisation methods into software libraries in order to facilitate the integration of the methods in production level visualization systems.

References

[1] Sven Bachthaler and Daniel Weiskopf. Continuous scatterplots. *TVCG*, 14 (6):1428–1435, 2008.

[2] Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel. One click mining—interactive local pattern discovery through implicit preference and performance learning. In *Proc. of KDD IDEA*, pages 27–35, 2013.

[3] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. Dis-function: Learning distance functions interactively. In *VAST*, pages 83–92, 2012.

[4] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proc. of CHI*, pages 167–176, 2011.

[5] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *SIGCHI*, pages 443–452, 2012.

[6] Tijl De Bie. An information theoretic framework for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 564–572, New York, NY, USA, 2011. ACM.

[7] Tijl De Bie. Subjective interestingness in exploratory data mining. In *International Symposium on Intelligent Data Analysis*, pages 19–31, Berlin, Heidelberg, 2013. Springer.

[8] Vladimir Dzyuba and Matthijs van Leeuwen. Interactive discovery of interesting subgroup sets. In *Proc. of IDA*, pages 150–161, 2013.

[9] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, and Chris North. Observation-level interaction with statistical models for visual analytics. In *VAST*, pages 121–130, 2011.

[10] Mary Anne Fisherkeller, Jerome H Friedman, and John W Tukey. Prim-9, an interactive multidimensional data display and analysis system. *Dynamic Graphics for Statistics*, pages 91–109, 1988.

[11] Jerome H Friedman and John W Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(9):881–890, 1974.

[12] Michael Gleicher. Explainers: Expert explorations with crafted projections. *TVCG*, 19(12):2042–2051, 2013.

[13] Sami Hanhijärvi. *Multiple Hypothesis Testing in Data Mining*. PhD thesis, Aalto University School of Science, 2012.

[14] Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, and Heikki Mannila. Tell me something I don't know: Randomization strategies for iterative data mining. In *Proc. of KDD*, pages 379–388, 2009.

[15] Julian Heinrich and Daniel Weiskopf. Continuous parallel coordinates. *TVCG*, 15(6):1531–1538, 2009.

[16] Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *Proc. of NIPS*, pages 857–864, 2003.

[17] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. Dna visual and analytic data mining. In *InfoVis*, pages 437–441, 1997.

[18] Peter J Huber. Projection pursuit. *Ann. Stat.*, 13(2):435–475, 1985.

[19] Eser Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *InfoVis*, volume 650, page 22, 2000.

[20] Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. A tool for subjective and interactive visual data exploration. In *Proc. of ECML-PKDD - Part III*, pages 3–7, 2016.

[21] Johannes Kehrer and Helwig Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE TVCG*, 19(3):495–513, 2013.

[22] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering the Information Age: Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[23] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, 1964.

[24] Jefrey Lijffijt, Panagiotis Papapetrou, and Kai Puolamäki. A statistical significance testing approach to mining the most informative set of patterns. *Data Mining and Knowledge Discovery*, 28(1):238–263, 2014.

[25] Shusen Liu, Bei Wang, Jayaraman J Thiagarajan, P-T Bremer, and Valerio Pascucci. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. In *Computer Graphics Forum*, volume 34, pages 271–280, 2015.

[26] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE TVCG*, 23(3):1249–1268, 2017.

[27] Sébastien Loisel. Numeric javascript. `http://www.numericjs.com/`.

[28] Adrian Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *TVCG*, 19(9):1526–1538, 2013.

[29] Bernard V. North, Dave Curtis, and Pak C. Sham. A note on the calculation of empirical p-values from Monte Carlo procedures. *Am. J. Hum. Gen.*, 71 (2):439–441, 2002.

[30] Daniel Paurat, Roman Garnett, and Thomas Gärtner. Interactive exploration of larger pattern collections: A case study on a cocktail dataset. In *Proc. of KDD IDEA*, pages 98–106, 2014.

[31] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[32] Kai Puolamaki, Panagiotis Papapetrou, and Jefrey Lijffijt. Visually controllable data mining methods. In *IEEE International Conference on Data Mining Workshops*, pages 409–417. IEEE, 2010.

[33] Kai Puolamäki, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Interactive visual data exploration with subjective feedback. In *Proc. of ECML-PKDD - Part II*, pages 214–229, 2016.

[34] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

[35] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[36] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. Interactive intent modeling: Information discovery beyond search. *Communications of the ACM*, 58(1):86–92, 2015.

[37] Michael Sedlmair, Andrada Tatu, Tamara Munzner, and Melanie Tory. A taxonomy of visual cluster separation factors. In *Computer Graphics Forum*, volume 31, pages 1335–1344, 2012.

[38] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *JCST*, 28(5):852–867, 2013.

[39] J. Thomas and K. Cook. *Illuminating the Path: Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.

[40] Warren S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[41] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008.

[42] Matthijs van Leeuwen and Lara Cardinaels. Viper — visual pattern explorer. In *Proc. of ECML–PKDD*, pages 333–336, 2015.

[43] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. S ee db: efficient data-driven visualization recommendations to support visual analytics. *VLDB*, 8(13):2182–2193, 2015.

[44] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *JMLR*, 11(Feb):451–490, 2010.

[45] Hadley Wickham, Dianne Cook, Heike Hofmann, and Andreas Buja. Graphical inference for infovis. *TVCG*, 16(6):973–979, 2010.

[46] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG*, 22(1):649–658, 2016.

[47] Eugene Wu and Arnab Nandi. Towards perception-aware interactive data visualization systems. In *Data Syst. Interactive Anal. Workshop*, 2015.

[48] Huijie Zhang, Quanle Liu, Dezhan Qu, Yafang Hou, and Bin Chen. Sampviz: An interactive multivariable volume visualization framework based on subspace analysis and multidimensional projection. *IEEE Access*, 2017.

[49] Hong Zhou, Weiwei Cui, Huamin Qu, Yingcai Wu, Xiaoru Yuan, and Wei Zhuo. Splatting the lines in parallel coordinates. In *Computer Graphics Forum*, volume 28, pages 759–766, 2009.

6

Interpretable Representations

Subjectively Interesting Subgroup Discovery on Real-valued Targets

Abstract Deriving insights from high-dimensional data is one of the core problems in data mining. The difficulty mainly stems from the fact that there are exponentially many variable combinations to potentially consider, and there are infinitely many if we consider weighted combinations, even for linear combinations. Hence, an obvious question is whether we can automate the search for interesting patterns and visualizations. In this chapter, we consider the setting where a user wants to *learn* as efficiently as possible about real-valued attributes. For example, to understand the distribution of crime rates in different geographic areas in terms of other (numerical, ordinal and/or categorical) variables that describe the areas. We introduce a method to find subgroups in the data that are maximally informative (in the formal Information Theoretic sense) with respect to a single or set of real-valued target attributes. The subgroup descriptions are in terms of a succinct set of arbitrarily-typed other attributes. The approach is based on the Subjective Interestingness framework FORSIED to enable the use of prior knowledge when finding most informative non-redundant patterns, and hence the method also supports iterative data mining.

6.1 Introduction

We introduce the central ideas by means of an example. Consider the situation that a user want to learn about crime demographics, based on the UCI Communities and Crime data[1] [27]. This data contains violent crime rates for all ($n = 1994$) districts in the US and over 120 other attributes describing demographic statistics of those districts. One method to learn about the relation between the 'number of violent crimes' attribute and the demographic attributes is to extract *subgroup patterns*, which are sets of data points where violent crime is surprisingly high (or low) and that share similar statistics for one or several demographic attributes. A subgroup pattern should be interpreted as 'for data points that fall within the specified statistics that describe the subgroup, violent crime is surprisingly low/high'.

For example, the top subgroup pattern—identified through the method introduced in this chapter—states that there are high violent crime rates in districts where many mothers are unmarried at the moment they give birth to their child (condition $PctIlleg >= 0.39$; mean violent crime rate 0.53 in subgroup vs. 0.24 overall). An illustration of the data coverage for this pattern is given in Fig. 6.1. The subgroup covers 20.5% of the data and may be interesting because the distribution of crime rates within this subgroup deviates substantially from the full data. If a user would have no prior expectations about the data, this pattern is highly informative.

Indeed, we may quantify how informative/interesting it is, in the Information Theoretic sense: the number of bits of information we gain about the data by learning about this pattern, which depends on the amount of data covered (more is better) and how much the distribution in the subgroup differs from our expectation (more is better; in this chapter we consider mean and variance statistics). Typically, we would like to weight this against how complex the description of the pattern is (number of attributes used to describe the subgroup plus the number of statistics presented to the user, fewer is better), such that our aim is to provide a maximal *information rate*.

This is precisely the contribution of this chapter. We quantify the Information Content (IC; the amount of information gained) and Description Length (DL; the complexity of the description) for *subgroup patterns*. However, while the example above has only one *target attribute* (the violent crime rate), we also do this for multivariate real-valued targets, in order to enable users to learn about multivariate distributions. Besides, while the example above is about a surprisingly high mean (violent crime rate), we quantify the IC and DL for both mean and (co-)variance statistics.

As hinted at in the example, the IC of a pattern is inherently *subjective*, i.e., particular to a user, because *how much you learn depends on your prior knowl-*

[1] `http://archive.ics.uci.edu/ml/datasets/communities+and+crime`

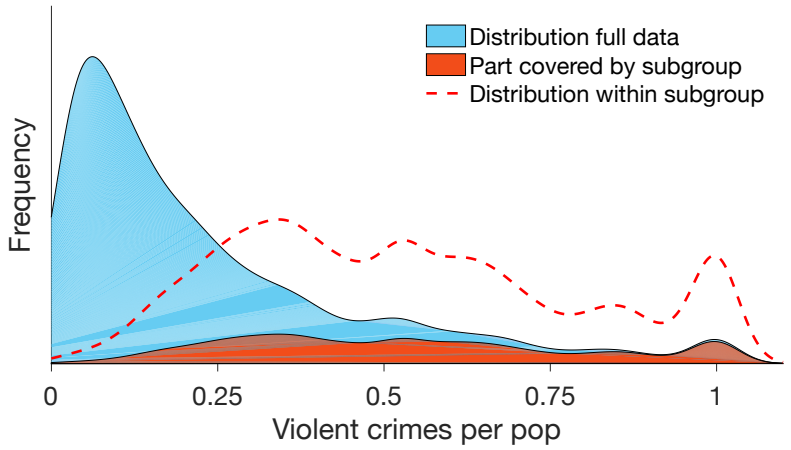

Figure 6.1: Distribution of violent crime over the full data (light blue area), part covered by the subgroup 'high rate of unmarried mothers' (red area), and distribution within the subgroup (red dotted line). Height of colored areas given by Gaussian-kernel smoothed estimates. The subgroup clearly covers a substantial amount of the data where the violent crime rate is relatively high.

edge. We implement this subjectivity by modeling a background distribution over the data space that is a Maximum Entropy distribution subject to constraints corresponding to the current knowledge of a user. This approach is known as FORSIED [6, 7] and also immediately enables iterative mining of non-redundant patterns without much additional effort.

We have implemented an algorithm to iteratively mine interesting patterns which is freely available as open source code. We have not studied the algorithmic problem in detail, but the implementation is based on beam search, a frequently employed approach in subgroup discovery. That is, it maintains a list of most interesting patterns of arity k, expands these to arity $k+1$ and selects the most interesting patterns again. Ultimately, it outputs the most interesting pattern found. It handles categorical, ordinal, and numerical *description attributes* (the demographic attributes in the example) and supports time constraints (e.g., stop after 1 minute of mining). The implementation is based on Cortana [23].

In summary, this chapter contributes the following:

1. We define a new pattern syntax for subgroups with a multivariate real-valued target distribution, called *location* and *spread* patterns. (Sec. 6.2.1)
2. We introduce a method to quantify their interestingness in a subjective manner. (Sec. 6.2.3)
3. Before that, we study how to incorporate prior knowledge into the background model, including previously identified patterns to enable iterative mining. (Sec. 6.2.2)

4. We present how to mine high-quality patterns using beam search and gradient descent. (Sec. 6.2.4)
5. We provide empirical evidence on four datasets that we can effectively find interesting patterns. (Sec. 6.3)

Discussion of related work is presented in Sec. 6.4, directions for future work and conclusions are given in Sec. 6.5. All code, including code for repeating the experiments, and links to the datasets are available at:
`http://www.interesting-patterns.net/forsied/sisd/`.

6.2 Methods

Overview. The high-level problem addressed in this chapter is:

Problem 1. Main Problem. *Iteratively inform the user about the mean and variance of subsets of data points that can be described concisely in terms of the description attributes, such that the rate of information gain of the user about the target attributes is maximized at each iteration.*

We first formalize the type of *pattern* shown to the user (Sec. 6.2.1). To explain how to find the most interesting patterns of this type (Sec. 6.2.4), we first need to formalize the background distribution (Sec. 6.2.2) and the interestingness of patterns (Sec. 6.2.3).

The formalization follows the FORSIED approach: we formalize the user's belief state about the target attributes by means of a background distribution, and quantify the IC of a pattern as the information (in its formal sense) the user gains about the target attributes by seeing the pattern. The Subjective Interestingness (SI) of a pattern is then formalized as the (subjective) IC divided by the DL of the pattern.

Notation. The data consists of a set of n pairs $(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)$, $i \in [n]$ (where $[n]$ is shorthand for $\{1, 2, \ldots, n\}$), called the *data points*. Here, the so-called *description attributes* of the ith data point $\hat{\mathbf{x}}_i \in \prod_{j=1:d_{\mathbf{x}}} \mathcal{X}_j$ is assumed to be a tuple of $d_{\mathbf{x}}$ attributes with domains $\mathcal{X}_j$, and $\hat{\mathbf{y}}_i \in \mathbb{R}^{d_{\mathbf{y}}}$ is a vector containing the values for $d_{\mathbf{y}}$ real-valued *target attributes*. We denote $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}'_1, \hat{\mathbf{y}}'_2, \cdots, \hat{\mathbf{y}}'_n)'$. In our setup the user is interested in gaining an understanding of the behavior of the target attributes in terms of the descriptions.

For example, the target attributes could contain healthcare-related attributes, whereas the description attributes could describe lifestyle choices (e.g., smoking or not, sedentary or active lifestyle, etc). Then, our method would yield insights into the healthcare target attributes, in terms of the lifestyle descriptions. In the example in Section 6.1, there is one target attribute (the violent crime rate) and over 120 description attributes.

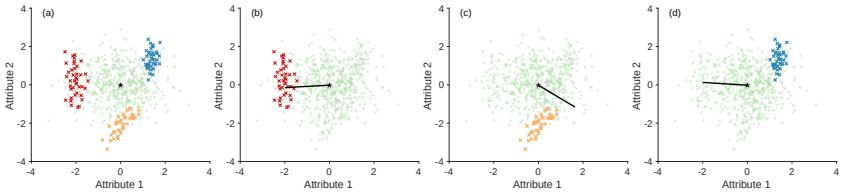

Figure 6.2: Patterns found in the synthetic data (§6.2.1,§6.3.1), (a) Data with the embedded patterns highlighted. (b)—(d) Top ranked pattern discovered in iterations 1—3. Light green circles are random data points, darker colored crosses the three embedded clusters. The black star represents the mean of the full data and the black lines are the angles of the most surprising variance direction. The two axes correspond to the only two target attributes.

We use hatted symbols to indicate these are empirical values. Non-hatted equivalents will be used to denote the respective random variables, e.g., $\mathbf{Y}$. They allow us to reason about the amount of uncertainty the user has about the data points. In general, standard face lower case symbols denote scalars, bold face lower case symbols denote tuples or vectors, upper case bold face symbols denote matrices, and upper case calligraphic letters denote sets.

6.2.1 Location and spread patterns

Subgroups, intentions, and extensions. We define patterns in terms of *subgroups*. A subgroup is defined by a set of *conditions* on the description attributes (the value combination is the subgroup *intention*) and by the set of data points for which the description attributes satisfy these conditions (the index set is the subgroup *extension*).

The intention is described in a pre-defined formal *description language*, such as in the form of a conjunction of conditions on individual metadata attributes. For $\mathcal{X}_j = \mathbb{R}$, such conditions are typically inequality conditions, and for $\mathcal{X}_j$ categorical they can be set in-/exclusion conditions. The extension is then specified by the index set $\mathcal{I} \subseteq [n]$ with $i \in \mathcal{I}$ iff $\hat{\mathbf{x}}_i$ satisfies the conditions.

Location and spread patterns. Subgroups tend to be informative if the target attribute values of data points in the extension $\{\hat{\mathbf{y}}_i | i \in \mathcal{I}\}$ are *unusual* in some sense. The way in which this set is unusual will be quantified by means of statistics—functions of this set of data points. For example, its empirical mean could be unusually far from what the user would expect, or its empirical variance around this mean could be unusually small or large along a certain direction.

To be precise, let us define two statistics $f_{\mathcal{I}} : \mathbb{R}^{n \times d_{\mathbf{y}}} \mapsto \mathbb{R}^{d_{\mathbf{y}}}$ and $g^{\mathbf{w}}_{\mathcal{I}}$:

$\mathbb{R}^{n \times d_\mathbf{y}} \mapsto \mathbb{R}$ as follows:

$$f_\mathcal{I}(\mathbf{Y}) = \sum\nolimits_{i \in \mathcal{I}} \mathbf{y}_i / |\mathcal{I}|, \text{ and} \tag{6.1}$$

$$g^\mathbf{w}_\mathcal{I}(\mathbf{Y}) = \sum\nolimits_{i \in \mathcal{I}} \left((\mathbf{y}_i - \hat{\mathbf{y}}_\mathcal{I})' \mathbf{w} \right)^2 / |\mathcal{I}|, \tag{6.2}$$

where $\hat{\mathbf{y}}_\mathcal{I} = \sum_{i \in \mathcal{I}} \hat{\mathbf{y}}_i / |\mathcal{I}|$ and $\mathbf{w} \in \mathbb{R}^k$ is a unit vector, i.e., $\mathbf{w}'\mathbf{w} = 1$. The first statistic (actually a set of $d_\mathbf{y}$ statistics), when evaluated on $\hat{\mathbf{Y}}$, quantifies the average vector of the data points in the extension (i.e., its average *location*), whereas the second quantifies the spread around that location. Patterns considered here are specified by an intention, which uniquely determines the extension $\mathcal{I} \subseteq [n]$, a unit vector $\mathbf{w}$, and the specification of the empirical values of one or both of the statistics $f_\mathcal{I}(\hat{\mathbf{Y}})$ and $g^\mathbf{w}_\mathcal{I}(\hat{\mathbf{Y}})$: we call it a *location pattern* when the former is specified, and a *spread pattern* when the latter is specified. We find that the spread of a subgroup cannot be interpreted straightforwardly without knowing its location, hence we only ever provide the user with spread patterns for subgroups for which the location pattern has been provided first. That is, we only explain the (co-)variance structure of subgroups for which the user already knows the precise mean value within the subgroup for all attributes.

Example. For the synthetic data shown in Fig 6.2a, a location pattern is an intention, e.g., 'Attribute3 = true', along with the mean of the subgroup, e.g., the dark red set of points. A spread pattern is an intention, a direction (a weight vector of unit length, as in Fig. 6.2b), and the magnitude of the variance in that direction.

6.2.2 Modelling the user's belief state

As we are interested in quantifying how informative a pattern is *to a particular user*, we quantify its informativeness (the IC) with respect to a model for the user's belief state. Patterns that contrast more strongly w.r.t. this belief state are more surprising and thus carry more information for the user. We model the user's belief state by the means of a so-called *background distribution*, represented by a density function p. This is a distribution over the possible data values (here, a distribution for $\mathbf{Y}$), which assigns a higher probability density to data values that are deemed more probable by the user. The general form of this approach is known as FORSIED [6, 7].

The initial background distribution, with density function p_0, can be estimated as the distribution of Maximum Entropy (MaxEnt) subject to constraints that express the user's knowledge, aka. the prior beliefs. The reason to use the MaxEnt distribution is that this is the only neutral choice, i.e., the only distribution that contains no other information [5]. Importantly, during the mining process the background distribution evolves, as each pattern shown to the user changes their belief state about the data. We first derive the initial background distribution, and then show how it can be updated to account for location and spread patterns.

Initial background distribution. To derive the initial background distribution, we need to assume what prior beliefs the user may have. We consider the case where the user expects the overall mean of $\hat{\mathbf{Y}}$ to be equal to a specified vector μ, and its covariance to be equal to a specified matrix $\mathbf{\Sigma}$. Notice that these need not be equal to the empirical statistics; they may be anything. The MaxEnt distribution subject to such expectations is well-known to equal a multivariate Normal distribution with μ and $\mathbf{\Sigma}$ as parameters:

$$p_0(\mathbf{Y}) \propto \exp\left(-\sum\nolimits_{i=1}^{n} (\mathbf{y}_i - \mu)' \, \mathbf{\Sigma}^{-1} \, (\mathbf{y}_i - \mu)/2\right). \tag{6.3}$$

The evolving background distribution. Given a pattern, the background distribution has to be updated to reflect the user's acquired knowledge. This can be done by minimally altering the background distribution while ensuring the statistic is (in expectation) as specified by the pattern. Here, *minimally* is naturally measured in terms of the Kullback-Leibler (KL) divergence. This approach is known as the *principle of minimum discrimination information*, a generalization of the MaxEnt principle.

We postulate, for now, that through subsequent updates in this way, the background distribution will continue to be a product of multivariate Normal distributions, although the means and covariances of the different data points may differ. I.e., after t iterations, the density function of the background distribution will be:

$$p_t(\mathbf{Y}) \propto \exp\left(-\sum\nolimits_{i=1}^{n} \left(\mathbf{y}_i - \mu_i^t\right)' \left(\mathbf{\Sigma}_i^t\right)^{-1} \left(\mathbf{y}_i - \mu_i^t\right)/2\right), \tag{6.4}$$

where data points may have differing means μ_i^t and covariance matrices $\mathbf{\Sigma}_i^t$. This holds for $t = 0$ (when $\mu_i^0 = \mu$ and $\mathbf{\Sigma}_i^0 = \mathbf{\Sigma}$ for all i), and the following shows that updating a distribution to account for location and spread patterns merely changes the parameter values, leaving the distribution's parametric form intact.

Background distribution updating for location patterns. To update p_t given a location pattern for a subgroup with extension $\mathcal{I}_{t+1}$, we must solve the following optimization problem:

$$p_{t+1} = \arg\min_q KL(q \mid\mid p_t) = E_q\left[\log\left(q(\mathbf{Y})/p_t(\mathbf{Y})\right)\right] \tag{6.5}$$

$$\text{subject to} \qquad E_q\left[f_{\mathcal{I}_{t+1}}(\mathbf{Y})\right] = \hat{\mathbf{y}}_{\mathcal{I}_{t+1}}, \tag{6.6}$$

with the additional technical constraint $E_q[1] = 1$ that guarantees that the distribution has a proper normalization.

Theorem 4. *Let p_t be a density function of the form of Eq. equation 6.4. Then, p_{t+1} has the same parametric form, with:*

$$\mu_i^{t+1} = \mu_i^t + \sum\nolimits_{i \in \mathcal{I}_{t+1}} (\hat{\mathbf{y}}_{\mathcal{I}_{t+1}} - \mu_i) / \left|\mathcal{I}_{t+1}\right|, \tag{6.7}$$

for $i \in \mathcal{I}_{t+1}$, and all other parameters unaltered.

Proof (outline only for brevity). Given the convexity of the KL-divergence and the linearity of the constraints, the optimization problem to be solved is convex and any stationary point is a global minimum. The Karush-Kuhn-Tucker (KKT) stationarity condition gives us the functional form of p_{t+1}:

$$p_{t+1} \propto p_t \exp\left(-\lambda' \sum\nolimits_{i \in \mathcal{I}_{t+1}} \mathbf{y}_i\right), \tag{6.8}$$

for a vector of KKT multipliers λ. Manipulating this expression shows that p_{t+1} is still of the form of Eq. (6.4), with $\mu_i^{t+1} = \mu_i^t + \mathbf{\Sigma}_i^t \lambda$ for $i \in \mathcal{I}_{t+1}$ and all other parameters unaltered. The optimal value of λ can be found by ensuring primal feasibility, yielding that $\lambda = \sum_{i \in \mathcal{I}_{t+1}} (\mathbf{\Sigma}_i^t)^{-1} (\hat{\mathbf{y}}_{\mathcal{I}_{t+1}} - \mu_i)$. Substituting this for λ in the expression for μ_i^{t+1} proves the theorem. □

Background distribution updating for spread patterns. To update the background distribution given a spread pattern for a subgroup with extension $\mathcal{I}_{t+1}$, we need to use the constraint

$$E_q\left[g^{\mathbf{w}}_{\mathcal{I}_{t+1}}(\mathbf{Y})\right] = \hat{v}^{\mathbf{w}}_{\mathcal{I}_{t+1}}, \tag{6.9}$$

in the KL-minimization problem, where for conciseness we denote the empirical variance as $\hat{v}^{\mathbf{w}}_{\mathcal{I}_{t+1}} \triangleq g^{\mathbf{w}}_{\mathcal{I}_{t+1}}(\hat{\mathbf{Y}})$.

Theorem 5. *Let p_t be a density function of the form of Eq. equation 6.4. Then, p_{t+1}, updated for a spread pattern with spread $\hat{v}^{\mathbf{w}}_{\mathcal{I}_{t+1}}$, has the same parametric form, with:*

$$\mu_i^{t+1} = \mu_i^t + \lambda \mathbf{w}'(\hat{\mathbf{y}}_{\mathcal{I}_{t+1}} - \mu_i^t)\mathbf{\Sigma}_i^t \mathbf{w} / \left(1 + \lambda \mathbf{w}' \mathbf{\Sigma}_i^t \mathbf{w}\right), \tag{6.10}$$

$$\mathbf{\Sigma}_i^{t+1} = \mathbf{\Sigma}_i^t - \lambda \mathbf{\Sigma}_i^t \mathbf{w}\mathbf{w}' \mathbf{\Sigma}_i^t / \left(1 + \lambda \mathbf{w}' \mathbf{\Sigma}_i^t \mathbf{w}\right), \tag{6.11}$$

for $i \in \mathcal{I}_{t+1}$, and all other parameters unaltered. The optimal value for λ is found as the (unique) zero of the following equation:

$$\sum_{i \in \mathcal{I}_{t+1}} \frac{\mathbf{w}' \mathbf{\Sigma}_i^t \mathbf{w}}{1 + \lambda \mathbf{w}' \mathbf{\Sigma}_i^t \mathbf{w}} + \sum_{i \in \mathcal{I}_{t+1}} \left(\frac{\mathbf{w}'(\hat{\mathbf{y}} - \mu_i^t)}{1 + \lambda \mathbf{w}' \mathbf{\Sigma}_i^t \mathbf{w}}\right)^2 = |\mathcal{I}_{t+1}| \hat{v}^{\mathbf{w}}_{\mathcal{I}_{t+1}}. \tag{6.12}$$

The proof is omitted for brevity. It is more tedious but analogous to the previous one.

Accounting for a set of location and spread patterns. If we want to take into account a set of location and spread patterns, the KL-divergence minimization problem needs to be solved with a constraint for each of these patterns. The problem remains convex, however, such that a coordinate-descent approach converges to

the global optimum. This means iteratively updating the background distribution for each of the patterns, until convergence. As long as the extensions of the different patterns have limited overlaps, as is the case in our experiments, convergence occurs very rapidly.

Implementation details. Rather than updating the parameters μ_i and $\mathbf{\Sigma}_i$, we actually update the *natural parameters* $\mathbf{\Sigma}_i^{-1}\mu_i$ and $-\frac{1}{2}\mathbf{\Sigma}_i^{-1}$ of these multivariate Normal distributions. This is numerically and computationally advantageous, but we feel it provides more insight to discuss the updates to μ_i and $\mathbf{\Sigma}_i$ above.

Also note that, maintaining and updating the background distribution may be costly if implemented naively. Each μ_i^t and $\mathbf{\Sigma}_i^t$ needs to be remembered and updating them involve summations over $\mathcal{I}_{t+1}$ terms. Yet, the number of *distinct* μ_i^t and $\mathbf{\Sigma}_i^t$ remains limited.[2]

6.2.3 Subjective Interestingness

Given a background distribution, [6] proposed that the Subjective Interestingness (SI) of a pattern can be computed as a ratio of two quantities: (a) the *Information Content* (IC) of a pattern, which is the negative log probability of the pattern under the background distribution; and (b) the *Description Length* (DL), which measures the effort a user has to make to understand and internalize the pattern.

To describe location patterns, we have to inform the user about the number of conditions in the pattern's intention, the conditions themselves, and the mean values for all attributes (to sufficient accuracy). For spread patterns, instead of the means, the vector $\mathbf{w}$ needs to be described, with its magnitude. All these parts of the code have constant length, except for the set of conditions, which has a length proportional to the number of conditions $|\mathcal{C}|$. Thus:

$$\text{DescriptionLength} = \gamma|\mathcal{C}| + \eta\ (+1),$$

where the $+1$ applies to spread patterns only because they have one more term then location patterns.

We discuss determining γ and η in Remark 8 below. Note that it does not matter whether the DL is reflective of reality *in absolute terms*, because the actual SI scores are irrelevant. What matters is the ranking, hence it is desirable that γ is chosen well relative to η.

As the IC (thus the SI) depends on the pattern type, we derive it first for location patterns and subsequently for spread patterns.

SI for location patterns. As the background distribution equation 6.4 for the target values $\mathbf{Y}$ of a data record is a normal distribution, the marginal distribution $p_{f_\mathcal{I}}$of the mean $f_\mathcal{I}(\mathbf{Y})$ of a subgroup $\mathcal{I}$ is again a normal distribution, with mean

[2]Indeed, $\mu_i^t = \mu_j^t$ and $\mathbf{\Sigma}_i^t = \mathbf{\Sigma}_j^t$ for all i and j such $i, j \in \mathcal{I}_s$ or $i, j \notin \mathcal{I}_s$ for all $s \in [t]$, since they will have been subjected to the same updates.

$\mu_{\mathcal{I}} = \sum_{i\in\mathcal{I}} \mu_i/|\mathcal{I}|$, and covariance $\mathbf{\Sigma}_{\mathcal{I}} = \sum_{i\in\mathcal{I}} \mathbf{\Sigma}_i/|\mathcal{I}|$. The IC of a location pattern with extension $\mathcal{I}$ is thus the negative log probability of the pattern. Written in full:

$$\begin{aligned} IC_f(\mathcal{I}) = &- \log p_{f_{\mathcal{I}}}\left(f_{\mathcal{I}}(\mathbf{Y})\right) = \log\left((2\pi)^{d_{\mathbf{Y}}}|\mathbf{\Sigma}_{\mathcal{I}}|\right)/2 \\ &+ (f_{\mathcal{I}}(\hat{\mathbf{Y}}) - \mu_{\mathcal{I}})'\mathbf{\Sigma}_{\mathcal{I}}^{-1}(f_{\mathcal{I}}(\hat{\mathbf{Y}}) - \mu_{\mathcal{I}})/2. \end{aligned} \tag{6.13}$$

The SI of a location pattern with extension $\mathcal{I}$ and statistic $f_{\mathcal{I}}$ reads:

$$SI_f(\mathcal{I}) = IC_f(\mathcal{I})/\left(\gamma|\mathcal{C}_{\mathcal{I}}| + \eta\right). \tag{6.14}$$

SI for spread patterns. While the SI of a location pattern can be computed analytically, evaluating the SI for a spread pattern is more complex. However, it can be approximated well.

If the patterns assimilated into the background so far do not overlap (i.e., non-intersecting extensions)[3], then after updating the background distribution with location information of the pattern, the parameter μ_i of the background model equals the observed mean of subgroup $\hat{\mathbf{y}}_{\mathcal{I}}$. So we can derive:

$$(\mathbf{y}_i - \hat{\mathbf{y}}_{\mathcal{I}})'\mathbf{w}(\mathbf{w}'\mathbf{\Sigma}_i\mathbf{w})^{-1/2} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ (normal distr.)}, \tag{6.15}$$

$$((\mathbf{y}_i - \hat{\mathbf{y}}_{\mathcal{I}})'\mathbf{w})^2/\left(\mathbf{w}'\mathbf{\Sigma}_i\mathbf{w}\right) \sim \chi^2_1 \text{ (Chi-squared, 1 d.f.)}. \tag{6.16}$$

Denote the chi-squared random variable by $\mathbf{c}_{i,1} = ((\mathbf{y}_i - \hat{\mathbf{y}}_{\mathcal{I}})'\mathbf{w})^2/(\mathbf{w}'\mathbf{\Sigma}_i\mathbf{w})$. Then, the variance statistic equation 6.2 is a linear combination of chi-squared random variables:

$$g^{\mathbf{w}}_{\mathcal{I}}(\mathbf{Y}) = \sum\nolimits_{i\in\mathcal{I}} \mathbf{w}'\mathbf{\Sigma}_i\mathbf{w} \cdot \mathbf{c}_{i,1}/|\mathcal{I}|. \tag{6.17}$$

The probability density function of a linear combination of chi-squared distributed random variables has been studied extensively, but a closed form analytic solution is unknown. Here we choose the state-of-art approximation proposed by [31]: Writing a_i for the coefficient $\mathbf{w}'\mathbf{\Sigma}_i\mathbf{w}/|\mathcal{I}|$, they prove that the distribution of $g^{\mathbf{w}}_{\mathcal{I}}(\mathbf{Y})$ can be accurately approximated by an affine function of a chi-squared random variable $\mathbf{c}_m$ with m degrees of freedom:

$$g^{\mathbf{w}}_{\mathcal{I}}(\mathbf{Y}) = \alpha\mathbf{c}_m + \beta, \text{ where } \alpha = \frac{\sum_{i\in\mathcal{I}_s} a_i^3}{\sum_{i\in\mathcal{I}_s} a_i^2}, \tag{6.18}$$

$$\beta = \sum_{i\in\mathcal{I}_s} a_i - \frac{\left(\sum_{i\in\mathcal{I}_s} a_i^2\right)^2}{\sum_{i\in\mathcal{I}_s} a_i^3}, \quad m = \frac{\left(\sum_{i\in\mathcal{I}_s} a_i^2\right)^3}{\left(\sum_{i\in\mathcal{I}_s} a_i^3\right)^2}.$$

[3] If the patterns used to update the background distribution do overlap, then $\mu_i \neq \hat{\mathbf{y}}_{\mathcal{I}}$ even after the update. So the random variable in Eq. equation 6.16 follows a non-central chi-squared distribution, hence the linear combination Eq. equation 6.17 also changes. In this case, we approximate the SI with the same computation for the non-overlapping situation.

Therefore the approximated probability density function reads:

$$p_{g_{\mathcal{I}}^{\mathbf{w}}}(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})) \approx \frac{\left(\left(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})-\beta\right)/\alpha\right)^{\frac{m}{2}-1} e^{-\frac{g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})-\beta}{2\alpha}}}{\left(\alpha \cdot 2^{\frac{m}{2}}\Gamma(m/2)\right)}.$$

Thus the IC for a spread pattern with extension $\mathcal{I}$ is given as:

$$\begin{aligned} \mathrm{IC}_g^{\mathbf{w}}(\mathcal{I}) = &-\log p_{g_{\mathcal{I}}^{\mathbf{w}}}\left(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})\right) \approx \log\left(2^{\frac{m}{2}}\Gamma(m/2)\right) \\ &+\alpha-(m/2-1)\log\left(\left(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})-\beta\right)/\alpha\right) \\ &+\left(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})-\beta\right)/(2\alpha). \end{aligned} \tag{6.19}$$

The SI is then given by

$$SI_g^{\mathbf{w}}(\mathcal{I}) = \mathrm{IC}_g^{\mathbf{w}}(\mathcal{I})/\left(\gamma|\mathcal{C}_{\mathcal{I}}|+\eta+1\right). \tag{6.20}$$

Remark 8. *In practice, the SI's from Eqs. (6.14) and (6.20) are only used for ranking the patterns, or even just for finding the single most interesting pattern. The absolute value of the SI is largely irrelevant in practice. Thus, we can set* $\eta = 1$ *without losing generality, such that only* γ *remains as a parameter, the value of which essentially depends on the 'coding scheme' used to present the pattern to the user.*

We do know of any principled approach to choose γ *well. Notice that the problem here is not to do model selection in the statistical sense, but rather the DL should be determined based on aspects of human cognition. In this chapter, we set* $\gamma = 0.1$ *throughout all the experiments. However, tuning* γ *biases the results toward more or fewer conditions to describe the subgroup and hence tuning could be useful.*

6.2.4 Search strategies

Overall approach. We have not studied the complexity formally, but the optimization problem for either pattern type appears to be very difficult. Tiling [12], a similar and easier-appearing problem, is already NP-hard. The score function here (the SI) is also not monotonic and, if the cardinality of metadata attributes is large, pattern enumeration, which then equals exhaustive search, is not a feasible strategy. For spread patterns, the search problem is essentially a dimensionality reduction problem. From empirical results, we learn that the search problem can have many local optima. Besides, there is no structure in the problem that struck us as easy to use.

Hence, we resort to optimization procedures that are commonly used in either scenario. In brief, to find location patterns that maximize Eq. equation 6.14, we employ beam search. For spread patterns, we first search for the best location pattern and after updating the background distribution with the location, we use

gradient descent to find the weight vector $\mathbf{w}$ that maximizes Eq. equation 6.20 for that subgroup. The procedures are outlined in more detail below.

Location pattern. Beam search systematically explores the conjunctions of conditions by expanding a limited set of conjunctions that have the largest SI so far. It evaluates conjunctions of conditions on metadata attributes in a level-wise manner. On each level, a limited list (beam width) of most promising combinations is maintained. On the next level, the algorithm exhaustively grows the combinations from the limited pattern list and maintains again the best. The mining process stops when all possible conjunctions of conditions are explored or a chosen stopping criterion is met, either a maximum search depth or time spent. Then, the best pattern found throughout the search is given as output.

Spread pattern. Finding the best spread pattern consists of two steps: (1) find the best location pattern and update the background distribution with that information, (2) for that location pattern find the most interesting direction in the target space. We have already described the first step; the second step can be formularized in terms of the following optimization problem:

$$\max_{\mathbf{w}} \; SI(g_{\mathcal{I}}^{\mathbf{w}}(\mathbf{Y})) \qquad \text{s.t. } \mathbf{w}'\mathbf{w} = 1. \tag{6.21}$$

Since the description length in SI is fixed for a specific extension $\mathcal{I}$, the problem (6.21) maximizes the entropy of a χ^2 distribution equation 6.19 over the unit sphere. To optimize $\mathbf{w}$, we apply the off-the-shelf manifold optimization tool Manopt [4] with the unit sphere as the manifold, and solve it with the gradient-based solver.[4]

6.3 Experiments

In this section we evaluate whether our method is able to find good location and spread patterns in terms of SI and whether the model updates work as expected. We also studied the pattern descriptions, to see whether the patterns found appear to be interesting. We conducted experiments on four datasets: one synthetic and three publicly available ones, of widely varying nature. The results for each dataset are described in the following subsections. The final subsection considers the scalability of the methods.

We used the beam search available within the data mining tool Cortana [23], using the following settings: descriptions on numerical metadata are based on $\geq$ and $\leq$ relations with four split points (1/5–4/5 percentiles). The beam width is set to 40 and the search depth is four conditions. The search logs the best 150 subgroups, with a maximum run time of 5 minutes.

[4] We computed the gradient analytically, but details are omitted due to lack of space.

Intention	SI Iter1	Iter 2	Iter 3	Iter 4
a3 = '1'	48.35	-1.13	-1.13	-1.13
a5 = '1'	47.49	47.49	-1.13	-1.13
a4 = '1'	39.49	39.49	39.49	-1.13
a4 = '0' $\wedge$ a3 = '1'	36.26	-0.85	-0.85	-0.85
a5 = '0' $\wedge$ a3 = '1'	36.26	-0.85	-0.85	-0.85
a3 = '0' $\wedge$ a5 = '1'	35.62	35.62	-0.85	-0.85
a4 = '0' $\wedge$ a5 = '1'	35.62	35.62	-0.85	-0.85
a3 = '0' $\wedge$ a4 = '1'	29.62	29.62	29.62	-0.85
a5 = '0' $\wedge$ a4 = '1'	29.62	29.62	29.62	-0.85
a5 = '0' $\wedge$ a4 = '0' $\wedge$ a3 = '1'	29.01	29.01	-0.68	-0.68

Table 6.1: Change in SI for the top patterns over four iterations (§6.3.1). All patterns have size 40.

6.3.1 Synthetic data

Data. We generated a dataset of 620 data points with two real-valued target attributes (attributes 1 and 2) and five binary descriptive attributes. We first sample 500 target values from the 2-D multivariate normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and then embed three subgroups each consisting of 40 points into the data, see Fig. 6.2a. Each subgroup has distance 2 from the mean but a different covariance structure: the variance along the main eigenvector is much larger than the other. The first three descriptive attributes (attributes 3–5) contain the true labels for subgroups p_1 to p_3; the other two (attributes 6 and 7) take values randomly sampled from a Bernoulli distribution with $p = 0.5$.

Setup. We set the mean and covariance of the background model equal to the empirical values of the full data. First, we tested whether our method could retrieve the embedded patterns. We performed the two-step spread pattern mining process for three iterations, and at each iteration we selected the top pattern to update the background distribution. Second, we corrupted the descriptive attributes by randomly flipping every 0 and 1 with a certain probability. Then, we checked up to what noise level the subgroups can still be retrieved.

Results. Figures 6.2b—6.2d show the top patterns in the first three iterations. Our method correctly found the embedded subgroups in the first three iterations by their displaced location from the expected center. It also retrieved the direction along which each subgroup's spread differs most from the full data covariance. Of course this is not so surprising, because for each embedded subgroup there is a description attribute setting the subgroup apart from the rest of the data.

To study the mining process in more detail, Table 6.1 shows the change in SI for the top 10 patterns from the first iteration in subsequent iterations. We observe

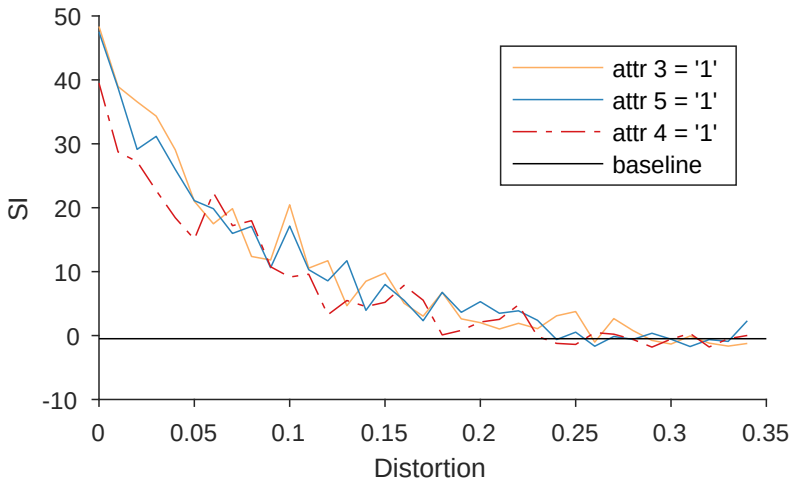

Figure 6.3: SI of subgroups in the synthetic data, (§6.3.1), corresponding to true descriptions when adding and removing points randomly to the subgroups.

that the three embedded subgroups were the highest-ranking patterns in the first three iterations (indeed they were the top 3 immediately because the subgroups induced by the true descriptions stand out so clearly from the rest of the data).

Once they were selected and used to update the background distribution, in the subsequent iterations the SI of the embedded subgroup patterns, and the SI of the derived patterns, dropped and remained low afterwards. Hence, updating the background distribution and the influence that should have on the IC scores of patterns worked as expected.

It can be observed also that the subgroups with more complex descriptions (e.g., a4 = '0' $\wedge$ a3 = '1') have lower SI, even though the extensions are equivalent to the corresponding a_i = '1' pattern. This is because their DL is higher, while their extension is equivalent. Note that non-redundancy in the description is indeed achieved naturally in a principled manner. Also worth noting is that the SI can be negative. This is due to that the IC is based on a probability density and not a mass.

The result of the retrieval experiment with noise added to the description attributes is given in Fig. 6.3. We find that all embedded patterns can still be recovered when the flipping probability is up to 0.22, and partially retrieved up to 0.25. These values correspond to adding a random set of points that is roughly three and four times the size of the embedded pattern (e.g., $(1 - 0.25) \cdot 40 = 30$ vs. $0.25 \cdot 480 = 120$). We conclude that the method is quite robust against noise.

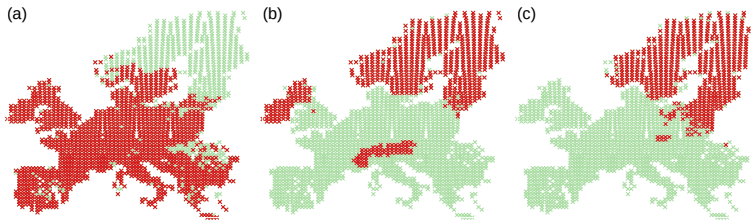

Figure 6.4: Explanation of what makes the first location pattern (Fig. 6.6a) interesting (also see Fig. 6.5). Presence maps of the first three species in the full data: (a) wood mouse, (b) mountain hare, and (c) moose.

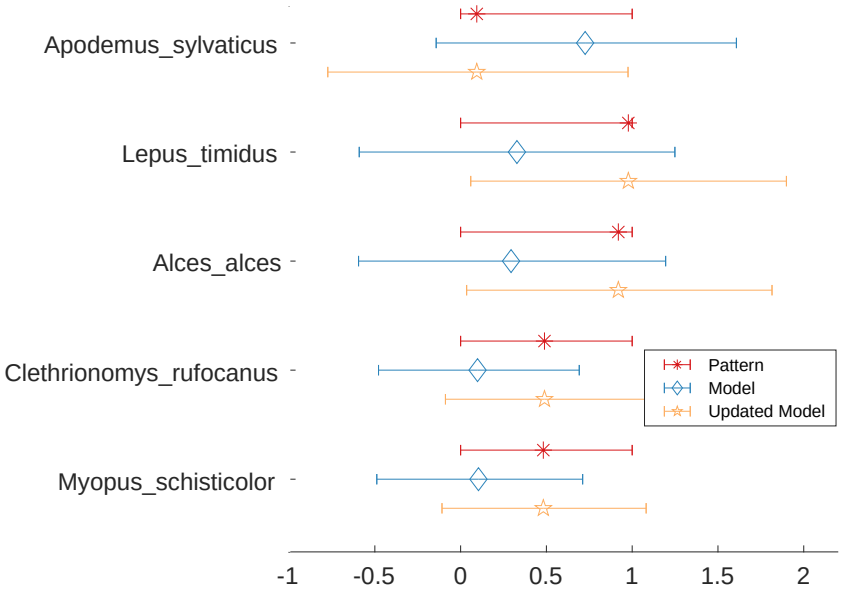

Figure 6.5: Explanation of what makes the first location pattern (Fig. 6.6a) interesting (also see Fig. 6.4). Observed and expected mean and 95% confidence interval of the most surprising species as ranked by SI.

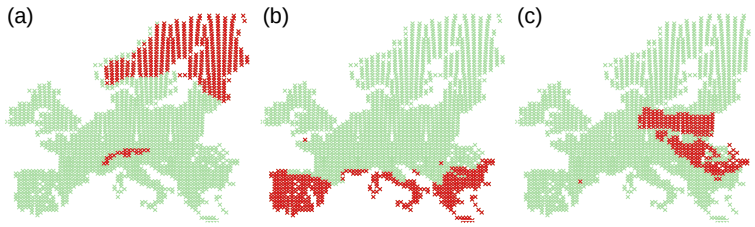

Figure 6.6: Extensions of the top three patterns found in the Mammal data (§6.3.2), (a) The first pattern covers northern Europe and part of the Alps. The intention is 'mean temperature in March $\leq -1.68\,^{\circ}C$*'. (b) The second pattern covers the very south of Europe. Its intention is 'average monthly rainfall in August* ≤ 47.62 *mm'. (c) The third pattern covers parts of eastern Europe. The intention is 'average monthly rainfall in October* ≤ 45.25 *mm* and *mean temperature of wettest quarter* $\geq 16.32\,^{\circ}C$*'.*

6.3.2 Mammal data

Data. The mammal data encompasses data from The Atlas of European Mammals and from WorldClim.org, as preprocessed by Heikinheimo et al. [13]. It contains records about the presence of species in 2220 cells located on a grid that covers Europe. Each record contains the geolocation, binary labels for the presence/absence of 124 mammals, as well as 67 climate condition indicators.

Setup. We used the presence/absence indicators as target attributes and climate indicators for descriptions. The location information was used only for visualization and interpretation. We again set the initial mean and covariance parameters of the background model equal to the empirical values.

We found that for binary target attributes, spread patterns are not truly interesting. This makes sense, because the variance of a Bernoulli random variable is uniquely determined by the mean. Hence, a spread pattern becomes a one dimensional location pattern. That the attributes are binary is another form of background knowledge that could in principle be incorporated into the method, but it would lead to different derivations and we did not study this. Instead, we studied only location patterns on this data.

Results. The geographic locations of the data points part of the subgroup for the top patterns found in the first three iterations are visualized in Fig. 6.6. The subrgoup intentions (combination of values that specifies the subrgoup) are given in the caption. The top pattern corresponds to locations that are relatively cold in late winter. In contrast, the second pattern covers locations that have an extremely dry summer, while the third pattern covers locations with a dry autumn and warm conditions in the months when most rain falls (which is the summer in that area).

We further investigated the distribution of the mammals within the subgroups.

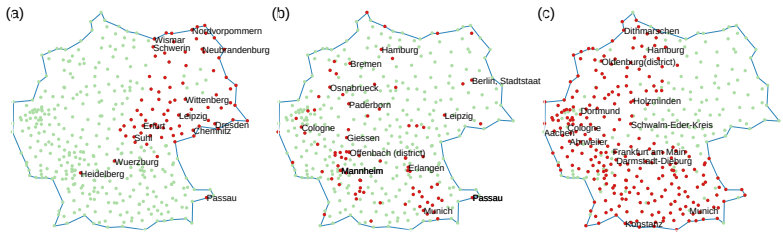

Figure 6.7: Geographic locations of data points covered by the top subgroup patterns found in the first three iterations on the Socio-economics data (§6.3.3): (a) "Children Pop. <= 14.1", (b) "Middle-aged Pop. >= 26.9", (c) "Children Pop. >= 16.4". The contents of these patterns is roughly as follows: (a) Low numbers of children are present in Eastern Germany, as well as in three cities with a very high percentage of students (Heidelberg, Passau, Wuerzburg). Here the Left party is popular is popular at the expense of all other parties (Fig. 6.8a). (b) These are larger cities with relatively many jobs. Here the Green party is more popular at the expense of Left. (c) This subgroup is almost the complement of (a), but not quite (e.g., Saarland and smaller cities in the Ruhr area are not covered). Here Left is impopular and all others are more popular than the country-wide averages.

Fig. 6.5 shows the mean values for the first pattern, and the mean and 95% confidence interval for the background model for the top five mammal species ranked by SI. Figures 6.4a–c show the actual occurrences of the top three species across Europe. The species ranked first is the wood mouse, which is wide-spread in the middle and southern Europe but not in the northern areas. The second species is the mountain hare, whose habitat mostly coincides with the area associated to the found location pattern. This indicates it thrives under harsh temperature condition. The third species, moose, is also wide-spread mostly in the same area.

By contrasting these ground-truth location maps for the species (Figs. 6.4a–c) against the subgroup location map (Fig. 6.6), we find that indeed this pattern could be highly informative. However, while the description is concise, the displacement in the target space does not appear to be sparse (it covers many species). To comprehend the pattern in full, one should look at all the attributes where the mean deviates from the expectation, not just at the top five. This means fully understanding the pattern is somewhat difficult.

Finally, notice that these three species correlate and the background model already accounts for that. Hence, the IC of the subgroup is much less than the sum over the three attributes if they would be considered individually. Nonetheless, the IC is very high.

Although not shown, we repeated this exercise for the second and third pattern. The subgroup patterns appear to be informative. For example, ranked by SI, the most surprising species for the second pattern are the absence of the stoat and the bank vole, who prefer a moist environment, and the presence of the Iberian

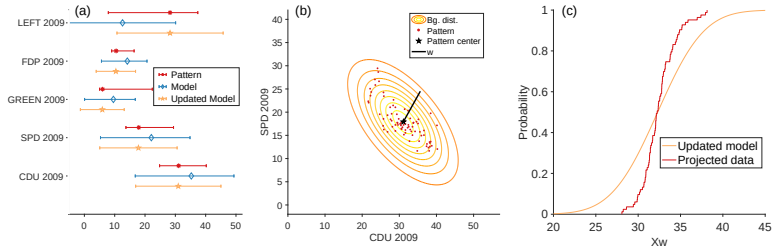

Figure 6.8: Spread pattern corresponding to the top pattern in Socio-economics data (§6.3.3, Fig. 6.7a). (a) Expected vs. observed distribution of the subgroup. the y-axis is ranked by SI, from top to bottom. (b) Expected vs. observed distribution for the pair of attributes with highest SI, after updating the background model with the location pattern. The contour plot shows the found weight vector (black line) along which the spread of the subgroup (red dots) has largest difference from the background model (contour lines). (c) The marginal CDF of background distribution and subgroup along $\mathbf{w}$ *after updating location.*

hare, who indeed lives exclusively in the area of the pattern. Thus, our method appears to find geographically meaningful location patterns that reveal the relationship between climate conditions and sets of animals that are absent/present in the corresponding area.

6.3.3 Socio-economics data case study

Data. The German socio-economic dataset [2] consists of socio-economic records of 412 administrative districts in Germany. The features are divided into three groups: election voting counts, age distribution, and workforce distribution. The voting percentages of the five largest political parties (CDU/CSU, SPD, FDP, Greens, and Left) in the 2009 German elections are also included. We added the geographic coordinates of each district center ourselves.

Setup. We used the vote count attributes as targets and the age and the work force attributes for the descriptions. Geolocations were used only for interpretation. Again, we set the initial mean and (co-)variance for the background distribution equal to the empirical values. In this case, that means we assume a user initially knows the overall voting behavior of the 2009 German elections.

We again performed three iterations of the subgroup discovery algorithm, but this time we studied both the location and the associated spread pattern in each iteration. To increase interpretability, we enforced a 2-sparsity constraint on $\mathbf{w}$, by optimizing it for each pair of target attributes separately and then selecting the result with the highest SI.

Results. Fig. 6.7 shows the top location patterns found, and Fig. 6.8 some explanation and the spread pattern for the top location pattern. Comparing the distribution

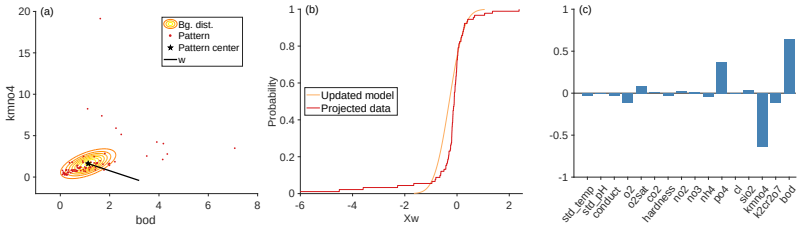

Figure 6.9: Top spread pattern found in the Water quality data (§6.3.4). (a) Subgroup vs. background distribution, along with the optimal projection vector $\mathbf{w}$*, projected on the two axes with highest weights. (b) CDF of subgroup and model along* $\mathbf{w}$*. (c) The weight vector* $\mathbf{w}$ *itself.*

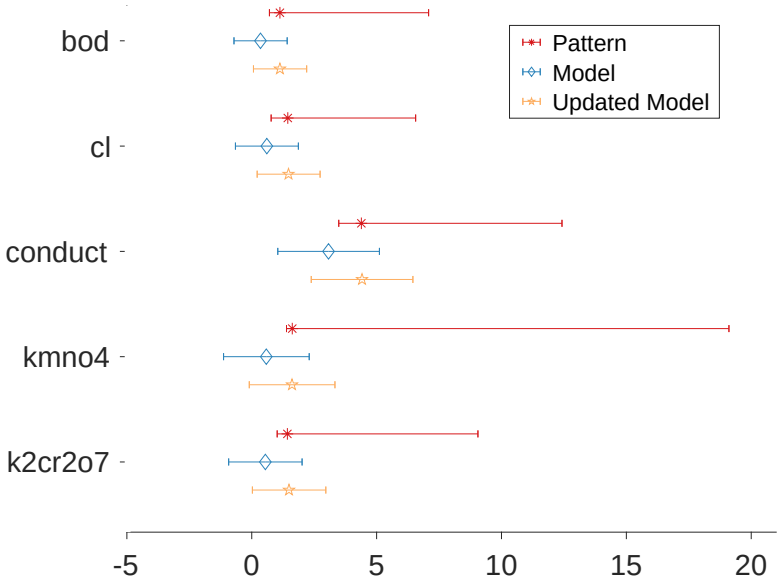

Figure 6.10: Observed and expected distribution of the top location pattern found in the Water quality data (§6.3.4), before and after updating location.

of the pattern against the expected distribution under the model (Fig. 6.8a, red and blue lines), we observe that the voting behavior in the corresponding districts deviates substantially from the full population: more votes for Left, fewer for all others. The intention of the pattern corresponds to districts with relatively few children; from the map we see the extension covers mainly East Germany.

Once we update the background distribution with the location pattern, the model mean of the pattern becomes the observed mean, see Fig. 6.8b. Given the updated background distribution, we find that the spread pattern with highest SI is related to the covariance between the social democrats (SPD) and Christian democrats (CDU), with weight vector $(0.5704, 0.8214)$ (see Fig. 6.8c).[5] As visualized in Fig. 6.8d, the variance in this direction is much smaller than expected. Of course since the votes add up to a constant, under the model we also expect negative correlations between the parties, but for this subgroup the anti-correlation is much stronger than expected. This indicates these parties really appear to battle for the same voters. However, we are not sufficiently knowledgeable of German politics to judge whether this is a solid observation.

Fig. 6.7 also shows the extensions for the top patterns in the second and third iterations. The second pattern has intention "Middle-aged Pop. $>=$ 26.9" and contains large cities. Within those districts, the Green party has relatively high vote counts, which comes at the expense of the Left party. The third pattern, "Children Pop. $>=$ 16.4", is mostly a complementary pattern to the first one (see Fig. 6.7a,c), except that many of the big cities (Munich, Berlin, Cologne, etc.) fall exactly between the two thresholds ($> 14.1, < 16.4$). The third pattern indeed covers locations where Left is unpopular and all other parties receive relatively many votes compared to the background model. In both the second and third location pattern, the corresponding spread pattern is a similar low-variance pattern as in Fig. 6.8. In our subjective opinion, these patterns appear to convey potentially highly interesting insights into this data.

6.3.4 Water quality data case study

Data. The River Water Quality dataset [10] consists of 1060 water quality records sampled from rivers in Slovenia. Each record contains measured values for 16 physical/chemical parameters and 14 bioindicators (7 plants, 7 animals), including a list of all taxa present and their density. The density of each taxon is recorded by an expert biologist at three different qualitative levels, where 1 means the taxon occurs incidentally, 3 frequently, and 5 abundantly.

[5] The 2d-contour plot of the subgroup is aggregated as the average pdf of the background model for each data point in the subgroup. The mean and covariance are the sub-vector and the sub-matrix that correspond to attributes indicated by the weight vector. This visualization is not fully accurate, as not all points have the same parameters. A single multivariate normal cannot represent the background model accurately.

Setup. We use the 16 physical/chemical parameters as targets and the 14 bioindicators as descriptors. Mean and (co-)variance of the initial background distribution were set to the empirical values.

Results. The top location pattern has intention "Amphipoda Gammarus fossarum $<= 0$ AND Oligochaeta Tubifex $>= 3$" and covers 91 records. Fig. 6.10a shows that the water samples fulfilling the description have an above-average biological oxygen demand (BOD), chlorine concentration (Cl), electrical conductivity, as well as $K_2Cr_2O_7$ and $KMnO_4$ (indicating chemical oxygen demand, COD).

In the second step our method finds, without enforcing it, a sparse weight vector placing high weights on BOD and $KMnO_4$ (Fig. 6.10d). The contour plot (Fig. 6.10b) indicates that along the most interesting spread direction, $\mathbf{w}$, the variance of the subgroup is much larger than expected. The CDF in Fig. 6.10c also confirms this. The main conclusion here is that, although the identified patterns are typically subgroups that are displaced from the center of the data, which is typically associated with having a smaller variance in comparison to the full data, it is also possible to find spread patterns corresponding to surprising higher-variance directions.

6.3.5 Scalability

We have not analyzed the algorithmic complexity of mining optimal location and spread patterns in detail, nor have we studied extensively how to find good solutions in practice. The computation time of the beam search algorithm can be controlled through the search parameters (number of solutions kept at each iteration, discretization strategy for numerical attributes, maximum number of conditions for the description) and it employs a timer. Of course it may not find the optimal pattern, but this strategy allows it to work on data of any size and dimensionality. Likewise, the heuristic solution to mine spread patterns typically outputs a pattern in very little time.

Notice that for both algorithms, the runtime is linear in the number of data points (i.e., to do the exact same computations on larger data is linear). One may add attributes without affected the computation time at the mining stage (background model discussed below), but of course to include them in candidate descriptions leads to an exponential growth in number of possible subgroup definitions. We feel it would be pointless to include a runtime experiment for these steps, as it is not feasible to compute the optimal solutions as a comparison, except on very small data.

What we can analyze is the runtime of fitting the background distribution. For all four real-world datasets, we mined location and spread patterns and measured the time it took to find the new MaxEnt distribution incorporating both previous and the newly identified pattern, for 20 iterations. The results are presented in 6.2.

	Location pattern				Spread pattern		
Iteration	GSE	WQ	Cr	Ma	GSE	WQ	Cr
Init	9.167	8.640	9.714	8.453			
1	0.13	0.16	0.12	13.72	0.10	0.10	0.11
2	0.09	0.16	0.08	33.09	0.08	0.05	0.08
3	0.12	0.31	0.09	62.61	0.06	0.12	0.09
4	0.25	0.52	0.11	120.44	0.11	0.13	0.13
5	0.33	0.92	0.16	184.33	0.14	0.18	0.20
6	0.49	1.41	0.19	250.23	0.19	0.19	0.27
7	0.68	1.94	0.30	399.90	0.26	0.32	0.44
8	0.91	2.57	0.41	602.54	0.37	0.36	0.50
9	1.16	3.07	0.56	796.38	0.38	0.37	0.65
10	1.49	4.00	0.80	1130.81	0.42	0.46	0.83
11	1.69	5.05	1.02	-	0.42	0.49	1.07
12	1.95	6.17	1.23	-	0.52	0.57	1.32
13	2.56	7.48	1.52	-	0.63	0.65	1.62
14	2.76	9.04	1.95	-	0.68	1.16	2.09
15	3.17	10.60	2.60	-	0.72	1.00	2.86
16	3.51	11.92	3.41	-	0.81	1.06	3.42
17	4.40	14.06	4.15	-	1.12	1.38	5.01
18	4.94	15.95	5.34	-	1.17	1.47	5.69
19	4.99	17.92	6.66	-	1.07	1.57	6.30
20	5.58	19.97	6.71	-	1.24	1.92	6.65

Table 6.2: Runtime measurements to update the background distribution with identified patterns. First row shows time (in seconds) to fit the initial distribution, consecutive rows the time until convergence when incorporating additional patterns. As the updates for location and spread patterns are different, these are reported independently (columns 2–5 and 6–9). Data sets: German Socio-Economics (GSE; $n = 412$*,* $d_{\mathbf{x}} = 13$*,* $d_{\mathbf{y}} = 5$*), Water Quality (WQ;* $n = 1060$*,* $d_{\mathbf{x}} = 14$*,* $d_{\mathbf{y}} = 16$*), Crime (Cr;* $n = 1994$*,* $d_{\mathbf{x}} = 122$*,* $d_{\mathbf{y}} = 1$*), Mammals (Ma;* $n = 2220$*,* $d_{\mathbf{x}} = 67$*,* $d_{\mathbf{y}} = 124$*).*

We find that after insertion of 10–20 location patterns, the time it takes to find the MaxEnt distribution becomes noticeable. This may not be so surprising, as there are at least $d_{\mathbf{y}}$ new constraints every time we insert a new location pattern. For the Mammals data, which has target dimension 124, the time quickly grows to durations that cannot be considered acceptable for interactive use. We also observe that for spread patterns, this problem does not occur because they are by definition of low rank (the weigth vector is not necessarily sparse but it is only a one-dimensional projection).

6.4 Related work

The pattern syntax introduced in this chapter can be considered a type of *Exceptional Model Mining* (EMM) [9, 21]. EMM can be seen as a multi-target generalization of *Subgroup Discovery* (SD) [17], which is a single-target supervised form of *Pattern Mining* [24]: the broad subfield of data mining where only a part of the data is described at a time, ignoring the coherence of the remainder.

Tasks similar to SD are Contrast Set Mining [1] and Emerging Pattern Mining [8]. Both these tasks have not been considered for multiple target attributes simultaneously, and hence differ from the current chapter in that they do not directly help in understanding interactions between variables. The relationships between Contrast Set Mining, Emerging Pattern Mining, and SD are extensively described in [20].

Distribution Rules [14] can be seen as an early instance of EMM with only one target. Umek et al. [29] do consider SD with multiple targets. They approach the attribute partition in the reverse way of EMM: candidate subgroups are generated by agglomerative clustering on the targets, and predictive modeling on the descriptors strives to find matching descriptions.

Redescription Mining by Galbrun et al. [11] is the closest related work to this chapter. It considers the case where a dataset contains two distinct parts, describing the same entities from two different viewpoints. Redescription Mining treats these two parts symmetrically: it seeks descriptions inducing the same subgroup, resulting in a rule of the form $A \simeq B$. In contrast, we consider the setting where the two parts play distinct roles: one part contains description attributes on which subgroups are defined, the other part forms the numeric data which we aim to learn about and hence on which the informativeness of subgroups is evaluated. This then results in rules of the form $A \Rightarrow B$.

Interestingly, Galbrun et al. [11, Fig. 8, Tab. 6, 7] also considered the problem of 'biological niche finding' on the Mammal data. However, none of the subgroups they report are the same as ours. Their version of the data also encompasses a slightly larger region, but it is anyway unsurprising that results are quite different. The score function in Redescription Mining is not based on how much the subgroups stand out from the overall data, but only on the accuracy of the redescription and its cover. Hence, we did not further compare the results of our method with theirs.

'*Subjective Interestingness*' was first used in the context of Association Rule Mining [25, 28]. These papers formalized the prior belief of a user in a belief system, and sought association rules that contrasted with these beliefs. We base our approach on the more recent and systematic approach named FORSIED [6, 7]. This framework has been applied successfully to a variety of data mining problems, such as mining relational patterns [22], community detection [30], cluster-

ing [18], and dimensionality reduction [15]. Maximum Entropy modeling for real-valued data has also been studied before [19], in order to compute the significance of the Weighted Relative Accuracy in SD. That method targets a different pattern syntax than what is introduced here and does not apply to EMM.

Finally, Boley et al. [3] recently introduced a score function for single-target SD where a reduction in variance adds to the interestingness score of a subgroup. While their approach is less general and the interestingsness score arguably less principled, they do study the algorithmic complexity of the problem in detail and derive a tight-optimistic-estimator-based branch and bound algorithm to find the globally best subgroup pattern very efficiently.

6.5 Discussion and Conclusion

Numerous unsupervised methods exist to make sense of real-valued datasets, most notably methods for dimensionality reduction and clustering. Labels (or more generally description attributes as in this chapter) associated with the data points are then often used to interpret these results, e.g., by measuring enrichment of certain labels within a cluster, or by coloring data points in a scatter plot of a 2-D projection of the data with a color depending on the labels of the points, for subsequent visual inspection. However, whether such analyses provide explanations or insights is a matter of coincidence: there is no a priori reason that clusters should be enriched, and there is no guarantee that equally colored points are grouped in a scatter plot.

Here, we propose an alternative approach, in directly using the description attributes to guide the search for surprising multivariate relations in the data. Resulting subgroups are then automatically explained well by the descriptions. Our approach contrasts with traditional supervised methods in focusing on *local* patterns: properties of the target attributes that apply only to subsets of the data defined in terms of conditions on their metadata. Arguably, with increasing amounts and resulting inhomogeneity of datasets, the importance of local patterns is bound to increase.

Our approach generalizes the literature on Subgroup Discovery and Exceptional Model Mining in being applicable for real-valued target attributes of arbitrary dimensionality, and in searching for multivariate local patterns across all these dimensions, including unusual covariance structures of subgroups in the data. Moreover, the interestingness of the patterns of this type is formalized in a rigorous manner, quantifying the amount of information the user gains by observing them. We have demonstrated that the resulting algorithms are effective and efficient, in theory and in practice.

In further work, we plan to remove the dependency on third party tools (Matlab and Cortana) and produce a standalone version of the method for public dissemi-

nation. Furthermore, it would be interesting to study similar pattern syntaxes for binary, categorical, and mixed sets of target attributes. Besides, although we have little hope to improve the search for optimal spread patterns, it may be feasible to devise a branch-and-bound approach to mine optimal location patterns efficiently. Indeed this appears to be the most relevant question to be addressed in the future. Finally, we aim to integrate this method with SIDE [16, 26], our online tool for exploration of numerical data, which currently does not use any labels or description attributes.

Acknowledgements. This work has been supported by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 615517, FWO (project no. G091017N, G0F9816N), the EU's Horizon 2020 research and innovation programme and the FWO under the Marie Skłodowska-Curie Grant Agreement no. 665501, the Academy of Finland (decision 288814), and Tekes (Revolution of Knowledge Work project).

References

[1] S. D. Bay and M.J. Pazzani. Detecting group differences: Mining contrast sets. *DMKD*, 5(3):213–246, 2001.

[2] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proc. of KDD-IDEA Workshop*, pages 27–35, 2013.

[3] Mario Boley, Bryan R. Goldsmith, Luca M. Ghiringhelli, and Jilles Vreeken. Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. In *Proc. of ECML-PKDD*, 2017.

[4] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *JMLR*, 15(1):1455–1459, 2014.

[5] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, 2nd edition, 2005.

[6] Tijl De Bie. An information theoretic framework for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 564–572, New York, NY, USA, 2011. ACM.

[7] Tijl De Bie. Subjective interestingness in exploratory data mining. In *International Symposium on Intelligent Data Analysis*, pages 19–31, Berlin, Heidelberg, 2013. Springer.

[8] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of KDD*, pages 43–52, 1999.

[9] W. Duivesteijn, A. Feelders, and A.J. Knobbe. Exceptional model mining - supervised descriptive local pattern mining with complex target concepts. *DMKD*, 30(1):47–98, 2016.

[10] S. Džeroski, D. Demšar, and J. Grbović. Predicting chemical parameters of river water quality from bioindicator data. *Appl. Intell.*, 13(1):7–17, 2000.

[11] E. Galbrun and P. Miettinen. From black and white to full color: extending redescription mining outside the boolean world. *SADM*, 5(4):284–303, 2012.

[12] Floris Geerts, Bart Goethals, and Taneli Mielikäinen. Tiling databases. In *Proc. of DS*, pages 278–289, 2004.

[13] H. Heikinheimo, M. Fortelius, J. Eronen, and H. Mannila. Biogeography of european land mammals shows environmentally distinct and spatially coherent clusters. *J. Biogeogr*, 34(6):1053–1064, 2007.

[14] A.M. Jorge, P.J. Azevedo, and F. Pereira. Distribution rules with numeric attributes of interest. In *Proc. of PKDD*, pages 247–258, 2006.

[15] B. Kang, J. Lijffijt, R. Santos-Rodríguez, and T. De Bie. Subjectively interesting component analysis: Data projections that contrast with prior expectations. In *Proc. of KDD*, pages 1615–1624, 2016.

[16] Bo Kang, Kai Puolamäki, Jefrey Lijffijt, and Tijl De Bie. A tool for subjective and interactive visual data exploration. In *Proc. of ECML-PKDD - Part III*, pages 3–7, 2016.

[17] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *AKDDM*, pages 249–271, 1996.

[18] K.-N. Kontonasios and T. De Bie. Subjectively interesting alternative clusterings. *MLJ*, 98(1):31–56, 2015.

[19] K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy modelling for assessing results on real-valued data. In *Proc. of ICDM*, pages 350–359, 2011.

[20] P. Kralj Novak, N. Lavrač, and G.I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *JMLR*, 10:377–403, 2009.

[21] D. Leman, A. Feelders, and A.J. Knobbe. Exceptional model mining. In *Proc. ECML-PKDD*, pages 1–16, 2008.

[22] J. Lijffijt, E. Spyropoulou, B. Kang, and T. De Bie. P-n-rminer: A generic framework for mining interesting structured relational patterns. *IJDSA*, 1(1): 61–76, 2016.

[23] M. Meeng and A. Knobbe. Flexible enrichment with cortana–software demo. In *Proc. of BeneLearn*, pages 117–119, 2011.

[24] K. Morik, J.-F. Boulicaut, and A. Siebes, editors. *Local Pattern Detection, International Seminar, Dagstuhl Castle, Germany, April 12-16, 2004, Revised Selected Papers*, volume 3539 of *LNCS*, 2005.

[25] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Proc. of KDD*, pages 94–100, 1998.

[26] Kai Puolamäki, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Interactive visual data exploration with subjective feedback. In *Proc. of ECML-PKDD - Part II*, pages 214–229, 2016.

[27] M. A. Redmond and A. Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *EJOR*, 141:660–678, 2002.

[28] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proc. of KDD*, pages 275–281, 1996.

[29] L. Umek and B. Zupan. Subgroup discovery in data sets with multi-dimensional responses. *IDA*, 15(4):533–549, 2011.

[30] Matthijs van Leeuwen, Tijl De Bie, Eirini Spyropoulou, and Cédric Mesnage. Subjective interestingness of subgraph patterns. *Machine Learning*, 105(1): 41–75, Oct 2016. ISSN 1573-0565. doi: 10.1007/s10994-015-5539-3. URL https://doi.org/10.1007/s10994-015-5539-3.

[31] J.-T. Zhang. Approximate and asymptotic distributions of chi-squared–type mixtures with applications. *JASA*, 100(469):273–285, 2005.

7 Conclusion and Future Work

7.1 Conclusion

Representation learning has gained enormous popularity due to its capability to capture rich information from the data and its easy-to-compute nature. Despite the success of representation learning, it currently has two limitations. First, high-dimensional data often has many aspects, a low-dimensional data representation is typically insufficient to capture all structure in the data and the most salient structure is often already known. It is not obvious how to capture the remaining information in a similarly effective way. Second, the structure in a dataset may exceed the representational power of Euclidean space. Hence, the data might be underrepresented.

To address the two issues, this thesis proposes a framework for learning Subjectively Interesting Data Representations (SIDR). The framework delineates how to take a prior about the data and find data representations that complement the prior.

First, by discounting the known salient structure, the SIDR framework enables complementary structure to be captured, allowing the remaining information to be captured effectively. Along this line, we developed a linear dimensionality reduction method called Subjectively Interesting Component Analysis (SICA) in order to explore the remaining information via linear projections. For complex non-linear structure remaining in the data, we further proposed Conditional t-distributed Stochastic Neighbor Embedding (ct-SNE), a conditioned version of

t-distributed Stochastic Neighbor Embedding. SICA and ct-SNE are evaluated in extensive case studies on both synthetic and (large) real-world datasets. The results show both methods effectively discount the prior knowledge and allow the remaining structure in the data to be efficiently explored.

Second, by encoding specific complex structure in the data as prior, the important information can be represented more accurately in Euclidean representations to capture. Combining the prior and the Euclidean representation, representation learning methods can yield better models of the data. This idea is readily applicable to network embeddings, where network structural properties such as (approximate) multipartiteness, certain degree distributions, or assortativity are difficult to express using Euclidean space. By applying SIDR, we derived Conditional Network Embeddings (CNE) that optimizes network representations with respect to certain prior knowledge about the network. We evaluated the performance of CNE on standard network analysis tasks such as link prediction and node classifications. Comparing to heuristic methods and state-of-art NE methods on a wide range of networks, CNE shows superior performance. This shows CNE is capable of better representing network data. Additionally, CNE also demonstrates potential for network visualization.

To enable real users to explore data using subjectively interesting linear projections, this thesis also presented an application of SIDR framework on iterative and interactive visual data analysis, named Subjectively Interesting Data Exploration (SIDE). Using SIDE, users can interactively select or label patterns in low-dimensional visualizations during their exploration. SIDE accumulates the learned patterns as prior and presents more informative representations to users. Case studies on both synthetic and real-world data show SIDE is useful for discovering subjectively interesting structure from data in an iterative and interactive manner.

Last but not least, this thesis takes a step along the direction for improving the interpretability of subjectively interesting linear projections. Introduced under the SIDR framework, Subjectively Interesting Subgroup Discovery (SISD) searches subjectively interesting representations that are both informative and descriptive. Case studies on synthetic and real-world datasets show the capability of SISD to provide interesting representations with concise descriptions.

7.2 Future work

Representation learning can be viewed as a learning process that disentangles the factors of variation in the observed data [1]. The superior performance of most representation learning methods is due to the implicit inductive biases (e.g., human knowledge) for disentangling the factors of variation from data that are encoded in the methods themselves [2]. As a representation learning framework, SIDR formalizes how to use priors to explicitly encode the inductive bias introduced by

a user. This enables us to peel the prior from the data and obtains embeddings that contain the remaining factors disentangled from the prior. As a result, the unmodeled information in the embedding become more salient. The combination of prior and embedding also gives a better model of the data. Under this perspective, a general research direction for SIDR is to generalize to other data types as well as to incorporate new types of priors. For time series data, the idea of subjective interestingness has been used to find informative motives with respect to a given prior [3]. For other data types such as image, we speculate SIDR framework can be used to untangle the complex information in the data from Euclidean embeddings to give a better representation (similar to CNE). Currently the prior types in SIDR are derived for specific purposes and data types. To extend the applicability of SIDR, a general language of encoding priors in SIDR would need to be developed.

For dimensionality-reduction instantiations SICA and ct-SNE, a further research direction is to stack the priors that encode information learnt by a user along his/her exploration and compute the next informative representation. SIDE achieves this goal. However, it visualizes different aspects of the data in a sequential manner. As different interactions of the user leads to different (ordering of) visualizations to be explored, users can easily lose track of the high-level picture of the data when investigating the different aspects sequentially. One solution is to keep a tree-like exploration history, and allow user to navigate through the history for retrospection. The visualization at a 'tree node' could be further reloaded in order to start a different exploration path. Such history tracking can remind users about the previously explored aspects and help them to piece together a holistic understanding about the data.

CNE has shown it superior performance in network analysis tasks. However, it still requires some hyper-parameter tuning when fitting different networks. Although CNE comes with a default hyper-parameter setting, its performance on different dataset still drastically varies based on the characteristics (e.g., size, density, multipartitienss) of a network. Thus, the next step would be to benchmark CNE on various types of networks and find the best parameters according to the characteristics of networks and downstream tasks. Additionally, an automatic hyper-parameter tuning module (e.g., via grid searching, Bayesian optimization) could be included to fine-tune the hyper-parameters for specific embedding cases.

Evaluation of the methods introduced here are mostly done through case studies, where the results produced by the methods are analyzed and reasoned against a set of assumption about the analysts. The main goal here is indeed to show these methods are able to produce data representations that are subjectively interesting to users. However, since subjective interestingness is user specific, the evaluation results would be more definitive if they are based on real user studies rather than a set of abstract assumptions. For this purpose, we could adapt the Creedo evaluation framework [4] for empirically evaluating knowledge discovery systems.

Creedo allows user studies to be easily configured for testing how well certain data analysis methods support real users to perform certain data analysis tasks based on certain evaluation criteria. That said, we would like to evaluate the usefulness of SIDR through user studies in the near future.

References

[1] Yoshua Bengio. *Deep Learning of Representations: Looking Forward*. CoRR, abs/1305.0445, 2013.

[2] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. *Relational inductive biases, deep learning, and graph networks*. CoRR, abs/1806.01261, 2018.

[3] Junning Deng, Jefrey Lijffijt, Bo Kang, and Tijl De Bie. *SIMIT: Subjectively Interesting Motifs in Time Series*. Entropy, 21(6):566, 2019.

[4] Mario Boley, Maike Krause-Traudes, Bo Kang, and Björn Jacobs. *Creedo—scalable and repeatable extrinsic evaluation for pattern discovery systems by online user studies*. In ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, page 20, 2015.