A kernel-based framework for learning graded relations from data

Willem Waegeman, Tapio Pahikkala, *Member, IEEE,* Antti Airola, Tapio Salakoski, Michiel Stock, Bernard De Baets

***Abstract*—Driven by a large number of potential applications in areas such as bioinformatics, information retrieval and social network analysis, the problem setting of inferring relations between pairs of data objects has recently been investigated intensively in the machine learning community. To this end, current approaches typically consider datasets containing crisp relations, so that standard classification methods can be adopted. However, relations between objects like similarities and preferences are often expressed in a graded manner in real-world applications. A general kernel-based framework for learning relations from data is introduced here. It extends existing approaches because both crisp and graded relations are considered, and it unifies existing approaches because different types of graded relations can be modeled, including symmetric and reciprocal relations. This framework establishes important links between recent developments in fuzzy set theory and machine learning. Its usefulness is demonstrated through various experiments on synthetic and real-world data. The results indicate that incorporating domain knowledge about relations improves the predictive performance.**

***Index Terms*—graded relations, fuzzy relations, reciprocal relations, transitivity, learning in graphs, kernel methods, machine learning**

I. Introduction

Relational data occurs in many predictive modeling tasks, such as forecasting the winner in two-player computer games [1], predicting proteins that interact with other proteins in bioinformatics [2], retrieving documents that are similar to a target document in text mining [3], investigating the persons that are friends of each other on social network sites [4], etc. All these examples represent fields of application in which specific machine learning and data mining algorithms have been successfully developed to infer relations from data; pairwise relations, to be more specific.

The typical learning scenario in such situations can be summarized as follows. Given a dataset of known relations between pairs of objects and a feature representation of these objects in terms of variables that might characterize the relations, the goal usually consists of inferring a statistical model that takes two objects as input and predicts whether the relation of interest occurs for these two objects. Moreover, since one aims to discover unknown relations, a good learning algorithm should be able to construct a predictive model that can generalize for unseen data, i.e., pairs of objects for which at least one of the two objects was not used to construct the model. As a result of the transition from predictive models for single objects to pairs of objects, new advanced learning algorithms need to be developed, resulting in new challenges with regard to model construction, computational tractability and model assessment.

As relations between objects can be observed in many different forms, this general problem setting provides links to several subfields of machine learning, like statistical relational learning [5], graph mining [6], metric learning [7] and preference learning [8]. More specifically, from a graph-theoretic perspective, learning a relation can be formulated as learning edges in a graph where the nodes represent information about the data objects; from a metric learning perspective, the relation that we aim to learn should satisfy some well-defined properties like positive definiteness, transitivity or the triangle inequality; and from a preference learning perspective, the relation expresses a (degree of) preference in a pairwise comparison of data objects.

The topic of learning relations between objects is also closely related to recent developments in fuzzy set theory. This article will elaborate on these connections via two important contributions: (1) the extension of the typical setting of learning crisp relations to real-valued and ordinal-valued relations and (2) the incorporation of domain knowledge about relations into the inference process by explicit modeling mathematical properties of these relations. For algorithmic simplicity, one can observe that many approaches only learn crisp relations, that is relations with only 0 and 1 as possible values, so that standard binary classifiers can be used. In this context, consider examples such as inferring protein-protein interaction networks or metabolic networks in bioinformatics [2], [9].

However, graded relations are observed in many real-world applications [10], resulting in a need for new algorithms that take graded relational information into account. Furthermore, the properties of graded relations have been investigated intensively in the recent fuzzy logic literature[1], and these properties are very useful to analyze and improve current algorithms. Using the mathematical properties of graded relations, constraints can be imposed to allow for incorporation of domain knowledge in the learning process, to improve predictive performance or simply to guarantee that a relation with the right properties is learned. This is definitely the

W. Waegeman, M. Stock and B. De Baets are with the Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000 Ghent, Belgium, email: forname.surname@ugent.be.

T. Pahikkala, A. Airola and T. Salakoski are with the Department of Information Technology and the Turku Centre for Computer Science, University of Turku, Joukahaisenkatu 3-5 B 20520 Turku, email: forname.surname@utu.fi.

[1] Often the term fuzzy relation is used in the fuzzy set literature to refer to graded relations. However, fuzzy relations should be seen as a subclass of graded relations. For example, reciprocal relations should not be considered as fuzzy relations, because they often exhibit a probabilistic semantics rather than a fuzzy semantics.

case for properties like transitivity when learning similarity relations and preference relations – see e.g. [11]–[14], but even very basic properties like symmetry, antisymmetry or reciprocity already provide domain knowledge that can steer the learning process. For example, in social network analysis, the notion "person A being a friend of person B" should be considered as a symmetric relation, while the notion "person A defeats person B in a chess game" will be antisymmetric (or, equivalently, reciprocal). Nevertheless, many examples exist, where neither symmetry nor antisymmetry necessarily hold, like the notion "person A trusts person B".

In this paper we present a general kernel-based approach that unifies all the above cases into one general framework where domain knowledge can be easily specified by choosing a proper kernel and model structure, while different learning settings are distinguished by means of the loss function. Let $Q(v, v')$ be a binary relation on an object space $\mathcal{V}$, then the following learning settings will be considered in particular:

- Crisp relations: when the restriction is made that $Q : \mathcal{V}^2 \rightarrow \{0, 1\}$, we arrive at a binary classification task with pairs of objects as input for the classifier.
- $[0, 1]$-valued relations: here it is allowed for relations to take the form $Q : \mathcal{V}^2 \rightarrow [0, 1]$, resulting in a regression type of learning setting. The restriction to the interval $[0, 1]$ is predominantly made because many mathematical frameworks in fields like fuzzy set theory and decision theory are built upon such relations, using the notion of a fuzzy relation, but in general one can account quite easily for real-graded relations by applying a scaling operation from $\mathbb{R}$ to $[0, 1]$.
- Ordinal-valued relations: situated somewhat in the middle between the other two settings, here it is assumed that the actual values of the relation do not matter but rather the provided order information should be learned.

Furthermore, one can integrate different types of domain knowledge in our framework, by guaranteeing that certain properties are satisfied. The following cases can be distinguished:

- Symmetric relations. Applications arise in many domains and metric learning or learning similarity measures can be seen as special cases requiring additional properties to hold, such as the triangle inequality for metrics and positive definiteness or transitivity properties for similarity measures. As shown below, learning symmetric relations can be seen as learning edges in an undirected graph.
- Reciprocal or antisymmetric relations. Applications arise here in domains such as preference learning, game theory and bioinformatics for representing preference relations, choice probabilities, winning probabilities, gene regulation, etc. We will provide a formal definition below, but, given a rescaling operation from $\mathbb{R}$ to $[0, 1]$, antisymmetric relations can be converted into reciprocal relations. Similar to symmetric relations, transitivity properties typically guarantee additional constraints that are definitely required for certain applications. It is, for example, well known in decision theory and preference modeling that transitive preference relations result in utility functions [15], [16]. Learning reciprocal or antisymmetric relations can be interpreted as learning edges in a directed graph.
- Ordinary binary relations. Many applications can be found where neither symmetry nor reciprocity holds. From a graph inference perspective, learning such relations should be seen as learning the edges in a bidirectional graph, where edges in one direction do not impose constraints on edges in the other direction.

Indeed, the framework that we propose below strongly relies on graphs, where nodes represent the data objects that are studied and the edges represent the relations present in the training set. The weights on the edges characterize the values of known relations, while unconnected nodes indicate pairs of objects for which the unknown relation needs to be predicted. The left graph in Figure 1 visualizes a toy example representing the most general case where neither symmetry nor reciprocity holds. Depending on the application, the learning algorithm should try to predict the relations for three types of object pairs:

- pairs of objects that are already present in the training dataset by means of other edges, like the pair (A,B),
- pairs of objects for which one of the two objects occurs in the training dataset, like the pair (E,F),
- pairs of objects for which none of the two objects is observed during training, like the pair (F,G).

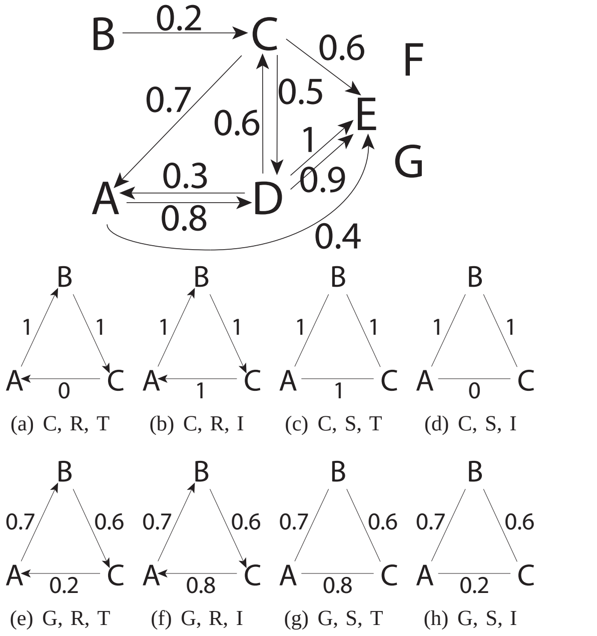
The graphs on the right-hand side in Figure 1 show examples of specific types of relations that are covered by our framework. The differences between these relations will become more clear in the following sections.

We start in Section 2 with a formal definition of our framework. The Kronecker pairwise product kernel is introduced as a general-purpose tool for modeling arbitrary binary relations. This claim is supported by Theorem II.1, a theoretical result stating that universal approximation can be obtained for the Kronecker product pairwise kernel. Subsequently, we analyze in Section 3 reciprocal and symmetric relations as two important special cases of our framework. It is shown that such prior knowledge can be easily incorporated by defining suitable kernel functions. In Section 4, we investigate additional properties of reciprocal and symmetric relations, such as transitivity and metric properties, while establishing important connections with existing kernel functions for paired comparisons and recent developments in fuzzy set theory. Further connections with related work are summarized in Section 5. Finally, Section 6 presents experimental results for case studies in different domains (game playing, document retrieval and ecology), emphasizing the generality of our framework. Well-known pairwise kernel functions are compared to illustrate that inclusion of domain knowledge influences the predictive performance. Scaling experiments confirm that this influence increases when the sample size decreases.

II. General framework

A. Notation and basic concepts

Let us start with introducing some notation. We assume that the data is structured as a graph $G = (\mathcal{V}, \mathcal{E}, Q)$, where $\mathcal{V}$ corresponds to the set of nodes, $\mathcal{E} \subseteq \mathcal{V}^2$ represents the set



(a) C, R, T (b) C, R, I (c) C, S, T (d) C, S, I

(e) G, R, T (f) G, R, I (g) G, S, T (h) G, S, I

Fig. 1. Top: example of a multi-graph representing the most general case, where no additional properties of relations are assumed. Bottom: examples of eight different types of relations in a graph of cardinality three. The following relational properties are illustrated: (C) crisp, (G) graded, (R) reciprocal, (S) symmetric, (T) transitive and (I) intransitive. For the reciprocal relations, (I) refers to a relation that does not satisfy weak stochastic transitivity, while (T) is showing an example of a relation fulfilling strong stochastic transitivity. For the symmetric relations, (I) refers a relation that does not satisfy T-transitivity w.r.t. the Łukasiewicz t-norm $T_{\mathbf{L}}(a,b) = \max(a+b-1,0)$, while (T) is showing an example of a relation that fulfills T-transitivity w.r.t. the product t-norm $T_{\mathbf{P}}(a,b) = ab$. See Section 4 for formal definitions of transitivity.

of edges, and the edges are associated with labels generated from an unknown underlying relation $Q : \mathcal{V}^2 \to [0,1]$. Relations are required to take values in the interval $[0,1]$ because some properties that we need are historically defined for such relations, but an extension to real-graded relations $h : \mathcal{V}^2 \to \mathbb{R}$ can always be realized. Consider $b \in \mathbb{R}^+$ and an increasing isomorphism $\sigma : [-b,b] \to [0,1]$ that satisfies $\sigma(x) = 1 - \sigma(-x)$, then we consider the $\mathbb{R} \to [0,1]$ mapping ∇ defined by:

$$\nabla(x) \quad = \quad \begin{cases} 0, & \text{if } x \leq -b \\ \sigma(x), & \text{if } -b \leq x \leq b \\ 1, & \text{if } b \leq x \end{cases}$$

and its inverse $\nabla^{-1} = \sigma^{-1}$.

Any real-valued relation $h : \mathcal{V}^2 \to \mathbb{R}$ can be transformed into a $[0,1]$-valued relation Q as follows:

$$Q(v,v') = \nabla(h(v,v'))\,, \quad \forall (v,v') \in \mathcal{V}^2\,, \tag{1}$$

and conversely by means of ∇^{-1}. In what follows we tacitly assume that ∇ has been fixed.

Following the standard notation for kernel methods, we formulate our learning problem as the selection of a suitable function $h \in \mathcal{H}$, where $\mathcal{H}$ is a hypothesis space, in particular a reproducing kernel Hilbert space (RKHS). More specifically, the RKHS supports in our case hypotheses $h : \mathcal{V}^2 \to \mathbb{R}$ denoted as

$$h(e) = \mathbf{w}^{\mathrm{T}}\Phi(e)\,, \tag{2}$$

with $\mathbf{w}$ a vector of parameters that needs to be estimated from training data, Φ a joint feature mapping for edges in the graph (see below) and $\mathbf{a}^{\mathrm{T}}$ the transpose of a vector $\mathbf{a}$. Let us denote a training dataset of cardinality q as a sequence $(e_i, y_{e_i})_{i=1}^q$ of edge-label pairs sampled from G. Then, we formally consider the following optimization problem, in which we select an appropriate hypothesis h from $\mathcal{H}$ for training data:

$$\operatorname*{argmin}_{h \in \mathcal{H}} \left\{ \frac{1}{q} \sum_{i=1}^{q} \mathcal{L}(h(e_i), y_{e_i}) + \lambda \|h\|_{\mathcal{H}}^2 \right\} \tag{3}$$

with $\mathcal{L}$ a given loss function, $\|\cdot\|_{\mathcal{H}}^2$ the traditional quadratic regularizer on the RKHS and $\lambda > 0$ a regularization parameter. According to the representer theorem [17], any minimizer $h \in \mathcal{H}$ of (3) admits a dual representation of the following form:

$$h(\overline{e}) = \mathbf{w}^{\mathrm{T}}\Phi(\overline{e}) = \sum_{i=1}^{q} a_i K^{\Phi}(e_i, \overline{e})\,, \tag{4}$$

with $a_i \in \mathbb{R}$ dual parameters, K^{Φ} the kernel function associated with the RKHS and Φ the feature mapping corresponding to K^{Φ} and $\mathbf{w} = \sum_{i=1}^{q} a_i \Phi(e_i)$.

The primal representation as defined in (2) and its dual equivalent (4) yield an RKHS defined on edges in the graph. In addition, we will establish an RKHS defined on nodes, as every edge consists of a couple of nodes. Given an input space $\mathcal{V}$ and a kernel $K : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, the RKHS associated with K can be considered as the completion of

$$\left\{ f \in \mathbb{R}^{\mathcal{V}} \;\middle|\; f(v) = \sum_{i=1}^{m} \beta_i K(v, v_i) \right\},$$

in the norm $\|f\|_K = \sqrt{\sum_{i,j} \beta_i \beta_j K(v_i, v_j)}$, where $\beta_i \in \mathbb{R}, m \in \mathbb{N}, v_i \in \mathcal{V}$.

B. Learning arbitrary relations

As mentioned in the introduction, both crisp and graded relations can be handled by our framework. To make a subdivision between different cases, a loss function needs to be specified. For crisp relations, one will typically use the hinge loss or the logistic loss. Conversely, if in a given application the observed relations are graded instead of crisp, other loss functions have to be considered. Hence, we will run experiments with a least-squares loss function:

$$\mathcal{L}(h(e), y) = (y_e - h(e))^2\,, \tag{5}$$

resulting in a regression type of learning setting.

So far, our framework does not differ from standard classification and regression algorithms. However, the specification of a more precise model structure for (2) offers a couple of new challenges. In the most general case, when no further restrictions on the underlying relation can be specified, the following Kronecker product feature mapping is proposed to express pairwise interactions between features of nodes:

$$\Phi(e) = \Phi(v,v') = \phi(v) \otimes \phi(v')\,,$$

where ϕ represents the feature mapping for individual nodes. A formal definition of the Kronecker product can be found in the appendix. As first shown in [18], the Kronecker product pairwise feature mapping yields the Kronecker product edge kernel (a.k.a. the tensor product pairwise kernel) in the dual representation:

$$K_{\otimes}^{\Phi}(e,\overline{e}) = K_{\otimes}^{\Phi}(v,v',\overline{v},\overline{v}') = K^{\phi}(v,\overline{v})K^{\phi}(v',\overline{v}')\,, \tag{6}$$

with K^{ϕ} the kernel corresponding to ϕ.

This section aims to formally prove that the Kronecker product edge kernel is the best kernel one can choose, when no further domain knowledge is provided about the underlying relation that generates the data. We claim that with an appropriate choice for K^{ϕ}, such as the Gaussian RBF kernel, the kernel K^{Φ} generates a class $\mathcal{H}$ of universally approximating functions for learning any type of relation. Before summarizing this important result, we recollect the classical concept of universal kernels (see Definition VII.1 in the appendix) introduced by [19]. With universality of the kernel K, we refer to the property that the RKHS induced by K can approximate any function in $C(\mathcal{V})$ arbitrarily well, where $C(\mathcal{V})$ is the set of real-valued continuous functions on $\mathcal{V}$. Using another classical result, the Stone-Weierstraß theorem (see Theorem VII.2 in the appendix, and e.g [20] for a more detailed description), we arrive at the following theorem concerning the Kronecker product pairwise kernels:

Theorem II.1. *Let us assume that the space of nodes $\mathcal{V}$ is a compact metric space. If a continuous kernel K^{ϕ} is universal on $\mathcal{V}$, then $K_{\otimes}^{\Phi}$ defines a universal kernel on $\mathcal{V}^2$.*

The proof can be found in the appendix. We would like to emphasize that one cannot conclude from the theorem that the Kronecker product pairwise kernel is the best kernel to use in all possible situations. The theorem only shows that the Kronecker product pairwise kernel makes a reasonably good choice, if no further domain knowledge about the underlying relation is known. Namely, the theorem says that given a suitable sample of data, the RKHS of the kernel contains functions that are arbitrarily close to any continuous relation in the uniform norm. However, the theorem does not say anything about how likely it is to have, as a training set, such a data sample that is representative of the approximating function. Further, the theorem only concerns graded relations that are continuous and therefore crisp relations and graded, discontinuous relations require more detailed consideration.

Other kernel functions might, of course, outperform the Kronecker product pairwise kernel in applications where domain knowledge can be incorporated in the kernel function. In the following section we discuss reciprocity, symmetry and transitivity as three relational properties that can be represented by means of more specific kernel functions. As a side note, we also introduce the Cartesian pairwise kernel, which is formally defined as follows

$$K_{C}^{\Phi}(v,v',\overline{v},\overline{v}') = K^{\phi}(v',\overline{v}')[v=\overline{v}] + K^{\phi}(v,\overline{v})[v'=\overline{v}']\,,$$

with $[.]$ the indicator function, returning one when both elements are identical and zero otherwise. This kernel was recently proposed by [21] as an alternative to the Kronecker product pairwise kernel. By construction, the Cartesian pairwise kernel has important limitations, since it cannot generalize to couples of nodes for which both nodes did not appear in the training dataset.

III. Special relations

If no further information is available about the relation that underlies the data, one should definitely use the Kronecker product edge kernel. In this most general case, we allow that for any pair of nodes in the graph several edges can exist, in which an edge in one direction does not necessarily impose constraints on the edge in the opposite direction. Multiple edges in the same direction can connect two nodes, leading to a multi-graph as in Figure 1, where two different edges in the same direction connect nodes D and E. This construction is required to allow repeated measurements. However, two important subclasses of relations deserve further attention: reciprocal relations and symmetric relations.

A. Reciprocal relations

This subsection briefly summarizes our previous work on learning reciprocal relations [22]. Let us start with a definition of this type of relation.

Definition III.1. *A binary relation $Q : \mathcal{V}^2 \rightarrow [0,1]$ is called a reciprocal relation if for all $(v,v') \in \mathcal{V}^2$ it holds that $Q(v,v') = 1 - Q(v',v)$.*

Definition III.2. *A binary relation $h : \mathcal{V}^2 \rightarrow \mathbb{R}$ is called an antisymmetric relation if for all $(v,v') \in \mathcal{V}^2$ it holds that $h(v,v') = -h(v',v)$.*

For reciprocal and antisymmetric relations, every edge $e = (v,v')$ in a multi-graph like Figure 1 induces an unobserved invisible edge $e_R = (v',v)$ with appropriate weight in the opposite direction. The transformation operator ∇ transforms an antisymmetric relation into a reciprocal relation. Applications of reciprocal relations arise here in domains such as preference learning, game theory and bioinformatics for representing preference relations, choice probabilities, winning probabilities, gene regulation, etc. The weight on the edge defines the real direction of such an edge. If the weight on the edge $e = (v,v')$ is higher than 0.5, then the direction is from v to v', but when the weight is lower than 0.5, then the direction should be interpreted as inverted, for example, the edges from A to C in Figures 1 (a) and (e) should be interpreted as edges starting from A instead of C. If the relation is 3-valued as $Q : \mathcal{V}^2 \rightarrow \{0, 1/2, 1\}$, then we end up with a three-class ordinal regression setting instead of an ordinary regression setting.

Interestingly, reciprocity can be easily incorporated in our framework.

Proposition III.3. *Let Ψ be a feature mapping on $\mathcal{V}^2$ and let h be a hypothesis defined by (2), then the relation Q of type (1) is reciprocal if Φ is given by*

$$\Phi_R(e) = \Phi_R(v,v') = \Psi(v,v') - \Psi(v',v)\,.$$

The proof is immediate. In addition, one can easily show that reciprocity as domain knowledge can be enforced in the dual formulation. Let us in the least restrictive form now consider the Kronecker product for Ψ, then one obtains for Φ_R the kernel $K^{\Phi}_{\otimes R}$ given by $K^{\Phi}_{\otimes R}(e, \overline{e}) =$

$$2\big(K^{\phi}(v, \overline{v})K^{\phi}(v', \overline{v}') - K^{\phi}(v, \overline{v}')K^{\phi}(v', \overline{v})\big)\,. \tag{7}$$

The following theorem shows that this kernel can represent any type of reciprocal relation.

Theorem III.4. *Let*

$$R(\mathcal{V}^2) = \{t \mid t \in C(\mathcal{V}^2), t(v, v') = -t(v', v)\}$$

be the space of all continuous antisymmetric relations from $\mathcal{V}^2$ to $\mathbb{R}$. If K^{ϕ} on $\mathcal{V}$ is universal, then for every function $t \in R(\mathcal{V}^2)$ and every $\epsilon > 0$, there exists a function h in the RKHS induced by the kernel $K^{\Phi}_{\otimes R}$ defined in (7), such that

$$\max_{(v,v') \in \mathcal{V}^2} \{|t(v, v') - h(v, v')|\} \leq \epsilon\,. \tag{8}$$

The proof can be found in the appendix.

B. Symmetric relations

Symmetric relations form another important subclass of relations in our framework. As a specific type of symmetric relations, similarity relations constitute the underlying relation in many application domains where relations between objects need to be learned. Symmetric relations are formally defined as follows.

Definition III.5. *A binary relation $Q : \mathcal{V}^2 \to [0, 1]$ is called a symmetric relation if for all $(v, v') \in \mathcal{V}^2$ it holds that $Q(v, v') = Q(v', v)$.*

Definition III.6. *A binary relation $h : \mathcal{V}^2 \to \mathbb{R}$ is called a symmetric relation if for all $(v, v') \in \mathcal{V}^2$ it holds that $h(v, v') = h(v', v)$.*

Note that ∇ preserves symmetry. For symmetric relations, edges in multi-graphs like Figure 1 become undirected. Applications arise in many domains and metric learning or learning similarity measures can be seen as special cases. If the relation is 2-valued as $Q : \mathcal{V}^2 \to \{0, 1\}$, then we end up with a classification setting instead of a regression setting.

Just like reciprocal relations, it turns out that symmetry can be easily incorporated in our framework.

Proposition III.7. *Let Ψ be a feature mapping on $\mathcal{V}^2$ and let h be a hypothesis defined by (2), then the relation Q of type (1) is symmetric if Φ is given by*

$$\Phi_S(e) = \Phi_S(v, v') = \Psi(v, v') + \Psi(v', v)\,.$$

In addition, by using mathematical properties of the Kronecker product, one obtains in the dual formulation an edge kernel that looks very similar to the one derived for reciprocal relations. Let us again consider the Kronecker product for Ψ, then one obtains for Φ_S the kernel $K^{\Phi}_{\otimes S}$ given by $K^{\Phi}_{\otimes S}(e, \overline{e}) =$

$$2\big(K^{\phi}(v, \overline{v})K^{\phi}(v', \overline{v}') + K^{\phi}(v, \overline{v}')K^{\phi}(v', \overline{v})\big)\,.$$

Thus, the substraction of kernels in the reciprocal case becomes an addition of kernels in the symmetric case. The above kernel has been used for predicting protein-protein interactions in bioinformatics [18] and it has been theoretically analyzed in [23]. More specifically, for some methods one has shown in the latter paper that enforcing symmetry in the kernel function yields identical results as adding every edge twice to the dataset, by taking each of the two nodes once as first element of the edge. Unlike many existing kernel-based methods for pairwise data, the models obtained with these kernels are able to represent any reciprocal or symmetric relation respectively, without imposing additional transitivity properties of the relations.

We also remark that for symmetry as well, one can prove that the Kronecker product edge kernel yields a model that is flexible enough to represent any type of underlying relation.

Theorem III.8. *Let*

$$S(\mathcal{V}^2) = \{t \mid t \in C(\mathcal{V}^2), t(v, v') = t(v', v)\}$$

be the space of all continuous symmetric relations from $\mathcal{V}^2$ to $\mathbb{R}$. If K^{ϕ} on $\mathcal{V}$ is universal, then for every function $t \in S(\mathcal{V}^2)$ and every $\epsilon > 0$, there exists a function h in the RKHS (2) induced by the kernel (7), such that

$$\max_{(v,v') \in \mathcal{V}^2} \{|t(v, v') - h(v, v')|\} \leq \epsilon.$$

The proof is analogous to that of Theorem III.4 (see appendix). As a side note, we remark that a symmetric and reciprocal version of the Cartesian kernel can be introduced as well.

IV. Relationships with fuzzy set theory

The previous section revealed that specific Kronecker product edge kernels can be constructed for modeling reciprocal and symmetric relations, without requiring any further background about these relations. In this section we demonstrate that the Kronecker product edge kernels $K^{\Phi}_{\otimes}$, $K^{\Phi}_{\otimes R}$ and $K^{\Phi}_{\otimes S}$ are particularly useful for modeling intransitive relations. Intransitive relations occur in a lot of real-world scenarios, like game playing [24], [25], competition between bacteria [26]–[31] and fungi [32], mating choice of lizards [33] and food choice of birds [34], to name just a few. In an informal way, Figure 1 shows with the help of examples what transitivity means for symmetric and reciprocal relations that are crisp and graded.

Despite the occurrence of intransitive relations in many domains, one has to admit that most applications are still characterized by relations that fulfill relatively strong transitivity requirements. For example, in decision making, preference modeling and social choice theory, one can argue that reciprocal relations like choice probabilities and preference judgments should satisfy certain transitivity properties, if they represent rational human decisions made after well-reasoned comparisons on objects [15], [35], [36]. For symmetric relations as well, transitivity plays an important role [37], [38], when modeling similarity relations, metrics, kernels, etc.

It is for this reason that transitivity properties have been studied extensively in fuzzy set theory and related fields.

For reciprocal relations, one traditionally uses the notion of stochastic transitivity [15].

Definition IV.1. *Let g be an increasing $[1/2,1]^2 \to [0,1]$ mapping. A reciprocal relation $Q : \mathcal{V}^2 \to [0,1]$ is called g-stochastic transitive if for any $(v_1, v_2, v_3) \in \mathcal{V}^3$*

$$\big(Q(v_1,v_2) \geq 1/2 \wedge Q(v_2,v_3) \geq 1/2\big)$$
$$\Rightarrow Q(v_1,v_3) \geq g(Q(v_1,v_2), Q(v_2,v_3))\,.$$

Important special cases are weak stochastic transitivity when $g(a,b) = 1/2$, moderate stochastic transitivity when $g(a,b) = \min(a,b)$ and strong stochastic transitivity when $g(a,b) = \max(a,b)$. Alternative (and more general) frameworks are FG-transitivity [39] and cycle transitivity [12], [13]. For graded symmetric relations, the notion of T-transitivity has been put forward [40], [41].

Definition IV.2. *A symmetric relation $Q : \mathcal{V}^2 \to [0,1]$ is called T-transitive with T a t-norm if for any $(v_1, v_2, v_3) \in \mathcal{V}^3$*

$$T(Q(v_1,v_2), Q(v_2,v_3)) \leq Q(v_1,v_3)\,. \tag{9}$$

Three important t-norms are the minimum t-norm $T_{\mathbf{M}}(a,b) = \min(a,b)$, the product t-norm $T_{\mathbf{P}}(a,b) = ab$ and the Łukasiewicz t-norm $T_{\mathbf{L}}(a,b) = \max(a+b-1,0)$.

In addition, several authors have shown that various forms of transitivity give rise to utility representable or numerically representable relations, also called fuzzy weak orders – see e.g. [15], [16], [42]–[44]. We will use the term ranking representability to establish a link with machine learning. We give a slightly specific definition that unifies reciprocal and symmetric relations.

Definition IV.3. *A reciprocal or symmetric relation $Q : \mathcal{V}^2 \to [0,1]$ is called ranking representable if there exists a ranking function $f : \mathcal{V} \to \mathbb{R}$ such that for all $(v,v') \in \mathcal{V}^2$ it respectively holds that*

1) *$Q(v,v') = \nabla(f(v) - f(v'))$ (reciprocal case) ;*
2) *$Q(v,v') = \nabla(f(v) + f(v'))$ (symmetric case) .*

The main idea is that ranking representable relations can be constructed from a utility function f. Ranking representable reciprocal relations correspond to directed acyclic graphs, and a unique ranking of the nodes in such graphs can be obtained with topological sorting algorithms. The ranking representable reciprocal relations of Figures 1 (a) and (e) for example yield the global ranking $A \succ B \succ C$. Interestingly, ranking representability of reciprocal relations and symmetric relations can be easily achieved in our framework by simplifying the joint feature mapping Ψ. Let $\Psi(v,v') = \phi(v)$ such that K^{Φ} simplifies to

$$K^{\Phi}_{fR}(e,\overline{e}) = K^{\phi}(v,\overline{v}) + K^{\phi}(v',\overline{v}') - K^{\phi}(v,\overline{v}') - K^{\phi}(v',\overline{v})$$
$$K^{\Phi}_{fS}(e,\overline{e}) = K^{\phi}(v,\overline{v}) + K^{\phi}(v',\overline{v}') + K^{\phi}(v,\overline{v}') + K^{\phi}(v',\overline{v})$$

when $\Phi(v,v') = \Phi_R(v,v')$ or $\Phi(v,v') = \Phi_S(v,v')$, respectively, then the following proposition holds.

Proposition IV.4. *The relation $Q : \mathcal{V}^2 \to [0,1]$ given by (1) and h defined by (2) with $K^{\Phi} = K^{\Phi}_{fR}$ (respectively $K^{\Phi} = K^{\Phi}_{fS}$) is a ranking representable reciprocal (respectively symmetric) relation.*

The proof directly follows from the fact that for this specific kernel, $h(v,v')$ can be respectively written as $f(v) - f(v')$ and $f(v) + f(v')$. The kernel K^{Φ}_{fR} has been used as a main building block in many kernel-based ranking algorithms (see e.g. [45], [46]). Since ranking representability of reciprocal relations implies strong stochastic transitivity of reciprocal relations, K^{Φ}_{fR} can represent this type of domain knowledge.

The notion of ranking representability is powerful for reciprocal relations, because the majority of reciprocal relations satisfy this property, but for symmetric relations it has a rather limited applicability. Ranking representability as defined above cannot represent relations that originate from an underlying metric or similarity measure. For such relations, one needs another connection with its roots in Euclidean metric spaces [37].

Definition IV.5. *A symmetric relation $Q : \mathcal{V}^2 \to [0,1]$ is called Euclidean representable if there exists a ranking function $f : \mathcal{V} \to \mathbb{R}^z$ such that for all pairs $(v,v') \in \mathcal{V}^2$ it holds that*

$$Q(v,v') = \nabla((f(v) - f(v'))^T (f(v) - f(v')))\,, \tag{10}$$

with a^T the transpose of a vector a.

Euclidean representability can be seen as Euclidean embedding or Multidimensional Scaling in a z-dimensional space [47]. In its most restrictive form, when $z = 1$, it implies that the symmetric relation can be constructed from the Euclidean distance in a one-dimensional space. When such a one-dimensional embedding can be realized, one global ranking of the objects can be found, similar to reciprocal relations. Nevertheless, although models of type (10) with $z = 1$ are sometimes used in graph inference [6] and semi-supervised learning [48], we believe that situations where symmetric relations become Euclidean representable in a one-dimensional space occur very rarely, in contrast to reciprocal relations. The extension to $z > 1$ on the other hand does not guarantee the existence of one global ranking, then Euclidean representability still enforces some interesting properties, because it guarantees that the relation Q is constructed from a Euclidean metric space with a dimension upper bounded by the number of nodes p. Moreover, this type of domain knowledge about relations can be incorporated in our framework. To this end, let $\Phi(v,v') = \Phi_S(v,v')$ and let $\Psi(v,v') = \phi(v) \otimes (\phi(v) - \phi(v'))$ such that K^{Φ} becomes $K^{\Phi}_{\mathrm{MLPK}}(e,\overline{e}) = (K^{\Phi}_{fR}(e,\overline{e}))^2 =$

$$\left(K^{\phi}(v,\overline{v}) + K^{\phi}(v',\overline{v}') - K^{\phi}(v,\overline{v}') - K^{\phi}(v',\overline{v})\right)^2.$$

This kernel has been called the metric learning pairwise kernel by [49]. As a consequence, the vector of parameters $\mathbf{w}$ can be rewritten as an $r \times r$ matrix $\mathbf{W}$ where $\mathbf{W}_{ij}$ corresponds to the parameter associated with $(\phi_i(v) - \phi_i(v'))(\phi_j(v) - \phi_j(v'))$ such that $\mathbf{W}_{ij} = \mathbf{W}_{ji}$.

Proposition IV.6. *If $\mathbf{W}$ is positive semi-definite, then the symmetric relation $Q : \mathcal{V}^2 \to [0,1]$ given by (1) with h defined by (2) and $K^{\Phi} = K^{\Phi}_{\mathrm{MLPK}}$ is an Euclidean representable symmetric relation.*

See the appendix for the proof. Although the model established by K^{Φ}_{MLPK} does not result in a global ranking, this model strongly differs from the one established with $K^{\Phi}_{\otimes S}$, since K^{Φ}_{MLPK} can only represent symmetric relations that exhibit transitivity properties. Therefore, one should use K^{Φ}_{MLPK} when, for example, the underlying relation corresponds to a metric or a similarity relation, while the kernel $K^{\Phi}_{\otimes S}$ should be preferably used for symmetric relations for which no further domain knowledge can be assumed beforehand.

V. Relationships with other machine learning algorithms

As explained in Section 2, the transition from a standard classification or regression setting to learning graded relations should be found in the specification of joint feature mappings over couples of objects, thereby naturally leading to the introduction of specific kernels. Any existing machine learning algorithm for classification or regression can in principle be adopted, if joint feature mappings are constructed explicitly. Since kernel methods avoid this explicit construction, they can often outperform non-kernelized algorithms in terms of computational efficiency [17]. As a second main advantage, kernel methods allow for expressing similarity scores for structured objects, such as strings, graphs and trees and text [50]. In our setting of learning graded relations, this implies that one should plug these domain-specific kernel functions into (6) or use the other pairwise kernels that are discussed in this paper. Such a scenario is in fact common practice in some applications of Kronecker product pairwise kernels, such as predicting protein-ligand compatibility in bioinformatics [51]. String kernels or graph kernels can be defined on various types of biological structures [52] and Kronecker product pairwise kernels then combine these object-based kernels into relation-based kernels (thus, node kernels versus edge kernels).

The edge kernels can be utilized within a wide variety of kernel methods. Since we focus on learning graded relations, one naturally arrives at a regression setting. In the following section, we run some experiments with regularized least-squares methods, which optimize (5) using a hypothesis space induced by kernels. The solution is found by solving a system of linear equations [50], [53].

Apart from kernel methods, we briefly mention a number of other algorithms that are somewhat connected, even though they provide solutions for different learning problems. If pairwise relations are considered between objects of two different domains, one arrives at a learning setting that is referred to as predicting labels for dyadic data [54]. Examples of such settings include link prediction in bipartite graphs and movie recommendation for users. As such, one could also argue that specific link prediction and matrix factorization methods could be applied in our setting as well, see e.g. [55]–[57]. However, these methods have been primarily designed for exploiting relationships in the output space, whereas feature representations of the objects are often not observed or simply irrelevant. Moreover, similar to the Cartesian pairwise kernel, these methods cannot be applied in situations where predictions need to be made for two new nodes that were not present in the training dataset.

TABLE I
Methods considered in the experiments

Abbreviation	Method
MPRED	Predicting the mean
$K^{\Phi}_{\otimes}$	Kronecker Product Pairwise Kernel
$K^{\Phi}_{\otimes S}$	Symmetric Kronecker Product Pairwise Kernel
$K^{\Phi}_{\otimes R}$	Reciprocal Kronecker Product Pairwise Kernel
K^{Φ}_{MLPK}	Metric Learning Pairwise Kernel
K^{Φ}_{C}	Cartesian Product Pairwise Kernel
K^{Φ}_{CS}	Symmetric Cartesian Pairwise Kernel

Another connection can be observed with multivariate regression and structured output prediction methods. Such methods have been occasionally applied in settings where relations had to be learned [9]. Also recall that structured output prediction methods use Kronecker product pairwise kernels on a regular basis to define joint feature representations of inputs and outputs [58], [59].

In addition to predictive models for dyadic data, one can also detect connections with certain information retrieval and pattern matching methods. However, these methods predominantly use similarity as the underlying relation, often in a purely intuitive manner, as a nearest neighbor type of learning, so they can be considered much more restrictive. Consider the example of protein ranking [60] or algorithms like *query by document* [3]. These methods simply look for rankings where the most similar objects w.r.t. the query object appear on top, contrary to our approach, which should be considered as much more general, since we learn rankings from any type of binary relation. Nonetheless, similarity relations will of course still occupy a prominent place in our framework as an important special case.

VI. Experiments

We test the ability of the pairwise kernels to model different types of relations, and the effect of enforcing prior knowledge about the properties of the learned relations. To this end, we train the regularized least-squares (RLS) algorithm to regress the relation values [50], [53]. We perform experiments on both symmetric and reciprocal relations, considering both synthetic and real-world data. In addition to the standard, symmetric and reciprocal Kronecker product pairwise kernels, we also consider the Cartesian kernel, the symmetric Cartesian kernel and the metric learning pairwise kernel. The performance measure used is the mean squared error (MSE). For statistical significance testing, we use the paired Wilcoxon-signed-rank test with significance level 0.05 where applicable. The conservative Bonferroni correction is applied to take into account multiple hypothesis testing, meaning that the required p-value is divided by the number of comparisons done.

A. Synthetic data: learning similarity measures

Experiments on synthetic data were conducted to illustrate the behavior of the different kernels in terms of the transitivity of the relation to be learned. A parametric family of cardinality-based similarity measures for sets was considered as the relation of interest [61]. For two sets A and B, let us

TABLE II
RESULTS FOR THE FIRST SIMILARITY MEASURE EXPERIMENT, THE TASK IS TO PREDICT WEIGHTS FOR UNOBSERVED EDGES IN A PARTIALLY OBSERVED RELATIONAL GRAPH.

Setting	(t, t', u, v)	MPRED	$K^{\Phi}_{\otimes}$	$K^{\Phi}_{\otimes S}$	K^{Φ}_{MLPK}	K^{Φ}_{C}	K^{Φ}_{CS}
Intransitive	(0,1,2,2)	0.01038	0.00908	0.00773	0.00768	0.00989	0.00924
$T_{\mathbf{L}}$-transitive	(0,1,1,0)	0.01514	0.00962	0.00781	0.00805	0.01155	0.00941
$T_{\mathbf{P}}$-transitive	(1,2,1,1)	0.00259	0.00227	0.00192	0.00188	0.00248	0.00231

TABLE III
RESULTS FOR THE SECOND SIMILARITY MEASURE EXPERIMENT, THE TASK IS TO PREDICT RELATIONS BETWEEN PREVIOUSLY UNSEEN NODES.

Setting	(t, t', u, v)	MPRED	$K^{\Phi}_{\otimes}$	$K^{\Phi}_{\otimes S}$	K^{Φ}_{MLPK}
Intransitive	(0,1,2,2)	0.01032	0.00995	0.00936	0.00971
$T_{\mathbf{L}}$-transitive	(0,1,1,0)	0.01515	0.01236	0.01166	0.01453
$T_{\mathbf{P}}$-transitive	(1,2,1,1)	0.00259	0.00251	0.00236	0.00242

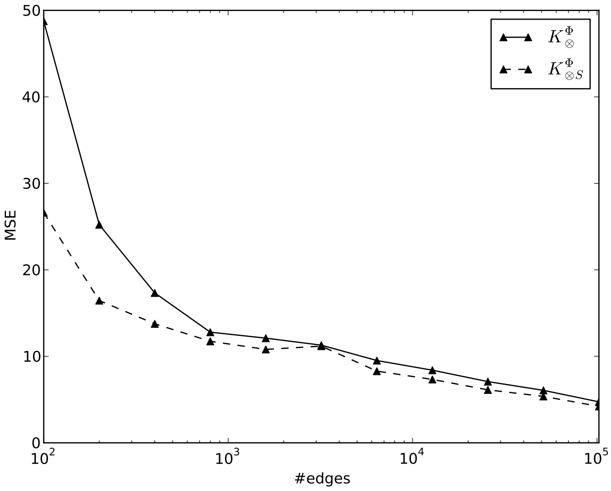


Fig. 2. The comparison of the ordinary Kronecker product pairwise kernel $K^{\Phi}_{\otimes}$ and the symmetric Kronecker product pairwise kernel $K^{\Phi}_{\otimes S}$ on the Newsgroups dataset. The mean squared error is shown as a function of the training set size.

define the following cardinalities:

$$\begin{aligned}\Delta_{A,B} &= |A \setminus B| + |B \setminus A|\,,\\ \delta_{A,B} &= |A \cap B|\,,\\ \nu_{A,B} &= |(A \cup B)^c|\,,\end{aligned}$$

then this family of similarity measures for sets can be expressed as:

$$S(A,B) = \frac{t\Delta_{A,B} + u\delta_{A,B} + v\nu_{A,B}}{t'\Delta_{A,B} + u\delta_{A,B} + v\nu_{A,B}}\,, \tag{11}$$

with t, t', u and v four parameters. This family of similarity measures includes many well-known similarity measures for sets, such as the Jaccard coefficient [62], the simple matching coefficient [63] and the Dice coefficient [64].

Three members of this family are investigated in our experiments. The first one is the Jaccard coefficient, corresponding to $(t,t',u,v) = (0,1,1,0)$. The Jaccard coefficient is known to be $T_{\mathbf{L}}$-transitive. The second member that we investigate was originally proposed by [65]. It corresponds to $(t,t',u,v) = (0,1,2,2)$ and it does not satisfy $T_{\mathbf{L}}$-transitivity, which is considered as a very weak transitivity condition. Conversely, the third member that we analyse has rather strong transitivity properties. It is given by $(t,t',u,v) = (1,2,1,1)$ and it satisfies $T_{\mathbf{P}}$-transitivity.

Features and labels for all three members are generated as follows. First we generate 20-dimensional feature vectors consisting of statistically independent features that follow a Bernoulli distribution with $\pi = 0.5$. Subsequently, the above-mentioned similarity measures are computed for each pair of features, resulting in a deterministic mapping between features and labels. Finally, to introduce some noise in the problem setting, 10% of the features are ad random swapped in a last step from a zero to a one or vice versa.

In the experiments, we always generate three data sets, a training set for building the model, a validation set for hyperparameter selection, and a test set for performance evaluation. We perform two kinds of experiments. In the first experiment, we have a single set of 100 nodes. 500 node pairs are randomly sampled without replacement to the training, validation and test sets. Thus, the learning problem here is, given a subset of the relation values for a fixed set of nodes, to learn to predict missing relation values. This setup allows us for testing also the Cartesian kernel, which is unable to generalize to completely new pairs of nodes. In the second experiment, we generate three separate sets of 100 nodes for the training, validation and test sets, and sample 500 edges from each of these. This experiment allows us for testing the generalization capability of the learned models with respect to new couples of nodes (i.e., previously unseen nodes). Here, the Cartesian kernel is not applicable, and thus not included in the experiment. The experiments are repeated 100 times, the presented results are means over the repetitions. The Gaussian RBF kernel was considered at the node level. For training RLS we solved the corresponding system of linear equations using matrix factorization, by considering an explicit regularization parameter. A grid search is conducted to select the width of the Gaussian RBF kernel and the regularization parameter, both are selected from the range $2^{-20}, \ldots, 2^{1}$.

TABLE IV
RESULTS FOR THE ECOLOGY EXPERIMENT.

Kernel	MPRED	$K^{\Phi}_{\otimes}$	$K^{\Phi}_{\otimes R}$	K^{Φ}_{MLPK}
MSE	0.02795	0.01082	0.01067	0.02877

The results for the experiments are presented in Tables II and III. In both cases all the kernels outperform the mean predictor, meaning that they are able to model the underlying relations. For all the learning methods, the error is lower in the first experiment than in the second one, demonstrating that it is easier to predict relations between known nodes, than to generalize to a new set of nodes. Enforcing symmetry is clearly beneficial, as the symmetric Kronecker product pairwise kernel always outperforms the standard Kronecker product pairwise kernel, and the symmetric Cartesian kernel always outperforms the standard one. Comparing the Kronecker and Cartesian kernels, the Kronecker one leads to clearly lower error rates. With the exception of the $T_{\mathbf{L}}$-transitive case in the second experiment, MLPK turns out to be highly successful in modeling the relations, probably due to enforcing symmetry of the learned relation. In the first experiment, all the differences are statistically significant, apart from the difference between the symmetric Kronecker product pairwise kernel and MLPK for the intransitive case. In the second experiment, all the differences are statistically significant. Clearly, including prior knowledge about symmetry helps to boost the predictive performance in this problem.

B. Learning the similarity between documents

In the second experiment, we compare the ordinary and symmetric Kronecker pairwise kernels on a real-world data set based on newsgroups documents[2]. The data was sampled from 4 newsgroups: rec.autos, rec.sport.baseball, comp.sys.ibm.pc.hardware and comp.windows.x. The aim is to learn to predict the similarity of two documents as measured by the number of common words they share. The node features correspond to the number of occurrences of a word in a document. The feature representation is high-dimensional and sparse, with more than 50000 possible features, the majority of which are zero for any given document. First, we sample separate training, validation and test sets each consisting of 1000 nodes. Second, we sample edges connecting the nodes in the training and validation set using exponentially growing sample sizes to measure the effect of sample size on the differences between the kernels. The sample size grid is $[100, 200, 400, \ldots, 102400]$. Again, we sample only edges with different starting and end nodes. When computing the test performance, we consider all the edges in the test set, except those starting and ending at the same node. The linear kernel is used at the node level. We train the RLS algorithm using conjugate gradient optimization with early stopping [66], optimization is terminated once the MSE on the validation set has failed to decrease for 10 consecutive iterations. Since we rely on the regularizing effect of early stopping, a separate regularization parameter is not needed. We do not include other types of kernels besides the Kronecker product pairwise kernels in the experiment. To the best of our knowledge, no algorithms that scale to the considered experiment size exist for the other kernel functions. Hence, this experiment mainly aims to illustrate the computational advantages of the Kronecker product pairwise kernel. The mean as prediction achieves an MSE around 145 on this data.

The results are presented in Figure 2. Even for 100 pairs the errors for both kernels are much lower than for mean predictor, showing that RLS succeeds with both kernels in learning the underlying relation. Increasing the training set size leads to a decrease in test error. Using the prior knowledge about the symmetry of the learned relation is clearly helpful. The symmetric kernel achieves a lower error than the ordinary Kronecker product pairwise kernel for all sample sizes, the largest differences are observed for the smallest sample sizes. For 100 training instances, the error is almost halved by enforcing symmetry.

C. Competition between species

Finally, we compare ordinary and reciprocal Kronecker pairwise kernels and the metric learning pairwise kernel on simulated data from an ecological model. The setup is based on the one described in [67]. This model provides an elegant explanation for the coexistence of multiple species in the same habitat, a problem that has puzzled ecologists for decades [68].

Imagine n species sharing a habitat and struggling for their share of the resources. One species can dominate another species based on k so-called limiting factors. A limiting factor defines an attribute that can give a fitness advantage, for example in plants the ability to photosynthesize or to draw minerals from the soil, resistance to diseases, etc. Each species is scored on each of its k limiting factors. The degree to which one species can dominate a competitor is relative to the number of limiting factors for which it is superior. All possible interactions can thus be represented in a tournament. In this framework relations are reciprocal and often intransitive.

For this simulation 400 species were simulated with 10 limiting factors. The value of each limiting factor is for each species drawn from a random uniform distribution between 0 and 1. Thus, any species v can be represented by a vector $\mathbf{f}$ of length k with the limiting factors as elements. The probability that a species v dominates species v' can easily be calculated as: $Q(v, v') = \frac{1}{k}\sum_{i=1}^{k} H(f_i - f_i')$, where $H(x)$ is the Heaviside step function.

Of the 400 species, 200, 100 and 100 were used for generating training, validation and testing data. For each subset, the complete tournament matrix was determined. From those matrices 1200 interactions were sampled for training, 600 for model validation and 600 for testing. No combination of species was used more than once. Using the limiting factors as features, we try to regress the probability that one species dominates another one using the ordinary and reciprocal Kronecker product pairwise kernels and the metric learning pairwise kernel. Gaussian kernel is applied as the node kernel. The validation set is used to determine the optimal regularization parameter and kernel width parameter from the grids 2^{-20}, 2^{-19} $\ldots$, 2^{4} and 2^{-10}, 2^{-9} $\ldots$, 2^{1}. To obtain statistically significant results the setup is repeated 100 times.

The results are shown in Table IV. The metric learning pairwise kernel gives rise to worse predictions than the mean as prediction. This is not surprising, as the MLPK cannot learn reciprocal relations. The ordinary Kronecker product pairwise kernel performs well and the reciprocal Kronecker product pairwise kernel performs even better. All the differences are statistically significant. The results show that using prior information on the types of relations to be learned can boost predictive accuracy.

VII. Conclusions

A general kernel-based framework for learning various types of graded relations was presented in this article. This framework extends existing approaches for learning relations, because it can handle crisp and graded relations. A Kronecker product feature mapping was proposed for combining the

[2] Available at: http://people.csail.mit.edu/jrennie/20Newsgroups/

features of pairs of objects that constitute a relation (edge level in a graph), and it was shown that this mapping leads to a class of universal approximators, if an appropriate kernel is chosen on the object level (node level in a graph).

In addition, we clarified that domain knowledge about the relation to be learned can be easily incorporated in our framework, such as reciprocity and symmetry properties. Experimental results on synthetic and real-world data clearly demonstrate that this domain knowledge can significantly improve generalization performance. Important links with recent developments in fuzzy set theory and decision theory can be established, by considering transitivity properties of relations.

Acknowledgments

W.W. is supported as a postdoc by the Research Foundation of Flanders (FWO Vlaanderen) and T.P. by the Academy of Finland (grant 134020).

Appendix

Definition VII.1 (Steinwart [19])**.** *A continuous kernel K on a compact metric space $\mathcal{V}$ (i.e. $\mathcal{V}$ is closed and bounded) is called universal if the RKHS induced by K is dense in $C(\mathcal{V})$, where $C(\mathcal{V})$ is the space of all continuous functions $f : \mathcal{V} \to \mathbb{R}$. That is, for every function $f \in C(\mathcal{V})$ and every $\epsilon > 0$, there exists a set of input points $\{v_i\}_{i=1}^m \in \mathcal{V}$ and real numbers $\{\alpha_i\}_{i=1}^m$, with $m \in \mathbb{N}$, such that*

$$\max_{x\in\mathcal{V}} \left\{ \left| f(v) - \sum_{i=1}^m \alpha_i K(v_i, v) \right| \right\} \le \epsilon.$$

Accordingly, the hypothesis space induced by the kernel K can approximate any function in $C(\mathcal{V})$ arbitrarily well, and hence it has the universal approximating property.

The following result is in the literature known as the Stone-Weierstraß theorem (see e.g [20]):

Theorem VII.2 (Stone-Weierstraß)**.** *Let $C(\mathcal{V})$ be the set of real-valued continuous functions on a compact metric space $\mathcal{V}$. If $\mathcal{A} \subset C(\mathcal{V})$ is a subalgebra of $C(\mathcal{V})$, that is,*

$$\forall f(v), g(v) \in \mathcal{A}, r \in \mathbb{R} : f(v) + rg(v) \in \mathcal{A}, f(v)g(v) \in \mathcal{A}$$

and $\mathcal{A}$ separates points in $\mathcal{V}$, that is,

$$\forall v, v' \in \mathcal{V}, v \neq v' : \exists g \in \mathcal{A} : g(v) \neq g(v'),$$

and $\mathcal{A}$ does not vanish at any point in $\mathcal{V}$, that is,

$$\forall v \in \mathcal{V} : \exists g \in \mathcal{A} : g(v) \neq 0,$$

then $\mathcal{A}$ is dense in $C(\mathcal{V})$.

A. Proofs

Proof: (**Theorem II.1**) Let us define

$$\mathcal{A} \otimes \mathcal{A} = \{t \mid t(v, v') = g(v)u(v'), g, u \in \mathcal{A}\} \tag{12}$$

for a compact metric space $\mathcal{V}$ and a set of functions $\mathcal{A} \subset C(\mathcal{V})$. We observe that the RKHS of the kernel $K_\otimes^\Phi$ can be written as $\mathcal{H} \otimes \mathcal{H}$, where $\mathcal{H}$ is the RKHS of the kernel K^ϕ.

Let $\epsilon > 0$ and let $t \in C(\mathcal{V}) \otimes C(\mathcal{V})$ be an arbitrary function which can, according to (12), be written as $t(v, v') = g(v)u(v')$, where $g, u \in C(\mathcal{V})$. By definition of the universality property, $\mathcal{H}$ is dense in $C(\mathcal{V})$. Therefore, $\mathcal{H}$ contains functions $\overline{g}, \overline{u}$ such that

$$\max_{v\in\mathcal{V}} \{|\overline{g}(v) - g(v)|\} \le \overline{\epsilon},\ \max_{v\in\mathcal{V}} \{|\overline{u}(v) - u(v)|\} \le \overline{\epsilon}\,,$$

where $\overline{\epsilon}$ is a constant for which it holds that

$$\max_{v,v'\in\mathcal{V}} \left\{ |\overline{\epsilon}\, g(v)| + |\overline{\epsilon}\, u(v')| + \overline{\epsilon}^2 \right\} \le \epsilon\,.$$

Note that, according to the extreme value theorem, the maximum exists due to the compactness of $\mathcal{V}$ and the continuity of the functions g and u. Now we have

$$\begin{aligned} &\max_{v,v'\in\mathcal{V}} \{|t(v, v') - \overline{g}(v)\overline{u}(v')|\} \\ &\le \max_{v,v'\in\mathcal{V}} \left\{ |t(v, v') - g(v)u(v')| + |\overline{\epsilon}\, g(v)| + |\overline{\epsilon}\, u(v')| + \overline{\epsilon}^2 \right\} \\ &= \max_{v,v'\in\mathcal{V}} \left\{ |\overline{\epsilon}\, g(v)| + |\overline{\epsilon}\, u(v')| + \overline{\epsilon}^2 \right\} \le \epsilon, \end{aligned}$$

which confirms the density of $\mathcal{H} \otimes \mathcal{H}$ in $C(\mathcal{V}) \otimes C(\mathcal{V})$.

According to Tychonoff's theorem, $\mathcal{V}^2$ is compact if $\mathcal{V}$ is compact. It is straightforward to see that $C(\mathcal{V}) \otimes C(\mathcal{V})$ is a subalgebra of $C(\mathcal{V}^2)$, it separates points in $\mathcal{V}^2$, it vanishes at no point of $C(\mathcal{V}^2)$, and it is therefore dense in $C(\mathcal{V}^2)$ due to Theorem VII.2. Consequently, $\mathcal{H} \otimes \mathcal{H}$ is also dense in $C(\mathcal{V}^2)$, and $K_\otimes^\Phi$ is a universal kernel on $\mathcal{V}^2$. ■

Proof: (**Theorem III.4**) Let $\epsilon > 0$ and $t \in R(\mathcal{V}^2)$ be an arbitrary function. According to Theorem II.1, the RKHS of the kernel $K_\otimes^\Phi$ defined in (6) is dense in $C(\mathcal{V}^2)$. Therefore, we can select a set of edges and real numbers $\{\alpha_i\}_{i=1}^m$, such that the function

$$u(v, v') = \sum_{i=1}^m \alpha_i K^\phi(v, v_i) K^\phi(v', v_i')$$

belonging to the RKHS of the kernel (6) fulfills

$$\max_{(v,v')\in\mathcal{V}^2} \{|t(v, v') - 4u(v, v')|\} \le \frac{1}{2}\epsilon\,. \tag{13}$$

We observe that, because $t(v, v') = -t(v', v)$, the function u also fulfills

$$\max_{(v,v')\in\mathcal{V}^2} \{|t(v, v') + 4u(v', v)|\} \le \frac{1}{2}\epsilon$$

and hence

$$\max_{(v,v')\in\mathcal{V}^2} \{|4u(v, v') + 4u(v', v)|\} \le \epsilon\,. \tag{14}$$

Let $\gamma(v, v') = 2u(v, v') + 2u(v', v)$. Due to (14), we have

$$|\gamma(v, v')| \le \frac{1}{2}\epsilon, \qquad \forall (v, v') \in \mathcal{V}^2\,. \tag{15}$$

Now, let us consider the function $h(v, v') =$

$$\sum_{i=1}^m \alpha_i 2 \left(K^\phi(v, v_i) K^\phi(v', v_i') - K^\phi(v', v_i) K^\phi(v, v_i') \right)\,,$$

which is obtained from u by replacing kernel (6) with kernel

(7). We observe that

$$\begin{aligned} h(v,v') &= 2u(v,v') - 2u(v',v) \\ &= 4u(v,v') - \gamma(v,v'). \end{aligned} \tag{16}$$

By combining (13), (15) and (16), we observe that the function h fulfills (8). ■

Proof: (**Proposition IV.6**) The model that we consider can be written as:

$$Q(v,v') = \nabla\left((\phi(v) - \phi(v'))^T \mathbf{W}(\phi(v) - \phi(v'))\right).$$

The connection with (10) then immediately follows by decomposing $\mathbf{W}$ as $\mathbf{W} = \mathbf{U}^T\mathbf{U}$ with $\mathbf{U}$ an arbitrary matrix. The specific case of $z = 1$ is obtained when $\mathbf{U}$ can be written as a single-row matrix. ■

References

[1] M. Bowling, J. Fürnkranz, T. Graepel, and R. Musick, "Machine learning and games," *Machine Learning*, vol. 63, no. 3, pp. 211–215, 2006.

[2] Y. Yamanishi, J. Vert, and M. Kanehisa, "Protein network inference from multiple genomic data: a supervised approach," *Bioinformatics*, vol. 20, pp. 1363–1370, 2004.

[3] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias, "Query by document," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain*, 2009, pp. 34–43.

[4] B. Taskar, M. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *Advances in Neural Information Processing Systems*, 2004.

[5] L. De Raedt, *Logical and Relational Learning*. Springer, 2009.

[6] J.-P. Vert and Y. Yamanishi, "Supervised graph inference," in *Advances in Neural Information Processing Systems*, vol. 17, 2005.

[7] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side information," in *Advances in Neural Information Processing Systems*, vol. 16, 2002, pp. 521–528.

[8] E. Hüllermeier and J. Fürnkranz, *Preference Learning*. Springer, 2010.

[9] P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc, "Inferring biological networks with output kernel trees," *BMC Bioinformatics*, vol. 8, no. 2, p. S4, 2007.

[10] J. Doignon, B. Monjardet, M. Roubens, and P. Vincke, "Biorder families, valued relations and preference modelling," *Journal of Mathematical Psychology*, vol. 30, pp. 435–480, 1986.

[11] Z. Switalski, "Transitivity of fuzzy preference relations - an empirical study," *Fuzzy Sets and Systems*, vol. 118, pp. 503–508, 2000.

[12] B. De Baets and H. De Meyer, "Transitivity frameworks for reciprocal relations: cycle-transitivity versus FG-transitivity," *Fuzzy Sets and Systems*, vol. 152, pp. 249–270, 2005.

[13] B. De Baets, H. De Meyer, B. De Schuymer, and S. Jenei, "Cyclic evaluation of transitivity of reciprocal relations," *Social Choice and Welfare*, vol. 26, pp. 217–238, 2006.

[14] S. Diaz, S. Montes, and B. De Baets, "Transitivity bounds in additive fuzzy preference structures," *IEEE Transactions on Fuzzy Systems*, vol. 15, pp. 275–286, 2007.

[15] R. Luce and P. Suppes, *Handbook of Mathematical Psychology*. Wiley, 1965, ch. Preference, Utility and Subjective Probability, pp. 249–410.

[16] U. Bodenhofer, B. De Baets, and J. Fodor, "A compendium of fuzzy weak orders," *Fuzzy Sets and Systems*, vol. 158, pp. 811–829, 2007.

[17] B. Schölkopf and A. Smola, *Learning with Kernels, Support Vector Machines, Regularisation, Optimization and Beyond*. The MIT Press, 2002.

[18] A. Ben-Hur and W. Noble, "Kernel methods for predicting protein-protein interactions," *Bioinformatics*, vol. 21 Suppl 1, pp. 38–46, 2005.

[19] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 67–93, 2002.

[20] W. Rudin, *Functional Analysis*, 2nd ed., ser. International Series in Pure and Applied Mathematics. New York: McGraw-Hill Inc., 1991.

[21] H. Kashima, S. Oyama, Y. Yamanishi, and K. Tsuda, "On pairwise kernels: An efficient alternative and generalization analysis." in *PAKDD*, ser. Lecture Notes in Computer Science, T. Theeramunkong, B. Kijsirikul, N. Cercone, and T. B. Ho, Eds., vol. 5476. Springer, 2009, pp. 1030–1037. [Online]. Available: http://dblp.uni-trier.de/db/conf/pakdd/pakdd2009.html#KashimaOYT09

[22] T. Pahikkala, W. Waegeman, E. Tsivtsivadze, T. Salakoski, and B. De Baets, "Learning intransitive reciprocal relations with kernel methods," *European Journal of Operational Research*, vol. 206, pp. 676–685, 2010.

[23] M. Hue and J. Vert, "On learning with kernels for unordered pairs," in *Proceedings of the 27th International Conference on Machine Learning, p.463-470, 2010*, 2010.

[24] B. De Schuymer, H. De Meyer, B. De Baets, and S. Jenei, "On the cycle-transitivity of the dice model," *Theory and Decision*, vol. 54, pp. 261–285, 2003.

[25] L. Fisher, *Rock, Paper, Scissors: Game Theory in Everyday Life*. Basic Books, 2008.

[26] B. Kerr, M. Riley, M. Feldman, and B. Bohannan, "Local dispersal promotes biodiversity in a real-life game of rock paper scissors," *Nature*, vol. 418, pp. 171–174, 2002.

[27] T. Czárán, R. Hoekstra, and L. Pagie, "Chemical warfare between microbes promotes biodiversity," *Proceedings of the National Academy of Sciences*, vol. 99, no. 2, pp. 786–790, 2002.

[28] M. Nowak, "Biodiversity: Bacterial game dynamics," *Nature*, vol. 418, pp. 138–139, 2002.

[29] B. Kirkup and M. Riley, "Antibiotic-mediated antagonism leads to a bacterial game of rock-paper-scissors in vivo." *Nature*, vol. 428, pp. 412–414, 2004.

[30] G. Károlyi, Z. Neufeld, and I. Scheuring, "Rock-scissors-paper game in a chaotic flow: The effect of dispersion on the cyclic competition of microorganisms," *Journal of Theoretical Biology*, vol. 236, no. 1, pp. 12–20, 2005.

[31] T. Reichenbach, M. Mobilia, and E. Frey, "Mobility promotes and jeopardizes biodiversity in rock-paper-scissors games," *Nature*, vol. 448, pp. 1046–1049, 2007.

[32] L. Boddy, "Interspecific combative interactions between wood-decaying basidiomycetes," *FEMS Microbiology Ecology*, vol. 31, pp. 185–194, 2000.

[33] S. Sinervo and C. Lively, "The rock-paper-scissors game and the evolution of alternative mate strategies," *Nature*, vol. 340, pp. 240–246, 1996.

[34] T. Waite, "Intransitive preferences in hoarding gray jays (*Perisoreus canadensis*)," *Journal of Behavioural Ecology and Sociobiology*, vol. 50, pp. 116–121, 2001.

[35] P. Fishburn, "Nontransitive preferences in decision theory," *Journal of Risk and Uncertainty*, vol. 4, pp. 113–134, 1991.

[36] A. Tversky, *Preference, Belief and Similarity*, E. Shafir, Ed. MIT Press, 1998.

[37] J. Gower and P. Legendre, "Metric and Euclidean properties of dissimilarity coefficients," *Journal of Classification*, vol. 3, pp. 5–48, 1986.

[38] F. Jäkel, B. Schölkopf, and F. Wichmann, "Similarity, kernels, and the triangle inequality," *Journal of Mathematical Psychology*, vol. 52, no. 2, pp. 297–303, 2008.

[39] Z. Switalski, "General transitivity conditions for fuzzy reciprocal preference matrices," *Fuzzy Sets and Systems*, vol. 137, pp. 85–100, 2003.

[40] B. De Baets and R. Mesiar, "Metrics and T-equalities," *Journal of Mathematical Analysis and Applications*, vol. 267, pp. 531–547, 2002.

[41] B. Moser, "On representing and generating kernels by fuzzy equivalence relations," *Journal of Machine Learning Research*, vol. 7, pp. 2603–2620, 2006.

[42] A. Billot, "An existence theorem for fuzzy utility functions: A new elementary proof," *Fuzzy Sets and Systems*, vol. 74, pp. 271–276, 1995.

[43] M. Koppen, "Random utility representation of binary choice probabilities: Critical graphs yielding critical necessary conditions," *Journal of Mathematical Psychology*, vol. 39, pp. 21–39, 1995.

[44] L. Fono and N. Andjiga, "Utility function of fuzzy preferences on a countable set under max-*-transitivity," *Social Choice and Welfare*, vol. 28, pp. 667–683, 2007.

[45] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. MIT Press, 2000, pp. 115–132.

[46] T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Järvinen, and J. Boberg, "An efficient algorithm for learning to rank from preference graphs," *Machine Learning*, vol. 75, no. 1, pp. 129–165, 2009.

[47] Z. Zhang, "Learning metrics via discriminant kernels and multidimensional scaling: Toward expected Euclidean representation," in *Proceedings of the Twentieth International Conference on Machine Learning, Washington D.C., USA*, 2003, pp. 872–879.

[48] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: a geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[49] J. Vert, J. Qiu, and W. S. Noble, “A new pairwise kernel for biological network inference with support vector machines,” *BMC Bioinformatics*, vol. 8 (Suppl 10), p. S8, 2007.

[50] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[51] L. Jacob and J. Vert, “Protein-ligand interaction prediction: an improved chemogenomics approach”, bioinformatics, 24(19):2149-2156, 2008,” *Bioinformatics*, vol. 241, pp. 2149–2156, 2008.

[52] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt, “Graph kernels,” *Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[53] C. Saunders, A. Gammerman, and V. Vovk, “Ridge regression learning algorithm in dual variables,” in *Proceedings of the International Conference on Machine Learning*, 1998, pp. 515–521.

[54] A. Menon and C. Elkan, “Predicting labels for dyadic data,” *Data Mining and Knowledge Discovery*, vol. 21, pp. 327–343, 2010.

[55] N. Srebro, J. Rennie, and T. Jaakkola, “Maxximum margin matrix factorization,” *Advances in Neural Processing Systems*, vol. 17, 2005.

[56] K. Miller, T. Griffiths, and M. Jordan, “Nonparametric latent feature models for link prediction,” *Advances in Neural Processing Systems*, vol. 22, pp. 1276–1284, 2009.

[57] N. Lawrence and R. Urtasan, “Nonlinear matrix factorization with gaussian processes,” in *Proceedings of the International Conference on Machine Learning*, 2009, pp. 601–608.

[58] Y. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and independent output variables,” *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[59] J. Weston, B. Schölkopf, O. Bousquet, T. Mann, and W. Noble, *Predicting structured data*. MIT Press, 2007, ch. Joint kernel maps, pp. 67–83.

[60] J. Weston, A. Eliseeff, D. Zhou, C. Leslie, and W. S. Noble, “Protein ranking: from local to global structure in the protein similarity networks,” *Proceedings of the National Academy of Science*, vol. 101, pp. 6559–6563, 2004.

[61] B. De Baets, H. De Meyer, and H. Naessens, “A class of rational cardinality-based similarity measures,” *J. Comput. Appl. Math.*, vol. 132, pp. 51–69, 2001.

[62] P. Jaccard, “Nouvelle recherches sur la distribution florale,” *Bulletin de la Sociétée Vaudoise de Sciences Naturelles*, vol. 44, pp. 223–270, 1908.

[63] R. Sokal and C. Michener, “A statistical method for evaluating systematic relationships,” *Univ. of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.

[64] L. Dice, “Measures of the amount of ecologic associations between species,” *Ecology*, vol. 26, pp. 297–302, 1945.

[65] R. Sokal and P. Sneath, *Principles of Numerical Taxonomy*. W. H. Freeman, 1963.

[66] T. Pahikkala, W. Waegeman, A. Airola, T. Salakoski, and B. De Baets, “Conditional ranking on relational data,” in *Proceedings of the European Conference on Machine Learning*, ser. Lecture Notes in Computer Science, J. Balczar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Springer Berlin / Heidelberg, 2010, vol. 6322, pp. 499–514.

[67] S. Allesina and J. M. Levine, “A competitive network theory of species diversity,” *Proceedings of the National Academy of Sciences*, vol. 108, pp. 5638–5642, 2011.

[68] G. E. Hutchinson, “The paradox of the plankton,” *The American Naturalist*, vol. 95, no. 882, pp. 137–145, 1961.

Willem Waegeman holds a M.Sc. degree in Computer Science (2004) and a Ph.D. in Engineering (2008) from Ghent University. Currently he is affiliated with the Department of Mathematical Modeling, Statistics and Bioinformatics at Ghent University and he holds a research grant from the Research Foundation of Flanders. In the past he has been a visiting researcher at the University of Marburg (Germany) and the University of Turku (Finland). His research interests include theoretical research in machine learning, data mining and statistics, as well as applications of these disciplines in the life sciences.

Tapio Pahikkala received his Bachelors, Masters, and PhD degrees from University of Turku, Finland, in 2002, 2003, and 2008, respectively. He is an Adjunct Professor of Computer Sciences with the Department of Information Technology, University of Turku, and currently holds a three-year postdoctoral research grant from the Academy of Finland. His research focuses on machine learning, pattern recognition, algorithmics, and computational intelligence. He has authored more than sixty peer reviewed scientific publications and served in program committees of numerous scientific conferences. Dr. Pahikkala also serves as a member on the IEEE Computational Intelligence Society (CIS) Social Media Subcommittee and the CIS GOLD Subcommittee.

Antti Airola received his M.Sc. in software engineering and D.Sc. (Tech.) in information and communication technology from University of Turku, Turku, Finland in 2006 and 2011. Currently he works as a postdoctoral researcher at the University of Turku. His research interests include both basic and applied research in machine learning, with special focus on regularized kernel methods and their applications in life sciences.

Tapio Salakoski is a Professor of Computer Science and the Head of Department of Information Technology at the University of Turku, Finland, and a Chairman of the Board for Turku Centre for Computer Science. His research interests are in machine learning and language technology in the BioHealth domain, especially bioinformatics and bioNLP, as well as in educational informatics, especially technology-enhanced learning of mathematics and programming. He is the head of Bioinformatics Laboratory, running an International Master’s Program in Bioinformatics. He has over 170 scientific publications in international journals and conference proceedings. He has organized and chaired international conferences and serves in editorial boards of scientific journals and program committees of conferences. He has lead many research projects and held numerous positions of trust involving both academia and industry.

Michiel Stock is a last-year M.Sc.-student in Bio-science Engineering at Ghent University, with a minor in computational biology.

Bernard De Baets (1966) holds an M.Sc. in Maths (1988), a Postgraduate degree in Knowledge Technology (1991) and a Ph.D. in Maths (1995), all *summa cum laude* from Ghent University (Belgium) and is a Government of Canada Award holder (1988). He is a Full Professor in Applied Maths (1999) at Ghent University, where he is leading KERMIT, the research unit *Knowledge-Based Systems*. He is an Honorary Professor of Budapest Tech (2006) and an IFSA Fellow (2011). His publications comprise more than 250 papers in international journals and about 50 book chapters. He serves on the Editorial Boards of various international journals, in particular as co-editor-in-chief of Fuzzy Sets and Systems. B. De Baets coordinates EUROFUSE, the EURO Working Group on Fuzzy Sets, and is a member of the Board of Directors of EUSFLAT and of the Administrative Board of the Belgian OR Society.