TUCS

Evgeni Tsivtsivadze

Learning Preferences with Kernel-Based Methods

Turku Centre *for* Computer Science

TUCS Dissertations
No 114, Apri 2009

Learning Preferences with Kernel-Based Methods

Evgeni Tsivtsivadze

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism on April 21, 2009, at 12 o'clock.

Turku Centre for Computer Science
University of Turku
Department of Information Technology
Joukahaisenkatu 3-5 B, FIN-20520, Turku
Finland

2009

Supervisors

Dr Jorma Boberg
Department of Information Technology
University of Turku
Finland

Professor Tapio Salakoski
Department of Information Technology
University of Turku
Finland

Reviewers

Dr Kai Yu
NEC Laboratories America
USA

Dr Wei Chu
Yahoo! Inc.
USA

Opponent

Professor Tom Heskes
Faculty of Science
University of Nijmegen
The Netherlands

ISBN 978-952-12-2277-1
ISSN 1239-1883

Abstract

Learning of preference relations has recently received significant attention in machine learning community. It is closely related to the classification and regression analysis and can be reduced to these tasks. However, preference learning involves prediction of ordering of the data points rather than prediction of a single numerical value as in case of regression or a class label as in case of classification. Therefore, studying preference relations within a separate framework facilitates not only better theoretical understanding of the problem, but also motivates development of the efficient algorithms for the task. Preference learning has many applications in domains such as information retrieval, bioinformatics, natural language processing, etc. For example, algorithms that learn to rank are frequently used in search engines for ordering documents retrieved by the query. Preference learning methods have been also applied to collaborative filtering problems for predicting individual customer choices from the vast amount of user generated feedback.

In this thesis we propose several algorithms for learning preference relations. These algorithms stem from well founded and robust class of regularized least-squares methods and have many attractive computational properties. In order to improve the performance of our methods, we introduce several non-linear kernel functions. Thus, contribution of this thesis is twofold: *kernel functions for structured data* that are used to take advantage of various non-vectorial data representations and the *preference learning algorithms* that are suitable for different tasks, namely efficient learning of preference relations, learning with large amount of training data, and semi-supervised preference learning. Proposed kernel-based algorithms and kernels are applied to the parse ranking task in natural language processing, document ranking in information retrieval, and remote homology detection in bioinformatics domain. Training of kernel-based ranking algorithms can be infeasible when the size of the training set is large. This problem is addressed by proposing a preference learning algorithm whose computation complexity scales linearly with the number of training data points. We also introduce sparse approximation of the algorithm that can be efficiently trained with large amount of data. For situations when small amount of labeled data but a large amount of unlabeled data is available, we propose a co-regularized preference learning algorithm.

To conclude, the methods presented in this thesis address not only the problem of the efficient training of the algorithms but also fast regularization parameter selection, multiple output prediction, and cross-validation. Furthermore, proposed algorithms lead to notably better performance in many preference learning tasks considered.

Acknowledgements

During my Ph.D years I had a privilege to work at the Turku Centre for Computer Science (TUCS) and this dissertation would not be possible without guidance and help I have received from my supervisors, colleagues, and friends whom I would like to warmly acknowledge here. First of all, I would like to thank my supervisors, Jorma Boberg, Tapio Salakoski, and Aleksandr Mylläri. From the start Jorma directed my research by sharing his perspectives on variety of machine learning problems that interested me a lot. I would like to thank Jorma for his support throughout my research and thesis-writing period as well as for his sound advice, teaching, and great company. I would like to thank Tapio for his supervision and encouragement in addressing challenging research questions. He has been the supervisor who also guided me as a group member of our lab, strengthening collaboration with other researchers working at TUCS. I would like to acknowledge Aleksandr for his supervision and support, particularly during the first years of my stay at TUCS. He has always had a good advice for me, would it be scientific or practical question. Many thanks to Jouni Järvinen who has always been able to make our long manuscripts much shorter with his sharp comments.

I would like to acknowledge the thesis reviewers, Wei Chu and Kai Yu whose suggestions increased readability and reduced ambiguity of the introductory part of the thesis. I would also like to thank Tom Heskes who kindly agreed to act as an opponent at the defense.

While working at TUCS I had a chance to learn from and to collaborate with many researchers: Filip Ginter, Tapio Pahikkala, Sampo Pyysalo, Hanna Suominen, Marketta Hiissa, Antti Airola and others. I am grateful to Tapio for inspiring discussions and fruitful collaboration. Thanks to Filip and Sampo who have been great source of insight on many problems ranging from natural language processing to machine learning. I would also like to thank Hanna and Antti who have been always prolific with new, cool ideas during our joint project. It has been a pleasure to work with all of you guys. Special thanks to Ion Petre and Vladimir Rogojin for organizing interesting seminars on biocomputing and bioinfomatics that were fun to attend. I would also like to acknowledge my collaborators and co-authors outside of

TUCS, Fabian Gieseke and Willem Waegeman. It was great to visit you in Dortmund and Ghent and to host you in Turku.

My particular gratitude goes to the administrative stuff of TUCS, Irmeli Laine, Tomi Mäntylä, Christel Donner, Satu Jääskeläinen, Ulrika Gustafsson, Monika Eklund, and others. Travel arrangements, organizing various events, etc. would have been impossible to accomplish without your help. There is your large contribution for making TUCS such a great place for conducting research.

The list of people who indirectly contributed to this thesis would not be complete without mentioning professors who supervised my undergraduate studies, Mzia Chogovadze and, later on, Levan Chkhaidze. They have influenced me with their ideas, working style, and personal charisma.

I am indebted to family for their support throughout these years. I would like thank my mother Natalia for her incredible friendship and continuous support. I thank my girlfriend Migena for her encouragement and invaluable help throughout this work. Finally, I am grateful to my grandfather Shota for being an inspiration and example of how to do things right. This thesis is dedicated to him.

To my grandfather

List of original publications included in the thesis

I Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T. (2008). Locality kernels for sequential data and their applications to parse ranking. *Applied Intelligence* (Online First).

II Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T. (2008). Kernels for text analysis. *Book chapter in Advances of Computational Intelligence in Industrial Systems* 116: 81–97.

III Tsivtsivadze, E., Boberg, J., Salakoski, T. (2007). Locality kernels for protein classification. In Giancarlo, R., Hannenhalli, S., eds., *Proceedings of the 7th International Workshop on Algorithms in Bioinformatics*, pages 2–11, Philadelphia, USA. Springer.

IV Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., Boberg, J. (2009). An efficient algorithm for learning to rank from preference graphs. Machine Learning, volume 75(1), pages 129–165.

V Tsivtsivadze, E., Pahikkala, T., Airola, A., Boberg, J., Salakoski, T. (2008). A sparse regularized least-squares preference learning algorithm. In Holst, A., Kreuger, P., Funk, P., eds., *Proceedings of the 10th Scandinavian Conference on Artificial Intelligence*, volume 173, pages 76–83, IOS Press.

VI Tsivtsivadze, E., Gieseke, F., Pahikkala, T., Boberg, J., Salakoski, T. (2008). Learning preferences with co-regularized least squares. In Hüllermeier, E., Fürnkranz, J., eds., *Proceedings of the ECML/PKDD Workshop on Preference Learning*, pages 52–66, Antwerp, Belgium.

List of related publications not included in the thesis

1. Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T. (2006). Locality-convolution kernel and its application to dependency parse ranking. In Ali, M., Dapoigny, R., eds., *Proceeedings of the IEA/AIE'06, Lecture Notes in Computer Science*, volume 4031, pages 610–618. Springer.

2. Tsivtsivadze, E., Pahikkala, T., Pyysalo, S., Boberg, J., Mylläri, A., and Salakoski, T. (2005). Regularized least-squares for parse ranking. In Famili, A. F., Kok, J. N., Peña, J. M., Siebes, A., and Feelders, A. J., eds., *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, pages 464–474. Springer.

3. Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., and Salakoski, T. (2008). Regularized least-squares for learning non-transitive preferences between strategies. In Raiko, T., Haikonen, P., and Vyrynen, J., eds., *Proceedings of the 13th Finnish Artificial Intelligence Conference.*

4. Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., and Salakoski, T. (2007). Learning to rank with pairwise regularized least-squares. In Joachims, T., Li, H., Liu, T.Y., and Zhai, C., eds., *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, pages 27–33.

5. Pahikkala, T., Tsivtsivadze, E., Boberg, J., and Salakoski, T. (2006). Graph kernels versus graph representations: a case study in parse ranking. In Gärtner, T., Garriga, G.C., Meinl, T., eds., *Proceedings of the ECML/PKDD'06 Workshop on Mining and Learning with Graphs.*

6. Ginter, F., Pahikkala, T., Pyysalo, S., Tsivtsivadze, E., Boberg, J., Järvinen, J., Mylläri, A., Salakoski, T. (2005). Information Extraction from Biomedical Text: The BioText Project. In Langemets, M., Penjam, P., eds., *HLT 2005*, pages 131–136, Institute of Cybernetics, Tallinn University of Technology.

Contents

Chapter 1

Introduction

"How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?" This central question defines the discipline of Machine Learning (Mitchell, 2006) that is at the cross-border of the Computer Science and Statistics. The answer to this question holds the key to develop systems that are able to learn useful information from the massive amounts of data and that will probably replace humans in many tasks. For example, machine learning methods have already made significant progress in many real-world applications, such as speech recognition, machine vision, robot control, and are extensively used in data-intensive domains to aid the scientific discovery process. There is no doubt that in the future machine learning methods would become increasingly important in modern society. This thesis concerns particular kind of machine learning methods, namely *kernel-based methods for learning preference relations.*

Preference learning (see e.g. Fürnkranz and Hüllermeier (2005)) is a challenging task because it involves prediction of ordering of the data points rather than a single numerical value as in case of regression or a class label as in case of classification problems. It has large number of applications in many domains such as information retrieval, bioinformatics, and natural language processing which will be discussed throughout this thesis. Kernel-based methods (see e.g. Shawe-Taylor and Cristianini (2004)), on the other hand, represent very well founded and robust class of algorithms with many attractive computational properties that make them particularly suitable for learning complex tasks.

1.1 Kernel-Based Methods

The kernel-based methods represent a class of pattern recognition algorithms that are used for various tasks such as classification, regression, preference learning, clustering, etc. One common approach for all kernel-based methods is the construction of the non-linear learning algorithm by mapping data into high dimensional feature space, namely substituting linear inner product by some kernel function. Thus, the values of kernel function of data objects correspond to the inner product in mapped space (Aronszajn, 1950; Scholkopf and Smola, 2001). Training the resulting kernel-based algorithm can be considered as training the original linear algorithm using the mapped objects in the feature space. The applications of kernel-based methods have had notable success in many areas. This thesis concerns with development of novel kernel-based methods for preference learning and their consequent application to parse ranking task in natural language processing, query-document ranking in information retrieval, and remote homology detection in bioinformatics domain.

There are several reasons why kernel-based methods are applicable to many real-world problems. Firstly, instead of manual construction of the feature space for the learning task, kernel functions provide an alternative way to design useful features automatically, therefore, allowing very rich representations. Secondly, kernels can be designed to incorporate a prior knowledge about the domain. This property allows to notably improve performance of the general learning methods and their simple adaptation to the specific problem. Finally, kernel methods are applicable in situations where data representation is not in a vectorial form, thus avoiding extensive pre-processing step.

Below we describe main properties of kernel functions and some basic methods for their construction (for in depth review see Aronszajn (1950); Scholkopf and Smola (2001); Herbrich (2002); Shawe-Taylor and Cristianini (2004)). The usage of the kernels in the learning algorithm allows to take advantage of the high dimensional feature space without computational penalty that usually grows with the number of dimensions. For studying kernel functions, properties of inner product spaces are used.

A vector space $\mathcal{F}$ is an inner product space, if there exists a real valued symmetric bilinear map $\langle\cdot,\cdot\rangle$ that satisfies

$$\langle z, z\rangle \geq 0, \forall z \in \mathcal{F} \tag{1.1}$$

and

$$\langle z, z\rangle = 0 \text{ if and only if } z = 0. \tag{1.2}$$

An inner product space is a vector space together with an inner product on it. If the inner product defines a complete metric, then the inner product space is called a Hilbert space. Properties of Hilbert spaces provide a formal basis of constructing and understanding new kernel functions. The map $\Phi : \mathcal{X} \rightarrow \mathcal{F}$ is called feature map from the input space $\mathcal{X}$ into the feature space $\mathcal{F}$. The kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is used to evaluate the inner product in the feature space with some feature map Φ as follows:

$$\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j). \tag{1.3}$$

If some function can be represented as an inner product in the feature space, then it is a valid kernel function. A common way of constructing new valid kernel functions is by using closure properties of kernels. If k_1 and k_2 are kernels over $\mathcal{X} \times \mathcal{X}$, $\alpha \in \mathbb{R}_+$, $f : \mathcal{X} \to \mathbb{R}$, $\Phi : \mathcal{X} \to \mathcal{F}$ and k_3 is a kernel over $\mathcal{F} \times \mathcal{F}$, and B is a positive semidefinite matrix of dimension $n \times n$, then the following functions are kernels as well:

$$\begin{array}{rcl} k(x_i, x_j) & = & k_1(x_i, x_j) + k_2(x_i, x_j) \\ k(x_i, x_j) & = & \alpha k_1(x_i, x_j) \\ k(x_i, x_j) & = & k_1(x_i, x_j) k_2(x_i, x_j) \\ k(x_i, x_j) & = & f(x_i) f(x_j) \\ k(x_i, x_j) & = & k_3(\Phi(x_i), \Phi(x_j)) \\ k(x_i, x_j) & = & x_i^t B x_j \text{ when } \mathcal{X} = \mathbb{R}^n. \end{array}$$

Further, let us define matrix $K \in \mathbb{R}^{m \times m}$ as follows:

$$K = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{pmatrix}$$

The matrix K is called kernel matrix. It can be shown to be positive semidefinite, however, in many cases we require positive definiteness, that is, $A^t K A > 0$ for all $A \in \mathbb{R}^m, A \neq 0$. This can be ensured by performing a small diagonal shift, that is, by adding λI to K, where $I \in \mathbb{R}^{m \times m}$ is the identity matrix and λ is a small positive real number. Kernel matrix contains the evaluation of the kernel function on all pairs of inputs. Below we demonstrate how kernel matrix is used in the learning algorithm.

1.1.1 Regularized Least-Squares Algorithm

One of the most fundamental and frequently applied to many real-world problems, particularly related to regression analysis, is the least-squares method. Initially proposed by Carl Friedrich Gauss around 1794 to determine the position of the asteroid based on previous observations, since then it has had many derivatives. A regularization applied to Gaussian least-squares allowed to deal with ill-posed problems, therefore avoiding numerical instabilities when solving a system of linear equations (Tikhonov and Arsenin, 1977). This method is also known as ridge regression. Recently, the algorithm has been formulated within kernel methods framework (Saunders et al., 1998) and termed kernel ridge regression. It is also referred in the literature as least-squares support vector machines (Suykens and Vandewalle, 1999), regularized least-squares (Poggio and Smale, 2003; Rifkin et al., 2003), proximal support vector machines (Fung and Mangasarian, 2001). In this thesis we refer to the algorithm as *regularized least-squares*. It is related to the Gaussian process (Rasmussen and Williams, 2005) and to many other methods such as kernel partial least-squares (Rosipal and Trejo, 2001), kernel Fisher linear discriminant analysis (Mika et al., 1999), and kernel independent and canonical correlation analysis (Bach and Jordan, 2003). Below we briefly formulate regularized least-squares and demonstrate how *kernel trick* allows the algorithm to learn non-linear relations.

Let us construct a training set from a given set of m data points. We define a data point $z = (x, s)$ to consist of an input $x \in \mathcal{X}$ and an output $s \in \mathbb{R}$, where $\mathcal{X}$, called the input space, can be any set. Further, let $X = (x_1, \ldots, x_m) \in (\mathcal{X}^m)^t$ be a sequence of inputs, where $(\mathcal{X}^m)^t$ denotes the set of row vectors whose elements belong to $\mathcal{X}$. Correspondingly, we define $S = (s_1, \ldots, s_m)^t \in \mathbb{R}^m$ be a sequence of the outputs. Finally, we define the training set to be $T = (X, S)$.

Let us denote $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$, and let $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$ be the hypothesis space. In order to construct an algorithm that selects a hypothesis f from $\mathcal{H}$, we have to define an appropriate cost function that measures how well the hypotheses fit to the training data. Following Schölkopf et al. (2001), we consider the framework of regularized kernel methods in which $\mathcal{H}$ is Reproducing Kernel Hilbert Space (RKHS) defined by a positive definite kernel function k. We also denote the sequence of feature mapped inputs as $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_m)) \in (\mathcal{F}^m)^t$ for all $X \in (\mathcal{X}^m)^t$. Then the kernel matrix is $K = \Phi(X)^t\Phi(X)$.

We define RKHS determined by the input space $\mathcal{X}$ and the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as

$$\mathcal{H} = \left\{ f(x) = \sum_{i=1}^{\infty} \beta_i k(x, x_i), \beta_i \in \mathbb{R}, x_i \in \mathcal{X}, \|f\|_k < \infty \right\},$$

where $\|f\|_k$ denotes the norm of the function f in $\mathcal{H}$. Using RKHS as our hypothesis space, we define the learning algorithm as

$$\mathcal{A}(T) = \operatorname*{argmin}_{f \in \mathcal{H}} J(f), \tag{1.4}$$

where

$$J(f) = c(f(X), S) + \lambda \|f\|_k^2, \tag{1.5}$$

$f(X) = (f(x_1), \ldots, f(x_m))^t$, c is a real valued cost function, and $\lambda \in \mathbb{R}_+$ is a regularization parameter controlling the trade-off between the cost on the training set and the complexity of the hypothesis. By the generalized representer theorem (Schölkopf et al., 2001), the minimizer of (1.5) has the following form:

$$f(x) = \sum_{i=1}^{m} a_i k(x, x_i), \tag{1.6}$$

where $a_i \in \mathbb{R}$, that is, the problem of finding the optimal hypothesis can be solved by finding the coefficients $a_i, 1 \leq i \leq m$. Using this notation, we rewrite $f(X) = KA$ and $\|f\|_k^2 = A^t KA$, where $A = (a_1, \ldots, a_m)^t$.

Note that Support Vector Machine (SVM) algorithm (Vapnik, 1998) can be obtained by choosing $c(f(X), S)$ in (1.5) to be a hinge loss

$$c(f(X), S) = \sum_{i=1}^{m} \max(1 - s_i f(x_i), 0). \tag{1.7}$$

We use least-squares cost function to measure how well a hypothesis $f \in \mathcal{H}$ is able to predict the labels of unseen data points:

$$c(f(X), S) = \sum_{i=1}^{m} (s_i - f(x_i))^2. \tag{1.8}$$

Rewriting the cost function (1.8) in a matrix form as

$$c(f(X), S) = (S - KA)^t (S - KA),$$

and the algorithm (1.4) is

$$\mathcal{A}(T) = \operatorname*{argmin}_{A} J(A),$$

where

$$J(A) = (S - KA)^t (S - KA) + \lambda A^t KA. \tag{1.9}$$

By taking the derivative of $J(A)$ with respect to A

$$\begin{aligned} \frac{d}{dA} J(A) &= -2K(S - KA) + 2\lambda KA \\ &= -2KS + (2KK + 2\lambda K)A, \end{aligned}$$

setting it to zero and solving with respect to A, we obtain:

$$A = (K + \lambda I)^{-1} S, \tag{1.10}$$

because K and therefore also $(K + \lambda I)$ is strictly positive definite.

The calculation of the solution (1.10) requires inversion of a $m \times m$-matrix. This operation is usually performed with methods whose computational complexities are $O(m^3)$, and hence the complexity of the regularized least-squares regression is cubic.

1.1.2 Examples of Kernel Functions for Structured Data

In some cases standard linear or Gaussian kernel functions (Shawe-Taylor and Cristianini, 2004) do not allow to take advantage of the structured representation of the data, thus leading to unsatisfactory results. To address this, various kernel functions for structured data have been recently proposed. Efficient computation of these kernel functions has been one of the central parts of the kernel engineering task together with incorporation of the prior knowledge into the kernel function. For example, string (Lodhi et al., 2002; Leslie and Kuang, 2004) and graph (Kondor and Lafferty, 2002; Gärtner et al., 2003; Kashima et al., 2004) kernels lead to notably better results in many applications by taking into account the structure of the data representation. As machine learning is a rapidly developing field, novel and more general kernels functions such as universal kernels (Caponnetto et al., 2008) are introduced. However, in this section we provide a brief overview of some widely used kernel functions that are closely related to the ones proposed within the scope of this thesis. First we describe the convolution framework for constructing kernels for structured data, which is based on the idea of defining kernels between the input objects by applying convolution sub-kernels for the parts of the objects. Then we describe some general string and graph kernel functions.

Convolution Framework

Following Haussler (1999) and Watkins (1999) let us consider $x \in X$ as a composite structure such that $x_1, \ldots, x_N$ are its parts, where x_n belongs to the set X_n for each $1 \leq n \leq N$, and N is a positive integer. We consider $X_1, \ldots, X_n$ as countable sets, however, they can be more general separable metric spaces (Haussler, 1999). Let us denote shortly $\widehat{x} = x_1, \ldots, x_N$. Then the relation "$x_1, \ldots, x_N$ are the parts of x" can be expressed as a relation R on the set $X_1 \times \ldots \times X_N \times X$ such that $R(\widehat{x}, x)$ is true if $\widehat{x}$ are the parts of x. Then we can define $R^{-1}(x) = \{\widehat{x} : R(\widehat{x}, x)\}$. Now let us suppose that $x, y \in X$ and there exist decompositions such that $\widehat{x} = x_1, \ldots, x_N$ are the parts of x and $\widehat{y} = y_1, \ldots, y_N$ are the parts of y. If we have the following kernel functions

$$k_n(x_n, y_n) = \langle \Phi(x_n), \Phi(y_n) \rangle, 1 \leq n \leq N,$$

to measure similarity between elements of X_n, then the kernel $k(x, y)$ measuring the similarity between x and y is defined to be the following generalized convolution:

$$k(x, y) = \sum_{\widehat{x} \in R^{-1}(x)} \sum_{\widehat{y} \in R^{-1}(y)} \prod_{n=1}^{N} k_n(x_n, y_n). \qquad (1.11)$$

String Kernels

Let us consider two strings $p = (p_1, \dots, p_{|p|})$ and $q = (q_1, \dots, q_{|q|})$. The similarity of p and q is obtained with the kernel

$$k(p, q) = \sum_{i=1}^{|p|} \sum_{j=1}^{|q|} \kappa(i, j). \tag{1.12}$$

By specializing κ in the general formulation (1.12), we obtain different similarity functions for strings. If we set $\kappa(i, j) = \delta(p_i, q_j)$, where

$$\delta(x, y) = \begin{cases} 0, & \text{if } x \neq y \\ 1, & \text{if } x = y, \end{cases}$$

then (1.12) equals to the number of matching characters in two strings. The spectrum kernel (Leslie et al., 2002a) is obtained by using

$$\kappa(i, j) = \prod_{l=0}^{h-1} \delta(p_{i+l}, q_{j+l}), \tag{1.13}$$

in (1.12). A generalization of the spectrum kernel, that computes similarity between two strings by taking into account subsequences rather than substrings, is called the mismatch kernel. Restricting the number of mismatches to m between the subsequences of length h, the (h, m)-mismatch kernel (Leslie et al., 2004) is obtained by using

$$\kappa(i, j) = \begin{cases} 0, & \text{if } \sum_{l=0}^{h-1} \delta(p_{i+l}, q_{j+l}) < h - m \\ 1, & \text{otherwise} \end{cases} \tag{1.14}$$

in (1.12). The spectrum kernel (1.13) is a special case of the mismatch kernel where $m = 0$. It can be observed that computing these kernels using naive approach is $O(k|p||q|)$. A much more efficient approach is to use trie data structure (as described in Leslie and Kuang (2004); Shawe-Taylor and Cristianini (2004)) that could reduce the computation to $O(k(|p| + |q|))$.

Graph Kernels

Another type of frequently used kernels for the text analysis tasks are graph kernels (see e.g. Pahikkala et al. (2006)). Graph kernels are primarily applicable in the situation when graph based annotation of the text is present. An example of this type of annotation is depicted on Figure 1.1. The kernels described in Gärtner (2002) and Kashima et al. (2004) are based on the random walks in two graphs. They can take into account various similarity measures, for example, walks with equal first and last labels of the vertices,

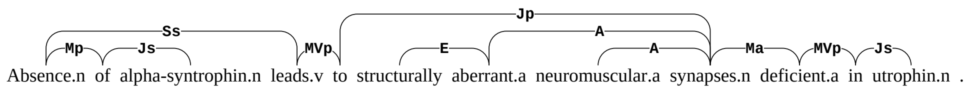
Ss
Mp
Js
MVp
Jp
E
A
A
Ma
MVp
Js
Absence.n of alpha-syntrophin.n leads.v to structurally aberrant.a neuromuscular.a synapses.n deficient.a in utrophin.n .

Figure 1.1: A sentence parsed with the link grammar parser (Sleator and Temperley, 1991) that adds to the text syntactical annotation describing different relations.

walks that have exactly all labels of the vertices matching, etc. Similarly to Gärtner (2002) and Gärtner et al. (2003), we use the following notations. We denote the set of real valued matrices of dimension $i \times j$ by $\mathcal{M}_{i\times j}(\mathbb{R})$. $[M]_{i,j}$ defines the element of matrix M that is located in the i-th row and j-th column. We define the set $\mathcal{L} = \{l_r\}, r \in \mathbb{N}$, to be the index set of all possible labels that could occur in the graph. Also, let $G = (V, E, h)$ be a graph consisting of the set of vertices V, the set of edges $E \subseteq V \times V$, and a function $h : V \to \mathcal{L}$ that assigns a label to each vertex of a graph. We suppose that the function h is represented as a label allocation matrix $L \in \mathcal{M}_{|\mathcal{L}|\times|V|}(\mathbb{R})$ so that $[L]_{i,j} = 1$ if the label of v_j is l_i and 0 otherwise. The adjacency matrix $W \in \mathcal{M}_{|V|\times|V|}(\mathbb{R})$ having the rows and columns indexed by the V and where

$$[W]_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

corresponds to the edge set of the G.

The rest of this section is mainly based on Gärtner et al. (2003). Let us consider two graphs G and G'. The vertex set of the graph $G_\times$ that would take into account common walks between the vertices of the G and G' is $V_\times \subseteq V \times V'$. The graph $G_\times$ has a vertex iff the labels of the vertices in corresponding graphs G and G' have the same label. Furthermore, there is an edge between the vertices in graph $G_\times$ iff there are edges between the corresponding vertices in both graphs G and G'. Let us denote the adjacency matrix of the graph $G_\times$ as $W_\times$. Then we have the product kernel

$$k_\times(G, G') = \sum_{i,j=1}^{|V_\times|} \left[\sum_{n=0}^{\infty} w_n W_\times^n \right]_{i,j}, \tag{1.15}$$

where $w_i \in \mathbb{R}, w_i > 0$, is the sequence of weights. This kernel can be computed efficiently using exponential or geometric series, if the limit in (1.15) exists.

Another example where similarity is measured based on the start and the end vertices of the random walk can be formulated as follows. We will use the fact that $[W^n]_{i,j}$ is the number of walks of length n from vertex

v_i to vertex v_j, where W^n denotes the nth power of the adjacency matrix of the graph G. Moreover, if the labels of the graph vertices are taken into account, $[LW^nL^t]_{i,j}$ corresponds to the number of walks of length n between vertices labelled l_i and l_j.

We denote by $\langle M, M'\rangle_F$ the Frobenius product of matrices M and M', that is, $\langle M, M'\rangle_F = \sum_{i,j}[M]_{i,j}[M']_{i,j}$. Further, let $\gamma \in \mathcal{M}_{n\times n}(\mathbb{R})$ be a positive semidefinite matrix. The kernels k_n between the graphs G and G' can be defined as follows:

$$k_n(G, G') \;=\; \textstyle\sum_{i,j=0}^{n}[\gamma]_{i,j}\langle LW^iL^t, L'W'^jL'^t\rangle_F. \tag{1.16}$$

It is interesting to notice that several specializations of this kernel function lead to different feature spaces and consequently have very different interpretations. If, for example, we set the $[\gamma]_{i,j} = \theta^i\theta^j$, where $\theta \in \mathbb{R}^+$ is a parameter, we obtain the kernel

$$\widehat{k_n}(G, G') = \langle L\Big(\sum_{i=0}^{n}\theta^iW^i\Big)L^t, L'\Big(\sum_{i=0}^{n}\theta^iW'^i\Big)L'^t\rangle_F, \tag{1.17}$$

which can be interpreted as an inner product in a feature space, in which there is a feature $\Phi_{k,l}$ per each label pair (k, l) so that its value $\Phi_{k,l}(G)$ for a graph G is a weighted count of walks of length up to n from the vertices labelled l to the vertices labelled k.

1.2 Learning Preference Relations

Recently, the task of learning preference relations (see e.g. Fürnkranz and Hüllermeier (2005)) has attracted considerable attention in machine learning research. This task involves prediction of ordering of the data points rather than a single numerical value as in case of regression or a class label as in case of classification problems.

Preference learning is often used in web search engines and recommender systems. Nowadays when large amount of information about customer preferences is available through different types of data, such as clickthrough or vote based, learning these preference relations could lead to significant increase in search engine or recommender system performance. Computerized methods for discovering preferences of individuals are useful in any field where focus towards personalization of products and services is needed.

Automated learning of preference relations constitutes a challenging task. Several ways of addressing this problem have been proposed, most of them having pros and cons. For example, a simple approach to learn preference relations is to regress the corresponding preference scores of the data points and order them accordingly afterwards. However, in this case the learning objective and the error function of the algorithm notably differ from each other. Thus, the performance of this approach is not usually satisfactory. Another possibility is to reduce the preference learning task to classification on the pairs of data points (Frank and Hall, 2001). A drawback of using this approach is that the number of the data points under consideration grows quadratically. However, several efficient ways of the reduction of preference learning to classification task have been proposed (Park and Fürnkranz, 2007; Ailon and Mohri, 2007). Recently, moving beyond pairwise preference learning was studied by Xia et al. (2008), where lists, rather than pairs of preference relations, are considered.

Various kernel-based algorithms have been proposed, such as RankSVM (Herbrich et al., 1999; Joachims, 2002), Gaussian processes for preference learning (Chu and Ghahramani, 2005b), collaborative ordinal regression (Yu et al., 2006), magnitude preserving ranking (Cortes et al., 2007), RankRLS (Pahikkala et al., 2007), and non kernel-based, such as RankBoost (Freund et al., 2003), RankNets (Burges et al., 2005), pairwise preference learning methods (Fürnkranz and Hüllermeier, 2003). Furthermore, many adaptations of these algorithms for various ranking problems have been studied. For example, SVMPerf (Joachims, 2006) is used to optimize information retrieval error measures. Sparse RankRLS (Tsivtsivadze et al., 2008b) makes ranking algorithm applicable in situations when large amount of data is available. Semi-supervised extension of the preference learning algorithms (see e.g. Chu and Ghahramani (2005a); Tsivtsivadze et al. (2008a)) can be used in case a little amount of labeled but large amount of unlabeled data is at hand.

In situations when not only preferences of the data points are available, but also their magnitudes, the magnitude preserving ranking algorithms can be used (see e.g. Pahikkala et al. (2007); Cortes et al. (2007)). Most of the time the underlying assumption for the data is that preference relations are transitive. Learning non-transitive preference relations (see e.g. Baets et al. (2006)) can be useful in many situations, for example, when using customer feedback in recommender systems, cyclic preferences can occur. Recently, we proposed a simple method for learning non-transitive preferences by reduction to the classification of the edges of the preference graph (Pahikkala et al., 2008).

The preference learning problem can be formulated within various settings depending on the type of the information provided to the learning algorithm as well as the problem at hand. For example, several problem settings have been recently proposed such as object ranking, label ranking (Fürnkranz and Hüllermeier, 2005), and multiple label ranking (Brinker and Hüllermeier, 2007).

In object ranking we aim to predict a preference relation among the objects in $\mathcal{X}$. In this case the preference relation is $\mathcal{P} \subseteq \mathcal{X} \times \mathcal{X}$. In label ranking the situation is different. Then we have also the set $\mathcal{Y}$ of labels. We would like to predict for any object $x \in \mathcal{X}$ a preference relation $\mathcal{P}_x \subseteq \mathcal{Y} \times \mathcal{Y}$ among the set of labels $\mathcal{Y}$, where each label $y \in \mathcal{Y}$ can be thought of as an alternative. An element $(y, y') \in \mathcal{P}_x$ means that the object x prefers the label y compared to y', also written as $y \succ_x y'$. As described in Fürnkranz and Hüllermeier (2005), one can distinguish between *weak preference* ($\succeq$) and *strict preference* ($\succ$), where $y \succ_x y' \Leftrightarrow (y \succeq_x y') \wedge (y' \not\succeq_x y)$. Furthermore, $y \sim_x y' \Leftrightarrow (y \succeq_x y') \wedge (y' \succeq_x y)$.

One commonly used example to illustrate this label ranking task is document ranking in information retrieval. Every input consists of the query and the document that is retrieved by the query. If query retrieves more than a single document then we obtain set of query document pairs, where each document corresponds to the label and the query to the object. Furthermore, because we would like to learn preferences between the documents that belong to the same query, we consider only the query-document pairs that contain exactly the same query to be relevant to the task. Similar problem occurs in natural language processing, namely parse ranking task. Here the sentence and the parse are considered as an object and a label, respectively. We are given a set of sentences and each sentence is associated with a set of parses. The task is to find the correct ordering of the parses of a sentence.

1.2.1 Learning Tasks

The problem of learning preference relations is related to standard classification and regression tasks. In particular, one can consider preference learning within classification framework as a problem of classifying pairs of data points. When the data points have not only preferences, but also magnitudes available, one can also cast preference learning as a regression problem.

Classification example
An input $x \in \mathcal{X}$ gets assigned to a single label $y_x \in \mathcal{Y}$. Then a set of pairwise label preferences is simply induced by the class labels, namely $\mathcal{P}_x = \{(y_x, y) | y \in \mathcal{Y} \setminus \{y_x\}\}$.

Regression example
This is similar to the classification example, but in this case an example $x \in \mathcal{X}$ gets assigned to a real valued label $y \in \mathbb{R}$. Now the set of preferences is obtained from real valued scores, which induce total order over the whole set of inputs.

Ranking example
An input $x \in \mathcal{X}$ gets assigned to the total order of the labels, that is, $\succ_x$ is transitive relation such that $y \succ_x y'$ or $y' \succ_x y$ holds for all pairs of labels (y, y').

One approach for learning preference relations is based on estimation of the utility (scoring) function. In this scenario, we can use the predicted scores to rank the data items accordingly (see e.g. Har-Peled et al. (2002); Pahikkala et al. (2007)). On the other hand, approaches for learning pairwise preferences are also widely used. In order to rank the labels for a new object, predictions for all pairwise label preferences are obtained and a ranking that is maximally consistent with these preferences is derived.

In the following sections we consider problems having a total order induced over the set of the preferences. This is the case when every data point has a real score associated with it, therefore indicating not only the direction of the preference but also its magnitude. In some real-world situations this is a typical setting. However, frequently preferences without magnitudes can be available to the learning algorithm (e.g. clickthrough data (Joachims, 2002)). We address the issue of learning from pairwise preferences without scores in Pahikkala et al. (2009).

1.2.2 Label Ranking in Scoring Setting

We assume that the (true) preference relation $\mathcal{P}_x$ is transitive and asymmetric for each object $x \in \mathcal{X}$. As training information, we are given a finite set $T = \{(z_i, s_i)\}_{i=1}^m$ of m data points, where each data point $(z_i, s_i) = ((x_i, y_i), s_i) \in (\mathcal{X} \times \mathcal{Y}) \times \mathbb{R}$ consists of an object-label tuple $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ and its score $s_i \in \mathbb{R}$. We say that two data points $((x, y), s)$ and $((x', y'), s')$ are *relevant*, iff $x = x'$. Considering two relevant data points $((x, y), s)$ and $((x, y'), s')$, we say that object x *prefers* label y to y', if $s > s'$. If $s = s'$, the labels are called *tied*. Accordingly, we write $y \succ_x y'$ if $s > s'$ and $y \sim_x y'$ if $s = s'$.

A *label ranking function* is a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ mapping each object-label tuple (x, y) to a real value representing the predicted relevance of the label y with respect to the object x. This induces for any object $x \in \mathcal{X}$ a transitive preference relation $\mathcal{P}_{f,x} \subseteq \mathcal{Y} \times \mathcal{Y}$ with $(f(x, y), f(x, y')) \in \mathcal{P}_{f,x} \Leftrightarrow f(x, y) \geq f(x, y')$. Informally, the goal of our ranking task is to find a label ranking function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that the ranking $\mathcal{P}_{f,x} \subseteq \mathcal{Y} \times \mathcal{Y}$ induced by the function for any object $x \in \mathcal{X}$ is a good "prediction" for the (true unknown) preference relation $\mathcal{P}_x \subseteq \mathcal{Y} \times \mathcal{Y}$.

1.2.3 Object Ranking

Similarly to label ranking setting, we are given a set of objects $\mathcal{X}$. However, here each object does not have a set of labels associated with it, but we would like to rank objects themselves according to some preference relation $\mathcal{P} \subseteq \mathcal{X} \times \mathcal{X}$. Every object has an associated score $s \in \mathbb{R}$, which can be interpreted as the "goodness" score, based on which ranking occurs.

As training information, we are given a finite set $T = \{(z_i)\}_{i=1}^m$ of m data points, where each data point $z_i = (x_i, s_i) \in (\mathcal{X} \times \mathbb{R})$ consists of an object $x_i \in \mathcal{X}$ and its score $s_i \in \mathbb{R}$. When considering two data points (x, s) and (x', s'), we say that the object x is *preferred* to x' if $s > s'$. If $s = s'$, the objects are called *tied*. Accordingly, we write $x \succ x'$ if $s > s'$ and $x \sim x'$ if $s = s'$. Similar description of the object ranking task is presented in Fürnkranz and Hüllermeier (2005), where preferences between the objects not necessarily have an associated magnitude.

An *object ranking function* is a function $f : \mathcal{X} \to \mathbb{R}$ mapping each object x to a real value from which the predicted preference (rank) with respect to the other objects in $x' \in \mathcal{X}$ can be deduced. This induces for any object $x \in \mathcal{X}$ a transitive preference relation $\mathcal{P}_f \subseteq \mathcal{X} \times \mathcal{X}$.

1.2.4 Measuring Ranking Performance

In order to measure the ranking performance, we adopt two commonly used measures, namely disagreement error and Kendall's rank correlation coefficient τ. We assume that we are given finite set $T = \{(z_i)\}_{i=1}^m$ of m data points as described Section 1.2.3. To formulate Kendall's correlation coefficient, let us define the function

$$F(z_i, z_j) = \begin{cases} 1 & \text{if } x_i \succ x_j \\ -1 & \text{if } x_j \succ x_i \\ 0 & \text{otherwise.} \end{cases}$$

Let us define the score $U(z_i, z_j)$ of a pair z_i and z_j to be the product

$$U(z_i, z_j) = F(z_i, z_j) F(f(z_i), f(z_j)).$$

If score is $+1$, then the rankings agree on the ordering of z_i and z_j, otherwise score is -1. The total score is defined as

$$U = \sum_{1 \le i < j \le m} U(z_i, z_j).$$

The number of all different pairwise comparisons of the data points that can be made is $\binom{m}{2} = \frac{1}{2} \cdot m\,(m-1)$. This corresponds to the maximum value of the total score, when agreement between the rankings is perfect. The correlation coefficient τ_a defined by Kendall (1970) is

$$\tau_a = \frac{U}{\frac{1}{2} \cdot m\,(m-1)}.$$

While τ_a is well applicable in many cases, there is an important issue that is not fully addressed by this coefficient, namely tied ranks. To take into account possible occurrences of tied ranks, Kendall proposes an alternative correlation coefficient

$$\tau_b = \frac{U}{\frac{1}{2}\sqrt{\sum_{i,j} F(z_i, z_j)^2 \cdot \sum_{i,j} F(f(z_i), f(z_j))^2}}, \tag{1.18}$$

where $1 \leq i, j \leq m$. With tied ranks the usage of τ_b is more justified than usage of τ_a. For example, if both rankings are tied except the last pair, then $\tau_b = 1$ indicating complete agreement between two rankings, while $\tau_a = \frac{2}{m}$. Both τ_a and τ_b performance measures can be used also in the label ranking setting. When considering the label ranking, the overall ranking performance can be computed by estimating the ranking of the labels associated with a single object and averaging over total number of the objects afterwards.

On the other hand, we can use a simple disagreement error (see e.g. Dekel et al. (2004)) to measure how well a hypothesis $f \in \mathcal{H}$ is able to predict the preference relations $\mathcal{P}_x$ for all instances $x \in \mathcal{X}$ (see Section 1.1.1). Thus, we consider the following cost function that captures the amount of incorrectly predicted pairs of relevant training data points:

$$d(f(Z), S, W) = \frac{1}{2n} \sum_{i,j=1}^{m} W_{i,j} \left| \operatorname{sign}\big(s_i - s_j\big) - \operatorname{sign}\big(f(z_i) - f(z_j)\big) \right|, \tag{1.19}$$

where $n = \sum_{i,j=1}^{m} W_{i,j}$, $f(Z) = (f(z_1), \ldots, f(z_m))^t \in \mathbb{R}^m$, $\operatorname{sign}(\cdot)$ is the signum function, $S = (s_1, \ldots, s_m)^t \in \mathbb{R}^m$ is a sequence of the outputs, and $W \in \mathbb{R}^{m \times m}$ denotes the adjacency matrix of the graph, that is, $W_{i,j} = 1$ iff edge (z_i, z_j) is relevant for the learning task and $W_{i,j} = 0$ otherwise.

We expect the learning algorithm minimizing the disagreement error (1.19) to perform well in ranking of the unseen data points. In Pahikkala et al. (2009, 2007) and Tsivtsivadze et al. (2008b), we demonstrate that by using least-squares approximation of (1.19) we are, in fact, regressing the differences $s_i - s_j$ with $f(z_i) - f(z_j)$ and in many cases this leads to significantly better performance compared to regressing the scores of individual data points and performing the ranking afterwards. The RankRLS algorithm can be derived in similar manner as RLS regression described in Section 1.1.1 by using using least-squares approximation of (1.19) instead of

(1.8). By denoting the Laplacian matrix $\mathcal{L} = D - W$, where D is a diagonal matrix whose entries are defined as $D_{i,i} = \sum_{j=1}^{m} W_{i,j}$, the objective function can be written as in Pahikkala et al. (2009):

$$J(A) = (S - KA)^t \mathcal{L}(S - KA) + \lambda A^t KA. \tag{1.20}$$

It can be shown that the minimizer of (1.20) is

$$A = (K\mathcal{L}K + \lambda K)^{-1} K\mathcal{L}S. \tag{1.21}$$

The computational complexity of the matrix inversion operation involved in (1.21) is $O(m^3)$, thus, training of RankRLS algorithm has the same complexity as training of standard RLS regression.

Chapter 2

Research Overview

2.1 Research Objectives

In this thesis, we aim at introducing efficient algorithms that can be applied for learning preference relations. These algorithms, formulated within the kernel-based methods framework, allow the use of various non-linear functions to improve the performance of the methods. Therefore, objectives of this thesis include development of *kernel functions for structured data* that are used to take advantage of various data representations and the *preference learning algorithms* that are suitable for different tasks, namely efficient learning of preference relations, learning with large amount of training data, and semi-supervised preference learning.

In particular, we plan to address computational efficiency of the preference learning algorithms by proposing the methods that scale linearly with the number of training data points. Furthermore, we aim at studying approximation techniques to speed up the training and possible extension of the algorithm that would make it applicable in situations when only small amount of labeled data, but a large amount of unlabeled data is available.

2.2 Kernels for Structured Data

In the following section we describe several kernel functions proposed within the scope of this thesis. Presented discussion is based on the publications I-III.

2.2.1 Locality Kernels for Parse Ranking

With availability of structured data in many areas, particularly in natural language processing (NLP) (see e.g. Ginter et al. (2004)), many applications of kernel-based algorithms have been recently presented (see e.g.

Shawe-Taylor and Cristianini (2004); Scholkopf and Smola (2001); Herbrich (2002)). For example, Collins and Duffy (2001) described convolution kernels for various discrete structures encountered in NLP tasks, which allow high dimensional representations of these structures in feature space.

We propose a framework for constructing kernels that take advantage of local correlations in sequential data. The kernels are designed to measure data similarities locally, within a small window built around each matching feature. Furthermore, we propose to incorporate positional information inside the window and consider different ways to do this. We call the kernel functions constructed within this framework locality kernels (Tsivtsivadze et al., 2008d).

The performance of the locality kernels is evaluated on the dependency parse ranking task in the biomedical domain. The parses are generated by the Link Grammar (LG) parser (Sleator and Temperley, 1991) which is applied to the BioInfer corpus (Pyysalo et al., 2007) containing 1100 annotated sentences. The parser is based on broad-coverage hand-written grammar and operates by generating all parses that are allowed by its grammar. To rank the generated parses, it uses built-in heuristics. However, its ranking performance has been found to be poor when applied to biomedical text (Pyysalo et al., 2004). Therefore, we consider improving ranking performance of the parser, by proposing a learning algorithm appropriate for the task.

We associate each sentence of the BioInfer corpus with a set of generated parses. The manual annotation of the sentence, present in the corpus, provides the correct parse. Each candidate parse is associated with a goodness score that indicates how close to the correct parse it is. The correct ranking of the parses associated with the same sentence is determined by this score. While the scoring induces a total order over the whole set of parses, the preferences between parses associated with different sentences are not considered in the parse ranking task.

Our method is based on the Regularized Least-Squares (RLS) algorithm (see e.g Tsivtsivadze et al. (2005) or Section 1.1.1). We use grammatically motivated features and the locality kernels to obtain significantly better ranking performance than that of built-in LG heuristics. We extend the results of Tsivtsivadze et al. (2006) by considering a framework for constructing kernels that take advantage of local correlations in sequential data and benchmarking locality kernels against the locality-improved (Zien et al., 2000) and spectrum (Leslie et al., 2002b) kernels, which also are designed to be applicable to sequential data. In all experiments, we apply the F-score based parse goodness function (Tsivtsivadze et al., 2005) and evaluate the ranking performance with Kendall's correlation coefficient τ_b (Kendall, 1970) described in Section 1.2.4. The results indicate that locality kernels notably outperform baseline methods and the performance gain is statistically significant.

2.2.2 Locality Kernels for Remote Homology Detection

An important task in computational biology is inference of the structure and function of the protein encoded in the genome. In many cases the similarity of protein sequences can imply structural and functional similarity as well. Therefore, detecting these similarities can lead to useful discoveries in pharmaceutical domain, as well as better drugs and shorter development time. This task (detecting similarities) can be formulated as a classification or bipartite ranking problem that treats proteins as a set of labelled examples which are in positive class if they belong to the same family and are in negative class otherwise.

Recently, there have been reported many applications of simple discriminative approach for detecting remote protein homologies. For example, Jaakkola et al. (2000) report good performance in protein family recognition by combining discriminative learning algorithm with Fisher kernel for extracting relevant features from the protein sequences. Liao and Noble (2003) improve the results by proposing a combination of pairwise sequence similarity feature vectors with Support Vector Machines (SVM) algorithm. Their algorithm called SVM-pairwise is performing significantly better than several other baseline methods such as SVM-Fisher, PSI-BLAST, and profile HMMs.

Despite notably better performance than many previously reported methods, the algorithms described in Jaakkola et al. (2000) and Liao and Noble (2003) have a drawback, namely they use expensive step of generating the features, which increases the overall computational time of the algorithm. Instead, the idea to use a simple kernel function that can be efficiently computed and which does not depend on any generative model or separate pre-processing step is considered by Leslie et al. (2002a). They show that simple sequence based kernel functions perform surprisingly well compared to other computationally expensive approaches.

Following this direction, we address the problem of protein sequence classification using the RLS algorithm with locality kernels (Tsivtsivadze et al., 2007). The features of the locality kernels are the sequences contained in a small window constructed around matching amino acids in the compared proteins. Proposed kernels make use of different similarity evaluations within the windows, namely *position insensitive matching*: amino acids that match are taken into account irrespective of their position, *position sensitive matching*: amino acids that match but have different positions are penalized, and *strict matching*: only amino acids that match and have the same positions are taken into account. By incorporating information about relevance of local correlations and positions of amino acids in the sequence into the kernel function, we demonstrate significantly better performance in protein classification on Structural Classification of Proteins

(SCOP) database (Hubbard et al., 1997) than that of the spectrum and the mismatch kernels (Leslie et al., 2002a, 2004; Leslie and Kuang, 2004). When used together with RLS, the performance of the locality kernels is comparable with some state-of-the-art methods of protein classification and remote homology detection.

2.2.3 Graph Kernels and Representations

Another frequently encountered data structure in many tasks is graph. One example where graph structures occur naturally are dependency parses (see Figure 1.1). Recently, kernel functions for instances that are represented by graphs were introduced by Gärtner (2002), Kashima and Inokuchi (2002), Kondor and Lafferty (2002). Motivated by this research, we propose graph representations and kernels for dependency parses and analyse the applicability of the graph kernels for the problem of parse ranking in the domain of biomedical texts (Pahikkala et al., 2006; Tsivtsivadze et al., 2008c).

Our approach builds on Tsivtsivadze et al. (2006), but in Tsivtsivadze et al. (2008c) we further developed the method by designing graph representations for data and adapting the kernels for these graphs. Our results show good performance in the task of parse ranking. We demonstrate that the proposed approach has several computational advantages, and can be considered as a generalization over previously described method (Tsivtsivadze et al., 2005). The results indicate that designing the graph representation is as important as designing the kernel function that is used as the similarity measure of the graphs.

2.3 Efficient Preference Learning Algorithms

In the following section we describe preference learning algorithms proposed within the scope of this thesis. Presented discussion is based on the publications IV-VI.

2.3.1 Pairwise Regularized Least-Squares

There are many tasks where the aim is not to learn how to correctly classify or regress data points, but to learn preference relations. For example, in information retrieval or collaborative filtering a challenging task is to predict user preferences based on the preferences of the other users. In such cases, it is typical to train algorithm on pairs of data points in order to find the preferences. When data point pairs are constructed explicitly, it can lead to quadratic increase in the size of dataset. To address this issue, we propose a preference learning algorithm that minimizes the regularized least-squares (RLS) error and has $O(m^3)$ computational complexity, where

m is the number of the data points (Pahikkala et al., 2007). Our algorithm possesses a set of computational properties (e.g. fast cross-validation, regularization parameter estimation) that makes it in some cases more efficient than other state-of-the-art (e.g. RankSVM) preference learning methods. We refer to our algorithm as RankRLS. Detailed empirical study in terms of runtime efficiency and predictive performance of the algorithm is reported in Pahikkala et al. (2009).

RankRLS can learn preferences that are relevant to the task in question. For each training data point, there is a vertex in the preference graph and two vertices are connected if the corresponding data point pair is relevant. This relevance information is conveyed into RankRLS by the Laplacian matrix of the graph. We consider the relevant pairs, because in some cases it may be harmful to the ranking performance to minimize the RLS error of all data point pairs. Our experiments on parse ranking task confirm that RankRLS trained to minimize the RLS error for every possible pair of training data points has a lower performance than RankRLS that minimizes the error only for the relevant pairs. Furthermore, we extend RankRLS algorithm so that it can be used to learn not only from scored data, but also from a given sequence of pairwise preferences and their magnitudes. We consider the case in which algorithm learns scoring (utility) function that maps each possible input to a real value. The function induces a total order for the inputs. The direction of preference between two inputs is obtained by comparing their predicted scores.

We demonstrate that the performance of the RankRLS method is comparable to RankSVM (Joachims, 2006) and RankBoost (Freund et al., 2003) in information retrieval task. Furthermore, RankRLS significantly outperforms naive regression approach in parse ranking task.

2.3.2 Large-Scale Preference Learning

In many cases preference learning problems can be cast as a classification of data point pairs (see e.g. Herbrich et al. (1999); Har-Peled et al. (2002); Fürnkranz and Hüllermeier (2003); Ailon and Mohri (2007)). However, when using this approach usually the number of data point pairs grows quadratically with respect to the size of the dataset. In situations when the size of the dataset is large, the training of the algorithm could be infeasible. We have previously proposed a preference learning algorithm (Pahikkala et al., 2007) having computational complexity same as that of the standard RLS regression, despite the fact that it takes into account data point pairs instead of the individual data points. When using dual version of the algorithm, the computational complexity is $O(m^3)$, where m is the size of the number of data points.

To even further reduce computational complexity, we propose sparse

RankRLS, a sparse regularized least-squares algorithm for learning preference relations (Tsivtsivadze et al., 2008b). The algorithm is in particular applicable to the situations when nonlinear kernel functions are used and the number of data points is large. Formally, the computational complexity of sparse RankRLS is $O(mr^2)$, where r is the number of basis vectors as described in Tsivtsivadze et al. (2008b). Various methods for subset selection have been proposed (see e.g.Vincent and Bengio (2002); Quinonero-Candela and Rasmussen (2005)), however, for simplicity and computational efficiency we consider random selection of basis vectors. In fact, r can be selected to be much smaller than m and in some cases it can be considered as a constant. Thus, our algorithm can efficiently perform ranking using a large amount of training data together with high dimensional feature representations. It can be observed that the algorithm approximates the error function that simply counts number of incorrectly ranked data points.

We consider the case where every input is associated with its real valued score. Thus, the total order of the data points can be obtained. The algorithm can be used for both object ranking and label ranking tasks (see e.g. Fürnkranz and Hüllermeier (2005) for in depth discussion about these types of tasks), however, we only consider label ranking. According to this setting, we define the input to consist of an object and its label.

A drawback of the standard RankRLS algorithm is that it can not be trained with large amount of data efficiently. We demonstrate that the proposed sparse RankRLS algorithm significantly outperforms basic RankRLS in this situation. Because training of the existing kernel based ranking algorithms, such as RankSVM (Herbrich et al., 1999; Joachims, 2002), could be infeasible when the size of the training set is large, sparse RankRLS is an efficient alternative to these methods, particularly, when nonlinear kernel functions are used.

2.3.3 Sparse Co-Regularized Least-Squares

In many real-world situations only a limited amount of labelled data and a large amount of unlabeled data is usually available to the learning algorithm. Therefore, methods that can use unlabeled data in the training process have been gaining more and more attention in recent years. Multi-view algorithms (Blum and Mitchell, 1998) are frequently used to take into account unlabeled data for increasing the learning performance. They split the attributes into independent sets and perform learning based on these different "views". Informally, the goal is to find a prediction function for each view such that it performs well on the labelled data and all predictions "agree" on the unlabeled data as well. An example task where multi-view approach is usually applied is the classification of web documents based on two very distinct views: one containing representation of the document based on text related feature and the other view containing only link related features. Closely related to this approach is the co-regularization framework described in Sindhwani et al. (2005), where the same idea of agreement maximization between the predictors is central. Formally, the algorithm searches for hypotheses from different Reproducing Kernel Hilbert Spaces (Schölkopf et al., 2001) such that the training error of each hypothesis on the labelled data is small and, at the same time, the hypotheses give similar predictions for the unlabeled data. Within this framework, the disagreement is taken into account via a co-regularization term.

We propose a semi-supervised preference learning algorithm that is based on the multi-view approach (Tsivtsivadze et al., 2008a). Our algorithm, that we call Sparse Co-RankRLS, minimizes a least-squares approximation of the ranking error and is formulated within the co-regularization framework. We consider a problem of learning a function capable of arranging data points according to a given preference relation (Fürnkranz and Hüllermeier, 2005). We aim at improving the performance of the learning algorithm by making it applicable to situations when only a small amount of labelled data, but a large amount of unlabeled data is available. Furthermore, the computation complexity of the algorithm is linear in the number of unlabeled data points, which is crucial when trained on large unlabeled datasets.

Empirical results reported in the literature show that the co-regularization approach works well for classification (Sindhwani et al., 2005), regression (Brefeld et al., 2006), and clustering (Brefeld and Scheffer, 2004) tasks. We extend it to preference learning task. Moreover, theoretical investigations demonstrate that the co-regularization approach reduces the Rademacher complexity by an amount that depends on the "distance" between the views (Rosenberg and Bartlett, 2007; Sindhwani and Rosenberg, 2008).

We test our algorithm, similarly to several other publications presented in this thesis, on a parse ranking task (Tsivtsivadze et al., 2005). However, we adopt slightly different setup from previously described, where we simulate unlabeled data by ignoring part of the labels present in the dataset. We demonstrate that Sparse Co-RankRLS is computationally efficient when it is trained on large datasets and the results are significantly better than the ones obtained with the standard RankRLS algorithm.

Chapter 3

Summary of Publications

This thesis consists of six publications. A short summary of each publication is presented in this chapter.

3.1 Kernels for Structured Data

Paper I: Locality Kernels for Sequential Data and their Applications to Parse Ranking (Tsivtsivadze et al., 2008d)

Kernel functions introduced within the scope of this thesis concern improving performance of the algorithm by constructing feature space that allows accurate and efficient learning. We describe a framework for constructing locality kernels that take advantage of the correlations in sequential data. These kernels use features extracted in the order of the appearance from the sequence, construct local windows around matching features in order to capture local correlations, and perform a position sensitive (or insensitive) evaluation of the features within the window. Final validation results demonstrate that the RLS algorithm with locality kernels performs significantly better than several other methods in parse ranking problem. Further, performance gain obtained with locality kernels is statistically significant. Although empirical evaluation of the locality kernels was conducted on parse ranking task, proposed kernels are not restricted for the particular domain and can be applied to any sequential data where positional matching, or local correlations can be useful.

Paper II: Kernels for Text Analysis (Tsivtsivadze et al., 2008c)

We demonstrate how to construct a kernel function that is appropriate for the task in question and provide illustrative examples of several widely used kernels. We start with basic principles for constructing kernels for natural language processing tasks and formulate "bag of words" kernels (Joachims, 1998), kernel for strings (Lodhi et al., 2002), word-sequence (Cancedda et al., 2003) and other types of kernels applicable for sequential data. We also demonstrate how to incorporate prior knowledge about problem domain into the kernel function. Furthermore, for the tasks where data has graph based structure (e.g. sentence with dependency annotation), we design graph representations and kernels. These kernels are used to generate features that are based on the start and end labels of random walks between vertices in the graphs. Thus, the feature vector corresponding to a data point is determined by the structure of the graph and the kernel function. Our graph representations and the graph kernels are applied to the problem of parse ranking in the domain of biomedical texts. Achieved performance is promising when compared to our previous studies (Tsivtsivadze et al., 2005) and could be even further improved by designing representations capturing additional prior knowledge about the problem to be solved. The results underline the importance of the design of a good graph representation for the data points as well as applicability of the graph kernels to various tasks in natural language processing, where the data points can be represented as graphs. Introduced graph representation and kernels allow prior knowledge about the learning problem to be incorporated into the learning algorithm. This can significantly improve performance of the method as demonstrated by our experiments.

Paper III: Locality Kernels for Protein Classification (Tsivtsivadze et al., 2007)

Remote homology detection is an important task in computational biology, because it can lead to inference of the structure and function of the protein encoded in the genome. To incorporate domain knowledge about the problem at hand, we make kernels that take advantage of local correlations between the amino acids in the protein sequence. Using the framework proposed in Paper I, we make locality kernels measure protein sequence similarities within a small window constructed around matching amino acids. The kernels incorporate positional information of the amino acids inside the window and allow a range of position dependent similarity evaluations. We use these kernels with regularized least-squares algorithm (RLS) for protein classification on the SCOP database. Our experiments demonstrate that the locality kernels perform significantly better than the spectrum and the mismatch kernels. When used together with RLS, performance of the locality kernels is comparable with some state-of-the-art methods of protein classification and remote homology detection.

3.2 Efficient Preference Learning Algorithms

Paper IV: An Efficient Algorithm for Learning to Rank from Preference Graphs (Pahikkala et al., 2009)

We propose a method for learning preference relations called RankRLS. Despite the fact that the number of possible data point pairs is quadratic, training complexity of this algorithm is $O(m^3)$, that is, the same as of the regularized least-squares regressor. RankRLS can learn preferences of the data points that are relevant to the task in question, both from the scored data and the data consisting of pairwise preferences without magnitudes. For this purpose, the Laplacian matrix of the preference graph is used to encode the information into RankRLS. We describe both primal and dual formulation of the algorithm. The primal form is applicable in situations when large amount of data points are available, but the amount of features is limited. The dual form is used in the opposite case. We also present the efficient methods for training, cross-validation, parameter selection, and multiple output learning for RankRLS. We describe the extension of RankRLS, initially proposed in Paper V, that is suitable for large scale learning and demonstrate its computational benefits. Finally, the evaluation of the preference learning algorithm is conducted on several tasks in information retrieval and natural language processing domains. The obtained results demonstrate that RankRLS algorithm is comparable to the state-of-the-art methods such as RankSVM. Moreover, it outperforms naive approaches of regressing the individual labels of the data points.

Paper V: A Sparse Regularized Least-Squares Preference Learning Algorithm (Tsivtsivadze et al., 2008b)

One drawback associated with RankRLS algorithm presented in Paper IV is that when using nonlinear kernel functions its computational complexity is cubic with respect to the number of training examples. Particularly, training of the algorithm can be infeasible on large datasets. To address this issue, we propose a sparse approximation of RankRLS whose training complexity is considerably lower than that of basic RankRLS. Formally, the computational complexity of the algorithm is $O(mr^2)$, where m is the number of training examples, and r is the number of basis vectors being much smaller than m. Similarly to RankRLS algorithm, the proposed method is formulated within kernel framework. The main advantage of sparse RankRLS is the computational efficiency when dealing with large amounts of training data together with high dimensional feature representations. The experiments indicate significantly better performance compared to basic RLS regressor, sparse RLS regressor, and basic RankRLS.

Paper VI: Learning Preferences with Co-regularized Least-Squares (Tsivtsivadze et al., 2008a)

Both RankRLS and sparse RankRLS are supervised learning algorithms, therefore, they can not take into account unlabeled data. However, in many cases labelled data is scarce and large amount of unlabelled data is available. These cases are frequently encountered in many real-world applications including the ones in natural language processing, bioinformatics, etc. Within the scope of this thesis, we propose a semi-supervised preference learning algorithm that is based on the multi-view approach. Our algorithm operates by constructing a predictor for each view and by choosing such prediction hypotheses that minimize the disagreement among all of the predictors on the unlabeled data. The algorithm is formulated within co-regularization framework, thus, we call it Sparse Co-RankRLS. The computational complexity of our algorithm is $O(M^3r^3 + M^2r^2n)$, where n is the number of unlabeled training examples, M is the number of views, and r is the number of basis vectors. Thus, our semi-supervised preference learning algorithm has a linear complexity in the number of unlabeled examples, which is useful in cases when a small amount of labelled, but a large amount unlabeled data is available for training. The experimental evaluation of the method shows significantly better performance of Sparse Co-RankRLS compared to the standard RankRLS algorithm.

Chapter 4

Conclusions

4.1 Research Conclusions

This thesis addresses the following issues concerning the development of the preference learning algorithms: efficient learning of pairwise preferences, sparse approximation for making the algorithms computationally feasible when training on large datasets, and semi-supervised learning in case when little amount of labelled, but large amount of unlabeled data is available. We develop not only computationally efficient algorithms but also procedures for fast regularization parameter selection, multiple output prediction, and cross-validation.

The algorithms are formulated within kernel methods framework, that makes them applicable to variety of tasks where the use of non-linear feature spaces is necessary. We further improve performance of the learning algorithm by proposing kernel functions that are tailored for the task in question, namely locality kernels for sequential data and graph kernels for data consisting of graph representations. By using these kernel functions, we not only are able to learn from structured data but also incorporate prior knowledge about the problem for the algorithm.

The overall results obtained in this work suggest that indeed, introduced kernel-based algorithms lead to notably better performance in many preference learning tasks considered. They can be successfully modified to accommodate cases where efficient training with large amount of data is necessary or labeled data is scarce but unlabeled data is abundant, which is the case in many real world problems. Furthermore, we demonstrate applicability of proposed methods to the parse ranking task in natural language processing, document ranking in information retrieval, and remote homology detection in bioinformatics domain. Due to general formulation of the algorithms they can be suitable for learning preference relations in other domain as well.

4.2 Future Work

Preference learning is rapidly growing research area with many novel concepts and algorithms being proposed. Still, there are many problems to be addressed both on theoretical and method development side. For example, regularization plays crucial role for many machine learning algorithms (e.g. sparse regularization, Tikhonov regularization, etc.). Recently, manifold regularization has been introduced (Belkin and Niyogi, 2004) for variety of semi-supervised learning tasks. Discovering regularizations that are particularly suitable for preference learning is one of the research directions that might lead to increase in performance and efficiency of the learning algorithms. Recently, we have demonstrated how to use large amount of unlabeled data to boost the performance of the algorithm while learning preferences (Tsivtsivadze et al., 2008a), however, much more detailed theoretical and empirical analysis is necessary.

Another challenging direction is multi-label (Brinker and Hüllermeier, 2007) multi-task preference learning that might be useful in many applications. Multi-task algorithms (Thrun and Pratt, 1998) aim at learning many related tasks together. This is done by taking advantage of the commonalities among the tasks. One particular example, where multi-task learning could be applied, is information retrieval field (e.g. search, or collaborative filtering problem (Yu and Tresp, 2005)) where simultaneous learning of preferences of the users can be accomplished.

In many cases there is little motivation to learn the complete ordering of the data points. For example, information retrieval measures such as Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002) can take into account the correct ranking only of the top N retrieved results. In bioinformatics, ROC50 score takes into account true positives only up to first 50 false positive results. Clearly, in this situation an algorithm that would concentrate on learning the top ranked data points correctly, on expense of possibly misordering the data points at the end of the list, might be useful. P-norm push ranking algorithm (Rudin, 2006) addresses this particular issue. Furthermore, SVMperf that is able to optimize NDCG measure has been recently proposed (Joachims, 2005, 2006). Despite encouraging results obtained using these methods, the challenge is further development of the computationally efficient algorithms to improve performance of correctly predicting ranks of top N data points.

Nowadays, when the amount of available data for the training of the algorithm increases rapidly, methods that are able to efficiently use large amount of data are necessary. Therefore, various approximation techniques that allow learning with less computational expense than standard methods are being proposed. For example, in Tsivtsivadze et al. (2008b) we demonstrate one such extension to RankRLS algorithm. Other type of approximation algorithms are particularly applicable in case data matrix is sparse. Simple gradient decent methods are proved to be invaluable for large-scale learning (see. e.g. Chu et al. (2005); Bottou et al. (2007)). Considering preference learning task, situations when algorithms have to scale to very large dataset are frequently occurring in recommender systems, web search engines, etc.

The concept of pairwise learning has been thoroughly investigated by many researchers. Moving beyond pairwise preferences is another promising direction. For example, in Xia et al. (2008) an approach is proposed where probabilistic ordering is considered on the level of the list of data points. Thus, algorithms that can take advantage not only of the pairs, but also lists of the preference relations, might lead to better results.

Bibliography

Ailon, N. and Mohri, M. (2007). An efficient reduction of ranking to classification. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, Lecture Notes in Computer Science, Heidelberg, Germany. Springer.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Bach, F. R. and Jordan, M. I. (2003). Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48.

Baets, B., Meyer, H., and Schuymer, B. (2006). Cyclic evaluation of transitivity of reciprocal relations. *Social Choice and Welfare*, 26(2):217–238.

Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning Journal*, 56(1-3):209–239.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100. New York, NY, USA.

Bottou, L., Chapelle, O., DeCoste, D., and Weston, J. (2007). *Large Scale Kernel Machines*. MIT Press, Cambridge, MA.

Brefeld, U., Gärtner, T., Scheffer, T., and Wrobel, S. (2006). Efficient co-regularised least squares regression. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 137–144, New York, NY, USA. ACM.

Brefeld, U. and Scheffer, T. (2004). Co-em support vector learning. In *Proceedings of the 21st International Conference on Machine Learning*, pages 121–128, New York, NY, USA. ACM.

Brinker, K. and Hüllermeier, E. (2007). Case-based multilabel ranking. In Veloso, M. M., editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 702–707.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96, New York, NY, USA. ACM.

Cancedda, N., Gaussier, E., Goutte, C., and Renders, J. (2003). Word sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Caponnetto, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2008). Universal multi-task kernels. *Journal of Machine Learning Research*, 9:1615–1646.

Chu, W. and Ghahramani, Z. (2005a). Extensions of Gaussian processes for ranking: Semi-supervised and active learning. In *Proceedings of the Learning to Rank Workshop at NIPS 2005.*

Chu, W. and Ghahramani, Z. (2005b). Preference learning with Gaussian processes. In *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pages 137–144, New York, NY, USA.

Chu, W., Ong, C. J., and Keerthi, S. S. (2005). An improved conjugate gradient scheme to the solution of least squares SVM. *IEEE Transactions on Neural Networks*, 16(2):498–501.

Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *NIPS*, pages 625–632. MIT Press.

Cortes, C., Mohri, M., and Rastogi, A. (2007). Magnitude-preserving ranking algorithms. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 169–176, New York, NY, USA. ACM.

Dekel, O., Manning, C., and Singer, Y. (2004). Log-linear models for label ranking. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Frank, E. and Hall, M. (2001). A simple approach to ordinal classification. In *Proceedings of the 12th European Conference on Machine Learning*, pages 145–156, London, UK. Springer-Verlag.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Fung, G. and Mangasarian, O. L. (2001). Proximal Support Vector Machine classifiers. In Provost, F. and Srikant, R., editors, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM Press.

Fürnkranz, J. and Hüllermeier, E. (2003). Pairwise preference learning and ranking. In Lavrac, N., Gamberger, D., Todorovski, L., and Blockeel, H., editors, *Proceedings of the 14th European Conference on Machine Learning, Lecture Notes in Computer Science*, volume 2837, pages 145–156. Springer.

Fürnkranz, J. and Hüllermeier, E. (2005). Preference learning. *Künstliche Intelligenz*, 19(1):60–61.

Gärtner, T. (2002). Exponential and geometric kernels for graphs. In *NIPS Workshop on Unreal Data: Principles of Modeling Nonvectorial Data.*

Gärtner, T., Flach, P. A., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In Schölkopf, B. and Warmuth, M. K., editors, *16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop (COLT-2003), Lecture Notes in Computer Science*, volume 2777, pages 129–143. Springer.

Ginter, F., Boberg, J., Järvinen, J., and Salakoski, T. (2004). New techniques for disambiguation in natural language and their application to biological text. *Journal of Machine Learning Research*, 5:605–621.

Har-Peled, S., Roth, D., and Zimak, D. (2002). Constraint classification: A new approach to multiclass classification. In *ALT '02: Proceedings of the 13th International Conference on Algorithmic Learning Theory*, pages 365–379, London, UK. Springer-Verlag.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz.

Herbrich, R. (2002). *Learning kernel classifiers: theory and algorithms.* MIT Press.

Herbrich, R., Graepel, T., and Obermayer, K. (1999). Support vector learning for ordinal regression. In *Proceedings of the 9th International Conference on Articial Neural Networks*, pages 97–102, London. Institute of Electrical Engineers.

Hubbard, T. J. P., Murzin, A. G., Brenner, S. E., and Chothia, C. (1997). SCOP: a structural classification of proteins database. *Nucleic Acids Research*, 25(1):236–239.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2):95–114.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of the 10th European Conference on Machine Learning, Lecture Notes in Computer Science*, volume 1398, pages 137–142, Heidelberg. Springer-Verlag.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, New York, NY, USA.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *International Conference on Machine Learning (ICML)*, pages 377–384.

Joachims, T. (2006). Training linear SVMs in linear time. In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, New York, NY, USA. ACM.

Kashima, H. and Inokuchi, A. (2002). Kernels for graph classification. In *ICDM Workshop on Active Mining*.

Kashima, H., Tsuda, K., and Inokuchi, A. (2004). *Kernels for graphs*, pages 155–170. MIT Press, Cambridge, MA, USA.

Kendall, M. G. (1970). *Rank correlation methods*. Griffin, London.

Kondor, R. I. and Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. In *ICML '02: Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Leslie, C., Eskin, E., and Noble, W. S. (2002a). The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575.

Leslie, C. and Kuang, R. (2004). Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research*, 5:1435–1455.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002b). The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575.

Liao, L. and Noble, W. S. (2003). Combining pairwise sequence similarity and Support Vector Machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, 10(6):857–868.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., and Müler, K. R. (1999). Fisher discriminant analysis with kernels. In *IEEE Workshop on Neural Networks for Signal Processing*.

Mitchell, T. M. (2006). The discipline of machine learning. Technical Report CMU-ML-06-108, Carnegie Mellon University.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., and Salakoski, T. (2007). Learning to rank with pairwise regularized least-squares. In Joachims, T., Li, H., Liu, T., and Zhai, C., editors, *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, pages 27–33.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., and Salakoski, T. (2008). Regularized least-squares for learning non-transitive preferences between strategies. In Raiko, T., Haikonen, P., and Vyrynen, J., editors, *Proceedings of the 13th Finnish Artificial Intelligence Conference (STeP 2008)*, pages 94–98. IOS Press.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., and Boberg, J. (2009). An efficient algorithm for learning to rank from preference graphs. *Machine Learning*, 75(1):129–165.

Pahikkala, T., Tsivtsivadze, E., Boberg, J., and Salakoski, T. (2006). Graph kernels versus graph representations: a case study in parse ranking. In Gärtner, T., Garriga, G. C., and Meinl, T., editors, *Proceedings of the ECML/PKDD'06 Workshop on Mining and Learning with Graphs (MLG'06)*.

Park, S.-H. and Fürnkranz, J. (2007). Efficient pairwise classification. In Kok, J. N., Koronacki, J., de Mántaras, R. L., Matwin, S., Mladenic, D., and Skowron, A., editors, *Proceedings of the 18th European Conference on Machine Learning, Lecture Notes in Computer Science*, volume 4701, pages 658–665. Springer.

Poggio, T. and Smale, S. (2003). The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50(5):537–544.

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8:50.

Pyysalo, S., Ginter, F., Pahikkala, T., Boberg, J., Järvinen, J., Salakoski, T., and Koivula, J. (2004). Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions. In Collier, N., Ruch, P., and Nazarenko, A., editors, *Proceedings of the JNLPBA Workshop at COLING'04, Geneva*, pages 15–21.

Quinonero-Candela, J. and Rasmussen, C. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press.

Rifkin, R., Yeo, G., and Poggio, T. (2003). Regularized least-squares classification. In Suykens, J., Horvath, G., Basu, S., Micchelli, C., and Vandewalle, J., editors, *Advances in Learning Theory: Methods, Model and Applications*, pages 131–154, Amsterdam. IOS Press.

Rosenberg, D. and Bartlett, P. L. (2007). The Rademacher complexity of co-regularized kernel classes. In Meila, M. and Shen, X., editors, *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 396–403.

Rosipal, R. and Trejo, L. J. (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123.

Rudin, C. (2006). Ranking with a p-norm push. In Lugosi, G. and Simon, H. U., editors, *Proceedings of the 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, Lecture Notes in Computer Science*, volume 4005, pages 589–604. Springer.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *ICML '98: Proceedings of the 15th International Conference on Machine Learning*, pages 515–521, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In Helmbold, D. and Williamson, R., editors, *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, pages 416–426, Berlin, Germany. Springer-Verlag.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: Support Vector Machines, regularization, optimization, and beyond.* MIT Press, Cambridge, MA, USA.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA.

Sindhwani, V., Niyogi, P., and Belkin, M. (2005). A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML Workshop on Learning with Multiple Views.*

Sindhwani, V. and Rosenberg, D. (2008). An RKHS for multi-view learning and manifold co-regularization. In McCallum, A. and Roweis, S., editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 976–983, Helsinki, Finland. Omnipress.

Sleator, D. D. and Temperley, D. (1991). Parsing English with a link grammar. Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Suykens, J. A. K. and Vandewalle, J. (1999). Least Squares Support Vector Machine classifiers. *Neural Process. Lett.*, 9(3):293–300.

Thrun, S. and Pratt, L., editors (1998). *Learning to learn.* Kluwer Academic Publishers, Norwell, MA, USA.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of ill-posed problems.* V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

Tsivtsivadze, E., Boberg, J., and Salakoski, T. (2007). Locality kernels for protein classification. In Giancarlo, R. and Hannenhalli, S., editors, *Proceedings of the 7th International Workshop on Algorithms in Bioinformatics*, pages 2–11, Philadelphia, PA, USA. Springer.

Tsivtsivadze, E., Gieseke, F., Pahikkala, T., Boberg, J., and Salakoski, T. (2008a). Learning preferences with co-regularized least squares. In Hüllermeier, E. and Fürnkranz, J., editors, *Proceedings of the ECML/PKDD Workshop on Preference Learning*, pages 52–66, Antwerp, Belgium.

Tsivtsivadze, E., Pahikkala, T., Airola, A., Boberg, J., and Salakoski, T. (2008b). A sparse regularized least-squares preference learning algorithm. In Holst, A., Kreuger, P., and Funk, P., editors, *Proceedings of the 10th Scandinavian Conference on Artificial Intelligence*, volume 173, pages 76–83. IOS Press.

Tsivtsivadze, E., Pahikkala, T., Boberg, J., and Salakoski, T. (2006). Locality-convolution kernel and its application to dependency parse ranking. In Ali, M. and Dapoigny, R., editors, *Proceeedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Lecture Notes in Computer Science*, volume 4031, pages 610–618. Springer.

Tsivtsivadze, E., Pahikkala, T., Boberg, J., and Salakoski, T. (2008c). Kernels for text analysis. *Book chapter in Advances of Computational Intelligence in Industrial Systems*, 116:81–97.

Tsivtsivadze, E., Pahikkala, T., Boberg, J., and Salakoski, T. (2008d). Locality kernels for sequential data and their applications to parse ranking. *Applied Intelligence*, (Online First).

Tsivtsivadze, E., Pahikkala, T., Pyysalo, S., Boberg, J., Mylläri, A., and Salakoski, T. (2005). Regularized least-squares for parse ranking. In Famili, A. F., Kok, J. N., Peña, J. M., Siebes, A., and Feelders, A. J., editors, *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, pages 464–474. Springer.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York.

Vincent, P. and Bengio, Y. (2002). Kernel matching pursuit. *Machine Learning*, 48(1-3):165–187.

Watkins, C. (1999). Dynamic alignment kernels. Technical Report CSD-TR-98-11, UL Royal Holloway.

Xia, F., Liu, T., Wang, J., Zhang, W., and Li, H. (2008). Listwise approach to learning to rank - theory and algorithm. In McCallum, A. and Roweis, S., editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 1192–1199. Omnipress.

Yu, K. and Tresp, V. (2005). Learning to learn and collaborative filtering. In *Proceedings of the Neural Information Processing Systems Workshop on Inductive Transfer: 10 Years Later, (NIPS 2005)*.

Yu, S., Yu, K., Tresp, V., and Kriegel, H. (2006). Collaborative ordinal regression. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pages 1089–1096, New York, NY, USA. ACM.

Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lengauer, T., and Muller, K. (2000). Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807.

Publication Reprints

Paper I

Locality kernels for sequential data and their applications to parse ranking

Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T. (2008). Applied Intelligence. (Online First).

Paper II

Kernels for text analysis

Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T. (2008). Book chapter in Advances of Computational Intelligence in Industrial Systems, volume 116: 81–97.

Paper III

Locality kernels for protein classification

Tsivtsivadze, E., Boberg, J., Salakoski, T. (2007). In Giancarlo, R., Hannenhalli, S., eds., Proceedings of the 7th International Workshop on Algorithms in Bioinformatics, pages 2–11, Philadelphia, USA. Springer.

Paper IV

An efficient algorithm for learning to rank from preference graphs

Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., Boberg, J. (2009). Machine Learning, volume 75(1), pages 129–165.

Paper V

A sparse regularized least-squares preference learning algorithm

Tsivtsivadze, E., Pahikkala, T., Airola, A., Boberg, J., Salakoski, T. (2008). In Holst, A., Kreuger, P., Funk, P., eds., Proceedings of the 10th Scandinavian Conference on Artificial Intelligence, volume 173, pages 76–83, IOS Press.

Paper VI

Learning preferences with co-regularized least squares

Tsivtsivadze, E., Gieseke, F., Pahikkala, T., Boberg, J., Salakoski, T. (2008).
In Hüllermeier, E., Fürnkranz, J., eds., Proceedings of the ECML/PKDD Workshop on Preference Learning, pages 52–66, Antwerp, Belgium.

ISBN 978-952-12-2277-1
ISSN 1239-1883

Evgeni Tsivtsivadze

Learning Preferences with Kernel-Based Methods

Evgeni Tsivtsivadze

Learning Preferences with Kernel-Based Methods