MULTIPLE KERNEL LEARNING METHODS AND THEIR APPLICATION IN YEAST PROTEIN SUBCELLULAR LOCALIZATION PREDICTION

Kristina Spirovska
Faculty of Computer Science and Engineering
Skopje, Republic of Macedonia

Ana Madevska Bogdanova, Ph.D
Faculty of Computer Science and Engineering
Skopje, Republic of Macedonia

ABSTRACT

Kernel methods are becoming more and more popular technique for solving machine learning problems. Recent advances in the field of Multiple Kernel Learning (MKL) have highlighted MKL as an attractive tool that can be applied in many supervised learning tasks. During the past decade, it has been shown that classifiers that use combinations of multiple kernels instead of classical single kernel-based ones attain significantly better results in certain problems.
We give an overview of the existing multiple kernel learning methods and present experimental results of their application in the bioinformatics domain.

Keywords — Multiple Kernel Learning (MKL), kernel methods, kernel function, SVM

I. INTRODUCTION

Kernel methods represent a family of algorithms used in pattern analysis. Support Vector Machines (SVMs), as part of this family, are basic tools in machine learning field especially for supervised learning problems such as classification. They are applied in diverse areas ranging from vision to bioinformatics or natural language processing.
The success of SVMs in these areas is often dependent on the choice of a good kernel – the main ingredient in all kernel methods. Kernels provide a general framework for data representation. [1]

A. Data representation

In order to apply a kernel method to a specific problem, we first have to find a way to represent the data. But data formats can vary. We may have numerical or textual data, graphs, trees, interactions, etc.
Let $S=(x_1, x_2,\ldots,x_n)$ be our data set which contain n objects that need to be classified. Each object comes from a set X, $x_i \in X$, i=1..n. This X set could be a set of proteins whose function needs to be predicted, set of users of some products which need to be analysed, set of texts that need to be classified, set of musical files that need to be categorized or even set of images that need to be analysed. Next step will be finding a representation for every object from S. Formally, this means that a representation $\phi(x) \in F$ is defined for each object $x \in X$. The data set S is then represented as the set of individual object representations $\phi(S)=(\phi(x_1),\ldots,\phi(x_n))$ and afterwards we design an algorithm to process the those data. This implies that we have to design different algorithms for processing data from different problems. [1, 7]

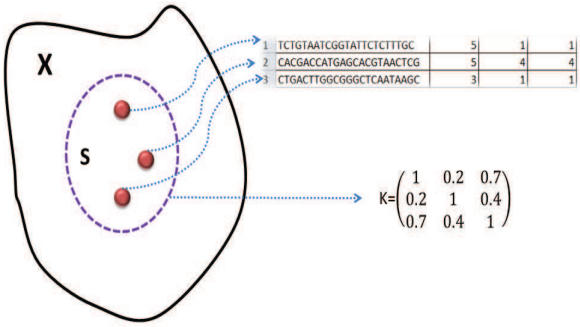

Figure 1: Two different ways of representing objects (DNA sequences and some of their characteristics), a classical way and a kernel method way.

Here we come to the kernel methods, who offer us an answer to this variety problem. Data are not represented as individual objects anymore, but through a set of pairwise comparisons.

B. Kernel methods

Kernel methods work by embedding data objects into vector space *F*, called feature space, and searching for linear relations in such a space. This embedding is defined implicitly, by specifying as inner product for the feature space via a kernel function:

$$K(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle \quad (1)$$

where $\Phi(x_1)$ and $\Phi(x_2)$ are the embedding of data items x_1 and x_2. We no more have the need to explicitly represent the mapping Φ, nor we need to know the nature of the feature space. All we need is evaluation of the kernel function which is often much easier than computing the coordinates of the points explicitly. The product of that evaluation is a kernel matrix K with good mathematical properties: it's symmetric, positive and semidefinite. So the key reason of success of kernel methods is the fact that kernel matrices take relationships that are implicit in the data and make them explicit, so that it is easier to detect patterns. Each kernel function thus extracts a specific type of information from a given dataset, thereby providing a partial description or view of the data. We can see a kernel matrix as a matrix of similarity measures between pairs of data objects. Doing that, it allows one to incorporate prior knowledge of the problem domain. What is of great importance here is the fact that the kernel contains all of the information about the relative positions of the input objects in the feature space and the actual learning algorithm is based only on the kernel function. So, we don't need to explicitly use the feature space in further

calculation. The training data only enter the algorithm through their entries in the kernel matrix, and never through their individual attributes. [1, 6]

II. MULTIPLE KERNEL LEARNING

Multiple kernel learning (MKL) has been pioneered by Lanckriet et al. [9] as an extension of single kernel SVM [8] to incorporate multiple kernels in classification. As we mentioned earlier, each kernel function provides a partial description or view of the data. If we want to combine more views of the data, we need to make a combination of several kernels.

In [2] it is given a taxonomy and review of several multiple kernel learning algorithms. They made a meaningful categorization of existing MKL methods identifying six key properties (enlisted by Fig.2).

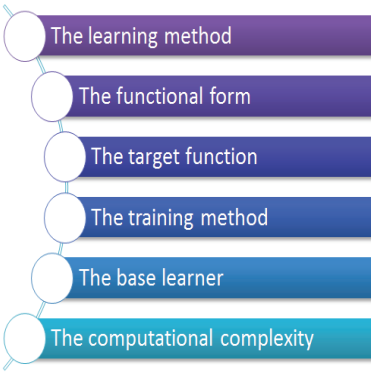

Figure 2: Key properties of MKL methods defined by [2].

A. *The learning method*

The existing MKL algorithms use different learning methods for determining the kernel combination. They can be divided into five major categories (shown by Fig. 3)

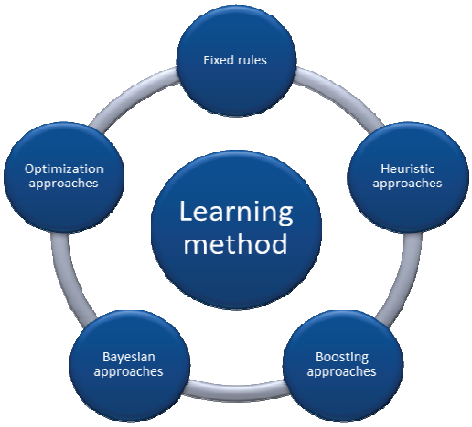

Figure 3: Major categories of MKL learning method defined by [2].

B. *The functional form*

The combination of kernel methods can be done in three different ways, as it's shown on Fig.4.

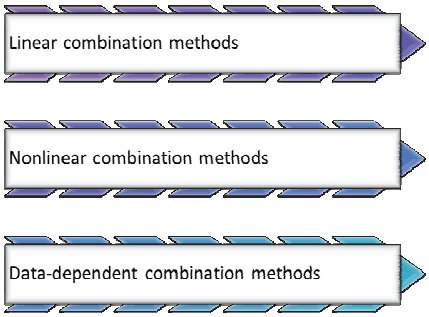

Figure 4: Three categories of functional form considering the way of combining of kernels according to [2].

Linear combination methods are most popular and they can be implemented as unweighted sum:

$$k(x_i, x_j) = \sum_{m=1}^{N} k_m(x_i^m, x_j^m) \qquad (2)$$

and weighted sum:

$$k_\alpha(x_i, x_j) = \sum_{m=1}^{N} \alpha_m k_m(x_i^m, x_j^m) \qquad (3)$$

where α denotes the kernel weights and N number of kernels. Nonlinear methods use nonlinear functions of kernels, like multiplication and exponentiation, and data-dependent combination methods assign specific kernel weights for each data instance identifying local distributions in the data.

C. *The target function*

Target functions can be optimized with selection of combination of function parameters. The three basic categories are shown in

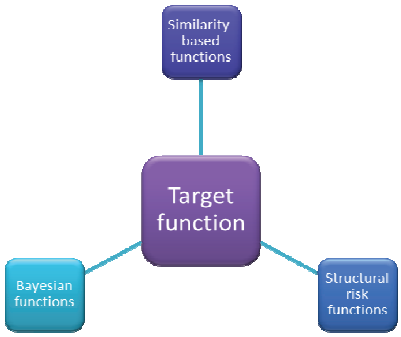

Figure 5: Major categories of MKL learning method defined by [2].

Similarity based functions calculate a similarity metric between the combined kernel matrix from the training data. The similarity between two kernel matrices can be calculated using kernel alignment, Euclidean distance or any other similarity measure.
Structural risk functions follow the structural risk minimization framework and try to minimize the sum of a regularization term that corresponds to the model complexity and an error term that corresponds to the system measure. Structural risk function can use $l1$-norm, $l2$-norm or a mixed norm on the kernel weights or feature spaces to pick the model parameters.
Bayesian functions measure the quality of the resulting kernel function constructed from candidate kernels using Bayesian formulation.

D. The training method

Existing MKL algorithms can be one-step methods or two-step methods. The former methods calculate the combination function parameters and the parameters of the combined base learner in a single pass.
Two-step methods use an iterative approach where at each iteration, first we update the combination function parameters while fixing the base learner parameters, and afterwards we update the parameters of the combined base learner, while fixing the combination function parameters.

E. The base learner

Every kernel-based learning algorithm can be transformed into an MKL algorithm. The most commonly used algorithms are: SVM and Support Vector Regression (SVR). Other popular methods are Kernel Fisher analysis (KFDA), Regularized Kernel Discriminant analysis (RKDA) and Kernel Ridge Regression (KRR).

F. The computational complexity

The computational complexity of MKL algorithm mainly depends on its training method and the computational complexity of its base learner.

III. Related works

During the past decade there have been a lot of advances in the field of multiple kernel learning. In 2004 Lanckriet et al. proposed a kernel-based framework to combine multiple types of data and successfully predicted the functional classes of yeast protein using an extended Support vector machine (SVM) algorithm. There are a lot of improvements in the learning algorithms like algorithm SimpleMKL proposed in [10] or the methods in [11, 14, 15]. Also there are lots of specialized applications of MKL in the field of Bioinformatics [12] used for solving specific problems like gene prioritization [13, 1]

IV. Experiments and results

One of the major goals in proteomic is functional annotation of unknown proteins. Protein's function is usually related to its subcellular localization, so predicting the subcellular localization of a protein can give one a key insight in what that protein function is. The machine learning offers us methods and tools that make that prediction more and more accurate. One of those methods is classification based on multiple kernel learning.

A. Shogun- A Large Scale Machine Learning Toolbox

Shogun is a machine learning toolbox, whose focus is on large scale kernel methods and especially on SVM.
The toolbox not only provides efficient implementations of the most common kernels, like the Linear, Polynomial, Gaussian and Sigmoid Kernel but also comes with a number of recent string kernels as e.g. the Locality Improved, Fischer, TOP, Spectrum, Weighted Degree Kernel (with shifts). Also SHOGUN offers the freedom of working with custom pre-computed kernels. One of its key features is the combined kernel which can be constructed by a weighted linear combination of a number of sub-kernels, each of which not necessarily working on the same domain. An optimal sub-kernel weighting can be learned using Multiple Kernel Learning. Currently SVM 2-class classification and regression problems can be dealt with. [16]
We use this machine learning toolbox in order to predict the protein subcellular localization in yeast with MKL.

B. Dataset description

For the purpose of the experiments in predicting the subcellular localization of proteins, we are using Yeast database from the UCI ML Repository. This dataset has 1484 records. Each record has nine feature values and one class value. These features are for signal sequence recognition such as transmembrane segments, mitochondrial proteins, endoplasmic reticulum (ER) recognition, peroxisomal protein recognition, vacuolar protein recognition, and nuclear protein recognition. The attributes of the Yeast database records are described in Table 1.

Table 1: Description and meaning of Yeast dataset attributes.

Attribute name	Meaning
Sequence name	Accession number for the SWISS-PROT data
Mcg	McGeoch's method for signal sequence recognition
Gvh	von Heijne's method for signal sequence recognition
Alm	Score of the ALOM membrane spanning region prediction program.
Mit	Score of discriminant analysis of the amino acid content of non-mitochondrial proteins
Erl	Presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute.

pox	Peroxisomal targeting signal in the C-terminus.
vac	Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins.
nuc	Score of discriminant analysis of nuclear localization signals

There are 10 localization patterns within a yeast cell, which defines the class attribute:

- cytosolic (CYT)
- endoplasmic reticulum lumen (ERL)
- extracellular (EXC)
- membrane protein with signal cleaved (ME1)
- membrane protein with signal uncleaved (ME2)
- membrane protein without N-terminal signal (ME3)
- mitochondrial (MIT)
- nuclear (NUC)
- peroxisomal (POX)
- vacuolar (VAC)

C. *Results*

In this section, we describe the results of prediction accuracy with single kernel SVM compared to multiple kernel SVM. The experiments were repeated with several kernels (linear, polynomial, Gaussian and RBF kernels and their combination). The data set was firstly divided into training and testing set using 10-fold method for cross validation. Each dataset is classified with SVM using previously mentioned kernels and their combination.
From Table 3 we can see that combination of multiple kernels gives us more precise prediction than the prediction made using single kernels.

V. FUTURE WORKS

As we mentioned earlier, the main ingredient in the kernel methods is the kernel function. If a person is well introduced in the nature of the problem and the data, he can propose a similarity function that can be used to calculate the kernel matrix. But sometimes we want to classify easily, without a kernel function proposal or any human interference. So the next step in our research is creating a generic system for classification based on multiple kernel learning principles.

VI. CONCLUSION

This paper presented a multiple kernel learning- based methods as very efficient techniques in supervised learning. Its purpose was to give taxonomy and review of several multiple kernel learning algorithms and to highlight their differences and similarities. Also it presented an experiment on real data set for better illustration and comparison of classification using a single kernel SVM and a combination of multiple kernel SVMs.
We can see that overall, using multiple kernels instead of a single one is a useful and promising technique.

REFERENCES

[1] Shi Yu, Leon-Charles Tranchevent, Bart De Moor, Yves Moreau, "Kernel-based Data Fusion for Machine Learning- Methods and Applications in Bioinformatics and Text Mining", *Springer* 2011

[2] Mehmet Gonen, Ethem Alpaydın,"Multiple Kernel Learning Algorithms", *Journal of Machine Learning Research 12* (pp. 2211-2268), 2011

[3]B. Scholkopf and A. J. Smola.," Learning with Kernels".

MIT Press, Cambridge, MA, 2002

[4] Francis R. Bach, Gert R. G. Lanckriet, Michael I. Jordan,"Multiple Kernel Learning, Conic Duality, and the SMO Algorithm", *Proceedings of the International Conference on Machine Learning* (pp. 6–13), 2004

[5] Lanckriet GRG, De Bie T, Cristianini N, Jordan MI, Noble WS, "A statistical framework for genomic data fusion.", *Bioinformatics* 2004, 20:2626-2635

[6] Gert R. G. Lanckriet, Minghua Deng, Nello Cristianini, Michael I. Jordan, William Stafford Noble, "Kernel-based data fusion and its application to protein function prediction in Yeast. ", *Proceedings of the Pacific Symposium on Biocomputing*, 2004

[7] Jean-Philippe Vert, Koji Tsuda, Bernhard Scholkopf, "A primer on kernel methods",2004

[8] http://www.support-vector-machines.org/ [online resource]

[9] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, Michael I. Jordan, "Learning the kernel matrix with semidefinite programming". *Proceedings of the 19th International Conference on Machine Learning,* 2002.

[10] Alain Rakotomamonjy, Francis R. Bach, Stephane Canu, Yves Grandvalet,"SimpleMKL", *Journal of Machine Learning Research 9* (pp. 2491-2521), 2009

[11] Francis R. Bach, "Consistency of the group Lasso and multiple kernel learning", *Journal of Machine Learning Research* (pp. 1179–1225), 2008.

[12] Shi Yu, Tillmann Falck, Anneleen Daemen, Leon-Charles Tranchevent, Johan AK Suykens, Bart De Moor, Yves Moreau, "L2-norm multiple kernel learning and its application to biomedical data fusion", Yu et al. *BMC Bioinformatics* 2010, 11:309

[13] De Bie T, Tranchevent LC, Van Oeffelen L, Moreau Y, "*Kernel-based data fusion for gene prioritization*", *Bioinformatics* 2007, 23:i125-i132.

[14] Kloft M, Brefeld U, Sonnenburg S, Laskov P, Müller K, Zien A: "Efficient and Accurate Lp-norm Multiple Kernel Learning", *Advances in Neural Information Processing Systems* 22 2009

[15] Kowalski M, Szafranski M, Ralaivola L: Multiple indefinite kernel learning with mixed norm regularization*" Proc of the 26th International Conference of Machine Learning* 2009

[16] http://www.shogun-toolbox.org/doc/en/current/index.html [online resource]