Kernel Methods for Predictive Sequence Analysis

Gunnar Rätsch[a] and Cheng Soon Ong[a,b]
[a] Friedrich Miescher Laboratory, Max Planck Society, Tübingen
[b] Max Planck Institute for Biological Cybernetics, Tübingen

Abstract

This tutorial is meant for a broad audience: Students, researchers, biologists and computer scientist interested in (a) an overview of general and efficient algorithms for statistical learning used in computational biology, (b) sequence kernels for the problems such as promoter or splice site detection. No specific knowledge will be required since the tutorial is self-contained and most fundamental concepts are introduced during the course.

The slides and additional tutorial material are available at `http://www.fml.mpg.de/raetsch/projects/gcbtutorial`

1 Introduction

The Machine Learning field evolved from the broad field of Artificial Intelligence, which aims to mimic intelligent abilities of humans by machines. In the field of Machine Learning one considers the important question of how to make machines able to "learn". Learning in this context is understood as inductive inference, where one observes examples that represent incomplete information about some "statistical phenomenon". In supervised learning, there is a label associated with each example. It is supposed to be the answer to a question about the example. If the label is discrete, then the task is called classification problem otherwise, for real-valued labels we speak of a regression problem. Based on these examples (including the labels), one is particularly interested in predicting the answer for other cases before they are explicitly observed. Hence, learning is not only a question of remembering but also of generalization to unseen cases.

2 Classification Algorithms

An important task in Machine Learning is classification, also referred to as pattern recognition, where one attempts to build algorithms capable of automatically constructing methods for distinguishing between different exemplars, based on their differentiating patterns. [39] described a pattern as "the opposite of chaos; it is an entity, vaguely defined, that could be given a name". Examples of patterns are human faces, text documents, handwritten letters or digits, EEG signals, and the DNA sequences that may cause a certain disease. More formally, the goal of a (supervised) classification task is to find a functional mapping between the input data X, describing the input pattern, to a class label Y (e.g. 1 or +1), such that $Y = f(X)$. The construction of the mapping is based on so-called training data supplied to the classification algorithm. The aim

is to accurately predict the correct label on unseen data. A pattern (also: "example") is described by its features. These are the characteristics of the examples for a given problem. For instance, in a face recognition task some features could be the color of the eyes or the distance between the eyes. Thus, the input to a pattern recognition task can be viewed as a two-dimensional matrix, whose axes are the examples and the features. Pattern classification tasks are often divided into several sub-tasks:

- Data collection and representation.
- Feature selection and/or feature reduction.
- Classification.

Data collection and representation are mostly problem-specific. Therefore it is difficult to give general statements about this step of the process. In broad terms, one should try to find invariant features, that describe the differences in classes as best as possible. Feature selection and feature reduction attempt to reduce the dimensionality (i.e. the number of features) for the remaining steps of the task. Finally, the classification phase of the process finds the actual mapping between patterns and labels (or targets). In many applications the second step is not essential or is implicitly performed in the third step.

3 Large Margin Classification Algorithms

Machine learning rests upon the theoretical foundation of Statistical Learning Theory [36] which provides conditions and guarantees for good generalization of learning algorithms. Within the last decade, large margin classification techniques have emerged as a practical result of the theory of generalization. Roughly speaking, the margin is the distance of the example to the separation boundary and a large margin classifier generates decision boundaries with large margins to almost all training examples. The two most widely studied classes of large margin classifiers are Support Vector Machines (SVMs) [2] and Boosting [35, 26].

In this tutorial we mainly consider Support Vector Machines and its applications to biological sequence analysis. SVMs work by mapping the training data into a feature space by the aid of a so-called kernel function and then separating the data using a large margin hyperplane (cf. Algorithm 1). Intuitively, the kernel computes a similarity between two given examples. Most commonly used kernel functions are RBF kernels $k(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2)$ and polynomial kernels $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$. The SVM finds a large margin separation between the training examples and previously unseen examples will often be close to the training examples. Hence, the large margin then ensures that these examples are correctly classified as well, i.e., the decision rule generalizes. For so-called positive definite kernels, the optimization problem can be solved efficiently and SVMs have an interpretation as a hyperplane separation in a high dimensional feature space [36, 27]. Support Vector Machines have been used on million dimensional data sets and in other cases with more than ten million examples [32]. Research papers and implementations can be downloaded from the kernel machines web-site `http://www.kernel-machines.org`.

Algorithm 1 Support Vector Machine with regularization parameter C and kernel k

Given labeled sequences $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and a kernel k, the SVM computes a function

$$f(s) = \sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b,$$

where the coefficients α_i are found by solving the optimization problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{w.r.t.} \quad & 0 \leq \alpha_i \leq C \text{ for } i = 1, \ldots, m \\ \text{subject to} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

4 Kernels on Biological Sequence Analysis

One of the most important benefits of using kernel methods is that one can apply the machinery developed for vector space methods directly to structured data. This is done via the "kernel trick" which enables efficient convex optimization methods to be applied to structured data such as sequences, trees and graphs. We will focus on kernels on sequences, and the reader is referred to [28, 3, 7, 6, 10, 12] for more general structures. Our introduction to string kernels will include their definition [13, 14], approaches to speed up kernel computation [30, 38] and several examples of applications to gene sequence comparison [22, 40, 23, 25, 32]. These kernels are the modeling tool that allow us to apply the algorithms presented in the previously on complex data structures arising in computational biology.

In the tutorial we discuss how a practitioner can construct kernels for a particular application with an emphasis to

- the spectrum and weighted degree kernels,
- the combination of known kernels and
- kernel design guidelines.

Finally, we give a brief overview of publicly available software:

- machine learning and optimization toolboxes and
- efficient kernel implementations.

With the course we will provide access to a novel kernel learning toolbox, called shogun, and give a tutorial introduction to the software (available at `http://www.fml.mpg.de/raetsch/projects/shogun`).

.

References

[1] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[2] C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[3] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5:1035–1062, 2004.

[4] CPLEX Optimization Incorporated, Incline Village, Nevada. *Using the CPLEX Callable Library*, 1994.

[5] R.O. Duda, P.E.Hart, and D.G.Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

[6] T. Gärtner, P.A. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M. K. Warmuth, editors, *Proc. Annual Conf. Computational Learning Theory*. Springer, 2003.

[7] D. Haussler. Convolutional kernels on discrete structures. Technical Report UCSC-CRL-99 - 10, Computer Science Department, UC Santa Cruz, 1999.

[8] T.S. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.*, 7:95–114, 2000.

[9] T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

[10] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proc. Intl. Conf. Machine Learning*, Washington, DC, United States, 2003.

[11] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[12] I. R. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. Intl. Conf. Machine Learning*, 2002.

[13] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[14] C. Leslie, E. Eskin, J. Weston, and W.S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4), 2003.

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

[16] L. Liao and W.S. Noble. Combining pairwise sequence similarity and support vector machines. In *Proc. 6th Int. Conf. Computational Molecular Biology*, pages 225–232, 2002.

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

[18] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446, 1909.

[19] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[20] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, pages 276–285, New York, 1997. IEEE.

[21] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

[22] G. Rätsch and S. Sonnenburg. Accurate splice site detection for *Caenorhabditis elegans*. In K. Tsuda B. Schoelkopf and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

[23] G. Rätsch, S. Sonnenburg, and C. Schäfer. Learning interpretable svms for biological sequence classification. *BMC Bioinformatics*, 7(Suppl 1):S9, February 2006.

[24] G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

[25] Gunnar Rätsch, Bettina Hepp, Uta Schulze, and Cheng Soon Ong. PALMA: Perfect alignments using large margin algorithms. In *German Conference on Bioinformatics*, 2006.

[26] R.E. Schapire. *The Design and Analysis of Efficient Learning Algorithms*. PhD thesis, MIT Press, 1992.

[27] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[28] B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.

[29] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 2001.

[30] S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller. New methods for splice-site recognition. In *Proc. International Conference on Artificial Neural Networks*, 2002.

[31] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[32] Sören Sonnenburg, Alexander Zien, and Gunnar Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–480, 2006.

[33] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.R. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14:2397–2414, 2002.

[34] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18:268S–275S, 2002.

[35] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[36] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[37] J.-P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. In K. Tsuda B. Schoelkopf and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

[38] S. V. N. Vishwanathan and A. J. Smola. Fast kernels for string and tree matching. In K. Tsuda, B. Schölkopf, and J.P. Vert, editors, *Kernels and Bioinformatics*, Cambridge, MA, 2004. MIT Press.

[39] W. Watanabe. *Pattern recognition: Human and mechanical*. Wiley, 1985.

[40] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. *BioInformatics*, 16(9):799–807, September 2000.