Incremental Kernel Machines for Protein Remote Homology Detection

Lionel Morgado[1] and Carlos Pereira[1,2]

[1] CISUC - Center for Informatics and Systems of the University of Coimbra
Polo II - Universidade de Coimbra, 3030-290 Coimbra, Portugal
lionel@student.dei.uc.pt, cpereira@dei.uc.pt
[2] ISEC - Instituto Superior de Engenharia de Coimbra
Quinta da Nora, 3030-199 Coimbra, Portugal
cpereira@isec.pt

Abstract. Protein membership prediction is a fundamental task to retrieve information for unknown or unidentified sequences. When support vector machines (SVMs) are associated with the right kernels, this machine learning technique can build state-of-the-art classifiers. However, traditional implementations work in a batch fashion, limiting the application to very large and high dimensional data sets, typical in biology. Incremental SVMs introduce an alternative to batch algorithms, and a good candidate to solve these problems. In this work several experiments are conducted to evaluate the performance of the incremental SVM on remote homology detection using a benchmark data set. The main advantages are shown, opening the possibility to further improve the algorithm in order to achieve even better classifiers.

Keywords: Kernel machines, incremental learning, protein classification.

1 Introduction

A traditional issue in bioinformatics is the classification of protein sequences into functional and structural groups based on sequence similarity. Despite being relatively easy to recognize homologues with high levels of similarity, remote homology detection is a much harder task. Approaches used for remote homology detection can be divided into three main groups: pairwise sequence comparison methods, generative models and discriminative classifiers. The most successful methods for remote homology detection are the discriminative, that combine SVMs [5] with special kernels [19, 20, 21, 22, 23]. The SVM is a powerful machine learning technique that combines high accuracy with good generalization, achieving state-of-the-art results. However, traditional SVM batch implementations present some limitations when faced with the high dimensional and large number of examples available in biology. Incremental SVMs can potentially bring the solutions to these issues, by means of their ability to add new information to an existing, already trained model.

In this work, some experiments are performed with a benchmark data set from SCOP [6] previously used on remote homology detection in order to evaluate the performance of an incremental SVM against the batch algorithm and PSI-BLAST.

E. Corchado et al. (Eds.): HAIS 2009, LNAI 5572, pp. 409–416, 2009.

An overview on incremental SVMs is presented in Section 2. Section 3 presents the description of the spectrum, mismatch and profile kernels which have been used, and Section 4 presents the experiments and results analysis. Final conclusions and reference to future work are given in the last Section.

2 Incremental Kernel Machines

Nowadays, the advances in technology allow collecting enormous amounts of data and joining it in very large data sets. Computational biology is one of such fields where the millions of available examples can also be characterized by very high and variable dimensionality. However, because traditional SVMs use all data in a batch train, both the models and algorithms complexity are overlapping the computational capacities available, limiting the application to this field. Since less information usually implies simpler models and lower memory requirements, reducing the number of train instances and the dimensionality of the data have been both explored approaches. By common sense, the easiest way to decrease the processing burden is to train only over one smaller set with randomly chosen examples. However the probability of excluding important information with this methodology is very high. A larger training set also represents an advantage, since the extra information can contribute to create more accurate models. Therefore, it is important to analyse every individual example, at least briefly. On the other hand, our knowledge in proteomics and genomics is constantly changing, taking repositories to suffer considerable modifications in relatively short periods of time, that demand frequent time consuming actualisations of the discriminative models. Considering these facts, a SVM that builds models step by step in an incremental/decremental fashion using a smaller number of instances each time should be a reality in computational biology.

The first incremental method proposed takes under consideration that the SVM solution only depends on the support vectors, therefore retraining a model consecutively in new blocks of data and the support vectors obtained from previous training sessions will yield the same result as training with all available points at once, because the support vectors are preserved along the process [7]. The exact formulation of incremental SVM learning was presented some years later [8], and brought the possibility to decrement or “unlearn” a model. The algorithm was extended to leave-one-out procedures, and adapted in a way to minimize the computational cost of recalculating a new solution when regularization parameter C and kernel parameters are changed [9]. Nevertheless, this algorithm presents some limitations associated to the use of all the already seen examples to get the final exact solution. An alternative that tries to solve this matter is SimpleSVM [10]. The SimpleSVM algorithm extends Poggio’s principles to the soft-margin case and combines it with block training to keep optimality over unconstrained Lagrangian multipliers. SimpleSVM has a good performance on data sets with few support vectors, however for large scale problems, Sequential Minimal Optimisation (SMO) is preferred [11]. SMO breaks the optimization problem down into two-dimensional sub-problems that may be solved analytically, eliminating the need for a numerical optimization algorithm such as conjugate gradient methods, this way shortening the processing time and the computational burden.

It is precisely from SMO that LASVM is derived [12]**.** This algorithm is an online kernel classifier based on the soft-margin SVM, that incrementally builds a discriminative model by adding or removing support vectors iteratively, using two different points each turn. New support vectors come from a direction search called PROCESS that involves at least one non support vector from the current kernel expansion, while REPROCESS can eliminate support vectors by changing to zero the weight coefficients of one or both the points analysed. In order to incrementally build the final discriminative model, each iteration demands storing a set of all the potential support vectors, Lagrange coefficients of the kernel expansion and the partial derivatives. A significant difference that arises when comparing LASVM to SimpleSVM is that the former doesn't seek the precise solution of the QP problem in each step but instead an approximation that improves the dual function. So, a finishing step similar to a simplified SMO may be necessary to improve performance on noisy data sets.

In fact, real-life problems are dynamic/online rather than static/batch, because information is prone to change. Some work has been developed around incremental/online classification [13, 14, 15] and regression problems [16, 17, 18], but a lot of research is still needed, in particular for biological data analysis.

3 Kernels for Proteins

Several kernels have been proposed for protein classification [19, 20, 21, 22, 23, 27]. The kernel function aims emphasizing important biological information while converting variable length strings that represent amino acids or nucleotides, into numeric fixed size feature vectors. This mapping is mandatory in the sense that the learning machine demands feature vectors with a fixed number of attributes and largely affect the final accuracy and complexity of the learning machine.

3.1 The Spectrum Kernel

The spectrum kernel [20] is a string kernel type that acts over an input space composed of all finite sequences of characters from an alphabet A with l elements, and maps it to a feature space with l^k dimensions that represent all the possible *k*-length contiguous subsequences that may be contained in a protein.
The feature map for sequence *x* is given by:

$$\Phi_k(x) = (\phi_\alpha(x))_{\alpha \in A^k}\ , \tag{1}$$

where $\phi_\alpha(x)$ contains the number of times subsequence α occurs in x .

Taking into account the definition of kernel, the *k*-spectrum kernel comes from the dot product:

$$K_k(x,y) = \langle \Phi_k(x), \Phi_k(y) \rangle \tag{2}$$

3.2 The Mismatch Kernel

The mismatch kernel [21] is an extension of the spectrum kernel. It measures sequence similarity based on shared occurrences of fixed-length patterns in the data, allowing mutations between them.

A *k*-length subsequence α of aminoacids can be described in a (k,m)-neighborhood $N_{(k,m)}(\alpha)$ defined by all the *k*-length subsequences β that differ from the original α by at most m mismatches.

The entry space uses the feature map:

$$\Phi_{(k,m)}(\alpha) = (\phi_\beta(\alpha))_{\alpha \in A^k}, \tag{3}$$

where $\phi_\beta(\alpha)$ contains the number of occurrences and where β belongs to $N_{(k,m)}(\alpha)$.

The mismatch kernel is given by:

$$K_{(k,m)}(x,y) = \langle \Phi_{(k,m)}(x), \Phi_{(k,m)}(y) \rangle, \tag{4}$$

and is equivalent to the spectrum kernel when no mismatches are allowed $(m = 0)$.

3.3 The Profile Kernel

The profile kernel [23] doesn't take as input the protein itself but rather profiles $P(x)$ of a sequence *x*. Profiles are statistically estimated from close homologues stored in a large sequence database, and can be defined as:

$$P(x) = \{p_i(a), a \in A\}_{i=1}^N, \tag{5}$$

with p_i being the emission probability of aminoacid a in position i and $\sum_{a \in A} p_i(a) = 1$ for every position i. Similarly to the mismatch kernel, mutations are considered. A significant difference is that here the probability of a mutation to occur is measured and only some cases are allowed, considering a score dependent on the position of the substring in the protein chain and a given threshold.

4 Experiments

Remote homology detection was used to evaluate the performance, structure complexity and processing time of incremental SVM algorithms comparatively to batch implementations. The following algorithms were applied: LIBSVM [24] (version 2.85) as the batch SVM, the incremental algorithm LASVM and PSI-BLAST, the most used method by the scientific community.

Due to the very high dimensionality of the feature space generated using string kernels, these were pre-computed in order to avoid computation problems. This methodology also allows planning computation in a way to avoid calculus redundancy. For the implementation, it was necessary to adapt LASVM to accept this kind of data as input.

A 2.4 GHz Intel Core 2 Quad CPU desktop computer with 4 GB RAM was used. PSI-BLAST was executed under Microsoft Windows XP, LIBSVM models were trained under the same operating system running a MATLAB interface, and LASVM was executed under gOS.

The profiles for the profile kernel were obtained with PSI-BLAST using 2 search rounds.

4.1 Data Set Description

The algorithms were tested with a SCOP benchmark data set previously used on remote homology detection [22]. The data set has 7329 domains and was divided according to 54 families. Remote homology detection is simulated by considering all domains for each family as positive test examples and sequences outside the family but belonging to the same superfamily as positive train examples. Negative examples are from outside the positive sequences fold, and were randomly divided into train and test sets in the same ratio as the positive examples.

To evaluate the quality of the created classifiers receiver operating characteristics (ROC) was used. A ROC curve consists in the plot of the true positives rate as a function of true negatives rate at varying decision thresholds, and expresses the ability of a model to correctly rank examples and separate distinct classes. The area under a ROC curve (AUC), also known as ROC score, is the most used performance measure extracted from ROC. A good model has AUC=1, a random classifier is expressed by an AUC $\approx 0,5$ and the worst case comes when AUC=0.

4.2 Results

The ROC scores (AUC) for the batch SVM, LASVM and PSI-BLAST, are given in Table 1. The kernel notation indicates the length of the subsequences taken under consideration and the number of mismatches allowed (for mismatch kernel) or the threshold value (for profile kernel).

As expected, the SVM with the profile kernel is the one that achieves better results, followed by the mismatch and spectrum kernel. Profile and mismatch kernels create models with even better performance than PSI-BLAST, showing its ability to evidence important biological information based on amino acid sequences alone. This quality is not an exclusive property of the batch algorithms, since LASVM exhibits an identical behaviour, creating models with equal or even superior results for some protein families. It was also verified that processing time is similar when training new models from the beginning with all data points.

The ability of the incremental algorithm to achieve an inferior number of support vectors than LIBSVM (as seen in Figure1), when describing the discriminative decision hyperplane, reveals an important contribution to complexity reduction, making this methodology suitable for large scale problems.

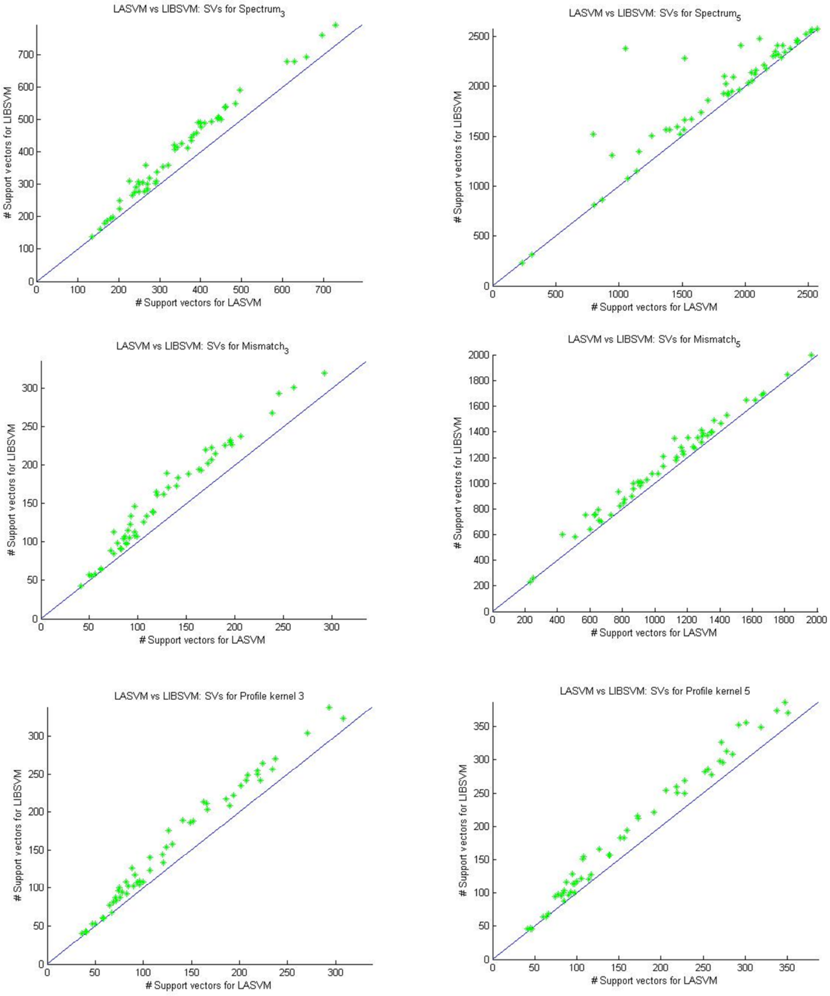

Fig. 1. Support vectors for LASVM plotted against LIBSVM for each family model. Results obtained for spectrum (3), spectrum (5), mismatch (3, 1), mismatch (5, 1), profile (3, 7.5) and profile (5, 7.5) in top left, top right, center left, center right, bottom left and bottom right, respectively.

Table 1. Mean ROC values for all experiments

Algorithm	Kernel	ROC (mean)
PSI-BLAST	-	0.8183
LIBSVM	Spectrum(3)	0.788
	Spectrum(5)	0.720
	Mismatch(3,1)	0.856
	Mismatch(5,1)	0.866
	Profile(3,7.5)	0.89
	Profile(5,7.5)	0.92
LASVM	Spectrum(3)	0.788
	Spectrum(5)	0.699
	Mismatch(3,1)	0.855
	Mismatch(5,1)	0.865
	Profile(3,7.5)	0.89
	Profile(5,7.5)	0.92

5 Conclusions and Future Work

This work proposes incremental SVM algorithms for protein remote homology detection. The presented results show that the incremental formulation, namely LASVM, achieves state-of-the-art results for this kind of task, bringing some advantages over the batch SVM, which by itself can get superior results to the widely accepted PSI-BLAST. The incremental SMO based SVM showed proficiency to generate discriminative models as good as or even better than batch LIBSVM, keeping, for the most families a reduced number of support vectors.

These good results and the potential of the approach encourage the application of the incremental algorithm with different kernels for online classification tasks, and in particular to large biological data sets.

Acknowledgments. This work was supported by FCT – Fundação para a Ciência e a Tecnologia, under Project BIOINK – PTDC/EIA/71770/2006.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: A basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
2. Smith, T.F., Waterman, M.S.: Identification of common molecular sub- Sequences. J. Mol. Biol. 147, 195–197 (1981)
3. Krogh, A., Brown, M., Mian, I., Sjolander, K., Haussler, D.: Hidden markov models in computational biology: Applications to protein modeling. J. Mol. Biol. 235, 1501–1531 (1994)
4. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Research 25, 3389–3402 (1997)
5. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)

6. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: A structural classification of proteins database for the investigation of sequences and structure. J. Mol. Biol. 247, 536–540 (1995)
7. Syed, N.A., Liu, H., Sung, K.K.: Incremental Learning with Support Vector Machines (1999)
8. Cauwenberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. Advances in Neural Information Processing Systems, vol. 13. MIT Press, Cambridge (2001)
9. Diehl, C.P., Cawenberghs, G.: SVM Incremental Learning, Adaptation and Optimization. In: Proceedings of the International Joint Conference on Neural Networks (2003)
10. Vishwanathan, S.V.N., Smola, A.J., Murty, M.N.: SimpleSVM. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington DC (2003)
11. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1999)
12. Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast Kernel Classifiers with Online and Active Learning. Journal of Machine Learning Research (2005)
13. Tax, D.M.J., Laskov, P.: Online SVM Learning: From Classification to Data Description and Back. Neural Networks for Signal Processing (2003)
14. Rüping, S.: Incremental Learning with Support Vector Machines. In: Proceedings of the 2001 IEEE International Conference on Data Mining (2001)
15. Laskov, P., Gehl, C., Krüger, S., Müller, K.: Incremental Support Vector Learning: Analysis, Implementaton and Applications. The Journal of Machine Learning Research 7, 1909–1936 (2006)
16. Martin, M.: On-line Support Vector Machine Regression. In: Proceedings of the 13th European Conference on Machine Learning (2002)
17. Ma, J., Theiler, J., Perkins, S.: Accurate On-line Support Vector Regression. Neural Computation 15, 2683–2703 (2003)
18. Parrella, F.: Online Support Vector Regression – A thesis presented for the degree of Information Science. Department of Information Science, University of Genoa, Italy (2007)
19. Jaakkola, T., Diekhans, M., Haussler, D.: Using the Fisher Kernel Method to Detect Remote Protein Homologies. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (1999)
20. Leslie, C., Eskin, E., Noble, W.: The spectrum kernel: a string kernel for SVM protein classification. In: Pacific Symposium on Biocomputing, vol. 7, pp. 566–575 (2002)
21. Leslie, C., Eskin, E., Weston, J., Noble, W.: Mismatch string kernels for SVM protein classification. Adv. Neural Inf. Process. Syst. 15, 1441–1448 (2002)
22. Weston, J., Leslie, C., Zhou, D., Elisseeff, A., Noble, W.S.: Semi-Supervised Protein Classification using Cluster Kernels. In: NIPS, vol. 17 (2003)
23. Kuang, R., Ie, E., Wang, K., Siddiqi, M., Freund, Y., Leslie, C.: Profile-based string kernels for remote homology detection and motif extraction. In: 3rd International IEEE Computer Society Computational Systems Bioinformatics Conference, Stanford, CA, pp. 152–160. IEEE Computer Society Press, Los Alamitos (2004)
24. Chang, C.C., Lin, C.J.: LIBSVM: a Library for Support Vector Machines (2004), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
25. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
26. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition 30(7), 1145–1159 (1997)
27. Busuttil, S., Abela, J., Pace, G.J.: Support Vector Machines with Profile-Based Kernels for Remote Protein Homology Detection. Genome Informatics 15(2), 191–200 (2004)