Locality Kernels for Protein Classification

Evgeni Tsivtsivadze, Jorma Boberg, and Tapio Salakoski

Turku Centre for Computer Science (TUCS)
Department of Information Technology, University of Turku
Joukahaisenkatu 3-5 B, FIN-20520 Turku, Finland
firstname.lastname@it.utu.fi

Abstract. We propose kernels that take advantage of local correlations in sequential data and present their application to the protein classification problem. Our 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.

1 Introduction

One important task in computational biology is inference of the structure and function of the protein encoded in the genome. The similarity of protein sequences may imply structural and functional similarity. The task of detecting these similarities can be formalized as a classification problem that treats proteins as a set of labeled examples which are in positive class if they belong to the same family and are in negative class otherwise.

Recently, applicability of this discriminative approach for detecting remote protein homologies has been demonstrated by several studies. For example, Jaakkola et al. [1] show that by combining discriminative learning algorithm and Fisher kernel for extraction of the relevant features it is possible to achieve a good performance in protein family recognition. Liao and Noble [2] further improve results presented in [1] by proposing 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.

The methods described in [1] and [2] use an expensive step of generating vector valued features for protein discrimination problems, which increases computational time of the algorithm. The idea to use a simple kernel function that can be efficiently computed and does not depend on any generative model or separate preprocessing step is considered by Leslie et al. in [3]. They show that

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 2–11, 2007.

simple sequence based kernel functions perform surprisingly well compared to other computationally expensive approaches.

In this study, we address the problem of protein sequence classification using the RLS algorithm with locality kernels similar to the one we proposed in [4]. The features used by the locality kernels represent sequences contained in a small window constructed around matching amino acids in the compared proteins. The kernels make use of the range of 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, *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 [5] than that of the spectrum and the mismatch kernels [3,6,7].

Previously, we have shown that the locality-convolution kernel [4] can be successfully applied to parse ranking task in natural language processing. The similarity of the data representation in cases of biological sequence and text, as well as results obtained in this study, suggest that locality kernels can be applied to tasks where local correlations and positional information within the sequence might be important.

The paper is organized as follows. In Section 2, we present overview of the RLS algorithm. In Section 3, we define notions of locality window, positional matching, and present locality kernels. In Section 5, we evaluate the applicability of the locality kernels for the task of protein classification and compare their performance with the spectrum and the mismatch kernels. We conclude this paper in Section 6.

2 Regularized Least-Squares Algorithm

Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)\}$, where $\mathbf{x}_i = (x_1, \ldots, x_n)^{\mathrm{T}}$, $\mathbf{x}_i \in S$ and $y_i \in \{0, 1\}$ be the set of training examples. The target output value y_i is a label value which is either 0, indicating that $\mathbf{x}_i$ does not belong to the class or 1 otherwise. The target output value is predicted by the regularized least-squares (RLS) algorithm [8,9]. We denote a matrix whose rows are $\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_t^{\mathrm{T}}$ as X and a vector of output labels as $\mathbf{y} = (y_1, \ldots y_t)^{\mathrm{T}}$. The RLS algorithm corresponds to solving following optimization problem:

$$\min_{\mathbf{w}} \sum_{i=1}^{t} (y_i - f(\mathbf{x}_i))^2 + \lambda \|\mathbf{w}\|^2, \tag{1}$$

where $f : S \rightarrow \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^n$ is a vector of parameters such that $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$, and $\lambda \in \mathbb{R}_+$ is a regularization parameter that controls the trade-off between fitting the training set accurately and finding the smallest norm for the function f.

Rewriting (1) in matrix form and taking derivative with respect to $\mathbf{w}$, we obtain

$$\mathbf{w} = (X^{\mathrm{T}}X + \lambda I)^{-1}X^{\mathrm{T}}\mathbf{y}, \tag{2}$$

where I denotes identity matrix of dimension $n \times n$. In (2) we must perform matrix inverse in dimension of feature space, that is $n \times n$. However, if the number of features is much larger than the number of training data points, a more efficient way is to perform inverse in the dimension of training examples. In that case, following [9], we present (2) as a linear combination of training data points:

$$\mathbf{w} = \sum_{i=1}^{t} a_i \mathbf{x}_i, \tag{3}$$

where

$$a = (K + \lambda I)^{-1}\mathbf{y} \tag{4}$$

and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel matrix that contains the pairwise similarities of data points computed by a kernel function $k : S \times S \to \mathbb{R}$. Finally, we predict an output of new data point as follows:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{y}^{\mathrm{T}}(K + \lambda I)^{-1}\mathbf{k}, \tag{5}$$

where $k_i = k(\mathbf{x}_i, \mathbf{x})$. Kernel functions are similarity measures of data points in the input space S, and they correspond to the inner product in a feature space H to which the input space data points are mapped. The kernel functions are defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle,$$

where $\Phi : S \to H$. Next we formulate the locality kernel functions that are used with the RLS algorithm for protein classification task.

3 Locality Kernels

There are three key properties of the locality kernels that make them applicable to the task of remote homology detection in the proteins. Firstly, the features used by these kernels contain amino acids that are extracted in the order of their appearance in the protein sequence. Secondly, local correlations within the protein sequence are taken into account by constructing a small window around the matching amino acids. Finally, positional information of the amino acids contained within window is used for similarity evaluation.

Let us consider proteins $\mathbf{p}, \mathbf{q}$ and let $\mathbf{p} = (p_1, \ldots, p_{|\mathbf{p}|})$ and $\mathbf{q} = (q_1, \ldots, q_{|\mathbf{q}|})$ be their amino acid sequences. The similarity of $\mathbf{p}$ and $\mathbf{q}$ is obtained with kernel

$$k(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{|\mathbf{p}|} \sum_{j=1}^{|\mathbf{q}|} \kappa(i, j). \tag{6}$$

By defining κ in the general formulation (6), we obtain different similarity functions between proteins. 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 (6) equals to the number of matching amino acids irrespective of their position in two sequences.

To take into account local correlations within a sequence, we construct small windows of length $2w+1$ around the matching amino acids. In addition we define real valued $(2w+1) \times (2w+1)$ matrix P that we use in the formulation of κ. The positional matrix P stores information about relevance of particular position in compared windows for the similarity evaluation task (see [10] for a related approach). Entries of P contain real valued coefficients that are defined for all possible position pairs within two windows. Below we propose several ways for selecting appropriate P for the task in question.

Let us consider following kernel function:

$$\kappa(i,j) = \delta(p_i, q_j) \sum_{h,l=-w}^{w} [P]_{h,l} \delta(p_{i+h}, q_{j+l}). \tag{7}$$

Note that the rows and the columns of the positional matrix P are indexed from $-w$ to w. Furthermore, we consider amino acids as mismatched when the indices $i+h$ and $j+l$ are not valid, e.g. $i+h < 1$ or $i+h > |\mathbf{p}|$. When we set $P = A$, where A is a matrix whose all elements are ones, we get κ that counts the matching amino acids irrespective of their positions in the two windows. As another alternative, we can construct a function that requires the positions of matching amino acid to be exactly the same. This is obtained by $P = I$, where I denotes the identity matrix. Furthermore, when P is a diagonal matrix whose elements are weights increasing from the boundary to the center of the window, we obtain a kernel that is related to the locality improved kernel proposed in [11]. However, if we do not require strict position matching, but rather penalize matches that have a different position within the windows, we can use a positional similarity matrix whose off-diagonal elements are nonzero and smaller than the diagonal elements. We obtain such a matrix, for example, by

$$[P]_{h,l} = e^{-\frac{(h-l)^2}{2\theta^2}}, \tag{8}$$

where $\theta \geq 0$ is a parameter. The choice of an appropriate κ is a matter closely related to the domain of the study. In Section 5 we show that positional information captured with (7) is useful and improves the classification performance.

When using (7) with different positional matrices in (6), we obtain the kernels which we call the locality kernels. Due to the kernel closure properties and positive semidefiniteness of matrix P, the locality kernels are indeed valid kernel functions. Our kernels could be considered within more general convolution framework described by Haussler [12]. From this point of view, we can distinguish between "structures" and "different decompositions" constructed by our kernels. Informally, we are enumerating all the substructures representing pairs of windows built around the matching amino acids in the proteins and calculating their similarity.

4 Spectrum and Mismatch Kernels

The spectrum kernel introduced in [3] (see also [9]) is very efficient kernel for sequence similarity estimation. It compares two sequences by counting the common contiguous subsequences of length v that are contained in both of them. Thus, the spectrum kernel can be considered as an inner product between vectors containing frequencies of the matching subsequences. For consistency, we present the spectrum and the mismatch kernels within already described framework for the locality kernels. For detailed feature map of these kernels, we refer to [7].

The spectrum kernel is obtained by using

$$\kappa(i,j) = \prod_{l=0}^{v-1} \delta(p_{i+l}, q_{j+l}), \tag{9}$$

in (6).

Leslie et al. [6] also proposed a more sensitive kernel function called the mismatch kernel. The intuition behind this approach is that similarity between two sequences is large if they share many similar subsequences. By restricting number of mismatches to m between the subsequences of length v, the (v,m)-mismatch kernel is obtained by using

$$\kappa(i,j) = \begin{cases} 0, & \text{if } \sum_{l=0}^{v-1} \delta(p_{i+l}, q_{j+l}) < v - m \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

in (6). The spectrum kernel (9) is a special case of the mismatch kernel where $m = 0$. Again, we consider amino acids as mismatched in (9) and (10), when the indices $i+l$ and $j+l$ are not valid, that is, $i+l > |\mathbf{p}|$ or $j+l > |\mathbf{q}|$.

5 Experiments

The experiments to evaluate performance of RLS with the locality kernels, the spectrum kernel, and the (v,m)-mismatch kernel are conducted on the SCOP [5] database. The aim is to classify protein domains into SCOP-superfamilies. We follow the experimental setup and use the dataset described in [2]. For each family, the protein domains within the family are considered positive test examples, and protein domains outside the family but within the same superfamily are considered as positive training examples. Negative examples are taken from outside of the positive sequences' fold and are randomly split into training and testing sets in the same ratio as positive examples. By this setup, we may simulate remote homology detection, because protein sequences belonging to different families but to the same superfamily are considered to be remote homologs in SCOP.

To measure performance of the methods, we use receiver operating characteristics (ROC) scores. The ROC score is the normalized area under a curve (AUC) that represents true positives as a function of false positives for varying classification thresholds [13,14]. When obtaining perfect classification, the ROC score is 1, and the random classification yields score of 0.5.

In Table 1, we present the best found parameters for the locality kernels with different positional matrices P, the spectrum and the (v, m)-mismatch kernels. The best found size of the window for the locality kernel is three ($w = 1$). The spectrum kernel has a parameter v corresponding to the size of subsequence and the mismatch kernel uses v and m, where m is the maximum number of allowed mismatches. The best found parameters for the spectrum and the mismatch kernels correspond to the ones reported in [3,6]. The RLS algorithm has the regularization parameter λ that controls the trade-off between the minimization of the training error and the complexity of the regression function. The results reported below are obtained with the best found combination of the parameters for every method.

The main results of the experiments are summarized in Figure 1. Each curve corresponds to RLS with specific kernel function for remote homology detection. Higher curves reflect more accurate classification performance. Each plotted data point represents the number of the families that have ROC score higher than the corresponding value. We observe that RLS with the position sensitive locality kernel with positional matrix (8) performs significantly better ($p < 0.05$) than RLS with the spectrum or the mismatch kernels. We evaluate statistical significance of the performance differences using Wilcoxon signed-ranks test. The locality kernel using positional matrix $P = I$ and a small window slightly looses to position sensitive locality kernel with matrix (8) in performance, whereas position insensitive locality kernel performs worst of all. Therefore, we do not present these results in Figure 1. We also observe that for the few families that are classified with high scores by all kernels the mismatch kernel is the best, however, for the rest of the families the locality kernel outperforms both the spectrum and the mismatch kernel.

In Figures 2 and 3 we give more detailed performance comparison of the locality, the spectrum and the mismatch kernels. Clearly, the classification

Table 1. The best found parameters used for conducting the experiments

Kernel	Positional matrix	Best parameters	Figures
(7)	$P = A$	$w = 1$	1, 2 and 3
	$[P]_{h,l} = e^{-\frac{(h-l)^2}{2\theta^2}}$	$w = 1, \theta = 0.9$	
	$P = I$	$w = 1$	
(9)		$v = 3$	1 and 2
(10)		$m = 1, v = 6$ and $m = 2, v = 8$	1 and 3

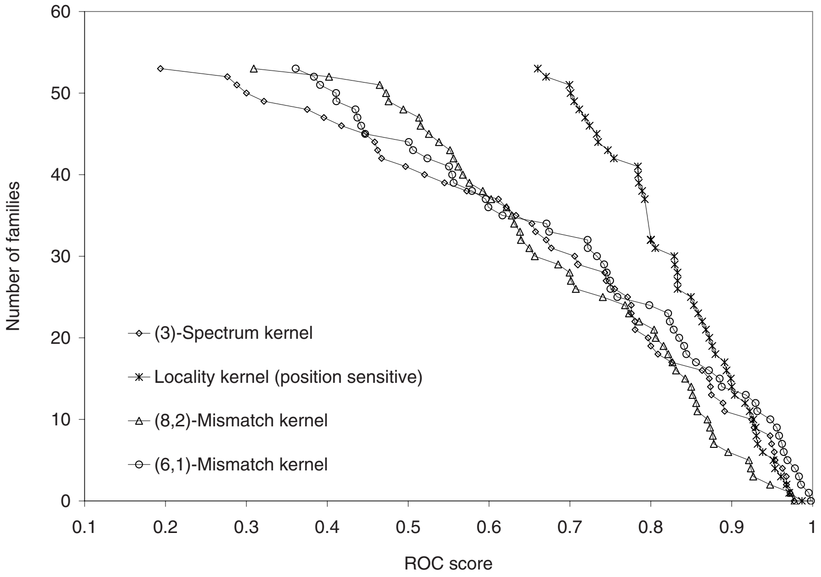

Fig. 1. Performance comparison of RLS with the locality (position sensitive), the spectrum (subsequences of length 3) and the mismatch (subsequences of length 6 and 8, and number of mismatches 1 and 2, respectively) kernels for remote homology detection using 54 families of the SCOP database. Each data point on the curve represents the number of the families having higher ROC score for the method.

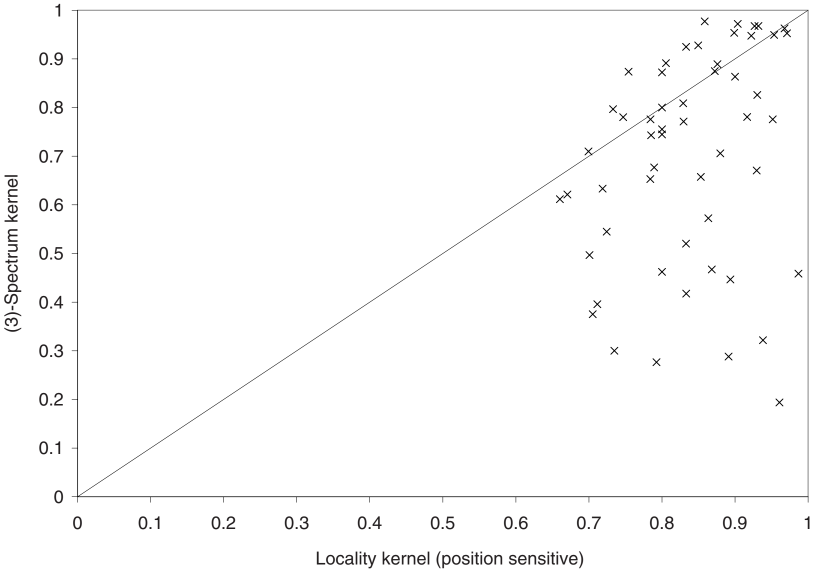

Fig. 2. Family-by-family performance comparison of RLS with the spectrum (subsequences of length 3) and the locality (position sensitive) kernels. The coordinates of each point are ROC scores obtained for one SCOP family.

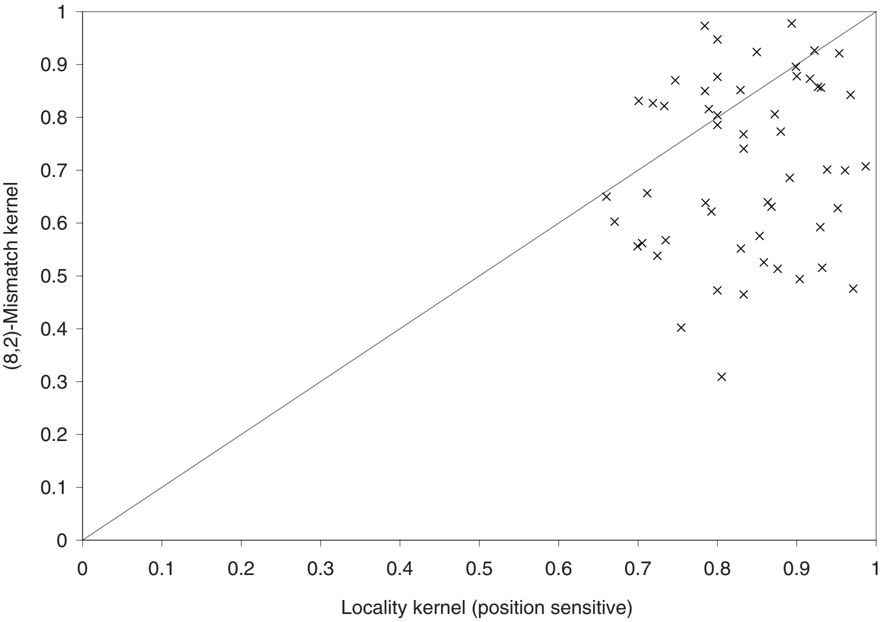

Fig. 3. Family-by-family performance comparison of RLS with the mismatch (subsequences of length 8, number of mismatches 2) and the locality (position sensitive) kernels. The coordinates of each point are ROC scores obtained for one SCOP family.

performance when using the position sensitive locality kernel is better than that of the spectrum and the mismatch kernels. In addition to the conducted experiments, we evaluated performance of the blended spectrum kernel [9], that is all subsequences of sizes from one to v are simultaneously compared, when measuring similarities between the proteins. However, performance of the blended spectrum kernel is not notably better than that of the spectrum kernel and its computation requires more time.

6 Conclusions

In this study, we propose kernels that take advantage of local correlations and positional information in sequential data and present their application to the protein classification problem. The locality kernels measure the protein similarities within a small window constructed around matching amino acids in both sequences. These kernels make use of the range of similarity evaluations within the windows, namely position insensitive matching, position sensitive matching, and strict matching.

We demonstrate that RLS with our locality kernels performs significantly better than RLS with the spectrum or the mismatch kernels in recognition of previously unseen families from the SCOP database. Throughout our experiments we observe that the locality kernels incorporating positional information perform better than the locality kernels that are insensitive to the positions of the amino acids within the windows containing protein subsequences. Although, we do not conduct experiments to compare performance of RLS with the locality kernels to other algorithms, by examining the results reported in [2,3,15], we may suggest that our method performs comparably with some state-of-the-art algorithms used for remote homology detection and protein classification. Moreover, our simple method does not require expensive step of generating vector valued features used in algorithms such as SVM-pairwise or SVM-Fisher.

In the future we plan to cast classification problem of protein sequences as a bipartite ranking task and we aim to obtain better classification performance by maximizing AUC instead of minimizing least squares error.

Acknowledgments

We would like to thank CSC, Finnish IT center for science, for providing us computing resources.

References

1. Jaakkola, T., Diekhans, M., Haussler, D.: A discriminative framework for detecting remote protein homologies. Journal of Computational Biology 7, 95–114 (2000)
2. Liao, L., Noble, W.S.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of Computational Biology 10, 857–868 (2003)
3. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for svm protein classification. In: Pacific Symposium on Biocomputing, pp. 566–575 (2002)
4. Tsivtsivadze, E., Pahikkala, T., Boberg, J., Salakoski, T.: Locality-convolution kernel and its application to dependency parse ranking. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 610–618. Springer, Heidelberg (2006)
5. Hubbard, T.J.P., Murzin, A.G., Brenner, S.E., Chothia, C.: Scop: a structural classification of proteins database. Nucleic Acids Research 25, 236–239 (1997)
6. Leslie, C.S., Eskin, E., Cohen, A., Weston, J., Noble, W.S.: Mismatch string kernels for discriminative protein classification. Bioinformatics 20, 467–476 (2004)
7. Leslie, C., Kuang, R.: Fast string kernels using inexact matching for protein sequences. J. Mach. Learn. Res. 5, 1435–1455 (2004)
8. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Amer. Math. Soc. Notice 50, 537–544 (2003)
9. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, USA (2004)
10. Pahikkala, T., Pyysalo, S., Ginter, F., Boberg, J., Järvinen, J., Salakoski, T.: Kernels incorporating word positional information in natural language disambiguation tasks. In: Russell, I., Markov, Z. (eds.) Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, Menlo Park, Ca., pp. 442–447. AAAI Press, Stanford, California, USA (2005)
11. Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lengauer, T., Muller, K.-R.: Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics 16, 799–807 (2000)

12. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz (1999)
13. Gribskov, M., Robinson, N.L.: Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. Computers & Chemistry 20, 25–33 (1996)
14. Fawcett, T.: Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs (2003)
15. Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., Leslie, C.: Profile-based string kernels for remote homology detection and motif extraction. J. Bioinform. Comput. Biol. 3, 527–550 (2005)