JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 3, 2009

Pp. 457–474
DOI: 10.1089/cmb.2008.0031

Learned Random-Walk Kernels and Empirical-Map Kernels for Protein Sequence Classification

RENQIANG MIN,[1] ANTHONY BONNER,[1] JINGJING LI,[2,3] and ZHAOLEI ZHANG[2,3]

ABSTRACT

Biological sequence classification (such as protein remote homology detection) solely based on sequence data is an important problem in computational biology, especially in the current genomics era, when large amount of sequence data are becoming available. Support vector machines (SVMs) based on mismatch string kernels were previously applied to solve this problem, achieving reasonable success. However, they still perform poorly on difficult protein families. In this paper, we propose two approaches to solve the protein remote homology detection problem: one uses a convex combination of random-walk kernels to approximate the random-walk kernel with the optimal random steps, and the other constructs an empirical-map kernel using a profile kernel. Both resulting kernels make use of a large number of pairwise sequence similarity information and unlabeled data; and have much better prediction performance than the best profile kernel directly derived from protein sequences. On a competitive Structural Classification Of Proteins (SCOP) benchmark dataset, the overall mean ROC_{50} scores on 54 protein families we obtained using both approaches are above 0.90, which significantly outperform previous published results.

Key words: protein classification, support vector machine, random walk kernels.

1. INTRODUCTION

CLASSIFYING BIOLOGICAL SEQUENCES is an important and challenging problem in both computational biology and machine learning. On the biological side, it helps to identify interesting sequence regions and protein domains that are related to a particular biological function; on the computational side, it motivates many novel and efficient new classification approaches specifically for sequence data. This problem has being addressed using different methods, including generative models (e.g., profile HMMs [Baldi et al., 1994; Krogh et al., 1994]), discriminative models (e.g., kernel SVMs [Jaakkola et al., 2000; Leslie et al., 2002, Liao and Noble, 2002]), and graph-based approaches (Tsuda et al., 2005).

[1]Department of Computer Science, University of Toronto, Toronto, Canada.
[2]Department of Molecular Genetics, University of Toronto, Toronto, Canada.
[3]Banting and Best Department of Medical Research, University of Toronto, Toronto, Canada.

It has been shown elsewhere (Jaakkola et al., 2000; Kuang et al., 2005; Leslie et al., 2002; Weston et al., 2005; Liao and Noble, 2002), that kernel support vector machines (SVMs) have better prediction performance on biological sequence data than other methods. Moreover, it was shown in Weston et al. (2005) that random-walk kernels (Szummer and Jaakkola, 2001) and empirical-map kernels (Scholkopf and Smola, 2002) produced promising results on protein remote homology detection. However, the process of deciding the optimal number of random steps in a random-walk kernel and the process of deciding the scaling parameter in an empirical-map kernel remain as challenging problems (Weston et al., 2005). In this paper, we present two approaches to address these problems that improve prediction accuracy. In the first approach, we use label information of training data and a positive linear combination of random-walk kernels to approximate the random-walk kernel with the optimum steps of a random walk, thereby obtaining a convex combination of random-walk kernels with different random-walk steps which achieves the best classification confidence on the labeled training set. In the second approach, we construct an empirical kernel map using profile kernels. The scaling parameter of the empirical map is decided by minimizing the Leave-One-Out (LOO) nearest neighbor classification error.

Section 2 gives a brief introduction to SVM classification based on mismatch-string kernels. Section 3.1 describes our first approach, a convex combination of random-walk kernels. Section 3.2 describes our second approach, empirical-map kernels based on profile kernels. Section 4 present experimental results of protein homology detection on the Structural Classificaion Of Proteins (SCOP) dataset. Section 5 concludes the paper with a discussion on the proposed methods and provides some ideas for future research.

2. SVM FOR BIOLOGICAL SEQUENCE CLASSIFICATION USING MISMATCH STRING KERNELS

A SVM (Scholkopf and Smola, 2002; Vapnik, 1995) is a discriminative model proposed especially for classification. Consider a two-class training set, $\{X, y\}$ and a test set U, where X is a matrix whose ith column, X_i, is the feature vector of data point i in the training set, U is a matrix whose jth column, U_j, is the feature vector of data point j in the test set, and y, a column vector whose ith component y_i is the label of data point i in the labeled set, $y_i \in \{-1, 1\}$, $X_i, U_j \in R^d$, $i = 1, \ldots, N$, $j = 1, \ldots, M$.

A linear SVM gives a separating hyper-plane that maximizes the margin between the sample data points of the two classes. The dual problem of a soft-margin SVM can be formulated as follows:

$$\begin{aligned} \max_{\alpha} \quad & 2\alpha^T \mathbf{1} - \alpha^T (K_{tr} \otimes yy^T)\alpha, \\ \text{s.t.} \quad & \alpha^T y = 0, \mathbf{0} \leq \alpha \leq C\mathbf{1}, \end{aligned} \tag{1}$$

where $\mathbf{1}$ and $\mathbf{0}$ are column vectors containing all ones and zeros respectively, $\otimes$ is the component-wise matrix multiplication operator, $K = [X|U]^T[X|U]$, is the dot product between feature vectors of pairwise data points, K_{tr} is the training part of K where $K_{tr} = X^T X$, and, C is the penalty coefficient penalizing margin violations. As the above dual problem is only dependent on dot-products between feature vectors, we can discard the original feature vectors of data points and calculate a kernel matrix K directly to represent the relationship between the original data points. As discussed in Scholkopf and Smola (2002), any symmetric positive semi-definite matrix can be used as a valid kernel matrix K. Therefore by constructing a kernel, K, we can map every data point, X_i, to a high-dimensional feature space, in which a SVM can be used to generate a separating hyper-plane.

For biological sequences, a kernel function can be used to map these sequences consisting of characters representing amino acids to a higher dimensional feature space on which a max-margin classifier is trained. All the computations of a SVM are performed on the dot products of the pairwise feature vectors stored in the kernel matrix. For example, suppose A is an alphabet of ℓ symbols ($\ell = 20$ for protein sequences), then k-mer string kernel maps every sequence in A to a ℓ^k-dimensional feature space in which coordinates are indexed by all possible sub-sequences of length k (k-mers). Specifically, the feature map of a k-mer string kernel is given by

$$\Phi_k(x) = (\Phi_{\alpha_1}(x), \Phi_{\alpha_2}(x), \ldots, \Phi_{\alpha_{|k}}(x))^T, \tag{2}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_{\ell^k}$ is an ordering of all the ℓ^k possible k-mers, and $\Phi_\alpha(x)$ is the number of occurrences of k-mer α in sequence x. The corresponding kernel matrix is

$$K_k(x, y) = \Phi_k(x)^T \Phi_k(y). \tag{3}$$

The mismatch string kernel extends this idea by accommodating mismatches when counting the number of occurrences of a k-mer in an input sequence. In particular, for any k-mer, α, let $N_{(\alpha,m)}$ be the set of all k-mers that differ from α by at most m mismatches. The kernel mapping and kernel matrix are then defined as follows:

$$\Phi_{(k,m)}(x) = (\Phi_{(k,m),\alpha_1}(x), \ldots, \Phi_{(k,m),\alpha_{\ell^k}}(x))^T, \tag{4}$$

$$\Phi_{(k,m),\alpha}(x) = \sum_{\beta \in N_{(\alpha,m)}(x)} \Phi_\beta(x), \tag{5}$$

$$K_{(k,m)}(x, y) = \Phi_{(k,m)}(x)^T \Phi_{(k,m)}(y). \tag{6}$$

A profile of a protein sequence is a sequence of multinomial distributions. Each position of a protein sequence's profile is a multinomial distribution on 20 amino acids, representing the emission probabilities of the 20 amino acids at each position in that sequence. A Profile Kernel (Kuang et al., 2005) extends the mismatch-string kernel by using additional profile information of each sequence. Instead of treating all k-mers with less than m mismatches similarly as the mismatch-string kernel described above, the profile-kernel examines these k-mers further by looking at the emission probabilities (profiles) at the mismatched positions and only accepts those mismatches that pass a certain threshold. The work-flow for constructing a profile kernel as described in Kuang et al. (2005) is shown in Figure 1. Each sequence has a profile, which

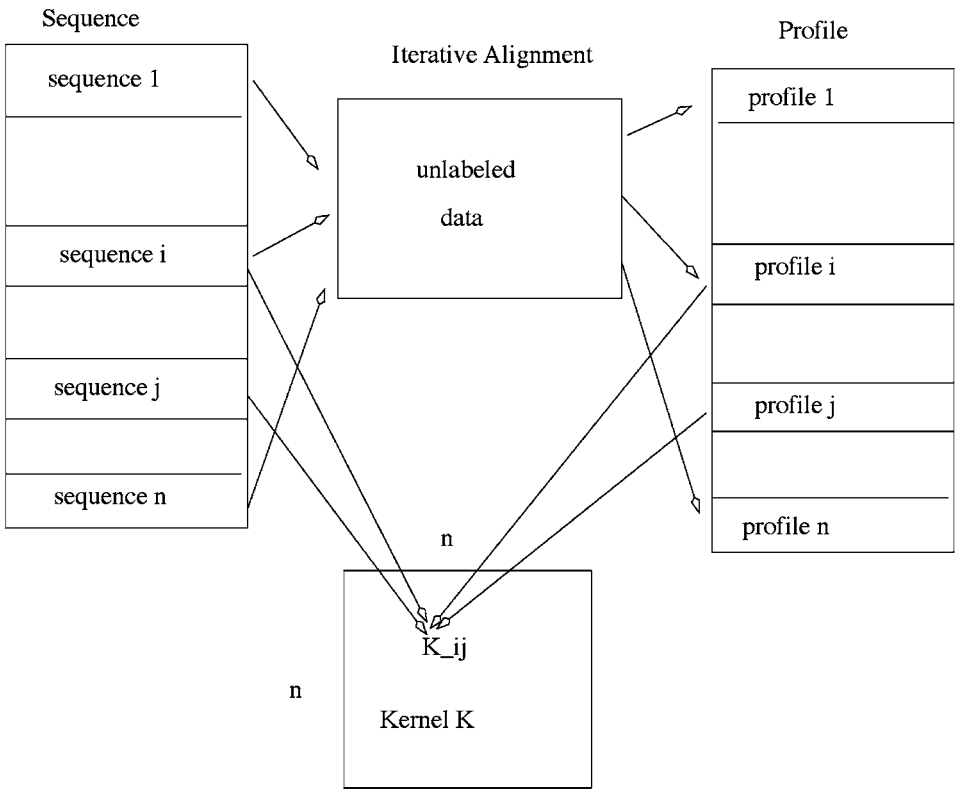

FIG. 1. The work-flow of constructing a profile kernel in Kuang et al. (2005).

is obtained by iteratively aligning each sequence to the sequences in an unlabeled set using PSI-BLAST (Altschul et al., 1997). Suppose we have a sequence $x = x_1 x_2 \ldots x_N$ of amino acids of length N, then $P(x) = \{p_i^x(a), a \in \Sigma\}_{i=1}^N$ is the profile of sequence x, where Σ is the set of 20 amino acids and $p_i^x(\cdot)$ is the multinomial distribution on the 20 amino acids at the ith position of the profile of sequence x. For example, $p_i^x(a)$, is the emission probability of amino acid a at position i, such that $\sum_{a\in\Sigma} p_i(a) = 1$ at each position i. In the Profile Kernel, the neighborhood of a k-mer $x[j+1 : j+k] = x_{j+1}x_{j+2}, \ldots, x_{j+k}$ in sequence x is defined as:

$$M_{(k,\sigma)}(P(x[j+1 : j+k])) = \left\{ \beta = b_1 \ldots b_k : -\sum_{i=1}^{k} \log p_{j+i}^x(b_i) < \sigma \right\}, \tag{7}$$

where the free parameter σ controls the size of the neighborhood, and $p_{j+i}^x(b)$ for $i = 1, \ldots, k$ is obtained from the profile of sequence x, $0 \le j \le |x| - k$. Further, $p_{j+i}^x(b)$ can be smoothed using the background frequency of amino acid b. The feature vector of sequence x in the Profile Kernel is defined as the following:

$$\Phi_{(k,\sigma)}(x) = \sum_{j=0}^{|x|-k} (\phi_{\beta_1}(P(x[j+1 : j+k])), \ldots, \phi_{\beta_{\ell^k}}(P(x[j+1 : j+k])))^T, \tag{8}$$

where $\beta_1, \ldots, \beta_{\ell^k}$ is an ordering of all possible k-mers, and the coordinate $\phi_\beta(P(x[j+1 : j+k]))$ is 1 if $\beta \in M_{(k,\sigma)}(P(x[j+1 : j+k]))$, and 0 otherwise. The profile kernel uses the profile to measure the mismatch information between different letters at each position of each sequence. Therefore, it's more accurate than the mismatch string kernel. In this paper, we will use the profile kernel discussed above as the base kernel in the derivation of the random-walk kernel and empirical-map kernel.

3. METHODS

3.1. Improved random-walk kernel for biological sequence classification

In this section, we will describe our improved random-walk kernels. Our motivation for using a random-walk kernel is its ability to coerce data points in the same cluster to stay closer while making data points in different clusters to stay farther apart by propagating similarity on both labeled data and unlabeled data (Szummer and Jaakkola, 2001; Weston et al., 2005). If we view a set of data points as a complete (or sparse) graph, in which the weights between data points are viewed as similarity scores, then we can make use of unlabeled data to help propagate similarity information through the whole graph. For example, we have a graph containing two labeled data points, i and j, and two unlabeled data points, s and t, i is highly similar to s, s is highly similar to t, and t is highly similar to j, but i and j are not very similar to each other in the given graph. After two steps of similarity propagation, i and j will become similar in a new similarity graph. When the similarity-propagation process is over, we hope that data points in the same class (having the same label) will stay relatively closer while data points in different classes (having different labels) will stay relatively farther apart (Chapelle et al., 2002; Szummer and Jaakkola, 2001; Weston et al., 2005). However, when the weight matrix connecting data points is not completely consistent with the labels of data points, excessive similarity propagation through the graph will harm the classification; therefore, we use label information to guide the similarity-propagation process on the graph. This motivated us to use the label information of training data to optimize the parameter in a random-walk kernel.

A t-step random-walk kernel is generally derived from a transition matrix with a t-step random walk by normalization and symmetrization. Given a base kernel K with positive entries (in this paper, we use profile kernels), the transition matrix P of a one-step random walk is defined as follows: let P_{ij} be the probability $P(x_i \to x_j)$, then after t steps of a random walk, the transition probability can be calculated as $P^t = (D^{-1}K)^t$, where D is a diagonal matrix with $D_{ii} = \sum_k K_{ik}$. Ideally, we want to use P^t as the kernel matrix for SVM classification. However, a kernel matrix must be a symmetric positive semi-definite matrix, therefore, we do the following manipulations to derive a kernel matrix from P^t. As described in

Weston et al. (2005), let $L = D^{-\frac{1}{2}} K D^{-\frac{1}{2}}$, with its eigen-decomposition, $L = U \Lambda U^T$, and $\tilde{L} = U \Lambda^t U^T$, where, t denotes the exponent, and, T, denotes the transpose. Then, the new kernel corresponding to a t-step random walk is calculated as $\tilde{K} = \tilde{D}^{-\frac{1}{2}} \tilde{L} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \tilde{L}_{ii}$. We can see that the derived kernel $\tilde{K}$ relates to the transition matrix after t-steps of a random walk P^t as follows: $\tilde{K} = \tilde{D}^{-\frac{1}{2}} D^{\frac{1}{2}} P^t D^{-\frac{1}{2}} \tilde{D}^{-\frac{1}{2}}$.

A random-walk kernel based on PSI-BLAST E-values has been tried in Weston et al. (2005) for protein remote homology detection. The challenge in random-walk kernels is how to decide the optimal number of random steps. Since random walks exploit both labeled data and unlabeled data to estimate the manifold structure of data, performing too many steps of a random walk can lead to the possibility of nearby clusters joining together, resulting in data points in different classes come closer. On the other hand, if the number of steps is too small, it can lead to a separation of data points in the same class. Our goal is to find the optimum number of steps that is most consistent with the class memberships of the data points. Using the label information of training data to learn the parameters of kernel functions has been successfully adopted by researchers. Related research can be found in elsewhere (Lanckriet et al., 2004; Min et al., 2007; Zhu et al., 2005).

Here, we need to learn the parameters of the random-walk kernel that achieves the goal of max-margin classification using the label information of training data. A brute-force solution to this problem results in a non-convex optimization problem, therefore, we propose using a positive linear combination of the base kernel and random-walk kernels from one step to m steps to calculate a new kernel to approximate the kernel with the optimum number of random steps by optimizing the dual objective function of the resulting SVM. We call the resulting kernel "improved random-walk kernel." Since every t-step random-walk kernel has trace n, if the base kernel also has trace n, by restricting the learned kernel to have trace n too, a positive linear combination of the base kernel and the random-walk kernels leads to a convex combination of these kernels, where n is the total number of training data and test data points. The result is the following optimization problem:

$$
\begin{aligned}
& \min_{\mu} \max_{\alpha} 2\alpha^T \mathbf{1} - \alpha^T (K_{tr} \otimes y y^T) \alpha, \\
\text{s.t.} \quad & \alpha^T y = 0 \\
& \mathbf{0} \le \alpha \le C \mathbf{1}, \\
& K = \mu_0 \tilde{K}^0 + \sum_{k=1}^{m} \mu_k \tilde{K}^k, \\
& \sum_{k=0}^{m} \mu_k = 1, \\
& \mu_k \ge 0, \qquad k = 0, \ldots, m,
\end{aligned} \tag{9}
$$

where $\tilde{K}^0$ is the base kernel for deriving the improved random-walk kernel, $\tilde{K}^k$ is the random-walk kernel with a k-step random walk, and m is the maximal number of random steps performed. The above optimization problem is a special case of the optimization problem discussed in Lanckriet et al. (2004). We follow the framework as shown in Lanckriet et al. (2004), and show that the above problem is equivalent to the following quadratically constrained convex optimization problem (for detailed derivations, see Appendix):

$$
\begin{aligned}
& \min_{\alpha, t} t, \\
\text{s.t.} \quad & t \ge \alpha^T (\tilde{K}^k_{tr} \otimes y y^T) \alpha - 2\alpha^T \mathbf{1}, \qquad k = 0, \ldots, m, \\
& \alpha^T y = 0 \\
& \mathbf{0} \le \alpha \le C \mathbf{1},
\end{aligned} \tag{10}
$$

where tr denotes the training part of the corresponding kernel. The optimal values of parameters $\mu_k, k = 0, \ldots, m$ are exactly the dual solution to the above quadratic constrained convex optimization problem. They can be found using the standard optimization software SeDuMi (Sturm, 1999) or MOSEK (Andersen and Andersen, 2000) which solve the primal and dual of an optimization problem simultaneously. For huge datasets, we can use SMO-like gradient-based algorithms (Platt, 1999) to solve the above problem. In this work, all the optimization problems were solved using MOSEK.

As described in Kondor and Lafferty (2002), the ideas of random walks and diffusion are closely related. Given a kernel matrix K, we can view it as a similarity matrix and compute the graph Laplacian as $Q = D - K$, where D is a diagonal matrix described in this section. Instead of taking the form of the tth power of the transition matrix P as in random-walk kernels, a diffusion kernel $K^{diffuse}$ takes a form of the matrix exponential of Q:

$$\begin{aligned} K^{diffuse} = e^{\beta Q} &= \lim_{n->\infty; n\in\mathcal{N}} \left(I + \frac{\beta Q}{n}\right)^n \\ &= I + \beta Q + \frac{\beta^2}{2} Q^2 + \cdots + \frac{\beta^t}{t!} Q^t + \cdots \\ &= \sum_i v_i e^{\beta\lambda_i} v_i^T, \end{aligned} \tag{11}$$

where β is a real parameter to control the diffusion, which is analogous to the minus inverse squared variance parameter in Gaussian kernels, I is an identity matrix, $\mathcal{N}$ is the integer set, and, v_i and λ_i are the ith eigenvalue and eigenvector of K, respectively. The first line in the above equation can be interpreted as a random walk with an infinite number of infinitesimally small steps. In this paper, we compute diffusion kernels based on profile kernels, and compare their performance to that of improved random-walk kernels shown in the experimental results section.

The computation of both a random-walk kernel and a diffusion kernel requires the eigen-decomposition of a base kernel, which has a worst-case time complexity $O(n^3)$. Computing the improved random-walk kernel described above requires solving in addition, the quadratically constrained convex optimization problem in equation 10, which has a worst-case time complexity $O(mn_{tr}^3)$ using an interior-point method, where n_{tr} is the number of training data points.

3.2. Empirical-map kernel

An empirical-map kernel based on PSI-BLAST E-values has been applied in the analysis of biological sequence data with reasonably good performance (Weston et al., 2005). We compared the LOO nearest neighbor classification errors on protein sequence classification produced using PSI-BLAST E-values to those produced using a normalized profile kernel, and found that the normalized profile kernel captures the neighborhood similarity much better than the PSI-BLAST E-values. This motivated us to use the normalized profile kernel to derive an empirical-map kernel for biological sequence classification. In Weston et al. (2005), the authors report their best result after tuning a scaling parameter; however, they do not provide a method for calculating this parameter. In contrast, here we propose three approaches for calculating the scaling parameter in the empirical-map kernel.

Given a similarity matrix S where S_{ij} is the similarity score between data points X_i and X_j, the empirical map for data point x is defined as:

$$\Phi^{emp}(x) = (e^{-\lambda S(x,X_1)}, e^{-\lambda S(x,X_2)}, \ldots, e^{-\lambda S(x,X_P)})^T, \tag{12}$$

where P is the number of available data points, including both labeled data points and unlabeled data points. The empirical-map kernel is defined as $K_{\lambda,ij} = \Phi^{emp}(X_i)^T \Phi^{emp}(X_j)$. The key to deriving the optimal empirical-map kernel is calculating the scaling parameter λ.

In this paper, we use the normalized profile kernel matrix as the similarity matrix. Given a profile kernel matrix K^{prof}, we normalize it such that every sequence has a unit feature vector (the norm is 1)

as follows:

$$K^{prof,norm} = \Delta^{-\frac{1}{2}} K^{prof} \Delta^{-\frac{1}{2}}, \tag{13}$$

where Δ is a diagonal matrix and $\Delta_{ii} = K_{ii}^{prof}$. Then the empirical-map kernel is given by:

$$K_{\lambda,ij}^{emp} = \sum_{k=1}^{P} e^{-\lambda(K_{ik}^{prof,norm} + K_{jk}^{prof,norm})}. \tag{14}$$

We normalize K^{emp} again so that every sequence has a unit feature vector, giving the following normalized empirical-map kernel:

$$K_{\lambda,ij}^{emp,norm} = \frac{\sum_{k=1}^{P} e^{-\lambda(K_{ik}^{prof,norm} + K_{jk}^{prof,norm})}}{\sqrt{\sum_{k=1}^{P} e^{-2\lambda K_{ik}^{prof,norm}} \sum_{k=1}^{P} e^{-2\lambda K_{jk}^{prof,norm}}}}. \tag{15}$$

One way to calculate λ is by cross validation; however, it is computationally expensive to search over a long list of candidate values and often this method fails to produce good values of λ. Alternatively, we can substitute K with $K_{\lambda}^{emp,norm}$ in Equation (1), perform the maximization with respect to α, and then perform the minimization with respect to λ. However, this problem is non-convex with respect to λ, and each iteration for calculating the optimal value of α with λ fixed involves a quadratic programming problem. Instead, in this paper, we propose three different approaches to calculate λ. The first approach calculates λ by maximizing the Kernel Alignment Score (KAS) (Cristianini et al., 2001). Given the labels of training data, the optimal kernel is given by $K^{opt} = YY^T$. We calculates λ by maximizing the alignment score as follows:

$$KAS = Trace(K^{emp,norm^T} K^{opt}) / \sqrt{Trace(K^{emp,norm^2}) Trace(K^{opt^2})}. \tag{16}$$

The second approach calculates λ in a way that encourages the similarities between data points within a class to be as large as possible. Given a kernel matrix K, we calculate the probability of sequence i and sequence j being in the same class as, $P_{ij} = \frac{K_{ij}}{\sum_{k=1}^{\ell} K_{ik}}$, where ℓ is the size of the labeled training set. We calculate the probability matrix $P_{\lambda}^{emp,norm}$ using $K_{\lambda}^{emp,norm}$, and ,P^{opt} using K^{opt}. To enforce the class-dependent constraint, we minimize the following KL-divergence between P^{opt} and $P_{\lambda}^{emp,norm}$:

$$KL = \sum_{ij} P_{ij}^{opt} \log[P_{ij}^{opt} / P_{\lambda,ij}^{emp,norm}]. \tag{17}$$

In the third approach, λ is chosen such that the normalized empirical-map kernel in Equation (15) corresponds to a good metric for defining a neighborhood consistent with the labels of the labeled data, i.e., we choose λ to minimize the LOO Nearest Neighbor classification error over the labeled dataset. To limit the search space, we use the optimal λs found by the first approach and the second approach as reference values, and we always take the smallest λ when there are several local minima of λ achieving equally good classification error.

All the three approaches of computing λ described above have a worst-case time complexity $O(n_{tr}^2)$. The third approach is often the most stable and often works best in practice, therefore, we suggest using this approach as the default approach for computing λ in the empirical-map kernel for possible future applications. Once λ is decided, the worst-case time complexity for computing an empirical-map kernel is $O(n^3)$ using traditional matrix multiplications, but this time complexity is reduced to $O(n^{2.376})$ using advanced matrix multiplication algorithms in Cohn et al. (2005). In contrast, computing an improved random-walk kernel has a worst-case time complexity $O(n^3)$ dominated by the eigen-decomposition of the base kernel matrix.

4. EXPERIMENTAL RESULTS ON PROTEIN REMOTE HOMOLOGY DETECTION

We determine the classification performance of the improved random-walk kernels and the empirical-map kernels against the profile kernels by comparing their ability to detect protein remote homology. We used the benchmark dataset, derived by Jaakkola from the SCOP database for this purpose (Jaakkola et al., 2000; Murzin et al., 1995). In SCOP, protein sequences are classified into a three-level hierarchy: Fold, Super-family, and Family, starting from the top. Remote homology is simulated by choosing all the members of a family as positive test data, some families in the same super-family of the test data as positive training data, all sequences outside the fold of the test data as either negative training data or negative test data, and sequences that are neither in the training set nor in the test set as unlabeled data. This data splitting scheme has been used in several previous papers (Jaakkola et al., 2000; Liao and Noble, 2002; Weston et al., 2005). We used the same training and test data split as that used in Liao and Noble (2002) and Weston et al. (2005). We used version 1.59 of the SCOP dataset (*http://astral.berkeley.edu*), in which no pair of sequences share more than 95% identity.

In the data splits, of most experiments, there are only a few positive test cases but, hundreds, or even thousands of negative test cases. The maximum number of positive test cases is usually below 30, but the maximum number of negative test cases is above 2600. The minimum number of positive test cases is 1, but the minimum number of negative test cases is still above 250. In the experiments with a very limited number of positive test cases and a large number of negative test cases, we can almost ignore the ranking of positive cases below 50 negative cases. In such situations, we consider the ROC_{50} score much more informative of prediction performance of different methods than the ROC score. Here, a ROC curve plots the rate of true positives as a function of the rate of false positives at different decision thresholds. The ROC score is the area under the curve. The ROC_{50} score is the ROC score computed up to the first 50 false positives. Thus, in our experiments, we only compare the ROC_{50} scores corresponding to different kernels.

Since the optimization procedure for calculating the convex combination coefficients for combining random-walk kernels is highly dependent on labels, we adopted the following approach: prior to training the SVM, we added to the positive training set labeled as positive, close homologs of the positive training data in the unlabeled set found by PSI-BLAST with E-value less than 0.05. When training the SVM based on random-walk kernels with a fixed number of random steps, diffusion kernels, and empirical-map kernels, we also used unlabeled data as discussed above. The improved random-walk kernel and the empirical-map kernel are based on the two profile kernels which produced the top two results on SCOP in Kuang et al. (2005). Both profile kernels were obtained by setting the k-mer length to 5 and the parameter σ to 7.5. However, the best profile kernel was obtained using the PSI-BLAST profile trained up to five search iterations while the second best profile kernel was obtained using the PSI-BLAST profile trained up to two search iterations. The profile kernels were cosinely normalized to have trace n as in Equation (13) before they were used for the SVM classification and the calculation of improved random-walk kernels. In Kuang et al. (2005), Leslie et al. (2002), and Weston et al. (2005), it has been shown that cosinely normalized mismatch string kernels including the profile kernels are very effective for protein classification. And in the experiments, the maximum number of steps m of random walks for the improved random-walk kernel was set to 6 (when it was set to 7, 8, 9, or 10, we saw an increasing computational time but no significant improvement in the results over that of $m = 6$). To compare improved random-walk kernels to diffusion kernels, the free parameter β in $K^{diffuse}$ was decided by five-fold cross validation. When using LOO Nearest Neighbor classification error to decide λ, we used the values found by the first and the second approaches as reference and limit the search space to regions around the reference values. We used a hard-margin SVM to identify protein remote homology; the free parameter C in the SVM was set to infinity, which has been shown to be very effective for protein classification (Weston et al., 2005).

Table 1 shows the ROC_{50} scores produced by the random-walk kernels with the best fixed number of random steps, the scores produced by the improved random-walk kernels, and, the scores produced by the diffusion kernels based on the best and second best profile kernels. It clearly shows that the improved random-walk kernels have much better performance than the profile kernels and the diffusion kernels. Moreover, based on the best profile kernel, the random-walk kernel with the best fixed number of random steps (two steps) has a worse performance than the base kernel; and based on both profile kernels, the diffusion kernels have a worse performance than the two base kernels. The poor performance of the

TABLE 1. OVERALL MEAN ROC_{50} SCORES OVER 54 PROTEIN FAMILIES CORRESPONDING TO DIFFERENT RANDOM-WALK KERNELS AND DIFFUSION KERNELS[a]

Random-walk kernel (RWK)	*Overall mean ROC_{50}*
2-step RWK using the second best profile kernel	0.847
Improved RWK using the second best profile kernel	0.867
Diffusion kernel using the second best profile kernel	0.746
Second best profile kernel (base kernel)	0.824
2-step RWK using the best profile kernel	0.862
Improved RWK using the best profile kernel	0.901
Diffusion kernel using the best profile kernel	0.790
Best profile kernel (base kernel)	0.874

[a]2-step random-walk kernels work best on the SCOP dataset among all the random-walk kernels with a fixed number of random steps. We see that improved random-walk kernels outperform random-walk kernels with the best fixed number of random steps and diffusion kernels.

diffusion kernels here is probably due to the very limited positive labeled data and the non-optimality of the parameter β decided by cross validation. From Table 1, we conclude that the convex combination of random-walk kernels is an effective way of using random walks. Table 2 lists the ROC_{50} scores by the empirical-map kernels with λ calculated using three different approaches. From Table 2, we see that the first and the second approaches have similar performance, while the third approach outperforms these two. In the remainder of this paper, the empirical-map kernel is taken in reference to the kernel with λ calculated using the third approach except where explicitly stated.

Table 3 gives the overall mean ROC_{50} scores over 54 protein families obtained by several previous representative approaches and our improved random-walk kernels and empirical-map kernels. It can be clearly seen that previous approaches except for the profile kernels have low ROC_{50} scores, below 0.70. The two profile kernels produce ROC_{50} scores above 0.80. In contrast, our improved random-walk kernels and empirical-map kernels produce ROC_{50} scores above 0.90. Because we have only a few positive test cases, but hundreds or even thousands of negative test cases in most of the 54 experiments, the mean ROC_{50} score produced by a random predictor is close to 0. We can see that all the approaches listed in the table have much better performance than a random predictor.

TABLE 2. OVERALL MEAN ROC_{50} SCORES OVER 54 PROTEIN FAMILIES CORRESPONDING TO DIFFERENT EMPIRICAL-MAP KERNELS WITH λ CALCULATED USING THREE DIFFERENT APPROACHES[a]

Empirical-map kernels	*Overall mean ROC_{50}*
Using the second best profile kernel [1]	0.862
Using the second best profile kernel [2]	0.866
Using the second best profile kernel [3]	0.878
Second best profile kernel (base kernel)	0.824
Using the best profile kernel [1]	0.904
Using the best profile kernel [2]	0.900
Using the best profile kernel [3]	0.911
Best profile kernel (base kernel)	0.874

[a][i] denotes the empirical-map kernel with λ calculated using the ith approach, $i = 1, 2$, or 3. From this table, we find that the third approach for deciding λ is the best. In contrast, the cross validation procedure (Min et al., 2007) to choose λ gives overall mean ROC_{50} scores 0.848 based on the second best profile kernel and 0.891 based on the best profile kernel, and the searching procedure is very slow.

TABLE 3. OVERALL MEAN ROC_{50} SCORES OVER 54 PROTEIN FAMILIES CORRESPONDING TO DIFFERENT KERNELS[a]

Methods	*Overall mean ROC_{50}*
eMOTIF (Ben-Hur and Brutlag, 2003; Kuang et al., 2005)	0.247
SVM-pairwise [PSI-BLAST] (Liao and Noble, 2002; Kuang et al., 2005)	0.533
Spectrum-kernel [PSI-BLAST] (Leslie et al., 2002)	0.545
Neighborhood (Weston et al., 2005)	0.699
Second best profile kernel (the second best result)	0.821
Best profile kernel (the best result)	0.874
Improved RWK using the second best profile kernel	0.867
Empirical-map kernel using the second best profile kernel	0.878
Improved RWK using the best profile kernel	0.901
Empirical-map kernel using the best profile kernel	0.911

[a]Here the empirical-map kernels refer to the kernels with λ calculated using the third approach. In this table, the top rows show the results produced by several previous representative approaches and the best published results; the middle rows show our results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the second best profile kernel; and the bottom rows show our results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the best profile kernel.

Table 4 shows the ROC_{50} scores for the most difficult protein family Glutathione S-transferases, N-terminal domain on which all the previous approaches produced very poor performance while our approaches performed well. In the experiment for this protein family, we have 13 positive test proteins and 927 negative test proteins, therefore, the ROC_{50} score produced by a random predictor should be close to 0, while the profile kernels and our proposed kernels performed much better than a random predictor.

Figures 2–8 show our results in detail. Figures 2 and 3 show the number of protein families that score above the different ROC_{50} threshold values for our kernels and the top two profile kernels (note that they are not ROC_{50} curves but the summarization of all the ROC_{50} scores). Figures 4–7 show the 10 largest improvements in ROC_{50} scores for our kernels over the top two best profile kernels. Figure 8 compares the SVM classification scores calculated using the best profile kernel and the scores calculated using the empirical-map kernel based on the best profile kernel on the most difficult protein family Glutathione S-transferases, N-terminal domain. On this special family, we tried to find the obvious changes of cluster patterns by visualizing the kernel matrices, but it turned out that the improvements of ROC_{50} scores were

TABLE 4. ROC_{50} SCORES ON THE MOST DIFFICULT PROTEIN FAMILY GLUTATHIONE S-TRANSFERASES, N-TERMINAL DOMAIN CORRESPONDING TO DIFFERENT KERNELS[a]

Methods	*ROC_{50} on the hardest protein family*
eMOTIF (Ben-Hur and Brutlag, 2003; Kuang et al., 2005)	0.000
SVM-pairwise [PSI-BLAST] (Liao and Noble, 2002; Kuang et al., 2005)	0.000
Spectrum-kernel [PSI-BLAST] (Leslie et al., 2002)	0.000
Neighborhood (Weston et al., 2005)	0.000
Second best profile kernel (the second best result)	0.045
Best profile kernel (the best result)	0.122
Improved RWK using the second best profile kernel	0.454
Empirical-map kernel using the second best profile kernel	0.455
Improved RWK using the best profile kernel	0.509
Empirical-map kernel using the best profile kernel	0.903

[a]Here the empirical-map kernels refer to the kernels with λ calculated using the third approach. In this table, the top rows show the results produced by several previous representative approaches and the best published results; the middle rows show our results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the second best profile kernel; and the bottom rows show our results using the improved Random-Walk Kernel (RWK) and the empirical-map kernel based on the best profile kernel.

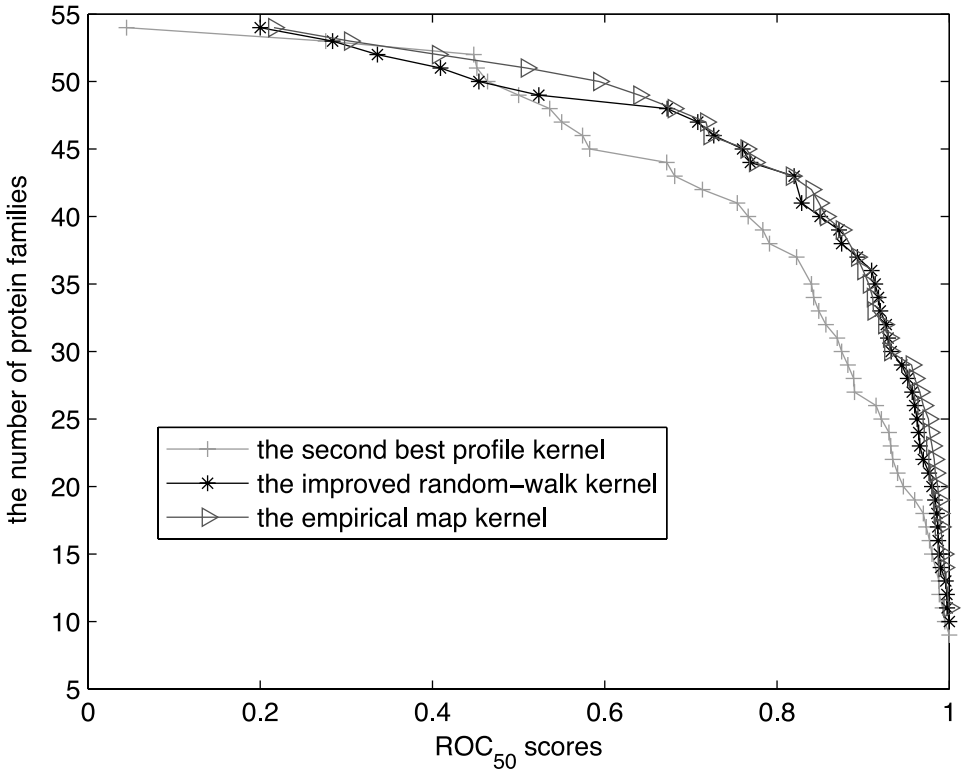

FIG. 2. The number of protein families with ROC_{50} scores above different thresholds for different kernels using the second best profile kernel.

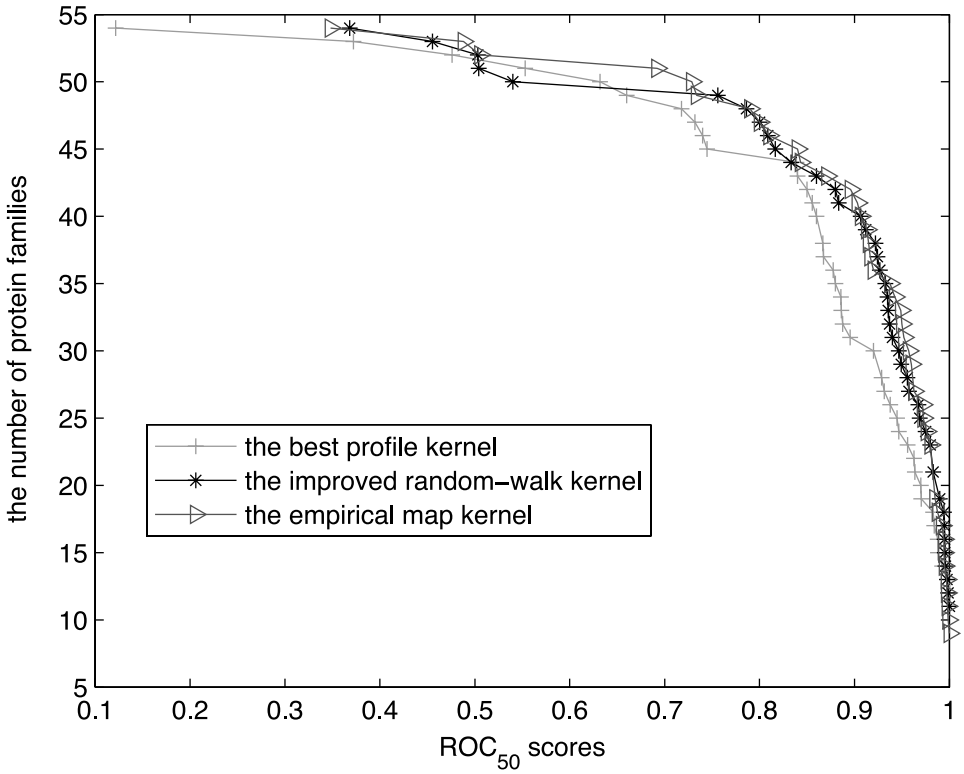

FIG. 3. The number of protein families with ROC_{50} scores above different thresholds for different kernels using the best profile kernel.

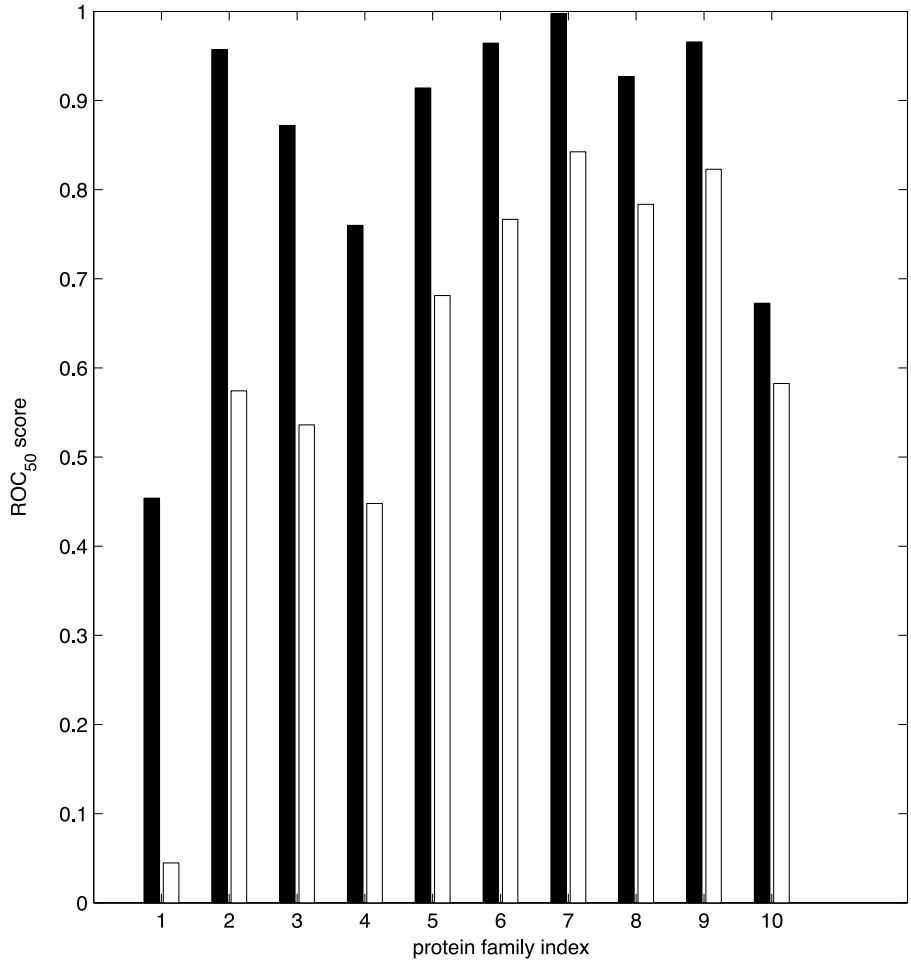

FIG. 4. The results obtained using the second best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the improved random-walk kernel based on the second best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the improved random-walk kernel based on the second best profile kernel, and the right white bar corresponds to the base kernel.

due to subtle changes of some kernel entries, which are hard to capture by eyes. Besides, we found that, based on the best profile kernel, the empirical-map kernel resulted in much more support vectors than the improved random-walk kernel and the base kernel. Out of 1943 training protein sequences, the empirical-map kernel with the learned $\lambda = 3$ resulted in 1901 support vectors, while the random-walk kernel and the base kernel resulted in 1026 and 1083 support vectors, respectively. In the empirical-map kernel, $\lambda = 3$ allows a lot of weak pairwise sequence similarities contributing to the construction of the kernel; moreover, almost every training sequence for this protein family was learned to be a support vector using this kernel. Therefore, the drastic improvement given by the empirical-map kernel for this protein family is probably due to the combination of a lot of weak pairwise sequence similarities, which might correspond to the combination of a lot of short sequence motifs. In our future work, we plan to extract sequence motifs that are crucial for determining each positive test protein's superfamily membership using the random-walk kernel and the empirical-map kernel to rank protein sub-sequences, and, we hope that we would be able to identify biologically meaningful motifs for the protein family Glutathione S-transferases, N-terminal domain and other families.

To determine whether the improvements obtained by the improved random-walk kernels and the empirical-map kernels are statistically significant, we performed Wilcoxon Matched-Pairs Signed-Ranks Tests on the differences between paired kernels. All the resulting p-values were below 0.05. The resulting p-value for the ROC_{50} score difference between the improved random-walk kernel based on the second best profile

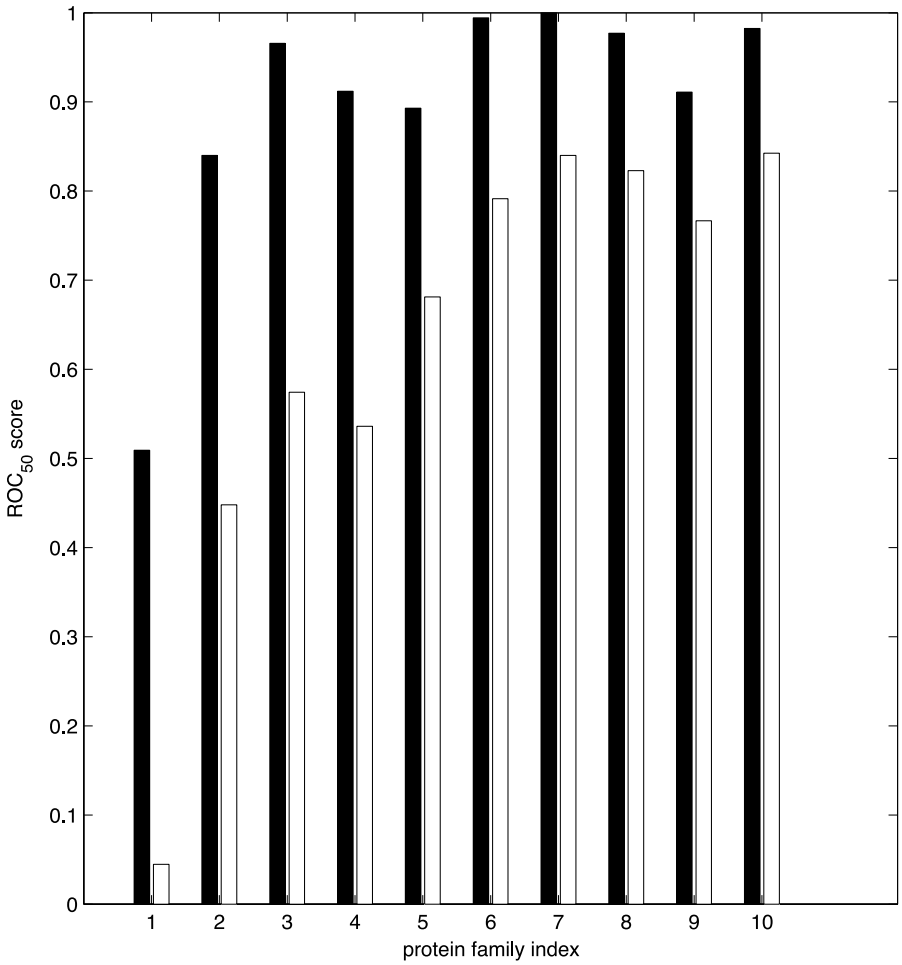

FIG. 5. The results obtained using the second best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the empirical-map kernel based on the second best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the empirical-map kernel based on the second best profile kernel, and the right white bar corresponds to the base kernel.

kernel and the base kernel was 1.52×10^{-4}. The p-value for the pair between the improved random-walk kernel based on the best profile kernel and the base kernel was 3.30×10^{-3}. The p-value for the pair between the empirical-map kernel based on the second best profile kernel and the base kernel was 1.67×10^{-4}, and the p-value for the pair between the empirical-map kernel based on the best profile kernel and the base kernel was 3.05×10^{-2}.

However, based on both the profile kernels, the p-values for the ROC_{50} score differences between the empirical-map kernels and the improved random-walk kernels are both greater than 0.15, which are not statistically significant. Since the empirical-map kernels produced higher mean ROC_{50} scores and have a lower order of computational time complexity than the improved random-walk kernels, we prefer to use the empirical-map kernels for future protein sequence classification tasks. We might consider the improved random-walk kernels to be better choices than the empirical-map kernels for some special problems in the special problem contexts there.

5. DISCUSSION

In this paper, we proposed two kernel learning approaches for protein remote homology detection based solely on protein sequence data. One approach approximates the optimal number of random steps in a

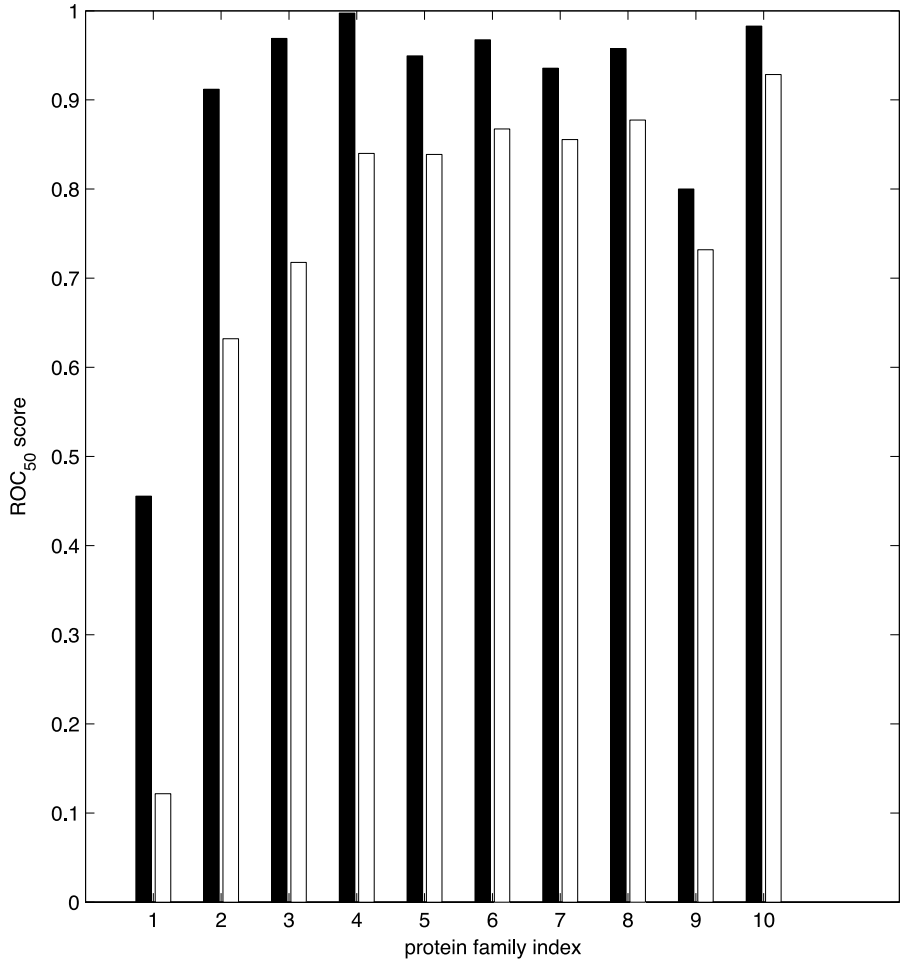

FIG. 6. The results obtained using the best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the improved random-walk kernel based on the best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the improved random-walk kernel based on the best profile kernel, and the right white bar corresponds to the base kernel.

random-walk kernel by calculating a convex combination of random-walk kernels with different numbers of random steps, and, the other approach uses profile kernels to derive empirical-map kernels with the scaling parameter λ calculated using a principled approach. The first approach scales down to a convex optimization problem avoiding concerns of local minima. The second approach initializes the value of λ in the empirical-map kernels by minimizing the KL divergence and maximizing the Kernel Alignment Score, and, then refining the value of λ by minimizing the Leave-One-Out nearest neighbor classification errors. It is a robust approach. We ran the procedure for calculating λ several times, each time obtaining the same refined value of λ on each protein family.

Both approaches make use of a large number of pairwise sequence similarities and unlabeled data to derive new kernels, which corresponds to new similarity metrics for pairwise sequences. In the first approach, pairwise sequence similarities contribute to defining the transition probability matrix for the random walks. The convex optimization procedure induces the new kernel to reflect the manifold structure of the sequences that is optimally consistent with the labeled training sequences.

In the second approach, the scaling parameter λ plays the role of selecting features in a soft way for the empirical-map kernel. When λ is small, small pairwise sequence similarities contribute weakly to the construction of the kernel based on the empirical map. When λ is large, only large pairwise sequence similarities contribute to the construction of the new kernel. Figure 9 illustrates how the feature component in the empirical map varies with distance between pairwise sequences for different λ values

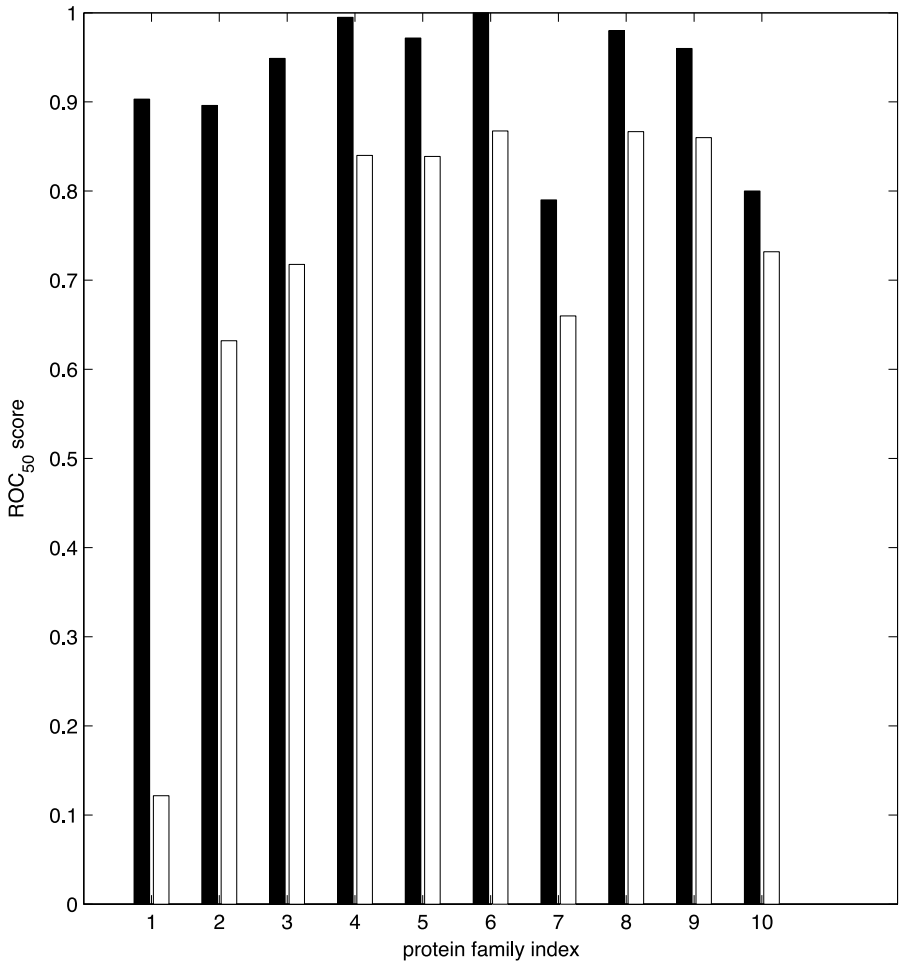

FIG. 7. The results obtained using the best profile kernel: the top 10 largest improvement in ROC_{50} scores out of 54 protein families for the empirical-map kernel based on the best profile kernel over the base kernel. In each group (for one protein family), the left black bar corresponds to the empirical-map kernel based on the best profile kernel, and the right white bar corresponds to the base kernel.

(note that the distances between pairwise sequences based on the normalized profile kernels are between 0 and 2). All three procedures in the second approach make use of the label information of training sequences to calculate λ in order to achieve good separability between positive sequences and negative sequences.

The experimental results on protein remote homology detection show that the improved random-walk kernels and the empirical-map kernels proposed here produce strikingly better results than previous methods, including the best approaches for solving this problem proposed to date. Out of 54 protein families, the best profile kernel produced ROC_{50} scores above 0.90 for 30 families, while the empirical-map kernel based on the best profile kernel produced ROC_{50} scores above 0.90 for 41 families. On one hand, this shows the effectiveness of the empirical map kernel, and on the other hand, it shows that the base kernel (the best profile kernel) has very good performance producing almost perfect results on more than half of the protein families. From Figures 4–7, we see that our approaches give more than 10% improvement over the base kernels on many difficult protein families. In particular, on the most difficult protein family Glutathione S-transferases, N-terminal domain on which all the previous approaches failed to produce useful results (ROC_{50} scores of zero or close to zero), our approaches produced very good results.

Our approaches are general and are readily applicable to other biological classification problems such as Transcription Factor Binding Site prediction and gene function prediction. The approaches described

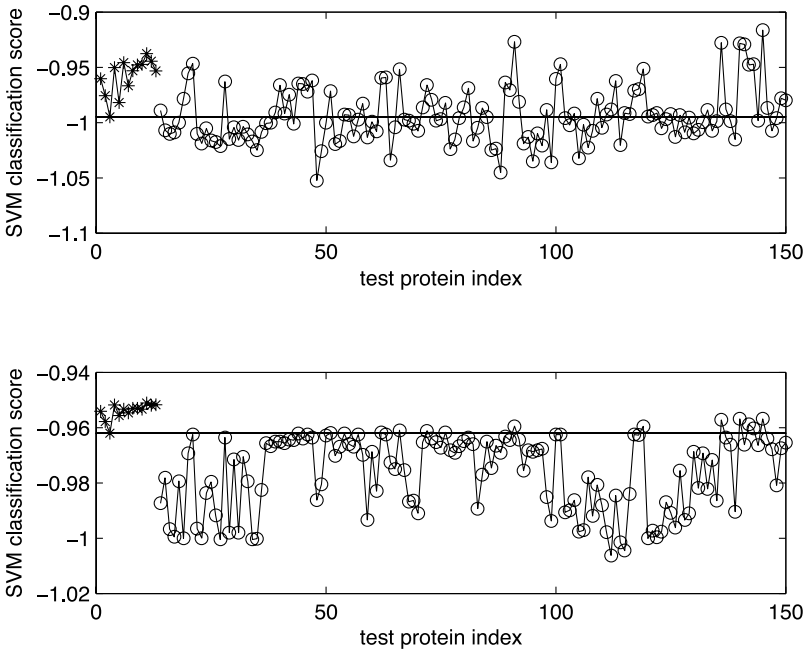

FIG. 8. The SVM classification scores calculated using the best profile kernel and the empirical-map kernel based on the best profile kernel on the most difficult protein family Glutathione S-transferases, N-terminal domain. **(Top plot)** Best profile kernel. **(Bottom plot)** Empirical-map kernel. In both plots, the stars represent all positive test proteins, and the circles represent some negative test proteins. In each plot, the horizontal line sits at the smallest value of the classification scores for all the positive test proteins. The top plot shows that if we want to classify most of the positive test proteins correctly by setting an appropriate threshold, there will be a lot of false positives; however, the bottom plot clearly shows that we can almost classify all the positive test proteins correctly by setting an appropriate threshold while only introducing a very small number of false positives.

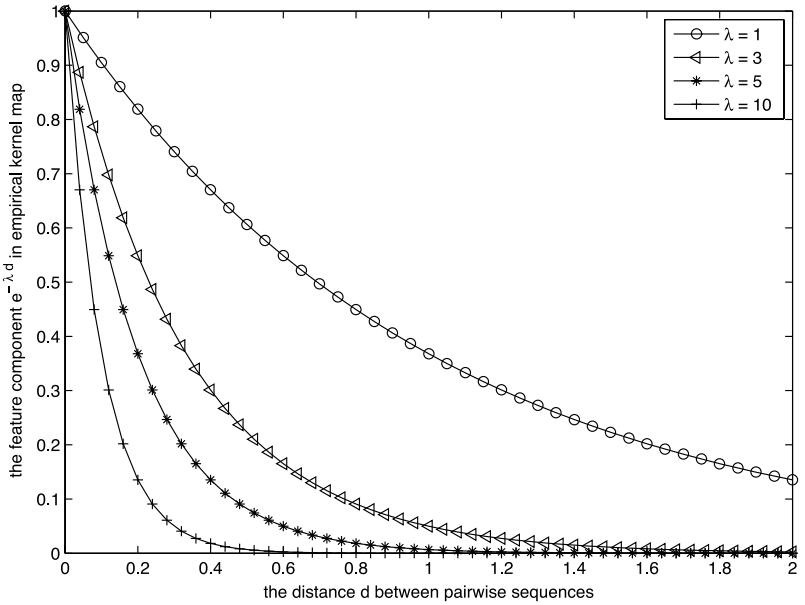

FIG. 9. This figure shows how the feature component $\exp(-\lambda d)$ in the empirical-map varies with the distance d between pairwise sequences for different λ values. When the normalized profile kernels are used to calculate the distances between pairwise sequences, the distances are always between 0 and 2.

here can also be applied to non-biological problems such as document classification, handwritten digit classification, and face recognition, where kernels are constructed on texts and images instead of on biological data.

6. APPENDIX

We now prove that the optimization problem in Equation (9) is equivalent to the optimization problem in Equation (10). It's easy to see that all the constraints in Equation (9) are linear thus convex with respect to α and μ. Let $\ell = 2\alpha^T\mathbf{1} - \alpha^T(K_{tr} \otimes yy^T)\alpha$, since only K_{tr} appears in ℓ in Equation (9), K_{tr} is the only part we need from K to solve Equation (9). ℓ is linear, thus convex with respect to μ. The Hessian of ℓ with respect to α is $-(K_{tr} \otimes yy^T)$, which is negative semi-definite, hence, ℓ is concave with respect to α. Therefore, we have the following equations:

$$\begin{aligned}
&\min_{\mu:\mu\geq\mathbf{0},\sum_{k=0}^{m}\mu_k=1}\ \max_{\alpha:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1}} 2\alpha^T\mathbf{1} - \alpha^T\left[\left(\sum_{k=0}^{m}\mu_k\tilde{K}_{tr}^k\right)\otimes yy^T\right]\alpha \\
&= \max_{\alpha:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1}}\ \min_{\mu:\mu\geq\mathbf{0},\sum_{k=0}^{m}\mu_k=1} 2\alpha^T\mathbf{1} - \alpha^T\left[\left(\sum_{k=0}^{m}\mu_k\tilde{K}_{tr}^k\right)\otimes yy^T\right]\alpha \\
&= \max_{\alpha:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1}}\ \min_{\mu:\mu\geq\mathbf{0},\sum_{k=0}^{m}\mu_k=1} \sum_{k=0}^{m}\mu_k[2\alpha^T\mathbf{1} - \alpha^T(\tilde{K}_{tr}^k \otimes yy^T)\alpha] \\
&= \max_{\alpha:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1}}\ \min_{k}[2\alpha^T\mathbf{1} - \alpha^T(\tilde{K}_{tr}^k \otimes yy^T)\alpha] \\
&= \max_{\alpha,t:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1},t\leq 2\alpha^T\mathbf{1}-\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha,k=0,\ldots,m} t \\
&= \min_{\alpha,t:\alpha^T y=0,\mathbf{0}\leq\alpha\leq C\mathbf{1},t\geq\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha-2\alpha^T\mathbf{1},k=0,\ldots,m} t
\end{aligned} \tag{18}$$

The first equality holds due to the special property of ℓ described above according to Boyd and Vandenberghe (2003). The second and third equalities hold due to the properties of the simplex defined by μ. The last two equalities hold due to the rewriting of the optimization problems in different formats. The last equality shows that the optimization problem in Equation (9) is equivalent to the optimization problem in Equation (10).

ACKNOWLEDGMENTS

We would like to thank Sumedha Gunewardena for proofreading and correcting this paper. We would also like to thank the anonymous reviewers for valuable comments and suggestions. We acknowledge funding support from Genome Canada through Ontario Genomic Institute.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Altschul, S.F., Madden, T.L., Schaffer, A.A., et al. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.

Andersen, E.D., and Andersen, A.D. 2000. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm, 197–232. *In* Frenk, H., Roos, C., Terlaky, T., et al., eds., *High Performance Optimization*. Kluwer Academic Publishers, New York.

Baldi, P., Chauvin, Y., Hunkapiller, T., et al. 1994. Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA* 91, 1059–1063.

Ben-Hur, A., and Brutlag, D. 2003. Remote homology detection: a motif based approach. *Proc. 11th Int. Conf. Intell. Syst. Mol. Biol.*

Boyd, S., and Vandenberghe, L. 2003. *Convex Optimization*. Stanford University, Stanford.

Chapelle, O., Weston, J., and Schoelkopf, B. 2002. Cluster kernels for semi-supervised learning. *NIPS (Neural Information Processing Systems).*

Cohn, H., Kleinberg, R., Szegedy, B., et al. 2005. Group-theoretic algorithms for matrix multiplication. *Proc. 46th Annu. Symp. Found. Comput. Sci.* 379–388.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., et al. 2001. On kernel-target alignment. *Adv. NIPS.*

Jaakkola, T., Diekhans, M., and Haussler, D. 2000. A discriminative framework for detecting remote protein homologies. *J. Comput. Biol.* 7, 95–114.

Kondor, R., and Lafferty, J. 2002. Diffusion kernels on graphs and other discrete structures. *Proc. ICML.*

Krogh, A., Brown, M., Mian, I., et al. 1994. Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.* 235, 1501–1531.

Kuang, R., Ie, E., Wang, K., et al. 2005. Profile-based string kernels for remote homology detection and motif extraction. *J. Bioinform. Comput. Biol.* 3, 527–550.

Lanckriet, G., Cristianini, N., Bartlett, P., et al. 2004. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.* 5, 27–72.

Leslie, C., Eskin, E., Weston, J., et al. 2002. Mismatch string kernels for SVM protein classification. *NIPS* 15.

Liao, C., and Noble, W.S. 2002. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proc. RECOMB.*

Min, R., Bonner, A., and Zhang Z. 2007. Modifying kernels using label information improves SVM classification performance. *Proc. Int. Conf. Mach. Learn. Appl.*.

Murzin, A.G., Brenner, S.E., Hubbard, T., et al. 1995. SCOP: a Structural Classification Of Proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536–540.

Sturm, J.F. 1999. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods Software* 12, 625–653.

Szummer, M., and Jaakkola, T. 2001. Partially labeled classification with Markov random walks. *Adv. NIPS* 14.

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

Scholkopf, B., and Smola, A.J. 2002. *Learning with Kernels*. MIT Press, Cambridge, MA.

Tsuda, K., Shin, H. H., and Scholkopf, B. 2005. Fast protein classification with multiple networks. *Bioinformatics* 21, 59–65.

Weston, J., Leslie, C., Ie, E., et al. 2005. Semi-supervised protein classification using cluster kernels. *Bioinformatics* 21, 3241–3247.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Zhu, X., Kandola, J., Ghahramani, Z., et al. 2005. Nonparametric transforms of graph kernels for semi-supervised learning. *Adv. NIPS* 17.

Address reprint requests to:
Dr. Zhaolei Zhang
Banting and Best Department of Medical Research
160 College Street, Room 608
Donnelly CCBR Building
Toronto, ON M5S 3E1, Canada

E-mail: Zhaolei.Zhang@utoronto.ca