FEATURE SELECTION FOR PAIRWISE SCORING KERNELS WITH APPLICATIONS TO PROTEIN SUBCELLULAR LOCALIZATION

Sun-Yuan Kung

Dept. of Electrical Engineering
Princeton University, USA

Man-Wai Mak

Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University,
Hong Kong SAR

ABSTRACT

In biological sequence classification, it is common to convert variable-length sequences into fixed-length vectors via pairwise sequence comparison. This pairwise approach, however, can lead to feature vectors with dimension equal to the training set size, causing the curse of dimensionality. This calls for feature selection methods that can weed out irrelevant features to reduce training and recognition time. In this paper, we propose to train an SVM using the full-feature column vectors of a pairwise scoring matrix and select the relevant features based on the support vectors of the SVM. The idea stems from the fact that pairwise scoring matrices are symmetric and support vectors are important for classification. We refer to this approach as vector-index-adaptive SVM (VIA-SVM). We compare VIA-SVM with other feature selection schemes—including SVM-RFE, R-SVM, and a filter method based on symmetric divergence (SD)—in protein subcellular localization. Results show that VIA-SVM is able to automatically bound the number of selected features within a small range. We also found that fusion of VIA-SVM and SD can produce more compact feature subsets without decreasing prediction accuracy, and that while VIA-SVM is superior for large feature-set size, the combination of SD and VIA-SVM performs better at small feature-set size.

Index Terms— Feature selection, pairwise scoring, kernel methods, SVM, subcellular localization

1. INTRODUCTION

The comparison of two temporal sequences are often hampered by the fact that the two sequences often have different lengths whether or not they belong to the same family. To overcome this problem, pairwise comparison between a sequence and a set of known sequences has been a popular scheme for creating fixed-size feature vectors from variable-length sequences [1–3]. Although this pairwise approach can usually create feature vectors with better discriminative properties, it also has its own limitation. The main problem is that the feature dimension is the same as the number of training patterns. This creates a curse of dimensionality, because the training set size could be very large. In fact, for the applications addressed in this paper, they are in the range of several thousands. The downside of such a curse of dimensionality is that it could hurt both training and recognition speed. Because a large number of sequences are being added to sequence databases in a daily basis, it is imperative to reduce the complexity of the pairwise scoring approach.

An obvious solution to the curse of dimensionality problem is to reduce the feature size and yet retaining the most important information critical for the classification of the training patterns. Research has found that just over 10% of the profile contributes 90% of the total score for positive training sequences [4], suggesting that some features are more relevant to classification than the others. The feature size can be reduced by either finding principle subspace or weeding out those less significant features. This paper takes the latter approach.

In this paper, we propose a novel method that uses the symmetric property of pairwise scoring matrices to select the relevant features. The method considers the columns of a pairwise scoring matrix as high-dimensional vectors and uses the column vectors to train a linear SVM. Because of the symmetric property of the score matrix, the row vectors with row indexes equal to the support vector indexes are identical to the support vectors. Also, because the support vectors define the decision boundary and margins of the SVM, they are critical for classification performance. Therefore, the support vector indexes are good candidates for selecting features for the column vectors, i.e., only the rows corresponding to the support vectors are retained. The column vectors with reduced dimension are then used to train another SVM for classification. Because the indexes of support vectors are used to select relevant features, we referred this method to as vector-index-adaptive SVM, or simply VIA-SVM.

We compared VIA-SVM with our recently proposed FDA-based feature selection [5], Guyon et al.'s SVM-RFE [6], and Zhang et al.'s R-SVM [7] in a subcellular localization benchmark, and found that this SV-based selection scheme not only avoids setting a cutoff point but also insensitive to the penalty factor in SVM training.

The paper is organized as follows. Section 2 defines the pairwise scoring kernels for SVM classification. The algorithmic details and theoretical analysis of VIA-SVM are discussed in Section 3. The method is then evaluated via a subcellular localization tasks in Section 4, which is followed by some concluding remarks in Section 5.

2. PAIRWISE SCORING KERNELS

Denote $\mathcal{D} = \{S^{(1)}, \ldots, S^{(T)}\}$ as a training set containing T protein sequences. Let us further denote the operation of PSI-BLAST[1] search given the query sequence $S^{(i)}$ as

$$\phi^{(i)} \equiv \phi(S^{(i)}) : S^{(i)} \longrightarrow \{\mathbf{P}^{(i)}, \mathbf{Q}^{(i)}\},$$

This work was in part supported by The Research Grant Council of the Hong Kong SAR (Project Nos. PolyU 5230/05E and A-PH18.

[1] To efficiently produce the profile of a protein sequence (called query sequence), the sequence is used as a seed to search and align homologous sequences from protein databases such as Swissprot [8] using the PSI-BLAST program [9].

where $\mathbf{P}^{(i)}$ and $\mathbf{Q}^{(i)}$ are the PSSM and PSFM of $S^{(i)}$, respectively.[2] Because these matrices are based on the information of a large number of sequences that are similar to the query sequence, they contain rich information about the remote homolog of the query sequence, which may help improves the prediction of subcellular locations and protein functions. Given the profiles of two sequences $S^{(i)}$ and $S^{(j)}$, we can apply the Smith-Waterman algorithm [10] and its affine gap extension [11] to align $\mathbf{P}^{(i)}$, $\mathbf{Q}^{(i)}$, $\mathbf{P}^{(j)}$, and $\mathbf{Q}^{(j)}$ to obtain the normalized profile-alignment score $\zeta(\phi^{(i)}, \phi^{(j)})$.[3]

The scores $\{\zeta(\phi^{(i)}, \phi^{(j)})\}_{i,j=1}^{T}$ constitute a symmetric matrix $\mathbf{Z}$ whose columns can be considered as T-dimensional vectors:

$$\boldsymbol{\zeta}^{(j)} = [\zeta(\phi^{(1)}, \phi^{(j)}) \ \ \ldots \ \ \zeta(\phi^{(T)}, \phi^{(j)})]^{\mathsf{T}} \quad j = 1, \ldots, T. \qquad (1)$$

An M-class protein prediction problem can now be solved by M one-vs-rest SVMs:

$$f_m(S) = \sum\nolimits_{j \in \mathcal{S}_m} y_{m,j} \alpha_{m,j} K(\phi(S), \phi(S^{(j)})) + b_m \qquad (2)$$

where S is an unknown sequence, $m = 1, \ldots, M$, $y_{m,j} \in \{+1, -1\}$, $\mathcal{S}_m$ contains the indexes of support vectors, $\alpha_{m,j}$ are Lagrange multipliers, and

$$K(\phi(S)), \phi(S^{(j)}) = g(\boldsymbol{\zeta}, \boldsymbol{\zeta}^{(j)})$$

is a kernel function.

Now we may consider the columns of a pairwise scoring matrix as high-dimensional vectors. This means that there are T feature vectors with dimension equal to the training set size. The T T-dimensional column vectors can be used to train M SVMs. Because of the high dimensionality, linear SVM is a preferred choice, i.e., $g(\boldsymbol{\zeta}, \boldsymbol{\zeta}^{(j)}) =< \boldsymbol{\zeta}, \boldsymbol{\zeta}^{(j)} >$. The class of S can then be obtained by $y(S) = \arg\max_{m=1}^{M} f_m(S)$, where M is the number of classes.

3. FEATURE SELECTION VIA VIA-SVM

The pairwise approach always results in feature vectors with extremely high dimension. This creates a problem known as the curse of dimensionality. An obvious solution is to reduce the feature size and yet retaining the most important information critical for the classification. The challenge thus lies in how to effectively determine those relevant features. Moreover, it is preferable to adopt a feature selection method which is tailor designed for the pairwise scoring vectors.

Assume that the training vectors are pre-arranged such that the vectors belonging to the same class are all grouped together, i.e., they are consecutively indexed. To design a feature selection algorithm for pairwise scoring vectors, we need to exploit the reflexive property of pairwise scoring matrices. The idea is based on the notion that support vectors are important for classification and pairwise scoring matrices are symmetric. (Namely, the elements of the i-th column of $\mathbf{Z}$ are identical to those in the i-th row.) This suggests a possible hypothesis:

> The support vector indexes are good candidates for selecting features for the column vectors, i.e., only the rows corresponding to the support vectors are retained.

Due to the symmetry property, if the j-th vector is a critical (supporting) vector for the decision boundary, then the j-th feature would also be a critical feature and therefore should be selected. We refer to this selection scheme as vector-index-adaptive SVM, or simply VIA-SAM.

3.1. Why Consider Only Support Vectors?

In VIA-SVM, the support vector indices are reused as feature selection indices. The use of support vectors to select relevant features is intuitively appealing because they are “critical” for establishing the decision boundary of SVM classifiers. Because of the symmetrical property of kernel matrices, the elements of the i-th column of $\mathbf{Z}$ are identical to those in the i-th row. If the i-th column of $\mathbf{Z}$ happens to be a support vector, the corresponding feature dimension (the i-th row of $\mathbf{Z}$) will also be critical for classification. On the other hand, non-support vectors are irrelevant for classification, so as their corresponding feature dimensions.

The weight vector of a linear SVM in Eq. 2 is given by $\mathbf{w} = \sum_{i \in \mathcal{S}} y_i \alpha_i \boldsymbol{\zeta}^{(i)}$, where the subscript m has been omitted for notational simplicity. The diagonal dominance [5] of the score matrix $\mathbf{Z}$ implies that a large value of α_i is likely to lead to a large value of w_i. By the same token, the non-support vectors are those correspond to $\alpha_i = 0$, and therefore their corresponding weight value w_i's are more likely to be smaller. Therefore, only those features corresponding to $\alpha_i > 0$ are considered for selection. Those corresponding to $\alpha_i = 0$ will be eliminated automatically.

The above interpretation of VIA-SVM is consistent with how SVM-RFE [6] selects features in that, in both methods, indexes with large weight will be chosen first. Moreover, they both prune the vectors/features corresponding to zero α_i. However, there is also an important difference, which lies in the treatment of the vectors/features corresponding to non-zero α_i. More exactly, in VIA-SVM, different types of support vectors receive different level of preference.

3.2. Treatments for Different Types of Support Vectors

Because the SVM-RFE takes the overall weight vector $\mathbf{w}$ into account, it only considers the Lagrange multiplers α_i but not the slack variables ξ_i. In contrast, the VIA-SVM considers both α_i (for identifying SVs) and ξ_i (for identifying outliers SVs and SVs that fall on or within the safety margin). In this sense, the VIA-SVM offers a more comprehensive coverage of all the critical factors made available by the SVM classifier.

In VIA-SVM, support vectors are not treated equally. Instead, they are divided into three levels of preferences:

1. **Most-preferred**: The SV is on the margin, if $0 < \alpha_i < C$ and $\xi_i = 0$, where C is the penalty factor in SVM training.
2. **Preferred**: The SV is in the fuzzy region, if $\alpha_i = C$ and $0 < \xi_i < 2$.[4]
3. **Non-preferred**: The SV is regarded as an outlier, if $\alpha_i = C$ and $\xi_i \geq 2$.

The reason of ruling out the outlier SVs is self-explanatory. The decision to have the marginal support vectors assigned the highest preference level can be justified on the basis that they offer relatively higher confidence than the fuzzy SVs.

The main difference between SVM-RFE and VIA-SVM lies in the differential treatment on the non-preferred support vectors. In

[2] The homolog information pertaining to the aligned sequences is represented by two matrices (profiles): position-specific scoring matrix (PSSM) and position-specific frequency matrix (PSFM). Both PSSM and PSFM have 20 rows and L columns, where L is the number of amino acids in the query sequence.

[3] See http://www.eie.polyu.edu.hk/∼mwmak/BSIG/PairProSVM.htm.

[4] When needed, the fuzzy region can be further divided into two subregions separated by the decision boundary.

SVM-RFE, features with $\alpha_i = C$ will be ranked high, regardless of whether the corresponding support vectors are outliers or not. On the other hand, VIA-SVM provides a mechanism to disregard these “unimportant” outliers, which may result in a more compact and representative set of features.

3.3. The VIA-SVM Algorithm

Feature selection in VIA-SVM is divided into two steps:

1. The score matrix $\mathbf{Z} = \{\zeta(\phi^{(i)}, \phi^{(j)})\}$ is used to train M SVMs (Eq. 2) from which M sets of support vector indexes $\mathcal{S}_m$ are determined. This results in a set of support vectors $\boldsymbol{\zeta}^{(j)} = [\zeta(\phi^{(1)}, \phi^{(j)}) \cdots \zeta(\phi^{(T)}, \phi^{(j)})]^\mathsf{T}$ for each class, where $j \in \mathcal{S}_m$.
2. For the m-th class, the indexes in $\mathcal{S}_m$ are used to select the feature dimensions (rows of $\mathbf{Z}$) of the column vectors to obtain vectors $\boldsymbol{\zeta}'^{(j)}$ of reduced dimension, where $j = 1, \ldots, T$. These vectors are then used to train another SVM for classification. This process is repeated for all classes.

These two steps are iterated N times (we set $N = 5$ in this work). Specifically, the features selected at the n-th iteration are used to train a new SVM in the $(n+1)$-th iteration, whose support vectors are subsequently used for determining the feature set in the $(n+2)$-th iteration, and so on. The classification accuracy on the training data at each iteration is recorded. At the end of the N-th iteration, the support vectors of the SVM with the highest training accuracy are used for selecting the final set of features. The column vectors with reduced dimension are then used to train another SVM for classification.

4. EXPERIMENTS ON PAIRWISE DATA

Two datasets were used for evaluating the performance of VIA-SVM and for comparing it against other feature selection algorithms. The first dataset is provided by Reinhardt and Hubbard [12]. It comprises 2427 amino acid sequences extracted from SWISSPROT 3.3, with each protein annotated with one of the four subcellular locations: cytoplasm, extracellular, mitochondrial, and nuclear. The second dataset was provided by Huang and Li [13]. It was created by selecting all eukaryotic proteins with annotated subcellular locations from SWISSPROT 41.0 and by setting the identity cutoff to 50%. The dataset comprises 3572 proteins with 11 classes. We used 5-fold cross validation for performance evaluation so that every sequence in the datasets will be tested.

4.1. Comparison of Three VIA Strategies

We shall first show some case studies and then we will provide theoretical justifications and tradeoff analysis on three VIA strategies.

Strategy 1 (ALL SVs): Select ALL SVs, i.e., marginal and fuzzy SVs with $0 < \alpha_i \leq C$.

Strategy 2 (Remove the non-preferred): Select all but outlier SVs (i.e., only keep those with $\xi_i < 2$).

Strategy 3 (Only the most-preferred): Select only “pure” marginal SVs ($\alpha_i < C$), while excluding those fuzzy and outlier SVs (i.e., $\alpha_i = C$).

Because Strategy 1 includes all SVs regardless of their types, it is likely to cause over selection, particularly when the penalty factor C is small. Strategy 2 is based on the notion that SVs lying beyond the margin of the opposite class are deemed to be outliers and therefore should be excluded. Note that in this strategy misclassified SVs that lie within the margin of separation will still be selected, which may lead to over selection when the penalty factor C is very small. In Strategy 3, in addition to outliers, SVs that are likely to be misclassified will also be excluded. In some cases, there may be many SVs falling on the fuzzy regions, and therefore, excluding all of these SVs may lead to under selection.

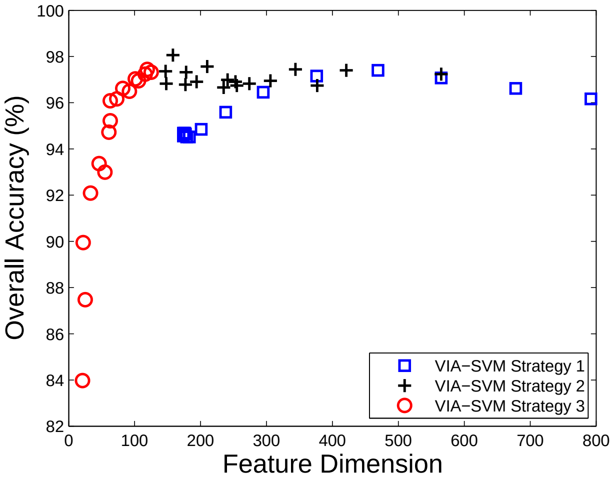


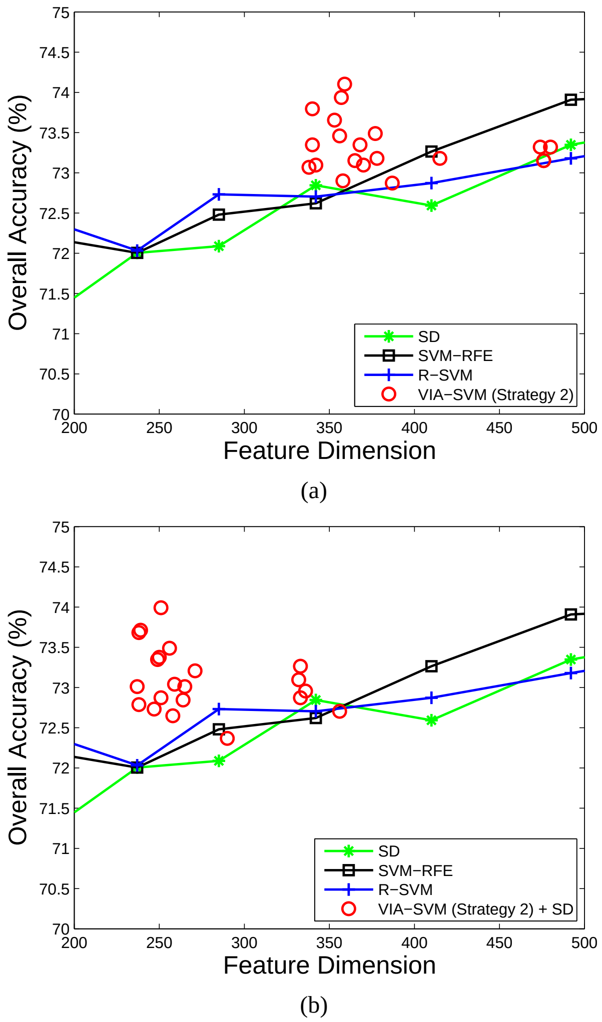
Fig. 1. Prediction performance of three strategies of VIA-SVM on the Reinhardt and Hubbard’s dataset when the penalty factor C varies from 0.004 to 8,000. See Section 4 for details of the strategies.

Figure 1 shows the performance of Strategies 1, 2, and 3 when the penalty factor C varies from 0.004 to 8,000. Note that the number of selected features (feature dimension) is automatically determined by the SVMs. The results show that Strategy 1 tends to select more features, conforming our earlier hypothesis that including all SVs will lead to over selection. These case studies suggest that Strategy 2, which selects all SVs except the outliers, seems to be the least sensitive to the penalty factor, because it can keep the number of features within a small range and maintain the accuracy at a constant level for a wide range of C.

4.2. Comparison with Other Classifiers

We compared the proposed VIA-SVM (Strategy 2) with our recently proposed symmetric divergency (SD) [5], SVM-RFE [6], and R-SVM [7] in the subcellular localization benchmarks mentioned earlier. Note that all these reference methods do not make use of the symmetric property of the pairwise scoring matrices in the selection process, because they are primarily designed for gene selections in microarray data where expression matrices are neither square nor symmetric.

Figure 2(a) shows the performance of these methods. Evidently, VIA-SVM is superior to other three methods in two aspects: (1) It outperforms the others at almost all feature dimensions and (2) it automatically bounds the number of selected features within a small range. A drawback of SD, SVM-RFE, and R-SVM is that they require a cutoff point for stopping the selection. On the other hand, VIA-SVM is insensitive to the penalty factor in SVM training and



(a)

(b)

Fig. 2. Prediction performance of symmetric divergence (SD), SVM-RFE, R-SVM, and VIA-SVM (Strategy 2) on Huang and Li's dataset. (b) Same as (a) but with the fusion of SD and VIA-SVM.

can avoid the need to set a cutoff point for stopping the feature selection process.

As shown in Figure 2(a), the performance of VIA-SVM is always better than that of other methods, but at the price of having a larger number of features. To overcome this weakness, we propose to combine this wrapper approach with a filter approach (SD in this case) to raise the selection standard. This is discussed in the following subsection.

4.3. Cascaded Fusion of SD and VIA-SVM

The feature selection process is divided into two stages.

Stage 1: Use VIA-SVM (Strategy 2) to select all-but-outlier SVs, i.e., only keep those with $\xi_i < 2$.

Stage 2: Use SD to sort the features found in Stage 1 and keep the most relevant x%.

In this work, we set x to 70. Figure 2(b) shows the fusion results on the Huang and Li's dataset mentioned earlier. A comparison between this figure and Figure 2(a) reveal that fusion can produce more compact feature subsets without decreasing prediction accuracy. We also note that while VIA-SVM is superior to others for large feature-set size (cf. Figure 2(a)), the combination of SD and VIA-SVM performs better at small feature-set size (cf. Figure 2(b)).

5. CONCLUSION

Based on several subcellular localization experiments, the VIA-SVM appears to be very promising. First, it seems to reach a useful dimension reduction while conceding minimum sacrifice on accuracy. This is supported by serval comparative studies. Second, the VIA-SVM can derive the subset size as well as the subset itself without having to guess the "right" penalty factor in SVM training nor the need to set a "right" cutoff point. In short, it automatically bounds the number of selected features within a small range.

6. REFERENCES

[1] L. Liao and W. S. Noble, "Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships," *J. Comput. Biol.*, vol. 10, no. 6, pp. 857–868, 2003.

[2] J. K. Kim, G. P. S. Raghava, S. Y. Bang, and S. Choi, "Prediction of subcellular localization of proteins using pairwise sequence alignment and support vector machine," *Pattern Recog. Lett.*, vol. 27, no. 9, pp. 996–1001, 2006.

[3] J. Guo, M. W. Mak, and S. Y. Kung, "Eukaryotic protein subcellular localization based on local pairwise profile alignment SVM," in *2006 IEEE International Workshop on Machine Learning for Signal Processing (MLSP'06)*, 2006, pp. 391–396.

[4] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie, "Profile-based string kernels for remote homology detection and motif extraction," *J. Bioinfom. Comput. Biol.*, vol. 3, pp. 527–550, 2005.

[5] M. W. Mak and S. Y. Kung, "A solution to the curse of dimensionality problem in pairwise scoring techniques," in *Int. Conf. on Neural Information Processing*, 2006, pp. 314–323.

[6] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389422, 2002.

[7] X. G. Zhang, X. Lu, Q. Shi, X. Q. Xu, H. C. E. Leung, L. N. Harris, J. D. Iglehart, A. Miron, J. S. Liu, and W. H. Wong, "Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data," *BMC Bioinformatics*, vol. 7, no. 197, 2006.

[8] http://www.expasy.org/sprot, ," .

[9] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, pp. 3389–3402, 1997.

[10] T. F. Smith and M. S. Waterman, "Comparison of biosequences," *Adv. Appl. Math.*, vol. 2, pp. 482–489, 1981.

[11] O. Gotoh, "An improved algorithm for matching biological sequences," *J. Mol. Biol.*, vol. 162, pp. 705–708, 1982.

[12] A. Reinhardt and T. Hubbard, "Using neural networks for prediction of the subcellular location of proteins," *Nucleic Acids Res.*, vol. 26, pp. 2230–2236, 1998.

[13] Y. Huang and Y. D. Li, "Prediction of protein subcellular locations using fuzzy K-NN method," *Bioinformatics*, vol. 20, no. 1, pp. 21–28, 2004.