Efficient Remote Homology Detection

Antolin Janssen[1], Evgeni Tsivtsivadze[1], Jorma Boberg[2], Tjeerd M.H. Dijkstra[1], and Tom Heskes[1]

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
{firstname.lastname}@science.ru.nl
[2] Department of Information Technology, University of Turku, Joukahaisenkatu 3-5 B, 20520 Turku, Finland
{firstname.lastname}@it.utu.fi

Abstract. We propose an efficient multi-class classification algorithm for remote homology detection (RHD). Unlike methods that treat RHD as a set of binary classification tasks, our algorithm solves a single multi-class classification problem by incorporating information about class-wise correlations among the proteins using joint kernel functions. Furthermore, the proposed method leads to notable reduction in computational time compared to binary classification algorithms. We evaluate our method on the Structural Classification of Proteins database and show that performance is better or comparable to several state-of-the-art algorithms for protein classification and remote homology detection.

1 Introduction

Remote homology detection (RHD) is a challenging and well studied problem in computational biology. A common approach for detecting remote homologs is based on the assumption that similarity of the protein sequences may also imply structural and functional similarity. Algorithms calculating similarity between proteins such as BLAST, PSI-BLAST (e.g. [1]) have been successfully used for this task. Recently a number of machine learning methods have been shown to be particularly applicable for the task where it is necessary to infer structure and function of the protein. The task of predicting the biological function of a protein can be formalized as a classification problem where proteins belonging to the same family (having same function) have the same class label.

This discriminative approach for detecting remote homology has been proposed in several studies e.g. [2, 3]. In recent work [4] the RHD task is treated as a ranking problem. It has been shown that algorithms developed within this (classification/ranking) framework are performing significantly better compared to the methods based solely on similarity evaluation (e.g. BLAST).

When considering detection of multiple homologs within a classification framework a common approach is to divide the task into several binary classification problems e.g. [3], each constituting the task of predicting homologs belonging to single family. This approach requires training of as many binary classifiers as

the number of available families and can be computationally prohibitive. Furthermore, by considering classification problems separately one does not take into account possible correlations between the "multi-homolog" proteins e.g. the proteins that are associated with more than one class label. In this study, we address the problem of protein sequence classification by using an efficient multi-class classification algorithm proposed in [5]. Our method leads to a significant decrease in computation time and gives results comparable to some of the state-of-the-art methods for RHD. We evaluate the performance of our algorithm on Structural Classification of Proteins (SCOP) database [6].

2 Algorithm

In this work we follow suggestions given in [7] and [8] on how to treat multi-class classification problems. The approach is known as error-correcting output codes (ECOC). The key idea is to construct the coding matrix $C \in \{-1, +1\}^{\kappa \times p}$, where p is some positive integer and κ is the number of classes, such that the rows of the matrix have good error correcting properties (e.g. a large Hamming distance). We note that this setting is a generalization of a one-versus-all scheme. Given a new example $\mathbf{x}'$ from input space $\mathcal{X}$, we can predict the corresponding label y' by finding the row of the coding matrix that is "closest" to $\mathbf{f} = (f_1(\mathbf{x}') \dots f_p(\mathbf{x}'))$, where $f_s(\mathbf{x}'), s = 1, \dots, p$ are the prediction functions constructed for each column of the output matrix.

1) Multiple Output Prediction

Our choice of basic learning algorithm is regularized least-squares (RLS) [9]. RLS has been shown to perform comparably to state-of-the-art supervised learning algorithms (e.g. SVM) and has several computational advantages one of which is efficient extension to handle multiple output prediction problems. Suppose instead of having a single column matrix for the outputs, we now have an $n \times p$-matrix, where p is the number of outputs and n is the number of data points. Denote the output matrix as $Y \in \mathbb{R}^{n \times p}$. In the context of multi-class classification using ECOC the rows of Y would be the same as those of the coding matrix C. We use the dataset $\mathcal{D} = (X, Y)$ originating from a set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ of data points, where $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)^t \in \mathcal{X}^n$, $Y = (\mathbf{y}_1, \dots, \mathbf{y}_n)^t \in \mathbb{R}^{n \times p}$. We write the minimization problem as

$$\min_{\mathbf{f}} J(\mathbf{f}, \mathcal{D}) = \sum_{i=1}^{p} \sum_{j=1}^{n} (y_{ji} - f_i(\mathbf{x}_j))^2 + \lambda \|f_i\|_{\mathcal{H}}^2, \tag{1}$$

thus, the problem at hand boils down to solving p independent regression tasks. It can be shown, when solving (1) in dual [9], that the coefficient vectors $A \in \mathbb{R}^{n \times p}$ that determine a minimizer of (1) can be computed as $A = (K + \lambda I)^{-1} Y$, where $K \in \mathbb{R}^{n \times n}$ is a kernel matrix. The problem of finding the optimal hypothesis can be solved by finding the coefficients $a_h^s, 1 \leq h \leq n$, for every output matrix

column $Y_{\cdot,s}$, $s = 1, \ldots, p$. We note that using a square loss function leads to an efficient multi-output regression solution, namely we obtain predictions for each output by inverting the kernel matrix only once and therefore the complexity of the algorithm is hardly increased compared to a standard single output problem.

2) Co-regularization on Joint Kernels

The co-regularization approach (e.g. [10, 11]) is used to take unlabeled data into the learning process. In this work we define a co-regularized algorithm that operates in a fully supervised setting, namely when the error of each function on the labeled data is small and prediction functions constructed using different feature spaces on the *same* dataset are similar. We consider N feature spaces $\mathcal{H}_1, \ldots, \mathcal{H}_N$ along with their corresponding kernel functions $k^{(v)} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, 1 \leq v \leq N$, in our co-regularization approach. We construct some of the views using a joint kernel function on inputs-outputs such as following Gaussian: $k((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = \exp\left[\frac{-\left|\left|(\mathbf{x},\mathbf{y})-(\mathbf{x}',\mathbf{y}')\right|\right|^2}{2\sigma^2}\right]$. In the classification task, we search for a vector $\mathbf{f} = (\mathbf{f}_1, \ldots, \mathbf{f}_N) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_N$ of prediction functions which minimizes

$$J(\mathbf{f}, \mathcal{D}) = \sum_{v=1}^{N} c(\mathbf{f}^{(v)}, \mathcal{D}) + \lambda \sum_{v=1}^{N} \|\mathbf{f}^{(v)}\|^2_{\mathcal{H}_v} + \nu \sum_{v,u=1}^{N} \tilde{c}(\mathbf{f}^{(v)}, \mathbf{f}^{(u)}, \mathcal{D}), \qquad (2)$$

where $\lambda, \nu \in \mathbb{R}^+$ are regularization parameters and $\tilde{c}$ is a loss function measuring the disagreement between the prediction functions of the views. By minimizing the objective function in (2) we obtain coefficient vectors for predicting the unknown class label.

3) Class Label Estimation

If joint kernels are used to construct the feature spaces in co-regularization, the unknown label has to be provided to the algorithm when making a prediction. Due to the fact that in multi-class classification problems the set of possible class labels that can be assigned to the new example is known, our strategy is to compute predictions by considering all possible labels. According to the representer theorem [12] the prediction function for (2) can be written as $\mathbf{f}^{(v)}(\mathbf{x}', \mathbf{y}') = \sum_{i=1}^{n} \mathbf{a}_i^{(v)} k^{(v)}((\mathbf{x}', \mathbf{y}'), (\mathbf{x}_i, \mathbf{y}_i))$, where $\mathbf{x}'$ is an unseen example, $\mathbf{y}' \in \{C_{1,\cdot}, \ldots C_{\kappa,\cdot}\}$ is the encoding of the label, and $\mathbf{a}_1^{(v)}, \ldots, \mathbf{a}_n^{(v)} \in \mathbb{R}^p$ are the coefficients. We take the average over all views $\mathbf{f}^*(\mathbf{x}', \mathbf{y}') = \frac{\sum_{v=1}^{N} \mathbf{f}^{(v)}(\mathbf{x}', \mathbf{y}')}{N}$, and define the loss based decoding function that calculates the distance between the prediction and the rows of the coding matrix $d_L(C_{i,\cdot}, \mathbf{f}^*(\mathbf{x}', \mathbf{y}')) = \sum_{j=1}^{p} (C_{i,j} - f_j^*(\mathbf{x}', \mathbf{y}'))^2$. Finally, we select the label to be assigned to the new data point $\mathbf{x}'$ by choosing the class i^* with the smallest distance: $i^* = \text{argmin}_i \, d_L(C_{i,\cdot}, \mathbf{f}^*(\mathbf{x}', \mathbf{y}'_i))$.

3 Experiments

To evaluate the performance of our algorithm we conduct experiments on the SCOP database [6]. The aim is to classify protein domains into SCOP-superfamilies. We follow the experimental setup and use the dataset described in [3]. 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 test sets in the same ratio as positive examples. The protein sequences belonging to different families but to the same superfamily are considered to be remote homologs in SCOP.

For each family (single binary classification task) we perform 10-fold cross-validation on the training set for selecting appropriate regularization and kernel parameters (we use the spectrum kernel [13]) for our algorithm. The RLS algorithm has regularization parameter λ that controls the trade-off between the minimization of the training error and the complexity of the regression function. Once the best parameters are found the algorithm is trained on the complete training set with the selected parameters and its performance is evaluated on a separate test set.

For the multi-class classification setting we randomize the complete dataset and use two thirds for training and reserve one third for testing purposes. We ensure that at least one example belonging to every class is present both in training and test sets. We again use 10-fold cross-validation on the training set to select the best parameters. Here in addition to λ we search for ν that controls the influence of the co-regularization term. Once the parameters are estimated we train the algorithm on the complete training set and evaluate the performance on the test set. Finally, we perform the whole experiment (multi-class classification) 10 times (every time re-randomizing whole dataset) and average the test set results.

To compare binary and multi-class classification we use the percentage of correctly classified instances above the random baseline. We note, that the random baseline for the multi-class classifier with 54 families is 1.85%, while for the binary classifier it is 50%. For example, if binary classifier predicts the correct label for 90% of the test examples we say that the percentage of correctly classified instances above the random baseline equals 40%. The results of the experiments are summarized in Figure 1. The plot shows the performance of the RLS algorithms with the spectrum kernel when treating RHD as a multiple binary classification problem. A point in the figure indicates the classification performance of the algorithm on a single family from SCOP dataset. The vertical lines depict the result obtained by our multi-class classification algorithm (JMcoreg) and standard multi-class SVM. It can be noted that our multi-class algorithm obtains the performance that is better than standard multi-class SVM and the average of RLS in binary classification, while notably decreasing computational time. Finally, we evaluate statistical significance of the performance differences over 10 reruns between SVM and our algorithm using a Wilcoxon signed-ranks

test. The test indicates that the differences are statistically significant ($p < 0.05$).

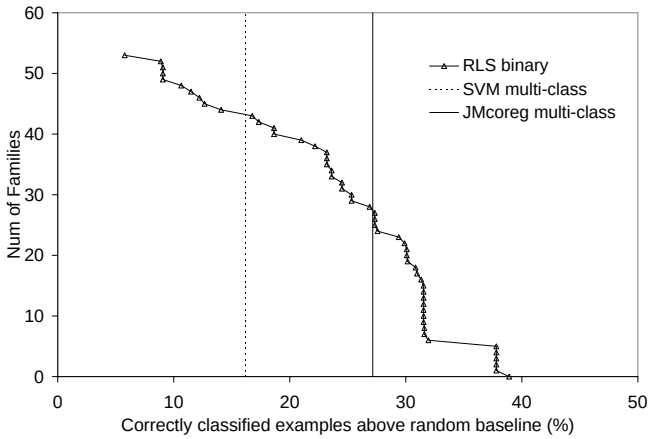

Fig. 1. Family-by-family performance comparison of RLS with the spectrum (subsequences of length 3) kernel. The coordinates of each point represent percentage of correctly classified examples (above the random baseline) obtained for one SCOP family. The vertical lines show the performance of the multi-class classification algorithms.

4 Conclusions

In this study, we propose an algorithm for efficient remote homology detection. Unlike many other methods our multi-class classification algorithm uses a co-regularization framework and allows construction of expressive features on input-output spaces. Furthermore, it leads to notable decrease in training time compared to algorithms that treat remote homology detection as a set of binary classification problems.

We test the performance of our algorithm on the Structural Classification of Proteins database and show that results are better or comparable to that of the state-of-the-art learning algorithms frequently applied for protein classification and remote homology detection[3], while leading to notable benefits in runtime.

[3] We have also conducted several experiments where the performance of the learning algorithm for binary classification problems was evaluated using AUC measure. By examining the results reported in [3, 13, 14], etc. we further suggest that proposed algorithm often gives results better or comparable to the algorithms frequently used for remote homology detection.

Acknowledgments

We acknowledge support from the Netherlands Organization for Scientific Research, in particular a Vici grant (639.023.604) and two CLS grants (635.100.020 and 635.100.026).

References

1. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic acids research **25** (1997) 3389–3402
2. Jaakkola, T., Diekhans, M., Haussler, D.: A discriminative framework for detecting remote protein homologies. Journal of Computational Biology **7** (2000) 95–114
3. 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** (2003) 857–868
4. Melvin, I., Weston, J., Leslie, C.S., Noble, W.S.: Rankprop: a web server for protein remote homology detection. Bioinformatics **25** (2009) 121–122
5. Tsivtsivadze, E., Birlutiu, A., Heskes, T.: Joint map co-regularization for multi-class classification. (2010) Submitted.
6. Hubbard, T.J.P., Murzin, A.G., Brenner, S.E., Chothia, C.: Scop: a structural classification of proteins database. Nucleic Acids Research **25** (1997) 236–239
7. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research **2** (1995) 263–286
8. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research **1** (2001) 113–141
9. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In: Advances in Learning Theory: Methods, Model and Applications, Amsterdam, IOS Press (2003) 131–154
10. Sindhwani, V., Niyogi, P., Belkin, M.: A co-regularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML Workshop on Learning with Multiple Views. (2005)
11. Brefeld, U., Gärtner, T., Scheffer, T., Wrobel, S.: Efficient co-regularised least squares regression. In: Proceedings of the International Conference on Machine learning, New York, NY, USA, ACM (2006) 137–144
12. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: Proceedings of the Conference on Computational Learning Theory, London, Springer (2001) 416–426
13. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for svm protein classification. In: Pacific Symposium on Biocomputing. (2002) 566–575
14. 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** (2005) 527–550