Maximum Feasibility Approach for Consensus Classifiers: Applications to Protein Structure Prediction

Aleksey Porollo[1], Rafal Adamczak[1], Michael Wagner[1] and Jaroslaw Meller[1,2]

[1]Pediatric Informatics, 3333 Burnet Avenue, Children's Hospital Research Foundation, Cincinnati, OH 45229, USA
{aporollo, radamczak, mwagner}@chmcc.org
[2]Department of Informatics, Nicholas Copernicus University, 87-100 Torun, Poland
jmeller@chmcc.org

Abstract

A novel strategy to optimize consensus classifiers for large classification problems is proposed, based on Linear Programming (LP) techniques and the recently introduced Maximum Feasibility (MaxF) heuristic for solving infeasible LP problems. For a set of classifiers and their normalized class dependent scores one postulates that the consensus score is a linear combination of individual scores. We require this consensus score to satisfy a set of linear constraints, imposing that the consensus score for the true class be higher than for any other classes.. Additional constraints may be added in order to impose that the margin of separation (difference between the true class score and false classes scores) for the consensus classifier be larger than that of the best individual classifier. Since LP problems defined this way are typically infeasible, approximate solutions with good generalization properties are found using interior point methods for LP in conjunction with the MaxF heuristic. The new technique has been applied to a number of classification problems relevant for protein structure prediction.

1. Introduction

Ensemble classifiers are an active area of research in the field of machine learning [1,2]. Many strategies, such as simple voting, linear combination based methods or boosting [3-6], have been proposed to find an improved consensus classifier, given a number of individual classifiers. Consensus classifiers are often able to improve significantly on the classification accuracy. Some important and relevant in bioinformatics examples include applications of neural network based classifiers for protein secondary structure prediction [7] or combining various individual scores into a consensus score for gene prediction [8].

Here, we introduce a novel strategy to optimize consensus classifiers for large problems, using LP techniques and the Maximum Feasibility heuristic for solving infeasible LP problems [9,10]. For a set of classifiers and their normalized class dependent scores one postulates that the consensus score is a linear combination of individual scores. Such defined total score is required to satisfy a set of linear constraints, imposing that the consensus score for the true class is higher than for any other class for each data point in the training.

The resulting LP problems are infeasible for classification problems that are not linearly separable in the feature space of individual classifiers scores. Our strategy to find an approximate solution is to identify a possibly large subset of inequalities that can be satisfied. In other words, we identify a subset of data points that can be classified using linear decision boundaries, with points difficult to classify excluded from the training. Such approximate solutions that achieve high accuracy and have good generalization properties may be found

efficiently using interior point methods for LP in conjunction with the MaxF heuristic.

Here, we briefly revisit the MaxF heuristic and then formally introduce the new approach for finding linear combination based classifiers and discuss strategies for solving the resulting infeasible LP problems (Methods section). The new technique is then applied to a number of classification problems relevant for protein structure prediction, including secondary structure and membrane domains prediction (Results section).

The protein folding problem, which is one of the central challenges in computational biology, consists of predicting the three-dimensional structure of a protein from its amino acid sequence. The methodology and modeling aspects of protein folding have been vastly discussed in the literature [11]. For the sake of completeness it suffice to say here that predicting secondary structures, i.e. locally ordered conformations taking shape of helices or beta strands, greatly facilitates fold recognition and functional annotations. The same concerns membrane domains prediction.

2. Methods

2.1. Maximum Feasibility Heuristic

The Maximum Feasibility (MaxF) [9,10] heuristic aims at finding an approximate solution, which satisfies a possibly large subset of an infeasible set of inequalities. The MaxF procedure is based on a special property of **interior point** algorithms for LP. Without a function to optimize the interior point algorithm places the solution at the "maximally feasible" point, which is away from any individual constraint. For problems with bound feasible polyhedra interior point algorithms converge to the so-called analytic center, when no objective function is used [12]. The idea behind MaxF heuristic is that the "maximally feasible" partial solution is likely to satisfy more constraints than an off-centered guess.

The MaxF heuristic starts from a certain initial guess of the solution and the subset of all the constraints that are satisfied by this initial guess. A series of "maximally feasible" approximations is then computed. The subset of all the inequalities satisfied by the previous approximation, which defines a feasible polyhedron, is solved using an interior point method. The new solution becomes our next "maximally feasible" approximation and satisfies at least as many constraints as the previous partial solution. If no further constraints are satisfied the procedure stops.

The choice of the initial guess of the solution is critical for the success of the MaxF heuristic. Finding the largest feasible subset of an infeasible problem is a NP-hard problem [13] and obtaining a good approximation cannot be guaranteed. However, in practice we observe significant improvement with respect to initial approximate solutions that are carefully chosen using a priori knowledge [9,10].

Another way to obtain an appropriate initial guess is to solve an elastic LP (eLP) problem, with a positive slack variable z_i added to each constraint:

$$\min \sum z_i \text{ subject to } \mathbf{A}\boldsymbol{\alpha} + \mathbf{z} \geq \varepsilon, \quad \varepsilon > 0, \ \mathbf{z} \geq 0 \ . \qquad (1)$$

Here, $\boldsymbol{\alpha}$ denotes the vector of unknowns that are target of optimization and **A** denotes the constraint matrix. The LP problem defined in (1) is always feasible and, by adding the sum of slack variables as the objective function, allows one to find approximate solutions of the original infeasible problem. We applied here the latter strategy.

The eLP finds a solution that effectively minimizes the misclassification error (sum of slacks), and might be influenced by outliers. Nevertheless, we observe in practice that it provides good initial approximations for

the problems considered here. These initial solutions are then improved in terms of margin of separation by subsequent MaxF iterations. Starting from a subset of separable data points, for which the slack variables are equal to zero, MaxF places the separating hyperplanes away from all the data points that are correctly classified by the initial guess.

The pPCx package by one of us (MW), which is a parallel interior point LP solver, was used to obtain results presented in this paper. We would like to comment that interior point methods for LP have superior, polynomial complexity and are very efficient. Problems with millions of constraints and hundreds of variables may be solved, e.g., in a few minutes on a cluster of Xeon CPU's, using the pPCx package [10].

2.2. MaxF based consensus classifiers

Let us consider a supervised classification problem with *N* real vectors from a certain feature space *X,* divided into *K* classes. A discrete set of class labels, conveniently chosen as $1,\ldots,K$, will be referred to as *Y*. A classifier Q is then a mapping from *X* to *Y*. For clarity of notation the *k*th class will be alternatively labelled as C_k - $\mathbf{x} \in X$ is classified as belonging to class C_k, if $Q(\mathbf{x}) = k$.

Consider now a number of individual models, M_i, $i = 1,\ldots,p$, that provide estimates for conditional probabilities of class C_k given the model and a vector in the feature space, $P(C_k \mid \mathbf{x}; M_i)$. For each model we define an individual classifier Q_i as:

$$Q_i(\mathbf{x}) = \underset{k=1,\ldots,K}{\arg\max}\ P(C_k \mid \mathbf{x}; M_i)\,. \qquad (2)$$

In other words, a data point **x** is assigned to the class with the highest probability**.** The goal is then to combine the individual models into a mixture (consensus) model.

We define a consensus classifier in the form of a linear combination of individual classifiers:

$$P(C_k \mid \mathbf{x}; M_c) = \sum_{i=1}^{p} \alpha_i P(C_k \mid \mathbf{x}; M_i)\,. \quad (3)$$

Note that the coefficients of the linear combination, which will be a target for optimization, are class independent here (as opposed to more general models with class dependent coefficients – see Results section). Linear decision boundaries for the consensus classifier are defined using again the simple rule:

$$Q_c(\mathbf{x}) = \underset{k=1,\ldots,K}{\arg\max}\ P(C_k \mid \mathbf{x}; M_c)\,. \qquad (4)$$

In supervised classification problem each training vector is assigned to its "true" class, which will also be called its "native" state in the context of applications to protein structure prediction. The true (or native) class will be referred to as C_n, where $Q^*(\mathbf{x}) = n$ is the true classifier (with the implicit dependence of index *n* on **x**).

In order to impose correct consensus predictions in the training, the following inequality constraints (with one inequality per data point) are used:

$$\sum_{i=1}^{p} \alpha_i P_i(C_n) \geq \sum_{i=1}^{p} \sum_{k \neq n} \alpha_i P_i(C_k)\,, \quad (5)$$

where coefficients $P_i(C_k) = P(C_k \mid \mathbf{x}; M_i)$ of the constraint matrix are obtained by applying individual classifiers. Thus, for each data point an inequality as defined in (5) is used to impose that consensus classifier of equation (3) assigns the highest (and larger than 0.5) probability to the true class of that point. A solution to the set of inequalities defined in (5) provides the coefficients α_i, and thus, a linear combination based classifier as defined in (3).

If the problem is feasible, i.e. when the data is linearly separable, the set of inequalities in (5) may be solved efficiently using LP techniques. Typically however, the problem is infeasible and heuristic approaches, such as combination of the elastic LP and

MaxF need to be applied. MaxF was shown before, in the context of protein structure recognition, to effectively filter out outliers that make impossible to separate exactly data points belonging to different classes [9].

The basic idea here is similar. By finding an approximate solution to an infeasible problem defined in (4) we identify points that are difficult to classify. Subsequent iterations of MaxF include only those data points that can be classified correctly (i.e. points that result in inequalities that are satisfied by current guess of the solution). Thus, the linear decision boundaries are optimized for a subset of data points that are separable. In addition, due to the "central" properties of interior point methods, discussed in the Introduction, the solutions that we obtain are away from any individual constraint, providing (at least in principle) a wide margin of separation and a good generalization.

Formulating the problem in terms of linear optimization with constraints opens a way for flexible generalizations. For example, one may impose that the margin of separation between the true and other classes should be at least as wide for the consensus classifier as for the individual classifier, which achieves best separation for a given point. This can be achieved by imposing (again for each vector in the training) additional inequalities of the following form:

$$P_{\mathrm{c}}(C_n) - P_{\mathrm{c}}(C_k) \geq \max_{i=1,p} [P_i(C_n) - P_i(C_k)]. \qquad (6)$$

Moreover, instead of considering positive and normalized conditional probabilities one may introduce a generalized classification problem in terms of real scores. One may also weaken the condition of equation (5) by decoupling inequalities for classes other than native. Replacing conditional probabilities for the *i*-th model by the corresponding score, S_i, and introducing one inequality for each non-native state we obtain the following set of inequalities:

$$\sum_{i=1}^{p} \alpha_i S_i(C_n, \mathbf{x}) \geq \sum_{i=1}^{p} \alpha_i S_i(C_k, \mathbf{x}) \quad \forall k \neq n \ \forall \mathbf{x}. \qquad (7)$$

The decision is made as previously: the class with the highest score is assigned to each data point.

3. Results

Preliminary results obtained using the new eLP/MaxF-based approach for protein membrane domain and secondary structure prediction are summarized in Table 1 and Tables 2 and 3, respectively. A set of inequalities defined in equation (7) is solved for each problem using the approach defined in section 2.1. The results are compared to that of several machine learning techniques, including decision trees (SSV [14] and C4.5), k-Nearest Neighbors, adaptable radial basis functions Neural Networks (FSM) [14], Support Vector Machines (SVMs) [15] and Linear Discriminat Analysis (LDA) [16].

Method	Training	Control	Software
Majority	72.1%	67.0%	-
kNN k=10	86.8%	71.8%	Tooldiag
SSV D. tree	85.4%	70.5%	GhostMiner
FSM	85.1%	71.1%	GhostMiner
SVM	86.7%	74.0%	SVMLight
LDA	83.8% (CV)	74.0%	Tooldiag
eLP/MaxF	86.7 (86.1)%	73.1 (72.8)%	pPCx

Table 1. Accuracy for membrane domain prediction.

For membrane domain prediction we used as the training set a curated set of 68 proteins that contained membrane domains and an additional set of 25 proteins as control. Out of the total number of 19,404 residues in the training, 7,704 were in membrane domains. The goal of the prediction is to assign to each amino acid residue one of the two states: membrane or non-membrane. We used as individual weak classifiers (or rather features in this case) twelve statistical scores, each of them assigning a score to a different type of profile (e.g. triplet of residues around the central residue) according to

observed frequency of this profile in a given class in the training set. These individual scores have low prediction accuracy (worse than the baseline). Nevertheless, as can be seen from Table 1, linear discrimination methods (linear SVM, LDA and eLP/MaxF) perform relatively well. Despite the fact that finding a large feasible subset could be potentially hindered by the low quality of individual "classifiers" (features), the LP based approach finds a solution close to that of LDA in terms of accuracy (73.1% when using 24 class dependent coefficients of linear expansion (7) and 72.8 with only 12 class independent coefficients). By combining predictions for adjacent residue the accuracy may be further elevated by about 10%, making this kind of simple predictor an attractive component of a more accurate membrane domain prediction system.

Method	Training:Pfam	Control:S174	Software
Majority	68.3%	67.5%	-
kNN k=10	71.8% (CV)	69.5%	Tooldiag
C4.5 D. tree	95.3%	64.3%	C4.5
LDA	73.5% (CV)	71.0%	Tooldiag
eLP/MaxF	78.8 (73.2)%	70.3 (69.7)%	pPCx

Table 2. Accuracy for coil vs. non-coil prediction.

The second problem that we consider is considerably larger. The training was derived from the Protein Families (Pfam) database and consists of 174,792 residues, which are divided into two classes: coil (no regular secondary structures) and non-coil (helices, beta strands). The feature space consists of 22 different statistical profiles, derived similarly to those for membrane proteins. Despite the still rather moderate size of the problem, we were unable to use either SVMLight or GhostMiner. Again, despite the fact that individual scores have low predictive power, their eLP/MaxF-optimized linear combination achieves accuracy close to that of LDA on the control set of 174 proteins with no homology to proteins in the training.

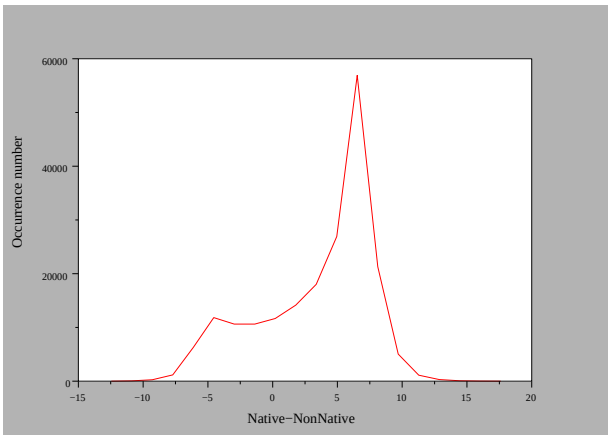

Figure 1. Distribution of differences between consensus scores for native and highest scoring non-native states.

The third problem deals with a consensus of 19 well-trained, NN-based classifiers for the three state (coil, helix, beta strand) secondary structure prediction. These individual predictors achieve accuracy between 71 and 74% in terms of the Q3 measure (three-state per residue accuracy), as opposed to about 78% for the state of the art PsiPRED method, which is itself a consensus of several classifiers [19]. Figure 1 shows the distribution of margins of separation between the native and the highest scoring non-native state for the eLP/MaxF consensus classifier obtained by solving a set of inequalities defined by equations (5) and (6). The use of constraints defined in (6) helps to provide solutions with wide separation margins. Indeed, most of the correctly predicted points (i.e. those with positive margin) are away from the decision boundary with a median separation of about 7. Therefore, by combining the eLP/MaxF consensus with a weighted majority voting for points with a small margin between the two highest scoring classes, we were able to obtain highly accurate predictions (that became part of our SABLE system: http://sable.cchmc.org), as shown in Table 3.

Control sets :	CASP	S174	S189
PsiPRED	80.4%	79.4%	78.7%
eLP/MaxF	81.0%	77.5%	78.8%

Table 3. Accuracy of the secondary structure prediction system obtained using LP-based consensus.

4. Conclusions

A new approach to optimize linear combination based classifiers is introduced. The Maximum Feasibility heuristic for finding approximate solutions to infeasible LP problems is applied to eliminate points that are difficult to classify from the training and to obtain a separating hyperplane for a feasible subset of the data. This approach can be applied to large classification problems with millions of data points and hundreds of variables. In particular, it may advantageous for optimizing consensus classifiers that are postulated as a linear combination of well-trained individual classifiers, while preserving the margin of separation for best classifier in a given region of the feature space. Using this novel strategy we were able to obtain highly accurate consensus classifiers for secondary structure predictions.

In light of the above, the proposed method appears to provide a general and flexible approach to large-scale, multiclass supervised classification problem. Compared to linear perceptron approach, which also produces separating hyperplanes but does not converge for infeasible problems, the present algorithm will efficiently find an approximate solution. Other linear discriminant methods, such as linear regression or LDA focus on centroids of the classes. MaxF based classifiers, similarly to SVM, focus on points close to decision boundaries. Contrary to SVM, though, points that are difficult to classify are first removed from the training. It is worth noting, however, that our strategy is consistent with attempts to achieve a better accuracy by using SVM iteratively, with separating hyperplanes computed for subsets of data points that may result in more robust decision boundaries [15,17].

It is also worth noticing that the standard formulation of the SVM algorithm involves solving a Quadratic Programming (QP) problem [17], which is numerically more expensive than LP. Moreover, multiclass generalizations of SVM are cumbersome [1,17] and the present approach may be an efficient alternative as long as linear discrimination is sufficient. While we present only few examples in the present work, we would expect that linear separation is sufficient in most cases when considering a consensus of well-trained individual classifiers.

References

[1] T. Hastie, R. Tibshirani and J. Friedman; *The Elements of Statistical Learning*, Springer, New York 2001

[2] A. Krogh and J. Vedelsby; *Neural Network Ensembles, Cross Validation and Active Learning*, Advances in Neural Information Processing Systems, MIT Press, 7: 231-238 (1995)

[3] L. Breiman; Bagging predictors, Machine Learning 24: 123-140 (1996)

[4] Y. Freund and R. E. Schapire; Experiments with a new boosting algorithm, in L. Saitta, ed., Machine Learning: Proceedings of the Thirteenth National Conference, Morgan Kaufman, pp. 148-156 (1996)

[5] F. Bauer and R. Kohavi; *An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants*, Machine Learning 36: 105-139 (1999)

[6] B. Mulgrew and C. F. N. Cowan; *Adaptive Filters and Equalisers*, Kluwer Academic Publ., Boston 1988

[7]] D. T. Jones; *Protein secondary structure prediction based on position-specific scoring matrices*, J. Mol. Biol. 292, 195-202 (1999)

[8] S. Salzberg, A. Delcher, K. Fasman and J. Henderson; *A Decision Tree System for Finding Genes in DNA*, J. Comp. Biol. 5: 667-680 (1998)

[9] J. Meller, M. Wagner, R. Elber; *Maximum Feasibility Guideline to the Design and Analysis of Protein Folding Potentials*, Journal of Computational Chemistry, **23**: 111-118 (2002).

[10] M. Wagner, J. Meller, R. Elber; *Large-Scale Linear Programming Techniques for the Design of Protein Folding Potentials*, Mathematical Programming, to appear (2003)

[11] R. A. Friesner and J. R. Gunn; *Computational Studies of Protein Folding*, Annu. Rev. Bioph. Biom. 25: 315-342 (1996)

[12] R. D. C. Monteiro and I. Adler; *Interior path following primal-dual algorithms: Convex quadratic programming*, Math. Program. 44: 43-66 (1989)

[13] N. Chakravarti; *Some results concerning post-infeasibility analysis*, Eur. J. Oper. Res. 73: 139-143 (1994)

[14] GhostMiner; W. Duch, R. Adamczak and K. Grabczewski; *A new methodology of extraction, optimization and application of crisp and fuzzy logical rules*, IEEE Transactions on Neural Networks, Vol. 11 (2): 277-306 (2001)

[15] T. Joachims; *Making large-Scale SVM Learning Practical*, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press 1999

[16] T.W Rauber, M.M. Barata and A.S. Steiger-Garcao; *A Toolbox for Analysis and Visualization of Sensor Data in Supervision*, Proceedings of the International Conference on Fault Diagnosis, Toulouse, Toulouse, France, 1993

[17] N. Cristianini and J. Shawe-Taylor; *An Introduction to Support Vector Machines*, Cambridge University Press 2002