A General Method for Combining Predictors Tested on Protein Secondary Structure Prediction

Jakob V. Hansen
Department of Computer Science, University of Aarhus
Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark

Anders Krogh
Center for Biological Sequence Analysis, Technical University of Denmark
Building 208, DK-2800 Lyngby, Denmark

Abstract

Ensemble methods, which combine several classifiers, have been successfully applied to decrease generalization error of machine learning methods. For most ensemble methods the ensemble members are combined by weighted summation of the output, called the linear average predictor. The logarithmic opinion pool ensemble method uses a multiplicative combination of the ensemble members, which treats the outputs of the ensemble members as independent probabilities. The advantage of the logarithmic opinion pool is the connection to the Kullback-Leibler error function, which can be decomposed into two terms: An average of the error of the ensemble members, and the ambiguity. The ambiguity is independent of the target function, and can be estimated using unlabeled data. The advantage of the decomposition is that an unbiased estimate of the generalization error of the ensemble can be obtained, while training still is on the full training set. These properties can be used to improve classification. The logarithmic opinion pool ensemble method is tested on the prediction of protein secondary structure. The focus is on how much improvement the general ensemble method can give rather than on outperforming existing methods, because that typically involves several more steps of refinement.

1 Introduction

Empirically it has proven very effective to average over an *ensemble* of neural networks in order to improve classification performance, rather than to use a single neural network. See [1, 2] for examples in protein secondary structure, [3] for an over-view of applications in molecular biology. Here we present a general ensemble method, and show how the error of an ensemble can be written as the average error of the ensemble members minus a term measuring the disagreement between the members (called the ensemble ambiguity). This proves that the ensemble is always better than the average performance – substantiating the empirical observations.

The ambiguity is independent of the training targets and can thus be estimated from unlabeled data. Therefore an *unbiased* estimate of the ensemble error can be found when combining cross-validation and ensemble training. This error estimate can be used for determining when to stop the training of the ensemble. We test this general approach to ensemble training on the secondary structure prediction problem. This problem is well suited for the method, because thousands of proteins without known structure are available for estimation of ambiguity. It is shown that the estimated error works well for stopping training, that the optimal training time is longer than for a single network, and that the method outperforms other ensemble methods tested on the same data.

2 The ensemble decomposition

An ensemble consists of M predictors f_i, which are combined into the combined predictor F. We consider the case where each predictor outputs a probability vector $\{f_i^1, \ldots, f_i^N\}$, where f_i^j is the estimated probability that input $\vec{x}$ belongs to class c_j. The ensemble has associated M coefficient $\{\alpha_1, \ldots, \alpha_M\}$ obeying $\sum_i \alpha_i = 1, \alpha_i \geq 0$.

It is very common to define the ensemble predictor as $F^{\mathrm{LAP}} = \sum_{i=1}^{N} \alpha_i f_i$, which is called the linear average predictor (LAP). The combined predictor for the logarithmic opinion pool (LOP) of the ensemble members is

$$F^j = \frac{1}{Z} \exp\left(\sum_{i=1}^{M} \alpha_i \log f_i^j\right), \tag{1}$$

where Z is a normalization factor given by $\sum_{j=1}^{N} \exp(\sum_{i=1}^{M} \alpha_i \log f_i^j)$. This combination rule is non-linear and asymmetric as opposed to the LAP. Unless otherwise stated, this is the combined predictor we are considering in this work.

An example set T consist of input-output pairs, where the output is also a probability vector $\{t^1, \ldots, t^N\}$. The examples are assumed generated by a target function t. The difference between the target t and the combined predictor F is measured by the Kullback-Leibler (KL) error function

$$E(t, F) = \sum_{j=1}^{N} t^j \log\left(\frac{t^j}{F^j}\right). \tag{2}$$

The error is zero if F^j is equal to t^j for all j. For all appearances of this error function the mean over the training set is implicitly taken. If the target probabilities are restricted to one and zero the error function (2) reduces to $E(t, F) = -\log(F^k)$, where t^k is one. This would be the case if the error function is used on a training set consisting of class examples.

The error in (2) can be decomposed into two terms with the LOP in (1)

$$E(t, F) = \sum_{i=1}^{M} \alpha_i E(t, f_i) - \sum_{i=1}^{M} \alpha_i E(F, f_i) = \big\langle E(t, f_i) \big\rangle - A(f), \tag{3}$$

where $A(f)$ is the ambiguity and $\big\langle \cdot \big\rangle$ is the weighted ensemble mean. By using Jensen's inequality it can be shown that ambiguity is always greater than or equal to zero. This implies that the error of the combined predictor always is less than or equal to the mean of the error of the ensemble members. We see that diversity among the ensemble members without increase in the error of each ensemble member will lower the error of the combined predictor. The decomposition in (3) also applies for the LAP ensemble with the quadratic error $E(t,F) = (t-F)^2$ replacing (2) [4]. The LOP decomposition is due to Tom Heskes [5, 6], although it is derived in a slightly different setting.

We see that the ambiguity term in (3) is independent of the target probability, which means that the ambiguity can be estimated using unlabeled data, or if the input distribution is known the ambiguity can be estimated without any data. Assuming we have estimates of the generalization error of the ensemble members and an estimate of the ambiguity, the estimated generalization error comes directly from (3). This can be achieved in the *cross-validation ensemble*, where the training set is divided into M equal sized portions. Ensemble member f_i is trained on all the portions except for portion i. The error on portion i is independent of training and can be used to estimate ensemble member f_i's generalization error. With the estimated ambiguity, this gives us a method for obtaining an unbiased estimate of the ensemble error and still use all the training data.

3 Tests on the protein secondary structure problem

The method is tested on the standard secondary structure problem, in which the task is to predict the three-class secondary structure labels α-helix, β-sheet or coil, which is anything else. There is much work on secondary structure prediction, for overviews see [3]. The currently most used method is probably the one presented in [1].

The LOP ensemble is suitable for the protein problem for several reasons. Firstly the protein problem is a classification problem, and the LOP ensemble method is tailor-made for classification, and secondly there is huge amount of unlabeled data, i.e. proteins of unknown structure, which can be used to estimate the ambiguity.

The ensemble members are chosen to be neural networks. The output of a neural network is post-processed by the SOFTMAX function defined as $f^j = e^{g_j} / \sum_k e^{g_k}$, where g^j is the linear output (the weighted sum of the hidden units) of output unit j. This ensures that the ensemble members obey $\sum_j f_i^j = 1, f_i^j \geq 0$. The ensemble coefficients are uniform. Learning is done with back-propagation and momentum. A window of 13 amino acids is used, together with a sparse coding of amino acid into a vector of size 20, so each network has 260 inputs and three outputs. Each network has a hidden layer with 50 nodes. All examples in the training set are used once and only once during each epoch. Weights are updated after a certain number of randomly chosen examples have been presented (batch-update). During training the batch size is increased every tenth epoch by a number of examples equal to the square root of the training set size. The learning rate is decreased inverse proportional to the batch size. Training is stopped after 500 epochs, or if the validation error is increased by more than 10 % over the best value.

Our data set consists of 650 non-homologous protein sequences with a total of about 130000 amino acid [7]. Four-fold cross-validation is used for each test, which means that the training set is divided into four equally sized sets. Training is done on three sets with 97000 amino acids in total, while testing is done on the remaining set of 33000 amino acids. The test set is rotated for each cross-validation run. The average of the four test runs is calculated.

In the cross-validation ensemble the validation error is the estimate of the generalization error calculated from (3) by using a set of 70000 unlabeled amino acids to estimate the ambiguity. Seven ensemble members are used, which means that each one was trained on about 83000 amino acids and validated on 14000. Apart from the cross-validation ensemble, we also test what we will term a simple ensemble, in which all the ensemble members are trained on the same training set of 83000 amino acids, and the validation error is calculated from an independent set of labeled data containing 14000 amino acids. Note that there is still the four-fold cross-validation as an 'outer loop' for both ensemble methods.

For every tenth epoch the ensemble validation error, the ensemble test error, and the average test error of the individual ensemble members were calculated. The graphs in Figure 1 shows for a single test run these values for the cross-validation ensemble.

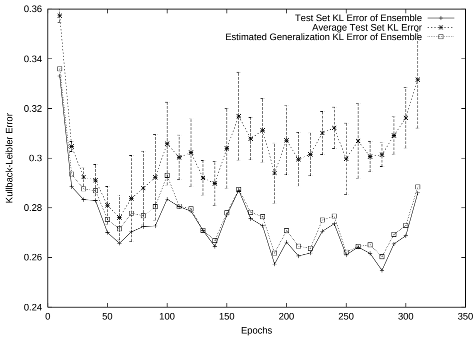

Figure 1: Cross-validation LOP ensemble

It is clearly seen that an ensemble is better than the average of the individual members, as it should be.
The cross-validation ensemble reaches a lower generalization error (represented by the test error) than the simple ensemble. This can be explained by the fact that the difference in training sets makes the ensemble members differ more than if they are trained on the same data, and this increases the ambiguity, which in turn lowers the generalization error.

In a practical application, one would select the ensemble at the training epoch with the lowest validation error. The various errors are shown for the ensemble selected in this manner. Using the validation error to select the ensemble training makes training dependent on the validation error, and therefore the estimate of the generalization error becomes biased. We will call the time of lowest validation error the stopping time. In

Table 1: The Kullback-Leibler errors and misclassification rates at stopping time.

Combination rule	LOP	LOP	LAP
Validation error	Cross-validation	Simple	Cross-validation
Train error function	KL	KL	MSE
Test error	0.2546	0.2585	0.6293
Average of indiv.	0.287 ± 0.010	0.2802 ± 0.0065	1.46 ± 0.12
Misclassification rate	0.3331	0.3385	0.3542
Ambiguity	0.0327	0.0217	0.8320

Table 1 the Kullback-Leibler error and misclassification rate at stopping time for the test runs is given. In these runs the validation error fluctuates by about ± 3 % around the test error. However, the oscillations of the validation error follows the oscillations of the test set error, as can be seen in Figure 1, so the validation error can still be used to find the lowest test error. The validation error is not always lowest when the test set error is lowest, so a measure of the usability of validation error for stopping is the average difference between the lowest test set error and the test set error at stopping time. For both types of ensembles this difference is as low as 0.0001 or close to 0.05 %. So the estimated generalization error can very accurately be used to find the right stopping time.

The cross-validation ensemble reaches a test error that is 1.5 % lower than for the simple ensemble, and a misclassification rate that is 1.6 % lower. The explanation is in a larger ambiguity for the cross-validation ensemble, since the average test error of the ensemble members are comparable. The ambiguity of cross-validation ensemble is 1.5 times the ambiguity for the simple ensemble.

As noted in section 2 the error of the combined predictor is always better than the average of the error of the ensemble members. Still, one of the ensemble members can be better than the combined predictor. For the cross-validation ensemble method the test error is 0.2546, while the average of the error of the ensemble members is 0.2873. The difference (the ambiguity) is 0.0327. A gain of 12.8 %, which is substantial. The standard deviation on the test error of the ensemble members is 0.010, so the ambiguity is more than three times larger. It is very unlikely that any ensemble member has a lower generalization error than the ensemble error. For the simple ensemble method the ambiguity is smaller: 0.0217 or a gain of 8.4 %. The standard deviation among the ensemble members is 0.0065, so the ambiguity is more than three times the deviation.

The lowest average error for the ensemble members do not have to happen when the ensemble error is lowest. Typically the lowest average generalization error of the ensemble members will be reached before the lowest generalization error for the ensemble, so the ensemble can actually gain from overfitting in the individual ensemble members. This effect can be seen in Figure 1. Also the optimal architecture for a simple predictor is often smaller than for ensemble members. A number of single predictors with different size hidden layer has been trained. The number of nodes in the hidden layer are varied from 3 to 400. The training must be done with a separate validation set, since there is no ambiguity for a single predictor. The best result is achieved with 10 hidden nodes giving an average generalization error of 0.2598, and misclassification rate of 0.3413, which is respectively 2.0 %, and 2.5 % more than for the cross-validation LOP ensemble.

A *standard* cross-validation ensemble using the MSE error function and LAP combination rule is trained on the same data as the cross-validation LOP ensemble. The validation error is calculated using (3) even though the outputs do not necessary sum to one. The validation error have lost it's meaning as an error, e.g. it can be negative, but it is still valid as an early stopping indicator. This is supported by the fact that the lowest misclassification rate on the test set is only 0.6 % lower than the misclassification rate on the test set at stopping time. The generalization error for the standard ensemble is much higher measured with the KL error function, namely 0.6293 or about 2.5 times more than the cross-validation LOP ensemble. This is not a fair comparison, since the LOP ensemble is trained to minimize the KL error. Another measure is the misclassification rate. For the standard ensemble the misclassification rate is 0.3542, which is 6.3 % more than the misclassification rate of the cross-validation LOP ensemble.

Surprisingly the benefit is not in the combination rule. A test run, where the LOP is replaced with the LAP, while training still uses the KL error, yields a generalization of 0.2543, which is essentially the same as for the LOP combination rule. The misclassification rate for the LAP is 0.3363, which 1.0 % more than the LOP.

4 Conclusion

It was shown how the generalization error of an ensemble of predictors using a logarithmic opinion pool (LOP) can be estimated using cross-validation on the training set and an estimate of the ambiguity from an independent unlabeled set of data. When testing on prediction of protein secondary structure it was shown that this estimate follows the oscillations of the error measured on an independent test set.

The estimated error can be used to stop training when it is at a minimum, and it was shown that the cross-validation LOP ensemble method is superior to single predictors and standard ensemble methods using mean square error function on the protein problem. The benefit is not as much in the combination rule, as in the use of the Kullback-Leibler error function and the target independent ambiguity term.

5 Acknowledgments

We would like to thank Claus Andersen and Ole Lund at CBS for sharing their protein data set. This work was supported by the Danish National Research Foundation.

References

[1] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70 % accuracy. *Journal of Molecular Biology*, 232(2):584–599, Jul 20 1993.

[2] S. K. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *Journal of Computational Biology*, 3:163–183, 1996.

[3] P. Baldi and S. Brunak. *Bioinformatics - The Machine Learning Approach*. MIT Press, Cambridge MA, 1998.

[4] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. The MIT Press, 1995.

[5] Tom Heskes. Bias/variance decompositions for likelihood-based estimators. *Neural Computation*, 10(6):1425–1433, 1998.

[6] Tom Heskes. Selecting weighting factors in logarithmic opinion pools. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[7] O. Lund, K. Frimand, J. Gorodkin, H. Bohr, J. Bohr, J. Hansen, and S. Brunak. Protein distance constraints predicted by neural networks and probability density functions. *Protein Engineering*, 10(11):1241–1248, 1997.