Combining Neural Network Regression Estimates Using Principal Components

Christopher J. Merz and Michael J. Pazzani
Dept. of Information and Computer Science
University of California, Irvine, CA 92717-3425 U.S.A.
{cmerz,pazzani}@ics.uci.edu

Category: Multiple Models.

1 Introduction

Combining a set of learned models[1] to improve classification and regression estimates has been an area of much research in machine learning and neural networks ([Wolpert92, Merz95, PerroneCooper92, LeblancTibshirani93, Breiman92, Meir95, Krogh95, Tresp95, ChanStolfo95]). The challenge of this problem is to decide which models to rely on for prediction and how much weight to give each.

The goal of combining learned models is to obtain a more accurate prediction than can be obtained from any single source alone. One major issue in combining a set of learned models is redundancy. *Redundancy* refers to the amount of agreement or linear dependence between models when making a set of predictions. The more the set agrees, the more redundancy is present. In statistical terms, this is referred to as the *multicollinearity problem.*

The focus of this paper is to describe and evaluate an approach for combining regression estimates based on principal components regression. The method, called PCR*, is then evaluated on several real-world domains to demonstrate its robustness versus a collection of existing techniques.

2 Motivation

The problem of combining a set of learned models is defined using the terminology of [PerroneCooper92]. Suppose two sets of data are given: a training set $\mathcal{D}_{Train} = (x_m, y_m)$ and a test set $\mathcal{D}_{Test} = (x_l, y_l)$. Now suppose $\mathcal{D}_{Train}$ is used to build a set of functions, $\mathcal{F} = f_i(x)$, each element of which approximates $f(x)$. The goal is to find the best approximation of $f(x)$ using $\mathcal{F}$.

[1] A learned model may be anything from a decision/regression tree to a neural network.

To date, most approaches to this problem limit the space of approximations of $f(x)$ to linear combinations of the elements of $\mathcal{F}$, i.e.,

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i f_i(x)$$

where α_i is the coefficient or weight of $f_i(x)$.

A variety of approaches to this problem have been taken. The simplest is to weight the models uniformly, i.e., $\alpha_i = 1/N$. [PerroneCooper92] call this approach the Basic Ensemble Method (BEM) and show that it is most effective when the errors made are small and independent. Since the errors tend to not be independent, they created a more versatile technique call the Generalized Ensemble Method (GEM) which is more capable of handling correlated errors. GEM is, in essence, a restricted form a linear regression where $\sum_{i=1}^{N} \alpha_i = 1$. Unconstrained linear least squares regression (LR) is another possible approach to this problem.

It will be demonstrated empirically later that BEM tends to be too restrictive by assuming complete independence in the errors of $\hat{f}_i(x)$. At the other end of the spectrum are GEM and LR which restrict the weights only slightly if at all and tend to be overly sensitive to the correlation in the errors of $\hat{f}_i(x)$.

The sensitivity of LR and GEM to correlation in $\mathcal{F}$, known as the multicollinearity problem, leads to unstable estimation of the weights, α. Consequently, the weights obtained from fitting the model to a particular sample may be far from their ideal values. To circumvent this problem, approaches have been developed which: 1) constrain the estimated regression coefficients so as to improve prediction performance (i.e., ridge regression, RIDGE [MontgomeryFriedman93]), 2) search for the coefficients via gradient descent procedures (i.e., Widrow-Hoff learning, GD and EG+- [KivinenWarmuth94]), or 3) build models which make decorrelated errors by adjusting the bias of the learning algorithm ([Opitz95]) or the data which it sees ([Meir95]). The third approach ameliorates, but does not solve, the problem because redundancy is an inherent part of the task of combining estimators.

The focus of this paper is on finding constrained regression coefficients. A new approach, based on principal components regression, is introduced which addresses the multicollinearity problem by automatically determining the appropriate level of constraint for the weights of the original learned models. Leblanc and Tibshirani [LeblancTibshirani93] have proposed several ways of constraining or *regularizing* the weights to help produce estimators with lower prediction error:

1. Shrink $\hat{\alpha}$ towards $(1/K, 1/K, \ldots, 1/K)^T$ where K is the number of learned models.
2. $\sum_{i=1}^{N} \alpha_i = 1$
3. $\alpha_i \geq 0, i = 1, 2 \ldots K$

Breiman ([Breiman92]) provides an intuitive justification for these constraints by pointing out that the more strongly they are satisfied, the more interpolative the weighting scheme is. In the extreme case, a uniformly weighted set of learned models is likely to produce a prediction *between* the maximum and minimum predicted values of the learned models. Without these constraints, there is no guarantee that the resulting predictor will stay near that range and generalization may be poor. The next section describes the proposed variant of principal components regression and explains how it provides a continuum of regularized weights for the original learned models.

3 Principal Components Regression

When dealing with the above mentioned multicollinearity problem, principal components regression ([DraperSmith81]) may be used to summarize and extract the "relevant" information from the learned models. The main idea of PCR is to map the original learned models to a set of (independent) principal components in which each component is a linear combination of the original learned models, and then to build a regression equation using the best number of the principal components to predict $f(x)$.

The advantage of this representation is that the components are sorted according to how much information (or variance) from the original learned models for which they account. Given this representation, the goal is to choose the number of principal components to include in the final regression by retaining the first k which meet a preselected stopping criteria. The basic approach is summarized as follows:

1. Do a principal components analysis (PCA) on the learned models' performances on the training data (i.e., do a PCA on the matrix, M, where $M_{i,j}$ is the j-th model's response for the i-th training example) to produce a set of principal components, $PC = \{PC_1, ..., PC_N\}$.
2. Use a stopping criteria to decide on k, the number of principal components to use.
3. Do a least squares regression on the selected components (i.e., include PC_i for $i \leq k$).
4. Derive the weights, α_i, for the original learned models by expanding
$$f_{PCR*} = \beta_1 PC_1 + ... + \beta_k PC_k$$
according to
$$PC_i = \gamma_{i,0} f_0 + ... + \gamma_{i,N} f_N ,$$
and simplifying for the coefficients of f_i. Note that $\gamma_{i,j}$ is the j-th coefficient of the i-th principal component.

The second step is very important because choosing too few or too many principal components may result in underfitting or overfitting, respectively. Ten-fold cross-validation is used to select k. For a given value of k, as each partition is held out it is evaluated on the regression equation derived from the pseudo-principal components. The pseudo-principal components are linear combinations of the models' predictions on the examples from the other nine partitions where the coefficients of this combination are the eigenvectors derived in step 1 for the *entire* data set. The k with the smallest cross-validation error is chosen.

Examining the spectrum of (N) weight sets derived in step four reveals that PCR* provides a continuum of weight sets spanning from highly constrained (i.e., weights generated from PCR_1 satisfy all three regularization constraints) to completely unconstrained (i.e., PCR_N is equivalent to unconstrained linear regression). To see that the weights, α, derived from PCR_1 are (nearly) uniform, recall that the first principal component accounts for where the learned models agree. Because the learned models are all fairly accurate they agree quite often so their first principal component weights, $\gamma_{1,*}$ will be similar. The γ-weights are in turn multiplied by a constant when PCR_1 is regressed upon. Thus, the resulting α_i's will be fairly uniform. The later principal components serve as refinements to those already included producing less constrained weight sets until finally PCR_N is included resulting in an unconstrained estimator much like LR, LRC and GEM.

Table 1: Results

Data	bodyfat		cpu		housing	
N	10	50	10	50	10	50
BEM	1.03(0.16)	1.04(0.16)	**38.57(7.88)**	**38.62(7.90)**	2.79(0.19)	2.77(0.18)
GEM	**1.02(0.17)**	**0.86(0.26)**	46.59(14.9)	227.54(197.9)	**2.72(0.20)**	**2.57(0.28)**
LR	**1.02(0.16)**	3.09(6.62)	44.9(13.8)	238.0(189.3)	**2.72(0.20)**	6.44(5.59)
RIDGE	**1.02(0.16)**	**0.826(0.27)**	44.8(13.7)	191.0(133.4)	**2.72(0.21)**	**2.55(0.26)**
GD	**1.03(0.16)**	1.04(0.16)	38.9(7.95)	38.8(7.99)	2.79(0.20)	2.77(0.18)
EGPM	1.03(0.17)	1.07(0.16)	**38.4(8.08)**	**38.0(7.86)**	2.77(0.20)	2.75(0.17)
PCR_1	1.04(0.15)	1.05(0.15)	39.0(7.76)	39.0(7.80)	2.78(0.21)	2.76(0.19)
PCR_N	**1.02(0.17)**	**0.848(0.27)**	44.8(13.7)	249.9(166.9)	**2.72(0.21)**	**2.57(0.29)**
PCR*	**0.99(0.16)**	**0.786(0.21)**	**40.3(10.0)**	**40.8(10.1)**	**2.70(0.21)**	**2.56(0.26)**

4 Experimental Results

The methods described were evaluated on three data sets: *cpu* and *housing* (from the UCI repository), and *bodyfat* (from the Statistics Library at Carnegie Mellon University). $\mathcal{F}$ consisted of neural networks with randomly initialized weights trained using Backpropogation ([Rumelhart86])[2]. Twenty trials were run for each of the data sets. Two sizes of $\mathcal{F}$ were tried (i.e., 10 and 50, respectively). As more models are included the linear dependence amongst them goes up showing how well the multicollinearity problem is handled[3]. Table 1 shows the average residual errors for the each of the methods on the three data sets. Bold-faced entries indicate methods which were *not* significantly different from the method with the lowest error (via two-tailed paired t-tests with $p \leq 0.05$).

PCR* is the only approach which is among the leaders for all three data sets. For the *bodyfat* and *housing* data sets the weights produced by BEM, PCR_1, GD, and EG+- tended to be too constrained, while the weights for LR tended to be too unconstrained for the larger collection of models. The less constrained weights of GEM, LR, RIDGE, and PCR_N severely harmed performance in the *cpu* domain where uniform weighting performed better.

The biggest demonstration of PCR*'s robustness is its ability to gravitate towards the more constrained weights produced by the earlier principal components when appropriate (i.e., in the *cpu* data set). Similarly, it uses the less constrained principal components closer to PCR_n when it is preferable as in the *bodyfat* and *housing* domains.

5 Conclusion

This investigation suggests that the principal components of a set of learned models can be useful when combining the models to form an improved estimator. It was demonstrated that the principal components provide a continuum of weight sets from highly regularized to unconstrained. An algorithm, PCR*, was developed which attempts to automatically select the number of these components which provides the lowest prediction error. Experiments on a collection of domains demonstrated PCR*'s ability to robustly handle redundancy in the set of learned models.

[2]There was no extreme effort to produce networks with more decorrelated errors. Even with such networks, the issue of extreme multicollinearity would still exist because $E[f_i(x)] = E[f_j(x)]$ for all i and j.

[3]This is verified by observing the eigenvalues of the principal components and values in the covariance matrix of the models in $\mathcal{F}$

Future work will be to improve upon PCR* and expand it to the classification task.

References

[Breiman84] Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

[Breiman92] Breiman, L. (1992). Stacked Regression. Dept of Statistics, Berkeley, TR No. 367.

[ChanStolfo95] Chan, P.K., Stolfo, S.J. (1995). A Comparative Evaluation of Voting and Meta-Learning on Partitioned Data *Proceedings of the Twelvth International Machine Learning Conference* (90-98). San Mateo, CA: Morgan Kaufmann.

[DraperSmith81] Draper, N.R., Smith, H. (1981). *Applied Regression Analysis.* New York, NY: John Wiley and Sons.

[KivinenWarmuth94] Kivinen, J., and Warmuth, M. (1994). Exponentiated Gradient Descent Versus Gradient Descent for Linear Predictors. Dept. of Computer Science, UC-Santa Cruz, TR No. ucsc-crl-94-16.

[Krogh95] Krogh, A., and Vedelsby, J. (1995). Neural Network Ensembles, Cross Validation, and Active Learning. In *Advances in Neural Information Processing Systems 7.* San Mateo, CA: Morgan Kaufmann.

[HansenSalamon] Hansen, L.K., and Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12* (993-1001).

[LeblancTibshirani93] Leblanc, M., Tibshirani, R. (1993) Combining estimates in regression and classification Dept. of Statistics, University of Toronto, TR.

[Meir95] Meir, R. (1995). Bias, variance and the combination of estimators. In *Advances in Neural Information Processing Systems 7.* San Mateo, CA: Morgan Kaufmann.

[Merz95] Merz, C.J. (1995) Dynamical Selection of Learning Algorithms. In Fisher, C. and Lenz, H. (Eds.) Learning from Data: Artificial Intelligence and Statistics, 5). Springer Verlag

[MontgomeryFriedman93] Mongomery, D.C., and Friedman, D.J. (1993). Prediction Using Regression Models with Multicollinear Predictor Variables. IIE Transactions, vol. 25, no. 3 73–85.

[Opitz95] Opitz, D.W., Shavlik, J.W. (1996). Generating Accurate and Diverse Members of a Neural-Network Ensemble. Advances in Neural and Information Processing Systems 8. Touretzky, D.S., Mozer, M.C., and Hasselmo, M.E., eds. Cambridge MA: MIT Press.

[PerroneCooper92] Perrone, M. P., Cooper, L. N., (1993). When Networks Disagree: Ensemble Methods for Hybrid Neural Networks. *Neural Networks for Speech and Image Processing*, edited by Mammone, R. J.. New York: Chapman and Hall.

[Rumelhart86] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Interior Representation by Error Propagation. *Parallel Distributed Processing, 1* 318–362. Cambridge, MASS.: MIT Press.

[Tresp95] Tresp, V., Taniguchi, M. (1995). Combining Estimators Using Non-Constant Weighting Functions. In *Advances in Neural Information Processing Systems 7.* San Mateo, CA: Morgan Kaufmann.

[Wolpert92] Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks, 5*, 241–259.