Statistica Sinica **16**(2006), 523-540

USING INPUT DEPENDENT WEIGHTS FOR MODEL COMBINATION AND MODEL SELECTION WITH MULTIPLE SOURCES OF DATA

We Pan, Guanghua Xiao and Xiaohong Huang

University of Minnesota

Abstract: With various sources and large amounts of genomic and proteomic data accumulating, the importance of integrative analyses of multiple sources of data has been increasingly recognized. A natural approach is to combine multiple models, each built on one source of data. A challenge however is to account for different local information contents of different sources of data: the choice of the weight on each candidate model (and thus each source of data) may depend on the input for which a prediction is to be made, suggesting that the constant weights used in most existing approaches may not be optimal. Here we propose an input-dependent weighting (IDW) scheme with the weight being the probability of each model's giving a correct prediction for the given input. The weights can be estimated based on regression using training data. We apply IDW to discriminating human heart failure etiology using two sources of gene expression data, and to gene function prediction by a combined analysis of gene expression and protein-protein interaction data. It is demonstrated that IDW may perform better than some standard approaches. Input-dependent weights can be also adopted as a criterion for model selection.

Key words and phrases: Classification, microarray data, model mixing, partial least squares, prediction.

1. Introduction

We consider the problem of prediction in the presence of multiple models. There are two approaches: model selection and model combination/mixing (Yang (2003)). In model selection, a model selection criterion measuring the predictive performance of a model or the probability of its being the true model is used to compare the models. Based on an estimate of the criterion using, e.g., AIC (Akaike (1973)), BIC (Schwarz (1978)), cross-validation (CV) (Stone (1974), Geisser (1975) and Efron (2004)), the bootstrap (Efron (1983), Efron (1986) and Efron and Tibshirani (1997)), or some more recently developed adaptive model selection criteria (George and Foster (2000), Shen and Ye (2002) and Shen, Huang and Ye (2004)), a "best" model is selected and then used in the

subsequent analysis. A drawback of model selection is its instability (Breiman (1996a)). In addition, one can argue that there is no single correct or best model (Box (1980)). As an alternative, one may choose to combine multiple models, usually by taking an equally or unequally weighted average of the outputs from the candidate models; a key is how to choose the weights (see Ripley (1996) and Hastie, Tibshirani and Friedman (2001) for reviews). Existing approaches include Bayesian model averaging (see Hoeting et al. (1999) for a review), stacking (Wolpert (1992) and Breiman (1996b)), bagging (Breiman (1996c) and LeBlanc and Tibshirani (1996)), random forests (Breiman (2001)), boosting (Freund and Schapire (1997)), ARM (Yang (2001)) and using AIC/BIC as weights (Burnham and Anderson (2002) and Hastie et al. (2001)). There have been discussions on the choice between model selection and model mixing (e.g., Yang (2003), Yuan and Yang (2003), Paik and Yang (2004) and Shen and Huang (2005)). The general conclusion is that the choice depends on the application.

In a prediction problem, based on a given training dataset one first builds a model, which can be either a selected model or a combination of multiple models. Then for each case of a given test dataset, one predicts the response value using the constructed model. In almost any of the existing approaches, a common feature is that neither the model selection criterion nor the weight in model mixing depends on the test data. For example, AIC/BIC for a candidate model only depends on the training data, not test data; the data-dependent penalty term in adaptive model selection criteria (George and Foster (2000) and Shen and Ye (2002)) also only depends on the training data; in BMA, the weight of each candidate model is the posterior probability of the model's being correct or selected given the training data, which again does not depend on test data. This can be a problem in some applications. In our motivating example, we have datasets collected from two independent institutions that are to be used to address the same scientific question. Due to the heterogeneity of patient populations and different study protocols, the datasets are quite different. It is possible that the datasets, and hence the models based on them, cover different subspaces of the input space: one works better in some subspace of the input while the other works better in another subspace. For a given new input, depending on which subspace it falls in or is closer to, we may choose to select or weight more on the corresponding model. Therefore, we propose using *test* input-dependent weights, shortened as input-dependent weights (IDWs) in the sequel, to select from or combine multiple models. An IDW for a candidate model is defined as the probability of the model's giving a correct prediction for a given input or, more generally, any reasonable measure to quantify the predictive performance of the model for the input. The weights can generally be estimated using regression techniques.

2. Methods

2.1. Input-dependent weights

Suppose we have M candidate models $f_1, \ldots, f_M$ that can be built based on a given training dataset (X_i, Y_i), with $X_i = (x_{1i}, \ldots, x_{pi})^T$ for $i = 1, \ldots, n$. Our goal is to predict the response Y^* for a new input X^*.

For concreteness, we first consider the classification problem where the response is $Y_i = 0$ or 1, and the output of each fitted model $\hat{f}_m(X_i)$ is either 0 or 1. In most existing approaches to model mixing, the output from combining models is

$$\hat{f}_G(X^*) = 1\left\{ \frac{\sum_{m=1}^{M} w_m \hat{f}_m(X^*)}{\sum_{m=1}^{M} w_m > c} \right\},$$

where c is a cut-off value used to dichotomize the output, and the w_m's are constant weights, independent of the input X^*. In general, w_m can be some (estimated) predictive performance measure, averaged over the input space for model f_m, such as AIC/BIC.

In contrast to constant weights, we propose using input-dependent weights (IDWs) $w_m(X^*)$, leading to the output

$$\hat{f}_{IDW}(X^*) = 1\left\{ \frac{\sum_{m=1}^{M} w_m(X^*) \hat{f}_m(X^*)}{\sum_{m=1}^{M} w_m(X^*) > c} \right\}, \tag{2.1}$$

where the weight

$$w_m(X^*) = \hat{\pi}(X^*; f_m)$$

is an estimate of $\pi(X^*; f_m)$, the probability of model f_m's giving a correct prediction for X^*, which can be estimated using the training data.

In our example, because we have a much larger number of predictor variables than the sample size (i.e., $p >> n$), we propose using a linear model to estimate $w_m(X^*)$. Specifically, for each model f_m, we create a binary response variable Z_{im} for $i = 1, \ldots, n$, and then fit a linear model

$$E(Z_{im}) = \pi(X; f_m) = \gamma_{0m} + \gamma_m^T X_i \;\text{ with }\; Z_{im} = 1\{Y_i = \hat{f}_m(X_i)\} \tag{2.2}$$

using partial least squares (PLS) (Wold et al. (1984)) with training data (X_i, Z_{im}) for $i = 1, \ldots, n$, obtaining estimates $\hat{\gamma}_{0m}$ and $\hat{\gamma}_m$ for the unknown regression

parameters, where γ_{0m} is a scalar and γ_m is a $p \times 1$ vector. The IDW for model f_m at an input X^* is

$$w_m(X^*) = \hat{\gamma}_{0m} + \hat{\gamma}_m^T X^*,$$

and constant weights can be regarded as the special case $\hat{\gamma}_m = 0$.

Since $w_m(X^*)$ is an estimate of a probability and there is no guarantee that the output of the linear model will be between 0 and 1, we may want to truncate $w_m(X^*)$ at 0 or 1 if it is smaller than 0 or larger than 1. Alternatively, one may want to use logistic regression (or its penalized form to account for $p >> n$) or other more flexible models to obtain better probability estimates. In addition, rather than using training outputs as proposed here to estimate γ_{0m} and γ_m, one may want to use some form of predictive outputs to construct Z_{im}. For example, using leave-one-out cross-validation (LOOCV), we build $\hat{f}_m^{(-i)}$ using the training data after deleting the ith observation (X_i, Y_i), and define $Z_{im} = 1\{Y_i = \hat{f}_m^{(-i)}(X_i)\}$. Then we can proceed as before. Although LOOCV is computationally more demanding, it may be necessary to use LOOCV when the candidate models over-fit training data to different degrees. In our latter example, the candidate models are built on different sources of data using the same type of classifier, and adopting (2.2) will not become an issue.

The IDW method can be applied equally to classification or regression where the output of each model is numerical. For example, in classification, the output of a classifier can be a probability. Then we can adopt

$$\hat{f}_{IDW}(X^*) = \frac{\sum_{m=1}^{M} w_m(X^*)\hat{f}_m(X^*)}{\sum_{m=1}^{M} w_m(X^*)}, \tag{2.3}$$

which is the final output for regression, and may need to be dichotomized by comparing to a cut-off value, e.g., $c = 1/2$, in classification.

Accordingly, we can define $Z_{im} = L(Y_i, \hat{f}_m(X_i))$ or $Z_{im} = L(Y_i, \hat{f}_m^{(-i)}(X_i))$, where $L(y, \hat{y})$ is a minus loss function for an observed response y and its predicted value $\hat{y}$. For example, in regression, it can be negative squared error: $L(y, \hat{y}) = -(y - \hat{y})^2$, or other robust (minus) loss functions. More generally, we may adopt any criterion $L((X_i, Y_i); f_m)$ that is (nearly) to be maximized under each candidate model f_m. For example, if there is an exact or approximate likelihood for each model f_m, it is natural to define $Z_{im} = L((X_i, Y_i); \hat{f}_m)$ or $Z_{im} = L((X_i, Y_i); \hat{f}_m^{(-i)})$, the (predicted) log-likelihood value at (X_i, Y_i) under the fitted model $\hat{f}_m$ or $\hat{f}_m^{(-i)}$. The argument for the choice between $\hat{f}_m$ or $\hat{f}_m^{(-i)}$ is the same as before.

The weights can be also used as a criterion for model selection: we select the model f_{m_0} with the largest weight $w_{m_0}(X^*) = \max_{1\le m\le M} w_m(X^*)$ and then use its output $\hat{f}_{m_0}(X^*)$ as the final prediction for X^*. Note that if we use the weights as a model selection criterion, using the constant weights will always select the same model for all the test data, whereas using IDWs may choose different models for different X_i^*.

Because we are going to use PLS, NSC and RF as the models in our main example, we review them briefly.

2.2. Partial least squares (PLS)

We use Y and $X = (x_1, \ldots, x_p)^T$ as generic notation to represent a binary response variable and a vector of p input variables. We code the response as $Y = 1$ for class 1 and $Y = 0$ for class 2. Given a training dataset $\{(X_i, Y_i) : i = 1, \ldots, n\}$, n iid copies of (X, Y), the goal is to fit a linear model of Y on X. When $n < p$, the commonly used ordinary least squares (OLS) may not work well, and partial least squares (PLS) was proposed as an alternative (Wold et al (1984)). Rather than regressing Y on x_j's, PLS first identifies a sequence of linear combinations of $(x_1, \ldots, x_p)$, $z_j = \alpha_j^T X$ for $j = 1, \ldots, q$, then regresses Y on $Z = (z_1, \ldots, z_q)$. Usually we have $q << p$ and $q < n$, and hence OLS can be used to obtain

$$\widehat{Y}_{PLS} = \beta_0 + \sum_{j=1}^{q} \beta_j z_j$$

with β's as OLS estimates. The choice of α_j is key, and it turns out that

$$\alpha_j = \text{argmax}_\alpha \text{Cov}(\mathbf{y}, \mathbf{X}\alpha)$$

with the constraints $||\alpha|| = 1$, $\alpha_l^T S\alpha = 0$ for $l = 1, \ldots, j-1$, where $\mathbf{y} = (Y_1, \ldots, Y_n)^T$ is the vector of observed Y_i's (in the training data), $\mathbf{X} = (X_1, \ldots, X_n)^T$ is the design matrix (i.e., matrix of observed X_i's), and S is the sample covariance matrix of X_i's (Frank and Friedman (1993)).

Simple algorithms exist to fit a PLS model (e.g., Hastie et al. (2001)). In practice, the number of linear components q has to be chosen, typically by CV.

Note that the class label (1/0) for the response Y is binary, but it is treated as numerical and the estimate $\widehat{Y}$ could be any real number. To predict the class of a new sample, if the estimated response $\widehat{Y}$ is greater than a threshold, e.g., 1/2, then we classify it into class 1; otherwise, class 2. There have been increasing applications of PLS to prediction with gene expression profiles (e.g., Nguyen and Rocke (2002), Hawkins et al. (2003), Huang and Pan (2003), Boulesteix (2004), Huang et al. (2004), Li and Gui (2004) and Tan et al. (2004)).

2.3. Nearest shrunken centroids (NSC)

Nearest shrunken centroids (NSC) was proposed for sample classifications with gene expression data when $p >> n$ (Tibshirani et al. (2002)). It combines the idea of a diagonalized linear discriminant analysis (DLDA) (e.g., McLachlan (1992) and Hastie et al. (2001)) with that of parameter shrinkage. For a K-class problem, suppose that $\bar{x}_{jk}$ is the sample mean of predictor j in class k of the training data, s_j^2 is the pooled sample variance of predictor j of the training data, and π_k is the prior probability of class k. The DLDA rule for a new sample $X^* = (x_1^*, \ldots, x_p^*)^T$ is

$$\delta_k(X^*) = \sum_{j=1}^{p} \frac{(x_j^* - \bar{x}_{jk})^2}{s_j^2} - 2\log \pi_k.$$

Define

$$d_{jk} = \frac{\bar{x}_{jk} - \bar{x}_j}{m_k s_j} \quad \text{with } m_k = \sqrt{\frac{1}{n_k} - \frac{1}{n}},$$

where n_k is the number of training samples in class k, and $\bar{x}_j$ is the overall sample mean of predictor j of the training data. By definition we have $\bar{x}_{jk} = \bar{x}_j + m_k s_j d_{jk}$. Let $d'_{jk} = \text{sign}(d_{jk})(|d_{jk}| - \Delta)_+$ for all j and k, where $a_+ = \max(a, 0)$ for any number a, and Δ is the shrinkage parameter to be chosen by CV. Substituting $\bar{x}_{jk}$ in the DLDA rule by $\bar{x}'_{jk} = \bar{x}_j - m_k s_j d'_{jk}$, we obtain an NSC rule

$$\delta'_k(X^*) = \sum_{j=1}^{p} \frac{(x_j^* - \bar{x}'_{jk})^2}{s_j^2} - 2\log \pi_k.$$

The new sample X^* is assigned to class $k_0 = \text{argmin}_k \delta'_k(x^*)$.

Note that if $d'_{jk} = 0$, then $\bar{x}'_{jk} = \bar{x}_j$, and predictor j is unused in the classification. This is in contrast to PLS, where each predictor is used in the final model.

2.4. Random forest (RF)

A random forest (RF) (Breiman (2001)) is an ensemble of classification trees (Breiman et al. (1984) and Zhang and Singer (1999)). It is designed to improve over a single classification tree. There are two random aspects that help generate multiple classification trees in RF. First, a bootstrap sample is repeatedly drawn from the original training data and used to build a classification tree. Second, in building a classification tree, rather than using the best splitting variable (i.e., gene here) from all the available variables at each node, it chooses the best from a small random subset of all the variables. Each tree is grown to maximum size and no pruning is pursued. To predict the class for a new sample, the sample

is applied to each tree and each tree votes by giving its prediction, then the majority vote is taken as the final prediction for the sample.

3. Main Example

3.1. Data

Our example concerns the use of gene expression profiles to discriminate heart failure etiology, which is important to guide appropriate therapy and determine prognosis for successful treatment. We have two datasets, called Minnesota (MN) data and PGA data, containing gene expression profiles of heart failure patients collected at the University of Minnesota and the Harvard University, respectively (Hall et al. (2004) and PGA (2004)). The two studies used Affymetrix HG-U133A chips containing ∼22,000 probe sets and HG-U133 plus 2 chips containing 54,675 probe sets respectively. The gene expression levels were summarized using Affymetrix Microarray Suite (MAS 5.0).

The patients were divided into two classes according to the underlying heart failure etiology, ischemic versus idiopathic. In the MN data, there were 10 patients in ischemic class and 13 in idiopathic class; for the PGA data, there were 11 and 14 in the two classes.

In order to combine the datasets, we matched the probe sets on a HG-U133 plus 2 chip with those on a HG-U133A chip. Only six probe sets on the HG-U133A chip could not be found on a HG-U133 plus 2 chip. Hence we used the remaining 22,277 probe sets in the following analyses with both datasets. To make the gene chips comparable to each other, we standardized the expression levels on each array by centering them at median 0 and scaling them by their interquartile range.

In previous work (Huang et al. (2005)), it was found that with any of five types of classifiers, including partial least squares (PLS), nearest shrunken centroids (NSC), and random forest (RF), it was much more difficult to predict the heart failure etiology for the MN data than for the PGA data, probably due to the heterogeneity of the study populations and the different gene chip platforms used. Using the first two components of a PLS model fitted to the combined data, Figure 3.1 shows the MN data points well separated from the PGA data, suggesting some inherent difference between the two datasets. As we show later, a model built using either the PGA data or the combined data may still not work well for the MN data. It seems reasonable that, for the purpose of prediction for a given test sample, depending on which study populations the test sample is closer to, we should weigh more on the model built using the corresponding data; this is exactly the idea of IDW. We use the data to demonstrate the good performance of our proposed IDW.

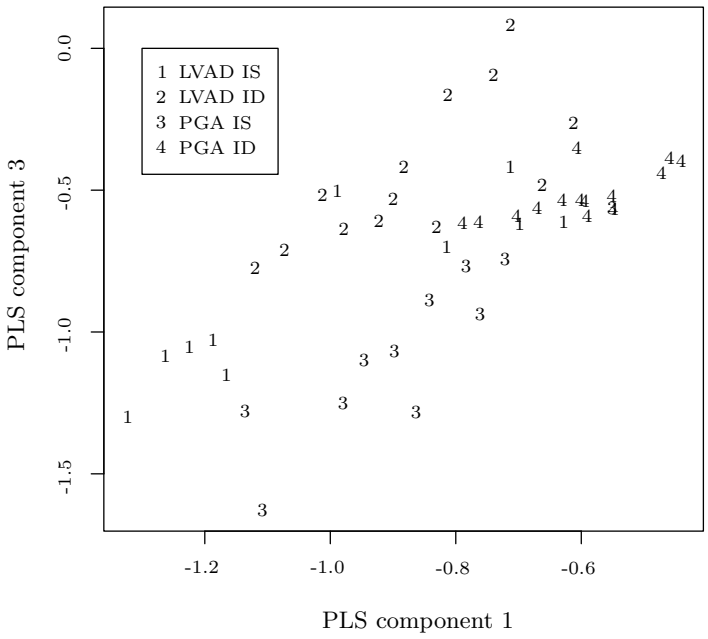

Figure 3.1. PLS plot for the MN data and PGA data. IS and ID refer to ischemic class and idiopathic class respectively.

3.2. Evaluation methods

For the purpose of comparison, in addition to the IDW method, we considered four more methods. The first three used the MN data alone, the PGA data alone, or the combined MN and PGA data to build a classifier, denoted as classifiers MN, PGA or MN+PGA, respectively. Two methods were used to combine the two base classifiers (i.e., MN and PGA), a simple average method and our proposed IDW. The simple average method weighted each of the two classifiers equally, implying that the combined model went with the decision that took the two classifiers with equal probability. Hence, if the two base classifiers give different predictions, the (expected) error of the combined model is 1/2. In the IDW method, the weights $w_m(X^*)$, $m = 1$ or 2, were calculated for any test input X^*, and the output of the combined model was the same as that of the classifier whose weight was larger. Note that if we use constant weights, because we only have two candidate classifiers to combine, it means that the decision is the model with the better overall performance.

We conducted three experiments. In the first, two candidate classifiers f_1 and f_2 were the nearest shrunken centroids (NSC) models built using the MN

and PGA data, respectively. In the second, we used PLS to build two classifiers f_1 and f_2 using the MN and PGA data, respectively, while in the third, RFs were used as the base classifiers f_1 and f_2. In each experiment, as in (2.2), PLS was used to fit a linear model, with all the predictors included, to estimate the probability of having a correct prediction under each fitted model f_1 or f_2 using the training data; if the output $\tilde{f}_m(X_i)$ of a base classifiers f_m is not binary, such as when PLS is used, we dichotomize the output as $\hat{f}_m(X_i) = 1(\tilde{f}_m(X_i) > 1/2)$.

There were two ways to combine the candidate models when their output was numerical, as described earlier. For example, suppose $\tilde{f}_m$ was the output of PLS model m, $m = 1$ or 2. We truncated $\tilde{f}_m$ at 0 or 1 if $\tilde{f}_m < 0$ or $\tilde{f}_m > 1$. Then, applying $\hat{f}_m = 1(\tilde{f}_m > c)$ or $\hat{f}_m = \tilde{f}_m$ to (2.1) or (2.3), respectively, resulted in one of the two implementations of the IDW method, called IDW-discrete and IDW-continuous, respectively. Similarly, using weights $w_1 = w_2 = 1/2$ in (2.1) and (2.3), we obtained two implementations of the simple average method, simple-discrete and simple-continuous. On the other hand, if the output of candidate models was binary, as in NSC, we always went with (2.1). All the results reported in Table 3.1 were based on implementing (2.1). Note that, because we only had two candidate models, IDW-discrete was equivalent to selecting the model with the larger weight.

For each experiment, we explored three ways of using the data. In Case (i), we used all the 22,277 genes as predictor variables. In Case (ii), we used only the 100 genes whose t-statistics had the largest absolute values to compare the two classes using the combined data. Finally, in Case (iii), we used the 100 genes whose t-statistics had the largest absolute values to compare the two classes using only the MN data.

Gene selection here was mainly for the purpose of perturbing the data. For example, gene selection in Case (iii) was designed to favor classifier MN so that it would work much better than classifier PGA for MN data, whereas classifier PGA would be better than classifier MN for PGA data, an interesting situation for the purpose of model selection or model mixing. In particular, we would like to investigate whether in this case our proposed IDW method could adaptively choose the better classifier depending on the input.

It is well-known that selecting genes using all the samples leads to a biased estimate of prediction error rate in the subsequent CV (Ambroise and McLachlan (2002) and Simon et al. (2003)). However, our goal was not to estimate the prediction error rate per se, but to compare the methods. Gene selection in Case (ii) did not favor any method and hence it was still fair to compare the methods; gene selection in Case (iii) favored classifier MN, which however, as to be shown later, was defeated by the IDW method.

Due to the small sample size, we used leave-one-out cross-validation (LOOCV) to estimate the number of prediction errors for each method. After submitting this article, we became aware of a new cross-validation method that works better than LOOCV for small samples and may be more appropriate here (Fu, Carroll and Wang (2005)).

3.3. Implementation

PLS, NSC and RF are implemented, respectively, in packages `pls.pcr`, `pamr` and `randomForest` in R (Ihaka and Gentleman (1996)), and are easy to use. With a given training dataset, 5-fold CV was used to select the number of components in PLS and the shrinkage parameter in NSC; otherwise, default parameter values of the R functions were used. For example, a RF was built with 500 classification trees, in which a random subset of $\{x_1, \ldots, x_p\}$ with size $\sqrt{p}$ were used as candidate splitting variables at each node.

3.4. Experiment 1

The results for Experiment 1 (and other two experiments) are summarized in Table 3.1. In Case (i), first, it seems that no method except IDW works well for the MN data, giving high misclassification error rates. Second, classifier PGA and IDW have good predictive performance for the PGA data. Third, neither classifier MN+PGA nor the simple average method works well for either MN or PGA data. Fourth, impressively, IDW always works best or nearly best, for either dataset.

In Case (ii), any method works (almost) equally well for the PGA data, and equally poorly for the MN data.

The conclusions in Case (iii) are similar to those in Case (i), except that classifier MN works much better for the MN data than it does in Case (i). Again IDW always works best or nearly best, for either dataset. Note the terrible performance of the classifier MN+PGA.

3.5. Experiment 2

First, we dichotomized the output of a PLS model using the cut-off value $c = 1/2$. Similar conclusions can be drawn as those for Experiment 1.

It may be argued that a Receiver Operating Characteristic (ROC) plot provides a more complete picture than the number of overall prediction errors. Increasing the cut-off value c gradually from 0 to 1, we obtain various numbers of false positives and false negatives, from which a ROC curve can be drawn and then used to measure the performance of any method.

Table 3.1. LOOCV errors (#Err) in three experiments (Exp 1–3), where NSC, PLS and RF were used as a base classifier. #MN was the frequency with which the MN classifier was selected by IDW, among which the number of misclassification errors (#Err) was also given.

Exp		Data	#Err of Method					Chosen by IDW	
			MN	PGA	MN+PGA	simple	IDW	#MN	#Err
1	Case (i)	MN data	8	10	11	9	6	18	5
		PGA data	11	5	9	8	5	2	1
		Both	19	15	20	17	11	20	6
	Case (ii)	MN data	10	10	9	10	10	19	7
		PGA data	2	3	3	2.5	2	19	2
		Both	12	13	12	12.5	12	38	9
	Case (iii)	MN data	5	11	9	8	6	16	5
		PGA data	11	3	11	7	3	6	1
		Both	16	14	20	15	9	22	6
2	Case (i)	MN data	9	11	13	10	10	17	6
		PGA data	12	3	4	7.5	3	0	0
		Both	21	14	17	17.5	13	17	6
	Case (ii)	MN data	7	8	6	7.5	7	21	6
		PGA data	1	1	2	1	1	18	0
		Both	8	9	8	8.5	8	39	6
	Case (iii)	MN data	3	14	4	8.5	5	19	3
		PGA data	16	3	6	9.5	3	0	0
		Both	19	17	10	18	8	19	3
3	Case (i)	MN data	9	11	10	10	9	22	9
		PGA data	11	1	2	6	1	1	0
		Both	20	12	12	16	10	23	9
	Case (ii)	MN data	6	4	7	5	5	17	3
		PGA data	4	1	1	2.5	1	8	1
		Both	10	5	8	7.5	6	25	4
	Case (iii)	MN data	3	15	3	9	5	21	3
		PGA data	11	2	3	6.5	2	5	1
		Both data	14	17	6	15.5	7	26	4

The ROC curves of the methods for the three cases were checked; the one for case (iii) is presented in Figure 3.2. The conclusions in Case (i) are: first, classifier MN is the worst; second, classifier PGA is among the best, along with IDW, followed by the simple average method; third, classifier MN+PGA is not good. In Case (ii), all the methods perform well. In Case (iii): first, classifier MN is the worst, followed by classifier PGA; second, IDW works best, followed by classifier MN+PGA, then the simple average method. These conclusions are in agreement with that drawn from Table 3.1.

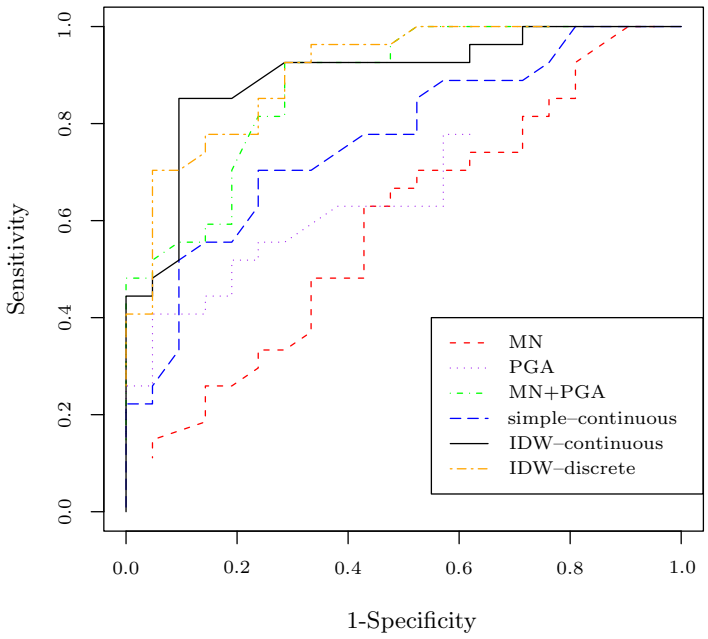

Figure 3.2. ROC curves for Experiment 2 Case (iii): using top 100 genes selected from the MN data.

3.6. Experiment 3

It is interesting to compare IDW with MN+PGA when the base classifier is an ensemble method, such as RF, which can better capture local features of inputs. Such a data-adaptive base classifier may automatically take account of the heterogeneity of the population, rendering model selection or model mixing largely unnecessary. However, an ensemble itself can be regarded as model mixing.

As shown in Table 3.1, MN+PGA with a RF as a base classifier has a much improved performance, but IDW is still among the best. Furthermore, in some applications where different base classifers are needed, as discussed elsewhere for gene function prediction with gene expression data and protein-protein interaction data (Xiao and Pan (2005)), popular ensemble methods, such as bagging, boosting and RF, cannot be directly applied, while the idea of IDW can.

3.7. Summary

Table 3.1 also gives frequencies with which IDW selected the MN classifier over the PGA classifier and the corresponding misclassification errors. In general, if the MN or the PGA classifier performs clearly better than the other,

IDW is more likely to select the winner. In summary, based on the predictive performance of the methods, IDW is often the leader.

4. Another Example

To partially demonstrate the generality and flexibility of IDW, we briefly consider an application to gene function prediction by combining two base classifiers built with gene expression data and protein-protein interaction data, respectively (Xiao and Pan (2005)).

Following Xiao and Pan (2005), we used the yeast data to predict 73 gene functions: for any given training data, base classifiers f_1 and f_2 were built using the two types of data; for any gene j and gene functional category F, f_1 and f_2 gave confidence votes on gene j's having function F, $V_1(j, F)$ and $V_2(j, F)$ respectively. To combine the two base classifiers, we used a logistic regression model

$$Logit[\pi(j, F)] = w_0 + w_1 V_1(j, F) + w_2 V_2(j, F),$$

where $\pi(j, F)$ was the true probability of gene j's having function F, and w_j's were constant weights to be estimated. Note that $w_1 = 0$ or $w_2 = 0$ corresponds to using only one source of data. The approaches taken in Xiao and Pan (2005) are similar to the ones described above.

To account for differing sensitivities and specificities of the base models for different functional categories, we consider a simplified implementation of IDW. Specifically, we used a logistic regression model

$$Logit[\pi(j, F)] = w_{0,F} + w_{1,F} V_1(j, F) + w_{2,F} V_2(j, F),$$

where the weights depended on a functional category F, an attribute of the input.

We used 10-fold CV to evaluate the methods. Specifically, we first used training data to build two base classifiers, obtained their confidence votes and then fitted the two logistic regression models to estimate the weights. Second, for each gene j and function F in the test data (i.e., left-out data un-used in training), from the fitted logistic regression models we obtained the estimated probability $\hat{\pi}(j, F)$. Third, for any threshold value c, if $\hat{\pi}(j, F) > c$, we classified gene j into functional category F; otherwise, no prediction was made. By changing c we obtained various sensitivities and specificities of the methods. The results of using only one source of data and of combining the two sources of data using constant weights and using IDW are presented as ROC plots in Figure 4.1, from which it can be seen clearly that (i) combining the two sources of the data improves over using only one source of data and (ii) IDW outperforms the standard method with constant weights.

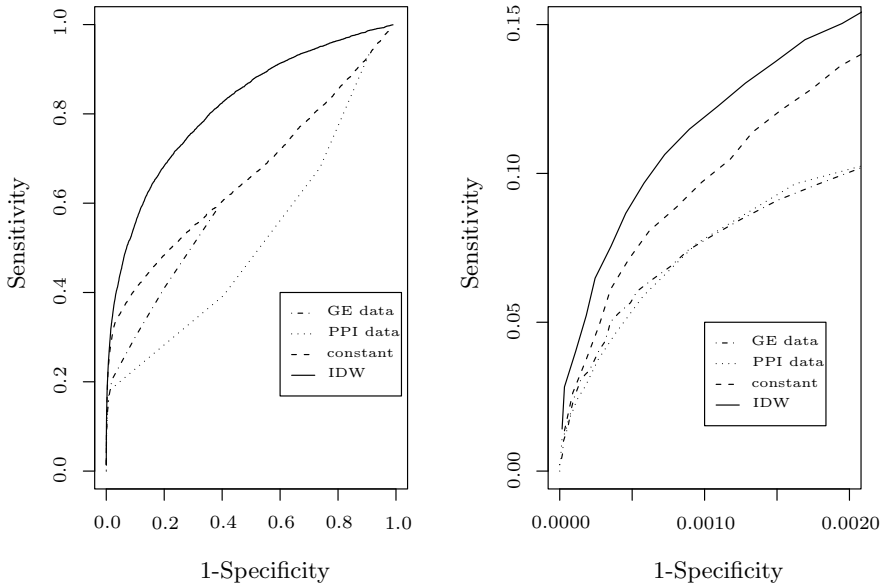

Figure 4.1. ROC curves for gene function prediction. The right panel is an enlarged part of the left one.

5. Discussion

This study was mainly motivated from the consideration of how to combine multiple models for multiple sources of data in computational biology, though the idea is general and equally applicable to combining models with a single source of data. Recent advances in high-throughput biotechnologies have generated large amounts, and various types, of genomic and proteomic data, such as DNA sequences, gene expression profiles, DNA-protein interactions and protein-protein interactions. It has been increasingly recognized that, in contrast to most standard approaches of analyzing a single source of data, a more powerful approach is to conduct integrative analyses of multiple sources of data. However, challenges remain. One of them is how to take account of the heterogeneity of the multiple sources of data and their varying sensitivities and specificities. In our main example, we have considered predicting the heart failure etiology using gene expression profiles collected from different patient populations and using different platforms of gene chips. Our other example concerned gene function prediction using gene expression data and protein-protein interaction data; the two sources of data have different information contents for different gene functional categories. The common approach of taking constant weights on multiple models built with individual sources of data is presumably suboptimal. Here

we have proposed an input-dependent weighting scheme, which can automatically select or weight more on the sources of data having higher sensitivities and specificities for a given input. In addition, because different sources of data may have different formats, different types of models may be needed. Our proposed IDW method has flexibility in allowing different types of candidate models to be combined.

The idea of using data-dependent weights to combine models has appeared in the literature. The best known is probably hierarchical mixtures of experts (HME) (Jordan and Jacobs (1994)). HME requires specifying each candidate model and data-dependent weights in some parametric forms, then estimating all the parameters in a single EM algorithm. Although it is neat to have a unified framework under maximum likelihood, HME loses the flexibility of allowing different types of candidate models, which may be necessary for different types of genomic and proteomic data. Furthermore, some non-likelihood based models, such as boosting, random forests, support vector machines (Vapnik (1998) and Lin et al. (2002)) and Psi-learning (Shen et al. (2003)), which are gaining popularity due to their good performance, may not be easily incorporated into HME. In addition, due to the well-known slow convergence of EM, it may not be computationally practical to fit HME for large datasets. Maclin (1998) considered using input-dependent weights to combine multiple models, but only in the context of boosting. Several authors have briefly mentioned that input-dependent weights can be used in stacking (Ripley (1996, p.66) and Hastie et al. (2001, p.253)), but no details are provided.

We emphasize that, in addition to combining models, IDW can be also used as a criterion for model selection, as illustrated in our example. In this paper, we propose using an estimated probability of making a correct prediction for a given input as a weight for a candidate model, but other measures on the predictivity of the model can be used. For example, if each candidate model is fitted using maximum likelihood, we can use the (predictive) log likelihood of the given input under a fitted candidate model as its weight. It has the same motivation as that of AIC, which is an asymptotically unbiased estimate of the *expected* predictive log likelihood. Of course, the main difference remains as before: our proposed weight depends on an input whereas AIC does not. Because a constant weight is usually an estimate of the *average* predictive performance of a model over its input space, whereas an IDW estimates the predictive performance of the model at a given input, the estimation of IDW, including the choice of a model such as (2.2), is in general more challenging. Further explorations and evaluations of the IDW method and its alternative implementations may be worthwhile.

Acknowledgement

The authors are grateful to the Editor and reviewers for helpful comments. WP and XH were partially supported by NIH grant HL65462 and a UM AHC Faculty Development grant, and GX by a Merck Fellowship.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proc. 2nd Int. Symp. Info. Theory* (Edited by B. N. Petrov and F. Csaki), 267-281. Akademia Kiado.

Ambroise, C. and McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Nat. Acad. Sci.* **99**, 6562-6566.

Boulesteix, A.-L. (2004). PLS dimension reduction for classification with microarray data. *Statist. Appl. Genetics Molecular Bio.* **3**, No. 1, Article 33.

Box, G. E. P. (1980). Sampling and Bayes' inference in scientific modelling and robustness (with discussion). *J. Roy. Statist. Soc. Ser. A* **143**, 383-430.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees.* Chapman and Hall, New York.

Breiman, L. (1996a). Heuristics of instability and stabilization in model selection *Ann. Statist.* **24**, 2350-2383.

Breiman, L. (1996b). Stacked regressions. *Machine Learning* **24**, 49-64.

Breiman, L. (1996c). Bagging predictors. *Machine Learning* **24**, 123-140.

Breiman, L. (2001). Random forests. *Machine Learning* **45**, 5-32.

Burnham, K. P. and Anderson, D. R. (2002). *Model Selection and Multimodel Inference: A Practical Information-theoretic Approach*, 2nd edition. Springer-Verlag, New York.

Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Amer. Statist. Assoc.* **78**, 316-331.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *J. Amer. Statist. Assoc.* **81**, 461-470.

Efron, B. (2004). The estimation of prediction error: covariance penalties and cross-validation (with discussion). *J. Amer. Statist. Assoc.* **99**, 619-642.

Efron, B. and Tibshirani, R. (1997). Improvement on cross-validation: the 632+ bootstrap method. *J. Amer. Statist. Assoc.* **92**, 548-560.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. System Sci.* **55**, 119-139.

Frank I. E. and Friedman J. H. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics* **35**, 109-135.

Fu, W. J., Carroll, R. J. and Wang, S. (2005). Estimating misclassification error with small samples via bootstrap cross-validation. *Bioinformatics* **21**, 1979-1986.

Geisser, S. (1975). The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.* **70**, 320-328.

George, E. I. and Foster, D. P. (2000). Calibration and empirical Bayes variable selection. *Biometrika* **87**, 731-747.

Hall, J. L., Grindle, S., Han, X., Fermin, D., Park, S., Chen, Y., Bache, R. J., Mariash, A., Guan, Z., Ormaza, S., Thompson, J., Graziano, J., de Sam Lazaro, S. E., Pan, S., Simari, R. D. and Miller, L. W. (2004). Genomic profiling of the human heart before and after mechanical support with a ventricular assist device reveals alterations in vascular signaling networks. *Physiological Genomics* **17**, 283-291.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York.

Hawkins, D. M., Wolfinger, R. D., Liu, L. and Young, S. S. (2003). Exploring blood spectra for signs of ovarian cancer. Manuscript.

Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statist. Sci.* **14**, 382-402.

Huang, X. and Pan, W. (2003). Linear regression and two-class classification with gene expression data. *Bioinformatics* **9**, 2072-2078.

Huang, X., Pan W., Park S., Han X., Miller L. W. and Hall J. (2004). Modeling the relationship between LVAD support time and gene expression changes in the human heart by penalized partial least squares. *Bioinformatics* **20**, 888-894.

Huang, X., Pan W., Grindle, S., Han X., Chen, Y., Park S., Miller L. W. and Hall J. (2005). A comparative study of discriminating human heart failure etiology using gene expression profiles. *BMC Bioinformatics* **6**, 205.

Ihaka, R. and Gentleman, R. (1996). A language for data analysis and graphics. *J. Comput. Graph. Statist.* **5**, 299-514.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.* **6**, 181-214.

LeBlanc, M. and Tibshirani, R. (1996). Combining estimates in regression and classification. *J. Amer. Statist. Assoc.* **91**, 1641-1650.

Li, H. and Gui, J. (2004). Partial Cox Regression Analysis for High-Dimensional Microarray Gene Expression Data. *Bioinformatics* **20**(Suppl 1), i208-i215.

Lin, Y., Lee, Y. and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning* **46**, 191-202.

Maclin, R. (1998). Boosting classifiers regionally. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 700-705. Madison, WI.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, New York.

Nguyen, D. V. and Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18**, 39-50.

Paik, M. and Yang, Y. (2004). Combining Nearest Neighbor Classifiers Versus Cross Validation Selection. *Statist. Appl. Genetics Molecular Bio.* **3**, No. 1, Article 12.

PGA(2004). NHLBI Program for Genomic Applications: Genomics of Cardiovascular Development, Adaptation, and Remodelling. Harvard Medical School. URL: http://www.cardiogenomics.org. Data downloaded in May 2004.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6**, 461-464.

Shen, X. and Ye, J. (2002). Adaptive model selection. *J. Amer. Statist. Assoc.* **97**, 210-221.

Shen, X., Tseng, G. C., Zhang, X. and Wong, W. H. (2003). On Psi-learning. *J. Amer. Statist. Assoc.* **98**, 724-734.

Shen, X., Huang, H.-C. and Ye, J. (2004). Adaptive model selection and assessment for exponential family models. *Technometrics* **46**, 306-317.

Shen, X. and Huang, H.-C. (2005). Optimal model assessment, model selection and model combination. Manuscript.

Simon, R., Radmacher, M. D., Dobbin, K. and McShane, L. M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J. Natl. Cancer Inst.* **95**, 14-18.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *J. Roy. Statist. Soc. Ser. B* **36**, 111-147.

Tan, Y., Shi, L., Tong, W., Hwang, G. T. G. and Wang, C. (2004). Multi-class tumor classification by discriminant partial least squares using microarray gene expression data and assessment of classification models. *Comput. Biol. Chemistry* **28**, 235-243.

Tibshirani, R., Hastie, R., Narasimhan, B. and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Nat. Acad. Sci.* **99**, 6567-6572.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley, New York.

Wold, S., Ruhe, A., Wold H. and W. J. Dunn III (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Statist. Comput.* **5**, 735-742.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks* **5**, 241-259.

Xiao, G. and Pan, W. (2005). Consensus clustering of gene expression data and its application to gene function prediction. Research Report 2005-012, Division of Biostatistics, University of Minnesota.

Yang, Y. (2001). Adaptive regression by mixing. *J. Amer. Statist. Assoc.* **96**, 574-588.

Yang, Y. (2003). Regression with multiple candidate models: selecting or mixing? *Statist. Sinica* **13**, 783-809.

Yuan, Z. and Yang, Y. (2003). Combining linear regression models: when and how? Manuscript. Available at http://www.public.iastate.edu/∼yyang/papers/index.html.

Zhang, H. and Singer, B. (1999). *Recursive Partitioning in the Health Sciences.* Springer-Verlag, New York.

Division of Biostatistics, School of Public Health, University of Minnesota, A460 Mayo, Minneapolis, MN 55455, U.S.A.

E-mail: weip@biostat.umn.edu

Division of Biostatistics, School of Public Health, University of Minnesota, A460 Mayo, Minneapolis, MN 55455, U.S.A.

E-mail: guanghx@biostat.umn.edu

Division of Biostatistics, School of Public Health, University of Minnesota, A460 Mayo, Minneapolis, MN 55455, U.S.A.

E-mail: xiaohong@biostat.umn.edu

(Received June 2005; accepted July 2005)