Efficient Learning by Combining Confidence-Rated Classifiers to Incorporate Unlabeled Medical Data

Weijun He[1], Xiaolei Huang[1], Dimitris Metaxas[1], and Xiaoyou Ying[2]

[1] Center for Computational Biomedicine Imaging and Modeling,
Division of Computer and Information Sciences, Rutgers University, NJ, USA
{weijunhe, xiaolei, dnm}@cs.rutgers.edu
[2] Sanofi-aventis, Bridgewater, NJ, USA
Xiaoyou.Ying@sanofi-aventis.com

Abstract. In this paper, we propose a new dynamic learning framework that requires a small amount of labeled data in the beginning, then incrementally discovers informative unlabeled data to be hand-labeled and incorporates them into the training set to improve learning performance. This approach has great potential to reduce the training expense in many medical image analysis applications. The main contributions lie in a new strategy to combine confidence-rated classifiers learned on different feature sets and a robust way to evaluate the "informativeness" of each unlabeled example. Our framework is applied to the problem of classifying microscopic cell images. The experimental results show that 1) our strategy is more effective than simply multiplying the predicted probabilities, 2) the error rate of high-confidence predictions is much lower than the average error rate, and 3) hand-labeling informative examples with low-confidence predictions improves performance efficiently and the performance difference from hand-labeling all unlabeled data is very small.

1 Introduction

In many learning algorithms in medical image analysis, the labeling of training data is often done manually. This process is quite time-consuming since a large set of training data is usually required. However, not all labeled data have the same level of effectiveness in improving a classifier. As in Support Vector Machines [1], only those "support vectors" that are located near the boundaries of different classes are the informative data that affect the final classifier. Hence if we can discover this type of "support vectors" in the unlabeled data, then we need only label these discovered informative data, include them in the training set and re-train the classifier. In this way, the amount of data to be labeled is greatly reduced without sacrificing the learning performance. In our approach we consider the confidence-rated classifiers that can predict a probability distribution over the labels for an example since the probability distribution enables us to determine the "informativeness" of the example.

J. Duncan and G. Gerig (Eds.): MICCAI 2005, LNCS 3749, pp. 745–752, 2005.

A single confidence-rated classifier, however, is often insufficient because in many medical images, multiple sets of features have very different characteristics and can not be effectively combined in a single classifier. For instance, image features are often grouped into different categories such as shape and texture. These feature sets have independent bases, and simply concatenating them into a single feature vector produces a complex, unstructured feature space that can potentially degrade learning and classification performance. To tackle this problem, in this paper we train separate confidence-rated classifiers on each category of features and then combine the predictions using Bayes rule, assuming conditional independence between classifiers trained on different feature sets. The classical voting classification algorithms, such as Bagging [2, 3] and AdaBoost [4, 5], are successful in improving the accuracy by combining multiple weak classifiers. Bauer and Kohavi [6] gave an empirical comparison of voting classification algorithms. However, these voting classification algorithms are generally applied to classifiers that just assign a label (not a probability) to an instance. Schapire and Singer [7] proposed new boosting algorithms using confidence-rated predictions, however, their extension to multi-class classification problems is not so straightforward. The new approach proposed in this paper for combining multiple confidence-rated classifiers based on Bayes rule efficiently addresses these problems.

Since our combining rule produces probability distributions over all labels for an unlabeled example, the predicted probabilities can be used to determine the "informativeness" of the example. Examples with high-confidence predictions are less informative than those with low-confidence predictions in improving the classifier. Hence it is more efficient to hand-label only those examples with low-confidence predictions. A classical method in the literature that improves learning by using unlabeled data is the co-training method [8, 9]. The basic idea is to organize the features of training examples into two different feature sets, and learn a separate classifier on each feature set. There are two assumptions in co-training. First, the two feature sets are redundant but not completely correlated. Second, each feature set would be sufficient for learning if enough data were available. Under these assumptions, the high-confidence predictions of one classifier on new unlabeled examples are expected to generate informative examples to enrich the training set of the other. However, these formal assumptions may not hold in many medical image applications that tend to have high complexity and dimensionality. In this paper, instead of trusting that each feature set is sufficient for learning, we determine the high-confidence predictions for new data by combining the opinions of all classifiers based on different feature sets. Our approach can be applied to multi-class classification problems directly.

2 Data Description and Preprocessing

The data we use are microscopic cell images. Each image consists of lots of cells in different developmental stages. The goal is to classify the cells into different stages and count the number of cells in each developmental stage. This problem has wide applications in the pharmaceutical industry for therapy evaluation.

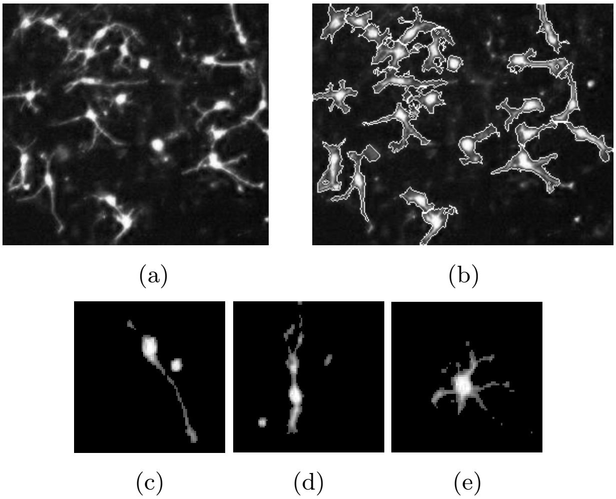

Fig. 1. (a) Cell image; (b) Segmented cell image; (c,d,e) are cells in developmental stage 0, 1 and 2 respectively

We first segment out the individual cells of interest. Since the cell nuclei are usually the brightest and cover a certain amount of area, we locate all cell nuclei by thresholding at a high intensity value and applying connected component analysis. Then we locate the regions occupied by all cells by thresholding the original image at a low intensity value. We apply dilation on all cell nuclei in the cell regions simultaneously until all cells are fully segmented. Finally we extract useful features from each cell for the purpose of classification. There are two categories of features: 1) binary image features, including roundness, eccentricity, solidity, extent and the diameter of a circle with the same area as the region; 2) gray image features, including mean and standard deviation of the gray level intensity. In the training phase, we first label a small set of cells from different developmental stages by hand, and include them in the initial training data set. The labels denote the different developmental stages. There are three developmental stages in our experiments (Fig. 1): 0 - beginning; 1 - immature; 2 - mature.

Because the hand-labeling phase is tedious and time consuming, the initial labeled training set we can acquire is limited. Our methodology is to tackle this problem by strategically adding unlabeled data into the training data set based on the evaluation of the unlabeled data using a confidence-rated classification mechanism. The proposed mechanism is described in the next sections.

3 Learning Framework

Our learning framework is outlined by the flowchart in Fig. 2. In the framework, we train multiple confidence-rated classifiers on separate groups of features. When presented with unlabeled data, each classifier produces a confidence

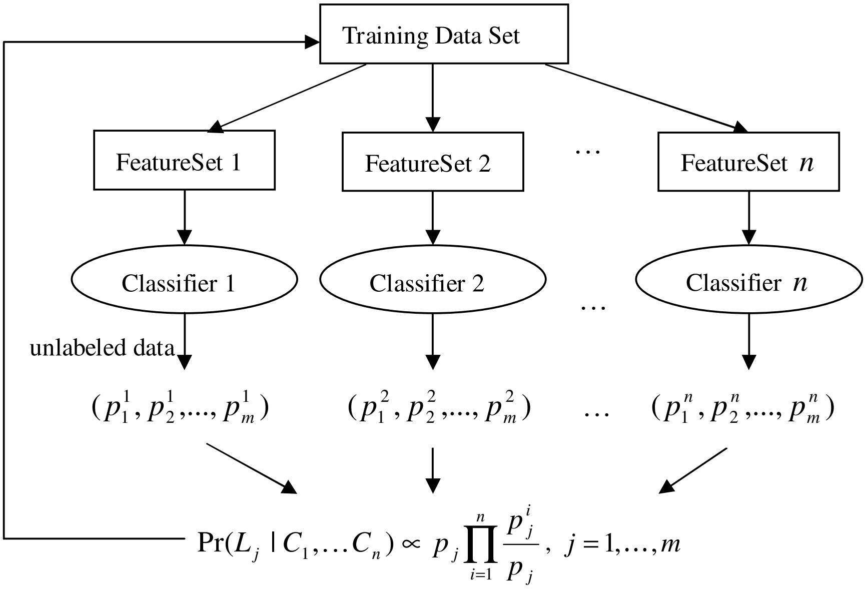
Training Data Set
FeatureSet 1
FeatureSet 2
…
FeatureSet n
Classifier 1
Classifier 2
…
Classifier n
unlabeled data
$(p_1^1, p_2^1, \ldots, p_m^1)$
$(p_1^2, p_2^2, \ldots, p_m^2)$
…
$(p_1^n, p_2^n, \ldots, p_m^n)$
$\Pr(L_j \mid C_1, \ldots C_n) \propto p_j \prod_{i=1}^{n} \frac{p_j^i}{p_j}, \; j = 1, \ldots, m$

Fig. 2. Learning framework

rate of its own prediction. Turney et al. [10] has argued that opinions from independent modules should be combined multiplicatively. We extend this idea to combine predictions of classifiers trained on different feature sets.

3.1 Combining Confidence-Rated Classifiers Using Bayes Rule

Our combining approach is to apply Bayes rule to evaluate the final confidence rate based on the confidence rates given by the multiple, independent classifiers. Suppose we have n classifiers, m labels, and classifier i is trained on the feature set i. p_j^i $(i = 1, \ldots, n, j = 1, \ldots, m)$ denotes the probability that classifier i assigns label j to an input unlabeled example. L_j denotes that the label of an example is j. Let $C_i = (p_1^i, p_2^i, \ldots, p_m^i), p_j = Pr(L_j)$. Using Bayes rule, we have

$$Pr(L_j|C_1, C_2, \ldots, C_n) = \frac{Pr(C_1, C_2, \ldots, C_n|L_j)Pr(L_j)}{Pr(C_1, C_2, \ldots, C_n)} \tag{1}$$

Assuming conditional independency of $C_1, C_2, \ldots, C_n$, we have

$$Pr(L_j|C_1, C_2, \ldots, C_n) \propto Pr(C_1|L_j)Pr(C_2|L_j) \ldots Pr(C_n|L_j)Pr(L_j) \tag{2}$$

Since $Pr(C_i|L_j) = \frac{Pr(L_j|C_i)Pr(C_i)}{Pr(L_j)}$, Eq. 2 can be rewritten as

$$Pr(L_j|C_1, C_2, \ldots, C_n) \propto Pr(L_j) \prod_{i=1}^{n} \frac{Pr(L_j|C_i)}{Pr(L_j)} \tag{3}$$

Table 1. The difference between our approach with naïve Bayes classifier

	Known	$Pr(L_j\|O_1,\ldots,O_q) \propto$
naïve Bayes classifier	$Pr(O_i\|L_j)$	$Pr(L_j)\prod_{i=1}^{q} Pr(O_i\|L_j)$
Our approach	$Pr(L_j\|C_i)$	$Pr(L_j)\prod_{i=1}^{n} \frac{Pr(L_j\|C_i)}{Pr(L_j)}$

Using our abbreviated notation, Eq. 3 can be simplified as

$$Pr(L_j|C_1, C_2, \ldots, C_n) \propto p_j \prod_{i=1}^{n} \frac{p_j^i}{p_j} \tag{4}$$

The above formula intuitively says that if the predicted probability p_j^i is greater than (equal to, less than) the prior probability p_j, then this prediction will increase (not affect, decrease) the final probability given to label j. The following theorem relates our approach to naïve Bayes classifier.

Theorem 1. *If Classifier $i, i = 1, \ldots, n$ are themselves naïve Bayes classifiers using disjoint feature sets, then the combined classifier is a naïve Bayes classifier using all features.*

Proof. Suppose the feature sets are $\phi = \{\phi_i, i = 1, \ldots, n\}$ and each classifier C_i is a naïve Bayes classifier based on the feature set ϕ_i. Then we have:

$$Pr(L_j|C_i) \propto Pr(L_j) \prod_{\alpha \in \phi_i} Pr(\alpha|L_j) \tag{5}$$

$$\begin{aligned} Pr(L_j|C_1, C_2, \ldots, C_n) &\propto Pr(L_j) \prod_{i=1}^{n} \frac{Pr(L_j|C_i)}{Pr(L_j)} \\ &\propto Pr(L_j) \prod_{i=1}^{n} \prod_{\alpha \in \phi_i} Pr(\alpha|L_j) \\ &\propto Pr(L_j) \prod_{\alpha \in \phi} Pr(\alpha|L_j) \qquad (6) \end{aligned}$$

Hence each classifier can be viewed as a mapping from a set of concrete features to a single abstract feature (a probability distribution over all labels). Our combining approach then predicts the final classification using Bayes rule on these abstract features. Denote the observation on feature $i, i = 1, \ldots, q$ as O_i. In table 1, we compare our combining approach with the naïve Bayes classifier. Our combining approach does not assume any particular class-conditional density model as the naïve Bayes classifier does for the continuous variables. Instead base classifiers trained on different feature sets are applied to generate probability distributions over the labels for an example.

3.2 Exploring the Unlabeled Data

Once we have acquired the combined probability distribution over an unlabeled example, we can use it to determine the "informativeness" of this example.

The intuition behind our approach is that not all unlabeled data have equal effectiveness in improving the classifier. For example, in SVM only the support vectors are used in determining the final classifier. If we know those support vectors, it would suffice to label only those data and train the classifier on them. So our strategy is to find those potential "support vectors", and present them only for hand-labeling. In this way, the amount of human efforts needed to acquire a large labeled training set is greatly reduced.

We notice that the support vectors are near the boundaries between two classes. And the classifier does not predict well their labels. So, the probabilities given by the classifier can be used to discover those informative unlabeled data. If the predicted probability that one unlabeled example belongs to a certain class is high, we include this example along with the predicted label directly into the training set. However, these data may only enlarge the training set without helping much to improve the classifier. On the other hand, if the probability of an unlabeled example belonging to any class is below some threshold, the current classifier is uncertain about the label of this example. Therefore, this type of data is most probably lying around the boundary between two classes. Hand-labeling these data that the current classifier is uncertain about and adding them into the training set will most efficiently improve the classifier. In this way, by quantitatively evaluating the relationship between the unlabeled data and the current classifier, we only need to label those "most profitable" unlabeled data without sacrificing much in performance.

4 Experiments

We tested our algorithm by collecting our data from 40 microscopic cell images, each containing about 50 cells of all developmental stages after segmentation. We classify the cells into 3 developmental stages: beginning, immature and mature. For each segmented cell, we extract two separate sets of features: one related to shape and geometry using the thresholded binary image, and the other related to intensity statistics based on the gray-level image. The multi-class logistic regression [11] classifier is applied on the binary and gray feature sets separately and we get two probability distributions for each unlabeled cell example. However our learning framework does not assume any specific base learning method. We choose logistic regression since it can predict a probability distribution and can be applied to a wide variety of situations as long as the difference between the logarithms of the class-conditional density function is linear in the variables.

In the first experiment, we compare our Bayesian combination approach which takes into account the prior probability of each class with multiplicative combination that does not consider the prior probabilities. The label we assign to an example is the label with the highest predicted probability. In table 2, we can see that combining the predictions without unlabeled data using Bayes rule is better than just simply multiplying the probabilities without considering the prior probability of each class. In each run, the prior probabilities are estimated from the training set (randomly selected 35 labeled examples). We expect that the performance gain would be greater if more feature sets were combined.

Table 2. Prediction accuracy without unlabeled data (average over 20 runs)

Binary features	Gray features	Multiplicative combination	Bayesian combination
81.1%	69.7%	79.3%	81.7%

Table 3. Prediction accuracy (a) with just initial 35 training data; (b) with hand-labeled "informative" unlabeled data, along with the number of such data; (c) with all 100 unlabeled data hand-labeled

runs	(a)	(b)	(c)
1	78.6%	91.1% (24)	92.9%
2	78.6%	83.9% (16)	83.9%
3	80.3%	82.1% (30)	83.9%
4	82.1%	83.9% (31)	85.7%
5	82.1%	87.5% (14)	87.5%

In the second experiment, we examine whether the confidently predicted unlabeled data are really correctly labeled and compute the proportion of this type of data among all unlabeled data. The result depends on the threshold applied on the combined probability. In this experiment, if the predicted probability of one example belonging to one class is higher than 90%, then we treat this example as a confidently predicted example. Over 20 runs, the percentage of the confidently labeled data over all unlabeled data is 60.9%. The average prediction accuracy for those confidently labeled data is 92.3%, which is much higher than the average prediction accuracy (81.7%). However if a higher labeling accuracy for the added unlabeled data is required, we need to increase the threshold, which will decrease the percentage of confidently labeled data. So there is a tradeoff here and the choice of the threshold depends on the application.

Finally, we show that hand-labeling a few informative examples with low-confidence predictions efficiently improves performance, and that the performance difference is small between hand-labeling the few informative examples and hand-labeling all unlabeled data. Similarly we need to set a threshold. In this experiment, if the maximal predicted probability of one example belonging to any class is lower than 80%, then we treat this example as an ambiguous example to the current classifier and we need to label it by hand. In table 3, we can see that, over five runs, the numbers of such ambiguous (i.e. "informative") unlabeled examples are 24, 16, 30, 31 and 14, which are much less than the total number of unlabeled examples (100). By labeling only this reduced number of unlabeled data, however, we achieve a performance that is comparable to that by labeling all unlabeled examples.

5 Discussions and Conclusions

In this paper, we have presented a Bayesian strategy for combining confidence-rated predictions of classifiers trained on different feature sets. Our method generates a probability distribution over the labels for an unlabeled example. We utilize these probability distributions to filter out two groups of unlabeled data. One group is the confidently labeled data. We add them directly into the training set. Compared to the co-training method, our approach combines the opinions from different classifiers to ensure that the self-labeled data are correct with very high probability. The other group of the filtered unlabeled data includes those potentially informative examples for whose labels the current classifier is uncertain. By hand-labeling only these informative data, we achieve comparable performance with hand-labeling all data. This results in greatly reduced training expense. Therefore the training phase of our method is not static, but dynamic.

References

1. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines and other kernel-based methods. Cambridge University Press (2000)
2. Breiman, L.: Bagging predictors. Machine Learning **26** (1996) 123–140
3. Quinlan, J.R.: Bagging, boosting, and C4.5. In: Proc. AAAI-96 Fourteenth National Conf. on Artificial Intelligence. (1996) 725–730
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55** (1997) 119–139
5. Schapire, R.E.: A brief introduction to boosting. In: Proc. of 16th Int'l Joint Conf. on Artificial Intelligence. (1999) 1401–1406
6. Bauer, E., Kohavi, R.: An empirical comparison of voting classification problems: Bagging, boosting and variants. Machine Learning **36** (1999) 105–142
7. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning **37** (1999) 297–336
8. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of the 1998 Conf. on Computational Learning Theory. (1998) 92–100
9. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual dectectors using co-training. In: Proc. of the Int'l Conf. on Computer Vision. (2003) 626–633
10. Turney, P., Littman, M., Bigham, J., Shnayder, V.: Combining independent modules to solve multiple-choice synonym and analogy problems. In: Proc. of the Int'l Conf. on Recent Advances in Natural Language Processing. (2003) 482–489
11. Anderson, J.A.: Logistic discrimination. In Krishnaiah, P.R., Kanal, L.N., eds.: Handbook of Statistics 2. (1982) 169–191