# A Framework for Incorporating Class Priors into Discriminative Classification

Rong Jin[1], Yi Liu[1]

[1] Department of Computer Science and Engineering, Michigan State University,
East Lansing, MI48824, U.S.A.
{rongjin, liu3}@cse.msu.edu

**Abstract.** Discriminative and generative methods provide two distinct approaches to machine learning classification. One advantage of generative approaches is that they naturally model the prior class distributions. In contrast, discriminative approaches directly model the conditional distribution of class given inputs, so the class priors are only implicitly obtained if the input density is known. In this paper, we propose a framework for incorporating class prior proportions into discriminative methods in order to improve their classification accuracy. The basic idea is to enforce that the distribution of class labels predicted on the test data by the discriminative model is consistent with the class priors. Therefore, the discriminative model has to not only fit the training data well but also predict class labels for the test data that are consistent with the class priors. Experiments on five different UCI datasets and one image database show that this framework is effective in improving the classification accuracy when the training data and the test data come from the same class proportions, even if the test data does not have exactly the same feature distribution as the training data.

## 1 Introduction

Machine learning approaches to classification usually fall either into the discriminative (or conditional modeling) category, or the generative category. Discriminative classification directly attempts to model $p(y\,|\,\vec{x})$ where $\vec{x}$ is the vector of input features and $y$ is the class label. Generative approaches model the joint distribution, split into the class prior and the class conditional density: $p(y,\vec{x}) = p(y)\,p(\vec{x}\,|\,y)$. One difference between them is the conditional model usually only focuses on the relationship between the input features and the class label, while the generative model has to explain both how the inputs are generated and how the class label is associated with the input data. One usually finds that state-of-the-art conditional models perform better than generative models on classification problems. More detailed studies of the comparison of conditional models and generative models can be found in [8].

However, compared to discriminative approaches, generative approaches have the advantage in that they are able to explicitly make use of class priors for predicting class labels. Knowledge of the class priors or proportions can be very useful in several contexts. Assume that the learning algorithm is provided with labeled training data and unlabeled test data. First, if the training data does not have the same proportions of points in each class as the test data, incorporating knowledge of class priors can help create a classifier that predict classes with the correct probabilities on the test data. Second, sometimes the class conditional probabilities $p(\vec{x} \mid y)$ are somewhat different from the training data to the test data. This is a situation which most learning algorithms are not designed for, but it occurs in practice due to the fact that the labeling process can be biased. Incorporating knowledge of the class priors can make a learning algorithm more robust to inaccuracies in the class conditional probabilities obtained from training data. In the extreme case, the inputs of the training data and the test data may be completely random numbers and only the class priors are consistent through the whole dataset. The best strategy of predicting class labels for the test data would be to always predict the class label with the highest prior. Unfortunately, since discriminative models focus on learning the mapping from input data to class label, $p(y \mid \vec{x})$, without representing the input density $p(\vec{x})$, most methods can't directly model or take advantage of the marginal class distribution, $p(y)$.

In this paper, we propose a framework that is able to *explicitly* incorporate class prior information into discriminative learning. This framework is based on the assumption that the class priors give a reasonably accurate description for the class distribution of the test data. Therefore, the discriminative model learned from the training data should not only explain the class labels for the training data well but also predict the class labels for the test data in such a way that the distribution of the predicted class labels for the test dataset is also coherent with the class priors. Clearly, this framework differs from the traditional approach for discriminative learning where the objective is to make the class labels predicted by the model consistent with the assigned class labels on the training dataset. Furthermore, our framework is able to utilize both training data and testing data in the construction of discriminative models, while traditional approaches for discriminative learning only take advantage of training data.

This framework can be useful when the training data and the test data have the same class priors but do not have exactly the same feature distributions and therefore the model learned from the training data may not be appropriate for the test data. Differences between the training data and the test data can be caused by the fact that either the sampling for the training data is quite different from the sampling for the test data, or the amount of training data is too small to give a good representation for the whole dataset. The other interesting aspect of this framework is that it allows the discriminative model to use a mixture of training data and test data. Thus, this framework is able to deal with learning problems in which only a small number of training examples are available and the majority of instances are unlabelled. Unlike previous works on the combination of labeled data and unlabeled data, which mainly focus on the generative model, the framework provides room for the discriminative model to take advantage of unlabeled data.

The rest of this paper is arranged as follows: section 2 will discuss related work. The formal description of the framework is presented in Section 3. Section 4 describes the empirical study of this framework. Conclusions and future work are presented in Section 5.

## 2 Related Work

As already discussed in the introduction, the main idea of this paper is to incorporate class prior information into discriminative learning. Therefore, it combines some aspects of learning both discriminative and generative models. There have been several studies on the improvement of discriminative models using information from a generative model [6]. The motivation of that work is based on the observation that sometimes the generative model captures properties of the input distribution that are meaningful to the discrimination problem. Therefore, if we can influence the discriminative model with a generative model that is specific to the problem, for example choosing a discriminative function that is coherent with the generative density, we may be able to gain better performance than using a generic discriminative model. Approaches based on this idea, such as combining the support vector machine with fisher kernels derived from a generative model, have shown significant improvement in classification problems. Unlike that work, in our framework we don't change the kernel function; instead, we only consider the class prior information as an extra hint to be used by the discriminative model.

Since this framework is taking advantage of both the training data and the test data, it is strongly related to the work on learning from the mixture of labeled and unlabeled data [9]. Many of works on this problem assume some form of generative model, which is used to explain both the labeled data (i.e. the inputs and the label) and the unlabeled data (i.e. just the inputs). In cases where only a small amount of data are labeled, a model learned from this data can be quite skewed and the incorporation of unlabeled data can help avoid idiosyncrasies of the labeled data to some extent. Unlike this work, our framework focuses on incorporating unlabeled data into discriminative training. Other works on learning from the mixture of labeled and unlabeled data have focused on using unlabeled data for model regularization and model selection. One example is the transductive SVM [7], where the classification margin is influenced both by the labeled and unlabeled data. Unlike their work, in this framework, we refine the learned model by only examining the discrepancy between the class distribution of the predicted labels of unlabeled data and the 'true' class priors.

## 3 Incorporating Class Priors into Discriminative Training

The basic logic behind this framework can be simply understood as follows: consider the case when the test data has quite different patterns from the training data. This situation can happen if there is very little training data, or as in many real applications, if the training data and testing data come from different sources. Then, applying the discriminative model that is learned from the training data directly to label the test data will be problematic. If we have prior knowledge on the class distribution for the test data, we may be able to find out the fact that the test data are noisy by simply examining the difference between the class priors and the distribution of the class labels for the test data predicted by the discriminative model. If there is a significant discrepancy between these two distributions, we will suspect that the learned model may not be appropriate for the test data and needs to be adjusted. In order to refine the discriminative model, we need to do two things: first, we can adjust the probability of classes for the test data computed from the original discriminative model in such a way that the averaged distribution of the predicted class labels for the test data is shifted toward the class priors. Then, the test data with the adjusted class probabilities will be included in the training data and a discriminative model will be retrained over the ‘enlarged training dataset’. The procedures of adjusting the class probabilities for test data using priors and retraining the discriminative model will be carried out iteratively until it reaches some local maximum.

### 3.1 Model Description

The essence of a discriminative model is the computation of the conditional probability for a class label $y$ given the input vector $\vec{x}$, i.e. $p(y \mid \vec{x})$. The learning of a discriminative model can be formalized as the search for a model that maximizes the log-likelihood of the training data, i.e.

$$L = \sum_{i \in Train} \log p(y_i \mid \vec{x}_i, \mathtt{M}) \tag{1}$$

where $\mathtt{M}$ stands for a discriminative model, $\vec{x}_i$ is the i$^{th}$ training data point and $y_i$ stands for its class label.

In order to incorporate class prior information into a model, a discriminative model will not only have to explain the training data well but also to predict class labels for the test data in such a way that the distribution of predicted class labels for the test data is consistent with class priors. Therefore, we need an extra term in Equation (1) that can account for the discrepancy between the two distributions. In the following sections, we will discuss three different approaches. To this end, for every instance in the test dataset, an unknown distribution over class labels is introduced. This represents the estimated distribution over classes, which will incorporate both the prior class constraints and the model predictions. Moreover, we will see that it considerably simplifies the computation in optimization. Let $r_k$ be this estimated class distribution for the k$^{th}$ data point in the test set, and value $r_{k,y}$ be the probability for the k$^{th}$ test data point to be in class $y$. To enforce the consistency between class priors of training data and test data, we impose the following constraint on the estimated class probability $r_{k,y}$, i.e.,

$$\forall y \quad \frac{1}{N_{test}} \sum_{k \in Test} r_{k,y} = p_y \tag{2}$$

Now, the next step is to connect the estimated class probability $r_{k,y}$ to our objective function. Of course, we want the distribution of class labels predicted by model $\mathtt{M}$, i.e., $p(y \mid \vec{x}_k, \mathtt{M})$, to be consistent with the estimated class distribution $r_{k,y}$. Therefore, the objective in (1) can be modified as:

$$L' = \sum_{i \in Train} \log p(y_i \mid \vec{x}_i, \mathtt{M}) + \sum_{k \in Test} \sum_{y} r_{k,y} \log \frac{p(y \mid \vec{x}_k, \mathtt{M})}{r_{k,y}} \tag{3}$$

In above, the KL divergence is introduced as the measurement of consistency between the estimated class distribution $r_{k,y}$ and the distribution of the predicted class labels $p(y \mid \vec{x}_k, \mathtt{M})$. By maximizing (3) under constraints (2), we ensure that the discriminative model is consistent with the estimated class distribution $r_{k,y}$, which indirectly forces consistency between the class priors of test data and of training data.

### 3.2 Finding the Optimal Solution

As indicated in Equation (3), the objective function contains two different sets of parameters, namely the model parameters $\mathtt{M}$ and the estimated class distribution $r_{k,y}$. Therefore, we can optimize the objective function in (3) by alternatively freezing one set of parameters. More specifically, we will first optimize the objective function in (3) using only the discriminative model parameters, and then search for the estimated class distributions that optimize (3) under the constraints in (2). It is not difficult to see that the strategy used in the optimization exactly corresponds to the intuition stated at the beginning of this section.

In the first step of optimization, the $r_{k,y}$ are held fixed (as target distributions for the test data) so the constraint in (2) is not relevant. Thus, the discriminative model can be trained with almost no modification except that both the training data and the test data are fed into the learning module. Of course, any discriminative classifier that accepts distributions as targets can be used here.

For the second step of optimization, we need to find the set of estimated class distributions that maximizes the objective function in (3) subject to the constraints in (2). Since parameters for the discriminative model are frozen, the objective function in (3) is simplified as:

$$L" = \sum_{i \in Test} \sum_{y} r_{k,y} \log \frac{p(y \mid \vec{x}_k, M)}{r_{k,y}} \tag{3'}$$

The problem of maximizing (3') under the constraints in (2) is exactly the same problem as solved in maximum entropy (ME) models [1]. The original version of maximum entropy model is to find a set of probabilities that not only maximize the entropy function and but also satisfy a set of linear constraints. This can be extended to the case when the objective function is not an entropy function but a KL divergence between the distribution to be optimized and a set of given probabilities, i.e. a *minimum relative entropy* (MRE) problem, which is exactly our problem.

## 4 Experiments

In this experiment, we examined the effectiveness of our model in terms of using class priors to improve classification accuracy. More specifically, we would like to address two scenarios of application for this framework:

1) *A scenario of a small number of labeled examples.* In this case, we will expose the system to a small number of labeled examples together with a large number of unlabeled examples. Under the assumption that a reliable estimation of class

**Table 1**: UCI datasets used in our experiments

| Data Set | Number of Feature | Number of Class | Number of Instance |
|---|---|---|---|
| Ecoli | 7 | 5 | 327 |
| Wine | 13 | 3 | 178 |
| Pendigit | 16 | 10 | 2000 |
| Iris | 14 | 3 | 154 |
| Glass | 10 | 5 | 204 |

priors are available, we can examine how well this model is able to improve the model by taking into count the large number of unlabeled data.

2) *A scenario of heterogeneous training and testing examples.* In this case, we assume that the training data are somehow different from testing data in some respects. Therefore, a classification model learned from training data is not appropriate for the testing data. By inspecting the discrepancy between the class priors and the class distribution of the predicted labels of unlabeled data, we expect to adjust our model to the characteristics of the testing data.

### 4.1 Experiment Design

The discriminative model used for the experiment is the conditional exponential model [1], in which conditional probability $p(y\,|\,\vec{x})$ is defined as $p(y\,|\,\vec{x}) = \exp(\vec{\lambda}_y \cdot \vec{x}) \big/ Z(\vec{x})$, where $\vec{\lambda}_y$ is the weight vector for class y and $Z(\vec{x})$ is the normalization factor. A conjugate gradient [10] is used to find the appropriate weight vectors.

To illustrate the effectiveness of our framework on the two different scenarios mentioned before, we tested the algorithm against two different groups of datasets. For the first scenario, we use five UCI datasets as the testbed. We use a small portion of each UCI dataset as training examples and leave majority of the dataset as testing examples. The detailed information about the five datasets is listed in Table 1. For the second scenario, we tested our algorithm on both the synthesized data that are generated from the above five UCI datasets and real image data. To simulate the difficult circumstance in which test data and training data have different feature distributions, for every feature, we uniformly randomly generate a weight factor ranging from 1 to 1.5 and multiple it with the corresponding feature of the testing data. By this 'corruption' procedure, the weights of the exponential model learned from the training data will not be appropriate for the test data because the scale of the test are changed. By testing our algorithm against the synthesized datasets, we are able to see how effectively our framework is able to adjust the model parameters $\vec{\lambda}_y$ according to the large number of unlabeled data.

The other dataset that we used for the second scenario is the image dataset. We use the images downloaded from the image directory of Google as the training examples and images from Corel image database as testing examples [3]. Six image categories are used in the experiment, i.e., category 'food', 'building', 'bird', 'sky', 'fish' and 'fruit & vegetable'. Each category contains 100 training images and 100 testing images. The training images are acquired by querying the image directory of Google with the name of categories as the query words. The top ranked 100 images from Google are used as the training examples. The testing images are collected by randomly sampling 100 images out of the corresponding categories from Corel database. Apparently, images downloaded from Google image database will be considerably different from images from Corel database. The extended color co-occurrence matrix [5] is used for image representation, which have shown its effectiveness in image classification. For each image, totally 500 image features are extracted. More detailed discussion about image classification can be found in [2, 4, 11].

The key component in this framework is the knowledge of class priors. To examine the impact of class prior accuracy on classification performance, we introduce three different ways of estimating class priors:

1) 'Empirical Estimate': Estimate the class priors only based on the training data. Since we use small portion of the data as training, this estimate of class prior can be quite inaccurate for the test data.
2) 'Optimal Estimate': Estimate the class priors based on the test data. Of course, this estimate gives the exact class distribution for the test data and is not realistic. However, the performance under this estimate gives a sense of an upper bound performance of our framework.
3) 'Intermediate Estimate': Estimate the class priors based on all the data including

**Table 2:** Classification Errors for UCI datasets when 25% data are used for training

| Data Set | No Prior | Empirical Estimate | Intermediate Estimate | Optimal Estimate |
|---|---|---|---|---|
| Ecoli | 16.1% | 20.6% | 16.7% | 16.0% |
| Wine | 15.1% | 15.0% | 9.1% | 8.0% |
| Pendigit | 8.8% | 12.4% | 8.7% | 8.0% |
| Iris | 5.6% | 16.0% | 4.5% | 3.7% |
| Glass | 9.8% | 14.2% | 3.9% | 2.7% |

**Table 3:** Classification Errors for UCI datasets when 10% data are used for training

| Data Set | No Prior | Empirical Estimate | Intermediate Estimate | Optimal Estimate |
|---|---|---|---|---|
| Ecoli | 32.4% | 26.9% | 21.2% | 21.8% |
| Wine | 20.8% | 26.0% | 15.0% | 15.1% |
| Pendigit | 11.8% | 17.9% | 11.8% | 11.6% |
| Iris | 7.5% | 23.6% | 5.3% | 4.4% |
| Glass | 5.7% | 27.8% | 2.2% | 2.6% |

the test data and training data. Definitely, this estimate will be better than the first case and worse than the second case.

### 4.2 Scenario 1: A Small Number of Labeled Examples

In this experiment, we set the training size to be 10% and 25% of the total dataset, respectively. The averaged classification errors based on cross validation for the proposed algorithm are listed in Table 2 and 3. We also included the classification results when no class priors information is used.

First, by comparing the performance listed in Table 2 to what listed in Table 3, it is clear that, by decreasing the amount of training examples from 25% to 10%, all learning methods on most UCI datasets suffers degradation in performance except the 'Glass' dataset. Second, comparing the proposed framework using different estimators of class priors, it is clear that the new framework with optimal estimator appears to have the best performance while the intermediate estimator gives the second best performance. The new framework with these two estimators of the class priors appears to substantially outperform the baseline model, i.e., the simple discriminative model without using class priors. This fact indicates that our algorithm is effective in improving the performance of discriminative classifier with reliable estimates of class priors. Third, the proposed algorithm with empirical estimates appears to perform significantly worse than the original discriminative model without using class priors. Since the empirical estimator bases its estimates on the empirical class distribution of training data and only small portion of training examples are available in the study, the empirical estimates usually gives poor estimation of class priors, which results in poor performance of the proposed algorithm. Based on this fact, we can see that, it is very important to our algorithm to have accurate estimates of class priors.

### 4.3 Scenario 2: Heterogeneous training data and testing data

In this subsection, we will test our algorithm against the case when the testing data have different feature distributions from the training data. This is a practically relevant scenario, which is rarely studied in machine learning. First, we will test the propose algorithm on the synthesized datasets, which are generated from the five UCI datasets. The 'corruption' procedure has already been described in section 4.1. The results for the proposed algorithm with three different estimators of class priors together with the discriminative model without class priors are listed in Table 4 and 5.

First, by comparing the results in Table 4 and 5 to the results in Table 2 and 3, it is clear that, by multiplying the testing data with a random weight factor, the performance of the discriminative model without using class priors suffers from a severe degradation. On the contrary, the proposed algorithm appears to suffer from much smaller degradation for all three different estimators. This fact indicates that, the proposed algorithm is robust to the 'corruption' on the features. Second, it is clear that, the proposed framework with all three different estimates is significantly better than the discriminative model without class priors. Unlike the results presented in previous scenario, where the empirical estimates gives rise to poor performance due to its inaccurate estimation on class priors, for this scenario, even the empirical estimate is able to result in a better performance than the original discriminative model for most datasets. The fact that the proposed algorithm is able to outperform the discriminative model without class priors indicates that our algorithm is effective in dealing with the situation when the testing data are quite different from the training data.

In addition to testing our algorithm against the synthesized dataset, we also examine our algorithm over the problem of image classification. The details of image databsets have already been described in the section 4.1. The downloaded 600 images are used as training examples and the 600 images from Corel database are used as testing instances. The classification error for the discriminative model without class priors is 66.9% and 60.9% for the proposed model assuming that the class priors equal to 1/6. Again, with accurate knowledge on class priors, we are able to decrease the classification error significantly. Notice that, the classification errors in this experiment is quite high, over 60%, while some image classification works show extremely good performance over Corel datasets. We believe the main reason for that is because the images from Google do not resemble the images from Corel in many cases. Table 6 shows two images of category 'bird' from both Google and Corel database. Apparently, the training examples are very different from testing images, either from the viewpoint of color or from the viewpoint of texture. We think that is the reason why this task is so hard and causes so large testing errors.

## 5. Conclusions

In this paper, we proposed a framework that is able to incorporate class prior information into training a  discriminative model. This algorithm can also be thought as a machine learning algorithm which allows a discriminative model to use a mixture of labeled and unlabeled data. In the empirical study over five different UCI datasets and Corel image database, this algorithm is able to improve the performance of the conditional exponential model significantly when the number of training examples is small and when the test data are heterogeneous from the training data. Therefore, we conclude that the new algorithm is able to help the performance even with inaccurate estimation for class priors and the improvement depends on the accuracy of the estimation. Usually large improvements were found when accurate class priors were incorporated into training but these improvements vanished when the class priors had substantial inaccuracies. Thus, more research work is needed in order to study how to improve the classification accuracy in case of inaccurate class priors.

## Acknowledgement

We thank Dr. Zoubin Ghahramani and Dr. Huan Liu for their discussion and suggestion on this work.

## References

1. Berger, A., S.D. Pietra, and V.D. Pietra, *A Maximum Entropy Approach to Natural Language Processing.* Computational Linguistics, 1996. **22**(1).
2. Chapelle, O., P. Haffner, and V. Vapnik, *Support Vector Machine for Histogram Based Image Classification.* IEEE Transaction on Neutral Network, 1999. **9**.
3. Corporation, C., *Corel Clipart & Photos*. 1999.
4. Goh, K.-S., E. Chang, and K.-T. Cheng. *Svm Binary Classifier Ensembles for Image Classification*. in Proceedings of the tenth international conference on Information and knowledge management. 2001.
5. Huang, J., et al. *Image Indexing Using Color Correlograms*. in Proccedings of IEEE Computer Vision and Pattern Recognition Conference. 1997.
6. Jaakkola, T. and D. Haussler. *Exploiting Generative Models in Discriminative Classifiers*. in Advance in Neutral Information Processing System 11. 1998.
7. Joachims, T. *Transductive Inference for Text Classification Using Support Vector Machines*. in Proceedings of The Sixteenth International Conference on Machine Learning (ICML 99). 1999.
8. Ng, A. and M. Jordan. *On Discriminative Vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes*. in Advances in Neural Information Processing Systems 14. 2002.
9. Seeger, M. *Learning with Labeled and Unlabeled Data.* Technical report Edinburgh University, 2001
10. Shewchuk, J. *An Introduction to the Conjugate Gradient Method without the Agonizing Pain.* Techinical Report School of Computer Science, Carnegie Mellon Unversity, 1994
11. Teytaud, O. and D. Sarrut. *Kernel Based Image Classification*. in Proceedings of International Conference on Artificial Neural Networks. 2001.