Extra-label information: experiments with view-based classification

A. Rakhlin, G. Yeo, and T. Poggio
Center for Biological and Computational Learning, MIT, Cambridge, USA

Abstract. Extra information is often readily available but not utilized in a classification paradigm. Here we explore using extra labels (profile faces and rotated faces) to aid in distinguishing faces versus non-faces. We propose a way to combine simple discriminant classifiers to build a more complex ones and justify the combination in a probabilistic setting.

1 Introduction

We would like to learn a distinction between two classes, C and D, given samples S_C and S_D from these classes. Now, assume additionally that we know that samples from class C have extra labels either C_1 or C_2. In other words, we have some additional information about the samples from one of the classes during training. At testing, this labeling is not given. The question is, when and how can this additional information be used to better learn a distinction between classes C and D.

An example of such a problem is face detection. We would like to learn to distinguish an image of a human face against a non-face. There is a lot of variation in face images, and the largest variation is coming from the rotation of the head. Instead of training on images of profile and frontal faces together, we would like to exploit additional information of whether a given training face is a frontal view or a profile view. This additional label is given to us for each training point. The question is, how can this information help us distinguish faces versus non-faces?

Prior work on view-based *recognition* includes work by Moghaddam et al. [2] on view-based eigenspace methods. In this work M eigenspaces were constructed, corresponding to M different views. Interpolating between these discrete views provided interesting results.

Schneiderman [5] claimed that a good view-invariant face classifier can be constructed from just two classifiers: one trained to recognize frontal faces, and one trained to recognize profile faces (mirror images can be used for the other frontal view thus giving a span of 180 degrees.) For cars Schneiderman claims to achieve good view-invariant performance with just 8 detectors (similarly taking advantage of the symmetry.) This is an example of the extra-label problem posed above. Mohhaddam's results and the interesting claim made by Schneiderman provided the motivation for the experiments in this paper.

2 Experiments

We can look at this problem from various view-points: clustering, classification, information-theory, and density estimation. In this paper we do not attempt to answer all the questions posed above, but rather to look at simple examples and get some intuition about the problem.

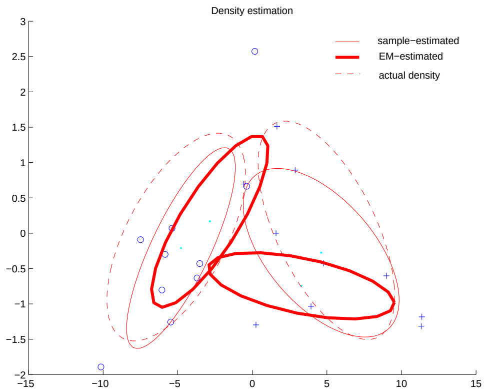

Figure 1: Density estimation with EM

Assume that examples from class C_1, C_2, and D are generated from unknown probability distributions p_{C_1}, p_{C_2}, p_D. Distributions will depend on the representation of objects (e.g. for images – pixel values, wavelet coefficients, simple object parts, etc). We would like the extra label to correspond to different clusters of C and D. So, the aim might be to choose a representation such that the three distributions are as different as possible. For example, if the divergence $D(p_{C_1}||p_{C_2})$ is small, then the distributions are similar and the additional information about the label C_1 vs C_2 is unlikely to be of any help. We should aim to maximize the difference between p_{C_1} and p_{C_2} by our choice of different representations. Approximating divergence is often not feasible, so other measures of how well the labels C_1 and C_2 correspond to clusters in space can be used. For example, we can compute a statistic based on the ratio of inter vs. intra-class distances in our samples as compared to a random labeling to find out how significant the given labels are.

The inverse problem is also interesting, but will not be considered in this paper: given the data (e.g. images of faces), first find the labels by clustering and by estimating the number of clusters using, for example, Gap statistic (see [7]). Then use this additional information to aid in making better classifiers.

2.1 Density estimation: toy problem

We now give a simple example to gain some geometric intuition. Consider a two-dimensional dataset C generated from two 2D Gaussian densities p_{C_1} and p_{C_2}. They have the same shape, but oriented at an angle to each other (see figure 1). The actual contours of the two Gaussians are the dashed ellipses. Now, we take a sample S_{C_1} and S_{C_2}, 10 points each. Assume that we know that there are two different clusters C_1 and C_2, but we are only given the combined set $C = C_1 + C_2$. If we run EM to estimate this mixture of Gaussians, we converge on a locally optimal solution (thick ellipses in figure 1). This is a well-known property of EM. If, on the other hand, we are given C_1 and C_2 as separate sets of samples (i.e. we are given labels on C), then we can better estimate p_C by estimating p_{C_1} and p_{C_2}. Although we have fewer points (this will be discussed later) in each of C_1 and C_2 separately than in $C = C_1 + C_2$, estimating each of the Gaussians using ML (thin ellipses in figure 1) gives better result than EM. Thus, for this particular setting, extra label helps density estimation. We can now easily imagine that a distribution of the other class D can be "around" these two clusters (e.g. non-faces in high-dimensional space around the clusters of frontal and profile faces), and therefore the

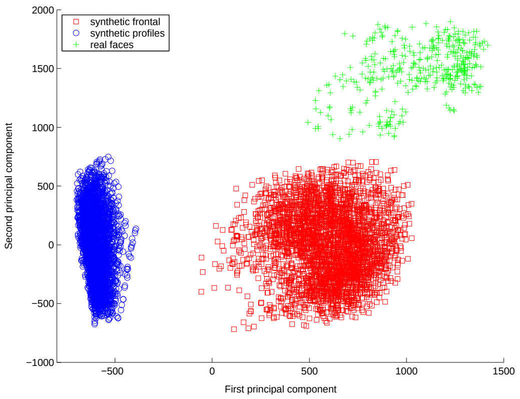

Figure 2: Plot of the first two PCA components. PCA was performed on synthetic images (frontal and profiles) and then all three datasets were projected to the PCA space.

extra label will lead to better density estimation and better generalization performance.

Note that the above example does not make any statements about other algorithms or other situations, but rather shows a setting where extra label does help. The result is also a consequence of particular behavior of EM for density estimation. Similar experiments can be carried out in high-dimensional spaces of actual images.

2.2 Faces: geometry of the problem

Consider a more concrete problem. We took a set C_1 of synthetic images of frontal faces, a set C_2 of synthetic images of profile faces [1], and a set C_1' of real faces, which contains frontal views with some rotation [2]. The size of images was 22x18, and our datasets were 3240, 3240, and 400 images, respectively.

The question that we asked is the following: do labels "frontal" and "rotated" correspond to different clusters in the high-dimensional space of image grayscale values? In other words, are the two distributions p_{C_1} and p_{C_2} different and generate well-separated clusters? If so, it is natural that the extra label information can be exploited to our advantage when classifying faces vs non-faces.

We found that the two clusters are indeed well-separated for the synthetic images. This is not too surprising because the synthetic images do not have much variability within each cluster C_1, C_2. Nonetheless, what's important is that there is a significant variability between two classes. PCA analysis of the two synthetic datasets C_1 and C_2 revealed that the clouds are well separated (figure 2). We then mapped the set of real images C_1' to the PCA space and found that its distribution differs from both of the synthetic distributions p_{C_1} and p_{C_2} along the second principal component, and that both C_1 and C_1' are well-separated from C_2 along the first principal component. In other words, the first principal component (which is responsible for almost all variation) makes a clear distinction between frontal and profile faces.

[1] Acknowledgments to Bernd Heisele for providing the synthetic images.

[2] ORL database, AT&T Laboratories, Cambridge.

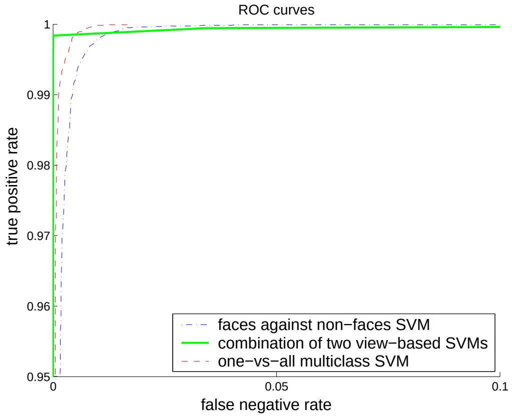

Figure 3: ROC curves for view-based classification

2.3 View-based classification

Next, we superimposed the face images (from the previous section) on backgrounds that do not contain faces, keeping the faces aligned in the center. Building on the intuition above, we trained three linear SVMs: f_1: C_1 vs D, f_2: C_2 vs D, and f_3: $C_1 + C_2$ vs D. We tested the generalization ability of each of the three SVMs on a test set containing unseen samples from C_1, C_2, and D. SVMs trained on one view of the face performed badly on the other view. Is it possible to combine classifiers f_1 and f_2 for separate views to get a better performance than with f_3? We calculated the Mahalanobis distance from each test point x to estimates of p_{C_1} and p_{C_2} based on the training examples. Output of each classifier was then weighted by the inverse of this distance:

$$f(x) = f_1(x) \cdot \frac{1}{d_{C_1}(x)} + f_2(x) \cdot \frac{1}{d_{C_2}(x)} \tag{1}$$

This combination of classifiers gave *better* results than other classification schemes we considered: 1) single linear SVM f_3 trained with no extra-label information, 2) classification based on thresholding Mahalanobis distance $d_{C_1}(x)$ and $d_{C_2}(x)$, and 3) one-vs-all SVM for three classes C_1, C_2, and D.

Motivation for the combination of classifiers above is the following. For simplicity, assume Gaussian distributions for the two subclasses C_1 and C_2. Equation (1) can be interpreted in terms of probabilities [3]. Mahalanobis distance d_{C_i} is directly related to the probability of point x belonging to class C_i:

$$p(x|C_i) = const \cdot e^{-\frac{1}{2} d_{C_i}(x)} \tag{2}$$

The main idea is to weight classifier $f_i(x)$ by the confidence that x belongs to class C_i:

$$f(x) \quad = \quad f_1(x) \cdot p(C_1|x) + f_2(x) \cdot p(C_2|x) \tag{3}$$

Note that $p(C_i|x) \propto p(x|C_i) \cdot \frac{1}{p(x)}$, so we get

$$f(x) \quad = \quad const_1 \cdot f_1(x) \cdot e^{-\frac{1}{2} \cdot d_{C_1}(x)} + const_2 \cdot f_2(x) \cdot e^{-\frac{1}{2} \cdot d_{C_2}(x)} \tag{4}$$

[3] see [4, 3] for probabilistic interpretation of SVM outputs.

The above combination gave similar results as equation (1) (with appropriate constants). At a closer inspection, we see that the inverse of the Mahalanobis distance in equation (1) roughly approximates the tail of the exponent in equation (4) (again, with appropriate constants) and in practice is much easier to compute.

This argument shows that the combination of classifiers in equation (1) can be motivated in a probabilistic framework. In general, we propose that a combination of n classifiers can be combined by adding the outputs of SVMs weighted by the inverse of the Mahalanobis distance to the corresponding clusters.

3 Discussion

We considered simple problems where extra-label information helps classification and density estimation. We also proposed a way to combine simple classifiers to build a more complex one.

One issue we have not considered so far is that the number of points for each extra label is smaller than the whole training set we are given. Therefore, fewer points are available for density estimation or classifier training for each separate label. We have a trade-off between having more extra information (many small clusters) versus having more training data (few larger clusters). As a limit, we can have a separate label for each training sample, and we will overfit. The problem is similar to splitting a given dataset for bagging (see [1]): each classifier is trained on a smaller number of points, but the *combination* might give better generalization results. The issue is also discussed in the stability framework in [6].

Using extra-label information is plausible from the biological point of view. For example, people visually recognize simple parts and combine them in some sophisticated way to form complex object representations. How to effectively combine simple classifiers in a large hierarchical structure is still an open question.

Acknowledgments

We would like to thank Sayan Mukherjee for helpful discussions.

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[2] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. In S. Nayar and T. Poggio, editors, *Early Visual Learning, Oxford University Press, 1996*, chapter 5, pages 99–130. Oxford University Press, 1996.

[3] S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J.P. Mesirov, and T. Poggio. Support vector machine classification of microarray data. AI Memo 1677, Massachusetts Institute of Technology, 2000.

[4] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, 1999.

[5] Henry Schneiderman. *A Statistical Approach to 3D Object Detection Applied to Faces and Cars*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.

[6] S. Mukherjee T. Poggio, R. Rifkin and A. Rakhlin. Bagging regularizes. AI Memo CBCL-214, Massachusetts Institute of Technology, 2002.

[7] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic, 2000.