Principled Hybrids of Generative and Discriminative Models

Julia A. Lasserre
University of Cambridge
Cambridge, UK
jal62@cam.ac.uk

Christopher M. Bishop
Microsoft Research
Cambridge, UK
cmbishop@microsoft.com

Thomas P. Minka
Microsoft Research
Cambridge, UK
minka@microsoft.com

Abstract

When labelled training data is plentiful, discriminative techniques are widely used since they give excellent generalization performance. However, for large-scale applications such as object recognition, hand labelling of data is expensive, and there is much interest in semi-supervised techniques based on generative models in which the majority of the training data is unlabelled. Although the generalization performance of generative models can often be improved by 'training them discriminatively', they can then no longer make use of unlabelled data. In an attempt to gain the benefit of both generative and discriminative approaches, heuristic procedure have been proposed [2, 3] which interpolate between these two extremes by taking a convex combination of the generative and discriminative objective functions.

In this paper we adopt a new perspective which says that there is only one correct way to train a given model, and that a 'discriminatively trained' generative model is fundamentally a new model [7]. From this viewpoint, generative and discriminative models correspond to specific choices for the prior over parameters. As well as giving a principled interpretation of 'discriminative training', this approach opens door to very general ways of interpolating between generative and discriminative extremes through alternative choices of prior. We illustrate this framework using both synthetic data and a practical example in the domain of multi-class object recognition. Our results show that, when the supply of labelled training data is limited, the optimum performance corresponds to a balance between the purely generative and the purely discriminative.

1. Introduction

Machine learning techniques are now widely used in computer vision. In many applications the goal is to take a vector $\mathbf{x}$ of input features and to assign it to one of a number of alternative classes labelled by a vector $\mathbf{c}$ (for instance, if we have C classes, then $\mathbf{c}$ might be a C-dimensional binary vector in which all elements are zero except the one corresponding to the class). Throughout this paper we will have in mind the problem of object recognition, in which $\mathbf{x}$ corresponds to an image (or a region within an image) and $\mathbf{c}$ represents the categories of object (or objects) present in the image, although the techniques and conclusions presented are much more widely applicable.

In the simplest scenario, we are given a training data set comprising N images $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ together with corresponding labels $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$, in which we assume that the images, and their labels, are drawn independently from the same fixed distribution. Our goal is to predict the class $\widehat{\mathbf{c}}$ for a new input vector $\widehat{\mathbf{x}}$, and so we require the conditional distribution

$$p(\widehat{\mathbf{c}}|\widehat{\mathbf{x}}, \mathbf{X}, \mathbf{C}). \tag{1}$$

To determine this distribution we introduce a parametric model governed by a set of parameters $\boldsymbol{\theta}$. In a *discriminative* approach we define the conditional distribution $p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the model. The likelihood function is then given by

$$L(\boldsymbol{\theta}) = p(\mathbf{C}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta}). \tag{2}$$

The likelihood function can be combined with a prior $p(\boldsymbol{\theta})$, to give a joint distribution

$$p(\boldsymbol{\theta}, \mathbf{C}|\mathbf{X}) = p(\boldsymbol{\theta})L(\boldsymbol{\theta}) \tag{3}$$

from which we can obtain the posterior distribution by normalizing

$$p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C}) = \frac{p(\boldsymbol{\theta})L(\boldsymbol{\theta})}{p(\mathbf{C}|\mathbf{X})} \tag{4}$$

where

$$p(\mathbf{C}|\mathbf{X}) = \int p(\boldsymbol{\theta})L(\boldsymbol{\theta})\, d\boldsymbol{\theta}. \tag{5}$$

Predictions for new inputs are then made by marginalizing the predictive distribution with respect to $\boldsymbol{\theta}$ weighted by the posterior distribution

$$p(\widehat{\mathbf{c}}|\widehat{\mathbf{x}}, \mathbf{X}, \mathbf{C}) = \int p(\widehat{\mathbf{c}}|\widehat{\mathbf{x}}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C})\, d\boldsymbol{\theta}. \tag{6}$$

In practice this marginalization, as well as the normalization in (5), are rarely tractable and so approximation, schemes such as variational inference, must be used. If training data is plentiful a point estimate for $\boldsymbol{\theta}$ can be made by maximizing the posterior distribution to give $\boldsymbol{\theta}_{\mathrm{MAP}}$, and the predictive distribution then estimated using

$$p(\widehat{\mathbf{c}}|\widehat{\mathbf{x}}, \mathbf{X}, \mathbf{C}) \simeq p(\widehat{\mathbf{c}}|\widehat{\mathbf{x}}, \boldsymbol{\theta}_{\mathrm{MAP}}). \tag{7}$$

Note that maximizing the posterior distribution (4) is equivalent to maximizing the joint distribution (3) since these differ only by a multiplicative constant. In practice, we typically take the logarithm before maximizing as this gives rise to both analytical and numerical simplifications. If we consider a prior distribution $p(\boldsymbol{\theta})$ which is constant over the region in which the likelihood function is large, then maximizing the posterior distribution is equivalent to maximizing the likelihood. In all cases, however, the key quantity for model training is the likelihood function $L(\boldsymbol{\theta})$. Discriminative methods give good predictive performance and have been widely used in many applications.

In recent years there has been growing interest in a complementary approach based on *generative* models, which define a joint distribution $p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})$ over both input vectors and class labels [4]. One of the motivations is that in complex problems such as object recognition, where there is huge variability in the range of possible input vectors, it may be difficult or impossible to provide enough labelled training examples, and so there is increasing use of semi-supervised learning in which the labelled training examples are augmented with a much larger quantity of unlabelled examples. A discriminative model cannot make use of the unlabelled data, as we shall see, and so in this case we need to consider a generative approach.

The complementary properties of generative and discriminative models have led a number of authors to seek methods which combine their strengths. In particular, there has been much interest in 'discriminative training' of generative models [2, 3, 12] with a view to improving classification accuracy. This approach has been widely used in speech recognition with great success [5] where generative hidden Markov models are trained by optimizing the predictive conditional distribution. As we shall see later, this form of training can lead to improved performance by compensating for model mis-specification, that is differences between the true distribution of the process which generates the data, and the distribution specified by the model. However, as we have noted, discriminative training cannot take advantage of unlabelled data. In particular, Ng et al. [8] show that logistic regression (the discriminative counterpart of a Naive Bayes generative model) works better than its generative counterpart, but only for a large number of training datapoints (large depending on the complexity of the problem), which confirms the need for using unlabelled data.

Recently several authors [2, 3] have proposed hybrids of the generative and discriminative approaches in which a model is trained by optimizing a convex combination of the generative and discriminative log likelihood functions. Although the motivation for this procedure was heuristic, it was sometimes found that the best predictive performance was obtained for intermediate regimes in between the discriminative and generative limits.

In this paper we develop a novel viewpoint [7] which says that, for a given model, there is a unique likelihood function and hence there is only one correct way to train it. The 'discriminative training' of a generative model is instead interpreted in terms of standard training of a different model, corresponding to a different choice of distribution. This removes the apparently ad-hoc choice for the training criterion, so that all models are trained according to the principles of statistical inference. Furthermore, by introducing a constraint between the parameters of this model, through the choice of prior, the original generative model can be recovered.

As well as giving a novel interpretation for 'discriminative training' of generative models, this viewpoint opens the door to principled blending of generative and discriminative approaches by introducing priors having a soft constraint amongst the parameters. The strength of this constraint therefore governs the balance between generative and discriminative.

In Section 2 we give a detailed discussion of the new interpretation of discriminative training for generative models, and in Section 3 we illustrate the advantages of blending between generative and discriminative viewpoints using a synthetic example in which the role of unlabelled data and of model mis-specification becomes clear. In Section 4 we show that this approach can be applied to a large scale problem in computer vision concerned with object recognition in images, and finally we draw some conclusions in Section 5.

2. A New View of 'Discriminative Training'

A parametric generative model is defined by specifying the joint distribution $p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})$ of the input vector $\mathbf{x}$ and the class label $\mathbf{c}$, conditioned on a set of parameters $\boldsymbol{\theta}$. Typically this is done by defining a prior probability for the classes $p(\mathbf{c}|\boldsymbol{\pi})$ along with a class-conditional density for each class $p(\mathbf{x}|\mathbf{c}, \boldsymbol{\lambda})$, so that

$$p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta}) = p(\mathbf{c}|\boldsymbol{\pi})p(\mathbf{x}|\mathbf{c}, \boldsymbol{\lambda}) \tag{8}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\lambda}\}$. Since the data points are assumed to be independent, the joint distribution is given by

$$L_{\mathrm{G}}(\boldsymbol{\theta}) = p(\mathbf{X}, \mathbf{C}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{c}_n|\boldsymbol{\theta}). \tag{9}$$

This can be maximized to determine the most probable (MAP) value of $\boldsymbol{\theta}$. Again, since $p(\mathbf{X}, \mathbf{C}, \boldsymbol{\theta}) =$

$p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C})p(\mathbf{X}, \mathbf{C})$, this is equivalent to maximizing the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{C})$.

In order to improve the predictive performance of generative models it has been proposed to use 'discriminative training' [12] which involves maximizing

$$L_{\mathrm{D}}(\boldsymbol{\theta}) = p(\mathbf{C}, \boldsymbol{\theta}|\mathbf{X}) = p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta}) \tag{10}$$

in which we are conditioning on the input vectors instead of modelling their distribution. Here we have used

$$p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})}{\sum_{\mathbf{c}'} p(\mathbf{x}, \mathbf{c}'|\boldsymbol{\theta})}. \tag{11}$$

Note that (10) is not the joint distribution for the original model defined by (9), and so does not correspond to MAP for this model. The terminology of 'discriminative training' is therefore misleading, since for a given model there is only one correct way to train it. It is not the training method which has changed, but the model itself.

This concept of discriminative training has been taken a stage further [2, 3] by maximizing a function given by a convex combination of (9) and (10) of the form

$$\alpha \ln L_{\mathrm{D}}(\boldsymbol{\theta}) + (1 - \alpha) \ln L_{\mathrm{G}}(\boldsymbol{\theta}) \tag{12}$$

where $0 \leqslant \alpha \leqslant 1$, so as to interpolate between generative ($\alpha = 0$) and discriminative ($\alpha = 1$) approaches. Unfortunately, this criterion was not derived by maximizing the distribution of a well-defined model.

Following [7] we therefore propose an alternative view of discriminative training, which will lead to an elegant framework for blending generative and discriminative approaches. Consider a model which contains an additional independent set of parameters $\widetilde{\boldsymbol{\theta}} = \{\widetilde{\boldsymbol{\pi}}, \widetilde{\boldsymbol{\lambda}}\}$ in addition to the parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\lambda}\}$, in which the likelihood function is given by

$$q(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\widetilde{\boldsymbol{\theta}}) \tag{13}$$

where

$$p(\mathbf{x}|\widetilde{\boldsymbol{\theta}}) = \sum_{\mathbf{c}'} p(\mathbf{x}, \mathbf{c}'|\widetilde{\boldsymbol{\theta}}). \tag{14}$$

Here $p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta})$ is defined by (11), while $p(\mathbf{x}, \mathbf{c}|\widetilde{\boldsymbol{\theta}})$ has independent parameters $\widetilde{\boldsymbol{\theta}}$.

The model is completed by defining a prior $p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}})$ over the model parameters, giving a joint distribution of the form

$$q(\mathbf{X}, \mathbf{C}, \boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) \prod_{n=1}^{N} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta}) p(\mathbf{x}_n|\widetilde{\boldsymbol{\theta}}). \tag{15}$$

Now suppose we consider a special case in which the prior factorizes, so that

$$p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta}) p(\widetilde{\boldsymbol{\theta}}). \tag{16}$$

We then determine optimal values for the parameters $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ in the usual way by maximizing (15), which now takes the form

$$\left[p(\boldsymbol{\theta}) \prod_{n=1}^{N} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta}) \right] \left[p(\widetilde{\boldsymbol{\theta}}) \prod_{n=1}^{N} p(\mathbf{x}_n|\widetilde{\boldsymbol{\theta}}) \right]. \tag{17}$$

We see that the resulting value of $\boldsymbol{\theta}$ will be identical to that found by maximizing (11), since it is the same function which is being maximized. Since it is $\boldsymbol{\theta}$ and not $\widetilde{\boldsymbol{\theta}}$ which determines the predictive distribution $p(\mathbf{c}|\mathbf{x}, \boldsymbol{\theta})$ we see that this model is equivalent in its predictions to the 'discriminatively trained' generative model. This gives a consistent view of training in which we always maximize the joint distribution, and the distinction between generative and discriminative training lies in the choice of model.

The relationship between the generative model and the discriminative model is illustrated using directed graphs in Figure 1.

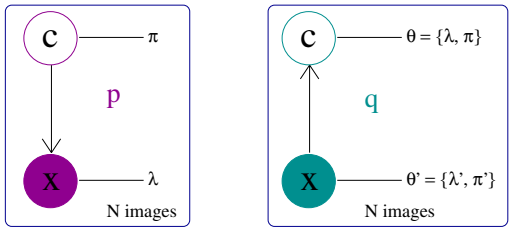

Figure 1. Probabilistic directed graphs, showing on the left, the original generative model, and on the right the corresponding discriminative model.

Now suppose instead that we consider a prior which enforces equality between the two sets of parameters

$$p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta}) \delta(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}). \tag{18}$$

Then we can set $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$ in (13) from which we recover the original generative model $p(\mathbf{x}, \mathbf{c}|\boldsymbol{\theta})$. Thus we have a single class of distributions in which the discriminative model corresponds to independence in the prior, and the generative model corresponds to an equality constraint in the prior.

2.1. Blending Generative and Discriminative

Clearly we can now blend between the generative and discriminative extremes by considering priors which impose a soft constraint between $\widetilde{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$. Why should we wish to do this?

First of all, we note that the reason why 'discriminative training' might give better results than direct use of the generative model, is that (15) is more flexible than (9) since it relaxes the implicit constraint $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$. Of course, if the generative model were a perfect representation of reality (in other words the data really came from the model) then increasing the flexibility of the model would lead to poorer

results. Any improvement from the discriminative approach must therefore be the result of a mis-match between the model and the true distribution of the (process which generates the) data. In other words, the benefit of 'discriminative training' is dependent on model mis-specification.

Conversely, the benefit of the generative approach is that it can make use of unlabelled data to augment the labelled training set. Suppose we have a data set comprising a set of inputs $\mathbf{X}_{\mathrm{L}}$ for which we have corresponding labels $\mathbf{C}_{\mathrm{L}}$, together with a set of inputs $\mathbf{X}_{\mathrm{U}}$ for which we have no labels. For the correctly trained generative model, the function which is maximized is given by

$$p(\boldsymbol{\theta}) \prod_{n \in \mathrm{L}} p(\mathbf{x}_n, \mathbf{c}_n|\boldsymbol{\theta}) \prod_{m \in \mathrm{U}} p(\mathbf{x}_m|\boldsymbol{\theta}) \tag{19}$$

where $p(\mathbf{x}|\boldsymbol{\theta})$ is defined by

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{c}'} p(\mathbf{x}, \mathbf{c}'|\boldsymbol{\theta}). \tag{20}$$

We see that the unlabelled data influences the choice of $\boldsymbol{\theta}$ and hence affects the predictions of the model. By contrast, for the 'discriminatively trained' generative model the function which is now optimized is again the product of the prior and the likelihood function and so takes the form

$$p(\boldsymbol{\theta}) \prod_{n \in \mathrm{L}} p(\mathbf{x}_c|\mathbf{x}_n, \boldsymbol{\theta}) \tag{21}$$

and we see that the unlabelled data plays no role. Thus, in order to make use of unlabelled data we cannot use a discriminative approach.

Now let us consider how a combination of labelled and unlabelled data can be exploited from the perspective of our new approach defined by (15), for which the joint distribution becomes

$$q(\mathbf{X}_{\mathrm{L}}, \mathbf{C}_{\mathrm{L}}, \mathbf{X}_{\mathrm{U}}, \boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) \left[\prod_{n \in \mathrm{L}} p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta}) p(\mathbf{x}_n|\widetilde{\boldsymbol{\theta}})\right] \left[\prod_{m \in \mathrm{U}} p(\mathbf{x}_m|\widetilde{\boldsymbol{\theta}})\right] \tag{22}$$

We see that the unlabelled data (as well as the labelled data) influences the parameters $\widetilde{\boldsymbol{\theta}}$ which in turn influence $\boldsymbol{\theta}$ via the soft constraint imposed by the prior.

In general, if the model is not a perfect representation of reality, and if we have unlabelled data available, then we would expect the optimal balance to lie neither at the purely generative extreme nor at the purely discriminative extreme.

As a simple example of a prior which interpolates smoothly between the generative and discriminative limits, consider the class of priors of the form

$$p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) \propto p(\boldsymbol{\theta}) p(\widetilde{\boldsymbol{\theta}}) \frac{1}{\sigma} \exp\left\{-\frac{1}{2\sigma^2} \|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\|^2.\right\} \tag{23}$$

If desired, we can relate σ to an α like parameter by defining a map from $(0, 1)$ to $(0, \infty)$, for example using

$$\sigma(\alpha) = \left(\frac{\alpha}{1 - \alpha}\right)^2. \tag{24}$$

For $\alpha \to 0$ we have $\sigma \to 0$, and we obtain a hard constraint of the form (18) which corresponds to the generative model. Conversely for $\alpha \to 1$ we have $\sigma \to \infty$ and we obtain an independence prior of the form (16) which corresponds to the discriminative model.

3. Illustration

We now illustrate the new framework for blending between generative and discriminative approaches using an example based on synthetic data. This is chosen to be as simple as possible, and so involves data vectors $\mathbf{x}_n$ which live in a two-dimensional Euclidean space for easy visualization, and which belong to one of two classes. Data from each class is generated from a Gaussian distribution as illustrated in Figure 2. Here the scales on the axes are equal, and

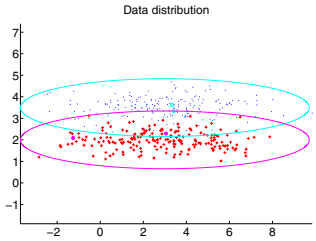

Figure 2. Synthetic training data, shown as red crosses and blue dots, together with contours of probability density for each of the two classes. Two points from each class are labelled (indicated by circles around the data points).

so we see that the class-conditional densities are elongated in the horizontal direction.

We now consider a continuum of models which interpolate between purely generative and purely discriminative. To define this model we consider the generative limit, and represent each class-conditional density using an isotropic Gaussian distribution. Since this does not capture the horizontally elongated nature of the true class distributions, this represents a form of model mis-specification. The parameters of the model are the means and variances of the Gaussians for each class, along with the class prior probabilities.

We consider a prior of the form (23) in which $\sigma(\alpha)$ is defined by (24). Here we choose $p(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}|\alpha) = p(\boldsymbol{\theta})\, N(\widetilde{\boldsymbol{\theta}}|\boldsymbol{\theta}, \sigma(\alpha))$, where $p(\boldsymbol{\theta})$ are the usual congugate priors (a gaussian prior for the means, a gamma prior for the variances, and a dirichlet for the class priors). This results in a proper prior.

The training data set comprises 200 points from each class, of which just 2 from each class are labelled, and the

test set comprises 200 points all of which are labelled. Experiments are run 10 times with differing random initializations and the results used to computer a mean and variance over the test set classification, which are shown by 'error bars' in Figure 3. We see that the best generalization occurs

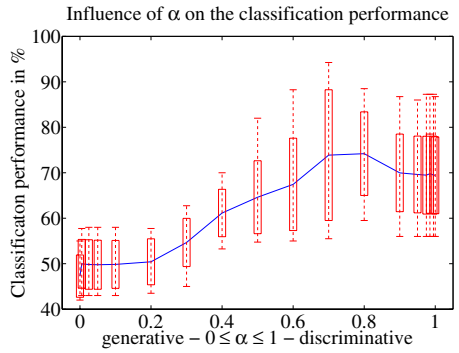

Figure 3. Plot of the percentage of correctly classified points on the test set versus α for the synthetic data problem.

for values of α intermediate between the generative and discriminative extremes.

To gain insight into this behaviour we can plot the contours of density for each class corresponding to different values of α, as shown in Figure 4. We see that a purely gen-

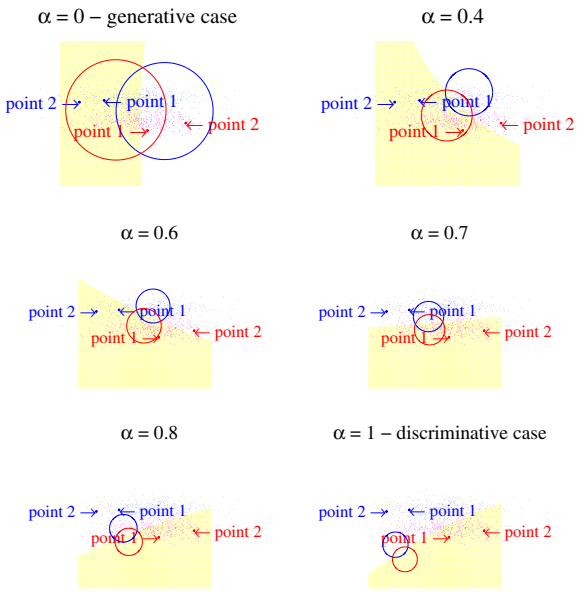

Figure 4. Results of fitting an isotropic Gaussian model to the synthetic data for various values of α. The top left shows $\alpha = 0$ (generative case) while the bottom right shows $\alpha = 1$ (discriminative case). The yellow area corresponds to points that are assigned to the red class.

erative model is strongly influenced by modelling the density of the data and so gives a decision boundary which is orthogonal to the correct one. Conversely a purely discriminative model attends only to the labelled data points and so misses useful information about the horizontal elongation of the true class-conditional densities which is present in the unlabelled data.

4. Object Recognition

We now apply our approach to a realistic application involving object recognition in static images. This is a widely studied problem which has been tackled using a range of different discriminative and generative models. The long term goal of such research is to achieve near human levels of recognition accuracy across thousands of object classes in the presence of wide variations in location, scale, orientation and lighting, as well as changes due to intra-class variability and occlusion.

4.1. The data

We used 8 different classes: airplanes, bikes, cows, faces, horses, leaves, motorbikes, sheep. The cows and sheep images come from Microsoft Research (http://www.research.microsoft.com/mlp), the airplanes, faces, leaves and motorbikes images come from the Caltech database (http://www.robots.ox.ac.uk/∼vgg/data), the bikes images were downloaded from the Technical University of Graz (http://www.emt.tugraz.at/∼pinz/data), and the horses images were downloaded from the Mathematical Sciences Research Institute (http://www.msri.org/people/members/eranb). Together these images exhibit a wide variety of poses, colours, and illumination, as illustrated by the sample images shown in Figure 5.

Each image contains one or more objects from a particular class, and the goal is to build a true multi-class classifier in which each image is assigned to one of the classes (rather than simply classifying each class separately versus the rest, which would be a much simpler problem).

All images were re-scaled to 300×200, and raw patches of size 48×48 were extracted on a regular grid of size 24×24 (i.e. every 24th pixel).

4.2. The features

Our features are taken from [11], in which the original RGB images are first converted into the CIE (L, a, b) colour space [6]. Each image is then convolved with 17 filters, and the set of corresponding pixels from each of the filtered images represents a 17-dimensional vector. All these feature vectors are clustered using K-means with $K = 100$. Since this large value of K is computationally costly in later stages of processing, PCA is used to give a 15-dimensional feature vector. Winn *et al.* [11] use a more powerful technique to reduce the number of features, but since this is a supervised method based on fully labelled training data, we

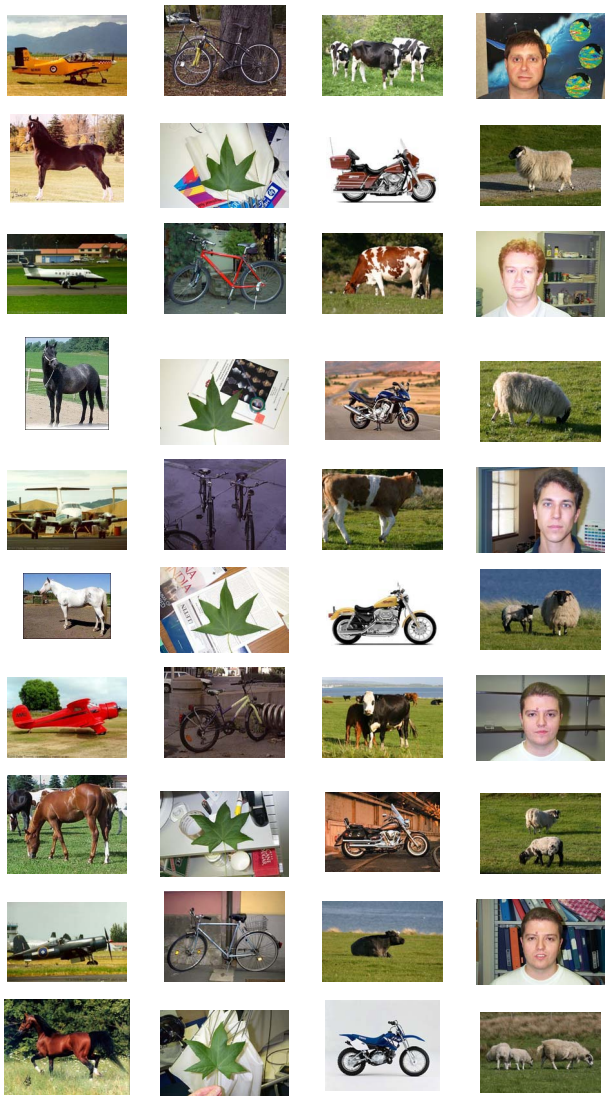

Figure 5. Sample images from the training set.

did not re-implement it here. The cluster centers obtained through K-means are called *textons* [10].

The filters are quite standard: the first three filters are obtained by scaling a Gaussian filter, and are applied to each channel of the colour image, which gives $3 \times 3 = 9$ response images. Then a Laplacian filter is applied to the L channel, at 4 different scales, which gives 4 more response images. Finally 2 DoG (difference of Gaussians) filters (one along each direction) are applied to the L channel, at 2 different scales, giving another 4 responses.

From these response images, we extract every pixel on a 4×4 grid, and apply K-means to obtain K textons. Now each patch will be represented by a histogram of these textons, i.e. by a K-dimensional vector containing the proportion of each texton. Textons were obtained from 25 training images per class (half of the training set). Note that the texton features are found using only unlabelled data. These vectors are then reduced using PCA to a dimensionality of 15.

4.3. The model

We consider the generative model introduced in [9], which we now briefly describe. Each image is represented by a feature vector $\mathbf{x}_n$, where $n = 1, \ldots, N$, and N is the total number of images. Each vector comprises a set of J patch vectors $\mathbf{x} = \{\mathbf{x}_{nj}\}$ where $j = 1, \ldots, J$. We assume that each patch belongs to one and only one of the classes, or to a separate 'background' class, so that each patch can be characterized by a binary vector $\boldsymbol{\tau}_{nj}$ coded so that all elements of $\boldsymbol{\tau}_{nj}$ are zero except the element corresponding to the class. We use $\mathbf{c}_n$ to denote the image label vector for image n with independent components $c_{nk} \in \{0, 1\}$ in which $k = 1, \ldots C$ labels the class.

The overall joint distribution for the model can be represented as a directed graph, as shown in Figure 6. We

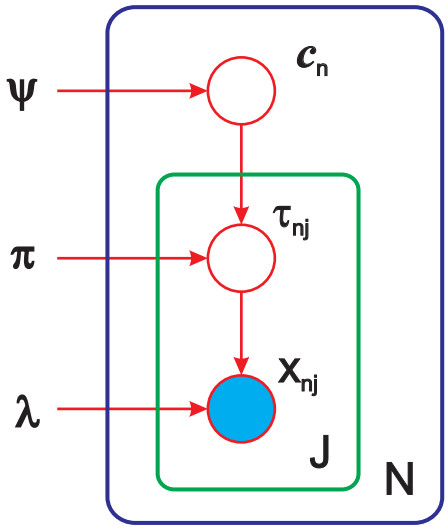

Figure 6. The generative model for object recognition expressed as a directed acyclic graph, for unlabelled images, in which the boxes denote 'plates' (i.e. independent replicated copies). Only the patch feature vectors $\{\mathbf{x}_{nj}\}$ are observed, corresponding to the shaded node. The image class labels $\mathbf{c}_n$ and patch class labels $\boldsymbol{\tau}_{nj}$ are latent variables.

can therefore characterize the model completely in terms of the conditional probabilities $p(\mathbf{c})$, $p(\boldsymbol{\tau}|\mathbf{c})$ and $p(\mathbf{x}|\boldsymbol{\tau})$. This model is most easily explained generatively, that is, we describe the procedure for generating a set of observed feature vectors from the model.

First we choose the overall class of the image according to some prior probability parameters ψ_k where $k = 1, \ldots, C$, and $0 \leqslant \psi_k \leqslant 1$, with $\sum_k \psi_k = 1$, so that

$$p(\mathbf{c}) = \prod_{k=1}^{C} \psi_k^{c_k}. \tag{25}$$

Given the overall class for the image, each patch is then drawn from either one of the foreground classes or the background ($k = C + 1$) class. The probability of generating a patch from a particular class is governed by a set of parameters π_k, one for each class, such that $\pi_k \geqslant 0$, constrained by the subset of classes actually present in the image. Thus

$$p(\boldsymbol{\tau}_j|\mathbf{c}) = \left(\sum_{l=1}^{C+1} c_l \pi_l\right)^{-1} \prod_{k=1}^{C+1} (c_k \pi_k)^{\tau_{jk}}. \quad (26)$$

Note that there is an overall undetermined scale to these parameters, which may be removed by fixing one of them, e.g. $\pi_{C+1} = 1$.

For each class, the distribution of the patch feature vector $\mathbf{x}$ is governed by a separate mixture of Gaussians which we denote by

$$p(\mathbf{x}|\boldsymbol{\tau}_j) = \prod_{k=1}^{C+1} \phi_k(\mathbf{x}_j; \boldsymbol{\lambda}_k)^{\tau_{jk}} \quad (27)$$

where $\boldsymbol{\lambda}_k$ denotes the set of parameters (means, covariances and mixing coefficients) associated with this mixture model.

If we assume N independent images, and for image n we have J patches drawn independently, then the joint distribution of all random variables is

$$\prod_{n=1}^{N} \left[p(\mathbf{c}_n) \prod_{j=1}^{J} p(\mathbf{x}_{nj}|\boldsymbol{\tau}_{nj}) p(\boldsymbol{\tau}_{nj}|\mathbf{c}_n) \right]. \quad (28)$$

Here we are assuming that each image has the same number J of patches, though this restriction is easily relaxed if required.

The graph shown in Figure 6 corresponds to unlabelled images in which only the feature vectors $\{\mathbf{x}_{nj}\}$ are observed, with both the image category and the classes of each of the patches being latent variables. It is also possible to consider images which are 'weakly labelled', that is each image is labelled according to the category of object present in the image. This corresponds to the graphical model of Figure 7 in which the node $\mathbf{c}_n$ is shaded. Of course, for a given size of data set, better performance is expected if all of the images are 'strongly labelled', that is segmented images in which the region occupied by the object or objects is known so that the patch labels $\boldsymbol{\tau}_{nj}$ become observed variables. The graphical model for a set of strongly labelled images is also shown in Figure 7. Strong labelling requires hand segmentation of images, and so is a time consuming and expensive process as compared with collection of the images themselves. For a given level of effort it will always be possible to collect many unlabelled or weakly labelled images for the same cost as a single strongly labelled image. Since the variability of natural images and objects is so vast we will always be operating in a regime in which the size of our data sets is statistically small (though they will often be computationally large).

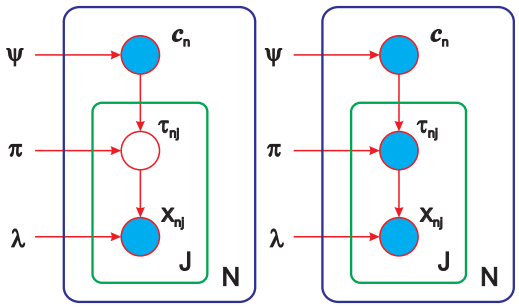

Figure 7. Graphical models corresponding to Figure 6 for weakly labelled images (left) and strongly labelled images (right).

For this reason there is great interest in augmenting expensive strongly labelled images with lots of cheap weakly labelled or unlabelled images in order to better characterize the different forms of variability. Although the two stage hierarchical model shown in Figure 6 appears to be more complicated than in the simple example shown in Figure 1, it does in fact fall within the same framework. In particular, for labelled images the observed data is $\{\mathbf{x}_n, \mathbf{c}_n, \boldsymbol{\tau}_{nj}\}$, while for 'unlabelled' images only $\{\mathbf{x}_n\}$ are observed. The experiments described here could readily be extended to consider arbitrary combinations of strongly labelled, weakly labelled and unlabelled images if desired.

If we let $\boldsymbol{\theta} = \{\psi_k, \pi_k, \boldsymbol{\lambda}_k\}$ denote the full set of parameters in the model, then we can consider a model of the form (22) in which the prior is given by (23) with $\sigma(\alpha)$ defined by (24), and the terms $p(\boldsymbol{\theta})$ and $p(\widetilde{\boldsymbol{\theta}})$ taken to be constant.

We use conjugate gradients to optimize the parameters. Due to lack of space we do not write down all the derivatives of the log likelihood function required by the conjugate gradient algorithm. However, the correctness of the mathematical derivation of these gradients, as well as their numerical implementation, can easily be verified by comparison against numerical differentiation [1]. The conjugate gradients is the most used technique when it comes to blending generative and discriminative models, thanks to its flexibility. Indeed, because of the discriminative component $p(\mathbf{c}_n|\mathbf{x}_n, \boldsymbol{\theta})$ which contains a normalising factor, an algorithm such as EM would require much more work, as nothing is directly tractable anymore. However, a comparison of the two methods is currently being investigated.

4.4. Results

We use 50 training images per class (giving 400 training images in total) of which 5 images per class (a total of 40) were fully labelled i.e. both the image and the individual patches have class labels. All the other images are left totally unlabelled, i.e. not even the category they belong to is given. Note that this kind of training data is 1) very cheap

to get and 2) very unusual for a discriminative model. The test set consists of 100 images per class (giving a total of 800 images), the task is to label each image.

Experiments are run 5 times with differing random initializations and the results used to compute a mean and variance over the test set classification, which are shown by 'error bars' in Figure 8. Note that, since there are 8 bal-

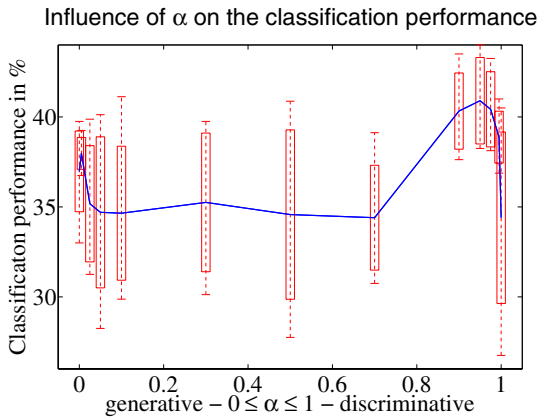

Figure 8. Influence of the term α on the test set classification performance.

anced classes, random guessing would give 12.5% correct on average. Again we see that the best performance is obtained with a blend between generative and discriminative extremes.

5. Conclusions

In this paper we have shown that 'discriminative training' for generative models can be re-cast in terms of standard training methods applied to a modified model. This new viewpoint opens the door to a wide range of new models which interpolate smoothly between generative and discriminative approaches and which can benefit from the advantages of both. The main drawback of this framework is that the number of parameters in the model is doubled leading to greater computational cost.

Although we have focussed on a specific application in computer vision concerned with object recognition, the techniques proposed here have very wide applicability.

A principled approach to combining generative and discriminative approaches not only gives a more satisfying foundation for the development of new models, but it also brings practical benefits. In particular, the parameter α which governs the trade-off between generative and discriminative is now a hyper-parameter within a well defined probabilistic model which is trained using the (unique) correct likelihood function. In a Bayesian setting the value of this hyper-parameter can therefore be optimized by maximizing the marginal likelihood in which the model parameters have been integrated out, thereby allowing the optimal trade-off between generative and discriminative limits to be determined entirely from the training data without recourse to cross-validation [1]. This extension of the work described here is currently being investigated.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. 7, 8

[2] G. Bouchard and B. Triggs. The trade-off between generative and discriminative classifiers. In *IASC 16th International Symposium on Computational Statistics*, pages 721–728, Prague, Czech Republic, august 2004. 1, 2, 3

[3] A. Holub and P. Perona. A discriminative framework for modelling object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego (California), USA, june 2005. IEEE Computer Society. 1, 2, 3

[4] T. Jebara. *Machine Learning: Discriminative and Generative*. Kluwer, 2004. 2

[5] S. Kapadia. *Discriminative Training of Hidden Markov Models*. Phd, University of Cambridge, Cambridge, U.K., March 1998. 2

[6] J. M. Kasson and W. Plouffe. An analysis of selected computer inter-change color spaces. *ACM Transactions on Graphics*, 11:373–405, 1992. 5

[7] T. Minka. Discriminative models, not discriminative training. Technical report, Microsoft Research, Cambridge, UK, 2005. 1, 2, 3

[8] A. Y. Ng and M. I. Jordan. On discriminative vs generative classifiers: A comparison of logistic regression and naive bayes. In *Neural Information Processing Systems*, pages 841–848, Vancouver, Canada, december 2001. MIT Press. 2

[9] I. Ulusoy and C. M. Bishop. Generative versus discriminative models for object recognition. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition, CVPR.*, San Diego, 2005. 6

[10] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. In *IJCV*, volume 62, pages 61–81, 2005. 6

[11] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *IEEE International Conference on Computer Vision*, Beijing, China, 2005. IEEE Computer Society. 5

[12] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *5th IEEE International Conference on Data Mining*, Houston (Texas), USA, november 2005. IEEE Computer Society. 2, 3