Comparison of Generative and Discriminative Techniques for Object Detection and Classification

Ilkay Ulusoy[1] and Christopher M. Bishop[2]

[1] METU, Computer Vision and Intelligent Systems Research Lab.
06531 Ankara, Turkey
http://www.eee.metu.edu.tr/∼ilkay
[2] Microsoft Research,
7 J J Thompson Avenue,
Cambridge, U.K.
http://research.microsoft.com/∼cmbishop

Abstract. Many approaches to object recognition are founded on probability theory, and can be broadly characterized as either generative or discriminative according to whether or not the distribution of the image features is modelled. Generative and discriminative methods have very different characteristics, as well as complementary strengths and weaknesses. In this chapter we introduce new generative and discriminative models for object detection and classification based on weakly labelled training data. We use these models to illustrate the relative merits of the two approaches in the context of a data set of widely varying images of non-rigid objects (animals). Our results support the assertion that neither approach alone will be sufficient for large scale object recognition, and we discuss techniques for combining the strengths of generative and discriminative approaches.

1 Introduction

In recent years many studies, both in machine learning and computer vision areas, have focussed on the problem of object recognition. The key challenge is to be able to recognize any member of a category of objects in spite of wide variations in visual appearance due to changes in the form and colour of the object, occlusions, geometrical transformations (such as scaling and rotation), changes in illumination, and potentially non-rigid deformations of the object itself. Since detailed hand-segmentation and labelling of images is very labour intensive, learning object categories from 'weakly labelled' data has been studied in recent years. Weakly labelled data means that training images are labelled only according to the presence or absence of each category of object. A major challenge presented by this problem is that the foreground object is accompanied by widely varying background clutter, and the system must learn to distinguish the foreground from the background without the aid of labelled data.

J. Ponce et al. (Eds.): Toward Category-Level Object Recognition, LNCS 4170, pp. 173–195, 2006.

Many of the current approaches to this problem rely on the use of local features obtained from small patches of the image. One motivation for this is that local patches can give information about an object even it is occluded. An other motivation is that the variability of small patches is much less than that of whole images and so there are much better prospects for generalization, in other words for recognizing that a patch from a test image is similar to patches in the training images. However, the patches must be sufficiently variable, and therefore sufficiently large, to be able to discriminate between the different object categories and also between objects and background clutter. A good way to balance these two conflicting requirements is to determine the object categories present in an image by fusing together partial ambiguous information from multiple patches. Probability theory provides a powerful framework for combining such uncertain information in a principled manner, and will form the basis for our research. We will also focus on the detection of objects within images by combining information from a large number of patches of the image.

Local features are obtained from small patches which are extracted from the local neighbourhood of interest points obtained in the image. Some of the interest point operators such as saliency [8], Difference of Gaussian (DoG) [11] and Harris-Laplace (HL) [12] are invariant to location, scale and orientation, and some are also affine invariant [12] to some extent. For the purposes of this chapter we shall consider the use of such generic operators. We will use some very common operators (Section 2) and feature description methods and will compare their effect in learning performance (Section 5).

Also, the locations of the patches which provide strong evidence for an object can give an indication of the location and spatial extent of that object. The probabilistic model of Fergus *et al.* [5] performed the localization of the object in an image by learning jointly the appearances and relative locations of a small set of parts whose potential locations are determined by the saliency detector [8]. Since their algorithm is computationally complex, the number of parts has to be kept small. In [10] a discriminative framework for the classification of image regions by incorporating neighborhood interactions is presented. But for two class classification only. In [4], the spatial relationship between patches was not considered but informative features (i.e. object features) were selected based on information criteria such as likelihood ratio and mutual information. However, in this supervised approach, hundreds of images were hand segmented. Finally, [19] extended the Gaussian Mixture Model (GMM) based approach of [4] to a semi-supervised case where a multi-modal GMM was trained to model foreground and background feature together. In their study, some uncluttered images of foreground were also used for the purpose of training their model. In this chapter, we do not attempt to model the spatial relationship between patches but instead focus on the comparison of generative with discriminative methods in the context of local patch labelling.

The object recognition problem is basically a classification problem and there are many different modelling approaches for the solution. These approaches can be classified into two main categories such as generative and discriminative. To understand the distinction between discriminative and generative approaches, consider a scenario in which an image described by a vector $\mathbf{X}$ (which might comprise raw pixel intensities, or some set of features extracted from the image) is to be assigned to one of K classes $k = 1, \ldots, K$. From basic decision theory [2] we know that the most complete characterization of the solution is expressed in terms of the set of posterior probabilities $p(k|\mathbf{X})$. Once we know these probabilities it is straightforward to assign the image $\mathbf{X}$ to a particular class to minimize the expected loss (for instance, if we wish to minimize the number of misclassifications we assign $\mathbf{X}$ to the class having the largest posterior probability).

In a discriminative approach we introduce a parametric model for the posterior probabilities, $p(k|\mathbf{X})$, and infer the values of the parameters from a set of labelled training data. This may be done by making point estimates of the parameters using maximum likelihood, or by computing distributions over the parameters in a Bayesian setting (for example by using variational inference).

By contrast, in a generative approach we model the joint distribution $p(k, \mathbf{X})$ of images and labels. This can be done, for instance, by learning the class prior probabilities $p(k)$ and the class-conditional densities $p(\mathbf{X}|k)$ separately. The required posterior probabilities are then obtained using Bayes' theorem

$$p(k|\mathbf{X}) = \frac{p(\mathbf{X}|k)p(k)}{\sum_j p(\mathbf{X}|j)p(j)} \tag{1}$$

where the sum in the denominator is taken over all classes.

Each modelling approach has some advantages as well as disadvantages. There are many recent studies dealing with the comparison of these two approaches with the final goal of combining the two in the best way. In [14] it was concluded that although the discriminative learning has lower asymptotic error, a generative classifier approaches its higher asymptotic error much faster. Very similar results were also obtained by [3] but they showed on a simulated data that this is only true when the models are appropriate for the data, i.e. the generative model models the data distribution correctly. Otherwise, if a mis-matched model was selected then generative and discriminative models behaved similarly, even with a small number of data points. In both [3] and [14] it was observed that as the number of data points is increased the discriminative model performs better. In [3] and [7] discriminative and generative learning were combined in an ad-hoc manner using a weighting parameter and the value of this parameter defines the extend to which discriminative learning is effective over generative learning. In [18] discriminative learning was performed on a generative model where background posterior probability was modelled with a constant.

In this chapter we will provide two different models, one from each approach, which are able to provide labels for the individual patches, as well as for the image as a whole, so that each patch is identified as belonging to one of the object categories or to the background class. This provides a rough indication of the location of the object or objects within the image. Again these individual patch labels must be learned on the basis only of overall image class labels. Our training set is weakly labelled where each image is labelled only according to the presence or absence of each category of object. Our goal in this chapter is not to find optimal object recognition system, but to compare alternative learning methodologies. For this purpose, we shall use a fixed data set. In particular, we consider the task of detecting and distinguishing cows and sheep in natural images. This set is chosen for the wide variability of the objects in order to present a non-trivial classification problem. We do not have any data set for background only. Various features used in this study are explained in Section 2. Our discriminative and generative models are introduced in Sections 3 and 4 respectively.

We use $\mathbf{t}_n$ to denote the image label vector for image n with independent components $t_{nk} \in \{0, 1\}$ in which $k = 1, \dots K$ labels the class. In our case $K = 3$ where the classes are cow, sheep and background. Each class can be present or absent independently in an image, and we make no distinction between foreground and background classes within the model itself. $\mathbf{X}_n$ denotes the observation for image n and this comprises as set of J_n patch vectors $\{\mathbf{x}_{nj}\}$ where $j = 1, \dots, J_n$. Note that the number J_n of detected interest points will in general vary from image to image.

We shall compare the two models in various aspects. First we will investigate how the models behave with weakly labelled data and then we will test how strongly labelled (i.e. images are segmented as foreground and background) and weakly labelled data can be used together in training the models. Experiments and results for this is given in Section 5.1. Secondly, we will test the models with various types of feature as inputs to see how feature type effects the models. Experiments and results for this is given in Section 5.2. Finally, as many previous studies did, we will see how training data quantity affects learning in the two different model types. Experiments and results for this is given in Section 5.3.

2 Feature Extraction

Due to the reasons that we have mentioned in the previous section,we will follow several recent approaches and use interest point detectors to focus attention on a small number of local patches in each image. This is followed by invariant feature extraction from a neighbourhood around each interest point.

We choose to work with Harris-Laplace (HL) [12] and Difference of Gaussian (DoG) [11] interest point operators because they are invariant to orientation and scale changes. In our earlier study [16] we have used DoG interest point detector with SIFT (Scale Invariant Feature Transform) descriptor. SIFT is invariant to illumination and affine (to some degree) changes and very suitable for DoG interest point detectors. However SIFT, being a 128 dimentional vector, brings a high computational load for model learning. Thus, in this chapter we will use 15 dimensional Local Jet (LJ) descriptor instead [9,6].

For the purpose of comparison, we will train our models using different feature types and see how they are effected by these choices. The two feature point operators, HL and DoG, will be used with the same feature descriptor (LJ). In Figure 1 a cow image is shown together with with HL and DoG feature point detectors in order to give more insight into these two types of operators. Here only feature points which have scale grater than 5 pixels are shown. As can be observed from the images, the DoG operator extracts uniform regions (leftmost image in Figure 1) and HL extracts corners (middle image in the figure) where the number of features extracted by HL is usually less than DoG.

The feature descriptor may be concatenated with colour information. The colour information is extracted from each patch based on [1]. Averages and standard deviations of (R, G, B), (L, a, b) and $(r = R/(R+G+B), g = G/(R+G+B))$ constitute the colour part of the feature vector. Lab is a device-independent colour space that attempts to uniformly represent colour as we perceive it. L is the lightness value, a is the red/green opponency and blue/yellow is represented on the b axis. As a result, if colour is also used as a feature descriptor then we will have a 31 dimensional feature vector.

Just for comparison purposes, we will also use square random patches as interest regions which are selected at random sizes and random positions all over the image. Since the size of a patch can vary between 1 pixel to the full size of the image, the patches will be scaled to 16 by 16 size. If each pixel's colour information is used directly to form a feature vector, this makes a feature vector of size 768 ($16 \times 16 \times 3$) and it is impossible to use this directly in our models (especially in the generative model). Thus, we compute first 15 Principle Component Analysis (PCA) coefficients for the gray scale patch and we obtain the colour feature as described in the previous paragraph. Again this makes a 31 dimensional feature vector. The number of random patches is selected to be approximately the same as the number of patches found by other interest point operators, which is around 100 for each image. In the rightmost image in Figure 1 the cow image with some of the random patches is also shown. We only show 10 random patches here. In Section 5.2, comparison of the two models when used with different features will be given in terms of patch labelling and image labelling. We will compare HL and DoG operators with LJ and colour feature, and random patches with PCA coefficients and colour feature.

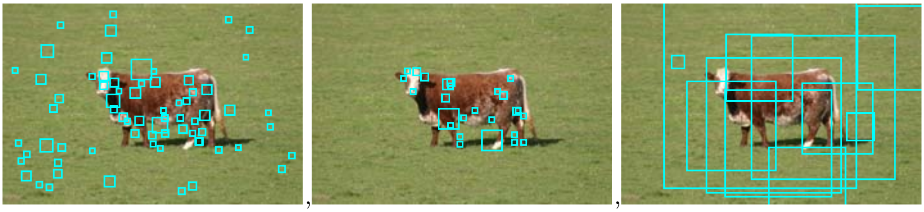


Fig. 1. Different interest point operators. Feature point locations are the centers of the squares and the size of a square shows the scale of that feature point. The three images show (left to right) DoG interest points, HL interest points and random patches.

3 The Discriminative Model with Patch Labelling

In a discriminative setting, the purpose is to learn the posterior probabilities. Since our goal is to determine the class membership of individual patches also, we associate with each patch j in an image n a binary label $\tau_{njk} \in \{0, 1\}$ denoting the class k of the patch. For the models developed in this chapter we shall consider these labels to be mutually exclusive, so that $\sum_{k=1}^{K} \tau_{njk} = 1$, in other words each patch is assumed to be either cow, sheep or background. Note that this assumption is not essential, and other formulations could also be considered. These components can be grouped together into vectors $\boldsymbol{\tau}_{nj}$. If the values of these labels were available during training (corresponding to strongly labelled images) then the development of recognition models would be greatly simplified. For weakly labelled data, however, the $\{\boldsymbol{\tau}_{nj}\}$ labels are hidden (latent) variables, which of course makes the training problem much harder.

We now introduce a discriminative model, which corresponds to the directed graph shown in Figure 2.

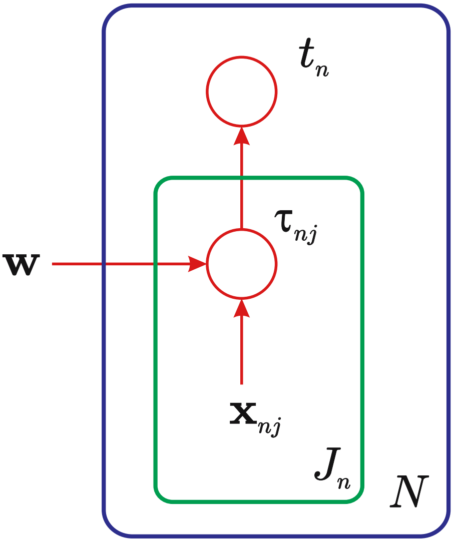


Fig. 2. Graphical representation of the discriminative model for object recognition

Consider for a moment a particular image n (and omit the index n to keep the notation uncluttered). We build a parametric model $y_k(\mathbf{x}_j, \mathbf{w})$ for the probability that patch $\mathbf{x}_j$ belongs to class k. For example we might use a simple linear-softmax model with outputs

$$y_k(\mathbf{x}_j, \mathbf{w}) = \frac{\exp(\mathbf{w}_k^{\mathrm{T}} \mathbf{x}_j)}{\sum_l \exp(\mathbf{w}_l^{\mathrm{T}} \mathbf{x}_j)} \tag{2}$$

which satisfy $0 \leqslant y_k \leqslant 1$ and $\sum_k y_k = 1$. More generally we can use a multi-layer neural network, a relevance vector machine, or any other parametric model that gives probabilistic outputs and which can be optimized using gradient-based methods. The probability of a patch label $\boldsymbol{\tau}_j$ is then given by

$$p(\boldsymbol{\tau}_j|\mathbf{x}_j) = \prod_{k=1}^{K} y_k(\mathbf{x}_j, \mathbf{w})^{\tau_{jk}} \tag{3}$$

where the binary exponent τ_{jk} simply pulls out the required term (since $y_k^0 = 1$ and $y_k^1 = y_k$).

Next we assume that if one, or more, of the patches carries the label for a particular class, then the whole image will. For instance, if there is at least one local patch in the image which is labelled 'cow' then the whole image will carry a 'cow' label (recall that an image can carry more than one class label at a time). Thus the conditional distribution of the image label, given the patch labels, is given by

$$p(\mathbf{t}|\boldsymbol{\tau}) = \prod_{k=1}^{K} \left[1 - \prod_{j=1}^{J} [1 - \tau_{jk}] \right]^{t_k} \left[\prod_{j=1}^{J} [1 - \tau_{jk}] \right]^{1-t_k} . \tag{4}$$

In order to obtain the conditional distribution $p(\mathbf{t}|\mathbf{X})$ we have to marginalize over the latent patch labels. Although there are exponentially many terms in this sum, it can be performed analytically for our model due to the factorization implied by the graph in Figure 2 to give

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}) &= \sum_{\boldsymbol{\tau}} \left\{ p(\mathbf{t}|\boldsymbol{\tau}) \prod_{j=1}^{J} p(\boldsymbol{\tau}_j|\mathbf{x}_j) \right\} \\ &= \prod_{k=1}^{K} \left[1 - \prod_{j=1}^{J} [1 - y_k(\mathbf{x}_j, \mathbf{w})] \right]^{t_k} \left[\prod_{j=1}^{J} [1 - y_k(\mathbf{x}_j, \mathbf{w})] \right]^{1-t_k} . \end{aligned} \tag{5}$$

This can be viewed as a probabilistic version of the 'noisy OR' function [15].

Given a training set of N images, which are assumed to be independent, we can construct the likelihood function from the product of such distributions, one for each data point. Taking the negative logarithm then gives the following error function

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{C} \{ t_{nk} \ln [1 - Z_{nk}] + (1 - t_{nk}) \ln Z_{nk} \} \tag{6}$$

where we have defined

$$Z_{nk} = \prod_{j=1}^{J_n} [1 - y_k(\mathbf{x}_{nj}, \mathbf{w})] . \tag{7}$$

The parameter vector $\mathbf{w}$ can be determined by minimizing this error (which corresponds to maximizing the likelihood function) using a standard optimization algorithm such as scaled conjugate gradients [2]. More generally the likelihood function could be used as the basis of a Bayesian treatment, although we do not consider this here.

Once the optimal value $\mathbf{w}_{\mathrm{ML}}$ is found, the corresponding functions $y_k(\mathbf{x}, \mathbf{w}_{\mathrm{ML}})$ for $k = 1, \ldots, K$ will give the posterior class probabilities for a new patch feature vector $\mathbf{x}$. Thus the model has learned to label the patches even though the training data contained only image labels. Note, however, that as a consequence of the 'noisy OR' assumption, the model only needs to label one foreground patch correctly in order to predict the image label. It will therefore learn to pick out a small number of highly discriminative foreground patches, and will classify the remaining foreground patches, as well as those falling on the background, as 'background' meaning non-discriminative for the foreground class. This will be illustrated in Section 5.1.

3.1 Soft Discriminative Model

In our discriminative model with probabilistic noisy OR assumption, if only one patch is labelled as belonging to a class, then the whole image is labelled as belonging to that class. We can soften this assumption by modelling the posterior probability of the image label using the logistic sigmoid function

$$p\left(t_k = 1|\mathbf{X}\right) = \frac{1}{1+e^{-Z_k}} \tag{8}$$

where Z_k is the sum over all patches

$$Z_k = \sum_{j=1}^{J} y_k\left(\mathbf{x}_j, \mathbf{w}\right) \tag{9}$$

where

$$y_k(\mathbf{x}_j, \mathbf{w}) = \mathbf{w}_k^{\mathrm{T}} \mathbf{x}_j \tag{10}$$

so that we are adding the log odds. It follows that the conditional distribution of target labels is given by

$$p\left(t_k|\mathbf{X}\right) = \left(\frac{1}{1+e^{-Z_k}}\right)^{t_n k} \left(1 - \frac{1}{1+e^{-Z_k}}\right)^{1-t_k}. \tag{11}$$

The distribution for the vector of target variables is then given by

$$p\left(\mathbf{t}|\mathbf{X}\right) = \prod_{k=1}^{K} p\left(t_k|\mathbf{X}\right). \tag{12}$$

However outputs of this model can not be directly used as patch label probabilities because they are not normalized and they don't satisfy $\sum_k y_k = 1$. This does not cause a problem in finding the most probable patch label. We can directly use the model outputs and choose the biggest one as patch label. However, when we need patch label probabilities then we need to normalize the model outputs over all possible patches and labels.

The error function for this soft discriminative model is given by the negative log likelihood, and takes the form

$$E\left(\mathbf{w}\right) = -\sum_{n=1}^{N}\sum_{k=1}^{K}\left\{Z_{nk}\left(t_{nk}-1\right) - \ln\left(1+e^{-Z_{nk}}\right)\right\}. \tag{13}$$

With this soft version, an improvement in both patch labelling and image labelling is obtained. Comparative results for the two discriminative models (probabilistic noisy OR and soft) are given in Section 5.1.

4 The Generative Model with Patch Labelling

Next we turn to a description of our generative model, whose graphical representation is shown in Figure 3. The structure of this model mirrors closely that of the discriminative model. In particular, the same class-label variables $\boldsymbol{\tau}_{nj}$ are associated with the patches in each image, and again these are unobserved and must be marginalized out in order to obtain maximum likelihood solutions.

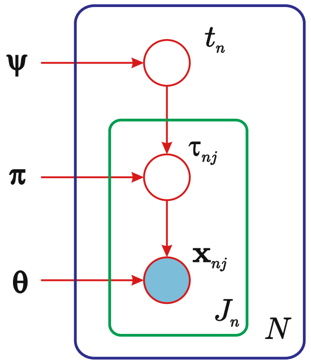


Fig. 3. Graphical representation of the generative model for object recognition

In the discriminative model we represented the conditional distribution $p(\mathbf{t}|\mathbf{X})$ directly as a parametric model. By contrast in the generative approach we model $p(\mathbf{t},\mathbf{X})$, which we decompose into $p(\mathbf{t},\mathbf{X}) = p(\mathbf{X}|\mathbf{t})p(\mathbf{t})$ and then model the two factors separately. This decomposition would allow us, for instance, to employ large numbers of 'background' images (those containing no instances of the object classes) during training to determine $p(\mathbf{X}|\mathbf{t})$ without concluding that the prior probabilities $p(\mathbf{t})$ of objects is small.

Again, we begin by considering a single image n. The prior $p(\mathbf{t})$ is specified in terms of K parameters ψ_k where $0 \leqslant \psi_k \leqslant 1$ and $k = 1, \ldots, K$, so that

$$p(\mathbf{t}) = \prod_{k=1}^{K} \psi_k^{t_k} (1 - \psi_k)^{1-t_k}. \tag{14}$$

In general we do not need to learn these from the training data since the prior occurrences of different classes is more a property of the way the data was collected than of the real world frequencies. (Similarly in the discriminative model we will typically wish to correct for different priors between the training set and test data using Bayes' theorem.)

The remainder of the model is specified in terms of the conditional probabilities $p(\boldsymbol{\tau}|\mathbf{t})$ and $p(\mathbf{X}|\boldsymbol{\tau})$. 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

$$p(\boldsymbol{\tau}_j|\mathbf{t}) = \left(\sum_{l=1}^{K} t_l \pi_l \right)^{-1} \prod_{k=1}^{K} (t_k \pi_k)^{\tau_{jk}}. \tag{15}$$

Note that there is an overall undetermined scale to these parameters, which may be removed by fixing one of them, e.g. $\pi_1 = 1$.

For each class k, the distribution of the patch feature vector $\mathbf{x}$ is governed by a separate mixture of Gaussians which we denote by $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$, so that

$$p(\mathbf{x}_j|\boldsymbol{\tau}_j) = \prod_{k=1}^{K} \phi_k(\mathbf{x}_j; \boldsymbol{\theta}_k)^{\tau_{jk}} \tag{16}$$

where $\boldsymbol{\theta}_k$ denotes the set of parameters (means, covariances and mixing coefficients) associated with this mixture model, and again the binary exponent τ_{jk} simply picks out the required class.

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

$$\prod_{n=1}^{N} p(\mathbf{t}_n) \prod_{j=1}^{J_n} \left[p(\mathbf{x}_{nj}|\boldsymbol{\tau}_{nj}) p(\boldsymbol{\tau}_{nj}|\mathbf{t}_n) \right]. \tag{17}$$

Since we wish to maximize likelihood in the presence of latent variables, namely the $\{\boldsymbol{\tau}_{nj}\}$, we use the EM algorithm. The expected complete-data log likelihood is given by

$$\sum_{n=1}^{N} \sum_{j=1}^{J_n} \left\{ \sum_{k=1}^{K} \langle \tau_{njk} \rangle \ln \left[t_{nk} \pi_k \phi_k(\mathbf{x}_{nj}) \right] - \ln \left(\sum_{l=1}^{K} t_{nl} \pi_l \right) \right\}. \tag{18}$$

In the E-step the expected values of τ_{nkj} are computed using

$$\langle \tau_{njk} \rangle = \sum_{\{\boldsymbol{\tau}_{nj}\}} \tau_{njk} p(\boldsymbol{\tau}_{nj}|\mathbf{x}_{nj}, \mathbf{t}_n) = \frac{t_{nk} \pi_k \phi_k(\mathbf{x}_{nj})}{\displaystyle\sum_{l=1}^{K} t_{nl} \pi_l \phi_l(\mathbf{x}_{nj})}. \tag{19}$$

Notice that the first factor on the right hand side of (15) has cancelled in the evaluation of $\langle \tau_{njk} \rangle$.

For the M-step we first set the derivative with respect to one of the parameters π_k equal to zero (no Lagrange multiplier is required since there is no summation constraint on the $\{\pi_k\}$) and then re-arrange to give the following re-estimation equations

$$\pi_k = \left[\sum_{n=1}^{N} J_n t_{nk} \left(\sum_{l=1}^{K} t_{nl} \pi_l \right)^{-1} \right]^{-1} \sum_{n=1}^{N} \sum_{j=1}^{J_n} \langle \tau_{njk} \rangle. \tag{20}$$

Since these represent coupled equations we perform several (fast) iterations of these equations before proceeding with the next EM cycle (note that for this purpose the sums over j can be pre-computed since they do not depend on the $\{\pi_k\}$).

Now consider the optimization with respect to the parameters $\boldsymbol{\theta}_k$ governing the distribution $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$. The dependence of the expected complete-data log likelihood on $\boldsymbol{\theta}_k$ takes the form

$$\sum_{n=1}^{N} \sum_{j=1}^{J_n} \langle \tau_{njk} \rangle \ln \phi_k(\mathbf{x}_{nj}; \boldsymbol{\theta}_k) + \text{const.} \tag{21}$$

This is easily maximized for each class k separately using the EM algorithm (in an inner loop), since (21) simply represents a log likelihood function for a weighted data set in which patch (n, j) is weighted with $\langle \tau_{njk} \rangle$. Specifically, we use a model in which $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$ is given by a Gaussian mixture distribution of the form

$$\phi_k(\mathbf{x}; \boldsymbol{\theta}_k) = \sum_{m=1}^{M} \rho_{km} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km}). \tag{22}$$

The E-step is given by

$$\gamma_{njkm} = \frac{\rho_{km} \mathcal{N}(\mathbf{x}_{nj} | \boldsymbol{\mu}_{km}, \boldsymbol{\Sigma}_{km})}{\sum_{m'} \rho_{km'} \mathcal{N}(\mathbf{x}_{nj} | \boldsymbol{\mu}_{km'}, \boldsymbol{\Sigma}_{km'})} \tag{23}$$

while the M-step equations are weighted by the coefficients $\langle \tau_{njk} \rangle$ to give

$$\begin{aligned}
\boldsymbol{\mu}_{km}^{\text{new}} &= \frac{\sum_n \sum_j \langle \tau_{njk} \rangle \gamma_{njkm} \mathbf{x}_{nj}}{\sum_n \sum_j \langle \tau_{njk} \rangle \gamma_{njkm}} \\
\boldsymbol{\Sigma}_{km}^{\text{new}} &= \frac{\sum_n \sum_j \langle \tau_{njk} \rangle \gamma_{njkm} (\mathbf{x}_{nj} - \boldsymbol{\mu}_{km}^{\text{new}})(\mathbf{x}_{nj} - \boldsymbol{\mu}_{km}^{\text{new}})^{\mathrm{T}}}{\sum_n \sum_j \langle \tau_{njk} \rangle \gamma_{njkm}} \\
\rho_{km}^{\text{new}} &= \frac{\sum_n \sum_j \langle \tau_{njk} \rangle \gamma_{njkm}}{\sum_n \sum_j \langle \tau_{njk} \rangle}.
\end{aligned}$$

If one EM cycle is performed for each mixture model $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$ this is equivalent to a global EM algorithm for the whole model. However, it is also possible to perform several EM cycle for each mixture model $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$ within the outer

EM algorithm. Such variants yield valid EM algorithms in which the likelihood never decreases.

The incomplete-data log likelihood can be evaluated after each iteration to ensure that it is correctly increasing. It is given by

$$\sum_{n=1}^{N}\sum_{j=1}^{J_n}\left\{\ln\left(\sum_{k=1}^{K} t_{nk}\pi_k\phi_k(\mathbf{x}_{nj})\right)-\ln\left(\sum_{l=1}^{K} t_{nl}\pi_l\right)\right\}.$$

Note that, for a data set in which all $t_{nk} = 1$, the model simply reduces to fitting a flat mixture to all observations, and the standard EM is recovered as a special case of the above equations.

This model can be viewed as a generalization of that presented in [19] in which a parameter is learned for each mixture component representing the probability of that component being foreground. This parameter is then used to select the most informative N components in a similar approach to [4] and [17] where the number N is chosen heuristically. In our case, however, the probability of each feature belonging to one of the K classes is learned directly.

Inference in the generative model is more complicated than in the discriminative model. Given all patches $\mathbf{X} = \{\mathbf{x}_j\}$ from an image, the posterior probability of the label $\boldsymbol{\tau}_j$ for patch j can be found by marginalizing out all other hidden variables

$$\begin{aligned} p\left(\boldsymbol{\tau}_j|\mathbf{X}\right) &= \sum_{\mathbf{t}}\sum_{\boldsymbol{\tau}/\boldsymbol{\tau}_j} p\left(\boldsymbol{\tau},\mathbf{X},\mathbf{t}\right) \\ &= \sum_{\mathbf{t}} p\left(\mathbf{t}\right)\frac{1}{\left(\sum_{l=1}^{K}\pi_l t_l\right)^{J}}\prod_{k=1}^{K}\left(\pi_k t_k\phi_k\left(\mathbf{x}_j\right)\right)^{\tau_{jk}}\prod_{i\neq j}\left[\sum_{k=1}^{K}\pi_k t_k\phi_k\left(\mathbf{x}_i\right)\right] \qquad (24) \end{aligned}$$

where $\boldsymbol{\tau} = \{\boldsymbol{\tau}_j\}$ denotes the set of all patch labels, and $\boldsymbol{\tau}/\boldsymbol{\tau}_j$ denotes this set with $\boldsymbol{\tau}_j$ omitted. Note that the summation over all possible $\mathbf{t}$ values, which must be done explicitly, is computationally expensive.

For the inference of image label we require the posterior probability of image label $\mathbf{t}$, which can be computed using

$$p\left(\mathbf{t}|\mathbf{X}\right) \propto p\left(\mathbf{X}|\mathbf{t}\right)p\left(\mathbf{t}\right) \qquad (25)$$

in $p(\mathbf{t})$ is computed from the coefficients $\{\psi_k\}$ for each setting of $\mathbf{t}$ in turn, and $p\left(\mathbf{X}|\mathbf{t}\right)$ is found by summing out patch labels

$$p\left(\mathbf{X}|\mathbf{t}\right) = \sum_{\boldsymbol{\tau}}\prod_{j=1}^{J} p\left(\mathbf{X},\boldsymbol{\tau}_j|\mathbf{t}\right) = \prod_{j=1}^{J_n}\frac{\sum_{k=1}^{K} t_k\pi_k\phi_k\left(\mathbf{x}_j\right)}{\sum_{l=1}^{K} t_l\pi_l}. \qquad (26)$$

5 Experiments and Results

In this chapter, we have used a test bed of weakly labelled images each containing either cows or sheep, in which the animals vary widely in terms of number, pose, size, colour and texture. There are 167 images in each class, and 10-fold cross-validation is used to measure performance. For the discriminative model we used a two-layer nonlinear network having 10 hidden units with 'tanh' activation functions. The network had 31 inputs, corresponding to the LJ or PCA coefficient with colour feature as discussed in Section 2 and 3 outputs (cow, sheep, background). For the generative model we used a separate Gaussian mixture for cow, sheep and background, each of which has 10 components with diagonal covariance matrices. In our earlier study [16] we used input vector of size 144 which consists of SIFT and colour features. Using a smaller feature vector this time brings computational benefit such as speed and computable covariance matrixes.

In the test phase of both discriminative and generative models, we input the patch features to the models and obtain the posterior probabilities of the patch labels as the outputs using (2) for probabilistic noisy OR discriminative model or (10) with normalization for soft discriminative model and (24) for the generative model. The posterior probability of the image label is computed as in (5) for probabilistic noisy OR model or (12) for the soft discriminative model and (25) for the generative case. We can therefore investigate the ability of the models both to predict the class labels of whole images and of their constituent patches. The latter is important for object localization.

5.1 Combining Strongly Labelled and Weakly Labelled Data for Training

Initial results with the generative model showed that with random initialization of the mixture model parameters it is incapable of learning a satisfactory solution [16]. We conjectured that this is due to the problem of multiple local maxima in the likelihood function (a similar effect was found by [19]). To test this, we used some segmented images for initialization purposes (but not for optimization) in our earlier study [16]. 30 cow and 30 sheep images were hand-segmented, and a patch which has any foreground pixel was labelled as foreground and a patch which has no foreground pixel was labelled as background. Features obtained from the patches belonging to each class were clustered using the K-means algorithm and the component centers of a class mixture model were assigned to the cluster centers of the respective class. The mixing coefficients were set to the number of points in the corresponding cluster divided by the total number of points in that class. Similarly, covariance matrices were computed using the data points assigned to the respective center.

In this chapter, we use these segmented images also for training optimization in order to give both models the same chance. In the generative case, including the segmented data into learning requires only a slight change in the expected complete-data log likelihood which becomes partially expected in this case:

$$\begin{aligned}&\sum_{n\in \mathrm{US}}\sum_{j=1}^{J_n}\left\{\sum_{k=1}^{K}\langle\tau_{njk}\rangle \ln\left[t_{nk}\pi_k\phi_k(\mathbf{x}_{nj})\right]-\ln\left(\sum_{l=1}^{K}t_{nl}\pi_l\right)\right\}\\&+\sum_{n\in \mathrm{S}}\sum_{j=1}^{J_n}\left\{\sum_{k=1}^{K}\tau_{njk} \ln\left[t_{nk}\pi_k\phi_k(\mathbf{x}_{nj})\right]-\ln\left(\sum_{l=1}^{K}t_{nl}\pi_l\right)\right\}\end{aligned} \qquad (27)$$

where S and US denote segmented and unsegmented image sets respectively. For segmented images $n \in \mathrm{S}$, τ_{nkj} values are already known. Including the segmented data to the generative model is very easy where we only need to assign known patch labels instead of their expected labels in the outer E step (19) mentioned in Section 4.

For the probabilistic noisy OR discriminative model, the error function becomes

$$E(\mathbf{w}) = -\sum_{n \in \mathrm{US}} \sum_{k=1}^{K} \left\{ t_{nk} \ln \left[1 - Z_{nk}\right] + \left(1 - t_{nk}\right) \ln Z_{nk} \right\} \\ - \sum_{n \in \mathrm{S}} \sum_{j=1}^{J_n} \sum_{k=1}^{K} \tau_{njk} \ln(y_k(\mathbf{x}_{nj}, \mathbf{w})) \qquad (28)$$

where the first term on the right hand side of the error function includes unsegmented images and is the image labelling error, while the second term includes segmented images and is the patch labelling error.

Similarly, for the soft discriminative model, the error function (29) consists of two parts: one with unlabelled data and the other with labelled data. These two parts need to be treated differently during all optimization steps.

$$E(\mathbf{w}) = -\sum_{n \in \mathrm{US}} \sum_{k=1}^{K} \left\{ Z_{nk} \left(t_{nk} - 1\right) - \ln \left(1 + e^{-Z_{nk}}\right) \right\} \\ - \sum_{n \in \mathrm{S}} \sum_{j=1}^{J_n} \sum_{k=1}^{K} \left(y_k(\mathbf{x}_{nj}, \mathbf{w}) - \tau_{njk}\right) \qquad (29)$$

To test the effect of labelled data on the generative model, we train the same generative model with and without labelled data and compared the results. When only unlabelled data is used (i.e. no initialization is performed) overall correct rate (ocr) for image labelling is obtained to be 46.50% which is worse than random labelling. When segmented data is used for initialization only then there is a significant increase in the performance where ocr becomes 59.37%. When the segmented data is used for training as well the performance is not effected much where ocr stays at 59.37%. In Figure 4 examples for generative model patch labelling are given for different situations where most probable label is assigned for each patch. Patch centers are shown by coloured dots where colour denotes the class (red, white, green for cow, sheep and background respectively). As can be observed from the image, without initialization patch labelling is as random (top image of the figure). Image labelling result for this particular sheep image is $\mathbf{t} = [1\ 0\ 1]$ for this sample run which means that this is a cow image. With initialization, most of the patches are labelled correctly (middle image in the figure). Image label for the same sheep is $\mathbf{t} = [1\ 1\ 1]$ this time which means there are both cow and sheep (as well as background) present in the image. When segmented data is also used for training (bottom image) patch labelling performance becomes better and sheep image is labelled correctly as $\mathbf{t} = [0\ 1\ 1]$.

Using segmented data for the probabilistic noisy OR discriminative model brings some problems. When labelled data is also used for training, although the patch labelling performance increases significantly image labelling performance degrades. For example, in Figure 4 patch labelling results during a sample run are given where the most probable label is assigned to each patch. Top image is an example which is obtained when segmented data is not used in training and ocr for this case is 62.50%. Image labelling result is correct for this particular cow with $\mathbf{t} = [0.99\, 0.50\, 1]$ which becomes $\mathbf{t} = [1\, 0\, 1]$ when 0.5 is used as a threshold for image label probability. Middle image is obtained when segmented data is used for training the model and ocr for this case is very low, 30%. In this case patch labelling is better but image label for this particular cow image is $\mathbf{t} = [1\ 0.83\ 1]$ which means that there is a high probability of sheep also. This is caused by a white (sheep) patch in the cow image. The bottom image is when the soft discriminative model is trained with segmented data where ocr becomes 78.1%. Patch labelling is as good as the previous case but this time image labelling is also correct $\mathbf{t} = [1\ 0\ 1]$ for this particular cow image although there are two white (sheep) patches. This shows that when we use segmented data and force the probabilistic noisy OR discriminative model to learn those patches as they are labelled then the discriminative power decreases because those patches may not be that discriminative. However this is not the case for soft discriminative model.

As we mentioned in Section 3.1 outputs are linear for our soft discriminative model and this means that outputs can take any real value. Thus, normalization is required for this model when we need patch label probabilities.

5.2 Comparison with Different Feature Types

In this section we will provide comparative results between our generative (G) and soft discriminative (D) model when they are used with different types of features such as HL operator with LJ and colour feature (HL-LJ+C), DoG operator with LJ and colour (DoG-LJ+C) and random patches with PCA coefficients and colour feature (R-PCA+C). Usually DoG feature point operator finds more points than HL operator does when applied on the same image. In the random selection case we define the number of feature points and their local extension. In order to eliminate the effect of data quantity in the comparison, we arranged the feature point extraction algorithms so that they produce roughly the same amount of feature points (around 100) for each image. Means and standard deviations of overall correct rate results over 10 fold runs are given in Table 1. Columns are for different feature types and rows are for different models.

As can be observed from the table, ocr for discriminative model is not effected much when different feature types are used. The best overall correct rate for the discriminative model is obtained by DoG-LJ+C feature and R-PCA+C feature causes the worst performance. The generative model produces highly different overall correct rates with different feature types. The best performance for the generative model is obtained by the random patches. With DoG-LJ+C and HL-LJ+C the performance is worse than the random patches.

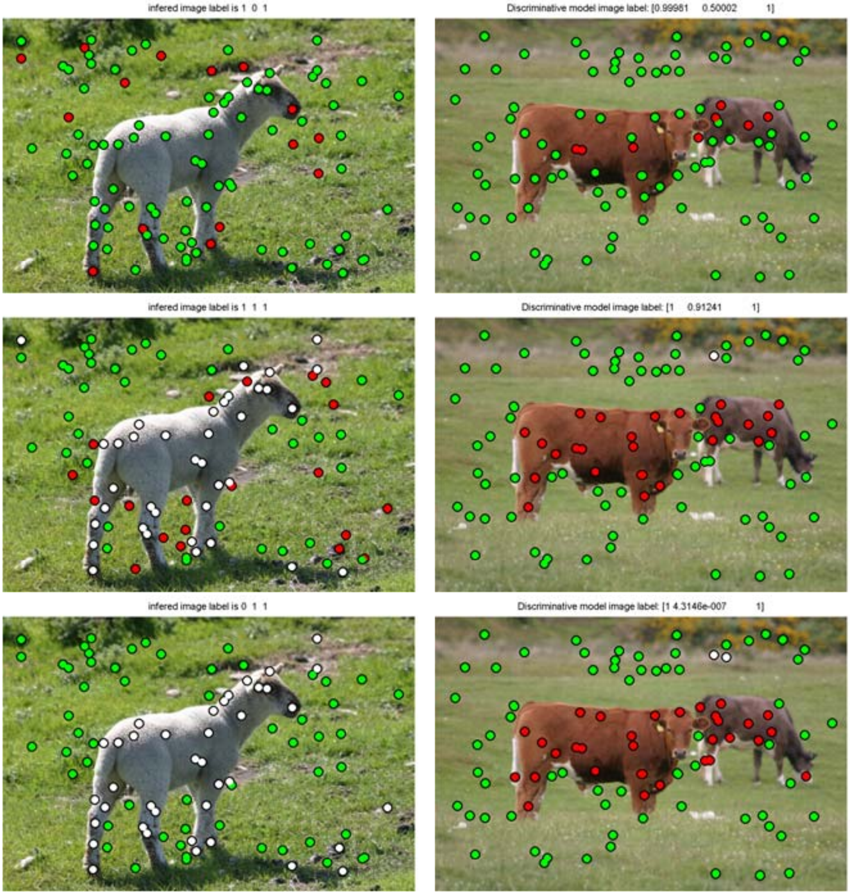

infered image label is 1 0 1
Discriminative model image label: [0.99981 0.50002 1]
infered image label is 1 1 1
Discriminative model image label: [1 0.91241 1]
infered image label is 0 1 1
Discriminative model image label: [1 4.3146e-007 1]

Fig. 4. Patch labelling results (red, white, green for cow, sheep and background respectively). Left column: Labelling results for the generative model where the most probable label is assigned to each patch. Patch labelling result in the top image is obtained when the generative model is trained without initialization. The middle image is when labelled data is used only for initializing the model. The bottom image is when the segmented images are used for both initializing and training the model. Right column: Labelling results for discriminative models where the most probable label is assigned to each patch. Top image is obtained when segmented data is not used in training of probabilistic noisy OR discriminative model. Middle row is when segmented data is used for training the same model. The bottom row is when the soft discriminative model is trained with segmented data.

It is also interesting to investigate the extent to which the discriminative and generative models correctly label the individual patches. In order to make a comparison in terms of patch labelling we use 12 hand segmented test images for each class. These segmented images are different from those we have used for initializing and training the models. Patch labels are obtained by (24) for the

Table 1. Means (M) and standard deviations (SD) of overall correct image label rate for different feature types: HL with LJ and colour (HL-LJ+C), DoG with LJ and colour (DoG-LJ+C) and random patches with PCA coefficients and colour (R-PCA+C)

	HL-LJ+C	DoG-LJ+C	R-PCA+C
D (M)(%)	80.63	89.38	78.13
D (SD)(%)	7.13	4.74	3.83
G (M)(%)	56.25	56.25	75.62
G (SD)(%)	6.25	9.88	2.61

generative model and by (10) for the soft discriminative model. Normalization is required for the discriminative model in order to obtain patch label probabilities. Various thresholds are used on patch label probabilities in order to produce ROC curves for the generative model and the soft discriminative model, as shown in Figure 5.

As can be observed from the plots the generative model patch labelling is better than the discriminative model patch labelling for all types of features and patch labelling with DoG operator with LJ and colour feature is better than other feature types.

Some examples of patch labelling for test images are given in Figure 6 for random patches and for DoG patches, and in Figure 7 for HL patches. In these figures each patch is assigned to the most probable class and patch centers are given with coloured dots where colour denotes the patch label.

5.3 Comparison for Training Data Quantity

We trained our models with various number of training data. We used 50 to 150 images with 25 intervals from each class for training and plot overall correct rate versus number of images used in training for both models in Figure 8. The left figure corresponds to the use of random patches, while the right figure corresponds to the use of DoG patches.

Similar results as [14] and [3] are obtained in this chapter also. Since the generative model performs the best with random patches (Section 5.2) we were expecting that with less data the generative model performance should be better than discriminative model. As can be observed from the left plots in Figure 8 the generative model performance is much better than the discriminative one for less data and as the quantity of data is increased discriminative model performance increases much faster than the generative model's performance. When DoG-LJ+C features are used, since the generative model does not perform well with this feature type, we were not expecting same type of behaviour. As can be seen in the right hand plots in Figure 8, the generative and the discriminative models behave nearly the same as we increase the data quantity but the discriminative model performs better than the generative model all the time.

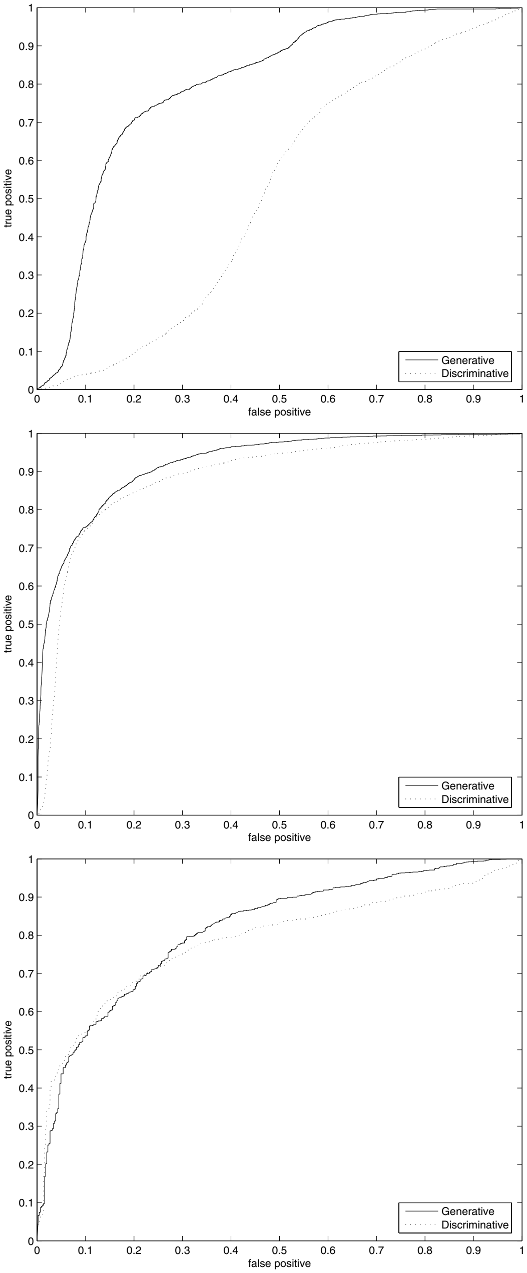
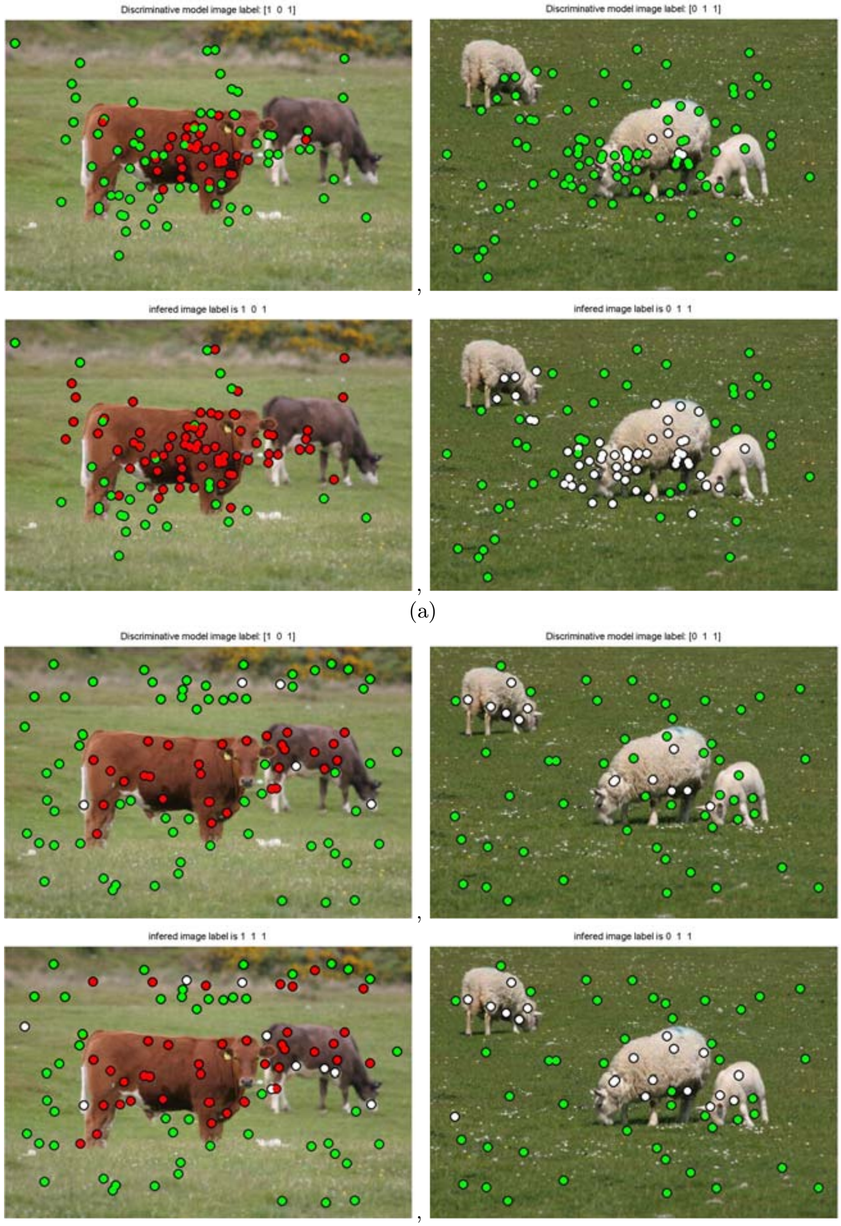


Fig. 5. ROC curves of patch labelling. Each figure contains two curves. One for the generative model and the other one for the discriminative model. Upper figure is for R-PCA+C patches. Center one is for DoG-LJ+C. Bottom one is for HL-LJ+C.


Discriminative model image label: [1 0 1]
Discriminative model image label: [0 1 1]
infered image label is 1 0 1
infered image label is 0 1 1
(a)
Discriminative model image label: [1 0 1]
Discriminative model image label: [0 1 1]
infered image label is 1 1 1
infered image label is 0 1 1

(b)

Fig. 6. Patch labelling examples for random patches (a) and for DoG patches (b). Results are shown for discriminative model (top row) and generative model (bottom row) for cow (left column) and sheep (right column) image. Red, white, green dots denote cow, sheep and background patches respectively and patch labels are obtained by assigning each patch to the most probable class.

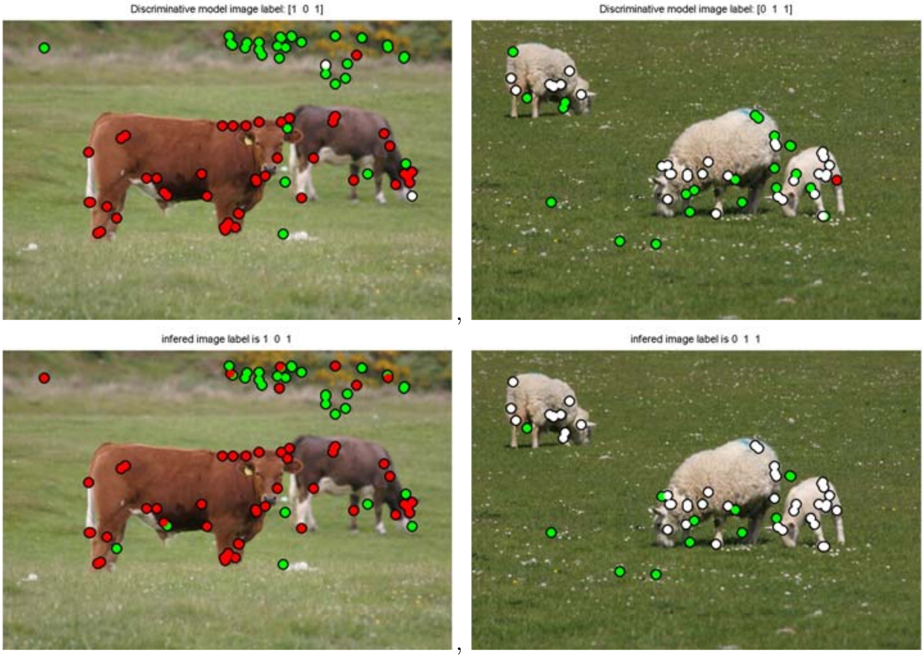

Discriminative model image label: [1 0 1]
Discriminative model image label: [0 1 1]
infered image label is 1 0 1
infered image label is 0 1 1

Fig. 7. Patch labelling examples for HL patches. Results for discriminative model (top row) nd generative model (bottom row) for cow (left column) and sheep (right column) image. Red, white, green dots denote cow, sheep and background patches respectively and patch labels are obtained by assigning each patch to the most probable class.

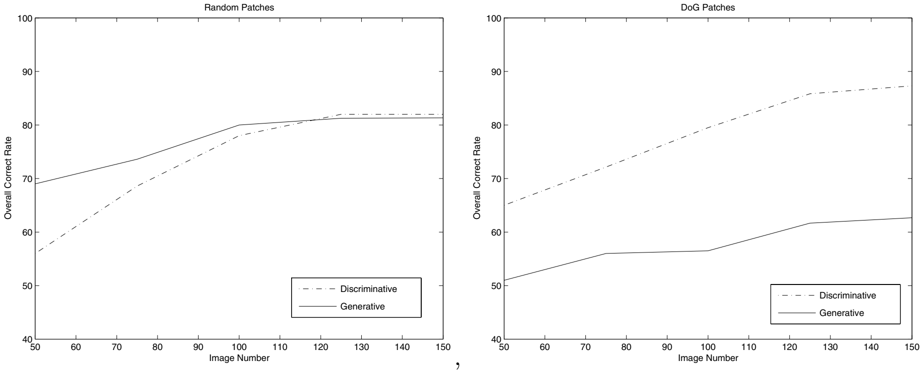


Fig. 8. Overall correct rate versus data number plots to show how the models behave as the data quantity is increased. Left figure is when random patches are used and the right figure is when DoG features are used.

6 Discussion

In our earlier study [16], we introduced novel discriminative (probabilistic noisy OR) and generative models. We used SIFT features only and showed that the probabilistic noisy OR discriminative model and the generative model have complementary strengths and limitations. The discriminative model is able to focus on highly informative features, while the generative model gives high classification accuracy, and also has some ability to localize the objects within the image. However, the generative model required careful initialization in order to achieve good results. Also, inference in such a generative model can be very complex. A discriminative model, on the other hand, is typically very fast once trained.

In this chapter, we have introduced a soft version of our previous probabilistic noisy OR discriminative model [16]. The soft discriminative model introduced here has a better patch labelling capability than probabilistic noisy OR one.

We have compared our soft discriminative and generative models in terms of using strongly labelled and weakly labelled data together in training. Combining these two data types is very easy in the generative model training but needs lots of variations in the discriminative case. The generative model, unlike the discriminative ones, could also benefit from the use of completely unlabelled images, although we have not conducted any experiments on this so far.

We have used several different feature point operators and feature extractors, and experimented with the effect of different feature types on the learning capacity of the models. First, we have compared the models in terms of image labelling performance. We have observed that the discriminative model is not effected very much when different feature types are used and the model performs the best with DoG-LJ+C (DoG operator with local jet and colour features). Random patches with PCA coefficients and colour features caused the worst performance for the discriminative model, while the opposite results are observed for the generative model. The performance of the generative model depends significantly on the choice of feature types, and the best performance is obtained with random features. We also compared the models in terms of patch labelling. In all cases the generative model outperforms the discriminative model in patch labelling. But the best patch labelling performance is obtained with DoG-LJ+C feature for both models. This is a very reasonable result because DoG operator extracts uniform regions as patches and in most cases a patch is either fully background or fully foreground. However in other cases most of the time, a patch may contain some foreground pixels as well as background pixels. In randomly selected patches this is more serious.

We have also compared the two models when different number of images are used for training. When this comparative experiment is performed using random patches as features, we have observed that with small number of data the generative model performs better than the discriminative model and as the data quantity increases the performances for both models increase but this increase is more marked for the discriminative model, so that the performance of the two approaches is similar for large data sets. When this comparative experiment is performed using DoG-LJ+C features, both models behaved nearly the same for all data quantities but the discriminative model performs better all the time as we increase the data quantity.

Our investigations suggest that the most fruitful approaches will involve some combination of generative and discriminative models. Indeed, this is already found to be the case in speech recognition where generative hidden Markov models are used to express invariance to non-linear time warping, and are then trained discriminatively by maximizing mutual information in order to achieve high predictive performance.

One promising avenue for investigation is to use a fast discriminative model to locate regions of high probability in the parameter space of a generative model, which can subsequently refine the inferences. Indeed, such coupled generative and discriminative models can mutually train each other, as has already been demonstrated in a simple context in [13].

One of the limitations of the techniques discussed here is the use of interest point detectors that are not tuned to the problem being solved (since they are hand-crafted rather than learned) and which are therefore unlikely in general to focus on the most discriminative regions of the image. Similarly, the invariant features used in our study were hand-selected. We expect that robust recognition of a large class of object categories will require that local features be learned from data.

Classifying individual patches is very hard because patches from different classes may seem similar due to the effects of illumination, pose, noise or similarity. This ambiguity can be solved by modeling the interactions between patches. The contextual information can be used in the form of spatial dependencies in the images. Markov Random Field models are traditional interaction models used in vision because they can incorporate spatial relationship constraints in a principled manner. For the purposes of this study we have ignored spatial information regarding the relative locations of feature patches in the image. However, most of our conclusions remain valid if a spatial model is combined with the local information provided by the patch features.

Acknowledgements

We would like to thank Antonio Criminisi, Geoffrey Hinton, Fei Fei Li, Tom Minka, Markus Svensen and John Winn for numerous discussions. The idea of replacing noise OR with with sum of log-odds arose in a discussion with Geoffrey Hinton.

References

1. K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
2. C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
3. G. Bouchard and B. Triggs. The trade-off between generative and discriminative classifiers. In *COMPSTAT*, 2004.

4. G. Dorko and C. Schmid. Selection of scale invariant parts for object class recognition. In *ICCV*, 2003.
5. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale invariant learning. In *CVPR*, 2003.
6. B. M. ter Haar Romay, L. M. J. Florach, A. H. Salden, and M. A. Viergever. Representation of local geometry in the visual system. *Biological cybernetics*, 55:367–375, 1987.
7. A. Holub and P. Perona. A discriminative framework for modelling object classes. In *CVPR*, 2005.
8. T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
9. J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
10. S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.
11. D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
12. K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.
13. R. Neal P. Dayan, G. E. Hinton and R. S. Zemel. The helmholtz machine. *Neural Computation*, pages 1022–1037, 1995.
14. A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, 2002.
15. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Net- works of Plausible Inference*. Morgan Kaufmann Publishers, 1998.
16. I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. In *CVPR*, 2005.
17. M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, 2003.
18. A. Bar-Hillel, T. Hertz, and D. Weinshall. Object class recognition by boosting a part-based model. In *CVPR*, 2005.
19. L. Xie and P. Perez. Slightly supervised learning of part-based appearance models. In *IEEE Workshop on Learning in CVPR*, 2004.