Efficient Learning of Relational Object Class Models

Aharon Bar Hillel Tomer Hertz Daphna Weinshall
School of Computer Science and Engineering and the Center for Neural Computation
Hebrew university of Jerusalem, Israel 91904
{ aharonbh, tomboy, daphna}@cs.huji.ac.il

Abstract

We present an efficient method for learning part-based object class models. The models include location and scale relations between parts, as well as part appearance. Models are learnt from raw object and background images, represented as an unordered set of features extracted using an interest point detector. The object class is generatively modeled using a simple Bayesian network with a central hidden node containing location and scale information, and nodes describing object parts. The model's parameters, however, are optimized to reduce a loss function which reflects training error, as in discriminative methods. Specifically, the optimization is done using a boosting-like technique with complexity linear in the number of parts and the number of features per image. This efficiency allows our method to learn relational models with many parts and features, and leads to improved results when compared with other methods. Extensive experimental results are described, using some common bench-mark datasets and three sets of newly collected data, showing the relative advantage of our method.

1 Introduction

One of the important organization principles of object recognition is the categorization of objects into object classes. Humans learn to categorize objects into classes from an early age, and usually begin by learning "basic categories", such as balls or chairs [14]. Categorization is a hard learning problem due to the large inner-class variability of object classes, in addition to the "common" object recognition problems of varying pose and illumination. Recently, there has been a growing interest in the task of object class recognition [11, 10, 3, 2] which can be defined as follows: given an image, determine whether the object of interest appears in the image (and perhaps also provide its location).

Following previous work [1, 11], in this paper we represent an object using a part-based model (see Fig. 1). Such models can capture the essence of an object class, since they represent both parts' appearance and invariant relations of location and scale between the parts. Part-based models are somewhat resistant to various sources of variability such as within-class variance, partial occlusion and articulation, and they may be convenient for indexing in a more complex system.

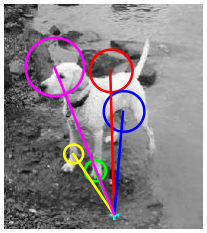

Figure 1. Dog image with our learnt model drawn on top. Each circle represents a part in the model. The parts relative location and scale are related to one another through a hidden center (better viewed in color).

Part-based approaches to object class recognition can be crudely divided into two types: (1) 'generative-model-based' methods (e.g., [11]) and (2) 'discriminative-model-free' methods (e.g., [2]). In the 'Generative-model based' approach a probabilistic model of the object class is learnt by likelihood maximization. The likelihood ratio test is used to classify new images. The main advantage of this approach is the ability to model relations between object parts. In addition, domain knowledge can be incorporated into the model's structure and priors [9]. 'Discriminative-model-free' methods seek a classification rule which discriminates object images from background images. The main advantage of discriminative methods is the direct minimization of a classification-based error function, which typically leads to superior classification results [4]. Additionally since these methods are model-free, they are usually computationally efficient.

In our current work, we try to enjoy the benefits of both worlds: The modeling power of the generative approach, with the accuracy and efficiency of discriminative optimization. We present a novel method for object class recognition, based on discriminative optimization of a simple generative object model. Specifically, we use a compact star-like Bayesian network as our generative model, and extend current discriminative boosting techniques to enable param-

eter optimization of this model. This combination provides some benefits which are not available in the purely generative or discriminative frameworks. Thus, in the framework of generative object modeling, our discriminative optimization allows - for the first time - efficient learning from unsegmented images, with complexity linear in P and N_f, the number of model parts and the number of features per image respectively. This is in sharp contrast to the $O(N_f^P)$ complexity of maximum-likelihood estimation [11], which remains essentially exponential even when a star-like relational model is used [12]. It also improves the behavior of feature selection during learning. From the discriminative perspective, a classifier based on a generative model allows for the natural treatment of spatial relations between model parts, which are not easily incorporated into current discriminative techniques.

In an earlier work [1] we considered discriminative optimization via boosting of a very simple generative model, in which parts were assumed to be independent, and only the parts's appearance was modeled (i.e without considering any relations between the parts). Here we extend the generative model to include dependencies between parts, modeling both parts' location and scale. The model, described in Section 2.2, includes a hidden variable to represent the object's center, and the location and scale of each part depend only on this hidden variable. Parts are therefore conditionally independent given the location of the 'hidden center'. In section 3 we show how to modify a boosting like algorithm in order to learn a model in which parts are only conditionally independent of one another. Unlike the boosting technique used in [1], which views boosting as gradient descent in function space [8], the modified boosting presented here is based on a new simpler view of boosting as gradient descent. Our final algorithm is a boosting extension with some elements from traditional gradient descent techniques.

In order to compare our algorithm to the previously suggested state-of-the-art generative and discriminative methods, we used the benchmark datasets used by both [11] (our generative competitor) and [2] (our discriminative competitor). Results are shown in Section 4, showing the advantage of our algorithm over both competitors. Our algorithm's performance becomes competitive even with a small number of parts, at low computational costs.

To further test our method, we collected three more challenging datasets containing images of chairs, dogs and humans, with matching backgrounds. We used these datasets to test the algorithm's performance under harder conditions: high visual similarity between object and background, and large pose and scale variability. We investigated the relative contribution of the appearance, location and scale components of our model, and showed the importance of incorporating relations between object parts. We experimented with a generic interest point detector [15], as well as with a discriminative interest point detector [5], and our results show a small advantage for the later.

2 A generative model

We represent an input image using a set of local descriptors obtained from an interest point detector. Some details regarding this process are given in Section 2.1. We then define a classifier over such sets of features using a generative object model. The model and the resulting classifier are described in Sections 2.2 and 2.3 respectively.

2.1 Feature extraction and representation

Our feature extraction and representation scheme mostly follows the scheme used in [11, 1]. Initially, images were rescaled to have uniform horizontal length of 200 pixels. We experimented with two feature detectors: (1) Kadir and Brady (KB) [15], and (2) Gao and Vasconcellos (GV) [5][1]. The KB detector is a generic detector that searches for circular regions of various scales, corresponding to the maxima of an entropy based score in scale space. The GV detector is a discriminative saliency detector, which searches for features that permit optimal discrimination between the object class and the background class. Given a set of labeled images from two classes, the algorithm finds a set of discriminative filters based on the principle of Maximal Marginal Diversity (MMD). It then identifies circular salient regions at various scales by pooling together the responses of the discriminative filters.

Both detectors produce an initial set of thousands of salient candidates for a typical image. We select a subset of N_f high scoring features with limited overlap (in our experiments, N_f varied from 13 to 228). Fig. 2(left) presents a set of 75 features detected using the KB detector. The selected regions are represented using the first 15 DCT coefficients (not including the DC) of a 11×11 subsample of the image patch. We concatenate 3 additional dimensions to the descriptor of each patch (or feature), corresponding to its x and y image coordinates and its scale respectively.

Each image I is therefore represented by an unordered set $F(I)$ of 18-dimensional vectors. Since our suggested algorithm's runtime is only linear in the number of image features, we can represent each image using a large pool of features, typically in the order of several hundred features per image. Note that purely generative methods typically use only 20 [11] or 40 [12] features , due to their high learning complexity.

[1]We thank Dashan Gao for making his code available to us, and providing useful feedback.

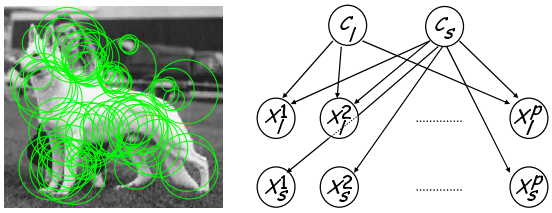

Figure 2. Left: Output of the KB interest point (or feature) detector, marked with green circles. Right: a Bayesian network specifying the dependencies between the hidden variables C_l, C_s and the parts scales and locations X_l^k, X_s^k for $k = 1, .., P$. The part appearance variables X_a^k are independent, and so they do not appear in this network.

2.2 Model structure

We consider a part-based model, where each part in a specific image I_i corresponds to a patch feature from $F(I_i)$. Denote the appearance, location and scale components of each vector $x \in F(I)$ by x_a, x_l and x_s respectively (with dimensions 15,2,1), where $x = [x_a, x_l, x_s]$. We can assume that the appearance of different parts are independent, but this is obviously not the case with the parts' scale and location. However, once we align the object instances with respect to location and scale, the assumption of part location and scale independence becomes reasonable. Thus we introduce a 3-dimensional hidden variable $C = (C_l, C_s)$ stating the location of the object and its scale. Our assumption is that locations and scales of different parts are conditionally independent given the hidden variable C, and so the joint distribution decomposes according to the graph in right panel of Fig 2.

For a model with P parts, the joint probability of $\{X^k\}_{k=1}^P$ and C takes the form

$$\mathrm{p}(\{\mathrm{X^k}\}_{\mathrm{k=1}}^{\mathrm{P}}, \mathrm{C}|\Theta) = p(C|\Theta) \prod_{k=1}^{P} p(X^k|C, \theta^k) = \quad (1)$$

$$p(C|\Theta) \prod_{k=1}^{P} \mathrm{p}(\mathrm{X_a^k}|\theta_a^k) p(X_l^k|C_l, C_s, \theta_l^k) p(X_s^k|C_s, \theta_s^k)$$

We assume a uniform probability for C and Gaussian conditional distribution for X_a, X_l, X_s as follows:

$$\begin{aligned} P(X_a^k|\theta_a^k) &= G(X_a^k|\mu_a^k, \Sigma_a^k) \quad (2) \\ P(X_l^k|C_l, C_s, \theta_l^k) &= G(\frac{X_l^k - C_l}{C_s}|\mu_l^k, \Sigma_l^k) \\ P(X_s^k|C_s, \theta_s^k) &= G(\log(X_s^k) - \log(C_s)|\mu_s^k, \sigma_s^k) \end{aligned}$$

where $G(\cdot|\mu, \Sigma)$ denotes the Gaussian density with mean μ and covariance matrix Σ. We index the model components a, l, s as $1, 2, 3$ respectively, and denote the log of these probabilities by $LG(x_j|C, \mu_j, \Sigma_j)$ for $j = 1, 2, 3$.

2.3 A model based classifier

Our input is not an ordered vector of parts, and so we ideally should consider all the $O(N_f^P)$ possible feature vectors that can be composed from the set $F(I)$. In order to compute the likelihood $P(I|M)$, we should average over all these vectors and all the possible values of the hidden variable C. We assume a uniform prior over the possible ordered vectors, and approximate the average as follows

$$\begin{aligned} P(I|M) &= K_0 \sum_C \sum_{\substack{(x^1,..,x^p)\in F(I)^P \\ x^i \neq x^j \, for \; i \neq j}} \prod_{k=1}^{P} P(x^k|C, \theta^k) \\ &\approx K_0 \sum_C \sum_{(x^1,..,x^p)\in F(I)^P} \prod_{k=1}^{P} P(x^k|C, \theta^k) \\ &\approx K_0 \max_C \max_{(x^1,..,x^p)\in F(I)^P} \prod_{k=1}^{P} P(x^k|C, \theta^k) \\ &= K_0 \max_C \prod_{K=1}^{P} \max_{x\in F(I)} P(x|C, \theta^k) \quad (3) \end{aligned}$$

where K_0 is a constant. In the first approximation we allow vectors with repeating features, which weren't allowed before. While not desirable, this approximation is necessary for the decomposition of the maximum operator achieved in the last line. In the second approximation above, the averages are replaced with the likelihood of the best vector and best hidden C. We prefer working with the best single vector since it uniquely identifies the parts, the object's location and its scale.

The decomposition of the maximum achieved is the key to efficient likelihood computation. If we consider N_c possible values for the hidden variable C, the maximum over the $N_c \cdot N_f^P$ arguments can be computed in $O(N_c N_f P)$ operations using this decomposition. However, the parameter optimization of such a model cannot be done by likelihood maximization: if feature repetition is allowed, the ML solution will choose the same (best) part P times. Maximum likelihood learning thus cannot decompose the likelihood by allowing feature repetition, and learning remains exponential even in a simple star model as suggested in [12].

The natural generative classifier compares the LRT statistic to a constant threshold ν, and it therefore requires a backgorund model in addition to the object model. Modeling a general background is clearly difficult, due to the diversity of objects and scenes that do not share simple common features. We hence approximate the background likelihood by a constant. Our LRT based classifier thus becomes

$$\begin{aligned} f(I) &= \log P(I|M) - \log P(I|BG) - \nu' \quad (4) \\ &= \max_C \sum_{k=1}^{P} \max_{x\in F(I)} \log p(x|C, \theta^k) - \nu \end{aligned}$$

3 Discriminative optimization

Given a set of labeled images $\{I_i, y_i\}_{i=1}^N$, we wish to find a classifier $f(I)$ which minimizes the exponential loss

$$L(f) = \sum_{i=1}^{N} \exp(-y_i f(I_i)) \tag{5}$$

This is the same loss minimized by the the Adaboost algorithm [13]. In Section 3.1 we learn a classifier of the form (4) using a variant of the boosting technique. We show that boosting can be naturally extended to handle classifiers of this form, despite the dependencies between parts due to the hidden variable C. In Section 3.2 we consider the optimization from a more general viewpoint of gradient descent, and present an algorithm (see Algorithm 1) which includes several enhancements to the pure boosting technique.

3.1 Boosting

Boosting is a common method which learns a classifier of the form $f(x) = \sum_{k=1}^{p} \alpha^k h^k(x)$ in a greedy fashion. Several papers [6, 8] have presented boosting as a greedy gradient descent of some loss function. In particular, the work of [8] has shown that the Adaboost algorithm [13] can be regarded as a greedy gradient descent of this loss in L^2 function space.

We suggest a simpler way to derive Adaboost, by considering the Taylor expansion of the exp loss. In what follows and throughout this paper, we use superscripts to indicate the boosting round in which a quantity is measured. At the p'th boosting round, we wish to extend the classifier f by $f^p(x) = f^{p-1}(x) + \alpha^p h^p(x)$. We first assume that α^p is infinitesimally small, and consider which weak hypothesis $h^p(X)$ is appropriate under such conditions. Since α^p is small, we can approximate (5) using the first order Taylor expansion. The derivative of $L(f)$ w.r.t. α^p is

$$\frac{dL(f)}{d\alpha^p} = -\sum_{i=1}^{N} \exp(-y_i f(x_i)) y_i h^p(x_i) \tag{6}$$

We denote $w_i = \exp(-y_i f(x_i))$, and derive the following Taylor expansion

$$L(f^p) \approx L(f^{p-1}) - \alpha^p \sum_{i=1}^{N} w_i^{p-1} y_i h^p(x_i) \tag{7}$$

Assuming $\alpha^p > 0$, the steepest descent of $L(f)$ is gained for some weak hypothesis h^p which maximizes $\sum_{i=1}^{N} w_i^{p-1} y_i h^p(x_i)$, and this maximization is the task of the weak learner. After the determination of $h^p(x)$, the coefficient α^p is determined by the direct optimization of the loss (5). This can be done in closed form only for weak hypotheses with the range of $\{1, -1\}$. In the general case numeric methods are employed, such as line search [13].

In order to derive a similar algorithm in our case, we cast our proposed classifier (4) in an equivalent form. Following [1], we re-parameterize the log-Gaussians to have a fixed covariance determinant of 1 and multiplicative coefficients. This leads to the following parametrization of (4)

$$f(I) = \max_{C} \sum_{k=1}^{P} \alpha^k h^k(I, C) - \nu \tag{8}$$

where for $k = 1, .., P$

$$h^k(I, C) = \max_{x \in F(I)} \sum_{j=1}^{3} \frac{\lambda_j^k}{\sum_{j=1}^{3} \lambda_j^k} LG(x_j^k | c, \mu_j^k, \Sigma_j^k) \tag{9}$$
$$|\Sigma_j^k| = 1 \ , \ \lambda_j^k > 0 \quad j = 1, 2, 3$$

In this parametrization α^k, which replaces the scale of the covariance matrix, can be thought of as the weight of hypothesis h^k, while $\lambda_i / \sum_{j=1}^{3} \lambda_j$ measures the relative weights of the appearance, location and scale components. In order to allow tractable maximization over C, we discretize it and consider only a finite grid of locations and scales with N_c possible values.

We can now derive a greedy loss minimization algorithm using (7) and the subsequent discussion. Denote the accumulated log-likelihood $ll(I, C) = \sum_{k=1}^{p} \alpha^k h^k(I, C)$ and $C^* = \arg\max_C ll(I, C)$. The derivative of $L(f)$ w.r.t. α_p is now

$$\frac{dL(f)}{d\alpha^p} = -\sum_{i=1}^{N} w_i y_i h^p(I_i, C_i^*) \tag{10}$$

and using the Taylor expansion we get

$$L(f^p) = L(f^{p-1}) - \alpha^p \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1}) \tag{11}$$

In analogy with the discussion following (7), the weak learner should now get as input $\{w_i^{p-1}, C_i^{*,p-1}\}_{i=1}^N$ and try to maximize the score $S(h^p) = \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1})$. This task is not essentially harder then the weak learner's task in regular boosting, since it 'assumes' that the value of the hidden variable C is known and set to its optimal value according to the previous hypotheses. In the first boosting round, when $C^{*,p-1}$ is not really defined, we only train the appearance component of the hypothesis. The relational components of this part are set to have low weights and default values.

Choosing α^p after the hypothesis $h^p(I, C)$ has been chosen is trickier than in standard boosting. First, it is more

computationally demanding since a change in the value of α^p requires to recompute the maximum over C as part of the classifiers estimation. Another difference lies in the existence of the threshold parameter ν in our model. Instead of searching for the value of α^p alone, we should search for the best update of both α^p and ν together. However, in [1] we show a closed form formula for the optimal ν. We then use a gradient descent procedure to optimize α^p, as shown in steps 2,3 of Alg. 1. Since the gradient of α^p in (10) depends on $\{C_i^*\}_{i=1}^N$ and $\{w_i\}_{i=1}^N$, we iterate in step 3 of Alg. 1 between gradient steps, inference of $\{C_i^*\}_{i=1}^N$, and updates of $\{w_i\}_{i=1}^N$. This loop must be preceded by the computation of the messages $h(i,c)$ in step 2.

3.2 Gradient descent

In [1] we considered two types of weak learners: selection-based and gradient-based. A selection based learner constructs a part hypotheses based on real image features, and returns the part hypothesis with the highest score $S(h)$. Such weak learners are commonly used in vision tasks [2, 7]. However, in [1] such weak learners were usually outperformed by gradient based learners. A gradient based learner learns a part hypothesis using gradient ascent dynamics on the score $S(h)$. The weights w_i^{p-1} in the score $S(h^p)$ do not depend on h^p, and so its derivative with respect to the vector of its parameters θ^p is given by the weighted sum

$$\frac{dS(h^p)}{d\theta^p} = \sum_{i=1}^{N} w_i^{p-1} y_i \frac{dh^p(I_i, C_i^{*,p-1})}{d\theta^p} \qquad (12)$$

The gradient dynamics, presented in step 1 of Alg. 1, iterate between gradient steps in θ and re-computation of the best parts and their scores.

In boosting the optimization of a new part is done in two sequential steps: Hypothesis parameters θ^p are optimized, then the mixture coefficient α^p. However, in our case feedback between these two optimizations is plausible, since a change in α^p may induce changes in C^* for some images and can therefore change the optimal $h^p(I, C)$. Such feedback can be introduced by using a more general gradient descent algorithm, constructed by: 1) differentiating $L(f)$ directly instead of its Taylor approximation, and 2) iterating small gradient steps on both α and θ^p in a single loop.

When one considers the update steps required for such a gradient descent algorithm, the derivatives w.r.t α^p and θ^p are similar to the ones used in (10), (12). The only difference is that now the gradient w.r.t θ depends on the weights $\{w_i^p\}$, and hence it is no longer constant w.r.t. h^p and α^p. Exact gradient descent therefore requires the recomputation of w_i at each gradient iteration, which is quite expensive computationally. We have experimented in the continuum between the 'boosting' and the 'gradient descent' approaches using Algorithm 1, which enclose the 'boosting' optimization loops of h^p and α^p in a third feedback loop. Setting the outer loop counter K_1 to 1 we get the booting optimization strategy, while setting K_1 to some large value and $K_2 = 1$,$K_3 = 1$ we can get exact gradient descent. A good complexity and performance trade-off is achieved with a version which is rather close to boosting, with the outer loop repeated several times. Our final optimization algorithm is hence essentially repeated, sequential calls of Algorithm 1 with such a parameter setting.

Algorithm 1 Optimization of part p

Input : $F(I_i)$, y_i , w_i, C_i^* $\quad i = 1, .., N$
$\quad\quad ll(i,c) \quad i = 1, .., N$, $c = 1, .., N_c$

initialize weak hypothesis using a selection learner :
Choose $\theta = \lambda_j, \mu_j, \Sigma_j \quad j = 1, .., 3 \quad , \quad \alpha = 0$
Set $[h(i, C_i^*), x^*(i)] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$

where $g(x,c) = \sum_{j=1}^{3} \frac{\lambda_j}{\sum_{j=1}^{3} \lambda_j} LG(x_j | c, \mu_j, \Sigma_j)$

Loop over $1, 2, 3$ K_1 iterations

1. Loop over a,b K_2 iterations (θ optimization)
   (a) Update weak hypothesis parameters
   $\theta = \theta + \eta \sum_{i=1}^{N} w_i y_i \frac{dg(x_i^*, c_i^*)}{d\theta}$
   (b) Update best part candidates for all images
   $[h(i, C_i^*), x_i^*] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$
2. Compute for all $i, c \quad h(i,c) = \max_{x \in F(I_i)} g(x,c)$
3. Loop over a,b,c K_3 iterations (α optimization)
   (a) Update α : $\quad \alpha = \alpha + \eta \sum_{i=1}^{N} w_i y_i h(i, C_i^*)$
   (b) Update hidden center for all images
   $[f^0(I_i), C_i^*] = \max, \arg\max_c ll(i,c) + \alpha h(i,c)$
   (c) Update $f(I_i)$ and the weights
   $\nu = \frac{1}{2} \log \left[\frac{\sum_{\{i: y_i = -1\}}^{N} \exp(f^0(I_i))}{\sum_{\{i: y_i = 1\}}^{N} \exp(-f^0(I_i))} \right]$
   $f(I_i) = f^0(I_i) - \nu$
   $w_i = exp(-y_i f(I_i))$

Set $ll^p(i,c) = ll(i,c) + \alpha h(i,c)$
Return θ, w_i, C_i^*,$ll^p(i,c) \quad i = 1, ..N \quad c = 1, .., N_c$

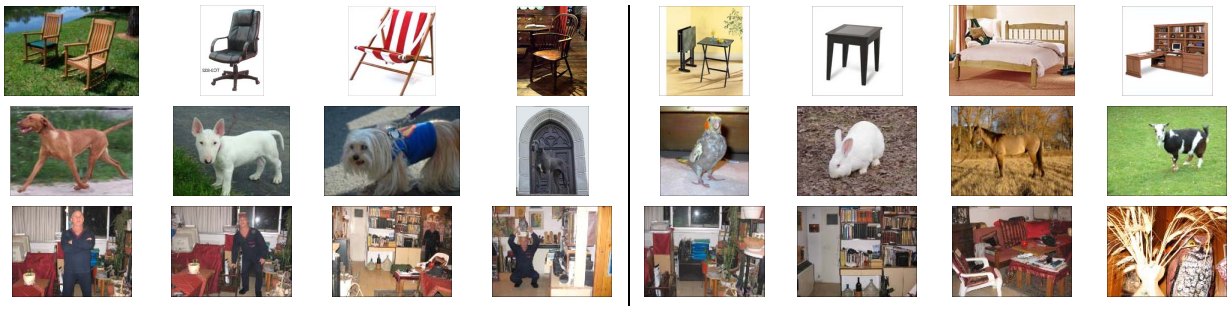

Figure 3. Images from the Chairs, Dogs and Humans datasets and their corresponding backgrounds. Object images appear on the left, background images on the right. In the second row, the two leftmost background images are of 'easy animals' and next are two 'hard animals' images. In the third row, the two leftmost object images belong to the easier image subset. The next two images are hard due to the person's scale and pose.

4 Experimental results

Datasets For comparison with other methods we used the Caltech datasets [11], which are publicly available. These datasets contain relatively small variance in scale and location, and the background images do not contain objects similar to the class objects. In order to test the algorithm under harder conditions, we compiled 3 new datasets with matching backgrounds.[2] These datasets contain images of Chairs (800 images), Dogs (500) and Humans (593).

In the Chairs and Dogs datasets, the objects are seen roughly from the same pose, but include large inner class variability, as well as some variability in location and scale. For the Chairs dataset we compiled a background dataset of Furniture which contained images of tables, beds and bookcases (200,200,100 images respectively). When possible (for tables and beds) images were aligned to a viewpoint isomorphic to the viewpoint of the chairs. As background for the Dogs dataset, we compiled two animal datasets: 'Easy Animals' contains 500 images of animals not similar to Dogs; 'Hard Animals' contains 250 images from the 'Easy Animals' dataset, and an additional 250 images of four-legged animals (horses, goats, etc.) in a pose isomorphic to the Dogs.

The Humans dataset was designed to include large variability in location, scale and pose - each person was photographed standing in 4 different scales (each 1.5 times larger than its predecessor), at various locations and with several articulated poses of the hands and legs. For this dataset we created a background dataset of 593 images which contain the sites in which the Humans images were taken. Fig. 3 shows a few images from our datasets.

Algorithm parameters In the experiments reported below we constructed models with up to 60 parts using Algorithm 1 with control parameters of $K_1 = 60, K_2 = 100, K_3 = 4$. Each image was represented using at most $N_f = 200$ features (KB detector) or $N_f = 240$ features (GV detector). The hidden location center values were equally spaced using a grid of 6×6 locations over the image. The hidden scale center had a single value, or 3 different values with a ratio of 0.63 between successive scales, resulting in a total of $N_c = 36, 108$ values respectively. In our experiments we set the covariance of the appearance and location models to σI because we discovered that covariance matrices estimation tended to overfit. We randomly selected half of the images from each dataset for training and used the remaining half for testing.

The learnt models Examples of the learnt models can be seen in Fig. 4. Most of the learnt parts have clear semantics in terms of object's parts. For example in the dog model we can clearly identify parts that correspond to the head, back, legs (both front and back), and the hip. The location models are gross, but clearly useful. In some cases both the appearance of parts and their modeled locations are exaggerated to enhance discriminative power.

Benchmark results In Table 1 we compare our results to those obtained by a purely generative approach [11][3] and a purely discriminative one [2][4] using the Caltech dataset. Both methods learn from an unordered set of features, obtained using an interest point detector. Following [11], the motorbikes, airplanes and faces datasets were tested against office background images, and the Cars rear dataset was tested against road background images. To allow for a clear comparison with [11], we used their exact train and test indexes and the same feature detector (KB). Our results were

[2]The datasets are available at `http://www.cs.huji.ac.il/~aharonbh/`.

[3]Note that the results reported in [11] (except for the cars data base) were achieved using manually scale-normalized images, while our method did not rely on any such rescaling.

[4]In an unpublished manuscript, this approach was reported to give better results using segmentation based features. We did not include these results since we wanted to compare the different learning algorithms using similar features.

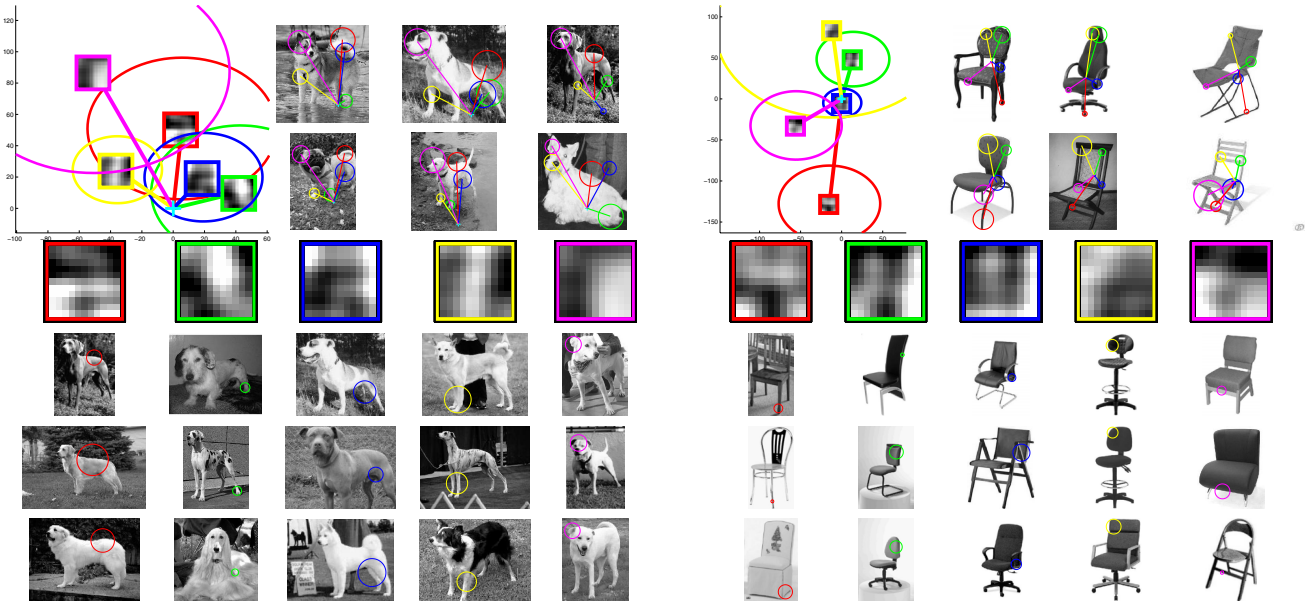

Figure 4. 5 parts from the dog and chair models. The top left drawing shows the spatial models of the 5 parts. Each part's mean location is surrounded by the 1 std line. The cyan cross indicates the location of the hidden 'center'. The top right pictures show dog/chair test images with the model implementation found. All dogs/chairs were successfully identified except for the one on the right-bottom corner. Below each model, the parts's mean appearance patches are shown. The last three rows present the 3 best scoring implementations of these parts, across all test images. Each column presents the implementations of the part shown above the column. The parts have clear semantic meaning, and repetitive locations. Most other parts behave similarly to the ones shown.

obtained without modeling scale, since it did not improve classification results when using the KB detector. This may be partially explained by noting that the Caltech datasets contain relatively small variance in scale. Error rates for our method were computed using the threshold learnt by our boosting algorithm. Results are presented for models with 7 parts (the number of parts used by [11]) and 50 parts. When 7 parts are used, our results are comparable to those of [11]. However, when 50 parts are used our algorithm outperforms both competitors in all but a single case.

Data Name	Our model 7 parts	Our model 50 parts	Fergus et. al	Opelt et. al
Motorbikes	7.8	4.9	7.5	7.8
Cars Rear	1.2	0.6	9.7	–
Airplanes	8.6	6.7	9.8	11.1
Faces	9.5	6.3	3.6	6.5

Table 1. Test error rates over the Caltech dataset of our method using 7 and 50 parts compared to a generative model approach [11] and a discriminative model-free boosting approach [2]. Algorithm's parameters were held constant across all experiments.

The importance of using location and scale models Table 2 shows a comparison of the test results when varying the model complexity. Specifically we present results when using only an appearance model, and when adding location and scale models. In this experiment we used features extracted using the GV detector [5]. We can see that although the appearance model produces very reasonable results, adding a location model significantly improves performance. The additional contribution of the scale model is only minor. Additionally, by comparing the results of our full blown model (A+L+S) to those presented in Tables 1,3, we can see that the GV detector usually provides somewhat better results than those obtain using the KB detector.

Data Name	A	A+L	A+L+S
Motorbikes	8.1	3.2	3.51
Cars Rear	4.0	1.4	0.6
Airplanes	15.1	15.1	12.1
Faces	6.1	5.2	3.8
Chairs	16.3	10.8	10.9

Table 2. Errors rates using models of varying complexity. (A) Appearance model alone. (A+L) Appearance and location models. (A+L+S) Appearance,location and scale models. Algorithm's parameters were held constant across all experiments.

Challenging datasets We used the Chairs and Dogs datasets to test the sensitivity of the algorithm to visual similarity between object and background images. We trained the Chairs dataset against the Caltech office background dataset, and against the furniture dataset described

above. The Dogs dataset was trained against 3 different backgrounds datasets: Caltechs 'office' background, 'Easy Animals' and 'Hard Animals'. The results are summarized in Table 3. As can be seen, our algorithm works well in cases where there are large differences between the object and background images. However, it fails to discriminate, for example, dogs from horses.

We used the Humans dataset to test the algorithm's sensitivity to variations in scale and object articulations. In order to obtain reasonable results on this hard dataset we had to reduce scale variability to 2 scales and restrict the variability in pose to hand gestures only - we denote this dataset by 'Humans restricted' (355 images).

Data	Background	Test Error
Chairs	Office	2.23
Chairs	Furniture	15.53
Dogs	Office	8.61
Dogs	Easy Animals	19.0
Dogs	Hard Animals	34.4
Humans	Sites	34.3
Humans rest.	Sites	25.9

Table 3. Error rates obtained on our new datasets of Chairs, Dogs and Humans. Results were obtained using the KB detector.

Large numbers of parts and features When hundreds of features are used per image, many features lie in the background of the image, and learning good parts requires good 'feature selection' behavior. Fig. 5 presents error rates as a function of parts and features number. Significant performance gains are obtained by scaling up this quantities, indicating that the algorithm is able to find good part models even in the presence of large amount of clutter features. This behavior should be contrasted with the generative learning of a similar model in [12], where increased numbers of parts and features do not in general lead to improved performance. Intuitively, maximum likelihood learning choose to model features which are frequent in object images, even if these are simple clutter features from the background, while discriminative learning naturally tend to selects only discriminative parts.

5 Conclusion

We have presented an object class recognition method, based on discriminative boosting-oriented optimization of a simple relational generative model. The method combines the natural treatment of spatial part relations, typical to generative classifiers, with the efficiency and the feature selection quality of discriminative systems. Our experiments show that the method indeed enjoys the benefits of geometrical modeling on the one hand, and the large numbers of parts and features on the other, and that it compares favorably with recent purely generative or purely discriminative systems.

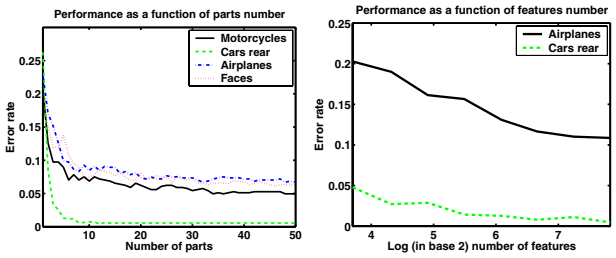

Figure 5. Left: Error rate as a function of the number of parts P in the model on the Caltech datasets for $N_f = 200$. Right: Error rate as a function of the number of image features N_f on Cars rear (easy) and Airplanes (Relatively hard) Caltech datasets, with $P = 30$. The X axis varies between 13 and 228 features in log scale. All results here were obtained using the KB detector

References

[1] Bar Hillel A., Hertz T., and Weinshall D. Object class recognition by boosting a part based model. In *CVPR*. IEEE Computer Society, 2005.

[2] Opelt A., Fussenegger M., Pinz A., and Auer P. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004.

[3] Shivani A. and Roth D. Learning a sparse representation for object detection. In *ECCV*. Springer, 2002.

[4] NG A.Y. and Jordan M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 2001.

[5] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. In *NIPS*, 2004.

[6] Friedman J. H., Hastie T., and Tibshirani R. Additive logistic regression: a statistical view ofboosting. *Ann. Statist.*, 28:337–407, 2000.

[7] Murphy K.P., Torralba A., and Freeman W. T. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *NIPS*, 2003.

[8] Mason L., Baxter J., Bartlett P., and Frean M. Boosting algorithms as gradient descent in function space. In *NIPS*, pages 512–518, 2000.

[9] F.F. Li, R. Fergus, and Perona P. A bayesian approach to unsupervised one shot learning of object catgories. In *ICCV*, 2003.

[10] Vidal-Naquet M. and Ullman S. Object recognition with informative features and linear classification. In *ICCV*, 2003.

[11] Fergus R., Perona P., and Zisserman A. Object class recognition by unsupervised scale invariant learning. In *CVPR*. IEEE Computer Society, 2003.

[12] Fergus R., Perona P., and Zisserman A. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.

[13] Schapire R.E. and Singer Y. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[14] Eleanor Rosch. Basic objects in natural categories. *Cognitive Psychology*, 8:382–349, 1976.

[15] Kadir T. and Brady M. Scale, saliency and image description. *IJCV*, 45(2):83–105, November 2001.