

A Discriminative Framework for Modelling Object Classes

Alex Holub and Pietro Perona
Computation and Neural Systems
California Institute of Technology
Pasadena, CA 91125
holub@caltech.edu, perona@caltech.edu

Abstract

Here we explore a discriminative learning method on underlying generative models for the purpose of discriminating between object categories. Visual recognition algorithms learn models from a set of training examples. Generative models learn their representations by considering data from a single class. Generative models are popular in computer vision for many reasons, including their ability to elegantly incorporate prior knowledge and to handle correspondences between object parts and detected features. However, generative models are often inferior to discriminative models during classification tasks. We study a discriminative approach to learning object categories which maintains the representational power of generative learning, but trains the generative models in a discriminative manner. The discriminatively trained models perform better during classification tasks as a result of selecting discriminative sets of features. We conclude by proposing a multi-class object recognition system which initially trains object classes in a generative manner, identifies subsets of similar classes with high confusion, and finally trains models for these subsets in a discriminative manner to realize gains in classification performance.

1 Introduction

Humans can easily recognize and distinguish thousands of visual categories. The best computer algorithms achieve only a fraction of human performance in terms of both the number of classes recognized and the accuracy in distinguishing between those classes. The impressive performance of the human system can be ascribed to both our ability to recognize the overall appearance of objects, and to detect subtle differences between very similar object class categories, such as the difference between male and female faces or mopeds and motorcycles.

Algorithms for learning representations of object class categories can be roughly grouped into two separate paradigms: Generative [1, 2, 3, 4, 5] and Discriminative [6, 7, 8,

9]. Generative object recognition algorithms create object class models using only the data of the class to be modelled. A significant benefit of generative models is their ability to elegantly handle missing data problems. In a local-feature based object recognition context a particularly important missing data problem is the mapping of detected features to object parts. In addition, generative methods tend to allow for elegant integration of prior knowledge [10]. Finally, generative techniques are well suited for modelling large numbers of object categories, as they easily allow for the introduction of new object classes. However, by not taking account the statistics of similar classes, these models can perform poorly when asked to classify similar object categories. Discriminative object recognition techniques, on the other hand, utilize the data from multiple object classes to create classifiers. SVMs [11] and Boosting [6] are examples of common discriminative techniques. Such methods tend to outperform their generative counterparts, but do not, in general, provide easy methods for handling missing data and incorporating prior knowledge. Our goal is to create object class models which take advantage of both the flexibility provided by generative methods and the classification performance increases provided by discriminative learning.

In this paper we use a principled probabilistic approach to extend the generative ‘Constellation Model’ [1, 2] framework for creating object class models to a discriminative setting. We directly optimize the conditional distribution while maintaining an underlying generative model (this approach is also known as maximizing the Conditional Likelihood or CL). Utilizing a generative framework in conjunction with a discriminative optimization has been previously proposed by other authors [12, 13, 14]. These studies do not observe substantial gains in using CL over generative approaches on traditional learning systems data-sets such as the UCI data-sets. One of the objectives of this study is to assess the performance of the discriminative method for local-feature based visual object recognition.

We conclude by proposing a general approach to object recognition which utilizes the relative merits of both the generative and discriminative learning paradigms. In this

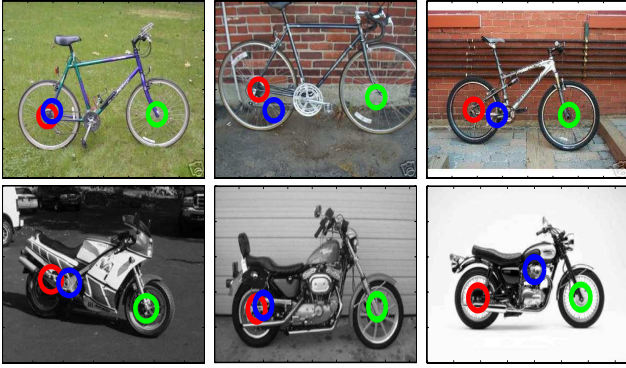


Figure 1: Examples of generative Bicycle and Motorcycle models. The circles indicate the positions of the best hypothesis. The generative models tend to model the wheels, which are similar in their relative positions and appearance for both classes.

system, models are initially trained in a generative fashion. Next, subsets of classes with high confusion are identified, which indicates similarity between classes. Each subset is then trained in a discriminative manner. The result is a set of class representations that model the unique aspects of all the classes.

This paper is organized as follows: First we will review the Constellation Model. We will then outline our discriminative learning approach. Finally, we will show examples of both generative and discriminative models, highlight their differences, and illustrate a system which utilizes both techniques.

2 Review of the Constellation Model

Our approach to object class modelling builds on earlier work by Weber et al. [1] and more recently that of Fergus et al. [2]. In this framework, an object is modelled by a ‘constellation’ of several parts, with each model containing information on both the appearance and relative position of each part. In the following experiments we use a simplified version of the constellation framework, which utilizes only three parts and does not include an explicit model for occlusion or relative scale. We use gaussian probability densities to represent the variations in appearance and shape of the three components. The parameters for our models are learned by extracting interesting features from a set of training images and using these features to maximize a model representation.

2.1 Feature Detection and Representation

Interesting locations, referred to as *features* or *interest points*, must be identified within all images. We accomplish

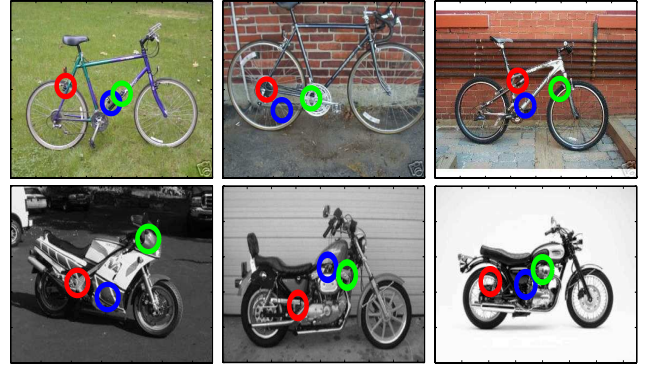


Figure 2: Examples of discriminative Bicycle and Motorcycle models. The best hypotheses tend to model the body and frame of both the bicycle and motorcycle classes respectively. The locations of the best hypotheses appear less consistent than in the ML models.

feature detection using either supervised or semi-supervised learning. For supervised learning, we manually select registered regions of images for learning from labelled images. In semi-supervised learning, the feature detection process is accomplished using the Kadir and Brady [15] detector, however the algorithm maintains knowledge of the class labels for each image¹.

For supervised learning, we manually register 9 features for use by our learning algorithm: left eye, right eye, left hairline, right hairline, center hairline, nose tip, left mouth, right mouth, and chin (see Figure 3). When features are missing or occluded we choose the closest position in the image as our feature. A constant scale was assumed for all features across all images for the supervised experiments.

We automate the feature selection process by using a feature detector in semi-supervised learning. The Kadir and Brady feature detector returns the positions, scales, and relative saliency of interest points within an image. The detector is well-tuned for detecting circular regions within images, including eyes and wheels. The interest points with the highest saliency are used as features for learning.

In order to construct an appearance representation for the salient points, we extract 11×11 pixel patches centered on the Kadir and Brady features. We scale the size of the patch extracted to correspond to the scale of detection and then sub-sample the patches to 11×11 matrices. We reduce the number of appearance parameters to be optimized by performing PCA on all patches from every image. We select the first K principal components for our appearance models, where K is typically 10. We use a matrix, A_i^c , of size

¹Note the departure from the terminology of [2] and [10] who consider their algorithms ‘unsupervised’. This distinction becomes particularly important with the advent of purely unsupervised object learning algorithms, i.e. [16]



Figure 3: (Left) Examples of features selected manually for Schwarzenegger. Nine total features are selected. (Center) The same image but with features found using the Kadir and Brady detector. The 15 most salient features are shown.

$F \times K$ to represent the appearances of all features within image i of class c , where F is the number of features used. F typically ranges from 25-30 for semi-supervised learning.

2.2 Shape Representation

We construct a shape model to represent the variations in position for each model part. We record the positions of all interest points within an image i for class c in the variable X_i^c . We attempt to find the optimal mean and variance of gaussian densities for the shape model. The positions of all model parts are relative to the first part. By conditioning on the first model component, the model becomes invariant to translations.

2.3 Generative Model

Here we present a generative framework which obtains maximum likelihood estimates for parameter values of each object class. Our goal is to find a set of model parameters θ_c , where c is a particular class of objects, which optimizes the appearance and relative positions of the patches extracted from images in that class. θ_c represents both the means and diagonal variance components. Consider a set of object classes ranging from $1..C$ and indexed by c , and the images belonging to each of these classes, ranging from $1..N_c$ and indexed by i . We have extracted both appearance, A_i^c , and shape, X_i^c , information from each image I_i^c . We assume that the shape and appearance models are independent of one another and that the images are I.I.D. The log likelihood of the training images given a particular parameter set θ is:

$$\sum_{i \in c} \log(p(I_i^c)) = \sum_{i \in c} \log(p(A_i^c | \theta_c) \cdot p(X_i^c | \theta_c)) \quad (1)$$

The maximum likelihood estimate will find the value of θ_c which optimizes the expression above. Given a particular image I_i^c , we obtain a set of interest points as described above. We must assign an interest point to a particular

model component. Since we do not a priori know which interest point belongs to which model component, we introduce a hypothesis variable h , which maps interest points to model parts. We order the interest points in ascending order of x-position. There are M model parts. This results in a total of $\binom{F}{M}$ unique combinations of interest points and parts, where each hypothesis h will assign a unique interest point to each model part.² We marginalize over the hypothesis variable to obtain the following expression for the log likelihood for a particular class:

$$= \sum_{i \in C} \log \left(\sum_h p(I_i^c, h | \theta_c) \right) \quad (2)$$

$$= \sum_{i \in C} \log \left(\sum_h p(A_i^c, h | \theta_c) \cdot p(X_i^c, h | \theta_c) \right) \quad (3)$$

This generative approach can lead to difficulties when attempting to distinguish between similar object categories. Fig. 1 shows several images from two similar classes, Bicycles and Motorbikes, and the corresponding locations of the best hypothesis. The relative positions and appearance of the parts seem similar between the two classes, leading to confusion during classification tasks.

3 Discriminative Model

Here we consider a discriminative formulation for learning object categories. Similar to the generative models described above, we assume that our models can be described by a parameter vector θ_c . However the discriminative approach maximizes the conditional distribution, $p(\theta_c | I_i^c)$, of all the classes given all the data. By maximizing the Conditional Likelihood (CL) expression, we are maximizing the probability that each image (represented by I_i^c) belongs to its own label (c):

$$\log \left(\prod_c \prod_{i \in c} p(\theta_c | I_i^c) \right) = \sum_c \sum_{i \in c} \left\{ \log(p(I_i^c | \theta_c)) + \log(p(c)) - \log(p(I_i^c)) \right\} \quad (4)$$

Next we expand $p(I_i^c)$ which corresponds to the probability of data-point from the current class belonging to any of the classes. We index the ‘competing’ classes by g . Furthermore, we remove the prior probability of a class, $p(c)$, as it is independent of the parameters we are optimizing, θ_c . We obtain:

²One of the significant limitations of the constellation model is the computational cost induced by the combinatorial explosion relating the number of interest points used and the number of model components.

$$\sum_c \sum_{i \in C} \left\{ \underbrace{\log(p(I_i^c | \theta_c))}_{\text{ML}} - \log \left(\sum_g p(I_i^c | \theta_g) p(g) \right) \right\} \quad (5)$$

The equation for maximizing CL consists of two terms. The first is the maximum likelihood term used in the generative approach, from which we subtract the second term, the probability of a data-point belonging to any other class. Intuitively, data-points are being pulled towards their own class label while being pushed away from other competing classes. Finally, we note that the relative strength of the ML and CL terms can be weighted using a term α , thereby allowing for a continuum between purely ML and purely CL models:

$$\sum_c \sum_{i \in C} \left\{ \underbrace{\log(p(I_i^c | \theta_c))}_{\text{ML}} - \alpha \cdot \log \left(\sum_g p(I_i^c | \theta_g) p(g) \right) \right\} \quad (6)$$

Where $\alpha = 0$ is an entirely ML approach and $\alpha = 1$ a pure CL approach. Varying the value of α is useful for illustrating the relative merits of generative and discriminative techniques. All our object models used a complete discriminative model with $\alpha = 1$ unless otherwise specified.

Returning to the Motorcycle and Bicycle classes (Fig. 2), we notice that the model parts for CL models tend to represent the body of the classes rather than the wheels. The features along the body intuitively seem to provide more discriminative power than the wheels for these object classes.

3.1 Model Testing

The performance of the discriminative models can be assessed by presenting a novel test image, I_i^c , and calculating the probability of this test image being generated by any given class: $p(I_i^c | \theta_c^*)$, where θ_c^* is an optimized CL model. Note that the term representing the competing classes in the discriminative framework optimization, $p(I_i^c)$, is the same for all classes. If the highest probability model corresponds to the class label for that image, the image is correctly classified. For the generative models, we utilize a likelihood ratio between the class models, which is equivalent to finding the model with highest probability for a particular image I_i^c (see [2] for a full derivation of the generative approach).

3.2 Model Optimization

We wish to maximize the expressions for both the generative and discriminative approaches derived above. We use the Expectation Maximization [17] (EM) algorithm to optimize the generative models. Learning is terminated after 100 iterations. The conjugate gradient algorithm is used to

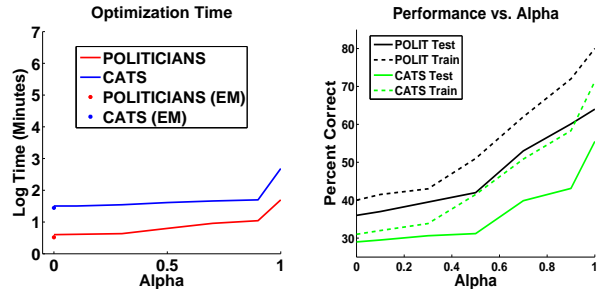


Figure 4: (Left) Optimization time as a function of α . Both ML and CL models are trained using conjugate gradient. At $\alpha = 0$ we have a purely generative approach, and for $\alpha = 1$ a purely discriminative approach. The time scale is in \log_{10} minutes. The CL models take longer to optimize, $6\times$ longer for the Politicians data-set and $10\times$ longer for the Cats data-set. Results shown for both a 9 interest points, 100 train images supervised ‘Politicians’ data-set and 20 interest points, 150 train images Cat Species data-set. Both data-sets contain 4 classes. Red and blue stars indicate the time taken to optimize using the EM algorithm. (Right) Train and Test performance as a function of α for the Politicians data-set (black curves) and the Cat Species data-set (green curves). Increasing the discriminative power increases the performance as well as the amount of over-fitting as measured in the difference between the train and test performance.

optimize the CL models. Conjugate gradient requires the derivative of the CL expression with respect to each parameter of the model. The derivatives can be easily obtained using the following expression:

$$\begin{aligned} \frac{\partial}{\partial \theta_c} \log(p(\theta_c | I_i^c)) = \\ \frac{\partial}{\partial \theta_c} \log p(I_i^c | \theta_c) - \frac{\partial}{\partial \theta_c} \log \sum_g p(I_i^c | \theta_g) p(g) \end{aligned} \quad (7)$$

θ_c represents both the mean and variance for the parts of the model. We also note that the parameters of a class θ_c appear in the background term of all competing classes, and these must be accounted for when taking the derivatives. Learning is terminated when either the gradient at a particular iteration is below a threshold or the maximum number of conjugate gradient iterations is reached. The maximum number of iterations is arbitrarily set to 100. We choose random initial starting conditions for all models to initialize both the EM and conjugate gradient optimization routines.

By varying the value of α we can explore how the computational cost changes as the model becomes more discriminative (see Fig. 4). Our experiments indicate a steep increase in computational time when moving from $\alpha = 0.9$ to $\alpha = 1$ with a corresponding minimal increase in performance, indicating that it might be more computationally favorable to not use the fully discriminative learning model.

The term representing the competing classes in the CL expression introduces many additional local minima to those found in ML. Large combinations of features and parts are computationally prohibitive using the current optimization technique. We compare both the performance and optimization time between the conjugate gradient algorithm with $\alpha = 0$ and the EM algorithm for the ML learned models. The EM algorithm is slightly faster but does not result in a noticeable change in performance.

4 Supervised Discrimination

Our initial experiments are conducted on a data-set consisting of 4 famous politicians. The Politicians data-set contains images of John Kerry, George Bush, Arnold Schwarzenegger, and Bill Clinton. Each data-set contains about 150 images of which typically 100-120 are used for training. The resolution is about 200×200 . See the appendix for information on data collection.

Figure 5 illustrates a typical set of discriminative and generative models for the politicians data-set. There are several interesting things to note. (1) The models found by ML and CL optimization differ as can be seen by the locations for the best fitting hypotheses. (2) The CL appearance models seem to be less homogenous between classes than their ML counterparts as indicated by the clustering of points within the ML appearance models and the general dispersion of points seen in the CL appearance models. (3) Features along the hairline are the highest probability hypotheses for the CL models, while the ML models prefer regions within the face. The increased performance exhibited by the CL models (see Figure 7) indicates that the appearance and mutual positions of features along the hairline is more discriminative.

5 Semi-Supervised Discrimination

We performed experiments on two visually similar sets of classes using semi-supervised learning, Bikes (Motorbikes, Bicycles) and Cat Species (House Cat, Tiger, Lion, Cougar), and one set of dissimilar classes, Human Faces and Airplanes. The Cat classes contained 240 images each with 200 being used for training, while the Bikes data-sets have about 450 images each with 370 used for training. We used a maximum of 30 interest points per image. The Cats data-set often contains very similar looking images, making the discrimination task particularly difficult. Figure 6 compares ML and CL generated models for the Cats data-set. We make several observations: (1) Both the appearance and shape models are more variable between classes for the CL optimized models than the corresponding shape and appearance models of the ML models. (2) The best hypothe-

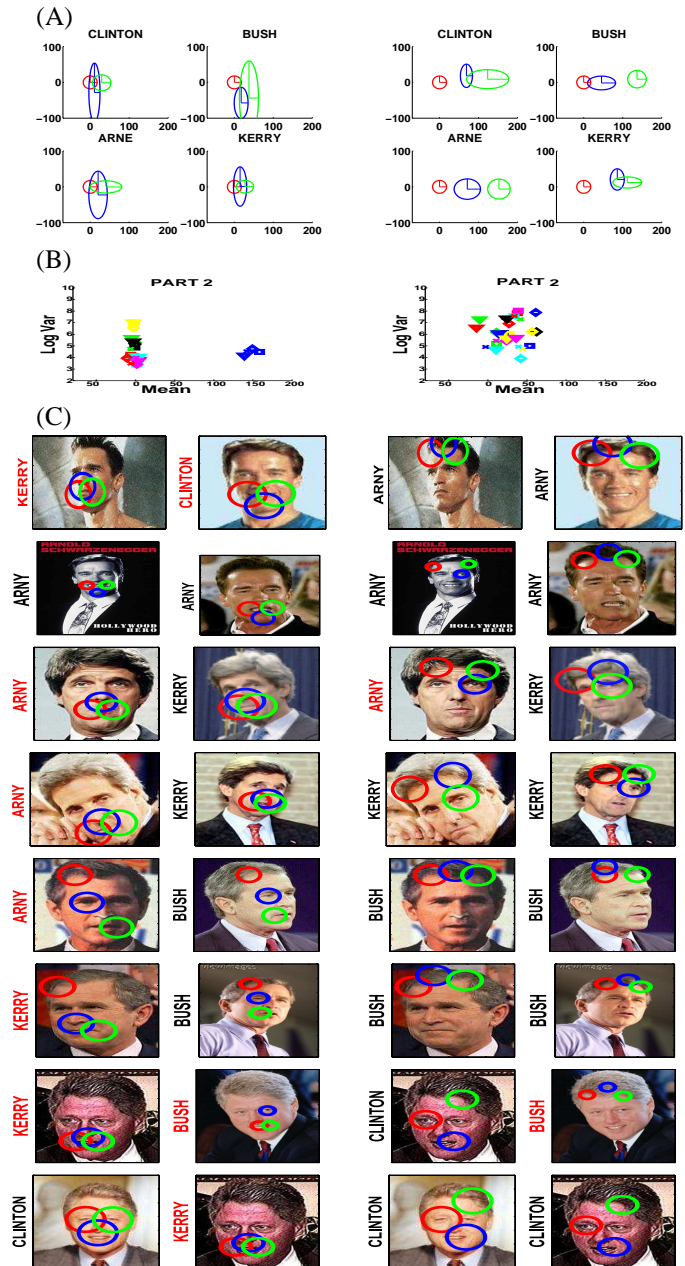


Figure 5: Generative (ML) and Discriminative (CL) Politician Models. Left column ML models, right column CL models. (A) Shape models for each class. The ovals represent the mean and variance of the gaussian model for each part. Each unique color circle represents a different part. (B) Plots of the mean vs the natural log of the variance for the first 7 PCA components. Only Part 2 is shown, although it is representative of the other parts. Each unique shape represents a different class and each color a different PCA coefficient. ML models are more tightly bunched between classes than the CL models. (C) The locations of the best matching hypothesis in the same images for both ML and CL models. Predicted labels are to the left of each image, with incorrect predicted labels in red.

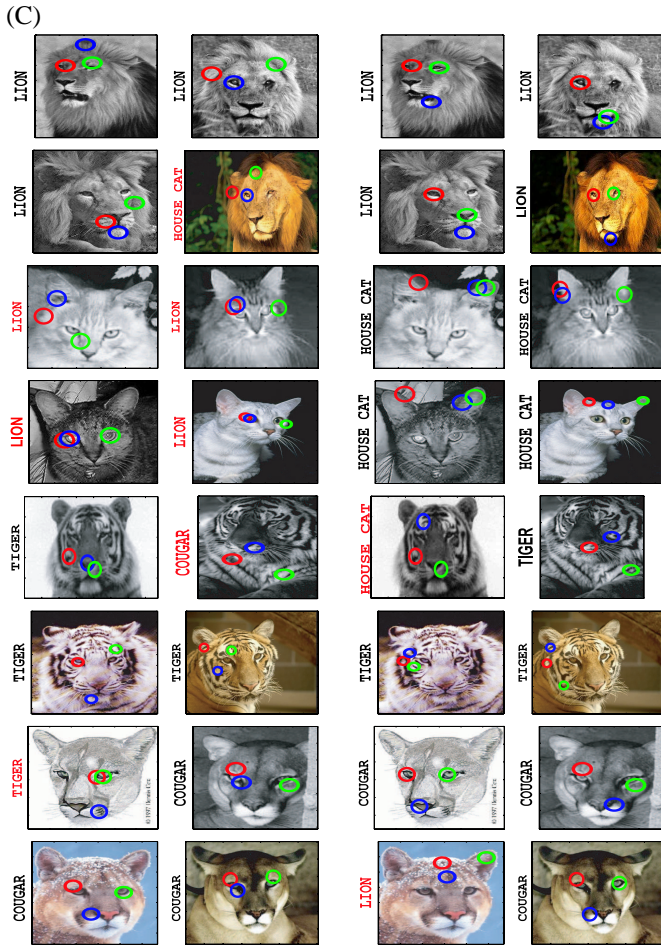
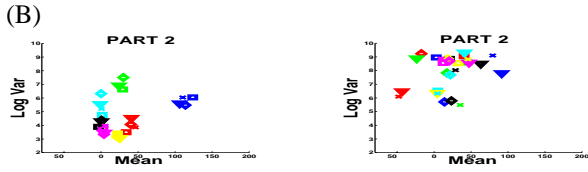
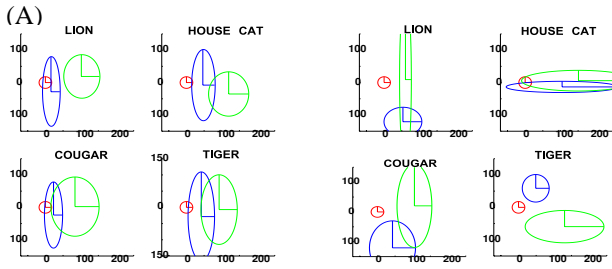


Figure 6: Generative (ML) and Discriminative (CL) Models of Cats Species. Plots are similar to those in Figure 5. (A) The CL shape classes are more variable between classes. (B) The CL appearance models also seem more variable between classes. (C) ML shape models are broad, indicating that the models are focusing on the appearance of patches rather than their relative positions for modelling the object classes. The result is less spatial consistency in the location of the best hypothesis. The CL shape models are a bit tighter indicating that there is useful discriminative power in the mutual positions of the parts.

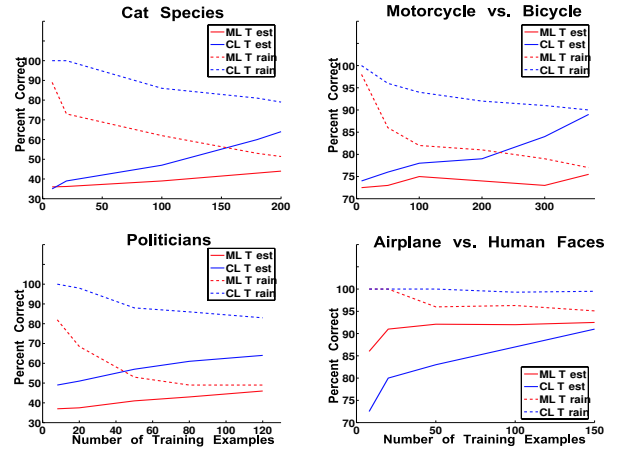


Figure 7: Performance plots as a function of the number of training examples. Data-sets shown are Cat Species (4 classes), Bikes (2 classes), Politicians (4 classes), and Airplanes vs. Human Faces (2 classes). Test errors are the solid lines and train errors are the dotted lines. Blue lines are CL trained models and red are ML trained models. CL tends to outperform the ML models when the classes are similar, but does not show significant performance improvements when the classes are distinct (e.g. Airplanes vs. Human Faces). We expect the train and test errors to converge when sufficient numbers of training examples are used. We notice overfitting for the discriminatively trained models. Performance was averaged over 3 experiments.

ses found seem less consistent for both ML and CL models than those found using the supervised Politicians data-sets.

Figure 7 compares the classification performance as a function of the number of training examples for all experiments. We notice performance improvements using the CL optimization compared to ML optimization for all but the Human Face vs. Airplane data-sets. However, discriminative learning resulted in significant overfitting in part due to the lack of an explicit regularization term within the discriminative expression. Consequentially, discriminative training requires more training examples to reach its optimal test-set performance. The Human Faces vs. Airplane experiment did not exhibit significant performance improvements for CL over ML, indicating that this discriminative framework may not be useful when the object classes of interest come from very different visual categories. We hypothesize that the generative framework will, in general, choose very different representations when the classes are dissimilar, thereby negating the potential benefits of discriminative learning.

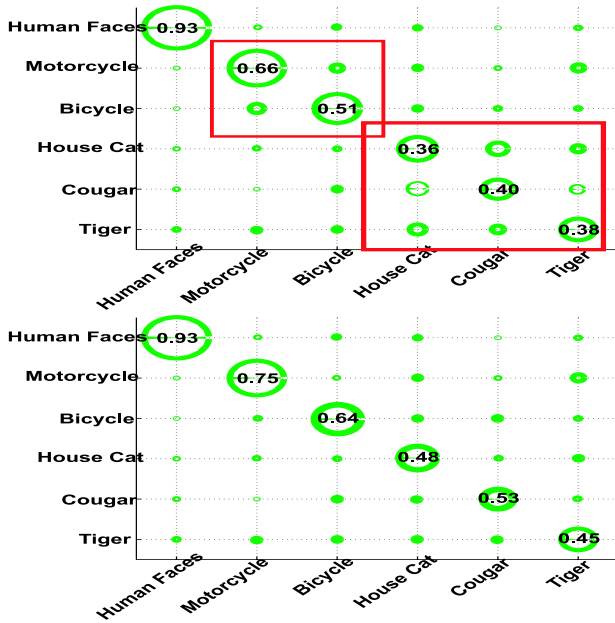


Figure 8: (Top) Initial confusion table for test sets on 6 classes. The x-axis indicates the true category of the image and the y-axis the predicted category for the image. An (x, y) entry indicates the fraction of times class x was classified as y . The values along the main diagonal are the percent of the time an image from class ‘ x ’ was correctly classified. The size of the green dots indicates the magnitude of confusion. Some subsets of classes have higher confusion, namely Cat Species and Bikes. (Bottom) Confusion table after discriminative learning of the Cat Species cluster and the Bike cluster. The performance on the Human Face class does not change. Both discriminative and generative models were trained using the same subset of data. Confused classes were identified using the Training set. Confusion tables shown indicate Test set performance.

6 Generative/Discriminative System

The previous sections indicate that the CL learning framework can result in substantial performance gains when the classes of interest are visually similar. But, these gains come at the price of both increased computational resources and large numbers of training examples. This suggests a natural system for training large numbers of object categories: (1) Initially train models in a generative fashion. (2) Identify areas of high confusion between classes. (3) Train these subsets of confused models with discriminative methods, using high numbers of training examples if necessary.

We implemented such a system using a set of 6 classes: Cats Species (House Cat, Tiger, Cougar), Bikes (Motorcycles, Bicycles), and Human Faces. The confusion table created from the generative models appeals to our semantic notion of similarity, with the subsets Cat Species and Bikes exhibiting higher confusion among themselves than between

other classes for both the train set (not shown) as well as the test set (shown in Figure 8). These 2 subsets of confused classes are good candidates for discriminative learning as identified by the confusion matrices generated by the training examples. Using the same training data used to create the generative models, we train two sets of discriminative models, for both the Cat Species and Bikes. This results in 4 discriminatively trained Cat Species models and 2 discriminatively trained Bike models. We pool test examples into the superset categories Bikes, Cat Species, or Human Faces according to the initial generatively trained models. This initial classification is followed by discriminative classification when the test examples fall either into the group of Bikes or Cat Species models.

Figure 8 illustrates the performance of first classifying using generative models followed by classification using discriminative models. The performance increases by about 10% for both the Cat Species and the Bike classes.

7 Conclusion

We have tested a discriminative learning paradigm on an underlying generative model which maximizes the conditional distribution. We show how this discriminative setting can be used to improve object categorization performance. We suggest that this discriminative setting is particularly useful when the object classes of interest are visually similar. Furthermore we suggest that discriminative classifiers seem to create object models accentuating the differences between classes. We also highlight the trade-off between the discriminative and generative formulations, with the discriminative technique generally outperforming its generative counterpart but requiring both a larger number of training images and greater computational resources. We concluded by proposing an object recognition system which initially trains models in a generative framework, then separates the representations of similar classes using discriminative learning.

Appendix: Data Collection

The Motorcycle, Airplane, and Human Face datasets are from the Caltech Image Data-Base located at www.vision.caltech.edu. We collected images of all other object from the web using the Google, Yahoo, and Lycos search engines. Images were sometimes cropped to emphasize the category of interest and reduce the number of non-object features detected for semi-supervised learning.

Acknowledgements

C. Rasmussen for making the conjugate gradient code 'minimize' publicly available and L. Zelnik, R. Fergus, F.F. Li, and G. Hinton for useful discussions.

References

- [1] M. Weber. *Unsupervised Learning of Models for Object Recognition*, Ph.D thesis, Department of Computational and Neural Systems, Caltech, Pasadena, CA, 2000.
- [2] R. Fergus, P. Perona, A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proc CVPR*, June 2003.
- [3] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, pp. 1150-1157, Sept. 1999.
- [4] H. Schneiderman, T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. CVPR*, 2000.
- [5] G. Dork, C. Schmid. Object class recognition using discriminative local features. *IEEE PAMI* (Submitted).
- [6] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pp. 39-45, 2001.
- [7] S. Kumar, M. Hebert. Discriminative Fields for Modeling Spatial Dependencies in Natural Images. In *Proc NIPS*, 2003.
- [8] A. Opelt, M. Fusseneger, A. Pinz, P. Auer. Generic Object Recognition with Boosting. *IEEE, PAMI*. submitted.
- [9] A. Torralba, K.P. Murphy, W.T. Freeman. Sharing visual features for multiclass and multiview object detection. In *Proc. CVPR*, pp. 762-769, 2004.
- [10] L. Fei-Fei, R.Fergus, P. Perona. A Bayesian approach to unsupervised One-Shot learning of Object categories. In *Proc. ICCV*, 2003.
- [11] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.
- [12] T. Jebara. *Discriminative, Generative and Imitative Learning*. PhD Thesis, Media Laboratory, MIT, December 2001.
- [13] G. Bouchard. The Trade-off Between Generative and Discriminative Classifiers. Technical Note, INRIA.
- [14] Y.D. Rubinstein, T. Hastie. Discriminative vs Informative Learning. In *AAAI*, 1997.
- [15] T. Kadir, M. Brady. Scale, saliency and image description. In *IJCV* 30(2), 1998.
- [16] R. Fergus, P. Perona, A. Zisserman. A Visual Category Filter for Google Images. In *Proc ECCV*, 2004.
- [17] A. Dempster, N. Laird, D. Rubin. Maximum Likelihood from incomplete data via the em algorithm. *JRSS B*, 39:1-38, 1976.