Learning to Detect Partially Labeled People

Yaron Rachlin[1], John Dolan[2], and Pradeep Khosla[1,2]
Department of Electrical and Computer Engineering, Carnegie Mellon University[1]
Robotics Institute, Carnegie Mellon University[2]
rachlin@ece.cmu.edu, jmd@cs.cmu.edu, pkk@ece.cmu.edu

Abstract

Deployed vision systems often encounter image variations poorly represented in their training data. While observing their environment, such vision systems obtain unlabeled data that could be used to compensate for incomplete training. In order to exploit these relatively cheap and abundant unlabeled data we present a family of algorithms called λMEEM. Using these algorithms, we train an appearance-based people detection model. In contrast to approaches that rely on a large number of manually labeled training points, we use a partially labeled data set to capture appearance variation. One can both avoid the tedium of additional manual labeling and obtain improved detection performance by augmenting a labeled training set with unlabeled data. Further, enlarging the original training set with new unlabeled points enables the update of detection models after deployment without human intervention. To support these claims we show people detection results, and compare our performance to a purely generative Expectation Maximization-based approach to learning over partially labeled data.

1. Introduction

Mobile robots and automated surveillance systems that rely on computer vision algorithms typically face an abundance of unlabeled data and a relatively small sample of labeled data. Even if labeled data are available in sufficient quantities during the training phase of an algorithm, operating conditions may change after system deployment, creating a need to retrain the system by augmenting its previous training set with new data. If an algorithm is able to take advantage of the unlabeled data the system collects, the retraining may be automated, forestalling the need for additional human intervention. To achieve these ends, we introduce a set of algorithms, λMEEM, which can exploit unlabeled data to improve detection performance.

λMEEM algorithms combine discriminative learning and estimation of the generative probability distribution of the data [1]. Our experiments indicate that by combining discrimination and prediction, λMEEM algorithms can outperform a strictly discriminative method in classification, and a strictly maximum likelihood approach in prediction. In this paper we demonstrate the efficacy of λMEEM in learning to detect people using partially labeled data.

In our semi-supervised scenario we provide the learning algorithm with significant quantities of unlabeled data collected passively by a moving camera. A user labels a fraction of the data to indicate to the algorithm the class memberships of interest. λMEEM is then used to fit a people detection model to these data. We do not seek to compete with state of the art people detection systems. Instead we choose a simple feature set and detection model in order to focus on the challenges associated with learning over partially labeled data. We compare our results to a purely generative Expectation Maximization (EM) [2, 5] based approach.

To explore the intuition behind concurrently learning a generative and discriminative model, consider the average probability of error of a classifier.

$$P(error) = \int P(error, x)dx = \int P(error \mid x)P(x)dx$$

The probability of error is a function of two terms, the probability of the data and the probability of making an error over the data. Typically we do not know $P(x)$ or $P(error \mid x)$. While generative models seek to accurately characterize $P(x)$, discriminative models attempt to minimize $P(error \mid x)$. During training, if the generative and discriminative models share parameters, λMEEM weaves together these two types of learning. The probabilities estimated by the generative model reweight the exemplars used in discriminative learning, such that mistakes on points likely to occur under our generative model receive increased emphasis, while mistakes over unlikely points receive decreased emphasis. Analogously, the discriminative model weights the exemplars so that points assigned to a class are emphasized by the generative model corresponding to that particular class.

Outside of a classification setting, one can obtain insight about concurrently learning discriminative and generative models by examining unsupervised learning. Consider K-means and EM, two algorithms widely used to find structure in unlabeled data. While K-means is

viewed as a clustering algorithm, the EM algorithm seeks to find a probabilistic model that maximizes the likelihood of the data. The relationship between K-Means and EM is well known [3]. Consider the case of EM applied to a Gaussian mixture model. While seeking parameters that maximize data likelihood, EM iteratively calculates the soft responsibilities/contributions of each Gaussian component to each data point. In contrast, K-means assigns hard responsibilities by partitioning the data points via a Euclidean distance metric. Assuming this form of responsibility, along with restrictions on the component priors and covariance matrices, demonstrates that K-means is a specialized form of EM. The restriction on the responsibilities required to convert EM to K-means states that only one component in the mixture of Gaussians should be responsible for any data point. K-means therefore focuses on partitioning the observed data, an inherently discriminative task, while EM attempts to make the data probable, a generative approach.

The difference between EM and K-means suggests a new framework for fitting models to data based on the relationship between partitioning the observed data and making the observed data likely. This paper presents such an algorithmic framework. EM lies at one extreme and makes the data probable, while a discriminative counterpart, Minimization of Error (ME), lies at the opposite extreme and seeks to find a decision-maker that partitions the data with minimum probability of error. Between the two extremes there exist an infinite number of convergent algorithms, called λMEEM, of which a generalization of K-means is one example.

The theory underlying λMEEM is explained in section 2. We introduce our detection model and feature sets in section 3. Section 4 contains experimental results.

2. λMEEM Algorithms

2.1 Introduction

In this section we introduce the λMEEM family of algorithms. We begin by reviewing relevant work in learning over partially labeled data, and defining the appropriate notation. We then introduce the discriminative ME algorithm, and review the generative EM algorithm. By combining the objective functions of ME and EM we obtain the MEEM algorithm. We then demonstrate that the manner in which the two objective functions are combined can be generalized to yield a set of algorithms called λMEEM. Detailed derivations can be found in [1].

Learning over partially labeled data is an active area of research in the machine learning community. Seeger [4] provides a detailed survey of learning over partially labeled data. Miller and Uyar [5] treat class membership as a latent variable and apply EM to learn by maximizing a joint likelihood function over both labeled and unlabeled data. This allows them to fit separate latent variable models to separate classes, and to utilize partially labeled data, but is not an inherently discriminative approach. Nigam, McCallum, Thrun, and Mitchell [6] apply EM to train a naïve Bayes classifier where the class membership is viewed as the missing data. Superior performance is demonstrated by learning over both labeled and unlabeled data. This approach is also not inherently discriminative in nature. Jaakkola and Haussler [7] combined discriminative and generative learning to obtain superior classification performance by utilizing a generative model trained over partially labeled data to estimate a Fisher Kernel, which they then used to train a Support Vector Machine (SVM). Beyond their use of the non-probabilistic SVM as a discriminative model, their approach differs from ours in that their estimation of the generative model is decoupled from the discriminative model. In our approach, the discriminative and generative models regularize each other during training.

2.2 Notation

The observed data $V = \{V_1,...,V_N\}$ is generated by sampling from mutually exclusive binary hidden/latent variables $H = \{H_1,...,H_K\}$ with associated probabilities $\pi = \{\pi_1,...,\pi_K\}, \Sigma\pi_k = 1$. A generative model, parameterized by θ_G, where $\pi \in \theta_G$, describes the probability distribution of the observed data $P(V \mid \theta_G)$.

Each point $V_i \in V$ belongs to some class $C_m \in C = \{C_1,...,C_M\}$. We define a class to be a subset of generative latent variables. There exists some function A, furnished by the user, that assigns each class to a subset of latent sources, $A: C \to 2^H$. A is not restricted to mapping classes to disjoint sets of latent variables.

If the data set V contains no information about the class membership of the observed points we refer to it as unlabeled. If labels indicating class membership accompany some/all of the data points $V_n \in V$ the data sequence is considered partially/fully labeled.

A decision function $D: (V, \theta) \to C$, described by parameters $\theta = \{\theta_G, \theta_D\}$, proposes a class label for each unlabeled data point V_n. Assume that we know that a latent variable H_k is responsible for generating the unlabeled observation V_n. The decision function D is considered to be in error if the class membership it proposes is inconsistent with the latent variable H_k. Inconsistency between the set of proposed class labels and the generating latent variable is determined via the A function. We define an error function E that embodies this definition.

$$E : (V, \theta, H_k) \rightarrow \{1, 0\}$$
$$E = \begin{cases} 1 & \text{if } H_k \in A(D(V, \theta)) \\ 0 & \text{if } H_k \notin A(D(V, \theta)) \end{cases} \quad (1)$$

For labeled data, we can evaluate our error function directly by examining the consistency of H_k with the true class membership via the A function. Access to the true class label allows us to bypass assigning class membership via the decision-maker.

2.3 Minimization of Error (ME)

The Minimization of Error (ME) algorithm [1] finds model parameters that maximize the probability of not making an error on the observed data:

$$\arg\max_{\theta} P(E = 0 \mid V, \theta) \quad (2)$$

This objective function is difficult to maximize directly due to the complex nature of the underlying parameter space, and in the case of unlabeled data, because we don't know the true class label. Instead one can derive a lower bound, which yields an iterative alternating maximization algorithm for objective function (2). We derive this lower bound by introducing latent variables and corresponding class memberships over the data. Latent variables are introduced by marginalizing in an EM-like manner, while class memberships are introduced by choosing a function A that maps class labels to a subset of the latent variables. Introducing latent variables enables the evaluation of error over unlabeled data points. Given that a particular latent variable occurs we can compare it to the class membership proposed by the decision-maker via A. If this latent variable corresponds to the class proposed, then the decision-maker is considered to be correct. One can therefore think of the probability of being in error over a data point as the probability of being generated by a latent variable inconsistent with the proposed class membership.

One marginalizes over the latent variables by introducing a probability distribution over these variables, $q(H)$. Jensen's inequality is then applied with respect to this distribution to obtain the desired lower bound.

$$\begin{aligned} &\arg\max_{\theta} P(E = 0 \mid V, \theta) \\ &\geq \arg\max_{\theta, q(H)} \int q(H) \log \frac{P(E = 0, H \mid V, \theta)}{q(H)} dH \\ &= \arg\max_{\theta, q(H)} \Delta(q(H), \theta) \end{aligned} \quad (3)$$

The lower bound derived in (3), $\Delta(q(H), \theta)$, is a function of two disjoint sets of parameters, θ and $q(H)$. θ represents the generative and decision model parameters, and $q(H)$ is a probability distribution over the latent variables. In a manner analogous to EM, maximizing Δ with respect to the distribution over the hidden variables is referred to as an E step, while maximizing with respect to the parameters θ is referred to as an M step. By iterating between the two steps one can maximize the lower bound Δ. The E step and its optimal solution, as well as the M step, are shown below.

$$\begin{aligned} &\text{E step: } q^{(i)}(H) := \arg\max_{q(H)} \Delta(q(H), \theta^{(i-1)}) \\ &\qquad \text{optimal } q^{(i)}(H) = P(H \mid E = 0, V, \theta^{(i-1)}) \\ &\text{M step: } \theta^{(i)} := \arg\max_{\theta} \Delta(q^{(i)}(H), \theta) \\ &\qquad = \arg\max_{\theta} \int q^{(i)}(H) \log P(E = 0, H \mid V, \theta) dH \end{aligned} \quad (4)$$

Though the ME algorithm maximizes a lower bound of the true objective function, it also monotonically maximizes this function. Since the true objective function is bounded above, and given that the objective function is monotonically non-decreasing under this algorithm, convergence is guaranteed.

2.4 Combining ME and EM to yield λMEEM

ME's derivation parallels the derivation of EM. The objective function of the EM algorithm seeks to make the observed data as likely as possible given the model:

$$\arg\max_{\theta} P(V \mid \theta) \quad (5)$$

Consider an objective function that seeks to maximize the joint probability of the data and of not making an error in classifying the data.

$$\begin{aligned} &\arg\max_{\theta} P(E = 0, V \mid \theta) \\ &= \arg\max_{\theta} P(E = 0 \mid V, \theta) P(V \mid \theta) \end{aligned} \quad (6)$$

This objective function is equal to maximizing the objective function of ME (2) multiplied by EM's objective function (5). An algorithm called MEEM maximizes this joint objective function [1]. If one assumes the generative model to be a mixture model and the observations to be independent, this algorithm corresponds to a generalized form of K-means.

In order to explore the relationship between maximizing the likelihood of the observed data and the probability of not making an error over these observations the following set of objective functions is proposed [1].

$$\arg\max_{\theta} P(E = 0 \mid V, \theta)^{\lambda} P(V \mid \theta)^{1-\lambda}, \lambda \in [0, 1] \quad (7)$$

Though this set of objective functions departs from a strictly probabilistic setting, the particular manner in which we relate the probability of the data and the probability of not making an error over the data preserves the convexity of ME and EM, allowing us to derive a family of convergent algorithms that co-learn discriminative and generative models called λMEEM. The parameter λ controls the characteristics of the objective function. λ=0 corresponds to EM and seeks to maximize data likelihood, while λ=1 corresponds to ME and seeks only to minimize errors over the data. Intermediate λ-valued objective functions seek to learn the probability of the data and minimum error decision models. By setting λ=0.5 we see that MEEM is also a member of this family of algorithms. These algorithms are summarized below.

As before, we cannot directly maximize the objective function (7) and therefore we derive a lower bound which will yield a tractable solution.

$$\begin{aligned}
&\arg\max_{\theta} P(E=0\mid V,\theta)^{\lambda} P(V\mid\theta)^{1-\lambda} \\
&\geq \arg\max_{\theta, q_\Delta(H), q_\Gamma(H)} \lambda\int q_\Delta(H)\log\frac{P(E=0,H\mid V,\theta)}{q_\Delta(H)}dH \\
&\qquad +(1-\lambda)\int q_\Gamma(H)\log\frac{P(V,H\mid\theta)}{q_\Gamma(H)}dH \\
&= \Lambda(q_\Delta(H), q_\Gamma(H), \theta)
\end{aligned} \tag{8}$$

The λMEEM lower bound concurrently evolves two probability distributions over the hidden variables, $q_\Delta(H)$ and $q_\Gamma(H)$. $q_\Gamma(H)$ describes the likelihood that a hidden variable generated the data. $q_\Delta(H)$ incorporates information about the classification decisions induced by the decision model by describing the likelihood of the data given that a class membership has been determined and thus a subset of the hidden variables has been categorically ruled out. These two posterior distributions are combined in a weighted manner, allowing one to integrate aspects of both distributions. A likelihood-based posterior can soften the hard partitions of a classification-based posterior.

We can prove that maximizing the λMEEM lower bound also maximizes the true objective function. The optimal E steps, and the M step are described below.

$$\begin{aligned}
&\text{Optimal E Steps: } q_\Delta(H) = P(H\mid E=0,V,\theta) \\
&\qquad\qquad\qquad\quad q_\Gamma(H) = P(H\mid V,\theta) \\
&\text{M Step: } \arg\max_{\theta} \lambda\int q_\Delta(H)\log P(E=0,H\mid V,\theta)dH \\
&\qquad +(1-\lambda)\int q_\Gamma(H)\log P(V,H\mid\theta)dH
\end{aligned} \tag{9}$$

To ground the discussion of this set of algorithms we will examine their application to specific generative and discriminative models.

2.5. Mixtures of Gaussians

We apply λMEEM to a mixture of a mixture of Gaussians (MMOG) and a maximum a posteriori (MAP) classifier. From a generative perspective, an MMOG is equivalent to a simple mixture of Gaussians, though from a discriminative perspective we model each class with a separate mixture of Gaussians. Assume data points are independently sampled from a mixture of a mixture of K Gaussians.

$$\begin{aligned}
p(V\mid H_1,\ldots,H_{MK}) &= p(V\mid\mu,\Sigma,\pi) \\
&= \prod_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\pi_{mk}\,p(V_n\mid\mu_{mk},\Sigma_{mk})
\end{aligned} \tag{10}$$

Assume each class corresponds to a disjoint subset of latent variables by defining the following A function: $A(C_i)=\{H_{(i-1)\cdot K+1},\ldots,H_{i\cdot K}\}$. Finally, assume that the decision maker D takes the form of a MAP classifier.

$$\begin{aligned}
D_{MAP}(V_n,\theta) &= \arg\max_{m}\left(p(A(C_m)\mid V_n)\right) \\
&= \arg\max_{m}\left(\sum_{k=1}^{K}\pi_{mk}\,p(V_n\mid\mu_{mk},\Sigma_{mk})\right)
\end{aligned} \tag{11}$$

The full derivation of the E-step for this model is omitted due to space limitations. The M step was performed using conjugate gradient descent over constrained parameters. Constraints on the covariance matrix parameters and mixing weights were necessary in order to ensure that the Gaussian probability distributions proposed by the algorithm were well defined.

3. Person Detection Model

3.1 Introduction

To provide a context for the evaluation of the use of partially labeled training data when training an appearance-based people detection model we develop a simple statistical people detection framework. We did not attempt to construct a state of the art detection system, examples of which are presented in [8, 9], but one that demonstrates good detection performance through the incorporation of partially labeled data in the training process.

3.2 Detection Model

Our object detection framework is based on a model which captures the statistics of a feature vector computed at each pixel. The color values at each pixel comprise our feature set. Taking RGB pixel values, we convert these values to CIE LAB space, and then discard the intensity

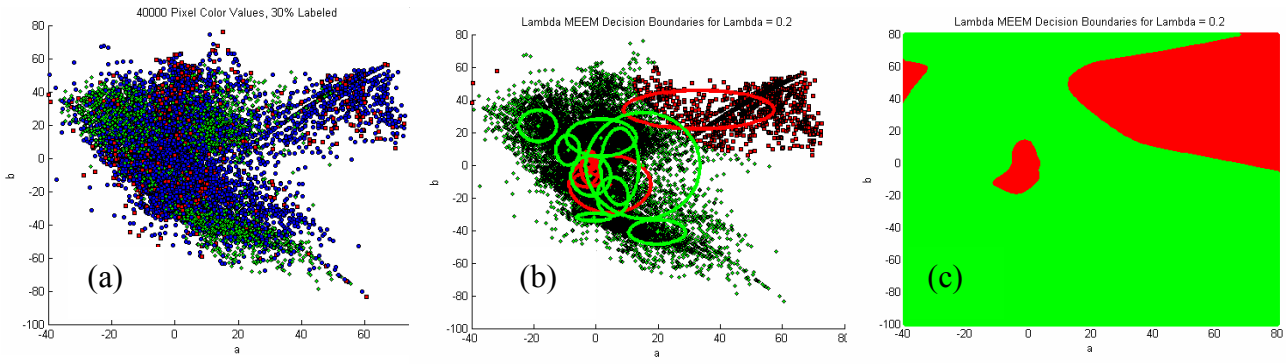

Figure 1. Plot (a) shows our partially labeled data. Blue circles correspond to unlabeled points, green diamonds to points labeled as 'background,' and red squares correspond to points labeled as 'person.' Plot (b) displays a λMEEM solution to the data in plot (a). Red squares indicate training points classified as people, and green diamonds indicate points classified as clutter. The ellipsoids indicate the one-standard-deviation contours of the Gaussians. The dot at the center of the ellipsoids indicates the mean. Ellipsoid color indicates class membership. Plot (c) demonstrates the decision boundaries induced by these Gaussians. Green regions correspond to parts of the color space classified as background, and red regions indicate portions of the color space classified as people.

information. Though color forms a poor feature space, particularly given the well-known color constancy problem [10], we chose to use it in order to maintain the simplicity of our detection model. By choosing a simple feature set we seek to focus solely on the learning algorithm.

Our training set contains a set of pictures taken from a moving camera in an indoor environment. Looking through the images, we selected several patches in the image and labeled them as 'person' or 'background.' A partially labeled color data set is shown in figure 1a. From these data we can see that in our color feature space the 'person' and 'background' classes overlap significantly. Furthermore, the multimodal locations of the labeled points suggest that the underlying class conditional probability distributions are not Gaussian in nature.

In our model, the distribution of values in the feature space is modeled using a mixture of axis-aligned Gaussians. We use nine Gaussians to model the person class and an additional nine Gaussians to model the background class. Using λMEEM we fit these models to the data. A λ=0.2 λMEEM model and its corresponding decision boundary are shown in figure 1. The value λ=0.2 was chosen using cross validation. From an initial examination of figure 1b the manner in which a λMEEM solution differs from a conventional EM-based mixture of Gaussians solution is not obvious. To highlight this difference we used EM [5] over the same partially labeled data to fit an identical type of model from the same initial point. Using a test image, we plotted at each pixel the ratio of the probability of a person to the probability of background. The results are shown in figure 2. λMEEM induces sharper class partitions than EM, and produces lower per pixel error rates as shown in figure 4.

To move from a pixel-based model to a person-based model we assume that the probability that a particular pixel was generated by a person as opposed to the background is determined not only by the pixel, but also by its neighbors. A neighborhood is defined as a rectangle centered at each pixel. Within this rectangle we assume that all pixels were generated independently by either the 'person' or 'background' model. Given some independence assumptions we compute the class membership of highest posterior probability for the center pixel. Figure 3 shows the ratio of the posterior of the 'person' to 'background' class for each pixel, and the corresponding people detection results for a test image.

Figure 2. We obtained the image on the left from our test set, and plotted pixel-wise plots of the ratio of the person to background model probabilities. Black pixels indicate low ratios, and white pixels indicate high ratios. The models were trained using the data plotted in figure 1a. The EM-based model ratios are shown in the middle. The λMEEM model based ratios are shown on the right. As can be seen in these plots, λMEEM induces sharper partitions between the person and background models than the EM-based models.

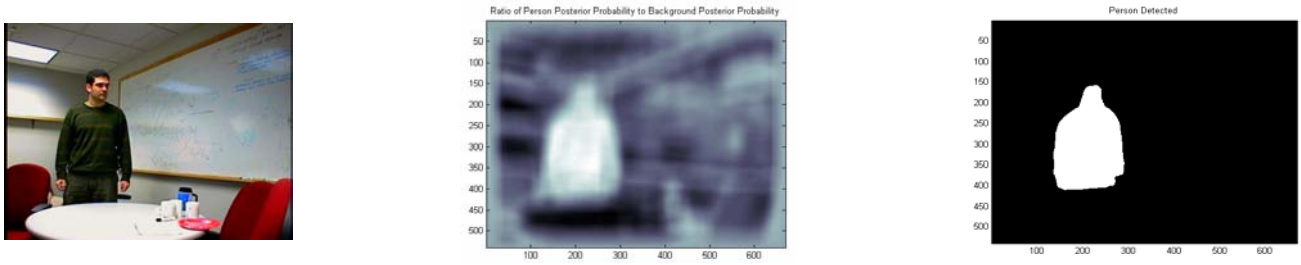

Figure 3. The image on the left was obtained from our test set. The image in the middle shows the ratio of the person class to background class posterior probabilities at each pixel. The image on the right displays the resulting person detection.

4. Data Augmentation Experiments

In figure 4 we summarize a set of experiments where we added unlabeled pixels to a set of thirty labeled pixels. Using our enlarged data sets we trained a people detection model using both an EM approach [6] and λMEEM. We used cross validation to determine a good choice of λ for each training set. The best lambda value decreased as the fraction of labeled data decreased, but always remained above 0. Though for both algorithms adding unlabeled data improves classification performance, λMEEM consistently achieves a lower error rate than EM on our test set. It is important to note that generalization performance does not improve monotonically with additional unlabeled data points for either algorithm.

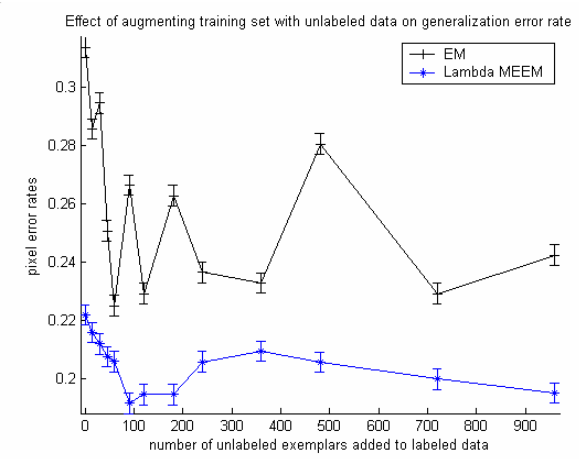

Figure 4. Variation in error rates over a test set for EM and λMEEM as a function of the number of unlabeled points added to thirty labeled points. Error bars were obtained using the Hoeffding inequality with a 95% confidence interval.

5. Conclusions

λMEEM algorithms can effectively exploit the abundance of unlabeled data available in vision applications such as people detection. Further, these algorithms can outperform purely EM based approaches to learning over partially labeled data. In the future we hope to apply λMEEM algorithms to sophisticated detection models, and to develop an approach to choosing λ beyond cross validation. We also hope to explore the non-monotonic effect on performance of augmenting our training set with additional unlabeled points.

6. References

[1] Rachlin, Y. (2002) A General Algorithmic Framework for Discovering Discriminative and Generative Structure in Data. MS Thesis, Department of Electrical & Computer Engineering, Carnegie Mellon University.

[2] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. J. Royal Stat. Soc. (B), 39, 1-38.

[3] Mackay, D. (2003) Information Theory, Inference, and Learning Algorithms. Draft 3.1415. http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html

[4] Seeger, M. (2001) Learning with labeled and unlabeled data. Technical Report, University of Edinburgh.

[5] Miller, D., Uyar, H. (1996) A Mixture of Experts classifier with learning based on both labeled and unlabeled data. Advances in Neural Information Processing Systems 9, 571-577.

[6] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (2000) Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, 39(2/3), 103-134.

[7] Jaakkola, T. S., Haussler, D (1998). Exploiting generative models in discriminative classifiers. Advances in Neural Information Processing Systems 11, 487-493.

[8] Mittal, A., Davis, L. (2002) M2Tracker: A Multi-view Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo. European Conference on Computer Vision(1), 18-36.

[9] Papageorgiou, C., Poggio, T. (1999) Trainable Pedestrian Detection. Proceedings of International Conference on Image Processing (4), 35-39.

[10] Rosenberg, R., Hebert, M., Thrun, S. (2001) Image Color Constancy Using KL-Divergence. Poster appeared at International Conference on Computer Vision.