Which Faces to Tag: Adding Prior Constraints into Active Learning

Ashish Kapoor[†], Gang Hua[‡], Amir Akbarzadeh[‡] and Simon Baker[†]
[†]Microsoft Research and [‡]Microsoft Corporation
1 Microsoft Way, Redmond WA 98052 USA
{akapoor, ganghua, amir, sbaker}@microsoft.com

Abstract

We introduce an algorithm that guides the user to tag faces in the best possible order during a face recognition assisted tagging scenario. In particular, we extend the active learning paradigm to take advantage of constraints known a priori. *For example, in the context of personal photo collections, if two faces come from the same source photograph, we know that they must be of different people. Similarly, in the context of video, we know that the faces from a single track must be of the same person. Given a set of unlabeled images and constraints, we use a probabilistic discriminative model that models the posterior distributions by propagating label information using a message passing scheme. The uncertainty estimate provided by the model naturally allows for active learning paradigms where the user is consulted after each iteration to tag additional faces. Our experiments show that performing active learning while incorporating* a priori *constraints provides a significant boost in many real-world face recognition tasks.*

1. Introduction

Tagging the identity of people in photos is an important tool in photo organization. Commercial systems such as Google Picassa [1] and Apple iPhoto [2] have recently added face recognition and clustering to help partially automate this process. Similarly, tagging people can be used to help organize and search personal video collections.

In classical face recognition [8, 17, 21], the goal is usually to optimize some form of recognition or verification rate on a probe set of test images, given a fixed gallery of training images (and possibly some generic training data of miscellaneous faces outside the probe and gallery sets) [12]. This goal is a good match to the requirements of most traditional applications of face recognition such as surveillance.

On the other hand, the primary goal in face tagging is to tag the faces as quickly and accurately as possible. The training set is no longer fixed and outside the control of the algorithm. As faces are tagged, the training set can be extended. One key question[1] is then which faces should be tagged first to maximize performance on the rest of the data. This question has received little attention in the face recognition literature, with the notable exception of [15].

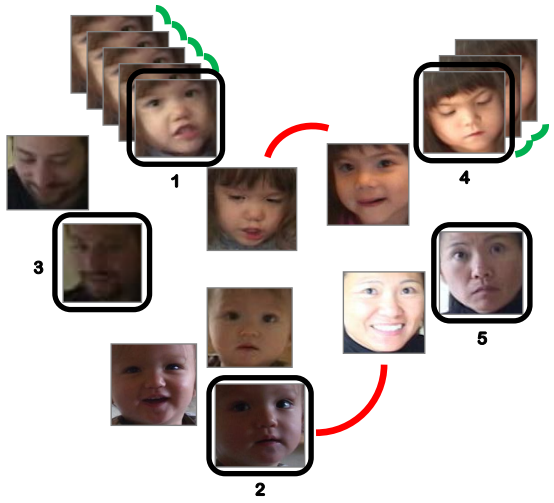

Figure 1. Our algorithm guides a user to tag faces in the best possible order. In particular, our approach takes advantage of constraints known *a priori.* Red lines show that two faces come from the same photograph and therefore must be of different people. Green lines show that the faces come from a single face track in a video and must be of the same person. The numbers are an illustration of the order in which our algorithm would ask the user to tag the faces. The algorithm exhibits a preference to tag faces that occur in photos with one or more other faces because tagging that face provides information about the other faces in that photo. Also, there is a preference to tag long tracks with pose variation.

The field of active learning provides a nice framework for choosing which faces to tag and has been a topic of considerable interest [6, 16, 4, 10, 11]. For example, Freund et al. [6] propose disagreement among a committee of clas-

[1]Another part of the problem is the design of the user interface. See [5] for example. User interface design is outside the scope of this paper. Instead we focus on the choice of which face images to tag.

sifiers as a criterion for active learning, and have shown an application to image classification [3]. Several authors have explored active learning in an Support Vector Machine (SVM) framework. Tong and Koller [16] select unlabeled cases to query based on minimizing the version space within the SVM. Chang et al. [4] have used active learning with SVMs for the task of image retrieval using color and texture features extracted from an image. Similarly, within the Gaussian process framework [9, 13], the method of choice has been to look at the expected informativeness of an unlabeled data point [10, 11]. In computer vision active learning has been employed for object categorization [9, 7], video annotation [20], and face tagging [15].

In this paper we extend the classical active learning paradigm and present a framework that allows the incorporation of additional sources of prior information. These additional constraints are what distinguishes our work from previous face tagging papers [15] and the more general active learning literature [6, 16, 4, 10, 11]. For example, if two faces appeared in the same unedited photo, the two faces cannot have the same identity. We call such constraints *non-match* constraints. Another example is in video. If faces are tracked and two images are from the same track, they must have the same identity. We call such constraints *match* constraints. Note that Tian et al.[15] perform partial clustering and assume that each cluster contains a single identity. In our framework, this can be regarded as a form of match constraint. However, this clustering-based match constraint is liable to errors in the clustering and the active learning approach relies on a number of heuristics. Our framework is more generally applicable to both match and non-match constraints, and more principled.

We begin by first proposing a probabilistic discriminative model that aims to induce a probability distribution over class labels by both considering the face images as well as known constraints. In particular our model consists of a Gaussian process (GP) prior [9, 13], which enforces a smoothness constraint on the labels, and a Markov random field (MRF) that enforces both the *match* and *non-match* constraints. We also propose an efficient variational message passing to infer the unobserved labels given the face regions and a set of known labels. Because of the probabilistic nature of our model, it provides all the information we need to develop an active learning criterion. In particular, our active learning selection criterion utilizes uncertainty estimates to determine the next face to tag.

Rather than evaluating our algorithm on standard face recognition benchmarks [14, 12], we evaluate it on a number of personal photo and video collection (one from a personal DV tape and one from a commercial TV sitcom.) We present results using both a held-out probe set and by treating the data as a single bag of images that need to be labeled. In both cases we show that the addition of constraints yields a significant boost in performance.

2. Active Learning with Constraints

Assume we are given a set of face images $\mathbf{X} = \{\mathbf{x}_i\}$. We partition this set into a set of labeled ones $\mathbf{X}_L$ with labels $\mathbf{t}_L = \{t_i | i \in L\}$ and a set of unlabeled ones $\mathbf{X}_U$. At the beginning of the tagging task, we have $\mathbf{X}_L = \emptyset$ and $\mathbf{X}_U = \mathbf{X}$. Our goal is to request as few labels as possible from a human oracle to maximize the classification rate over the entire set of images $\mathbf{X}$. (In our experimental results, we also include classification rates on a held-out testing subset.)

If we treat each face image independently, a standard active learning criterion such as uncertainty [6] or information gain [11] can be used to determine the next face to tag at each step. For example, we can use off-the-shelf prediction algorithms such as SVMs [16] or Gaussian Process (GP) models [9] to infer the posterior distribution $p(\mathbf{t}_U|\mathbf{X}, \mathbf{t}_L)$ over unobserved labels $\mathbf{t}_U = \{t_i | i \in U\}$. This distribution can then be used in the active learning criterion. However, we would like to model the dependencies between images and below we describe a discriminative model that utilizes contextual constraints in order to classify unlabeled images and determine what face to tag next.

We present a probabilistic model that utilizes such constraints correctly to propagate information pertinent to content as well as other known constraints in order to infer the unobserved labels. Our framework considers pairwise constraints between images that specify whether two faces are the same or different. *Non-Match constraints* between two face images mean that the images must have different labels. *Match constraints* mean that the two face images must have the same label. We assume that a set of the match and non-match constraints have been provided:

$$\mathbf{NMC} = \{(t_i, t_j) \,|\, t_i \neq t_j\}$$
$$\mathbf{MC} = \{(t_i, t_j) \,|\, t_i = t_j\}$$

The remainder of this section is organized as follows. We begin in Section 2.1 by presenting a discriminative model which incorporates the constraints. We proceed in Section 2.2 to describe how inference can be performed in this model to compute the posterior distribution over the unobserved labels:

$$p\left(\mathbf{t}_U|\mathbf{X}, \mathbf{t}_L\right) \tag{1}$$

given the input face images $\mathbf{X}$, the labels added so far $\mathbf{t}_L$, and the constraints, both non-match **NMC** and match **MC**. Finally, in Section 2.3 we show how $p(\mathbf{t}_U|\mathbf{X}, \mathbf{t}_L)$ can be used in an active learning criterion.

2.1. A Discriminative Model with Constraints

We propose a model that consists of a network of predictions that interact with one another such that the decision of each predictor is influenced by the decision of its neighbors. Specifically, given match and non-match constraints we induce a graph where every vertex corresponds to a label t_i,

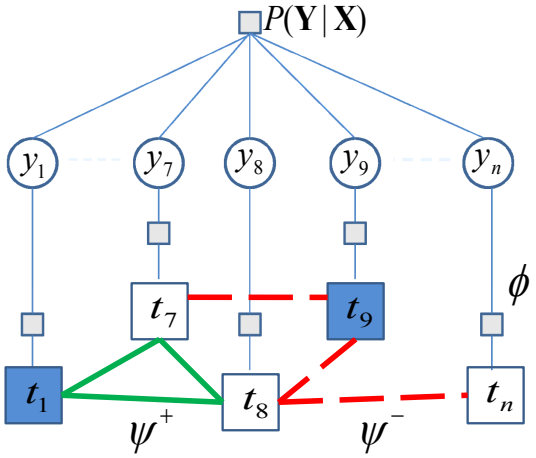

Figure 2. Factor graph depicting the proposed discriminative model. The shaded nodes correspond to the observed labels (training data) and the thick green and dashed red line correspond to match and non-match constraints respectively.

$i \in L \cup U$, and is connected to its neighbors according to the given constraint. We will denote the set of edges corresponding to match and non-match edges as $\mathcal{E}^+$ and $\mathcal{E}^-$ respectively.

Figure 2 illustrates the factor graph corresponding to the proposed model. The class labels $\{t_i : i \in L \cup U\}$ are denoted by squares and influence each other based on different match (green lines) and non-match (dashed red lines) constraints. In addition to these constraints, our model also imposes smoothness constraints using a GP prior [13]. We introduce latent variables $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$ that use a GP prior to enforce the assumption that *similar* points should have similar prediction. In particular, the latent variables are assumed to be jointly Gaussian and the covariance between two outputs y_i and y_j is typically specified using a kernel function applied to $\mathbf{x}_i$ and $\mathbf{x}_j$. Formally, $p(\mathbf{Y}|\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\mathbf{K}$ is a kernel matrix[2] with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and encodes similarity between pairs of face regions. For the rest of the discussion in this section we assume that we are given a kernel matrix $\mathbf{K}$. We describe more details on how we compute this kernel matrix in section 3.1.

Given a pool of images, the model induces a conditional probability distribution $p(\mathbf{t}, \mathbf{Y}|\mathbf{X})$ using the GP prior $p(\mathbf{Y}|\mathbf{X})$ and potential functions ϕ, ψ^+ and ψ^-. Here ϕ encodes the compatibility of a label t and the corresponding latent variable y. Further, ψ^+ and ψ^- encode the pairwise label compatibility according to the match and non-match constraints respectively. Thus, the conditional distribution induced by the model can be written as:

$$p(\mathbf{t}, \mathbf{Y}|\mathbf{X}) = \frac{1}{Z} p(\mathbf{Y}|\mathbf{X}) \prod_{i=1}^{n} \phi(\mathbf{y}_i, t_i) \times \prod_{(i,j)\in\mathcal{E}^+} \psi^+(t_i, t_j) \prod_{(i,j)\in\mathcal{E}^-} \psi^-(t_i, t_j)$$

where Z is the partition function (normalization term) and the potentials ϕ, ψ^+ and ψ^- take the following form:

$$\phi(\mathbf{y}_i, t_i) \propto e^{-\frac{||\mathbf{y}_i - \bar{\mathbf{t}}_i||^2}{2\sigma^2}}$$
$$\psi^+(t_i, t_j) = \delta(t_i, t_j)$$
$$\psi^-(t_i, t_j) = 1 - \delta(t_i, t_j).$$

Here, $\delta(\cdot,\cdot)$ is the Dirac delta function and evaluates to 1 whenever the arguments are equal, and zero otherwise. Also, $\bar{t}_i$ is the indicator vector corresponding to t_i and σ^2 is the noise parameter and determines how tight the relation between the smoothness constraint and the final label is. By changing the value of σ we can emphasize or de-emphasize the effect of the GP prior.

Note that in absence of any match and non-match constraints the model reduces to a multi-class classification scenario with GP models [9, 13]. Further, the model is akin to a conditional random field (CRF), although modeling a large multiclass problem as a CRF with kernels is non-trivial due to the large number of parameters that would be required to solve such a problem. The proposed model does not suffer from these problems as GPs are non-parametric models.

In summary, our model provides a powerful framework for modeling non-linear dependencies using priors induced by kernels. Also note that this is a discriminative framework as we never model $P(\mathbf{X})$, the high dimensional underlying density of observations. Moreover, as we will see in the next section this model will allow us to perform message passing to resolve the smoothness, match and non-match constraints and infer the unobserved variables in an efficient manner. Finally, the probabilistic nature of the approach provides us with valid probabilistic quantities that can be used to perform active selection of the unlabeled points.

2.2. Inference in the Model

Given some labeled data the key task is to infer $p(\mathbf{t}_U|\mathbf{X}, \mathbf{t}_L)$ the posterior distribution over unobserved labels $\mathbf{t}_U = \{t_i | i \in U\}$. Performing exact inference is prohibitive in this model primarily due to two reasons. First, notice that the joint distribution is a product of a Gaussian (GP prior and unary potentials) and non-Gaussian terms (pairwise match ψ^+ and non-match constraints ψ^-). More importantly, the match and the non-match constraints might induce loops in the graph making exact inference intractable. We resort to approximate inference techniques

[2]This kernel matrix is a positive semidefinite matrix and is akin to the kernel matrix used in classifiers such as SVMs.

in order to get around this problem. In particular we perform an approximate inference by maximizing the variational lower bound by assuming that the posterior over the unobserved random variable $\mathbf{Y}$ and $\mathbf{t}_U$ can be factorized:

$$F = \int_{\mathbf{t}_U,\mathbf{Y}} q(\mathbf{t}_U)q(\mathbf{Y}) \log \frac{p(\mathbf{t}_U, \mathbf{Y}|\mathbf{X}, \mathbf{t}_L)}{q(\mathbf{t}_U)q(\mathbf{Y})}$$
$$\leq \log \int_{\mathbf{t}_U,\mathbf{Y}} p(\mathbf{t}_U, \mathbf{Y}|\mathbf{X}, \mathbf{t}_L),$$

where $q(\mathbf{Y})$ is assumed to be a Gaussian distribution and $q(\mathbf{t}_U)$ is a discrete joint distribution over the unobserved labels. The approximate inference algorithm aims to compute good approximations $q(\mathbf{Y})$ and $q(\mathbf{t}_U)$ to the real posteriors by iteratively optimizing the above described variational bound. Specifically, given the approximations $q^k(\mathbf{Y}) \sim \mathcal{N}(\mathbf{M}^k, \Sigma^k)$ and $q^k(\mathbf{t}_U)$ from the k^{th} iteration the update rules are as follows:

$$q^{k+1}(\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X})\phi(\mathbf{Y}_L, \mathbf{t}_L)\phi(\mathbf{Y}_U, q^k(\mathbf{t}_U))$$
$$q^{k+1}(\mathbf{t}_U) \propto \Psi^+\Psi^-\phi(\mathbf{M}^k_U, \mathbf{t}_U)$$

For clarity, we have collected the product of unary potential terms in ϕ and all the labeled instantiation of local potentials, match and non-match constraints in Ψ^+ and Ψ^- respectively. The first update equation considers the current beliefs about the unlabeled data and incorporates it in updating $q(\mathbf{Y})$. Notice that the first update equation is just a product of Gaussian terms and can be computed easily:

$$q^{k+1}(\mathbf{Y}) \sim \mathcal{N}(\mathbf{M}^{k+1}, \Sigma^{k+1}) \text{ Where:}$$
$$\mathbf{M}^{k+1} = \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1} \begin{bmatrix} \bar{\mathbf{t}}_L \\ q^{k+1}(\mathbf{t}_U) \end{bmatrix}$$
$$\Sigma^{k+1} = \mathbf{K} - \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{K}.$$

Again $\bar{\mathbf{t}}_L$ denotes the indicator matrix where everything is zero except the (i, j) entry is set to 1 if $\mathbf{x}_i$ is labeled as class j. This update operation is the same as inferring $\mathbf{Y}$ by considering distributions over unlabeled data in addition to the labeled images.

The second update equation similarly considers the local potentials $\phi(\mathbf{M}^k_U, \mathbf{t}_U)$ induced by the posterior $q^k(\mathbf{Y})$ over the latent variables and needs to resolve the pairwise constraints in order to compute the updated distribution over $\mathbf{t}_U$. Notice that the second update equation for $q^{k+1}(\mathbf{t}_U)$ has the same form as a Markov Random Field. In particular Ψ^+ and Ψ^- are the edge potentials while $\phi(\mathbf{M}^k, \mathbf{t}_U)$ are the local potentials. Consequently we can use loopy Belief Propagation to first compute the marginal probabilities $q^{BP}(t_i)$ for all $i \in U$ and set $q^{k+1}(\mathbf{t}_U) = \prod_{i\in U} q^{BP}(t_i)$. Note that in presence of cycles in the graph doing loopy belief propagation till convergence will provide approximations for the marginal distributions. The pseudocode for the message passing scheme is provided in Algorithm 1.

Algorithm 1 Inferring the Unknown Labels

function probOut = Infer($\mathbf{K}$, $\bar{\mathbf{t}}_L$, $\mathcal{E}^+$, $\mathcal{E}^-$)

Compute $\mathbf{A} = \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}$

Initialize:
$q^0(\mathbf{t}_U) = 0$ and $q^0(\mathbf{Y}) = \mathcal{N}(\mathbf{M}^0, \Sigma^0)$
where $\mathbf{M}^0 = \mathbf{A}[\bar{\mathbf{t}}_L; q^0(\mathbf{t}_U)]$and $\Sigma^0 = \mathbf{K} - \mathbf{AK}$

for k = 0 to Maximum Iterations or Convergence **do**

Update $q(\mathbf{t}_U)$:
Do LoopyBP over MRF induced by $\mathcal{E}^+$, $\mathcal{E}^-$ and ϕ
$q^{k+1}(\mathbf{t}_U) = \prod_{i\in U} q^{BP}(t_i)$

Update $q(\mathbf{Y})$:
$\mathbf{M}^{k+1} = \mathbf{A}[\bar{\mathbf{t}}_L; q^{k+1}(\mathbf{t}_U)]$

end for

Return probOut = $q(\mathbf{t}_U)$

Alternating between the above described updates can be considered as message passing between a classifier and a constraint resolution scheme. By doing the update on $q(\mathbf{t}_U)$, the constraints are imposed on classification results and are resolved by performing belief propagation. Similarly, by updating $q(\mathbf{Y})$ any new information learnt about the unlabeled points is propagated to the classifier in order to update its class beliefs. By iterating between these two updates the model consolidates information from both components, and thus provides a good approximation of the true posterior.

The computational complexity of the first update step is $O(N^3)$ due to the inversion of the $N \times N$ matrix, where N is the total number of data points. However, note that the particular term $\mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}$ needs to be computed only once and can be reused in every iteration. Inference using loopy belief propagation is the dominating factor in the computational complexity for the second update equation. However, these constraint graphs are often sparse and consequently inference can be run fairly efficiently for most of the real-world photo collections.

2.3. Use in an Active Learning Criterion

The task in active learning is to seek the label for one of the unlabeled examples and then update the classification model by incorporating it into the existing training set. The goal is to select the sample that would maximize the benefit in terms of the discriminatory capability of the system.

A popular heuristic with non-probabilistic classification schemes is to first establish the confidence of the estimates using the distance from the classification boundary (margin) and select the closest to the margin. However, our Bayesian model provides a full posterior distribution over the class labels for the unlabeled points which can be used for active learning.

As in standard active learning methods [11, 16], measures such as uncertainty or information gain can be used. Formally, we can write these two selection criteria as:

$$\mathcal{UN} : \mathbf{x}^* = \arg\max_{i \in U} H(t_i)$$
$$\mathcal{INFO} : \mathbf{x}^* = \arg\max_{i \in U} H(\mathbf{t}_U \setminus i) - \mathsf{E}_{t_i}[H(\mathbf{t}_U \setminus i | t_i)]$$

Here, $H(\cdot) = -\sum_{c \in \text{classes}} p_c \log(p_c)$, where $p_c = q^{BP}(t_i = c)$, and in our model denotes Shannon entropy and is a measure of uncertainty. The uncertainty criterion seeks to select the face image with most uncertainty, whereas the information gain criterion seeks to select a data point that has the highest expected reduction in uncertainty over all the other unlabeled points. Either of these criteria can be computed given the inferred posteriors; however we note that the information gain criterion is far more expensive to compute as it requires us to do repeated inference by considering all possible labels for every unlabeled data point. The uncertainty criterion on the other hand is very simple and often guides active learning with reasonable amount of gains [9, 16]. In this work we will consider uncertainty as the primary active learning criterion, although please note that extension to other information theoretic schemes is possible.

3. Experiments

We consider two application scenarios that naturally provide constraints. First, we consider the task of tagging faces in a personal photo collection. Associating identities of faces in a photo collection can greatly enhance the photo browsing experience. The scenario naturally induces non-match constraints between faces that appear in the same photo. The other application scenario we consider is tagging faces in videos. Tagging the faces in a home video allows the video to be searched more easily. For example, it would be possible to find all the shots of a particular family member. Besides inducing non-match constraints this scenario also leads to match constraints. By using a tracking algorithm we can induce match-constraints for faces belonging to the same track. Below we describe the procedure we used to detect and extract faces and determine the constraints.

3.1. Face Processing Pipeline

Our face recognition pipeline uses the Viola-Jones face detector [18] to detect face regioins. In the case of videos, the pipeline first breaks the video into shots using a color-histogram based shot detector. A matching algorithm is then used to assemble the face regions into tracks. Missed detections are filled in by interpolating between previous and subsequent frames.

Constraint Generation: The *non-match constraints* are generated by considering each photo in the collection or each frame in the video. A non-match constraints is added for any pair of faces that appear in the same photograph or the same video frame, and for which the face regions do not overlap. Similarly, a *match constraint* is added for each pair of faces that appear in the same track. In our implementation tracking is very conservative and we can be sure that each track just contains a single person. A result of this conservative algorithm is that tracks may be broken into multiple shorter ones that definitely contain the same person. This may result in slightly less information, but avoids erroneous constraints.

Feature Extraction: We applied two different facial feature extraction algorithms to the face regions. The first was a simple eigenface algorithm [17]. The face regions were re-sampled and normalized to 64×64 grayscale patches with zero mean and unit variance. Principle components analysis was then performed and enough eigenvectors are selected to retain 95% of the empirical variance. The face patches were then projected into the principle components. The distance metric used is a simple L1 distance.

The second algorithm utilized the face recognition pipeline proposed by Wright and Hua [19]. Each detected face is first geometrically rectified and then photometrically normalized. The rectified face region is then partitioned into overlapping small patches. A local image descriptor is extracted from each patch. The location of each patch is appended to the corresponding image descriptor to come up with a joint spatial-appearance descriptor. Each augmented descriptor is then quantized by a set of pre-trained randomized projection trees (RPTrees). The final face representation is a bag of quantized indices of these spatial-appearance descriptors, i.e., a sparse histogram. We refer to this representation as RPTrees. Following Wright and Hua [19], the distance metric is the inverse document frequency (IDF) weighted L1 distance between two sparse histograms.

Given the matrix $\mathbf{D}^{sq}$ of the squared distances using either of these representations, we induce a Gaussian Process kernel $\mathbf{K} = [K_{ij}]$ such that $K_{ij} = \exp(-\frac{D^{sq}_{ij}}{\text{mean}(\mathbf{D}^{sq})})$. Note, that this kernel is positive semi-definite and can be used in any kernel based classification algorithm including SVM and GP classification.

3.2. Description of Data

We performed experiments on 4 datasets. Two of these datasets came from personal photo collections. The other two were extracted from video footage. We describe the data in detail below.

HomeDV: is a set of faces extracted from a 51 minute home video. The video footage contains 5 people, 4 members of a family (mom and 3 young children) and a house guest. The dataset contains 420 tracks. Each track is then

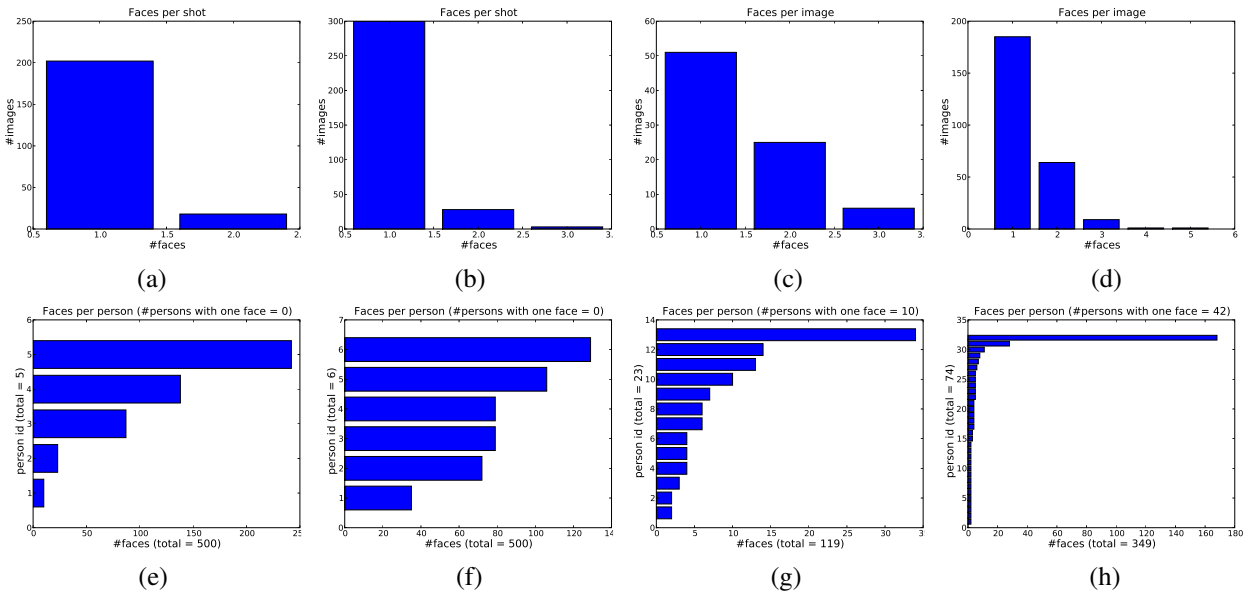

Figure 3. Histograms of faces per image ((a) HomeDV, (b) Sitcom, (c) London Trip, (d) Birthday Party) and faces per person ((e) HomeDV, (f) Sitcom, (g) London Trip, (h) Birthday party).

sampled every 5 frames yielding 5742 faces. We conduct our experiments on a sample of 500 faces randomly selected from this data.

Sitcom: is extracted from the concatenation of 4 episodes of the TV show "Friends," resulting in approximately 80 minutes of video. We just extract the faces of the 6 main characters, Rachel, Joey, Ross, Chandler, Monica, and Pheobe. The dataset contains 1282 tracks. Again, each track is then sampled every 5 frames yielding 16720 faces and we conduct our experiments on a random selection of 500 faces.

Note that although HomeDV and Sitcom have similar numbers of subjects, in HomeDV most of the faces are from 3 people, the 3 children, whereas in Sitcom the distribution of the number of face regions per person is much more uniform. Figure 3 includes histograms of the number of faces per image ((a) and (e)) in the video and another histogram of the number of face regions ((b) and (f)) for each subject in these videos.

London Trip: contains photos from a group trip to London. There are 23 people in the collection with a total of 119 faces. The distribution for faces per image is shown in Figure 3(c). The distribution of the faces per person, shown in Figure 3(g), is more spread out than Birthday (described next). In this set there are 10 persons that only appear once.

Birthday party: this dataset comes from a personal photo collection and contains 74 people with a total of 349 faces. The distribution of faces per image and faces per person is included in Figure 3(d) and (h) respectively. Notice that there are 42 persons that only appear once. One person, the owner of the photos, appears almost 170 times.

3.3. Results

We present results in terms of recognition rate, both estimated on a held-out "validation" set and estimated over the complete pool of data that our algorithm is being applied to. The accuracy on the validation set is interesting when we care about building good classifiers that would be able to work beyond the available pool of examples, whereas accuracy on the other pool of images closely reflects the realistic task of just tagging a particular photo/image collection. To generate the test set, we randomly held out 50% of the data. The remaining 50% was used for active learning. The active learning scheme selected a single face image in every round. In an application, multiple images might be selected greedily. But selecting one per round is more natural for evaluation purposes.

We compare 4 different methods. The first is our full algorithm, with the constraints. The second is active learning on our model without the constraints. The third is an SVM based active learning algorithm [16] that is commonly used for comparison purposes. The fourth is a random sampling baseline. All the experiments were performed 50 times by randomly splitting the datasets into the validation set and the pool of unlabeled images available for active learning. The results include average results over these 50 trials and the standard error. We fix $\sigma = 10^{-5}$ for the proposed model (with and without constraints) and correspondingly for the SVM set the parameter $C = 10^5$. We also tried other settings of these parameters and found that the results did not differ significantly. Also note that we use the same distance matrices and the induced kernels for all different schemes.

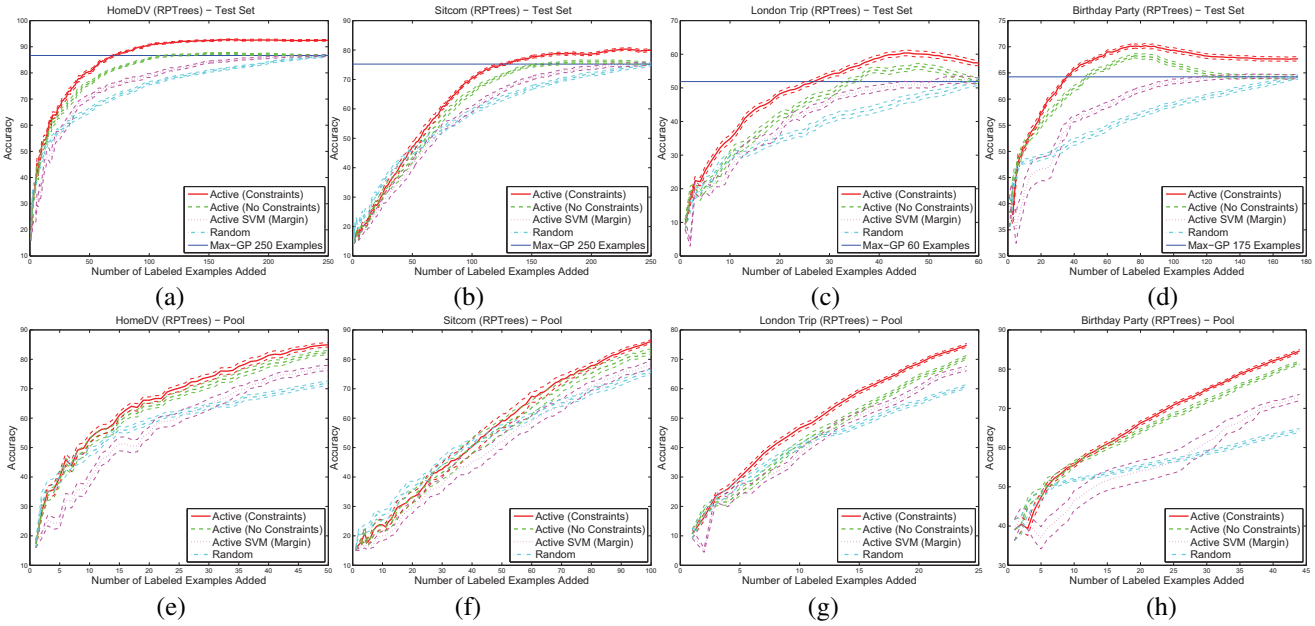

Figure 4. Recognition performance on the held-out test set (top row) and the pool of data where active learning algorithm is operating (bottom row) for (a) and (e) HomeDV, (b) and (f) Sitcom, (c) and (g) London Trip and (d) and (h) Birthday Party datasets. The above graphs show mean performance over 50 different runs and the dotted lines signify standard error. These plots highlight that modeling the constraints is advantageous for the purpose of active learning. These results used RPTrees representations. For results on Eigenspace representation please see the supplementary material.

The results using the RPTrees representation[3] on the held out set are included in Figure 4 (top row). The blue horizontal line is the average recognition rate obtained using a Gaussian-Process classifier trained on all the active learning data. As such, it upper-bounds the recognition rate we expect from all the algorithms (without constraints) when all examples have been selected. This property can be seen in the figures with the algorithms except the one which uses the constraints reaching this point. The two main points to note, however, are: (1) the algorithm which uses the constraints reaches an even higher level of performance through the incorporation of this additional information, and (2) the recognition rate of the algorithm which uses the constraints increases the fastest as examples are added.

The results on the pool of data used for active learning are shown in Figure 4 (bottom row) and Table 1. We truncate these plots after 40% of the examples have been labeled to make it easier to see the difference. When extended, all algorithms achieve 100% accuracy and the proposed algorithm always stays on top of the other curves. As with the held-out validation data, the main point to note is that the recognition rate of our algorithm with the constraints increases the fastest as examples are added. The results show that the method with constraints is choosing better examples.

[3]In this section we only report results on the RPTrees due to space constraints. For results using Eigenspace representation please see the supplementary material.

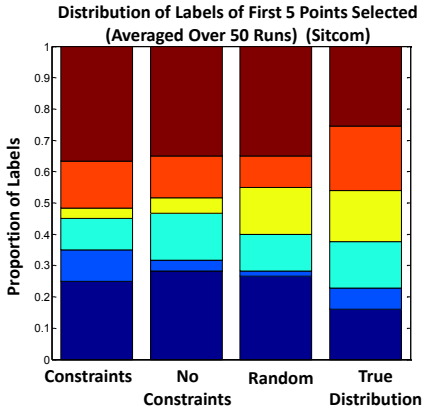

Figure 5. The distribution of the first five labels sampled according to the different active learning schemes (averaged over 50 runs). The different colors correspond to different classes and the length of each color segment is in proportion to the number of sampled examples belonging to a particular class.

In Table 1 we include numerical results which consist of the number of examples that need to be added to achieve 75% accuracy. Again, averages are reported over 50 trial runs, along with the standard error. As above, these results confirm that the recognition rate of our algorithm with the constraints increases the fastest as examples are added, reaching the 75% rate with significantly fewer examples.

Next, we analyze the behavior of different active learning selection strategies. Figure 5 shows an average distribution over the class of the first five selected examples

Table 1. Number of labels required to achieve 75% accuracy on the complete active learning pool. The results are averaged over 50 runs and the standard error is shown in the parenthesis.

	Our Model (Constraints)	**Our Model (No Constraints)**	**SVM (Margin)**	**Random**
HomeDV	**29.3±1.2**	33.2±1.1	40.4±1.9	55.5±2.1
Sitcom	**71.4**±1.5	76.8±1.8	87.5±2.2	94.8±1.8
London Trip	**24.8±0.4**	27.5±0.5	30.7±0.6	36.0±0.6
B'day Party	**30.9±0.5**	34.6±0.7	48.2±1.5	73.9±1.5

for the different active learning schemes and compares it to the true distribution of the labels in the data. The figure is generated by averaging distribution over the 50 runs. Each color corresponds to a particular class and the length of a color segment in one bar corresponds to the proportion of examples in a particular class that were picked by the active learning strategy. We see that the distribution over labels for the random selection strategy closely matches the true underlying distribution. On the other hand, the active learning scheme using the proposed model deviates fairly significantly. There is also a significant difference in distributions between the constrained and the unconstrained model. Note that the amount of 'yellow' labels sampled by the constrained model is fairly different when compared to the true distribution of the labels. The results in Figures 4 and 5 suggest that active learning with the proposed method to model constraints can guide the selection to achieve significant gains.

Inferring labels for 250 unlabeled images using 250 training images takes around 0.45 seconds on a dual 3.0 GHz 64-bit Intel-Xeon machine. Our implementation is in MATLAB except the loopy belief propagation, which is implemented in C.

4. Conclusion

We have extended the active learning paradigm to include constraints. We used our framework to develop an algorithm that chooses which faces a user should tag first in face recognition assisted face tagging scenarios. We considered two types of constraints. Non-match constraints mean that two examples must have different labels. An example of such a constraint occurs when two people appear in the same photograph. Match constraints mean that two examples must have the same label. An example of such a constraint occurs when faces are tracked in video. We demonstrated that the addition of such constraints can improve the performance of active learning.

We considered the scenario where one face is presented to the user at a time. A variety of other user interfaces are possible, [1, 2, 5]. Instead of myopic "one face at a time labeling" we also seek to investigate non-myopic selective sampling where an optimal subset of unlabeled faces are selected. Other future directions include considering other contextual cues to induce more constraints and applying the framework to other novel scenarios.

References

[1] http://picasaweb.google.com/.

[2] http://www.apple.com/ilife/iphoto/.

[3] Y. Abramson and Y. Freund. Active learning for visual object recognition. Technical report, UCSD, 2004.

[4] E. Y. Chang, S. Tong, K. Goh, and C. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*, 2005.

[5] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *CHI*, 2007.

[6] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3), 1997.

[7] A. Holub, P. Perona, and M. Burl. Entropy-based active learning for object recognition. In *CVPR workshop on Online Learning for Classification*, 2008.

[8] T. Kanade. *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, 1973.

[9] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with Gaussian Processes for object categorization. In *ICCV*, 2007.

[10] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian Process method: Informative vector machines. *NIPS*, 2002.

[11] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4), 1992.

[12] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *CVPR*, 2009.

[13] C. E. Rasmusen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[14] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *PAMI*, 25(12), 2003.

[15] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang. A face annotation framework with partial clustering and interactive labeling. In *CVPR*, 2007.

[16] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *ICML*, 2000.

[17] M. Turk and A. Pentland. Face recognition using eigenfaces. In *CVPR*, 1991.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[19] J. Wright and G. Hua. Implicit elastic matching with randomized projections for pose-variant face recognition. In *CVPR*, 2009.

[20] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *ICCV*, 2003.

[21] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, pages 399–458, 2003.