Exploring Album Structure for Face Recognition in Online Social Networks

Jason Hochreiter[a,*], Zhongkai Han[a], Syed Zain Masood[a], Spencer Fonte[a], Marshall Tappen[a]

[a] *University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816*

Abstract

In this paper, we propose an album-oriented face-recognition model that exploits the album structure for face recognition in online social networks. Albums, usually associated with pictures of a small group of people at a certain event or occasion, provide vital information that can be used to effectively reduce the possible list of candidate labels. We show how this intuition can be formalized into a model that expresses a prior on how albums tend to have many pictures of a small number of people. We also show how it can be extended to include other information available in a social network. Using two real-world datasets independently drawn from Facebook, we show that this model is broadly applicable and can significantly improve recognition rates.

Keywords: face recognition, online social networks, structural SVM

1. Introduction

Traditional face recognition systems have relied on image features to identify the individuals in photographs. The advent of popular social networks, such as Facebook, which host photo albums and make it possible to tag photos with user identities, adds a new dimension of data that can be used to help identify faces.

In Facebook, as in most photo management services, photos are grouped into albums. These albums are a rich source of information because they often correspond to trips, events, or specific groups of people. In this paper, we show how the organizational structure of photos into albums can be used to significantly increase recognition accuracy. In addition, as will be shown in Section 3, our model based on using album information can be applied to significantly more pictures than other models that exploit co-occurrence in photos, such as [1].

Much of the contribution of this work lies in modeling how individuals tend to co-occur in photo albums. The basic model is constructed with the underlying idea that albums tend to contain multiple photos of a small number of people, such as an album containing photographs from a trip. An album may contain many photos, but it is likely that the individuals pictured in the album will be dominated by the small number of people that participated in the event.

After introducing this basic model, we will then show how it can be improved and extended by considering other factors such as previous co-occurrence in an album, friendship information, and the identity of the person who uploaded the photo to the social network.

The rest of the paper follows this rough outline:

- Section 2 illustrates the difficulties of working with data from publicly available social networks.
- Section 3 shows that a CRF model based on limiting the number of individuals appearing in an album is useful for a significant portion of photos on Facebook. This section will show how this model can also be applied more widely than just modeling co-occurrence in photographs. Following this, Section 4 discusses related work.
- Sections 5 through 7 describe the design of the model, inference with the model, and the training procedure for the model.
- Section 8 describes experiments showing how this model significantly increases recognition performance. Most importantly, for the two datasets we test, this approach increases accuracy by around 20% over a baseline classifier.
- In Section 9, we describe a simple stochastic coordinate descent approach to learn the model parameters. That a technique having such low complexity requirements can achieve comparable results illustrates the flexibility of our model.

2. Reproducible Research on Social Network Data

As will be discussed in Section 8, we validate our methods on real data gathered from Facebook. One of the difficulties in working with social network data is the ability to share that data. Common datasets, such as the PASCAL challenge [2] and the Middlebury Stereo database [3] have facilitated significant advances in vision technology. However, sharing social network data is problematic due to privacy issues since it involves sharing the information of both users and their friends on social networks.

While it could be argued that anonymizing the type of data used for face recognition by various transformations could protect the identities of users, researchers have been effective at

*Corresponding author
Email address: jasonhochreiter@gmail.com (Jason Hochreiter)

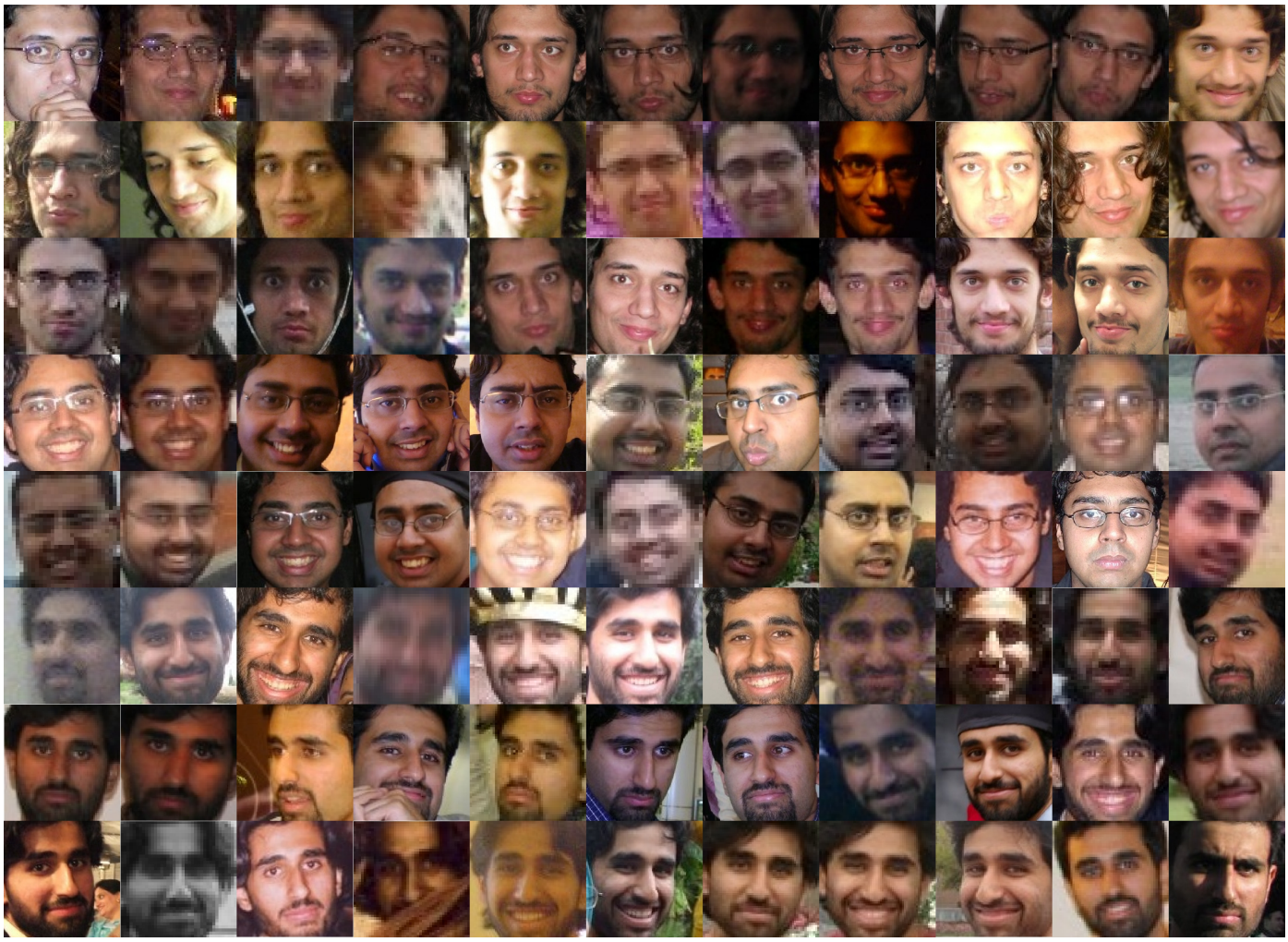

Figure 1: Sample images for three users from our dataset. The dataset contains a variety of facial poses, expressions, and image quality.

de-anonymizing data. In 2008, researchers de-anonymized significant portions of a dataset released by Harvard [4]. Given the constant threat of having anonymized data cracked, future efforts are required to gather realistic social network data that can be distributed.

To more strongly validate our results, we used two different datasets downloaded from the accounts of different individuals and two implementations of our algorithms created by different members of our group. As will be reported in Section 8, both experiments confirm the benefits of the proposed approach.

An example of facial images obtained from three users appears in Figure 1. These are raw images obtained from Facebook; there is no guarantee of quality or consistency of facial pose and expression in the images.

3. The Applicability of an Album Prior

A key question facing this work is whether users tend to organize albums in a way that makes this prior useful. Our study on photo albums in Facebook indicates that a prior based on the assumption that albums tend to contain multiple photos of a small number of people is much more applicable than a model that relies on co-occurrence inside a photo, such as [1]. To evaluate the usefulness of this prior on the occurrence of individuals in albums, we used the Facebook API to download all pictures visible to a single user's account, similar to [1]. In total, we were able to capture 8078 pictures containing 2849 different people across 1649 albums. In all of the pictures, we only considered faces that had been tagged by a user. To ensure the accuracy of tags, we applied the OpenCV face detector to each tagged photograph we downloaded [5]; if the detector did not indicate the presence of at least one face in a photo, we discarded it. In total, we collected 11724 facial images.[1]

In this collection of photographs, over 5735 photographs, or 71% of the photographs, only contained one tagged face. Presuming that, for the vast majority of photographs, all of the faces have been tagged, this means that a model based on co-occurrence inside a photograph [1] would help with recognition in roughly only 29% of photographs.

In measuring the applicability of our model, we set a high standard for its usefulness by assuming that an album prior

[1]This data eventually became the first of our two datasets. See Section 8.1.

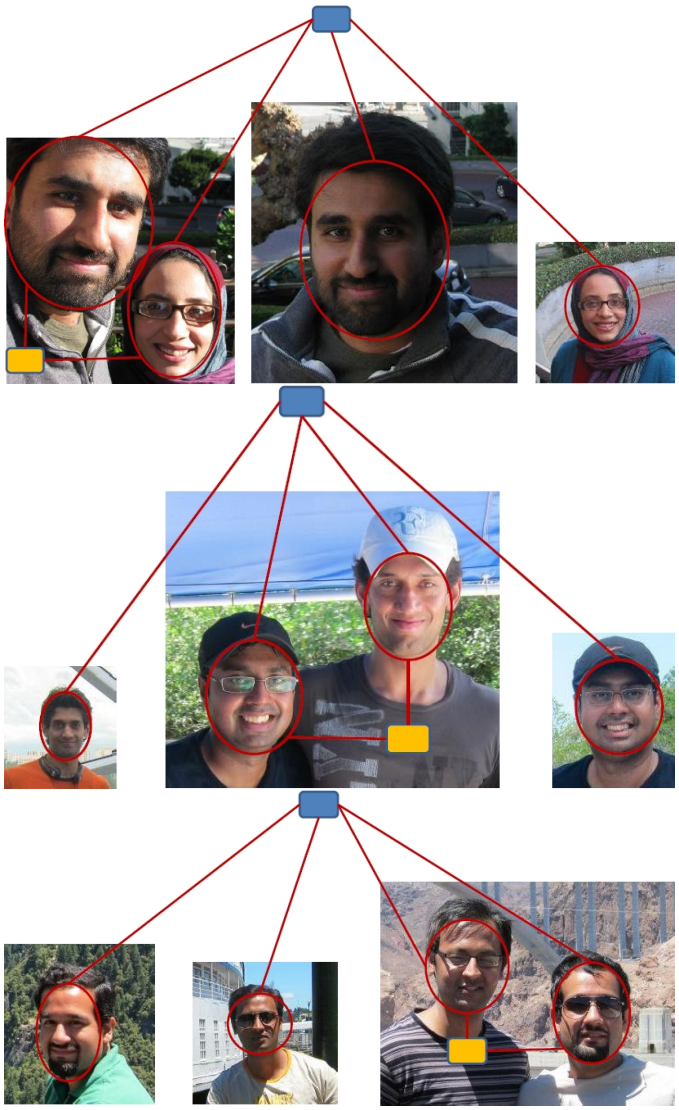

Figure 2: The factor graph illustrating the importance of modeling how people occur in albums instead of just pictures. Each row corresponds to photographs taken from a specific album. A model based on just photograph co-occurrence, shown in yellow, is only able to work with one of the three photographs, while a model based on occurrence in an album can help recognize faces in all three pictures.

would only be useful if there are at least twice as many photographs in an album as people occurring in that album. Despite this high threshold, we found that an album prior could still be applied to albums containing 57% of the photos in our dataset. Roughly speaking, a prior based on limiting occurrence in albums can be applied to nearly twice as many photographs as a prior just based on co-occurrence inside an photograph, such as the model in [1].

In Section 8.4, we consider the scenario in which this high standard is not met. Training and testing on all albums – for some of these albums, the number of labels is more than half of the number of photographs – results in only a 2% performance degradation as compared to training and testing on only albums that satisfy this threshold, indicating the applicability of our approach.

Figure 2 demonstrates why an album-based model can be applied more widely. Each row corresponds to photographs taken from a specific album. For each row, only one photograph has two people in it. Thus, a model based on photograph co-occurrence could only be applied to one of the photographs, while all three could be considered in an album-based model.

An important contribution of this work is that we show a model where the album structure can be considered in an efficient inference scheme.

3.1. *Quality of Downloaded Data*

The metadata provided by downloaded photographs and albums may be incomplete or inaccurate. The former case is often due to privacy concerns; the data we are able to download may be limited by the privacy settings of particular users on Facebook. We find this to be of little consequence: even without the capability to download all possible photographs of a given user, we still obtained a large amount of data for training and testing purposes.

Moreover, metadata such as the identity of persons tagged in photographs may be incorrect. Based on our observations, the most typical cause of inaccurate tags is that a user tags something *other than* a person with a photograph; for instance, in Facebook, as tagging a user in a photograph sends him or her a message, some users intentionally tag a user who does not appear in a particular photograph in order to ensure that this user sees the image. For example, for photographs at a party for which participants brought in a special food dish, one user tagged each dish with the name of the person who prepared it. To prevent this case, we applied the OpenCV face detector to each image, as discussed previously, keeping only those photographs with a confirmed face.

However, this does not address the issue of improperly tagged faces. While it may be possible to remove noisy data – e.g., by clustering – we assume that a tagged face has been tagged correctly. While the presence of inaccurate data may affect results, we still obtained significant improvements in classification accuracy.

4. Related Work

The most similar work on this problem is the model of co-occurrence in photographs proposed by Stone et al. [1]. In this model, photographs gathered from Facebook were used to model the frequency with which individuals appeared together in photographs.

A model like that used in [1] would be difficult to extend to album-level co-occurrence because modeling co-occurrence in photographs implicitly assumes that individuals are not repeated. However, in albums, individuals may be repeated many times. In addition, the model in [1] was based on a fully-connected graph between individuals in the photograph. This was manageable because only a small number of people appeared in any photograph. Extending these fully-connected pairwise relationships between all faces appearing in an album would

lead to a large, fully-connected graph representing the album. The density of the edges in the graph could pose serious challenges for inference in the model.

Instead, we propose a model based on penalizing album labelings by how many individuals appear in the album, similar to the label cost from [6]. This makes it possible to perform inference with an efficient, greedy approximation that performs well. Moreover, in [1], only photos containing exactly two faces are considered; using an album-level model has no such restriction.

There have also been many studies conducted in the recent past on the problem of person recognition that use contextual information for improved results. Predominantly, two kinds of contextual information are used: social context information and personal context information.

Personal contextual information, such as hair and clothing, can provide useful information to characterize a person, because such features do not change during a short period of time, which means the same person tends to wear the same clothes or has the same hairstyle across different pictures. Many researchers [7, 8, 9, 10] combine these features with face recognition results to improve recognition accuracy. However, the pictures in online communities might be captured over a long period of time and uploaded at the same time, so such personal information could be unstable.

Compared with personal contextual features, social contextual information captures the relationships between people. This is believed to be more robust because relationships change little over a long period of time. The most widely used social feature is co-occurrence [11, 9, 12]. In [11], event and location groupings of pictures are obtained based on time and locations of photographs. Later, picture co-occurrence is used to provide a relevant short list of identities likely to appear in new photographs. Gallagher and Chen [12] use pairwise co-occurrence to calculate the grouping prior distribution, which models the probability of a group of people appearing in the same picture. Zhao and Liu [9] first cluster pictures into albums based on time and then combine picture co-occurrence and personal contextual information with face recognition results to achieve recognition accuracy for each album. These methods will narrow down the candidates to a small group of people, such as family members [9, 10] or the people going on the same trip [8]. For an online community, however, the candidates are all the people in the online community related to the uploader, and effectively narrowing down the list of candidates is the main concern in this paper.

The embedded social network in online communities provides useful information for recognition tasks. In [1], social network information such as friendship, pairwise co-occurrence and face recognition score are incorporated into a CRF model. Specifically, each detected face in one picture is regarded as a node in the graph model, and the edges between nodes are the social relationships. In this way, the total energy is the weighted sum of all the potentials in the graph.

Both the CRF model of [1] and our model use similar social contextual information, but the two models use this information in a different manner. The application scope is different: the CRF model tries to model the relations within a picture and ignores the relations between the pictures, while our model focuses on a bigger view – the entire album. In addition, the model presented in [1] does not contain links between images; only intra-image relationships are captured. In our model, these costs generate a factor that connects to all images in the album.

Caption text is used by [13, 14, 12] to facilitate recognition. Although picture captions are available in Facebook albums, their relevance and reliability is questionable, especially when considering all possible labels.

4.1. Comparison to [1]

As will be explained in Section 6, we make use of various social metadata – including encoding the notion of friendship (as allowed within the social network), co-occurrence, and uploaders – but we use these in the context of albums rather than individual photographs, as in [1].

According to [1], Facebook users co-occur in photographs with only 9% of their friends, on average; we have observed similar trends. Thus, considering co-occurrence can help limit the number of candidate friends that must be considered within photographs. We build on this phenomenon and take it a step further: in [1], the fact that photographs uploaded to social networks are arranged in albums – essentially the next level in the hierarchy – is not considered. Albums cover specific events involving specific people at specific times and locations. It is, therefore, natural to assume that not all of these 9% of friends that may co-occur in photos will be present in a single album. Leveraging this information helps reduce the friends candidate list even further.

This leads to our introduction of the personal label cost, explained in Section 6.1. This relies on the notion that albums tend to have photographs of a limited number of people. We qualitatively evaluated this claim: Figure 3 shows the distribution of the number of people tagged in albums for our second dataset (see Section 8.1). Clearly, most albums contain very few people, and only a small number of albums feature more than 15. On average, around 6 unique people are tagged in an album.

While our datasets represent only a small sample of Facebook data, they provide an insight as to how photo co-occurrence is related to albums and how album-level co-occurrence can improve over the model of [1].

5. Album-Oriented Face Recognition

In this section, we describe how to construct a model based on occurrence in albums. This model is based on introducing a label cost, similar to the label cost described in [6]. Essentially, a cost is assigned to each label present in the album, regardless of whether the label appears more than once, effectively limiting the number of people appearing in an album. In this section, we first present a framework for incorporating data cost and label cost; then, we introduce the specific data cost used. Section 5.3 will introduce the inference strategy used.

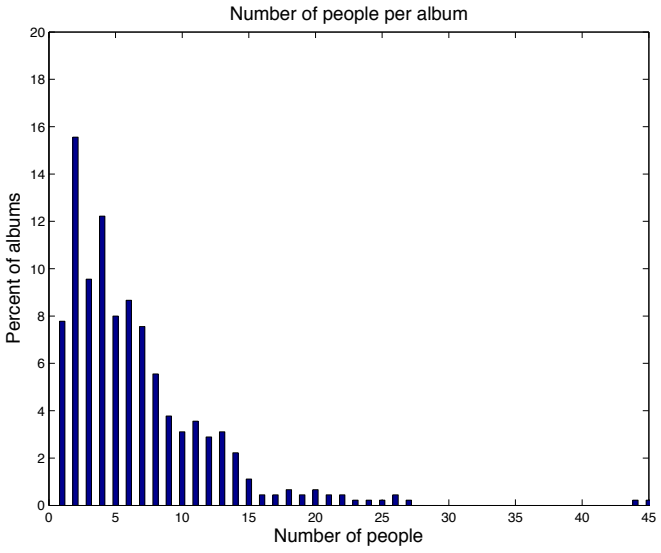

Figure 3: A visualization of the number of unique people that appear in an album. The y-axis shows the percentage of albums that contain the number of people represented by the x-axis. On average, these albums contain around 6 unique people. This data was computed using the raw data downloaded for our second dataset before pruning it (see Section 8.1).

5.1. Face Recognition Using Label Cost

The central goal is to correctly label the faces in an album F. Thus, the album F can be thought of as a set of face images. Following the notation used in [15], the vector $\vec{y}$ will denote the labelings of each face image in one album, making its fth entry y_f the label of the fth facial image in album F. A traditional face recognition system can be described by the energy function

$$E(\vec{y}) = D(\vec{y}; \vec{x}) = \sum_{f \in F} D_f(y_f; \vec{x}_f), \tag{1}$$

where D_f is the data cost of assigning label y_f to face $f \in F$. The vector $\vec{x}_f$ is the vector of features gathered from the image corresponding to the facial image f.

Our model builds on this by adding costs based on who is in a particular labeling $\vec{y}$, ignoring how many times that label appears in $\vec{y}$. With a slight abuse of notation, we will use the notation $l \in \vec{y}$ to denote that the label l appears at least once in the label vector $\vec{y}$. Formally, we express this as

$$l \in \vec{y} \Leftrightarrow \exists f \text{ s.t. } y_f = l. \tag{2}$$

Following [6], we will refer to costs based on this type of membership as label costs. Suppose there are only N candidates $\{l_1, \ldots, l_N\} \in \mathcal{L}$ for each label y_f. The basic system in Equation (1) will be extended as

$$E(\vec{y}) = \sum_{f \in F} D_f(y_f; \vec{x}_f) + C(\vec{y}), \tag{3}$$

where $C(\vec{y})$ is the label cost of album labeling $\vec{y}$. This label cost combines several types of social network information. The specific form of this cost will be introduced in Section 6.

This model is interesting in that recognition is simultaneously performed on all images in the album.

5.2. Data Cost

The data cost can be thought of as the result from a baseline, image-only, face recognition system. In this paper, the data cost is implemented using the high-dimensional V1-like features proposed in [16, 17, 18], who showed that excellent face-recognition results could be achieved by linear classification of very high-dimensional feature vectors extracted from the image.

Assuming that $\mathcal{L}$ denotes the set of possible labels for each face and that there are N labels in $\mathcal{L}$, the data cost is computed using the negative log of a soft-max function:

$$D(\vec{y}; \vec{x}) = \sum_{f \in F} -\log \frac{e^{-V_{y_f}(\vec{x}_f)}}{\sum_{l=1}^{N} e^{-V_l(\vec{x}_f)}}, \tag{4}$$

where $\vec{x}_f$ is vector of features extracted from an image f. The outer summation is computed over all images in the album F. The functions $V_1(\vec{x}), \ldots, V_N(\vec{x})$ are linear combinations of the feature vector $\vec{x}$ with weight vectors, as is standard in linear classifiers.

Because the length of $\vec{x}$ is huge, containing 86400 entries, we use a multi-class generalization of the LogitBoost and GentleBoost algorithms [19], combined with regression stumps, to greedily select a subset of features. In our experiments, we found that a classifier using only 400 of the 86400 features performed nearly as well as linear classification using the entire feature vector.

5.3. Inference

Inference in Equation (3) is an NP-hard problem. However, this model is similar to the well-studied *uncapacitated facility location (UFL)* problem [20], so it is possible to use a greedy algorithm which would yield a $O(\log |\mathcal{L}|)$-approximation ([21, 20]) in this case. As we will show, our problem is not a standard *UFL* problem as the label costs will depend on each other when we introduce the social label cost. Although there is no theoretical guarantee that a greedy algorithm will give a good approximation, a greedy method worked well in our experiments.

This greedy algorithm, which is described in detail in Algorithm 1, operates by adding one label to the album that maximizes the energy function into Equation (3) at each iteration. For each iteration, the greedy method scans all the available candidates and selects the best one. It stops when adding new candidate labels does not result in further improvement.

6. Label Cost

The central idea of this paper is to use the label cost to effectively constrain the number of labels in an album, so it is important to assign an appropriate label cost to different labels. In this section, we first present how to construct a label cost; after that, we will illustrate each component in detail.

The total cost for adding an individual to the album is the combination of two costs: a personal label cost and a social label cost. The basic idea behind this strategy is that every label should pay its personal cost to enter into the album. The social

Algorithm 1 Greedy approximation

Define:
Q: queue containing all the candidate labels in $\mathcal{L}$
l_i: ith label in Q
L: the set of all labels appearing in the album
$\vec{w}$: learned weight vector (Section 7.2)
$\Psi(\vec{x}, \vec{y})$: feature vector depicting input/output relation (see Sections 7.1 and 7.2 for details)

Initialize:
$L^* = \emptyset$
$E^* = -\infty$

while $Q \neq \emptyset$ **do**
  for $i = 1$ *to* $|Q|$ **do**
    $L = L^* \cup l_i$
    $E_i = \max_{\vec{y}} \vec{w}^T \Psi(\vec{x}, \vec{y})$
  end for
  $\hat{E} = \max_i E_i$
  $\hat{l} = l_i$
  if $E^* < \hat{E}$ **then**
    $E^* = E_i$
    $L^* = L^* \cup \hat{l}$
    Eject(Q, $\hat{l}$)
  else
    break;
  end if
end while
return: L^*

label cost represents the compatibility of different labels in the album. Formally, we express the total cost for a label as

$$C(\vec{y}) = C_{personal}(\vec{y}) + C_{social}(\vec{y}), \tag{5}$$

where $C_{personal}(\vec{y})$ is the personal label cost for each label and $C_{social}(\vec{y})$ represents the social cost for incorporating label l. The costs are described in the following sections.

6.1. Personal Label Cost

The personal label cost expresses the idea that a limited number of people should appear in an album. We define this cost as

$$C_{personal}(\vec{y}) = \sum_{l \in \mathcal{L}} \lambda \mathrm{I}(l, \vec{y}), \tag{6}$$

where $\mathrm{I}(l, \vec{y})$ is an indicator function defined as

$$\mathrm{I}(l, \vec{y}) = \begin{cases} 1 & \text{if } l \in \vec{y} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This definition ensures that every label entering into the album pays its cost, which is the weight λ learned (see Section 7).

It is also possible to consider a per-person label cost: that is, for each label $l \in \mathcal{L}$, there is an associated label cost λ_l. As our two datasets contain 25 and 15 labels, respectively, this results in a dramatic increase in the number of parameters that need to be learned. We discuss this further in Section 9.

6.2. Social Label Cost

The social label cost function is similar to the personal cost formulation but is based on information from the social network. Given a particular labeling $\vec{y}$, the social cost of including a label l in the labeling is computed by summing costs representing the interaction between label l and all other labels in the labeling $\vec{y}$, which are formally expressed as

$$C_{social}(\vec{y}) = \sum_{l \in \mathcal{L}} S(l, \vec{y}) \mathrm{I}(l, \vec{y}), \tag{8}$$

where

$$S(l, \vec{y}) = \sum_{j \in \mathcal{L}} (\alpha_f C_f(l, j) + \alpha_c C_{co}(l, j)) \mathrm{I}(l, \vec{y}) + \alpha_u C_u(l). \tag{9}$$

The three components introduced below are the social costs derived from the social network:

- **Friendship Cost**. This cost measures whether individuals co-occurring in the album are friends:

$$C_f(i, j) = \begin{cases} 0 & i \text{ and } j \text{ are friends} \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

- **Co-Occurrence Cost**. This cost is similar to the friendship cost but measures whether the individuals in an album have ever co-occurred in albums in the training data:

$$C_{co}(i, j) = \begin{cases} 0 & i \text{ and } j \text{ have co-occurred} \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

- **Uploader Cost**. The uploader cost uses the history contained in an uploader's previous photos. Like the previous two costs, it is based on an indicator function. Here, we define a potential which captures the relationship between an individual and the owner of the photo album[2]:

$$C_u(i) = \begin{cases} 0 & \text{if } i \text{ has appeared in images} \\ & \text{uploaded by the owner of } F \\ 1 & \text{otherwise.} \end{cases} \tag{12}$$

7. Training

Training the parameters for the structural model is a challenging problem. We use the Structural SVM (SSVM) because it can optimize parameter values even if inference can only return approximate solutions, as is the case in this model [22]. In this section, we will first introduce the SSVM method; after that, we demonstrate how to transform the energy in Equation (3) into the SSVM expression.

[2]This model assumes that all photos in an album have been uploaded by a single user.

7.1. Structural SVM

The SSVM deals with the general problem of learning a mapping from the inputs $x \in X$ to discrete outputs $y \in Y$ based on a training sample set $S = \{(x_i, y_i) | (x_i, y_i) \in X \times Y\}$. The label here is in a general form, which could be a numbered label in the case of multiclass classification or a parsing tree for a sentence in natural language parsing.

During training, the SSVM tries to learn a discriminant function over input/output pairs of the form

$$f(x, y, w) = w^T \Psi(x, y), \tag{13}$$

where Ψ generates a feature vector which depicts relations between inputs and outputs and w are model parameters that need to be learned.

A prediction is made by maximizing f over the response variable for a specific given input x. Formally, this can be expressed as

$$h(x) = \arg\max_{y \in Y} f(x, y). \tag{14}$$

To learn this map, SSVMs solve the following quadratic program

$$\min_{w, \xi \geq 0} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i,$$

$$\forall i, \forall y \in Y \backslash y_i : w^T \Psi(x_i, y_i) \geq w^T \Psi(x_i, y) + \Delta(y_i, y) - \xi_i,$$

where $\Delta(y_i, y)$ is the loss function indicating how far $h(x_i)$ is from the true output y_i.

Introducing a constraint for every wrong output is typically intractable, and Joachims proposed a cutting plane algorithm which defines a separation oracle and finds the most violated label, constructs a sufficient subset of the constraints using these labels and iteratively solves the QP only over this subset; this guarantees polynomial time runtime and correctness [15].

7.2. Learning the Weight Vector

The parameters in our model that need to be learned are the personal label cost weight λ and the three weights α_f, α_{co} and α_u for the social costs. These parameters can be optimized using the SSVM training procedure by expressing the energy in Equation (5) as the inner product of the weight vector $\vec{w}$ and feature vector $\Psi(\vec{x}, \vec{y})$. This can implemented by representing $\vec{w}$ as

$$\vec{w} = [1, \lambda, \alpha_f, \alpha_{co}, \alpha_u]. \tag{15}$$

The vector $\Psi(\vec{x}, \vec{y})$ is defined as

$$\Psi(\vec{x}, \vec{y}) = [D(\vec{y}; \vec{x}), C_{person}(\vec{y}), Z_f(\vec{y}), Z_{co}(\vec{y}), Z_u(\vec{y})]^T, \tag{16}$$

where

$$Z_f(\vec{y}) = \sum_{l \in \vec{y}} \sum_{j \in \vec{y}} C_f(l, j) \tag{17}$$

$$Z_{co}(\vec{y}) = \sum_{l \in \vec{y}} \sum_{j \in \vec{y}} C_{co}(l, j) \tag{18}$$

$$Z_u(\vec{y}) = \sum_{l \in \vec{y}} C_u(l) \tag{19}$$

For the small number of parameters in this model, we also had success with a randomized coordinate ascent approach, though the structural SVM solution was much faster; see Section 9.

8. Experiments

In this section, we present experimental results conducted on two datasets collected from Facebook. In the following section, we will first give a detailed description of the datasets; then, we compare our results with a baseline system. After that, we combine our method with the model proposed in [1].

8.1. Datasets

As described in Section 1, we replicated our experiments on two different datasets, using different implementations of our algorithms. These experiments were conducted separately on each dataset.

Both datasets were gathered from the Facebook accounts of volunteers, using a downloading application similar to that used in [1]. The datasets differed in the users used to capture the photographs. In the first dataset, the photographs were gathered from all of the albums visible to one user. The second dataset was gathered from a larger set of volunteers that agreed to let us access photographs in their Facebook accounts, using the same permissions mechanism that any Facebook application can use to access personal data. We accessed all photos in albums available to our downloading application. We also stored available social network information, including friendship relationships, the identity of the uploader of each picture, and the way in which people have co-occurred in tagged photographs.

Not surprisingly, although these applications gave us access to photographs of hundreds of individuals, most individuals only had a handful of photographs. To ensure that we had enough data to properly perform both training and evaluation, we culled the dataset to include only individuals with a large number of samples available. This resulted in one dataset with 1951 facial images of 25 people across 481 albums and a second dataset with 1994 facial images of 15 people across 234 albums. Both datasets were constructed as detailed in Section 3; we downloaded photos using the Facebook API and kept only tagged images that the OpenCV face detector labeled as faces.

Each dataset is then partitioned into three parts:

1. A set of albums and images to be used to train the weights for the data cost. These images are used to train the linear classification weights for the image-only training.
2. A second set of albums and images that will be used to train the weights for the personal label and social costs. It is necessary to use a second training set because the training process used to learn the linear weights can separate the training data perfectly. If the training data in the first partition is used, then the data cost will be unrealistically accurate, and not enough weight will be given to the personal label and social costs.
3. A third set to be used as a testing set. In our experiments, this test set included 454 images of faces across 81 albums in our first dataset and 515 images of faces across 78 albums in our second dataset.

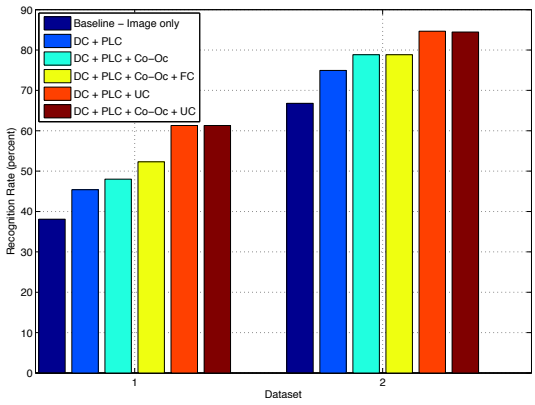

Figure 4: This figure shows a visualization of the improvement provided by incorporating label-based priors into recognition. DC refers to the Data Cost, PLC refers to the Personal Label Cost, Co-Oc refers to the Co-Occurrence Cost, FC refers to the Friendship Cost, and UC refers to the Uploader Cost. In both datasets, adding a personal label cost significantly increases recognition performance over the image-only baseline face recognition system (described in Section 5.2). Adding additional social costs also leads to further improvements. In both of the datasets, over 400 photographs were tested.

The albums are divided chronologically to simulate how photographs would arrive in a social network: the first partition contains the oldest albums, while the third contains the newest.

8.2. Results

Figure 4 summarizes our results for both datasets. The baseline bar in the graph shows the accuracy achieved using only the data cost, which is computed using a linear classifier and the image features extracted from each face image.

Implementing a personal label cost, as in Equation (3), leads to an improvement in recognition accuracy for both datasets. This cost is based on the number of labels present in the album. The accuracy for the first dataset is improved from 38.1% (baseline) to 45.4%, and the accuracy for the second dataset is improved from 66.8% (baseline) to 75.0%.

The remaining bars in Figure 4 show the accuracies of different combinations of social costs from Section 6.2. Most importantly, for both datasets, incorporating some form of a social label cost improves recognition performance considerably. Out of all of the social costs we explored, the uploader cost, which penalizes a potential candidate in a photograph if he or she has never appeared in any album uploaded by the owner of the photograph, seems to be the most important. The combination of the data cost, personal label cost, co-occurrence cost, and uploader cost improves accuracy on the first dataset from 38.1% to 61.3% and improves the accuracy on the second dataset from 66.8% to 84.5%.

Figure 5 shows the manner in which each social cost – taken individually – improves recognition performance on each dataset. The vertical axis shows the recognition rate on both datasets as the weight on a single cost in the model is increased. In these experiments, only a single cost is considered, in addition to the base cost from the image information.

Compared to the baseline of image-only recognition, adding even a single social cost leads to improvements for both datasets. While these costs related to social metadata continue to aid performance or level out as the weight increases, accuracy begins to decrease once the label cost becomes too high; when this happens, even correct labels can be heavily penalized and prevented from being considered for a given album.

The addition of the uploader cost in particular leads to significant improvements. The importance of the uploader cost is intuitive. A user who takes a photograph of a person and uploads it to Facebook is likely to do so again, regardless of their friendship status or previous album co-occurrence history. If we assume that an album is a collection of many pictures of a small group of people, it follows that all of a user's photographs – across all albums – are likely primarily pictures of a specific group of people. Moreover, this social cost is more broadly applicable, as it can be applied to albums containing only one person.

8.3. Consistency of Results Between Datasets

For both datasets, incorporating the label-based priors yields consistent improvements: around 7% for the first and 8% for the second. Likewise, the inclusion of additional social costs further improves recognition performance on the two sets, up to a maximum gain of 23% for the first and 18% for the second.

We observe, though, that simply using the image-based data cost produces varied results across the datasets. This variation is most likely due to the differing number of people present in each of these datasets and not a result of our learning methodology. As expected, the dataset featuring more people – in this case, the first – achieves a lower accuracy. However, comparable performance gains achieved on both datasets when utilizing the personal and social label costs indicate the usefulness of our system across various scenarios.

8.4. Albums With Many People

While the results in the previous section demonstrate that the addition of a label cost can significantly improve recognition results, this cost is designed for albums where the number of photos outnumbers the number of individuals. While we have shown that this behavior is common, we cannot expect it to hold across all albums.

To investigate this issue, we characterized each album with a ratio measuring the number of individuals in the album with respect to the number of facial images; we will refer to this value as the *IdentityRatio*:

$$\text{IdentityRatio} = \frac{|\mathcal{L}_F|}{|F|}, \tag{20}$$

where $|\mathcal{L}_F|$ is number of individuals appearing in album F and $|F|$ is the number of facial images in F. Because the set $\mathcal{L}$ is used to denote all possible labels, $\mathcal{L}_F$ denotes the labels in album F.

For the purpose of experimentation, we manually choose a threshold of 0.5 for the IdentityRatio and so only apply the label cost to albums having a value ≤ 0.5. The threshold is selected based on the observation that the label cost can be applied to

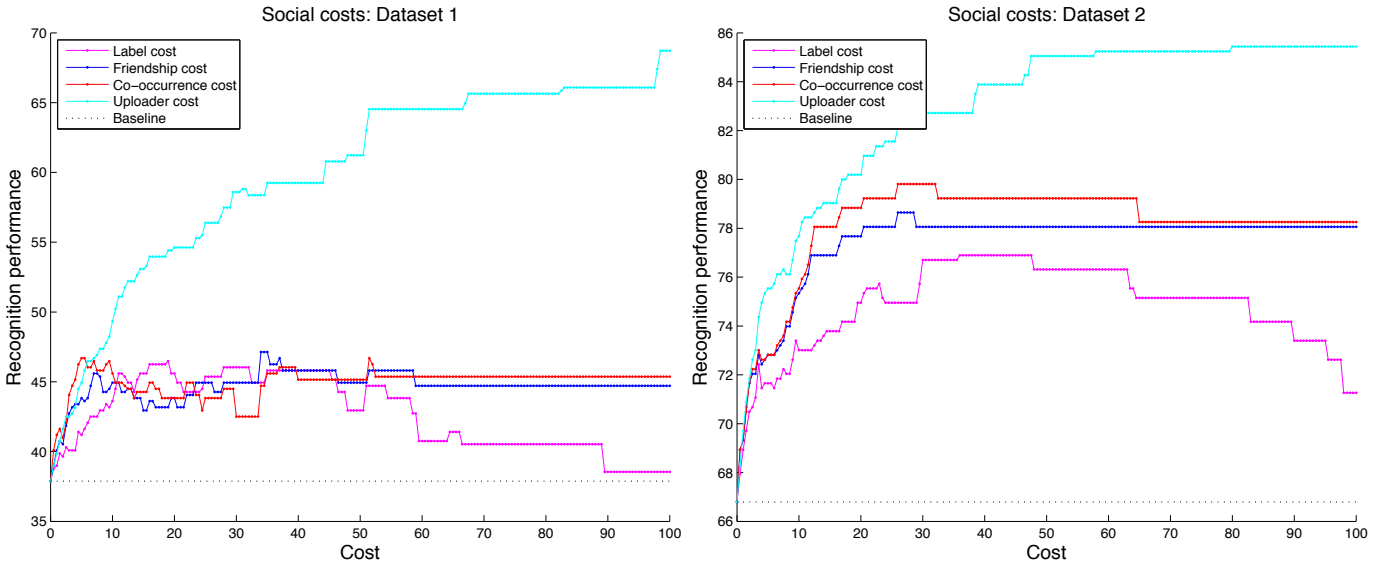

Figure 5: The effect of each social cost on recognition performance. The vertical axis shows the accuracy obtained as the weight assigned to a single cost (label and social costs) varies; this weight is shown along the horizontal axis. The left subgraph shows the results for the testing set of our first dataset, which contains 25 people across 454 facial images, while the right subgraph shows the results for the testing set of our second dataset, which contains 15 people across 515 facial images. In both graphs, the dotted line indicates baseline performance, which uses only image information.

the simplest possible album, which contains two facial images belonging to only one person.

Using this ratio, we can split our training sets into one set of albums well-suited for the label cost (i.e. those albums with IdentityRatio ≤ 0.5) and another set containing all albums, some of which could be problematic (i.e. some albums may have IdentityRatio > 0.5). The resulting performance difference is tiny: when training and testing on "good" albums, which are consistent with our label cost prior, we see only a very slight improvement – approximately 2% – over training and testing on the complete dataset. This suggests that incorporating the label cost does not induce instability into the recognition system.

8.5. Comparison with Image Co-Occurrence Model

Our use of a CRF model makes it convenient to use it to aid a model based on co-occurrence in photographs, similar to [1]. Accordingly, we culled our dataset to find photographs containing two individuals in our set[3]. This was difficult because a number of photos contained multiple individuals, but our tests are limited to a subset of individuals. Because of the restriction in finding photographs with two people, our training set was limited to 33 pictures. The testing set contained 32 for the same reason. While this set is too small to make statistically significant results, our preliminary experiments have shown that when these models can be combined, recognition rates improve by around 5%.

9. Stochastic Coordinate Descent

While the SSVM model detailed in Section 7.1 performs well on this dataset, it is designed for models where exact inference can be performed in the underlying model. If only approximate inference is possible, the SSVM learning optimization is not guaranteed to converge. As an alternative, we considered the use of a simple stochastic coordinate descent approach to optimize the parameters of the model without the complexity required by the SSVM. We found that this simple approach achieves competitive results across different collections of social costs and allows us to easily consider the use of per-person label costs. Each such experiment was performed independently of the others.

9.1. Method

Each experiment begins with the selection of available social costs to change as well as the form of label cost; there may be no label cost used, a single label cost for all labels, or a per-person label cost. Any cost not used is set to zero. For each iteration, a randomly chosen cost is modified by either adding or subtracting a random amount – though we ensure that no cost is of the wrong sign. On a given iteration, a set of costs that leads to improved performance on the training set is kept as the current "best" set of costs. Because of the complexity of the cost space we consider, we also allow an iteration with equivalent or worse performance to become the current best based on a random chance. Similarly, there is a small random chance that the costs are reset to the overall best costs. Using these two modifications allows us to escape local minima.

When running experiments, we run the optimization several times, each time initializing all of the social costs to zero, then taking the best performing result.

This success of this technique, shown in the results below, illustrates that there is significant flexibility in the type of optimization used to implement this system.

[3]The number of people per photograph was limited to two, as in [1], to make brute-force inference possible during training.

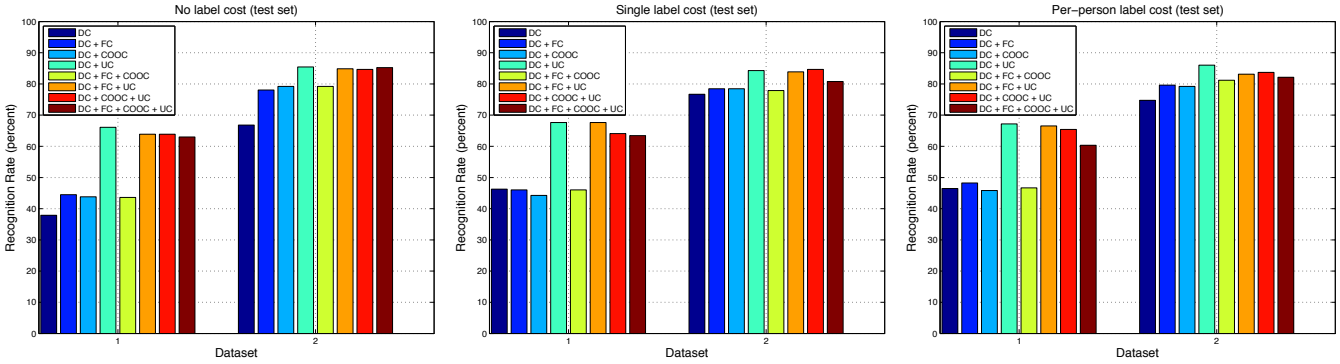

Figure 6: Comparison of test set recognition performance improvement provided by various combinations of social costs. Left: social costs with no label cost. Middle: social costs with a single label cost. Right: social costs with a per-person label cost. DC refers to the data cost, FC to the friendship cost, COOC to the co-occurrence cost, and UC to the uploader cost.

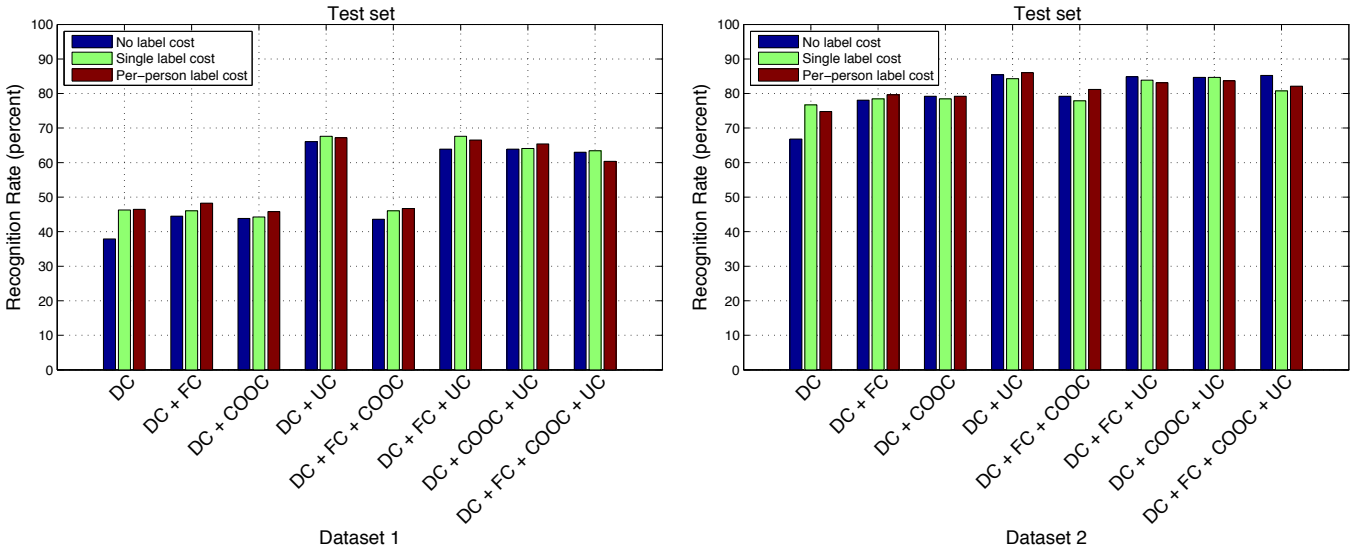

Figure 7: Comparison of recognition performance by type of label cost considered on the testing set of albums. The left and right graphs show the results for the first and second dataset, respectively. For each combination of social costs considered, the results using no label cost, a single label cost, and a per-person label cost are grouped together. DC refers to the data cost, FC to the friendship cost, COOC to the co-occurrence cost, and UC to the uploader cost.

9.2. Results

The recognition performance improvements resulting from each combination of social costs considered is shown in detail in Figure 6. In order, this figure demonstrates our experimental results using no label cost, a single label cost, and a per-person label cost for both datasets. For both datasets, the incorporation of some form of social metadata leads to improved results over the image-only baseline. The contribution of the uploader cost is especially evident here; the friendship and co-occurrence costs result in smaller improvements. Regardless of the form of label cost considered, the impact of each combination of social costs is relatively consistent over both datasets.

Figure 7 shows an alternate view of these results; here, we focus on the impact of the type of label cost used: no label cost, a single label cost, and a per-person label cost on the testing set of albums. For the training set, the use of some form of a label cost almost always led to improved recognition performance, regardless of any social costs considered. As expected, using a per-person label cost as opposed to a single label cost consistently resulted in improvements of a few percentage points across all experiments at the cost of dramatically increasing the number of parameters that need to be optimized. Comparable results hold for the testing set of albums; however, the single label cost and per-person label costs learned on the training set perform quite similarly on this set. In a few cases, the use of a label cost led to slightly degraded test accuracy – mainly when several social costs were included. Slower convergence is expected for such experiments, as there are additional degrees of freedom. For both sets of albums, the gain afforded using a label cost is especially large when no social costs are used.

We ran each experiment for 500 iterations. On average, each test converged in about 200 iterations.

10. Conclusions and Future Work

The album storage structure for photos in a social network provides a strong source of information regarding the identities of the people in those photographs. This paper has introduced a structural SVM-based system that is able to exploit this information. As shown in the experiments, utilizing this information leads to significant improvements in recognition accuracy.

In future work, we hope to better integrate models like the work of Stone *et al.* [1] that model co-occurrence inside pictures. We would also like to explore the use of other social features. For instance, relationship status may be used to further improve recognition performance, as we can expect spouses to appear in photos together frequently. Moreover, we would also like to consider "friend-of-friend" data – that is, if two users who are not friends share a mutual friend, they may be more likely to co-occur in a photograph or album than two users who are not friends and have no friends in common. Fortunately, our model can easily be extended to consider such information. However, friend-of-friend information is not easily obtained via Facebook due to privacy concerns.

11. Acknowledgements

This work was funded by NSF grants IIS-0905387 and IIS-0916868.

References

[1] Z. Stone, T. Zickler, T. Darrell, Autotagging Facebook: Social network context improves photo annotation, in: In Proceedings of CVPR Workshop on Internet Vision, IEEE, 2008.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results, http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html, 2011.

[3] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, 2002.

[4] M. Parry, Harvard's Privacy Meltdown - Harvard Researchers Accused of Breaching Students' Privacy, in: The Chronicle of Higher Education, 2011.

[5] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools (2000).

[6] A. Delong, A. Osokin, H. Isack, Y. Boykov, Fast approximate energy minimization with label costs, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 2173–2180.

[7] D. Anguelov, K. chih Lee, S. Gokturk, B. Sumengen, Contextual identity recognition in personal photo albums, 2007, pp. 1–7.

[8] J. Sivic, C. L. Zitnick, R. Szeliski, Finding people in repeated shots of the same scene, in: Proceedings of the British Machine Vision Conference, 2006.

[9] M. Zhao, S. Liu, Automatic person annotation of family photo album, in: Proc. International Conf. on Image and Video Retrieval, 2006, pp. 163–172.

[10] L. Zhang, L. Chen, M. Li, H. Zhang, Automated annotation of human faces in family albums, in: In MULTIMEDIA 03: Proceedings of the Eleventh ACM International Conference on Multimedia, ACM Press, 2003, pp. 355–358.

[11] M. Naaman, H. Garcia-Molina, A. Paepcke, R. B. Yeh, Leveraging Context to Resolve Identity in Photo Albums, Technical Report 2005-2, Stanford InfoLab, 2005. URL: http://ilpubs.stanford.edu:8090/694/.

[12] A. Gallagher, T. Chen, Using group prior to identify people in consumer images, 2007, pp. 1–8.

[13] T. L. Berg, E. C. Berg, J. Edwards, D. A. Forsyth, Who is in the picture, in: In Neural Information Processing Systems(NIPS), MIT Press, 2006, pp. 264–271.

[14] T. L. Berg, E. C. Berg, J. Edwards, M. Maire, R. White, Y. whye Teh, E. Learned-miller, D. A. Forsyth, Names and faces in the news, in: In Proc. CVPR, IEEE Computer Society, 2004, pp. 848–854.

[15] T. Joachims, T. Finley, C.-N. Yu, Cutting-plane training of structural SVMs, Machine Learning 77 (2009) 27–59.

[16] N. Pinto, J. DiCarlo, D. Cox, How far can you get with a modern face recognition test set using only simple features?, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 0 (2009) 2591–2598.

[17] N. Pinto, J. J. Dicarlo, D. D. Cox, Establishing Good Benchmarks and Baselines for Face Recognition, in: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Marseille France, 2008. URL: http://hal.inria.fr/inria-00326732/PDF/pinto-dicarlo-cox-eccv-2008-lfw_final.pdf.

[18] N. Pinto, D. Doukhan, J. J. DiCarlo, D. D. Cox, A high-throughput screening approach to discovering good forms of biologically inspired visual representation, PLoS Computational Biology 5 (2009).

[19] J. Friedman, T. Hastie, R. Tibshirani, Additive Logistic Regression: a Statistical View of Boosting, The Annals of Statistics 38 (2000).

[20] A. A. Kuehn, M. J. Hamburger, A heuristic program for locating warehouses, Management Science 9 (1963) 643–666.

[21] D. S. Hochbaum, Heuristics for the fixed cost median problem, Mathematical Programming 22 (1982) 148–162.

[22] T. Finley, T. Joachims, Training structural SVMs when exact inference is intractable, in: International Conference on Machine Learning (ICML), 2008, pp. 304–311.