Constrained Clustering and Its Application to Face Clustering in Videos

Baoyuan Wu *[1,2], Yifan Zhang[1], Bao-Gang Hu[1], and Qiang Ji[2]

[1]NLPR, CASIA, Beijing 100190, China
[2]Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

In this paper, we focus on face clustering in videos. Given the detected faces from real-world videos, we partition all faces into K disjoint clusters. Different from clustering on a collection of facial images, the faces from videos are organized as face tracks and the frame index of each face is also provided. As a result, many pairwise constraints between faces can be easily obtained from the temporal and spatial knowledge of the face tracks. These constraints can be effectively incorporated into a generative clustering model based on the Hidden Markov Random Fields (HMRFs). Within the HMRF model, the pairwise constraints are augmented by label-level and constraint-level local smoothness to guide the clustering process. The parameters for both the unary and the pairwise potential functions are learned by the simulated field algorithm, and the weights of constraints can be easily adjusted. We further introduce an efficient clustering framework specially for face clustering in videos, considering that faces in adjacent frames of the same face track are very similar. The framework is applicable to other clustering algorithms to significantly reduce the computational cost. Experiments on two face data sets from real-world videos demonstrate the significantly improved performance of our algorithm over state-of-the-art algorithms.

1. Introduction

We are interested in clustering faces into groups corresponding to individuals appearing in real-world videos. A successful solution of this problem can be applied to many fields, including automatically determining the cast of a feature-length film, content based video retrieval, rapid browsing and organization of video collections, automatic collection of large-scale face data set, etc. However, this task is challenging. In real-word videos, lighting conditions, facial expressions and head pose may drastically change the appearance of faces. Partial occlusions caused by objects such as glasses and hair also cause problems. In addition, the blur caused by motion and low spatial resolution are also common. Hence, the uncontrolled imaging condition in real-world videos raise many difficulties for face clustering.

Related works. Face clustering is a task of grouping faces by visual similarity. It is closely related to face recognition but has several different aspects. For face recognition, it is assumed that the number of persons and a training facial image data set are known beforehand. The data set consists of certain labeled facial images, which is used for training a classifier. When testing, each querying facial image can be classified by the trained classifier and the best matching person ID is returned. Hence, face recognition can be considered as a supervised classification problem. In contrast, since no labelled faces are provided, face clustering is considered as a unsupervised problem.

Although a large body of work has been conducted on face recognition, face clustering is a rather novel topic with few publications in the literature. Those works can be grouped into two categories: purely data-driven methods and clustering with prior knowledge. Most data-driven methods are fully unsupervised, and focus on obtaining a good distance measure or mapping raw data to a new space for better representing the structure of the inter-personal dissimilarities from the unlabeled faces[10, 11, 19, 12, 1]. Fitzgibbon and Zisserman [10] proposed an affine invariant distance measure to achieve robustness to face pose changing. To the best of our knowledge, this work is the first attempt for face clustering in movies. Then they [11] extended their work to a Joint Manifold Distance (JMD), where each subspace represents a set of facial images of the same person detected in consecutive video frames. Wang et al. [19] proposed a Manifold-Manifold Distance (MMD), in which a nonlinear manifold is divided into several local linear subspaces. MMD integrates the distances between pair of subspaces respectively from one of the involved manifolds. Hu et al. [12] introduced a between-set distance called Sparse Approximated Nearest Point (SANP)

*{bywu, yfzhang, hubg}@nlpr.ia.ac.cn, qji@ecse.rpi.edu

distance, where the dissimilarity of two sets is measured as the distance between their nearest points. Arandjelovic and Cipolla [1] clustered faces over face appearance manifolds in an anisotropic manifold space which exploits the coherence of dissimilarities between manifolds.

In addition to fully unsupervised methods, another kind of data-driven clustering methods try to utilize some partial supervision to help clustering. Prince and Elder [18] combined clustering with a Bayesian approach to count the number of different people that appear in a collection of face images. The parameters of a generative model describing the face manifold are learned from the training data. This model enables the computation of the posterior probability over possible clusterings, so that Bayesian model selection can be applied to compare partitions of varying sizes. Du and Chellappa [8] presented an on-line context-aided face association method, which uses a Conditional Random Fields (CRFs) to combine multiple contextual features. Wolf et al. [23] described a set-to-set similarity measure, the Matched Background Similarity (MBGS), which can tell the differences between images with similar background, so that the similarities due to pose, lighting, and viewing conditions can be ignored.

The aforementioned methods are all purely data-driven, which only exploit information contained in the data. Their performance significantly depends on the data quality. Considering the huge uncertainty in real-world videos, purely data-driven methods are expected to be unstable. Instead, prior knowledge could be exploited to guide the clustering in order to achieve robustness and increase the generalization ability of the methods. Berg et al. [4] considered using extra information to enhance the face clustering, where the faces are collected from web news pages. A set of names automatically captured from associated news captions are employed to supervise the clustering. However, such text-based labels are not always available for faces in videos. Fortunately, there is readily useful prior knowledge for face clustering in videos: The faces in the same face track must be the same person, no matter how different the appearances of the faces look like; If two face tracks overlaps in some frames, then faces in these two tracks must be different persons, no matter how same they look like. Thus, we can easily obtain plenty of must-link and cannot-link constraints without much extra cost. However, few works have exploited such constraints. In [21], constraints are exploited to modify the distance matrix and to guide the clustering to satisfy such constraints. However, the method is very computationally expensive. As reported in [21], it takes about 6 days on a data set of 10000 faces. The latest work on face clustering with constraints is presented in [7], called unsupervised logistic discriminative metric learning (ULDML). A metric is learned such that must-linked faces are close, while cannot-linked faces are far from each other.

In the literature of constrained clustering, many methods have been proposed to exploit pairwise constraints to guide the clustering, such as COP-KMEANS [22], constrained EM [20], HMRF-KMeans [2] and PPC [17]. COP-KMEANS embeds constraints in hard manner, while the other three adopt the soft constraints. However, the weights of these soft constraints are totally user-defined. In contrast, the weights of constraints can be easily adjusted through learning in our algorithm.

Method overview. In this paper we propose a probabilistic constrained clustering model based on Hidden Markov Random Fields. Together with the pairwise constraints, the local smoothness assumptions are also incorporated into the model to achieve the robust performance. The local smoothness is implemented in two levels: the label-level smoothness means that *if two observations x_i and x_j are similar, then their labels y_i and y_j should be similar*; the constraint-level smoothness tells that *given a must-link (cannot-link) between x_1 and x_2, if x_3 is close to x_2, then it is assumed that there is also a must-link (cannot-link) between x_1 and x_3* [14][16]. The proposed model is optimized by the *simulated field algorithm* [6]. Such that the parameters are learned effectively, and the weights of pairwise constraints can be easily adjusted. Note that there are many issues in the whole process of face clustering in videos, such as face detection, face tracking, face features and determining the number of persons, etc. However, in this paper we only focus on showing how and to what extent the pairwise constraints can help the face clustering in videos. Our task can be briefly described as follows: *given the number of persons K, the detected faces and their tracks, and features of each face, we partition all faces into K disjoint clusters.*

The main contributions of this work are highlighted in three aspects. (1) We propose a probabilistic constrained clustering model based on HMRFs, in which the pairwise constraints, label-level and constraint-level local smoothness assumptions are incorporated together to guide the clustering process. (2) The model parameters are effectively learned through the simulated field algorithm, and the weights of constraints can be easily adjusted. In contrast, in many existing constrained clusterings [2][17], no practical suggestions are provided to determine the weights of constraints. (3) We present an efficient clustering framework specially for face clustering in videos, considering that faces in adjacent frames of the same face track are very similar. Any clustering algorithm can be directly used in this framework to significantly reduce the computational cost.

The remaining of this paper is structured as follows: In Section 2, a constrained clustering model based on HMRFs is introduced; The optimization of the proposed model is given in Section 3; Section 4 shows the face clustering experiments on two face data sets in videos, followed by the conclusions in Section 5.

2. A constrained clustering based on Hidden Markov Random Fields

2.1. Problem formulation

Given a unlabeled data set $X = \{x_1, x_2, \ldots, x_n | x_i \in \mathbb{R}^d\}$, our goal is to partition it into K (predefined) disjoint clusters. The latent label set is denoted as $Y = \{y_1, y_2, \ldots, y_n\}, y_i \in \{1, 2, \ldots, K\}$. A pairwise constraint set $C = \{C_{ml}, C_{cl}\}$ is also provided to guide the clustering, such that the clustering result should satisfy these constraints. The must-link constraints $C_{ml} = \{(x_i, x_j)\}$ indicates that x_i and x_j should be in the same cluster. The cannot-link constraints $C_{cl} = \{(x_i, x_j)\}$ requires that x_i and x_j should be partitioned into different clusters. A $n \times n$ symmetric matrix $W = \{w_{ij}\}$ is adopted to represent all the pairwise constraints: if $(x_i, x_j) \in C_{ml}$, then $w_{ij} = 1$; if $(x_i, x_j) \in C_{cl}$, then $w_{ij} = -1$; if $(x_i, x_j) \notin C$, $w_{ij} = 0$. As we will show later, through constraint propagation, $-1 \leq w_{ij} \leq 1$: $w_{ij} > 0$ means must-link, $w_{ij} < 0$ means cannot-link; $|w_{ij}|$ denotes the confidence of the constraint, i.e., the cost of violating this constraint.

2.2. Hidden Markov Random Fields

Hidden Markov Random Fields [13] is a commonly used generative model. It is defined based on two assumptions: (a) given the latent variables Y, the observed variables X are independent, i.e. $P(X|Y) = \prod_{i=1}^{n} p(x_i|y_i)$; (b) given the observed variables X, Y constitute a Markov network. The correlations between Y are embedded by a neighbourhood system. Its general formulation is as follows [13]:

$$\begin{aligned} P(X,Y) &= P(X|Y)P(Y) \\ &= \frac{1}{Z}\prod_{i=1}^{n}\psi_u(x_i,y_i)\prod_{i=1}^{n}\prod_{j\in N_i}\psi_p(y_i,y_j), \end{aligned} \quad (1)$$

where ψ_u denotes the unary potential function, ψ_p denotes the pairwise potential function, and Z is the partition function, N_i represents the neighbourhood set of x_i.

2.2.1 Unary potential function ψ_u

$\psi_u(x_i, y_i)$ embeds the correlation between the observation x_i and its latent label y_i. There have been many different methods to design the unary potential. For simplicity, we assume it as a Gaussian distribution, as follows:

$$\psi_u(x_i, y_i) = \mathcal{N}(x_i|\boldsymbol{\mu}_{y_i}\boldsymbol{\Sigma}_{y_i}), \quad (2)$$

where $\boldsymbol{\mu}_{y_i}$ is the mean vector of the y_ith cluster, $\boldsymbol{\Sigma}_{y_i}$ denotes the covariance matrix. $\Theta = \{(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)\}$ denotes the parameter set of ψ_u. Θ should be learned in optimization.

2.2.2 Pairwise potential function ψ_p

$\psi_p(y_i, y_j)$ embeds the correlation between y_i and y_j. All correlations in Y can be represented by a neighbourhood system, denoted as a $n \times n$ symmetric matrix $V = \{v_{ij}\}$. $v_{ij} > 0$ means the positive correlation, i.e. y_i and y_j should be same; $v_{ij} < 0$ means the negative correlation, i.e. y_i and y_j should be different; $v_{ij} = 0$ means no correlation. Since V plays a key role in the pairwise potential, we present a detailed discussion about how to generate V in the next section. Here we firstly show how to utilize a generated V in pairwise potential, as follows:

$$\begin{aligned} \psi_p(y_i, y_j) &= \exp(-\beta\phi(y_i, y_j)) = \exp\big(\beta[\textstyle\sum_{v_{ij}\geq 0} v_{ij} \\ &(-1 + \delta(y_i, y_j)) + \textstyle\sum_{v_{ij}<0} v_{ij}\delta(y_i, y_j)]\big), \end{aligned} \quad (3)$$

where $\phi(y_i, y_j)$ is called pairwise energy function, β is a positive trade-off parameter between unary and pairwise potential, and it will be learned. δ is defined as follows: if $y_i = y_j$, then $\delta(y_i, y_j) = 1$, else $\delta(y_i, y_j) = 0$. Equation (3) has an intuitive interpretation in probability: when $v_{ij} > 0$, if $y_i \neq y_j$, then $\psi_p(y_i, y_j) = \exp(-\beta|v_{ij}|) < 1$, i.e., the prior probability of this configuration will decrease; when $v_{ij} < 0$, if $y_i = y_j$, then $\psi_p(y_i, y_j) = \exp(-\beta|v_{ij}|) < 1$, i.e., the prior probability of this configuration will decrease; when $v_{ij} = 0$, $\psi_p(y_i, y_j) = 1$, i.e., the prior probability of any configuration of y_i and y_j is equal. In summary, when a correlation exists between y_i and y_j, configurations that violate the correlation should be penalized in probability. The penalty degree depends on the confidence of the correlation , i.e., $|v_{ij}|$.

2.3. The neighbourhood system V

The neighborhood system V takes the key role in the HMRFs model, since it embeds the correlations between the latent variables Y. In this paper we present a combined neighborhood system, including the normalized affinity matrix, the propagated pairwise constraints and a trade-off parameter.

Normalized affinity matrix. Its formulation is as follows [15]:

$$V^{ls} = L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, \quad A_{ij} = \exp^{(-d^2(x_i,x_j)/\sigma_i\sigma_j)}, \quad (4)$$

where A is the affinity matrix of a k-nn graph, i.e., if x_j is not within the k-nearest neighbors of x_i, then $A_{ij} = 0$. Note that $A_{ii} = 0$. $k = 10$ in our experiments. L is its normalized affinity matrix. The degree matrix D is diagonal, and $D_{ii} = \sum_{j=1}^{n} A_{ij}$. $d(x_i, x_j)$ denotes the distance between x_i and x_j (here the Euclidean distance is used). $\sigma_i = d(x_i, x_h)$, where x_h is the hth nearest neighbor of x_i. Following the suggestion in [15], we set $h = 7$. V^{ls} embeds the label-level local smoothness assumption.

Propagated pairwise constraints. In many existing works [20][2][17], the original pairwise constraint matrix W are used to represent the correlations between Y. However, W is often sparse, especially when given a few constraints. It fails to provide enough information to guide the clustering process. To alleviate this limitation, pairwise constraints can be propagated to gain many soft constraints, based on the constraint-level local smoothness assumption. [16] presents a closed-form propagation method, as follows:

$$V^{pc} = (1-\alpha)^2(I - \alpha L)^{-1}W(I - \alpha L)^{-1}, \quad (5)$$

where user-defined constant $\alpha = \frac{\gamma}{1+\gamma} \in (0,1)$, which can be seen as the propagation degree: when $\alpha \to 0$, $V^{pc} = W$, i.e., the propagation degree is small; when $\alpha \to 1$, V^{pc} is far from W, i.e., the propagation degree is large. It has been proven that $|V_{ij}^{pc}| \leq 1$ [16]. It is considered that there is a soft constraint between x_i and x_j: $\overline{W}_{ij}^{*} > 0$ means a soft must-link; $V_{ij}^{pc} < 0$ means a soft cannot-link. From the interpretation of Equation (3), the propagated constraints totally satisfy the requirement of the neighborhood system.

Combined neighborhood system. Constraint propagation leads to many more constraints than given constraints. However, it is found that the magnitudes of most propagated constraints are too small to represent correlations between samples, especially for the local correlations between nearby samples. V^{ls} just compensates for this limitation. A natural choice is to combine V^{pc} and V^{ls}, as follows:

$$\psi_p(y_i, y_j) = \exp\big\{\beta * \big(\lambda \sum_{V_{ij}^{ls} \geq 0} V_{ij}^{ls}(-1 + \delta(y_i, y_j)) + [\sum_{v_{ij}^{pc} \geq 0} v_{ij}^{pc}(-1 + \delta(y_i, y_j)) + \sum_{v_{ij}^{pc} < 0} v_{ij}^{pc}\delta(y_i, y_j)]\big)\big\}. \quad (6)$$

Equation (6) has an intuitive interpretation: it penalizes the configuration that violates either propagated constraints (including given constraints and constraint-level local smoothness) or label-level local smoothness. As a result, it enforces the clustering to satisfy both of these two terms. This neighborhood system is denoted as $V^{com} = (V^{ls}, V^{pc}, \lambda)$. The user-defined constant λ represents the trade-off between the constraints and label-level local smoothness.

3. Objective function and its optimization

3.1. Objective function

In Section 2, we have presented a generative model based on HMRFs, with four different neighborhood systems. For the clustering task, a natural choice is the maximization of the log complete likelihood function. The objective function is as follows:

$$\mathcal{L}(Y, \Theta, \beta) = \log P(Y, X|\Theta, \beta)$$
$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\delta(y_i, k)\big(log|\mathbf{\Sigma}_k| + (x_i - \boldsymbol{\mu}_k)^T\mathbf{\Sigma}_k^{-1}$$
$$(x_i - \boldsymbol{\mu}_k)\big) - \beta\sum_{i=1}^{n}\sum_{j\in N_i}\phi(y_i, y_j) - \log Z, \quad (7)$$

where the pairwise energy function $\phi(y_i, y_j)$ is defined in Equations (3) and (6). Since we choose V^{com} as the neighborhood system, the corresponding clustering model is denoted as HMRF-com.

3.2. Optimization: simulated field algorithm

Given the neighborhood system, Y is not independent with each other. As a result, the exact computation of Z, the learning of β and the inference of Y become intractable. We have to use some approximate methods to gain a suboptimal solution. In this paper, we adopt an approximate method called *simulated field algorithm* [6]. Its main idea is: when treating a particular latent variable y_i, ignoring the fluctuations of its neighbors, through fixing the states of the neighbors; as a result, the overall computation reduces to deal with independent variables. The main procedure is shown in Algorithm 1. Its computational complexity is $O(TKn(r + d^2) + n^2)$, where T denotes the number of iterations in optimization, and r is the maximal number of neighbors for any sample. Since V^{com} is used as the neighborhood system, $r = n - 1$. Actually r can be significantly decreased through making the neighborhood system sparse. However, we do not make this processing in this paper.

- **Given** $\Theta^{(t-1)}$, $\beta^{(t-1)}$, $P(Y|X, \Theta^{(t-1)}, \beta^{(t-1)})$, **update** $\overline{Y}^{(t)}$

Firstly, for all latent variables, $\overline{Y}^{(t)}$ are simulated from $P(y_i|x_i, \overline{y}_{N_i}^{(t,i)}, \Theta^{(t-1)}, \beta^{(t-1)})$, in a sequential form. $\overline{y}^{(t,i)} = \{\overline{y}_1^{(t)}, \ldots, \overline{y}_{i-1}^{(t)}, \overline{y}_{i+1}^{(t-1)}, \overline{y}_n^{(t-1)}\}$. Note that $\overline{Y}^{(t)}$ are temporal configurations of the labels, in order to ease the following computations. They should be distinguished from the predicted labels. Then, for each latent variable, set its neighbors to $\overline{Y}^{(t)}$ and replace the marginal distribution $P(Y|\beta^{(t-1)})$ by the following formulation:

$$P(Y|\overline{Y}^{(t)}, \beta^{(t-1)}) = \prod_{i=1}^{n} P(y_i|\overline{y}_{N_i}^{(t)}, \beta^{(t-1)})$$
$$= \prod_{i=1}^{n}\frac{P(y_i, \overline{y}_{N_i}^{(t)}, \beta^{(t-1)})}{\sum_{y_i=1}^{K} P(y_i, \overline{y}_{N_i}^{(t)}, \beta^{(t-1)})} \quad (8)$$

Algorithm 1 Simulated field algorithm for HMRF-based clustering

Input: unlabeled data set X, pairwise constraints C, α, λ
Output: predicted labels of Y and parameters Θ and β

Step 1 Use K-means to initialize the labels Y and the parameters $\Theta^{(0)}$; Setting $\beta^{(0)} = 0$;
Step 2 Compute the neighborhood system V as described in Section 2.3;
Step 3 Set $t = 1$, repeat the following three steps, until convergence:
(3.1) given parameters $\Theta^{(t-1)}$, $\beta^{(t-1)}$ and the posterior probability $P(Y|X, \Theta^{(t-1)}, \beta^{(t-1)})$, update configurations $\overline{Y}^{(t)}$;
(3.2) given configurations $\overline{Y}^{(t)}$, update the posterior probability $P(Y|X, \Theta^{(t-1)}, \beta^{(t-1)})$ and parameters $\Theta^{(t)}$, $\beta^{(t)}$.
(3.3) set $t = t + 1$;
Step 4 Set $Y^* = \arg\max_Y P(Y|X, \Theta^*, \beta^*)$, then output Y^*, Θ^* and β^*.

where $P(y_i, \overline{y}_{N_i}^{(t)}, \beta^{(t-1)}) = \frac{1}{Z}\exp\big(-\beta^{(t-1)}\phi(y_i, \overline{y}_{N_i}^{(t)})\big)$. Note that the partition function Z is eliminated in $P(y_i|\overline{y}_{N_i}^{(t)}, \beta^{(t-1)})$. The problem becomes tractable.

- **Given** $\overline{Y}^{(t)}$ **, update** $P(Y|X, \Theta^{(t-1)}, \beta^{(t-1)})$**,** $\Theta^{(t)}$**,** $\beta^{(t)}$

In this step, parameters learning and inference are done by the Expectation Maximization (EM) algorithm. Following the work in [6], we run a single iteration in this step.

E step: compute the posterior probability and the expectation of the joint log likelihood function

$$P(y_i|x_i, \overline{y}_{N_i}^{(t)}, \Theta^{(t-1)}, \beta^{(t-1)}) = \frac{\psi_u(y_i, x_i, \Theta^{(t-1)})P(y_i|\overline{y}_{N_i}^{(t)}, \beta^{(t-1)})}{\sum_{y_i=1}^{K}\psi_u(y_i, x_i, \Theta^{(t-1)})P(y_i|\overline{y}_{N_i}^{(t)}, \beta^{(t-1)})}, \quad (9)$$

$$\mathcal{Q}(\Theta^t, \beta^t|\Theta^{t-1}, \beta^{t-1}) = E_{Y|X,\Theta^{t-1},\beta^{t-1},\overline{y}^{(t)}}\big(\log P(Y, X|\Theta, \beta, \overline{Y}^{(t)})\big). \quad (10)$$

M step: update the parameters

$$\Theta^{(t)} = \arg\max_{\Theta}\sum_{i=1}^{n}\sum_{y_i=1}^{K}P(y_i|x_i, \overline{y}_{N_i}^{(t)}, \Theta^{(t-1)}, \beta^{(t-1)})\log\psi_u(y_i, x_i, \Theta), \quad (11)$$

$$\beta^{(t)} = \arg\max_{\beta}\sum_{i=1}^{n}\sum_{y_i=1}^{K}P(y_i|x_i, \overline{y}_{N_i}^{(t)}, \Theta^{(t-1)}, \beta^{(t-1)})\log P(y_i|\overline{y}_{N_i}^{(t)}, \beta). \quad (12)$$

Since $\psi_u(y_i, x_i, \Theta)$ is Gaussian distribution, the closed-form solution for Θ is be gained easily. For β, we find a local optimal value in each iteration through the local search method [3]. Note that $\mathcal{Q}(\Theta^t, \beta^t|\Theta^{t-1}, \beta^{t-1})$ is an approximation of the expectation of the objective function $\mathcal{L}(Y, \Theta, \beta)$.

4. Face clustering in videos

4.1. Comparison of clustering algorithms

Three state-of-the-art clustering algorithms of different types are compared in our experiments:

Traditional clustering: K-means [5] is used as baseline to check whether the prior knowledge can help the clustering.

Constrained clustering: penalized probabilistic clustering (PPC) [17] embeds pairwise constraints in Gaussian mixture models. Its main limitation is the requirement of a large number of constraints. Besides, the learning of the mixture coefficient parameter π is solved by grid search in [17], which is infeasible when the dimension of data is larger than 5. For simplicity, in our experiments we set $\pi = \frac{1}{K}$. Magnitudes of all constraints in PPC are set to 1.

Specific algorithm for face clustering in videos: as mentioned in Section 1, the face clustering presented in [21] is not compared, due to its high computational cost. We compare with ULDML [7], in which positive pairs are generated based on must-link constraints, while negative pairs based on cannot-link constraints. Then a Mahalanobis metric is learned through the logistic regression, such that the distance between two faces in a positive pair is small, while the distance between two faces in a negative pair is large. In order to decrease the computational cost, in our experiments PCA [5] is firstly adopted to project the original face feature into a 100-dimensional space[1], then ULDML is run on the projected data set. Following the work in [7], we also use the low-rank constraint on the covariance matrix, i.e., $M = L^T L$, where L is a 35×100 matrix. In [7], the authors define a distance matrix between face tracks, based on the learned metric between faces. Then a complete-link hierarchical clustering method is run on this distance matrix to predict the label of each face track. This algorithm is denoted as ULDML-cl. Note that the other three algorithms are all doing clustering on faces, rather than on face tracks. So we also compare with K-means utilizing the learned metric, denoted as ULDML-km. Another important issue in ULDML is the initialization of L. In experiments it is found that the performance of the random initialization with L2 normalization is often worse than the performance of the initialization using PCA. In our experiments the results using the PCA initialization are reported.

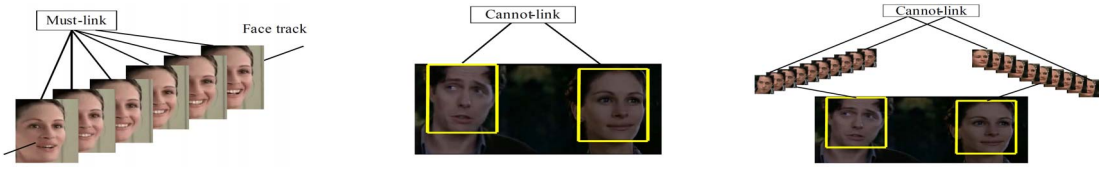
4.2. An efficient clustering framework

The computational complexity of Algorithm 1 is $O(TKn(r + d^2) + n^2)$. The number of detected faces from videos is often up to several thousands or more. Such a size takes a long time using HMRF-com or PPC. However, it is found that the faces in adjacent frames of the same track are very similar. Taking advantage of this characteristic, we develop an efficient clustering framework. The main idea is firstly doing clustering on a subset of faces, then determining the labels of all faces based on the labels of the subset. Details are presented in Algorithm 2. If most face tracks are pure, then the clustering accuracy of this framework would not become worse than the accuracy of clustering on the whole data set directly. The following experiments will demonstrate this point. In our experiments, PPC and HMRF-com (Algorithm 1) are adopted in Stage 2 of Algorithm 2 respectively. In order to check the effect on clustering accuracy of Algorithm 2, we run K-means in two ways: (a) run K-means directly on the whole data set (projected to the same dimension with the subset using PCA), denoted as Kmeans-1; (b) adopt K-means in Stage 2 of Algorithm 2, denoted as Kmeans-2. Since ULDML is originally designed to do clustering on face tracks, it is run on the whole data set. The computational complexity of using HMRF-com in this framework is $O(TKn_s(r + d^2) + n_s^2 + n)$, where n_s is the size of the subset, and here $r = n_s - 1$. Since $\frac{n_s}{n} < 0.1$

[1] In our experiments, since the summation of 100 largest eigenvalues takes about 90% of the summation of all eigenvalues, this projection will not lose much information of the original data set.

data set	person	face	dimension	track	overlapped track	must-link	cannot-link
BF0502-whole [9]	6	17737	1937	229	20	954005	116762
BF0502-subset	6	687	10	229	20	687	180
Notting-Hill-whole [24]	5	4660	18000	76	6	210293	11759
Notting-Hill-subset	5	456	5	76	6	1140	216

Table 1. Two face data sets from different real-world videos.



(a) (b) (c)

Figure 1. Detected faces from Notting-Hill [24]: (a) faces in the same track are must-linked; (b) two faces in the same frame are cannot-linked; (c) faces from two overlapped tracks are cannot-linked.

Algorithm 2 An efficient clustering framework for face clustering in videos

Input: the whole face data set X, the whole set of pairwise constraints C
Output: the predicted labels of each faces Y

Stage 1 Construct a subset X^s by uniformly sampling a fixed number (user defined) of faces from each face track, and determine the corresponding subset of pairwise constraints C^s;
Stage 2 Adopt a clustering algorithm on X^s with C^s, and predict their labels Y^s;
Stage 3 Determine the labels of all faces Y based on Y^s. Specifically, for all faces in one face track, their labels are determined as the mode value among the labels in Y^s of this track. For example, 5 faces are sampled from a track, and their labels are predicted as (3,1,2,3,3) in Stage 2. Then the labels of all faces in this track are determined as 3.

often holds in practice, the complexity is significantly reduced compared with Algorithm 1.

4.3. Two face data sets in videos

We evaluate the clustering algorithms on two publicly available face data sets: BF0502 [9] and Notting-Hill [24]. BF0502 is derived from the TV series " Buffy the Vampire Slayer". The detected faces of 6 main casts are used, including 17337 faces in 229 tracks. 3 faces are uniformly sampled from each track, then a subset of 687 faces is gained. Each face is represented as a 1937 dimensional vector. Notting-Hill is detected from the movie " Notting Hill". Faces of 5 main casts are used, including 4660 faces in 76 tracks. The original data set does not provide the feature of each face. For simplicity, we represent each face by the pixel intensities of RGB channels, a 18000 dimensional vector. 6 samples are uniformly sampled from each track, then a subset of 456 faces is gained. In order to avoid overfitting and reduce the computational cost, on both subsets we use PCA to project the original feature space to a lower dimensional space. The number of projected dimension is user-defined, and we suggest it is equal to or slightly larger than the number of clusters. The pairwise constraints are derived from three sources: (a) the faces in the same face track should be must-linked; (b) two faces in the same frame should be cannot-linked; (c) if two face tracks are overlapped, i.e., two faces from them appear in the same frame, then faces in one track should be cannot-linked with faces in the other track. The transitivity of must-links are utilized here. A simple example is shown in Figure 1. Details of both data sets are summarized in Table 1.

4.4. Clustering results

Clustering evaluation. For simplicity, we compute the accuracy based on the confusion matrix, which is derived from the match between the predicted labels of all faces and the ground-truth labels.

Parameter setting. For two user-defined constants, α is chosen from $\{0.05, 0.1, \ldots, 0.95\}$, and $\lambda \in \{10^0, 10^{-1}, \ldots, 10^{-6}\}$. As mentioned in Section 3.2, Equation (10) is an approximation of the expectation of the objection function. Its final value after optimization can be utilized to determine α and λ. More specifically, α and λ that lead to larger value of Equation (10) are preferred. Another issue is setting the search space of β. To shrink this search space, we amplify the absolute value of each constraint, i.e., $v_{ij}^{(ls)} \times 100$ and $v_{ij}^{(pc)} \times 100$. Then the search space of β is set to $\{0.1 : 0.1 : 1.5\}$. Note that the cost of constraint violation depends on α, λ, β together. Since α, λ are easily determined and β is automatically learned, the weights of constraints can be easily adjusted.

We test all algorithms in three cases: with cannot-links, with must-links and with all-links. Such a setting provides a clear view of different effects of different constraints. Each algorithm is repeated 30 times in each case. The averaged

accuracy and standard deviation are computed as the output.

Clustering accuracy. The results on BF0502 are shown in Table 2. HMRF-com outperforms all others in both must-links and all-links: about 4-18.7% increases in must-links, and about 1-11% increases in all-links. In cannot-links, HMRF-com is slightly lower than PPC. The results on Notting-Hill are shown in Table 3. The accuracies of HMRF-com are higher than all others (except ULDML-cl) in all cases: about 3.5-7.8% increases in cannot-links, about 2-12% increases in must-links and about 5.5-15% increases in all-links. ULDML-cl gives degenerate results in all cases. Kmeans-2 are higher than the accuracies of Kmeans-1 on both data sets. This demonstrates that Algorithm 2 can maintain or even gives higher accuracy compared with running clustering on the whole data set directly. Note that accuracies on the subset are not shown in Table 2 and 3. Denote the accuracy on subset as HMRF-com-sub, and on the whole data as HMRF-com-whole. On BF-0502, HMRF-com-sub is about 4% lower than HMRF-com-whole, while about 7% on Notting-Hill. The gap mainly depends on sampling and data. Similar gaps occur in other methods.

A clustering example of HMRF-com on Notting-Hill is shown in Figure 2. The purities of 5 clusters (top-down) are $90.66\%, 70.72\%, 100\%, 93.95\%$ and 43.47% respectively. The first and third cluster take over 50% of the whole data set, while the last cluster takes less than 10%, so that the clustering accuracy is up to 85.26%. As shown in Figure 2, the faces have different resolutions, different head poses, different facial expressions and partial occlusions in Notting-Hill. Obviously 85.26% is an encouraging clustering result on such a data set. Since face images are not provided in [9], the clustering result on BF0502 are not shown.

Figure 2. A clustering result of HMRF-com on Notting-Hill-whole, including 5 clusters shown in 5 rows respectively. 11 faces are randomly chosen from each cluster. The incorrect faces are highlighted by the red rectangle, and their numbers are approximately equal to their proportions in each cluster.

Robustness. Since all above clustering methods give local optimal results, the robustness is an important measure of their performances. The robustness can be reflected from the standard deviation, i.e., a small standard deviation means good robustness. As shown in Table 2 and 3, PPC has smaller standard deviations than Kmeans-s in most cases. It tells that pairwise constraints can alleviate the uncertainty in faces. HMRF-com has much lower standard deviations than PPC and Kmeans-s in all cases. It benefits from the combination of pairwise constraints and local smoothness. Although given a fixed initialization by PCA, ULDML-km on Notting-Hill still has a large standard deviation. Its standard deviations on BF0502 are small, but its accuracies are low. Since complete-link hierarchical clustering gives a fixed result, standard deviations of ULDML-cl on both data sets are 0. However, its performance varies dramatically on different data sets: On BF0502, it gives a much better result than K-means, while a degenerate result on Notting-Hill. We consider that the instability of ULDML is mainly due to the isolation between metric learning and clustering. Metric learning pursues the constraint satisfaction, while clustering wants to satisfy the cluster assumption. Their goals are different and there are no direct relations in theory. Although a well learned metric is expected to help clustering, there is no guarantee. In contrast, we perform constraint satisfaction and clustering simultaneously, in a unified model.

Computational complexity. The computational complexity of HMRF-com in Algorithm 2 is $O(TKn_s(r+d^2)+n_s^2+n)$. The complexity of metric learning in ULDML is $O(Tn(n+d)\times 35)$. Specifically, on BF0502, the average time of HMRF-com is about 119.26s, while ULDML takes 410.56s. On Notting-Hill, HMRF-com takes 19.58s, while ULDML takes 84.24s. The time of PPC is about half of HMRF-com. Computations of distance matrix between samples and dimensionality reduction by PCA are not included, because they are shared in above methods.

All above comparisons demonstrate the superior performance of our algorithm to state-of-the-art algorithms. Note that we have studied the influences of different parts in HMRF-com, including affinity matrix, constraint propagation, combination, constraint weights and sampling. We find constraint propagation and constraint weights make the key contributions to the performance. Other parts bring additional benefits. Due to the page limitation, the details are not presented in this paper.

Methods	Cannot-link	Must-link	All-link
Kmeans-1	39.31±4.51	39.31±4.51	39.31±4.51
Kmeans-2	42.05±5.45	42.05±5.45	42.05±5.45
PPC	**46.07±5.52**	43.64±4.61	42.54±3.98
ULDML-km	44.08±2.8	29.05±2.84	41.62±0
ULDML-cl	42.72±0	39.01±0	49.29±0
HMRF-com	45.96±1.46	**47.77±3.31**	**50.30±2.73**

Table 2. Clustering accuracies on BF0502-whole.

Methods	Cannot-link	Must-link	All-link
Kmeans-1	69.16±3.22	69.16±3.22	69.16±3.22
Kmeans-2	73.43±8.12	73.43±8.12	73.43±8.12
PPC	70.26±8.98	79.71±2.14	78.88±5.15
ULDML-km	69.64±9.43	72.66±12.78	73.18±8.66
ULDML-cl	38.76±0	51.72±0	36.87±0
HMRF-com	**76.94±3.58**	**81.33±0.43**	**84.39±1.47**

Table 3. Clustering accuracies on Notting-Hill-whole.

5. Conclusions

This paper has showed how to utilize the readily available pairwise constraints to help face clustering in videos. Together with pairwise constraints, label-level and constraint-level local smoothness assumptions are also incorporated into the neighborhood system of the Hidden Markov Random Fields. It provides enough constraints on the labels to guide the clustering, such that it shows a promising and robust performance. Model parameters are learned through simulated field algorithm, and the weights of constraints can be easily adjusted. An efficient clustering framework is also specifically developed for face clustering in videos, considering that faces in adjacent frames of the same face track are very similar. It not only reduces the computational cost significantly, but also maintains the clustering accuracy. This framework demonstrates the feasibility of our algorithm to scale to larger face data sets from videos, which will be one of our future work. Experiments on two face data sets from real-world videos have demonstrated the outstanding performance of our algorithm.

6. Acknowledgements

The work was completed when the first author was a visiting student at Rensselaer Polytechnic Institute (RPI), supported by a scholarship from China Scholarship Council (CSC). We thank CSC and RPI for their supports. Qiang Ji's involvement in this work is supported in part by a grant from the US National Science Foundation (NSF, No. 1145152). This work is also supported in part by the National Natural Science Foundation of China (NSFC, No. 61075051 for Bao-Gang Hu and Baoyuan Wu, No. 61202325 for Yifan Zhang). We thank the reviewers for their constructive comments. We also thank Ramazan Gokberk Cinbis for his helpful discussion during this work.

References

[1] O. Arandjelovic and R. Cipolla. Automatic cast listing in feature-length films with anisotropic manifold space. In *CVPR*, 2006.

[2] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney. Probabilistic semi-supervised clustering with constraints. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*, chapter 5, pages 71–98. MIT Press, 2006.

[3] R. Battiti, M. Brunato, and F. Mascia. *Reactive Search and Intelligent Optimization*, volume 45 of *Operations research/Computer Science Interfaces*. Springer Verlag, 2008.

[4] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *CVPR*, 2006.

[5] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[6] G. Celeux, F. Forbes, and N. Peyrard. Em procedures using mean field-like approximations for markov model-based image segmentation. *Pattern Recognition*, 36(1):131–144, 2003.

[7] R. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in tv video. In *ICCV*, 2011.

[8] M. Du and R. Chellappa. Face association across unconstrained video frames using conditional random fields. In *ECCV*, 2012.

[9] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy automatic naming of characters in tv video. In *British Machine Vision Conference*, 2006.

[10] A. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *ECCV*, 2002.

[11] A. Fitzgibbon and A. Zisserman. Joint manifold distance: a new approach to appearance based clustering. In *CVPR*, 2003.

[12] Y. Hu, A. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

[13] D. Koller and N. Friedman, editors. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[14] Z. Li, J. Liu, and X. Tang. Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In *Proceedings of the 25th International Conference on Machine Learning*, pages 576–583, 2008.

[15] Z. Lihi and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2004.

[16] Z. Lu and H. H. S. Ip. Constrained spectral clustering via exhaustive and efficient constraint propagation. In *Proceedings of the 11th European conference on Computer vision*, pages 1–14, 2010.

[17] Z. Lu and K. Todd. Penalized probabilistic clustering. *Neural Computation*, 19(6):1528–1567, 2007.

[18] S. Prince and J. Elder. Bayesian identity clustering. In *Canadian Conference on Computer and Robot Vision*, 2010.

[19] R.Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

[20] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *Advances in Neural Information Processing Systems*, 2004.

[21] N. Vretos, V. Solachidis, and I. Pitas. A mutual information based face clustering algorithm for movie content analysis. *Image and Vision Computing*, 2011.

[22] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrdl. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, pages 577–584, 2001.

[23] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534, 2011.

[24] Y. Zhang, C. Xu, H. Lu, and Y. Huang. Character identification in feature-length films using global face-name matching. *IEEE Transactions on Multimedia*, 11(7):1276–1288, 2009.