Who and Where: People and Location Co-Clustering

Zixuan Wang
Electrical Engineering
Stanford University
zxwang@stanford.edu

ABSTRACT

The goal of the image clustering is to group semantically related images together. This task is very important, particularly in the era with an immense volume of user uploaded online photos. In this paper, we consider the clustering problem on images where each image contains patches in people and location domains. While many previous work ignore the correlations between patches from different domains, we propose a co-clustering framework, which combines the visual appearance of patches in each domain and cross-domain relations. The objective of the clustering becomes minimizing the variance within the cluster in each domain, at the same time maximizing the consistency across both domains. We design a semi-supervised kernel k-means algorithm to approximate the solution. Experimental evaluations show that the convergence is reached fast and the clustering performance is promising. The last but not the least, we provide the MapReduce version to solve the scalability problem on a large corpus of images.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

Keywords

Co-clustering, Face Recognition, Location Recognition

1. INTRODUCTION

Given a large corpus of images, we want to cluster them such that images semantically related are grouped in one cluster. Semantics of an image refer to the information that image carries. For example, the face on the image is usually used to identify *who*. The background of the image refers to the location *where* the person was. All components together can convey what story has happened. In our case, we focus on two entities: *who* and *where*. Online social networks host huge volumes of images nowadays. Such explosive amount of digital photos on the web makes the clustering task crucial, in particular, when users apply image search over a person or a location. Combining with timestamps, reliable people and location clusters are also very helpful to tell stories of physical events through image corpus.

While there has been considerable work on automatic face recognition [2, 27] in images and even a modest effort on location recognition [4, 3], the coupling of the two is basically unexplored, though people and locations are highly inter-correlated. An image which contains both people and location implies the co-occurrence of instances in two domains. A collection of such images can shed light on the clustering on each domain. For example, multiple photos taken at the same private location increase the confidence that similar faces on those photos are from a same person. Similarly, within a short time window, the same person on several photos indicates the affinity of locations.

Our framework, shown in Fig. 1, consists of two domains: people and locations. We consider three types of relations between people and location domains: (1) *people-people* (2) *location-location* (3) *people-location*. A set of image patches is extracted and described in each domain. The similarity between patches within each domain is defined based on the visual appearances. We then take into account the co-occurrence relations across *people-location* domains to enhance the clustering in each domain. The co-occurrence constraints are satisfied if patches from two domains appear in a same image. This relationship reflects the consistency of clustering results which is not embodied from visual appearances in a single domain.

We formulate the clustering task as an optimization problem which aims to minimize the within cluster distances and maximize the consistency across domains. We show this problem can convert to the semi-supervised kernel k-means clustering similar to [14]. The difference is that we generate clustering results for two domains at the same time, and during the iterative clustering process, constraints across domains and within domain keep updated. The main idea is that the clustering result in one domain can aid the clustering in the other domain. We validate our approach with photos gathered from personal albums and a set of public photos crawled from Flickr.

When dealing with the set of online photos, the scalability of the framework becomes a challenge. Considering search people over billion photos, it's almost impossible to store all images in one machine and never mention the case when the

application requires real-time response. Therefore it is important to have a distributed algorithm. We develop a scalable solution of our framework and apply Map-Reduce on both the pre-process and alternative clustering without any loss of accuracy. We conduct the Map-Reduce framework on online datasets and observe a significant improvement with respect to the running time. Our main contributions are as follows.

1. The co-clustering algorithm is proposed for image clustering, focusing on people and locations. The framework couples both domains and explores underlying cross-domain relations. Our algorithm can simultaneously produce the clustering results of people and location, and outperform clustering separately on each domain and the baseline co-clustering algorithm.

2. Our algorithm is formulated as an optimization problem, which can be solved by through semi-supervised kernel k-means. It is robust and converges fast in practice.

3. Our algorithm can fit in the MapReduce framework to tackle the scalability challenge without any loss of accuracy. The experiment on the online dataset demonstrates our approach is scalable.

The paper is organized as follows: Section 2 outlines the recent work on the people and location clustering. Section 3 sketches the system framework , and describes the problem definition, formulation and the detail of algorithm. Section 4 shows the setup of our experiments and the results. Section 5 concludes the paper and discusses the future work.

2. RELATED WORK

Face is an important kind of visual objects in images, and it's crucial to identify people. It is well-structured and contains abundant information, therefore, it can provide accurate links between images. In recent years, there have been a lot of efforts in face detection [26], recognition [2, 27] and clustering [1]. The basic idea is to either represent a face as one or multiple feature vectors, or parameterize the face based on some template or deformable models. In addition to treating faces as individual objects, some researchers have been seeking for help from context information, such as background, people co-occurrence, etc. Davis et al. [6] developed a context-aware face recognition system that exploits GPS-tags, timestamps, and other meta-data. Song et al. [24] proposed an adaptive scheme to combine face and clothing features based on the timestamps. Lin et al. [16] proposed a unified framework to jointly recognize the people, location and event in a photo collection based on a probabilistic model.

In recent years, most location clustering algorithms are relying on the bag of words model [20]. First of all, interest points are detected [18] and described by an invariant descriptor [17]. Then the descriptors are quantized into a vocabulary of visual words by approximate k-means [20]. Metadata associated with images are also exploited to help the location clustering. Large-scale location clustering has been recently demonstrated in [15, 28], which use the GPS information to reduce the large-scale task down into a set of smaller tasks. Hays et al. [10] proposed an algorithm for estimating a distribution over geographic locations from the query image using a purely data-driven scene matching approach. They leveraged a dataset of over 6 million GPS-tagged images from the Flickr. When the temporal data is available in the corpora, it also helps to localize sequences of images. In [12], a prior of human travel patterns is added, which is learned from millions of images, to infer geographic location for sequences of time-stamped images. Chen and Grauman [3] leveraged the time information and travel patterns within a Hidden Markov Model (HMM) to robustly estimate locations for sequences of tourist photos.

K-means is one of the most popular clustering algorithms. A major drawback to k-means is that it can not separate clusters that are non-linearly separable in the input space. Two recent approaches have emerged for tackling such a problem. One is kernel k-means [7], in which points are mapped to a higher-dimensional feature space using a nonlinear function. The other approach is the spectral clustering algorithm [19], which uses the eigenvectors of an affinity matrix to obtain a clustering of the data. A popular objective function used in spectral clustering is to minimize the normalized cuts [22]. Kulis et al. [14] showed that the spectral clustering can be formulated as a weighted kernel k-means clustering problem.

3. APPROACH

In this section, we present the co-clustering framework to simultaneously cluster images in people and location domains. We have two major steps. The first step is pre-processing. We extract face and location patches from the corpus of images, and compute the visual features. The next step is co-clustering. The people-people, location-location and people-location relations are considered. Below has the detail for each step.

3.1 Pre-processing

We here describe how to extract features from people and location domains, and discover the relations between both domains.

People Domain. We use Viola-Jones face detector [26] to extract face patches from an image. To obtain high accuracy, a nested detector is applied to reduce the false positive rate. From the collection of images, every face has a corresponding face patch. All face patches are normalized to the same size. We adopt the algorithm in [25] to detect seven facial landmarks from each extracted face patch. For each input face patch, four landmarks (outer eye corners and mouth corners) are registered to the pre-defined positions using the perspective transform. Then all seven facial landmarks are aligned by the computed perspective transform. For each landmark, two SIFT descriptors of different scales are extracted to form the face descriptor. We build a face graph over all face patches in the image collection. In the graph, each vertex represents a face patch. The weight of the edge is the similarity of face descriptors of two face patches.

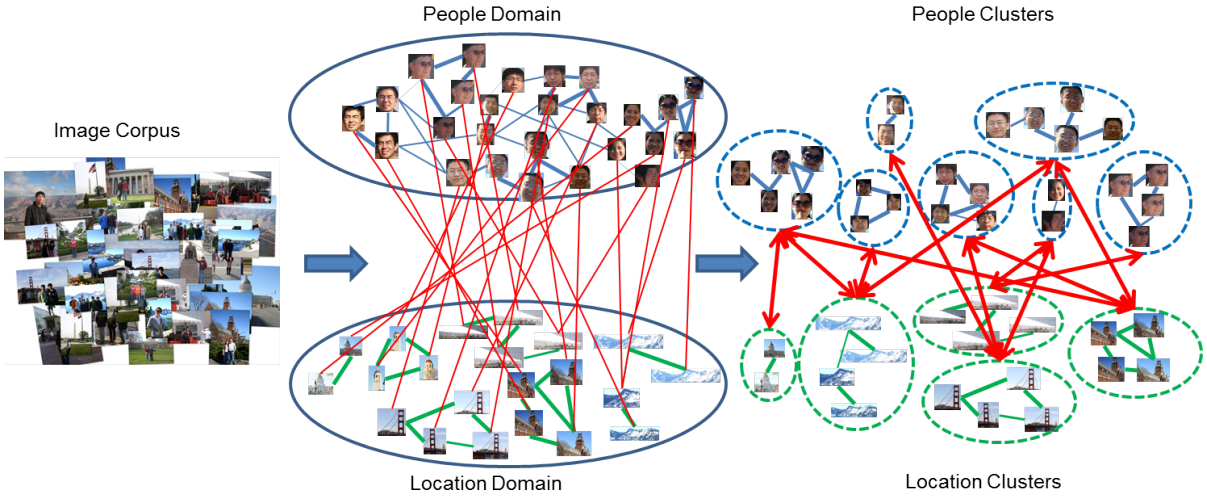


Figure 1: The framework of people and locations co-clustering. The red lines across two domains represent the co-occurrence relations.

Location Domain. For each image, Hessian affine covariant detector [18] is used to detect interest points. The SIFT descriptor [17] is extracted on every interest point. The method similar to the work of Heath et al. [11] is used to discover the shared locations in the image collection. The content-based image retrieval [20] is applied to find top related images, and avoids quadratic pairwise comparisons. Lowe's ratio test [17] is used to find the initial correspondences and RANSAC [8] is used to estimate the affine transform between a pair of images and compute feature correspondences between images. For every location patch, two types of features are extracted: a bag of visual words [23] and a color histogram. The bag of words descriptor summarizes the frequency that prototypical local SIFT patches occur. It captures the appearance of component objects. For images taken in an identical location, this descriptor will typically provide a good match.The color histogram characterizes certain scene regions well. These three types of features are concatenated to represent the location patch. A location graph is built similarly to the face graph. Each vertex in the graph represents a location patch. The weight of the edge is the similarity of location descriptors of two location patches.

Inter-relations across domains. To co-cluster across the people and location domains, several basic assumptions are made to reveal the relations between domains as follows.

Cannot Match Link. First, one person cannot appear twice in one image. Therefore, there is a *cannot match link* between a pair of face patches in the same image. Here we do not consider the exceptions like the photo collage or mirrors in the image. Second, if two locations are far away according to the ground truth e.g. GPS signals, and two face patches appear in these two locations during a short time period, there is a *cannot match link* between this pair of patches. This assumption comes from that people cannot teleport within a short time period, for example, one people cannot appear in San Francisco and in New York within an hour.

Must Match Link. First, two location patches are connected by a *must match link* if there is an affine transform found between them in the location graph construction. Because the links verified by RANSAC have high accuracy, we trust that they connect patches in the same location. Second, two location patches are connected by a *must match link* if they appear in the same image. Two different buildings may appear in the same image, therefore, in our setting, one location is defined as an area which may contain different backgrounds. Third, two location patches are connected by a *must match link* if they co-occur with the same people within a short time period. This assumption also comes from the fact that people cannot move too fast.

Possible Match Link. In the people domain, two face patches that appear in the same location but not in the same image probably belong to the same people. If there are places that a person takes more than one photos, the clustering results in the location domain can help cluster the people domain, through the strong co-occurrence relations. This is true if the place has special meaning to the person, for example, his/her home or office, where he/she visits frequently. However, the assumption is not always true. For example, at tourist attractions, every people would take photos there. Therefore, the locations do not contribute much for the clustering in people domain. A weight is needed for the locations to distinguish the public locations and private locations. Private location is more helpful for clustering in people domain. Public location will introduce many noise.

3.2 Problem Formulation

We formulate our people and location co-clustering as an optimization problem. Given a set of feature vectors $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, the goal of the standard k-means in each domain is to find a k-way disjoint partitioning $(S_1, \ldots, S_k)$ such that

the following objective is minimized:

$$f_{\text{kmeans}} = \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in S_c} ||\mathbf{x}_i - \mathbf{m}_c||^2 \quad (1)$$

where $\mathbf{m}_c$ is the cluster center of c. The objective can be rewritten using the fact that:

$$\sum_{c=1}^{k} \sum_{\mathbf{x}_i \in S_c} 2||\mathbf{x}_i - \mathbf{m}_c||^2 = \sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in S_c} \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{|S_c|} \quad (2)$$

The matrix E is defined as pairwise squared Euclidean distances among the data points, such that $E_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||^2$. We introduce an indicator vector $\mathbf{z}_c$ for the cluster S_c.

$$\mathbf{z}_c(i) = \begin{cases} 1 & \text{if } i \in S_c \\ 0 & \text{if } i \notin S_c \end{cases} \quad (3)$$

$\mathbf{z}_c^{\mathrm{T}}\mathbf{z}_c$ is the size of cluster S_c, and $\mathbf{z}_c^{\mathrm{T}} E \mathbf{z}_c$ gives the sum of E_{ij} over all $\mathbf{x}_i$ and $\mathbf{x}_j$ in S_c. Now the matrix $\tilde{Z}$ is defined such that the cth column of $\tilde{Z}$ is equal to $\mathbf{z}_c/(\mathbf{z}_c^{\mathrm{T}}\mathbf{z}_c)^{1/2}$. $\tilde{Z}$ is an orthonormal matrix, $\tilde{Z}^{\mathrm{T}}\tilde{Z} = I_k$, and the objective f_{kmeans} is:

$$\begin{aligned} \underset{\tilde{Z}}{\text{minimize}} \quad & \text{tr}(\tilde{Z}^{\mathrm{T}} E \tilde{Z}) \\ \text{subject to} \quad & \tilde{Z}^{\mathrm{T}}\tilde{Z} = I_k \end{aligned} \quad (4)$$

Let N_F be the number of face patches and N_L be the number of location patches. k_F is the number of face clusters and k_L is the number of location clusters. By considering the relations between people and location domains, we write the objective as:

$$\begin{aligned} \text{minimize} \quad & f_F + f_L - f_{FL} - f_{LF} \\ \text{subject to} \quad & f_F = \text{tr}(\tilde{Z}_F^{\mathrm{T}} E_F \tilde{Z}_F), \\ & f_L = \text{tr}(\tilde{Z}_L^{\mathrm{T}} E_L \tilde{Z}_L), \\ & f_{FL} = \sum_{i=1}^{t} \text{tr}(M_i^{\mathrm{T}} M_i), \\ & f_{LF} = \text{tr}(N^{\mathrm{T}} N), \\ & M_i = \tilde{Z}_F^{\mathrm{T}} C_{FL}^{\mathrm{T}} T_i \tilde{Z}_L, \\ & N = \tilde{Z}_F^{\mathrm{T}} C_{FL}^{\mathrm{T}} P \tilde{Z}_L, \\ & \tilde{Z}_F^{\mathrm{T}}\tilde{Z}_F = I_{k_F}, \\ & \tilde{Z}_L^{\mathrm{T}}\tilde{Z}_L = I_{k_L}. \end{aligned} \quad (5)$$

E_F and E_L are pairwise squared Euclidean distance matrices in people and location domains. To integrate the must match constraints and cannot match constraints, the distance of the must match link is set to 0 and the distance of the cannot match link is set to $+\infty$. f_F and f_L with constraints $\tilde{Z}_F^{\mathrm{T}}\tilde{Z}_F = I_{k_F}$ and $\tilde{Z}_L^{\mathrm{T}}\tilde{Z}_L = I_{k_L}$ are the standard k-means optimization problems in people and location domains respectively.

The binary people-location co-occurrence matrix C_{FL} is defined as: the ith column of C_{FL} is the location patches that co-occur with the face patch i. For example, if the first column of C_{FL} is $(0, 0, 1, 0, 1, 0, \dots)^{\mathrm{T}}$, which means the first face patch co-occurs with the third and the fifth location patches in the same image.

$C_{FL}\tilde{Z}_F$ is a clustering of location patches which is based on the face clustering result $\tilde{Z}_F$. Our goal is to maximize the consistency between the location clustering $\tilde{Z}_L$ and $C_{FL}\tilde{Z}_F$. Location patches are weighted differently to reflect different semantic meanings of the people and location interactions. It is not difficult to discover the similarity between the definitions of f_{FL} and f_{LF} except the weight matrix T_i and P. f_{FL} optimizes the consistency that locations co-occur with the same people during a short time period should be one location. T_i is a $N_L \times N_L$ binary diagonal matrix that non-zero entries on the diagonal indicate these location patches are taken within a short time period. For example, $T_i = \text{diag}(0, 1, 0, 1, 1, \dots)$ means the second, the fourth and the fifth location patches have similar timestamps. There are t time constraints that are automatically learned from the meta-data of images.

f_{LF} optimizes the consistency that private locations are useful to identify people. P is a $N_L \times N_L$ diagonal weight matrix. It defines a score for each location patches. The private locations have larger weights and the private locations have small weights. The diagonal matrix P is defined as:

$$P_{ii} = \frac{\log(k_F/N_{F_{L_i}})}{\log(k_F)} \quad (6)$$

where P_{ii} is approximate 0 at public locations such as landmarks and it is approximate 1 at private locations. L_i is the location cluster that i belongs to. $N_{F_{L_i}}$ is the number of people appear in location L_i.

3.3 Alternative Optimization

The optimization problem (5) is not convex when the optimization variables involve $\tilde{Z}_F$ and $\tilde{Z}_L$. Therefore, we use the alternative optimization by fixing variables in one domain and optimize on other variables and do this iteratively. When fixing variables, e.g. $\tilde{Z}_F$. The problem becomes a semi-supervised kernel k-means problem, which can be solved easily. We solve the problem following this sequence until the convergence: $\tilde{Z}_L \to \tilde{Z}_F \to P \to \tilde{Z}_L \to \tilde{Z}_F \to P \to \cdots$. The first $\tilde{Z}_F$ and $\tilde{Z}_L$ are computed using the standard kernel k-means without cross-domain relations. After the initial clustering results are known, the weight matrix P can be computed using equation (6) and in the following iteration, the semi-supervised kernel k-means is used to integrate the cross-domains relations.

3.3.1 Semi-supervised Kernel K-means

We now briefly describe the existing semi-supervised kernel k-means algorithm [14]. The objective is written as the minimization of:

$$\sum_{c=1}^{k} \sum_{\mathbf{x}_i, \mathbf{x}_j \in S_c} \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{|S_c|} - \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ c_i = c_j}} \frac{2w_{ij}}{|S_c|} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ c_i = c_j}} \frac{2w_{ij}}{|S_c|} \quad (7)$$

where $\mathcal{M}$ is the set of must match link constraints, $\mathcal{C}$ is the set of cannot match link constraints, w_{ij} is the penalty cost for violating a constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, and c_i refers to the cluster label of $\mathbf{x}_i$. The first term in this objective function is the standard k-means objective function, the second term is a reward function for satisfying must match link constraints, and the third term is a penalty function for violating cannot match link constraints. The penalties and rewards are normalized by cluster size: if there are two points that have a cannot match link constraint in the same

cluster, we will penalize higher if the corresponding cluster is smaller. Similarly, we will reward higher if two points in a small cluster have a must match link constraint. Thus, we divide each w_{ij} by the size of the cluster that the points are in.

Let A be the similarity matrix $A_{ij} = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$ and let $\tilde{A}$ be the matrix such that $\tilde{A}_{ij} = A_{ii} + A_{jj}$. Then, $E = \tilde{A} - 2A$. By replacing E in the trace minimization, the problem is equivalent to the minimization of $\mathrm{tr}(\tilde{Z}^{\mathrm{T}}(\tilde{A} - 2A - 2W)\tilde{Z})$. We calculate $\mathrm{tr}(\tilde{Z}^{\mathrm{T}}\tilde{A}\tilde{Z})$ as $2\mathrm{tr}(A)$, which is a constant and can be ignored in the optimization. This leads to a maximization of $\mathrm{tr}(\tilde{Z}^{\mathrm{T}}(A+W)\tilde{Z})$. If we define a matrix $K = A + W$, our problem is expressed as a maximization of $\mathrm{tr}(\tilde{Z}^{\mathrm{T}}K\tilde{Z})$ and is mathematically equivalent to unweighted kernel k-means [7]. The iterative solution of the kernel k-means in shown in Algorithm 1.

Algorithm 1 KERNEL-KMEANS (K, k, t_{max})

Input: K: kernel matrix, k: number of clusters, t_{max}: maximum number of iterations
Output: $\{\pi_c\}_{c=1}^{k}$: final partitioning of the points

Initialize the k clusters $\{\pi_c^{(0)}\}_{c=1}^{k}$ randomly
while not converge or $t_{max} > t$ **do**
  For each point $\mathbf{x}_i$ and every cluster c, compute $d(\mathbf{x}_i, \mathbf{m}_c) = K_{ii} - \frac{2\sum_{x_j \in \pi_c} K_{ij}}{\sum_{x_j \in \pi_c} 1} + \frac{\sum_{x_j, x_l \in \pi_c} K_{jl}}{(\sum_{x_j \in \pi_c} 1)^2}$
  Find $c^*(\mathbf{x}_i) = \mathrm{argmin}_c d(\mathbf{x}_i, \mathbf{m}_c)$
  Update clusters as $\pi_c^{(t+1)} = \{\mathbf{x}_i : c^*(\mathbf{x}_i) = c\}$
end while
Return $\{\pi_c^{(t+1)}\}_{c=1}^{k}$

3.3.2 Alternative Optimization

If $\tilde{Z}_F$ is fixed and $\tilde{Z}_L$ is optimized. The objective f_{LF} can be written as:

$$f_{LF} = \mathrm{tr}(\tilde{Z}_L^{\mathrm{T}}P^{\mathrm{T}}C_{FL}\tilde{Z}_F\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}P\tilde{Z}_L) \tag{8}$$

The objective f_{FL} can be written as:

$$f_{FL} = \sum_{i=1}^{t}\mathrm{tr}(\tilde{Z}_L^{\mathrm{T}}T_i^{\mathrm{T}}C_{FL}\tilde{Z}_F\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}T_i\tilde{Z}_L) \tag{9}$$

We obtain the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \mathrm{tr}(\tilde{Z}_L^{\mathrm{T}}(2A_L + \sum_{i=1}^{t} W_{Li} + Q_L)\tilde{Z}_L) \\ \text{subject to} \quad & W_{Li} = T_i^{\mathrm{T}}C_{FL}\tilde{Z}_F\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}T_i, \\ & Q_L = P^{\mathrm{T}}C_{FL}\tilde{Z}_F\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}P, \\ & \tilde{Z}_L^{\mathrm{T}}\tilde{Z}_L = I_{k_L}. \end{aligned} \tag{10}$$

where A_L is the affinity matrix in the location domain. This optimization problem can be solved by setting the kernel matrix $K_L = 2A_L + \sum_{i=1}^{t} W_{Li} + Q_L$.

Similarly, if $\tilde{Z}_L$ is fixed and $\tilde{Z}_F$ is optimized. The objective f_{LF} can be rewritten using the fact that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ as:

$$f_{LF} = \mathrm{tr}(\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}P\tilde{Z}_L\tilde{Z}_L^{\mathrm{T}}P^{\mathrm{T}}C_{FL}\tilde{Z}_F) \tag{11}$$

The objective f_{FL} can be written as:

$$f_{FL} = \sum_{i=1}^{t}\mathrm{tr}(\tilde{Z}_F^{\mathrm{T}}C_{FL}^{\mathrm{T}}T_i\tilde{Z}_L\tilde{Z}_L^{\mathrm{T}}T_i^{\mathrm{T}}C_{FL}\tilde{Z}_F) \tag{12}$$

We obtain the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \mathrm{tr}(\tilde{Z}_F^{\mathrm{T}}(2A_F + \sum_{i=1}^{t} W_{Fi} + Q_F)\tilde{Z}_F) \\ \text{subject to} \quad & W_{Fi} = C_{FL}^{\mathrm{T}}T_i\tilde{Z}_L\tilde{Z}_L^{\mathrm{T}}T_i^{\mathrm{T}}C_{FL}, \\ & Q_F = C_{FL}^{\mathrm{T}}P\tilde{Z}_L\tilde{Z}_L^{\mathrm{T}}P^{\mathrm{T}}C_{FL}, \\ & \tilde{Z}_F^{\mathrm{T}}\tilde{Z}_F = I_{k_F}. \end{aligned} \tag{13}$$

where A_F is the affinity matrix in the face domain. This optimization problem can be solved by setting the kernel matrix $K_F = 2A_F + \sum_{i=1}^{t} W_{Fi} + Q_F$.

4. EVALUATIONS

We conduct experiments on two datasets to validate our approach. The first dataset contains images collected from personal albums with labeled ground truth. The second one uses a larger dataset crawled from online photo service: Flickr. K-means with constraints [13] is used as the baseline algorithm. Clustering on the each single domain by normalize cut [22] and Kmeans without any constraint are also compared. The RandIndex [21] is used to evaluate the performance of the clustering, which is defined as follows:

$$R = \frac{a+b}{0.5n(n-1)} \tag{14}$$

where a is the number of pairs that have the same label in the ground truth and also have the same label in the clustering result; b is the number of pairs that have different labels in the ground truth and have different labels in the clustering result. Intuitively, $a + b$ can be considered as the number of agreements between the ground truth and the clustering result. Larger value of R implies better clustering result.

4.1 Personal Albums

This dataset contains 111 images collected from personal albums. In total it has 11 people and 13 locations. The ground truth of this dataset is shown in Fig. 2. In the people domain, a pre-trained frontal face Viola Jones face detector and the nested nose detector are used to detect face with minimum size of 30×30 pixels. The detected face patches are normalized to 128×128 pixels, then warped to the canonical pose using the perspective transform. Seven landmarks are identified on each face patch. At each landmark, two SIFT descriptors of size $8, 16$ are extracted. The dimension size of features in the people domain is $1,792$.

In the location domain, the top 50 image candidates are selected for the pairwise geometric verification. In each image, the bounding box of matched interest points is extracted as the location patch. A bag of words histogram (1000 visual words), 256-bin color histogram are extracted from each location patch. The dimension size of features in the location domain is $1,256$. We use a weight ratio of $1:1$ for BoW:color features respectively. All feature vectors are $L2$ normalized. In total, there are 146 face patches and 266 location patches.

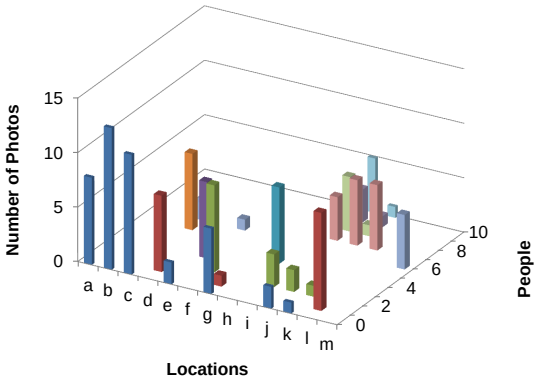


Figure 2: The ground truth of the personal albums dataset. Locations are represented by letters from a to m. People are represented by numbers from 0 to 10. The height of each bar denotes the number of photos for one people appear at one locations. There are 11 people and 13 locations in total.

In the dataset, each image associates a timestamp stored in the Exif header. The mean-shift [9] is used to cluster images in the time sequence and a matrix T_i is defined for each cluster of images. We cluster the face and location patches using the normalized cuts based on their appearance features as the baseline. K-means with constraints are also compared by taking into account of the must match links and cannot match links in each domain. Fig. 3 shows the co-clustering result in the people domain comparing to the result obtained from the baseline. Fig. 4 shows the co-clustering result in the location domain compared with the baseline.

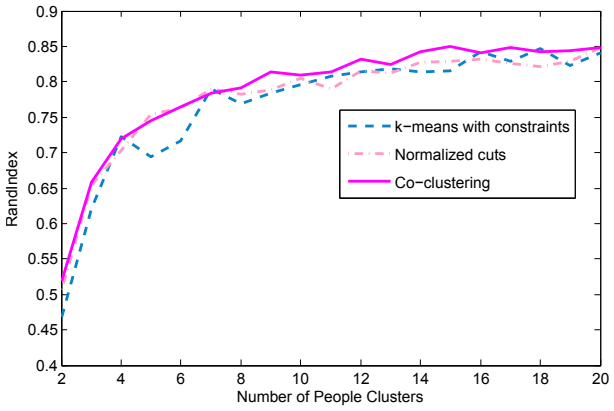


Figure 3: It shows the RandIndex in the people domain as the number of clusters changes. The ground truth of the number of clusters is 11.

The convergence of our Co-clustering algorithm is reached less than ten iterations, which is quite fast. From Fig. 3 and Fig. 4, we observe the steady improvement on the clustering results when the number of clusters is larger than 2. The k-means with constraints are quite sensitive to the number of clusters. The best RandIndex values of methods across all K values are ordered as: Co-clustering, k-means-with-constraints and Normalize cuts. The values for these methods except Co-clustering do not vary much. The performance gain of Co-clustering in the location domain is very significant. It's mainly resulted from the must match link within the location domain. For the people domain, the difference in the clustering performance is not as big as the location domain, however, the steady increase over K is still promising. The improvement is caused by the help of the links between private locations and people.

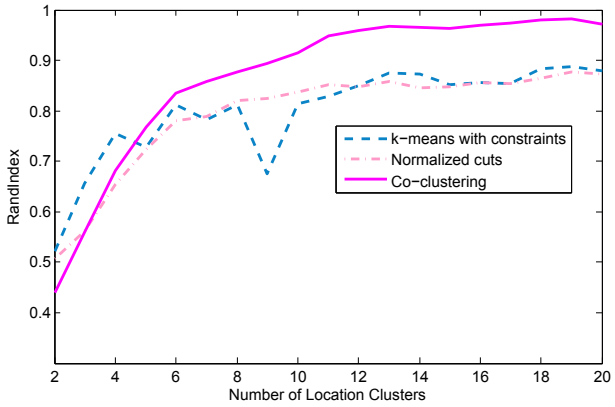


Figure 4: It shows the RandIndex in the location domain as the number of clusters changes. The ground truth of the number of clusters is 13.

4.2 Online Photo Sets

Dataset preparation: We use 140 names of public figures to query Flickr and filter out images without geo-location information. In total, we collect 53,800 images. We then use the Viola-Jonce's face detector to filter out images without faces. The ground truth of the people domain is obtained directly from names. The ground truth of the location is obtained by clustering the longitude and latitude associated with images. We use the agglomerative clustering to discover location clusters. We consider each geo-location data including the longitude and the latitude as a point in the two dimensional space. Initially, we have n points and assign them to n different clusters. In each iteration of the clustering algorithm, we merge two clusters if the distance between two clusters is the minimum among all pairs of clusters. We keep merging clusters until the minimum distance in each iteration is above a threshold or the number of clusters we want to obtain is reached. In this dataset, we set the number of locations to be 100.

After the pre-processing that throws away images without face, the dataset contains 1,446 images. The total number of people in this dataset is 132. Each image contains exact one face patch. The image upload time is used as the timestamp. Because the image is not dense enough, it is possible that one location patch does not appear in more than two images. In this case, we use the whole image as a single location patch. The statistics of this dataset is shown in Table 1. The pipeline to extract features for face and location patches is the same as that used in the personal

Table 2: The running time of the algorithm. The numbers in parenthesis are the time used using MapReduce on a 4 node cluster.

Location Preprocessing	Face Preprocessing	Alternative Optimization
224(62) mins	38(9) mins	48(14) mins

Table 1: Statistics of the Flickr dataset

Images	Face Patches	Location Patches	Must Links
1,446	1,446	2,380	1,544

Algorithm 2 LOCATION-PREPROCESSING ($\{I_i\}$, k)

Input: $\{I_i\}$: image collection, k: maximum number of image candidates for verification

Output: $\{(i,j)\}$: location patch pairs with must match links

Compute the image features (color histogram, bag of words) for each image i.

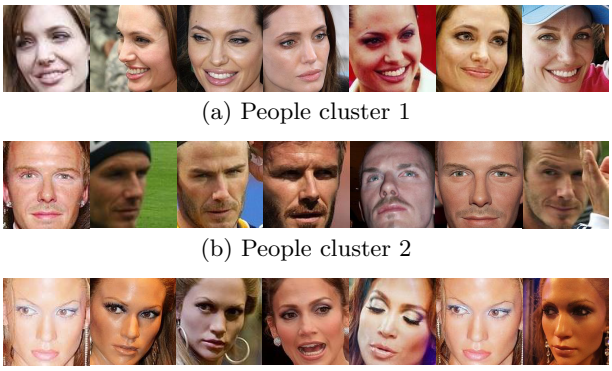
Mapper Find k nearest neighbors $id_1^i, \cdots, id_k^i$ in the feature space for image i, and emit k pairs: $< id_1^i, i >, \cdots, < id_k^i, i >$.

Reducer Verify the pairwise matching using RANSAC. If there exists a geometric transform between a pair of images, the output the must match pair.

albums dataset. However, because of the big dimension of image feature vector and the size of the data, the processing time is quite slow on the single machine. To apply the algorithm on applications which need to search over billions of images, the approach which fits in Map-Reduce framework is very critical. We develop the Map-Reduce solution of our framework to speed up the computation. Algorithm 2 has the detail for the pre-processing for the location domain, which takes the most time in the pre-processing step due to the geometric verification. The pre-processing for the people domain is easier because we only need extract feature vector for each face patch, we can distribute the computation to multiple mappers without any reducer. The approximated Map-Reduce Kernel K-means has been well studied. We adopt the algorithm in [5]. We deploy the whole process on a 4 node cluster using the MapReduce framework, the running time of each step is shown in Table 2.

Figure 7 shows RandIndex values on the people domain comparing k-means, Normalized cuts and Co-clustering over different K values. Figure 8 shows results on the location domain using the same methods. Figures show Co-clustering has good performance in both domains. The improvement is not as big as that in the personal album dataset. It is mainly caused by the noise of the image set. The ground truth of the location domain is clustered by geo-location information which is not necessary equal to the location in the image. The ground truth of the people domain could also contain noise e.g. different people with the same name may appear together within one cluster. One future work is to find efficient algorithm to deal with the noise. Nevertheless, Co-clustering still perform well which implies the inter-correlation of people and location domains indeed helps.

5. CONCLUSION

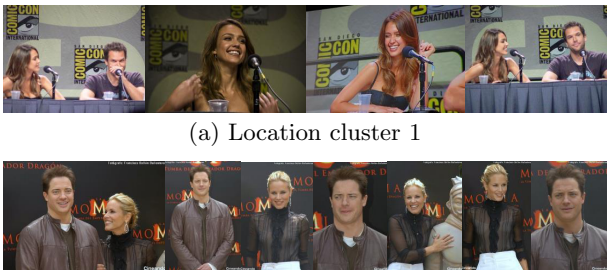


(a) People cluster 1

(b) People cluster 2

(c) People cluster 3

Figure 5: People cluster examples.



(a) Location cluster 1

(b) Location cluster 2

Figure 6: Location cluster examples

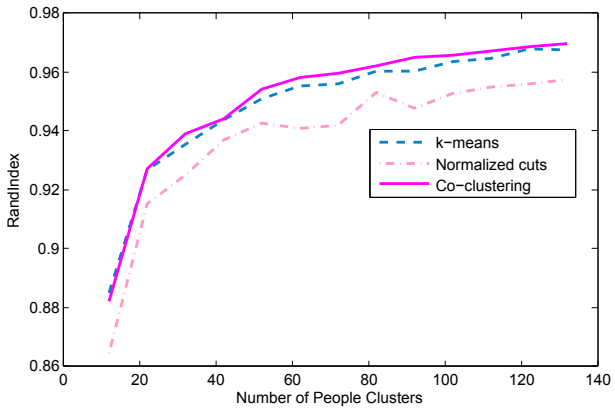


Figure 7: It shows the RandIndex in the people domain as the number of clusters changes. The ground truth of the number of clusters is 132.

We present a novel framework to co-cluster the people and location simultaneously. The relations across domains are

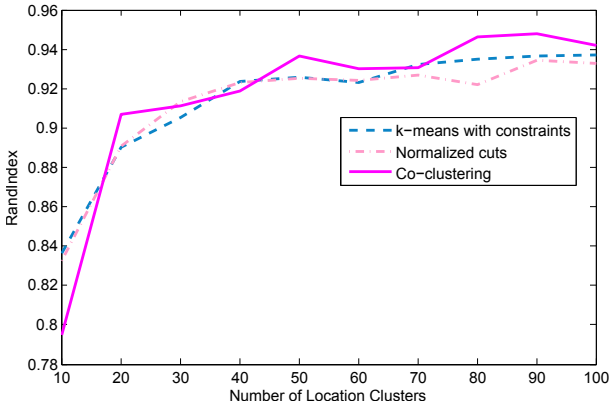


Figure 8: It shows the RandIndex in the location domain as the number of clusters changes. The ground truth of the number of clusters is 100.

used to help the clustering. An optimization problem is formulated and an iterative semi-supervised kernel k-means clustering algorithm is proposed to solve the problem. We validate our approach using images taken from personal albums with ground truth. The experiment show that our algorithm converges fast and performs better than clustering in the single domain and the baseline co-clustering algorithm. In addition, we conduct a larger scale experiment based on the MapReduce framework and exhibit that our framework is scalable.

Our work gives some insight that clustering algorithms taking cross domain relations into account can achieve better results. The clustering framework in our paper can be applied into other applications that having similar cross-domain relations. In the future, we plan to handle the noise introduced by public locations, and compare our algorithm with other co-clustering algorithms. It is interesting to explore whether our algorithm works well in other areas, for example, paper and author clustering. It's very difficult to obtain the ground truth of a large scale dataset, we are seeking a good way to measure the clustering performance over very large scale photo corpus, perhaps considering the aid of crowd source.

6. REFERENCES

[1] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, E. Learned-Miller, and D. Forsyth. Names and faces in the news. *CVPR*, 2004.

[2] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. *CVPR*, 2010.

[3] C.-Y. Chen and K. Grauman. Clues from the beaten path: Location estimation with bursty sequences of tourist photos. In *CVPR*, 2011.

[4] D. Chen, G. Baatz, Köser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. 2011.

[5] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain. Approximate kernel k-means: solution to large scale kernel clustering. In *KDD*, 2011.

[6] M. Davis, M. Smith, J. Canny, N. Good, S. King, and R. Janakiraman. Towards context-aware face recognition. In *ACM MM*, 2005.

[7] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *KDD*, 2004.

[8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[9] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 1975.

[10] J. Hays and A. Efros. IM2GPS: estimating geographic information from a single image. *CVPR*, 2008.

[11] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas. Image webs: Computing and exploiting connectivity in image collections. In *CVPR*, 2010.

[12] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *ICCV*, 2009.

[13] S. R. Kiri Wagstaff, Claire Cardie and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML*, 2000.

[14] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A kernel approach. In *ICML*, 2005.

[15] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008.

[16] D. Lin, A. Kapoor, G. Hua, and S. Baker. Joint people, event, and location recognition in personal photo collections using cross-domain context. In *ECCV*, 2010.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[18] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[21] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 2000.

[23] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[24] Y. Song and T. Leung. Context-aided human recognition–clustering. *ECCV*, 2006.

[25] M. Uřičář, V. Franc, and V. Hlaváč. Detector of facial landmarks learned by the structured output SVM. In *VISAPP*, 2012.

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

[27] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *CVPR*, 2011.

[28] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, 2009.