To appear in Proc. IEEE ICB 2015

An Efficient Approach for Clustering Face Images

Charles Otto
Michigan State University
ottochar@msu.edu

Brendan Klare
Noblis
Brendan.Klare@noblis.org

Anil Jain
Michigan State Universtiy
jain@msu.edu

Abstract

Investigations that require the exploitation of large volumes of face imagery are increasingly common in current forensic scenarios (e.g., Boston Marathon bombing), but effective solutions for triaging such imagery (i.e., low importance, moderate importance, and of critical interest) are not available in the literature. General issues for investigators in these scenarios are a lack of systems that can scale to volumes of images of the order of a few million, and a lack of established methods for clustering the face images into the unknown number of persons of interest contained in the collection. As such, we explore best practices for clustering large sets of face images (up to 1 million here) into large numbers of clusters (approximately 200 thousand) as a method of reducing the volume of data to be investigated by forensic analysts. Our analysis involves a performance comparison of several clustering algorithms in terms of the accuracy of grouping face images by identity, run-time, and efficiency in representing large datasets of face images in terms of compact and isolated clusters. For two different face datasets, a mugshot database (PCSO) and the well known unconstrained dataset, LFW, we find the rank-order clustering method to be effective in clustering accuracy, and relatively efficient in terms of run-time.

1. Introduction

As the deployment of surveillance cameras and mobile devices continues to grow, so does the size and frequency of image and video collections. In the context of forensic investigations, this represents a major issue as the exploitation of such imagery must proceed in a timely manner. Few examples are more relevant than the Boston Marathon bombing, where tens of thousands of images and videos needed to be analyzed during a time sensitive investigation [14]. Other common cases that require the investigation of media collections include identifying perpetrators and victims in child exploitation cases, an understanding of which persons exist in a collection of social media (such as imagery from gang and terrorist networks), and organizing media collections from hard drives (personal computer or servers).

The first step when investigators analyze such data is to triage the imagery. That is, the data must be filtered and organized in a manner that allows manual resources to be deployed to the most potentially useful face imagery. Often critical in this process is a clustering of the images into possibly distinct subjects in the imagery. In turn, human analysts can look through the clusters of identities to determine who may be relevant to the case at hand. While the subsequent steps from the clustering process can vary, common next steps include tagging subjects with their identity if it is known, submitting imagery to an external face recognition system for identification, or adding the subject to watch lists if they cannot be identified.

Several classic clustering challenges exist when applied to face images. These include:

- Large dataset size (millions of face images).
- Large number of classes: a crowd may contain a large number of individuals (tends of thousands, if not more).
- High intra-class variability and small inter-class separation: images are captured under unconstrained conditions, with uncooperative subjects, in difficult imaging environments.
- Unknown number of clusters: the number of individuals present in the collected data is not known a priori, but may contain tens of thousands of clusters.
- Variable number of samples per cluster: some individuals may be present in only a few images or video frames, others in many.

Despite being critical to some of the most sensitive of law enforcement cases involving face recognition, clustering of face images has received relatively little attention (see Section 2). Aside from the classical challenges mentioned above, other application-specific issues include: (i) the lack of unified frameworks for exploiting face media, (ii) a lack of understanding of what clustering algorithms are the most accurate given a large number of samples (n), subjects (C),

Table 1. A summary of related works in face clustering.

Publication	Features	Clustering method	# Images	# Subjects
Ho et al. [8]	Gradient and Pixel intensity features	Spectral clustering	1,147	66
Zhao et al. [19]	2D-HMM + contextual	Hierarchical clustering	1,500	8
Cui et al. [5]	LBP, clothing color + texture	Spectral	400	5
Tian et al. [15]	Image + contextual	Partial clustering	1,147	34
Zhu et al. [20]	Learning-based descriptor [3]	rank-order hierarchical	1,322	53
Vidal and Favaro [16]	Joint subspace learning and clustering		2,432	38
Wang et al. [18]	Learning-based descriptor [3]	rank-order hierarchical	500K	5,749
Ours	Component Based and COTS	multiple	1 million	195,494

and well tuned facial features (d), and (iii) how to scale the clustering process to accommodate both time sensitive investigations and limited computing resources.

This work provides a unified framework for clustering face images at scale. Contributions of our work include: (i) the largest scale evaluation of face clustering to date, (ii) the use of face recognition algorithms representative of state of the art approaches (as opposed to weaker features such as pixels), and (iii) a unified framework for ingesting, enrolling, comparing, and clustering face images amid the aforementioned classical and application-specific challenges.

2. Background

The clustering problem is well studied in pattern recognition, statistics, and machine learning literature (Jain [10] provides a survey). Less studied is the challenging problem of clustering face images. An important consideration in handling face images is that since there is no universally agreed upon face representation or distance metric, the clustering results depend not only on the choice of clustering algorithm, but also on the quality of the underlying face representation and metric. Table 1 lists some prior works on face clustering, with the face representation and clustering algorithm used, along with the largest dataset size employed in terms of face images, as well as number of subjects.

Ho et al. [8] develop variations on spectral clustering wherein the affinity matrix is computed based on (i) assuming a Lambertian object (with fixed camera/object positioning), and then computing the probability that two face images are of the same object (same convex polyhedral cone in the image space), or (ii) the local gradients of the images being compared. They report results on the Yale-B, and PIE-66 datasets (dataset size is 1,147 images). Assuming a fixed camera position is not realistic, so it is difficult to credit the assumptions used in the conic affinity method; additionally, the face datasets used are rather small.

Zhao et al. [19] develop an application for clustering personal photograph collections. Their approach is to combine a variety of contextual information (including time based clustering, and the probability for certain people to appear together in images) with identity estimates obtained via a 2D-HMM, and hierarchical clustering results based on body detection. Their method is evaluated on a dataset of 1,500 face images of 8 individuals.

Cui et al. [5] develop a semi-automatic tool for annotating photographs, which employs clustering as an initial method for organizing photographs. LBP features are extracted from detected faces, and color and texture features are extracted from detected bodies. Spectral clustering is performed, and the clustering results can then be manually adjusted by the human operator. Evaluation is done on a dataset consisting of 400 photographs of 5 subjects. Tian et al. [15] further develop this approach, incorporating a probabilistic clustering model, which incorporates a "junk" class, allowing the algorithm to discard clusters which do not have tightly distributed samples.

Zhu et al. [20] develop a dissimilarity measure based on the rankings of two samples being compared in the opposite samples nearest neighbor lists (formed using a basic distance metric), and perform hierarchical clustering based on that rank-order distance function. The primitive feature representation used is the result of unsupervised learning [3]. The clustering method is evaluated on several small datasets (the largest of which contains only 1,322 face images). Wang et al. [18] primarily develop an approximate k-nn graph construction method; in one of their experiments they apply this method to construct the nearest neighbor lists required by [20], on a dataset containing 500K images.

Vidal and Favaro [16] develop a joint subspace learning and clustering approach which derives several subspaces from the input dataset which best capture clusters in the data. They evaluate the method on the extended Yale-B database.

For clustering images in general, rather than faces in particular, Liu et al. [12] (i) extracted Haar wavelet features from images, (ii) applied a distributed algorithm consisting of an approximate nearest neighbor step, (iii) generated an initial set of clusters by applying a distance threshold to the nearest neighbor lists, and (iv) applied a union-find algorithm to get a final set of clusters. Clustering was performed

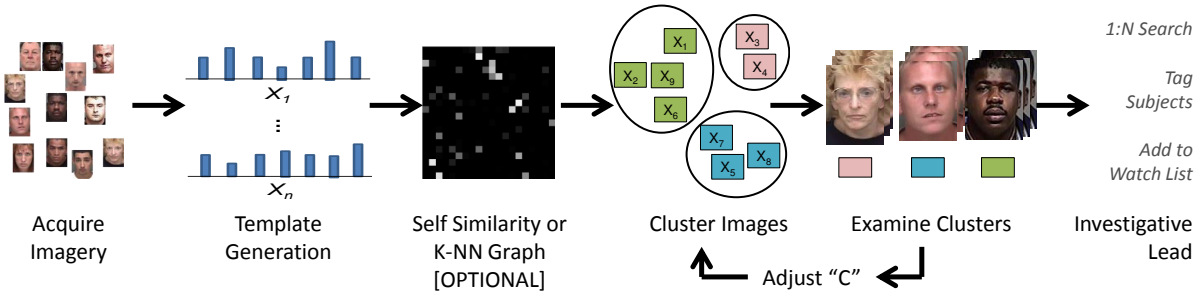

Figure 1. Face clustering is a vital, yet time consuming, process for triaging large sets of images. Shown is an overview of the process for clustering face images in forensic scenarios. Given a corpus of images acquired in an investigation, the first step is to perform enrollment to generate a representation (template) of all faces. An optional step is to compute the self similarity, or an approximate k-nearest neighbor (k-NN) graph between all faces templates. As shown in this work, performing this step greatly improves the efficiency of the clustering process. Using either the k-NN graph or the templates/feature vectors, clustering is next performed to group all subjects in the corpus into distinct clusters. Because the number of subjects is not known a priori, this process is often repeated with different values of C until it best approximates the number of subjects. Finally, the resultant information could be used to build investigative leads. This paper provides a unified framework for clustering face images that addresses both the clustering accuracy and efficiency considerations.

on approximately 1.5 billion unlabeled images, along with an evaluation on 3,385 labeled images.

Foo et al. [6] consider a related problem, the detection of near-duplicate images in large datasets. In this case, rather than grouping images of people by identity, the goal is to identify near-duplicate images, which may be the result of various image processing operations, such as cropping, rotation, colorspace conversion, etc. Their image representation consists of applying a visual words approach to local PCA-SIFT descriptors, indexed with a Locality Sensitive Hashing (LSH) scheme. The clustering method used is a union-find algorithm. Evaluation was performed by generating a synthetic set of near duplicate images, and performing clustering in the presence of a separate noise set; the largest dataset used was 300,000 images.

3. Approach

The basic process of clustering an unlabeled set of face images consists of two major parts: feature extraction from face images, followed by the application of a clustering algorithm. For clustering algorithms leveraging local neighborhood information (such as the rank-order clustering method of Zhu et al. [20], or spectral clustering leveraging k-nearest neighbor graphs), the clustering step may further be broken down into a (re-usable) nearest neighbor computation step, and a final clustering step based on the nearest neighbor information.

3.1. Face Recognition Algorithms

Two face recognition algorithms are used in this study: (i) a component-based algorithm (listed as Component) based on the method presented by Bonnen *et al*. [2], which was implemented within the open-source OpenBR framework [11], and (ii) a commercial off the shelf (COTS) matcher, which is anonymized due to licensing agreements, but is one of the top performing algorithms in the NIST FRVT 2014 evaluations (listed as COTS). As is typically the case, commercial algorithms do not allow access to underlying feature vectors; as such, certain clustering approaches described in this paper are only presented with the "Component" face recognition algorithm.

The component algorithm can be outlined as follows: detect keypoints using the STASM library [13]; based on the detected keypoints extract, local regions containing the subject's nose, eyes, mouth, and eyebrows; extract LBP and HOG features from each extracted region; apply PCA for dimensionality reduction; and finally, concatenate the features from each local region, and apply LDA on the resulting feature vector.

3.2. k-NN Graph Construction

A k nearest neighbor (k-NN) graph is a weighted graph where each instance (a face image in our case) has edges connecting the other k closest instances. Here, the weights are similarity values from the respective face recognition algorithms. To exactly compute a k-NN graph, the entire self similarity matrix needs to first be computed. In turn, a sorting process (or similar approach) is performed to find the nearest instances. From a memory perspective, it is more efficient to store a k-NN graph instead of a full similarity matrix; the k-NN graph can also be referred to as a sparse matrix representation of the full similarity matrix.

For clustering algorithms which leverage nearest neighbor information, such as rank-order clustering or some variations of spectral clustering, computing the nearest neigh-

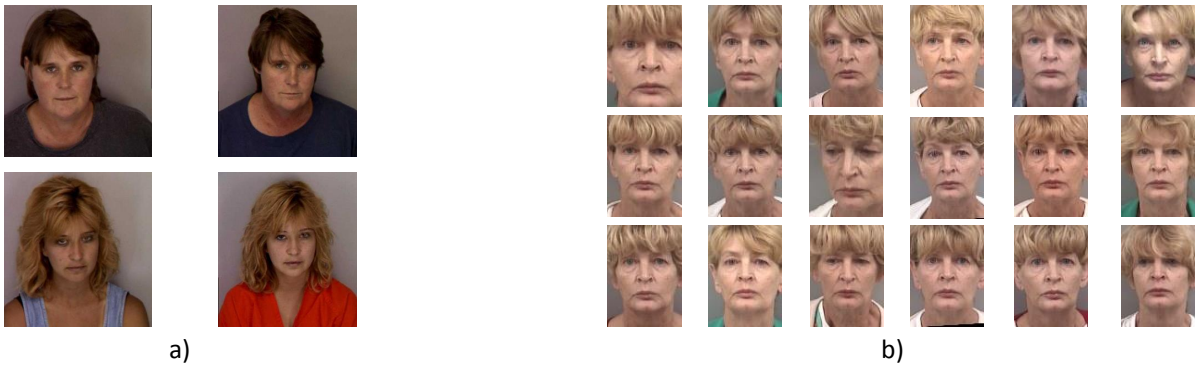

Figure 2. Clustering results: (a) heterogeneous (unsuccessful), and (b) homogeneous (successful) clusters, from the PCSO dataset, generated via rank-order clustering with Component features.

bors of every sample constitutes a major computational cost. In the brute force manner described above, given n samples, the computational cost is O(n^2). Thus, even if the basic comparison method is relatively fast, on large datasets the cost of computing the nearest neighbors will dwarf the cost of enrolling the face images (see Table 4).

3.2.1 Parallel k-NN Graph Construction

One obvious approach to speed up nearest neighbor computation is parallelization; the nearest neighbors of every sample may be computed simply by comparing each sample against the gallery in parallel. While such a parallelization method is efficient, it can only produce a speedup linear with the amount of additional hardware employed; meanwhile, the computational cost of processing larger datasets increases with the square of dataset size.

3.2.2 Approximate k-NN Graph Construction

The disparity between the speedup achievable via parallelization, and quadratic cost of computing nearest neighbors suggests that it may be valuable to purse a subquadratic approximation method for computing the nearest neighbors. As such, we explore the implications of using the "glue" method of Chen et al. [4], which employs a divide and conquer algorithm to achieving O(n^t) runtime, where t is configurable such that $1 \leq t \leq 2$.

The original algorithm is a recursive procedure outlined as follows:

1. If the number of instances in the current set is less than a threshold, compute them exhaustively.
2. Otherwise, compute a separating hyperplane using Lanczos bisection.
3. Divide the feature vectors into 3 sets, the two sets separated by a hyperplane, and a third set overlapping both.
4. Recurse to step 1 above on all 3 sets.
5. Refine results from step 4 by, for each point, checking if the children of the currently found neighbors are closer than the current k-NN candidates.

We parallelize this procedure by recursing on the two disjoint sets found in step 3 in parallel. Then, after those recursive calls finish, we recurse on the overlap partition and perform refinement. Since the two sets separated by the hyperplane are disjoint, those distances computed in one half will not be needed in the other, allowing us to process them independently. Threads are managed via a thread pool, so although we set up two separate jobs in each recursive call, the total number of threads used is fixed to the number of available cores on the computer.

The runtime of the algorithm is determined by the degree of overlap chosen for the middle partition. In our experiments the runtime used is O($n^{1.3}$). Subsequent experiments will demonstrate the tradeoff present at runtime to compute nearest neighbors vs. approximation (and ultimately clustering) accuracy.

3.3. Clustering Algorithms

We study three well known clustering algorithms: k-means, spectral clustering, and the rank-order method of Zhu et al. [20]. The k-means algorithm is widely used in general, spectral clustering has been used in several prior works on face clustering, and the rank-order method has been tested on relatively large datasets.

3.3.1 k-means

In k-means, the clustering problem is defined as minimizing the total square distance of a set of feature vectors to the nearest of C cluster centers. Finding the exact solution to the k-means objective is not feasible, so in practice an

a) b)

Figure 3. Clustering Results: (a) heterogeneous cluster (unsuccessful), and (b) homogeneous cluster (successful) , from the LFW dataset, generated via rank-order clustering with Component features.

approximate solution is typically reached via Lloyd's algorithm, which can be outlined as follows: (i) initialize cluster centers (we follow the k-means++ seeding procedure of Arthur and Vassilvitskii [1]), (ii) assign each point in the dataset to the nearest cluster center, (iii) recompute cluster centers as the mean of all feature vectors assigned to each center, and (iv) repeat steps (ii)-(iii) until convergence.

3.3.2 Spectral Clustering

Spectral clustering [17] approaches the problem from a graph theory perspective. The first step is to construct an adjacency matrix for the target feature vectors, describing the dataset as a graph. If no inherent graph structure is known, as is the case for general face clustering, the adjacency matrix can be constructed in several ways. One option is to construct a fully connected graph, wherein each value in the adjacency matrix is the similarity between the corresponding samples; otherwise, a sparse adjacency matrix may be constructed, by either retaining all edges with a similarity above a threshold, or retaining a fixed number of edges with the greatest weights.

After the adjacency matrix is defined, the normalized Laplacian is computed, followed by the top C eigenvectors of the normalized Laplacian, and then a new matrix is formed whose columns consist of the computed eigenvalues. Considering each row of this matrix a new sample (corresponding to the n original samples), k-means clustering is carried out on the new data representation.

3.3.3 Rank-Order Clustering

The rank-order clustering algorithm proposed by Zhu et al. [20], similar to the method of Gowda and Krishna [7], is a form of agglomerative hierarchical clustering, leveraging a sophisticated distance metric. The overall procedure for agglomerative hierarchical clustering, given some distance metric, is to initialize all samples to be separate clusters, then iteratively merge the two closest clusters together. This requires defining a cluster-to-cluster distance metric. In this case, the distance between two clusters is considered to be the minimum distance between any two samples in the clusters.

The first distance metric used in Rank-Order clustering is given by:

$$d(a,b) = \sum_{i=1}^{O_a(b)} O_b(f_a(i))$$

where $f_a(i)$ is the ith face in the neighbor list of a, and $O_b(f_a(i))$ gives the rank of face $f_a(i)$ in face b's neighbor list. This asymmetric distance function is then used to define a symmetric distance between two faces as:

$$D(a,b) = \frac{d(a,b) + d(b,a)}{min(O_a(b), O_b(a))}$$

The symmetric rank order distance function gives low values if the two points are close to each-other (are high in the opposite data point's rank list), and have several neighbors in common.

4. Datasets

4.1. PCSO Subsets

The Pinellas County Sheriff's Office (PCSO) dataset is a set of mugshot images available in the public domain through Florida's "Sunshine" laws. The full dataset consists of approximately 1.4 million images of 400,000 subjects (Figure 2 displays some examples). Images in the PCSO dataset have an average interpupillary distance (IPD) of approximately 109 pixels. We have sampled several subsets of this dataset, with sizes listed in Table 2. Subjects were

# Images (# Subjects)	Rank-Order Clustering			k-Means	Spectral
	Component	Component*	COTS	Component	Component
1,001 (201)	0.88 (242)	0.88 (243)	0.90 (172)	0.49 (201)	0.74 (201)
10,002 (2,150)	0.87 (2,937)	0.85 (3,235)	0.94 (2,090)	0.40 (2,150)	0.53 (2,150)
50,002 (10,908)	0.83 (15,047)	0.75 (18,631)	0.93 (10,304)	0.34 (10,908)	-
100,004 (21,996)	0.79 (31,262)	0.70 (40,471)	0.91 (20,655)	0.33 (21,996)	-
1,000,008 (195,494)	0.64 (246,785)	0.49 (442,956)	0.76 (159,118)	-	-

Table 2. Clustering accuracy, and number of clusters (reported as "F-Measure (# Clusters)") as a function of dataset size on the PCSO dataset using different clustering algorithms (Rank-Order, k-Means, and Spectral), and different face recognition algorithms (Component and COTS). Entries labeled Component* use the approximate nearest neighbor method discussed in 3.2.2. The k-Means and Spectral cluster results use the Component face recognition algorithm features. An entry of "-" means that the corresponding algorithm could not finish the clustering in a reasonable amount of time.

Dataset	Rank-Order Clustering		
	Component	Component*	COTS
LFW	0.33 (4,235)	0.33 (4,231)	0.39 (5,049)
LFW+	0.15 (647k)	0.14 (770k)	-

Table 3. Clustering accuracy, and number of clusters (reported as "F-Measure (# Clusters)") on the original, and augmented LFW datasets. LFW contains 5,749 subjects, the LFW+ dataset contains all LFW subjects plus an unknown number of additional subjects. Component* indicates that the approximation method discussed in Section 3.2.2 was used to compute the nearest neighbors for the rank-order clustering algorithm.

randomly drawn from the PCSO dataset, under the condition that each subject selected had at least two images in the dataset. Since the subjects in each subset were sampled uniformly from all available subjects in the complete dataset, the distribution of number of images per subject remains roughly the same for all sizes of PCSO subsets.

4.2. LFW and LFW+ Unconstrained Face Datasets

We also evaluate clustering performance on the well known Labeled Faces in the Wild (LFW) dataset [9] (some examples are shown in Figure 3). In order to consider a more challenging scenario, we augment LFW with 1 million images collected via crawling the internet to define the LFW+ dataset. These images were filtered to only include images with faces detectable by the OpenCV implementation of the Viola-Jones face detector, similar to the procedure used to select LFW images. Since ground truth identity information is unavailable for the additional 1 million images, performance on the augmented dataset is calculated by computing precision and recall while only considering data for which identity labels are available.

5. Experiments

5.1. Clustering Accuracy

We evaluate clustering performance using pairwise precision/recall. Precision is defined as the average fraction of face image pairs assigned to a cluster with matching class labels, and recall is defined as the average fraction of face image pairs belonging to the same class assigned to the same cluster. F-measure is a summary statistic for precision/recall, defined as $F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$. Table 2 contains F-measure values for the evaluated clustering algorithms and matchers on the PCSO datasets. For the rank-order algorithm, the score threshold is varied, and the score threshold and consequent number of clusters resulting in the highest F-measure is reported. The best results in terms of F-measure are typically attained using a somewhat higher number of clusters than is present in the ground truth, although using an arbitrarily high number of clusters is punished since eventually losses in recall offset gains in precision. For k-means and spectral clustering, the exact (true) number of clusters is specified.

Clustering accuracy, as expected, generally decreases as dataset size increases, with a significant accuracy dropoff on the 1 million image PCSO dataset. The approximate k-NN method results in worse overall accuracy than the brute force method, and the gap in performance increases with dataset size, up to a 0.15 gap in F-measure on the one million dataset.

Results on the original and augmented LFW datasets are reported in Table 3. For both the Component and COTS matchers, face recognition performance is significantly worse on the unconstrained LFW images, leading to relatively low clustering accuracy.

In terms of clustering algorithms, rank-order clustering consistently has the most accurate results, followed by spectral clustering, followed by k-means. Comparing face matchers, the best results are attained using the COTS matcher for all datasets, although since no feature vectors are available, neither k-means nor the approximate k-NN graph construction method can be used with this matcher. The relative performance of the face matchers is consistent across datasets, and in all cases clustering accuracy decreases with increasing dataset size. Overall, the clustering accuracy decreases dramatically on the one million image

Dataset Size (# Subjects)	Enroll	NN	Approximate NN	k-Means	Spectral	Rank-Order
1,001 (201)	0:00.14	0:00.01	0:00.01	0:00.02	0:00.12	0:00.01
10,002 (2,150)	0:02.02	0:00.09	0:00.10	0:02.47	1:59.28	0:00.01
50,002 (10,908)	0:09.42	0:03.51	0:01.26	1:12.12	-	0:00.02
100,004 (21,996)	0:18.23	0:15.37	0:03.44	4:50.15	-	0:00.04
1,000,008 (195,494)	3:05.38	26:40.37	1:15.26	-	-	0:02.43

Table 4. Runtimes for different stages of the face clustering framework (enrollment, nearest neighbor (NN) computation, and clustering). Enrollment is performed with the Component algorithm. Approximate NN denotes the use of the algorithm described in Section 3.2.2. Times are listed as Hours:Minutes.Seconds. All times were measured on a 20-core server.

# Images (# Subjects)	Enroll	NN	R-O
1,001 (201)	0:00.32	0:00.13	0:00.01
10,002 (2,150)	0:04.13	0:00.50	0:00.01
50,002 (10,908)	0:36.48	0:04.12	0:00.02
100,004 (21,996)	1:09.52	0:19.57	0:00.03
1,000,008 (195,494)	12:50.31	32:34.26	0:02.51

Table 5. Runtimes for the COTS algorithm on PCSO subsets in Hours:Minutes.Seconds; column R-O is for the Rank-Order algorithm.

dataset, to at best 0.76 F-measure, from 0.91 on the 100,000 mugshot dataset.

Some examples of successful, and unsuccessful clusters are shown in Figures 2 and 3, generated using the rank-order clustering algorithm with Component features. It seems better clusters are formed when the number of faces images for a subject is large, as in Figures 2(b) and 3(b)

5.2. Runtime

Tables 4 and 5 break down the runtime of the evaluated clustering algorithms on several datasets. Runtimes were measured using a server with 20 cores clocked at 2.5GHz, leveraging available multi-threading.

Enrollment by a particular face matcher is a necessary first step in the clustering process. Enrollment time can be a significant portion of total runtime, particularly for small datasets; however, enrollment time is linear with the number of images, and will be dwarfed by other costs as the dataset size increases. Both the rank-order and spectral clustering algorithms compute a set of nearest neighbors for each sample. This cost is initially low for the Component algorithm since the actual comparison function is quite efficient; however, the computation cost increases with the square of dataset size, and becomes the dominant cost for datasets on the order of one million faces.

For rank-order clustering, nearest neighbor computation is the dominant cost for large datasets, followed by the cost of enrollment. The actual clustering step itself is rather quick, since all distance computations are already done. On the other hand, spectral clustering, which also computes nearest neighbors, has significant additional costs in eigenvector computation, as well as a k-means clustering step. The cost of computing the eigenvectors is cubic with datasets size, and quickly dominates both enrollment and nearest neighbor calculation.

k-means does not compute a k-nn graph; however, its basic loop (which compares each sample to the current cluster centers) has a runtime comparable to the cost of computing the k-NN graph. Since the number of clusters C is within a constant factor of the total number of samples (approximately 5 samples per cluster), O(nC) operations (such as comparing all samples to all cluster centers) are in fact O(n^2), and the cost per iteration of the k-means algorithm becomes quite high for large datasets. In fact, even after running the algorithm for 4 days on the 1 million image PCSO dataset it failed to converge.

5.3. Dataset Summarization

We can evaluate clustering results by measuring the consistency of the results with the ground truth identity labels; however, this does not directly address the application of summarizing a dataset to allow an analyst to investigate it more efficiently. We therefore adapt the penetration/hit rate plot typically used to evaluate indexing applications, and plot the fraction of dataset retained after replacing all members of a cluster with a single exemplar (Penetration Rate) vs. the fraction of distinct identities still represented in the reduced dataset (Subject Hit Rate). A tradeoff between the degree of consolidation vs. number of subjects retained is observed, and several operating points can be evaluated by varying the number of clusters the dataset is reduced to. Figure 4 plots the penetration vs. hit rate for the rank-order clustering algorithm on the 1 million image PCSO subset.

In practice, 90% of subjects are retained while still reducing the effective dataset size to approximately one third of its original size. This shows that for subjects with a large number of face images in the dataset, the clustering is very effective. The 90% of subjects remaining in the dataset have relatively few images per subject. In this sense, the face clustering is effective in identifying dense clusters from noisy background clusters.

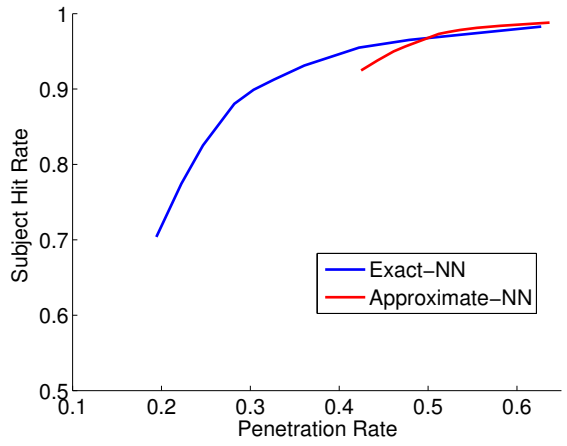

Figure 4. Hit rate vs. Penetration rate for the PCSO 1 million image dataset. Results are shown for Rank-Order Clustering, with features provided by the Component algorithm.

6. Conclusions

We have examined the challenging problem of face clustering from the perspective of applications in forensics and law enforcement. This application entails clustering a very large number of unconstrained face images (say, a million) into a very large, but unknown number of clusters (say, 100,00). Of the several clustering methods evaluated, rank-order clustering consistently displayed a good tradeoff between clustering accuracy and computational requirements. Further, the runtime characteristics of the algorithm (performance bound by k-NN computation) easily allows for use with varying thresholds (useful for evaluating different possible numbers of cluster centers). Although the method is relatively efficient, the O(n^2) computational cost of computing the k-NN graph eventually limits its utility, which can be remedied to an extent by applying an approximation method (at the cost of clustering accuracy). Lastly, we observe that for large datasets (on the order of 1 million images), while the clustering accuracy decreases, it is still able to identify some subject-specific (homogeneous) clusters, provided the number of face images of the subject is large.

Our ongoing work includes exploring the use of (i) incorporating pairwise constraints (must-link and cannot-link) and (ii) leveraging clustering ensembles to improve the clustering performance.

7. Acknowledgement

This research was supported by the National Institute of Justice (NIJ) grant 2011-IJ-CX-K057.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

[2] K. Bonnen, B. F. Klare, and A. K. Jain. Component-based representation in automated face recognition. *IEEE TIFS*, 8(1):239–253, 2013.

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *CVPR*, pages 2707–2714. IEEE, 2010.

[4] J. Chen, H.-r. Fang, and Y. Saad. Fast approximate k nn graph construction for high dimensional data via recursive lanczos bisection. *JMLR*, 10:1989–2012, 2009.

[5] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *SIGCHI conf. on Human factors*, pages 367–376. ACM, 2007.

[6] J. J. Foo, J. Zobel, and R. Sinha. Clustering near-duplicate images in large collections. In *ACM Workshop on multimedia information retrieval*, pages 21–30, 2007.

[7] K. C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978.

[8] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, volume 1, pages I–11, 2003.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, Univ. of Massachusetts, Amherst, October 2007.

[10] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[11] J. C. Klontz, B. F. Klare, S. Klum, A. K. Jain, and M. J. Burge. Open source biometric recognition. In *IEEE BTAS*, pages 1–8, 2013.

[12] T. Liu, C. Rosenberg, and H. A. Rowley. Clustering billions of images with large scale nearest neighbor search. In *WACV*, pages 28–28, 2007.

[13] S. Milborrow and F. Nicolls. Active Shape Models with SIFT Descriptors and MARS. *VISAPP*, 2014.

[14] B. S. Swann. FBI video analytics priority initiative. In *17th Annual Conference & Exhibition on the Practical Application of Biometrics*, 2014.

[15] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang. A face annotation framework with partial clustering and interactive labeling. In *CVPR*, pages 1–8, 2007.

[16] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, 2014.

[17] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[18] J. Wang, J. Wang, G. Zeng, Z. Tu, R. Gan, and S. Li. Scalable k-nn graph construction for visual descriptors. In *CVPR*, pages 1106–1113, 2012.

[19] M. Zha, Y. Teo, S. Liu, T. Chua, and R. Jain. Automatic person annotation of family photo album. In *Image and Video Retrieval*, pages 163–172. Springer, 2006.

[20] C. Zhui, F. Wen, and J. Sun. A rank-order distance based clustering algorithm for face tagging. In *CVPR*, pages 481–488, 2011.