Crowdclustering with Sparse Pairwise Labels: A Matrix Completion Approach

Jinfeng Yi[†], Rong Jin[†], Anil K. Jain[†], Shaili Jain[*]
[†]Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
{yijinfen, rongjin, jain}@cse.msu.edu
[*]Department of Computer Science, Yale University, New Haven, CT 06520, USA
shaili.jain@yale.edu

Abstract

Crowdsourcing utilizes human ability by distributing tasks to a large number of workers. It is especially suitable for solving data clustering problems because it provides a way to obtain a similarity measure between objects based on manual annotations, which capture the human perception of similarity among objects. This is in contrast to most clustering algorithms that face the challenge of finding an appropriate similarity measure for the given dataset. Several algorithms have been developed for crowdclustering that combine partial clustering results, each obtained by annotations provided by a different worker, into a single data partition. However, existing crowdclustering approaches require a large number of annotations, due to the noisy nature of human annotations, leading to a high computational cost in addition to the large cost associated with annotation. We address this problem by developing a novel approach for crowclustering that exploits the technique of **matrix completion**. Instead of using all the annotations, the proposed algorithm constructs a **partially observed** similarity matrix based on a subset of pairwise annotation labels that are agreed upon by most annotators. It then deploys the matrix completion algorithm to complete the similarity matrix and obtains the final data partition by applying a spectral clustering algorithm to the completed similarity matrix. We show, both theoretically and empirically, that the proposed approach needs only a small number of manual annotations to obtain an accurate data partition. In effect, we highlight the trade-off between a large number of noisy crowdsourced labels and a small number of high quality labels.

Introduction

Crowdsourcing is a new business model that has grown rapidly in recent years. It provides an easy and relatively inexpensive way to accomplish small-scale tasks, such as *Human Intelligence Tasks* (HITs), and to effectively utilize human capabilities to solve difficult problems. Typically, in a crowdsourcing scenario, each human worker is asked to solve a part of a big problem, and a computational algorithm is then developed to combine the partial solutions into an integrated one. Crowdsourcing has been exploited by a number of machine learning tasks (e.g., classification, clustering and segmentation) that require object (e.g., image) labeling or annotation (Tamuz et al. 2011; Wauthier and Jordan 2011; Karger, Oh, and Shah 2011; Raykar and Yu 2011; Yan et al. 2011; Vijayanarasimhan and Grauman 2011; Welinder et al. 2010).

In this work, we focus on crowdclustering that applies the crowdsourcing technique to data clustering. Given a collection of objects to be clustered, a subset of objects is first sampled in each HIT, and a worker is asked to annotate the subset of objects in the HIT based on their own opinion. The annotation task can either be grouping objects based on their similarities or describing individual objects by multiple keywords; the annotation results are usually summarized in the form of pairwise constraints. The keyword annotation is transformed into binary pairwise constraints by checking if two objects share common annotated keywords. The results of each HIT, which can be considered as a (partial) local clustering of the objects in that HIT, are then combined to form a data partitioning of the entire data set.

The main advantage of crowdclustering is that it explores the crowdsourcing technique to address one of the key challenges in data clustering, namely how to define the similarity measure between objects. A typical clustering algorithm measures the similarity between two objects (data points) based on their attributes, which often does not reflect human perception of inter-object similarity. In contrast, crowdclustering utilizes human power in acquiring pairwise similarities by asking each worker to perform clustering on a subset of objects, thereby defining a similarity measure between pairs of objects based on the percentage of workers who put them into the same cluster.

The core of crowdclustering is to combine the partial clustering results, generated by individual workers, into a complete data partition. One way to address this challenge is ensemble clustering (Fred and Jain 2002; Strehl and Ghosh 2002), as suggested in (Gomes et al. 2011). There are, however, two special challenges in applying ensemble clustering to the crowdclustering problem. First, since each worker can only deal with a subset of the entire dataset, only partial clustering results are available in the ensemble for combination. This is in contrast to most ensemble clustering studies that require a clustering of the complete dataset from individual partitions. Second, since different human workers may have different clustering criterion, they may produce various partial clustering results. This usually introduces a significant amount of noise and inter-worker variations in their cluster-

ing results. As a consequence, we often observe a large number of uncertain data pairs for which about half of the human workers put them into the same cluster while the other half do the opposite. These uncertain data pairs can mislead the ensemble clustering algorithms to create inappropriate data partitions.

To address the potentially large variations in the pairwise annotation labels provided by different workers (i.e. whether or not two objects should be assigned to the same cluster), a Bayesian generative model was proposed for crowdclustering in (Gomes et al. 2011). It explicitly models the hidden factors that are deployed by individual workers to group objects into the same cluster. The empirical study in (Gomes et al. 2011) shows encouraging results in comparison to the ensemble clustering methods. However, one limitation of the Bayesian approach for crowdclustering is that in order to discover the hidden factors for clustering decision, it requires a sufficiently large number of manual annotations, or HITs. This results in high cost, both in computation and annotation, which limits the scalability to clustering large data sets.

To overcome the limitation of the Bayesian approach, we propose a novel crowdclustering approach based on the theory of matrix completion (Candès and Tao 2010). The basic idea is to first compute a *partially observed* similarity matrix based only on the **reliable** pairwise annotation labels, or in other words, the labels that are in agreement with most of the workers. It then completes the partially observed similarity matrix using a matrix completion algorithm, and obtains the final data partition by applying a spectral clustering algorithm (Ng, Jordan, and Weiss 2001) to the completed similarity matrix.

The main advantage of the matrix completion approach is that only a small number of pairwise annotations are needed to construct the partially observed similarity matrix. This way, we can obtain a clustering accuracy similar to the Bayesian methods, with a substantial reduction in the number of workers and/or the number of HITs performed by individual workers. The high efficiency of the proposed algorithm in exploiting manual annotations arises from a key observation, i.e. the complete similarity matrix for all the objects is generally of low rank (Jalali et al. 2011). According to the matrix completion theory (Candès and Tao 2010), when an $n \times n$ matrix is of low rank, it can be perfectly recovered given only a very small portion of entries (i.e. $O(\log^2 n/n)$). Another advantage of the proposed crowclustering algorithm is that by filtering out the uncertain data pairs, the proposed algorithm is less sensitive to the noisy labels, leading to a more robust clustering of data.

Crowdclustering by Matrix Completion

The key idea of the proposed algorithm is to derive a *partially observed* similarity matrix from the partial clustering results generated by individual workers, where the entries associated with the uncertain data pairs are marked as *unobserved*. A matrix completion algorithm is applied to complete the partially observed similarity matrix by filtering out the unobserved entries. Finally, a spectral clustering algorithm (Ng, Jordan, and Weiss 2001) is applied to the completed similarity matrix to obtain the final clustering. Below, we describe in detail the two key steps of the proposed algorithm, i.e., the *filtering step* that removes the entries associated with the uncertain data pairs from the similarity matrix, and the *matrix completion step* that completes the partially observed similarity matrix.

The notations described below will be used throughout the paper. Let N be the total number of objects that need to be clustered, and m be the number of HITs. We assume that the true number of clusters in the data is known a priori. [1] Given the partial clustering result from the k-th HIT, we define a similarity matrix $W^k \in \mathbb{R}^{N\times N}$ such that $W^k_{ij} = 1$ if objects i and j are assigned to the same cluster, 0 if they are assigned to different clusters, and -1 if the pairwise label for the two objects can not be derived from the partial clustering result (i.e. neither object i nor object j is used in the HIT). Finally, given a subset of object pairs $\Delta \subset \{(i,j),\ i,j = 1,\dots N\}^2$, we define a matrix projection operator $\mathcal{P}_\Delta : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ that takes a matrix B as the input and outputs a new matrix $\mathcal{P}_\Delta(B) \in \mathbb{R}^{N\times N}$ as

$$[\mathcal{P}_\Delta(B)]_{ij} = \begin{cases} B_{ij} & (i,j) \in \Delta \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

This projection operator is to guarantee that only the reliable entries in the matrix can be projected into the space where we apply matrix completion.

Filtering Entries with Unlabeled and Uncertain Data Pairs

The purpose of the filtering step is to remove the uncertain data pairs from the manual annotations. To this end, given the m similarity matrices $\{W^k\}_{k=1}^m$ obtained from individual workers, we first compute matrix $A = [A_{ij}] \in \mathbb{R}^{N\times N}$ as the average of $\{W^k\}_{k=1}^m$, i.e.,

$$A_{ij} = \begin{cases} \sum_{k=1}^m \frac{W^k_{ij} I(W^k_{ij}\geq 0)}{\sum_{l=1}^m I(W^l_{ij}\geq 0)} & \sum_{l=1}^m I(W^l_{ij} \geq 0) > 0, \\ -1 & \text{otherwise} \end{cases}$$

where $I(z)$ is an indicator function that outputs 1 when z is true and zero, otherwise. We introduce the indicator function $I(W^k_{ij} \geq 0)$ in the above equation so that only the labeled pairs of objects will be counted in computing A.

Since $A_{ij} \in [0,1]$ for a labeled data pair (i.e. $A_{ij} \geq 0$) measures the percentage of HITs that assign objects i and j to the same cluster, it can be used as the basis for the uncertainty measure. In particular, we define the set of reliable data pairs whose labelings are agreed upon by the percentage of workers as

$$\Delta = \{(i,j) \in [N]\times[N] : A_{ij} \geq 0, A_{ij} \notin (d_0, d_1)\}$$

where $d_0 < d_1 \in [0,1]$ are two thresholds that will be determined depending on the quality of the annotations. We

[1] We can relax this requirement by estimating the number of clusters via some heuristic, by considering the number of clusters as the rank of the completed matrix A.

[2] The detailed definition of Δ would be given in the next subsection.

then construct the partially observed similarity matrix $\tilde{A}$ as follows

$$\tilde{A}_{ij} = \begin{cases} 1 & (i,j) \in \Delta, A_{ij} \geq d_1 \\ 0 & (i,j) \in \Delta, A_{ij} \leq d_0 \\ \text{unobserved} & (i,j) \notin \Delta \end{cases} \quad (2)$$

Completing the Partially Observed Matrix

The second step of the algorithm is to reconstruct the full similarity matrix $A^* \in \mathbb{R}^{N \times N}$ based on the partially observed matrix $\tilde{A}$. To this end, we need to make several reasonable assumptions about the relationship between $\tilde{A}$ and A^*.

A simple approach is to assume $\tilde{A}_{ij} = A^*_{ij}, \forall (i,j) \in \Delta$; in other words, assume that all the observed entries in matrix $\tilde{A}$ are correct. This, however, is unrealistic because $\tilde{A}$ is constructed from the partial clustering results generated by different workers, and we expect a significant amount of noise in individual clustering results. Thus, a more realistic assumption is $\tilde{A}_{ij} = A^*_{ij}$ for *most* of the observed entries in Δ. We introduce the matrix $E \in \mathbb{R}^{N \times N}$ to capture the noise in $\tilde{A}$, i.e.,

$$\mathcal{P}_\Delta(A^* + E) = \mathcal{P}_\Delta(\tilde{A}), \quad (3)$$

where $\mathcal{P}_\Delta$ is a matrix projection operator defined in (1). Under this assumption, we expect E to be a sparse matrix with most of its entries being zero.

The assumption specified in equation (3) is insufficient to recover the full similarity A^* as we can fill the unobserved entries (i.e., $(i,j) \notin \Delta$) in A^* with any values. An additional assumption is needed to make it possible to recover the full matrix from a partially observed one. To this end, we follow the theory of matrix completion (Candès and Tao 2010) by assuming the full similarity A^* to be of low rank. It was shown in (Jalali et al. 2011) that when the similarity matrix A^* is constructed from a given clustering (i.e. $A^*_{ij} = 1$ when objects i and j are assigned to the same cluster and zero, otherwise), its rank is equal to the number of clusters. As a result, when the number of clusters is relatively small compared to N, which is typically the case, it is reasonable to assume A^* to be of low rank.

Combining the two assumptions (E is sparse and A is of low rank) together leads to the following approach, to recover the full similarity matrix A^* from the partially observed matrix $\tilde{A}$. We decompose $\tilde{A}$ into the sum of two matrices E and A^*, where E is a sparse matrix that captures the noise in $\tilde{A}$ and A^* is a low rank matrix that gives the similarity between any two objects. Based on this idea, we cast the matrix recovery problem into the following optimization problem

$$\min_{A',E} \ \text{rank}(A') + C\|E\|_1 \ \text{s.t.} \ \mathcal{P}_\Delta(A' + E) = \mathcal{P}_\Delta(\tilde{A}) \quad (4)$$

where $\|X\|_1 = \sum_{ij} |X_{ij}|$ is the ℓ_1 norm of matrix X that measures the sparsity of X. Parameter $C > 0$ is introduced to balance the two objectives, i.e., finding a low rank similarity matrix A' and a sparse matrix E for noise. Section 3.3 presents an approach to automatically determine the value of C.

One problem with the objective function in (4) is that it is non-convex because $\text{rank}(\cdot)$ is a non-convex function (Candès and Tao 2010). It is therefore computationally challenging to find the optimal solution for (4). To address this challenge, we follow (Candès and Tao 2010) and replace $\text{rank}(L)$ in (4) with its convex surrogate $|A'|_*$, the trace norm of matrix A'. This allows us to relax (4) into the following convex optimization problem

$$\min_{A',E} \ |A'|_* + C\|E\|_1 \ \text{s. t.} \ \mathcal{P}_\Delta(A' + E) = \mathcal{P}_\Delta(\tilde{A}). \quad (5)$$

We use the efficient first order algorithm developed in (Lin et al. 2010) to solve the optimization problem in (5).

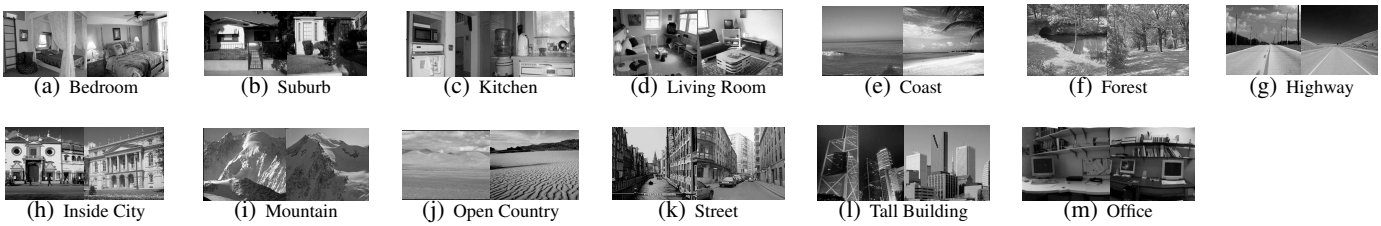
A theoretical question is whether the similarity matrix obtained by (5) is close to the true similarity matrix A^*. Our theoretical analysis gives a positive answer to this question. More specifically, under appropriate conditions about the eigenvectors of A^* (assumptions **A1** and **A2** given in the appendix), A^* can be **perfectly** recovered by (5) if the number of noisy data pairs is significantly smaller than the number of observed data pairs. More details of our theoretical analysis can be found in the appendix.

Given the completed similarity matrix A^* obtained from (5), we apply the spectral clustering algorithm (Ng, Jordan, and Weiss 2001) to compute the final data partition, which is essentially an application of k-means algorithm (MacQueen and others 1967) to the data projected into the space of the top r eigenvectors of A^*. Compared to the other kernel based clustering methods (e.g., kernel k-means), spectral clustering is more robust to the noise in the similarity matrix due to the projection of data points into the space spanned by the top eigenvectors.

Selecting Parameter Values

Parameter C in (5) plays an important role in deciding the final similarity matrix. Since no ground truth information (true cluster labels) is available to determine C, we present a heuristic for estimating the value of C.

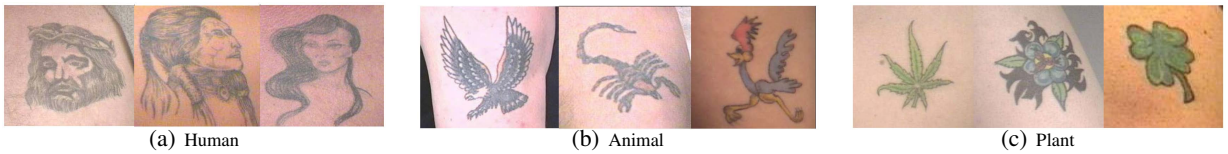
We assume that the N objects to be clustered are roughly evenly distributed across clusters; a similar assumption was adopted in normalized cut algorithm (Shi and Malik 2000). Based on this assumption, we propose to choose a value of C that leads to the most balanced distribution of objects over different clusters. To this end, we measure the imbalance of data distribution over clusters by computing $\sum_{i,j=1}^{N} A'_{i,j} = \mathbf{1}^\top A' \mathbf{1}$, where $\mathbf{1}$ is a vector of all ones. Our heuristic is to choose a value for C that minimizes $\mathbf{1}^\top A' \mathbf{1}$. The rationale behind the imbalance measurement $\mathbf{1}^\top A' \mathbf{1}$ is the following: Let $N_1, \cdots, N_r$ be the number of objects in the r clusters. Since $\mathbf{1}^\top A' \mathbf{1} = \sum_{k=1}^{r} N_k^2$ and $\sum_{k=1}^{r} N_k = N$, without any further constraints, the optimal solution that minimizes $\mathbf{1}^\top A' \mathbf{1}$ is $N_i = N/r, i = 1, \ldots, r$, the most balanced data distribution. Hence, $\mathbf{1}^\top A' \mathbf{1}$, to some degree, measures the imbalance of data distribution over clusters. The experimental results show that this heuristic works well. (Due to space limitation, we omit the comparison between different C values in the experiments section.)



(a) Bedroom (b) Suburb (c) Kitchen (d) Living Room (e) Coast (f) Forest (g) Highway

(h) Inside City (i) Mountain (j) Open Country (k) Street (l) Tall Building (m) Office

Figure 1: Some sample images from the 13 categories in the Scenes data set



(a) Human (b) Animal (c) Plant

Figure 2: Some sample images from the three categories in the Tattoo data set

Experiments

In this section, we first demonstrate empirically that the proposed algorithm can achieve similar or better clustering performance as the Bayesian approach for crowdclustering (Gomes et al. 2011) with significantly lower running time. We further show that, as we reduce the number of pairwise labels, either by reducing the number of workers, or by reducing the number of HITs performed by each worker, the proposed algorithm significantly outperforms the Bayesian approach.

Data Sets

Two image data sets are used for clustering:

- *Scenes Data Set*: This is a subset of the larger Scenes image data set (Fei-Fei and Perona 2005) which has been used in the previous study on crowdclustering (Gomes et al. 2011). It is comprised of $1,001$ images belonging to 13 categories. Figure 1 shows sample images of each category from this data set. To obtain the crowdsourced labels, 131 workers were employed to perform HITs. In each HIT, the worker was asked to group images into multiple clusters, where the number of clusters was determined by individual workers. Pairwise labels between images are derived from the partial clustering results generated in HITs. The data we used, including the subset of images and the output of HITs, were provided by the authors of (Gomes et al. 2011).
- *Tattoo Data Set*: This is a subset of the Tattoo image database (Jain, Lee, and Jin 2007). It contains $3,000$ images that are evenly distributed over three categories: human, animal and plant. Some sample images of each category in the Tattoo data set are shown in Figure 2. Unlike the Scenes data set where the objective of HIT was to group the images into clusters, the workers here were asked to annotate tatoo images with keywords of their choice. On average, each image is annotated by three different workers. Pairwise labels between images are derived by comparing the number of matched keywords between images to a threshold (which is set to 1 in our study).

Baseline and evaluation metrics

Studies in (Gomes et al. 2011) have shown that the Bayesian approach performs significantly better than the ensemble clustering algorithm (Strehl and Ghosh 2002), and Nonnegative Matrix Factorization (NMF) (Li, Ding, and Jordan 2007) in the crowdclustering setting. Hence, we use the Bayesian approach for crowdclustering as the baseline in our study.

Two metrics are used to evaluate the clustering performance. The first one is the normalized mutual information (NMI for short) (Cover and Thomas 2006). Given the ground truth partition $\mathcal{C} = \{C_1, C_2, \ldots, C_r\}$ and the partition $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_r\}$ generated by a clustering algorithm, the normalized mutual information for partitions C and C' is given by

$$NMI(C, C') = \frac{2MI(C, C')}{H(C) + H(C')},$$

where $MI(X, Y)$ represents the mutual information between the random variables X and Y, and $H(X)$ represents the Shannon entropy of random variable X.

The second metric is the pairwise F-measure (PWF for short). Let $\mathcal{A}$ be the set of data pairs that share the same class labels according to the ground truth, and let $\mathcal{B}$ be the set of data pairs that are assigned to the same cluster by a clustering algorithm. Given the pairwise precision and recall that are defined as follows

$$\text{precision} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|}, \quad \text{recall} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{B}|},$$

the pairwise F-measure is computed as the harmonic mean of precision and recall, i.e.

$$PWF = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

Both NMI and PWF values lie in the range $[0, 1]$ where a value of 1 indicates perfect match between the obtained partition by a clustering algorithm and the ground truth partition and 0 indicates completely mismatch. Besides clustering accuracy, we also evaluate the efficiency of both algorithms by measuring their running time. The code of the

Table 1: Clustering performance and running time of the proposed algorithm (i.e. matrix completion) and the baseline algorithm (i.e. Bayesian method) on two data sets

Data sets	Scenes Data Set			Tattoo Data Set		
	NMI	PWF	CPU time (seconds)	NMI	PWF	CPU time (seconds)
Matrix Completion	0.738	0.584	6.02×10^2	0.398	0.595	8.85×10^3
Bayesian Method	0.764	0.618	5.18×10^3	0.292	0.524	4.79×10^4

(a) Highway and Inside city (b) Bedroom and Kitchen (c) Mountain and Open country (d) Tall building and Street

Figure 3: Sample image pairs that are grouped into the same cluster by more than 50% of the workers but are assigned to different clusters according to the ground truth.

Table 2: Performance of the proposed clustering algorithm as a function of different threshold values and the percentage of 1 entries in the matrix $\tilde{A}$ that are consistent with the cluster assignments for the Scenes data set

Threshold d_1	0.1	0.3	0.5	0.7	0.9
Consistency percentage	18.02%	28.10%	35.53%	43.94%	61.79%
NMI	0.507	0.646	0.678	0.700	0.738
PWF	0.327	0.412	0.431	0.445	0.584

baseline algorithm was provided by the authors of (Gomes et al. 2011). Both the baseline algorithm and the proposed algorithm were implemented in MATLAB and run on an Intel Xeon 2.40 GHz processor with 64.0 GB of main memory.

Experimental results with full annotations

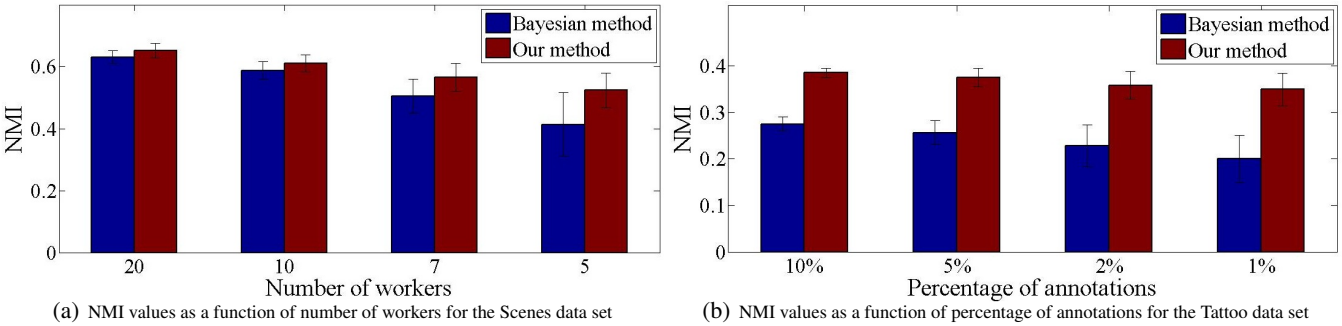
To evaluate the clustering performance of the proposed algorithm, our first experiment is performed on the Scenes and Tattoo data sets using all the pairwise labels derived from the manual annotation process. For both data sets, we set d_0 to 0. We set d_1 to 0.9 and 0.5 for the Scenes and Tattoo data sets, respectively. Two criteria are deployed in determining the value for d_1: (i) d_1 should be large enough to ensure that most of the selected pairwise labels are consistent with the cluster assignments, and (ii) it should be small enough to obtain sufficiently large number of entries with value 1 in the partially observed matrix $\tilde{A}$. Table 1 summarizes the clustering performance and running time (CPU time) of both algorithms.

We observed that for the Scenes data set, the proposed algorithm yields similar, though slighlty lower, performance as the Bayesian crowdclustering algorithm but with significantly lower running time. For the Tattoo data set, the proposed algorithm outperforms the Bayesian crowdclustering algorithm in both accuracy and efficiency. The higher efficiency of the proposed algorithm is due to the fact that the proposed algorithm uses only a subset of reliable pairwise labels while the Bayesian crowdclustering algorithm needs to explore all the pairwise labels derived from manual annotation. For example, for the Scenes data set, less than 13% of image pairs satisfy the specified condition of "reliable pairs". The small percentage of reliable pairs results in a rather sparse matrix $\tilde{A}$, and consequently a high efficiency in solving the matrix completion problem in (4). The discrepancy in the clustering accuracy between the two data sets can be attributed to the fact that many more manual annotations are provided for the Scene dataset than for the Tattoo data set. As will be shown later, the proposed algorithm is more effective than the Bayesian method with a reduced number of annotations.

We also examine how well the conditions specified in our theoretical analysis (see Appendix) are satisfied for the two image data sets. The most important condition used in our analysis is that a majority of the reliable pairwise labels derived from manual annotation should be consistent with the cluster assignments (i.e. $m_1 - m_0 \geq O(N \log^2 N)$). We found that for the Scenes data set, 95% of the reliable pairwise labels identified by the proposed algorithm are consistent with the cluster assignments, and for the Tattoo data set, this percentage is 71%.

We finally evaluate the significance of the filtering step for the proposed algorithm. First, we observe that a large portion of pairwise labels derived from the manual annotation process are inconsistent with the cluster assignment. In particular, more than 80% of pairwise labels are inconsistent with the cluster assignment for the Scenes data set. Figure 2 shows some example image pairs that are grouped into the same cluster by more than 50% of the workers but belong to different clusters according to the ground truth.

To observe how the noisy labels affect the proposed algorithm, we fix the threshold d_0 to be 0, and vary the threshold d_1 used to determine the reliable pairwise labels from 0.1



(a) NMI values as a function of number of workers for the Scenes data set

(b) NMI values as a function of percentage of annotations for the Tattoo data set

Figure 4: NMI values as a function of number of workers and percentage of annotations for two data sets

to 0.9. Table 2 summarizes the clustering performance of the proposed algorithm for the Scenes data set with different values of d_1 and the percentage of resulting reliable pairwise labels that are consistent with the cluster assignments. Overall, we observe that the higher the percentage of consistent pairwise labels, the better the clustering performance.

Experimental results with sampled annotations

The objective of the second experiment is to verify that the proposed algorithm is able to obtain an accurate clustering result even with a significantly smaller number of manual annotations. To this end, we use two different methods to sample the annotations: for the Scenes data set, we use the annotations provided by 20, 10, 7 and 5 randomly sampled workers, and for the Tattoo data set, we randomly sample $10\%, 5\%$, 2% and 1% of all the annotations. Recall that for the Tattoo data set we have only 3 annotators per image. Then we run both the baseline and the proposed algorithm on the sampled annotations. All the experiments in this study are repeated five times, and the performance averaged over the five trials is reported in Figure 4 (due to space limitation, we only report the NMI values).

As expected, reducing the number of annotations deteriorates the clustering performance for both the algorithms. However, the proposed algorithm appears to be more robust and performs better than the baseline algorithm for all levels of random sampling. The robustness of the proposed algorithm can be attributed to the fact that according to our analysis, to perfectly recover the cluster assignment matrix, the proposed algorithm only requires a small number of reliable pairwise labels (i.e. $O(N \log^2 / N)$). In contrast, the Bayesian crowdclustering algorithm requires a large number of manual annotations to overcome the noisy labels and to make a reliable inference about the hidden factors used by different workers to group the images. As a consequence, we observe a significant reduction in the clustering performance of the Bayesian approach as the number of manual annotations is decreased.

Conclusion and Discussion

We have presented a matrix completion framework for crowdclustering. The key to the proposed algorithm is to identify a subset of data pairs with reliable pairwise labels provided by different workers. These reliable data pairs are used as the seed for a matrix completion algorithm to derive the full similarity matrix, which forms the foundation for data clustering. Currently, we identify these reliable data pairs based on the disagreement among workers, and as a result, a sufficient number of workers are needed to determine which data pairs are reliable. An alternative approach is to improve the quality of manual annotations. Given that our matrix completion approach, needs only a small number of high quality labels, we believe that combining appropriately designed incentive mechanisms with our matrix completion algorithm will lead to greatly improved performance. In (Shaw, Horton, and Chen 2011), the authors discussed different incentive mechanisms to improve the quality of work submitted via HITs. In particular, they studied a number of incentive mechanisms and their affect on eliciting high quality work on Turk. They find that a mechanism based on accurately reporting peers' responses is the most effective in improving the performance of Turkers. As part of our future work, we plan to investigate the conjunction of appropriate incentive mechanisms with clustering algorithms for this problem. Another direction for our future work is to combine the pairwise similarities obtained by HITs and the object attributes.

Acknowledgement:

This research was supported by ONR grant no. N00014-11-1-0100 and N00014-12-1-0522. Also, we would like to thank Ryan Gomes and Prof. Pietro Perona for providing us the code of their algorithm, the specific subset of images of the Scenes data set they used and the outputs of the HITs.

References

Candès, E. J., and Tao, T. 2010. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory* 56(5):2053–2080.

Chandrasekaran, V.; Sanghavi, S.; Parrilo, P. A.; and Willsky, A. S. 2011. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization* 21(2):572–596.

Cover, T. M., and Thomas, J. A. 2006. *Elements of Information Theory (2nd ed.)*. Wiley.

Fei-Fei, L., and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition, volume 2, 524–531.

Fred, A. L. N., and Jain, A. K. 2002. Data clustering using evidence accumulation. In *International Conference on Pattern Recognition*, volume 4, 276–280.

Gomes, R.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. In *NIPS*.

Jain, A. K.; Lee, J.-E.; and Jin, R. 2007. Tattoo-ID: Automatic tattoo image retrieval for suspect and victim identification. In *Advances in Multimedia Information Processing–PCM*, 256–265.

Jalali, A.; Chen, Y.; Sanghavi, S.; and Xu, H. 2011. Clustering partially observed graphs via convex optimization. In *ICML*, 1001–1008.

Karger, D.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *NIPS*.

Li, T.; Ding, C. H. Q.; and Jordan, M. I. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Seventh IEEE International Conference on Data Mining*, 577–582.

Lin, Z.; Chen, M.; Wu, L.; and Ma, Y. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Arxiv preprint arXiv:1009.5055*.

MacQueen, J., et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 14. California, USA.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Raykar, V., and Yu, S. 2011. Ranking annotators for crowdsourced labeling tasks. In *NIPS*.

Shaw, A. D.; Horton, J. J.; and Chen, D. L. 2011. Designing incentives for inexpert human raters. In *ACM Conference on Computer Supported Cooperative Work* (2011), 275–284.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(8):888–905.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *JMLR* 3:583–617.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. 2011. Adaptively learning the crowd kernel. In Getoor, L., and Scheffer, T., eds., *ICML*, 673–680. Omnipress.

Vijayanarasimhan, S., and Grauman, K. 2011. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 1449–1456.

Wauthier, F., and Jordan, M. 2011. Bayesian bias mitigation for crowdsourcing. In *NIPS*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Yan, Y.; Rosales, R.; Fung, G.; and Dy, J. 2011. Active learning from crowds. In *Proceedings of the Int. Conf. on Machine Learning (ICML)*.

Appendix A: Theoretical Analysis for Perfect Recovery using Eq. 5

First, we need to make a few assumptions about A^* besides being of low rank. Let A^* be a low-rank matrix of rank r, with a singular value decompsition $A^* = U\Sigma V^\top$, where $U = (\mathbf{u}_1, \ldots, \mathbf{u}_r) \in \mathbb{R}^{N\times r}$ and $V = (\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathbb{R}^{N\times r}$ are the left and right eigenvectors of A^*, satisfying the following incoherence assumptions.

- **A1** The row and column spaces of A^* have coherence bounded above by some positive number μ_0, i.e.,
  $$\max_{i\in[N]} \|P_U(\mathbf{e}_i)\|_2^2 \le \frac{\mu_0 r}{N}, \quad \max_{i\in[N]} \|P_V(\mathbf{e}_i)\|_2^2 \le \frac{\mu_0 r}{N}$$
  where $\mathbf{e}_i$ is the standard basis vector.
- **A2** The matrix $E = UV^\top$ has a maximum entry bounded by $\frac{\mu_1\sqrt{r}}{N}$ in absolute value for some positive μ_1, i.e. $|E_{i,j}| \le \frac{\mu_1\sqrt{r}}{N}, \forall (i,j) \in [N]\times[N]$,

where P_U and P_V denote the orthogonal projections on the column space and row space of A^*, respectively, i.e.

$$P_U = UU^\top, \quad P_V = VV^\top$$

To state our theorem, we need to introduce a few notations. Let $\xi(A')$ and $\mu(A')$ denote the low-rank and sparsity incoherence of matrix A' defined by (Chandrasekaran et al. 2011), i.e.

$$\xi(A') = \max_{E\in T(A'), \|E\|\le 1} \|E\|_\infty \tag{6}$$

$$\mu(A') = \max_{E\in \Omega(A'), \|E\|_\infty \le 1} \|E\| \tag{7}$$

where $T(A')$ denotes the space spanned by the elements of the form $\mathbf{u}_k\mathbf{y}^\top$ and $\mathbf{x}\mathbf{v}_k^\top$, for $1 \le k \le r$, $\Omega(A')$ denotes the space of matrices that have the same support to A', $\|\cdot\|$ denotes the spectral norm and $\|\cdot\|_\infty$ denotes the largest entry in magnitude.

Theorem 1. *Let $A^* \in \mathbb{R}^{N\times N}$ be a similarity matrix of rank r obeying the incoherence properties (**A1**) and (**A2**), with $\mu = \max(\mu_0, \mu_1)$. Suppose we observe m_1 entries of A^* recorded in $\tilde{A}$ with locations sampled uniformly at random, denoted by $\mathcal{S}$. Under the assumption that m_0 entries randomly sampled from m_1 observed entries are corrupted, denoted by Ω, i.e. $A^*_{ij} \ne \tilde{A}_{ij}, (i,j) \in \Omega$. Given $\mathcal{P}_\mathcal{S}(\tilde{A}) = \mathcal{P}_\mathcal{S}(A^* + E^*)$, where E^* corresponds to the corrupted entries in Ω. With*

$$\mu(E^*)\xi(A^*) \le \frac{1}{4r+5}, \quad m_1 - m_0 \ge C_1\mu^4 n(\log n)^2,$$

and C_1 is a constant, we have, with a probability at least $1 - N^{-3}$, the solution $(A', E) = (A^*, E^*)$ is the unique optimizer to (5) provided that

$$\frac{\xi(A^*) - (2r-1)\xi^2(A^*)\mu(E^*)}{1 - 2(r+1)\xi(A^*)\mu(E^*)} < \lambda < \frac{1 - (4r+5)\xi(A^*)\mu(E^*)}{(r+2)\mu(E^*)}$$

We skip the proof of Theorem 1 due to space limitation. As indicated by Theorem 1, the full similarity matrix A^* can be recovered if the number of observed correct entries (i.e., m_1) is significantly larger than the number of observed noisy entries (i.e., m_0).