Clustering by Comparison: Stochastic Block Model for Inference in Crowdsourcing

Ramya Korlakai Vinayak RAMYA@CALTECH.EDU
Babak Hassibi HASSIBI@SYSTEMS.CALTECH.EDU
California Institute of Technology, Pasadena, CA, USA.

Abstract

The focus of this article is clustering objects by crowdsourcing pairwise comparison tasks. We propose graph clustering as an inference tool for this task. We interpret the Stochastic Block Model, which is a popular generative model for graph clustering in this context and make connections to the Dawid-Skene Model that is widely used in inference for crowdsourced tasks.

Looking at models and algorithms in graph clustering from the point of view of inference from crowdsourced comparison tasks gives rise to several interesting questions. If we are clustering n objects, making all $\binom{n}{2}$ comparisons is too expensive. Instead we want to cluster the objects with only a subset of pairwise comparisons. These comparisons when performed by non-experts gives rise to noisy answers. One way to overcome the noise is to make multiple queries for same the comparison. However, the budget and time puts a constraint on the number of questions we can ask. It is interesting to investigate whether one is better off collecting a larger number of noisy answers compared to fewer but more reliable answers.

Another interesting question that arises is whether we need complicated models to get high accuracy in clustering by pairwise comparisons? From experiments on real data and simulations, we observe that Stochastic Block Model though simplistic in its approach towards crowdsourcing, performs well in clustering data using pairwise comparisons.

1. Introduction

Supervised learning tasks need labeled data sets for training and testing. Creating such a dataset by having experts on the respective field label the collected data is both time consuming and expensive. In (Raykar et al., 2010; Sorokin & Forsyth, 2008; Vijayanarasimhan & Grauman, 2014; Von Ahn et al., 2008; Welinder et al., 2010) it is shown that crowdsourcing can be a good source to collect labels for data. Apart from generating labeled data, clustering objects from crowdsourced pairwise comparisons has also been used as a tool to learn similarity mapping (Yi et al., 2012).

Consider the specific example of labeling a set of images of dogs of different breeds. One could show a set of images to each worker on a crowdsourcing platform, and ask him/her to identify the breed of dog in each of those images. But such a task would require the workers to be experts in identifying the dog breeds. A more reasonable task is to ask the workers to compare pairs of images, and for each pair, answer whether they think the dogs in the images are of the same breed or not. Then we can cluster images based on the aggregate responses such that the images of dogs of the same breed are in the same cluster. Given n images, there are $\binom{n}{2}$ distinct pairs of images, and it is hopeless to compare all possible pairs. So, we need to cluster the data with partial comparisons.

Clustering broadly refers to the process of grouping together the data points or objects that are similar to each other (Jain et al., 1999). It is a powerful tool for pattern recognition and is widely used in various applications like data mining (Ester et al., 1995; Xu et al., 1999), social networks (Domingos & Richardson, 2001; Fortunato, 2010; Mishra et al.), bioinformatics (Xu et al., 2002; Yang & Lonardi, 2005) and other machine learning tasks. The type of clustering algorithm to be used depends on the data and the application. In a wide variety of applications, the data is in the form of graphs. In these scenarios, one is interested in identifying group of nodes in the graph that are more connected to each other than to the nodes outside the

This is a part of workshops in ICML 2015 Copyright 2015 by the author(s).

group.

In this article, we look at graph clustering as an inference tool for clustering images from crowdsourced pairwise comparison tasks. We focus on the image labeling task experiment from our paper (Vinayak et al., 2014b) in this light. We interpret the Stochastic Block Model in the context of inference in Crowdsourcing (Section 2). Also we compare the clustering results from k-means, Spectral Clustering with the output from these clustering algorithm with preprocessed input (where convex algorithms are used for preprocessing the data) (Section 4).

In (Gomes et al., 2011) Bayesian estimation is used to infer clusters from pairwise comparisons. Non-expert workers are modeled as linear classifiers and the algorithm estimates the clusters as well as the linear estimators. However such models are complicated with large number of hyperparameters to be tuned. Furthermore, the exact model would depend on the type of objects being compared. Unlike such complicated models, the Stochastic Block Model is very simple in its approach: applying graph clustering algorithms like spectral clustering and convex algorithms based on low-rank + sparse decomposition on the pairwise comparison graph does not require the knowledge of model parameters and does not depend on the type of objects being compared. From our experiments (Section 4) we observe that clustering algorithms like k-means and spectral clustering when combined with preprocessing by convex programs (Section 3.3), perform very well. In the future, we want to compare inference from complex Bayesian models, maximum likelihood estimation from popular crowdsourcing models with graph clustering algorithms.

2. Comparing Models for Clustering and Inference in Crowdsourcing

2.1. Dawid-Skene Model

Dawid-Skene model (Dawid & Skene, 1979) is one of the most popular models for answers collected on crowdsourced tasks. Consider a task with the binary answer like the pair-wise comparison task. In this model, each worker j is capable of answering any given question correctly with probability μ_j. This model does not account for the difficulty of the question itself.

2.2. Two-coin Dawid-Skene Model

In the two-coin Dawid-Skene Model, the ability of the worker to give correct answer depends on the true answer. For a binary $0/1$ classification task, each worker j is associated with a 2×2 confusion matrix,

$$\tilde{\mathbf{C}}_j = \begin{bmatrix} \mu_{j,0} & 1-\mu_{j,0} \\ 1-\mu_{j,1} & \mu_{j,1} \end{bmatrix} \tag{2.1}$$

where $\mu_{j,0}$ is the probability that worker j correctly answers the question when the true answer is 0, and $\mu_{j,1}$ is the probability that he/she answers correctly when the true answer is 1.

2.3. Stochastic Block Model

Stochastic Block Model (SBM) (Condon & Karp, 2001; Holland et al., 1983) is one of the most widely used model for graph clustering. It is an extension of random graphs. In a random graph on n nodes, any two nodes are linked independently with probability p. Consider a graph on n nodes and K clusters. Any two nodes in this graph are connected with probability p_i if both the nodes are in cluster i, else they are connected with probability q. Let $\mathbf{A} = \mathbf{A}^T$ be the adjacency matrix of a graph on n nodes with K disjoint clusters of size n_i each, $i = 1, 2, \cdots, K$. Let $1 \geq p_i \geq 0$, $i = 1, \cdots, K$ and $1 \geq q \geq 0$. For $l > m$, $\mathbf{A}_{l,m} = 1$ with probability p_i if both l and m are in the same cluster i, and $\mathbf{A}_{l,m} = 1$ with probability q if l and m are in different clusters. If $p_i > q$ for each $i = 1, \cdots, K$, then we expect the density of edges to be higher inside the clusters compared to outside.

2.4. Stochastic Block Model for Inference in Crowdsourcing

For the problem of clustering images by pairwise comparisons, we can look at the data aggregate data collected as a partially filled adjacency matrix. In this scenario, we can use Stochastic Block Model as generative model. SBM can be more specific in terms of distinguishing the classes when the true answer is 1. The answer given by a worker depends not only on if the true answer is 1 or 0, but when the true answer is 1 what class the images belong to. That is:

- If the true answer is 0, then $\mu_{j,0} = q$ for all workers j.
- If the true answer is 1, and the images being compared are from class i, then $\mu_{j,1} = p_i$ for all workers j.

So when $p_i > q$, it means that workers are more likely to identify similar images and dissimilar images correctly.

Stochastic Block Model with a common edge probability p for all clusters as applied to this would give same confusion matrix to all workers:

$$\tilde{\mathbf{C}}_j = \begin{bmatrix} 1-q & q \\ 1-p & p \end{bmatrix} \tag{2.2}$$

The Stochastic Block Model is more general than the two-coin Dawid-Skene model with respect to distinguishing questions, but is restrictive in terms of distinguishing worker abilities as it assumes all workers have same ability.

Definition 2.1 (Partial Observation Model for SBM) *Let $\mathbf{A} = \mathbf{A}^T$ be the adjacency matrix of a random graph generated according to the Stochastic Block Model. Let $0 < r \leq 1$. Each entry of the adjacency matrix $\mathbf{A}$ is observed independently with probability r. Let $\mathbf{A}^{obs}$ denote the observed adjacency matrix. Then for $l > m$: If both nodes l and m are in the same cluster i,*

$$\mathbf{A}^{obs}_{l,m} = \begin{cases} 1 \text{ w.p. } rp_i, \\ 0 \text{ w.p. } r(1-p_i), \\ * \text{ w.p. } 1-r. \end{cases} \tag{2.3}$$

If the nodes l and m are not in the same cluster i,

$$\mathbf{A}^{obs}_{l,m} = \begin{cases} 1 \text{ w.p. } rq \\ 0 \text{ w.p. } r(1-q) \\ * \text{ w.p. } 1-r, \end{cases} \tag{2.4}$$

where $*$ denotes unknown.

3. Clusteirng

3.1. k-means Clusteirng

Algorithm 1 K-Means Algorithm

```
Input: Data points: ${\mathbf{x}_1, \ldots, \mathbf{x}_n}$ Number of clusters: $K$
Output: Cluster allocations: ${y_1, \ldots, y_n}$
Initialize cluster centers ${\mathbf{c}_1^{(0)}, \ldots, \mathbf{c}_K^{(0)}}$ randomly from data points
$t = 1$
repeat
    /* Cluster Allocation */
    for $i = 1, \ldots, n$ do
        $y_i = \underset{j}{argmin} ||\mathbf{x}_i - \mathbf{c}_j^{(t-1)}||$
    end
    /* Update Cluster Centers */
    for $j = 1, \ldots, K$ do
        $\mathbf{c}_j^{(t)} = \frac{1}{|\{i: y_i = j\}|} \sum_{i: y_i = j} \mathbf{x}_i$
    end
    $t = t + 1$
until no change in cluster allocation;
```

K-means it one of the most popular and widely used clustering algorithm. Given a set of data points in a metric space, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, k-means algorithm finds k cluster centers which are average of the nodes in the respective clusters. Each node in a cluster is closer to it cluster center compared to other cluster centers. A greedy algorithm to do this is Algorithm 1

3.2. Spectral Clustering

Algorithm 2 Spectral Clustering (Ng et al., 2001)

```
Input: Adjacency Matrix: $\mathbf{A}$ Number of clusters: $K$
Output: Cluster allocations: ${y_1, \ldots, y_n}$
Compute the modified Laplacian $\mathcal{L}$
Compute Top $K$ eigenvectors $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$ of $\mathcal{L}$
Normalize $\mathbf{U}$ such that row norm = 1
Run K-means Algorithm on the rows of normalized $\mathbf{U}$ as data and $K$ as number of clusters to be found
```

Spectral clustering is another popular clustering algorithm. Given adjacency matrix of a graph, define modified graph Laplacian as follows,

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, \tag{3.1}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{A}_{ij}$. Spectral Clustering uses the top-k eigenvectors of $\mathcal{L}$ to cluster the nodes in the graph. There are various versions of spectral clustering algorithm. We use Algorithm 2 from (Ng et al., 2001).

3.3. Convex Algorithms

In this section we review convex algorithms for clustering based on low-rank plus sparse decomposition of the adjacency matrix. In the case of unweighted graphs, an ideal clustered graph is a union of disjoint cliques. Given the adjacency matrix of an unweighted graph with clusters (denser connectivity inside the clusters compared to outside), we can interpret it as an ideal clustered graph with missing edges inside the clusters and erroneous edges in between clusters. Recovering the low-rank matrix corresponding to the disjoint cliques is equivalent to finding the clusters.

The idea of using convex optimization for clustering has been proposed in (Ailon et al., 2013; Ames, 2013; Ames & Vavasis, 2011; 2014; Chen et al., 2012; 2013; Jalali et al., 2011; Oymak & Hassibi, 2011; Vinayak et al., 2014a;b; Xu et al., 2010). While each of these works differ in certain ways, the common approach they use for clustering is inspired by recent work on low-rank matrix recovery and completion via regularized nuclear norm (trace norm) minimization (Candes & Recht, 2009; Candes & Romberg, 2006; Candès et al., 2011; Chandrasekaran et al., 2011; 2012).

Consider the following convex program for regularized nuclear norm minimization (based on robust PCA):

Simple Convex Program: (Vinayak et al., 2014b)

$$\begin{aligned} &\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \;\; \|\mathbf{L}\|_\star + \lambda\|\mathbf{S}\|_1 \\ &\text{subject to} \\ &\quad 1 \geq \mathbf{L}_{i,j} \geq 0 \;\text{ for all } i,j \in \{1,2,\dots n\} \\ &\quad \mathbf{L}^{obs} + \mathbf{S}^{obs} = \mathbf{A}^{obs} \end{aligned} \tag{3.2}$$

where $\lambda \geq 0$ is the regularization parameter, $\|.\|_\star$ is the nuclear norm (sum of the singular values of the matrix), and $\|.\|_1$ is the l_1-norm (sum of absolute values of the entries of the matrix). $\mathbf{S}$ is the sparse error matrix that accounts for the missing edges inside the clusters and erroneous edges outside the clusters on the observed entries. $\mathbf{L}^{obs}$ and $\mathbf{S}^{obs}$ denote entries of $\mathbf{L}$ and $\mathbf{S}$ that correspond to the observed part of the adjacency matrix.

Program 3.2 is very simple and intuitive. Further, it does not require any information other than the observed part of the adjacency matrix.

It is not difficult to see that, when the edge probability inside the cluster is $p < 1/2$, that (as $n \to \infty$) Program 3.2 will return $\mathbf{L}^0 = 0$ as the optimal solution (since if the cluster is not dense enough it is more costly to complete the missing edges). To overcome this hiccup, consider the following modification:

Improved Convex Program: (Vinayak et al., 2014b)

$$\begin{aligned} &\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \;\; \|\mathbf{L}\|_\star + \lambda\|\mathbf{S}\|_1 \\ &\text{subject to} \\ &\quad 1 \geq \mathbf{L}_{i,j} \geq \mathbf{S}_{i,j} \geq 0 \;\text{ for all } i,j \in \{1,2,\dots n\} \\ &\quad \mathbf{L}_{i,j} = \mathbf{S}_{i,j} \text{ whenever } \mathbf{A}^{obs}_{i,j} = 0 \\ &\quad \text{sum}(\mathbf{L}) \geq |\mathcal{R}| \end{aligned} \tag{3.3}$$

As before, $\mathbf{L}$ is the low-rank matrix corresponding to the ideal cluster structure and $\lambda \geq 0$ is the regularization parameter. However, $\mathbf{S}$ is now the sparse error matrix that accounts only for the missing edges inside the clusters on the observed part of adjacency matrix.

As before, $\mathbf{L}$ is the low-rank matrix corresponding to the ideal cluster structure and $\lambda \geq 0$ is the regularization parameter. However, $\mathbf{S}$ is now the sparse error matrix that accounts only for the missing edges inside the clusters on the observed part of adjacency matrix.

If $\mathcal{R}$ is not known, it is possible to solve Problem 3.3 for several values of $\mathcal{R}$ until the desired performance is obtained. Our empirical results reported in Section 4.1, suggest that the solution is not very sensitive to the choice of $\mathcal{R}$.

Table 1. Empirical Parameters from the real data. (Section 4.1)

Params	Value	Params	Value
n	473	r	0.1500
K	3	q	0.1929
n_1	172	p_1	0.7587
n_2	151	p_2	0.6444
n_3	150	p_3	0.7687

4. Experiments

4.1. Clustering Images: Amazon MTurk Experiment

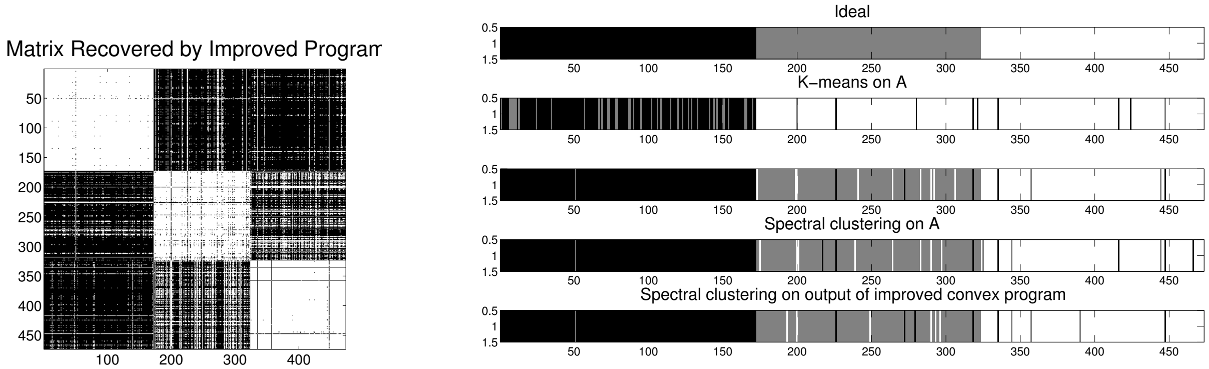
Let us revisit our example of clustering dogs of different breeds from introduction. Here we are interested in creating a clustered dataset that can be used for training supervised models. Instead of requiring an expert to label images, we will put a simpler task of comparing pairs of images on crowdsourcing platform and get a partial comparison graph from non-experts. Our goal is to recover the underlying clustering of the images from this noisy and incomplete comparison graph.

Image Data Set: We used images of 3 different breeds of dogs : Norfolk Terrier (172 images), Toy Poodle (151 images) and Bouvier des Flandres (150 images) from the Standford Dogs Dataset (Khosla et al., 2011). We uploaded all the 473 images of dogs on an image hosting server (we used imgur.com). A sample of the images is in Figure 3.

MTurk Task: We used Amazon Mechanical Turk (Buhrmester et al., 2011) as the platform for crowdsourcing. For each worker who accepted the task, we showed 30 pairs of images chosen randomly from the $\binom{n}{2}$ possible pairs. The task assigned to the worker was to compare each pair of images, and answer whether they think the dogs belong to the same breed or not. If the worker's response is a "yes", then there we fill the entry of the adjacency matrix corresponding to the pair as 1, and 0 if the answer is a "no".

Collected Data: We recorded around 608 responses. We were able to fill $16,750$ out of $111,628$ entries in $\mathbf{A}$. That is, we observed 15% of the total number of entries. Compared with true answers (which we know a priori), the answers given by the workers had around 23.53% errors (3941 out of 16750). The empirical parameters for the Stochastic Block Model with partial observations from data obtained is shown Table 1.

We ran Program 3.3 with regularization parameter, $\lambda = 1/\sqrt{n}$ and the size of the cluster region, $\mathcal{R} = 0.125\binom{n}{2}$. We did not notice much difference in the solution when we varied $\mathcal{R}$. As long as it is not too small or too large, the program performs reasonably well. Figure 1a shows the recovered matrix. Entries with value 1 are depicted by white and 0 is depicted by black. In Figure 1b we compare the



(a)

(b) Comparing k-means and spectral clustering.

Figure 1. (a) Result of using Program 3.3 on the real data set. (b) Comparing the clustering output of running k-means and spectral clustering directly on A (with unknown entries set to 0) and on the rounded output of Program 3.3 (Section 4.1).

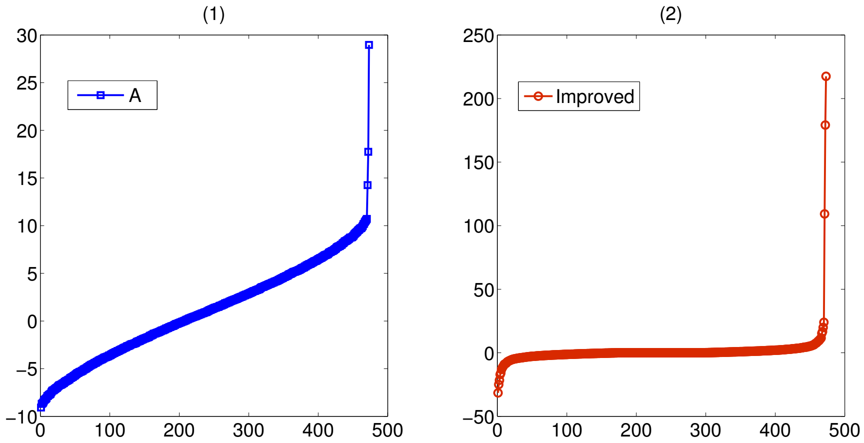


Figure 2. Plot of sorted eigen values for (1) Adjacency matrix with unknown entries filled by 0, (2) Recovered adjacency matrix from Program 3.3

clusters output by running the k-means and spectral clustering algorithms directly on the adjacency matrix $\mathbf{A}$ (with unknown entries set to 0) to that obtained by running them on the rounded matrix recovered after running Program 3.3. The overall error with k-means was 40.8% whereas the error drops to 7.19% we used the matrix recovered from Programs 3.3. Spectral Clustering performs even better with error of 3.38% when run on rounded output of Program 3.3 (see Table 2). Further, note that for running the k-means and spectral clustering algorithms we need to know the exact number of clusters. A common heuristic is to identify the top K eigen values that are much larger than the rest. In Figure 2 we plot the sorted Eigen values for the adjacency matrix $\mathbf{A}$ and the recovered matrx. For the matrix recovered after running Program 3.3, we can see that the top 3 eigen values are very easily distinguished from the rest which fall flat much more drastically compared to $\mathbf{A}$.

From the empirical parameters computed on the collected data (Table 1), we observe that on average workers found it difficult to cluster the toy poodles as one class compared to the other breeds. A sample of the data is shown in Figure 3. Norfolk Terrier and Bouvier des Flandres seem to have similar colors, whereas the Toy Poodle could be of different colors. Note that factors such as color, grooming, posture, face visibility etc. can result in confusion while comparing image pairs. Also, note that the ability of the workers to distinguish the dog breeds is neither guaranteed nor uniform. Thus, the edge probability inside and outside clusters are not uniform. Nonetheless, k-means and spectral clustering on the rounded output of Program 3.3, are quite successful in clustering the data with only 15% observations.

Figure 3. Sample images of three breeds of dogs that were used in the MTurk experiment. (Section 4.1)

Table 2. Number of miss-classified images for the real data. (Section 4.1)

Clusters→	#1	#2	#3	Total	Success %
K-means on A	39	150	4	193	59.20
K-means on output of 3.3	1	29	4	34	92.81
Spectral Clustering on A	1	11	7	19	95.98
Spectral Clustering on output of 3.3	1	10	5	16	**96.62**

4.2. Simulations

For all the simulations we generate adjacency matrix from according to the Partially Observed Stochastic Block Model defined in Definition 2.1.

4.2.1. Different cluster sizes

Consider a graph on $n = 1000$ nodes, with 4 clusters of sizes $[150, 200, 300, 350]$. Edge probability inside clusters is $p = 0.7$, edge probability between clusters is $q = 0.3$ and observation probability is $r = 0.15$. We run k-means and spectral clustering directly on the observed adjacency matrix by setting unknown entries to zero, and on the output of the Improved Convex algorithm. The average number of misclassified images (average over 10 experiments) are shown in the Table 3. We observe that the smallest cluster is the hardest to recover (Figure 5). Note that both K-means and spectral clustering need to know the number of clusters. A heuristic to pick K is to look at the top eigen values of the adjacency matrix. If the graph has K clusters, the top K eigen values are separated from the rest. As clusters get smaller, the gap between K-th and $K+1$-th eigen value decreases and hence it becomes hard to distinguish it. Even though running spectral clustering on the observed adjacency matrix directly does fairly well, we input number of clusters as 4. If we look at the eigen value plot in Figure 4, we see that only top 3 eigen values are distinguishable from the rest. So, without the prior knowledge that the number of clusters is 4, spectral clustering on the observed matrix directly would have merged the small cluster with one of the remaining clusters giving 3 clusters. However, if we look at the eigen values of the rounded output of the improved convex program, the 4-th largest eigen values is clearly resolved from the rest. Thus, there is an advantage in preprocessing the data.

4.2.2. Different edge densities

Recall that in the Stochastic Block Model the answer for a pair-wise comparison depends only on the cluster to which the images belong. However in reality it is affected by the difficulty of the images themselves. For example, some images might be blurry making it hard for any worker to answer well regardless of which cluster the images belong to. In order to check how robust clustering algorithms are in such case, we consider the following simulation. Consider a grpah on $n = 1000$ nodes with 4 clusters of equal size 250. Observation probability is $r = 0.15$. Edge probability inside the clusters is iid $\mathcal{N}(p, \sigma^2)$ with $p = 0.55$. Edge probability between the clusters is iid $\mathcal{N}(q, \sigma^2)$, with $q = 0.30$. Standard deviation $\sigma = 0.25$. Hence even if two images being compared are from the same class, the edge probability can vary. Further note that p and q are just one standard deviation apart. Table 4 shows the average number of misclassified images (average over 10 experiments). We see that clustering algorithms, especially k-means and Spectral clustering on the rounded output of 3.3 performs very well with 98.61% and 98.55% success respectively.

4.2.3. Comparison of majority Vote versus noisy observations

The answers obtained by crowdsourcing are noisy as the workers are non-experts. This noise can be reduced by taking multiple independent measurements for each comparison and taking the majority vote. However, this improvement in quality of answers comes the cost of increase in the number of queries. If we have a budget of M queries, we can obtain M noisy pair-wise comparison or obtain

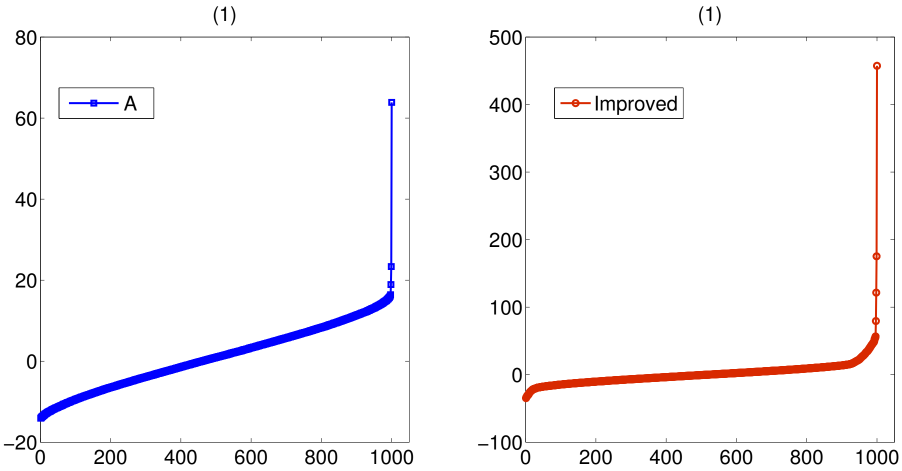


Figure 4. Plot of sorted eigen values of (1) the adjacency matrix with unobserved entries set to zero and that of (2) rounded output of improved convex program. Synthetic graph on $n = 1000$ nodes with 4 different clusters of sizes $[150, 200, 300, 350]$ with $p = 0.7$, $q = 0.3$ and $r = 0.15$ (Section 4.2.1).

Table 3. Average number (over 10 experiments) of miss-classified images for simulation on 1000 node graph with 4 clusters of different sizes (Section 4.2.1).

Clusters→	#1	#2	#3	#4	Total	Success %
K-means on A	76.20	8.80	39.90	24.60	149.50	85.05
K-means on output of 3.3	7.9	21.1	5.0	3.50	37.50	96.25
Spectral Clustering on A	9.20	12.50	16.20	15.90	53.80	94.62
Spectral Clustering on output of 3.3	8.10	6.80	5.20	5.90	26.00	**97.40**

$\lfloor M/T \rfloor$ pair-wise comparisons that are of better quality by making T independent repetitions.

In this experiment we want to compare fewer but less noisy measurements versus more measurements which are noisy. Consider a graph on $n = 1000$ nodes with $K = 4$ clusters of equal sizes. Observation probability $r = 0.18$. So we can ask 18% of the possible $\binom{1000}{2}$ edges comparison questions. Edge probability inside clusters is $p = 0.7$ and between clusters is $q = 0.3$. We compare the clustering results for the following two cases: (a) Different pairs are compared in each query; (b) Each query is made 3 times and a majority vote is taken. Table 5 shows the average (over 10 experiments) number of misclassified images. For the given set of parameters, the simulations suggest that making more measurements that are noisy is better than fewer measurements that are of better quality. In the future work, we would like to investigate this for different set of parameters to understand the trade-off both through simulations and analysis.

5. Conclusion

In this article, we considered the problem of clustering objects from crowd-sourced pairwise comparisons. We looked at the generative model for clustering and the clustering algorithms from the point of view of inference from crowdsourced pairwise comparisons. Experiments on the real data set and synthetic data sets show that clustering algorithms can be quite effective for this task. In the future, we would like to compare clustering algorithms and Bayesian inference algorithms.

References

Ailon, Nir, Chen, Yudong, and Xu, Huan. Breaking the small cluster barrier of graph clustering. *CoRR*, abs/1302.4549, 2013.

Ames, Brendan P. W. Robust convex relaxation for the planted clique and densest k-subgraph problems. 2013.

Ames, Brendan P. W. and Vavasis, Stephen A. Nuclear norm minimization for the planted clique and biclique problems. *Math. Program.*, 129(1):69–89, September 2011. ISSN 0025-5610.

Ames, Brendan P. W. and Vavasis, Stephen A. Convex optimization for the planted k-disjoint-clique problem. *Math. Program.*, 143(1-2):299–337, 2014.

Buhrmester, Michael, Kwang, Tracy, and Gosling, Samuel D. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6 (1):3–5, January 2011. ISSN 1745-6924.

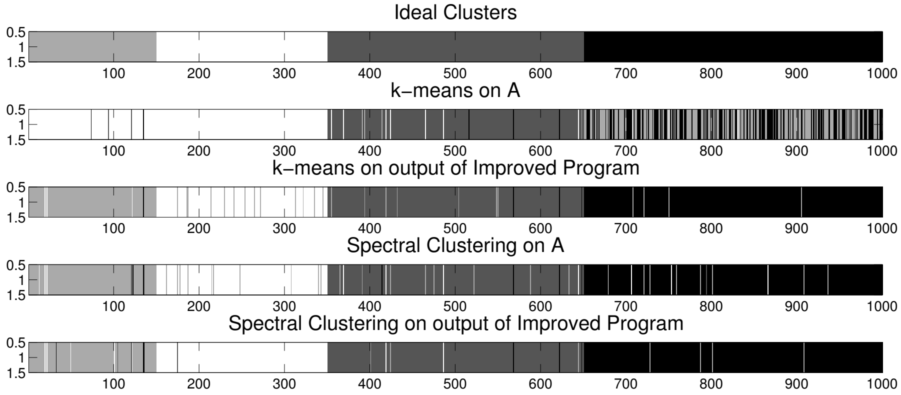


Figure 5. Comparing the clustering output of k-means and spectral clustering on the adjacency matrix with unobserved entries set to 0 and on the rounded output of improved convex program. Synthetic graph on $n = 1000$ nodes with 4 different clusters of sizes $[150, 200, 300, 350]$ with $p = 0.7$, $q = 0.3$ and $r = 0.15$ (Section 4.2.1).

Table 4. Average number (over 10 experiments) of miss-classified images for simulation on 1000 node graph with 4 clusters of equal size, with varying edge densities (Section 4.2.2).

	Avg. # of misclassified images	Success %
K-means on A	34.70	96.53
K-means on output of 3.3	13.90	**98.61**
Spectral Clustering on A	43.00	95.70
Spectral Clustering on output of 3.3	14.50	**98.55**

Candes, Emmanuel J. and Recht, Benjamin. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6): 717–772, December 2009. ISSN 1615-3375.

Candes, Emmanuel J. and Romberg, Justin. Quantitative robust uncertainty principles and optimally sparse decompositions. *Found. Comput. Math.*, 6(2):227–254, April 2006. ISSN 1615-3375.

Candès, Emmanuel J., Li, Xiaodong, Ma, Yi, and Wright, John. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. ISSN 0004-5411.

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

Chandrasekaran, Venkat, Parrilo, Pablo A., and Willsky, Alan S. Rejoinder: Latent variable graphical model selection via convex optimization. *CoRR*, abs/1211.0835, 2012.

Chen, Yudong, Sanghavi, Sujay, and Xu, Huan. Clustering sparse graphs. In Bartlett, Peter L., Pereira, Fernando C. N., Burges, Christopher J. C., Bottou, Lon, and Weinberger, Kilian Q. (eds.), *NIPS*, pp. 2213–2221, 2012.

Chen, Yudong, Jalali, Ali, Sanghavi, Sujay, and Caramanis, Constantine. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

Condon, Anne and Karp, Richard M. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18(2):116–140, 2001.

Dawid, A. P. and Skene, A. M. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28 (1):20–28, 1979. ISSN 00359254.

Domingos, Pedro and Richardson, Matt. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pp. 57–66, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X.

Ester, M., Kriegel, H.-P., and Xu, X. A database interface for clustering in large spatial databases. In *Proceedings of the 1st international conference on Knowledge Discovery and Data mining (KDD'95)*, pp. 94–99. AAAI Press, August 1995.

Fortunato, Santo. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. ISSN 0370-1573.

Gomes, Ryan, Welinder, Peter, Krause, Andreas, and Perona, Pietro. Crowdclustering. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Granada, Spain.*, pp. 558–566, 2011.

Holland, Paul W., Laskey, Kathryn Blackmond, and Leinhardt, Samuel. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983. ISSN 0378-8733.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. ISSN 0360-0300.

Table 5. Comparing the performance of clustering with more but noisy measurements versus fewer measurements of better quality (obtained by majority voting). Average number (over 10 experiments) of miss-classified images for simulation on 1000 node graph with 4 clusters of equal size. (Section 4.2.3).

	Avg. # of misclassified images	Success %
K-means on A	10.20	98.98
K-means on A (Majority Voting)	37.70	96.23
K-means on output of 3.3	2.30	**99.77**
K-means on output of 3.3 (Majority Voting)	31.70	96.83
Spectral Clustering on A	9.90	99.01
Spectral Clustering on A (Majority Voting)	46.80	95.32
Spectral Clustering on output of 3.3	2.5	**99.75**
Spectral Clustering on output of 3.3 (Majority Voting)	34.90	96.51

Jalali, Ali, Chen, Yudong, Sanghavi, Sujay, and Xu, Huan. Clustering partially observed graphs via convex optimization. In Getoor, Lise and Scheffer, Tobias (eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pp. 1001–1008, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Khosla, Aditya, Jayadevaprakash, Nityananda, Yao, Bangpeng, and Fei-Fei, Li. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Mishra, Nina, Schreiber, Robert, Stanton, Isabelle, and Tarjan, Robert. In Bonato, Anthony and Chung, Fan R. K. (eds.), *Algorithms and Models for the Web-Graph*, chapter 5, pp. 56–67. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-77003-9.

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856. MIT Press, 2001.

Oymak, S. and Hassibi, B. Finding Dense Clusters via "Low Rank + Sparse" Decomposition. *arXiv:1104.5186*, April 2011.

Raykar, Vikas C., Yu, Shipeng, Zhao, Linda H., Valadez, Gerardo Hermosillo, Florin, Charles, Bogoni, Luca, and Moy, Linda. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322, August 2010. ISSN 1532-4435.

Sorokin, A. and Forsyth, D. Utility data annotation with Amazon Mechanical Turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW &#039;08. IEEE Computer Society Conference on*, pp. 1–8. IEEE, June 2008. ISBN 978-1-4244-2339-2.

Vijayanarasimhan, Sudheendra and Grauman, Kristen. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014.

Vinayak, Ramya Korlakai, Oymak, Samet, and Hassibi, Babak. Sharp performance bounds for graph clustering via convex optimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pp. 8297–8301, 2014a.

Vinayak, Ramya Korlakai, Oymak, Samet, and Hassibi, Babak. Graph clustering with missing data: Convex algorithms and analysis. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada*, pp. 2996–3004, 2014b.

Von Ahn, Luis, Maurer, Benjamin, McMillen, Colin, Abraham, David, and Blum, Manuel. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1468, 2008.

Welinder, Peter, Branson, Steve, Belongie, Serge, and Perona, Pietro. The multidimensional wisdom of crowds. In *Neural Information Processing Systems Conference (NIPS)*, 2010.

Xu, Huan, Caramanis, Constantine, and Sanghavi, Sujay. Robust pca via outlier pursuit. In Lafferty, John D., Williams, Christopher K. I., Shawe-Taylor, John, Zemel, Richard S., and Culotta, Aron (eds.), *NIPS*, pp. 2496–2504. Curran Associates, Inc., 2010.

Xu, Xiaowei, Jäger, Jochen, and Kriegel, Hans-Peter. A fast parallel clustering algorithm for large spatial databases. *Data Min. Knowl. Discov.*, 3(3):263–290, September 1999. ISSN 1384-5810.

Xu, Ying, Olman, Victor, and Xu, Dong. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18(4):536–545, 2002.

Yang, Qiaofeng and Lonardi, Stefano. A parallel algorithm for clustering protein-protein interaction networks. In *CSB Workshops*, pp. 174–177. IEEE Computer Society, 2005. ISBN 0-7695-2442-7.

Yi, Jinfeng, Jin, Rong, Jain, Anil K., Jain, Shaili, and Yang, Tianbao. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, Nevada, United States.*, pp. 1781–1789, 2012.