TENSOR-BASED CROWDSOURCED CLUSTERING VIA TRIANGLE QUERIES

Ramya Korlakai Vinayak[1], Tijana Zrnic[2], Babak Hassibi[1]

[1]California Institute of Technology, Pasadena, CA, USA, [2]University of Novi Sad, Novi Sad, Serbia

ABSTRACT

We consider the problem of crowdsourced clustering of a set of items based on queries of the similarity of triple of objects. Such an approach, called triangle queries, was proposed in [1], where it was shown that, for a fixed query budget, it outperforms clustering based on edge queries (i.e, comparing pairs of objects). In [1] the clustering algorithm for triangle and edge queries was identical and each triangle query response was treated as 3 separate edge query responses. In this paper we directly exploit the triangle structure of the responses by embedding them into a 3-way tensor. Since there are 5 possible responses to each triangle query, it is a priori not clear how best to embed them into the tensor. We give sufficient conditions on non-trivial embedding such that the resulting tensor has a rank equal to the underlying number of clusters (akin to what happens with the rank of the adjacency matrix). We then use an alternating least squares tensor decomposition algorithm to cluster a noisy and partially observed tensor and show, through extensive numerical simulations, that it significantly outperforms methods that make use only of the adjacency matrix.

Index Terms— Crowdsourced Clustering, Tensor Decomposition

1. INTRODUCTION

Crowdsourcing - the process of collecting data from workers on platforms such as Amazon Mechanical Turk for various applications has recently become quite popular [2, 3]. The workers on these platforms are often *non-experts* and hence the answers obtained will be *noisy*. Therefore both problems of designing *queries* and designing *algorithms* for inferring quality data from such non-expert workers are of importance.

Crowdsourced Clustering: [4, 5, 1]. Consider the task of collecting labels of unlabelled images, e.g, of birds of different species. To label the image of a bird, a worker should either have some expertise regarding bird species, or should be trained, both of which are expensive. However, answering a comparison question, such as, "Do these two birds belong to the same species?" is much easier than the labeling task and does not require expertise or training. Though different workers might use different criteria for comparison, e.g, color of feathers, shape, size etc., the hope is that, averaged over the crowd workers, we will be able to reasonably resolve the clusters (and label each).

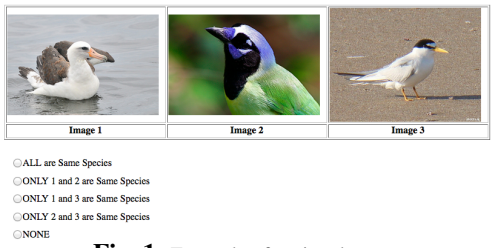

Fig. 1. Example of a triangle query.

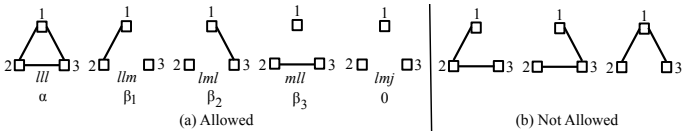

Fig. 2. Configurations for a triangle query that are (a) observed and (b) not allowed.

In [1] we considered the problem of query design for crowdsourced clustering and showed that for a fixed query budget, we can obtain better quality answers (and hence better clustering) by making random triangle queries, where three items are compared per query (Figure 1) as compared to making random edge queries where a pair of items are compared. However, in [1], the information obtained from the triangle queries was embedded into an adjacency matrix which was input to graph clustering algorithms. Such an embedding treats each triangle query as 3 separate edges and ignores the triangle structure itself. A more natural embedding is to consider a tensor where the query result for each triple of items $\{i, j, k\}$ is embedded into the ijk-th entry of a 3-way tensor.

Entry A_{ij} of an adjacency matrix of a graph holds information about the *pair* of nodes $\{i, j\}$, which has two possible configurations, *edge*, encoded by $A_{ij} = 1$ and *no edge*, encoded by $A_{ij} = 0$. The true adjacency matrix, $\mathbf{A}^*$, obtained by this simple encoding has a low-rank structure that reflects the underlying clusters and the rank is equal to the number of clusters. A triangle query has 5 possible answers (Figure 2(a)): (1) All items are similar, denoted by lll, (2) Items 1 & 2 are similar, denoted by llm, (3) Items 1 & 3 are similar, denoted by lml, (4) Items 2 & 3 are similar, denoted by mll and (5) None are similar, denoted by lmj. So, we need an encoding scheme with 5 alphabets to embed the information obtained from a triangle query. Moreover, we also would like the true tensor, $\mathbf{T}^*$, obtained by this embedding to have a low-rank structure that reflects the underlying clusters.

Our Contributions: In this paper, we propose a general encoding scheme for filling a tensor from triangle queries (Section 3.1) and provide sufficient conditions on this encoding scheme to give a true tensor with unique (up to scaling and permutations) CP-decomposition of rank equal to the number of clusters (Section 3.2, 3.3). We also provide extensive numerical simulations (Section 4) that show that using tensor decomposition methods can improve over clustering obtained via the adjacency matrix.

2. TENSORS: A QUICK RECAP

A *tensor* is a multidimensional array. [6] provides a very good survery on tensors. In this paper we will focus on 3-way tensors. In this section we provide a few properties that are relevant for our results.

A rank-1 tensor is an outer product of 3 vectors $\mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z}$ with $(\mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z})_{ijk} = x_i y_j z_k$. A rank-$K$ tensor, $\mathbf{T}$ can be written as a sum of K rank-1 tensors (CP-decomposition): $\mathbf{T} = \sum_{l=1}^{K} \mathbf{u}_l \otimes \mathbf{v}_l \otimes \mathbf{w}_l = \mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}$, where $\mathbf{u}_l, \mathbf{v}_l, \mathbf{w}_l \in \mathbb{C}^n$, $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_K]$, $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_K]$ and $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K]$. Recall that the *Kruskal rank* of a matrix $\mathbf{A}$, denoted by krank($\mathbf{A}$), is the maximal number K, such that any set of K columns of $\mathbf{A}$ is linearly independent. The CP-decomposition of a tensor is unique up to scaling and permutations of the factors under mild conditions:

Theorem 2.1 (Kruskal) *[7, 8] The CP-decomposition of a $n \times n \times n$ tensor, $\mathbf{T} = \mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}$ (with $\mathbf{U}, \mathbf{V}, \mathbf{W}$ being $n \times K$ matrices), is unique up to scaling and permutations if krank$(\mathbf{U})$ + krank$(\mathbf{V})$ + krank$(\mathbf{W}) \geq 2K + 2$.*

3. MAIN RESULTS

Consider a graph on n items with K disjoint clusters. In this section we present a scheme to encode the answers to the triangle queries in a tensor and provide sufficient conditions that guarantee a unique (up to scaling and permutations) rank-K CP-decomposition of the true tensor.

3.1. Encoding Scheme for Embedding Triangle Queries

Recall that a triangle query has 5 possible configurations (Figure 2(a)). We propose the following encoding scheme to embed the response to the query $\{i, j, k\}$:

1. If $\{i, j, k\}$ are in the same cluster, $T_{ijk} = \alpha \neq 0$.
2. If i, j are in the same cluster but k is not, $T_{ijk} = \beta_1 \neq 0$.
3. If i, k are in the same cluster but j is not, $T_{ijk} = \beta_2 \neq 0$.
4. If j, k are in the same cluster but i is not, $T_{ijk} = \beta_3 \neq 0$.
5. If $\{i, j, k\}$ are all in different clusters, $T_{ijk} = 0$.

And $T_{iii} = \alpha$, $\forall i$. Note that $\mathbf{T}$ is *not* a symmetric tensor in general. However it does have the following symmetries:

1. If $\{i, j, k\}$ is of the configuration lll, then $T_{ijk} = T_{jik} = T_{ikj} = T_{jki} = T_{kij} = T_{kji} = \alpha$. Also, $T_{iij} = T_{iji} = T_{jii} = \alpha$, and similarly for all the permutations of $\{jji, iik, kki, jjk, kkj\}$.
2. If $\{i, j, k\}$ is of the configuration llm, then $T_{ijk} = T_{jik} = \beta_1$, $T_{ikj} = T_{jki} = \beta_2$, and $T_{kij} = T_{kji} = \beta_3$. Further, $T_{iij} = T_{jii} = T_{jij} = T_{jji} = T_{jij} = T_{ijj} = \alpha$. Also, $T_{iik} = T_{jjk} = \beta_1, T_{iki} = T_{jkj} = \beta_2, T_{kii} = T_{kjj} = \beta_3$.

Similarly for other configurations.

3.2. Low-Rank Tensor Structure

Let $\mathbf{c}_i \in \mathbb{R}^n$ denote the indicator vector of cluster i. That is, $\mathbf{c}_{i_j} = 1$ if node $j \in$ cluster i and 0 otherwise. Let $\mathbf{C} := [\mathbf{c}_1, \cdots, \mathbf{c}_K] \in \mathbb{R}^{n \times K}$. Note that each row of $\mathbf{C}$ has a single 1, since each item belongs to only one cluster. Let $\mathbf{T}^*$ be the full tensor filled using the scheme in Section 3.1 via triangle queries when the there is no noise:

$$\mathbf{T}^* = \alpha \sum_{l=1}^{K} \mathbf{c}_l \otimes \mathbf{c}_l \otimes \mathbf{c}_l + \beta_1 \sum_{l=1}^{K} \sum_{\substack{m=1 \\ m \neq l}}^{K} \mathbf{c}_l \otimes \mathbf{c}_l \otimes \mathbf{c}_m + \beta_2 \sum_{l=1}^{K} \sum_{\substack{m=1 \\ m \neq l}}^{K} \mathbf{c}_l \otimes \mathbf{c}_m \otimes \mathbf{c}_l + \beta_3 \sum_{l=1}^{K} \sum_{\substack{m=1 \\ m \neq l}}^{K} \mathbf{c}_m \otimes \mathbf{c}_l \otimes \mathbf{c}_l. \tag{3.1}$$

The true adjacency matrix $\mathbf{A}^* = \sum_{l=1}^{K} \mathbf{c}_l \mathbf{c}_l^\top = \mathbf{C}\mathbf{C}^\top$, has a low-rank structure, with rank$(\mathbf{A}^*) = K$, the number of clusters. Our goal is to understand if the true tensor $\mathbf{T}^*$ (3.1) has such a low-rank structure. In particular, we want to write $\mathbf{T}^*$ as: $\mathbf{T}^* = \sum_{l=1}^{K} \mathbf{u}_l \otimes \mathbf{v}_l \otimes \mathbf{w}_l = \mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}$, where $\mathbf{u}_l, \mathbf{v}_l, \mathbf{w}_l \in \mathbb{C}^n$, $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_K]$, $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_K]$ and $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K]$.

The following theorem provides the conditions on the encoding scheme (Section 3.1) that is sufficient for $\mathbf{T}^*$ to have a unique rank-K CP-decomposition.

Theorem 3.1 *For the encoding scheme in Section 3.1, $\mathbf{T}^*$ is a rank K tensor for $K \geq 2$ with* unique *(up to scaling and permutations) CP-decomposition if the following hold:*

1. $\beta_1\beta_2 + \beta_2\beta_3 + \beta_3\beta_1 \neq 0$.
2. $\alpha = \beta_1 + \beta_2 + \beta_3 - K\frac{\beta_1\beta_2\beta_3}{\beta_1\beta_2 + \beta_2\beta_3 + \beta_3\beta_1}$.
3. $\beta_1 \neq -\beta_2$, $\beta_2 \neq -\beta_3$, $\beta_3 \neq -\beta_1$.

3.3. Proof of Theorem 3.1

We observe that $\mathbf{T}^*$ (3.1) can be re-written as: $\mathbf{T}^* = \sum_{l,m,n=1}^{K} \mathbf{B}_{lmn} \mathbf{c}_l \otimes \mathbf{c}_m \otimes \mathbf{c}_n$, where $\mathbf{B}$ is a $K \times K \times K$ tensor (Tucker Decomposition [6]). Suppose we can write $\mathbf{B}$ as a sum of K rank-1 tensors:

$$\mathbf{B} = \sum_{l=1}^{K} \mathbf{f}_l \otimes \mathbf{g}_l \otimes \mathbf{h}_l = \mathbf{F} \otimes \mathbf{G} \otimes \mathbf{H}, \tag{3.2}$$

where $\mathbf{f}_l, \mathbf{g}_l, \mathbf{h}_l \in \mathbb{C}^K$, $\mathbf{F} := [\mathbf{f}_1, \cdots, \mathbf{f}_K]$, $\mathbf{G} := [\mathbf{g}_1, \cdots, \mathbf{g}_K]$, and $\mathbf{H} := [\mathbf{h}_1, \cdots, \mathbf{h}_K]$. Then, $\mathbf{T}^*$ has the following rank-K CP decomposition: $\mathbf{T}^* = (\mathbf{CF}) \otimes (\mathbf{CG}) \otimes (\mathbf{CH})$. So, proving the following lemma is sufficient to prove Theorem 3.1.

Lemma 3.2 *For the encoding scheme in Section 3.1, $\mathbf{B}$ is a rank K tensor for $K \geq 2$ with unique (up to scaling and permutations) CP-decomposition if the conditions in Theorem 3.1 are satisfied.*

Proof We prove Lemma 3.2 by first constructing $\mathbf{F}, \mathbf{G}, \mathbf{H} \in \mathbb{C}^{K \times K}$ that satisfy (3.2) and then showing that it is unique.

Construction: Comparing with (3.1), we note that the $l-$th panel of $\mathbf{B}$, where the third index is kept fixed to l (which

gives a matrix), has the following structure:

$$\mathbf{B}_{(:,:,l)} = \begin{bmatrix} \beta_1 & 0 & \cdots & \beta_3 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \beta_1 & \beta_3 & 0 & \cdots & 0 \\ \beta_2 & \cdots & \beta_2 & \alpha & \beta_2 & \cdots & \beta_2 \\ 0 & \cdots & 0 & \beta_3 & \beta_1 & \cdots & 0 \\ 0 & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \beta_3 & 0 & \cdots & \beta_1 \end{bmatrix}$$

$$= \beta_1 \mathbf{I} + \begin{bmatrix} \mathbf{1} & \mathbf{e}_l \end{bmatrix} \begin{bmatrix} 0 & \beta_3 \\ \beta_2 & K\delta \end{bmatrix} \begin{bmatrix} \mathbf{1}^\top \\ \mathbf{e}_l^\top \end{bmatrix}, \quad (3.3)$$

where $\delta := \frac{\alpha - \beta_1 - \beta_2 - \beta_3}{K}$, $\mathbf{1}$ is a vector of all 1's and $\mathbf{e}_l$ is the standard vector with $\mathbf{e}_l(l) = 1$ and all other entries 0. We note that $\mathbf{B}$ is a *circulant* tensor in the following sense:

$$\mathbf{B}_{(:,:,l+1)} = Z\mathbf{B}_{(:,:,l)}Z^\top, \text{ where } Z := \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}.$$

This is analogous to a circulant matrix in which the columns get circularly shifted. Note that from (3.2), we can write, $\mathbf{B}_{(:,:,l)} = \mathbf{F}\mathbf{D}_{\mathbf{h}_l}\mathbf{G}^\top$, where $\mathbf{D}_{\mathbf{h}_l} = \text{diag}(h_{l1}, \cdots, h_{lK})$. The circulant structure of $\mathbf{B}$ suggests a circulant structure for the factors. Since circulant matrices are diagonalized by Fourier transforms, we may write: $\mathbf{F} := \mathscr{F}\mathbf{\Lambda}\mathscr{F}^\dagger$ and $\mathbf{G}^\top := \mathscr{F}\mathbf{\Xi}\mathscr{F}^\dagger$, where $\mathscr{F}$ is $K \times K$ Fourier matrix normalized with $1/\sqrt{K}$, $\mathbf{\Lambda}$ and $\mathbf{\Xi}$ are diagonal matrices. So, $\mathscr{F}^\dagger\mathbf{B}_{(:,:,l)}\mathscr{F} = \mathbf{\Lambda}\mathscr{F}^\dagger\mathbf{D}_{\mathbf{h}_l}\mathscr{F}\mathbf{\Xi} = \mathbf{\Lambda}\mathbf{A}^{(l)}\mathbf{\Xi}$, where $\mathbf{A}^{(l)} := \mathscr{F}^\dagger\mathbf{D}_{\mathbf{h}_l}\mathscr{F}$ is a circulant matrix. Using (3.3), we can verify that $\mathscr{F}^\dagger\mathbf{B}_{(:,:,l)}\mathscr{F}$ has the following structure:

$$\begin{bmatrix} \beta_1+\beta_3+\beta_2+\delta & (\beta_3+\delta)\eta^{(l-1)} & \cdots & (\beta_3+\delta)\eta^{(K-1)(l-1)} \\ (\beta_2+\delta)\eta^{-(l-1)} & \beta_1+\delta & \cdots & \delta\eta^{(K-2)(l-1)} \\ (\beta_2+\delta)\eta^{-2(l-1)} & \delta\eta^{-(l-1)} & \cdots & \delta\eta^{(K-3)(l-1)} \\ \vdots & \vdots & \ddots & \vdots \\ (\beta_2+\delta)\eta^{-(K-1)(l-1)} & \delta\eta^{-(K-1)(l-1)} & \cdots & \beta_1+\delta \end{bmatrix},$$

where $\eta = e^{j2\pi/K}$, the $K-$th root of unity. Note that the submatrix obtained by removing first row and first column of the above matrix is toeplitz. So, the corresponding submatrix of $\mathbf{\Lambda}\mathbf{A}^{(l)}\mathbf{\Xi}$, where $\mathbf{A}^{(l)}$ is circulant, should also be toeplitz . This gives us conditions (we omit the details for reasons of space) using which we can show that $\mathbf{\Lambda} = \text{diag}(\lambda, 1, \cdots, 1)$ and $\mathbf{\Xi} = \text{diag}(\mu, 1, \cdots, 1)$. By comparing the entries of $\mathscr{F}^\dagger\mathbf{B}_{(:,:,l)}\mathscr{F}$ to those of $\mathbf{\Lambda}\mathbf{A}^{(l)}\mathbf{\Xi}$, the following should hold:

$$\beta_1+\beta_2+\beta_3+\delta = \lambda\mu(\beta_1+\delta),\ \beta_2+\delta = \mu\delta,\ \beta_3+\delta = \lambda\delta$$

Note that, if $\delta = 0$, then $\beta_2 = \beta_3 = 0$ which is not allowed in the scheme considered. So, assuming $\delta \neq 0$, we get:

$$\lambda = \frac{\beta_3+\delta}{\delta},\ \mu = \frac{\beta_2+\delta}{\delta},\ \beta_1+\beta_3+\beta_2+\delta = \lambda\mu(\beta_1+\delta).$$

Using the expressions for λ and μ, we can solve for δ and hence α in terms of $\beta_i,\ i = 1, 2, 3$ and K:

$$\delta = \frac{-\beta_1\beta_3\beta_2}{\beta_1\beta_3+\beta_3\beta_2+\beta_1\beta_2},\ \beta_1\beta_2+\beta_2\beta_3+\beta_3\beta_1 \neq 0$$

$$\alpha = \beta_1+\beta_3+\beta_2 - K\frac{\beta_1\beta_3\beta_2}{\beta_1\beta_3+\beta_3\beta_2+\beta_1\beta_2} \quad (3.4)$$

Recall that $\mathbf{A}^{(l)} = \mathscr{F}^\dagger\mathbf{D}_{\mathbf{h}_l}\mathscr{F}$. So, the diagonal of $\mathscr{F}\mathbf{A}^{(l)}\mathscr{F}^\dagger$ gives the $l-$th row of $\mathbf{H}$. More elaborate calculations, omitted here for the reasons of space, show that $\mathbf{h}_l = \beta_1\mathbf{1} + K\delta\mathbf{e}_l$. Thus, $\mathbf{H}$ is a circulant matrix (as expected) with eigenvalues: $K\delta[(\beta_1+\delta)/\delta, 1, \cdots, 1]$ (Fourier transform of the first row).

Uniqueness: If $\mathbf{F}, \mathbf{G}, \mathbf{H}$ are full rank, then their kruskal rank is K, and hence from Theorem 2.1, the CP-decomposition of $\mathbf{B}$ (3.2) is unique. If $\beta_3+\delta \neq 0$ (i.e, $\lambda \neq 0$) and $\beta_2+\delta \neq 0$ (i.e, $\mu \neq 0$), then $\mathbf{F}$ and $\mathbf{G}$ are full rank. Further, if $\beta_1+\delta \neq 0$, and $\delta \neq 0$, then $\mathbf{H}$ is also full rank. Using the expression for δ in (3.4), these sufficient conditions translate to $\beta_1 \neq -\beta_2, \beta_1 \neq -\beta_3, \beta_2 \neq -\beta_3$. ■

3.4. Discussion

1. The encoding scheme is a function of $\{\beta_1, \beta_2, \beta_3, K\}$ as they fix the value of α.
2. $\mathbf{T}^*$ with the encoding considered is not orthogonal in general, i.e, the factors $\mathbf{F}, \mathbf{G}, \mathbf{H}$ of $\mathbf{B}$ and hence $\mathbf{U}, \mathbf{V}, \mathbf{W}$ of $\mathbf{T}^*$ might not be orthogonal.
3. $\mathbf{T}^*$ with the encoding considered is not symmetric in general, unless $\beta_1 = \beta_2 = \beta_3$.
4. It is interesting to note that we can recover the adjacency matrix $\mathbf{A}$ from $\mathbf{T}$, even when we encode all single edge results to the same, i.e, $\beta_1 = \beta_2 = \beta_3 = \beta \implies \alpha = (3 - K/3)\,\beta$, as long as $\alpha \neq \beta$ (when $K = 6$).
5. In general, let $\beta_i = \beta e^{-j\phi_i}$ and $\gamma := e^{j\phi_1} + e^{j\phi_2} + e^{j\phi_3}$. Then from (3.4), $\alpha = \beta(|\gamma|^2 - K)/\bar{\gamma}$.

4. NUMERICAL EXPERIMENTS

In this section we numerically compare the performance of clustering on the adjacency matrix $\mathbf{A}$ to that obtained using the tensor $\mathbf{T}$, both filled by the same set of triangle queries. Let r determine the sparsity (to fix a budget on the number of triangle queries). We generate answers to $\lceil \frac{r}{3}\binom{n}{2} \rceil$ random triangle queries using two different models (described in Section 4.2). $\mathbf{T}$ is filled using the encoding scheme and the symmetries described in Section 3.1 for three different encodings: $\beta = [1, 1, 1]$, $\beta = [1, 2, 3]$ and $\beta = [1, \frac{-1+i}{\sqrt{2}}, \frac{-1-i}{\sqrt{2}}]$. Note that both $\mathbf{A}$ and $\mathbf{T}$ have a lot of missing entries as only a small subset of triples are observed. We use spectral clustering [9] on $\mathbf{A}$ (unobserved entries set to 0).

4.1. Clustering via Tensor Decomposition

Let Ω be the set of triangle queries and $\mathbf{\Omega}$ be an $n \times n \times n$ tensor with $\Omega_{ijk} = 1$ if T_{ijk} is observed. We consider following simple CP-decomposition objective :

$$\min_{\mathbf{U},\mathbf{V},\mathbf{W}} \sum_{i,j,k=1}^{n} \{\Omega_{ijk}(T_{ijk} - \sum_{l=1}^{K} u_{il}v_{jl}w_{kl})\}^2, \quad (4.1)$$

which is a non-convex optimization problem. Note that we fix the number of clusters. We solve (4.1) iteratively using alternating least squares (ALS). In each time step t, ALS updates $\mathbf{U}^t, \mathbf{V}^t, \mathbf{W}^t$, one variable at a time assuming the other two to be fixed. So, assuming $\mathbf{U} = \mathbf{U}^{t-1}, \mathbf{V} = \mathbf{V}^{t-1}$ to be fixed, we can re-write the objective (4.1) as: $\min_{\mathbf{W}} \sum_{i,j,k=1}^{n} \{\Omega_{ij,k}(T_{ij,k} - \sum_{l=1}^{K} M_{ij,l}w_{kl})\}^2$, where

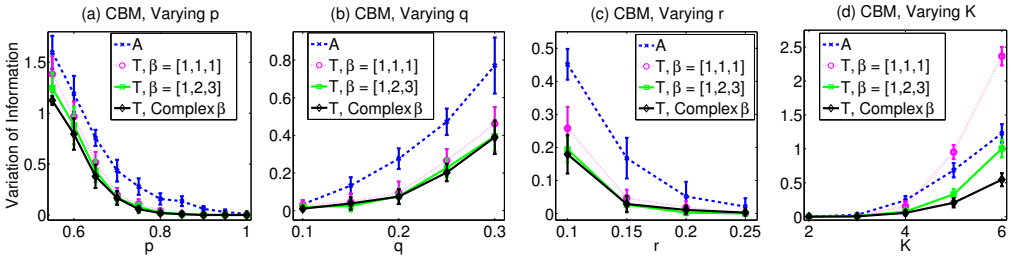

Fig. 3. Comparison of VI (averaged over 10 experiments) for clustering using the tensor (filled using different encoding schemes) compared to that obtained using adjacency matrix, for varying different parameters for the Conditional Block Model (Section 4.3).

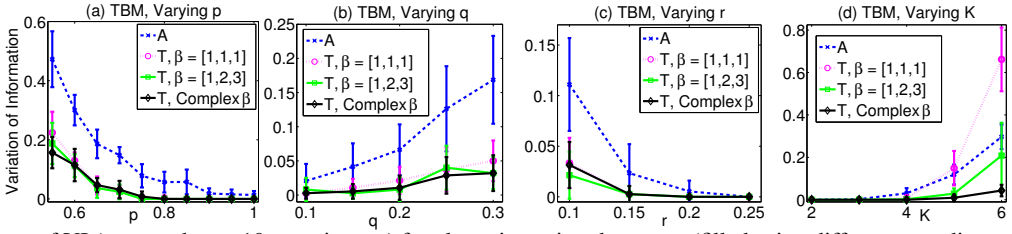

Fig. 4. Comparison of VI (averaged over 10 experiments) for clustering using the tensor (filled using different encoding schemes) compared to that obtained using adjacency matrix, for varying different parameters for the Triangle Block Model (Section 4.3)

$M_{ij,l} := u_{il}^{t-1} v_{jl}^{t-1}$. Note that $\mathbf{M} = \mathbf{U}^{t-1} \odot \mathbf{V}^{t-1} \in \mathbb{C}^{n^2 \times K}$, where $\odot$ denotes Khatri-Rao product. Let $\mathbf{\Omega}^{(w)}, \mathbf{T}^{(w)} \in \mathbb{C}^{n^2 \times n}$ be matricized $\mathbf{\Omega}$ and $\mathbf{T}$ respectively, whose rows are indexed by ij and columns by k of the tensors. Define, $\mathbf{D}_k^{(w)} := \text{diag}(\mathbf{\Omega}^{(w)}(:,k))$, where $(:,k)$ stands for k-th column. Each row of $\mathbf{W}$, equivalently, each column of $\mathbf{W}^\top$ can be solved for by the following least squares problem: $||\mathbf{D}_k^{(w)} \mathbf{T}^{(w)}(:,k) - \mathbf{D}_k^{(w)} \mathbf{M} (\mathbf{W}(k,:))^\top||^2$. We use the clustering $\mathbf{C}^0$ output by the spectral clustering on $\mathbf{A}$ to initialize: $[\mathbf{U}^0, \mathbf{V}^0, \mathbf{W}^0] = [\mathbf{C}^0\mathbf{F}, \mathbf{C}^0\mathbf{G}, \mathbf{C}^0\mathbf{H}]$. We implemented the ALS algorithm on MATLAB using the sparse tensor functions from tensor toolbox [10, 11]. We then run k-means on $[\hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{W}}]$ obtained by ALS to get clusters from $\mathbf{T}$.

4.2. Models for Triangle Queries

We consider two models for generating answers to the triangle queries [1]: *Triangle Block Model* (**TBM**) and *Conditional Block Model* (**CBM**), both of which are derived from the popular Stochastic Block Model (SBM) [12].

SBM is a random graph model for a graph with disjoint clusters. Given the cluster assignements to the nodes, the edges of the graph are independently generated. Edge probability inside the clusters is p and between the clusters is q. In the context of crowdsourcing, if a worker compares items i and j that belong to same cluster, then she will correctly say they are similar with probability p. If they are not in the same cluster, then the probability that she will make an error and say they are similar is q.

For both the TBM and the CBM, given a triple $\{i, j, k\}$, the 3 edges $\{ij, jk, ki\}$ are generated using the SBM. If the configuration thus obtained is one of the 3 configurations that is not allowed (Figure 2(b)), then: (1) TBM assumes that the crowd worker can resolve this to the correct configuration; (2) CBM will regenerate the 3 edges until one of the allowed configuration is obtained. More detailed descriptions of these models are available in [1].

4.3. Simulation Results

Consider a graph on $n = 450$ nodes with $K = 3$ clusters of equal size. We vary the following parameters:

(a)**Varying** p**:** Let $q = 0.25, r = 0.1$. We vary the edge density inside the clusters p from 0.55 to 1 in steps of 0.05.

(b) **Varying** q**:** Let $p = 0.7, r = 0.1$. We vary the edge density between the clusters q from 0.1 to 0.25 in steps of 0.05.

(c) **Varying** r**:** Let $q = 0.25, p = 0.7$. We vary the sparsity parameter r from 0.1 to 0.25 in steps of 0.05.

(d) **Varying** K**:** Consider a graph on $n = 480$ nodes with clusters of equal sizes and $K = [2, 3, 4, 5, 6]$ and hence the cluster sizes get varied. Let $q = 0.25, p = 0.7, r = 0.2$.

Figures 3 and 4 show the results for the CBM and TBM respectively. We compare the output clustering with the ground truth via *variation of information* (VI) [13] which is defined for two clusterings (partitions) of a dataset and has information theoretical justification. Smaller values of VI indicate a closer match and VI $= 0$ means that the clusterings are identical. We compare clustering on $\mathbf{A}$ (dashed blue line) to that obtained by clustering $\mathbf{T}$ when $\beta = [1, 1, 1]$ (dotted pink line), $\beta = [1, 2, 3]$ (solid green line) and complex $\beta = [1, \frac{-1+i}{\sqrt{2}}, \frac{-1-i}{\sqrt{2}}]$ (solid black line). Note that $\beta = [1, 1, 1]$ performs worse as K increases (Figures 3(d), 4(d); note that when $K = 6, \alpha = \beta$). We also note that clustering on $\mathbf{T}$ encoded with different β outperforms that obtained by $\mathbf{A}$. In particular, the complex βs uniformly outperform others.

5. CONCLUSION AND FUTURE WORK

In this paper we considered the problem of crowdsourced custering via triangle queries. We proposed an encoding scheme to embed the answers to triples in a tensor and provided sufficient conditions for it to give a true tensor of rank equal to the number of clusters. We also showed, through extensive simulations, that using tensor decomposition for clustering significantly improves the clustering obtained via the adjacency matrix. Future work will focus on improved clustering algorithms that exploit the sparse structure of the noise.

6. REFERENCES

[1] R. K. Vinayak and B. Hassibi, "Crowdsourced clustering: Querying edges vs triangles," in *Neural Information Processing Systems Conference (NIPS)*, 2016.

[2] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1297–1322, 2010.

[3] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, EMNLP '08, pp. 254–263.

[4] R. Gomez, P. Welinder, A. Krause, and P Perona, "Crowdclustering," in *Neural Information Processing Systems Conference (NIPS)*, 2011.

[5] R. K. Vinayak, S. Oymak, and B. Hassibi, "Graph clustering with missing data: Convex algorithms and analysis," in *Neural Information Processing Systems Conference (NIPS)*, 2014.

[6] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[7] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95 – 138, 1977.

[8] Nicholas D Sidiropoulos, Rasmus Bro, and Georgios B Giannakis, "Parallel factor analysis in sensor array processing," *IEEE transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[9] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[10] B. W. Bader, T. G. Kolda, et al., "Matlab tensor toolbox version 2.6," Available online, February 2015.

[11] B. W. Bader and T. G. Kolda, "Efficient MATLAB computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, December 2007.

[12] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109 – 137, 1983.

[13] M. Meilă, "Comparing clusterings: an information based distance," *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007.