A Provable Generalized Tensor Spectral Method for Uniform Hypergraph Partitioning

Debarghya Ghoshdastidar DEBARGHYA.G@CSA.IISC.ERNET.IN
Ambedkar Dukkipati AD@CSA.IISC.ERNET.IN
Departiment of Computer Science & Automation,
Indian Institute of Science, Bangalore – 560012, India

Abstract

Matrix spectral methods play an important role in statistics and machine learning, and most often the word 'matrix' is dropped as, by default, one assumes that similarities or affinities are measured between two points, thereby resulting in similarity matrices. However, recent challenges in computer vision and text mining have necessitated the use of multi-way affinities in the learning methods, and this has led to a considerable interest in hypergraph partitioning methods in machine learning community. A plethora of "higher-order" algorithms have been proposed in the past decade, but their theoretical guarantees are not well-studied. In this paper, we develop a unified approach for partitioning uniform hypergraphs by means of a tensor trace optimization problem involving the affinity tensor, and a number of existing higher-order methods turn out to be special cases of the proposed formulation. We further propose an algorithm to solve the proposed trace optimization problem, and prove that it is consistent under a planted hypergraph model. We also provide experimental results to validate our theoretical findings.

1. Introduction

The underlying problem in most clustering approaches is to optimize a certain objective function involving pairwise relations among all data instances, where the optimization is performed over all possible cluster assignment matrices. Various relaxations of this NP-hard problem are common in practice. For instance, k-means (Lloyd, 1982) uses a greedy approach for achieving a local optimum, while spectral clustering does not restrict the optimization to binary-valued assignment matrices (Donath & Hoffman, 1973). On the other hand, non-negative matrix factorization (Lee & Seung, 2001) derives an approximation of the assignment matrix by decomposing the similarity matrix.

Still, there are numerous applications, where pairwise similarities are either inappropriate for the purpose or provide poor performance, for example, subspace clustering (Agarwal et al., 2005), graph matching (Duchenne et al., 2011) etc. To tackle such problems, a wide class of algorithms, often coined as "higher-order" methods, have been proposed in the last decade. The basic idea in these approaches is to use a m-way similarity measure with $m > 2$. While it is common to simply pose this as an optimization problem, where the objective function is justified from various perspectives (Kim et al., 2011; Rota Bulo & Pelillo, 2013), there are works that formulate above problem as a hypergraph partitioning problem and use hypergraph reduction techniques (Agarwal et al., 2005; 2006; Arias-Castro et al., 2011) or alternative solution strategies (Hein et al., 2013; Karypis & Kumar, 2000). Since, the constructed hypergraph is m-uniform, *i.e.*, every edge spans m vertices, it is more natural to exploit the structure of the m^{th}-order affinity tensor of the hypergraph. Hence, one can partition the hypergraph using higher-order SVD of tensors (Govindu, 2005), non-negative tensor factorization (Shashua et al., 2006) or tensor power iterations (Duchenne et al., 2011).

Despite the wide variety of solution strategies, one finds that most of the above algorithms attempt to solve a similar optimization problem. This is not surprising because, as in the case of graph based clustering (Shi & Malik, 2000), the primary objective in above methods is to obtain a grouping with high intra-cluster similarity. Unfortunately, unlike the case of graphs, there still remains a lack of unified treatment of above methods from a common perspective. This paper fills the gap between graph and uniform hypergraph partitioning by providing a general notion of associativity maximization and its formulation as a tensor trace maximization problem involving the affinity tensor of the hyper-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

graph. Our formulation encompasses normalized spectral clustering as well as several existing higher-order methods.

Another concern with the related literature is the absence of provable higher-order clustering algorithms. Most standard clustering algorithms with pairwise information have been well studied using tools from spectral graph theory (Chung, 1997) or matrix theory (Stewart & Sun, 1990). In fact, perturbation bounds (Ng et al., 2002) and consistency results (von Luxburg et al., 2008; Rohe et al., 2011) have become standard techniques for proving correctness of spectral clustering and similar algorithms. On the other hand, spectral theory of uniform hypergraphs (Cooper & Dutle, 2012; Hu & Qi, 2012) is still in its early stage, and is yet to provide guarantees for partitioning algorithms. Moreover, the role of tensor decompositions (De Lathauwer et al., 2000; Lim, 2005) and tensor perturbation analysis (Anandkumar et al., 2014) in higher-order clustering is rarely studied in the literature. Till date, error bounds for clustering have only been studied in the context of hybrid linear modeling (Arias-Castro et al., 2011), and recently, in a planted partition setting (Ghoshdastidar & Dukkipati, 2014). Our second contribution is an algorithm for solving the proposed trace maximization problem. We prove that, under a planted hypergraph model, the proposed algorithm is consistent. The techniques used in our results are significantly different from existing analysis (Arias-Castro et al., 2011; Ghoshdastidar & Dukkipati, 2014). We also supplement our studies with numerical results on benchmark datasets.

2. Partitioning Uniform Hypergraphs

An undirected hypergraph is a structure on n vertices, $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where each edge connects a subset of vertices and may have a non-negative weight associated with it. The aim is to partition $\mathcal{V}$ into k disjoint sets, $\mathcal{V}_1, ..., \mathcal{V}_k$, based on the presence or weight of edges. For m-uniform hypergraphs, every edge is a collection of exactly m vertices and the associated weights are often termed as m-way affinities. One often represents such affinities in a m^{th}-order n-dimensional symmetric tensor, $\mathbf{W}$, usually called the affinity tensor of the hypergraph. A special case is that of graphs (2-uniform hypergraphs), where the affinities are given by a n-dimensional symmetric matrix W. In normalized spectral clustering, one partitions the graphs such that the normalized cut of the partition is minimized. The problem can be alternatively formulated in terms of maximization of normalized associativity (Shi & Malik, 2000) that can be expressed as a trace maximization problem (von Luxburg, 2007)

$$\underset{H}{\text{maximize}} \ \text{Trace}(H^T \overline{W} H), \tag{1}$$

where the maximum is taken over all matrices $H \in \mathbb{R}^{n\times k}$ with $H_{ij} = \frac{1}{\sqrt{|\mathcal{V}_j|}} \mathbf{1}_{\{v_i \in \mathcal{V}_j\}}$. Here, $\overline{W} = D^{-1/2} W D^{-1/2}$ is the normalized affinity matrix, and D is a diagonal matrix containing degrees of every vertex. One can see that H contains orthonormal columns, and hence, a spectral relaxation of the NP-hard problem in (1) is usually solved as

$$\underset{Z\in\mathbb{R}^{n\times k}}{\text{maximize}} \ \text{Trace}\left(Z^T \overline{W} Z\right) \ \text{ s.t. } Z^T Z = I\,, \tag{2}$$

which is maximized by the k leading eigenvectors of $\overline{W}$.

2.1. Maximizing Normalized Associativity

We formulate an objective for partitioning hypergraphs that is similar in essence to the objective of spectral clustering given in (1)-(2). For this, we introduce the following notion of associativity in a uniform hypergraph.

Definition 1 (Normalized Associativity)**.** *Let $\mathcal{V}_1 \subseteq \mathcal{V}$ represent a set of vertices in a m-uniform hypergraph with m^{th}-order affinity tensor $\mathbf{W} \in \mathbb{R}^{n\times n\times \ldots \times n}$. The associativity of $\mathcal{V}_1$ is defined as*

$$\text{Assoc}(\mathcal{V}_1) = \sum_{v_{i_1}, v_{i_2}, \ldots, v_{i_m} \in \mathcal{V}_1} \mathbf{W}_{i_1 i_2 \ldots i_m}.$$

Further, if $\mathcal{V}_1, \ldots, \mathcal{V}_k$ is such that $\cup_{i=1}^k \mathcal{V}_i \subseteq \mathcal{V}$ and $\mathcal{V}_i \cap \mathcal{V}_j = \phi$ for all $i \neq j$, then the normalized associativity of the partition is defined as

$$\text{N-associativity}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{i=1}^{k} \frac{\text{Assoc}(\mathcal{V}_i)}{|\mathcal{V}_i|^{m/2}}\,, \tag{3}$$

where $|\mathcal{V}_i|$ denotes the size of i^{th} partition.

For $m = 2$, the above normalization is similar to that used in ratio cut (von Luxburg, 2007). For higher values of m, the normalizing factor increases to counter the increase in the number of edges. Our objective for hypergraph partitioning is based on maximization of above notion of associativity. We later show that this leads to standard tensor problems more naturally as compared to a cut formulation. One can find definitions of hypergraph cuts and Laplacians and related partitioning approaches in (Hein et al., 2013; Hu & Qi, 2012). We now provide an equivalent problem based on tensor trace maximization similar to (2). To make the notation simple, we use following definition from (De Lathauwer et al., 2000).

Definition 2 (mode-k product)**.** *Let $A \in \mathbb{R}^{p\times n_k}$ and m^{th}-order tensor $\mathbf{W} \in \mathbb{R}^{n_1\times n_2\times \ldots \times n_m}$. The mode-$k$ product of $\mathbf{W}$ and A is a m^{th}-order tensor, represented as $\mathbf{W} \times_k A \in \mathbb{R}^{n_1\times \ldots n_{k-1}\times p\times n_{k+1}\times \ldots n_m}$, whose elements are*

$$(\mathbf{W} \times_k A)_{i_1 \ldots i_{k-1} j i_{k+1} \ldots i_m} = \sum_{i_k=1}^{n_k} \mathbf{W}_{i_1 \ldots i_{k-1} i_k i_{k+1} \ldots i_m} A_{j i_k}.$$

Theorem 3. *Let* $\mathbf{W}$ *be the affinity tensor of a* m*-uniform hypergraph, and* $\mathbf{1}_{\{\cdot\}}$ *denote indicator function. The problem of partitioning the vertices into* k *disjoint clusters while maximizing normalized associativity* (3) *is equivalent to*

$$\underset{H}{\text{maximize}} \ \text{Trace}(\mathbf{W} \times_1 H^T \times_2 \ldots \times_m H^T), \quad (4)$$

where the maximum is taken over all matrices $H \in \mathbb{R}^{n \times k}$ *with* $H_{ij} = \frac{1}{\sqrt{|\mathcal{V}_j|}} \mathbf{1}_{\{v_i \in \mathcal{V}_j\}}$.

Though the above problem is NP-hard, one can observe that the columns of H are orthonormal. So, we may relax (4) as

$$\underset{Z \in \mathbb{R}^{n \times k}: Z^T Z = I}{\text{maximize}} \ \text{Trace}(\mathbf{W} \times_1 Z^T \ldots \times_m Z^T), \quad (5)$$

which is similar to the spectral relaxation in (2).

2.2. Related Problems in Tensor Algebra

In this section, we discuss the connections of the problem in (5) to general problems studied in the context of tensors. A dedicated discussion on the literature of higher-order clustering is postponed till the next section.

In (5), we essentially maximize the trace of a tensor via orthogonal transformations. This has been previously studied in the signal processing community for blind source separation problems (Comon, 2014). One also solves a variant of this where the sum of squares of diagonal elements is maximized. Such an objective leads to the approach studied in (Ghoshdastidar & Dukkipati, 2014). Thus, our formulation complements the above work in the sense that both methods try to perform higher-order clustering by formulating the problem as two well-known tensor problems related to maximization of diagonal terms.

Our proposed partitioning objective (5) is also related to the tensor eigenvalue problem (Lim, 2005), where the m^{th}-order tensor $\mathbf{W}$ is viewed as a m-linear functional. This is easy to observe since for any $x_1, \ldots, x_m \in \mathbb{R}^n$, $\mathbf{W}(x_1, \ldots, x_m) := \mathbf{W} \times_1 x_1^T \times_2 \ldots \times_m x_m^T$ is a scalar function, linear in each argument $x_1, \ldots, x_m$. The maximizers of this function under unit ℓ_2-norm constraints are known as the ℓ_2-eigenvectors of $\mathbf{W}$ (Lim, 2005). Based on this definition, we observe the following regarding (5).

Corollary 4. *Let* $z_1, \ldots, z_k$ *be the columns of* Z*, then*

$$\text{Trace}\left(\mathbf{W} \times_1 Z^T \ldots \times_m Z^T\right) = \sum_{j=1}^{k} \mathbf{W}(z_j, ..., z_j), \quad (6)$$

where each term in the sum is the normalized associativity of individual clusters. So, if one relaxes the orthogonality constraint in (5) *but retains the constraint* $\|z_j\|_2 = 1$ *for all* j*, then the stationary points of the trace maximization are matrices whose columns are* ℓ_2*-eigenvectors of* $\mathbf{W}$.

3. Relation with Existing Clustering Methods

The purpose of this section is to show that a wide variety of pairwise and higher-order clustering algorithms solve some variant of the optimization problems in (4)-(5). This establishes the generality of the associativity based objective presented in Section 2.1.

Spectral Methods. The similarity of the formulation in (5) to standard spectral clustering (2) is quite evident, though one may note that, in our framework, one considers a graph with affinities given by $\overline{W}$ instead of W. In this respect, our framework is more similar to a spectral relaxation of the k-means algorithm (Zha et al., 2001). However, the decoupling of the normalization from the eigenvalue problem gives the freedom to use alternative normalizations. For instance, one can use a doubly stochastic normalization (Zass & Shashua, 2006), which gives superior performance.

Non-negative tensor factorization. Shashua et al. (2006) generalized the use of non-negative matrix factorization for clustering to the case of tensors. The objective is to approximate a hyperstochastic affinity tensor $\mathbf{W}$ by a sum of k non-negative rank-one tensors. Note that for a vector $h \in \mathbb{R}^n$, the m^{th}-order rank-one tensor, $h^{\otimes m}$, has the entries $(h^{\otimes m})_{i_1 i_2 \ldots i_m} = h_{i_1} h_{i_2} \ldots h_{i_m}$. The reason for such an approximation is that $\mathbf{W}$ contains the total probability that m points lie in same cluster, whereas, under conditional independence, each rank-one tensor represent the joint probability that m vertices lie in each cluster. So the optimization problem in (Shashua et al., 2006) is

$$\underset{z_1, .., z_k \in \mathbb{R}^n_+}{\text{minimize}} \left\| \mathbf{W} - \textstyle\sum_{j=1}^{k} z_j^{\otimes m} \right\|_o^2 \text{ s.t. } z_i^T z_j = \mathbf{1}_{\{i=j\}}, \quad (7)$$

where $\|\cdot\|_o^2$ is the sum of squares of entries in the tensor. Non-negativity of z_i's is due to probabilistic reason, and orthogonality ensures hard clustering. The objective functions of (4) and (7) are related by following relation.

$$\left\| \mathbf{W} - \textstyle\sum_{j=1}^{k} z_j^{\otimes m} \right\|_o^2 = \|\mathbf{W}\|^2 + k - 2\,\text{Trace}\left(\mathbf{W} \times_1 Z^T \times_2 \ldots \times_m Z^T\right), \quad (8)$$

where $Z = [z_1 \ldots z_k]$. It immediately follows that (7) is a relaxation of (4), which is tighter than (5) because of the non-negativity constraint.

Hypergraph reduction by clique averaging. A common approach to partitioning general hypergraphs is by reducing it to a graph, and subsequently partitioning the equivalent graph. Two standard reductions include the clique and star expansions (Agarwal et al., 2005). In fact, Agarwal et al. (2006) showed that a variety of popular hypergraph Laplacian formulations are equivalent one of these expansions. Moreover, it is known that for uniform hypergraphs, the eigenvectors for both expansions are similar (Agarwal et al., 2006), proving generality of clique expansion.

We show that the use of clique expansion for hypergraph partitioning is a relaxation of (5). This is true as the affinity matrix obtained after clique averaging is

$$W^c = \sum_{i_3,..,i_m} \frac{\mathbf{W}_{..i_3 \ldots i_m}}{(m-2)!} = \frac{\mathbf{W}(\cdot,\cdot,1_n,\ldots,1_n)}{(m-2)!}, \quad (9)$$

where we view W^c as a bilinear functional, and 1_n is a vector of all ones. A similar reduction of $\mathbf{W}$ was used in the case of local linear approximation based grouping (Arias-Castro et al., 2011). The final form in (9), ignoring constant scaling, clearly indicates that a spectral clustering of the reduced graph is same as finding orthonormal vectors $z_1, \ldots, z_k$ that maximizes

$$\sum_{j=1}^{k} W^c(z_j, z_j) = \sum_{j=1}^{k} \mathbf{W}(z_j, z_j, 1_n \ldots, 1_n). \quad (10)$$

From the representation in (6), we clearly see that the above problem is a relaxation of (5), where we fix the last $m-2$ arguments to reduce (5) to a spectral problem.

Matching via tensor power iterations. In the matching problem, given two sets of points (for instance, sets of features in two images), one needs to extract the points of correspondence. Let s be the size of each set, then one can find s^2 candidate matches. If $X \in \{0,1\}^{s \times s}$ denotes the correspondence matrix, then $\|X\|_F^2 = s$. Duchenne et al. (2011) optimizes a score function over a vectorized form of X as

$$\underset{x \in \{0,1\}^{s^2}}{\text{maximize}} \sum_{i_1,..,i_m} \mathbf{W}_{i_1 \ldots i_m} x_{i_1} \ldots x_{i_m} \text{ s.t. } \|x\|_2^2 = s, \quad (11)$$

where $\mathbf{W}$ is constructed from m-way similarities among the candidate matches. A similar optimization has been considered in (Lee et al., 2011). Duchenne et al. (2011) relaxed the search space to $\mathbb{R}^{s^2}$, and solved (11) using tensor power iterations. Since, the objective in (11) is simply $\mathbf{W}(x, \ldots, x)$, above problem and its relaxation are identical to (4)-(5), where one finds a single cluster that maximizes normalized associativity, *i.e.*, $k = 1$.

Methods with ℓ_1-norm constraint. Given a m^{th}-order similarity tensor $\overline{\mathbf{W}}$ among n data instances, this class of algorithms extract a cluster by solving a generic problem

$$\underset{x \in [0,1]^n}{\text{maximize}} \sum_{i_1,..,i_m} \overline{\mathbf{W}}_{i_1 .. i_m} x_{i_1} \ldots x_{i_m} \text{ s.t. } \|x\|_1 = 1. \quad (12)$$

The justification for such an optimization varies in different approaches. For instance, Rota Bulo & Pelillo (2013) originally viewed the above objective as the expected payoff when each of m players in an evolutionary game chooses one of n vertices of the hypergraph. The solution of (12), x, is the probability distribution corresponding to the equilibrium strategy of the game (Rota Bulo & Pelillo, 2013).

On the other hand, (12) corresponds to maximizing the ensemble m-way affinity of a cluster in (Liu et al., 2010; Leordeanu & Sminchisescu, 2012). In practice, the use of ℓ_1-norm makes the solution sparse, and hard clustering is achieved by sequentially extracting clusters and removing them from the problem. The latter works further restrict the search space to $[0, \epsilon]^n$ to extract larger clusters.

Though (12) involves a ℓ_1-norm constraint, we claim that the problem is a special case of (5). This can be seen by constructing a $(2m)$-uniform hypergraph on the given n vertices, whose affinities are given by the tensor $\mathbf{W}$ with

$$\mathbf{W}_{i_1 i_2 \ldots i_{2m}} = \overline{\mathbf{W}}_{i_1 i_3 \ldots i_{2m-1}} \mathbf{1}_{\{i_l = i_{l+1} \forall l = 1,3,..,m-1\}}. \quad (13)$$

Now, for $x \in [0,1]^n$, $\|x\|_1 = 1$ and $y = \sqrt{x}$ (element-wise root), one can verify that $\mathbf{W}(y, \ldots, y) = \overline{\mathbf{W}}(x, \ldots, x)$ and $\|y\|_2 = 1$. This immediately implies that the equivalence of (12) with the single cluster case of (5).

4. Higher-order Clustering Algorithm

Here, we propose a spectral method that solves a relaxed form of (5). We also provide a theoretical analysis of this algorithm under a planted partition model. Note that, in the literature, theoretical guarantees for higher-order clustering algorithms have been mostly overlooked. We note that some of the tools used in our analysis are quite different from existing analysis for clustering (Arias-Castro et al., 2011; Ghoshdastidar & Dukkipati, 2014) and in the case of tensors (Anandkumar et al., 2014).

4.1. The Proposed Method

The proposed method, listed in Algorithm 1, relaxes the problem in (5), following the lines of the clique averaging technique. To be precise, in (5), we replace Z in mode-3 to mode-m multiplication by a $n \times k$ matrix with all entries as $\frac{1}{\sqrt{n}}$. The substituted matrix is not orthonormal, but has columns of unit norm. Retaining Z in modes-1 and 2 relaxes (5) to a matrix eigendecomposition problem.

Algorithm 1 Tensor Spectral Hypergraph Partitioning

input m-way affinity tensor $\mathbf{W}$; number of partitions k.

1: Compute matrix $A = \mathbf{W} \times_3 \left(\frac{1_n}{\sqrt{n}}\right)^T \ldots \times_m \left(\frac{1_n}{\sqrt{n}}\right)^T$, where 1_n is a n-dimensional vector of ones.
2: Compute matrix $Z \in \mathbb{R}^{n \times k}$ of k leading orthonormal eigenvectors of A.
3: Cluster rows of Z into k clusters by k-means.

output Assign node-i to partition-j if row-i of Z lies in cluster-j.

Before we proceed to the analysis of the algorithm few comments are in place. Note that the Algorithm 1 is listed

in such a way that it is suitable for a theoretical analysis. In practice, one can incorporate few modifications, for instance, one can normalizes the rows of Z before performing k-means (Ng et al., 2002). Moreover, the complexity of Algorithm 1 is $O\left(kn^m\right)$. Such complexity is due to the computation of the tensor, and is common in tensor based approaches. In practice, one uses sampling techniques to reduce the complexity (Ghoshdastidar & Dukkipati, 2015).

4.2. Consistency of Algorithm 1

We analyze Algorithm 1 under a planted partition model defined as follows. Consider a m-uniform random hypergraph on n vertices, where every m-edge occurs with probability $q \in [0, 1)$. Divide the vertices into k disjoint classes of sizes $n_1, \ldots, n_k$. Let $c_1, \ldots, c_k \in \{0, 1\}^n$ be the assignment vector for each class, and for $j = 1, \ldots, k$, let $p_j \in [0, 1 - q]$. Generate additional m-edges within each class such that for vertices in class-j, m-edges occur with probability $(p_j + q)$. The goal of Algorithm 1 is to determine the true assignments $c_1, \ldots, c_k$ from the affinity tensor $\mathbf{W}$ of a random realization of the hypergraph.

We note that the above model includes a number of learning problems, where higher-order methods are used. For instance, the models for subspace clustering and feature matching presented in (Ghoshdastidar & Dukkipati, 2014) are both special cases of the above model. We also mention here that the model in (Ghoshdastidar & Dukkipati, 2014) in more general, and allows complicated random hypergraphs. However, our model suffices for standard learning problems. Few settings are mentioned below.

Example 1. When all classes are of equal size, and $p_1 = \ldots = p_k$ are strictly positive, we obtain the model for subspace clustering (Ghoshdastidar & Dukkipati, 2014).

Example 2. One may extend this model to incorporate the presence of outliers. The outliers are modelled as an additional class that is not strongly connected, *i.e.*, $p_{k+1} = 0$.

Example 3. For the matching problem, there are $k = 2$ classes – the set of correct matches of size $\sqrt{n}$, while the remaining $n - \sqrt{n}$ matches are incorrect, and hence, $p_2 = 0$.

The following result bounds the error incurred by Algorithm 1 under the above random model.

Theorem 5. *If there exists n_0 such that $\delta_n > \sqrt{n \log n}$ for all $n \geq n_0$, then the number of misclustered vertices is $O\left(\frac{k n_{\max} n \log n}{\delta_n^2}\right)$ almost surely as $n \to \infty$. Thus, Algorithm 1 is consistent whenever $\delta_n = \omega\left(\sqrt{k n_{\max} n \log n}\right)$.*

We validate the significance of Theorem 5 in the case of subspace clustering model (Example 1). Similar results can be studied for other models. We let the number of partitions to grow with the size of the hypergraph as $k = O(n^{1/2m})$. This has been considered in (Ghoshdastidar & Dukkipati, 2014).

Corollary 6. *Let $k = O(n^{1/2m})$ in the subspace clustering problem where $n_j = \frac{n}{k}$ and $p_j = p$ for all $j = 1, \ldots, k$. Then almost surely as $n \to \infty$, the misclustering error of Algorithm 1 is at most*

$$|M_n| = O\left(\frac{\log n}{p^2 n^{m-3+\frac{1}{m}}}\right) \quad \text{for all } m \geq 2, \text{ and } p > 0.$$

Thus, Algorithm 1 is almost surely consistent for all $m \geq 3$, while $\frac{|M_n|}{n}$ eventually vanishes for $m = 2$.

We observe that the above bounds are quite similar to the bounds obtained for the HOSVD based algorithm in (Govindu, 2005; Ghoshdastidar & Dukkipati, 2014), which is subsequently referred to as HOSVD. To elaborate on their differences, we note that under the setting of Corollary 6, the misclustering error for HOSVD is

$$|M_n^{HOSVD}| = O\left(\frac{(\log n)^2}{p^4 n^{m-3+\frac{1}{2m}}}\right) \quad \text{for } m \geq 2, p > 0.$$

Comparison of above two results clearly shows that the error bounds for Algorithm 1 are clearly better than that of HOSVD, particularly in terms of the density gap p. More precisely, if the gap between the intra-cluster and inter-cluster edge probabilities are small or decrease with n, then Algorithm 1 is less affected compared to the HOSVD algorithm in (Ghoshdastidar & Dukkipati, 2014).

4.3. Proof of Theorem 5

Here, we present an outline of the proof of Theorem 5. The proofs of the technical lemmas are given in the supplementary material. The key to our analysis is the correctness of Algorithm 1 in the expected case. One can verify that in presence of the partitions, the expected affinity tensor can be expressed as

$$\mathcal{W} := \mathsf{E}[\mathbf{W} | c_1, \ldots, c_k] = \sum_{j=1}^{k} p_j c_j^{\otimes m} + q 1_n^{\otimes m}, \qquad (14)$$

where $1_n^{\otimes m}$ is a rank-one tensor formed from vector 1_n. We note that $\mathcal{W}$ has a CP-decomposition of rank $(k+1)$, which means that $\mathcal{W}$ can be expressed as a sum of $(k+1)$ rank-1 tensors, *i.e.*, each of the $(k+1)$ terms can be written as a m-way outer product of a vector. But, the vectors forming the rank-1 terms are not orthogonal or incoherent. Hence, one cannot use standard tensor perturbation bounds (Anandkumar et al., 2014) to comment on the error of determining the rank-one terms. The next result shows that Algorithm 1 is accurate in the expected case. We require the following.

Let $j^* = \arg\min_j \left(p_j n_j^{m-1}\right)$,

$$g = \frac{1}{n^{(0.5m-1)}} \left(\min_{j \neq j^*} \left(p_j n_j^{m-1}\right) - \left(p_{j^*} n_{j^*}^{m-1}\right) \right),$$

$$\text{and} \quad \delta_n = \left(\frac{p_{j^*} n_{j^*}^{m-1}}{2n^{(0.5m-1)}} + \frac{gqn_{j^*} n^{(0.5m-1)}}{2(g+qn^{0.5m})} \right). \tag{15}$$

The quantity δ_n is a lower bound on the eigen-gap that separates the largest k eigenvalues from other eigenvalues.

Lemma 7. *Let $\mathcal{W}$ be the input for Algorithm 1, and let $\mathcal{Z} \in \mathbb{R}^{n \times k}$ be the corresponding eigenvector matrix. If $\delta_n > 0$ for the model, then the rows i and j of $\mathcal{Z}$ are identical if and only if vertices i and j belong to the same class.*

Above result implies that the k-means algorithm clusters all rows accurately in this case. However, in practice, we work with a random realization $\mathbf{W}$ instead of $\mathcal{W}$, which can be viewed as a perturbation of $\mathcal{W}$, ie $\mathbf{W} = \mathcal{W} + \mathcal{E}$. We quantify the perturbation in terms of the following definition of norm of a tensor (Anandkumar et al., 2014).

Definition 8 (Operator norm)**.** *The operator norm of a m^{th}-order n-dimensional tensor $\mathcal{E}$ is defined as*

$$\|\mathcal{E}\|_{op} = \max_{\|x_1\|_2 = \ldots = \|x_m\|_2 = 1} |\mathcal{E}(x_1, \ldots, x_m)| \,,$$

where the maximum of the m-linear functional is taken over all vectors $x_1, \ldots, x_m \in \mathbb{R}^n$ with $\|x_i\|_2 = 1$ for all i.
In addition, if $\mathcal{E}$ is symmetric, it is known that (Chen et al., 2012)

$$\|\mathcal{E}\|_{op} = \max_{\|x\|_2 = 1} |\mathcal{E}(x, \ldots, x)| \,.$$

Let Z be the eigenvector matrix associated with $\mathbf{W}$. The following result provides an upper bound on the perturbation of Z from $\mathcal{Z}$ in terms of the perturbation $\mathcal{E}$.

Lemma 9. *If $\|\mathcal{E}\|_{op} < \delta_n$, then there exists an orthonormal (rotation) matrix $Q \in \mathbb{R}^{k \times k}$ such that*

$$\|Z - \mathcal{Z}Q\|_F \leq \frac{\sqrt{2k}\|\mathcal{E}\|_{op}}{\delta_n} \,,$$

where $\|\cdot\|_F$ is the Frobenius norm.

The above result is useful only if $\|\mathcal{E}\|_{op}$ is reasonably small. This is ensured in the following lemma, where we use an ϵ-net argument to bound $\|\mathcal{E}\|_{op}$ with high probability.

Lemma 10. *For any $\lambda > 0$,*

$$\mathsf{P}(\|\mathcal{E}\|_{op} > \lambda) \leq 2(1+2m)^n \exp\left(-\frac{2\lambda^2}{m! m^2}\right).$$

One needs to choose λ appropriately. In our case, choosing $\lambda = \sqrt{n \log n}$ ensures that, for any m, the above bound vanishes as $n \to \infty$. So, $\|\mathcal{E}\|_{op} \leq \sqrt{n \log n}$ almost surely as $n \to \infty$. This bound is appears to be significant even in the context of existing works on random tensors. For instance, though bounds on the expected operator norm are known (Nguyen et al., 2010), Lemma 10 provides a similar, yet simpler, bound that holds with high probability.

Combining the bound on $\|\mathcal{E}\|_{op}$ with Lemma 9, we obtain a bound on the difference in eigenspaces for the random and expected cases. This result becomes interesting due to the some keys observations (Rohe et al., 2011). These are summarized in the next lemma, which uses the following notations. Let $n_{\max} = \max_j n_j$. Let γ_i be the i^{th} row of $\mathcal{Z}Q$, and α_i be the center of the cluster in which i^{th} row of Z is grouped. The following result characterizes the set

$$M_n = \left\{ i \in \{1, \ldots, n\} : \|\alpha_i - \gamma_i\|_2 \geq \frac{1}{\sqrt{2n_{\max}}} \right\}.$$

Lemma 11. *Whenever $i \notin M_n$ and vertices i,j are in different classes, $\|\alpha_i - \gamma_i\|_2 < \|\alpha_i - \gamma_j\|_2$. Also, if global optimum is achieved in k-means, then*

$$|M_n| \leq 8n_{\max} \|Z - \mathcal{Z}Q\|_F^2 \,. \tag{16}$$

The above lemma claims that for any vertex not in M_n, the k-means objective is reduced if the vertex is correctly clustered. Hence, all misclustered vertices must belong to M_n, and a bound on the number of such vertices is given by (16). We note that, in practice, standard k-means algorithm finds a local minimum. However, there are variants of the k-means algorithm, for instance (Kumar et al., 2004), which provide a near-optimal solution with error at most $(1+\epsilon)$ times the optimal error for some $\epsilon > 0$. Using such a method allows one to relax the condition of global optimality, while the bound in (16) increases only by a constant factor of $(1+\epsilon)^2$. At this stage, one can combine the above results to arrive at the claim of Theorem 5.

5. Experimental Results

We now numerically demonstrate the performance of Algorithm 1 in a number of problems.

5.1. Partitioning Uniform Random Hypergraphs

We first compare the performance of Algorithm 1 with the HOSVD based algorithm in an artificial setting based on the planted partition model. Here, we randomly generate uniform hypergraphs from the model described in Section 4.2. In Figure 1, we consider bi-partitioning of a m-uniform hypergraph, with inter-class edge probability $q = 0.2$, and density gap $p_1 = p_2 = 0.1$. So, within class edges occur with probability $(p_1 + q) = (p_2 + q) = 0.3$. We consider 2, 3 and 4-uniform hypergraphs with varying number of nodes. The results are averaged over 50 independent runs. Figure 1 shows that error incurred by Algo-

rithm 1 is less than HOSVD algorithm. It also shows that the error reduces for tensors of higher order.

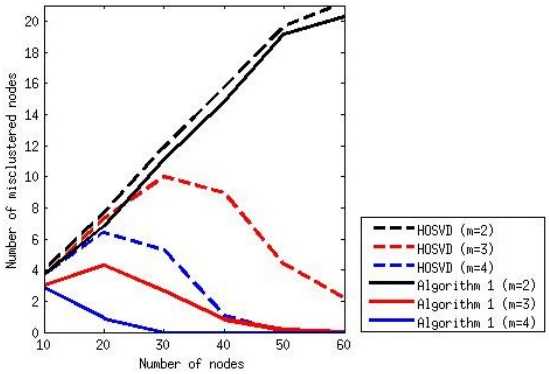

Figure 1. Number of nodes misclustered by HOSVD based approach (dashed lines) and Algorithm 1 (solid lines) as the total number of nodes increases. The black, red and blue lines correspond to cases with $m = 2$, 3 and 4, respectively.

We conduct another study on bi-partitioning 3-uniform hypergraphs, where we fix $q = 0.2$ but the density gaps, p_1, p_2, are varied. We let $p_1 = p_2 = p$, which varies over $\{0.025, 0.05, 0.075, 0.1\}$. Figure 2 shows the number of misclustered nodes, averaged over 50 runs, as the hypergraph grows. Note that the problem becomes harder as p reduces, and the performance of HOSVD is highly affected. But, the effect is much less in case of Algorithm 1.

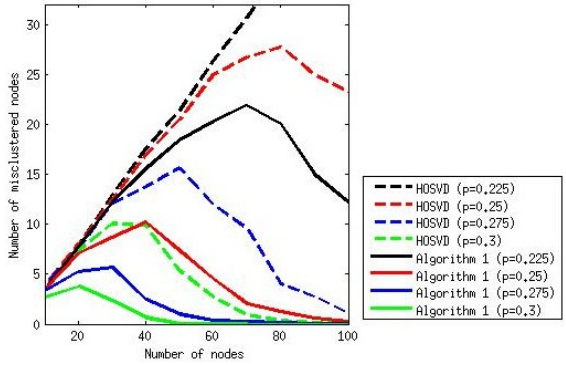

Figure 2. Number of nodes misclustered by HOSVD based approach (dashed lines) and Algorithm 1 (solid lines) as total number of nodes increases. The black, red, blue and green lines correspond to density gap $p = 0.025, 0.05, 0.075$ and 0.1, respectively.

5.2. Comparison with Pairwise Similarity

Before discussing about problems, where higher-order relations are essential, we present a comparative study with normalized spectral clustering (Ng et al., 2002). In spectral clustering, one often defines the pairwise similarity between two data instances x, y as $\exp(-\beta\|x-y\|_2^2)$ with β being a tuning parameter. We show that simple 3-way extension of this gives more robust performance. For any 3 points, x, y, z, we compute the similarity among them as

Dataset	Spectral	Algorithm 1
Iris	0.117±0.138	0.094±0.114
Vertebral Column	0.345±0.017	0.333±0.002
Wine	0.342±0.084	0.331±0.040
Ionosphere	0.325±0.000	0.316±0.000
Haberman's Survival	0.423±0.082	0.392±0.000
Blood Transfusion	0.325±0.000	0.319±0.000

Table 1. Performance of spectral clustering and Algorithm 1.

$$\exp\left(-\beta \max\left\{\|x-y\|_2^2, \|y-z\|_2^2, \|z-x\|_2^2\right\}\right). \quad (17)$$

We run spectral clustering and Algorithm 1, with above 3-way similarity, on some 2 and 3 class datasets from UCI repository (Frank & Asuncion, 2010), where each dataset is normalized. Table 1 reports the mean and standard deviation of the fractional error incurred by both algorithms over 100 runs of k-means. The results show that slight reduction in error is obtained in case of Algorithm 1. This is expected since the 3-way similarity in (17) is essentially constructed from pairwise relations. Moreover, in most cases, there is a significant reduction in the standard deviation of the errors. Since randomness is only at the k-means steps, this implies that 3-way relations provide better embedding of the data points, making the k-means step more consistent.

The above results indicate that even simple higher-order extensions can improve the performance of the algorithm. We show a sample result of how this approach can be extended to image segmentation. Figure 3 shows an image of a maze, which is divided into two segments based on only color and distance information. We can see that better segments are obtained when Algorithm 1 is run with the 3-way similarity of (17). We note that above approaches are quite simple, and do not use any filters or post-processing of the segments. Better segments can be obtained using sophisticated hypergraph based methods (Kim et al., 2011).

In large datasets, for instance, in image segmentation, it is expensive to compute the 3-way tensor. So, we use the following approximation. We observe that in Algorithm 1, the matrix A can be approximated (upto a scaling factor) as

$$A_{ij} = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \mathbf{W}_{ijl} \approx \frac{1}{\sqrt{n}} \sum_{p=1}^{s} \mathbf{W}_{ijl_p}, \quad (18)$$

where we randomly select s out of n samples. Alternatively, this implies that one select s data points, and constructs s similarity matrices, each computed by fixing one data. A is approximated as a sum of these matrices.

Figure 3. **(left)** Original image; **(middle)** Segments obtained from spectral clustering; **(right)** Segments obtained from Algorithm 1.

5.3. Subspace (Line) Clustering

We now focus on problems, where higher-order clustering is required as pairwise relations are inappropriate in these cases. For instance, in subspace clustering, each cluster is formed by points that closely represent a subspace of dimension less than the data dimension. In this section, we conduct experiments on the line clustering problem, where each cluster is a one-dimensional subspace (line).

We generate three random lines in $[-1, 1]^5$, and sample 20 points from each line. The points are perturbed by Gaussian noise of standard deviation $\sigma = 0.02$ and 0.05, respectively. For each value of σ, 20 random examples are generated, and clustered using the higher-order approaches discussed in this paper. Two such instances are shown in Figure 4. A 3-uniform hypergraph is constructed with affinities among three points as $\exp\left(-\beta(\sigma_2^2+\sigma_3^2)\right)$, where σ_i is i^{th} singular value of the 5×3 matrix containing data vector in each column. Note that $(\sigma_2^2+\sigma_3^2)$ is the least squared error of fitting a line through these three points.

For clustering, we use non-negative tensor factorization (SNTF) (Shashua et al., 2006), game theoretic method with ℓ_1-norm constraint (HGT) (Rota Bulo & Pelillo, 2013), clique averaging (CA) (Agarwal et al., 2005) and Algorithm 1. All these algorithms solve the trace maximization problem (5). In addition, we also run HOSVD based algorithm (Ghoshdastidar & Dukkipati, 2014) and a hypergraph partitioning algorithm popular in VLSI community (hMETIS) (Karypis & Kumar, 2000). Table 2 shows the percentage errors incurred by each method. As noted before, Algorithm 1 is better than HOSVD. The relaxation in CA is similar to Algorithm 1, and so their performances are comparable. However, SNTF does not relax the problem (5), and hence, achieves minimum error. hMETIS is poor than CA and Algorithm 1. HGT is known to be good in removing outliers, but also labels true data as outliers.

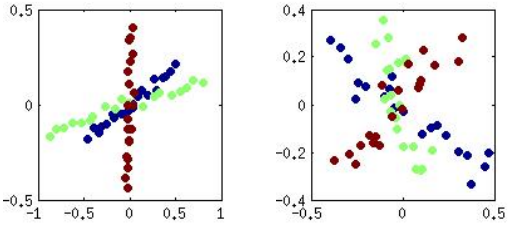

Figure 4. **(left)** 2-dimensional projection of three random lines in $[-1, 1]^5$ perturbed by Gaussian noise with $\sigma = 0.02$; **(right)** three lines perturbed by Gaussian noise with $\sigma = 0.05$.

Algorithm	Error ($\sigma = 0.02$)	Error ($\sigma = 0.05$)
SNTF	2.50	8.58
hMETIS	4.50	11.75
HGT	8.33	22.17
HOSVD	5.17	12.58
CA	3.33	10.92
Algorithm 1	3.25	10.33

Table 2. Mean percentage error for different algorithms.

A real-world application of subspace clustering is encountered in motion segmentation, where one needs to group different moving objects in a video. We conducted experiments on the Hopkins 155 database (Tron & Vidal, 2007), and the results are given in the supplementary material. The results show that Algorithm 1 performs better than most approaches, but best results are obtained by sampled variants of HOSVD (Jain & Govindu, 2013; Ghoshdastidar & Dukkipati, 2015). Thus, the accuracy of Algorithm 1 may be improved by using better sampling techniques.

6. Concluding Remarks

This paper provides a unified objective for partitioning uniform hypergraphs by maximizing the normalized associativity of the partition. This general idea appears to be at the heart of various higher-order clustering algorithms, but was not previously formalized in the literature. The above objective can be posed as a tensor trace maximization problem (Theorem 3) that is quite similar in spirit to the underlying problem of spectral clustering. Theorem 5 provides an almost sure error bound for the proposed Algorithm 1 in a random setting. The result shows higher-order clustering is consistent, and Algorithm 1 is provably better than the approach in (Ghoshdastidar & Dukkipati, 2014). Experiments validate this fact, and also provide insights into different formulations of the trace maximization problem.

In summary, this paper addresses two important aspects that has been crucial for the popularity of matrix spectral methods, but has been overlooked in the case of higher-order methods: **(1)** a well-defined objective for hypergraph partitioning, and **(2)** guarantees on the error incurred by tensor-based clustering. In the process of addressing above aspects, we also provide more general results. For instance, the operator norm of tensors are often used to quantify perturbations in recent literature (Anandkumar et al., 2014). Lemma 10 provides a bound on the tail probability of the operator norm that is applicable in other problems as well.

Acknowledgments

D. Ghoshdastidar is supported by Google Ph.D. Fellowship in Statistical Learning Theory.

References

Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., and Belongie, S. Beyond pairwise clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 838–845, 2005.

Agarwal, S., Branson, K., and Belongie, S. Higher order learning with graphs. In *Proceedings of the International Conference on Machine Learning*, pp. 17–24, 2006.

Anandkumar, A., Ge, R., Hsu, D., and Kakade, S. M. A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15: 2239–2312, 2014.

Arias-Castro, E., Chen, G., and Lerman, G. Spectral clustering based on local linear approximations. *Electronic Journal of Statistics*, 5:1537–1587, 2011.

Chen, B., He, S., Li, Z., and Zhang, S. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107, 2012.

Chung, F. R. K. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Comon, P. From source separation to blind equalization: Contrast based approaches. In *International Conference on Image and Signal Processing*, 2014.

Cooper, J. and Dutle, A. Spectra of uniform hypergraphs. *Linear Algebra and its Applications*, 436(9):3268–3292, 2012.

De Lathauwer, L., De Moor, B., and Vandewalle, J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Donath, W. E. and Hoffman, A. J. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

Duchenne, O., Bach, F., Kweon, I.-S., and Ponce, J. A tensor-based algorithm for high-order graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(12):2383–2395, 2011.

Frank, A. and Asuncion, A. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, 2010.

Ghoshdastidar, D. and Dukkipati, A. Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems*, 2014.

Ghoshdastidar, D. and Dukkipati, A. Spectral clustering using multilinear SVD: Analysis, approximations and applications. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

Govindu, V. M. A tensor decomposition for geometric grouping and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1150–1157, 2005.

Hein, M., Setzer, S., Jost, L., and Rangapuram, S. The total variation on hypergraphs-learning on hypergraphs revisited. In *Advances in Neural Information Processing Systems*, pp. 2427–2435, 2013.

Hu, S. and Qi, L. Algebraic connectivity of even uniform hypergraph. *Journal of Combinatorial Optimization*, 24: 564–579, 2012.

Ipsen, I. C. F. and Nadler, B. Refined perturbation bounds for eigenvalues of hermitian and non-hermitian matrices. *SIAM Journal on Matrix Analysis and Applications*, 31 (1):40–53, 2009.

Jain, S. and Govindu, V. M. Efficient higher-order clustering on the grassmann manifold. In *IEEE International Conference on Computer Vision*, 2013.

Karypis, G. and Kumar, V. Multilevel k-way hypergraph partitioning. *VLSI Design*, 11(3):285–300, 2000.

Kim, S., Nowozin, S., Kohli, P., and Yoo, C. D. Higher-order correlation clustering for image segmentation. In *Advances in Neural Information Processing Systems*, 2011.

Kumar, A., Sabharwal, Y., and Sen, S. A simple linear time $(1+\epsilon)$-approximation algorithm for geometric k-means clustering in any dimensions. In *Proceedings-Annual Symposium on Foundations of Computer Science*, 2004.

Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 556–562, 2001.

Lee, J., Cho, M., and Lee, K. M. Hyper-graph matching via reweighted random walks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

Leordeanu, M. and Sminchisescu, C. Efficient hypergraph clustering. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Lim, L.-H. Singular values and eigenvalues of tensors: a variational approach. In *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 129–132, 2005.

Liu, H., Latecki, L. J., and Yan, S. Robust clustering as ensembles of affinity relations. In *Advances in Neural Information Processing Systems*, pp. 1414–1422, 2010.

Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pp. 849–856, 2002.

Nguyen, N. H., Drineas, P., and Tran, T. D. Tensor sparsification via a bound on the spectral norm of random tensors. *arXiv preprint*, (arXiv:1005.4732), 2010.

Rohe, K., Chatterjee, S., and Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39(4):1878–1915, 2011.

Rota Bulo, S. and Pelillo, M. A game-theoretic approach to hypergraph clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(6):1312–1327, 2013.

Shashua, A., Zass, R., and Hazan, T. Multi-way clustering using super-symmetric non-negative tensor factorization. In *European Conference on Computer Vision*, pp. 595–608, 2006.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Stewart, G. W. and Sun, J. *Matrix Perturbation Theory*. Academic Press, 1990.

Tron, R. and Vidal, R. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

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

von Luxburg, U., Belkin, M., and Bousquet, O. Consistency of spectral clustering. *Annals of Statistics*, 36(2): 555–586, 2008.

Zass, R. and Shashua, A. Doubly stochastic normalization for spectral clustering. In *Advances in Neural Information Processing Systems*, 2006.

Zha, H., He, X., Ding, C., Simon, H., and Gu, M. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*, 2001.