Approximation Algorithms for Tensor Clustering

Stefanie Jegelka[1], Suvrit Sra[1], and Arindam Banerjee[2]

[1] Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany
{jegelka,suvrit}@tuebingen.mpg.de
[2] Univ. of Minnesota, Twin Cities, Minneapolis, MN, USA
banerjee@cs.umn.edu

Abstract. We present the *first* (to our knowledge) approximation algorithm for tensor clustering—a powerful generalization to basic 1D clustering. Tensors are increasingly common in modern applications dealing with complex heterogeneous data and clustering them is a fundamental tool for data analysis and pattern discovery. Akin to their 1D cousins, common tensor clustering formulations are NP-hard to optimize. But, unlike the 1D case no approximation algorithms seem to be known. We address this imbalance and build on recent co-clustering work to derive a tensor clustering algorithm with approximation guarantees, allowing metrics and divergences (e.g., Bregman) as objective functions. Therewith, we answer two open questions by Anagnostopoulos et al. (2008). Our analysis yields a constant approximation factor independent of data size; a worst-case example shows this factor to be tight for Euclidean co-clustering. However, empirically the approximation factor is observed to be conservative, so our method can also be used in practice.

1 Introduction

Tensor clustering is a recent generalization to the basic one-dimensional clustering problem, and it seeks to partition an order-m input tensor into coherent sub-tensors while minimizing some cluster quality measure [1, 2]. For example, in *co-clustering*, which is a special case of tensor clustering with $m = 2$, one simultaneously partitions rows and columns of an input matrix to obtain coherent submatrices, often while minimizing a Bregman divergence [3, 4].

Being generalizations of the 1D case, common tensor clustering formulations are also NP-hard to optimize. But despite the existence of a vast body of research on approximation algorithms for 1D clustering problems (e.g., [5–10]), there seem to be *no* published approximation algorithms for tensor clustering. Even for (2D) co-clustering, there are only two recent attempts [11] and [12] (from 2008). Both prove an approximation factor of $2\alpha_1$ for Euclidean co-clustering given an α_1-approximation for k-means, and show constant approximation factors for ℓ_1 ([12] only for binary matrices) and ℓ_p-norm [11] based variants.

Tensor clustering is a basic data analysis task with growing importance; several domains now deal frequently with tensor data, e.g., data mining [13], computer graphics [14], and computer vision [2]. We refer the reader to [15]

for a recent survey about tensors and their applications. The simplest tensor clustering scenario, namely, co-clustering (also known as bi-clustering) is more established [12, 4, 16–18]. Tensor clustering is less well known, though several researchers have considered it before [1, 2, 19–21].

1.1 Contributions

The main contribution of this paper is the analysis of an approximation algorithm for tensor clustering that achieves an approximation ratio of $O(p(m)\alpha)$, where m is the order of the tensor, $p(m) = m$ or $p(m) = m^{\frac{1}{\log_3 2}}$, and α is the approximation factor of a corresponding 1D clustering algorithm. Our results apply to a fairly broad class of objective functions, including metrics such as ℓ_p norms, Hilbertian metrics [22, 23], and divergence functions such as Bregman divergences [24] (with some assumptions). As corollaries, our results solve two open problems posed by [12], viz., whether their methods for Euclidean co-clustering could be extended to Bregman co-clustering, and if one could extend the approximation guarantees to tensor clustering. The bound also gives insight into properties of the tensor clustering problem. We give an example for the tightness of our bound for squared Euclidean distance, and provide an experimental validation of the theoretical claims, which forms an additional contribution.

2 Background & Problem

Traditionally, "center" based clustering algorithms seek partitions of columns of an input matrix $\mathsf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ into clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, and find "centers" $\boldsymbol{\mu}_k$ that minimize the objective

$$J(\mathcal{C}) = \sum\nolimits_{k=1}^{K} \sum\nolimits_{\boldsymbol{x} \in \mathcal{C}_k} d(\boldsymbol{x}, \boldsymbol{\mu}_k), \tag{2.1}$$

where the function $d(\boldsymbol{x}, \boldsymbol{y})$ measures cluster quality. The "center" $\boldsymbol{\mu}_k$ of cluster $\mathcal{C}_k$ is given by the mean of the points in $\mathcal{C}_k$ when $d(\boldsymbol{x}, \boldsymbol{y})$ is a Bregman divergence [25]. Co-clustering extends (2.1) to seek simultaneous partitions (and centers $\boldsymbol{\mu}_{IJ}$) of rows and columns of X, so that the objective function

$$J(\mathcal{C}) = \sum\nolimits_{I,J} \sum\nolimits_{i \in I, j \in J} d(x_{ij}, \boldsymbol{\mu}_{IJ}), \tag{2.2}$$

is minimized; $\boldsymbol{\mu}_{IJ}$ denotes the (scalar) "center" of the cluster described by the row and column index sets, viz., I and J. We generalize formulation (2.2) to tensors in Section 2.2 after introducing some background on tensors.

2.1 Tensors

An order-m tensor A may be viewed as an element of the vector space $\mathbb{R}^{n_1 \times \ldots \times n_m}$. An individual entry of A is given by the multiply-indexed value $a_{i_1 i_2 \ldots i_m}$, where $i_j \in \{1, \ldots, n_j\}$ for $1 \leq j \leq m$. For us, the most important tensor operation is **multilinear matrix multiplication**, which generalizes matrix multiplication [26]. Matrices *act* on other matrices by either left or right multiplication. Similarly, for an order-m tensor, there are m dimensions on which a matrix may

act. For $\mathsf{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, and matrices $\mathsf{P}_1 \in \mathbb{R}^{p_1 \times n_1}$, ..., $\mathsf{P}_m \in \mathbb{R}^{p_m \times n_m}$, *multilinear multiplication* is defined by the action of the P_i on the different dimensions of A, and is denoted by $\mathsf{A}' = (\mathsf{P}_1, \ldots, \mathsf{P}_m) \cdot \mathsf{A} \in \mathbb{R}^{p_1 \times \cdots \times p_m}$. The individual components of A' are given by $a'_{i_1 i_2 \ldots i_m} = \sum_{j_1, \ldots, j_m = 1}^{n_1, \ldots, n_m} p^{(1)}_{i_1 j_1} \cdots p^{(m)}_{i_m j_m} a_{j_1 \ldots j_m}$, where $p^{(k)}_{ij}$ denotes the ij-th entry of matrix P_k. The *inner product* between two tensors A and B is defined as

$$\langle \mathsf{A}, \mathsf{B} \rangle = \sum\nolimits_{i_1, \ldots, i_m} a_{i_1 \ldots i_m} b_{i_1 \ldots i_m}, \tag{2.3}$$

and this inner product satisfies the following natural property (which generalizes the familiar $\langle \mathsf{Ax}, \mathsf{By} \rangle = \langle \mathsf{x}, \mathsf{A}^\top \mathsf{By} \rangle$):

$$\langle (\mathsf{P}_1, \ldots, \mathsf{P}_m) \cdot \mathsf{A}, (\mathsf{Q}_1, \ldots, \mathsf{Q}_m) \cdot \mathsf{B} \rangle = \langle \mathsf{A}, (\mathsf{P}_1^\top \mathsf{Q}_1, \ldots, \mathsf{P}_m^\top \mathsf{Q}_m) \cdot \mathsf{B} \rangle. \tag{2.4}$$

Moreover, the *Frobenius norm* is $\|\mathsf{A}\|^2 = \langle \mathsf{A}, \mathsf{A} \rangle$. Finally, we define an arbitrary *divergence* function $d(\mathsf{X}, \mathsf{Y})$ as an elementwise sum of individual divergences, i.e.,

$$d(\mathsf{X}, \mathsf{Y}) = \sum\nolimits_{i_1, \ldots, i_m} d(x_{i_1, \ldots, i_m}, y_{i_1, \ldots, i_m}), \tag{2.5}$$

and we will define the scalar divergence $d(x, y)$ as the need arises.

2.2 Problem Formulation

Let $\mathsf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ be an order-m tensor that we wish to partition into coherent sub-tensors (or clusters). In 3D, we divide a cube into smaller cubes by cutting orthogonal to (i.e., along) each dimension (Fig. 1). A basic approach is to minimize the sum of the divergences between individual (scalar) elements in each cluster to their corresponding (scalar) cluster "centers". Readers familiar with [4] will recognize this to be a "block-average" variant of tensor clustering.

Assume that each dimension j ($1 \le j \le m$) is partitioned into k_j clusters. Let $\mathsf{C}_j \in \{0, 1\}^{n_j \times k_j}$ be the cluster indicator matrix for dimension j, where the ik-th entry of such a matrix is one if and only if index i belongs to the k-th cluster ($1 \le k \le k_j$) for dimension j. Then, the *tensor clustering* problem is (*cf.* 2.2):

$$\underset{\mathsf{C}_1, \ldots, \mathsf{C}_m, \mathsf{M}}{\text{minimize}} \quad d(\mathsf{A},\ (\mathsf{C}_1, \ldots, \mathsf{C}_m) \cdot \mathsf{M}), \quad \text{s.t. } \mathsf{C}_j \in \{0, 1\}^{n_j \times k_j}, \tag{2.6}$$

where the tensor M collects all the cluster "centers."

3 Algorithm and Analysis

Given formulation (2.6), our algorithm, which we name **Co**mbination **Te**nsor **C**lustering (CoTeC), follows the simple outline:

1. Cluster along each dimension j, using an approximation algorithm to obtain clustering C_j; Let $\boldsymbol{C} = (\mathsf{C}_1, \ldots, \mathsf{C}_m)$
2. Compute $\mathsf{M} = \operatorname{argmin}_{\mathsf{X} \in \mathbb{R}^{k_1 \times \cdots \times k_m}} d(\mathsf{A}, \boldsymbol{C} \cdot \mathsf{X})$.
3. Return the tensor clustering $(\mathsf{C}_1, \ldots, \mathsf{C}_m)$ (with representatives M).

Remark 1: Instead of clustering one dimension at a time in Step 1, we can also cluster along t dimensions simultaneously. In such a *t-dimensional clustering* of an order-m tensor, we form groups of order-$(m - t)$ tensors.

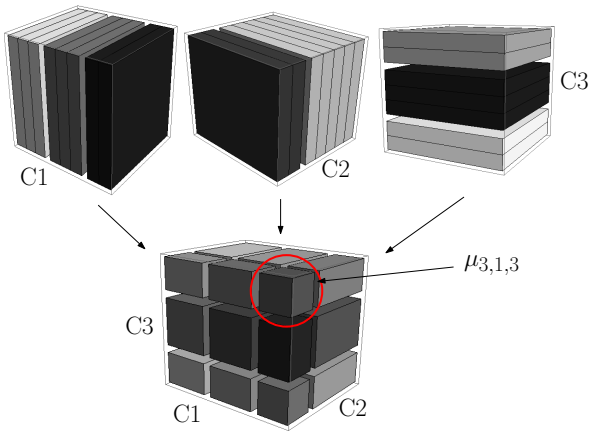


Fig. 1. CoTeC: Cluster along dimensions one (C1), two (C2), three (C3) separately and combine the results; $\mu_{3,1,3}$ is the mean of sub-tensor (cluster) (3,1,3); The various clusters in the final tensor clustering are color coded to indicate combination of contributions from clusters along each dimension.

Our algorithm might be counterintuitive to some readers as merely clustering along individual dimensions and then combining the results is against the idea of "co"-clustering, where one *simultaneously* clusters along different dimensions. However, our analysis shows that dimension-wise clustering suffices to obtain strong approximation guarantees for tensor clustering—a fact often observed empirically too. It is also easy to see that CoTeC runs in time $O((m/t)T(t))$, if the subroutine for dimension-wise clustering takes $T(t)$ time.

The main contribution of this paper is the following approximation guarantee for CoTeC, which we prove in the remainder of this section.

Theorem 1 (Approximation). *Let* A *be an order-*m *tensor and let* $\mathcal{C}_j$ *denote its clustering along the* j*th subset of* t *dimensions (*$1 \leq j \leq m/t$*), as obtained from a multiway clustering algorithm with guarantee* α_t[3]*. Let* $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_{m/t})$ *denote the induced tensor clustering, and* $J_{OPT}(m)$ *the best* m*-dimensional clustering. Then,*

$$J(\mathcal{C}) \leq p(m/t)\rho_d\alpha_t J_{OPT}(m), \quad \textit{with} \tag{3.1}$$

1. $\rho_d = 1$ *and* $p(m/t) = 2^{\log_2 m/t}$ *if* $d(x,y) = (x-y)^2$,
2. $\rho_d = 1$ *and* $p(m/t) = 3^{\log_2 m/t}$ *if* $d(x,y)$ *is a metric*[4].

Thm. 1 is quite general, and it can be combined with some natural assumptions (see §3.3) to yield results for tensor clustering with general divergence functions (though ρ_d might be greater than 1). For particular choices of d one can perhaps derive tighter bounds, though for squared Euclidean distances, we provide an explicit example (Fig. 2) that shows the bound to be tight in 2D.

3.1 Analysis: Theorem 1, Euclidean Case

We begin our proof with the Euclidean case, i.e., $d(x,y) = (x-y)^2$. Our proof is inspired by the techniques of [12]. We establish that given a clustering algorithm

[3] We say an approximation algorithm has guarantee α if it yields a solution that achieves an objective value within a factor $O(\alpha)$ of the optimum.

[4] The results can be trivially extended to λ-relaxed metrics that satisfy $d(x,y) \leq \lambda(d(x,z) + d(z,y))$; the corresponding approximation factor just gets scaled by λ.

which clusters along t of the m dimensions at a time[5] with an approximation factor of α_t, CoTeC achieves an objective within a factor $O(\lceil m/t \rceil \alpha_t)$ of the optimal. For example, for $t = 1$ we can use the seeding methods of [8, 9] or the stronger approximation algorithms of [5]. We assume without loss of generality (wlog) that $m = 2^h t$ for an integer h (otherwise, pad in empty dimensions).

Since for the squared Frobenius norm, each cluster "center" is given by the mean, we can recast Problem (2.6) into a more convenient form. To that end, note that the individual entries of the means tensor M are given by (*cf.* (2.2))

$$M_{I_1 \dots I_m} = \frac{1}{|I_1| \cdots |I_m|} \sum\nolimits_{i_1 \in I_1, \dots, i_m \in I_m} a_{i_1 \dots i_m}, \tag{3.2}$$

with index sets I_j for $1 \le j \le m$. Let $\overline{\mathsf{C}}_j$ be the normalized cluster indicator matrix obtained by normalizing the columns of C_j, so that $\overline{\mathsf{C}}_j^\top \overline{\mathsf{C}}_j = \mathsf{I}_{k_j}$. Then, we can rewrite (2.6) in terms of projection matrices P_j as:

$$\underset{\mathcal{C}=(\overline{\mathsf{C}}_1, \dots, \overline{\mathsf{C}}_m)}{\text{minimize}} \quad J(\mathcal{C}) = \|\mathsf{A} - (\mathsf{P}_1, \dots, \mathsf{P}_m) \cdot \mathsf{A}\|^2, \quad \text{s.t. } \mathsf{P}_j = \overline{\mathsf{C}}_j \overline{\mathsf{C}}_j^\top. \tag{3.3}$$

Lemma 1 (Pythagorean). *Let* $\boldsymbol{P} = (\mathsf{P}_1, \dots, \mathsf{P}_t)$, $\boldsymbol{P}^\perp = (\mathsf{I} - \mathsf{P}_1, \dots, \mathsf{I} - \mathsf{P}_t)$ *be* collections *of projection matrices* P_j*, and* $\boldsymbol{S}$ *and* $\boldsymbol{R}$ *be arbitrary collections of* $m - t$ *projection matrices. Then,*

$$\|(\boldsymbol{P}, \boldsymbol{S}) \cdot \mathsf{A} + (\boldsymbol{P}^\perp, \boldsymbol{R}) \cdot \mathsf{B}\|^2 = \|(\boldsymbol{P}, \boldsymbol{S}) \cdot \mathsf{A}\|^2 + \|(\boldsymbol{P}^\perp, \boldsymbol{R}) \cdot \mathsf{B}\|^2.$$

Proof. Using $\|\mathsf{A}\|^2 = \langle \mathsf{A}, \mathsf{A} \rangle$ we can rewrite the l.h.s. as

$$\|(\boldsymbol{P}, \boldsymbol{S}) \cdot \mathsf{A} + (\boldsymbol{P}^\perp, \boldsymbol{R}) \cdot \mathsf{B}\|^2 = \|(\boldsymbol{P}, \boldsymbol{S}) \cdot \mathsf{A}\|^2 + \|(\boldsymbol{P}^\perp, \boldsymbol{R}) \cdot \mathsf{B}\|^2 + 2\big\langle (\boldsymbol{P}, \boldsymbol{S}) \cdot \mathsf{A}, (\boldsymbol{P}^\perp, \boldsymbol{R}) \cdot \mathsf{B} \big\rangle,$$

from which the last term is immediately seen to be zero using Property (2.4) and the fact that $\mathsf{P}_j^\top \mathsf{P}_j^\perp = \mathsf{P}_j(\mathsf{I} - \mathsf{P}_j) = \mathbf{0}$. □

Some more notation: Since we cluster along t dimensions at a time, we recursively partition the initial set of all m dimensions until (after $\log(m/t) + 1$ steps), the sets of dimensions have length t. Let l denote the level of recursion, starting at $l = \log(m/t) = h$ and going down to $l = 0$. At level l, the sets of dimensions will have length $2^l t$ (so that for $l = 0$ we have t dimensions). We represent each clustering along a subset of $2^l t$ dimensions by its corresponding $2^l t$ projection matrices. We gather these projection matrices into the collection $\boldsymbol{P}_i^l$ (note boldface), where the index i ranges from 1 to 2^{h-l}.

We also need some notation to represent a complete tensor clustering along all m dimensions, where *only a subset* of $2^l t$ dimensions are clustered. We pad the collection $\boldsymbol{P}_i^l$ with $m - 2^l t$ identity matrices for the non-clustered dimensions, and call this padded collection $\boldsymbol{Q}_i^l$. With recursive partitioning of the dimensions, $\boldsymbol{Q}_i^l$ subsumes $\boldsymbol{Q}_j^0$ for $2^l(i-1) < j \le 2^l i$, i.e.,

$$\boldsymbol{Q}_i^l = \prod\nolimits_{j=2^l(i-1)+1}^{2^l i} \boldsymbol{Q}_j^0.$$

[5] One could also consider clustering differently sized subsets of the dimensions, say $\{t_1, \dots, t_r\}$, where $t_1 + \cdots + t_r = m$. However, this requires unilluminating notational jugglery, which we can skip for simplicity of exposition.

At level 0, the algorithm yields the collections $\boldsymbol{Q}_i^0$ and $\boldsymbol{P}_i^0$. The remaining clusterings are simply *combinations*, i.e., products of these level-0 clusterings. We denote the collection of $m - 2^l t$ identity matrices (of appropriate size) by $\boldsymbol{I}^l$, so that $\boldsymbol{Q}_1^l = (\boldsymbol{P}_1^l, \boldsymbol{I}^l)$. Accoutered with our notation, we now prove the main lemma that relates the combined clustering to its sub-clusterings.

Lemma 2. *Let* A *be an order-*m *tensor and* $m \geq 2^l t$*. The objective function for any* $2^l t$*-dimensional clustering* $\boldsymbol{P}_i^l = (\boldsymbol{P}_{2^l(i-1)+1}^0, \ldots, \boldsymbol{P}_{2^l i}^0)$ *can be bound via the sub-clusterings along only one set of dimensions of size* t *as*

$$\|\mathsf{A} - \boldsymbol{Q}_i^l \cdot \mathsf{A}\|^2 \leq \max_{2^l(i-1) < j \leq 2^l i} 2^l \|\mathsf{A} - \boldsymbol{Q}_j^0 \cdot \mathsf{A}\|^2. \tag{3.4}$$

We can always (wlog) permute dimensions so that any set of 2^l clustered dimensions maps to the first 2^l ones. Hence, it suffices to prove the lemma for $i = 1$, i.e., the first 2^l dimensions.

Proof. We prove the lemma for $i = 1$ by induction on l.

Base: Let $l = 0$. Then $\boldsymbol{Q}_1^l = \boldsymbol{Q}_1^0$, and (3.4) holds trivially.

Induction: Assume the claim holds for $l \geq 0$. Consider a clustering $\boldsymbol{P}_1^{l+1} = (\boldsymbol{P}_1^l, \boldsymbol{P}_2^l)$, or equivalently $\boldsymbol{Q}_1^{l+1} = \boldsymbol{Q}_1^l \boldsymbol{Q}_2^l$. Using $\boldsymbol{P} + \boldsymbol{P}^\perp = \boldsymbol{I}$, we decompose A as

$$\begin{aligned}
\mathsf{A} &= (\boldsymbol{P}_1^{l+1} + \boldsymbol{P}_1^{l+1\perp}, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} \quad = \quad (\boldsymbol{P}_1^l + \boldsymbol{P}_1^{l\perp}, \boldsymbol{P}_2^l + \boldsymbol{P}_2^{l\perp}, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} \\
&= (\boldsymbol{P}_1^l, \boldsymbol{P}_2^l, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} + (\boldsymbol{P}_1^{l\perp}, \boldsymbol{P}_2^l, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} + (\boldsymbol{P}_1^l, \boldsymbol{P}_2^{l\perp}, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} + (\boldsymbol{P}_1^{l\perp}, \boldsymbol{P}_2^{l\perp}, \boldsymbol{I}^{l+1}) \cdot \mathsf{A} \\
&= \boldsymbol{Q}_1^l \boldsymbol{Q}_2^l \cdot \mathsf{A} + \boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^l \cdot \mathsf{A} + \boldsymbol{Q}_1^l \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A} + \boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A},
\end{aligned}$$

where $\boldsymbol{Q}_1^{l\perp} = (\boldsymbol{P}_1^{l\perp}, \boldsymbol{I}^l)$. Since $\boldsymbol{Q}_1^{l+1} = \boldsymbol{Q}_1^l \boldsymbol{Q}_2^l$, the Pythagorean Property 1 yields

$$\|\mathsf{A} - \boldsymbol{Q}_1^{l+1} \cdot \mathsf{A}\|^2 = \|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^l \cdot \mathsf{A}\|^2 + \|\boldsymbol{Q}_1^l \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A}\|^2 + \|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A}\|^2.$$

Combining the above equalities with the assumption (wlog) $\|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^l \cdot \mathsf{A}\|^2 \geq \|\boldsymbol{Q}_1^l \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A}\|^2$, we obtain the inequalities

$$\begin{aligned}
\|\mathsf{A} - \boldsymbol{Q}_1^l \boldsymbol{Q}_2^l \cdot \mathsf{A}\|^2 &\leq 2\big(\|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^l \cdot \mathsf{A}\|^2 + \|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A}\|^2\big) \\
&= 2\|\boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^l \cdot \mathsf{A} + \boldsymbol{Q}_1^{l\perp} \boldsymbol{Q}_2^{l\perp} \cdot \mathsf{A}\|^2 \;=\; 2\|\boldsymbol{Q}_1^{l\perp} (\boldsymbol{Q}_2^l + \boldsymbol{Q}_2^{l\perp}) \cdot \mathsf{A}\|^2 \\
&= 2\|\boldsymbol{Q}_1^{l\perp} \cdot \mathsf{A}\|^2 \;=\; 2\|\mathsf{A} - \boldsymbol{Q}_1^l \cdot \mathsf{A}\|^2 \\
&\leq 2 \max_{1 \leq j \leq 2^l} \|\mathsf{A} - \boldsymbol{Q}_j^l \cdot \mathsf{A}\|^2 \;\leq\; 2 \cdot 2^l \max_{1 \leq j \leq 2^{l+1}} \|\mathsf{A} - \boldsymbol{Q}_j^0 \cdot \mathsf{A}\|^2,
\end{aligned}$$

where the last step follows from the induction hypothesis (3.4), and the two norm terms in the first line are combined using the Pythagorean Property. □

Proof. (*Thm. 1, Case 1*). Let $m = 2^h t$. Using an algorithm with guarantee α_t, we cluster each subset (indexed by i) of t dimensions to obtain $\boldsymbol{Q}_i^0$. Let $\boldsymbol{S}_i$ be the optimal sub-clustering of subset i, i.e., the result that $\boldsymbol{Q}_i^0$ would be if α_t were 1. We bound the objective for the collection of all m sub-clusterings $\boldsymbol{P}_1^h = \boldsymbol{Q}_1^h$ as

$$\|\mathsf{A} - \boldsymbol{Q}_1^h \cdot \mathsf{A}\|^2 \leq 2^h \max_j \|\mathsf{A} - \boldsymbol{Q}_j^0 \cdot \mathsf{A}\|^2 \leq 2^h \alpha_t \max_j \|\mathsf{A} - \boldsymbol{S}_j \cdot \mathsf{A}\|^2. \tag{3.5}$$

The first inequality follows from Lemma 2, while the last inequality follows from the α_t approximation factor that we used to get sub-clustering $\boldsymbol{Q}_j^0$.

So far we have related our approximation to an optimal sub-clustering along a set of dimensions. Let us hence look at the relation between such an optimal sub-clustering $\boldsymbol{S}$ of the first t dimensions (via permutation, these dimensions correspond to an arbitrary subset of size t), and the optimal tensor clustering $\boldsymbol{F}$ along all the $m = 2^h t$ dimensions. Recall that a clustering can be expressed by either the projection matrices collected in $\boldsymbol{Q}_1^l$, or by cluster indicator matrices C_i together with the mean tensor M, so that

$$(\mathsf{C}_1, \ldots, \mathsf{C}_{2^l t}, \boldsymbol{I}^l) \cdot \mathsf{M} = \boldsymbol{Q}_1^l \cdot \mathsf{A}.$$

Let C_j^S and C_j^F be the dimension-wise cluster indicator matrices for $\boldsymbol{S}$ and $\boldsymbol{F}$, respectively. By definition, $\boldsymbol{S}$ solves

$$\min_{\mathsf{C}_1,\ldots,\mathsf{C}_t,\mathsf{M}} \|\mathsf{A} - (\mathsf{C}_1, \ldots, \mathsf{C}_t, \boldsymbol{I}^0) \cdot \mathsf{M}\|^2, \quad \text{s.t. } \mathsf{C}_j \in \{0,1\}^{n_j \times k_j},$$

which makes $\boldsymbol{S}$ even better than the sub-clustering $(\mathsf{C}_1^F, \ldots, \mathsf{C}_t^F)$ induced by the optimal m-dimensional clustering $\boldsymbol{F}$. Thus,

$$\begin{aligned} \|\mathsf{A} - \boldsymbol{S} \cdot \mathsf{A}\|^2 &\leq \min_{\mathsf{M}} \|\mathsf{A} - (\mathsf{C}_1^F, \ldots, \mathsf{C}_t^F, \boldsymbol{I}^0) \cdot \mathsf{M}\|^2 \\ &\leq \|\mathsf{A} - (\mathsf{C}_1^F, \ldots, \mathsf{C}_t^F, \boldsymbol{I}^0)(\mathsf{I}, \ldots, \mathsf{I}, \mathsf{C}_{t+1}^F, \ldots, \mathsf{C}_m^F) \cdot \mathsf{M}^F\|^2 \\ &= \|\mathsf{A} - \boldsymbol{F} \cdot \mathsf{A}\|^2, \end{aligned} \tag{3.6}$$

where M^F is the tensor of means for the optimal m-dimensional clustering. Combining (3.5) with (3.6) yields the final bound for the combined clustering $\mathcal{C} = \boldsymbol{Q}_1^h$,

$$J_m(\mathcal{C}) = \|\mathsf{A} - \boldsymbol{Q}_1^h \cdot \mathsf{A}\|^2 \leq 2^h \alpha_t \|\mathsf{A} - \boldsymbol{F} \cdot \mathsf{A}\|^2 = 2^h \alpha_t J_{\mathrm{OPT}}(m),$$

which completes the proof of the theorem. □

Tightness of Bound: How tight is the bound for CoTeC implied by Thm. 1? The following example shows that for Euclidean co-clustering, i.e., $m = 2$, the bound is tight. Specifically, for every $0.25 > \gamma > 0$, there exists a matrix for which the approximation is as bad as $J(\mathcal{C}) = (m - \gamma) J_{\mathrm{OPT}}(m)$.

Let ϵ be such that $\gamma = 2\epsilon(1+\epsilon)^{-2}$. The optimal 1D row clustering $\mathcal{C}_1$ for the matrix in Figure 2 groups rows $\{1,2\}$ and $\{3,4\}$ together, and the optimal column clustering is $\mathcal{C}_2 = (\{a,b\},\{c,d\})$. The co-clustering loss for the combination is $J_2(\mathcal{C}_1, \mathcal{C}_2) = 8 + 8\epsilon^2$. The optimal co-clustering, grouping columns $\{a,d\}$ and $\{b,c\}$ (and rows as $\mathcal{C}_2$) achieves an objective of $J_{\mathrm{OPT}}(2) =$

	a	b	c	d
1	$-\epsilon$	-1	ϵ	1
2	1	ϵ	-1	$-\epsilon$
3	$10-\epsilon$	9	$10+\epsilon$	11
4	11	$10+\epsilon$	9	$10-\epsilon$

Fig. 2. Matrix with co-clustering approximation factor $2 - 2\epsilon(1+\epsilon)^{-2}$.

$4(1+\epsilon)^2$. Relating these results, we get $J_2(\mathcal{C}_1, \mathcal{C}_2) = (2-\gamma)J_{\mathrm{OPT}}(m)$. However, this example is a worst-case scenario; the average factor is much better in practice, as revealed by our experiments (§4). The latter combined with the structure of this negative example suggest that with some assumptions on the data, one can probably obtain tighter bounds. Also note that the bound holds for a CoTeC -like scheme treating dimensions separately, but not necessarily for *all* approximation algorithms.

3.2 Analysis: Theorem 1, Metric Case

Now we present our proof of Thm. 1 for the case where $d(x, y)$ is a metric. For this case, recall that the tensor clustering problem is

$$\underset{(\mathsf{C}_1,\ldots,\mathsf{C}_m),\mathsf{M}}{\text{minimize}}\; J(\mathcal{C}) = d(\mathsf{A}, (\mathsf{C}_1, \ldots, \mathsf{C}_m) \cdot \mathsf{M}), \quad \text{s.t. } \mathsf{C}_j \in \{0,1\}^{n_j \times k_j}. \tag{3.7}$$

Since in general the best representative M is not the mean tensor, we cannot use the shorthand $\boldsymbol{P} \cdot \mathsf{A}$ for M, so the proof is different from the Euclidean case.

The following lemma is the basis of the induction for this case of Thm. 1.

Lemma 3. *Let A be of order, $m = 2^h t$, and $\boldsymbol{R}_i^l$ the clustering of the i-th subset of $2^l t$ dimensions (for $l < h$) with an approximation guarantee of $\alpha_{2^l t}$—$\boldsymbol{R}_i^l$ combines the $\mathcal{C}_j$ in a manner analogous to how $\boldsymbol{Q}_i^l$ combines projection matrices. Then the combination $\boldsymbol{R}^{l+1} = \boldsymbol{R}_i^l \boldsymbol{R}_j^l$, $i \neq j$, satisfies*

$$\min_{\mathsf{M}}\; d(\mathsf{A}, \boldsymbol{R}^{l+1} \cdot \mathsf{M}) \;\leq\; 3\alpha_{2^l t} \min_{\mathsf{M}} d(\mathsf{A}, \boldsymbol{F}^{l+1} \cdot \mathsf{M}),$$

where $\boldsymbol{F}^{l+1}$ is the optimal joint clustering of the dimensions covered by $\boldsymbol{R}^{l+1}$ (as before, we always assume that $\boldsymbol{R}_i^l$ and $\boldsymbol{R}_j^l$ cover disjoint subsets of dimensions).

Proof. Without loss of generality, we prove the lemma for $\boldsymbol{R}_1^{l+1} = \boldsymbol{R}_1^l \boldsymbol{R}_2^l$. Let $\mathsf{M}_i^l = \operatorname{argmin}_{\mathsf{X}} d(\mathsf{A}, \boldsymbol{R}_i^l \cdot \mathsf{X})$ be the associated representatives for $i = 1, 2$, and $\boldsymbol{S}_i^l$ the optimal 2^l-dimensional clusterings. Further let $\boldsymbol{F}_1^{l+1} = \boldsymbol{F}_1^l \boldsymbol{F}_2^l$ be the optimal 2^{l+1}-dimensional clustering. The following step is vital in relating objective values of $\boldsymbol{R}_1^{l+1}$ and $\boldsymbol{S}_i^l$. The optimal sub-clusterings will eventually be bounded by the objective of the optimal $\boldsymbol{F}_1^{l+1}$. Let $L = 2^{l+1}$, and

$$\widehat{\mathsf{M}} = \operatorname*{argmin}_{\mathsf{X}} d(\boldsymbol{R}_1^l \mathsf{M}_1^l, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \mathsf{X}), \quad \mathsf{X} \in \mathbb{R}^{k_1 \times \ldots \times k_L \times n_{L+1} \ldots \times n_m}.$$

Let i,j be multi-indices running over dimensions 1 to 2^l, and $2^l + 1$ to 2^{l+1}, respectively; let r be the multi-index covering the remaining $m - L$ dimensions. The multi-indices of the clusters defined by $\boldsymbol{R}_1^l$ and $\boldsymbol{R}_2^l$, respectively, are I and J. Since $\widehat{\mathsf{M}}$ is the element-wise minimum, we have

$$\begin{aligned} d(\boldsymbol{R}_1^l \cdot \mathsf{M}_1^l, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \widehat{\mathsf{M}}) &= \sum_{I,J} \sum_{i \in I, r} \min_{\mu_{IJr} \in \mathbb{R}} \sum_{j \in J} d((\mu_1^l)_{Ijr}, \mu_{IJr}) \\ \leq \sum_{I,J} \sum_{i \in I, r} \sum_{j \in J} d((\mu_1^l)_{Ijr}, (\mu_2^l)_{iJr}) \quad &= \quad d(\boldsymbol{R}_1^l \cdot \mathsf{M}_1^l, \boldsymbol{R}_2^l \cdot \mathsf{M}_2^l). \end{aligned}$$

Using this relation and the triangle inequality, we can now relate the objectives for the combined clustering and for the optimal sub-clusterings:

$$\begin{aligned}\min_{\mathsf{M}^{l+1}} \; & d(\mathsf{A}, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \mathsf{M}^{l+1}) \leq d(\mathsf{A}, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \widehat{\mathsf{M}}) \\ & \leq d(\mathsf{A}, \boldsymbol{R}_1^l \cdot \mathsf{M}_1^l) + d(\boldsymbol{R}_1^l \cdot \mathsf{M}_1^l, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \widehat{\mathsf{M}}) \\ & \leq d(\mathsf{A}, \boldsymbol{R}_1^l \cdot \mathsf{M}_1^l) + d(\boldsymbol{R}_1^l \cdot \mathsf{M}_1^l, \boldsymbol{R}_2^l \cdot \mathsf{M}_2{}^l) \\ & \leq 2d(\mathsf{A}, \boldsymbol{R}_1^l \cdot \mathsf{M}_1^l) + d(\mathsf{A}, \boldsymbol{R}_2^l \cdot \mathsf{M}_2^l) \\ & \leq 2\alpha_{2^l t} \min_{\mathsf{X}_1} d(\mathsf{A}, \boldsymbol{S}_1^l \cdot \mathsf{X}_1) + \alpha_{2^l t} \min_{\mathsf{X}_2} d(\mathsf{A}, \boldsymbol{S}_2^l \cdot \mathsf{X}_2). \end{aligned} \tag{3.8}$$

However, owing to the optimality of $\boldsymbol{S}_1^l$, we have

$$\min_{\mathsf{X}_1^l} \; d(\mathsf{A}, \boldsymbol{S}_1^l \cdot \mathsf{X}_1^l) \;\leq\; \min_{\mathsf{Y}^l} \; d(\mathsf{A}, \boldsymbol{F}_1^l \cdot \mathsf{Y}^l) \;\leq\; \min_{\mathsf{Y}^{l+1}} \; d(\mathsf{A}, \boldsymbol{F}_1^l \boldsymbol{F}_2^l \cdot \mathsf{Y}^{l+1}),$$

and analogously for $\boldsymbol{S}_2^l$. Plugging this inequality into (3.8) we get

$$\min_{\mathsf{M}^{l+1}} d(\mathsf{A}, \boldsymbol{R}_1^l \boldsymbol{R}_2^l \cdot \mathsf{M}^{l+1}) \;\leq\; 3\alpha_{2^l t} \min_{\mathsf{Y}^{l+1}} \; d(\mathsf{A}, \boldsymbol{F}_1^l \boldsymbol{F}_2^l \cdot \mathsf{Y}^{l+1}) = 3\alpha_{2^l t} \min_{\mathsf{Y}^{l+1}} d(\mathsf{A}, \boldsymbol{F}_1^{l+1} \cdot \mathsf{Y}^{l+1}). \square$$

Proof. (*Thm. 1, Case 2*). Given Lemma 3, the proof of Thm. 1 for the metric case follows easily by induction if we hierarchically combine the sub-clusterings and use $\alpha_{2^{l+1}t} = 3\alpha_{2^l t}$, for $l \geq 0$, as stated by the lemma. $\square$

3.3 Implications

We now mention several important implications of Theorem 1.

Clustering with Bregman divergences. Bregman divergence based clustering and co-clustering are well-studied problems [25, 4]. Here, the function $d(x, y)$ is parametrized by a strictly convex function f [24], so that $d(x, y) = B_f(x, y) = f(x) - f(y) - f'(y)(x - y)$. Under the assumption (also see [5, 6])

$$\sigma_L \|x - y\|^2 \leq B_f(x, y) \leq \sigma_U \|x - y\|^2, \tag{3.9}$$

on the curvature of the divergence $B_f(x, y)$, we can invoke Thm. 1 with $\rho_d = \sigma_U / \sigma_L$ The proofs are omitted for brevity, and may be found in [27]. We would like to stress that such curvature bounds seem to be necessary to guarantee *constant* approximation factors for the underlying 1D clustering—this intuition is reinforced by the results of [28], who avoided such curvature assumptions and had to be content with a *non-constant* $O(\log n)$ approximation factor for information theoretic clustering.

Clustering with ℓ_p-norms. Thm. 1 (metric case) immediately yields approximation factors for clustering with ℓ_p-norms. We note that for binary matrices, using $t = 2$ and the results of [11] we can obtain the slightly stronger guarantee

$$J(\mathcal{C}) \leq 3^{\log_2(m)-1}(1 + \sqrt{2})\alpha_1 J_{\mathrm{OPT}}(m).$$

Exploiting 1D clustering results. Substituting the approximation factors α_1 of existing 1D clustering algorithms in Thm. 1 (with $t = 1$) instantly yields specific bounds for corresponding tensor clustering algorithms. Table 1 summarizes these results, however we omit proofs for lack of space—see [27] for details.

Table 1. Approximation guarantees for Tensor Clustering Algorithms. K^* denotes the maximum number of clusters, i.e., $K^* = \text{argmax}_j\, k_j$; c is some constant.

Problem Name	Approx. Bound	Proof
Metric tensor clustering	$J(\mathcal{C}) \leq m(1+\epsilon)J_{\text{OPT}}(m)$	Thm. 1 + [6]
Bregman tensor clustering	$E[J(\mathcal{C})] \leq 8mc(\log K^* + 2)J_{\text{OPT}}(m)$	(3.9), Thm. 1 + [7]
Bregman tensor clustering	$J(\mathcal{C}) \leq m\sigma_U\sigma_L^{-1}(1+\epsilon)J_{\text{OPT}}(m)$	(3.9), Thm. 1 + [5]
Bregman co-clustering	Above two results with $m = 2$	as above
Hilbertian metrics	$E[J(\mathcal{C})] \leq 8m(\log K^* + 2)J_{\text{OPT}}(m)$	See [27]

4 Experimental Results

Our bounds depend strongly on the approximation factor α_t of an underlying t-dimensional clustering method. In our experiments, we study this close dependence for $t = 1$, wherein we compare the tensor clusterings arising from different 1D methods of varying sophistication. Keep in mind that the comparison of the 1D methods is to see their impact on the tensor clustering built on top of them.

Our experiments reveal that the empirical approximation factors are usually smaller than the theoretical bounds, and these factors depend on statistical properties of the data. We also observe the linear dependence of the CoTeC objectives on the associated 1D objectives, as suggested by Thm. 1 (for Euclidean) and Table 1 (2nd row, for KL Divergence).

Further comparisons show that in practice, CoTeC is competitive with a greedy heuristic SiTeC (**Si**multaneous **Te**nsor **C**lustering), which *simultaneously* takes *all* dimensions into account, but lacks theoretical guarantees. As expected, initializing SiTeC with CoTeC yields lower final objective values using fewer "simultaneous" iterations.

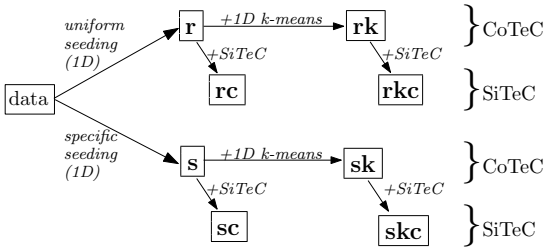


Fig. 3. Tensor clustering variants.

We focus on Euclidean distance and KL Divergence to test CoTeC. To study the effect of the 1D method, we use two seeding methods, uniform, and distance based (weighted farthest first) drawing. The latter ensures 1D approximation factors for $E[J(\mathcal{C})]$ by [7] for Euclidean clustering and by [8, 9] for KL Divergence.

We use each seeding by itself and as an initialization for k-means to get four 1D methods for each divergence (see Fig. 3). We refer to the CoTeC combination of the corresponding independent 1D clusterings by abbreviations: (1) '**r:**' uniformly sample centers from the data points and assign each point to its closest center; (2) '**s:**' sample centers with distance-specific seeding [7–9] and assign each point to its closest center; (3) '**rk:**' initialize Euclidean or Bregman k-means with '**r**'; (4) '**sk:**' initialize Euclidean or Bregman k-means with '**s**'.

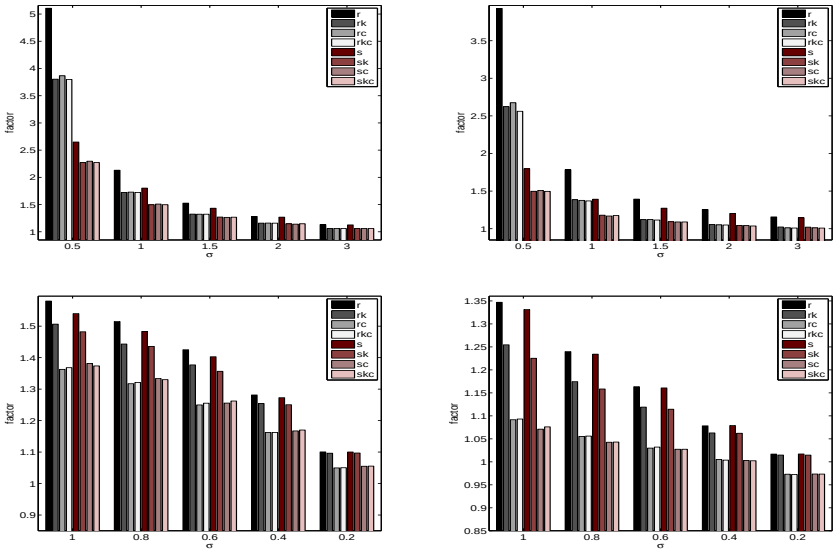


Fig. 4. Approximation factors for 3D clustering (left) and co-clustering (right) with increasing noise. Top row: Euclidean distances, bottom row: KL Divergence. The x axis shows σ, the y axis the empirical approximation factor.

The SiTeC method we compare to is the minimum sum-squared residue co-clustering of [29] for Euclidean distances in 2D, and a generalization of Algorithm 1 of [4] for 3D and Bregman 2D clustering. Additionally, we initialize SiTeC with the outcome of each of the four CoTeC variants, which yields four versions (of SiTeC), namely, **rc**, **sc**, **rkc**, and **skc**, initialized with the results of '**r**', '**s**', '**rk**', and '**sk**', respectively. These variants inherit the guarantees of CoTeC, as they monotonically decrease the objective value.

4.1 Experiments on Synthetic Data

For a controlled setting with synthetic data, we generate tensors A of size $75 \times 75 \times 50$ and 75×75, for which we randomly choose a $5 \times 5 \times 5$ tensor of means M and cluster indicator matrices $\mathsf{C}_i \in \{0,1\}^{n_i \times 5}$. For clustering with Euclidean distances we add Gaussian noise (from $\mathcal{N}(0, \sigma^2)$ with varying σ) to A, while for KL Divergences we use the sampling method of [4] with varying noise.

For each noise-level to test, we repeat the 1D seeding 20 times on each of five generated tensors and average the resulting 100 objective values. To estimate the approximation factor α_m on a tensor, we divide the achieved objective $J(\mathcal{C})$ by the objective value of the "true" underlying tensor clustering. Figure 4 shows the empirical approximation factor $\hat{\alpha}_m$ for Euclidean distance and KL Divergence. Qualitatively, the plots for tensors of order 2 and 3 do not differ.

Table 2. (i) Improvement of CoTeC and SiTeC variants upon '**r**' in %; the respective reference value (J_2 for '**r**') is shaded in gray. (ii) Average number of SiTeC iterations.

Bcell, Euc.

(i)

k_1	k_2		CoTeC x	CoTeC xk	SiTeC xc	SiTeC xkc
20	3	x=r	$5.75 \cdot 10^5$	31.66	20.05	33.05
		x=s	18.83	32.24	24.61	33.36
20	6	x=r	$5.56 \cdot 10^5$	49.13	35.26	50.37
		x=s	34.97	50.55	43.93	51.66
50	3	x=r	$5.63 \cdot 10^5$	31.10	14.77	31.76
		x=s	15.25	32.58	19.14	33.17
50	6	x=r	$5.18 \cdot 10^5$	47.55	34.63	48.41
		x=s	36.22	49.83	43.77	50.55

Bcell, KL

(i)

k_1	k_2		CoTeC x	CoTeC xk	SiTeC xc	SiTeC xkc
20	3	x=r	$3.37 \cdot 10^{-1}$	17.59	22.23	23.26
		x=s	10.54	18.44	22.99	22.98
20	6	x=r	$3.15 \cdot 10^{-1}$	18.62	24.51	25.43
		x=s	11.76	20.52	25.69	26.23
50	3	x=r	$3.20 \cdot 10^{-1}$	15.70	20.12	21.07
		x=s	9.61	17.24	20.85	21.33
50	6	x=r	$2.85 \cdot 10^{-1}$	16.38	21.61	22.57
		x=s	11.86	18.63	23.24	23.13

(ii)

k_1	k_2	rc	rkc	sc	skc
20	3	7.0 ± 1.4	2.0 ± 0.2	3.9 ± 1.0	2.2 ± 0.5
20	6	11.3 ± 2.3	2.6 ± 0.8	5.1 ± 2.0	2.7 ± 0.7
50	3	6.2 ± 1.9	2.0 ± 0.0	3.5 ± 2.0	2.0 ± 0.0
50	6	8.1 ± 2.1	2.1 ± 0.3	4.1 ± 1.6	2.0 ± 0.0

(ii)

k_1	k_2	rc	rkc	sc	skc
20	3	10.6 ± 2.8	7.5 ± 2.0	7.4 ± 1.8	7.0 ± 2.2
20	6	12.6 ± 3.4	8.8 ± 2.9	8.4 ± 2.1	8.1 ± 2.0
50	3	9.1 ± 2.3	6.2 ± 1.3	6.9 ± 1.8	6.0 ± 1.3
50	6	10.5 ± 1.8	7.7 ± 2.1	8.1 ± 2.3	6.9 ± 1.0

In all settings, the empirical factor remains below the theoretical factor. The reason for decreasing approximation factors with higher noise could be lower accuracy of the estimates of J_{OPT} on the one hand, and more similar objective values for all clusterings on the other hand. With low noise, distance-specific seeding **s** yields better results than uniform seeding **r**, and adding k-means on top (**rk**,**sk**) improves the results of both. With Euclidean distances, CoTeC with well-initialized 1D k-means (**sk**) competes with SiTeC. For KL Divergence, though, SiTeC still improves on **sk**, and with high noise levels, 1D k-means does not help: both **rk** and **sk** are as good as their seeding only counterparts **r**, **s**.

4.2 Experiments on Biological Data

We further assess the behavior of our method with gene expression data[6] from multiple sources [30–32]. For brevity, we only introduce two of the data sets here for which we present more detailed results; more datasets and experiments are described in [27].

The matrix *Bcell* [30] is a (1332×62) lymphoma microarray dataset of chronic lymphotic leukemia, diffuse large Bcell leukemia and follicular lymphoma. The order-3 tensor *Interferon* consists of gene expression levels from MS patients treated with recombinant human interferon beta [32]. After removal of missing values, a complete $6 \times 21 \times 66$ tensor remained. For experiments with KL Divergence, we normalized all tensors to have their entries sum up to one. Since our analysis concerns the objective function $J(\mathcal{C})$ alone, we disregard the "true" labels, which are available for only one of the dimensions.

For each data set, we repeat the sampling of centers 30 times and average the resulting objective values. Panel (i) in Table 2 (order-2), and in Table 3 (order-3) show the objective value for the simplest CoTeC variant '**r**' as a baseline, and

[6] We thank Hyuk Cho for kindly providing us his preprocessed 2D data sets.

the relative improvements achieved by other methods. The methods are encoded as **x**, **xk**, **xc**, **xkc**, where **x** stands for **r** or **s**, depending on the row in the table.

Interferon, KL

(i) k_1	k_2	k_3		x	xk	xc	xkc
2	2	2	x=r	$9.71 \cdot 10^{-1}$	38.58	42.46	43.53
			x=s	25.07	36.67	43.53	43.74
2	2	3	x=r	$8.17 \cdot 10^{-1}$	41.31	46.06	46.31
			x=s	33.63	43.90	46.82	47.16
2	2	4	x=r	$7.11 \cdot 10^{-1}$	39.79	44.05	45.62
			x=s	38.01	46.09	51.30	51.35

Table 3. (i) Improvement of CoTeC and SiTeC variants upon '**r**' in %; the respective reference value (J_3 for '**r**') is shaded in gray.

Figure 5 summarizes the average improvements for all five order-2 data sets studied in [27]. Groups indicate methods, and colors indicate seeding techniques. On average, a better seeding improves the results for all methods: the gray bars are higher than their black counterparts in all groups. Just as for synthetic data, 1D k-means improves the CoTeC results here too. SiTeC (groups 3 and 4) is better than CoTeC with mere seeding (**r**,**s**, group 1). Notably, for Euclidean distances, combining good 1D clusters obtained by k-means (**rk**,**sk**, group 2) is on average better than SiTeC initialized with simple seeding (**rc,sc**, group 3). For KL Divergences, on the other hand, SiTeC still outperforms all CoTeC variations. Given the limitation to single dimensions, CoTeC performs surprisingly well in comparison to SiTeC. Additionally, SiTeC initialized with CoTeC converges faster to better solutions, further underscoring the utility of CoTeC.

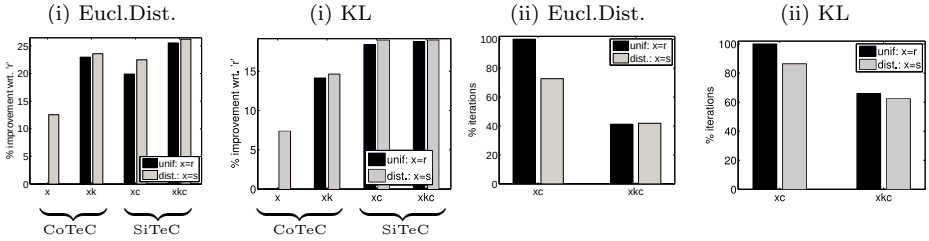


Fig. 5. (i) % improvement of the objective $J_2(\mathcal{C})$ with respect to uniform 1D seeding (**r**), averaged over all order-2 data sets and parameter settings (details in [27]). (ii) average number of SiTeC iterations, in % with respect to initialization by **r**.

Relation to 1D Clusterings Our experiments support the theoretical results and the intuitive expectation that better 1D clusterings yield better CoTeC solutions. Can we quantify this relation?

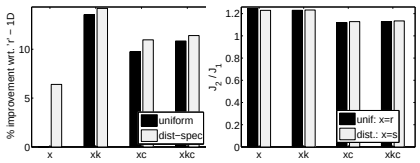
Theorem 1 suggests a linear dependence of the order-m factor α_m on α_1. However, these factors are difficult to check empirically when optimal clusterings are unknown. However, on one matrix $J_{\mathrm{OPT}}(2)/J_{\mathrm{OPT}}(1)$ is constant, so if the approximation factors are tight (up to a constant factor), the ratio

$$J_2(C_1, C_2)/J_1(C_i) \approx (\alpha_2/\alpha_1)\ J_{\mathrm{OPT}}(2)/J_{\mathrm{OPT}}(1), \quad i = 1, 2$$

only depends on α_2/α_1. Stating $\alpha_2 = 2\alpha_1\rho_d$, Thm. 1 predicts J_2/J_1 to be *independent* of the 1D method, i.e., of α_1, and constant on one matrix.

The empirical ratios J_2/J_1 in Figure 6 support this prediction, which suggests that for CoTeC the quality of the multi-dimensional clustering directly depends on the quality of its 1D components, both in theory *and* in practice.

Fig. 6. Left: average improvement of 1D clusterings (components) with respect to 'r'. Right: average ratio J_2/J_1, both for the same clusterings as in Figure 5.



5 Conclusion

In this paper we presented CoTeC, a simple, and to our knowledge the first approximation algorithm for tensor clustering, which yielded approximation results for Bregman co-clustering and tensor clustering as special cases. We proved an approximation factor that grows linearly with the order of the tensor, and showed tightness of the factor for the 2D Euclidean case (Fig. 2), though empirically the observed factors are usually smaller than suggested by the theory.

Our worst-case example also illustrates the limitation of CoTeC, i.e., to ignore the interaction between clusterings along multiple dimensions. Thm. 1 thus gives hints how much information maximally lies in this interaction. Analyzing this interplay could potentially lead to better approximation factors, e.g., by developing a co-clustering specific seeding technique. Using such an algorithm as a subroutine in CoTeC will yield a hybrid that combines CoTeC's simplicity with better approximation guarantees.

Acknowledgment AB was supported in part by NSF grant IIS-0812183.

References

1. Banerjee, A., Basu, S., Merugu, S.: Multi-way Clustering on Relation Graphs. In: SIAM Conf. Data Mining (SDM). (2007)
2. Shashua, A., Zass, R., Hazan, T.: Multi-way Clustering Using Super-Symmetric Non-negative Tensor Factorization. LNCS **3954** (2006) 595–608
3. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: KDD. (2003) 89–98
4. Banerjee, A., Dhillon, I.S., Ghosh, J., Merugu, S., Modha, D.S.: A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. JMLR **8** (2007) 1919–1986
5. Ackermann, M.R., Blömer, J.: Coresets and Approximate Clustering for Bregman Divergences. In: ACM-SIAM Symp. on Disc. Alg. (SODA). (2009)
6. Ackermann, M.R., Blömer, J., Sohler, C.: Clustering for metric and non-metric distance measures. In: ACM-SIAM Symp. on Disc. Alg. (SODA). (April 2008)
7. Arthur, D., Vassilvitskii, S.: `k-means++`: The Advantages of Careful Seeding. In: ACM-SIAM Symp. on Discete Algorithms (SODA). (2007) 1027–1035
8. Nock, R., Luosto, P., Kivinen, J.: Mixed Bregman clustering with approximation guarantees. In: Eur. Conf. on Mach. Learning (ECML). LNAI 5212 (2008)

9. Sra, S., Jegelka, S., Banerjee, A.: Approximation algorithms for Bregman clustering, co-clustering and tensor clustering. Technical Report 177, MPI for Biological Cybernetics (2008)
10. Ben-David, S.: A framework for statistical clustering with constant time approximation algorithms for K-median and K-means clustering. Mach. Learn. **66**(2-3) (2007) 243–257
11. Puolamäki, K., Hanhijärvi, S., Garriga, G.C.: An approximation ratio for biclustering. Inf. Process. Letters **108**(2) (2008) 45–49
12. Anagnostopoulos, A., Dasgupta, A., Kumar, R.: Approximation algorithms for co-clustering. In: Symp. on Principles of Database Systems (PODS). (2008)
13. Zha, H., Ding, C., Li, T., Zhu, S.: Workshop on Data Mining using Matrices and Tensors. KDD (2008)
14. Hasan, M., Velazquez-Armendariz, E., Pellacini, F., Bala, K.: Tensor Clustering for Rendering Many-Light Animations. Eurographics Symp. on Rendering **27** (2008)
15. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review **51**(3) (2009) to appear.
16. Hartigan, J.A.: Direct clustering of a data matrix. J. of the Am. Stat. Assoc. **67**(337) (March 1972) 123–129
17. Cheng, Y., Church, G.: Biclustering of expression data. In: Proc. ISMB, AAAI Press (2000) 93–103
18. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD. (2001) 269–274
19. Bekkerman, R., El-Yaniv, R., McCallum, A.: Multi-way distributional clustering via pairwise interactions. In: ICML. (2005)
20. Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., Belongie, S.: Beyond pairwise clustering. In: IEEE CVPR. (2005)
21. Govindu, V.M.: A tensor decomposition for geometric grouping and segmentation. In: IEEE CVPR. (2005)
22. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press (2001)
23. Hein, M., Bosquet, O.: Hilbertian metrics and positive definite kernels on probability measures. In: AISTATS. (2005)
24. Censor, Y., Zenios, S.A.: Parallel Optimization: Theory, Algorithms, and Applications. Oxford University Press (1997)
25. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman Divergences. JMLR **6**(6) (October 2005) 1705–1749
26. de Silva, V., Lim, L.H.: Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. SIAM J. Matrix Anal. & Appl. **30**(3) (2008) 1084–1127
27. Jegelka, S., Sra, S., Banerjee, A.: Approximation algorithms for Bregman coclustering and tensor clustering. In: arXiv: cs.DS/0812.0389. (v3,2009)
28. Chaudhuri, K., McGregor, A.: Finding metric structure in information theoretic clustering. In: Conf. on Learning Theory, COLT. (July 2008)
29. Cho, H., Dhillon, I.S., Guan, Y., Sra, S.: Minimum Sum Squared Residue based Co-clustering of Gene Expression data. In: SDM. (2004) 114–125
30. Kluger, Y., Basri, R., Chang, J.T.: Spectral biclustering of microarray data: Coclustering genes and conditions. Genome Research **13** (2003) 703–716
31. Cho, H., Dhillon, I.: Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. IEEE/ACM Tran. Comput. Biol. Bioinf. **5**(3) (2008) 385–400
32. Baranzini, S.E., *et al.*: Transcription-based prediction of response to IFNβ using supervised computational methods. PLoS Biology **3**(1) (2004)