Approximation Algorithms for Bregman Co-clustering and Tensor Clustering*

Stefanie Jegelka
MPI for Biological Cybernetics
72070 Tübingen, Germany

Suvrit Sra
MPI for Biological Cybernetics
72070 Tübingen, Germany

Arindam Banerjee
Univ. of Minnesota
MN 55455, USA

Abstract

In the past few years powerful generalizations to the Euclidean k-means problem have been made, such as Bregman clustering [7], co-clustering (i.e., simultaneous clustering of rows and columns of an input matrix) [3, 18], and tensor clustering [8, 34]. Like k-means, these more general problems also suffer from the NP-hardness of the associated optimization. Researchers have developed approximation algorithms of varying degrees of sophistication for k-means, k-medians, and more recently also for Bregman clustering [2]. However, there seem to be no approximation algorithms for Bregman co- and tensor clustering. In this paper we derive the first (to our knowledge) guaranteed methods for these increasingly important clustering settings. Going beyond Bregman divergences, we also prove an approximation factor for tensor clustering with arbitrary separable metrics. Through extensive experiments we evaluate the characteristics of our method, and show that it also has practical impact.

1 Introduction

Partitioning data points into clusters is a fundamentally hard problem. The well-known Euclidean k-means problem that partitions the input data points (vectors in \( \mathbb{R}^d \)) into \( K \) clusters while minimizing sums of their squared distances to corresponding cluster centroids, is an NP hard problem [19] (exponential in \( d \)). However, simple and frequently used procedures that rapidly obtain local minima exist since a long time [23, 28].

Because of its wide applicability and importance, the Euclidean k-means problem has been generalized in several directions. Specific examples relevant to this paper include:

- Bregman clustering [7], where instead of minimizing squared Euclidean distances one minimizes Bregman divergences (which are generalized distance functions, see [6, 10] or [13] for details),
- Bregman co-clustering [8] (which includes both Euclidean [16] and information-theoretic co-clustering [18] as special cases), where the set of input vectors is viewed as a matrix and one simultaneously clusters rows and columns to obtain coherent submatrices (co-clusters), while minimizing a Bregman divergence, and
- Tensor clustering or multiway clustering [34], especially the version based on Bregman divergences [8], where one simultaneously clusters along various dimensions of the input tensor.

For these problems too, the commonly used heuristics perform well, but do not provide theoretical guarantees (or at best assure local optimality). For k-means type clustering problems—i.e., problems

*A part of the theory of this paper appeared in MPI-TR-#177 [35].
that group together input vectors into clusters while minimizing “distance” to cluster centroids—there exist several algorithms that approximate a globally optimal solution. We refer the reader to [1, 2, 6, 27], and the numerous references therein for more details. In stark contrast, approximation algorithms for tensor clustering are much less studied. We are aware of only two very recent attempts (both papers are from 2008) for the two-dimensional special case of co-clustering, namely, [4] and [31]—and both of the papers follow similar approaches to obtain their approximation guarantees. Both prove a $2\alpha_1$-approximation for Euclidean co-clustering, Puolamäki et al. [31] an additional factor of $(1 + \sqrt{2})$ for binary matrices and an $\ell_1$ norm objective, and Anagnostopoulos et al. [4] a factor of $3\alpha_1$ for co-clustering real matrices with $\ell_p$ norms. In all factors $\alpha_1$ is an approximation guarantee for clustering either rows or columns. In this paper, we build upon [4] and obtain approximation algorithms for tensor clustering with Bregman divergences and arbitrary separable metrics such as $\ell_p$-norms. The latter result is of particular interest for $\ell_1$-norm based tensor clustering, which may be viewed as a generalization of k-medians to tensors. In the terminology of [7], we focus on the “block average” versions of co- and tensor clustering.

Additional discussion and relevant references for co-clustering can be found in [9], while for the lesser known problem of tensor clustering more background can be gained by referring to [3, 8, 10, 21, 29, 34].

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 $O(m\alpha)$, where $m$ is the order of the tensor and $\alpha$ 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 $\ell_p$ norms or Hilbertian metrics [24, 33], and divergence functions such as Bregman divergences [13] (with some assumptions). As corollaries, our results solve two open problems posed by [4], 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. Owing to the structure of the algorithm, our results also give insight into properties of the tensor clustering problem as such, namely, a bound on the amount of information inherent in the joint consideration of several dimensions.

In addition, we provide extensive experimental validation of the theoretical claims, which forms an additional contribution of this paper.

2 Background

Traditionally, “center” based clustering algorithms seek partitions of columns of an input matrix $X = [x_1, \ldots, x_n]$ into clusters $C = \{C_1, \ldots, C_K\}$, and find “centers” $\mu_k$ that minimize the objective

$$J(C) = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, \mu_k),$$

where the function $d(x, y)$ measures cluster quality. The “center” $\mu_k$ of cluster $C_k$ is given by the mean of the points in $C_k$ when $d(x, y)$ is a Bregman divergence $\ell_2$. Co-clustering extends (2.1) to seek simultaneous partitions (and centers $\mu_{I,J}$) of rows and columns of $X$, so that the objective function

$$J(C) = \sum_{I,J} \sum_{i \in I, j \in J} d(x_{ij}, \mu_{I,J}),$$

is minimized; $\mu_{I,J}$ denotes the (scalar) “center” of the cluster described by the row and column index sets, viz., $I$ and $J$. Formulation (2.2) is easily generalized to tensors, as shown in Section 2.2 below. However, we first recall basic notation about tensors before formally presenting the tensor clustering problem. Tensors are well-studied in multilinear algebra [22], and they are gaining importance in both data mining and machine learning applications [26, 34].
2.1 Tensors

A large part of the material in this section is derived from the well-written paper of de Silva and Lim [17]—their notation turns out to be particularly suitable for our analysis. An order-\(m\) tensor \(A\) may be viewed as an element of the vector space \(\mathbb{R}^{n_1 \times \cdots \times n_m}\) (in this paper we denote matrices and tensors using sans-serif letters). An individual component of the tensor \(A\) is represented by the multiply-indexed value \(a_{i_1 i_2 \ldots i_m}\), where \(i_j \in \{1, \ldots, n_j\}\) for \(1 \leq j \leq m\).

Multilinear matrix multiplication

For us the most important operation on tensors is that of multilinear matrix multiplication, which is a generalization of the familiar concept of matrix multiplication. Matrices act on other matrices by either left or right multiplication. For an order-3 tensor, there are three dimensions on which a matrix may act via matrix multiplication. For example, for an order-3 tensor \(A \in \mathbb{R}^{n_1 \times n_2 \times n_3}\), and three matrices \(P \in \mathbb{R}^{p_1 \times n_1}\), \(Q \in \mathbb{R}^{p_2 \times n_2}\), and \(R \in \mathbb{R}^{p_3 \times n_3}\), multilinear matrix multiplication is the operation defined by the action of these three matrices on the different dimensions of \(A\) that yields the tensor \(A' \in \mathbb{R}^{p_1 \times p_2 \times p_3}\). Formally, the entries of the tensor \(A'\) are given by

\[
a'_{i_1 i_2 \ldots i_m} = \sum_{i,j,k=1}^{n_1,n_2,n_3} p_{i} q_{nj} r_{nk} a_{ijk}, \tag{2.3}
\]

and this operation is written compactly as

\[
A' = (P, Q, R) \cdot A. \tag{2.4}
\]

Multilinear multiplication extends naturally to tensors of arbitrary order. If \(A \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}\), and \(P_1 \in \mathbb{R}^{p_1 \times n_1}, \ldots, P_m \in \mathbb{R}^{p_m \times n_m}\), then \(A' = ((P_1, \ldots, P_m) \cdot A) \in \mathbb{R}^{p_1 \times \cdots \times p_m}\) has components

\[
a'_{i_1 i_2 \ldots i_m} = \sum_{j_1, \ldots, j_m=1}^{n_1, \ldots, n_m} p_{1}^{(1)} \cdots p_{m}^{(m)} a_{j_1 \ldots j_m}, \tag{2.5}
\]

where \(p_{ij}^{(k)}\) denotes the \(ij\)-th entry of matrix \(P_k\).

Example 2.1 (Matrix Multiplication). Let \(A \in \mathbb{R}^{n_1 \times n_2}\), \(P \in \mathbb{R}^{p \times n_1}\), and \(Q \in \mathbb{R}^{q \times n_2}\) be three matrices. The matrix product \(PAQ\) can be written as the multilinear multiplication \((P, Q) \cdot A\).

Proposition 2.2 (Basic Properties). The following properties of multilinear multiplication are easily verified (and generalized to tensors of arbitrary order):

1. **Linearity:** Let \(\alpha, \beta \in \mathbb{R}\), and \(A\) and \(B\) be tensors with same dimensions, then

\[
(P, Q) \cdot (\alpha A + \beta B) = \alpha (P, Q) \cdot A + \beta (P, Q) \cdot B
\]

2. **Product rule:** For matrices \(P_1, P_2, Q_1, Q_2\) of appropriate dimensions, and a tensor \(A\)

\[
(P_1, P_2) \cdot ((Q_1, Q_2) \cdot A) = (P_1Q_1, P_2Q_2) \cdot A
\]

3. **Multilinearity:** Let \(\alpha, \beta \in \mathbb{R}\), and \(P, Q,\) and \(R\) be matrices of appropriate dimensions. Then, for a tensor \(A\) the following holds

\[
(P, \alpha Q + \beta R) \cdot A = \alpha (P, Q) \cdot A + \beta (P, R) \cdot A
\]

Vector Norms

The standard vector \(\ell_p\)-norms can be easily extended to tensors, and are defined as

\[
\|A\|_p = \left( \sum_{i_1, \ldots, i_m} |a_{i_1 \ldots i_m}|^p \right)^{1/p}, \tag{2.6}
\]

for \(p \geq 1\). In particular for \(p = 2\) we get the “Frobenius” norm, also written as \(\|A\|_F\).
Inner Product

The Frobenius norm induces an inner-product that can be defined as

$$\langle A, B \rangle = \sum_{i_1, \ldots, i_m} a_{i_1 \ldots i_m} b_{i_1 \ldots i_m},$$  \hspace{1cm} (2.7)$$

so that $\|A\|_F^2 = \langle A, A \rangle$ holds as usual.

**Proposition 2.3.** The following property of this inner product is easily verified (a generalization of the familiar property $(A, B) = \langle x, A^T B \rangle$ for vectors):

$$((P_1, \ldots, P_m) \cdot A, (Q_1, \ldots, Q_m) \cdot B) = \langle A, (P_1^T Q_1, \ldots, P_m^T Q_m) \cdot B \rangle. \hspace{1cm} (2.8)$$

**Proof:** Using definition (2.5) and the inner-product rule (2.7) we have

$$((P_1, \ldots, P_m) \cdot A, (Q_1, \ldots, Q_m) \cdot B) = \sum_{i_1, \ldots, i_m} p_{i_1 j_1 k_1} \cdots p_{i_m j_m k_m} a_{j_1 \ldots j_m} b_{k_1 \ldots k_m},$$

$$= \sum_{j_1 \ldots j_m} \left( \sum_{i_1, \ldots, i_m} p_{i_1 j_1 k_1} \cdots p_{i_m j_m k_m} \right) a_{j_1 \ldots j_m} b_{k_1 \ldots k_m},$$

$$= \sum_{j_1 \ldots j_m} (P_1^T Q_1)_{j_1 k_1} \cdots (P_m^T Q_m)_{j_m k_m} a_{j_1 \ldots j_m} b_{k_1 \ldots k_m} = \sum_{j_1 \ldots j_m} a_{j_1 \ldots j_m} b'_{j_1 \ldots j_m} = \langle A, B' \rangle,$$

where $B' = (P_1^T Q_1, \ldots, P_m^T Q_m) \cdot B$.

Divergence

Finally, we define an arbitrary divergence function $d(X, Y)$ between two $m$-dimensional tensors $X, Y$ as an elementwise sum of individual divergences, i.e.,

$$d(X, Y) = \sum_{i_1, \ldots, i_m} d(x_{i_1, \ldots, i_m}, y_{i_1, \ldots, i_m}),$$  \hspace{1cm} (2.9)$$

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

### 2.2 Tensor clustering

Let $A \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ be an order-$m$ tensor that we wish to partition into coherent sub-tensors (or clusters). 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 $j$ ($1 \leq j \leq m$) is partitioned into $k_j$ clusters. Let $C_j \in \{0, 1\}^{n_j \times k_j}$ 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 ($1 \leq k \leq k_j$) for dimension $j$. Then, the tensor clustering problem is (cf. 2.2):

$$\minimize_{C_1, \ldots, C_m, M} d(A, (C_1, \ldots, C_m) \cdot M), \quad \text{s.t. } C_j \in \{0, 1\}^{n_j \times k_j},$$  \hspace{1cm} (2.10)$$

where the tensor $M$ collects all the cluster “centers.”

### 3 Algorithm and Analysis

Given formulation (2.10), our algorithm, which we name Combination Tensor Clustering (CoTeC), follows the simple outline:

1. Cluster along each dimension $j$, using an approximation algorithm to obtain clustering $C_j$; Let $C = (C_1, \ldots, C_m)$.
2. Compute $M = \arg\min_{X \in \mathbb{R}^{n_1 \times \cdots \times n_m}} d(A, C \cdot X)$.
3. Return the tensor clustering $(C_1, \ldots, C_m)$ (with representatives $M$).
Instead of clustering one dimension at a time, we can also cluster along $t$ dimensions at a time, which we will call $t$-dimensional clustering. For an order-$m$ tensor, with $t = 1$ we form groups of order-$(m - 1)$ tensors. For illustration, consider an order-3 tensor $A$ for which we group matrices when $t = 1$. For the first dimension we cluster the objects $A(i, :, :)$ (using MATLAB notation) to obtain cluster indicators $C_1$; we repeat the procedure for the second and third dimensions. The approximate tensor clustering will be the combination $(C_1, C_2, C_3)$. As we assumed the Bregman divergences to be separable, the sub-tensors, e.g., $A(i, :, :)$ can be simply treated as vectors.

Apart from [4, 31], all approximation guarantees refer to one-dimensional clustering algorithms. Any one-dimensional approximation algorithm can be used as a base method for our scheme outlined above. For example, the method of Ackermann and Blömer [1], or the more practical Bregman clustering approaches of [30, 35] are two potential choices, though with different approximation factors. Clustering along individual dimensions and then combining the results to obtain a tensor clustering might seem counterintuitive to the idea of “co”-clustering, where one simultaneously clusters along different dimensions. However, our analysis will show that dimension-wise clustering suffices to obtain strong approximation guarantees for tensor clustering—a fact often observed empirically too. At the same time, our Theorem 1 bounds the amount of information that can lie in the simultaneous consideration of multiple dimensions.

3.1 Results

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

**Theorem 3.1 (Approximation).** Let $A$ be an order-$m$ tensor and let $C_j$ 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 $\alpha_j$. Let $C = (C_1, \ldots, C_{m/t})$ denote the induced tensor clustering, and $J_{OPT}(m)$ the best $m$-dimensional clustering. Then,

$$J(C) \leq p(m/t)\rho_d\alpha_j J_{OPT}(m), \text{ with}$$

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 $p(m/t) = 2m/t$ if $d(x, y)$ is a metric.\(^2\)

Theorem 3.1 is quite general, and it can be combined with some natural assumptions (see Section 3.5) to yield results for tensor clustering with more general divergence functions too (here $\rho_d$ might be greater than 1).

3.2 Analysis: Theorem 3.1 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 [4]. We establish that given a clustering algorithm which clusters along $t$ of the $m$ dimensions at a time\(^1\) with an approximation factor of $\alpha_t$, our CoTeC algorithm achieves an objective within a factor $O([m/t]\alpha_t)$ of the optimal. For example, for $t = 1$ we can use the seeding methods of [30, 35] or the stronger approximation algorithms of [1]. 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 2.10 into a more convenient form. To that end, note that the individual entries of the

---

\(^1\)Both [30, 35] discovered essentially the same method for Bregman clustering, though the analysis of [30] is somewhat sharper.

\(^2\)We say an approximation algorithm has guarantee $\alpha$ if it yields a solution that achieves an objective value within a factor $O(\alpha)$ of the optimum.

\(^3\)The results can be trivially extended to $\lambda$-relaxed metrics that satisfy $d(x, y) \leq \lambda(d(x, z) + d(z, y))$; the corresponding approximation factor just gets scaled by $\lambda$.

\(^4\)One could also consider clustering differently sized subsets of the dimensions, say $\{t_1, \ldots, t_r\}$, where $t_1 + \cdots + t_r = m$. However, this requires unilluminating notational juggling, which we can skip for simplicity of exposition.
means tensor $M$ are given by (cf. (2.3))

$$\mu_{I_1, \ldots, I_m} = \frac{1}{|I_1| \cdots |I_m|} \sum_{i_1 \in I_1, \ldots, i_m \in I_m} a_{i_1 \ldots i_m}, \quad (3.2)$$

with index sets $I_j$ for $1 \leq j \leq m$. Let $\mathbb{T}_j$ be the normalized cluster indicator matrix obtained by normalizing the columns of $C_j$, so that $\mathbb{T}_j \mathbb{C}_j = I_{k_j}$. Then, we can rewrite (2.10) in terms of projection matrices $P_j$ as:

$$\text{minimize} \quad J(C) = \|A - (P_1, \ldots, P_m) \cdot A\|^2_F, \quad \text{s.t.} \quad P_j = \mathbb{T}_j \mathbb{T}_j^\top. \quad (3.3)$$

**Lemma 3.2** (Pythagorean). Let $P = (P_1, \ldots, P_t)$, $S = (P_{t+1}, \ldots, P_m)$, and $P^\perp = (1-P_1, \ldots, 1-P_t)$ be collections of projection matrices $P_j$. Then,

$$\| (P, S) \cdot A + (P^\perp, R) \cdot B \|^2 = \| (P, S) \cdot A\|^2 + \| (P^\perp, R) \cdot B\|^2,$$

where $R$ is a collection of $m - t$ projection matrices.

**Proof.** Using $\| A \|^2_F = (A, A)$ we can rewrite the l.h.s. as

$$\| (P, S) \cdot A + (P^\perp, R) \cdot B\|^2 = \| (P, S) \cdot A\|^2 + \| (P^\perp, R) \cdot B\|^2 + 2 \langle (P, S) \cdot A, (P^\perp, R) \cdot B \rangle.$$

The last term is immediately seen to be zero using Property (2.8) and the fact that $P_j^\top P^\perp_j = P_j(1-P_j) = 0$. □

**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 $2^l$ (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 $P^l_t$ (note boldface), where the index $i$ ranges from 1 to $2^l - 1$.

**Example 3.3.** Consider an order-8 tensor where we group $t = 2$ dimensions at a time. Then, $h = \log(m/t) = 2$ and we have 3 levels. We recursively divide the set of dimensions in the middle, i.e., $\{1, \ldots, 8\}$ into $\{1, \ldots, 4\}$ and $\{5, \ldots, 8\}$ and so on, ending with $\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}$. The projection matrix for dimension $i$ is $P_i$, and the full tensor clustering is represented by $(P_1, \ldots, P_8)$. For each level $l = 0, 1, 2$, individual collections of projection matrices $P^l_t$ are

$$P^2_1 = (P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8),$$
$$P^1_1 = (P_1, P_2, P_3, P_4), \quad P^1_2 = (P_5, P_6, P_7, P_8),$$
$$P^0_1 = (P_1, P_2), \quad \ldots, \quad P^0_4 = (P_7, P_8).$$

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 $P^l_t$ with $m - 2^l t$ identity matrices for the non-clustered dimensions, and call this padded collection $Q^l_t$. With recursive partitioning of the dimensions, $Q^l_i$ subsumes $Q^0_j$ for $2^l(i-1) < j \leq 2^l i$, so that

$$Q^l_i = \prod_{j = 2^l (i-1) + 1}^{2^l i} Q^0_j.$$

At level 0, the algorithm yields the collections $Q^0_1$ and $P^0_t$. 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 $I^l$, so that $Q^l_i = (P^l_i, I^l)$. Accoutered with our notation, we now prove the main lemma that relates the combined clustering to its sub-clusterings.
Lemma 3.4. Let $A$ be an order-$m$ tensor and $m \geq 2^t$. The objective function for any $2^t$-dimensional clustering $F_{i} = (F_{0}^{0}, F_{0}^{(1)}, \ldots, F_{0}^{2^t})$ can be bound via the sub--clusterings along only one set of dimensions of size $t$ as

$$\|A - Q_{i} \cdot A\|_{F}^{2} \leq \max_{2^t \leq 2^{(i-1)}} 2^{i} \|A - Q_{j} \cdot A\|_{F}^{2}. \quad (3.4)$$

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

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

Base: Let $l = 0$. Then $Q_{i} = Q_{i}^{0}$, and (3.4) holds trivially.

Induction: Assume the claim holds for $l \geq 0$. Consider a clustering $F_{i}^{l+1} = (F_{i}^{l}, F_{2}^{l})$, or equivalently $Q_{i}^{l+1} = Q_{1}^{l} Q_{2}^{l}$. Using $P + \mathbf{P} = \mathbf{I}$, we decompose $A$ as

$$A = (P_{i}^{l+1} + P_{i}^{l+1} \cdot A + (P_{i}^{l+1} + P_{2}^{l} + P_{2}^{l} \cdot I^{l+1}) \cdot A = Q_{i}^{l} Q_{1}^{l} \cdot A + Q_{i}^{l} Q_{2}^{l} \cdot A + Q_{1}^{l} Q_{2}^{l} \cdot A,$$

where $Q_{1}^{l} = (P_{1}^{l} + I^{l})$. Since $Q_{i}^{l+1} = Q_{1}^{l} Q_{2}^{l}$, the Pythagorean Property yields

$$\|A - Q_{i}^{l+1} \cdot A\|^{2} = \|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2} + \|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2} + \|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2}$$

Combining the above equalities with the assumption (wlog)

$$\|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2} \geq \|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2},$$

we obtain the inequalities

$$\|A - Q_{i}^{l} Q_{2}^{l} \cdot A\|^{2} \leq 2(\|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2} + \|Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2})$$

$$= 2\|Q_{1}^{l} Q_{2}^{l} \cdot A + Q_{1}^{l} Q_{2}^{l} \cdot A\|^{2} = 2\|Q_{1}^{l} (Q_{1}^{l} Q_{2}^{l} + Q_{2}^{l}) \cdot A\|^{2}$$

$$= 2\|Q_{1}^{l} \cdot A\|^{2} \leq 2\|A - Q_{1}^{l} \cdot A\|^{2} \leq 2 \cdot 2^{l} \cdot \max_{1 \leq j \leq 2^{l+1}} \|A - Q_{j}^{l} \cdot A\|^{2},$$

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

Proof. (Thm. 3.7 Case 1). Let $m = 2^{3t}$. Using an algorithm with guarantee $\alpha_{t}$, we cluster each subset (indexed by $i$) of $t$ dimensions to obtain $Q_{i}^{0}$. Let $S_{i}$ be the optimal sub-clustering of subset $i$, i.e., the result that $Q_{i}^{0}$ would be if $\alpha_{t}$ were 1. We bound the objective for the collection of all $m$ sub-clusterings $F_{i}^{0} = Q_{1}^{l}$ as

$$\|A - Q_{1}^{l} \cdot A\|_{F}^{2} \leq 2^{h} \max_{j} \|A - Q_{j}^{l} \cdot A\|^{2} \leq 2^{h} \alpha_{t} \max_{j} \|A - S_{j} \cdot A\|_{F}^{2}. \quad (3.5)$$

The first inequality follows from Lemma 3.4 while the last inequality follows from the $\alpha_{t}$ approximation factor that we used to get sub-clustering $Q_{i}^{0}$.

So far we have related our approximation to an optimal sub-clustering across a set of dimensions. Let us hence look at the relation between such an optimal sub-clustering $S$ of the first $t$ dimensions (via permutation, these dimensions correspond to an arbitrary subset of size $t$), and the optimal tensor clustering $F$ across all the $m = 2^{3t}$ dimensions. Recall that a clustering can be expressed by either the projection matrices collected in $Q_{1}^{l}$, or by cluster indicator matrices $C_{i}$ together with the mean tensor $M$, so that

$$(C_{1}, \ldots, C_{2^{l+1}}, I^{l}) \cdot M = Q_{1}^{l} \cdot A.$$
3.3 Analysis: Theorem 3.1 Metric case

Let $C_j^S$ and $C_j^F$ be the dimension-wise cluster indicator matrices for $S$ and $F$, respectively; By definition, $S$ solves
\[
\min_{C_1, \ldots, C_t, M} \|A - (C_1, \ldots, C_t, I^0) \cdot M\|_F^2, \quad \text{s.t. } C_j \in \{0, 1\}^{n_j \times k_j},
\]
which makes $S$ even better than the sub-clustering $(C_1^F, \ldots, C_t^F)$ induced by the optimal $m$-dimensional clustering $F$. Thus,
\[
\|A - S \cdot A\|_F^2 \leq \min_M \|A - (C_1^F, \ldots, C_t^F, I^0) \cdot M\|_F^2 \\
\leq \|A - (C_1^F, \ldots, C_t^F, I^0)(1, \ldots, 1, C_{t+1}^F, \ldots, C_m^F) \cdot M^F\|_F^2 \\
= \|A - F \cdot A\|_F^2,
\]
(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 $C = Q^h_i$,
\[
J_m(C) = \|A - Q^h_i \cdot A\|_F^2 \leq 2^h \alpha_i \|A - F \cdot A\|_F^2 = 2^h \alpha_i J_{OPT}(m),
\]
which completes the proof of the theorem.

\[
\square
\]

3.3 Analysis: Theorem 3.1 Metric case

Now we present our proof of Thm. 3.1 for the case where $d(x, y)$ is a metric, such as an $\ell_p$ distance or separable Hilbertian metric. For this case, recall that the tensor clustering problem is
\[
\min_{(C_1, \ldots, C_m, M)} J(C) = d(A, (C_1, \ldots, C_m) \cdot M), \quad \text{s.t. } C_j \in \{0, 1\}^{n_j \times k_j}. \tag{3.7}
\]

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

Proof. We will split the dimensions in a different way. Let $R^i_\ell$ be the collection of clusterings of dimensions $i, \ldots, i + \ell - 1$. $R^i_\ell$ combines the $C_j$ in a manner analogous to how $Q^i_1$ combines projection matrices. For simplicity, the proof here is for clustering single dimensions at a time, but it generalizes in a straightforward way to chunks of $t$ dimensions, leading to a factor $2m/t$ instead of $2m$.

Let us first prove a relation for any subset of the last $m - i + 1$ dimensions. Let $R^i_{m-i+1} = R^m_{m-i+1}$. Let $M_i^1 = \arg\min_X d(A, R^i_\ell \cdot X)$ be the optimal representatives for the clustering collections $R^i_\ell$ and $R^m_{m-i+1}$, and
\[
\tilde{M}_i = \arg\min_X d(R^i_\ell M_i^1, R^i_\ell R^m_{m-i+1} \cdot X), \quad X \in \mathbb{R}^{n_1 \times \ldots \times n_{i-1} \times k_i \times \ldots \times k_m}.
\]

The index $i$ will run over dimension $i$, and the multi-indices, $r, j$ over dimensions $1, \ldots, i - 1$ and $i + 1, \ldots, m$, respectively. The indices $I$ and multi-indices $J$ refer to the clusterings in $R^i_\ell$ and $R^m_{m-i+1}$, respectively. Since $M_i$ is the element-wise minimum, we have
\[
d(R^i_\ell M_i^1, R^i_\ell R^m_{m-i+1} \cdot M_i) = \sum_{I,J} \sum_{I,J} \min_{\mu_{I,J} \in \mathbb{R}} \sum_{j \in J} d((\mu^{(1)}_I)_{I,J}, \mu_{I,J}^{(m-i+1)}) \\
\leq \sum_{I,J} \sum_{I,J} \sum_{j \in J} d((\mu^{(1)}_I)_{I,J}, (\mu^{m-i+1}_{m-i+1})_{I,J}) = d(R^i_\ell M_i^1, R^i_\ell R^m_{m-i+1} \cdot M_i^{m-i}).
\]

We use this relation and the triangle inequality to break down $R^m_i$ into its single-dimensional parts.
We then relate the objectives of these parts to the optimal single-dimensional clusterings \( S^1_i \).

\[
\min_{M^m} \ d(A, R^1_1 R^{m-1}_2 : M^m) \leq d(A, R^1_1 R^{m-1}_2 : \hat{M}_1)
\]
\[
\leq d(A, R^1_1 : M^m_1) + d(R^1_1 : M^m_1, R^{m-1}_2 : \hat{M}_1)
\]
\[
\leq d(A, R^1_1 : M^m_1) + d(R^1_1 : M^m_1, R^{m-1}_2 : M^{m-1}_2)
\]
\[
\leq 2d(A, R^1_1 : M^m_1) + d(A, R^{m-1}_2 : M^{m-1}_2)
\]
\[
\leq \ldots
\]
\[
\leq 2 \sum_{i=1}^m d(A, R^1_i : M^m_i) \leq 2 \sum_{i=1}^m \alpha_1 \min_{X^1_i} d(A, S^1_i : X^1_i). \tag{3.9}
\]

For (3.8), we applied the same steps as before to \( R^1_1 \) and \( R^{m-2}_1 \), and then continued this breakdown, always splitting off the first dimension. The last relation follows from the 1D approximation algorithm that was used. What is left is to bound (3.9) by the objective for the optimal single-dimensional clusterings \( F \cdot M_F = F_1 F_2 \ldots F_m : M_F \). Note that, since non-clustered dimensions have identity matrices, the cluster parts commute: \( F_1 F_2 X = F_2 F_1 X \). Owing to the optimality of \( S^1_i \), we have

\[
\min_{X^1_i} d(A, S^1_i : X^1_i) \leq \min_{Y^m} d(A, F_i : Y^m) \leq \min_{Y^m} d(A, F_i (F^1_1 \ldots F^1_{i-1} F^1_{i+1} \ldots F^1_m Y^m)) = d(A, F : M_F)
\]

for any term in the sum (3.9). Thus, it follows that

\[
\min_{M^m} \ d(A, R^m_1 : M^m) \leq 2 \sum_{i=1}^m \alpha_1 \min_{X^1_i} d(A, S^1_i : X^1_i) \leq 2m \alpha_1 d(A, F : M_F),
\]

which completes the proof.

\[\square\]

### 3.4 Theorem 3.1 with Bregman divergences

Theorem 3.1 also applies to Bregman divergences, i.e., divergences that can be bounded in terms of squared Euclidean distances and for which the best representative is the tensor of means defined in Equation (3.2) \( \hat{Q} \).

The *Bregman divergence* \( B_f(x, y) \) between scalars \( x \) and \( y \) is defined as \( 12 \ 13 \)

\[
B_f(x, y) = f(x) - f(y) - f'(y)(x - y), \tag{3.10}
\]

for a given strictly convex function \( f \). With \( f = \frac{1}{2} x^2 \) the divergence (3.10) reduces to the familiar Euclidean distance \( \frac{1}{2}(x - y)^2 \), while for \( f(x) = x \log x \) it turns into the (generalized) KL Divergence. For tensors, we extend Definition (3.10) by considering separable Bregman divergences, so that

\[
B_f(X, Y) = \sum_{i_1, \ldots, i_m} B_f(x_{i_1 \ldots i_m}, y_{i_1 \ldots i_m}).
\]

Let \( \sigma_U \) and \( \sigma_L \) be upper and lower bounds, respectively, with \( \sigma_L > 0 \), such that

\[
\sigma_L B_f(x, y) \leq ||x - y||^2 \leq \sigma_U B_f(x, y) \tag{3.11}
\]

for all \( x, y \) in the convex hull of the entries of the given tensor \( A \). For KL-divergence, the data must then be bounded away from zero.

Since the means tensor is the best representative \( \arg\min_X B_f(A, R : A) \) for a clustering \( R \), we again use use projection matrices to express clusterings. Let \( Q^1_i \) be, as above, the full combination of projection matrices from dimension-wise clustering, and \( F = \arg\min_Q B_f(A, Q : A) \) the optimal
m-dimensional tensor clustering. Then we know that

\[
B_f(A, Q_i^j) \leq \sigma_U \|A, Q_i^j\|^2 \\
\leq \sigma_U 2^{\log_2 m/t} \max_j \|A - Q_0^j \cdot A\|^2 \\
\leq \frac{\sigma_U}{\sigma_L} 2^{\log_2 m/t} D(A, Q_0^j \cdot A) \\
\leq \frac{\sigma_U}{\sigma_L} 2^{\log_2 m/t} B_f(A, F \cdot A),
\]

so \( \rho_d = \frac{\sigma_U}{\sigma_L} \). Inequality (3.12) follows from Lemma 3.4 and Inequality (3.13) from an argumentation analogous to Equation (3.6).

Curvature bounds as in (3.11) seem to be necessary for Bregman divergences to guarantee constant approximation factors for the underlying 1D clustering—this intuition is reinforced by the results of [14], who avoided such curvature assumptions and had to be content with a non-constant \( O(\log n) \) approximation factor for information theoretic clustering.

### 3.5 Implications

To obtain concrete bounds for a variety of tensor clustering problems, we can use Theorem 3.1 for \( t = 1 \) or \( t = 2 \) with existing 1D approximation factors \( \alpha_i \) from the literature. Table 1 summarizes the results.

#### 3.5.1 1D factors for Metric and Bregman clustering

The \((1 + \epsilon)\) approximation factor for 1D clustering by Ackermann et al. [2] applies to all metrics. It leads to an \( m \)-dimensional approximation factor of \( \alpha_m = p(m/t)(1 + \epsilon) \). Arthur and Vassilvitskii [6] prove a guarantee in expectation of \( \alpha_1 = 8(\log K + 2) \) for \( K \) clusters with Euclidean k-means, resulting in an expected \( \alpha_m = 8p(m/t)(\log K + 2) \).

For Bregman clustering, we arrive at similar results with the approximation factor by Ackermann and Blömer [1] or the extension of [6] in [30, 35].

#### 3.5.2 Hilbertian metrics

A special example of metrics are Hilbertian metrics [24, 33] that arise from conditionally positive definite (CPD) kernels. A real valued function \( C : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R} \) is called a conditionally positive definite (CPD) kernel on \( \mathcal{S} \) if for any positive integer \( n \), any choice of \( n \) elements \( x_i \in \mathcal{S} \), \( (i)_1^n \equiv i = 1, \ldots, n \) and any choice of \( n \) reals \( u_i \in \mathbb{R} \) such that \( \sum_i u_i = 0 \), we have \( \sum_{i,j=0}^n u_i u_j C(x_i, x_j) \geq 0 \) \([11, 33]\). The following remarkable result \([32]\) connects CPD kernels and Hilbertian metrics, i.e., metrics which can be isometrically embedded in Hilbert space: There exists a Hilbert space \( \mathcal{H} \) of real-valued functions on \( \mathcal{S} \), and a mapping \( \Phi : \mathcal{S} \rightarrow \mathcal{H} \) such that

\[
\|\Phi(x) - \Phi(y)\|^2 = -C(x, y) + \frac{1}{2}(C(x, x) + C(y, y)) = d_C(x, y),
\]

if and only if \( C(\cdot, \cdot) \) is a CPD kernel. Hence, given a CPD kernel \( C \), one can construct a Hilbertian metric \( d_C(x, y) \) which behaves like the squared Euclidean distance in the Hilbert space. The corresponding kernel is \( K(x, y) = \frac{1}{2}(C(x, y) - C(x, a) - C(y, a) + C(a, a)) \) for some fixed \( a \in \mathcal{S} \).

Here, we choose \( \mathcal{S} \subseteq \mathbb{R} \) and define the distance of tensors \( X, Y \) as

\[
d_C(X, Y) = \sum_{i_1, \ldots, i_m} d_C(x_{i_1, \ldots, i_m}, x_{i_1, \ldots, i_m}).
\]

Since the argument by [6] for their kmeans++ is independent of the dimensionality, it can be generalized from Euclidean distance to distances in a Hilbert space.
Lemma 3.5 (1D Hilbertian Metric Clustering). For any 1D clustering with a Hilbertian metric $d_C$, one can construct a \text{kmeans++} based initialization followed by iterative updates using kernel k-means such that if $C$ is the final clustering, then
\[ E[J(C)] \leq 8(\log K + 2)J_{OPT}. \] (3.14)

Proof. Using $d_C(x, y) = \|\Phi(x) - \Phi(y)\|^2$, we can use the initialization by [6] in the Hilbert space on the mapped data points $\Phi(x)$, since it only depends on squared Euclidean distances or inner products, independent of the dimensionality of the space. Finally, the objective function can always be improved by running kernel k-means starting from the \text{kmeans++} initialization. \qed

Together with Theorem 3.1, Lemma 3.5 directly leads to a tensor clustering guarantee for Hilbertian metrics:
\[ E[J(C)] \leq 8m(\log K^* + 2)J_{OPT}(m), \] (3.15)
where $K^* = \max_{1 \leq j \leq m} k_j$ is the maximum number of clusters across all dimensions.

3.5.3 2D factor for binary $\ell_1$ clustering

Applying the results of [31] for binary matrices as $\alpha_2$ yields the slightly stronger bound for $\ell_1$ tensor clustering:
\[ J(C) \leq 3^{\log_2(m)} - 1 \{1 + \sqrt{2}\}\alpha_1J_{OPT}(m). \]

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

| Problem Name                  | Approx. Bound                              | Proof                      |
|-------------------------------|--------------------------------------------|----------------------------|
| Metric tensor clustering       | $J(C) \leq m(1 + \epsilon)J_{OPT}(m)$    | Thm. 3.1 + [2]             |
| Bregman tensor clustering      | $E[J(C)] \leq 8mc(\log K^* + 2)J_{OPT}(m)$| [3.11], Thm. 3.1 + [30, 35] (using [6]) |
| Bregman tensor clustering      | $J(C) \leq m\sigma_U\sigma_L^{-1}(1 + \epsilon)J_{OPT}(m)$ | [3.11], Thm. 3.1 + [1]   |
| Bregman co-clustering         | Above two results with $m = 2$             | as above                   |
| Hilbertian metrics            | $E[J(C)] \leq 8m(\log K^* + 2)J_{OPT}(m)$ | Thm. 3.1 + Lemma 3.5       |

4 Experiments

Our bounds depend strongly on the approximation factor $\alpha_t$ 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. 3.1 (for Euclidean) and Table 1 (2nd row, for KL-Divergence).

Further comparisons show that in practice, CoTeC is competitive with a greedy heuristic SiTeC (Simultaneous Tensor Clustering), 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.

Regarding divergences, we focus on Euclidean distance and KL-divergence to test CoTeC. To study the effect of the 1D method, we use two seeding methods for each divergence, uniform and distance-based drawing. The latter seeding ensures 1D approximation factors for $E[J(C)]$ by [6] for Euclidean clustering and by [30, 33] for KL-divergence.

We use each seeding by itself and as an initialization for \text{k-means} to get four 1D methods for each divergence. We refer to the CoTeC combination of the corresponding independent 1D clusterings by abbreviations:
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 $75 \times 75$, for which we randomly choose a $5 \times 5 \times 5$ tensor of means $M$ and cluster indicator matrices $C_i \in \{0, 1\}^{n_i \times 5}$. For clustering with Euclidean distances we add Gaussian noise (from $N(0, \sigma^2)$ with varying $\sigma$) to $A$, while for KL-Divergences we use the sampling method of [9] 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 $\alpha_m$ on a tensor, we divide the achieved objective $J(C)$ by the objective value of the “true” underlying tensor clustering. Figure 1 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. 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 $J(C)$ 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.

In summary, the empirical approximation factor does depend on the data, but in general seems to be lower than the theoretical worst-case value.

4.2 Experiments on real data

We further assess the behavior of CoTeC on a number of real-world gene expression data sets.

The first three of our data sets, Bcell (1332 × 62), AllAml (2088 × 72) and Breast (21906 × 77) are gene expression microarray data sets, and described in detail in [25]. Bcell is a lymphoma microarray dataset of chronic lymphotic leukemia, diffuse large Bcell leukemia and follicular lymphoma. During preprocessing only those genes were selected whose minimum expression level was above $e^{-1000}$. Microarray data for B-cell and T-cell acute lymphocytic leukemia and acute myelogenous leukemia is collected in AllAml. Our data matrix is restricted to those genes whose ratio of maximum to

---

5 We thank Hyuk Cho for kindly providing us the preprocessed data.
Figure 1: 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 $\sigma$, the y axis the empirical approximation factor.
Conclusions

In this paper we presented a simple, and to our knowledge the first approximation algorithm for Bregman and metric tensor clustering. Our approximation factor grows linearly with the order $m$ of the tensor for Bregman divergences, and is slightly superlinear in $m$ for arbitrary metrics. It is always linear in the quality of the sub-clusterings.

Our experiments demonstrated the dependence of the multi-dimensional clustering on the single-dimensional clusterings, confirming the dependence stated in the theoretical bound. On real-world data, the approximation algorithm is also suitable as an initialization for a simultaneous co-clustering algorithm, and endows the latter with its approximation guarantees. In fact the approximation algorithm by itself can also yield reasonable results in practice.

In our experiments we used single-dimensional clusterings with guarantees for our overall approximation algorithm. An interesting direction for future work is the development of a simultaneous approximation algorithm, such as a specific co-clustering seeding scheme of multi-dimensional centers, which can be then used as a subroutine by our tensor clustering algorithm.
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. | | | | | | | Bcell, KL | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k_1$ | $k_2$ | $r$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ | $k_1$ | $k_2$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ |
| 5 3 6 | 6.00$\pm$10$^6$ | 20.98 | 18.37 | 26.44 | | | 5 3 7 | 3.37$\pm$10$^6$ | 15.01 | 20.87 | 21.13 | |
| s | 8.52 | 24.97 | 22.83 | 29.53 | s | | 1.53 | 14.31 | 20.43 | 20.26 | |
| 5 6 r | 5.94$\pm$10$^6$ | 30.88 | 26.09 | 34.72 | | | 5 6 r | 3.60$\pm$10$^6$ | 15.76 | 21.23 | 21.62 | |
| s | 16.97 | 33.35 | 32.06 | 37.33 | s | | 3.24 | 16.22 | 21.37 | 21.21 | |
| 20 4 6 | 5.75$\pm$10$^6$ | 31.66 | 20.05 | 33.05 | | | 20 4 6 | 3.37$\pm$10$^6$ | 17.59 | 22.23 | 23.26 | |
| s | 18.83 | 32.24 | 24.71 | 33.36 | s | | 10.54 | 18.44 | 22.99 | 22.98 | |
| 20 6 r | 5.56$\pm$10$^6$ | 49.13 | 35.26 | 50.37 | | | 20 6 r | 3.15$\pm$10$^6$ | 18.92 | 24.51 | 25.43 | |
| s | 34.97 | 50.55 | 43.93 | 51.66 | s | | 11.76 | 20.52 | 25.69 | 26.23 | |
| 50 4 6 | 5.63$\pm$10$^6$ | 31.10 | 17.77 | 31.76 | | | 50 4 6 | 3.20$\pm$10$^6$ | 15.70 | 20.12 | 21.97 | |
| s | 15.25 | 32.58 | 19.14 | 33.17 | s | | 9.61 | 17.24 | 20.85 | 21.33 | |
| 50 6 r | 5.18$\pm$10$^6$ | 47.55 | 34.63 | 48.41 | | | 50 6 r | 2.85$\pm$10$^6$ | 16.38 | 21.61 | 22.57 | |
| s | 36.22 | 49.83 | 43.77 | 50.55 | s | | 11.86 | 18.63 | 23.24 | 23.13 | |

| Breast, Euc | | | | | | | Breast, KL | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k_1$ | $k_2$ | $r$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ | $k_1$ | $k_2$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ |
| 5 2 r | 1.43$\pm$10$^6$ | 22.96 | 20.48 | 24.47 | | | 5 2 r | 2.70$\pm$10$^6$ | 8.08 | 12.81 | 12.23 | |
| s | 2.69 | 21.92 | 19.42 | 24.32 | s | | 1.77 | 7.98 | 13.19 | 12.38 | |
| 5 4 r | 1.42$\pm$10$^6$ | 26.49 | 25.85 | 27.30 | | | 5 4 r | 2.67$\pm$10$^6$ | 11.88 | 17.56 | 17.31 | |
| s | 10.38 | 26.72 | 26.67 | 27.95 | s | | 3.60 | 11.95 | 18.10 | 18.29 | |
| 10 2 r | 1.41$\pm$10$^6$ | 22.13 | 19.46 | 25.26 | | | 10 2 r | 2.66$\pm$10$^6$ | 8.01 | 11.44 | 12.37 | |
| s | 7.77 | 21.66 | 19.20 | 25.09 | s | | 2.45 | 7.96 | 12.34 | 12.46 | |
| 10 4 r | 1.57$\pm$10$^6$ | 26.36 | 24.09 | 28.93 | | | 10 4 r | 2.50$\pm$10$^6$ | 11.17 | 16.54 | 17.92 | |
| s | 9.79 | 26.87 | 26.44 | 29.90 | s | | 4.97 | 13.53 | 19.50 | 19.31 | |
| 20 2 r | 1.41$\pm$10$^6$ | 22.46 | 10.42 | 26.21 | | | 20 2 r | 2.63$\pm$10$^6$ | 6.27 | 9.72 | 9.95 | |
| s | 8.16 | 22.54 | 19.43 | 26.16 | s | | 2.93 | 8.78 | 11.69 | 11.61 | |
| 20 4 r | 1.57$\pm$10$^6$ | 27.95 | 23.44 | 31.74 | | | 20 4 r | 2.56$\pm$10$^6$ | 11.73 | 17.42 | 17.78 | |
| s | 10.55 | 28.31 | 25.83 | 32.44 | s | | 3.45 | 12.21 | 17.51 | 17.45 | |

(ii) $k_1$ $k_2$ | $r$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ | $k_1$ $k_2$ | $r$ | $x$ | $x_k$ | $x_c$ | $x_{kc}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 2 | 4.6 ± 2.4 | 1.2 ± 0.4 | 4.0 ± 1.6 | 1.8 ± 0.4 | | 5 2 | 5.2 ± 2.0 | 3.6 ± 2.0 | 4.9 ± 2.6 | 3.1 ± 1.8 | |
| 5 4 | 4.9 ± 1.8 | 1.0 ± 0.2 | 3.0 ± 0.9 | 1.2 ± 0.5 | | 5 4 | 5.6 ± 1.8 | 3.6 ± 1.9 | 4.4 ± 1.2 | 3.5 ± 1.4 | |
| 10 2 | 3.4 ± 1.4 | 2.0 ± 0.2 | 2.6 ± 1.0 | 2.0 ± 0.0 | | 10 2 | 4.0 ± 1.8 | 2.5 ± 1.0 | 4.4 ± 2.8 | 2.7 ± 1.7 | |
| 10 4 | 4.3 ± 1.8 | 2.0 ± 0.5 | 3.0 ± 0.9 | 2.1 ± 0.3 | | 10 4 | 5.1 ± 1.4 | 4.0 ± 1.7 | 5.2 ± 1.7 | 3.7 ± 1.3 | |
| 20 2 | 2.9 ± 1.3 | 2.0 ± 0.0 | 2.7 ± 1.0 | 2.0 ± 0.0 | | 20 2 | 3.6 ± 1.8 | 2.3 ± 0.9 | 3.2 ± 1.5 | 2.1 ± 0.5 | |
| 20 4 | 3.9 ± 1.3 | 2.1 ± 0.3 | 3.4 ± 1.8 | 2.0 ± 0.2 | | 20 4 | 5.2 ± 1.9 | 3.5 ± 1.8 | 4.3 ± 1.6 | 2.8 ± 1.2 | |
Table 3: (i) Improvement of CoTeC and SiTeC variants upon ‘r’ in %; the respective reference value \((J_2 \text{ for ‘r’})\) is shaded in gray. (ii) Average number of SiTeC iterations.

| \(k_1 \ k_2\) | AllAml, Euc. | AllAml, KL |
|----------------|-------------|-------------|
|                | \(k_1\)     | \(k_2\)     | \(k_1\)     | \(k_2\)     |
|                | \(x\)       | \(xk\)      | \(xc\)      | \(xkc\)     | \(x\)       | \(xk\)      | \(xc\)      | \(xkc\)     |
| \(5\ \ 3\ \ r\) | 6.06 \times 10^{-5} | 49.26 | 49.19 | 50.54 | 5.64 \times 10^{-5} | 43.92 | 47.14 | 46.73 |
|                | s           | 40.63 | 48.62 | 50.27 | 50.71 | s           | 33.39 | 43.12 | 46.68 | 46.44 |
| \(10\ \ 3\ \ r\) | 5.31 \times 10^{-5} | 47.02 | 47.01 | 48.69 | 4.67 \times 10^{-5} | 40.11 | 41.72 | 42.57 |
|                | s           | 40.83 | 48.65 | 49.51 | 50.10 | s           | 31.04 | 39.87 | 42.78 | 42.75 |
| \(20\ \ 3\ \ r\) | 4.37 \times 10^{-5} | 39.75 | 38.02 | 41.78 | 3.78 \times 10^{-5} | 29.29 | 32.67 | 32.24 |
|                | s           | 34.26 | 41.06 | 42.70 | 43.28 | s           | 20.58 | 29.74 | 33.90 | 34.07 |

Table 4: (i) Improvement of CoTeC and SiTeC variants upon ‘r’ in %; the respective reference value \((J_2 \text{ for ‘r’})\) is shaded in gray. (ii) Average number of SiTeC iterations.

| \(k_1 \ k_2\) | Leukemia, Euc. | Leukemia, KL |
|----------------|---------------|-------------|
|                | \(k_1\)     | \(k_2\)     | \(k_1\)     | \(k_2\)     |
|                | \(x\)       | \(xk\)      | \(xc\)      | \(xkc\)     | \(x\)       | \(xk\)      | \(xc\)      | \(xkc\)     |
| \(5\ \ 3\ \ r\) | 7.61 \times 10^{-5} | 5.48 | 5.77 | 6.74 | 1.82 \times 10^{-5} | 5.11 | 7.15 | 7.52 |
|                | s           | 0.17 | 5.54 | 5.73 | 6.78 | s           | 0.36 | 4.93 | 7.19 | 7.51 |
| \(10\ \ 3\ \ r\) | 7.57 \times 10^{-5} | 6.53 | 7.18 | 7.75 | 1.81 \times 10^{-5} | 6.00 | 8.13 | 8.76 |
|                | s           | 0.14 | 6.79 | 6.77 | 7.79 | s           | 0.44 | 6.08 | 8.18 | 8.76 |
| \(20\ \ 3\ \ r\) | 7.30 \times 10^{-5} | 3.79 | 5.97 | 7.25 | 1.71 \times 10^{-5} | 3.81 | 7.58 | 7.90 |
|                | s           | 0.33 | 3.75 | 5.54 | 7.25 | s           | -0.21 | 3.65 | 7.32 | 7.35 |
| \(75\ \ 2\ \ r\) | 7.15 \times 10^{-5} | 4.90 | 7.34 | 8.93 | 1.08 \times 10^{-5} | 4.74 | 9.31 | 9.35 |
|                | s           | 0.60 | 5.00 | 8.00 | 9.06 | s           | 1.08 | 5.16 | 9.70 | 9.75 |
| \(75\ \ 3\ \ r\) | 7.26 \times 10^{-5} | 3.66 | 5.87 | 6.89 | 1.66 \times 10^{-5} | 3.36 | 6.92 | 6.95 |
|                | s           | 0.02 | 3.67 | 5.23 | 6.88 | s           | -0.35 | 2.85 | 6.60 | 6.30 |

| \(k_1 \ k_2\) | Mill, Euc. | \(k_1 \ k_2\) | \(k_1 \ k_2\) |
|----------------|------------|---------------|---------------|
|                | \(x\)       | \(xk\)      | \(xc\)      | \(xkc\)     | \(rc\) | \(rkc\) | \(sc\) | \(skc\)     |
| \(3\ \ 2\ \ r\) | 6.52 \times 10^{-5} | 10.54 | 11.26 | 11.45 | 7.6 \times 10^{-5} | 4.5 \times 10^{-2} | 8.0 \times 10^{-2} | 4.6 \times 10^{-2} |
|                | s           | 1.41 | 10.62 | 11.20 | 11.46 | s           | 4.2 \times 10^{-2} | 2.0 \times 10^{-2} | 3.8 \times 10^{-1} | 2.0 \times 10^{-0} |
| \(50\ \ 3\ \ r\) | 6.83 \times 10^{-5} | 8.40 | 12.53 | 13.21 | 4.7 \times 10^{-2} | 2.0 \times 10^{-2} | 4.2 \times 10^{-2} | 2.1 \times 10^{-2} |
|                | s           | 1.12 | 8.23 | 12.35 | 13.17 | s           | 4.4 \times 10^{-2} | 2.0 \times 10^{-2} | 4.3 \times 10^{-2} | 2.0 \times 10^{-2} |
| \(75\ \ 3\ \ r\) | 6.75 \times 10^{-5} | 7.84 | 11.69 | 12.52 | 4.4 \times 10^{-2} | 2.0 \times 10^{-2} | 4.3 \times 10^{-2} | 2.0 \times 10^{-2} |
|                | s           | 0.84 | 7.86 | 11.68 | 12.52 | s           | 4.4 \times 10^{-2} | 2.0 \times 10^{-2} | 4.3 \times 10^{-2} | 2.0 \times 10^{-2} |
References

[1] M. R. Ackermann and Johannes Blömer. Coresets and Approximate Clustering for Bregman Divergences. In Proc. 20th ACM-SIAM Symposium on Discrete Algorithms (SODA ’09), 2009. To appear.

[2] M. R. Ackermann, J. Blomer, and C. Sohler. Clustering for metric and non-metric distance measures. In ACM-SIAM SODA, April 2008.

[3] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In IEEE CVPR, 2005.

[4] A. Anagnostopoulos, A. Dasgupta, and R. Kumar. Approximation algorithms for co-clustering. In PODS, 2008.

[5] S. A. Armstrong. Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics, 30:41–47, 2002.

[6] D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. In SODA, pages 1027–1035, 2007.

[7] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. JMLR, 6(6):1705–1749, October 2005.

[8] A. Banerjee, S. Basu, and S. Merugu. Multi-way Clustering on Relation Graphs. In SIAM Data Mining, 2007.

[9] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. JMLR, 8:1919–1986, 2007.

[10] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In ICML, 2005.

[11] C. Berg, J. Christensen, and P. Ressel. Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions. Springer-Verlag, 1984.

[12] L. M. Bregman. The relaxation method of finding the common point of convex sets and its applications to the solution of problems in convex programming. U.S.S.R. Computational Mathematics and Mathematical Physics, 7(3):200–217, 1967.

[13] Y. Censor and S. A. Zenios. Parallel Optimization: Theory, Algorithms, and Applications. Oxford University Press, 1997.

[14] K. Chaudhuri and A. McGregor. Finding metric structure in information theoretic clustering. In Conf. on Learning Theory, COLT, July 2008.

[15] H. Cho and I. Dhillon. Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 5(3):385–400, 2008.

[16] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum Sum Squared Residue based Co-clustering of Gene Expression data. In Proc. 4th SIAM International Conference on Data Mining (SDM), pages 114–125, Florida, 2004. SIAM.

[17] V. de Silva and L.-H. Lim. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. SIAM J. on Matrix Analysis and Applications, 30(3):1084–1127, 2008.

[18] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In Proc. ACM SIGKDD 2003, pages 89–98, 2003.
[19] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.

[20] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caliguri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[21] V. M. Govindu. A tensor decomposition for geometric grouping and segmentation. In *IEEE CVPR*, 2005.

[22] W. H. Greub. *Multilinear Algebra*. Springer, 1967.

[23] J. A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[24] M. Hein and O. Bosquet. Hilbertian metrics and positive definite kernels on probability measures. In *AISTATS*, 2005.

[25] Y. Kluger, R. Basri, and J. T. Chang. Spectral biclustering of microarray data: Co-clustering genes and conditions. *Genome Research*, 13:703–716, 2003.

[26] T. G. Kolda and J. Sun. Scalable Tensor Decompositions for Multi-aspect Data Mining. In *ICDM*, 2008.

[27] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time (1 + $\epsilon$)-approximation algorithms for k-means clustering in any dimensions. In *IEEE Symp. on Foundations of Comp. Sci.*, 2004.

[28] S. P. Lloyd. Least squares quantization in PCM. *IEEE Tran. on Inf. Theory*, 28(2):129–136, 1982.

[29] B. Long, X. Wu, and Z. Zhang. Unsupervised learning on k-partite graphs. In *SIGKDD*, 2006.

[30] R. Nock, P. Luosto, and J. Kivinen. Mixed bregman clustering with approximation guarantees. In *Euro. Conf. on Mach. Learning (ECML)*, LNAI 5212, 2008.

[31] K. Puolamäki, S. Hanhijärvi, and G. C. Garriga. An approximation ratio for biclustering. *Inf. Process. Lett.*, 108(2):45–49, 2008.

[32] I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of American Mathematical Society*, 44(3):522–536, 1938.

[33] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2001.

[34] A. Shashua, R. Zass, and T. Hazan. Multi-way Clustering Using Super-Symmetric Non-negative Tensor Factorization. *LNCS*, 3954:595–608, 2006.

[35] S. Sra, S. Jegelka, and A. Banerjee. Approximation algorithms for bregman clustering co-clustering and tensor clustering. Technical Report 177, MPI for Biological Cybernetics, Oct. 2008.

[36] H. Zha, C. Ding, T. Li, and S. Zhu. Workshop on Data Mining using Matrices and Tensors. KDD, 2008.
$\text{sig} = 1.0$

\begin{align*}
\log(k_1)
\end{align*}
\( \% \text{ impr.} \) vs. \( \sigma \)

- \( r/r \)
- \( r_k/r \)
- \( s/r \)
- \( s_k/r \)
