CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

Part I

Mathematical foundations

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

1 Tensor models: solution methods and applications

Shiqian Ma, Bo Jiang, Xiuzhen Huang, and Shuzhong Zhang

This chapter introduces several models and associated computational tools for tensor data analysis. In particular, we discuss: tensor principal component analysis, tensor low-rank and sparse decomposition models, and tensor co-clustering problems. Such models have a great variety of applications; examples can be found in computer vision, machine learning, image processing, statistics, and bio-informatics. For computational purposes, we present several useful tools in the context of tensor data analysis, including the alternating direction method of multipliers (ADMM), and the block variables optimization techniques. We draw on applications from the gene expression data analysis in bio-informatics to demonstrate the performance of some of the aforementioned tools.

1.1 Introduction

One rich source of *big data* roots is the high dimensionality of the data formats known as *tensors*. Specifically, a complex-valued m-dimensional or mth-order *tensor* (a.k.a. m-way multiarray) can be denoted by $\mathcal{F} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_m}$, whose dimension in the ith direction is $n_i, i = 1, \ldots, m$. Vector and matrix are special cases of tensor when $m = 1$ and $m = 2$, respectively. In the era of big data analytics, huge-scale dense data in the form of tensors can be found in different domains such as computer vision [1], diffusion magnetic resonance imaging (MRI) [2–4], the quantum entanglement problem [5], spectral hypergraph theory [6], and higher-order Markov chains [7]. For instance, a color image can be considered as 3D data with row, column, color in each direction, while a color video sequence can be considered as 4D data, where time is the fourth dimension. Therefore, how to extract useful information from these tensor data becomes a very meaningful task.

On the other hand, the past few years have witnessed an emergence of sparse and low-rank matrix optimization models and their applications in data sciences, signal processing, machine learning, bioinformatics, and so on. There have been extensive investigations on low-rank matrix completion and recovery problems since the seminal

Big Data over Networks, ed. Shuguang Cui, Alfred O. Hero III, Zhi-Quan Luo, and José M. F. Moura.
Published by Cambridge University Press. © Cambridge University Press 2015.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

works of [8–11]. Some important variants of sparse and low-rank matrix optimization problems such as robust principal component analysis (PCA) [12, 13] and sparse PCA [14] have also been studied. A natural extension of the matrix to higher-dimensional space is the tensor. Traditional matrix-based data analysis is inherently two-dimensional, which limits its ability in extracting information from a multi-dimensional perspective. Tensor-based multi-dimensional data analysis has shown that tensor models can take full advantage of the multi-dimensional structures of the data, and generate more useful information. For example, Wang and Ahuja [1] reported that the images obtained by tensor PCA technique have higher quality than those from matrix PCA.

Stimulated by the need of big data analytics, and motivated by the success of compressed sensing and low-rank matrix optimization, it is important and timely to study methods for analyzing massive tensor data.

Before proceeding let us introduce notations that will be used throughout this chapter. We use $\mathbf{R}^n$ to denote the n-dimensional Euclidean space. A tensor is usually denoted by a calligraphic letter, as $\mathcal{A} = (\mathcal{A}_{i_1 i_2 \cdots i_m})_{n_1 \times n_2 \times \cdots \times n_m}$. The space where $n_1 \times n_2 \times \cdots \times n_m$-dimensional real-valued tensor resides is denoted by $\mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$. We call $\mathcal{A}$ super-symmetric if $n_1 = n_2 = \cdots = n_m$ and $\mathcal{A}_{i_1 i_2 \cdots i_m}$ is invariant under any permutation of $(i_1, i_2, \ldots, i_m)$, i.e., $\mathcal{A}_{i_1 i_2 \cdots i_m} = \mathcal{A}_{\pi(i_1, i_2, \ldots, i_m)}$, where $\pi(i_1, i_2, \ldots, i_m)$ is any permutation of indices $(i_1, i_2, \ldots, i_m)$. The space where $\underbrace{n \times n \times \cdots \times n}_{m}$ super-symmetric tensors reside is denoted by $\mathbf{S}^{n^m}$. Special cases of tensors are vector ($m = 1$) and matrix ($m = 2$), and tensors can also be seen as a long vector or a specially arranged matrix. For instance, the tensor space $\mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$ can also be seen as a matrix space $\mathbf{R}^{(n_1 \times n_2 \times \cdots \times n_{m_1}) \times (n_{m_1+1} \times n_{m_1+2} \times \cdots \times n_m)}$, where the row is actually an m_1-way array tensor space and the column is another $(m - m_1)$-dimensional tensor space. Such connections between tensor and matrix re-arrangements will play an important role in this chapter. As a convention in this chapter, if there is no other specification we shall adhere to the Euclidean norm (i.e. the L_2-norm) for vectors and tensors; in the latter case, the Euclidean norm is also known as the Frobenius norm, and is sometimes denoted as $\|\mathcal{A}\|_F = \sqrt{\sum_{i_1, i_2, \ldots, i_m} \mathcal{A}^2_{i_1 i_2 \cdots i_m}}$. For a given matrix X, we use $\|X\|_*$ to denote the nuclear norm of X, which is the sum of all the singular values of X. Regarding the products, we use $\otimes$ to denote the outer product for tensors; that is, for $\mathcal{A}_1 \in \mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$ and $\mathcal{A}_2 \in \mathbf{R}^{n_{m+1} \times n_{m+2} \times \cdots \times n_{m+\ell}}$, $\mathcal{A}_1 \otimes \mathcal{A}_2$ is in $\mathbf{R}^{n_1 \times n_2 \times \cdots \times n_{m+\ell}}$ with

$$(\mathcal{A}_1 \otimes \mathcal{A}_2)_{i_1 i_2 \cdots i_{m+\ell}} = (\mathcal{A}_1)_{i_1 i_2 \cdots i_m} (\mathcal{A}_2)_{i_{m+1} \cdots i_{m+\ell}}.$$

The inner product between two tensors $\mathcal{A}_1$ and $\mathcal{A}_2$ residing in the same space $\mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$ is denoted

$$\mathcal{A}_1 \bullet \mathcal{A}_2 = \sum_{i_1, i_2, \ldots, i_m} (\mathcal{A}_1)_{i_1 i_2 \cdots i_m} (\mathcal{A}_2)_{i_1 i_2 \cdots i_m}.$$

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

Under this light, a multi-linear form $\mathcal{A}(x^1, x^2, \ldots, x^m)$ can also be written in inner/outer products of tensors as

$$\mathcal{A} \bullet (x^1 \otimes \cdots \otimes x^m) := \sum_{i_1,\ldots,i_m} \mathcal{A}_{i_1,\ldots,i_m} (x^1 \otimes \cdots \otimes x^m)_{i_1,\ldots,i_m} = \sum_{i_1,\ldots,i_m} \mathcal{A}_{i_1,\ldots,i_m} \prod_{k=1}^{m} x_{i_k}^k.$$

1.2 Tensor models

1.2.1 Sparse and low-rank tensor optimization models

We first consider the common-background and sparse-foreground decomposition for the tensor data. To this end, we propose two tensor models below. The first model is to write a given tensor $\mathcal{A} \in \mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$ as the sum of three tensors: $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$. That is $\mathcal{A} = \mathcal{X} + \mathcal{Y} + \mathcal{Z}$, while $\mathcal{X}$ is in the form of $\mathcal{X} = \bar{\mathcal{X}} \otimes e$ where $\bar{\mathcal{X}} \in \mathbf{R}^{n_1 \times n_2 \times \cdots \times n_{m-1}}$ is a $(m-1)$-dimensional tensor and e is the all-one vector, and $\mathcal{Z}$ is the noise tensor. Specifically, the model in question is given by [15]

$$\begin{array}{ll} \min & \|\mathcal{Y}\|_1 \\ \text{s.t.} & \bar{\mathcal{X}} \otimes e + \mathcal{Y} + \mathcal{Z} = \mathcal{A} \\ & \|\mathcal{Z}\|_F \le \delta. \end{array} \tag{1.1}$$

Note that $\mathcal{A}$ thus has a common-tensor structure in the sense that all the $\mathbf{R}^{n_1 \times n_2 \times \cdots \times n_{m-1}}$-dimensional subtensors of $\mathcal{X}$ are the same. We now give more details about the physical meaning of model (1.1). For ease of presentation, we assume $m = 3$ at this moment. In this case, $\mathcal{A}$ consists of n_3 matrices $\mathcal{A}_1, \ldots, \mathcal{A}_{n_3}$ with the same size $n_1 \times n_2$. The equality constraint in (1.1) indicates that each matrix $\mathcal{A}_i$ can be decomposed into three parts: the common part (matrix $\bar{\mathcal{X}}$), the sparse part (matrix $\mathcal{Y}_i$), and the noisy part (matrix $\mathcal{Z}_i$). In many real applications, the last dimension in tensor $\mathcal{A}$ denotes time. Model (1.1) implies that the subtensors of $\mathcal{A}$ along time are almost the same, but different from each other with certain sparse changes captured in $\mathcal{Y}$ and small noises captured in $\mathcal{Z}$. By solving model (1.1) one can identify the common part and detect the changing part that results in significant difference among the subtensors. It should be pointed out that, even in the matrix case, our common-tensor model (1.1) is theoretically different from the low-rank + sparse decomposition of the robust PCA model proposed by Candes *et al.* [12] and Chandrasekaran *et al.* [13]. The L_1 norm in the objective of (1.1) naturally promotes the sparsity in tensor $\mathcal{Y}$. Recently, a similar model was also considered independently by Li *et al.* [16] in the context of image processing.

A common observation for huge-scale data analysis is that the data exhibit a low-dimensional property, or the most-representative part lies in low-dimensional subspace. Along with this line, we can model the background fluctuation by a low-rank tensor and achieve another optimization model:

$$\begin{array}{ll} \min & \text{rank}(\bar{\mathcal{X}}) + \rho\|\mathcal{Y}\|_1 \\ \text{s.t.} & \bar{\mathcal{X}} \otimes e + \mathcal{Y} + \mathcal{Z} = \mathcal{A} \\ & \|\mathcal{Z}\|_F \le \delta, \end{array} \tag{1.2}$$

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

where rank($\bar{\mathcal{X}}$) denotes the CP rank of $\bar{\mathcal{X}}$ and its precise definition can be described as follows.

Definition 1.1 Suppose $\mathcal{X} \in \mathbf{R}^{n_1 \times n_2 \times \cdots \times n_m}$, the CP rank of $\mathcal{X}$ denoted by rank($\mathcal{X}$) is the smallest integer r such that

$$\mathcal{F} = \sum_{i=1}^{r} a^{1,i} \otimes \cdots \otimes a^{m,i}, \tag{1.3}$$

where $a^{k,i} \in \mathbf{R}^{n_k}$ for all $1 \leq i \leq r$ and $1 \leq k \leq m$.

The idea of decomposing a tensor into an (asymmetric) outer product of vectors was first introduced and studied by Hitchcock in 1927 [17, 18]. This concept of tensor rank became popular after its rediscovery in the 1970s in the form of CANDECOMP (canonical decomposition) by Carroll and Chang [19] and PARAFAC (parallel factors) by Harshman [20]. Consequently, CANDECOMP and PARAFAC are further abbreviated as "CP" in the context of "CP rank" by many authors in the literature. In the next subsection, we will introduce the CP rank for super-symmetric tensors.

1.2.2 Tensor principal component analysis

Principal component analysis (PCA) plays an important role in applications arising from areas such as data analysis, dimension reduction, and bioinformatics, among others. PCA finds a few linear combinations of the original variables. These linear combinations, which are called principal components (PCs), are orthogonal to each other and explain most of the variance of the data. PCs provide a powerful tool to compress data along the direction of maximum variance to reach the minimum information loss.

Although the PCA and eigenvalue problem for the matrices have been well studied in the literature, the research of PCA for tensors is still underdeveloped. The tensor PCA is of great importance in practice and has many applications in computer vision [1], diffusion magnetic resonance imaging (MRI) [2–4], quantum entanglement problem [5], spectral hypergraph theory [6] and higher-order Markov chains [7]. Similar to its matrix counterpart, the problem of finding the PC that explains the most of the variance of a tensor $\mathcal{T}$ (with degree m) can be formulated as:

$$\begin{array}{ll} \min & \|\mathcal{T} - \lambda x^1 \otimes x^2 \otimes \cdots \otimes x^m\| \\ \text{s.t.} & \lambda \in \mathbf{R},\ \|x^i\| = 1, i = 1, 2, \ldots, m, \end{array} \tag{1.4}$$

which is equivalent to

$$\begin{array}{ll} \max & \mathcal{T}(x^1, x^2, \ldots, x^m) \\ \text{s.t.} & \|x^i\| = 1, i = 1, 2, \ldots, m. \end{array} \tag{1.5}$$

Let us call the above solution the *leading* PC. Once the leading PC is found, the other PCs can be computed sequentially via the so-called "deflation" technique. For instance, the second PC is defined as the leading PC of the tensor subtracting the leading PC from the original tensor, and so forth. The theoretical basis of such a deflation procedure for tensors is not exactly sound, although its matrix counterpart is well established (see

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

[21] and the references therein for more details). However, the deflation process does provide a heuristic way to compute multiple principal components of a tensor, albeit approximately. Thus in the rest of this paper, we focus on finding the leading PC of a tensor.

Problem (1.5) is also known as the best rank-one approximation of tensor $\mathcal{T}$, which has been studied in [22]. By embedding $\mathcal{T}$ into a larger tensor (for instance, see Section 8.4 in [23]), problem (1.5) can be reformulated as

$$\begin{array}{ll} \max & \mathcal{F}(x, x, \ldots, x) \\ \text{s.t.} & \|x\| = 1, \end{array} \tag{1.6}$$

where $\mathcal{F}$ is a super-symmetric tensor. Problem (1.6) is NP-hard and is called the maximum Z-eigenvalue problem in [24] and the nonlinear eigenproblem in [25]. Although a systematic study of the eigenvalues and eigenvectors for a real symmetric tensor was first conducted by Lim [26] and Qi [24] independently in 2005, Kofidis and Regalia in 2001 already showed that blind deconvolution can be formulated as a nonlinear eigenproblem [25]. Note that various methods have been proposed to find the Z-eigenvalues [27–31], which, however, may correspond only to local optimums, although some efforts on heuristics for finding global optimal solution were made (see, e.g., [25, 28]). In this chapter, we shall focus on finding the global optimal solution of (1.6).

In the subsequent analysis, for convenience we assume m to be even, i.e. $m = 2d$ in (1.6), where d is a positive integer, as this assumption is essentially not restrictive (see [23]). Therefore, we will focus on the following problem of largest eigenvalue of an even-order super-symmetric tensor:

$$\begin{array}{ll} \max & \mathcal{F}(\underbrace{x, \ldots, x}_{2d}) \\ \text{s.t.} & \|x\| = 1, \end{array} \tag{1.7}$$

where $\mathcal{F}$ is a $2d$th-order super-symmetric tensor. In particular, problem (1.7) can be equivalently written as

$$\begin{array}{ll} \max & \mathcal{F} \bullet \underbrace{x \otimes \cdots \otimes x}_{2d} \\ \text{s.t.} & \|x\| = 1. \end{array} \tag{1.8}$$

Now we introduce the so-called CP rank for even-order super-symmetric tensors.

Definition 1.2 Suppose $\mathcal{F} \in \mathbf{S}^{n^{2d}}$, the CP rank of $\mathcal{F}$ denoted by rank($\mathcal{F}$) is the smallest integer r such that

$$\mathcal{F} = \sum_{i=1}^{r} \lambda_i \underbrace{a^i \otimes \cdots \otimes a^i}_{2d}, \tag{1.9}$$

where $a_i \in \mathbf{R}^n$, $\lambda_i \in \{1, -1\}$.

Thus, given any $2d$th-order super-symmetric tensor form $\mathcal{F}$, we call it *rank one* if $\mathcal{F} = \lambda \underbrace{a \otimes \cdots \otimes a}_{2d}$ for some $a \in \mathbf{R}^n$ and $\lambda \in \{1, -1\}$.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

In the following, to simplify the notation, we denote

$$\mathbb{K}(n,d) = \left\{ k = (k_1, \ldots, k_n) \in \mathbb{Z}_+^n \;\middle|\; \sum_{j=1}^n k_j = d \right\}$$

and

$$\mathcal{X}_{1^{2k_1} 2^{2k_2} \ldots n^{2k_n}} := \mathcal{X}_{\underbrace{1\ldots1}_{2k_1}\underbrace{2\ldots2}_{2k_2}\ldots\underbrace{n\ldots n}_{2k_n}}.$$

By letting $\mathcal{X} = \underbrace{x \otimes \cdots \otimes x}_{2d}$ we can further convert problem (1.8) into:

$$\begin{array}{ll} \max & \mathcal{F} \bullet \mathcal{X} \\ \text{s.t.} & \sum\limits_{k \in \mathbb{K}(n,d)} \frac{d!}{\prod_{j=1}^n k_j!} \mathcal{X}_{1^{2k_1} 2^{2k_2} \ldots n^{2k_n}} = 1, \\ & \mathcal{X} \in \mathbf{S}^{n^{2d}}, \ \operatorname{rank}(\mathcal{X}) = 1, \end{array} \tag{1.10}$$

where the first equality constraint is due to the fact that

$$\sum_{k \in \mathbb{K}(n,d)} \frac{d!}{\prod_{j=1}^n k_j!} \prod_{j=1}^n x_j^{2k_j} = \|x\|^{2d} = 1.$$

Thus, the tensor PCA problem can be viewed as a tensor optimization problem with rank-one constraint, which is the extreme case of low-rank tensor optimization.

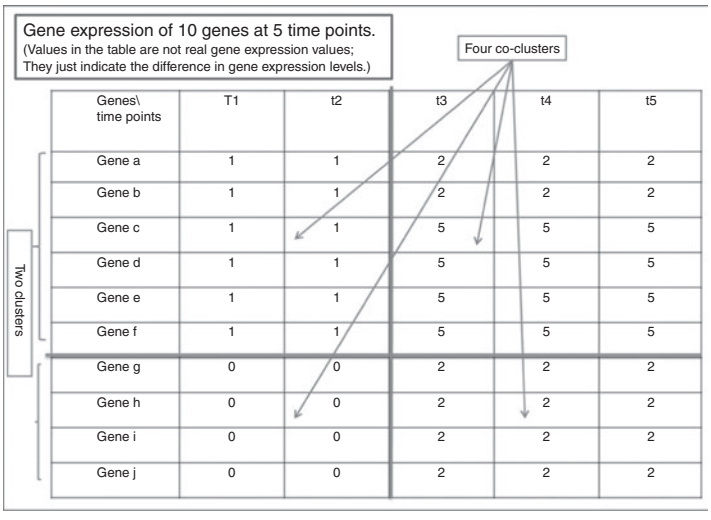
1.2.3 The tensor co-clustering problem

While genome data are relatively static, gene expression, which reflects gene activity, is highly dynamic. Patterns of gene expression change dramatically based on cell type, developmental stage, disease state, and in response to a wide variety of biological or environmental factors. In addition, both the kinetics and amplitude of changes in gene expression can have biological and biomedical significance. Gene expression of the cell could be used to infer the cell type, state, stage, and cell environment, and may indicate a homeostasis response or a pathological condition and thus relate to development of new medicines, drug metabolism, and diagnosis of diseases [32–34]. High-throughput gene expression techniques (such as microarray, next-generation sequencing and third-generation sequencing technologies) are generating huge amounts of high-dimensional genome-wide gene expression data (e.g. 4D with genes vs. timepoints vs. conditions vs. tissues). While the availability of these data presents unprecedented opportunities, it also presents major challenges for extractions of biologically meaningful information from the mountain-like gene expression data. In particular, it calls for effective computational models, equipped with efficient solution methods, to categorize gene expression data into biologically relevant groups in order to facilitate further functional assessment of important biological and biomedical processes. Classical clustering and co-clustering analysis of gene expression data is a worthy approach in this endeavor [35, 36] (Figure 1.1).

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information



Genes\ time points	T1	t2	t3	t4	t5
Gene a	1	1	2	2	2
Gene b	1	1	2	2	2
Gene c	1	1	5	5	5
Gene d	1	1	5	5	5
Gene e	1	1	5	5	5
Gene f	1	1	5	5	5
Gene g	0	0	2	2	2
Gene h	0	0	2	2	2
Gene i	0	0	2	2	2
Gene j	0	0	2	2	2

Figure 1.1 This figure illustrates the idea of clustering and co-clustering analysis. This is a table of 10 genes expression at five different time points. According to classical clustering, there are two clusters of genes {a, b, c, d, e, f}, {g, h, i, j}. For co-clustering, there could be four co-clusters, as shown.

Clustering as an effective approach, is usually applied to partition gene expression data into groups, where each group aggregates genes with similar expression levels. A lot of research has been conducted in clustering: cf. [37] for classical clustering in gene expression analysis, where the author discussed two classes of clustering (hierarchical clustering and partitioning), and three popular clustering methods (Eisen hierarchical clustering [38], *k*-means [39], and self-organizing map (SOM) method [40]). The classical clustering methods cluster genes into a number of groups based on their similar expression on all the considered conditions.

The concept of *co-clustering* was first introduced to 2D gene expression data analysis by Cheng and Church [41]. The co-clustering method can cluster genes and conditions simultaneously and thus can discover the similar expression of a certain group of genes on a certain group of conditions and vice versa. Readers may refer to [42] for a comprehensive comparison of the popular co-clustering approaches. Recently there are developed approaches for 3D gene expression data clustering analysis [43–46].

Essentially, the principle of current clustering and co-clustering models is to conduct *partitions* based on the assignment of a gene and/or a condition to a specific cluster or co-cluster. However, even a slightly less explicitly expressed function of the gene, which may be very important to know, can get lost under the principle of *sole assignment* of each gene to one co-cluster in the clustering analysis. In fact, it is widely known that one enzyme or a group of enzymes may get involved in more than one pathway, and one particular gene may be co-regulated with different groups of genes under different conditions and different development stages. The current clustering and co-clustering models are not designed to allocate more than one assignment per gene. Note that

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-107-09900-5 - Big Data Over Networks
Edited by Shuguang Cui, Alfred O. Hero III, Zhi-quan Luo and José M. F. Moura
Excerpt
More information

post-processing for merging identified clusters or co-clusters into overlapping groups [41, 47] could not address the issue. Motivated by this urgent need from the real-world gene expression data analysis, we develop a novel identification model based on tensor optimization that is capable of recognizing more than one assignment for one element, to better accommodate the reality of complex biological systems.

To illustrate the ideas, let us start by considering the conventional co-clustering formulation. Suppose that $\mathcal{A} \in \mathbf{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is a d-dimensional tensor. Let $I_j = \{1, 2, \ldots, n_j\}$ be the set of indices on the jth dimension, $j = 1, 2, \ldots, d$. We wish to find a p_j-partition of the index set I_j, say $I_j = I_1^j \cup I_2^j \cup \cdots \cup I_{p_j}^j$, where $j = 1, 2, \ldots, d$, in such a way that each of the *subtensor* $A_{I_{i_1}^1 \times I_{i_2}^2 \times \cdots \times I_{i_d}^d}$ is as tightly packed up as possible, where $1 \le i_j \le n_j$ and $j = 1, 2, \ldots, d$. The notion that plays an important role in our model is the so-called *mode product* between a tensor $\mathcal{X}$ and a matrix P. Suppose that $\mathcal{X} \in \mathbf{R}^{p_1 \times p_2 \times \cdots \times p_d}$ and $P \in \mathbf{R}^{p_i \times m}$. Then, $\mathcal{X} \times_i P$ is a tensor in $\mathbf{R}^{p_1 \times p_2 \times \cdots \times p_{i-1} \times m \times p_{i+1} \times \cdots \times p_d}$, whose $(j_1, j_2, \ldots, j_{i-1}, j_i, j_{i+1}, \ldots, j_d)$th component is defined by

$$(X \times_i P)_{j_1, j_2, \ldots, j_{i-1}, j_i, j_{i+1}, \ldots, j_d} = \sum_{\ell=1}^{p_i} X_{j_1, j_2, \ldots, j_{i-1}, \ell, j_{i+1}, \ldots, j_d} P_{\ell, j_i}.$$

Let $\mathcal{X}_{j_1, j_2, \ldots, j_{i-1}, j_i, j_{i+1}, \ldots, j_d}$ be the value of the co-cluster

$$(j_1, j_2, \ldots, j_{i-1}, j_i, j_{i+1}, \ldots, j_d) \text{ with } 1 \le j_i \le p_i, i = 1, 2, \ldots, d.$$

Let an assignment matrix $Y^j \in \mathbf{R}^{n_j \times p_j}$ for the indices for jth array of tensor $\mathcal{A}$ be:

$$Y_{ik}^j = \begin{cases} 1, & \text{if } i \text{ is assigned to the } k\text{th partition } I_k^j; \\ 0, & \text{otherwise.} \end{cases}$$

Then, we introduce a *proximity* measure $f(s) : \mathbf{R} \to \mathbf{R}_+$, with the property that $f(s) \ge 0$ for all $s \in \mathbf{R}$ and $f(s) = 0$ if and only if $s = 0$. The co-clustering problem can be formulated as

$$\begin{aligned} \min \quad & \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_d=1}^{n_d} f\left(\mathcal{A}_{j_1, j_2, \ldots, j_d} - (\mathcal{X} \times_1 Y^1 \times_2 Y^2 \times_3 \cdots \times_d Y^d)_{j_1, j_2, \ldots, j_d}\right) \\ \text{s.t.} \quad & \mathcal{X} \in \mathbf{R}^{p_1 \times p_2 \times \cdots \times p_d}, \\ & Y^j \in \mathbf{R}^{n_j \times p_j} \text{ is a row assignment matrix, } j = 1, 2, \ldots, d. \end{aligned} \tag{1.11}$$

We may consider a variety of proximity measures. For instance, if $f(s) = |s|^2$ then (1.11) can be written as

$$\begin{aligned} \min \quad & \|\mathcal{A} - \mathcal{X} \times_1 Y^1 \times_2 Y^2 \times_3 \cdots \times_d Y^d\|_F^2 \\ \text{s.t.} \quad & \mathcal{X} \in \mathbf{R}^{p_1 \times p_2 \times \cdots \times p_d}, \\ & Y^j \in \mathbf{R}^{n_j \times p_j} \text{ is a row assignment matrix, } j = 1, 2, \ldots, d. \end{aligned} \tag{1.12}$$

Note that our co-identification model could accommodate different evaluation and objective functions. Therefore, different co-clustering approaches previously developed in the literature could be considered as special cases of our approaches. Besides the norms L_1, L_2, L_∞, our model could use any Bregman divergence functions [48] instead;

© in this web service Cambridge University Press www.cambridge.org