CO-CLUSTERING AS MULTILINEAR DECOMPOSITION WITH SPARSE LATENT FACTORS

Evangelos E. Papalexakis, Nicholas D. Sidiropoulos

Dept. of ECE, TU Crete, 73100 Chania - Greece; (vagelis, nikos)@telecom.tuc.gr

ABSTRACT

The K-means clustering problem seeks to partition the columns of a data matrix in subsets, such that columns in the same subset are 'close' to each other. The co-clustering problem seeks to simultaneously partition the rows and columns of a matrix to produce 'coherent' groups called co-clusters. Co-clustering has recently found numerous applications in diverse areas. The concept readily generalizes to higher-way data sets (e.g., adding a temporal dimension). Starting from K-means, we show how co-clustering can be formulated as constrained multilinear decomposition with sparse latent factors. In the case of three- and higher-way data, this corresponds to a PARAFAC decomposition with sparse latent factors. This is important, for PARAFAC is unique under mild conditions - and sparsity further improves identifiability. This allows us to uniquely unravel a large number of possibly overlapping co-clusters that are hidden in the data. Interestingly, the imposition of latent sparsity pays a collateral dividend: as one increases the number of fitted co-clusters, new co-clusters are added without affecting those previously extracted. An important corollary is that co-clusters can be extracted incrementally; this implies that the algorithm scales well for large datasets. We demonstrate the validity of our approach using the ENRON corpus, as well as synthetic data.

1. INTRODUCTION

The classical (e.g., K-means) clustering problem seeks to partition the columns of a data matrix in subsets, such that columns in the same subset are 'close' to each other. The co-clustering problem seeks to simultaneously partition the rows and the columns of a matrix to produce 'coherent' groups called co-clusters. An example could be a customer vs. product data matrix, where one is not interested in clustering customers or products, but in jointly detecting subsets of customers buying select products (e.g., online retailing). This is also referred to as bi-clustering. Co-clustering has recently found numerous applications in diverse areas, ranging from the analysis of gene co-expression to network traffic and social network analysis [5, 8, 4, 10, 1]. The concept readily generalizes to higher-way data sets (e.g., adding a temporal dimension); yet there are very few papers dealing with three-way co-clustering (tri-clustering) [15, 11, 16] and no systematic study of three- and higher-way co-clustering, to the best of our knowledge. This is important because the algebraic properties of three- and higher-way data are very different from those of two-way (matrix) data; see, for example [13].

Hard co-clustering is a generalization of K-means, which is NP-hard. Several heuristic approaches and some disciplined approximations have been proposed in the literature for both hard and soft bi-clustering [5, 8, 4, 10, 1]. Starting from basic K-means and its extensions, we show how co-clustering can be formulated as a constrained multilinear decomposition with sparse latent factors. In the case of three- and higher-way data, this corresponds to a PARAFAC decomposition with sparse latent factors. This has important implications for co-clustering, for PARAFAC is unique under mild conditions; and sparsity further improves identifiability. This allows us to uniquely unravel a large number of possibly overlapping co-clusters that are hidden in the data - something impossible with matrix methods. Interestingly, the imposition of latent sparsity pays a collateral dividend: as one increases the number of fitted co-clusters, new co-clusters are added without affecting those previously extracted. This is not true for PARAFAC without latent sparsity - which is not, and generally cannot be an orthogonal decomposition, by virtue of uniqueness. An important corollary of this 'additivity' is that the co-clusters can be equivalently recovered one by one, in deflation mode. This is important because fitting a rank-one component is far easier computationally, implying that the approach remains operational even for large datasets. We demonstrate our methodology using the ENRON e-mail corpus, as well as synthetic data.

Supported by ARL/ERO W911NF-10-1-0464 and TU Crete seed funds.

2. FORMULATION

Clustering as a constrained outer product decomposition: Consider the familiar problem of clustering a set of vectors $\left\{\mathbf{x}_j \in \mathbb{R}^I\right\}_{j=1}^J$ in K clusters. The goal is to find $K << J$ cluster means $\left\{\boldsymbol{\mu}_k \in \mathbb{R}^I\right\}_{k=1}^K$ and an assignment of each $\mathbf{x}_j$ to a best-matching cluster $k^*(j)$ such that $\sum_j |\mathbf{x}_j - \boldsymbol{\mu}_{k^*(j)}|^2$ (or other suitable mismatch cost) is minimized. In matrix algebra terms, the problem can be posed as follows. Define $\mathbf{X} := [\mathbf{x}_1, \cdots, \mathbf{x}_J]$ $(I \times J)$, $\mathbf{M} := [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ $(I \times K)$, and an assignment matrix $\mathbf{A} := [\mathbf{a}_1, \cdots, \mathbf{a}_K]$ $(J \times K)$ having binary elements $\mathbf{A}(j,k) = \mathbf{a}_k(j) \in \{0,1\}$ and rows satisfying $\sum_{k=1}^K \mathbf{A}(j,k) = 1, \forall j$ (i.e., each row sums to 1). The most widely used version of the clustering problem, known as *K-means* clustering, can then be written as

$$\min_{\mathbf{M},\mathbf{A} \in \{0,1\}^{J \times K} \bigcap \mathcal{RS}} ||\mathbf{X} - \mathbf{M}\mathbf{A}^T||_F^2,$$

where $\mathcal{RS}$ denotes the set of matrices with the property that each row sums to 1. K-means clustering is NP-hard; for this reason, iterative algorithms based on the *Lloyd-Max* iteration are typically used to compute suboptimal solutions, often with good results.

Note that K-means clustering is equivalent to finding a best-fitting approximation (in the least squares sense) of the matrix $\mathbf{X}$ as a sum of K outer products

$$\min \left|\left|\mathbf{X} - \left(\boldsymbol{\mu}_1 \mathbf{a}_1^T + \cdots + \boldsymbol{\mu}_K \mathbf{a}_K^T\right)\right|\right|_F^2,$$

but the loadings in one mode are constrained: $\mathbf{A} \in \{0,1\}^{J \times K} \bigcap \mathcal{RS}$. The binary constraint $\mathbf{A}(j,k) = \mathbf{a}_k(j) \in \{0,1\}, \forall j,k$ corresponds to the usual case of 'hard' clustering: every data vector either belongs to a certain cluster or not. The $\mathcal{RS}$ constraint $\sum_{k=1}^K \mathbf{A}(j,k) =$

1, $\forall j$ ensures that every data vector belongs to one and only one cluster - no data vector is left 'orphan' and the clusters are non-overlapping. Relaxing the binary 0-1 constraints to non-negativity while maintaining the $\mathcal{RS}$ constraint corresponds to 'soft' clustering (overlapping clusters); the magnitude of $\mathbf{A}(j,k)$ now indicates how well $\mathbf{x}_j$ fits in cluster k. Replacing the $\mathcal{RS}$ constraint by its 'lossy' counterpart $\sum_{k=1}^{K}\mathbf{A}(j,k) \leq 1$ (or even dropping it altogether) emphasizes the extraction of tight clusters at the expense of not modeling 'outlying' data points. This is often well-justified in the context of exploratory data analysis. From this point on, we focus on soft lossy (co-) clustering. We also assume non-negative data, and impose non-negativity on all latent variables - as is often needed for interpretability.

Co-clustering as constrained outer product decomposition: K-means and related approaches cluster whole vectors - meaning that all elements of a given vector are considered when making clustering decisions, and vectors that are clustered together are ideally 'close' in each and every coordinate. A single cluster is modeled as a rank-one outer product plus noise: $\mathbf{C} = \boldsymbol{\mu}\mathbf{a}^T + \text{noise}$, where $\boldsymbol{\mu}$ is unconstrained and $\mathbf{a}$ is binary; i.e., $\mathbf{a}(j) \in \{0,1\}$, with 1's in those elements corresponding to columns of the data matrix that belong to the given cluster. The vector $\mathbf{a}$ will typically be sparse, because most data columns *will not* belong to any given cluster - at least when $K > 2$ and the cluster populations are roughly balanced.

There are many applications where certain vectors are close only for a certain subset of their elements, and we need to spot this pattern. A good example is gene expression data, where the rows of the data matrix $\mathbf{X}$ correspond to genes, the columns to experimental conditions, and the objective is to detect patterns of joint gene expression and the conditions under which this happens. Note that we do not know *a priori* which genes are expressed together, or under which conditions. Another example is marketing, where rows correspond to products, columns to customers, and the objective is *not* to cluster the products or the customers, but rather to detect (possibly overlapping) groups of customers that tend to buy certain subsets of products. This is the *co-clustering* (in this case *bi-clustering*) problem which has recently generated significant interest in numerous disciplines [5, 8, 4, 10, 1]. In social network analysis, co-clustering can be used to detect social groups (often called 'cliques') engaging in certain types of social behavior.

Whereas one-sided clustering involves selection (which columns belong to the given cluster) in one mode, co-clustering involves selection on both modes (rows and columns). This can be modeled as $\mathbf{G} = \mathbf{b}\mathbf{a}^T + \text{noise}$, where $\mathbf{b}$ and $\mathbf{a}$ are both sparse. When only relative expression matters, we can relax the binary constraint on the elements of $\mathbf{b}$ and $\mathbf{a}$, possibly retaining non-negativity when appropriate. Assuming non-negative data $\mathbf{X}(i,j) \geq 0, \forall i,j$, and focusing on overlapping (soft) lossy co-clustering, the problem can then be formulated as

$$\min_{\mathbf{B}\geq\mathbf{0},\mathbf{A}\geq\mathbf{0}} ||\mathbf{X} - \mathbf{B}\mathbf{A}^T||_F^2,$$

where the inequalities apply element-wise, and the columns of $\mathbf{A}$ $(J \times K)$ and $\mathbf{B}$ $(I \times K)$ should typically contain many zeros. One may attempt to use PCA or NMF [9] for co-clustering; however, the columns of $\mathbf{A}$ and $\mathbf{B}$ will be very dense, destroying all support information which is crucial in co-clustering applications. PCA imposes orthogonality, which is artificial and limits analysis to non-overlapping co-clusters if non-negativity is also imposed. Enforcing sparsity is ideally accomplished by penalizing the number of non-zero elements (the ℓ_0 norm), however this yields an intractable optimization problem. Recent research has shown that a practical alternative is to use an ℓ_1 penalty in lieu of the ℓ_0 norm. Enforcing sparsity can be achieved using alternating sparse regression [12]

$$\min_{\mathbf{B}\geq\mathbf{0},\mathbf{A}\geq\mathbf{0}} ||\mathbf{X} - \mathbf{B}\mathbf{A}^T||_F^2 + \lambda\sum_{i,k}|\mathbf{B}(i,k)| + \lambda\sum_{j,k}|\mathbf{A}(j,k)|\,.$$

Ignoring sparsity constraints, the problem is a non-negative matrix factorization, which is generally non-unique. Enforcing sparsity improves conditional uniqueness [14], although the uniqueness of sparse bilinear factorizations as above is currently an open problem.

Extension to three- and higher-way co-clustering: PARAFAC with sparse latent factors: In many cases, one works with data sets indexed by three or more variables, instead of two (as in matrices). A good example is several batches of gene expression data measured over several experimental conditions in two or more occasions or by different labs. Another is social network data, such as the ENRON e-mail corpus, where we have e-mail counts from sender to receiver as a function of time, stored in a three-way array $\underline{\mathbf{X}}$ whose (i,j,n)-th element $\underline{\mathbf{X}}(i,j,n)$ is the number of packets send by transmitting node i to receiving node j during time interval n. The natural generalization of bi-clustering to tri-clustering is to consider a trilinear outer product decomposition

$$\underline{\mathbf{X}} \cong \sum_{k=1}^{K} \mathbf{a}_k \odot \mathbf{b}_k \odot \mathbf{c}_k, \tag{1}$$

where $\odot$ stands for the outer product, i.e., $[\mathbf{a}_k \odot \mathbf{b}_k \odot \mathbf{c}_k]\,(i,j,n) = \mathbf{a}_k(i)\mathbf{b}_k(j)\mathbf{c}_k(n)$, and all vectors should be sparse. Without constraints, the above is known as the PARAFAC decomposition [6], and K is the exact or 'essential' rank of $\underline{\mathbf{X}}$, depending on whether one seeks an exact or approximate decomposition. Note that latent sparsity is key here, because the whole point of co-clustering is to *select* subsets along each mode. Even without sparsity, however, the PARAFAC decomposition is unique under relatively mild conditions - even in certain cases where $K >> \min(I,J)$ (e.g., see [13]). This means that our formulation of (overlapping and lossy) three-way co-clustering can *reveal the true latent patterns* in the data when used as an exploratory tool, even for a large number of co-clusters.

There are very few papers on tri-clustering in the literature [15, 11, 16] (note that tri-clustering is very different from K-means clustering of three-way data, as considered, e.g., in [7]). Off-the-shelf non-negative PARAFAC has been used for tri-clustering of web data in [16], albeit without motivation as to why it is an appropriate tool for co-clustering. A hybrid PARAFAC-Tucker model is proposed in [11], again without clear motivation regarding its application to co-clustering. Still, these are the closest pieces of work, and so will use non-negative PARAFAC as a baseline for comparison in our simulations. We underscore, however, that latent sparsity is key in our present context, because the whole point of co-clustering is to *select* subsets along each mode. Latent factor sparsity has not been considered in the aforementioned references, which did not start from a 'first principles' formulation, as we did.

One may wonder if there is a need to impose sparsity in our present context, in light of uniqueness of unconstrained (or non-negative) PARAFAC. The answer is two-fold. First, in practice we compute truncated PARAFAC approximations, instead of a full decomposition; noise and unmodeled dynamics will thereby render the extracted factors non-zero everywhere, with probability one. This destroys the support information that is crucial for co-clustering. Enforcing latent sparsity suppresses noise and automatically selects the desired support in all modes, simultaneously. Second, the imposition of sparsity (and non-negativity) improve uniqueness, thereby allowing stable extraction of more co-clusters than would otherwise be possible with plain PARAFAC. For these two reasons, sparsity constraints are very important here.

3. PARAFAC WITH SPARSE LATENT FACTORS

Motivated by the aforementioned considerations, we propose the following formulation of tri-clustering

$$\min_{\mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0},\mathbf{C}\geq\mathbf{0}} \|\underline{\mathbf{X}} - \sum_{k=1}^{K} \mathbf{a}_k \odot \mathbf{b}_k \odot \mathbf{c}_k\|_F^2 + \lambda \sum_{i,k} |\mathbf{A}(i,k)| + \lambda \sum_{j,k} |\mathbf{B}(j,k)| + \lambda \sum_{n,k} |\mathbf{C}(n,k)|, \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{I\times K}$, $\mathbf{B} \in \mathbb{R}^{J\times K}$, and $\mathbf{C} \in \mathbb{R}^{N\times K}$ contain vectors $\mathbf{a}_k, \mathbf{b}_k, \mathbf{c}_k$ respectively; $\geq$ is interpreted element-wise; and λ is a regularization parameter that trades-off sparsity for least-squares fit. PARAFAC is usually fitted using alternating least squares (ALS), wherein each of the three matrices is updated using least squares conditioned on the other two matrices in a cyclic fashion [6]. In plain ALS, each conditional update is a linear least squares problem, including non-negativity constraints when appropriate. In our context (which includes the ℓ_1 penalty terms), each conditional update can be shown to be tantamount to a Lasso problem [14], which can be solved optimally in a variety of ways. We use a simple element-wise coordinate descent algorithm for the Lasso step. In practice, we may initialize $\mathbf{A}, \mathbf{B}, \mathbf{C}$ using non-negative ALS (as implemented in the N-way Toolbox for Matlab [2]), followed by alternating Lasso. As we will see shortly, however, a fortuitous side-benefit of our formulation enables far simpler computation of the dominant co-clusters in an incremental fashion.

4. EXPERIMENTAL EVALUATION

ENRON e-mail corpus: We used a summary version of the ENRON dataset stored in a three-way array $\underline{\mathbf{X}}$ of size $168 \times 168 \times 44$, containing the number of e-mails exchanged between 168 employees over 44 months (spanning from 1998 to 2002), indexed by sender, receiver, and month. Similar to [3], we first compress the dynamic range of the raw data using a logarithmic transformation: each non-zero entry of $\underline{\mathbf{X}}$ is mapped to $x' = \log_2(x) + 1$. We then fit a non-negative PARAFAC model with sparse latent factors to extract the dominant co-clusters. In our present context, each co-cluster captures a 'clique' and its temporal evolution. Table 1 summarizes the extracted cliques, which turn out to match the structure of the organization remarkably well - e.g., the label 'legal' in Table 1 means that the corresponding co-cluster contains the employees in the legal department (plus/minus one employee in all cases reported in Table 1). Furthermore, Table 1 indicates that increasing K yields a nested sequence of co-clusters. This can also be appreciated by looking at the corresponding temporal co-cluster profiles for $K \in \{1, 2, 3\}$, shown in Fig. 1. We will soon return to this property, but for the moment let us provide further evidence that our co-clustering analysis passes sanity checks. The temporal profiles for $K = 5$ are plotted in Fig. 2. Two distinct peaks can be identified in the temporal communication patterns among the various cliques. Namely, the first peak is located in the months between the end of 2000 and the middle of 2001 (points 26-33 in Fig. 2), when a change of CEO occurred. The second peak corresponds to bankruptcy, and can be located in the months between September and November 2001 (points 36-38 in Fig. 2). In [3], four class labels are identified: *Legal, Executive/Govt. Affairs, Executive, Pipeline*. The same class labels are also identified in [11], where non-negative PARAFAC is used, among other methods. Our results (cf. Table 1) are consistent with [3, 11], but our cliques are far cleaner, containing fewer outliers due to the imposition of sparsity.

Synthetic Data: Starting from an all-zero array $\underline{\mathbf{X}}$ of size $80 \times 80 \times 8$, we implant three co-clusters: $\underline{\mathbf{X}}_{20:24,20:24,1:3} = 4\mathbf{I}_{5,5,3}$; $\underline{\mathbf{X}}_{40:44,70:74,2:5} = 2\mathbf{I}_{5,5,4}$; and $\underline{\mathbf{X}}_{37:41,73:77,4:8} = 4\mathbf{I}_{5,5,5}$, where $I_1 : I_2$ denotes a range of indices and $\mathbf{I}_{I,J,K}$ denotes an $I \times J \times K$ array of unit elements. We then add i.i.d. sparse Gaussian noise, to make the problem a bit more challenging. The number of co-clusters is set to $K = 3$. Fig. 3 shows the input data summed across the third (temporal) mode, and the co-clusters extracted using PARAFAC with non-negative sparse latent factors (NN SLF) for $\lambda = 12$. One can easily verify that the support of each co-cluster is perfectly recovered, and noise has been effectively removed. For comparison, the results of PARAFAC with non-negativity (NN) but without latent sparsity ($\lambda = 0$) are shown in Fig. 4. Observe how the two overlapping clusters have been merged into one; a 'phantom' co-cluster has emerged; and the loss of localization (leakage) due to noise. It is important to note here that thresholding the results of non-negative PARAFAC (Fig. 4) in a post-processing step may reduce leakage, but it will not recover the correct support information and the phantom co-cluster will of course remain.

Nesting revisited: In both cases considered above we observed that increasing K yields a nested sequence of co-clusters. This is not limited to the specific datasets, as we shall see next. It is important to note that 'plain' non-negative PARAFAC (corresponding to $\lambda = 0$) does not enjoy this property. This suggests that a sufficiently large λ is needed for the property to hold at least approximately. Indeed, we have observed that this qualitative property holds with increasing accuracy as λ increases, and for higher values of K. For numerical assessment, we simulated data containing two overlapping co-clusters, similar to the synthetic data used above. Table 2 summarizes how well the property holds for $\lambda = 5,\ 15$ and $\lambda = 0$ (the latter corresponds to plain non-negative PARAFAC). The effect of positive (and increasing) λ is evident. Qualitatively similar observations have been made on completely unstructured (fully random) data.

5. DISCUSSION

Starting from first principles, we have formulated a new approach to co-clustering as constrained multilinear decomposition with sparse latent factors. In future work, we plan to investigate i) automatic ways of choosing λ; ii) using imbalanced sparsity penalties for the different modes; iii) theoretically justifying the nesting property observed in experiments; and iv) analyzing the uniqueness potential of PARAFAC with latent sparsity.

6. REFERENCES

[1] A. Anagnostopoulos, A. Dasgupta, R. Kumar, "Approximation Algorithms for Co-Clustering," in *Proc. PODS 2008*, June 9-12, 2008, Vancouver, BC, Canada.

[2] C.A. Andersson, R. Bro, "The N-way Toolbox for MATLAB," *Chemometrics and Intelligent Laboratory Systems*, 2000; see `http://www.models.life.ku.dk/nwaytoolbox`

[3] B.W. Bader, R.A. Harshman, T.G. Kolda, "Temporal analysis of social networks using three-way DEDICOM," Sandia National Laboratories TR SAND2006-2161, 2006.

[4] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, "A generalized maximum entropy approach to Bregman co-clustering and matrix approximation," *Journal of Machine Learning Research*, vol. 8, pp. 1919-1986, Aug. 2007.

[5] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pp. 93-103, AAAI Press, 2000.

[6] R.A. Harshman, "Foundations of the Parafac procedure: models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1-84, 1970.

[7] H. Huang, C. Ding, D. Luo, T. Li, "Simultaneous tensor subspace selection and clustering: the equivalence of high order SVD and k-means clustering," in *Proc. 14th ACM SIGKDD*, pp. 327-335, 2008.

[8] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: Coclustering genes and conditions," *Genome Research*, vol. 13, pp. 703-716, 2003.

[9] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[10] S.C. Madeira and A.L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24-45, Jan.-Mar. 2004.

[11] W. Peng, T. Li, "Temporal relation co-clustering on directional social network and author-topic evolution," *Knowledge and Information Systems*, pp. 1-20, March 2010; DOI 10.1007/s10115-010-0289-9

[12] I. Schizas, G.B. Giannakis, and N.D. Sidiropoulos, "Exploiting Covariance-domain Sparsity for Dimensionality Reduction," in *Proc. IEEE CAMSAP 2009*, Dec. 13-16, 2009, Aruba, Dutch Antilles.

[13] T. Jiang, N.D. Sidiropoulos, "Kruskals Permutation Lemma and the Identification of CANDECOMP/PARAFAC and Bilinear Models with Constant Modulus Constraints," *IEEE Trans. on Signal Processing*, vol. 52, no. 9, pp. 2625-2636, Sept. 2004.

[14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[15] L. Zhao, M.J. Zaki, "Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data," in *Proc. ACM SIGMOD 2005*, p. 705.

[16] Q. Zhou, G. Xu, Y. Zong, "Web Co-clustering of Usage Network Using Tensor Decomposition," in *Proc. 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, pp. 311-314, 2009.

$K = 1$	Legal	-	-	-	-
$K = 2$	Legal	Executive, Govt. affairs	-	-	-
$K = 3$	Legal	Executive, Govt. affairs	Trading	-	-
$K = 4$	Legal	Executive, Govt. affairs	Trading	Pipeline	-
$K = 5$	Legal	Executive	Executive, Govt. affairs	Trading	Pipeline

Table 1. Extracted co-clusters for ENRON ($\lambda = 30$)

PARAFAC w/ NN SLF	**A**-mode error	**B**-mode error	**C**-mode error
noisless ($\lambda = 5$)	2.4×10^{-5}	1.0×10^{-5}	1.3×10^{-5}
noisless ($\lambda = 15$)	5.8×10^{-5}	4.0×10^{-5}	1.8×10^{-5}
noisy ($\lambda = 5$)	3.7×10^{-6}	5.4×10^{-6}	1.0×10^{-5}
noisy ($\lambda = 15$)	1.7×10^{-6}	3.8×10^{-6}	2.1×10^{-6}
PARAFAC w/ NN	**A**-mode error	**B**-mode error	**C**-mode error
noisless	0.1597	0.0024	0.0048
noisy	0.4479	0.0076	0.002

Table 2. Numerical assessment of 'nesting'. $\mathbf{A}$-mode error := max(abs($\mathbf{A}_2(:,1) - \mathbf{A}_3(:,1)$)), where the subscript denotes the fitted rank (number of co-clusters); and likewise for modes $\mathbf{B}$ and $\mathbf{C}$.

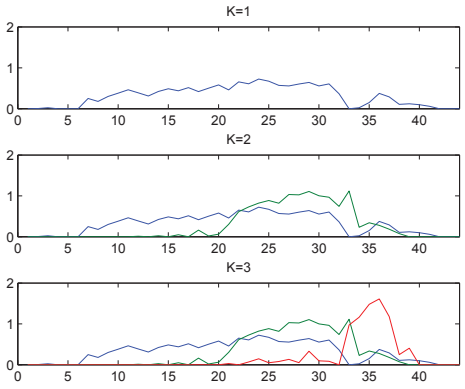

Fig. 1. Temporal co-cluster profiles for $K \in \{1, 2, 3\}$ and $\lambda = 30$.

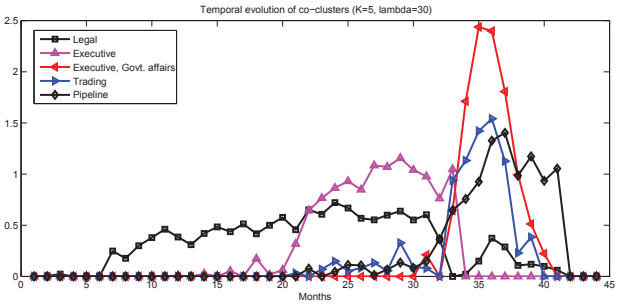

Fig. 2. Temporal co-cluster profiles for $K = 5$ amd $\lambda = 30$.

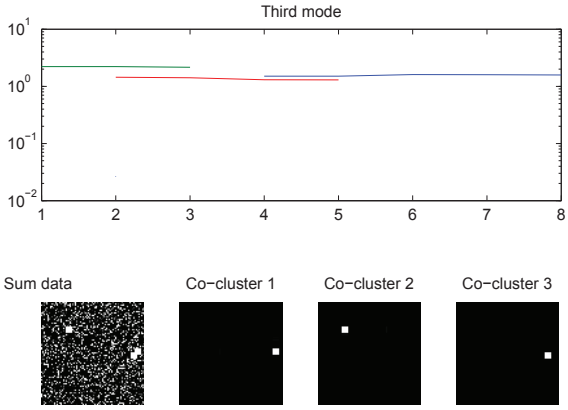

Fig. 3. Overlapping co-clusters: PARAFAC w/ NN SLF ($\lambda = 12$)

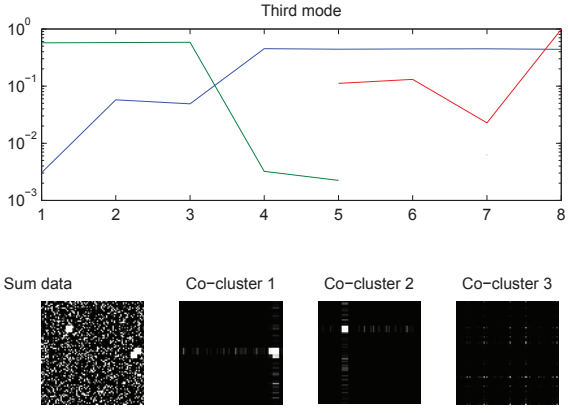

Fig. 4. Overlapping co-clusters: PARAFAC w/ NN ($\lambda = 0$)