Apprentissage spectral d'une représentation unidimensionnelle pour la visualisation de données

Lazhar Labiod, Mohamed Nadif

LIPADE, Université Paris Descartes,45 rue des Saints Pères Paris 75006, France

Résumé La visualisation des données qui a attiré une très grande attention cette dernière décennie est un outil puissant pour une meilleure compréhension des données. Dans ce papier, nous proposons un nouveau cadre théorique pour la visualisation des données. Ce cadre est basé sur une décomposition en valeurs singulières (SVD) de rang 1 et s'appuie sur la recherche des vecteurs singuliers appropriés de la matrice de données normalisées A. Cela implique le calcul d'une décomposition tronquée en valeurs singulières, particulièrement appropriée à la construction d'une incorporation unidimensionnelle pour les lignes et colonnes de A. La visualisation simplifiée et informative de A consiste en une simple permutation des lignes et des colonnes selon les premiers vecteurs singuliers à gauche et à droite ordonnés. En effet, cette tâche permet une réorganisation optimale de A en blocs homogènes conduisant à réveler une structure facilement interprétable. Enfin, nous relions notre approche au domaine du co-clustering spectral et montrons son utilité dans ce contexte.

1 Introduction

Prominent authors in the discipline of information visualization [1] have identified that the data mining community gives minimal attention to information visualization. However, they pointed that there are hopeful signs that the narrow bridge between data mining and information visualization will be expanded in the coming years. Bertin [5] has described the visualization procedure as *simplifying without destroying* and was convinced that simplification was *no more than regrouping similar things*. Spath [2] considered such matrix permutation approaches to have a great advantage in contrast to the cluster algorithms, *because no information of any kind is lost, and because the number of clusters does not have to be presumed, it is easily and naturally visible*. Murtagh [3], Arabie and Hubert [4] have referred to similar advantages calling such an approach *a non-destructive data analysis*, emphasizing the essential property that no transformation or reduction of the data itself takes place.

In certain problems it may be useful to perform co-clustering, where both objects and features are assigned to groups simultaneously. One approach to the co-clustering problem is to view it as the task of partitioning a weighted bipartite graph. Dhillon [8] proposed a spectral approach to approximate the optimal

normalised cut of a bipartite graph, which was applied for document clustering. This involves computing a truncated singular value decomposition (SVD) of a suitably normalized term-document matrix, constructing an embedding of both terms and documents, and applying k-means to this embedding to produce a simultaneous k-way partitioning of both documents and terms. Finally, data visualization is obtained by reorganizing rows and columns data according to the co-clustering result. Despite the advantages of co-clustering, all methods require the knowledge of the number of blocks, in this paper we will not tackle co-clustering but we will see how an appropriate visualization of data leads to a reorganization into homogeneous blocks, and we will show the usefulness of our approach in the context of co-clustering. We propose a new theoretical framework, specifically we develop an efficient iterative procedure to find one dimensional embedding of both rows and columns data. This involves an optimal simultaneous rows and columns data reordering. We show that the solution is given by the leading left and right singular vectors of data matrix.

The rest of paper is organized as follows. Section 2 introduces the problem formulation and the aims. Section 3 is devoted to the proposed algorithm for data visualization. Section 4 presents numerical experiments on real and simulated data. Finally, the conclusion summarizes the advantages of our contribution.

2 Problem formulation

Let A be a $m \times n$ data matrix, it can be viewed as a weighted bipartite graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. G is said to be bipartite if its vertices can be partitioned into two sets I and J such that every edge in E has exactly one end in I and the other in $J : V = I \cup J$. The data matrix A can be viewed as a weighted bipartite graph where each node i in I corresponds to a row and each node j in J corresponds to a column. The edge between i and j has weight a_{ij} denoting the element of the matrix in the intersection between row i and columns j. For convenience of discussion we also call the vertices in I as the the documents (rows) while vertices in J as words (columns). The adjacency matrix of a bipartite graph is :

$$\mathbf{B} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{1}$$

We define a columns stochastic data matrix associated to $\mathbf{B}$ as follow,

$$\mathbf{S} = \mathbf{D}^{-1}\mathbf{B} = \begin{bmatrix} 0 & D_r^{-1}A \\ D_c^{-1}A^T & 0 \end{bmatrix} \quad \text{where} \quad \mathbf{D} = \begin{bmatrix} D_r & 0 \\ 0 & D_c \end{bmatrix} \tag{2}$$

The matrices D_r and D_c, are diagonal such that $D_r = diag(A\mathbb{1})$ and $D_c = diag(A^T\mathbb{1})$. Let us focus our attention on the first leading eigenvectors of $\mathbf{S}$, since $\mathbf{S}$ is nonnegative and stochastic, due to the Frobenius theorem the first vector is also nonnegative and constant. The power method is the well known

technique used to compute the leading eigenvector of $\mathbf{S}$. The power method consists in the following iterative process :

$$\pi^{(t)} = \mathbf{S}^{(t)}\pi^{(0)} \quad \text{and} \quad \pi^{(t)} = \frac{\pi^{(t)}}{||\pi^{(t)}||}. \tag{3}$$

By Perrons-Frobenius theorem [6] all eigenvalues are real and are contained in $[-1, 1]$. Since $\mathbf{S}$ is stochastic, it is known that for every right eigenvector there is a corresponding left eigenvector that corresponds to the same eigenvalue $\lambda_1 = 1$, the greatest eigenvalue and called the Perron root. The right eigenvector corresponding to the uniform distribution $(\frac{1}{m+n}, ..., \frac{1}{m+n}, ..., \frac{1}{m+n})^T$. the corresponding left eigenvector $\pi = \mathbb{1}$ represents the constant left eigenvector of $\mathbf{S}$ so that $\pi^T\mathbb{1} = m + n$. In the matrix notation we have $\pi = \mathbf{S}\pi$ and $\mathbf{S}\mathbb{1} = \mathbb{1}$.

At first sight, this process might seem uninteresting since it eventually leads to a vector with all rows and columns coincide for any starting vector. However our practical experience shows that, first the vector π very quickly collaps into rows and columns blocks and these blocks move towards each other relatively slowly. If we stop the power method iteration at this point, the algorithm would have a potential application for data reordering. The structure of $\pi^{(t)}$ during short-run stabilization makes the discovery of data ordering straightforward. The key is to look for values of $\pi^{(t)}$ that are approximately equal and reordering data accordingly.

3 Rank one SVD algorithm for data visualization

Given, for instance, a document×word data matrix A, let us consider $\pi = \begin{bmatrix} u \\ v \end{bmatrix}$, where $u \in \mathbf{R}_+^m$ and $v \in \mathbf{R}_+^n$. The upper part of π i.e. u is for documents weight and the lower part u is for words weights. Exploiting now the diagonal structure of $\mathbf{S}$, then we can write

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & D_r^{-1}A \\ D_c^{-1}A^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \Leftrightarrow \begin{cases} u = D_r^{-1}Av & \text{(a)} \\ v = D_c^{-1}A^Tu & \text{(b)} \end{cases} \tag{4}$$

For numerical computation of the leading singular vectors, we use rank one SVD algorithm, which is a variation of the power method adapted to rectangular data matrix. This iterative process starts with arbitrary vector u^0 and repeatedly performs the updates of v and u by alternating between formulas (a) and (b) given in 4 until convergence.

The documents and words weights are collected by u, v respectively ; the corresponding component values of u and v give document and word weights, respectively. We can sort both sets of words and documents in decreasing (or increasing) order of their weights and reorganize data matrix accordingly to reveal a structure into homogeneous blocks of A. We have developed a mutually reinforcing optimization procedure to exploit duality between both sets of documents and words. Thus, if a word is shared by many documents associated with

Algorithm 1 : R1SVD

Input : data $A \in \mathbf{R}_+^{m\times n}$, *threshold*, $D_r = Diag(A\mathbb{1})$ and $D_c = Diag(A^T\mathbb{1})$
Output : u, λ, v
Initialize : $\tilde{u} = D_r^{-1}A\mathbb{1}$, $,u = \frac{\tilde{u}}{||\tilde{u}||}$
repeat

$$\tilde{v}^{(t+1)} = D_c^{-1}A^T u^{(t)}$$
$$v^{(t+1)} = \frac{\tilde{v}^{(t+1)}}{||\tilde{v}^{(t+1)}||}$$
$$\tilde{u}^{(t+1)} = D_r^{-1}Av^{(t)}$$
$$u^{(t+1)} = \frac{\tilde{u}^{(t+1)}}{||\tilde{u}^{(t+1)}||}$$
$$\gamma^{(t+1)} \leftarrow ||u^{(t+1)} - u^{(t)}|| + ||v^{(t+1)} - v^{(t)}||$$

until stabilization of u, v, $|\gamma^{(t+1)} - \gamma^{(t)}| \leq threshold$

a block, then the word has a high weight associated with the block. On the other hand, if a document is shared by many words associated with a block, then the document has high weight associated with the same block.

4 Experiments

We now provide experimental results to illustrate the behavior of the R1SVD algorithm. We argue that R1SVD allows us to capture the trends of objects over a subset of attributes and then reorganizes data matrix into homogeneous blocks. We apply our algorithm on different real data sets and word-document simulated data sets (using a Bernoulli latent block model [7]). Different patterns are considered to show the ability of the R1SVD algorithm to discover the hidden blocks in data without fixing any parameters on the rows and columns ordering. Further R1SVD seems to have the potential to address the question of the number of clusters underlying the data ; it detects the suitable number of blocks by analyzing the evolution of the first right and left singular vectors u and v of the matrix A. We now try to visualize the homogeneous block structure that might be dis-

Table 1. 16 Townships Data.

	Townships															
Characteristics	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
High School	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Agricult Coop	0	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0
Rail station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
One Room School	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
Veterinary	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
No Doctor	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Police Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Land Reallocation	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0

covered by our algorithm. Figure 1 shows the original characteristics- townships data matrix (Table 1) and the reordered matrix (Table 2) obtained by arranging rows and columns based on the sorted u and v. the figure reveals the hidden sparsity structure of both characteristics and townships clusters. The three diagonal blocks in figure 1 correspond to the three clusters. It clearly appears that

Table 2. Reorganization of townships and characteristics after co-clustering.

Characteristics	H	K	B	C	D	G	L	O	M	N	J	I	A	P	F	E
High School	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Railway Station	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Police Station	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agricult Coop	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Veterinary	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Land Reallocation	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
One Room School	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
No Doctor	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

we can characterize each cluster of townships by a cluster of characteristics : {H, K} by {High School, Railway Station,Police Station}, {B, C, D, G, L, O} by {Agricult Coop, Veterinary,Land Reallocation} and {M, N, J, I, A, A, P, F, E} by {One Room School, No Doctor, No Water Supply}.

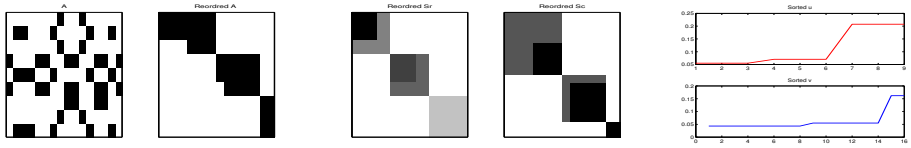

Figure 1. Townships : reordered $Sr = AA^T$ according u and $Sc = A^TA$ according v.

The three diagonal blocks in figures 2 and 3 correspond to the three clusters. The rough block diagonal structure indicates the cluster structure relation between documents and words. Hence by exploiting the duality of the data and features, incorporating the features information in data reordering at each stage, our algorithm yields better reordering solution that one dimensional clustering approaches, especially for high dimensional sparse datasets. The R1SVD al-

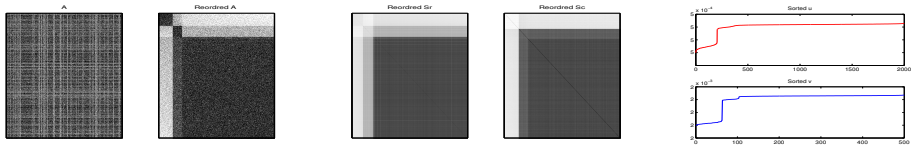

Figure 2. Data1 : reordered $Sr = AA^T$ according u and $Sc = A^TA$ according v.

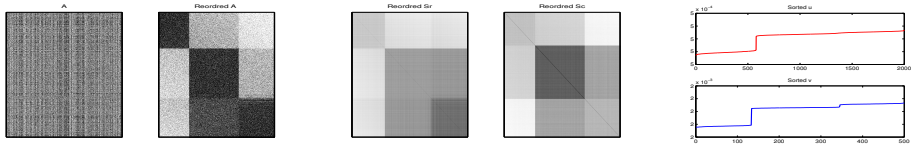

Figure 3. Data2 : reordered $Sr = AA^T$ according u and $Sc = A^TA$ according v.

gorithm seems to have the potential to address the question of the number of clusters underlying the data, it detects the suitable number of blocks by analyzing the evolution of the first left u of A_r and right singular vectors v of

Table 3. Confusion Matrix evaluation on rows and columns data.

data1 (rows)	data2 (rows)	data1 (columns)	data2 (columns)
0 205 0	0 0 795	0 0 40	155 0 0
1614 0 5	626 0 0	397 0 0	0 133 0
0 0 176	0 579 0	0 63 0	0 0 212

A_c. A performance study has been conducted to evaluate our method. In this subsection, we try to answer the question ; is this reordering meaningful ?. In order to be able to answer this question we use confusion matrix to measure the clustering performance of the co-clustering result provided by our method. The co-clustering task is to recover groups of rows and columns. After the learning stage, the clusters indicators are given by the vectors u and v. It can be seen that our method reconstructs efficiently all co-clusters for balanced and unbalanced data sets used in our experiments. From table 3, we observe that data reordering provided by R1SVD can be useful in a co-clustering context. It is very interesting to underly the fact that obtained visualization does not destroy data and, unlike most co-clustering algorithms, it does not require the number of blocks.

5 Conclusion

In this paper we have presented an iterative matrix-vector multiplication procedure called rank-one SVD for data visualization. The procedure consists in applying iteratively an appropriate stochastic adjacency data matrix associated to a bipartite graph, and then compute the first leading left singular vector associated to the eigenvalue λ_1 of this matrix. Stopping the algorithm after a few iterations involves a visualization of data matrix into homogeneous blocks. This approach appears therefore very interesting in co-clustering context.

Références

1. B. B. Bederson and B. Shneiderman, The Craft of Information Visualization : Readings and Reflections, San Francisco, Morgan Kaufmann, 2003.
2. H. Spath, Cluster Analysis Algorithms for Data Reduction and Classification of Objects, Chichester, UK, Ellis Horwood, 1980.
3. F. Murtagh, Book review : W. Gaul and M. Schader, Eds., Data, expert knowledge and decisions, Heidelberg : Springer-Verlag, 1988, viii + 380, J Classification 6 (1989), 129-132.
4. P. Arabie and L. J. Hubert, An overview of combinatorial data analysis, In Clustering and Classification, P. Arabie, L. J. Hubert, and G. De Soete, eds. River Edge, World Scientific, 1996, 5-63.
5. J. Bertin, Graphics and Graphic Information Processing, Berlin, Walter de Gruyter (Translated by W. J. Berg and P. Scott), 1981.
6. Horn, R. A., Johnson, C. R. (1986). Matrix analysis.Cambridge, U.K. : Cambridge University Press.
7. G. Govaert and M. Nadif, "Block clustering with Bernoulli mixture models : Comparison of different approaches," *Computational Statistics and Data Analysis*, 52, pp. 233-3245, 2008.
8. I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," *ACM SIGKDD*, pp. 269-274, 2001.