

A Clustering Model Based on Matrix Approximation with Applications to Cluster System Log Files

Tao Li and Wei Peng

School of Computer Science
Florida International University
11200 SW 8th street
Miami, FL, 33199
{taoli, wpeng002}@cs.fiu.edu

Abstract. In system management applications, to perform automated analysis of the historical data across multiple components when problems occur, we need to cluster the log messages with disparate formats to automatically infer the common set of semantic situations and obtain a brief description for each situation. In this paper, we propose a clustering model where the problem of clustering is formulated as matrix approximations and the clustering objective is minimizing the approximation error between the original data matrix and the reconstructed matrix based on the cluster structures. The model explicitly characterizes the data and feature memberships and thus enables the descriptions of each cluster. We present a two-side spectral relaxation optimization procedure for the clustering model. We also establish the connections between our clustering model with existing approaches. Experimental results show the effectiveness of the proposed approach.

1 Introduction

1.1 Background on System Log Files

With advancement in science and technology, computing systems are becoming increasingly more complex with an increasing variety of heterogeneous software and hardware components. They are thus becoming increasingly more difficult to monitor, manage and maintain. A popular approach to system management is based on analyzing system log files. The data in the log files describe the status of each component and record system operational changes.

The heterogeneous nature of the system makes the data more complex and complicated. As we know, a typical computing system contains different devices (e.g., routers, processors, and adapters) with different software components (e.g., operating systems, middleware, and user applications), possibly from different providers (e.g., Cisco, IBM, and Microsoft). These various components have multiple ways to report events, conditions, errors and alerts. The heterogeneity and inconsistency of log formats make it difficult to automate problem determination [5]. For example, there are many different ways for the components to report the start up process. Some might log “the component has started”, while others might say that “the component has changed the state from

starting to running”. Imagine that we would like to automatically perform the following rule: if any component has started, notify the system operators. Given the inconsistent content and sometimes subtle differences in the way components report the “started” process, writing a program to automate this simple task is difficult, if not impossible [10]. One would need to know all the messages that reflect the “started” status, for all the components involved in the solution. Every time a new component is installed, the program has to be updated by adding the new component’s specific terminology for reporting “started” situations. This makes it difficult to perform automated analysis of the historical event data across multiple components when problems occur.

To perform automated analysis of the historical event data across multiple components when problems occur, we need to categorize the text messages with disparate formats into common situations [10]. Clustering techniques are then needed to automatically *infer the common set of situations* from historical data and *obtain a brief description for each situation*. This would create consistency across similar fields and improve the ability to correlate across multiple component logs.

1.2 Clustering

As a fundamental and effective tool for efficient organization, summarization, navigation and retrieval of large amount of documents, clustering has been very active and enjoying a growing amount of attention with the ever-increasing growth of the on-line information. The clustering problem can be intuitively described as the problem of finding, given a set W of some n data points in a multi-dimensional space, a partition of W into classes such that the points within each class are *similar* to each other. The clustering problem has been studied extensively in machine learning [11], databases [7, 13], and statistics [2] from various perspectives and with various approaches and focuses.

Despite significant research on various clustering methods, few attempts have been made to obtain the descriptions for each cluster. In this paper, we present a clustering model¹ where the problem of clustering is formulated as matrix approximations. The model explicitly characterizes the data and feature memberships and thus enables the descriptions of each cluster. The goal of clustering is then transformed to minimizing the approximation error between the original data matrix and the reconstructed matrix based on the cluster structures. We provide an optimization procedure based on two-side spectral relaxation. In addition, we show the connections between our model with other clustering algorithms.

The rest of the paper is organized as follows: Section 2 introduces the notations and describes the general clustering model, Section 3 presents the optimization procedures based on two-side spectral relaxations, Section 4 presents the experimental results on system log data, finally, our discussions and conclusions are presented in Section 5.

2 The Clustering Model

We first present the clustering model for clustering problem. The notations used in the paper are introduced in Table 1.

¹ In this paper, we use model and framework interchangeably.

| | |
|--------------------------------|--|
| $W = (w_{ij})_{n \times m}$ | The Data set |
| $D = (d_1, d_2, \dots, d_n)$ | Set of data points |
| $F = (f_1, f_2, \dots, f_m)$ | Set of features |
| K | Number of clusters for data points |
| C | Number of clusters for features |
| $P = \{P_1, P_2, \dots, P_K\}$ | Partition of D into K clusters |
| $i \in P_k, 1 \leq k \leq K$ | i -th data point in cluster P_k |
| p_1, p_2, \dots, p_K | Sizes for the K data clusters |
| $Q = \{Q_1, Q_2, \dots, Q_C\}$ | Partition of F into C clusters |
| q_1, q_2, \dots, q_C | Sizes for the C feature clusters |
| $j \in Q_c, 1 \leq c \leq C$ | j -th feature in cluster Q_c |
| $A = (a_{ik})_{n \times K}$ | Matrix designating the data membership |
| $B = (b_{jc})_{m \times C}$ | Matrix designating the feature membership |
| $X = (x_{kc})_{K \times C}$ | Matrix specifies/indicates the association between data and features or the cluster representation |
| $\text{Trace}(M)$ | Trace of the Matrix M |

Table 1. Notations used throughout the paper.

The model is formally specified as follows:

$$W = AXB^T + E \quad (1)$$

where matrix E denotes the error component. The first term AXB^T characterizes the information of W that can be described by the cluster structures. A and B designate the cluster memberships for data points and features, respectively. X specifies cluster representation. Let \hat{W} denote the approximation AXB^T and the goal of clustering is to minimize the approximation error (or *sum-of-squared-error*)

$$\begin{aligned} O(A, X, B) &= \|W - \hat{W}\|_F^2 \\ &= \text{Trace}[(W - \hat{W})(W - \hat{W})^T] \\ &= \sum_{i=1}^n \sum_{j=1}^m (w_{ij} - \hat{w}_{ij})^2 \end{aligned} \quad (2)$$

$$= \sum_{i=1}^n \sum_{j=1}^m (w_{ij} - \sum_{k=1}^K \sum_{c=1}^C a_{ik} b_{jc} x_{kc})^2 \quad (3)$$

Note that the Frobenius norm, $\|M\|_F$, of a matrix $M = (M_{ij})$ is given by $\|M\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$.

3 The Optimization Procedure

Without loss of generality, we assume that the rows belong to a particular cluster are contiguous, so that all data points belonging to the first cluster appear first and the

second cluster next, etc ². Then A can be represented as $A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$. Note that

$A^T A = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & p_K \end{bmatrix}$ is a diagonal matrix with the cluster size on the diagonal. The

inverse of $A^T A$ serves as a weight matrix to compute the centroids. Hence, in general, if A and B denote the cluster membership, then we have $A^T A = \text{diag}(p_1, \dots, p_K)$ and $B^T B = \text{diag}(q_1, \dots, q_C)$ are two diagonal matrices.

Double K-Means Suppose $A = (a_{ik}), a_{ik} \in \{0, 1\}, \sum_{k=1}^K a_{ik} = 1, B = (b_{jc}), b_{jc} \in \{0, 1\}, \sum_{c=1}^C b_{jc} = 1$. Thus, based on Equation 3, we obtain

$$\begin{aligned} O(A, X, B) &= \|W - \hat{W}\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m (w_{ij} - \sum_{k=1}^K \sum_{c=1}^C a_{ik} b_{jc} x_{kc})^2 \\ &= \sum_{k=1}^K \sum_{c=1}^C \sum_{i \in P_k} \sum_{j \in Q_c} (w_{ij} - x_{kc})^2 \end{aligned} \quad (4)$$

For fixed P_k and Q_c , it is easy to check that the optimum X is obtained by $x_{kc} = \frac{1}{p_k q_c} \sum_{i \in P_k} \sum_{j \in Q_c} w_{ij}$. In other words, X can be thought as the matrix of centroids for the two-side clustering problem and it represents the associations between the data clusters and the feature clusters [3]. $O(A, X, B)$ can then be minimized via a two-side iterative procedure (i.e., the natural extensions of the K-means type algorithm for two-side cases [1, 3, 8]).

Spectral Relaxation If we relax the conditions on A and B , requiring $A^T A = I_K$ and $B^T B = I_C$, we would obtain an optimization procedure based on a two-side spectral relaxation. Similar ideas have been explored in for gene expression data in [4]. Here we illustrated in our clustering model. Note that

$$\begin{aligned} O(A, X, B) &= \|W - AXB^T\|_F^2 \\ &= \text{Trace}((W - AXB^T)(W - AXB^T)^T) \\ &= \text{Trace}(WW^T) + \text{Trace}(XX^T) - 2\text{Trace}(AXB^T W^T) \end{aligned}$$

² This can also be applied to column clusters.

Since $\text{Trace}(WW^T)$ is constant, hence minimizing $O(A, X, B)$ is equivalent to minimizing

$$O'(A, X, B) = \text{Trace}(XX^T) - 2\text{Trace}(AXB^TW^T). \quad (5)$$

The minimum of Equation 5 is achieved where $X = A^TWB$ as $\frac{\partial O'}{\partial X} = X - A^TWB$.

Plugging $X = A^TWB$ into Equation 5, we have

$$\begin{aligned} O'(A, X, B) &= \text{Trace}(XX^T) - 2\text{Trace}(AXB^TW^T) \\ &= \text{Trace}(A^TWBB^TW^TA) - 2\text{Trace}(AA^TWBB^TW^T) \\ &= \text{Trace}(WW^T) - 2\text{Trace}(A^TWBB^TW^TA) \end{aligned}$$

Since the first term $\text{Trace}(WW^T)$ is constant, minimizing $O'(A, X, B)$ is thus equivalent to maximizing $\text{Trace}(A^TWBB^TW^TA)$.

Let $G = WB$, then $\text{Trace}(A^TWBB^TW^TA) = \text{Trace}(A^TGG^TA)$.

Proposition 1 *Given B , $\text{Trace}(A^TGG^TA)$ can be maximized by constructing A with the eigenvectors of GG^T corresponding to the K largest eigenvalues.*

Note that $\text{Trace}(A^TWBB^TW^TA) = \text{Trace}(B^TW^TAA^TWB)$. Denote $H = W^TA$. Similarly, we have

Proposition 2 *Given A , $\text{Trace}(B^THH^TB)$ can be maximized by constructing B with the eigenvectors of HH^T corresponding to the C largest eigenvalues.*

Proposition 1 and Proposition 2 can be proved via matrix computations [6] and they lead to an alternating optimization procedure to maximize $\text{Trace}(A^TWBB^TW^TA)$, i.e., update B to maximize $\text{Trace}(A^TWB^TW^TB^TA)$ and update A to maximize $\text{Trace}(B^TW^TAA^TWB)$. The alternative optimization procedure can be thought as a two-side generalization of the spectral relaxation [12]. After obtaining the relaxed A and B , the final cluster assignments of the data points and features are obtained by applying ordinary K-means clustering in the reduced spaces. A short description of the clustering procedure is presented as *Algorithm 1*.

4 Experiments

We performed experimental studies to 1) show that the clustering model can identify the inherent structure in real application studies on system log files, and 2) verify that our proposed clustering method can improve the clustering performance. Due to space limit, we only present a case study on clustering system log files, showing that the cluster model can identify the inherent structures of the datasets.

4.1 Log Data Generation

The log files used in our experiments are collected from several different machines with different operating systems using logdump2td (NT data collection tool) developed at IBM T.J. Watson Research Center. The raw log files contains a free-format ASCII description of the event. In our experiment, we apply clustering algorithms to group the messages into different situations. To pre-process text messages, we remove stop words and skip HTML labels.

Algorithm 1 Two-Side Spectral Relaxation

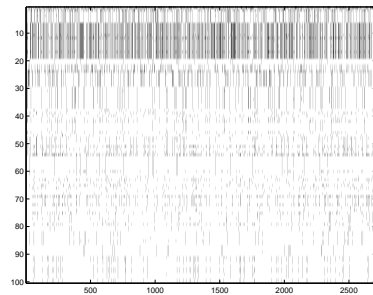
Input: ($W_{n \times m}$, K and C)Output: P, Q : set of clusters;**begin**

- 1 Initialize A ;
 2. **Iteration:** Do while the stop criterion is not met
begin
 - 2.1 Update B to maximize $\text{Trace}(A^T W B W^T B^T A)$
 - 2.2 Compute $X = A^T W B$
 - 2.3 Update A to maximize $\text{Trace}(B^T W^T A A^T W B)$**end**
 3. Get the final clusterings P and Q
- end**

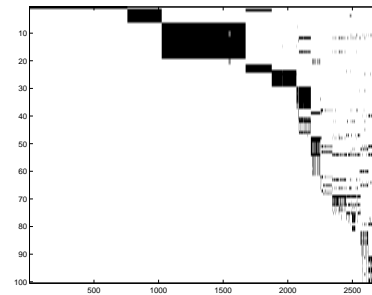
4.2 Experimental Results on Log Data

The general cluster framework introduced in Section 2 explicitly models both data and feature assignments. With the feature assignments, we can get the distinguishing words for each cluster and consequently obtain a description for the cluster. We use *Algorithm 1* described in Section 3 in our experiments.

Figure 1 shows the original word-document matrix of the log file and the reordered matrix obtained by arranging rows and columns based on the cluster assignments. The figure reveals the hidden sparse structure of both the document message and word clusters.



(a) Original Dataset



(b) Dataset after Reordering

Fig. 1. Visualization of the original message-data matrix and the reordered document-data matrix.

Table 2 lists the discriminating words for several clusters. We can derive meaningful common situations from the cluster results. For example, cluster 1 mainly concerns

| Cluster Number | Words |
|----------------|---|
| 1 | product, configuration, completed |
| 2 | inventory, server, respond, network, connection, party, root |
| 3 | create, temporary, file |
| 4 | exist, directory, domain, contacted, contact, failed, certificate, enrollment |
| 5 | profile, service, version, faulting, application, module, fault, address |
| 6 | completed, update, installation |
| 7 | service, started, application, starting |
| 8 | stopped, restarted, completed, failed, shell, explorer |

Table 2. keywords and their clusters

product configuration, Cluster 2 is about aspects related to a connection to another component, Cluster 3 describes the problem of creating temporary files etc.

The case study on clustering log message files for computing system management provides a successful story of applying the cluster model in real applications. The log messages are relatively short with a large vocabulary size [9]. Hence they are usually represented as sparse high-dimensional vectors. In addition, the log generation mechanisms implicitly create some associations between the terminologies and the situations. Our clustering model explicitly models the data and feature assignments and is also able to exploit the association between data and features. The synergy of these factors leads to the good application on system management.

5 Discussions and Conclusions

Based on different constraints on the matrices A , B and X , our cluster model encompasses different clustering algorithms. The relationships between our clustering model and other well-known clustering approaches can be briefly summarized in Figure 2.

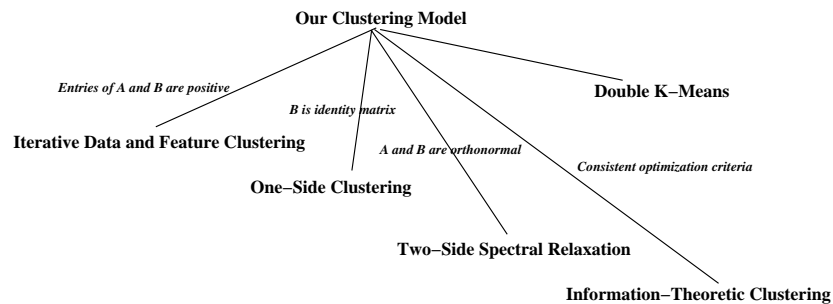


Fig. 2. Relations of Our Clustering Models and Other Approaches.

In this paper, we present a clustering model and investigate its applications to cluster system log data. The model explicitly characterizes the data and feature memberships and thus enables the descriptions of each cluster. A two-side spectral relaxation

method is presented as the optimization procedure for clustering. In addition, we also establish the connections between our clustering model with existing approaches. Experimental results show the effectiveness of the proposed approach.

Acknowledgment

This project is supported by an IBM Shared University Research(SUR) award and an IBM Faculty Award. Wei Peng is supported by a Florida International University Presidential Graduate Fellowship.

References

1. D. Baier, W. Gaul, and M. Schader. Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring. In R. Klar and O. Opitz, editors, *Classification and Knowledge Organization*, pages 577–566. Springer, 1997.
2. Marsha Berger and Isidore Rigoutsos. An algorithm for point clustering and grid generation. *IEEE Trans. on Systems, Man and Cybernetics*, 21(5):1278–1286, 1991.
3. William Castillo and Javier Trejos. Two-mode partitioning: Review of methods and application and tabu search. In K. Jajuga, A. Sokolowski, and H.-H. Bock, editors, *Classification, Clustering and Data Analysis*, pages 43–51. Springer, 2002.
4. Hyuk Cho, Inderjit S. Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the SIAM Data Mining Conference*, 2004.
5. Gary Dudley, Neeraj Joshi, David M. Ogle, Balan Subramanian, and Brad B. Topol. Autonomic self-healing systems in a cross-product it environment. *International Conference on Autonomic Computing*, pages 312–313, 2004.
6. Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.
7. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 73–84. ACM Press, 1998.
8. Vichi Maurizio. Double k-means clustering for simultaneous classification of objects and variables. In S. Borra, R. Rocci, M. Vichi, and M. Schader, editors, *Advances in Classification and Data Analysis*, pages 43–52. Springer, 2001.
9. Jon Stearley. Towards informatic analysis of syslogs. In *Proceedings of IEEE International Conference on Cluster Computing*, Sept. 2004.
10. Brad Topol, David Ogle, Donna Pierson, Jim Thoensen, John Sweitzer, Marie Chow, Mary Ann Hoffmann, Pamela Durham, Ric Telford, Sulabha Sheth, and Thomas Studwell. Automating problem determination: A first step toward self-healing computing systems. IBM White Paper, October 2003. <http://www-106.ibm.com/developerworks/autonomic/library/ac-summary/ac-prob.html>.
11. Andrew Webb. *Statistical Pattern Recognition*. Wiley, 2002.
12. Hongyuan Zha, Xiaofeng He, Chris Ding, and Horst Simon. Spectral relaxation for k-means clustering. In *Proceedings of Neural Information Processing Systems*, 2001.
13. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 103–114. ACM Press, 1996.