Intelligent Data Analysis 9 (2005) 603–620
IOS Press

Learning quantifiable associations via principal sparse non-negative matrix factorization

Chenyong Hu[a,*], Benyu Zhang[b], Yongji Wang[a,e], Shuicheng Yan[c], Zheng Chen[b], Qing Wang[a] and Qiang Yang[d]
[a]*Lab for Internet Software Technologies, Institute of Software Chinese Academy of Sciences, Beijing, 100080, P.R. China*
E-mail: {huchenyong, ywang, wq}@itechs.iscas.ac.cn
[b]*Microsoft Research Asia, 49 Zhichun Road, Beijing, 100080, P.R. China*
E-mail: {byzhang,zhengc}@microsoft.com
[c]*Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong*
E-mail: scyan@ie.cuhk.edu.hk
[d]*Department of Computer Science, Hong Kong University of Science and Technology*
E-mail: qyang@cs.ust.hk
[e]*Key Lab of Computer Science, Institute of Software Chinese Academy of Sciences, Beijing, China*

Received 27 October 2004
Revised 26 December 2004
Accepted 13 Februry 2005

Abstract. Association rules are traditionally designed to capture statistical relationship among itemsets in a given database. To additionally capture the quantitative association knowledge, Korn et.al. recently propose a paradigm named Ratio Rules [6] for quantifiable data mining. However, their approach is mainly based on Principle Component Analysis (PCA), and as a result, it cannot guarantee that the ratio coefficients are non-negative. This may lead to serious problems in the rules' application. In this paper, we propose a new method, called Principal Sparse Non-negative Matrix Factorization (PSNMF), for learning the associations between itemsets in the form of Ratio Rules. In addition, we provide a support measurement to weigh the importance of each rule for the entire dataset. Experiments on several datasets illustrate that the proposed method performs well for discovering latent associations between itemsets in large datasets.

Keywords: Data mining, association rules, non-negative matrix factorization, principal sparse non-negative matrix factorization

1. Introduction

Association rules are one of the major representations in expressing the knowledge discovered from large databases. The problem of association rules mining (ARM) in large transactional databases is introduced in [1,5], and the basic idea is to discover some important and interesting associations among

*Corresponding author.

the itemsets such that the presence of some items in a transaction will imply the presence of other items in the same transaction. The form of such an association is as follows:

$$\{bread, milk\} => butter\ (80\%)$$

The above form means that customers who buy "bread" and "milk" are likely to buy butter with 80% confidence. Inspired by the work in [1], several fast algorithms based on the level-wise Apriori framework [2,15] and partitioning [10] are proposed to remedy the performance bottleneck of Apriori. In addition, several novel mining techniques, such as parallel algorithms [11,21], uncertain algorithms [4, 12] and other techniques [9,13,19], also receive much attention lately.

Most of the prevalent approaches assume that the transactions only carry Boolean information and ignore the valuable knowledge inherent in the quantities of the items. To find association rules, the Boolean approaches assume that all we need to know is whether an item is contained in a transaction. Thus, Boolean association rules have the advantages that they are easy to interpret. However, the major drawback is that a given data matrix V (with e.g. amounts spent per customer per product) has to be converted to a binary matrix by treating non-zero amounts as plain "1"s. This approach simplifies the data mining algorithms, but tends to lose lots of valuable information.

In fact, since the quantities of the items in many datasets contain valuable information for us, it is necessary to provide a definition of association rules when the datasets contain quantitative attributes. Several efficient algorithms for mining quantitative association rules have been proposed in the past [3, 17]. A notable algorithm is the work [6], where they provided a stronger set of rules as *Ratio Rules*. A rule under that framework is expressed in the following form:

$$bread : milk : butter = a : b : c$$
$$(a, b, c\ are\ arbitrary\ numerical\ values)$$

This rule states that for each a amount spent on bread, a customer normally spends a b amount on milk and a c amount on butter. Given such a definition, Ratio Rules allow quantitative information to be expressed in many practical applications, including forecasting such as *"if a customer spends $\$a$ on bread, how much will s/he spend on butter?"*, and "what-if" scenarios such as *"we expect the demand for bread, how much butter should we stock up on?"*.

Principal Component Analysis (PCA) is often used in data mining applications to discover the *eigen-vectors* of a dataset. Ratio Rules [6] can represent the quantitative associations between items as the principal *eigen-vectors* of a data matrix, where the values a, b and c in the example above correspond to the projections of the *eigen-vectors* in the space defined by bread, milk and butter. Because PCA factorization requires only the orthogonality in matrix factorization, the ratio coefficients of rules (elements of *eigen-vectors*) can be either positive or negative. An example of Ratio Rules containing a negative value is

$$Shoe :\ Coat :\ Hat = 1 : -2 : -5$$

Obviously, this rule loses the intuitive appeal of associations between items when containing negative values, because a customer's spending should always be positive (there is no consideration of profit here). In this paper, we present a method to address this problem.

Our method amounts to a novel application of Non-negative Matrix Factorization (NMF) [7]. Like PCA, the NMF is aimed at learning latent components; unlike PCA, the NMF imposes the non-negativity constraints in the matrix factorization to ensure that all principle components are positive. However, we

cannot directly apply NMF for our purpose. We observe that although all the coefficients of the latent components learned by NMF are non-negative, it is still difficult to explain that these latent components represent the latent associations between items in a quantifiable dataset. Therefore, we need to provide a bridge to bring NMF closer to association rules.

In this work, we propose a novel method called *Principal Sparse Non-Negative Matrix Factorization* (PSNMF), which adds the sparsity constraints in the standard NMF [7]. Furthermore, we extend the definition of Ratio Rules with all the ratio coefficients constrained to be non-negative. An example of such Ratio Rules according to this definition is:

$$bread : milk : butter = 1 : 2 : 5$$

This rule implies that the customer who spends \$ 1 on bread tends to buy \$ 2 of milk and \$ 5 of butter. We will illustrate that the Ratio Rules by PSNMF can also support a variety of important tasks such as forecasting and answering "what-if" scenarios.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 describes the problem and the intuition behind the Ratio Rules. Section 4 introduces our new algorithm (PSNMF) and contrasts it with PCA. Section 5 presents the experimental results. Section 6 concludes the paper. The convergence of the PSNMF learning procedure is given in Appendix.

2. Related work

2.1. Association rules

Association rules algorithms find rules of the form $X \Rightarrow Y$ where X and Y are the disjoint sets of the items. The data used in the notable Apriori algorithms [1] are market basket data that are naturally binary (two-valued) referred as Boolean. Either an item has been purchased by a customer and is in his/her market basket, indicated by a value of 1 (true), or it has not, indicated by a value of 0 (false). Although finding association rules in two-valued categorical data has been well investigated, problems occur when trying to find these types of the rules in data with pure numeric (quantitative) or mixed numeric and categorical (qualitative) values. Srikant et al. extend the traditional definition to include quantitative data, and propose quantitative association rules in [17]. A notable algorithm is the work [6], where they provide a stronger set of rules as *Ratio Rules.* The following lists some simple examples for these types of association rules:

Boolean association rules [16,20]:

$$\{bread, milk\} => butter\ (80\%)$$

Quantitative association rules [17]:

$$\langle bread : [2 - 5]\rangle => \langle butter : [1 - 2]\rangle$$

Ratio Rules [6]:

$$bread : milk : butter = a : b : c$$
$$(a, b, c\ is\ arbitrary\ value.)$$

In the Ratio Rules listed above, a, b and c are used to denote the relative coefficients of the corresponding items, that is, the association among such items can be represented as $a : b : c$. The method in [6] is

based on the Principal Component Analysis (PCA), where each Ratio Rule corresponds to an *eigen-vector* found by the PCA. Because PCA factorization requires only the orthogonality in matrix factorization, the ratio coefficients of the rules can be of arbitrary numerical values. Obviously it is not interpretable in the context of association rules.

2.2. Approaches for matrix factorization

Motivated by the method [6], we extend the definition of Ratio Rules to learn hidden associations in transaction sets based on Principal Sparse Non-negative Matrix Factorization. In fact, the PSNMF algorithm is presented based on Non-negative Matrix Factorization (NMF). We will introduce the Non-negative Matrix Factorization in detail in the following sections. Here, let's briefly review some other prevalent approaches in matrix factorization first.

2.2.1. Latent Semantic Analysis (*LSA*)

A classic algorithm arising from linear algebra, LSA decomposes the matrix in three matrices by a truncated Singular Value Decomposition (SVD)

$$A \cong USV^T$$

where $A \in \mathbb{R}^{m\times n}, U \in \mathbb{R}^{m\times r}, S \in \mathbb{R}^{r\times r}$ and $V \in \mathbb{R}^{r\times n}$. The operation performs the optimal least-square projection of the original space onto a space of the reduced dimensionality R. The subspace representation has been empirically shown to capture to some degree the semantic relationships across terms in corpus. LSA has been extensively used in text analysis, and more recently to improve retrieval of multimedia news documents [22]. Unfortunately, LSA lacks a clear probabilistic interpretation [18]. Therefore, PLSA [18] is provided as a probabilistic version of LSA which focuses on good estimation of all the parameters using the EM algorithm.

2.2.2. Linear Discriminant Analysis (*LDA*)

Linear Discriminant Analysis (LDA), also called Fisher Discriminant Analysis (FDA), is proposed to pursue a low dimensional subspace that can best discriminate the samples from different classes. Suppose $W \in \mathbb{R}^{d\times p}$ is the linear projection matrix, LDA aims to maximize the so-called Fisher criterion:

$$J(W) \in \left|W^T S_b W\right| / \left|W^T S_w W\right|$$

where

$$S_b = \sum_{i=1}^{c} p_i (m_i - m)(m_i - m)^T, S_w = \sum_{i=1}^{c} p_i E(u_i - m)(u_i - m)^T$$

is called inter-class scatter matrix and intra-class scatter matrix, respectively, where c is the number of classes, m is the mean of all the samples, m_i is the mean of the samples belonging to class i, and p_i is the prior probability for a sample belonging to class i. The projection matrix W can be obtained by solving the following generalized *eigen-vectors* decomposition problem:

$$S_b w = \lambda S_w w$$

There are at most $c-1$ nonzero *eigen-values*, so the upper bound of p is $c-1$; and at least $d+c$ data samples are required to make it possible that S_w is not singular. These constrains limit the application of LDA. Furthermore, it is difficult for LDA to handle large size datasets when the dimension of the feature space is high.

2.2.3. Independent Component Analysis (*ICA*)

The problem of Independent Component Analysis (ICA) has been studied by many authors in recent years, and in the simplest form of ICA, it is assumed that we have a sequence of observations $\{x_k\}$ with each observation vector x generated according to:

$$x = As$$

where $s = (s_1, \ldots, s_n)^T$ is a vector of real independent random variable (*the sources*), A is a nonsingular $n \times n$ real mixing matrix. The task in ICA is to identify A using the assumption of independence of the $s_i s$ and hence, to construct an immixing matrix $B = RA^{-1}$ giving $y = Bx = BAs = Rs$, where R is a matrix which permutes and scales the sources. Typically, we assume that the sources have unit variance, with any scaling factor being absorbed into the mixing matrix A, so y will be a permutation of s with just a sign ambiguity.

Common methods for ICA involve higher order cumulates such as kurtosis or the use of autocorrelation differences between the sources. The observations x are often assumed to be zero-mean, or transformed to be so, and are commonly prewhitened by a matrix V satisfying $z = Vx$ so that $E\{zz^T\} = I$ before an optimization algorithm is applied to z.

3. Problem definition

The problem that we tackle is as follows. Given a large set of N customers and M products organized in an $N \times M$ matrix V where each row corresponds to a customer transaction (e.g., market basket databases), and the entity v_{ij} gives the dollar amount spent by customers on the products. The goal is to find all Ratio Rules of the form:

$$v_1 : v_2 : v_3 : \ldots : v_M \ (v_i \geqslant 0)$$

The above form means that there are some latent components, which represent the non-negative associations between the items of the dataset V. That is, customers who buy the items will spend $v_1, v_2, \ldots$ respectively on each itemset with sufficient frequency.

Figure 1(a) lists a large set of N customers and $M(M = 2)$ products organized in a $N \times M$ matrix V. Each row vector of the matrix can be thought of as an M-dimensional point. Figure 1 (b) lists such 2-points distribution in the graphical form. Here we assume that the dataset is made up of two clusters. Each cluster corresponds to a latent component that implies the association between items in the dataset. Given this set of N points, our goal is to capture the latent components, which represent the associations between items. We list two Ratio Rules discovered by PCA [6] in Fig. 2(a), where one contains negative values:

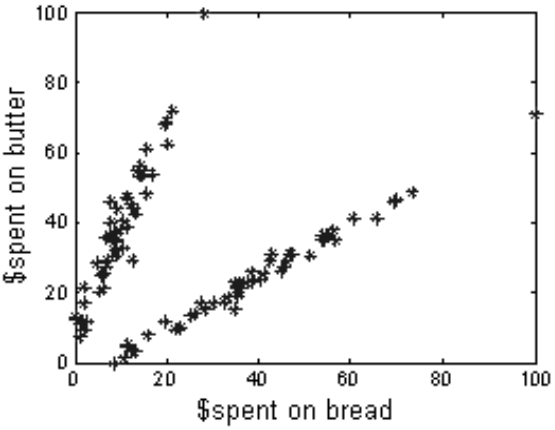
$$bread : butter = -0.77 : 0.64$$

According to the above definition of Ratio Rules, the negative association between items ("bread" and "butter") does not make sense. However, the PCA does not prevent this from happening. Furthermore, the Ratio Rules which tend to represent the associations between items often do not give the latent associations behind the distribution of these points. In Fig. 1(b), it is obvious that Ratio Rules found by PCA deviate the latent associations between items.

In fact, from Fig. 1(b) we find that the latent associations are not mutually orthogonal, while the method by PCA imposes the orthogonality constraint on these Ratio Rules. Therefore, Ratio Rules

	bread ($)	*butter ($)*
T_1	2.50	4.39
T_2	3.91	8.44
T_3	3.99	11.56
T_4	3.65	8.20
T_5	4.99	15.58
T_6	4.64	14.05
T_7	6.92	24.69
T_8	2.75	4.08
T_9	6.14	3.67
…	…	…
T_n	4.36	2.55

(a)



(b)

Fig. 1. Data matrix with 2-dimension in table form (a) and its counterpart in graphical form (b).

found by PCA cannot reflect the latent associations among the items correctly. Compared to Fig. 2(a), (b) illustrates the Ratio Rules captured by our proposed PSNMF in this work. Surprisingly, every rule can be treated as an association between items in the two clusters respectively. Furthermore, our method that is based PSNMF guarantees that all the values of Ratio Rules are positive. For example:

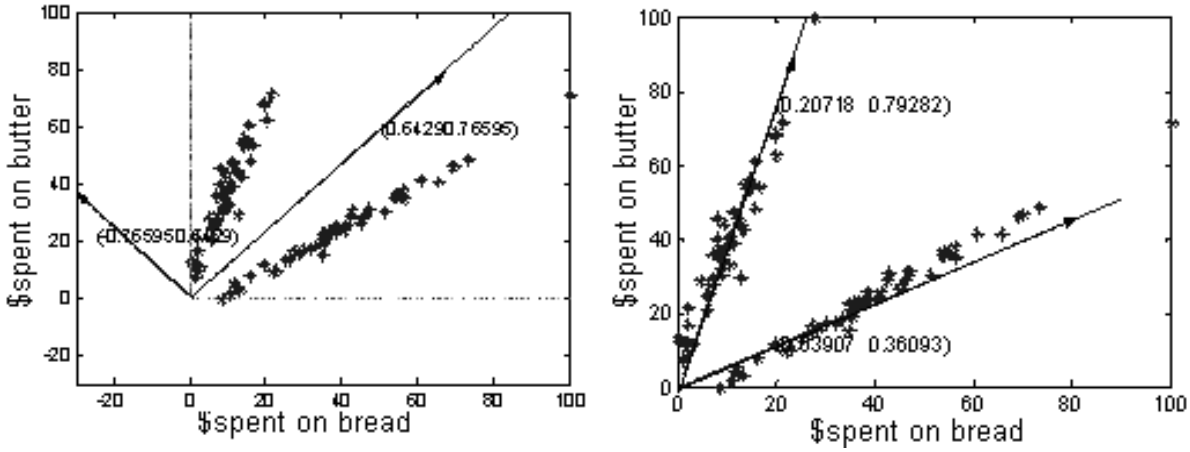
$$bread : butter = 0.21 : 0.79$$

$$bread : butter = 0.64 : 0.36$$

From Fig. 2(b), we can find that the customers of one cluster mainly depend on the first rule where their relative spending amounts between bread and butter are closely to the ratio (0.21:0.79). Furthermore, customers of another cluster are related by the ratio (0.64:0.36). In addition, in this work, a *support* measurement is designed to illustrate the importance of each rule for the entire dataset.

4. Principal sparse non-negative matrix factorization

Let a set of N M-dimension training records be given as a $M \times N$ matrix V, with each column consisting of the M non-negative attribute values of records. Denote a set of $P \ll M$ basis components by a $M \times P$ matrix W, where each record can be represented as a linear combination of the basis



(a) (b)

Fig. 2. Latent associations identified by PCA (a) and PSNMF (b).

components using the approximate factorization:

$$V \approx WH \tag{1}$$

where H is a $P \times N$ matrix, and the entities of H are the coefficients of the corresponding basis components. Korn et al. [6] apply the Principal Component Analysis (PCA) in matrix factorization. PCA factorization requires that the basis components which are the columns of W to be orthogonal and the rows of H to be mutually orthogonal. PCA imposes no other constraints than the orthogonality constraint, and hence, it allows the entries of W and H to be either positive or negative. In representing the associations between items, anomalies such as the following can happen, which is clearly not desirable: $bread : butter = 1 : -2$.

4.1. Non-negative matrix factorization (*NMF*)

NMF [7,8] is a method to obtain a representation of data using non-negativity constraints. As we will see, these constraints lead to a part based representation because they allow only additive, not subtractive, combinations of the original data [8]. Given an initial database expressed by a matrix V, where each column is an M-dimensional non-negative vector of the original database, it is possible to find two new matrices (W and H) in order to approximate the original matrix $V_{i,\mu} \approx (WH)_{i,\mu} = \sum_{j=1}^{p} W_{i,j} H_{j,\mu}$. The dimensions of the factorized matrices W and H are $M \times P$ and $P \times N$ respectively, and usually, P is chosen so that $(M + N)P < MN$. Each column of the matrix W contains a basis vector while each column of the matrix H contains the weights needed to approximate the corresponding column in V using the basis from W In the PCA context, each column of the matrix W represents an *eigen-vector* and the factorized matrix of H represent the *eigen-projections*. In contrast to PCA, NMF does not allow negative entries in the factorized matrices W and H permitting the combination of multiple basis components to represent an object.

In order to estimate the factorization matrices, an objective function has to be defined by using the

I-divergence of V from Y

$$D(V||Y) = \sum_{i,j} \left(v_{ij} \log \frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij} \right) \tag{2}$$

As the measurement of fitness for factorizing V into $WH \triangleq Y = [Y_{ij}]$, a NMF factorization is defined as

$$\begin{aligned} &\min_{W,H} \ D(V \,\|WH\,) \\ &s.t\ W, H \geqslant 0,\ \sum_i w_{ij} = 1\ \forall j \end{aligned} \tag{3}$$

where $W, H \geqslant 0$ means that all entries of W and H are non-negative. The above optimization can be done by using multiplicative update rules [7].

4.2. Sparse non-negative matrix factorization (*SNMF*)

Although NMF is successful in Matrix Factorization, the NMF model as defined by the constrained minimization of Eq. (2) does not impose the sparse constraints. Therefore, it can hardly yield a factorization, which reveals local sparse features in the dataset V. We wish to find a decomposition in which the latent components are sparse, implying that the projection coefficients over all training records have probability densities which are highly peaked at zero and have heavy tails. This basically means that any given training records can be represented using only a few significantly non-zero latent coefficients. Related sparse coding is proposed in the work of [14] for matrix factorization.

Inspired by the original NMF and sparse coding, the aim of this work is to propose a new algorithm we call Sparse Non-negative Matrix Factorization (SNMF), which imposes the sparse and non-negative constraints. Considering the sparse constraint, we require that any given record should be represented with only a few significantly non-zero latent coefficients. Consequently, we can impose the sparse constraint by $\sum_j \|l_j\|_1 = \min$, which $\|l_j\|_1$ is the 1-normal of the projection of the j-th record. Therefore, we put forward the following constrained divergence as the objective function for SNMF:

$$\begin{aligned} &D(V||Y) = \sum_{i,j} \left(v_{ij} \log \frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij} \right) + \lambda \sum_j \|l_j\|_1 \\ &where\ l_j = (h_{1j}, h_{2j}, h_{3j}, \ldots, h_{pj})^T\ denotes\ the\ column\ of\ H. \end{aligned} \tag{4}$$

where $WH \triangleq Y = [Y_{ij}]$,because all the coefficients are non-negative, we can simplify the 1-normal expression with: $\|l_j\|_1 = \sum_{k=1}^{p} \|h_{kj}\| = \sum_{k=1}^{p} h_{kj}$. In addition, parameter λ is obtained by experience and is generally assumed a non-negative positive constant. As we know, when λ is set with zero, we find formula Eq. (4) is essentially the objective function of Non-negative Matrix Factorization, and the factorization procedure is only constrained by the non-negative characteristic. With the increasing of λ, the effects of sparse constraints will get stronger and the accuracy will decrease. In this work, according to our experiments, we set the value of λ with 1. As the measurement of fitness for factorizing V into $WH \triangleq Y = [Y_{ij}]$, a SNMF factorization is defined as:

$$\min_{W,H} \ D(V \,\|WH\,)$$

$$s.t\ \forall i,j:\ W_{ij} \geqslant 0, H_{ij} \geqslant 0,\ \text{and}\ \forall i\ \ \|w_i\|_1 = 1 \tag{5}$$

$where\ w_i\ denotes\ the\ i:th\ column\ of\ W.$

Since we have chosen a linear activation penalty (i.e. minimum the 1-norm of the column of H) to measure sparseness, as for the objective function Eq. (5), a sparse solution to the above-constrained minimization can be found by using the following update rules:

$$h_{kl} = h_{kl} \sum_i v_{il} \frac{w_{ik}}{\sum_k (w_{ik} h_{kl})} \Big/ \left(\sum\nolimits_i w_{ik} + \lambda\right) \tag{6}$$

$$w_{kl} = w_{kl} \sum_j v_{kj} \frac{h_{lj}}{\sum_l w_{kl} h_{lj}} \Big/ \sum\nolimits_j h_{lj} \tag{7}$$

Note that the solution to minimizing the criterion function is not unique. If W and H are the solutions, then, $WD,\ HD^{-1}$ will also form a set of solutions for any positive diagonal matrix D. To make the solution unique, we further require that the 1-normal of the column vector in matrix W is one. When normalizing matrix W, matrix H needs to be adjusted accordingly, so that WH does not change.

$$w_{kl} = w_{kl} \Big/ \sum\nolimits_k w_{kl} \tag{8}$$

$$h_{kl} = h_{kl} \sum\nolimits_k w_{kl} \tag{9}$$

It is proved that the objective function is non-increasing under the above iterative updating rules and that the convergence of the iteration is guaranteed (see Appendix).

4.3. Principal SNMF

When the dataset V is decomposed with W and H, the column vectors of W make up a new basis components space, and each column value of H represents the corresponding projection on the new basis component space. The non-negative restriction on these coefficients results in the additive nature of NMF. In other words, every row coefficient of H is the affection fact of a corresponding column basis for whole dataset V. As a whole, the sum of every row vector of H represents the importance of corresponding base for the whole dataset V. Therefore, we define a *support* measurement after normalizing every column of H:

$$h_{kl} = h_{kl} \Big/ \sum\nolimits_k h_{kl} \tag{10}$$

Definition. For every rule (column vector) of W, we define a support measurement:

$$support(w_i) = \sum_j h_{ij} \Big/ \sum_{ij} h_{ij} \tag{11}$$

$where\ rule(w_i)\ denotes\ the\ column\ of\ W$

$support(w_i) \in [0,1],\ and\ \sum_i support(w_i) = 1$

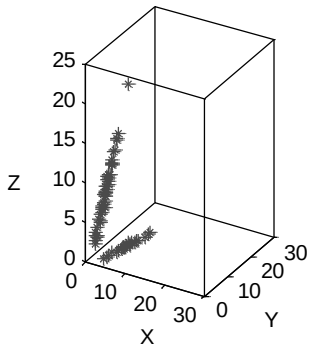


Fig. 3. Synthetic dataset with two clusters.

Consequently, we can measure the importance of each rule for the entire dataset by their *support* values. The more value of *support* implies the more importance of such rule for the entire dataset.

In order to select the principal k rules as Ratio Rules to denote the associations between items in the dataset, firstly, we rank the whole rules in descending by the *support* value of each rule. And then, retain the first $k$principal rules as Ratio Rules because they are more important than the other rules for the entire dataset. About the selection of k value, a simple method is taken such as:

$$\min_k \left(\frac{\sum_{i=1}^{k} support(w_i)}{\sum_{i=1}^{M} support(w_i)} > threshold \right) \tag{12}$$

From the above form Eq. (12), the Ratio Rules are obtained effectively according that the sum of k *support* values of Ratio Rules cover *threshold* (*i.e. 90%*) of the grand total *support* values.

5. Experiments

Experiments are performed on some datasets (synthetic and real datasets) to illustrate that our proposed method is effective in mining Ratio Rules between items on quantitative matrix. In addition, it is also proven that our Ratio Rules can work on the binary matrix.

5.1. Ratio rules on quantitative matrix

At the beginning of this section, we will illustrate the different characteristics of the PSNMF and PCA by two experimental results.

5.1.1. Synthetic dataset

We have applied both the PSNMF and the PCA to a dataset consisting of two clusters, which contains 25 Gaussian distribution points on x-y plane (the model parameters are set with $\mu = [3, 5]$, $\delta = [1, 1.2; 1.2, 2]$) and 50 points on y-z plane. (Fig. 3) (Generated with parameters $\mu = [3, 5]$, $\delta = [2, 1.6; 1.6, 2]$).

5.1.2. Ratio rules on PSNMF

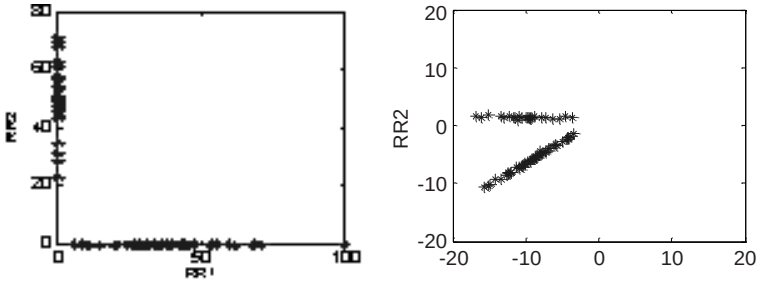
Table 1 lists all the rules (columns of W) and the corresponding *support* values. According to the *support* measurement, we rank these rules: $RR_1 > RR_2 > RR_3$. Since $support(w_1) +$

Table 1
Rules based on PSNMF

PSNMF	RR_1	RR_2	RR_3
(X)	0.000	0.696	0.020
(Y)	0.493	0.304	0.980
(Z)	0.507	0.000	0.000
$Sum(w_i)$	49.88	21.636	3.488
$Support(w_i)$	0.665	0.289	0.046

Table 2
Rules based on PCA

PCA	RR_1	RR_2
X)	-0.518	0.720
(Y)	-0.765	-0.148
(Z)	-0.383	-0.678



(a) In PSNMF Subspace (b) In PCA Subspace

Fig. 4. Data projection in PSNMF and PCA subspaces.

$support(w_2)= 0.9535 > 90\%$, we can obtain the following Ratio Rules:

$$rule_1 :: \ X : Y : Z \ => 0 : 0.493 : 0.507 \ (0.6650)$$
$$rule_2 :: \ X : Y : Z \ => 0.696 : 0.304 : 0 \ (0.2885)$$
$$where \ X, Y, Z \ represent \ the \ items \ (columns \ of \ matrix)$$

In addition, Fig. 4(a) indicates all the data projection in PSNMF subspace possess sparse property. That is, every record can be well represented with only a few significantly non-zero latent coefficients. For example, 2/3 records (the cluster with distribution on y-z plain) mostly depend on $rule_1$ and others on $rule_2$ respectively. Therefore, the corresponding *support* value (0.665) of $rule_1$ is consistent with intuition.

5.1.3. Ratio rules on PCA

According to the method by PCA, we retain two corresponding *eigen-vectors* as Ratio Rules in Table 2. From Table 2, we find some entities of the Ratio Rule contain negative values which lack intuition to explain the associations between such items. Furthermore, no measurement is given to rank the importance of Ratio Rules for the entire dataset, thus we can not partition these Ratio Rules which is more important for the entire dataset. In addition, from Fig. 4(b), the projection of dataset on PCA space does not have the sparsity features, which are the features of PSNMF instead.

Table 3
Ratio Rules by PSNMF from NBA

field	RR_1	RR_2	RR_3
Games			0.450
Minute			0.013
Points Per Game			0.010
Rebound Per Game		0.117	
Assists per Game	0.206		
Steals	0.220		
Block Shots			
Total Rebound			
Fouls		0.263	
Field Goals			
3 Points			

Table 4
Ratio Rules by PCA from NBA

field	RR_1	RR_2	RR_3
Games			-0.586
Minute		0.280	0.332
Points Per Game			0.389
Rebound Per Game		-0.374	
Assists per Game	0.167		
Steals	0.229		
Block Shots			
Total Rebound			
Fouls		$-$ 0.320	
Field Goals			
3 Points			

5.1.4. Real Dataset: NBA (459×11)

The dataset comes from basketball statistics from the 1997–1998 seasons and is made up of 459 records/players, and each record owns 11 attributes values such as Minutes, Point per Game, Assist per Game, etc. The reason why we select this dataset is that it can give an intuitive meaning of such latent associations. Table 3 presents the first three Ratio Rules (RR_1, RR_2, RR_3) by PSNMF. Based on a general knowledge of basketball and through examination of these rules, we conjecture the RR_1 represent the agility of a player, which gives the ratio of Assists per Game and Steals, is 0.206:0.220 $\approx$ 1:1. It means that the average player who possess one time of assist per game will be also steal the ball one time. RR_2 shows the number of rebounds per game is correlated with Fouls times in a 0.117:0.263 $\approx$ 1:2.25 ratio, and this Ratio Rule can be interpreted with: an average player who makes better in rebound usually are easy to make more fouls per game. In this case, traditional methods can not give such ratio information behind the dataset. In additional, we list the results according to the method by PCA and finding some entities of Ratio Rule contain negative values (such as RR_2) in the Table 4. Obviously, it lacks intuition to explain the association that a player who adds 0.28 minute play time will obtain ($-$ 0.374) rebound per game and ($-$ 0.320) times of fouls.

5.2. Ratio rules on boolean matrix

Experiments on a Boolean dataset are performed to show that our proposed algorithm can also work well in the traditional dataset. First, we run our approach on binary data (e.g., market basket), the goal is to see if Ratio Rules could distinguish between three different groups of items where the groups are food

Table 5
Binary matrix of market basket (a) and its first three Ratio Rules (b)

	Bread	*Milk*	*Butter*	*Tire*	*Bulb*	*Oil*	*Shirt*	*Pants*
Fesco	1	1	1	0	0	0	0	0
Cookie	1	1	1	0	0	0	0	0
Tom	1	1	1	0	0	0	0	0
John	0	0	0	1	1	1	0	0
Tayor	0	0	0	1	1	1	0	0
Stan	0	0	0	1	1	1	0	0
Lee	0	0	0	0	0	0	1	1
Seung	0	0	0	0	0	0	1	1
Noise	0	1	0	0	1	0	0	1

(a)

field	RR_1	RR_2	RR_3
Bread	**0.330**	0.000	0.000
Milk	**0.331**	0.001	0.001
Butter	**0.333**	0.002	0.007
Tire	0.000	**0.328**	0.000
Bulb	0.002	**0.335**	0.013
Oil	0.001	**0.331**	0.000
Shirt	0.001	0.003	**0.492**
Pants	0.000	0.000	**0.487**
Support	49.8(%)	30.1(%)	20.1(%)

(b)

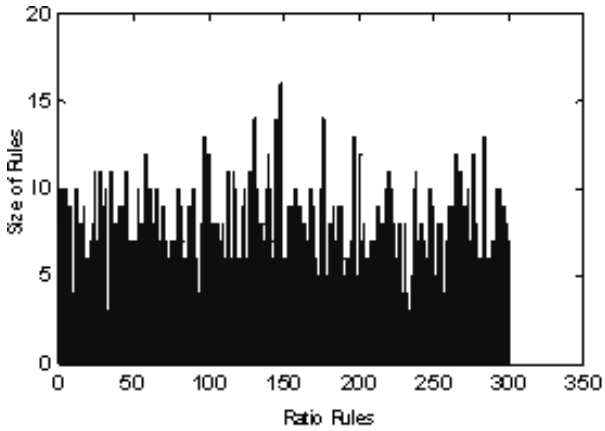


Fig. 5. Size of ratio rules.

(*Bread, Milk, Butter*), automotive (*Tire, Bulb, Oil*), and clothes (*Shirt, Pant*). Most of the matrix rows represented transactions involving items from one and only one group. In other words, given any pair of rows, all the items are either from exactly the same group or from two mutually disjoint groups. The rest of the rows were *'noise'*, which is generated by randomly selecting items across separate groups, and it is possible that representatives from several groups could be chosen. This matrix format is illustrated in Table 5(a). Table 5(b) shows the Ratio Rules for this type of matrix with 420 rows and with the 8 attributes listed above. The rows comprise of 47.8% from the first group, 28.6% from the second group, 19% from the third group and 4.8% noise, and the ratio of these three groups is 5:3:2. The dominate values of each rule vector are highlighted. As we will see, such three Ratio Rules are essentially: $RR_1 : Bread : Milk : Butter =$ 1:1:1, $RR_2 : Tire : Bulb : Oil =$ 1:1:1 and $RR_3 : Shirt : Pants =$ 1:1 It is surprising that their corresponding *support* value is closed to 5:3:2. In this case, the Ratio Rules are able to identify almost perfectly the three groups despite the presence of noise. RR_1 represents the "food" group, RR_2 the " automotive" group, and RR_3 the "clothes" group.

Another experiment is implemented on the synthetic dataset named "Sync" which is obtained with a commonly adopted dataset generator available from IBM Almaden2.[1] The dataset consists of 10000

[1] "Sync" Dataset is available at www.almaden.ibm.com/cs/quest/syndata.html.

Table 6
Characteristics of dataset "Syn"

Transaction	Items	Max tran size	Average size	Min tran size
10000	300	18	6	1

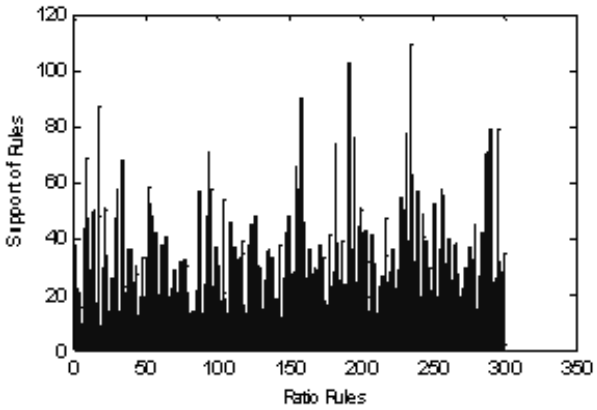


Fig. 6. Support of ratio rules.

records and 300 attributes. Table 6 lists the structure of such data. In this experiment, since most item values in each transact are zero, we find most ratio coefficients of Ratio Rules are also zero. We define a ratio itemset with the set of some non-zero entities of the rules, and the size of each itemset is listed in Fig. 5 and we find most of the sizes of each rule are very shorter than the entire item size 300 which means our rules own sparisity characteristics. Furthermore, we can measure the importance of each rule for the entire dataset by ranking the *support* value of these rules (Fig. 6). As we know, the higher the score of *support* represent the more importance than others. Therefore, according to (12), we can set a threshold to 90%, and then obtain $k = 220$ rules whose *support* values are higher than others as Ratio Rules which represent the latent associations within the dataset.

6. Conclusions

In essential, one contribution in this work is that we propose a novel method, called Principal Sparse Non-negative Matrix Factorization (PSNMF) for learning sparse non-negative components in matrix factorization. This work aims to learn latent components with sparse features, which are called Ratio Rules to express the quantifiable associations between itemsets. In addition, a *support* measurement is designed for weighting the importance of each rule for the entire dataset. Experimental results illustrate that our Ratio Rules are more suited for representing latent quantifiable associations between items effectively. Finally, although our approach is essentially a special case of non-negative matrix factorization, we believe that the proposed constraints are of significance in learning hidden components inherent in the non-negative data.

Acknowledgements

The authors, Chenyong Hu, Yongji Wang, Qing Wang, are partly supported by the National Natural Science Foundation of China under Grant Nos. 60373053, 50273026; the hundred Talents of the Chinese

Academy of Sciences; the National High-Tech Research and Development Plan of China (863) under Grant No. 2002AA116080. The work was partly done when Chenyong was an intern in Microsoft Research Asia. We are very grateful to the anonymous reviewers and editors.

References

[1] R. Agrawal, T. Imielinski and A.N. Swami, *Mining association rules between sets of items in large databases*, in Proceedings of the 1993 ACM SIGMOD, 1993, 207–216.
[2] R. Agrawal and R. Srikant, *Fast algorithms for mining association rules*, in Proceedings of the 20th VLDB, 1994, 487–499.
[3] Y. Aumann and Y. Lindell, *A statistical theory for quantitative association rules*, in Proceedings of the fifth ACM SIGKDD, 1999, 261–270.
[4] J.W. Guan, D.A. Bell and D.Y. Liu, *The Rough Set Approach to Association Rule Mining*, in Proceedings of the Third IEEE International Conference on Data Mining, 2003, 529–532.
[5] J. Han and Y. Fu, *Discovery of Multiple-Level Association Rules from Large Databases*, in Proceedings of the 21th VLDB, 1995, 420–431.
[6] F. Korn, A. Labrinidis, Y. Kotidis and C. Faloutsos, *Ratio Rules: A New Paradigm for Fast*, Quantifiable Data Mining, in Proceedings of the 24th VLDB, 1998, 582–593.
[7] D.D. Lee and H.S. Seung, Algorithms for Nonnegative Matrix Factorization, *Proceedings of the Advances in Neural Information Processing Systems* **13** (2001), 556–562.
[8] D.D. Lee and H.S. Seung, Learning the Parts of Objects by Non-Negative Matrix Factorization, *Nature* **401** (1999), 788–791.
[9] W. Li, J. Han and J. Pe, *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules*, in Proceedings of the 2001 IEEE International Conference on Data Mining, 2001, 369–376.
[10] J. Lin and M.H. Dunham, *Mining Association Rules: Anti-skew Algorithms*, Proceedings of the Fourteenth International Conference on Data Engineering, 1998, 486–493.
[11] M.E. Otey, C. Wang, S. Parthasarathy et al., *Mining Frequent Itemsets in Distributed and Dynamic Databases*, in Proceedings of the 2003 IEEE International Conference on Data Mining, 2003, 617–620.
[12] M. Kaya and R. Alhajj, *Facilitating Fuzzy Association Rules Mining by Using Multi-Objective Genetic Algorithms for Automated Clustering*, in Proceedings of the 2003 IEEE International Conference on Data Mining, 2003, 561–564.
[13] E.K.K. Ng, A.W.-C. Fu and K. Wang, *Mining Association Rules from Stars*, in Proceedings of the 2002 IEEE International Conference on Data Mining, 2002, 322–329.
[14] B.A. Olshausen and D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature* **381** (1996), 607–609.
[15] J.S. Park, M.-S. Chen and P.S. Yu, Using a Hash-Based Method with Transaction Trimming for Mining Association Rules, *IEEE Transactions on Knowledge and Data Engineering* **9**(5) (1997), 813–825.
[16] A. Schuster, R. Wolff and D. Trock, *A High-Performance Distributed Algorithm for Mining Association Rules*, in Proceedings of the 2003 IEEE International Conference on Data Mining, 2003, 291–298.
[17] R. Srikant and R. Agrawal, *Mining quantitative association rules in large relational tables*, in Proceedings of the 1996 ACM SIGMOD, 1996, 1–12.
[18] T. Hofmann, Unsupervised Learning by Probabilistic Latent Semantic Analysis, *Machine Learning* **42**(1–2) (2001), 177–196.
[19] H. Wang, C.S. Perng, S. Ma and P.S. Yu, *Mining Associations by Pattern Structure in Large Relational Tables*, in Proceedings of the 2002 IEEE International Conference on Data Mining, 2002, 482–489.
[20] R. Wolff and A. Schuster, *Association Rule Mining in Peer-to-Peer Systems*, in Proceedings of the 2002 IEEE International Conference on Data Mining, 2003, 291–298.
[21] M.J. Zaki, S. Parthasarathy, M. Ogihara and W. Li, New Parallel Algorithms for Fast Discovery of Association Rules, *Data Mining and Knowledge Discovery: An International Journal* **4**(1) (1997), 343–373.
[22] R. Zhao and W.I. Grosky, Narrowing the semantic gap improved text-based web document retrieval using visual features, *IEEE Transactions on Multimedia* **4**(2) (2002), 189–200.

Appendix

Section A

To prove the convergence of the leaning algorithm (6)–(7), an auxiliary function $G(H, Z')$ is given for objective function $L(Z)$ with the properties that $G(Z, Z') \geqslant L(Z))$ and $G(Z, Z) = L(Z)$, we will show that the multiplicative update rule corresponds to setting, at each iteration, the new state vector to the values that minimize the auxiliary function:

$$Z^{(t+1)} = \arg\min_{z} G(Z, Z^t) \tag{13}$$

Then the objective function $L(Z)$ is non-increasing when Z is updated using Eq. (13), because of

$$L(Z^{(t+1)}) \leqslant G(Z^{(t+1)}, Z^t) \leqslant G(Z^t, Z^t) = L(Z^t)$$

Updating H: with W fixed, H is updated by minimizing $L(H) = D(V \,\|WH\,)$. An auxiliary function is constructed for $L(H)$ as:

$$G(H, H') = \sum_{i,j} v_{ij} \log v_{ij} - \sum_{i,j,k} v_{ij} \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \left(\log (w_{ik} h_{kj}) - \log \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \right) + \sum_{i,j} y_{ij} - \sum_{i,j} v_{ij} + \lambda \sum_{i,j} h_{ij}$$

Since the entries of W are non-negative, it is easy to verify $\sum_j \|l_j\|_1 = \sum_{i,j} h_{ij}$, therefore, it is not difficult to testify $G(H, H) = L(H)$. The following proves $G(H, H') \geqslant L(H)$. Because $\log(\sum_k w_{ik} h_{kj})$ is a convex function, the following holds for all i, j and $\sum_k \mu_{ijk} = 1$

$$-\log(\textstyle\sum_k w_{ik} h_{kj}) \leqslant -\left(\sum_k \mu_{ijk} \log \frac{w_{ik} h_{kj}}{\mu_{ijk}} \right) \left(where\ \mu_{ijk} = \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \right)$$

Then

$$-\log \left(\textstyle\sum_k w_{ik} h_{kj} \right) \leqslant -\sum_k \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \left(\log w_{ik} h_{kj} - \log \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \right)$$

Thus, $G(H, H') \geqslant L(H)$.
To minimize $L(H)$, we update H by

$$H^{(t+1)} = \arg\min_{H} G(H, H^t)$$

This minimum is easily found by taking $\frac{\partial G(H,H')}{\partial h_{kl}} = 0$ for all $k,\ l$

$$\frac{\partial G(H, H')}{\partial h_{kl}} = -\sum_i v_{i.l} \frac{w_{ik} h'_{kj}}{\sum_k w_{ik} h'_{kj}} \frac{1}{h_{kl}} + \sum_i b_{i.k} + \lambda = 0$$

Solving for H, this gives

$$h_{kl} = h'_{kl} \sum_i v_{il} \frac{w_{ik}}{\sum_k (w_{ik} h'_{kl})} \Big/ \left(\sum_i w_{ik} + \lambda \right)$$

which is the desired updated H.

Updating W: with H fixed, W is updated by minimizing $L(W) = D(V \| WH)$. The auxiliary function for $L(W)$ is:

$$G(W, W') = \sum_{i,j} v_{ij} \log v_{ij} - \sum_{i,j,k} v_{ij} \frac{w'_{ik} h_{kj}}{\sum_k w'_{ik} h_{kj}} \left(\log (w_{ik} h_{kj}) - \log \frac{w'_{ik} h_{kj}}{\sum_k w'_{ik} h_{kj}} \right) + \sum_{i,j} y_{ij} - \sum_{i,j} v_{ij} + \lambda \sum_{i,j} h_{ij}$$

It is easily to prove $G(W, W) = L(W)$ and $G(W, W') \geqslant L(W)$. Likewise, by taking $\frac{\partial G(W,W')}{\partial w_{kl}} = 0$, we can get:

$$w_{kl} = w'_{kl} \sum_j v_{kj} \frac{h_{lj}}{\sum_k w'_{kl} h_{lj}} \Big/ \sum_j h_{lj}$$

Section B

In the above section, we have illustrated that the objective function is non-increasing under the above iterative updating rules (6)–(7). As we know, the development of the PSNMF algorithm also introduces normalization where all of the basis vectors in W are of unit length with respect to the one-norm. In this section, we will prove that such normalization applied in PSNMF does not lose the monotonicity of the two updates.

Theorem: Given a Non-negative Matrix Factorization $\widetilde{V} = WH$ introduced by the objection function Θ, for any vector norm $\mathbb{N} = \|.\|$, the basis vector W may be normalized to have columns of unit length in $\mathbb{N}$ without losing the monotone convergence of the generating objective function Θ.

Proof: First define a diagonal matrix $\mathbb{N}$ whose elements are the norms of each of r columns of the basis factor W:

$$N = \begin{bmatrix} \|w_1\| & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\ 0 & \|w_2\| & 0 & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \|w_i\| & & & \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & \ldots & 0 & \|w_r\| \end{bmatrix}.$$

Our strategy is to multiply N and its inverse $N^{-1} = diag\left(\frac{1}{\|w_1\|}, \frac{1}{\|w_2\|}, \ldots, \frac{1}{\|w_r\|} \right)$, between the factors in each iteration of the PSNMF algorithm induced by Θ. Then:

$$\widetilde{V} = WH = WN^{-1}NH = \widehat{W}\widehat{H} \tag{14}$$

where

$$\widehat{W} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \frac{w_1}{\|w_1\|} & \frac{w_2}{\|w_2\|} & \cdots & \frac{w_r}{\|w_r\|} \\ \vdots & \vdots & & \end{bmatrix}$$

so that $\widehat{W}$ now contains columns of unit length in $\mathbb{N}$ and $\widehat{H}$ is scaled appropriately (by N).In applying this normalization, we do not lose the equality in Eq. (14).Therefore, we conclude that the normalized factorization $\widehat{W}\widehat{H}$ obtained from any factorization WH results in monotonicity of Θ maintains this monotonicity.

This theorem shows explicitly that the theoretical error of $\widetilde{V}$ is not affected by the normalization process. For the implementation of the algorithm, the above proof allows us to enforce a specified norm without requiring more operations for each iteration. We therefore benefit from the computational ease of enforcing the "natural" one-norm (L1) in the W update before applying $N^{-1}N$ to the final factorization.