Mining Ratio Rules Via Principal Sparse Non-Negative Matrix Factorization

Chenyong Hu[1],Benyu Zhang[2],Shuicheng Yan[3],Qiang Yang[4],Jun Yan[3],Zheng Chen[2],Wei-Ying Ma[2]

[1] Institute of Software, CAS,Beijing, P.R. China
huchenyong@itechs.iscas.ac.cn

[2] Microsoft Research Asia, Beijing
{byzhang,zhengc,wyma}@microsoft.com

[3] LMAM, Peking University, Beijing, P.R. China
yanjun@math.pku.edu.cn
v-scyan @msrchina.research.microsoft.com

[4] Hong Kong University of Science and Technology
qyang@cs.ust.hk

Abstract

Association rules are traditionally designed to capture statistical relationship among itemsets in a given database. To additionally capture the quantitative association knowledge, F.Korn et al recently proposed a paradigm named Ratio Rules [4] 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 coefficient is 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.

1. Introduction

Association rules are one of the major representations in representing the knowledge discovered from large databases. The problem of association rule mining (ARM) in large transactional databases was introduced in [1, 3], Its basic idea is to discover important and interesting associations among the data items. The form of such association is as following:

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

To find association rules, most prevalent approaches assume the transactions only carry Boolean information and ignore the valuable knowledge inherent in the quantities of the items. In fact, considering that the quantities of the items normally contain valuable information for us, it is necessary to provide a definition of quantitative association rules when the datasets contain quantitative attributes. Several efficient algorithms for mining quantitative association rules have been proposed in the past [2, 7]. A notable algorithm is the work [4], where they provided a stronger set of rules as *Ratio Rules*. A rule under this framework is expressed in the following form:

$$bread : milk : butter = a : b : c$$
$$(a, b, c \ \textit{is arbitrary numerical values})$$

This rule states that for each *a* amount spent on bread, a customer normally spends *b* amount on milk and *c* amount on butter.

Principal Component Analysis (PCA) is often used to discover the *eigen-vectors* of a dataset. Ratio Rules [4] can represent the quantitative associations between items as the principal *eigen-vectors*, where the values *a, b* and *c* in the example above correspond to the projections of the eigenvector. Because the element of *eigen-vector* can be either positive or negative, sometime the ratio coefficient of Ratio Rules may contain negative value, such as

$$Shoe : \ Coat : \ Hat = 1 : \text{-}2 : \text{-}5$$

Obviously, such rule loses the intuitive appeal of associations between items, because a customer's spending should always be positive.

Our method amounts to a novel application of non-negative matrix factorization (NMF) [5]. However, we cannot directly apply NMF for our purpose, because it is still difficult to explain that these latent components represent the latent association between items in a quantifiable dataset. We need to provide a bridge to bring NMF closer to association rules.

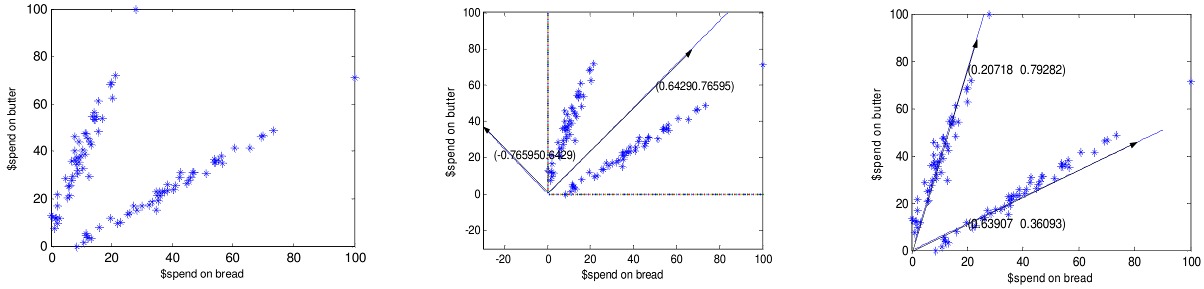
In this work, we propose a novel method called *Principal Sparse Non-Negative Matrix Factorization* (PSNMF), which adds the sparsity constraint as well as the non-negativity constraint in the standard NMF[5].

The rest of the paper is organized as follows: Section 2 describes the problem and the intuition behind the Ratio Rules. Section 3 introduces our new algorithm (PSNMF). Section 4 presents the experimental results. Section 5 concludes the paper. The convergence of PSNMF learning procedure is provided in Appendix.

2. Problem Definition

The problem that we tackle is as follows. Given a $N \times M$ matrix V (e.g., market basket databases), the entity v_{ij} gives the amount spent by customers on the product. The goal is to find all Ratio Rules of the form:

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE



(a) A data matrix with 2-dimension (b) Ratio Rules identified by PCA c)Ratio Rules identified by PSNMF

Fig 1. A data matrix and latent associations discovered by PCA and PSNMF

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

The above form means that customers who buys the items will spend v_1, v_2 … respectively on each itemset.

Fig1.(a) lists distribution of the matrix V which is organized with N customers and M $(M = 2)$ products Here we assume that the dataset is consisted with two clusters. Our goal is to capture the associations between items. We list two Ratio Rules discovered by PCA[4] in Fig.1 (b), where one contains negative values:

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

Obviously, the negative association between items (“bread” and “butter”) does not make sense. Furthermore, it is obvious that the Ratio Rules deviate with the latent associations behind the distribution of these points.

In fact, from Fig 1.(a), 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 based on PCA cannot truly reflect the latent associations among the items correctly. Compared to Fig.1 (b), Fig 1(c) illustrates the Ratio Rules captured by our proposed PSNMF. Surprisingly, each rule could be treated as an association in the two clusters respectively.

3. Principal Sparse Non-Negative Matrix Factorization (PSNMF)

Given a $M \times N$ non-negative matrix V, denote a set of $P \ll M$ basis components by a $M \times P$ matrix W, where each transaction (column vector) can be represented as a linear combination of the basis components using the approximate factorization:

$$V \approx WH \qquad (1)$$

where H is a $P \times N$ coefficients matrix.

3.1 Non-negative Matrix Factorization(NMF)

Because the entries of W and H calculated by PCA may contain negative values, NMF [5] is proposed as a procedure for matrix factorization which imposes non-negative instead of orthogonal constraint, and NMF uses the I-divergence of V from Y, which is defined as

$$D(V \parallel Y) = \sum_{i,j} (v_{ij} \log \frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}) \qquad (2)$$

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

$$\min_{W,H} D(V \| WH) \qquad s.t \ W, \ H \geq 0, \ \sum_i w_{ij} = 1 \ \ \forall j \qquad (3)$$

The above optimization can be done by using multiplicative update rules [5].

3.2 Sparse Non-negative Matrix Factorization (SNMF)

Although NMF is successful in Matrix Factorization, the NMF model does not impose the sparse constraints. Therefore, it can hardly yield a factorization, which reveals local sparse features in the data V. Related sparse coding is proposed in the work of [6] for matrix factorization.

Inspired by the original NMF and sparse coding, the aim of our work is to propose Sparse Non-negative Matrix Factorization (SNMF), which imposes the sparse and non-negative constraint. Therefore, we put forward the following constrained divergence as objective function:

$$D(V \parallel Y) = \sum_{i,j} (v_{ij} \log \frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}) \ + \lambda \sum_j \left\| l_j \right\|_1 \qquad (4)$$

$$l_j = \left(h_{1j}, h_{2j}, h_{3j}, \ldots, h_{pj}\right)^T \ denotes \ the \ column \ of \ H.$$

where $WH \triangleq Y = [Y_{ij}]$, and λ obtained by experience was assumed a positive constant. As the measurement of fitness for factorizing V into $WH \triangleq Y = [Y_{ij}]$, a SNMF factorization is defined as:

$$\min_{W,H} D(V \| WH) \qquad (5)$$

$$s.t \ \forall i, j: \ W_{ij} \geq 0, \ \ H_{ij} \geq 0, \ and \ \forall i \ \left\| w_i \right\|_1 = 1$$

Notice that we have chosen to measure sparseness by a linear activation penalty (i.e. minimum the 1-norm of the column of H). A Sparse solution to the above constrained minimization can be found by the following update rules:

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

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

To make the solution unique, we further require that the 1-normal of the column vector in matrix W is one. In addition, matrix H needs to be adjusted accordingly.

$$w_{kl} = w_{kl} \Big/ \sum_k w_{kl} \tag{8}$$

$$h_{kl} = h_{kl} \sum_k w_{kl} \tag{9}$$

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

3.3 Principal SNMF

When the dataset V is decomposed with W and H , each column value of H represents the corresponding projection on the basis space W . As a whole, the sum of every row vector of H represents the importance of corresponding base. Therefore, we define a *support* measurement after normalizing every column of H :

$$h_{kl} = h_{kl} \Big/ \sum_k h_{kl} \tag{10}$$

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

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

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 whole dataset.

In order to select the principal *k* rules as Ratio Rules, firstly, we rank the whole rules in descending by the *support* value. And then, retain the first *k* principal rules as Ratio Rules because they are more important than others. 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 above (12), Ratio Rules are obtained effectively according that the sum of *k* *support* values of rules cover *threshold (i.e.90%)* of the grand total *support* values.

4. Experiments

Synthetic dataset:

We have applied both the PSNMF and the PCA to a dataset that consists of two clusters, which contains 25 Gaussian distribution points on x-y plain (generated with mu=[3;5], sigma=[1,1.2;1.2,2]) and 50 points on y-z plain. (Fig2.)(Generated with mu=[3;5], sigma=[2,1.6;1.6,2]).

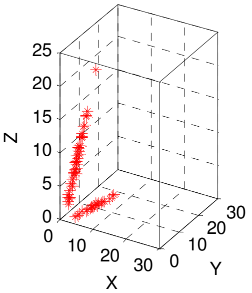


Fig 2.Dataset with two clusters

Table 1. Ratio Rules based on PSNMF and PCA

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	21.64	3.488
$Support(w_i)$	0.665	0.289	0.046

(a)Based on PSNMF

PCA	RR_1	RR_2
(X)	-0.52	0.72
(Y)	-0.77	-0.15
(Z)	-0.38	-0.68

(b) Based on PCA

Table 1.(a) lists all the rules and corresponding *support* values. After ranking such rules, Ratio Rules are obtained since $support(w_1) + support(w_2) = 0.9535 > 90\%$.

$rule_1 :: \; X:Y:Z \;=> 0:0.493:0.507$ (0.6650)

$rule_2 :: \; X:Y:Z \;=> 0.696:0.304:0$ (0.2885)

For example, 2/3 transactions (the cluster with distribution on y-z plain) are mostly depended on $rule_1$ and others on $rule_2$. Therefore, the corresponding *support* value (0.665) of $rule_1$ does not contradict with intuition. Otherwise, Table 1.(b) lists the Ratio Rules by PCA which is difficult to explain the negative association obviously.

Real Dataset: NBA (459×11)

This dataset comes from basketball statistics obtained from the 97-98 season, including minutes played, Point per Game, Assist per Game, etc. The reason why we select this dataset is that it can give a intuitive meaning of such latent associations. Table 2 presents the first three Ratio Rules (RR_1 , RR_2 , RR_3) by the PSNMF. Based on a general knowledge of basketball, 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, and so does $RR_2 (0.117: \; 0.263 \approx 1{:}2.25)$. In this case, traditional method cannot give such information behind the dataset.

Table 2. 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		
Fouls		0.263	
3Points			

5. Conclusion

In this work, we proposed Principal Sparse Non-Negative Matrix Factorization (PSNMF) for learning sparse non-negative components in matrix factorization. It aims to learn latent components which are called Ratio Rules. Experimental results illustrate that our Ratio Rules are more suited for representing associations between items than that by PCA.

References

[1] Agrawal, R., Imielinski, T. and Swami, A.N., Mining association rules between sets of items in large databases. In *the Proc. of the ACM SIGMOD*, (1993), 207-216.

[2] Aumann, Y. and Lindell, Y., A statistical theory for quantitative association rules. In *the Proc. KDD*, (1999).

[3] Han, J. and Fu, Y., Discovery of Multiple-Level Association Rules from Large Databases. In *Proc. of the VLDB*, (1995), 420--431.

[4] Korn, F., Labrinidis, A., Kotidis, Y. and Faloutsos, C., Ratio rules: A new paradigm for fast, quantifiable data mining. In *the Proc. of the VLDB*, (1998), 582-593.

[5] Lee, D.D. and Seung, H.S., Algorithms for nonnegative matrix factorization. In *Proc. of the Advances in Neural Information Processing Systems 13*, (2001), 556-- 562.

[6] Olshausen, B.A. and Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature 381(1996)*. 607--609.

[7] Srikant, r. and Agrawal, R., Mining quantitative association rules in large relational tables. In *Proc. of the ACM SIGMOD* (1996).

Appendix

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 $G(Z,Z')\geq L(Z))$ 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:

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

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

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

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

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

Since 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 $G(H,H')\geq L(H)$. Because $\log(\sum_{k}w_{ik}h_{kj})$ is a convex function, the following holds for all *i* ,*j* and $\sum_{k}\mu_{ijk}=1$:

$$-\log(\sum_{k}w_{ik}h_{kj})\leq-(\sum_{k}\mu_{ijk}\log\frac{w_{ik}h_{kj}}{\mu_{ijk}}) \quad (where\ \mu_{ijk}=\frac{w_{ik}h_{kj}^{'}}{\sum_{k}w_{ik}h_{kj}^{'}})$$

thus

$$-\log(\sum_{k}w_{ik}h_{kj})\leq-(\sum_{k}\frac{w_{ik}h_{kj}^{'}}{\sum_{k}w_{ik}h_{kj}^{'}}\left(\log w_{ik}h_{kj}-\log\frac{w_{ik}h_{kj}^{'}}{\sum_{k}w_{ik}h_{kj}^{'}}\right)$$

Thus, $G(H,H')\geq L(H)$.

To minimize $L(H)$, we update H by:

$$H^{(t+1)}=\arg\min_{H}G(H,H^{t}) \qquad (14)$$

for all kl

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

Solving for H , this gives:

$$h_{kl}=h_{kl}^{'}\sum_{i}v_{il}\frac{w_{ik}}{\sum_{k}\left(w_{ik}h_{kl}^{'}\right)}\Big/\left(\sum_{i}w_{ik}+\lambda\right)$$

which is the desired updated H .

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

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

It is easily to prove $G(W,W)=L(W)$ and $G(W,W')\geq L(W)$ likewise. we can get:

$$w_{kl}=w_{kl}^{'}\sum_{j}v_{kj}\frac{h_{lj}}{\sum_{k}w_{kl}^{'}h_{lj}}\Big/\sum_{j}h_{lj}$$

This completes the proof.