Features for unsupervised document classification

S H Srinivasan
Applied Research Group
Satyam Computer Services Ltd.
14 Langford Avenue, Lalbagh Road
Bangalore - 560 025, INDIA
SH_Srinivasan@satyam.com

Abstract

Unsupervised document classification is an important problem in practical text mining since training data is seldom available. In this paper we study the problem of term selection and the performance of various features for unsupervised text classification. The features studied are: principal components, independent components, and non-negative components. The clustering algorithm used is based on bipartite graph partitioning (Zha et al., 2001). The evaluation is performed using the newsgroups corpus.

1 Introduction

Many natural language processing applications require text classification. Labeled data for training is typically unavailable in many situations. So unsupervised classification techniques are called for. Classification – both supervised and unsupervised – is usually done in two steps: feature extraction and classification. Feature extraction is basically a change of representation of the raw data.

The most common representation used in document retrieval is the vector representation or the "bag-of-words" feature representation (Salton, 1989). Each document is represented as a vector in the term space. Let $t_1, t_2, \cdots, t_n$ be the *terms* we use to represent the documents.[1] This collection usually ignores very high and very low frequency terms. The terms can be arranged in some convenient order. The documents are represented as vectors of dimension n. Let v be the vector corresponding to a document d. We set v_i to 1 if term t_i occurs in d. Otherwise v_i is set to 0. This representation uses information about presence or absence of terms in the document. All the other information is ignored. Some representations use frequency information. The vectors corresponding to several documents can be arranged in a matrix – the so called "term-document" matrix. See (Berry et al., 1999) for a review of vector space representation. Other representation schemes can be in the vector space framework.

Classification is done in the feature space. Several techniques exist for classification – naive bayes, support vector machines, k-means clustering, neural networks etc. See (Sebastiani, 2002) for a comprehensive survey. In this paper, we use the bipartite graph matching technique proposed in (Zha et al., 2001) for classification. See also (Shi and Malik, 1997). Bipartite graph matching (also known as spectral clustering) has several attractive properties including the property that if the data has a "good" clustering, the algorithm will find an "optimal" one. See (Kannan et al., 2000) for definitions and details.

The term-by-document matrix can be considered as a representation of a weighted bipartite graph – the vertex sets being terms (T) and documents (D). The clustering technique partitions the vertices into disjoint sets ($T = T_1 \cup T_1^c$ and $D = D_1 \cup D_1^c$) such that intra cluster weight (between T_1 & D_1 and T_1^c & D_1^c) is maximized and inter cluster weight (between T_1 & D_1^c and T_1^c & D_1) is minimized.

The overall classification scheme is shown in figure 1. It can be seen that the final clustering scheme depends on the connectivity and weights of the bipartite graph. Any scheme which changes the connectivity or weights is likely to change the clustering results. In this paper we explore the following aspects.

[1] Terms are obtained from words after stemming.

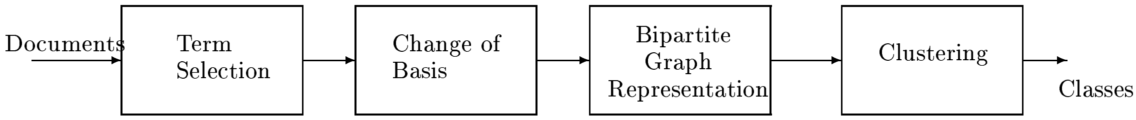

Figure 1: Document classification scheme.

term selection: The terms used for representing the documents are crucial to the success of the classification scheme. Several techniques exist for term selection: document frequency, information gain, etc (Sebastiani, 2002). In this paper, we use an iterative technique in conjunction with information gain.

basis selection: Terms form the basis of the original representation. These bases may not be optimal for the classification task. We explore the following representation schemes: singular vectors, independent vectors, and non-negative parts.

While these techniques have been applied to document processing (for example, (Dumais et al., 1988), (Kolenda and Hansen, 2000), (Lee and Seung, 1999)), there is no systematic comparison of these techniques for unsupervised document classification. The same clustering algorithm – bipartite graph partitioning – is used in the final stage.

The paper is organized as follows. Section 2 introduces the representation schemes used, section 3 describes the bipartite graph partitioning, and section 4 lists the experimental setting and results.

2 Feature representation schemes

The term-by-document matrix is a representation of the documents using one type of basis: terms. It is possible to extract other underlying bases which may aid classification. For example, matrix diagonalization (Strang, 1980) produces an orthogonal collection of basis vectors that are ordered according to their "importance".

In this section, we provide a brief description of singular value decomposition, independent component analysis, and non-negative matrix factorization. In the following discussion, let A be the $m \times n$ document-by-term matrix. Here m is the number of documents and n is the number of words (or terms) used in the representation.

2.1 Singular value decomposition

Matrix diagonalization can be performed only on square matrices. In contrast, singular value decomposition (SVD) exists for rectangular matrices also. SVD is a generalization of matrix diagonalization. The SVD of A can be written as

$$A = U \Sigma V^T$$

where U and V are $m \times m$ and $n \times n$ orthogonal matrices and Σ is a $m \times n$ diagonal matrix (Strang, 1980). The (non-zero) diagonal entries of Σ are called singular values. The columns of U corresponding to the singular values form a basis for the column space of A and those of V form a basis for the row space of A. Since the document vectors lie in the row space, the columns of V form a basis for the document vectors. The use of SVD in document retrieval is known as latent semantic analysis (Dumais et al., 1988). LSA has been applied for the resolution of synonymy and polysemy in document retrieval.

In this paper, we take the columns of V as the basis for the document space. We represent the document vectors in terms of these and then perform the clustering. In other words, the matrix VA^T is subjected to bipartite matching. Since this matrix can have negative entries, we use the absolute values.

2.2 Independent component analysis

Independent Component Analysis (ICA) is a generalization of stochastic interpretation of matrix diagonalization. Diagonalization, when applied to stochastic vectors, produces mutually uncorrelated basis. ICA produces a *statistically independent* basis. Thus the independent components of a matrix A can be thought of a collection of statistically independent sources for the rows (or columns) of A (Lee et al., 1998). The decomposition reveals the sources as well as mixing coefficients. The $m \times n$ matrix A is decomposed as

$$A = WS + N$$

where S is the $r \times n$ source signal matrix, W is the $m \times r$ mixing matrix, and N is the matrix of noise signals. Here r is the number of independent sources. The above decomposition can be performed for any number of independent components and the sizes of W and S vary accordingly. We subject the matrix SA^T to clustering. (Absolute values are used, as before.) In this case, the documents are represented in terms of r independent components instead of words. We use the Fast ICA algorithm for performing the decomposition (Hyvarinen, 1999).

ICA has been used for dimensionality reduction and representation of word histograms (Kolenda and Hansen, 2000).

2.3 Non-negative matrix factorization

In the above matrix factorizations, the components or basis vectors can have negative entries. The term-by-document matrix itself does not have negative entries. Thus negative factors are difficult to interpret. In particular, we cannot interpret them as *parts* of objects like documents. Non-negative matrix factorization (NMF) attempts a factorization in which the components have non-negative entries. The NMF of A is given by

$$A = WH$$

where the factors W and H contain only non-negative entries. The interpretation in (Lee and Seung, 1999) for the above decomposition is as follows: The columns of the $m \times n$ matrix A are the signals, the columns of the $m \times r$ matrix W are the basis signals, and the $r \times n$ matrix H is the mixing matrix. (Here r is the number of parts or non-negative components.) If we use the columns of W as the new basis, the new bipartite graph can be represented as $W^T A$. [In our case, the the signals of interest are documents and documents form the *rows* of A. Hence we subject $B = A^T$ to factorization. Let $B = WH$. The matrix used for clustering is $W^T B = W^T A^T$.]

Non-negative matrix factorization has been shown to discover semantic features (Lee and Seung, 1999).

3 Bipartite graph partitioning

Consider a weighted bipartite graph such as the term-document graph. Let the vertex sets be X and Y and the weight matrix be W. A partition of the graph into two subgraphs can be represented by a vertex partition of X and Y. Let $A \subseteq X$ and $B \subseteq Y$. Then the partition is given by the subgraphs induced by A & B and A^c & B^c. The *cut* between A and B is defined as

$$\text{cut}(A, B) = W(A, B^c) + W(A^c, B)$$

where

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

The normalized cut, *Ncut*, is defined as (Shi and Malik, 1997) (Zha et al., 2001)

$$\begin{aligned} \text{Ncut}(A, B) \quad = \quad & \frac{\text{cut}(A, B)}{W(A, Y) + W(X, B)} \\ & + \frac{\text{cut}(A^c, B^c)}{W(A^c, Y) + W(X, B^c)} \end{aligned}$$

The problem of finding partition with *minimum Ncut* can be posed as an eigenvalue problem (Shi and Malik, 1997) (Zha et al., 2001). The following is the algorithm derived in (Zha et al., 2001).

Let e be a vector of all 1s (of appropriate dimension).

1. Compute diagonal matrices D_X and D_Y as
$$We = D_X e, W^T e = D_Y e$$
2. Let $\tilde{W} = D_X^{-1/2} W D_Y^{-1/2}$
3. Compute the *second* largest left and right singular vectors of $\tilde{W}$, $\tilde{x}$ and $\tilde{y}$.
4. Find cut points c_x and c_y for $x = D_X^{-1/2} \tilde{x}$ and $y = D_Y^{-1/2} \tilde{y}$.
5. Let $A = \{i : x_i \geq c_x\}$, $A^c = \{i : x_i < c_x\}$, $B = \{j : y_j \geq c_y\}$, and $B^c = \{j : y_j < c_y\}$,

The graphs $G(A, B)$ and $G(A^c, B^c)$ can be further partitioned.

4 Experiments

To test the effectiveness of different feature representations, we use the scheme of (Zha et al., 2001). The task used is binary clustering - discrimination between two news groups. The corpus used is the 20 newsgroups database.[2] The newsgroups and the associated labeling scheme of (Zha et al., 2001) are listed below:

```
NG1: alt.atheism
NG2: comp.graphics
NG3: comp.os.ms-windows.misc
NG4: comp.sys.ibm.pc.hardware
NG5: comp.sys.mac.hardware
NG6: comp.windows.x
NG7: misc.forsale
NG8: rec.autos
NG9: rec.motorcycles
NG10: rec.sport.baseball
NG11: rec.sport.hockey
NG12: sci.crypt
NG13: sci.electronics
NG14: sci.med
NG15: sci.space
NG16: soc.religion.christian
NG17: talk.politics.guns
NG18: talk.politics.mideast
NG19: talk.politics.misc
NG20: talk.religion.misc
```

We consider three binary classification tasks: between NG1 & NG2, NG10 & NG11, and NG18 & NG19. It can be seen that NG1 and NG2 are well-separated, NG10 and NG11 have some overlap, and NG18 and NG19 have more overlap. To compare with the results of (Zha et al., 2001), we perform clustering for four different sample sizes: 50, 100, 150, 200 and 250 files for the second class.[3] The first class always has 50 files.

Each run of the experiment was performed as follows.

1. The required number of files were chosen randomly from each class.

2. The documents were represented using N terms. (In the experiments reported below, $N = 200$.) The terms were chosen according to maximum information gain criterion.

   The information gain between terms and documents is defined as

   $$IG(t,d) = \sum_d \sum_t p(t,d) \log \left(\frac{p(t,d)}{p(t)p(d)} \right) \quad (1)$$

   where t is term and d is document. This was done using the *bow* toolkit.[4]

3. The documents were represented using the N terms. We call this representation in *term* basis.

4. The document vectors were also represented using other bases – singular vectors, independent components, and non-negative components.

5. The resulting matrices were subjected to bipartite graph partitioning. The best partitions were calculated by choosing c_x and c_y which minimize the objective function (Shi and Malik, 1997).

The results of the experiments are shown in table 1. The results of (Zha et al., 2001) are shown in table 2 for comparison. The differences in results can be attributed to

1. Zha et al. (2001) use maximum mutual information to choose terms.

2. While we choose 200 terms in all experiments, number of terms used by Zha et al. (2001) is not available.

It can be seen that the performance confirms to our initial expectation: classification accuracies are smaller when the classes are not well-separated. It is surprising that the default term-basis performs better than other derived bases.

In all the derived bases, we have an extra degree of freedom: the number of components chosen. In the previous set of experiments, we used 200 independent components and 100 non-negative components. We reduced the number of independent components and non-negative components by half and the results are shown in table 3. ICA performs better in the presence of uncertainty (NG18/NG19 and unbalanced data sets for NG10/NG11).

[2] The database is available from http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.

[3] (Zha et al., 2001) use the sample sizes; 50, 100, 150, and 200.

[4] The toolkit is also available from http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.

Newsgroups: NG1/NG2

Mixture	Terms	SVD	ICA	NMF
50/50	95.16 ± 0.14	93.00 ± 0.81	86.26 ± 0.31	93.32 ± 0.62
50/100	93.51 ± 0.16	87.47 ± 1.58	88.42 ± 0.42	91.16 ± 0.73
50/150	93.33 ± 0.09	87.57 ± 0.23	86.90 ± 1.18	87.00 ± 0.30
50/200	88.18 ± 0.57	84.50 ± 0.55	86.02 ± 1.93	80.74 ± 0.34
50/250	82.24 ± 0.59	78.41 ± 0.26	79.14 ± 2.01	78.16 ± 0.28

Newsgroups: NG10/NG11

Mixture	Terms	SVD	ICA	NMF
50/50	75.42 ± 1.39	84.68 ± 0.58	60.74 ± 0.41	78.79 ± 1.69
50/100	81.02 ± 0.82	74.74 ± 1.56	68.49 ± 0.56	63.02 ± 0.98
50/150	74.40 ± 1.45	63.10 ± 0.66	61.13 ± 0.19	60.35 ± 0.18
50/200	74.29 ± 0.85	59.92 ± 0.33	61.12 ± 0.27	61.87 ± 0.18
50/250	71.56 ± 1.06	58.37 ± 0.20	59.72 ± 0.29	65.68 ± 0.18

Newsgroups: NG18/NG19

Mixture	Terms	SVD	ICA	NMF
50/50	62.95 ± 0.34	60.89 ± 0.26	62.11 ± 0.37	59.16 ± 0.31
50/100	69.26 ± 0.48	66.89 ± 0.37	64.22 ± 0.25	64.22 ± 0.27
50/150	68.58 ± 0.42	66.55 ± 0.30	67.50 ± 0.34	66.29 ± 0.13
50/200	67.41 ± 0.91	68.12 ± 0.15	69.62 ± 0.25	68.14 ± 0.18
50/250	70.08 ± 1.43	67.94 ± 0.18	68.71 ± 0.10	68.15 ± 0.10

Table 1: Experimental results for three binary classification tasks. Percent correct figures are reported. The statistics were measure over 20 runs for each case.

Mixture	NG1/NG2	NG10/NG11	NG18/NG19
50/50	92.12 ± 3.52%	74.56 ± 8.93%	73.66 ± 10.53%
50/100	90.57 ± 3.11%	67.13 ± 7.17%	67.23 ± 7.84%
50/150	88.04 ± 3.90%	58.30 ± 5.99%	65.83 ± 2.79%
50/200	82.77 ± 5.24%	57.55 ± 5.69%	61.23 ± 9.88%

Table 2: The results of (Zha et al., 2001).

4.1 Term selection

Note that when the initial terms selection is done using equation 1, the documents are not labeled. But after first round of classification, we have tentative class labels available. This can be used to refine the choice of terms. For this we use class labels instead of document labels in equation 1. The overall scheme is shown in figure 2. Figure 3 shows how the scores improve with this iterative term selection.

We can use this idea to improve classification accuracy. Table 4 gives the *best case* results. *The figures are obtained by using the knowledge of classes when choosing terms using equation 1.* These can be viewed classification accuracies when best term selection is performed. Figure 4 shows the actual improvement of scores for a particular case.

5 Related work

There have been several attempts to improve document clustering accuracy. (Moore et al, 1999) use hypergraph partitioning and show that the resulting clusters have smaller entropy compared to other clustering techniques. (Strehl et al.,) study the effect of various similarity measures and clustering algorithms on clustering web pages. They conclude that cosine and extended Jaccard similarities and weighted graph partitioning give good results. (Dhillon

80.95 ± 5.58	88.20 ± 6.75
86.37 ± 0.14	90.57 ± 0.57
72.39 ± 4.45	72.64 ± 3.73
75.51 ± 4.00	73.29 ± 3.10
73.59 ± 3.61	71.37 ± 2.80

60.55 ± 0.53	70.10 ± 1.68
75.61 ± 1.14	62.67 ± 0.22
76.29 ± 1.46	61.18 ± 0.15
74.21 ± 0.78	61.79 ± 0.36
79.05 ± 1.18	64.09 ± 0.15

58.75 ± 0.36	60.55 ± 0.27
62.67 ± 0.40	63.97 ± 0.14
68.18 ± 0.24	65.79 ± 0.24
74.10 ± 0.21	68.25 ± 0.09
76.82 ± 0.36	69.79 ± 0.07

Table 3: Classification accuracies when the number of independent and non-negative components are reduced. The three tables correspond to NG1/NG2, NG10/NG11, and NG18/NG19. The columns of each table correspond to ICA and NMF. The rows correspond to different mixture compositions: 50/50, 50/100, 50/150, 50/200, and 50/250. It can be seen that ICA performs better than term-basis when the classes are overlapping (NG18/NG19).

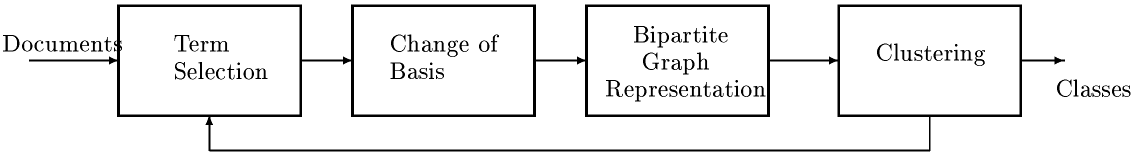

Figure 2: Iterative term selection.

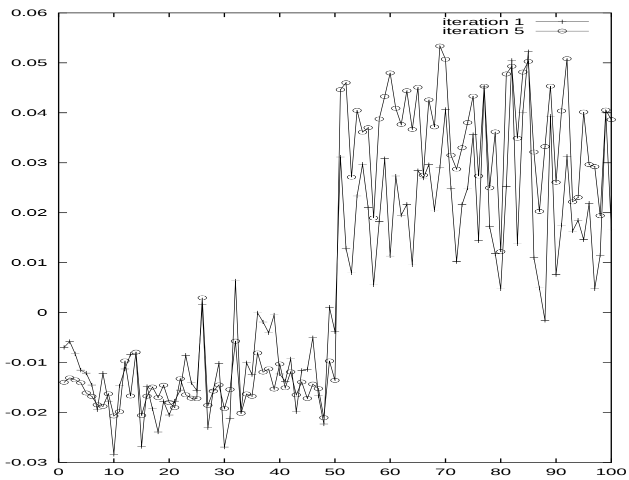

Figure 3: Evolution of scores for binary classification (iterations 1 and 5). 50 documents from NG1 and an equal number of documents from NG2 are chosen. For ease visualization, scores of documents from NG1 occur before those of NG2. Terms for first iteration are chosen using equation 1. Terms for subsequent iterations are chosen using the classification labels of previous iteration in the same equation. It can be seen that the discrimination increases.

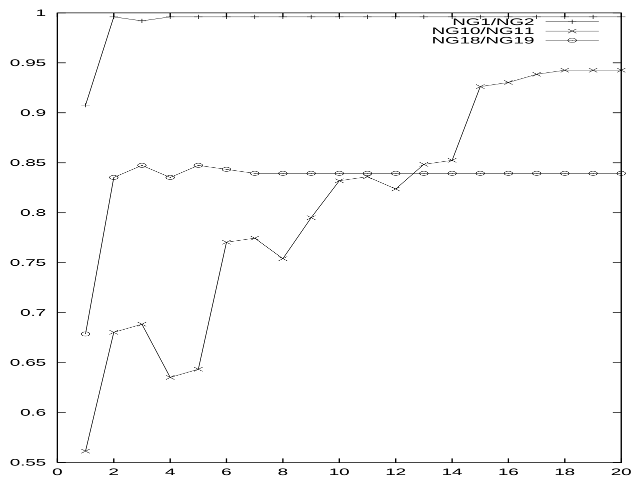

Figure 4: Classification accuracy for iterative feature selection.

and Modha, 2001) propose spherical k-means clustering and compare it to SVD. The basis produced by spherical k-means is more localized. The algorithms have been tried on a variety of datasets. (None of the above papers use the 20 newsgroups dataset.)

6 Conclusions

The experiments presented here capture two common scenarios in document classification: class overlap and data insufficiency. Term-based representations perform well when there is little overlap and small amount of data. ICA-based representations perform well when there is sufficient data even for overlapping cases. A crucial parameter in case of ICA and NMF is the number of components chosen. Choosing the optimal number of components will be an interesting extension of the work reported here.

Newsgroups: NG1/NG2

Mixture	Terms	SVD	ICA	NMF
50/50	99.30 ± 0.00	99.30 ± 0.00	97.40 ± 0.01	99.70 ± 0.00
50/100	99.08 ± 0.01	99.50 ± 0.01	97.83 ± 0.01	97.33 ± 0.04
50/150	92.88 ± 2.39	92.31 ± 2.23	92.94 ± 1.76	86.75 ± 0.75
50/250	99.07 ± 0.00	96.18 ± 0.07	98.67 ± 0.00	83.16 ± 0.13

Newsgroups: NG10/NG11

Mixture	Terms	SVD	ICA	NMF
50/50	98.00 ± 0.04	95.50 ± 0.07	96.80 ± 0.11	98.20 ± 0.03
50/100	97.80 ± 0.01	93.47 ± 0.15	96.87 ± 0.04	92.07 ± 0.08
50/150	97.50 ± 0.01	91.25 ± 0.25	97.50 ± 0.02	84.05 ± 0.40
50/250	97.36 ± 0.01	88.04 ± 0.22	97.92 ± 0.01	83.92 ± 0.85

Newsgroups: NG18/NG19

Mixture	Terms	SVD	ICA	NMF
50/50	90.80 ± 0.47	73.00 ± 0.99	91.70 ± 0.09	79.90 ± 0.86
50/100	94.07 ± 0.07	85.19 ± 0.31	94.52 ± 0.03	86.89 ± 0.33
50/150	91.70 ± 0.23	86.65 ± 0.37	94.80 ± 0.03	85.95 ± 0.24
50/250	91.48 ± 0.17	81.92 ± 1.43	93.00 ± 0.02	85.40 ± 0.15

Table 4: Results for "best case" term selection.

References

M W Berry, Z Drmac, and E R Jessup. 1999. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41:335–362.

I S Dhillon and D S Modha. 2001. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175.

S T Dumais, G W Furnas, T K Landauer, and S Deerwester. 1988. Using latent semantic analysis to improve information retrieval. In *Proceedings of CHI'88*.

A Hyvarinen. 1999. Fast and robust fixed-point algorithms for indepdendent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634.

R Kannan, S Vempala, and A Vetta. 2000. On clusterings: Good, bad, and spectral. In *Proceedings of FOCS*, pages 367–377.

T Kolenda and L Hansen. 2000. Independent components in text. In Mark Girolami, editor, *Advances in Independent Component Analysis*, chapter 13. Springer Verlag.

D D Lee and S Seung. 1999. Learning parts of objects by non-negative matrix factorization. *Nature*, 401:788–791.

T.-W Lee, M Girolami, A J Bell, and T J Sejnowski. 1998. A unifying information-theoretic framework for independent component analysis. *International Journal on Mathematical and Computer Modeling.*

J Moore et al. 1999. Web page categorization and feature selection using associate rule and principal component clustering. *Decision Support Systems*, 27(3):329–341.

G Salton. 1989. *Automatic text processing: The transformation, analysis, and retrieval of information by computer.* Addison-Weslsy.

F Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys.*

J Shi and J Malik. 1997. Normalized cuts and image segmentation. In *Proc of IEEE Conference on Computer Vision and Pattern Recognition*, June.

Gilbert Strang. 1980. *Linear algebra and its applications.* Academic Press.

A Strehl, J Ghosh, and R Mooney. Impact of similarity measures on web-page clustering. In *Workshop of Artificial Intelligence for Web Search (AAAI 2000).* AAAI.

H Zha, X He, C Ding, H Simon, and M Gu. 2001. Bipartite graph partitioning and data clustering. In *Proc CKIM 01*, Nov 5-10.