Document Retrieval and Clustering: from Principal Component Analysis to Self-aggregation Networks

Chris H.Q. Ding, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. chqding@lbl.gov

Abstract. We first extend Hopfield networks to clustering bipartite graphs (words-to-document association) and show that the solution is the principal component analysis. We then generalize this via the min-max clustering principle into a self-aggregation networks which are composed of scaled PCA components via Hebb rule. Clustering amounts to an updating process where connections between different clusters are automatically suppressed while connections within same clusters are enhanced. This framework combines dimension reduction with clustering via neural networks and PCA. Self-aggregation networks can also improve information retrieval performance. Applications are presented.

1 Introduction

Clustering documents[11] is a challenging problem because of the very high dimensionality; in vector space model, the dimensionality is the size of vocabulary. In recent years, dimension reduction techniques such as principal component analysis (PCA) (which is also called Latent semantic indexing (LSI)[2]) are popularly used to project the documents into the low-dimensional space.

Feedforward networks[1] via backpropagation has been widely used for classification tasks such as text categorization [20]. Although Hopfield associative-memory networks[10] is not suitable for classification, it has the flexibility to be adopted for solving combinatorial problems[9] such as traveling saleman problem, graph partitioning, etc.

In this paper, we explore the relationship between data clustering and dimension reduction via the neural networks connection. We show that using Hopfield networks to cluster the bipartite graph (word-document association matrix), PCA is the solution. This provides justification for clustering using PCA (see §2).

By appropriately modifying the clustering objective function according to a min-max clustering principle, we obtain a min-max cut clustering algorithm whose equations are essentially rescaling of those for PCA (see §3).

Using scaled PCA components we can construct self-aggregation networks which have the unique property of cluster self-aggregation: connections between different clusters are automatically suppressed while connections within same clusters are enhanced. An indepth analysis of self-aggregation (SA) networks are provided (see §4).

We use SA networks for document retrieval and obtained improved retrieval precision. We also use SA networks for clustering documents and words simultaneously, and obtain substantially better results than the K-means method (see §5).

2 Hopfield networks for clustering documents

In the rectangular $m \times n$ term-document association matrix $B = (b_{ij})$, each row represents a word and is denoted by an r-node in a weighted bipartite graph shown in Fig.1. Each column represents a document and is denoted by a c-node. Element b_{ij} in the matrix represents the counts of co-occurrence of row object r_i and column object c_j, and is represented by a weighted edge between r_i and c_j.

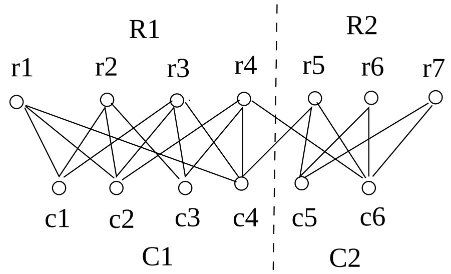

Figure 1: A bipartite graph with r-nodes and c-nodes. The dashed line indicates a possible partitioning.

Hopfield networks can be used to partition an standard undirected graph [9]. In this section, we extend Hopfield networks for partition bipartite graph, and show that the relaxed version of the Hopfield networks for bipartite graphs is precisely the Latent Semantic Indexing.

We wish to partition the r-type nodes of R into two parts R_1, R_2 and simultaneously partition the c-type nodes of C into two parts C_1, C_2, based on the clustering principle of minimizing between-cluster association and maximizing within-cluster association (see Fig.1). We use indicator vector $\mathbf{f}$ to determine how to split R into R_1, R_2:

$$f(i) = \begin{cases} 1 & \text{if} \quad r_i \in R_1 \\ -1 & \text{if} \quad r_i \in R_2 \end{cases} \tag{1}$$

and use $\mathbf{g}$ to determine how to split C into C_1, C_2:

$$g(i) = \begin{cases} 1 & \text{if} \quad c_i \in C_1 \\ -1 & \text{if} \quad c_i \in C_2 \end{cases} \tag{2}$$

(For presentation purpose, we index the nodes such that nodes within same cluster are indexed contiguously. The clustering algorithms presented are independent to this assumption. Bold face lower case letters are vectors. Matrices are denoted by upper case letters.) Thus we may write

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}^{(+)} \\ \mathbf{f}^{(-)} \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \mathbf{g}^{(+)} \\ \mathbf{g}^{(-)} \end{pmatrix} \tag{3}$$

With this indexing, the association matrix is

$$B = \begin{pmatrix} B_{R_1,C_1} & B_{R_1,C_2} \\ B_{R_2,C_1} & B_{R_2,C_2} \end{pmatrix} \tag{4}$$

It is convenient to convert the bipartite graph into an undirected graph. We follow standard procedure and combine the two types nodes to one by setting

$$\mathbf{q} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \; W = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}, \tag{5}$$

This induces an undirected graph G, whose adjacency matrix is the symmetric weight matrix W.

Consider the following objective function,

$$J_{\text{cut}}(C_1, C_2; R_1, R_2) = \frac{1}{2}\mathbf{q}^T W \mathbf{q} \tag{6}$$

$$= s(B_{R_1,C_1}) + s(B_{R_2,C_2}) - s(B_{R_1,C_2}) - s(B_{R_2,C_1})$$

where

$$s(B_{R_1,C_2}) \equiv s(R_1, C_2) \equiv \sum_{r_i \in R_1, c_j \in C_2} b_{ij},$$

and $s(B_{R_2,C_1}), s(B_{R_1,C_1}), s(B_{R_2,C_2})$ are similarly defined. $s(B_{R_1,C_1})$ is the association within cluster 1 (see Fig.1), and we call it the self-association. $s(B_{R_2,C_2})$ is the self-association of cluster 2. $s(B_{R_1,C_2})$ and $s(B_{R_2,C_1})$ are the overlaps between different clusters.

We propose a *min-max clustering principle*: data points are grouped into clusters such that the overlaps $s(B_{R_1,C_2})$, $s(B_{R_2,C_1})$ between different clusters are minimized while cluster self-similarities (B_{R_1,C_1}), $s(B_{R_2,C_2})$ are maximized[5]. Maximizing $s(B_{R_1,C_1}) + s(B_{R_2,C_2})$ while minimizing $s(B_{R_1,C_2})$ $+ s(B_{R_2,C_1})$ is equivalent to maximizing the objective function $J_{\text{cut}}(\mathbf{q})$.

Using Hopfield network [10, 9], the solution is obtained by the update rule

$$q^{(t+1)}(i) = \text{sgn}[\sum_j w_{ij} q^{(t)}(j)].$$

where $\mathbf{q}^{(t)}$ is the value of $\mathbf{q}$ at t-th update. This equation can be written in vector form $\mathbf{q}^{(t+1)} = \text{sgn}[W\mathbf{q}^{(t)}]$. One can verify that $J_{\text{cut}}(\mathbf{q})$ monotonically decreases in this update.

If one relaxes $q(i)$ from discrete indicators to continuous values in $(-1, 1)$, the solution $\mathbf{q}$ satisfies

$$W\mathbf{q} = \lambda \mathbf{q}. \tag{7}$$

Now utilizing the explicit structures of W and $\mathbf{q}$, we have

$$\begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}. \tag{8}$$

which is identical to

$$B\mathbf{g} = \lambda \mathbf{f}, \quad B^T \mathbf{f} = \lambda \mathbf{g}. \tag{9}$$

The solutions to these two equations are the singular value decomposition (SVD) of B. To see clearly, upon substitutions, we have

$$(BB^T)\mathbf{f} = \lambda^2 \mathbf{f}, \quad (B^T B)\mathbf{g} = \lambda^2 \mathbf{g}. \tag{10}$$

This verifies that $\{\mathbf{f}_i\}$ are left singular vectors and $\{\mathbf{g}_i\}$ are right singular vectors of the SVD of B:

$$B = \sum_{k=1}^{m} \mathbf{f}_k \lambda_k \mathbf{g}_k^T = F_m \Lambda_m G_m^T. \tag{11}$$

We summarize these results in
Theorem 1. Using Hopfield networks to maximize the objective function $J_{\text{cut}}(\mathbf{q})$ of Eq.(6), the solutions for clustering indicators are given by SVD of B.

Several further results can be obtained. First, note that SVD of B are precisely the *Latent Semantic Indexing* [2]. Thus we conclude that Hopfield networks for clustering leads to LSI. The partitioning indicator vectors are the LSI index vectors.

Second, because $s(B_{R_1,C_1}) + s(B_{R_2,C_2}) + s(B_{R_1,C_2}) + s(B_{R_2,C_1}) = \sum_{ij} b_{ij} \equiv s$ is a constant for a given association matrix B, we have $J_{\text{cut}} = s - 2[s(B_{R_1,C_2}) + s(B_{R_2,C_1})]$. Therefore, maximizing $J_{\text{cut}}(\mathbf{q})$ is equivalent to minimizing

$s(B_{R_1,C_2})+s(B_{R_2,C_1})$ alone. In graph theory, $s(B_{R_1,C_2})+s(B_{R_2,C_1})$ is the sum of weights on the edges being cut, and is called cutsize. Therefore, PCA is equivalent to MinCut in graph theory. It is well known that MinCut often leads to skewed cuts. This imbalance will be addressed in §3.

Thirdly, all these are connected to K-means clustering. Consider the K-means squared error objective function,

$$J_{\text{Kmeans}} = \sum_{k=1}^{K}\sum_{\mathbf{x}_i\in c_k}||\mathbf{x}_i-\mathbf{c}_k||^2 = \sum_{k=1}^{K}\sum_{\mathbf{x}_i,\mathbf{x}_j\in c_k}\frac{||\mathbf{x}_i-\mathbf{x}_j||^2}{n_k} \tag{12}$$

$$\simeq \frac{1}{\bar{n}_k}\sum_{k=1}^{K}\sum_{\mathbf{x}_i,\mathbf{x}_j\in c_k}||\mathbf{x}_i-\mathbf{x}_j||^2 \tag{13}$$

$$= \frac{1}{\bar{n}_k}\left[\sum_{ij}||\mathbf{x}_i-\mathbf{x}_j||^2 - \sum_{p\neq q}\sum_{\mathbf{x}_i\in c_p}\sum_{\mathbf{x}_j\in c_q}||\mathbf{x}_i-\mathbf{x}_j||^2\right] \tag{14}$$

where $\mathbf{x}_j$ is the j-th document: $B=(\mathbf{x}_1,\cdots,\mathbf{x}_2)$; $\mathbf{c}_k, n_k$ are the centroid and size of k-th cluster, and $\bar{n}_k$ is a suitable constant represents approximately the number of points in a cluster on average. In Eq.(14), the first term is a constant, and the second term is the sum of distances between documents in different clusters, which is analogous to overlapping association between different clusters, $s(B_{R_1,C_2})+s(B_{R_2,C_1})$. Therefore, Hopfield network (and PCA) has a nice connection to the K-means clustering: one minimizes the between-cluster associations (similarities) whereas the other maximizes the between-cluster distances (di-similarities).

All results in this section for bipartite graphs can be immediately extended to an undirected graph, $G(A)$, with adjacency matrix A. The clustering objective function Eq.6 becomes

$$J_{\text{cut}}(C_1,C_2) = s(A_{C_1,C_1}) + s(A_{C_2,C_2}) - 2s(A_{C_1,C_2}) \tag{15}$$

where $s(A_{C_1,C_2})$ is defined similar to $s(B_{R_1,C_2})$. The clustering indicators $\mathbf{g}$ of Eq.2 via the Hopfield network are given by the eigenvector of $A\mathbf{g}=\lambda\mathbf{g}$.

3 MinMaxCut

Approximately speaking, the above Hopfield network of maximizing Eq.6 is equivelant to

$$\min \frac{s(B_{R_1,C_2})+s(B_{R_2,C_1})}{s(B_{R_1,C_1})+s(B_{R_2,C_2})}. \tag{16}$$

Maximization of $s(B_{R_1,C_1})+s(B_{R_2,C_2})$ does not guarrentee the balance of the two terms; in fact it often happens that $s(B_{R_1,C_1}) \gg s(B_{R_2,C_2})$ or $s(B_{R_1,C_1}) \ll s(B_{R_2,C_2})$. To prevent this imbalance of cluster self-associations, we add a cluster balance condition in the min-max clustering principle that $s(B_{R_1,C_1})$, $s(B_{R_2,C_2})$ are maximized *individually* while overlap associations $s(B_{R_1,C_2})+s(B_{R_2,C_1})$ are minimized. This leads to the MinMaxCut objective

$$J_{\text{MMC}}(C_1,C_2;R_1,R_2) = \frac{s(B_{R_1,C_2})+s(B_{R_2,C_1})}{2s(B_{R_1,C_1})} + \frac{s(B_{R_1,C_2})+s(B_{R_2,C_1})}{2s(B_{R_2,C_2})} \tag{17}$$

in contrast to J_{cut} in Eq.(6).

To find an efficient algorithm to compute the optimal solution according to $J_{\text{MMC}}(C_1,C_2;R_1,R_2)$ we proceed as follow. First, we write the weight matrix W explicitly,

$$W = \begin{pmatrix} 0 & 0 & B_{R_1,C_1} & B_{R_1,C_2} \\ 0 & 0 & B_{R_2,C_1} & B_{R_2,C_2} \\ B^T_{R_1,C_1} & B^T_{R_2,C_1} & 0 & 0 \\ B^T_{R_1,C_2} & B^T_{R_2,C_2} & 0 & 0 \end{pmatrix} \tag{18}$$

Now we re-order the indices of the nodes,

$$\mathbf{q} = \begin{pmatrix} \mathbf{f}^{(+)} \\ \mathbf{f}^{(-)} \\ \mathbf{g}^{(+)} \\ \mathbf{g}^{(-)} \end{pmatrix} \quad \Rightarrow \quad \mathbf{q} = \begin{pmatrix} \mathbf{f}^{(+)} \\ \mathbf{g}^{(+)} \\ \mathbf{f}^{(-)} \\ \mathbf{g}^{(-)} \end{pmatrix},$$

i.e., nodes with Cluster 1 are indexed contiguously irrespect wether they are r-nodes or c-nodes. With this reordering, W becomes[22]

$$W = \begin{pmatrix} 0 & B_{R_1,C_1} & 0 & B_{R_1,C_2} \\ B^T_{R_1,C_1} & 0 & B^T_{R_2,C_1} & 0 \\ 0 & B_{R_2,C_1} & 0 & B_{R_2,C_2} \\ B^T_{R_1,C_2} & 0 & B^T_{R_2,C_2} & 0 \end{pmatrix} \tag{19}$$

This can be viewed as an undirected graph, with adjacency matrix

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}. \tag{20}$$

From this, Eq.(17) can be written as

$$J_{\text{MMC}} = \frac{s(W_{12})}{s(W_{11})} + \frac{s(W_{12})}{s(W_{22})}. \tag{21}$$

Eq.(21) is the min-max cut objective function for undirected graph [5]. One can show that

$$\min_{\mathbf{q}} J_{\text{MMC}}(\mathbf{q}) \Rightarrow \min_{\mathbf{q}} \frac{\mathbf{q}^T(D-W)\mathbf{q}}{\mathbf{q}^T D\mathbf{q}}, \tag{22}$$

subject to $\mathbf{q}^T W\mathbf{e} = \mathbf{q}^T D\mathbf{e} = 0$, where $D=(d_i)$ is a diagonal matrix and $d_i=\sum_j w_{ij}$ is the degree of node i and $\mathbf{e}=(1,\cdots,1)^T$. We relax $q(i)$ from discrete indicators to real values in $(-1,1)$. The solution of $\mathbf{q}$ for minimizing the Rayleigh quotient of Eq.(22) is given by $(D-W)\mathbf{q} = \lambda D\mathbf{q}$, which can be written as

$$W\mathbf{q} = \zeta D\mathbf{q},\ \zeta = 1-\lambda. \tag{23}$$

For convenience, we define $\mathbf{z} = D^{1/2}\mathbf{q}$, and write Eq.(24) as a standard eigenvalue problem:

$$\widehat{W}\mathbf{z} = (D^{-1/2}WD^{-1/2})\mathbf{z} = \zeta\mathbf{z}. \tag{24}$$

Finally, coming back to the bipartite graph, we have

$$D = \begin{pmatrix} D_r & 0 \\ 0 & D_c \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} D_r^{1/2}\mathbf{f} \\ D_c^{1/2}\mathbf{g} \end{pmatrix}. \tag{25}$$

Substituting into Eq.(24), we have

$$\begin{pmatrix} 0 & \widehat{B} \\ \widehat{B}^T & 0 \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \zeta\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}, \tag{26}$$

where

$$\widehat{B} = D_r^{-1/2}BD_c^{-1/2}. \tag{27}$$

The solutions to Eq.(26) are SVD of $\widehat{B}$ (that SVD is the solution to Eq.24 for bipartite graph is noted earlier[22, 3].) We emphasize that Eq.(26) is identical Eq.(8), with the correspondence relationship

$$B \Rightarrow \widehat{B}, \quad \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \Rightarrow \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}. \tag{28}$$

(see also the similarity between Eq.(23) and Eq.(7).) Therefore, the net effect of MinMaxCut of Eq.(17) over the simple MinCut objective Eq.(6) or Eq.(16) is the scaling of the association matrix B in Eq.(27). However, with this scaling, the self-aggregation property emerges.

4 Self-Aggregation Networks

Just as the Hopfield networks is the solution to Mincut objective, we propose the self-aggregation networks as the K-way clustering solution to MinMax Cut.

We introduce nonlinear scaling factors, diagonal matrices D_r (each element is the sum of a row, see Eq.25) and D_c (each element is the sum of a column). Let $B = D_r^{1/2}\widehat{B}D_c^{1/2}$, where $\widehat{B}$ is defined in Eq.(27). Applying SVD on $\widehat{B}$, we obtain

$$B = D_r^{1/2}(\sum_k^m \mathbf{u}_k\zeta_k\mathbf{v}_k^T)D_c^{1/2} = D_r\sum_k^m \mathbf{f}_k\zeta_k\mathbf{g}_k^T D_c. \tag{29}$$

We call $\mathbf{f}_k = D_r^{-1/2}\mathbf{u}_k$ and $\mathbf{g}_k = D_c^{-1/2}\mathbf{v}_k$ *scaled* PCA components. In data clustering perspective, they are just the *relaxed* clustering indicators, see Eq.(25). (We note that there are a number of different approaches for nonlinear PCA [7, 13, 15, 16].)

In Hopfield networks, a pattern $\mathbf{f}_1$ is encoded into the objective function as $\mathbf{f}_1\mathbf{f}_1^T$ (the Hebb rule); multiple patterns are encoded additively: $\mathbf{f}_1\mathbf{f}_1^T + \cdots + \mathbf{f}_k\mathbf{f}_k^T$. In our problem, a pattern is a cluster partitioning indicator vector. Let $F_K = (\mathbf{f}_1, \cdots, \mathbf{f}_K)$, and $G_K = (\mathbf{g}_1, \cdots, \mathbf{g}_K)$, and

$$Q_K = (\mathbf{q}_1, \cdots, \mathbf{q}_K) = \begin{bmatrix} F_K \\ G_K \end{bmatrix}. \tag{30}$$

We call $Q_K Q_K^T = \sum_{k=1}^K \mathbf{q}_k\mathbf{q}_k^T$ the generalized self-aggregation (SA) network. From the relation,

$$Q_K Q_K^T = \begin{bmatrix} F_K F_K^T & F_K G_K^T \\ G_K F_K^T & G_K G_K^T \end{bmatrix}. \tag{31}$$

we see that $F_K F_K^T = \sum_{k=1}^K \mathbf{f}_k\mathbf{f}_k^T$ is the SA network for row objects, $G_K G_K^T = \sum_{k=1}^K \mathbf{g}_k\mathbf{g}_k^T$ is the SA network for column objects, and $F_K G_K^T = \sum_{k=1}^K \mathbf{f}_k\mathbf{g}_k^T$ is the SA network for row-column associations,

The SA networks defined above share an important feature: *cluster self-aggregation*. Using neural networks language, we call $(F_K G_K^T)_{ij}$ the connection (association) between nodes i, j. Self-aggregation amounts to an connection weight updating process where connections between different clusters are automatically suppressed while connections within same clusters are enhanced.

In the following we provide a theoretical analysis and prove this fundamental property for SA networks. The development follows a perturbation analysis framework[4, 14, 6] by decomposing $\widehat{W}$ in Eq.(24) as

$$\widehat{W} = \widehat{W}^{(0)} + \widehat{W}^{(1)}$$

where $\widehat{W}^{(0)}$ corresponds to the case where no overlap (connection) exists between different clusters and $\widehat{W}^{(1)}$ corresponds to the case where small overlaps exist between different clusters.

4.1 Well separated clusters

In this case, the connections between two clusters (edges cross the cut line in Fig.1) do not exist. In the association matrix, this is reflected by $B_{R_p,C_q} = 0, p \neq q$ [see Eq.(4)]. We have

Theorem 2. When overlaps among K clusters are zero, the K scaled PCA components $\mathbf{q}_1, \cdots, \mathbf{q}_K$ get the same maximum eigenvalue: $\zeta_k = 1, k = 1, \cdots K$. Each $\mathbf{q}_k$ is a multistep (piecewise-constant) function (assuming objects within a cluster are indexed consecutively). In the scaled PCA subspaces, objects within the same cluster self-aggregate into a single point. □

The proof is a few algebraic manipulations. For simplicity, we illustrate the proof by providing a concrete $K = 3$ example. The solutions to Eq.(24) are

$$\mathbf{x}^{(1)} = \frac{1}{\sqrt{2s_{11}}} \begin{bmatrix} D_{r11}^{1/2}\mathbf{e}_{r1} \\ 0 \\ 0 \\ D_{c11}^{1/2}\mathbf{e}_{c1} \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{x}^{(2)} = \frac{1}{\sqrt{2s_{22}}} \begin{bmatrix} 0 \\ D_{r22}^{1/2}\mathbf{e}_{r2} \\ 0 \\ 0 \\ D_{c22}^{1/2}\mathbf{e}_{c2} \\ 0 \end{bmatrix}$$

etc. Here $D_{rpq} = \mathrm{diag}(B_{pq}\mathbf{e}_{rq})$, $(p, q = 1, \cdots, K)$, $\mathbf{e}_{rq} = \mathbf{e}$ with the size of p-th row block; $D_{cpq} = \mathrm{diag}(B_{pq}\mathbf{e}_{cq})$, $\mathbf{e}_{cq} = \mathbf{e}$ with the size of p-th column block; and $s_{pq} = s(B_{R_p,C_q})$. Note that $s_{pq} \neq s_{qp}$. Let

$$X_K = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(K)}). \tag{32}$$

For any K-dim vector $\mathbf{y} = (y(1), \cdots, y(K))^T$,

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} = \mathbf{q} = D^{-1/2} X_K \mathbf{y} = \begin{bmatrix} y(1)\ \mathbf{e}_{r1}/(2s_{11})^{1/2} \\ \vdots \\ y(K)\ \mathbf{e}_{rK}/(2s_{KK})^{1/2} \\ y(1)\ \mathbf{e}_{c1}/(2s_{11})^{1/2} \\ \vdots \\ y(K)\ \mathbf{e}_{cK}/(2s_{KK})^{1/2} \end{bmatrix} \tag{33}$$

is an eigenvector of Eq.(23). Now any K orthonormal $\{\mathbf{y}_1, \cdots, \mathbf{y}_K\}$ leads to K eigenvectors $\{\mathbf{q}_1, \cdots, \mathbf{q}_K\} \equiv Q_K$.

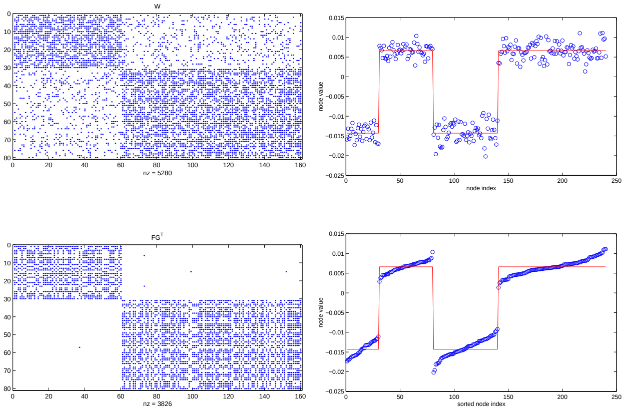

Figure 2: Left-top: adjacency matrix of a bipartite graph of 2 dense clusters (diagonal blocks) with random overlaps (off-diagonal blocks). Left-bottom: $F_K G_K^T$. The overlaps are reduced significately due to self-aggregation. Right: computed $\mathbf{q}_2$ (cycles) and the approximation from Theorem 3 (solid line), in original index order (top panel) and in sorted index order (bottom panel).

In the space spanned by F_K the coordinate of data object i is $\mathbf{r}_i = (\mathbf{f}_1(i), \cdots, \mathbf{f}_K(i))^T$; From Eq.33, data objects within a cluster self-aggregate to (are located at) the same point. Furthermore, $F_K F_K^T$ gives the clusters for row objects, the word clusters (see Fig.3):

$$F_K F_K^T = \begin{bmatrix} \mathbf{e}_{r1}\mathbf{e}_{r1}^T/2s_{11} & 0 & 0 \\ 0 & \mathbf{e}_{r2}\mathbf{e}_{r2}^T/2s_{22} & 0 \\ 0 & 0 & \mathbf{e}_{r3}\mathbf{e}_{r3}^T/2s_{33} \end{bmatrix}. \tag{34}$$

In the space spanned by G_K the coordinate of data object i is $\mathbf{r}_i = (\mathbf{g}_1(i), \cdots, \mathbf{g}_K(i))^T$; once again, data objects within a cluster are self-aggregate to the same point. Furthermore, $G_K G_K^T$ gives the clusters for column objects, i.e, the document clusters (see Fig.3):

$$G_K G_K^T = \begin{bmatrix} \mathbf{e}_{c1}\mathbf{e}_{c1}^T/2s_{11} & 0 & 0 \\ 0 & \mathbf{e}_{c2}\mathbf{e}_{c2}^T/2s_{22} & 0 \\ 0 & 0 & \mathbf{e}_{c3}\mathbf{e}_{c3}^T/2s_{33} \end{bmatrix} \tag{35}$$

In both SA networks $F_K F_K^T, G_K G_K^T$, the overlap connections are identically zero as expected. However, connections within same clusters are enhanced significantly: every pair of two objects i, j within a cluster acquires the same connection strength even if objects i, j may not be connected in the original association matrix B.

SA network $F_K G_K^T$ gives the association between row objects and column objects. The self-aggregation gives the sharpened row-column associations (see Figs.2).

$$F_K G_K^T = \begin{bmatrix} \mathbf{e}_{r1}\mathbf{e}_{c1}^T/2s_{11} & 0 & 0 \\ 0 & \mathbf{e}_{r2}\mathbf{e}_{c2}^T/2s_{22} & 0 \\ 0 & 0 & \mathbf{e}_{r3}\mathbf{e}_{c3}^T/2s_{33} \end{bmatrix} \tag{36}$$

This is useful for document retrieval (see §5.1).

4.2 Overlapping Clusters

In clustering, the useful case is that clusters overlap. Here we assume that the overlaps are small and provide a perturbation analysis. We have the following results:
Theorem 3. At the first order, the solutions to Eq.(24) are the following: the highest K eigenvectors have the form

$$\mathbf{q} = D^{-1/2} X_K \mathbf{y},$$

where X_K is given in Eq.(32) and $\mathbf{y}$ and the eigenvalue λ $(\zeta = 1 - \lambda)$ satisfy the eigensystem

$$\Gamma \mathbf{y} = \lambda \mathbf{y}. \tag{37}$$

Γ has the form $\Gamma = \Omega^{-1/2}\ \bar{\Gamma}\ \Omega^{-1/2}$, where

$$\bar{\Gamma} = \begin{bmatrix} h_{11} & -s_{12} - s_{21} & \cdots & -s_{1K} - s_{K1} \\ -s_{21} - s_{12} & h_{22} & \cdots & -s_{2K} - s_{K2} \\ \vdots & \vdots & \cdots & \vdots \\ -s_{K1} - s_{1K} & -s_{K2} - s_{2K} & \cdots & h_{KK} \end{bmatrix} \tag{38}$$

where $\quad h_{kk} = \sum_{p \neq k}(s_{kp} + s_{pk})$
and $\quad \Omega = \text{diag}(2s_{11}, 2s_{22}, \cdots, 2s_{KK})$. □

The proof is bit involved and will be omitted here. This theorem captures several important features of SA networks, which are embedded in the solution to Eq.(37). Note that the $K \times K$ matrix Γ is symmetric semi-positive definite. Here we list two corollaries:

Corollary 2.1. For $K = 2$, the second lowest eigenvalue of Γ is

$$\lambda_2 = (s_{12} + s_{21})/2s_{11} + (s_{12} + s_{21})/2s_{22},$$

which is precisely the min-max cut clustering objective J_{MMC} in Eq.(17). Therefore, the smaller λ_2, the better quality of the resulting clusters. The corresponding eigenvector is

$$\mathbf{q}_2 = D^{-1/2} X_2 \mathbf{y}_2 = \sqrt{\frac{s_{22}}{2s_{11}}} \begin{bmatrix} \mathbf{e}_{r1} \\ 0 \\ \mathbf{e}_{c1} \\ 0 \end{bmatrix} - \sqrt{\frac{s_{11}}{2s_{22}}} \begin{bmatrix} 0 \\ \mathbf{e}_{r2} \\ 0 \\ \mathbf{e}_{c2} \end{bmatrix}.$$

Thus we automatically recover the partitioning indicators. All these indicate SA networks is a highly consistent and principled framework for clustering. The lowest eigenvector is $\mathbf{q}_1 = (1, \cdots, 1)$. $Q_2 = (\mathbf{q}_1, \mathbf{q}_2)$ constructed from these two eigenvectors have the forms given in Eqs.(4,4,4).

Corollary 2.2. The K eigenvectors $Y_K = (\mathbf{y}_1, \cdots, \mathbf{y}_K)$ of Γ satisfy $Y_K^T Y_K = I_K$. The square orthonormal matrix Y_K is full rank under general conditions, thus $Y_K Y_K^T = I_K$. Using $Q_K = D^{-1/2} X_K Y_K$. and constructing the SA networks, $F_K F_K^T$, $G_K G_K^T$ and $F_K G_K^T$, they will have the same block diagonal structures of Eqs.(4,4,4).

Corollary 2.2 provides the theoretical basis for using $F_K F_K^T$ and $G_K G_K^T$ for clustering, and $F_K G_K^T$ for improving retrieval.

Example. We apply the above analysis to a bipartite graph example with association matrix shown in Fig.2. The bipartite graph has two dense clusters with large overlaps between them. The indicator vector $\mathbf{q}_2$ computed directly from Eq.(24) together with that from Theorem 3 are also shown in Fig.2. They agree reasonably. The eigenvalue values from Eq.(24) and Theorem 3 also agree reasonably well: $\lambda_2 = 0.456$, $\tilde{\lambda}_2 = 0.477$. $F_K G_K^T$ gives a sharpened association matrix (Fig.2) where the overlap between the two clusters are greatly reduced. $F_K F_K^T$ and $G_K G_K^T$ computed from Eq.(24) are shown in Fig.3. They are close to the analysis results (Corollary 2.2). $F_K F_K^T$ gives clusters for row objects (words) and $G_K G_K^T$ gives clusters for column objects (documents).

In self-aggregation, data objects move towards each other guided by connectivity, as connection weights between different clusters are suppressed and connections within same clusters are enhanced. This is similar to the self-organizing map [12], where feature vectors self-organize into a 2D feature map while data objects remain fixed. In Hopfield network, features are stored (encoded) as associative memories, whereas in SA networks, connection weights are dynamically adjusted to learn the patterns in an unsupervised way.

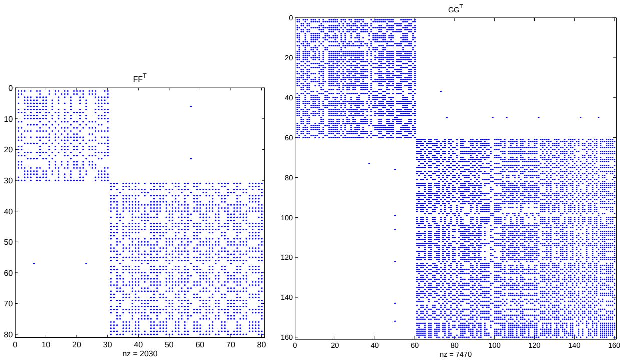

Figure 3: Left: $F_K F_K^T$ for clustering row objects. Right: $G_K G_K^T$ for clustering column objects.

5 Applications of SA networks

5.1 Document Retrieval

We first apply SA networks to document retrieval. That clustering can help retrieval is suggested by the Clustering Hypothesis [17]: if a document $\mathbf{x}_i$ is highly relevant to a query $\mathbf{q}$, then documents very similar to $\mathbf{x}_i$ (defined by cosine similarity) are likely to be relevant to the query as well. In many previous work, documents are first clustered and query is then matched to the cluster centroids[19]. However, the experimental results so far indicates clustering had not helped the retrieval precision [19, 8]. (A recent different usage is to cluster the retrieved documents to group them into different topics[8].)

SA networks presents a new approach to use clustering for retrieval. Here the cluster structure is embedded in $F_K G_K^T$ which is very similar to the original word-to-document matrix. We truncate the expansion in Eq.(29) at K and set $\zeta_k = 1$,

$$B \simeq D_r \sum_{k=1}^{K} \mathbf{f}_k \mathbf{g}_k^T D_c = D_r F_K G_K^T D_c = (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_n), \quad (39)$$

the jth column $\tilde{\mathbf{x}}_j$ is the representation of SA network for document j. The relevance r_j of document $\tilde{\mathbf{x}}_j$ for query $\mathbf{q}$ through the cluster structure is simply $r_j = \cos(\mathbf{q}, \tilde{\mathbf{x}}_j)$. If the clusters are well separated, all documents within a cluster will have *same* relevance to a query (see Eq.(36) and Fig.2 left-bottom panel), and thus all documents of the most relevant cluster will be retrieved, even though their original vector-space representations ($\{\mathbf{x}_j\}$, columns in B) could differ considerably. The self-aggregation makes this possible. In practice, overlaps exist; documents most similar to each other will have similar $\tilde{\mathbf{x}}_j$ and will get very similar relevance score using the cosine similarity metric. Therefore, Eq.(39) is a convenient and natural way to incorporate clustering information into retrieval.

We define the total relevance as the combination of the keywords matching (KM) and SA network matching:

$$r_j = \cos(\mathbf{q}, \mathbf{x}_j) + \alpha \cos(\mathbf{q}, \tilde{\mathbf{x}}_j) \qquad (40)$$

We call this self-aggregation improved keywords matching (SAI-KM). In all experiments below, $\alpha = 0.5$

We apply this retrieval method to 4 standard IR test datatsets: Medline (1033 docs, 30 queries), Cranfield (1400 docs, 225 queries), CACM (3204 docs, 64 queries) and NPL (11429 docs, 93 queries) collections.

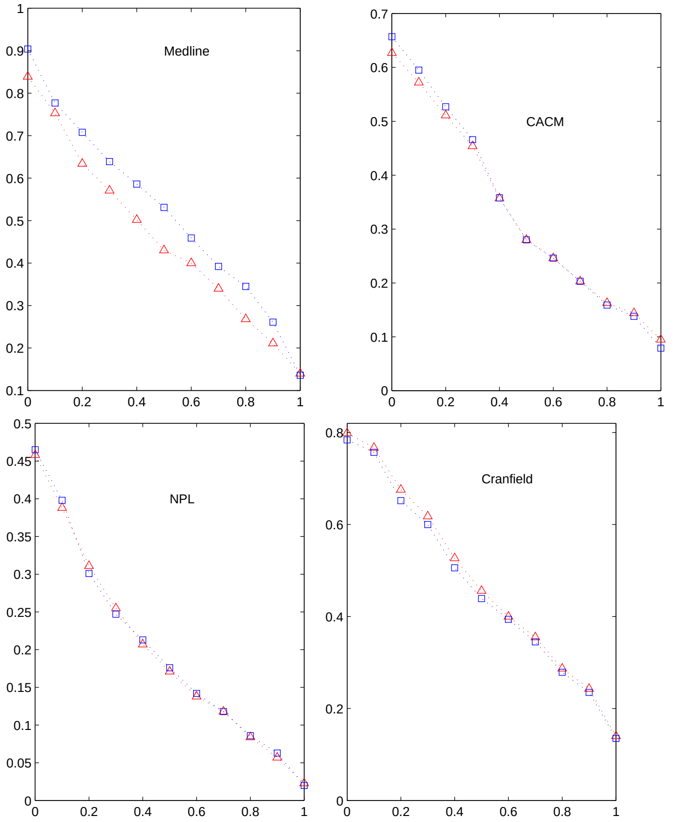

Figure 4: Precision-recall curves for Medline, CACM, NPL and Cranfield collections. △ for keywords matching, □ for SA-improved keywords matching.

Precision-recall curves for Medline, CACM, NPL and Cranfield collections are shown in Fig.4. The average precisions are summarized in Table 1. Here we use `tf.idf` term weighting. K=10 for Medline, K=20 for all others. On Medline, SAI-KM clearly improves the retrieval precisions at all recall levels. It is interesting to note that LSI also performs well on Medline. The advantage of SA network is that we only store $2K$ vectors F_K, G_K, whereas LSI typically use $K = 200$, about 20 times more storage.

For CACM and NPL, SAI-KM improves precision at low recall levels (0-10%). We note that retrieval precision at low recall are important because in practice user usually check the few top returned documents only.

For Cranfield, SAI-KM performs slightly worse than standard keywords matching. We note that clustering hypothesis were first experimented on this collection and the results are generally inferior to keywords matching [18, 19]. By examining the SA networks for Cranfield, the cluster structure is not detectable, i.e, this collection does not have clear sub-structures.

In summary, comparing to standard keywords matching, SA network improved retrieval achieves substantially better retrieval precision for Medline, improves slightly at low recall for CACM and NPL, and performs slightly worse for Cranfield. This represents a significant progress from earlier work summarized in [19].

	Med	CACM	NPL	Cran
KM	0.463	0.331	0.201	0.478
SAI-KM	0.522	0.337	0.203	0.467

Table 1: Average 11-point retrieval precision.

5.2 Document Clustering

We apply SA networks clustering method on newsgroup articles in 5 newsgroups (see Fig.5). 100 news articles are randomly selected from each newsgroup. 1000 words are selected based on mutual information. The term-document association matrix B are solved by SVD. The results are shown in Fig.5. Here we emphasize the fact that words aggregate into clusters in the K-dim space F_K (see Eq.36) while documents are simultaneous clustered using $G_K G_K^T$. The clustering accuracy $[\sum_k t_{kk}/N, T = (t_{ij})$ is the contingency table] of the clustering results is 86%. In comparison, the standard K-means methods has a clustering accuracy of 66%, while two improved K-means methods achieves 76-80% [21].

6 Summary

We present a document clustering framework connecting PCA, Kmeans with Hopfield networks. The min-max cut clustering objective inforces cluster balance and leads to scaled PCA. Networks constructed with scaled PCA components via Hebb rule has the unique and desirable self-aggregation property. SA networks improves document retrieval and provides an effective multi-K clustering algorithm, as shown by a number of experiments.

Acknowledgements. This work is supported by Department of Energy (Office of Science) under contract DE-AC03-76SF00098 through a LDRD grant. LBNL-51339, August 2002,

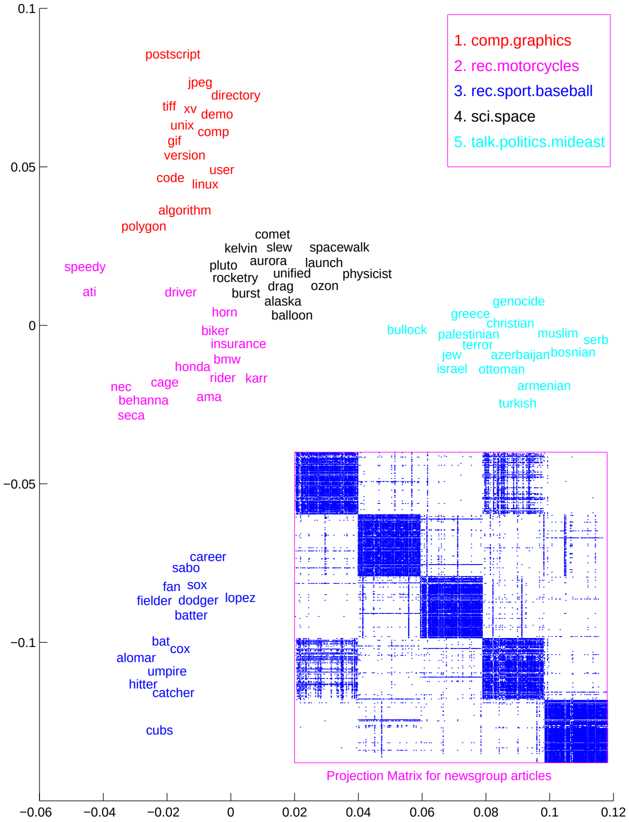

Figure 5: Word aggregation in F_K space while news articles from 5 newsgroups are simultaneously clustered using SA network $G_K G_K^T$ shown in the insert. Several words in *motorcycles* are brand names, and several words in *baseball* are players' names.

References

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Info. Sci*, 41:391–407, 1990.

[3] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *Proc. ACM Int'l Conf Knowledge Disc. Data Mining (KDD 2001)*.

[4] C. Ding, X. He, and H. Zha. A spectral method to separate disconnected and nearly-disconnected web graph components. In *Proc. ACM Int'l Conf Knowledge Disc. Data Mining (KDD)*, pages 275–280, August 2001.

[5] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. *Proc. IEEE Int'l Conf. Data Mining*, pages 107–114, 2001.

[6] C. Ding, X. He, H. Zha, and H. Simon. Unsupervised learning: self-aggregation in scaled principal component space. *Proc. 6th European Conf. Principles of Data Mining and Knowledge Discovery (PDKK 2002)*, pp.112-124.

[7] T. Hastie and W. Stuetzle. Principal curves. *J. Amer. Stat. Assoc*, 84:502–516, 1989.

[8] M. A. Hearst and J. O. Paderson. Re-examining the cluster hypothesis: Scatter/gather on retrieval results. *Proc. SIGIR'96*, 1996.

[9] J. Hertz, R.G. Palmer, and A. Krogh. *Introduction to the Theory of Neural Computation*. Perseus Publishing, 1991.

[10] J.J. Hopfield. Neural networks and physical systems with emergent collective computation abilities. *Proc. Nat'l Acad Sci USA*, 79:2554–2558, 1982.

[11] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.

[12] T. Kohonen. *Self-organization and Associative Memory*. Springer-Verlag, 1989.

[13] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[14] J. Mathews and R.L. Walker. *Mathematical Methods of Physics*. Addison-Wesley, 1971.

[15] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[16] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[17] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[18] E.M. Voorhees. The cluster hypothesis revisited. *Proc. SIGIR 1985*, 1985.

[19] P. Willett. Recent trends in hierarchical document clustering. *Information Processing and Management*, 24, 1988.

[20] Y. Yang. An evaluation of statistical approaches to text categorization. *J. Information Retrieval*, 1:67–88, 1999.

[21] H. Zha, C. Ding, M. Gu, X. He, and H.D. Simon. Spectral relaxation for k-means clustering. *Proc. Neural Info. Processing Systems (NIPS 2001)*, Dec. 2001.

[22] H. Zha, X. He, C. Ding, M. Gu, and H.D. Simon. Bipartite graph partitioning and data clustering. *Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM 2001)*, pages 25–31, 2001.