Data Mining and Applied Linear Algebra

Moody Chu*
Department of Mathematics
North Carolina State University
Raleigh, North Carolina 27695-8205, USA
chu@math.ncsu.edu

Abstract

In this era of hyper-technological innovation, massive amounts of data are being generated at almost every level of applications in almost every area of disciplines. Extracting interesting knowledge from raw data, or data mining in a broader sense, has become an indispensable task. Nevertheless, data collected from complex phenomena represent often the integrated result of several interrelated variables, whereas these variables are less precisely defined. The basic principle of data mining is to distinguish which variable is related to which and how the variables are related. In many situations, the digitized information is gathered and stored as a data matrix. It is often the case, or so assumed, that the exogenous variables depend on the endogenous variables in a linear relationship. Retrieving "useful" information therefore can often be characterized as finding "suitable" matrix factorization. This paper offers a synopsis from this prospect on how linear algebra techniques can help to carry out the task of data mining. Examples from factor analysis, cluster analysis, latent semantic indexing and link analysis are used to demonstrate how matrix factorization helps to uncover hidden connection and do things fast. Low rank matrix approximation plays a fundamental role in cleaning the data and compressing the data. Other types of constraints, such as nonnegativity, will also be briefly discussed.

1 Introduction

Data analysis is pervasive throughout science, engineering and business applications. An essential task in almost every discipline is to analyze a certain data to search for relationships between a set of exogenous and endogenous variables. There are two special concerns in data analysis.

First, most of the information gathering devices or methods at present have only finite bandwidth. One thus cannot avoid the fact that the data collected often are not exact. For example, signals received by antenna arrays are often contaminated by instrumental noises; astronomical images acquired by telescopes are often blurred by atmospheric turbulence; database prepared by document indexing are often biased by subjective judgment; and even empirical data obtained in laboratories often do not satisfy intrinsic physical constraints. Before any deductive sciences can further be applied, it is important to first reconstruct or represent the data so that the inexactness is reduced while certain feasibility conditions are satisfied.

Secondly, complex systems always entail multiple variables. Data observed for these systems are the convoluted action of these variables. When these variables are less precisely defined, the actual information contained in the original data might be overlapping and ambiguous. A reduced system model could provide a fidelity near the level of the original system, while facilitating the extraction of previously hidden knowledge for important decision making.

Among a wide variety of data mining techniques, classification, regression, factor analysis and principal component analysis are some of the most commonly employed methods for accomplishing the goal of reducing the number of variables and detecting structures among the variables. One common ground in the various approaches for noise removal, model reduction, feasibility reconstruction, and so on, is to replace the original data by a lower dimensional representation obtained via subspace approximation. The notion of low rank approximations therefore arises. This paper casts these data mining techniques as a problem of matrix factorization and suggests fundamentals of models and methods from linear algebra prospective.

2 Linear Model

Let $Y = [y_{ij}] \in \mathbb{R}^{n\times\ell}$ denote the matrix of the "observed" data, which is to be analyzed. Each entry y_{ij} rep-

*This research was supported in part by the National Science Foundation under grants DMS-0505880 and CCF-0732299.

resents, in a broad sense, the *score* obtained by entity j on variable i. One way to characterize the interrelationships among multiple variables that contribute to the observed data Y is to assume that y_{ij} is a linearly weighted score by entity j over several "factors". We shall temporarily assume that there are m factors, but often it is precisely the point that the factors are to be retrieved in the mining process. A linear model, therefore, assumes the relationship

$$Y = AF, \tag{1}$$

where $A = [a_{ik}] \in \mathbb{R}^{n\times m}$ is the loading matrix with a_{ik} denoting the *influence* of factor k on variable i, and $F = [f_{kj}] \in \mathbb{R}^{m\times \ell}$ is the scoring matrix with f_{kj} denoting the *response* of entity j to factor k. Depending on the applications, there are many ways to interpret the meaning of the linear model.

A brief description of some applications below should demonstrate the points. Pay attention to the fact that the data are subject some additional constraints.

2.1 Air Quality Management

The receptor model is an observational technique commonly employed by the air pollution research community which makes use of the ambient data and source profile data to apportion sources or source categories [6]. Assume that there are m sources which contribute n chemical species to ℓ samples. The *mass balance equation* within this system can be expressed via the relationship,

$$y_{ij} = \sum_{k=1}^{m} a_{ik} f_{kj}, \tag{2}$$

where y_{ij} is the elemental concentration of the ith chemical measured in the jth sample, a_{ik} is the gravimetric concentration of the ith chemical in the kth source, and f_{kj} is the airborne mass concentration that the kth source has contributed to the jth sample. In a typical scenario, only values of y_{ij} are observable whereas neither the sources are known nor the compositions of the local particulate emissions are measured. Thus, a critical question is to estimate the number m, the compositions a_{ik}, and the contributions f_{kj} of the sources.

Note that in this receptor model, there is a physical constraint imposed upon the data. That is, the source compositions a_{ik} and the source contributions f_{kj} must all be nonnegative. The identification and apportionment, therefore, becomes a nonnegative matrix factorization problem of Y.

2.2 Image Articulation Library

In biometric identification applications, it is sometimes desirable to process data sets of images represented by column vectors as composite objects in many articulations, poses, or separated parts. The factorization in the linear model would enable the identification and classification of intrinsic "parts" that make up the object being imaged by multiple observations [5, 11]. More specifically, each column $\mathbf{y}_j$ of a nonnegative matrix Y now represents n pixel values of one image. The columns $\mathbf{a}_k$ of A are basis elements in $\mathbb{R}^n$. The columns of F, belonging to $\mathbb{R}^m$, can be thought of as coefficient sequences representing the ℓ images in the m basis elements. In other words, the relationship

$$\mathbf{y}_j = \sum_{k=1}^{m} \mathbf{a}_k f_{kj} \tag{3}$$

can be thought of as that there are m *standard parts* $\mathbf{a}_k$ in a variety of positions and that each image represented as a vector $\mathbf{y}_j$ is made by superposing these parts together in specific ways by a mixing matrix represented by F. Those parts, being images themselves, are necessarily nonnegative. The superposition coefficients, each part being present or absent, are also necessarily nonnegative.

2.3 Latent Semantic Indexing

Assume that textual documents are collected in an *indexing matrix* $H = [h_{ik}]$ in $\mathbb{R}^{n\times m}$. Each document is represented by one row in H. The entry h_{ik} represents the *weight* of one particular *term* k in document i whereas each term could be defined by just one single word or a string of phrases. One commonly used term-weighting scheme to enhance discrimination between various documents and to enhance retrieval effectiveness is to define $h_{ik} = t_{ik} g_k n_i$, where t_{ik} captures the relative importance of term k in document i, g_k weights the overall importance of term k in the entire set of documents, and $n_i := \left(\sum_{k=1}^{m} t_{ik} g_k\right)^{-1/2}$ is the scaling factor for normalization, which is necessary because, otherwise, one could artificially inflate the prominence of document i by padding it with repeated pages or volumes. Note that after the normalization, rows of H are of unit length.

With the indexing matrix H in place, one can retrieve information for a given query. Each query is represented as a column vector $\mathbf{q}_j = [q_{1j}, \ldots, q_{mj}]^\top$ in $\mathbb{R}^m$ where q_{kj} represents the weight of term k in the query j. To measure how the query $\mathbf{q}_j$ matches the documents, we calculate the column vector

$$\mathbf{c}_j = H\mathbf{q}_j \tag{4}$$

and rank the relevance of documents to $\mathbf{q_j}$ according to the *scores* in $\mathbf{c}_j$.

The computation thus far seems to be merely the vector-matrix multiplication. This is so only if H is a "reasonable" representation of the relationship between documents and terms. In practice, however, the matrix H is never exact.

A major challenge in the field has been to represent the indexing matrix and the queries in a more compact form so as to facilitate the computation of the scores [4, 13]. In this context, the standard parts $\mathbf{a}_k$ indicated in (3) may be interpreted as subcollections of some "general concepts" contained in these documents.

Note the different emphasis in these examples. In some cases the data matrix Y is given and the factors in A are to be retrieved while in other cases the terms in the indexing matrix H are predetermined and the scores $\mathbf{c}_j$ are to be calculated.

3 Matrix Factorization

The relationship of $Y = AF$ in the linear model (1) is only the first level matrix factorization. To effectuate this decomposition, our thought needs to go deeper.

3.1 Factor Retrieval

A common practice is to assume that the data in Y are standard scores, i.e., each row of Y has been normalized to have mean 0 and standard deviation 1. The matrix

$$R := \frac{1}{\ell} YY^\top, \tag{5}$$

therefore represents the correlation matrix of all n variables. Assume further that all sets of factors being considered are uncorrelated with each other and, similar to Y, that the scores in F for each factor are also normalized. Then it is true that $FF^\top = \ell I_m$, where I_m stands for the identity matrix in $\mathbb{R}^{m\times m}$. It follows that the correlation matrix R can be expressed directly in terms of the loading matrix A, i.e.,

$$R = AA^\top. \tag{6}$$

Factor extraction now becomes a problem of decomposing the correlation matrix R into the product $AA^\top$.

As a whole, the ith row of A may be interpreted as how the data variable i is weighted across the list of current factors. If the norm of this row, called the *communality* of variable i, is small, it suggests that this specific variable is of little consequence to the current list of factors. On the other hand, the kth column of A may be interpreted as correlations of the data variables with that particular kth factor. Those data variables with high factor loadings are considered to be more like the factor in some sense and those with zero or near-zero loadings are treated as being unlike the factor. The quality of this likelihood, called the *significance* of the corresponding factor, is measured by the norm of the kth column of A. One basic idea in factor analysis is to rewrite the loadings of variables over some newly selected factors so as to manifest more clearly the correlation between variables and factors.

Suppose the newly selected factors are expressed in terms of columns of the orthogonal matrix

$$V := [\mathbf{v}_1, \ldots, \mathbf{v}_m] \in \mathbb{R}^{m\times m}. \tag{7}$$

Then the rewriting of factor loadings with respect to V is mathematically equivalent to a change of basis, i.e., A is now written as $B := AV$. Selecting V so that the significance levels among the factors are more manifestly differentiated is the critical step in the practice of factor analysis. Ideally, we want to concentrate the loadings on as few factors as possible. In this way, we discover influential factors and reduce the dimension of the model.

Note that because $VV^\top = I_m$, the very same observed data now is decomposed as $Y = AF = (AV)(V^\top F) = BG$ with $B = AV$ and $G = V^\top F$ representing, respectively, the factor loadings and uncorrelated standard factor scores corresponding to the factors in V. From this we also see that the correlation matrix $R = AA^\top = BB^\top \in \mathbb{R}^{n\times n}$ is independent of factors selected. This identity is important in that the new factors can be retrieved directly from the correlation matrix R without reference to any previously defined loading matrix A.

3.2 Centroid Decomposition

The centroid decomposition amounts to a procedure of defining a new coordinate system representing what are called the *centroid factors* via successive rank reduction. The most important feature of this approach is that loadings with respect to the centroid factors can be calculated without the knowledge of the data matrix Y. The computation only depends on the correlation matrix R. The centroid decomposition, motivated by the simplicity of its geometry, had been used as a convenient way to retrieve factors by hand calculation before the most powerful singular value decomposition was developed.

We demonstrate how the first centroid factor can be calculated. Temporarily assuming that a loading matrix $A_1 \in \mathbb{R}^{n\times m}$ is given, the coordinate axes in $\mathbb{R}^m$ represent a set of m abstractly defined factors. Denoting each row of A_1 as a point in the factor space $\mathbb{R}^m$, the arithmetic mean of these points could be an indicator for the collective trend of the variables and thus constitutes the essential idea of a centroid factor. The centroid of these n variables is given trivially by the column vector

$$\mathbf{c}_1 := \frac{A_1^\top \mathbf{1}_n}{n} = \left[\frac{\sum_{i=1}^n a_{i1}}{n}, \ldots, \frac{\sum_{i=1}^n a_{im}}{n}\right]^\top, \tag{8}$$

where $\mathbf{1}_n$ denote the column vector $\mathbf{1}_n := [1, \ldots, 1]^\top \in \mathbb{R}^n$. The first *centroid factor* is defined to be the normalized vector

$$\mathbf{v}_1 := \frac{\mathbf{c}_1}{\|\mathbf{c}_1\|}. \tag{9}$$

The new loadings of variables with respect to this new factor $\mathbf{v}_1$, i.e., the first column $\mathbf{b}_1$ of the new loading matrix B (which is yet to be found), is given by $\mathbf{b}_1 = A_1\mathbf{v}_1$ which can be rewritten as

$$\mathbf{b}_1 = A_1 \frac{A_1^\top \mathbf{1}_n}{\|A_1^\top \mathbf{1}_n\|} = \frac{R_1\mathbf{1}_n}{\sqrt{\mathbf{1}_n^\top R_1 \mathbf{1}_n}}. \tag{10}$$

Note that the first loading vector $\mathbf{b}_1$ is extracted directly from R_1. No reference to A_1 or $\mathbf{v}_1$ is needed.

Once the first centroid factor is found, the system can easily be orthogonally reduced to remove any information along $\mathbf{v}_1$ from A_1 (or R_1). The above procedure can then be repeated to find the next generalized centroid. Each application of this centroid factor retrieval will reduce the rank of the loading matrix by one [2]. The procedure therefore has to come to a stop in finitely many steps. In this way, with the recurrence

$$A_i = A_{i-1} - A_{i-1}\mathbf{v}_{i-1}\mathbf{v}_{i-1}^\top, \quad i = 2, \ldots, r, \tag{11}$$

where $\mathbf{v}_i$ is the generalized centroid factor of A_i, r is the rank of A_1, and with the loadings $\mathbf{b}_i = A_i\mathbf{v}_i$, we may write

$$A = A_1 = \mathbf{b}_r\mathbf{v}_r^\top + \ldots + \mathbf{b}_1\mathbf{v}_1^\top, \tag{12}$$

which is called a *centroid decomposition* of A. In practice, the iteration terminates whenever the significance level $\|\mathbf{b}_i\|$ is downgraded to a preset threshold, resulting a low rank approximation to A and, hence, to Y.

3.3 Singular Value Decomposition

It is worthwhile to point out first the statistical meaning behind the singular value decomposition. Consider a general random column vector $\mathcal{X}$ in $\mathbb{R}^n$ with a certain unspecified (or unknown) distribution. Let $\mathcal{E}[\mathcal{X}]$ denote the expected value of $\mathcal{X}$ and $\mathrm{cov}(\mathcal{X}) := \mathcal{E}[(\mathcal{X} - \mathcal{E}[\mathcal{X}])(\mathcal{X} - \mathcal{E}[\mathcal{X}])^\top] \in R^{n\times n}$ the *covariance matrix* of $\mathcal{X}$. The deterministic matrix $\mathrm{cov}(\mathcal{X})$ enjoys a spectral decomposition

$$\mathrm{cov}(\mathcal{X}) = \sum_{j=1}^{n} \lambda_j \mathbf{p}_j\mathbf{p}_j^\top, \tag{13}$$

where we also assume that eigenvalues are arranged in the descending order $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Because $\mathbf{p}_1, \ldots, \mathbf{p}_n$ form an orthonormal basis for R^n, we can express the random column variable $\mathcal{X}$ as

$$\mathcal{X} = \sum_{j=1}^{n} (\mathbf{p}_j^\top \mathcal{X})\mathbf{p}_j. \tag{14}$$

Note that columns in the matrix $P := [\mathbf{p}_1, \ldots, \mathbf{p}_n]$ are deterministic vectors themselves, the randomness of $\mathcal{X}$ therefore must come solely from the randomness of the combinations coefficients in (14). Let $\boldsymbol{\alpha} := P^\top\mathcal{X}$. Then $\boldsymbol{\alpha}$ is a random vector with moments,

$$\mathcal{E}[\boldsymbol{\alpha}] = P^\top\mathcal{E}[\mathcal{X}], \tag{15}$$

$$\mathrm{var}(\boldsymbol{\alpha}) = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}, \tag{16}$$

showing that its components are mutually stochastically independent. Now that our random vector $\mathcal{X}$ is comprised of random contributions from each of the n directions $\mathbf{p}_j$, $j = 1, \ldots, n$, whereas the contribution from each direction is governed independently by the distribution of the corresponding random variable α_j, it is intuitively correct from a statistical point of view that those coefficients α_j with larger variances should represent a more integral part in the stochastic nature of $\mathcal{X}$. It is in this context that we may *rank* the importance of corresponding eigenvectors $\mathbf{p}_j$ as *essential* components of the variable $\mathcal{X}$ according to the magnitude of λ_j.

If it becomes desirable to approximate the random variable $\mathcal{X}$ by another unbiased yet *simpler* variable $\hat{\mathcal{X}}$, we see that $\hat{\mathcal{X}}$ had better capture those components corresponding to larger λ_j in the expression (14). More specifically, by a simpler variable $\tilde{\mathcal{X}}$ we mean a random variable limited to a *lower dimensional* subspace. Our goal then is to find a proper subspace $\mathcal{S}$ of $\mathbb{R}^n$ and a particular random vector $\tilde{\mathcal{X}}$ over $\mathcal{S}$ such that $\mathcal{E}[\|\mathcal{X} - \tilde{\mathcal{X}}\|^2]$ is minimized.

To quantize $\tilde{\mathcal{X}}$, observe that given any m-dimensional subspace $\mathcal{S}$, there exists a matrix $K \in \mathbb{R}^{n\times m}$ such that columns of the matrix product PK, with P given by (13), form a basis for $\mathcal{S}$. Any unbiased random variable $\tilde{\mathcal{X}}$ restricted to $\mathcal{S}$ can then be expressed in the form

$$\tilde{\mathcal{X}} = PK\boldsymbol{\beta}$$

where $\boldsymbol{\beta}$ stands for a certain (column) random variable in $\mathbb{R}^m$. We may further assume that components in $\boldsymbol{\beta}$ are mutually independent. It follows that $\mathcal{E}[\|\mathcal{X} - \tilde{\mathcal{X}}\|^2] = \mathcal{E}[\|\boldsymbol{\alpha} - K\boldsymbol{\beta}\|^2]$. The minimum-variance problem is now reduced to the problem of finding K and $\boldsymbol{\beta}$ so that the variance $\mathcal{E}[\|\boldsymbol{\alpha} - K\boldsymbol{\beta}\|^2]$ is minimized.

The minimum-variance estimate is readily available. Indeed, given $\boldsymbol{\beta}$, the optimal matrix K is completely determined and is given by [12]

$$K = \mathcal{E}[\boldsymbol{\alpha}\boldsymbol{\beta}^\top](\mathcal{E}[\boldsymbol{\beta}\boldsymbol{\beta}^\top])^{-1}. \tag{17}$$

In fact, each β_i is the minimum-variance estimate of the corresponding α_i. We know further that

$$\mathcal{E}[\|\boldsymbol{\alpha} - K\boldsymbol{\beta}\|^2] = \mathcal{E}[\boldsymbol{\alpha}^\top\boldsymbol{\alpha}] - \mathcal{E}[\boldsymbol{\alpha}^\top K\boldsymbol{\beta}]. \tag{18}$$

Thus to obtain the minimum-variance approximation of $\mathcal{X}$, it only remains to choose $\boldsymbol{\beta}$ so that

$$\mathcal{E}[\boldsymbol{\alpha}^\top K\boldsymbol{\beta}] = \left\langle \mathcal{E}[\boldsymbol{\alpha}\boldsymbol{\beta}^\top](\mathcal{E}[\boldsymbol{\beta}\boldsymbol{\beta}^\top])^{-1}, \mathcal{E}[\boldsymbol{\alpha}\boldsymbol{\beta}^\top] \right\rangle \tag{19}$$

is maximized. This nonlinear optimization problem turns out to have a simple solution [3]. Among all unbiased variables restricted to any m-dimensional subspaces in $\mathbb{R}^n$, the random variable

$$\hat{\mathcal{X}} := \sum_{j=1}^{m} (\mathbf{p}_j^\top \mathcal{X}) \mathbf{p}_j \tag{20}$$

is the best linear minimum-variance estimate of $\mathcal{X}$ in the sense that $\mathcal{E}[\|\mathcal{X} - \hat{\mathcal{X}}\|^2]$ is minimized. Additionally, the random variable $\hat{\mathcal{X}}$ defined in (20) also minimizes $\|\text{cov}(\hat{\mathcal{X}}) - \text{cov}(\mathcal{X})\|$.

Suppose now columns of $X \in \mathbf{R}^{n\times\ell}$ represent ℓ random samples of the variable $\mathcal{X}$. By the law of large numbers, many of the stochastic properties of $\mathcal{X}$ can be recouped from X when ℓ is large enough. The question is how to retrieve a sample data matrix from X to represent the minimum-variance approximation $\hat{\mathcal{X}}$ of $\mathcal{X}$.

The connection lies in the fact that $\hat{\mathcal{X}}$ is simply the projection of $\mathcal{X}$ onto the subspace spanned by $\{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$. Since the covariance matrix $R = \frac{1}{\ell} X X^\top$ of the samples converges to $\text{cov}(\mathcal{X})$, if

$$R = \sum_{i=1}^{n} \mu_i \mathbf{u}_i \mathbf{u}_i^\top \tag{21}$$

denotes the spectral decomposition of R with eigenvalues $\mu_1 \geq \ldots \geq \mu_n$ and orthonormal eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$, then the projection

$$\hat{X} := \sum_{j=1}^{m} (\mathbf{u}_j^\top X) \mathbf{u}_j. \tag{22}$$

should represent samples of the best low dimensional minimum-variance estimate $\hat{\mathcal{X}}$ to $\mathcal{X}$. The *low dimension* estimate $\hat{\mathcal{X}}$ to the (continuous) random variable $\mathcal{X}$ is now comfortably translated into a *low rank* approximation $\hat{X}$ to the (discrete) random sample matrix X.

Indeed, the singular value decomposition of X

$$X = U \Sigma V^\top = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \boldsymbol{v}_i^\top \tag{23}$$

shares the same eigenvectors of R as its left singular vectors with singular values $\sigma_i = \sqrt{n\mu_i}$, $i = 1, \ldots n$. The popular notion of the truncated singular value decomposition of X, $\sum_{i=1}^{m} \sigma_i \mathbf{u}_i \boldsymbol{v}_i^\top$, is precisely $\hat{X}$ defined in (22). The truncated singular value decomposition represents random samples of the best minimum-variance linear estimate $\hat{\mathcal{X}}$ to $\mathcal{X}$ among all possible m-dimensional subspaces.

4 Link Analysis

When mining data through very large scale of objects, matrix factorization becomes increasingly difficult. Adding or deleting information requires updating or downdating the current factorization, which can be slow; and determining optimal rank m is not obvious, which can lead to inaccuracy. Regardless, this process is an inevitable task. For effective search, most search engines continually index documents, mine and retrieve information, and store the data in an organized way for quick reference when needed. But then another issue arises. Taking today's World Wide Web for example, a query usually can bring up deluging information which must be sorted again to reveal the most relevant pages. Link analysis turns out to be another area where linear algebra can help to tackle this ranking problem. We outline two ideas in the context of Web search engines.

4.1 HITS Algorithm

Given a query, assume that n Web pages have been matched through some search mechanism. Without some ordering of the pages, the user may still be lost at sea since n could be a huge number. The HITS algorithm is meant to attach an importance rating to these pages so that the most important pages are sorted to the top of the list [7].

Let $\mathbb{I}_i$ and $\mathbb{O}_i$ denote the set of pages linking into and out of the page P_i, respectively. One way to measure the importance of page P_i is to assign an authority score a_i and a hub score h_i, recording the degrees of linkages into or out of P_i, respectively. Starting with uniform scores, $h_i^{(0)} = \frac{1}{n}$, the pages compete for their authorities and hub reputations. It makes sense to trade the scores according to the rules,

$$a_i^{(k)} = \sum_{j:P_j\in\mathbb{I}_i} h_j^{(k-1)}, \quad h_i^{(k)} = \sum_{j:P_j\in\mathbb{O}_i} a_j^{(k)}. \tag{24}$$

Let L denote the adjacency matrix where $L_{ij} = 1$ if $P_j \in \mathbb{O}_i$; and 0, otherwise. We can represent the evolution of scores via the matrix form,

$$\mathbf{a}^{(k)} = L^\top \mathbf{h}^{(k-1)}, \quad \mathbf{h}^{(k)} = L\mathbf{a}^{(k)}, \tag{25}$$

where $\mathbf{a}^{(k)} = [a_1^{(k)}, \ldots, a_n^{(k)}]^\top$ and so on. It follows that

$$\mathbf{a}^{(k)} = (L^\top L)\mathbf{a}^{(k-1)}, \quad \mathbf{h}^{(k)} = (LL^\top)\mathbf{h}^{(k-1)}. \tag{26}$$

With appropriate normalization (which interprets the scores as percentages), this algorithm amounts to the power method that computes the dominant eigenvectors of $L^\top L$ and $LL^\top$, respectively. The limit points of these iterations, if exist, provide a ranking of importance for each page.

4.2 PageRank Algorithm

PageRank, employed by Google, is another link analysis technique that provides accuracy for Web surfing. Let $|\mathbb{O}_i|$ denote the number of out links from page P_i. The PageRank

algorithm differs from the HITS algorithm mainly in that it distributes importance equally to all linked pages and thus conveniently introduces the notion of probability. The PageRank r_i for page P_i is defined to be

$$r_i^{(k)} := \sum_{j:P_j \in \mathbb{I}_i} \frac{r_j^{(k-1)}}{|\mathbb{O}_j|}. \tag{27}$$

Let H denote the modified adjacency matrix where $H_{ij} = \frac{1}{|\mathbb{O}_i|}$ if $P_j \in \mathbb{O}_i$; and 0, otherwise. The evolution of the row vector $\mathbf{r}^{(k)} = [r_1^{(k)}, \ldots r_n^{(k)}]$ can be expressed as

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} H. \tag{28}$$

Note that H is a row stochastic matrix and each $\mathbf{r}^{(k)}$ is a probability distribution vector. Surfing on Web is thus interpreted as a random walk on the graph defined by the hyperlinks [10].

To avoid the possibility that $\mathbb{O}_i$ is an empty set, and more so to ensure convergence, the hyperlink matrix H is further modified to become

$$G = \alpha \left(H + \frac{\mathbf{a}\mathbf{1}^\top}{n} \right) + (1-\alpha)\frac{\mathbf{1}\mathbf{1}^\top}{n}, \tag{29}$$

where $\mathbf{a}, \mathbf{1} \in \mathbb{R}^n$ are column vectors with $a_i = 1$ if $\mathbb{O}_i = \emptyset$; and 0, otherwise; $\mathbf{1}$ has 1 in all entries, and $\alpha \in [0,1]$ is a parameter. In this way, G remains row stochastic, but is also irreducible and aperiodic. The stationary distribution vector

$$\mathbf{r} = \mathbf{r}G \tag{30}$$

exists, is unique, and provides a ranking of importance for each page [1].

Finding the stationary distribution vector $\mathbf{r}$ is certainly not a easy job, because the Google matrix G has indexed billions of pages and the size is constantly growing. For that reason, iterative method is perhaps the only choice of method and convergence rate becomes a concern. The rescue lies with the parameter α. As a stochastic matrix, the power method applied to G converges at the rate of its second largest eigenvalue $|\lambda_2|$, which has been proved to be $|\lambda_2| = \alpha$ precisely [9]. It has been said that Google uses $\alpha = .85$ and the PageRank can be found to be within 10^{-4} accuracy by 50 iterations regardless the size of the matrix.

The mathematical theory above does not address all issues arising in real-life applications. Link structure over the web is extremely dynamical. The PageRank needs update periodically. The mechanism of updating an old PageRank is still an open question.

5 Structural Constraints

In sciences and engineering, data values out of mathematical models are often constrained for the sake of feasibility and interpretability. The fact of the matter has been demonstrated in our applications in Section 2, where constraints such as nonnegativity or constant row sums must be imposed. There are also situations in settings such as DNA microarray analysis or drug discovery where the data must be refrained to either 0 or 1, resulting a binary decomposition [8].

Very few of currently available mining techniques are readily generalizable to take structured data into account. Nonnegative matrix factorization, for example, plays a major role in a wide range of important applications, yet there is still little theory on how the factorization can be robustly and efficiently accomplished. At present, most algorithms are data specific and the rate of convergence is slow. Structured low rank approximation should be an important area for future research.

References

[1] K. Bryan and T. Leise. The $\$25,000,000,000$ eigenvector: the linear algebra behind Google. *SIAM Rev.*, 48(3):569–581 (electronic), 2006.

[2] M. T. Chu and R. E. Funderlic. The centroid decomposition: relationships between discrete variational decompositions and SVDs. *SIAM J. Matrix Anal. Appl.*, 23(4):1025–1044 (electronic), 2002.

[3] M. T. Chu and G. H. Golub. *Inverse eigenvalue problems: theory, algorithms, and applications*. Oxford University Press, New York, 2005.

[4] I. S. Dhillon and D. M. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning J.*, 42:143–175, 2001.

[5] D. Donoho and V. Stodden. When does nonnegative matrix factorization give a correct decomposition into parts? Technical report. Proc. 17th Ann. Conf. Neural Infromation Processing Systems, NIPS 2003.

[6] P. K. Hopke. *Receptor Modeling for Air Quality Management*. Elsevier, Amsterdam, Hetherlands, 1991.

[7] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[8] M. Koyutürk, A. Grama, and N. Ramakrishnan. Nonorthogonal decomposition of binary matrices for bounded-error data compression and analysis. *ACM Trans. Math. Software*, 32(1):33–69, 2006.

[9] A. N. Langville and C. D. Meyer. A survey of eigenvector methods for Web information retrieval. *SIAM Rev.*, 47(1):135–161 (electronic), 2005.

[10] A. N. Langville and C. D. Meyer. *Google's PageRank and beyond: the science of search engine rankings*. Princeton University Press, Princeton, NJ, 2006.

[11] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Natural*, 401:788–791, 1999.

[12] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons Inc., New York, 1997.

[13] H. Park, M. Jeon, and J. B. Rosen. Lower dimensional representation of text data in vector space based information retrieval. In *Computational Information Retrieval*, pages 3–23. SIAM, 2001.