Date of Acceptance	Grade
December 2007	5/5

Instructor

Saara Hyvönen, Ella Bingham, Roman Yangarber

Recommendation of Multimedia Objects Using Metadata and Link Analysis

Davin Wong

Helsinki November 21, 2007

UNIVERSITY OF HELSINKI

Department of Computer Science

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section	Laitos – Institution – Department
Faculty of Science	Department of Computer Science

Tekijä – Författare – Author
Davin Wong

Työn nimi – Arbetets title – Title
Recommendation of Multimedia Objects Using Metadata and Link Analysis

Oppiaine – Läroämne – Subject
Computer Science

Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages
M. Sc. Thesis	November 2007	55

Tiivistelmä – Referat – Abstract

We investigate methods for recommending multimedia items suitable for an online multimedia sharing community and introduce a novel algorithm called UserRank for ranking multimedia items based on link analysis. We also take the initiative of applying EigenRumor from the domain of blogosphere to multimedia. Furthermore, we present a strategy for making personalized recommendation that combines UserRank with collaborative filtering. We evaluate our method with an informal user study and show that results obtained are promising.

Avainsanat – Nyckelord – Keywords
Link Analysis, Collaborative Filtering, Recommendation Systems

Säilytyspaikka – Förvaringställe – Where deposited
Kumpula Science Library, serial number C-

Muita tietoja – Övriga uppgifter – Additional information

Table of Content

Chapter 1. Introduction

We are about to embark on a quest to build a recommendation system for an online community based multimedia sharing service. We fully understand the wisdom in the old saying that "Rome wasn't built in a day"; nevertheless, we are curious to see whether we can cobble together a decent recommendation system with efforts and time typical of a Master's thesis. In this chapter, we begin our mission by laying out our motivation, goal and plan of action.

1.1 Background

With the advent of Internet, information sharing has become significantly easier than before. The prevalence of broadband has also made sharing photos and videos a reality. Many online services that offer publishing and sharing of photos and videos have spawned over the recent years. Major players in the market include Yahoo Flickr, Google YouTube, Photobucket, Google Picasa, Zooomr, etc. As an example of the popularity of these services, at the very minute that this sentence is being typed, Photobucket has had over 3,911,633,403 images uploaded by users to date and is increasing at a rate of about 70 images per second!

Not all services are created equal, some have more features and offerings than others but the core functionalities are similar across the board. For instance, many sites allow users to add metadata to photos such as title, description and tags. Tags are essentially keywords or terms used to describe a photo and they are typically single words in lower case. For example, a cat photo may be tagged with the term "cat", "kissa" (Finnish word for cat), "kitten", "kitty", "feline", "garfieldthecat", "orange", "rambunctious", etc. The purpose of tagging is to enable text-based retrieval and management of photos; so for example, somebody who wishes to view cat photos can search for all photos with the tag "cats". Another common feature appearing in many photo sharing services is the ability for users to comment or rate each other's photos. This social engagement in fact is the tie that binds the users of a service into an active community of contributors and peer reviewers, and it is also the basis for our study.

1.2 Objective

The central focus of this thesis is to investigate and attempt to design a multimedia recommendation system. We would like to be able to proactively suggest multimedia items that may be of interest to users based on some measures of interestingness. The proposed system is predictive and unintrusive in that recommendations are drawn by passively observing and analyzing the activities collected about users and their items from an online multimedia sharing service. Since such data typically accumulates over time as the size of the user base increases and the amount of activities grows, the recommendations offered by the system are also implicitly dynamic and adaptive to the growing breadth and depth of the data.

Apparently, the success of the recommendation system hinges on the definition of interestingness which is an admittedly rather slippery and elusive concept. While this may seem like a formidable task at first, in fact, a simple definition of interestingness exists and is prevalent among sites that let users rate items online.[1] The most interesting items can be defined as the ones that received the highest average rating from the community. This formulation of interestingness is global in the sense that the recommendation produced by the measure is the same for all users. The advantage of global interestingness is that it maximizes viewership of the topmost interesting items. This maximum exposure may entice contributors to strive for the top interestingness ranking by contributing better items all the time. As the site becomes more prominent, these top spots not only offer fame to the contributors but can also carry monetary compensation as well.[2] Inevitably, with the users being humans, the battleground for the top is also littered with treachery and manipulation. A

[1] Flickr is different in that its recommendations are computed using some interestingness algorithms according to their patent applications [15, 16]. Unfortunately, the algorithm is proprietary and is not disclosed to public.
[2] The author is aware of at least one instance where an advertisement agency purchased exclusive rights

simple rating averaging mechanism for computing global interestingness can be easily gamed by dishonest users. One of our goals, therefore, is to devise an alternative formulation of global interestingness that is more resilient against manipulation.

Earlier, we described the importance and benefits of interestingness from the point of view of contributors, we now switch gears and put ourselves in the shoes of viewers. If we think of the recommendation system as a marketing ploy where we promote items to potential buyers, then global interestingness is a one-size-fits-all strategy and completely ignores the nuances between the preferences and needs of individuals. The chance of having all the top recommended items appealing broadly to the individual interests of the masses is rather slim. To boost our hypothetical sales, what we would like is a personalized interestingness algorithm that takes into account user preferences by studying users' past behavior based on the information that was collected about them from the online service. The development of such a personalized algorithm is the second target of this thesis.

1.3 Methodology

For the purpose of this study, we have available to us a small data set extracted from Flickr which is an online photo sharing service. The data set composes of 2524 users who collectively own 2,177,103 photos. Of these photos, 34.97% are commented by users within the set. On average, each user commented 302 photos and marked 120 photos as their favorites. Some of the photos commented by a user can include his/her own photos while photos that are marked as favorites belong to other users. In terms of tags, there are 577,353 unique tags in the collection. While there are also other types of metadata available in the data set such as title and description of photos and albums, we concentrate on using tags alone as the textual representation for photos. We do not attempt to analyze the photos through any image processing or recognition means to look for clues of what is in the photo. Our

to photos from Flickr members for marketing use.

recommendation engine is based entirely on tags, ownership and evaluation information. By evaluation, we mean the trail of comments and favorite photos left behind by the users.

To ascertain the quality of the recommendation produced by the global and personalized interestingness algorithm, we conducted an informal user study. We chose 26 of the 2524 users from the data set and had them grade the proposed recommendation based on several criteria such as general interestingness, personal appeal, aesthetics and overall quality of the photos recommended.

For the rest of the discussion, Chapter 2 and 3 lay out the theoretical foundation for the thesis. Much of the proposed algorithm relies on linear algebra and with insights borrowed from web mining algorithms, in particular, PageRank and HITS. Unfortunately, Linear Algebra is an extensive field so it is beyond the scope of this thesis to cover its entirety. Rather, only important and germane concepts are drawn and reviewed in Chapter 2. In Chapter 3, we devote the entire discussion to PageRank and HITS. An understanding of the inner workings of PageRank and HITS will allow us to develop our own algorithm in the following chapters. The thrust of the thesis is in Chapter 4 and 5. We first explore and develop various methods to rank photos by global interestingness in Chapter 4. In Chapter 5, we utilize the results from the previous chapter to customize recommendation to individual users. We put our algorithms to test in Chapter 6 where we discuss the results of our user study. In Chapter 7, we conclude our discussion and explore future possibilities.

Chapter 2. Concepts from Linear Algebra

This chapter provides a cursory overview of some of the important results from linear algebra that will be used in the discussion of our topic. By no means is it intended as a crash course on linear algebra but merely as a refresher for concepts already familiar to the readers. For a more complete and thorough treatment of the subject, the readers is advised to consult the materials provided in the references [6, 11, 13].

Vector

A vector is an abstraction with magnitude and direction and can be visualized as an arrow on a plane. Using the origin as a point of reference in Euclidean space, a vector can be thought of as a set of coordinates. By convention, we write a vector as a vertical array of numbers such as $v = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$. To address the k^{th} element of the vector, we subscript the index next to the variable as such, $v_1=3$ and $v_2=5$. A vector can also be multiplied by a scalar to increase or decrease its magnitude, or added/subtracted by another vector to alter both magnitude and direction. While the example vector given above is two dimensional, a vector can have any dimension. We say that a vector is of size n if it has n elements or dimensions.

Vectors and matrices can be considered as the elementary data structures in linear algebra. The study of their properties and operations lies in the core of linear algebra. One of the ways that we will be using vectors is to store the interestingness scores for multimedia items.

Vector Dot Product

Two vectors of the same size can be mapped to a scalar through the dot product operation. The dot product, also called the scalar product or Euclidean inner product, sums the component wise products of two vectors. For example, for the two vectors x and y such that,

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \text{ and } y = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

the dot product x●y is 1·4+2·5+3·6=32. An important concept related to dot product is orthogonality. We say two vectors are orthogonal if their dot product is 0.

Vector Norm

The norm of a vector is its length and there is more than one way to measure the length of a vector. Perhaps the most notable way is the 2-norm or Euclidean-norm which is related to the Pythagorean Theorem and is defined as,

$$\|v\|_2 = \sqrt{v \bullet v} = \sqrt{\sum_{k=1}^{n} v_k^2}$$

Another example of norms is the 1-norm better known as Manhattan norm and is defined as,

$$\|v\|_1 = \sum_{k=1}^{n} |v_k|$$

Sometimes, when we are operating with vectors, it is convenient to normalize the vector so its norm is 1. We called such normalized vectors unit vectors since they have unit length. Different norms produce different unit vectors. Below, we show the unit vectors for 2-norm and 1-norm where any vector starting from the origin and ending on the circle or square is a unit vector.

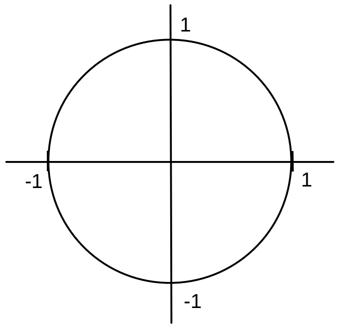

Figure 2.1 2-norm unit vectors

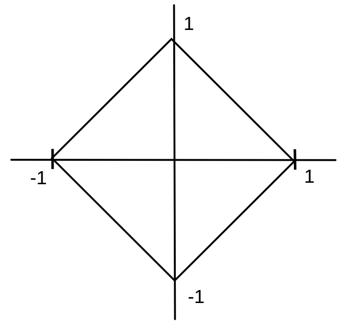

Figure 2.2 1-norm unit vectors

Cosine Similarity

The dot product described earlier has a geometric interpretation. If u and v are two vectors of the same dimension and we align them so that they share the same initial point, then the angle between them is,

$$\theta = \cos^{-1}\left(\frac{u \bullet v}{\|u\|_2 \|v\|_2}\right)$$

$$= \cos^{-1}\left(\frac{u}{\|u\|_2} \bullet \frac{v}{\|v\|_2}\right)$$

In other words, the angle between any two vectors is the inverse cosine of the dot product of the two vectors normalized to unit length. The cosine of the angle is known as the cosine similarity.

The cosine similarity of two vectors plays a very important role in information retrieval. Often times, in text mining, documents are treated as bags of words without any order between the words. Each document is represented as a vector where the elements in the vector are the frequencies of the terms in the document scaled by some factor. Two documents are considered similar if the angle between their vectors is small. In terms of cosine similarity, this means the value approaches 1. (The vectors are nonnegative as the elements in the vector represent frequencies, so the cosine of the angle is always between 0 and 1.) When two documents are identical, their cosine similarity is 1.

For our purpose, we use cosine similarity to measure the similarity between users through the tags associated with the photos they own. For example, suppose we have three users with the following distribution of tag frequencies,

	cats	dogs	flowers
user 1	8	2	0
user 2	5	5	2
user 3	0	10	2

Each row of the table lists the number of times a tag occurs in the collection of photos owned by the user. To compute the cosine similarity between user 1 and 2, we first compute the Euclidean norm of the two vectors,

$$u_1 = \begin{bmatrix} 8 \\ 2 \\ 0 \end{bmatrix} \text{ and } u_2 = \begin{bmatrix} 5 \\ 5 \\ 2 \end{bmatrix}$$

$$\|u_1\|_2 = \sqrt{8^2 + 2^2 + 0^2} = \sqrt{68}$$

$$\|u_2\|_2 = \sqrt{5^2 + 5^2 + 2^2} = \sqrt{54}$$

The cosine similarity between u_1 and u_2 is then,

$$\cos\theta = \frac{8 \cdot 5 + 2 \cdot 5 + 0 \cdot 2}{\sqrt{68} \cdot \sqrt{54}} = 0.83$$

If we carried out the cosine similarity between all pairs of users, we obtain the following results.

	User 1	User 2	User 3
User 1	1	0.8251	0.2378
User 2	0.8251	1	0.7206
User 3	0.2378	0.7206	1

From the table, we conclude that user 1 is more similar to user 2 than to user 3, and that user 2 is more similar to user 1 than to user 3.

Matrix

A matrix is a two dimensional array of numbers. In link analysis, we use a matrix to describe the relationship between a set of entities. For example, suppose we are dealing with web pages, then a matrix can be used to describe the hyperlink relationship between web pages. To illustrate this with a small example, consider the following miniature web with 4 web pages,

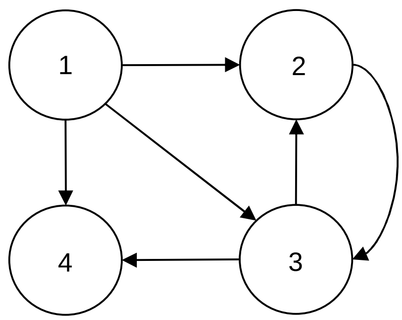

Figure 2.3 Link structure of web

The arrows on the graph denote hyperlinks between web pages. We can define a matrix A so that the element in the i^{th} row and j^{th} column is 1 if there exists a link from the i^{th} page to the j^{th} page. This matrix A would look like,

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this case, our matrix A is a square matrix but typically, a matrix can be of any sizes. We say a matrix is of size m-by-n if it has m rows and n columns.

By convention, an upper case alphabet is used to denote a matrix. To refer to an element in a matrix, we use the lower case of the matrix symbol and append the row and column indices to be referenced. For example, the element in the second row and third column of A is referenced by $a_{2,3}$ which in our example is equal to 1. Sometimes, we may also wish to refer to any element in A, we do so with $a_{i,j}$ (or a_{ij} as shorthand) where i refers to any row and j refers to any column. At other times, we may also need to refer to any column or row in a matrix. To address any column, we use $a_{\cdot j}$ and for row, we use $a_{i\cdot}$. It should be apparent that $a_{\cdot j}$ can be treated as a column vector while $a_{i\cdot}$ is a row vector.

A common matrix that we often encounter is the identity matrix, denoted as I. The identity matrix is a square matrix with ones along the main diagonal and zeros everywhere else.

Matrix Multiplication

A matrix can be multiplied by a vector to produce another vector or by a matrix to produce another matrix. We first define matrix-vector multiplication and from that, we can see that matrix-matrix multiplication follows naturally. To multiply a matrix with a vector, the dimension has to match. For an m-by-n matrix A to be post-multiplied by a column vector x, x must be of size n. The product of the multiplication Ax is a column vector of size m. Formally, matrix-vector multiplication is defined as,

$$b = Ax = \sum_{j=1}^{n} x_{\cdot j} a_{j}$$

We can interpret the product as a weighted sum of the column vectors of A. For example, suppose A = $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and x = $\begin{bmatrix} 5 \\ 6 \end{bmatrix}$, then b = Ax = $5\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 6\begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 17 \\ 39 \end{bmatrix}$.

A matrix times a matrix also has a dimension compatibility requirement. A p-by-m matrix can only be multiplied by an m-by-n matrix and the result is a p-by-n matrix. Formally, let A be a p-by-m matrix and C be an m-by-n matrix, then B=AC is defined as,

$$b_{\cdot j} = Ac_{\cdot j} = \sum_{k=1}^{m} c_{kj} a_{\cdot k}$$

In other words, the j^{th} column of B is the weighted sum of the column vectors of A and the weights are the elements in the j^{th} column of C. For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } C = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} \rightarrow AC = \left[5\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 6\begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad 7\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 8\begin{bmatrix} 2 \\ 4 \end{bmatrix} \right] = \begin{bmatrix} 17 & 23 \\ 39 & 53 \end{bmatrix}$$

<u>Matrix Transposition</u>

The transposition operation for a matrix interchanges the row and column elements of the matrix. Formally, let $A=[a_{ij}]$, the transpose of A denoted by A' is $[a_{ji}]$. For a square matrix, this operation is akin to flipping the elements in the matrix along the main diagonal, for example, for matrix $A=\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $A'=\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$.

In general, for a column vector which is an m-by-1 matrix, its transpose is the row vector which is a 1-by-m matrix.

<u>Matrix Norm</u>

Like vectors, matrices have norms as well. A common way to introduce matrix norm is through the vectors that a matrix operates on. Formally, a vector induced matrix norm is defined as,

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

Plainly speaking, this matrix norm is the maximum stretching ratio induced by the matrix on a vector.

<u>Symmetric Matrix</u>

A symmetric matrix is a square matrix that is equal to its transpose. It can be shown that the product of a matrix with its transpose forms a symmetric matrix. We give an example here without proof.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ and } AA' = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

Matrix Inverse

A square matrix B is invertible if there exists a matrix C such that CB = BC = I where I is the identity matrix. The matrix C is called the inverse of B while B is the inverse of C. Not all square matrices are invertible but if an inverse exists, it must be unique. By convention, we denote the inverse of an invertible matrix B as B^{-1}, hence, $B^{-1}B = BB^{-1} = \mathrm{I}$.

For our purpose, we will focus on a special class of matrices called the diagonal matrices which can easily be shown to be invertible. We already encountered the identity matrix I which is a diagonal matrix. In general, a matrix D is a diagonal matrix if $d_{ij}=0$ where $i \neq j$. The definition requires that all elements off the main diagonal of the matrix are zeros while the elements along the diagonal can have any values including zeros. A diagonal matrix is invertible if and only if all of its diagonal elements are nonzero. The inverse D^{-1} is also a diagonal matrix with diagonal elements being the reciprocal of the diagonal elements of D. For example,

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \text{ and } D^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \rightarrow DD^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathrm{I}$$

Orthogonal Matrix

A matrix U is orthogonal if the product of the matrix with its transpose is the identity matrix, that is, $U'U = UU' = \mathrm{I}$. In other words, the transpose of the matrix is also the inverse of the matrix. We will encounter the orthogonal matrix again when we discuss eigenvalue decomposition.

Eigenvalues and Eigenvectors

For a square matrix A, if there exists a scalar λ and a non-zero vector x such that Ax=λx, then we say λ is an eigenvalue for A and x is called an eigenvector corresponding to λ, and together, the pair (λ,x) is called an eigenpair for A. Plainly speaking, an eigenvector for a matrix A is a vector such that when multiplied by the matrix its direction remains unchanged or reversed while its magnitude is scaled by λ. For example, for matrix

$$B = \begin{bmatrix} 2 & -3 & 1 \\ 1 & -2 & 1 \\ 1 & -3 & 2 \end{bmatrix},$$

the eigenpairs are (0, $\begin{bmatrix} -0.58 \\ -0.58 \\ -0.58 \end{bmatrix}$), (1, $\begin{bmatrix} 0.82 \\ 0.41 \\ 0.41 \end{bmatrix}$) and (1, $\begin{bmatrix} -0.68 \\ 0.02 \\ 0.73 \end{bmatrix}$).

There are numerous properties related to eigenvalues and eigenvectors. The ones that we will encounter are:

1. λ is an eigenvalue of A if and only if λ is an eigenvalue of A'.
2. |λ| ≤ ||A|| where ||.|| denotes any norm and λ is an eigenvalue of A.

Spectral Radius

The spectral radius of a matrix A is the magnitude of the largest eigenvalue(s) for A and is denoted by ρ(A). The eigenvalues for a matrix can be real or complex (all matrices and vectors are assumed to be real in this paper however). If we plot the eigenvalues for A on a complex plane that is centered at the origin, then all the eigenvalues must lie within the spectral radius from the origin. The circle with spectral radius centered at the origin on the plane is called the spectral circle of A. We give an example of a spectral circle in Figure 2.4 for the eigenvalue set {-1, 0, ½}.

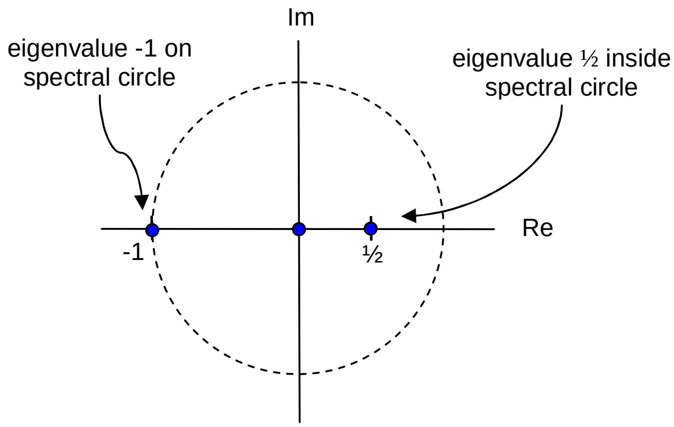

Figure 2.4 Spectral circle with radius ρ(A)=1.

Similarity Transformation

If matrix X is invertible, then the map $A \rightarrow X^{-1}AX$ is called a similarity transformation of A. We say that A and B are similar if $B = X^{-1}AX$ for some invertible matrix X. One very important property of similarity transformation is that it preserves spectral properties such as eigenvalues, i.e. if matrix A and B are similar, then their eigenvalues are identical. Two well known instances of similarity transformation are eigenvalue decomposition and Schur factorization.

Eigenvalue Decomposition

An eigenvalue decomposition for matrix A exists if A can be factorized into the form $X\Lambda X^{-1}$ where X is invertible and Λ is diagonal. The column vectors $x_{\cdot j}$ are the eigenvectors of A and the diagonal elements Λ_{ij} are the eigenvalues corresponding to eigenvectors $x_{\cdot j}$.

Not all square matrices have eigenvalue decomposition. A special class of matrices that has eigenvalue decomposition includes the symmetric matrices. The eigenvectors of symmetric matrices are also orthogonal, i.e. the dot product between any pair of eigenvectors is 0. As a result, matrix X can be made orthogonal by normalizing the eigenvectors by their Euclidean norm.

Schur Factorization

Every square matrix A has a Schur factorization of the form QTQ' where Q is an orthogonal matrix and T is an upper triangular matrix. The Schur factorization of A may not be unique. When T is a diagonal matrix, Schur factorization is the same as eigenvalue decomposition. Since the Schur factorization is a similarity transformation, the eigenvalues of A and T are identical. In fact, since T is triangular, the eigenvalues are located on its main diagonal. Because of this property, the factorization is often used to analyze the eigenvalues of a matrix. We will be using Schur factorization quite frequently when we discuss PageRank. The proof of Schur factorization by induction on the dimension of A is straightforward and we provide it here for the convenience of reference [6, 14:187].

Suppose A is an m-by-m matrix where m ≥ 2 (the case for m=1 is trivial) and let λ be an eigenvalue of A with corresponding normalized eigenvector x. Let U be an orthogonal matrix with the first column equal to x, then

$$AU = U\begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix} \text{ for some submatrix B and C}$$

Since U is invertible, we rewrite the above as $U'AU = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix} = T_1$. By induction hypothesis, we know C has a Schur factorization VT_2V'. To find Q, the orthogonal matrix that factorizes A, we let $Q = U\begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$, then

$$\begin{aligned} Q'AQ &= \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}' U'AU \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & V' \end{bmatrix} \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix} \\ &= \begin{bmatrix} \lambda & B \\ 0 & V'C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} \lambda & BV \\ 0 & V'CV \end{bmatrix}$$

$$= \begin{bmatrix} \lambda & BV \\ 0 & T_2 \end{bmatrix}$$

$$= T$$

Power Method

Power Method is a numerical iterative algorithm for computing the dominant eigenvalue and corresponding eigenvector of a matrix. It is also perhaps the slowest method available but it has the advantage that it requires minimal storage space in carrying out the steps making it suitable for computation involving a large matrix. For our purpose, we use the power method to find the dominant eigenvector only. The dominant eigenvector is the eigenvector corresponding to the largest eigenvalue in magnitude. The iterative steps for finding the dominant eigenvector x of matrix A are,

```
x0 = e/||e||
while ||xk-xk-1|| > some threshold
    qk = Axk-1
    xk = qk/||qk||
```

Figure 2.5 Power Method for finding dominant eigenvector.

where e is a vector with 1s and ||.|| denotes any vector norm. The purpose for normalizing q_k by $||q_k||$ is to avoid overflow so theoretically speaking, it is not needed. Without the normalization step, the method simply becomes $x_k = Ax_{k-1}$ which is equivalent to computing $A^k x_0$.

We can visualize the operation of Power Method by following the course of x_k through the iterations. In each execution of the loop, the multiplication Ax_{k-1} pulls x_k toward the direction of the dominant eigenvector. Repeating executions of the loop further amplifies the pull until at some point the changes becomes small enough that the loop terminates and the approximated dominant eigenvector x is returned.

We mentioned earlier that Power Method is slow. To be exact, if A has eigenvalues $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$, then the convergence rate is $|\lambda_2|/|\lambda_1|$.

<u>Perron-Frobenius Theorem</u>

Perron-Frobenius Theorem for nonnegative matrices enables us to understand the spectral characteristics of a matrix. Two important results from the theorem are irreducibility and primitivity. A matrix P is reducible if there exists a similarity transformation such that $X'PX=\begin{bmatrix} A & B \\ 0 & C \end{bmatrix}$ where A, B and C are matrices. Irreducibility can be made more clearly understood if we interpret P as an adjacency matrix for a directed graph. For example, the directed graph illustrated in Figure 2.3 can be represented by the adjacency matrix,

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

The element in the i^{th} row and j^{th} column of P denotes the existence of a link from node j to node i. P is irreducible if and only if there exists a directed path between any two vertices on the graph. In graph theory terminology, such graph is also called strongly connected. For the example above, P is not irreducible because there is no path that connects from node 2 to node 1.

Perron-Frobenius theorem states that if $A \geq 0$ is an irreducible square matrix, then the following must be true [9:172]:

1. A unique real eigenvalue $\lambda > 0$ exists that lies on the spectral circle of A, hence, no other eigenvalue has magnitude bigger than λ.

2. There exists an eigenvector x with all elements $x_i > 0$ such that $Ax=\lambda x$.

Irreducibility alone does not guarantee that λ is the only eigenvalue on the spectral circle, e.g. $-\lambda$ might be on the spectral circle as well. However, if λ is the only eigenvalue on the spectral circle, then the matrix is also called primitive. Primitivity is very important when we discuss PageRank in Chapter 3. There are at least two possible tests for primitivity when A is a square nonnegative matrix [9:174],

1. A is primitive if and only if $A^m > 0$ for some $m > 0$.
2. A is primitive if A is irreducible with at least one positive diagonal element.

Chapter 3. Web Mining Algorithms

Besides the enormous amount of content available on the World Wide Web, the web also holds a very special kind of data that prior to the invention of PageRank [2] and HITS [8] went largely unnoticed and untapped by the information retrieval community. The basis of the web is formed by hyperlinks connecting web pages together. These hyperlinks are like referrals. A page may link to another page because the other page may hold some valuable information that the viewer of current page may also like to check out. From the point of view of the linked page, the links it receives are tantamount to endorsement by other pages. The web then can be viewed as a giant recommendation system where each page recommends other pages while it receives recommendations from others. In this context, two separate algorithms, PageRank and HITS, were developed coincidentally around the same time by two different groups that exploit the hyperlink structure of the web. The algorithms differ by the way they assign importance to the web pages based on knowledge of the collective recommendations and the mechanism for arriving at the rankings. In this chapter, we will pay homage to PageRank and HITS but it is also worth to mention that other graph-based algorithms have been developed over the years such as TrafficRank [13] and SALSA [10].

3.1 PageRank

PageRank, the key ingredient behind the popular Google search engine, was invented in 1998 by Sergey Brin and Larry Page who were then graduate students at Stanford University [2]. The idea behind PageRank is surprisingly simple yet effective. Exploiting the hyperlink structure of the web, PageRank surmises that each web page has a prestige score that ties to the number of inlinks a page receives. The more prestigious sources linking to a page, the more prestigious the page becomes. The presumption is that a web page with high prestige score tends to contain more valuable information than one with low prestige; hence, it is more useful to a searcher and deserves a higher position in the search results. To see why this might be the case, we need to articulate the idea more rigorously.

First, we define the weighted adjacency matrix Q such that,

$$q_{ij} = \frac{1}{N_j} \text{ if web page j links to page i, otherwise 0}$$

where N_j is the number of outlinks page j has. The non-zero elements in each column of Q represent the weights of the outbound links from a page. The non-zero elements in each row of Q represent the weights of the inbound links to a page. The weighting scheme enables us to interpret the outbound links of a page as probabilistic transitions from the page. For example, for a page with three outlinks, each outbound destination has a one-third chance of being reached from the page. We can visualize the entire web then as a giant network of state transitions with each state corresponding to a web page. The prestige that we seek traverses the states and diffuses through the network until equilibrium is reached assuming one exists.

Formally, we define prestige as,

$$r = Qr \tag{3.1.1}$$

where r is a column vector of prestige scores for the pages. The above equation says that the prestige of a page is the sum of the prestige of the sources weighted by the probabilities of transition from the sources to the page. For example, for the following weighted adjacency matrix and prestige score,

$$Q = \begin{bmatrix} 0 & 0 & a \\ b & 0 & c \\ d & e & 0 \end{bmatrix} \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

The prestige for page 2 is $r_2 = br_1 + 0r_2 + cr_3$ and the prestige for page 3 is $r_3 = dr_1 + er_2 + 0r_3$. The prestige scores that we are looking for are the prestige scores at equilibrium with Equation 3.1.1. Interpreting Equation 3.1.1 as an eigenvalue problem, it assumes eigenvalue 1 exists and we are looking for the corresponding eigenvector. Obviously, whether eigenvalue 1 exists depends on Q. If we can somehow guarantee its existence, then we have a chance for solving the equation. (If we would be more lenient with Equation 3.1.1 and instead define prestige as $\lambda r = Qr$ where λ is the largest positive eigenvalue of Q, then we might stand a better chance of finding a solution though the sensibility of the solution would still need to be verified by experiments on test data.) A more practical problem is that the size of matrix Q limits our choice of computation methods for solving the equation. For the World Wide Web, the number of web pages is in the billions and the size of matrix Q is billions to the power of 2. Even when sparsity is considered since many entries in Q are likely 0, the space requirement to carry out the computation is still immense. The only viable option is the Power Method which requires minimal intermediate storage in carrying out the iterative steps toward solution. A sacrifice has to be made though in choosing the Power Method, the nature of the method requires that Q^k converges to a stable matrix as $k \to \infty$. What can possibly go wrong is that there is more than one distinct eigenvalue on the spectral circle of Q. To illustrate the problem, suppose +1 and -1 are both eigenvalues of Q with spectral radius 1, then by Schur Factorization,

$$\begin{aligned} Q^k &= (UTU')^k \\ &= UTU'UTU'...UTU' \\ &= UTIT...ITU' \\ &= UT^kU' \end{aligned}$$

where U is an orthogonal matrix and T is upper-triangular. This means that the convergence of Q^k rests on T^k. Since T is upper-triangular with eigenvalues of Q along the main diagonal, the main diagonal of T^k consists of eigenvalues of T raised to the k^{th} power,

$$T^k = \begin{bmatrix} 1^k & * & * & \cdots \\ 0 & (-1)^k & * & \\ & & \ddots & \\ & & & \end{bmatrix}$$

where * denotes any real number. With eigenvalue -1 to the k^{th} power toggling between -1 and +1, it is apparent that Q^k will not converge.

It turns out that we can guarantee eigenvalue 1 exists and is the only distinct eigenvalue on the spectral circle of Q if we modify Q slightly. The enabler behind the maneuver is Perron-Frobenius Theorem. The key properties from the Perron-Frobenius Theorem that we utilize are irreducibility and primitivity.

3.1.1 Random Surfer

Earlier, we envisioned Equation 3.1.1 as prestige flowing across a network, we now take a slightly different interpretation. The original conception of PageRank was based on the notion of a random surfer. The random surfer navigates the web by following the links of the web pages. Each outlink on a page is chosen randomly by the surfer. Over time, the surfer may revisit the same page over and over again simply by navigating the link structure. The idea is that the proportion of time spent by the surfer on a given page reflects the importance of that page and the pages that are linked from an important page must also be relatively important as well. All is swell except that some of the pages on the web have no outlinks. The random surfer, therefore, gets stuck on those pages. To avoid getting stuck, Brin and Page altered the weighted adjacency matrix Q so that all linkless pages now link to every other pages, in other words, the prestige of the formerly linkless page is now propagated and shared evenly by all other pages so no particular page benefits from the adjustment exclusively. Formally, the new weighted adjacency matrix P is,

$$p_{ij} = \frac{1}{N_j} \text{ if web page j links to page i, otherwise } \frac{1}{\tau}$$

where τ is the total number of web pages available on the web. In effect, we turn P into a column stochastic matrix in that each column of the matrix sums to 1. The stochasticity of the matrix means that eigenvalue 1 exists since P'e=e where e is a column vector of ones, implies 1 is an eigenvalue of both P and P' (the eigenvalues of a matrix and its transpose are identical). Furthermore, eigenvalue 1 is also on the spectral circle of P, i.e. no other eigenvalue has magnitude greater than or equals to 1, because $\rho(P) \leq ||P||_1 = 1$ where $||P||_1$ is the largest column sum of P.

3.1.2 Teleportation

We have now guaranteed the existence of eigenvalue 1 but to be able to compute PageRank by Power Method, we still need to ensure eigenvalue 1 is the only eigenvalue on the spectral circle of P. To this end, Brin and Page proposed the idea of teleportation. The insight behind teleportation is that on any given page, our random surfer may get bored and decide to jump to another page without following the links on the page. This action is akin to entering a random URL address on the location bar on most browsers. Hence, each page has a minute chance of transitioning to any other pages on the web by default. To maintain stochasticity for the sake of eigenvalue 1, we further modify the weighted adjacency matrix P to G where,

$$G = \alpha P + (1-\alpha)\frac{ee'}{n} \qquad (3.1.2.1)$$

for some $0 \leq \alpha \leq 1$, n is the number of web pages on the web and e is a column vector with n ones. The matrix G is sometimes called the Google matrix. What the equation does is to reserve a fraction of the transition weights from a page and redistribute them evenly across all pages. We illustrate this with a small example. Let $\alpha = 0.8$ and $P = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$, then $G = 0.8\begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} + (1-0.8)\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.5 \\ 0.1 & 0.5 \end{bmatrix}$.

The Google matrix is not only stochastic but it has the additional property that it is irreducible and primitive. By irreducibility, we mean that all the modified pages on the web are now strongly connected, i.e. each page is reachable from any other page through a sequence of link traversals. By Perron-Frobenius Theorem, irreducibility guarantees that our eigenvalue 1 is unique and has corresponding eigenvector with positive elements. While we were not seeking for such feature explicitly, it is still good to have since this guarantees the prestige scores are unique. What we would like to know is whether eigenvalue 1 is the only eigenvalue on the spectral circle of G. Primitivity of the Perron-Frobenius Theorem ensures us this is the case. The litmus test is the property that a square nonnegative matrix A is primitive if and only if $A^m > 0$ for some $m > 0$ which in our case is trivially so since G is a positive matrix to begin with [9:174].

3.1.3 Rate of Convergence

We have established that the Google matrix has a spectral radius of 1 and that eigenvalue 1 is the only eigenvalue on the spectral circle. Consequently, we can rest assured that the Power Method will converge iteratively to a unique stable solution. We also know that the Google matrix is primitive, so its second largest eigenvalue must be less than one, hence, the convergence rate should be $\lambda_2/\lambda_1=\lambda_2$. Since primitivity was the result of teleportation, it is natural to wonder how the teleportation parameter α affects the rate of convergence. To answer this question, we analyze Equation 3.1.2.1 to see how α affects the eigenvalues of the Google matrix by applying Schur Factorization to G to reveal its eigenvalues. The outline of our approach is to find a common orthogonal matrix U that can be used to factorize the matrices in the first and second term of Equation 3.1.2.1 [4:151, 9:46].

First, we decompose the rank one matrix ee'/n in the second term of Equation 3.1.2.1 into its Schur Factorization form $U'ee'/nU$ where U is an orthogonal matrix. We note that ee'/n has an eigenvalue 1 with any multiple of e as corresponding eigenvector. For reasons that will become obvious when we factorize P, we choose $\hat{e}$ which has Euclidean norm 1 as the eigenvector instance and let the first column of U be $\hat{e}$ and the rest of the columns of U be some U_1 so that $U = [\hat{e}\ U_1]$ is orthogonal. The Schur Factorization of ee'/n is then,

$$
\begin{aligned}
\Lambda &= U'\frac{ee'}{n}U \\
&= \frac{1}{n}\begin{bmatrix} \hat{e}' \\ U_1' \end{bmatrix} ee' \begin{bmatrix} \hat{e} & U_1 \end{bmatrix} \\
&= \frac{1}{n}\begin{bmatrix} \hat{e}'e \\ U_1'e \end{bmatrix} \begin{bmatrix} e'\hat{e} & e'U_1 \end{bmatrix} \\
&= \frac{1}{n}\begin{bmatrix} \frac{n}{\sqrt{n}} \\ 0 \end{bmatrix} \begin{bmatrix} \frac{n}{\sqrt{n}} & 0 \end{bmatrix} \\
&= \frac{1}{n}\begin{bmatrix} n \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

From the result of the factorization, we see that U_1 is not constrained by Λ and is free to take on any values as long as U is an orthogonal matrix.

Next, we apply Schur Factorization to the column stochastic matrix P in the first term of Equation 3.1.2.1 but first we note that the factorization of P is equivalent to the transpose of the factorization of P',

$$Q'P'Q = (Q'P'Q)' = Q'PQ$$

where Q is an orthogonal matrix. Therefore, by finding the Schur Factorization for P', we also find the Schur Factorization for P by taking its transpose. We proceed to factorize P' which is easier since we know P' has an eigenvalue 1 with any multiple of e as corresponding eigenvector. For the orthogonal matrix Q, we can re-use U from above since it already has ê as its first column with subsequent orthogonal columns open for choosing, hence,

$$
\begin{aligned}
T &= U'P'U \\
&= \begin{bmatrix} \hat{e}' \\ U_1' \end{bmatrix} P' \begin{bmatrix} \hat{e} & U_1 \end{bmatrix}
\end{aligned}
$$

$$= \begin{bmatrix} \hat{e}' \\ U_1' \end{bmatrix} \begin{bmatrix} P'\hat{e} & P'U_1 \end{bmatrix}$$

$$= \begin{bmatrix} \hat{e}' \\ U_1' \end{bmatrix} \begin{bmatrix} \hat{e} & P'U_1 \end{bmatrix}$$

$$= \begin{bmatrix} \hat{e}'\hat{e} & \hat{e}'P'U_1 \\ U_1'\hat{e} & U_1'P'U_1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \hat{e}'P'U_1 \\ 0 & U_1'P'U_1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & B \\ 0 & C \end{bmatrix}$$

where $B = \hat{e}'P'U_1$ and $C = U_1'P'U_1$. We can repeat Schur Factorization for C and so on until we have a complete upper triangular T. Since T and P' are similar, their eigenvalues are preserved. As a matter of fact, since T is a triangular matrix, the eigenvalues are located along the main diagonal of T so C must contain the rest of the eigenvalues of P' along its diagonal.

We are now ready to put together our proof of convergence. Let the eigenvalues of P' be $\{1, \lambda_2, ..., \lambda_n\}$, the Schur Factorization of G,

$$\begin{aligned} U'G'U &= U'(\alpha P' + (1-\alpha)ee')U \\ &= \alpha U'P'U + (1-\alpha)U'ee'U \\ &= \alpha \begin{bmatrix} 1 & B \\ 0 & C \end{bmatrix} + (1-\alpha) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \alpha B \\ 0 & \alpha C \end{bmatrix} \end{aligned}$$

shows that the eigenvalues of G are $\{1, \alpha\lambda_2, ..., \alpha\lambda_n\}$. The convergence rate by Power Method, therefore, is $\alpha\lambda_2$. In the worst case when $\lambda_2=1$, the rate of convergence becomes α. By adjusting the level of teleportation, we are in fact controlling the rate of convergence.

3.2 Hypertext Induced Topic Search (HITS)

HITS was invented by Jon Kleinberg in 1998 at IBM Almaden Research Center around the same time PageRank was invented [8]. Similar to PageRank, HITS utilizes the hyperlink structure of the web to rank web pages but unlike PageRank, the ranking for web pages is based on two criteria which are the authority and hub score. The definition for authority and hub score are mutually dependent and recursive. A web page is considered a good authority if it is linked by many good hubs while a web page is considered a good hub if it links to many good authorities. A user can select the type of web pages she desires by weighing the authority and hub score. For example, a web page with high hub score might be a directory containing links to good information about a subject while a web page with high authority score may contain a good exposé to a topic.

To appreciate the intricacy of the HITS idea, we first define formally the adjacency matrix B to describe the hyperlink structure,

$$b_{ij} = 1 \text{ if web page i links to page j, otherwise } 0$$

Using the adjacency matrix, we formulate the authority score as,

$$a = \varphi B'h$$

where a is a column vector containing the authority scores, h is the column vector of hub scores and φ is a proportionality constant. Similarly, the hub score is defined as,

$$h = \omega Ba$$

where ω is a proportionality constant. We can solve for either equation by substituting a or h into the other which yields the following pair of equations for the authority and hub score,

$$a = \varphi\omega B'Ba \qquad (3.2.1)$$

$$h = \varphi\omega BB'h \qquad (3.2.2)$$

where matrix B'B and BB' are known as the authority and hub matrix. Solving Equation 3.2.1 and Equation 3.2.2 becomes an eigenvalue problem and we will show that a stable solution exists (though not unique) when we apply Power Method. To avoid a proliferation of symbols, we will only demonstrate the steps for Equation 3.2.1 but the same procedure applies to Equation 3.2.2.

3.2.1 HITS Solution

From Chapter 2, we know that the authority matrix being symmetric has eigenvalue decomposition with orthogonal eigenvectors. It turns out that the eigenvalues of B'B are also nonnegative. To prove this, let λ be an eigenvalue of B'B and x be corresponding eigenvector, then,

$$\lambda x'x = x'\lambda x = x'B'Bx = (Bx)'(Bx) = \sum_k (Bx)_k^2 \geq 0$$

Since $x'x \geq 0$, $\lambda x'x \geq 0$ implies that λ must also be ≥ 0. While the eigenvalues for B'B are nonnegative, they are not necessarily distinct due to reducibility. Suppose the eigenvalues for B'B are $\lambda_1 = \lambda_2 > \lambda_3 \geq \ \dots \ \geq \ \lambda_n$ sorted in descending magnitude, the power method converges by computing the k^{th} power of B'B multiplied by some initial vector a_0,

$$a = (B'B)^k a_0 \qquad (3.2.1.1)$$

Let $U\Lambda U'$ be the eigenvalue decomposition for B'B, where Λ is the diagonal matrix containing the sorted eigenvalues of B'B and U is the orthogonal matrix with the corresponding eigenvectors, then the k^{th} power of B'B is,

$$\begin{aligned}(B'B)^k &= (U\Lambda U')^k \\ &= U\Lambda U'U\Lambda U' \dots U\Lambda U' \\ &= U\Lambda I\Lambda \dots I\Lambda U' \\ &= U\Lambda^k U'\end{aligned}$$

From this, we see that the computation of Equation 3.2.1.1 depends only on the diagonal matrix Λ^k as $k\to\infty$. To examine the convergence rate, we normalize Λ^k by the dominant eigenvalue λ_1^k so that,

$$\frac{\Lambda^k}{\lambda_1^k} = \begin{bmatrix} 1 & 0 & 0 & & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & \left(\frac{\lambda_3}{\lambda_1}\right)^k & & 0 \\ & & & \ddots & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{\lambda_n}{\lambda_1}\right)^k \end{bmatrix}$$

This indicates that the convergence rate depends on the ratio λ_3/λ_1 which is < 1. As $k\to\infty$, $(\lambda_i/\lambda_1)^k = 0$ for $i \geq 3$ so $\Omega=\Lambda^k/\lambda_1^k$ becomes,

$$\Omega = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & \\ \vdots & & & & \ddots \end{bmatrix}$$

Basically, all the eigenvalues drop out except the two dominant ones, λ_1 and λ_2. Multiplying $a = U\Omega U'a_0$ out term by term, we obtain,

$$a = (u_1'a_0)u_1 + (u_2'a_0)u_2$$

where u_1 and u_2 are the eigenvectors corresponding to λ_1 and λ_2, respectively. This shows that the convergent authority vector is not unique due to reducibility but is a linear combination of the leading eigenvectors where the constants in the combination are the scaled dot products of the initial vector a_0 with the dominant eigenvectors.

There are other variations of HITS that guarantees uniqueness of solution such as by applying PageRank's "teleportation" [12]. As it stands, the HITS formulation is a powerful one and also quite versatile. As we shall see in the next chapter, we can adapt a similar mutually recursive definition for our purpose of ranking multimedia items.

Chapter 4. Finding Globally Interesting Multimedia Items

The goal of this chapter is to come up with ways to identify multimedia items that are interesting to the general public irrespective of individual user preferences. Since our data set consists only of photos, we will use the term multimedia item and photo interchangeably without loss of generality by assuming that multimedia items have the same properties as photos, i.e. they are owned and evaluated by users and items are annotated with tags. The photos that we want to recommend in this chapter are the ones that carry broad appeal to everybody. As we shall see in the next chapter, we can incorporate this concept of universal interestingness into our other objective of finding interesting photos that are localized to specific users.

So what exactly constitutes interestingness? This is a difficult question and there is no single consensual and decisive factor that makes a photo interesting. For example, we can claim a photo is interesting by its aesthetic quality. Alternatively, we can base interestingness by theme. If a photo depicts a well-liked theme, then it is perhaps interesting. Another possibility would be to grade a photo based on feedback by previous viewers. This approach, in fact, is implemented in some online multimedia sharing applications that allow users to rate or comment each other's photos and the most reviewed photos are the ones deemed most interesting. While this approach is sound, a simple counting of votes is vulnerable to manipulation by malicious users. To encompass all these qualities into a single interestingness measure is futile as some of these criteria like aesthetic value are subjective and immeasurable. Also, the criteria themselves may not be compatible with each other. Instead, we will hone our definition on a single criterion which can be most fittingly described as popularity similar to the simple vote counting scheme but more robust. As we will see shortly, we can gauge the interestingness/popularity of a photo objectively by analyzing its association with interesting/popular users. The two types of association possible between users and photos are ownership and evaluation.

In the following sections, we introduce three popularity based methods which bear resemblance to PageRank and HITS. Each of these methods takes a different approach in assigning interestingness scores to photos. We will show that not all of them are applicable to our data set as indicated by the results of our informal experiment. As a baseline for comparison, we will use the top 100 photos with the most commentators for reference. To expedite the discussion, we will first standardize the symbols and elucidate the common intuition behind the methods.

4.1 Notation and Intuition

Our entire discussion in this chapter revolves around two vectors and two matrices. The vectors encapsulate the interestingness for photos and users while the matrices describe the relationship between photos and users. We start off with defining the matrices first. The ownership matrix B is defined as,

$$b_{ij} = 1 \text{ if photo i belongs to user j, otherwise } 0$$

The nonzero entries in each column of B indicate the photos owned by a user while the nonzero entries in each row of B indicate the owner of the photo. Since each photo can only have one owner, B is row stochastic. In other words, the sum of the row entries must equal to 1. Next, we define the evaluation matrix E as,

$$e_{ij} = \frac{1}{N_j} \text{ if user j evaluated photo i, otherwise } 0$$

where N_j is the number of photos evaluated by user j. By evaluation, we mean any one of the following actions taken by a user toward a photo such as dropping comments for a photo, rating a photo or marking a photo as favorite. Furthermore, we make no distinction between actions taken toward one's own photos or photos of others.

We surmise that users who own interesting photos know what is interesting and their knowledge of interestingness influences them to pick out interesting photos to evaluate. The emphasis here is that interesting photos are bridged by interesting users (users who know what is interesting) through ownership and evaluation, and vice versa, interesting users are bridged by interesting photos through ownership and evaluation. The relationship between interesting photos and interesting users can be modeled by the equations,

$p = Eu$	Interesting photos are evaluated by interesting users.	(4.1.1)
$u = E'p$	Interesting users evaluate interesting photos.	(4.1.2)
$p = Bu$	Interesting photos belong to interesting users.	(4.1.3)
$u = B'p$	Interesting users own interesting photos.	(4.1.4)

where p is a column vector containing the global interestingness score for photos and u is a column vector containing the interestingness score for users. We can combine any of the equations by substituting for p and u to create a PageRank/HITS type formulation, some of which are given below with their interpretation.

$p = EB'p$	Interesting photos are evaluated by owners of interesting photos.	(4.1.5)
$u = B'Eu$	Interesting users own photos evaluated by interesting users.	(4.1.6)
$p = BB'p$	Interesting photos are owned by users with interesting photos. Equivalently, interesting photos are pointed to by interesting owners and interesting owners point to interesting photos.	(4.1.7)
$p = EE'p$	Interesting photos are evaluated by users who evaluate interesting photos. Equivalently, interesting photos are pointed to by interesting evaluators and interesting evaluators point to interesting photos.	(4.1.8)

Of course, any of the resulting formulations is only a hypothesis in modeling the proposed global interestingness of users and photos. Their validity still needs to be verified by experiments. As it turns out, not all of them yield desirable results. In the sections to follow, we will address each of the above formulations and test them against our data set.

4.2 MediaRank (p = EB'p)

From a computational aspect, this equation resembles PageRank. Here, B' is a column stochastic matrix since each photo has one and only one owner and E is almost column stochastic except for some columns of zeros which correspond to users who have not reviewed any photos. We can follow the exact steps as PageRank to seek a solution. First, we convert E to be column stochastic by replacing all 0-columns with column vectors of uniform distribution. With both E and B' now column stochastic, their product EB' is necessarily column stochastic. Next, we apply "teleportation" to eliminate any zero elements in EB'. Since EB' is now primitive, we know a solution exists for p=EB'p. We can solve for the stable solution p using Power Method.

To check the validity of the model, we carried out a small experiment on our test data to rank the approximately two million photos in the collection. Each photo received an interestingness score from p which is in the range from 0 to 1. We picked the top 100 photos for evaluation. To our disappointment, the results from the test showed that the top ranked photos were rather mundane, the kind that looked like snapshots of ordinary everyday's events or somebody's family photos. The top ranked photos also received very few comments indicating that they were not popular either.

To understand the failure, we need to take a closer look at the matrix EB'. The (i, j) entry of EB' is the i^{th} row of E multiplied by the j^{th} column of B'. The i^{th} row of E contains the weights of evaluation from each user for photo i. The j^{th} column of B' indicates the owner of photo j. The (i, j) entry of EB' is nonzero only if the owner of photo j evaluated photo i and the weight received by photo i from photo j is the reciprocal of the number of photos reviewed by the owner of photo j. We illustrate this with an example.

Let $E=\begin{bmatrix} 0 & 1 \\ 0.5 & 0 \\ 0.5 & 0 \end{bmatrix}$, $B'=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$, the product $EB'=\begin{bmatrix} 0 & 1 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 0 \end{bmatrix}$ is represented in graphical form by,

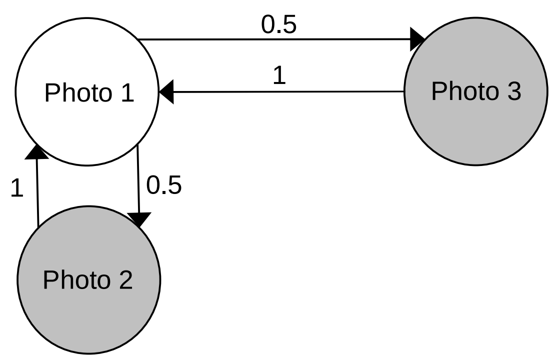

Figure 4.2.1 Graphical interpretation of matrix EB'. Shaded photos are owned by user 2 and unshaded photo is owned by user 1.

We see that the weights from photo 1 to 2 and from photo 1 to 3 are smaller than vice versa because user 1 who owns photo 1 evaluated more photos than user 2. If we interpret EB' as a Markov chain where the arrows on the above graph represent probabilities of transition, then photo 1 is more likely to be visited than photo 2 or 3. Photo 1 received higher ranking because its evaluator reviewed few photos while photo 2 and 3 received lower ranking because their evaluator reviewed many photos. The difference in the amount of evaluation contributed by users can be quite high. Besides some users being more active than others in evaluating photos, the number of evaluations also accumulates over time for older users further reducing the outbound weight of the photos owned by those users. It would be wrong to favor newer or less active users over the more senior or active ones by reducing the influence of the latter group in the recommendation decision process. For this fact alone, we can eliminate p=EB'p as a contender.

4.3 UserRank (u = B'Eu)

Our second algorithm takes an indirect approach in assigning interestingness ranking to photos. We do not rank photos directly. Instead we rank users first and from the ranked users we assign interestingness to photos through their association with users. Just like the previous section, the equation u=B'Eu resembles PageRank in form. The product B'E can be transformed into a primitive matrix which in turn guarantees a stable solution for u. The (i, j) entry of B'E is the i^{th} row of B' multiplied by the j^{th} column of E. The i^{th} row of B' contains the photos owned by user i. The j^{th} column of E contains the weighted evaluation of photos by user j. The (i, j) entry of B'E is nonzero only if any of the photos owned by user i were evaluated by user j and the actual value is the proportion of user j's evaluations directed to user i. We illustrate this with an example.

Let $E=\begin{bmatrix} 0 & \frac{1}{3} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 \end{bmatrix}$, $B'=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$, the product $B'E=\begin{bmatrix} 0 & \frac{1}{3} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{2}{3} & 0 \end{bmatrix}$ can be interpreted graphically as,

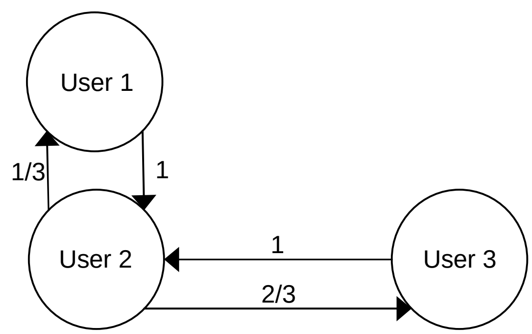

Figure 4.3.1 Graphical interpretation of matrix B'E

From the graph of B'E, we see that 1/3 of user 2's evaluations goes to user 1 while the other 2/3 goes to user 3 and in return, user 2 gets full evaluation from user 1 and 3. We can imagine the graph as depicting a competition for attention between the users. Each user is vying for the attention of the other users by submitting interesting photos. In our example, user 2 receives the most attention followed by user 3 and then user 1.

Since we surmised earlier that users with interesting photos tend to evaluate other interesting photos, we can identify interesting photos by following the ownership and evaluations of interesting users. For each photo, we obtain the ownership score and the evaluation score from the interestingness score of the users as,

$$p_{ownership} = Bu$$
$$p_{evaluation} = \hat{E}u$$

where $\hat{E}$ is the unnormalized version of E, that is, $\hat{e}_{ij} = 1$ if user j evaluated photo i, otherwise 0. The reason being we do not want to penalize active and senior users as discussed in section 4.2. There exists other variations of $\hat{E}$ that also work for our data set, for example, $\hat{e}_{ij} = 1/N_j^k$ where N_j is the number of photos evaluated by user j and k<0.5 gives acceptable results as well. The photo evaluation score $p_{evaluation}$ is the sum of the evaluators' interestingness scores weighted by $\hat{E}$. The photo ownership score $p_{ownership}$ is the interestingness score of the owner. To obtain the final ranking for a photo, we can use a convex combination of $p_{ownership}$ and $p_{evaluation}$,

$$p = \gamma \frac{p_{ownership}}{\left\| p_{ownership} \right\|_1} + (1-\gamma) \frac{p_{evaluation}}{\left\| p_{evaluation} \right\|_1} \qquad (4.3.1)$$

where γ is a real number from 0 to 1. We can either use a cutoff threshold to return interesting photos meeting some minimum score or we can pick the top M photos to return as recommendation. We can vary γ to control the significance of prominent ownership. When γ=1, the recommended photos are sorted by the popularity of the owners in descending order. When γ is small, the recommendation tends to be less dominated by any single user.

As a sanity check prior to our user study, we tested UserRank with γ=0.1 and $\hat{e}_{ij}=1$ as the parameter settings for our data set. We examined the top 100 photos with the highest interestingness scores. Compared to the photos in the previous section, the UserRank photos were much more presentable and professional looking. In terms of the commentator baseline, 35% of the photos were identical to top 100 photos with the most commentators. In fact, as we will see in the next section, many of the UserRank photos also appeared in the top photos from EigenRumor.

In addition to being able to return interesting photos, UserRank also the advantage that different “shades” of photo interestingness scores can be produced quickly by adjusting γ in Equation 4.3.1. The bulk of UserRank computation lies in solving the equation u=B'Eu for the dominant eigenvector u. Once the user interestingness scores are obtained, the interestingness scores for photos can be computed simply from Equation 4.3.1. By adjusting γ, we can produce different shades of recommendation to allow viewers the option to choose the best one.

4.4 EigenRumor (p = BB'p and p = EE'p)

The EigenRumor algorithm was originally designed for ranking blogs [5]. The blogosphere has the same general properties as an online multimedia sharing application in that blog entries are authored by a blogger and the entries may have links to the entries of other bloggers thereby forming indirect ”evaluation” links from bloggers to the entries of other bloggers. Both ownership and evaluation characteristics are present in blogosphere and the EigenRumor exploits these two properties to derive scores for bloggers and blog entries.

In the original words of the authors of EigenRumor, three scores are defined: the authority scores for bloggers, the hub scores for bloggers, and the reputation scores for blog entries. The authority score expresses to what degree the blogger contributed entries in the past that were aligned with the direction of the community. The idea is that bloggers who submit entries coinciding with the interest of the community are good authorities of the community. In terms of our notation, the authority score can be expressed as

$$a = B'r \qquad (4.4.1)$$

where vector a holds the authority scores for bloggers and vector r holds the reputation scores for blog entries. The hub score captures to what degree the blogger evaluated entries in the past that were aligned with the direction of the community. The idea is that bloggers who evaluate entries that coincide with the interest of the community are good hubs of the community. In our notation, this can be expressed as,

$$h = E'r \tag{4.4.2}$$

where vector h holds the hub scores for bloggers. Finally, the reputation score is the level of support a blog entry receives from bloggers either through ownership or evaluation. The belief is that an entry submitted by good authority or evaluated by good hubs tends to follow the direction of the community. Expressed in our notation, the reputation score is,

$$r = \alpha Ba + (1-\alpha)Eh \tag{4.4.3}$$

where α governs the weight of the authority and hub score and vector r holds the reputation scores. Replacing the authority and hub term with Equation 4.4.1 and Equation 4.4.2, respectively, the reputation score becomes,

$$r = \alpha BB'r + (1-\alpha)EE'r$$

Since BB' and EE' are both symmetric matrices, their convex combination must also be symmetric. For clarity, we rewrite the above formula as,

$$r = Sr \tag{4.4.4}$$

where $S = \alpha BB' + (1-\alpha)EE'$ and is a symmetric matrix. From Equation 4.4.4, we see that this is essentially the HITS equation and we have already seen how HITS can be solved in Chapter 3.2.

Applying EigenRumor to our problem domain is straightforward since Equation 4.4.4 is a convex combination of Equation 4.1.7 and 4.1.8 but instead of blog entries, we are dealing with multimedia items. While the intuition that guided our derivation differs somewhat, the outcome is the same.

As sanity check prior to the user study, we tested EigenRumor on our data set using the same normalization procedure for B and E as described in the original paper for blogs, that is,

$$b_{ij} = \frac{1}{\sqrt{M_j}} \text{ if photo i belongs to user j, otherwise 0}$$

$$e_{ij} = \frac{1}{\sqrt{N_j}} \text{ if user j evaluated photo i, otherwise 0}$$

where M_j and N_j is the total number of items owned and evaluated by user j, respectively. For α, we found that a small value works best for our data set so we set it to 0.1. We inspected the top 100 photos with the highest interestingness score and found them to be the same quality as the ones returned by UserRank. Comparing to the commentator baseline, 55% of the photos also appeared in the set. In terms of the top 100 photos returned by UserRank, 55% of the photos were identical between the two sets.

Chapter 5. Finding Personalized Interesting Multimedia Items

In the previous chapter, we were interested in finding globally interesting photos irrespective of individual user preferences and we examined UserRank and EigenRumor as two potential algorithms. In this chapter, we take user preferences into account and attempt to produce localized recommendation based on user preferences. For our approach, we use collaborative filtering which attempts to propose recommendation based on what other users of similar interests have liked in the past [1, 3:115]. In Chapter 2, we introduced cosine similarity as a way of measuring how close two vectors are. We also gave an example for computing the similarity between two photos by comparing the tag frequencies of the photos. We shall now apply this idea to match users with similar interests. To put it more precisely, we will match users through the tags associated with the photos they owned. The assumption is that the set of tags chosen by a user to describe her photos bears the telltale sign of her preference. To recommend interesting photos to that user, we find all other users with similar preference and return their interesting photos to the user. The way we determine interesting photos is by using either UserRank or EigenRumor. To expedite testing, we chose UserRank as the plug-in method but a more careful comparison between UserRank and EigenRumor should be conducted in the future. We should state at this point that we can also use the tags associated with the photos evaluated by users for computing user similarity, however, our own sanity check suggested that the tags of owned photos are more indicative of user preferences than the tags of evaluated photos. As usual, we begin our discussion by fixing our notation in Section 5.1 before delving into the algorithm in Section 5.2

5.1 Notation

In Chapter 4.1, we defined the ownership matrix B where

$$b_{ij} = 1 \text{ if photo i belongs to user j, otherwise } 0$$

We now define the ranked ownership matrix P as,

$$p_{ij} = \text{rank}(j) \text{ if photo j belongs to user i, otherwise } 0$$

where rank(j) is the global interestingness score for photo j obtained by any one of the working methods in Chapter 4. Basically, the ranked ownership matrix P is identical to the transpose of ownership matrix B except the 1's are replaced by the global interestingness score of the photos. We also add to our repository the tag-photo matrix D defined as,

$$d_{ij} = 1 \text{ if photo j contains tag i, otherwise } 0$$

From matrix B and D, we derive the unnormalized tag-user matrix $\hat{A}=DB$ so that,

$$\hat{a}_{ij} = \text{number of photos owned by user j with tag i}$$

Each column in $\hat{A}$ lists the tag frequencies for the photos owned by a user. For example, suppose we only have one user in our system who owns two photos so that $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $D = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The second row of the tag-photo matrix D contains all ones which indicate that the two photos share a common tag. We compute the unnormalized tag-user matrix as $\hat{A} = DB = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Our final addition is the normalized tag-user matrix A which is derived from $\hat{A}$ by normalizing each column of $\hat{A}$ with its Euclidean norm. For the example above, this produces $A = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}$.

5.2 Personalized Recommendation Algorithm

At this point, we have all the necessary artifacts to customize recommendations to individual users. First, we obtain the similarity between the users by computing the pairwise cosine similarity for the columns of A so that two users are similar if their tag usages are similar. In matrix notation, this is equivalent to computing A'A which is a symmetric matrix. Each row or column in A'A contains the cosine similarities between a user and all other users and the values are from 0 to 1, inclusively. The elements along the diagonal are the cosine similarities between a user and herself which is always 1. For our test data, A'A is a 2524x2524 matrix.

For each user, we find all similar users and recommend their photos based on the ranking of the photos which is computed by the global interestingness score of a photo weighed by the cosine similarity between the owner of the photo and the target user. A photo would obtain a high score if it is globally interesting and the owner of the photo is similar to the user. In matrix form, this is the same as computing,

$$R = A'AP \tag{5.2.1}$$

where R is the recommendation matrix. Each row in R holds the ranking for 2,177,103 photos customized to a user. For our test data, R is a 2,524 x 2,177,103 matrix since A'A is a 2,524 x 2,524 matrix and P is a 2,524 x 2,177,103 matrix. As a sanity check prior to the user study, we inspected the top 100 ranked photos recommended to a small set of randomly chosen users and found that the approach gave sensible results. For example, for users who own many photos tagged with ”cat”, a sizable portion of the recommendations composed of cat photos and the photos were of presentable quality.

While it is reasonable that we stop here and recommend photos according to Equation 5.2.1, we can further refine our algorithm by relaxing the similarity requirement between users. To motivate the need, consider the case where we have three users. User 1 owns only photos tagged with ”lizard”, user 2 owns photos tagged with ”lizard” and ”snake”, and user 3 owns only photos tagged with ”snake”. If we were to compute A'A for all three users, then user 1 and 2 are similar to each other as well as user 2 and 3 but user 1 and 3 are not similar at all. It is not wrong, of course, to only recommend user 2's photos to user 1, but it may not be a bad idea to also return some of the user 3's photos as well since there is a chance user 1 who enjoys lizards may also like snakes and user

3's snake photos might be more interesting than user 2's snake photos. Therefore, what we would like to have in the user similarity matrix A'A is that user 1 and 3 are also similar due to their common similarity to user 2, however, the strength of this similarity is less than the direct similarity between user 1 and 2 or between user 3 and 2. Taking this idea a bit further, we can extend the notion of indirect similarity to span any number of intermediate users so that any two users are indirectly similar if there exists a path of pair-wise similarity along the path of users.

We now formally redefine our recommendation matrix R to account for indirect similarity:

$$R = (c_1(A'A) + c_2(A'A)^2 + \ldots + c_n(A'A)^n)P \qquad (5.2.2)$$

where $c_1 > c_2 > \ldots > c_n$. The constant c_k dampens similarity over long distance of indirection. The parameter n represents the distance. When n=1, the formula reduces to Equation 5.2.1 scaled by an innocuous constant. Each $c_k(A'A)^k$ term in Equation 5.2.2 represents the similarity between users at k distance away. The computation of Equation 5.2.2 for large n requires enumerating powers of A'A which are time consuming especially if A'A is large. It turns out that we can avoid this if we allow the constants c_k to take on certain values. Since A'A is a symmetric matrix, we know eigenvalue decomposition exists for A'A with orthogonal eigenvectors. Let $U\Lambda U'$ be the eigenvalue decomposition for A'A with U containing the orthogonal eigenvectors of A'A, then Equation 5.2.2 becomes

$$\begin{aligned} R &= (c_1 U\Lambda U' + c_2(U\Lambda U')^2 + \ldots + c_n(U\Lambda U')^n)P \\ &= (c_1 U\Lambda U' + c_2(U\Lambda^2 U') + \ldots + c_n(U\Lambda^n U'))P \\ &= U(c_1\Lambda + c_2\Lambda^2 + \ldots + c_n\Lambda^n)U'P \\ &= U(\sum_{k=1}^{n} c_k\Lambda^k)U'P \end{aligned}$$

If we choose c_k to be $1/\mu^k$ where $\mu > \rho(\Lambda)$, in other words, c_k is less than the reciprocal of the largest eigenvalue of Λ (largest element along the main diagonal of Λ), then the sum $\sum_{k=1}^{n} \frac{\Lambda^k}{\mu^k}$ is known as the Neumann series and converges to $(I-\frac{\Lambda}{\mu})^{-1} - I$ [11:126]. Substituting back into the above formula, we obtain,

$$\begin{aligned} R &= U((I - \frac{\Lambda}{\mu})^{-1} - I)U'P \\ &= (U(I - \frac{\Lambda}{\mu})^{-1}U' - UIU')P \\ &= (U(I - \frac{\Lambda}{\mu})^{-1}U' - I)P \\ &= U(I - \frac{\Lambda}{\mu})^{-1}U'P - P \end{aligned}$$

Since we only recommend photos that are not self owned, the second term has no bearing on the final recommendations, hence, we can compact the formula to,

$$R = U(I - \frac{\Lambda}{\mu})^{-1}U'P \qquad (5.2.3)$$

Compared to Equation 5.2.2, we now only have one parameter to tune which is *μ*. We can optimize the recommendations by selecting a suitable value for *μ*. For the data set, varying *μ* from 1.01 to 100 shows a discernible difference in the recommendations for users. The smaller the *μ* is, the more homogeneous the recommendations become across the users. For larger *μ*, the recommendations are more localized to individual users. This seems to agree with the role of *μ* as a dampener for distanced similarity. After sampling the recommendations for a small set of randomly chosen users, we judged the best value for *μ* to be 2ρ(Λ). (Interestingly, in the context of bibliographic citation networks or the web where A'A represents the co-citation matrix or the authority matrix, *μ* in fact controls the weighting between the measurement of relatedness (when *μ* is large) and importance (when *μ* is small) of papers or web pages in the summation $\sum_{k=1}^{\infty} \frac{(A'A)^k}{\mu^k}$ [7].)

Before we end this chapter and ready ourselves to recommend photos to users, it is worthwhile to note that the eigenvalue decomposition of A'A is related to its singular value decomposition (SVD). From a data compression point of view, we can approximate A'A by using its truncated SVD thus reducing the number of steps needed to carry out the computation of R. Also, while beyond the scope of this thesis, it may be worthwhile to examine the structure and composition of the SVD for it may reveal some special characteristics of the data set. Finally, as an additional improvement to performance, we can pre-process the tags prior to matching users. For example, we can use a semantic lexicon like WordNet to cluster synonyms and we can also stem the tags to their root forms thereby reducing the tag space.

Chapter 6. Evaluation

So far, we relied solely on our own judgment in evaluating the recommendation produced by the global and personalized interestingness algorithms. To seek a second opinion and also as a more thorough sanity check, we asked 26 users from the data set who own at least 50 tagged and publicly shared photos to rate the top 100 recommended photos proposed to them by our algorithms. We chose UserRank as the plug-in for the personalized interestingness algorithm simply because we did not have the resources for complete test coverage. For each person, we gave 3 sets of recommendation created by varying the degree of personalization factor μ in Equation 5.2.3. The values chosen were $1.0001*\rho(\Lambda)$, $2*\rho(\Lambda)$ and $10000*\rho(\Lambda)$; in this order, the recommendations ranged from entirely global to wholly personalized. The order of which the sets were served to the volunteers was random. For each set, the volunteers rated the quality of the recommended photos by completing the following questionnaire:

Overall, on a scale of 1-5 with 5 being the best,

1. Do you find these photos interesting?
2. Are these photos related to your interests?
3. Are the photos aesthetically pleasing?
4. How would you rate the general quality of the recommendation?
   Why (optional)?

The intent of the questions was to let the testers grade the recommendations according to general interestingness, personal appeal, aesthetics and overall quality. Unfortunately, not all volunteers were able to complete the trial. Of the initial 26 volunteers, we only received completed surveys from 19 of them. Therefore, we only examined the results from the 19 respondents. We processed the scores given for each question from the completed questionnaires and calculated the means and standard deviations. For each question, we then plotted the average score and standard deviation as a function of the degree of personalization μ. In the sections to follow, we present the evaluation results for each of the questions.

6.1 General Interestingness

For Question #1 which asked the participants to evaluate the sets based on whatever interestingness criteria s/he had in mind, the results are given in the following figure.

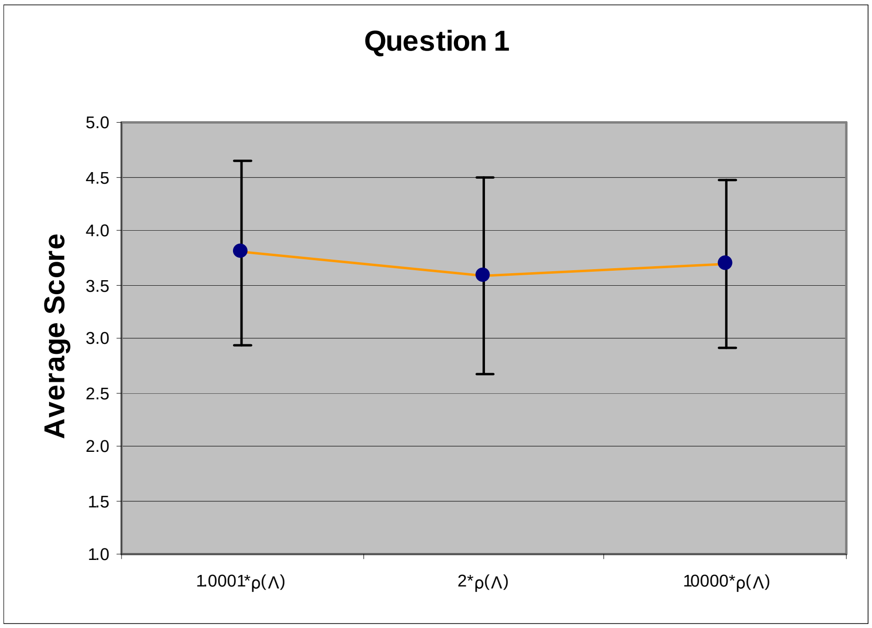

Figure 6.1 General interestingness of the recommendation. Error bars indicate 1 standard deviation in both directions. Left: global, middle: intermediate, right: personalized

For all 3 sets of recommendation, on average the testers found all of them to be somewhat interesting with the global set 1.0001*ρ(Λ) leading the score slightly. Admittedly, the sample size was too small to draw a definitive conclusion but it did seem to suggest that the photos returned were of certain quality and not merely random photos which we would expect the score to be close to 1.0.

6.2 Personal Appeal

Question #2 was more specific than Question #1 in that it asked the volunteers to rate the recommendations in direct relation to what they like or to put it more simply, by personal appeal. To succeed, the algorithm must be able to predict the preferences and needs of the participants and at the same time avoid recommending something that the users dislike. We expected the 10000*ρ(Λ) set which was the personalized set to score higher in this category than the 1.0001*ρ(Λ) set which is the global set. Unfortunately, this was not the case as indicated in the plot below.

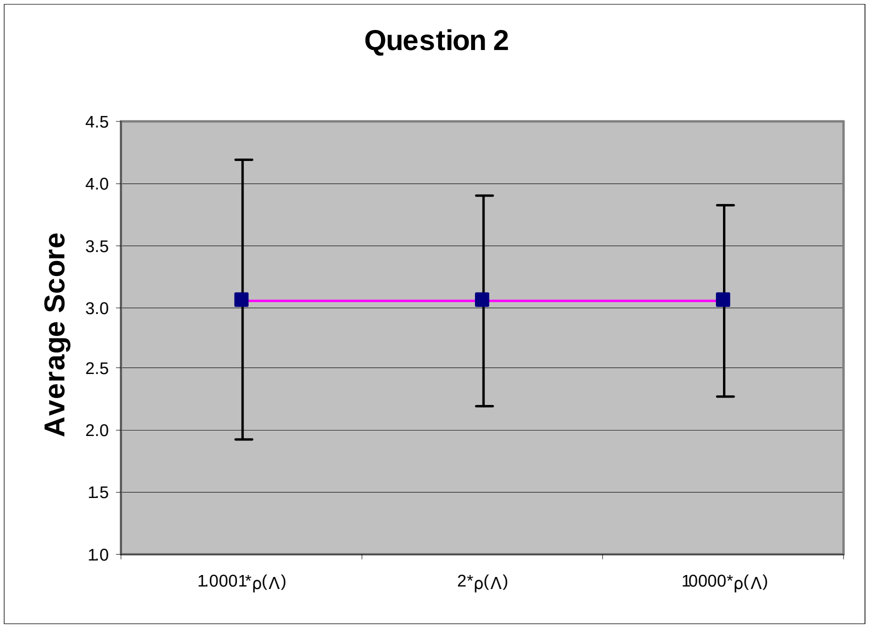

Figure 6.2 Personal appeal of the recommendation. Error bars indicate 1 standard deviation in both directions. Left: global, middle: intermediate, right: personalized

For all 3 sets of recommendation, the scores were neutral with average scores of around 3.1. While the personalized set did not perform as expected, the spread of the average score was smaller than the global set. In fact, there was a noticeable trend in the narrowing of the spread across μ from the global set to the personalized set. This indicated that the personal appeal scores assigned to the personalized set were more even across the users which implied that the personalized recommendation was less random in nature in capturing user preferences.

6.3 Aesthetics

Question #3 asked the volunteers to grade the recommendations based on aesthetics alone. As the old adage "beauty is in the eye of the beholder" goes, we left it up to the testers to decide what is beautiful. We anticipated that the global interesting set would excel in this category since the photos in the set tended to be more popular and we believe that aesthetics plays an important part in popularity. The plot for question #3 below indeed seems to suggest this.

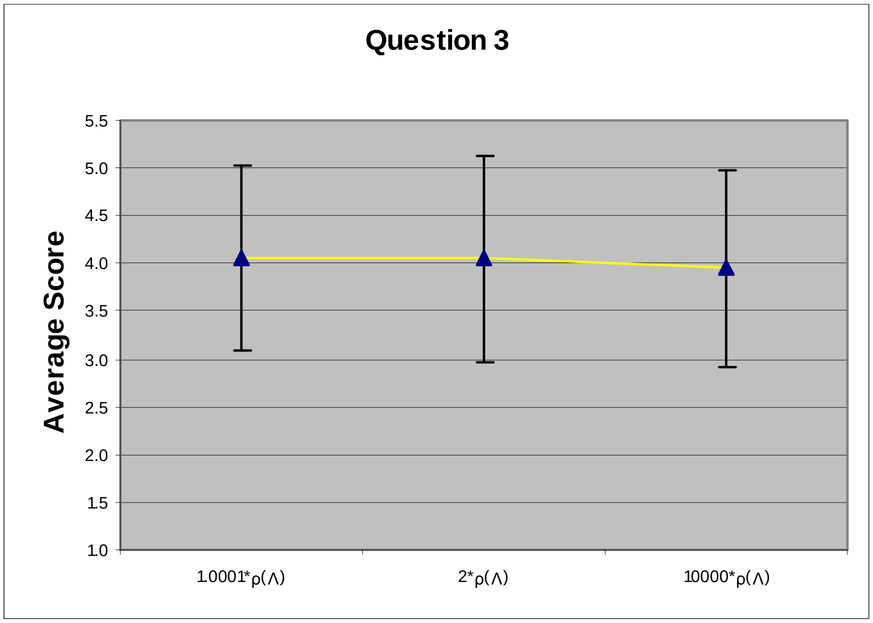

Figure 6.3 Aesthetics of the recommendation. Error bars indicate 1 standard deviation in both directions. Left: global, middle: intermediate, right: personalized

In fact, not only the global set performed well but all the other sets produced rather good results. Again, this indicated that the photos recommended were of certain caliber and not merely random photos or mundane photos of everyday lives. As a side note and to exemplify the old adage, one of the test participants deemed the photos in the global set to be "too pretty" hence gave it a neutral score of 3.

6.4 Overall Quality

Question #4 was the exit summarative question which let the testers graded the overall recommendation based on whatever factors he or she decided. Being the last question to be answered on the questionnaire, we suspected the criteria chosen by the testers were most likely influenced by the first three questions.

Our true motivation behind this question was to collect feedback from the testers on their evaluation process and what they were looking for. We obtained quite many valuable inputs this way but first we give the plot of the results below.

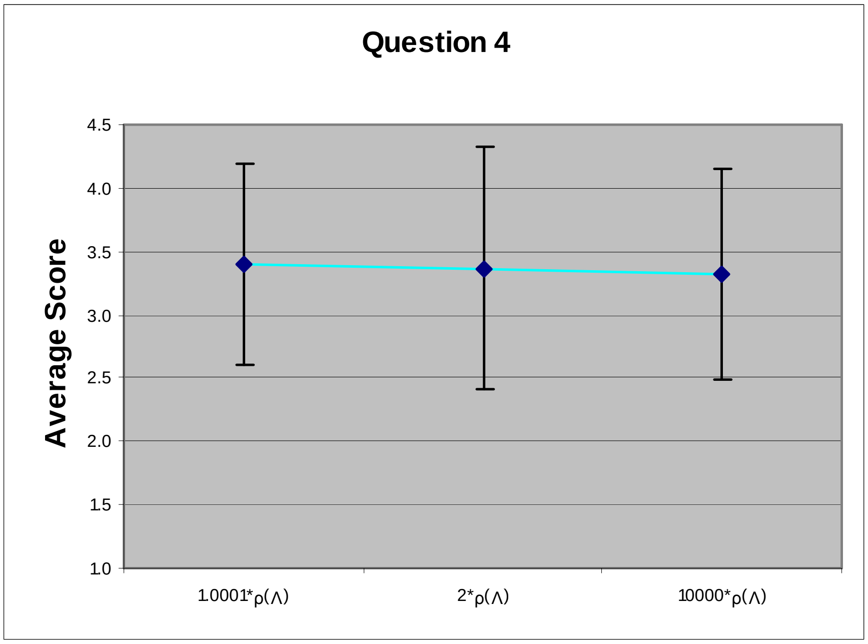

Figure 6.4 Quality of the recommendation. Error bars indicate 1 standard deviation in both directions. Left: global, middle: intermediate, right: personalized

Overall, the scores were above average for all three sets with the global set leading the pack slightly. To summarize the important points collected from the participants, variety seemed to be a very important criterion among some of the testers. They did not prefer photos that were dominated by few categories, themes or subjects. There were also apparent likes and dislikes for some testers. For example, one tester preferred dog photos over cat photos and another preferred people photos over photos of scenery. There were also praises when photos of contacts and friends were shown or when photos of places they have visited were included in the set. Also, some testers judged the photos by how much they had been altered digitally such as by using Photoshop. For this group, they tended to prefer photos that were more authentic and natural.

Chapter 7. Conclusion

In this paper, we introduced the global and personalized interestingness measures for multimedia items that could be used as part a recommendation system for an online community based multimedia sharing service. We also tested the performance of these two measures by conducting an informal user evaluation study. Our first measure of interestingness is global in the sense that it yields recommendations that are uniform across users. Such a global measure is useful when the purpose of the recommendation is to showcase and promote the best items that a site has to offer. For new visitors to the site, the globally interesting set of items offers a good first impression and entices the visitors to become members. For contributors of items, being showcased for all to see is rewarding and an encouragement to continually upload quality items to compete for the top spots. Our second measure of interestingness is about personalization. Rather than recommending only the best items, personalized interestingness takes into account of user interests and preferences by individualizing the recommendation. Personalization is important for positive user experience. In addition, personalized recommendation can be regarded as target marketing if the purpose is to promote items for sales or advertisement associated with items.

We sought the lowest hanging fruits in coming up with the global and personalized interestingness measures and their algorithms. The focus of our study was limited to analyzing the link structure of evaluation and ownership of multimedia items and tags associated with items. We made no effort in analyzing the content of the multimedia items by any image processing means. To limit the scope of the study, we used a group of 2524 users and the 2,177,103 photos owned by them from an online photo sharing service as our data set. We chose 19 of the 2524 users to rate the recommendations produced by the algorithms. While our test group was too small to draw any definitive conclusion, the results seem to suggest that both global and personalized interestingness measures were able to recommend photos of certain aesthetic quality and general interestingness to the evaluators. In terms of personalization, the personalized interestingness measure did not produce any significant results in relating the recommendation to user interests and preferences. We speculated that one of the reasons for the failure might be that the data set was too small with too few users for collaborative filtering to work. Another possible cause could be that since we only had access to publicly shared photos owned by the users, the photos that the users chose to share to the world do not necessarily reflect their true preferences.

We gathered some valuable feedback from the evaluation study that raises the possibility of improving the algorithm in the future. Variety in the recommendation was deemed quite important by the evaluators. One way to achieve this is to avoid having the recommendation set dominated by a few item owners. Another way is to select interesting items from different categories or themes. We can add a post-processing step after the recommendation algorithm to implement this objective. For the personalized interestingness algorithm, we can boost the interestingness score of items that are owned by contacts of the recipient of the recommendation. It is likely that items owned by friends are more personally interesting than items owned by strangers. Currently, the personalized recommendation algorithm relies on the tag frequencies of items owned by a user to compute similarity with other users. Unfortunately, not all users tag their items and even if all the items were tagged, the quality of the tags may be poor. To supplement the deficiency of tags, we can also add previous search terms of the users, interest group affiliation, and previous item click-throughs as a basis for computing user similarity. If we are limited to using tags only, we can still expand their coverage to include tags of items commented by the user and the tags of his/her favorite items. As a further improvement to the algorithm, we can also return not only interesting items owned by users similar to the recommendation recipient but also interesting items that were commented or marked as favorites by the similar users.

Finally, the recommendation system we proposed is passive and unsupervised in that it mines and analyzes the data of users and their interaction with items in order to make recommendation. Some evaluators in the user study expressed interest in giving interactive feedback to the system to guide the recommendation. This can be a possibility worth investigating in the future. One possible implementation would be to let users rate each recommended item individually on a Boolean scale and then use the Boolean labels along with tags for classification of user preference. By having users optionally engaged in the recommendation process, the quality of the personalized recommendation may improve significantly and at the same time, provide a richer user experience through the interaction with the system.

References

[1] Adomavicius G., Tuzhilin A., Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6,* pp. 734-749, Jun., 2005.

[2] Brin S., Page L., The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 1998.

[3] Chakrabarti S., *Mining the Web*. Morgan Kaufmann, 2003.

[4] Eldén L., *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, 2006.

[5] Fujimura K., Inoue T., Sugisaki M., The EigenRumor Algorithm for Ranking Blogs. *Workshop on the Weblogging Ecosystem*, May 2005.

[6] Golub G., Van Loan C., *Matrix Computations*. Johns Hopkins University Press, 1996.

[7] Ito T., Shimbo M., Kudo T., Matsumoto Y., Application of Kernels to Link Analysis. *In Proc. 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, pp. 586-592, 2005.

[8] Kleinberg J., Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM, 46,* 1999.

[9] Langville A., Meyer C., *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[10] Lempel R., Moran S., The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. *The Ninth International World Wide Web Conference.* ACM Press, 2000.

[11] Meyer C., *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[12] Ng A., Zheng A., Jordan M., "Stable Algorithms for Link Analysis".

Proceedings of the 24th Annual International ACM SIGIR Conference, ACM, 2001.

[13] Tomlin J., A New Paradigm for Ranking Pages on the World Wide Web. *The Twelfth International World Wide Web Conference*. ACM Press, 2003.

[14] Trefethen L., Bau D., *Numerical Linear Algebra*. SIAM, 1997.

[15] US patent application 20060242139, “Interestingness ranking of media objects”, published 2006-02-08.

[16] US patent application 20060242178, “Media object metadata association and ranking”, published 2006-02-08.