DATA MINING - CLASS 3

1. Google and Hits - continue

1.1. Reminder: Terms from previous lesson.

Definition 1.1. Thin SVD - A matrix M_{m*n} with rank(M)=k, can be represented as a product of 3 matrices: U_K, Σ_K, V_K^T :

$$M = U_K \cdot \Sigma_K \cdot V_K^T$$

- **Matrices dimensions:**
  U_K**:** m*k
  Σ_K**:** k*k
  V_K^T**:** k*n
- Σ **Properties:**
  (1) Σ is a diagonal matrix:
  $$\begin{pmatrix} \sigma_1 & & \bigcirc \\ & \ddots & \\ \bigcirc & & \sigma_k \end{pmatrix}$$
  (2) The values $\sigma_1 \ldots \sigma_n$ are called **"M's Singular values"**
- U_K, V_K **Properties:** U_K, V_K are orthonormal matrices, meaning:
  (1) Their columns form a basis of $\mathbb{R}^K$
  (2) The Norm of each column is 1

1.2. Google Vs. Hits.

The following table sums up how Google and Hits calculate their hyper-links based rank:

Google	**Hits**
$Q(q) \prec \sum_{p:p\rightarrow q} \frac{Q(p)}{OutDegree(p)}$	$A(q) \prec \sum_{p:p\rightarrow q} H(p)$
	$H(p) \prec \sum_{q:p\rightarrow q} A(q)$

Notation:

A(p): Authority rank of page p- How good are p's references
H(p): Hub rank of page p- How good are the authorities pointed by p
Q(p): Google's PageRank of page p- How good are the pages that point to p

1.3. The connection between singular values and eigenvalues.

Lemma 1.2. *MM^T is a symmetric matrix*

Proof.
Fact From Linear algebra: $(AB)^T = B^T A^T$
$\Longrightarrow (MM^T)^T = M^{T^T} M^T = MM^T$ □

Remark 1.3.

Fact 1: Symmetric matrix has the same eigenvalues from both right and left
Fact 2: MM^T dimensions are m*m, $M^T M$ dimensions are n*n

Lemma 1.4. *$\sigma_1^2 \dots \sigma_k^2$ are the eigenvalues of MM^T, and of $M^T M$*

Proof.

$$MM^T = U_K \Sigma_K \overbrace{V_K^T V_K} \Sigma_K^T U_K^T = U_K \overbrace{\Sigma_K \Sigma_K^T} U_K^T = U_K \Sigma_K^2 U_K^T$$

We used here orthonormality of V_K columns (therefore $V_K^T V_K = I$), and Σ_K being a diagonal matrix.

Now, let's take the i column of U_K, and prove that it is a left eigenvector of MM^T, with eigenvalue σ_i^2:

$$\vec{u}_i^T MM^T = \overbrace{\vec{u}_i U_K} \Sigma_K^2 U_K^T = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \overbrace{\Sigma_K^2 U_K^T} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} \sigma_1^2 * \vec{u}_1^T \\ \sigma_2^2 * \vec{u}_2^T \\ \vdots \\ \vdots \\ \sigma_k^2 * \vec{u}_k^T \end{pmatrix} = \sigma_i^2 * \vec{u}_2^T = \vec{u}_2^T * \sigma_i^2$$

We see that for each $i \leq k$, σ_i^2 is an eigenvalue of MM^T with the eigenvector $\vec{u}_i^T$. Since MM^T is symmetric, σ_i^2 is also right eigenvalue of the matrix.

□

That's why many times articles will talk about eigenvalues and eigenvectors of MM^T instead of signular values singular vectors of M

Remark 1.5.

Fact: The 1st singular vector $\vec{v}_1$ is all-positive

1.4. **Another way to look at singular values.**

There is another way to look at M's singular values:

Let's look at the following matrix:

$$\begin{pmatrix} \bigcirc & M^T \\ M & \bigcirc \end{pmatrix}$$

This is a square matrix- it's dimensions are (m+n)*(m+n). What are it's eigenvalues?

It's easy to see that for each $i \le k$:

$$\begin{pmatrix} \bigcirc & M^T \\ M & \bigcirc \end{pmatrix} \begin{pmatrix} \vec{v}_i \\ \vec{u}_i \end{pmatrix} = \begin{pmatrix} M^T\vec{u}_i \\ M\vec{v}_i \end{pmatrix}$$

Now, since:

$$M^T\vec{u}_i = (U_K\Sigma_K V_K^T)^T\vec{u}_i = V_K\Sigma_K U_K^T\vec{u}_i = V_K\Sigma_K \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \vec{v}_i * \sigma_i$$

and since:

$$M\vec{v}_i = U_K\Sigma_K V_K^T\vec{v}_i = U_K\Sigma_K \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \vec{u}_i * \sigma_i$$

We receive that:

$$\begin{pmatrix} \bigcirc & M^T \\ M & \bigcirc \end{pmatrix} \begin{pmatrix} \vec{v}_i \\ \vec{u}_i \end{pmatrix} = \sigma_i \begin{pmatrix} \vec{v}_i \\ \vec{u}_i \end{pmatrix}$$

Meaning that σ_i are the eigenvalues of the above matrix

In the same way, we receive that $-\sigma_i$ are eigenvalues of the above matrix, with eigenvectors $\begin{pmatrix} \vec{v}_i \\ -\vec{u}_i \end{pmatrix}$.

1.5. **Hits algorithm** .

As we mentioned before, authority is defined by hub, and hub is defined by authority ($\prec$ means that the values are normalized somehow after they're calculated):

$$A(q) \prec \sum_{p:p\to q} H(p), \quad H(p) \prec \sum_{q:p\to q} A(q)$$

Such a relation is called: "mutually reinforcing relationship". How do we solve it?

We will represent the pages-hyperlinks graph as a simple adjacency matrix M, where:

$$M(p,q) = 1 \Longleftrightarrow p \to q$$

Now, if we mark the Authority of all the nodes as a vector $\vec{a}$, and the Hub as vector $\vec{h}$, we may represent the above formula as:

$$\vec{a} = \frac{M^T\vec{h}}{\|M^T\vec{h}\|}, \quad \vec{h} = \frac{M\vec{a}}{\|M\vec{a}\|}$$

Assigning $\vec{h}$ leads to:

$$\vec{a} = \frac{M^T M\vec{a}}{\| M^T M\vec{a} \|}$$

Assuming $\| M^T M\vec{a} \| \neq 0$, we receive that:

$$M^T M\vec{a} = \| M^T M\vec{a} \| \, \vec{a}$$

Meaning that $\vec{a}$ is an eigenvector of $M^T M$, or in other words- $\vec{a}$ is a right singular vector of M (One of the columns of V_k). In the same way we receive that $\vec{h}$ is one of the columns of U_k.

The question now is, which of the vectors $\vec{v_1}, \ldots \vec{v_n}$ is the $\vec{a}$ we seek? All of them fit the equality, but which one is the real authority vector?

Although all of the $\vec{v_1}, \ldots \vec{v_n}$ vectors fit the equality, some of them might contain negative values. That is not good for us, since we defined authority to be at the range $0 \ldots 1$. The only vector that can be assured to be all-positive is, as we said, $\vec{v}_1$.

Conclusion:

$$\vec{a} = \vec{v}_1$$

$$\vec{h} = \vec{u}_1$$

2. LATENT SEMANTIC ANALYSIS

2.1. Bag of words model .

Let's define a binary vector that contain an entry for each word of the English language.

Representation of a document as "bag of words": Each entry contain 1 if the word exists in the document, and 0 if not:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & . & . & . & . & 1 \end{pmatrix}$$

This simple representation is useful, although:

- It loses the order of words in the document
- It ignores the number of occurrences of each word in the document

The most basic classification algorithm: Similarity between two documents is measured through the cosine of the angle between their vector representations (Small angle $\Rightarrow$ high similarity):

$$Similarity = cos(\theta) = \frac{(\vec{v}, \vec{u})}{|\vec{v}||\vec{u}|}$$

Notice that both vectors are very sparse. Therefore, usually the multiplication will be close to zero.

This classification method has at least two major problems:

1. **Synonymy:** Different words may have the same meaning. For example: Car and Automobile.
2. **Polysemy:** A word may have more than one meaning. For Example: Jaguar can be either a car or an animal.

Our goal will be to solve the synonymy problem. We will start by representing the problem as a matrix A of words and documents:

$$\begin{array}{cc} & \textbf{m} - \textbf{Documents} \\ \textbf{n} - \textbf{Words} & \begin{pmatrix} 0 & 1 & . & . & 0 \\ 1 & 0 & . & . & 0 \\ 0 & 1 & . & . & 1 \\ . & . & . & . & . \\ 1 & 1 & . & . & 0 \end{pmatrix} \end{array}$$

In order to try to solve the problem of synonymy, we will reduce the dimension of the matrix, trying to merge rows that represent words with similar meaning.

A procedure of SVD on A, assuming that rank(A)= K, will give us the following decomposition:

$$A = U_k \Sigma_k V_k^T$$

For reducing the dimension of A, we will take i < K (Usually we will take i $\ll$ k) , and define a matrix A_i in the following way:

$$A_i = U_i \Sigma_i V_i^T$$

Where:

- U_i**:** Received by taking the first i columns of U. dim(U_i)=m*i
- Σ_i**:** Received by taking the upper-left i*i sub-matrix of Σ. dim(Σ_i)=i*i
- V_i^T**:** Received by taking the first i rows of V^T. dim(Σ_i)=i*n

What is $U_i^T A_i$? it is the casting of A_i on the basis of $\mathbb{R}^i$ spanned by $U_i columns$. We narrow the matrix A from m*n to i*n, and this way we eliminate some of the noise.

Now we want to expend again the reduced matrix to its original dimensions, by multiplying from the left by U_i:

$$U_i U_i^T A_i$$

This method solves the problem of synonymy: In the above example, reducing the dimension would add the word "automobile" to the word "car". Expending the dimensions again would restore the words to their location. Now, searching "automobile", will also give results of "car", because many of the other terms related to the topic appear with both words.

Definition: **<u>Uniform matrix</u>-** A matrix where each cell contain 0 or 1 with probability $\frac{1}{2}$.

As an example to the method above, we will take 2 unrelated terms: cars and sails. We assume that documents that discuss cars, usually don't discuss sails. We also assume that within each category the terms form a uniform matrix:

$$\begin{array}{l} \\ \textit{Cars terminology} \\ \\ \textit{sails terminology} \end{array} \left(\begin{array}{cc} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}}_{Uniform-100*100000} & \\ & \underbrace{\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}}_{Uniform-100*100000} \end{array} \right)$$

Now we will search a matrix A' , with rank(A')=2, which is as close as possible to A:

$$A' = U_2 U_2^T A$$

What is U_2? it is a matrix composed of 2 spanning vectors:

(1) 1 at the terminology of cars, and 0 at all other cells
(2) 1 at the terminology of sails, and 0 at all other cells

$$\begin{array}{cc} \textit{Cars} & \textit{Sails} \\ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \end{array}$$

If there are terms which are common to both types of documents, they will be zeroed. Because of the uniform nature of both parts of the matrix, and because of the multiplication, we will receive 2 blocks of $\frac{1}{2}$'s:

$$U_2 U_2^T A = A' = \begin{pmatrix} 1/2 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 & 1/2 \end{pmatrix}$$