COS 493: Algorithms on Massive Data Sets 04/25/02

Lecture 22

Professor Moses Charikar *Scribe: Shubha Nabar*

In the last class we went over the Hubs and Authorities algorithm for page ranking and showed how these calculations could be related to linear algebraic calculations. We talked about the Singular Value Decomposition (SVD) technique, where we showed that any $m \times n$ matrix, A, could be decomposed as follows -

$$A_{m\times n} = U\Sigma V^T, \text{ where } \Sigma \text{ is a diagonal matrix.}$$

SVD is a useful tool used to identify hidden connections. For instance, it could be used to generate low rank approximations to matrices.

Matrix Approximation

Let x be a vector. Then the norm (size) of x, denoted by $\|x\|$, can be measured in many ways. For instance,

$$\|x\|_2 = (\Sigma x_i^2)^{1/2}$$

If A is a matrix, then the norm of A is defined as

$$\|A\| = \max_{\|x\|=1} \|A \cdot x\|$$

Let A be the matrix that we want to approximate by a matrix B of rank k, i.e. we want $\|A-B\|$ to be as small as possible over all matrices B of rank k. We can think of A as some sort of an underlying structure with noise. Then B would identify the underlying structure and $A-B$ would represent the noise. Using SVD, we can represent A as $U\Sigma V^T$, i.e.

$$\begin{bmatrix} & \boxed{u_i} & \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} \boxed{\quad v_i \quad} \end{bmatrix}$$

$$\text{Thus } A = \sum \sigma_i u_i v_i^T, \text{where } \begin{array}{l} \|u_i\| = 1, u_i \cdot u_j = 0 \\ \|v_i\| = 1, v_i \cdot v_j = 0 \end{array}$$

Assume $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_n$. A rank k approximation of the matrix A would then be given by

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Then $\|A - A_k\| = \min_{B \text{ of rank } k} \|A - B\|$. This is true for all different measures of the norm.

Latent Semantic Indexing

Consider a set of documents containing several terms. We can construct a matrix, A, the columns of which represent the documents and the rows of which represent the terms present in the documents. Thus a_{ij} would be 1 if term i were present in document j, and 0 otherwise. Two documents would be considered similar if the dot product of the corresponding column vectors were high. Similarly, two terms would be considered similar if the dot product of the corresponding row vectors were high. This method however, does not work so well. This is because you somehow want to weight the occurences by the similarity of the documents they occur in as well.

A better measure of the similarity of documents and terms can be obtained by taking a rank k approximation of A, and using this as the matrix of document and term vectors. Conceptually, this is like taking each document vector and projecting it onto k dimensions (hopefully the most important k dimensions). Each of the k dimensions is a cluster of terms, and A_k gives the weight of the documents on each cluster. Taking the SVD gets rid of the noise. Recently, attempts have been made to theoretically justify the success of SVD analysis.

Papadimitriou, Raghavan, Tamaki and Vempala in 1998, considered a specific model according to which documents are generated. According to this model there are k fundamental hidden topics. There are T_i terms associated with each topic i, and $T_i \cap T_j = \emptyset, \; for \; i \neq j$. Each document draws at random from these sets of terms. They showed that SVD would identify the k fundamental topics.

In the same paper, they then considered another model for generating the documents. In this model they assumed that the vocabulary sets of different topics could intersect, but each topic would have a core set of terms that would be disjoint from the core sets of terms of other topics. Assume T_i' is the core set of terms for document i. Then $T_i' \cap T_j' = \emptyset$

Let $|T_i'| \geq (1-\epsilon)|T_i|$. Let v_i be the document vector for document i obtained after doing the SVD analysis. They showed that

$$v_i \cdot v_j \geq 1 - O(\epsilon) \; if \; i \; and \; j \; are \; on \; the \; same \; topic, \; and$$

$$v_i \cdot v_j \leq O(\epsilon) \; if \; i \; and \; j \; are \; not \; on \; the \; same \; topic.$$

In a more recent paper, Azar, Fiat, Karlin, McShery and Saia (in 2001) proved that if entry a_{ij} in matrix A represented the probability that term i occured in document j, and if A was of small rank, then the SVD analysis would be successful.

All the techniques thus far have been based on the assumption that some distribution produces the documents and terms.

Web Search Via Hub Synthesis - Achlioptes, Fiat, Karlin, McShary

Suppose there exist k fundamental topics. Every web page, p, has a hub vector, H_p associated with it, that says how good a hub it is in each of these topics. Likewise, it also has an authority vector, A_p that says how good an authority it is in each of these topics.

$$H_p = (h_1 \; h_2 \ldots h_k)$$

$$A_p = (a_1 \; a_2 \ldots a_k)$$

We can observe the links between pages and the vocabulary of the pages. If p is a hub on a particular topic and q is an authority on the same topic, $prob(link\ from\ p\ to\ q) = H_p \cdot A_q$. We also assume that there is a hub vocabulary and an authority vocabulary for each topic. After the query is generated, we can obtain the query vector, Q, which gives a certain weight for each topic. Then $Q \cdot A_p$ would give the ranking of the page.

In reality, we do not actually have these vectors, but can come up with a spectral analysis that takes the query terms and tries to infer the query vector. This is similar to the SVD analysis.

Another use of Latent Semantic Indexing (LSI) is in Cross-Lingual IR. In general, LSI is useful when you have hidden connections between terms and documents that you are trying to find out.