Seminar talk:

Page 1

Welcome to my seminar talk, my name is Vahe Eminyan and my seminar topic is Latent Semantic Indexing

Page 2

This is the overview of my presentation

At the beginning there is an introduction, and then there is a background section. Then we will look at the original paper and our emphasised aspect.

Then LSI by random projection the and we will summarise the topic and draw a conclusion

Page 3

We start with the motivation.

Nowadays large datasets are organized in tabular form and represented as matrices.

For example, if we have a huge dataset of document we can represent it with help of term document matrix. Or if we consider the dataset of a film company, we can represent the relation of movies and users in a matrix.

For example, look at this table. We have m documents and a vocabulary with n words (stop word removal is done). We represent it as an n by m matrix. Where the entry ij represents if the term i occurs in document j.

Interesting aspects to investigate are:

Find documents semantically associated with a query

OR to recommend a new movie to a user.

Page 4

To solve such kind of problems there are many different methods and one of them is the so called Latent Semantic Indexing. The name of it comes from the analysis of textual data. It finds the latent (hidden) semantic structure of the data.

And the rough idea is to represent the original term-document matrix as a product of three matrices. This process is based on the singular value decomposition of a matrix.

Page 5

Now let me introduce the singular value decomposition of a matrix.  
Every n by m matrix can be represented as the product of three matrices. U times D times V transposed.

U is a column orthonormal matrix: its columns are called left singular vectors of matrix A

V is a column orthonormal matrix. Its columns are the right singular vectors of A

And D is a diagonal matrix. The diagonal elements of the matrix are called singular values of the matrix A and are ordered in decreasing order. (They are nonnegative and greater than 0).

Page 6

Now lets consider an example of SVD for a rank 3 matrix. A term document matrix is represented a product of three matrices. The matrix U can be interpreted as the term-topic similarity matrix, the matix V as the topic-document similarity matrix. And the diagonal elements of matrix D show the strength “importance” of each topic.

Page 7

LSI considers the matrix A\_k the rank k approximation of matrix A. We keep the first k columns of U and V and the first k singular values.

Here the A\_k approximation for the example of the last slide.

To find the most relevant documents for a query we map the query with the matrix U\_k to lower dimensional space and then apply cosine similarity to fund the similar documents in D\_k times V\_k transposed.

Page 8

There is a famous theorem from Eckart and Young that states:

Among all n by m matrices C of rank at most k, A\_k is the one that minimizes the squared value of the Frobenius norm of the matrix A minus C.

And from the example we can see that the both matrices A and A\_k are similar to each other.

Page 9

LSI has shown strong empirical results.

There are two important aspects.

1. Why does LSI find semantically related documents
2. How can we reduce the computational complexity

In the scientific paper on which my seminar is based on . Papadimitrou et al. investigated both of the questions:

1. They have shown that, under certain constraints on the term-document matrix, semantically related documents are mapped to similar vectors
2. nstead of LSI use LSI by random projection.This reduces the computational complexity
   1. First we map the original term-document matrix into a lower dimensional space
   2. Then use LSI on the lower dimensional matrix

In this presentation, we focus on the second aspect

Page 10

First let us understand the dimensionality reduction of a matrix using random projection.

For a given n by m matrix A we use another matrix R of dimensions l by n to reduce the dimensionality of A.

By preserving the pairwise distances between any two points.

We call the the lower dimensional matrix B is defined as R transposed times A and scaled with square root of n divided by l. B is now an ell times m matrix. So we basically reduce the number of rows.

Regarding the random projection there is a Lemma from Johnson and Lindenstrass that states the following:

Let v be a unit vector and H be a random l dimensional subspace through the origin, the random varible X denote the square of the length of the projection of v onto H.

Suppose epsilon is in the range (0, 0.5) And suppose 24 log n < 1 < sqrt(n). Then the expected value of X is l/n, and

The probability that the absolute difference of X and l / n larger is than epsilon times l/n. Is smaller than 2 times this term. (Informally said for the large l this probability is very low because l is in the negative exponent).

Page 11

After understanding how the random projection works we now move to the LSI by random projection.

1. First we apply a random projection on A. We project each column of A into l dimensional space where l is a small value greater than k.
2. Then apply rank O(k) LSI on B. (because of the random projection, the number of singular values kept may have to be slightly increased)

This leads to an improvement of the computational complexity. Furthermore, Papadimitriou et al have proven a theorem that the original matrix A after applying random projection and then LSI is almost as good recovered as by directly using LSI.

Before moving to the formal formulation of the theorem and its proof we introduce some important background knowledge and notations.

Page 12

These are our vector notations for 4 matricers we need in the proof.

First one is the original term document matrix

A\_k is the rank k approximation of A

B is the matrix a after random projection and scaling

B\_2k is the 2k approximation of the matrix A.

Page 13

Then we continue with the following lemma.

Let epsilon be a positive constant. If \ell is larger than this value for a sufficiently large constant c, then for p = 1,…,\ell. we have the following property:

For each singular value of the matrix B. The squared value of this singular value is greater than or equal to this term.

A corollary of that is the following:

The summation of the squared value of the first 2k singular values of B. Is larger then or equal to (1-epsilon) times squared frobenius norm of A\_k.

Page 14

Then we introduce another important lemma which we will use in the proof of the main theorem.

It states that the squared frobenis norm of the difference of A and A\_k is equal

Sum of the squared values of k +1th till nth singular values of A.

And the last thing we need is the theorem called Parsevals identity which states.  
If b1,,,bn is an orthonormal basis for a space S. Then for each vector in that space We have the following property: The squared length of that vector can be written as summation of s times b\_i squared for all i in i =1 to n.

Page 15

Finally we introduce the main theorem of this presentation , suggested from Papadimitriou et al. WE have the following:

Squared frobenius norm of the difference of A and B\_2k is less than or equal to the sum of squared frobenius norm of the difference of A and A\_k plus 2 epsilon times the squared frobenius norm of A.

Informally, the theorem states that the original matrix A after applying random projection and then LSI is

almost as good recovered as by using one-step LSI on the original matrix.

Page 16

Now lets prove this theorem. Here on the right top corner of every slide I put the theorem we want to proof.

Just remember we have the matrices A, A\_k B and B\_2k and their vector notations.

B1 till bn are orthonormal vectors spanning the row space of A and B2k.

Hence using the Parsevals identity we can write that the squared frobenius norm of the difference of A and B\_2k as follows.

Then for the right side for each element the A minus B\_2k times b\_i is equal to 0 for i in range i = 1, ….2k. Because the dot product of b with itself is 1.

And with all others is zero hence for i equal 2k +1 till n we have (a -B2k) b\_i = Atimes B\_i.

Page 17

Now we continue the proof. The first line is from the last slide. Then we rewrite it as

Summation squared lengths of vectors A times b\_i for I in 2k+1 till n. Then we write it as the difference of two terms. Once the summation over all n and once the summation form I = 1 till 2k.

And using the parsvals identity for the last line we get the following. Basically the left term is the squared Frobenius norm of the matrix A.

Page 18

On the other hand we consider the sqared frobenius norm of the difference of A and A\_k.

Using Lemma 5 we can rewrite it as the sum of the squared singular values from k+1 to n. Then using the definition of the frobenius norm we can write it as squared frobenius norm if A minus squared frobenius norm of A\_k.

Page 19

Now we consider the difference between squared frobenisus norm of A -B2k and A minus A\_k.

Based on the last steps we can rewrite it as squared frobenius norm of A\_k minus this summation.

This is equivalent to the following term : We just bring the squared frobenius norm of the difference of A -A\_k to the right and simplify.

Now look at the theorem. We almos have the form we want Just need to show that the right part of this summation is less then or equal to 2 epsilon times squared frobenius norm of A.

Page 20

For the next step first we show that (1 + epsilon) times this summatuion is grater than or equal to summation of squared values of the singular values of B.

Instead of the singular values we write B times b\_i. And then substitute instead of B its definition with A. Then we pull out the square root of n ivided by l getting this term.

Page 21.

Now from Johnson – Lindenstrauss lemma we get that for very large \ell the following inequality for fixed I with high probabaility.

Hence with high probability we have the (1 + epsilon) times this summation is grater than or equal to summation of squared singular values of B for I = 1 to 2k.

Page 22

Then we devide both sides by (1 + epsilon) and using corollary 4 we get that the it is greater than or equal to (1-epsilon) divided (1+epsilon) times square frobenius norm of A\_k. And this is greater than or equal to 1-2espilon times sqared frobenius norm of A\_k.

Page 23

Now remember the equation 12. We had this equality and in order to show the theorem we had to show that the right part of summation is less then or equal to 2 epsilon times squared frobenius norm of A.

Now we substitute the result of erquation 22 into equation 12. This means instead of the sum of squared lengths of A times b\_i we use use (1-12espilon) times squared frobenis norm of A\_k.

And then simplify it.

the frobenius norm of a matrix is always larger than or equal than the frobenius norm of its rank k approximation. Hence we get our final result, which we wanted to proof.

Page 24

Now lets compare the computational complexity of both methods.

Given a term-document matrix A of sitze n times m.

The time complexity of the one-step LSI is in O of m times n times c if A is sparse matrix with about c nonzero entries per column.

The time complexity of two step LSI, i.e LSI by rndom projection can accumulated from the following steps:

The random projection into \ell dimension is in O(mcl).

LSI computation is in O(Ml^2)

Together the complexity is in O(m(cl + l^2)) with l in O ( log n divided by epsilon squared).

Hence the total time complexity is in O ( m times (squared value of logarithm n + c times logarithm n)

This is better complexity than O(mnc) because of the logarithm.

Page 25

In this slide I would like to summarize the entire topic and draw a conclusion.

We have analyzed the LSI an SVD based matemetaical teqnique for information retriewal.

Papadimitriou et al. analyzed two important aspects:

Why does find semantically related documents and how to reduce the computational complexity. (This was our main focus).

They have shown that the LSI by random projection leads to a reduction of the computational time , while preventing the expressivnesss of the original matrix.

There are newer teqniques for information retrieval which are based on neural networks like (Graph neural networks etc.)