Introduction to Data Mining

Summary of Lecture 4

Prof. Amos Fiat

Fall 2002

Scribe by Nisim Peled and Dina Dushnik

In this paper we will discuss two subjects:

- Latent semantic analysis - in continuation of the last lecture [1].
- The probably approximately correct learning model - presented by the rectangle learning game example [2].

1. Latent Semantic Analysis

1.1 Introduction

Spectral techniques have proven to be useful in a variety of data mining applications. In this paper we will present the use of *Latent Semantic Analysis* for data mining algorithms.
The data will be presented by the *Data Generative Model* under the assumption that the data of interest can be represented as an object/attribute matrix. This model is described at section 1.2.
On the input matrix A we will use a *thin-SVD* (singular value decomposition) as follows:

$$A = U\Sigma V^T$$

We keep only the first K singular values and denote the new matrix as A_k:

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

For more details on *thin-SVD* see lecture number 3 notes.
The goal of this process is to get information about the "true" data. By drooping the singular values, which are smaller then K, we in fact get rid of the "noise". A justification to this process will be given at section 1.3.

1.2. The Data Generative Model

We assume that there is a probability matrix A, where A_{ij} describes the probability of the i-th attribute of the j-th object and $0 < A_{ij} < 1$.
For example we can describe the probability that a word will appear in a document. Each document is represented by a vector V of k fundamental terms, where V_i is the weight that represent the connection between the documents and the term (or subject). In the same manner we also describe the words. By multiplying the words and the documents vectors we will get a matrix A, where A_{ij} is the probability that the word i appears in document j, denoted by P_{ij}.
The actual values of each attribute is then obtained by sampling from the distribution A. We denote the resulting values by $\tilde{A}$. For example if the probability that a word i appears in document j is P_{ij}, we will have $A_{ij} = P_{ij}$. Then we obtain the actual values as follows:

$$\tilde{A}_{ij} = 1 \text{ in probability } P_{ij}$$
$$\tilde{A}_{ij} = 0 \text{ in probability } 1 - P_{ij}$$

Where $\tilde{A}_{ij} = 1$ means the word i appears in document j and $\tilde{A}_{ij} = 0$ means it is not. See figure 1.

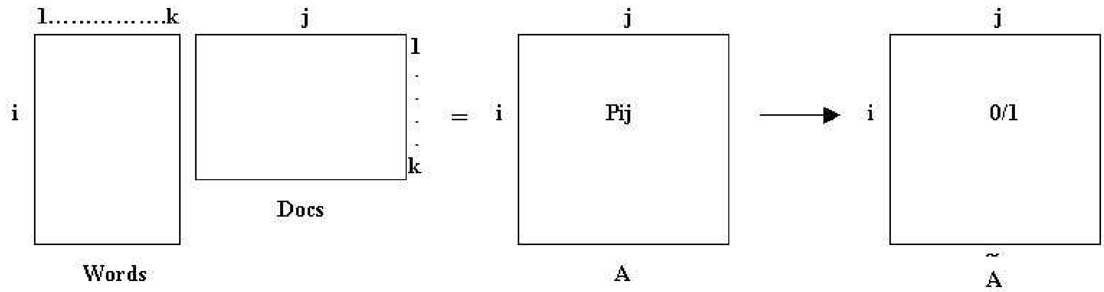

Figure 1: The process of calculating $\tilde{A}$

In this process we add "noise" to A and obtain $\tilde{A}$. The "noise" is denote by an error matrix E. E can be calculated as follows:

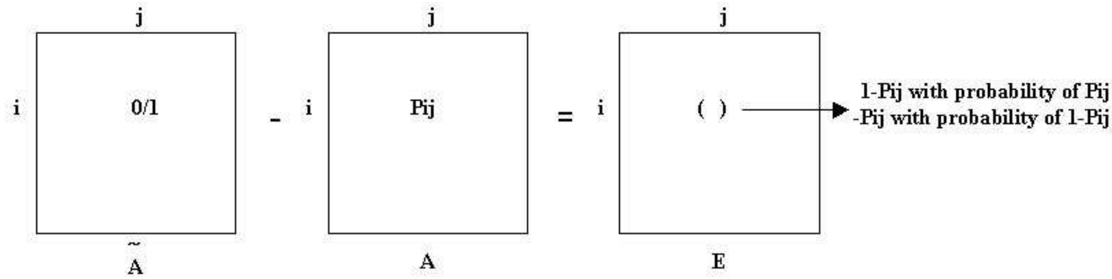

Figure 2: The error matrix E

From that we can conclude that:

$$\tilde{A} = E + A$$

As was described at the introduction, if we keep only the first K singular values of the *thin-SVD* of $\tilde{A}$, we would get $\tilde{A}_k$ and by that get rid of the "noise" (or E). This will be explained at the next section.

1.3 Justification of the process

The basic assumption of this model is that it is enough to use a low ranked matrix A, from rank k, in order to describe the data "well". In fact this is a philosophically argument. It is claimed that people, products, documents, terms, etc., are all inherently determined by or associated with a small number of fundamental properties, where each individual person, product, word, etc., can be described by a weight vector of these base properties. Thus, the justification that A can be from a low rank.
Now we have to justify the way we choose the K vectors from $\tilde{A}$ in order to obtain meaningful information about A:
Let $\tilde{A} = A + E$ where A is a low rank matrix, from rank K and E is a new matrix, whose entries are independent random variables with probability of +-1/2. We will perform *thin-SVD* on A and get:

$$A = U\Sigma V^T$$

The norm of the random matrix E is $\|E\|_2 = \sqrt{n}$. The k singular value of A is $U(\sqrt{n})$. This means that in order to get rid of the influence of E we have to drop the singular values of $\tilde{A}$ which are smaller than K. This can be done by performing *thin-SVD* on $\tilde{A}$:

$$\tilde{A} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$$

And choosing the first K singular values:

$$\tilde{A}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^T$$

In conclusion we can see that this technique allows us to get a clear view of the "true" data whiteout any information on the "noise" or the error.

2. The Probably Approximately Correct Learning Model

2.1 A Rectangle Learning Game Definitions

we will present the idea of the probably approximately correct learning model by the rectangle learning game example.
Consider a one-player learning game:

- The object of the game is to learn an unknown axis-align rectangle R (his sides are parallel to the coordinate axes). See figure 3.
- The player gets information about R (location/size) just through the following process:
  1. Each time the player press the button a random point is chosen in the plane according to some fixed probability distribution D.
  2. The player is given the point with a label which determine if the point is inside R (denote by +) or outside (denote by -).
- The goal of the player is to find a rectangle R' which is close approximation to R ,while using minimum examples and as little computation as possible.

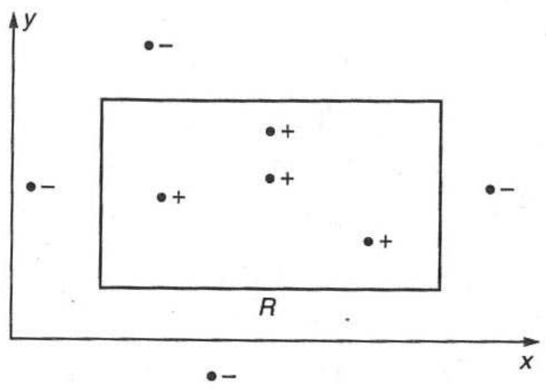

Figure 3: The target rectangle R in the plane along with a sample of positive and negative samples.

- We measure the player's hypothesis by choosing a random point from the same distribution D and check whether the player can correctly decide whether the point is inside or outside R. As a meter of fact we will check the error of R' by computing the probability that a random point from D will fall inside the area of $R \triangle R'$ where:

$$R \triangle R' = (R - R') \bigcup (R' - R)$$

2.2 Hypothesis Error Evaluation

Now we would like to show that for any target rectangle R and any distribution D, and for any small values ϵ and δ ($0 < \epsilon$, $\delta \leq 1/2$), and for suitably chosen value of sample (button press) size m we can make assert that with probability at least $1 - \delta$, the tightest-fit rectangle has error at most ϵ with respect to R and D. See figure 4.

We can express the difference R-R' as a union of four rectangular strips. if we can guarantee that the weight under D of each strip (that is, the probability with respect to D of falling in the strip) is at most $\epsilon/4$ then we can conclude that the error of R' is at most $4(\epsilon/4) = \epsilon$.

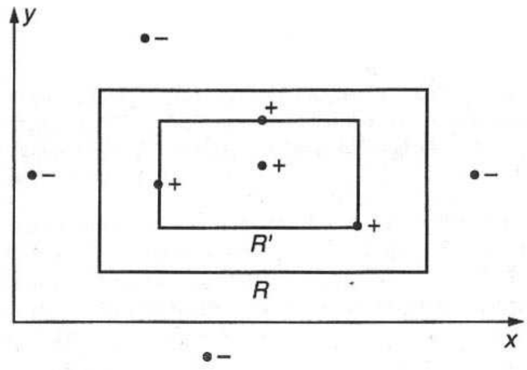

Figure 4: The tightest-fit rectangle R' defined by the samples.

lets check the weight of strip T' (see figure 5):
define T to be the rectangular strip along the inside top of R which encloses exactly weight $\epsilon/4$ under D. Because of the way that we define T, T' will weight $\epsilon/4$ under D if and only if T' includes T. This will happened if and only if no point in T is a + point (or else R' must extend upwards into T).
We can also say that because of the definition of T, the probability that a single draw from the distribution D misses the region T is exactly $1 - \epsilon/4$.
Therefore the probability that m independent draw will all miss the region T is exactly $(1 - \epsilon/4)^m$.
The same goes for all three other rectangular region R-R'. That's why the probability that one of the four regions will weight greater than $\epsilon/4$ is at most $4(1 - \epsilon/4)^m$ (according to the union bound).
Provided that we choose m to satisfy $4(1-\epsilon/4)^m \leq \delta$, then with probability $1-\delta$ over the m samples, the weight of the error region R -R' will be bounded by ϵ.
If we use the inequality $(1 - x) \leq e^{-x}$ we see that any value of m satisfying $4e^{-\epsilon m/4} \leq \delta$ also satisfies the previous condition. Dividing by 4 and taking natural logarithms of both sides gives $-\epsilon m/4 \leq \ln(\delta/4)$, or equivalently $m \geq (4/\epsilon)\ln(4/\delta)$.

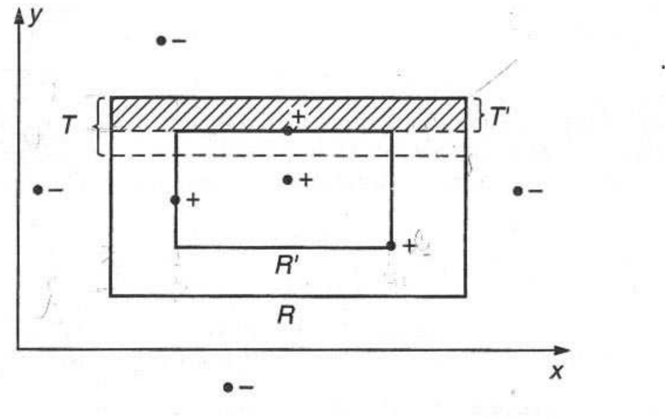

Figure 5: Analysis of the error contributed by the top shaded strip T'.

2.3 Summarize

If we use $(4/\epsilon)\ln(4/\delta)$ samples to form a hypothesis rectangle R', we can assert that with probability at least $1-\delta$, R' will misclassify a new point with probability at most ϵ.
Notice that the new point that we are trying to classify should be drawn according to the same distribution from which the samples were chosen.

References

[1] Y.Azar, A.Fiat, A.Karlin, F.McSherry, J.Saia, ”Spectral Analysis of Data”, Proceedings of STOC, 2001

[2] M.Kearns, V.Vazirani, ”An Introduction to Computational Learning Theory”, 1994.