Matrix-Theoretic Corpus Characterizations

Scribed by Gilly Leshed, Blazej Kot

This lecture is the start of a round-about introduction to Latent Semantic Indexing (LSI), via first explaining the ideas and the apparent magic underlying the Singular Value Decomposition (SVD).

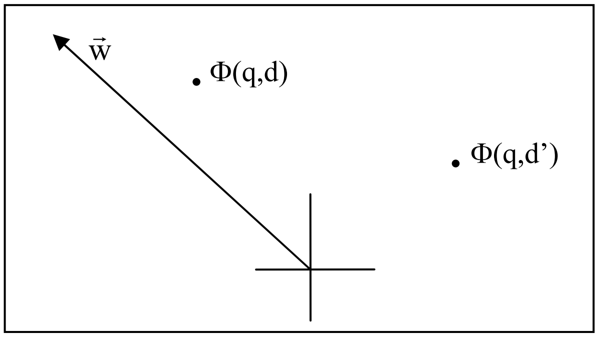
1 Implicit Feedback and Support Vector Machines Recap

We addressed using implicit feedback in a framework in which the available information is represented as points in a high-dimensional space. The dimensions of the space were the attributes of the vectors, where each vector represents a document-query dyad:

- Axis in the space ≈ feature / variable
- Vector component ≈ value of the variable for the item under representation

Implicit feedback created specific constraints, for example, that document d is *more relevant* than document d' for a given query q.

The "magic" weight vector $\vec{w}$ gives the preferred "direction" with respect to relevance to queries. $\vec{w}$ represents the fact that vectors farther along its direction are more preferred.



In this example, the horizontal axis could be, for example, (# Finnish terms - # English terms), and therefore points could obtain negative values as well. The vertical axis could be, for example, the value of $\cos(\vec{q},\vec{d})$.

The points in the space do not depend on the feedback information implicitly extracted from the user, but, they are query-dependent. Thus, classification is based on knowing the user's query. However, finding the weight vector $\vec{w}$ does depend on the feedback information, as discussed in the previous lecture.

The question that arises is, therefore, perhaps for a general purpose it might be useful to find a general characterization of the corpus, even without knowing the query. This could give us some indication of the structure of the corpus, or some inherent properties of the corpus. The idea is that we could wisely select the dimensions of the vector space, or the features along which we characterize the corpus, even without knowing the queries. We begin with representing the corpus as a *feature-document matrix*, which is presented as a *term-document matrix* in the next section.

2 Term-Document Matrix

Assume the corpus consists of *n* documents, each of which can be represented by an *m*-element feature vector, $\vec{d}^{(1)},...,\vec{d}^{(n)}$. We can combine these by writing them as columns of a matrix:

$$D = \begin{bmatrix} | & & | \\ \vec{d}^{(1)} & ... & \vec{d}^{(n)} \\ | & & | \end{bmatrix}$$

We will refer to elements of this matrix as $D_j^{(i)}$ where the superscript *i* refers to the matrix column, i.e. the document index, and the subscript *j* refers to the matrix row, i.e. the feature index.

Each element of a document vector corresponds to the value of one document feature. These features can be as simple as the number of times a vocabulary term occurs in the document; hence the label *Term-Document Matrix*.

3 Measuring Corpus Spread

Given this representation, we can now ask what the corpus looks like. In other words, how are the documents "spread" around the space? Are they evenly spread around, indicating that the corpus is less structured, or are they laid out in small clusters of related documents? This spread indicates the degree to which the corpus is varied in the vector space. Note that the spread depends greatly on the selected features, which could be other than term occurrences in documents. Selecting certain features could increase or decrease the spread of the corpus.

One approach to this question could be by looking at the span of the document vectors:

$$s = \text{span}(\{\vec{d}^{(1)},...,\vec{d}^{(n)}\})$$

We can then consider the dimensionality of the span $\equiv \dim(\text{span}(\{\vec{d}^{(1)},...,\vec{d}^{(n)}\}))$ as a measure of the "complexity" of the corpus.

Note that dim(s) is the rank of the term-document matrix, and recall from linear algebra:

$$r = \dim(s) = \text{rank}(D) = \text{rank}(D^T)$$

It may be hard to perceive how the rank of D is equal to the rank of D^T, especially because each one of them represents different things:

- D has *n* columns of documents, and shows how the values are distributed in each document between the features
- D^T has *m* columns of features, and shows how the values are distributed in each feature between the documents

$\underline{r = 1}$

To gain an understanding of what $r = \text{rank}(D)$ measures, let's first consider the case $r = 1$. In this case there exists a *basis vector* $\vec{b} \in \Re^m$ such that each document $\vec{d}^{(i)}$ in the corpus can be represented as a scalar multiple of this basis vector:

$$\vec{d}^{(i)} = \alpha_i \vec{b} \text{ for some } \alpha_i \in \Re$$

This means that in the corpus there is a single direction along which all document vectors lie. Furthermore, if we require that $\|\vec{b}\|_2 = 1$, then the basis vector is unique, up to sign.[1]

$1 \le r \le \min(m,n)$

In the general case, $1 \le r \le \min(m,n)$. The upper limit is there since the rank of a matrix cannot be larger than the smallest between the number of columns and rows of the matrix. In this case there exist r basis vectors $\vec{b}^{(1)},\ldots,\vec{b}^{(r)} \in \Re^m$ such that each document in the corpus can be represented as a linear combination of them:

$$\vec{d}^{(i)} = \sum_{l=1}^{r} \alpha_l^{(i)} \vec{b}^{(l)} \text{ for some } \alpha_l^{(i)} \in \Re$$

The basis vectors $\vec{b}^{(1)},\ldots,\vec{b}^{(r)} \in \Re^m$ are orthonormal, i.e. each is unit-length in 2-norm and they are mutually orthogonal.

Notice that given $\vec{b}^{(1)},\ldots,\vec{b}^{(r)}$ each document can now be represented as r scalar values (the $\alpha_l^{(i)}$'s), linearly combining the basis vectors to give rise to the document vector. So to represent the document, instead of a vector with m scalar elements, we could in principle use vectors with r scalar elements instead:

$$\vec{d}^{(i)} = \sum_{l=1}^{m} \vec{d}^{(i)} e^{(l)} = \sum_{l=1}^{r} \alpha_l^{(i)} \vec{b}^{(l)}$$

Where $e^{(l)}$ is the l^{th} elementary basis vector:

$$e^{(l)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \longleftarrow \\ \\ \\ \\ \end{matrix}$$

m elements,
l^{th} element is 1, all
others are 0

Thereby, we see that we can reduce the dimensionality of the vector space chosen to represent the documents from m to r without loss of information. This is known as *dimension reduction.* Note that each column in the matrix $[\vec{b}^{(1)},\ldots,\vec{b}^{(r)}]$ is still m elements long, however, as indicated by the equality above, the number of vectors required to represent each vector $\vec{d}^{(i)}$ was reduced from m to r.

[1] We saw earlier how features can have negative values, if we assume that they are not necessarily term counts.

4 Restricting the Linear Combinations

Generally speaking, there could be an infinite number of basis vector sets. However, we would like to get a better sense of what a “good” basis set would be. We have been looking at the dimensionality of the span of the document vectors as a measure of complexity. The span is defined as the space of all linear combinations of the document vectors. By restricting the allowed linear combinations, so that instead of all possible ones we only allow those which satisfy a certain property, we could perhaps obtain more useful information from the basis set.

A brief note about notation: we can represent a linear combination of the documents in the corpus as a sum which is equivalent to a matrix multiplication:

$$\sum_{i=1}^{n} \alpha_i \vec{d}^{(i)} = D\vec{\alpha} = \begin{bmatrix} | & & | \\ d^{(1)} & \cdots & d^{(n)} \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \alpha_1 d_1^{(1)} + \alpha_2 d_1^{(2)} + \cdots + \alpha_n d_1^{(n)} \\ \vdots \\ \alpha_1 d_m^{(1)} + \alpha_2 d_m^{(2)} + \cdots + \alpha_n d_m^{(n)} \end{bmatrix}$$

where $\vec{\alpha}$ is an *n*-element column vector consisting of the coefficients for each document vector. One way to think about this is to think of D *as an operator* that is applied to $\vec{\alpha}$, to create the linear combination of document vectors according to the coefficients $\vec{\alpha}$. We can restrict the kind of linear combinations we allow by restricting the values $\vec{\alpha}$ can take.

4.1 Fractional Assignment

Fractional assignment of the $\vec{d}^{(i)}$'s requires that the sum of all elements of $\vec{\alpha}$ is 1 and that each element of $\vec{\alpha}$ is nonnegative:

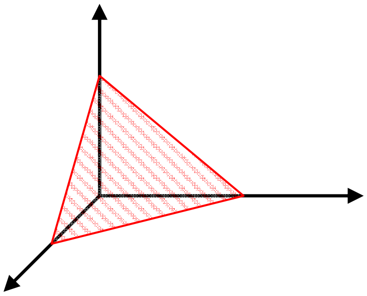
$$\sum_{i} \alpha_i = 1 \text{ and } \alpha_i \geq 0$$

To get a sense of the effect of using fractional assignment, consider a corpus of 3 documents, each represented by a two-element feature vector:

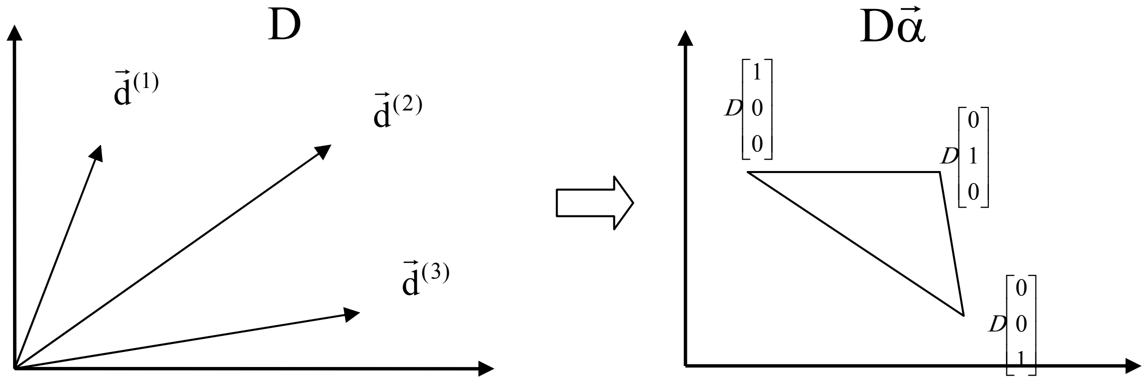
This means the corpus matrix D is a $m = 2$ x $n = 3$ matrix: $D = \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}$

$\vec{\alpha}$ will therefore be a 3-element vector.

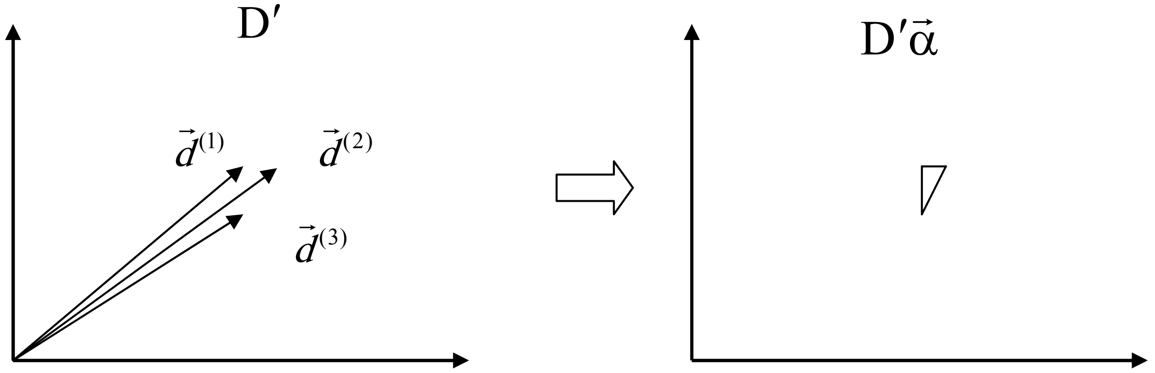
Because we are using fractional assignment, all $\vec{\alpha}$'s must end at a point in the positive octant of the plane passing through the points (1,0,0), (0,1,0), (0,0,1), as shown below:



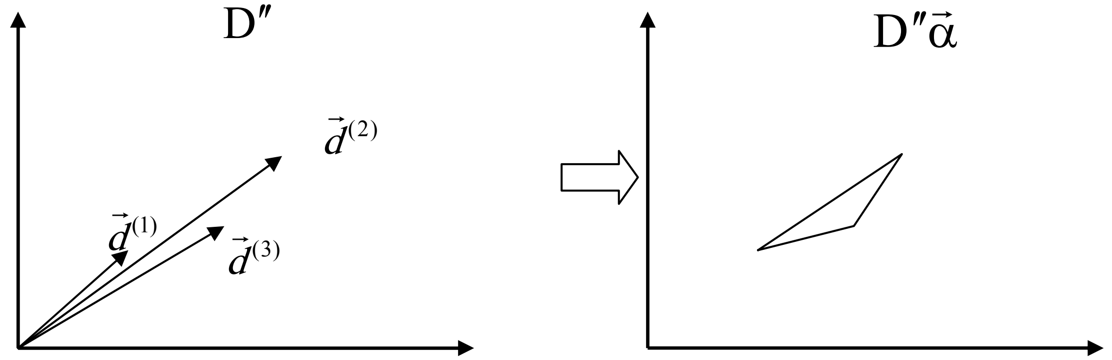
If the corpus D has a big spread, we can ask what all the possible values that $D\vec{\alpha}$ can take are, remembering that $\vec{\alpha}$ is restricted to fractional assignment. One way to look at this is to plot the values for the three extreme options of $\vec{\alpha}$: (1,0,0), (0,1,0), (0,0,1). Obviously, these will correspond exactly to the three documents in the corpus, since each value of $\vec{\alpha}$ will select one document from D with weight 1, and the others with weight 0. Since these are the extreme values of $\vec{\alpha}$, and matrix multiplication is linear, we know what we have plotted are the extreme values of $D\vec{\alpha}$. This means all possible values of $D\vec{\alpha}$ lie within the convex hull shown below:


D
$\vec{d}^{(1)}$
$\vec{d}^{(2)}$
$\vec{d}^{(3)}$
$D\vec{\alpha}$
$D\begin{bmatrix}1\\0\\0\end{bmatrix}$
$D\begin{bmatrix}0\\1\\0\end{bmatrix}$
$D\begin{bmatrix}0\\0\\1\end{bmatrix}$

The question that arises is what this convex hull would look like for different corpora. If the document vectors all point at the same direction:



As can be seen, the *area* of the convex hull is much smaller in the case of document vectors all pointing toward the same direction than the case in which directions are spread out. This suggests that perhaps we can use the area of the hull as a measure of the complexity of the corpus. However, before setting fractional assignment as the "ultimate" approach, consider another corpus, which has little variation in the direction of the document feature vectors, but large variation in the document feature vector lengths:



The convex hull in this case has a larger area, although the corpus is not obviously more varied in terms of directionality of the vectors. This suggests that a more subtle measure is required, perhaps by considering the direction in which the triangle points. This will be the topic of the next lecture.

Questions

We saw in class that the term-document matrix D can be represented using a set of $r = \text{Rank}(D)$ basis vectors $\vec{b}^{(1)},\ldots,\vec{b}^{(r)} \in \Re^m$. Each vector $\vec{d}^{(i)} \in D$ can be obtained using the basis vectors and *r* coefficients $\alpha_1^{(i)},\ldots,\alpha_r^{(i)}$ in the following way: $\vec{d}^{(i)} = \sum_{l=1}^{r} \alpha_l^{(i)} \vec{b}^{(l)}$

1) What do the basis vectors represent in this idea?

<u>Answer:</u> The number of basis vectors is the lowest dimension into which the corpus can be reduced without losing information (given that we know some information is already lost in the VSM bag-of-words approach). They can be viewed as *r* 'topics', 'meanings', or 'domains' covered by the corpus. Note that these do not necessarily correspond to 'topics', since, for example, there is no guarantee that 'topics' corresponding to different basis vectors are really separate. For example, consider a corpus where the only two terms are “*cat*” and “*feline*”: There will be two orthogonal basis vectors, but really only one topic. Each of the original documents of the corpus can be reconstructed by linearly combining together the basis vectors. Each element $b_j^{(l)}$ in the basis vector $\vec{b}^{(l)}$ expresses the value of term j for the 'pattern' represented by the *lth* basis vector. Thus, while the row $d_j^{(1)},\ldots,d_j^{(n)}$ of the document matrix represents the distribution of term j across all documents, $b_j^{(1)},\ldots,b_j^{(r)}$ represent the distribution of term j across all 'basic patterns'.

2) What do the $\vec{\alpha}$ coefficient vectors represent?

Answer: Each $\alpha_l^{(i)}$ captures the weight of the basis vector $\vec{b}^{(l)}$ for the reconstruction of vector $\vec{d}^{(i)}$. In other words, it can be viewed as the degree to which the document's meaning is connected to the meaning represented by the l^{th} basis vector. We can therefore conceive vector $\vec{d}^{(i)}$ as a weighted average of the patterns covered by the corpus.

3) As we discussed in class, the purpose of representing n term-document vectors using r basis vectors ($r \leq n$) is to find a new representation of the corpus with a lower dimension. Under what circumstances would the vector space dimensionality reduction be beneficial? What would be the advantages and disadvantages of this idea?

Answer: Dimensionality reduction of the term-document matrix is reasonable when we know that the original data was generated from something that is in a lower dimension. It then makes sense to attempt to find the representation that corresponds to that reduced dimensionality.

One clear benefit of the term-document matrix dimensionality reduction is that once created, the new, smaller representation presumably requires less computing power. Given a query, we do not need to examine all the documents, but all the basis vectors to find which basis vectors match the query vector. This can be done by finding the projection of the query vector onto the new space created by the basis vectors, i.e. finding the $\vec{\alpha}^q$ coefficient vector such that $\vec{q} = \sum_{l=1}^{r} \alpha_l^q \vec{b}^{(l)}$. Those basis vectors $b^{(l)}$ of which the corresponding α_l^q's are large, are referred to as 'matching' the query. We can then reconstruct the original documents from the basis vectors that comply with the query.

Another advantage of this approach is that adding new documents to the corpus does not require reconstructing the set of basis vectors from the entire corpus. If adding a document does not increase the rank of the corpus, then the corresponding document vector can be represented using the already existing basis vectors. On the other hand, if we know that the addition of a given document increases the rank, then that document is orthogonal to the current basis vectors and so could be length normalized to become the next basis vector. Since all the previous document vectors were orthogonal to this new one, we know that their projection onto the new basis vector is zero.

A disadvantage of dimensionality reduction refers to its interpretation: it lends itself for the belief that the set of basis vectors correspond to the set of meanings encompassed by the corpus. However, the basis vectors could also correspond to the set of term patterns in the corpus that do not necessarily follow any natural understanding of the meanings concealed within the corpus.

4) The following are 9 titles of books found in Amazon.com:

1. When Elephants Weep: The Emotional Lives of Animals
2. The Social Lives of Dogs
3. Hen's Teeth and Horse's Toes
4. The Panda’s Thumb
5. The Cat in the Hat
6. Brown Bear, Brown Bear, What Do You See?
7. The Very Hungry Caterpillar
8. Never Ride Your Elephant to School
9. Don't Let the Pigeon Stay Up Late!

How would you represent these books given a dimension lower than 9?

Answer: Note that the first four books are popular scientific books about animals. The next five are all children’s books in which the main figures are animals. We can postulate that the underlying term-pattern of these books follow some conventions stemming from the topics of these books. All books contain names of animals. But the first four books are more likely to contain a greater vocabulary and more “scientific” terms, and the next five books consist of a smaller vocabulary and more “childish” terms that we expect to find in children’s books. Thus, these nine books could possibly be represented using three basis vectors, representing three domains: scientific concepts, animals, and children. Notice that the orthogonality constraint for the choice of these basis vectors requires that no one basis vector can be a linear combination of the other two. For example, the ‘animals’ vector and the ‘children’ vector cannot be combined to construct the ‘scientific concepts’ vector.