CS 630 Lecture 18: 04/06/2006

Lecturer: Lillian Lee Scribes: Asif-ul Haque, Benyah Shaparenko

---

1 Review of the SVD

For a document matrix $D \in \Re^{m \times n}$, the SVD of D (which is "more or less" unique depending on the singular values):

$$D = \underbrace{\begin{bmatrix} | & & | \\ \vec{u}^{(1)} & \cdots & \vec{u}^{(r)} \\ | & & | \end{bmatrix}}_{\text{left singular vectors}} \underbrace{\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}}_{\text{singular values} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0} \underbrace{\begin{bmatrix} - & \vec{v}^{(1)} & - \\ & \vdots & \\ - & \vec{v}^{(r)} & - \end{bmatrix}}_{\text{right singular vectors}}$$

In short, $D = U\Sigma V^T$ where U is orthonormal and has dimension $m \times r$, Σ has dimension $r \times r$ and V is orthonormal and has dimension $n \times r$ where m is the total number of terms in the vocabulary, n is the number of documents in the matrix and r is the rank of the matrix D.

Eigenvectors

Often people have this misconception that the singular vectors are eigenvectors of D. It is not too hard to see why this is wrong. We can do SVD for any matrix – not just for square matrices. But eigenvectors exist only for square matrices. For a $m \times n$ matrix D and a $n \times 1$ matrix (vector) u, Du has dimension $m \times 1$ – it can never be any multiple of u because of dimension mismatch unless $m = n$. However, we can consider the matrix D^TD, which is square and symmetric, implying that the full range of eigenvalues exist (some may have multiplicity greater than one). Σ is symmetric i.e. $\Sigma^T = \Sigma$. U and V are orthonormal i.e. $U^TU = I$ and $V^TV = I$.

$$\begin{aligned} D^TD &= (U\Sigma V^T)^T(U\Sigma V^T) \\ &= (V\Sigma^TU^T)(U\Sigma V^T) \\ &= V\Sigma U^TU\Sigma V^T \\ &= V\Sigma^2V^T \end{aligned}$$

$$(D^TD)\vec{v}^{(l)} = (V\Sigma^2V^T)\vec{v}^{(l)} = \underbrace{\sigma_l^2}_{\text{eigenvalue}} \vec{v}^{(l)}$$

2 LSI

Although the idea of projecting high-dimensional data onto lower dimensional space was known for years, LSI (re)emerged by the paper by Deerwester, Dumais, Furnas, Landauer, Harshman '90. In LSI, we approximate Σ with $\hat{\Sigma}$ where $\hat{\sigma}_i = \sigma_i$ for $1 \leq i \leq k$ and $\hat{\sigma}_i = 0$ for $k < i \leq r$ where k is a parameter that can be changed. So $\hat{D} = U\hat{\Sigma}V^T$ is an approximation of D. Eckart and Young proved in 1939 that this is an optimal (in terms of matrix norm of the difference matrix, specifically, in both $||D - \hat{D}||_2$ and $||D - \hat{D}||_F$, but don't worry about this) rank-k approximation of D. The columns of $\hat{D}$ are $\hat{\vec{d}_i}$s – the new representation of the documents. They can used for whatever purpose the original document vectors were used for. Often U, $\hat{\Sigma}$ and V are convenient for computation, and the full matrix $\hat{D}$ need not be actually formed.

2.1 LSI: Some success stories

- Improved ad-hoc retrieval (sometimes). In other words, cosines in the new subspace can be more representative of query/doc similarity.
- Grading essays. It has been observed that human grading and using LSI to grade essays (considering the essays as a "bag of words") are remarkably close. For the "Intelligent Essay Assessor" (Foltz, Laham, Landauer '99), "the IEA scores agreed with human experts as accurately as expert scores agreed with each other." Of course this casts a shadow of doubt on the grading scheme used by the humans (or on the essays that are being graded).
- Matches human TOEFL test-taker scores. Landauer, Dumais '97 showed that the problems on selecting the appropriate synonym in TOEFL can be solved just as well as humans. For this, they used the Grolier's encyclopedia ($\sim$ 30k documents) as the corpus and computed the term (row) similarity in the subspace. They found $k = 300$ gives the optimal performance and hence made the contentious remark that human knowledge has only 300 dimensions.
- May capture important co-occurrence patterns. It could be conceptual, e.g., "probabilistic" and "stochastic" are used as synonyms. Or, it could be collocational, e.g., "humpty" and "dumpty" appearing together frequently.
- Varying dimensionality is like searching over similarity functions (trying different linear combinations of features).
- The Eckart-Young theorem is a "positive" result showing something is optimal rather than many other theorems that provide negative results like "finding the optimal is NP-complete" which is not useful in any constructive sense.

2.2 LSI: Disadvantages

The following disadvantages are not necessarily just for LSI, but could also be for any similar methods.

- There are no guarantees for IR performance. However, we kind of expected this to be the case.
- The $\vec{u}^{(l)}$'s are not interpretable. This may or may not matter depending on what the purpose for finding the $\vec{u}^{(l)}$ vectors is. In the case of IR, where we just use LSI and these vectors to improve performance without regard for interpretation, we probably don't care about what $\vec{u}^{(l)}$ means.
- The representation is not sparse. The new document vectors have many non-zero entries. (cf. Non-negative matrix factorization (Seung and Lee) which attempts to address this problem.)
- SVD is expensive. Calculating the full SVD of the entire matrix is $O(m^2n)$ or $O(n^2m)$. (Note: the subspaces implicitly represented by the SVD are usually fairly stable for small perturbations in the data. On the other hand, the singular vectors aren't necessarily stable: one outlier can affect the ellipsoid shape a lot.)

- What about dimensionality selection? There is an inherent tradeoff between generalization and faithfulness to the original data.

2.3 Explanatory Features

One of the issues with LSI was that the $\vec{u}^{(l)}$'s are not interpretable. To solve this problem, can we get k more "explanatory" features, where "explanatory" functions in some way like a topic? We could think about using latent variable methods, e.g., PLSI (Hofmann '99).

PLSI: Probabilistic LSI

Suppose that we've been given a set of topic (language) models $t_1 \cdots t_k$. The idea is to model a document as some distribution over these topic models t_i. For example, some document might be 50% about cats and 50% about the 300 dimensions of human knowledge. (Cf. our previous LM scenarios, where each document's terms were only generated by one topic model). The document's contents are $\hat{P}(v|d) = \sum_t P(t|d)P(v|t)$. (Note: there "should" be a dependence on d in the last quantity, but because we make a conditional independence assumption, it goes away.) So, a document's contents are produced by first generating a topic, then picking a term "from" that topic, then picking another topic and choosing a term according to that topic model, and so on.

Why is this called PLSI? First, it is because we have (probabilistic) arguably semantic factors derived as "explanations" for the data. Second, it is possible to make the probabilities look something like LSI.

$$\begin{aligned}
\hat{P}(v,d) &= P(d)\hat{P}(v|d) \\
&= P(d)\sum_t P(t|d)P(v|t) \\
&= P(d)\sum_t \frac{P(d|t)P(t)}{P(d)}P(v|t) \\
&= \sum_t P(d|t)P(t)P(v|t)
\end{aligned}$$

It is possible to view $P(t)$ as analogous to a singular value, and $P(d|t)$, $P(v|t)$ as analogous to components of singular vectors. Furthermore, the last summation actually looks like a three-matrix product expanded out. However, PLSI itself has nothing to do with the SVD. (For instance, no orthogonality constraints hold).

The learning problem to learn/infer $P(t)$, $P(t|d)$, and $P(v|t)$ is hard. Hofmann proposes an algorithm and finds that PLSI performance in ad-hoc retrieval is good, outperforming LSI (when interpolation with the original cosine values is performed). Some of the results for PLSI are reproduced in the lecture handout, section I.

2.4 Dimension Selection

Next, we will look at another one of the issues, namely, dimensionality selection. Selecting the number of dimensions is hard, and the "optimal" value is surely application- and dataset-independent. One option is to have a penalty for more factors, which can be encoded directly into the optimization function. Here is a method which encodes this penalty implicitly.

Chinese Restaurant Process: (Blei, Griffiths, Jordan, Tenenbaum '03)

There are an infinite set of factors, which correspond to the infinite number of tables in a Chinese restaurant. (The tables are round, and so can be considered to have unbounded capacity.) Customers represent data points, which arrive from time to time. Upon arriving, the customer selects a table. The customer picks the table l with a probability proportional to the number of items at table l, or else picks an empty table (with some low probability). This process exhibits "rich-get-richer" clustering. The intuition for the purposes of inference is that we prefer a small number of factors, but if a data point comes along that really isn't described well by any of the previous factors, then it can start a new factor if necessary.

3 Problems

1. If we assume that LSI works the way many people think it does, why might we want to ignore the first singular value/vector (or maybe even the first few singular values/vectors)? LSI already throws away the singular values $\sigma_{k+1} \cdots \sigma_r$ and their corresponding vectors. The suggestion here is to likewise throw away a singular value/vector on the opposite end as well.

2. Consider the matrix $B = D^T D$, and let $\vec{w}$ be a normalized vector in $\Re^n$, and let $\vec{z}$ be the normalized version of $B\vec{w}$.
   (a) Can we think of a meaning for the entry B_{ij}?
   (b) Using this interpretation, does the entry z_i make any sense?
   (c) If $\vec{w}$ is an eigenvector corresponding to the eigenvalue $\lambda = \sigma^2$ of B, what is the significance of $\vec{w}$?

3. (open-ended) In the Chinese restaurant process, the customers (data points) walked into the restaurant and selected a table (factor) to sit down at. Given n customers who are described by m dimensions, with probabilities p_i of the customer selecting a new table (except for the very first customer), how many tables are likely to be chosen after all n customers are seated? Does it make sense to ask what the "expected value" of the number of tables is? How might one go about calculating this quantity in the Chinese restaurant case? What implications does this calculation have on the reducibility of the data? Given this calculation for the number of tables to be used, what implications does this have for clustering?

Answers

1. The largest singular values correspond to groups of terms that have high weights. After all, the singular values represent the lengths along the axes of the hyper-ellipsoids. Depending on the weighting scheme used for the term weights, the first singular value/vector may correspond to the very common words that appear often across the corpus (this actually happens in practice). For example, if every document mentions the terms "humpty" and "dumpty," then they are very strongly collocationally correlated, even more so than many other term combinations. Consequentially, even though the terms are very common and thus useless for IR, they may still be included in the first singular value/vector. Alternatively, performing stopword removal and using an IDF weighting can ameliorate this situation since they address the same issue.

2. (a) B_{ij} is the dot product of the document vectors $\vec{d_i}$ and $\vec{d_j}$ i.e. $B_{ij} = \vec{d_i} \cdot \vec{d_j}$. If the document vectors are normalized, B_{ij} means the cosine of the angle between the two vectors.

   (b) $z_i = \alpha \sum_j w_j(\vec{d_i} \cdot \vec{d_j}) = \alpha\vec{d_i} \cdot \sum_j w_j\vec{d_j}$ where α is a normalizing constant. So z_i is the dot product between $\vec{d_i}$ and the weighted sum of all the document vectors where the weights are the values of $\vec{w}$.

   (c) Since $\vec{w}$ is an eigenvector and $\vec{z}$ is normalized, $\vec{z} = \vec{w}$. One meaning could be that the weight w_i is the dot product of the document vector $\vec{d_i}$ and the weighted sum of all the document vectors. In other words these weights reflect some sort of directionality between a document vector and the weighted sum of all vectors. But is this meaningful in any sense?

   The eigenvectors of B are the right singular vectors of D:

$$B = D^T D = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma^2 V$$

   These vectors are already normalized. The right singular vectors have as much of an interpretation as the left singular vectors do, except for the term space – so one potential interpretation is as a "dominant" document-occurrence pattern (in the sense of having a large associated singular value, not necessarily in the case of being representative of the ("term") corpus).

3. One would expect, even for a given number of customers and dimensions, the probability of selecting a new table and the number of tables varies widely depending on the data. If the data are very reducible, e.g. in the degenerate case that all the document vectors were linear combinations of each other, then everyone should sit at the same table. However, if the term document matrix D happened to be the identity matrix, then we would probably hope that everyone sits by himself or herself. The expected value of the number of tables depends on the probabilities p_i that are chosen. In the Blei et al. paper, the first customer sits at a table. Other customers m sit at a new table with probability $\frac{\gamma}{\gamma+m+1}$, where γ is a parameter. Steele and Aldous conclude that the number of tables is $O(log\ n)$ by doing something similar in Section 1.1.5 of "Introduction [to the interface of probability and algorithms]" which can be found at http://www-stat.wharton.upenn.edu/∼steele/Publications/PDF/ItPAat.pdf.

   Since the Chinese restaurant process can be thought of as a clustering algorithm, the number of tables could correspond to the number of clusters. It is known that, in clustering, asking what the "correct" number of clusters is, even given a particular data set, is often very difficult. Answering the same question in general is even more difficult, maybe even impossible. Although the Chinese restaurant model may provide the math background to actually calculate the expected value of the number of tables, are we willing to assign to this value the interpretation that this is the number of clusters?