Latent Semantic Indexing and Nonnegative Matrix Factorization – A Survey

James Musk
jamesmusk@berkeley.edu
23851414

Claire Asselstine
claire.asselstine@berkeley.edu
23628013

1. INTRODUCTION

In this paper we inspect one of the first works to analyze Latent Semantic Indexing as an information technique. We then move too identify potential problems which have been pointed out with this model in terms of its ability to determine latent factors, and then review proposed solutions to this problem. Second, we move to nonnegative matrix factorization, and analyze its runtime as performed by Arora, Ge, Moitra and Kannan in their 2011 paper.

1.1 A Motivation For LSI

Latent Semantic Indexing is an information retrieval method which uses Singular Value Decomposition to identify connections between terms and concepts in sets of documents given as an unstructured list of words. This has an application to problems including that of search engines, discovering ranking documents by their ”relevance” to another list of words. It is also used to do document categorization and clustering.

The idea originates in the mid 1960s when Borko and Bernik first proposed Factor analysis. People have now found that it can help solve problems in many many fields including Social Networking, Software Engineering, System Administration, and Education. Four professors from the University of Joensuu in Finland demonstrated the use of Probabilistic Latent Semantic Analysis for automatically grading essays. Yes, there have been many other arguably more complex and meaningful computational feats, but we believe that their paper stands as a beacon of success in showing that the Computer Science industry can do great things for society as a whole.

1.2 Obstacles

There are many obstacles in creating a model that will represent and solve specific events in natural language processing. As Hofmann states in his novel paper on Probabilistic Latent Semantic Indexing, ”one of the fundamental problems is to learn the meaning and usage of words in a data-driven fashion, i.e., from some given text corpus, possibly without further linguistic prior knowledge.” Meaning and usage are both ideas which are hard to grasp or calculate. A model should be expected to be able to take what was actually written and infer what was meant to be written, or what was referred to. The two fundamental obstacles are synonymy and polysemy.

Polysems create a large problem because they could unnaturally highly correlate two distinct ideas. For example, the words ”drum” and ”beat” will occur documents about music, and the words ”compete” and ”beat” will occur in documents about sports. Since the word ”beat” has multiple meanings, we have the problem that unless a model is very smart, documents about music and sports will be represented as similar.

A second problem is synonyms. Consider the scenario that one document uses the words ”dance”, ”groove”, and ”boogie”, and another uses ”jump”, ”jive”, and ”jig.” If a model is not powerful, then these documents will be represented as dissimilar.

To summarize, polysemy and synonymy are the largest problems in LSI and can very likely lead to false positive similarity and false negative similarity. There are many more intricacies to language which come up, but from what we can tell these are the most important for models to be able to handle with some accuracy.

2. LATENT SEMANTIC INDEXING

2.1 Corpus Model

We represent a document as a ”bag of words” in which the order of the words do not matter. This is one of the first and most basic assumptions that LSI makes. Let there be n different valid words in our dictionary, every document d will be represented in $\mathbf{R}^n$ by some n dimensional vector v_d. The i^{th} element of this vector v_d will be the number of occurrences of word i in document d. A corpus is any set of such documents. It is easy to see that stacking all document vectors horizontally will create one large term-document matrix. This is the end goal for a probabilistic model to be able to produce.

The corpus model that we create will be defined by four things: U, T, S, and D. We define U as a set of all n terms in the universe. We define a topic as a distribution over U, and T as a set of all topics. The idea is that each topic will be very different from the uniform distribution. This is intuitively because documents about a certain topic will contain some words very frequently, and others much more rarely. For example, if our topic is ”techno” we expect the word ”beat” to have a higher probability of coming up than the word ”skiing”. The third element we will need to define a corpus model is style. This will be represented as a stochastic matrix S which is n by n which will map a document to another, changing the frequency of each word. Finally we define D as a probability distribution of convex combinations of topics, convex combinations of styles, and positive integers (these integers will be used as the length of each document).

It must be noted that under this construction, each document can have any convex combination of all topics, any convex combination of styles, and a length of any positive integer. It can also be observed that style adds an extra transformation on a distribution of words, thus, if styles are too extreme (very drastically change the frequencies of words) then this could make it much more difficult decipher between documents which are purely from distinct topics. Therefore we would have some expectation that styles will preserve the distance between vectors.

A single document is defined by a topic, t which is a convex combination of T, a style s which is a convex combination of S, and a length L. The words in the document are created by sampling L times from ts. An entire corpus of documents can be created by sampling from D to obtain combinations of t, s, and L.

This is the full construction of a corpus, which is laid by Papadimitriou et all in their 1998 paper, however, LSI had been used for decades previous to this.

2.2 LSI Algorithm

2.3 The simplest case

A first analysis of Latent Semantic Indexing began by peering into the simplest case, and mathematically proving that if you can make very strict assumptions about your documents, then LSI will indeed discover the original corpus.

Two important terms to define are "pure" and "$\epsilon-$separable. A document is pure if it is only associated with a single topic. A corpus is $\epsilon-$separable where $0 \leq \epsilon < 1$ if the set of terms U_T associated with each topic are mutually disjoint, and each probability that T assigns to all terms in U_T is at least $1 - \epsilon$.

A final term to define is the skew of the documents. The skew is a measure of how parallel similar documents are, and how orthogonal different topics are. In mathematical terms, this is to say that a document is δ-skewed if for all d_1 and d_2 documents from the same topics, $v_{d_1} v_{d_2} \geq 1 - \delta ||v_{d_1}||||v_{d_2}||$. For all d_1 and d_2 documents from different topics, $v_{d_1} v_{d_2} \leq \delta ||v_{d_1}||||v_{d_2}||$.

It is clear that we must be able to compute some bounds on the how large the skew will be. Let us take the strict assumption that our documents are pure, ϵ-separable, and the probability that each topic assigns to each term is sufficiently small. Papadimitriou, Raghavan, Tamaki, and Vempala found that in this case, the rank-k LSI is $O(\epsilon) - skewed$ on C with probability 1 - $O(m^{-1})$.

We begin with A, a term-document matrix representing a corpus, and define C_i as the documents in our corpus belonging to topic i. Let us first start with the case when $\epsilon = 0$, it will be easy to show that it is also true for small ϵ once this is done. C_i will only contain terms from set U_i, the set of terms most associated with topic i. The matrix A will have a set of blocks B_i, the rows corresponding to the terms in U_i and the columns corresponding to the documents in C_i. Because of this property, $A^T A$ is block-diagonal, with blocks $B_i^T B_i$. As in the Laplacian Matrix we learned in lecture, $B_i^T B_i$ is an adjacency matrix of a random bipartite multigraph because the terms are sampled randomly. Using the spectral theory of graphs, we have that $\frac{\lambda'_i}{\lambda_i} \rightarrow 0$ with probability 1 as $\tau \rightarrow 0$ and $|C_i| \rightarrow \inf$. To show this we prove that the conductance of $B_i^T B_i$ is high. Once we have shown this we have shown that the eigenvalues have these properties. We can now say that if the sample size m = $|C|$ is sufficiently large, and the random term probability τ is sufficiently small, which also means that the size of the primary set of terms for each topic is sufficiently large, the k largest eigenvalues of $A^T A$ are the λ_i, $1 \leq i \leq k$, with high probability. The probability bound comes from bounding the high probabilty of $B_i^T B_i$ having high conductance. The notable property is that each λ_i comes from a distinct $B_i^T B_i$ block. Suppose now that our sample C indeed enjoys this property. Let u_i denote the eigenvector of $B_i^T B_i$ corresponding to eigenvalue λ_i and let u_i be its extention to the full term space, obtained by padding zero entries for terms not in T_i. Then, the k-dimensional LSI-space for corpus C is spanned by the mutually orthogonal vectors u_i, $1 \leq i \leq k$. When a vector v_d representing a document d $\in C_i$ is projected onto this space, the projection is a scalar multiple of u_i, because v_d is orthogonal to u_j for every j $\neq$ i.

Now let's look at the case where $\epsilon > 0$, the term document matrix A can be written as $A = B + F$, where B consists of blocks B_i as above and F is a matrix with small $||L||_2$-norm (not exceeding ϵ by much, with high probability). AS observed in the above analysis for the case $\epsilon = 0$, the invaraint subspace of $B_i^T B_i$ corresponding to its largest k eigenvalues is an ideal representation space for representing documents according to their topics. Our hope is that the small perturbation F does not prevent LSI from indentifying W_k with small errors.

Let W_k' denote the k dimensional space the rank k LSI identifies. The ϵ separability of the corpus model implies that the two norm of the document-term matrix is $\mathcal{O}(epsilon)$. We use this fact to say that the two-norm of the differnce between the matrix representations of W_k and W_k' is $\mathcal{O}(\epsilon)$. Since W_k' is a small perturbation of W_k, projecting a vector document onto W_k' yields a vector close, in its direction, to u_i - the dominating eigenvector of $B_i^T B_i$. Therefore, the LSI representations of two documents are almost in the same direction if they belong to the same topic and are nearly orthogonal if they belong to differnt topics. (Papadimitriou et. all, 1998)

2.4 Random Projection

Singular Value decomposition can be computationally very expensive in very high dimensions containing many documents. It can be shown that the Johnson Lindenstrass Lemma can be utilized to first apply a random projection in order to speed up the runtime of LSI. This initial random projection is not meant to bring related documents together, only to represent the data on a more reasonably sized space. Let this dimension be l where $l > k$ (k being final dimenstion after SVD, and l satisfies the JL lemma for a given ϵ, so $l = O(\frac{logn}{\epsilon^2})$. The paper specifically uses an l satisfying $24logn < l < \sqrt{n}$. A random projection onto l dimensions will preserve the distance between any two points be with an error of ϵ. So for any two documents d_1 and d_2 which are assigned the n dimensional vectors v_{d_1} and v_{d_2}, after undergoing a random projection onto an l dimensional space, the the JL lemma gives the following bounds:

$$(1-\epsilon)||v_{d_1}-v_{d_2}|| \leq ||f(v_{d_1})-f(v_{d_2})|| \leq (1+\epsilon)||v_{d_1}-v_{d_2}||$$

The inner products are also preserved, more precisely, if all vectors are of length no greater than 1, the inner product between any two document vectors will change by no more than 2ϵ.
When A is our total term document matrix, and R is a random column-orthonormal matrix mapping from dimension n to l (R has n rows and l columns), let B be defined in the following way:

$$B = \sqrt{\frac{n}{l}} R^T A$$

B is the way we can represent our data after being mapped onto the lower dimensional space, and scaled accordingly. Therefore the SVDs of A and B are:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$B = \sum_{i=1}^{t} \lambda_i u_i v_i^T$$

We aim to compare the accuracy between performing SVD on A, versus creating B and then finding the SVD of on a lower dimention. The following theorem does exactly that:

$$||A - B_{2k}||_F^2 \leq ||A - A_k||_F^2 + 2\epsilon||A||_F^2$$

To phrase this in english, $||A - A_k||_F^2$ is a measure of the information retrieved from direct LSI, and it can be found that "the matrix obtained by random pro jection followed by LSI expanded to twice the rank recovers almost as much as the matrix obtained by direct LSI." (11)

How much computational time was saved moving from direct LSI to this two step process? If we let c be the average number of terms in a document, then the time needed to compute a projection onto l projections is $O(mcl)$, and LSI afterwards is $O(ml^2)$. This makes a total of $O(ml(l+c))$, where $l = O(logn/\epsilon)$ Let us compare this to the original LSI, which takes $O(mnc)$. The two step process is aymptotically better!

2.5 Assumptions

Some assumptions about a data set can be reasonable, and others may be far too strict in most cases. For example, the authors of this paper defend the assumption of ϵ-separability on the basis that most documents are preprocessed to exclude stop words.(Papadimitriou et. all, 1998) However, other assumptions, such as purity or the absence of style, might be far too extreme. In these cases, these early theorems and bounds may simply defend the notion of LSI as a way to problem solve, rather than prove bounds on information retrieval accuracy.

3. FOLLOWING BREAKTHROUGHS

3.1 Probabilistic Latent Semantic Analysis

As stated, the previous paper was one of the first pieces of analysis as LSI. Upon more inspection, it was criticized for a few specific reasons, which lead to new ideas. One comes from Thomas Hofmann from University of California, Berkeley. In his paper "Probabilistic Latent Semantic Analysis" he proposes a similar model to Latent Semantic Analysis, however, he argues that "this approach is more principled than standard Latent Semantic Analysis, since it possesses a sound statistical foundation."

One problem some point out with Latent Semantic Analysis is that documents are not only highly correlated if they have many similar words, but also if they exclude similar words. For example, if one document has the topic "techno" and another topic has the topic "furniture" then they will both have a very low probability of containing the word "precipitation." Using regular LSI, just based on a negative correlation to the word "precipitation" two documents of very different topics could be represented as more similar.

Instead of using SVD to map data to a lower dimension, Hofmann uses a factor space to perform the same kind of dimensionality reduction purely based on words which documents have in common. This makes more intuitive sense as well as being more mathematically sound. Instead of minimizing the Frobenius norm, PLSA relies on the likelihood function of multinomial sampling and "aims at an explicit maximization of the predictive power of the model." (HOfmann, 291) According to Hofmann, this new approach "yields substantial and consistent improvements over Latent Semantic Analysis in a number of experiments." (Hofmann, 296)

4. NONNEGATIVE MATRIX FACTORIZATION

Next we introduce the problem of Nonnegative Matrix Factorization (NMF) as discussed a 2011 paper by Arora et all. In the Nonnegative Matrix Factorization (NMF) problem we are given an n by m nonnegative matrix M and an integer r > 0. Our goal is to express M as AW where A and W are nonnegative matrices of size n by r and r by m respectively. We can get an exact representation of this factorization or we could find an approximate one. In many applications this makes sense. When we perform this approximation, we try to minimize the Frobenius norm $||M - AW||_F$; we refer to this as Approximate NMF. This problem has a rich history spanning quantum mechanics, probability theory, data analysis, polyhedral combinatorics, communication complexity, demography, chemometrics, etc. In the past decade NMF has become enormously popular in machine learning, where A and W are computed using a variety of local search heuristics. Vavasis recently proved that this problem is NP-complete. (Arora et all, 2011) (Without the restriction that A and W be nonnegative, both the exact and approximate problems can be solved optimally via the singular value decomposition.)

We define r as the inner dimension, the nonnegative rank of M to be the smallest value of r such that there exists a nonnegative factorization of M with inner dimension r. For large r it has been proved that this problem is NP-complete. We, for the most part, are not interested in large r factorizations. This is because a large part of these factorizations is done for revealing the underlying structure as well as for dimensionality reduction. It's clear if r is large you are not reducing the storage space of your M matrix significantly. A large r does not do much for revealing the underlying structure of data either. Let's take the example as in LSI where M is a term document matrix. We would like a decomposition into a small r because we can think of r as our set of topics so we would like to decompose our documents into a small set of topics. The size of r is essentially the number of hidden variables. To take the example of our term-document matrix further let M_{ij} represent the frequency of occurance of the i^{th} term in the j^{th} document in the database. In this context, a NMF computes r topics which are each a distribution on words corresponding to the r columns of A and each document a column of M can be expressed as a distribution on topics given by the corresponding column of W. This example will be a useful metaphor for thinking about nonnegative factorization. In particular it justifies the assertion r should be small the number of topics should be much smaller than the total number of documents in order for this representation to be meaningful.

The authors of the paper pose several questions including these: What assumptions can be made about M in order to compute the NMF any faster? It has already been proven that this is NP-hard for large r, but what happens when r is small?

4.1 Simplicial Factorization

One requirement that could be necessary to make for A is that its columns be linearly independent. This is equivalent to stating that in LSI, there cannot exists a document that can be represented by two completely different convex combinations of different sets of topics. This is defended as a fair assumption. NMF with this added assumption is referred to as SF, or Simplicial Factorization. One conclusion that has been proven is that "There is an algorithm for the Exact SF problem (where r is the target inner dimension) that runs in $O((nm)^(r^2))$ time" (3). They then follow this with an even stronger statement: There is little hope of running much faster.

4.2 Hardness Result

One of the most interesting results which Arora, Ge, Kannan, and Moitra came to is about the hardness of nonnegative matrix factorization. They proved in their 2011 paper that if we can find a simplicial factorization in $nm^{o(r)}$ then 3-SAT can be solved in subexponential time. Therefore, unless the world of complexity is not as we presume, NMF cannot be solved exactly in $nm^{o(r)}$. Let us inspect from a broad view how they did this.

First, we must introduce a problem called d-SUM, which takes an input of N values in $[0, 1]$ and outputs whether there exists a set of d numbers summing to exactly $d/2$. It has been proven previously that if this has a $N^{o(d)}$ solution, then 3SAT has a subexponential algorithm (12). Arora, et. all prove their theorem about hardness by reducing d-SUM to an instance of Intermediate Simplex, which has been proved to be equivalent to Simplical Factorization.

As a quick side note, one interesting point comes out of the equality between Intermediate Simplex and Simplical Factorization. When SF is easy – when rank(M) = 2, the equivalent Intermediate Simplex problem is nolonger a multi-dimensional polyhedron, but on one dimension, so the polyhedron is simply an interval. The reduction from d-SUM to Intermediate Simplex utilizes d very specific geometric gadgets. Each i^{th} gadget has three variables $\{x_i, y_i, z_i\}$, and the solution of each gadget corresponds to one number in the original list of numbers in d-SUM. There are several constraints given are that for all gadgets, $x_i, y_i \in [0, 1]$, $z_i \in [0, 2]$. We also have a single extra variable $w \in [0, 1]$. The gadgets are used to define constraints on the polyhedron P in an instance of intermediate simplex.

As each solution to a gadget is corresponds to one number in the original list of inputs for $d - SUM$, the solution using the gadgets must simply be decoded to give a solution for $d - SUM$. The authors were able to fully prove that this reduction is complete and sound.

5. CONCLUSION

LSI solves two big problems that we run into when trying to parse text. The first of these is synonymy: multiple words having similar meanings. This can cause a problem when doing search queries. For example, in my database I search for documents relating to films", but nothing comes up because my documents use the word movies instead. The other problem is that of polysemy: a word having many meanings. When searching the word box, I get documents about people fighting instead of storage containers. LSI solves these problems by employing SVD to reveal the underlying topics of our data. Basically were creating a new basis for all our documents where our basis vectors are the eigenvectors that correspond to the underlying topics of our documents. Now if we have a document, represented as a term vector, we find out what topics its comprised of by projecting it onto our new basis. We can choose the size of our basis by only taking the k principle eigenvalues. These intuitively will be the topics that show up the most in our documents. What Papadimtriou showed in his paper is that if we make assumptions about the structure of our corpus model, then we conduct a Mathematical analysis of the quality of the information retrieval gained by LSI. We can also use the familiar concept of dimensionality reduction and perform a random linear mapping of our data to a much lower dimension to greatly speed up our LSI algorithm. This paper from Papadimitriou was one of the first analyses done on LSI and since then there has been a lot of breakthrough in the field. We studied another breakthrough called Nonnegative Matrix Factorization, which solves an issue that SVD has. In some sense, the decomposition into topics generated via SVD is inconsistent with our intuitive notion of what a topic is. This is because the eigenvectors generated by SVD have negative and positive values - these vectors are orthogonal. For example, imagine two documents - one talks about techno the other speaks about backpacking. These topics would both be negatively correlated with mentioning the word painting - that is both documents are unlikely to use that word and therefore SVD considers them similar. This is inconsistent with our intuitive understanding of grouping documents into topics. We expect similarity based on similar words only. This is the problem nonnegative matrix factorization solves - requiring the factorization to be nonnegative means that documents are judged to be similar based on what words they contain.

6. REFERENCES

S. Arora, R. Ge, R. Kannan, A. Moitra. Computing a Nonegative Matrix Factorization – Provably. 2011.

C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis. PODS 1998.

T. Hofmann. Probabilistic Latent Semantic Analysis. UAI, pp 289-296, 1999.