cmi

CHENNAI
MATHEMATICAL
INSTITUTE

MSc. Data Science

Taming Text with the SVD

Authors:
Aanchal
MDS202001
Aditi
MDS202002
Aditya
MDS202003
Animesh
MDS202004

Supervisor:
Dr Kavita Sutar
Lecturer, Chennai Mathematical Institute
ksutar@cmi.ac.in

PROJECT REPORT

Work contribution:

Aanchal MDS202001	Implementing the paper with code Plotting the graphs (figures in this paper) Parts of section 3(Examples) Finding out about related material Working out a running example (For plot in section 8, Observations) Parts of Abstract and Section 1(Introduction)
Aditi MDS202002	Finding out about related material Parts of Abstract and Section 1(Introduction) Section 4,5,6,7,8,9 and Reference Finding out about peripheral/related material and further advances Making slides and presentation Presenting part of the talk.
Aditya MDS202003	contributed to making slides and presentation (did the example of frequency matrix in section 1.1, have written section 1.2 and example 1 for section 3 and did similar part for slides also) have calculated svd and pca for the matrix of section 1.1 participated in presenting
Animesh MDS202004	Understanding the content of the paper and helping the group to understand it Collecting relevant videos and some study material for theory and code Helped Aanchal for implementing code in some places helped in making slides and reports (have written section introduction ,section 1.1, section 2 and 1st point of section 8 and similar part for the slides also.) have participated in presenting

Abstract

In SAS Text Miner, vector space models are used to represent text. In this framework, we consider distinct words of documents as variables and each document is considered as observation. But in this process the number of variables becomes too enormous to handle. As a result, dimension reduction becomes a crucial aspect of text mining solutions to easily model the documents. In order to reduce dimension we used Singular Value Decomposition(SVD) and Principal Components Analysis(PCA) on the class of documents and compare the results.

1 Introduction

Mining text at a semantic level is a very difficult task because of the complexity involved in understanding documents. Thus, most text mining software focus at the level of relationships between words in documents. We try to understand the pattern that exists between documents by examining the frequency of all words in each document.
Here we use the terms 'Collection' ,'terms', and 'documents' . Collection is a finite set of documents and document may be a paragraph,headline,any kind of sentence and terms are each words that will be use to make the 'bag of words.

1.1 Document to vector transform

We use the vector space model to represent documents which uses the 'bag of words' approach. In this approach, collections of n documents are now going to transform to a vector of length m, where m is the number of unique terms that are indexed in the collection. **The i-th entry is simply the frequency of term i in the document. The vector for each document is generally very sparse (i.e., it contains a high proportion of zeroes) because few of the terms in the collection as a whole are contained in any one given document**.

Example:
Suppose we have 3 documents. Number of times a term appears in each document are as follows:

Term	d1	d2	d3
"profile"	0	0	1
"frequency"	2	1	0
"mean"	1	0	1
"load"	0	0	1
"become"	1	1	0
"exit"	0	1	0

If m is the number of distinct terms in a collection of n documents, then let A be the $m \times n$ matrix that represents this collection. This matrix, where terms are rows and documents are columns, is known as the **term-document frequency matrix**.

Collections of even a few documents can contain a thousand of words. This presents a problem when building predictive models due to the "curse of dimensionality". As the collection size increases, the term-document frequency matrix becomes very sparse because very few of the unique words in the collection are actually contained in any single document

Here we can measure similarity and distance between two documents using dot product and angle. Suppose for the documents d1 ,d2 and d3

$$d1.d2 = 3$$

and

$$d1.d3 = 1$$

then d1 can be said to be more similar to d2 than d3.

1.2 Dimension reduction

For a practical purpose due to high dimension of documents, it is harder to work with this term-document frequency matrix. That is why we need to project the high dimension vector space to lower dimension.Here each terms are each dimensions. This dimension reduction procedure can be done in two way

1) **Eliminating terms :**

Start lists: keep only a fixed, predetermined set of terms for the analysis. This is done by learning from previous analysis of similar data, and removing terms that were not useful in previous data.

Roll-up terms: restrict the number of variables to the specified number of highest weighted terms. This approach is dynamic, that is we assign weights to each term during the current analysis itself and take the highest weighted terms.

Text Miner also includes a few other methods, which does not guarantee dimension reduction at a large scale like the previous 2 methods. They are:

- **Stop Lists** - Removing a set of predetermined terms from analysis.
- **Synonym Lists** - Remove terms that are similar by mapping one term to another.
- **Stemming** - Map multiple term to a root form. based on the root form.
- **Remove singly occurring terms in the document.**
- **Remove numeric terms in the document**

Using the above mention techniques we did not address problems like:

Synonymy: Words that are not similar in any sense can be used in a similar meaning, like a computer can be said to be 'frozen' or 'hanged', but they are very different in meaning actually.

Polysemy: Words that can be used in different contexts. For example, we can use the word 'set' in several different forms.

Term Correlation: Words that are related to each other even though they do not mean similar things, like 'error' and 'message',

This makes cleaning by removing terms not so efficient at reducing dimension as large portion of terms still remain.

That is why we use PCA or SVD kind of things on our matrix for dimension reduction.

2) **Using dimension reduction tools like SVD or PCA**

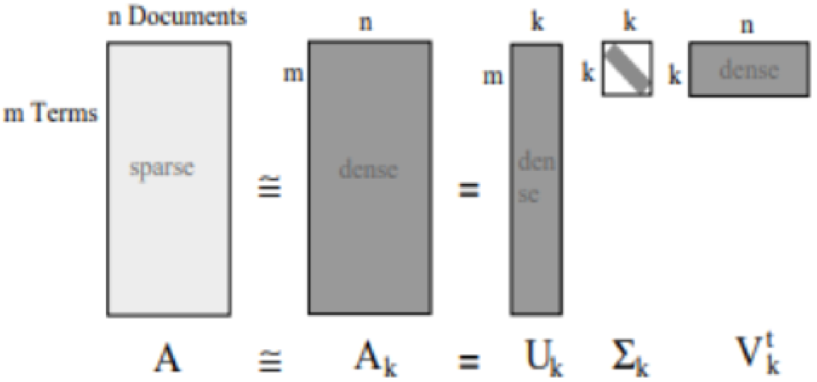
For dimension reduction here we do projection, But we want to lose data as much less as possible. That is why normal projection to a axis or a plane can cause a lose of data. But using SVD technique we project to the space which gives best least square fitting .And PCA made projection of data to those dimensions which have higher variance. The details procedure are described in next section.

2 Procedure

SVD: If A be an $m \times n$ matrix, then singular value decomposition (SVD) of A is $A = U\Sigma V^t$,
where U is an m × r orthogonal matrix whose columns make up the left singular vectors of A , Σ is an $r \times r$ dimensional diagonal matrix whose diagonal elements are termed singular values and V is an r × n orthogonal matrix whose columns form the right singular vectors of A.
Singular values are all greater than or equal to zero and, by convention, are ordered from largest to smallest.
Now suppose that rank of the matrix A is r and we want to approximate A by a rank k matrix where $k \leq r$. Then according to SVD rules we can generate generates a rank k matrix, A_k, that is the best rank-k approximation to A in terms of least-squares best fit and

$$A \approx A_k = U_k \Sigma_k V_k^t$$

Note that even though A is very sparse, A_k is meant to be a dense matrix of much lesser dimension(when k is sufficiently smaller than r) but not losing significant information from A.



Here we can see that the columns of U_k forms an orthonormal basis for the k-dimensional document space. So, an m dimensional document d can be projected to k-dimensional space using this U_k as

$$\hat{d} = U_k^T d$$

And ith row of U_k gives weight to the ith component of d to form $\hat{d}$.

PCA:

PCA is a dimension reduction tool like SVD, here we project the data to those dimensions which have higher variances.
Suppose that A is our term document frequency matrix of $m \times n$,then subtracting the mean of each column of A from each entry in that column and form the matrix B. And Covariance matrix $C = BB^T$. Now $C = X\Lambda X^{-1}$ is eigenvalue decomposition of C. From property of SVD we know that, each columns of X are left singular vector of B and root of each eigen vector are singular value of B. And Now we sort these eigen vector and eigen values on their principal values, where ith principal value $= \frac{\lambda_i}{\sum_{j=1}^{r} \lambda_j}$
After sorting suppose we get U from X. Now similarly like SVD for reducing the matrix A to a lower rank matrix A_k (suppose rank of $A = r$ and $k < r$)

$$\hat{d} = U_k^t \cdot d$$

where U_k is consisting of first k columns of U.

3 Examples

We performed SVD and PCA on three examples, one small 6 dimensional example with 3 documents and two high dimensional datsets which will be discussed later in this section. First, let us consider the small dataset. Suppose our term document matrix A is the one which was shown before in section 1.1:

Term	d1	d2	d3
"profile"	0	0	1
"frequency"	2	1	0
"mean"	1	0	1
"load"	0	0	1
"become"	1	1	0
"exit"	0	1	0

SVD: Now after doing SVD on it for k=2

So our U_2 , Σ_2 and V_2 are

$$U_2 = \begin{bmatrix} -0.06 & 0.55 \\ -0.78 & -0.18 \\ 0.36 & 0.55 \\ 0.06 & 0.55 \\ 0.48 & -0.18 \\ 0.18 & -0.18 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2.82 & 0 \\ 0 & 1.73 \end{bmatrix},$$

$$V_2^t = \begin{bmatrix} 0.84 & 0.51 & 0.17 \\ 0 & -0.32 & 0.95 \end{bmatrix}$$

Now, we obtain $\hat{A} = U_2^t \cdot A = \begin{bmatrix} 2.39 & 1.43 & 0.48 \\ 0 & -0.55 & 1.64 \end{bmatrix}$, which gives us the reduced document matrix.

Dividing each column vector by its norm, we get the final reduced term document matrix as:

$$\hat{A} = \begin{bmatrix} 1 & 0.93 & 0.28 \\ 0 & -0.36 & 0.96 \end{bmatrix}$$

We can now represent the original 6 dimensional matrix in 2 dimensions!

PCA: In PCA, we have reduced matrix A similarly for PCA:

$$\hat{A} = \begin{bmatrix} -0.96 & -0.63 & -1.59 \\ -0.79 & 0.91 & -0.12 \end{bmatrix}$$

Dividing each vector by its norm, we get the final reduced matrix as:

$$\hat{A} = \begin{bmatrix} -0.77 & -0.57 & -1 \\ -0.64 & 0.82 & -0.08 \end{bmatrix}$$

Here too, we can represent the 6-dimensional documents in just 2-dimensions.

To compare both the approaches, we implemented SVD and PCA on high dimensional datasets. We used two datsets: dataset containing text files related to programming from archives.textfiles.com and movie reviews dataset from Movie Review Data.

The programming dataset contains 261 documents with 77959 unique words(terms) and the movie review data contains 2000 documents with 39399 unique words. We considered columns as documents and rows as unique words(terms) elsewhere in the report but in our code columns correspond to unique words and rows correspond to documnents as we wanted linear combinations of items instead of documents.

The raw data contained many uninterpretable words as well as terms like numbers that are less significant for predictive modelling. So first we performed cleaning of the text by removing non english words, words containing only numbers and words with sing letter. The dimensions were then reduced to 261 × 27572 for programming dataset and 2000 × 18481 for movie reviews dataset.

As mention in section 1.2, SAS text miner includes a few methods to reduce dimensions. We had applied methods 4 and 5 while cleaning process and after cleaning we applied 1, 2 and 3 i.e. removed stop words and mapped all the synonyms or the stem words a word to that word. We observed that after applying all these methods we still have 17223 unique terms left in programming dataset and 13671 unique terms left in the movie reviews dataset. These steps ere not highly effective in reducing the dimensions and this was mentioned in section 1.2 that these does not guarantee dimension reduction at a large scale.

Thus, as we still are left with a very large dimensional dataset, there arises a need to look for other better and more effective dimension reduction techniques like SVD and PCA. SAS text miner uses SVD and as PCA is a similar technique, we have studied and compared both. So, we performed SVD and PCA on the dataset remaining after applying the aforementioned methods.

Link to the notebooks:

- Programming dataset
- Movie reviews dataset

4 Advantage of using SVD

In real life data set, to use a data mining model, data reduction is an essential part. **SVD** comes handy in this situation in order to derive significantly fewer variables. As we know the huge numbers of variables are extremely difficult to handle without dimension reduction, we have two options in SAS text miner to reduce dimension.

- Term Elimination
- SVD

Term elimination will cost us hugely by loosing important information as we are blindly eliminating variables without considering their significance. but the later technique, **SVD** allows us to reduce the number of variable significantly without causing as much data loss as in term elimination, since in this process we are taking linear combination of all the variables and thus we are considering the importance of each variable properly.

5 Drawbacks of using SVD

- SVD is very resource intensive, computationally. It requires a large amount of RAM to complete SVD computation.
- Since SVD dimensions are linear combinations of original variables, sometimes it blurs out distinguishable document.
  For example, if we have two documents, one explaining about wheat and other one explaining about corn, then SVD blurs out these two distinguishable documents. But a simple term based classification could have easily distinguish these documents.
- SVD fails to analyse documents with rich interaction between terms and documents. For instance, suppose our collection of documents contain only author's name. Then there is almost no co-occurrence of terms in any of the documents. Thus we will form very sparse term-frequency matrix, where rank is very small and SVD will place most of them at the origin of the reduced sub-space.

6 Number of Dimension

The choice of number of dimension is essential. If number of dimension is too less then the model will fail to describe the crucial aspect of the collection. If the it is too high then there will be unnecessary noise. Thus training a model will be really difficult. So choosing a proper number for dimension is a really crucial aspect of text mining.
But the question is **"How to choose the number of dimension?"**. This part is still being researched. In practice there is an upper bound of few hundred dimensions. In some cases, we stop when the total variance of all data starts to converge.

For SVD, stop when the total variance of the data reached at a certain percentage by using sum of the first k v_i's obtained by the following formula,

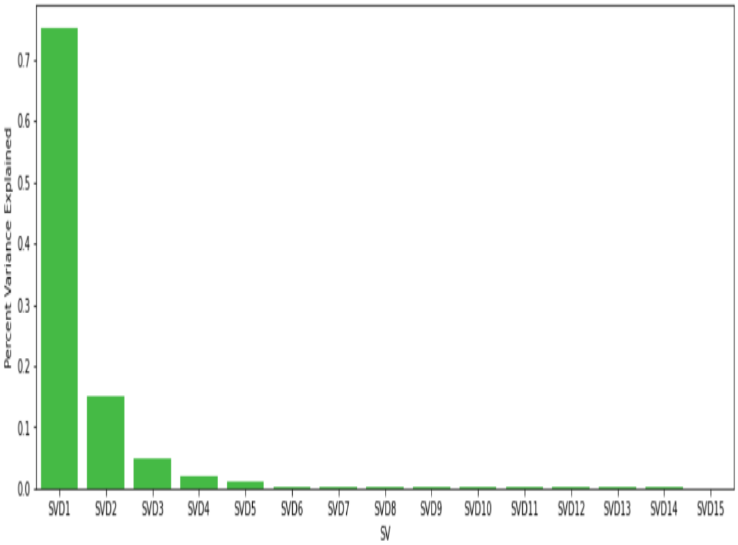
$$v_i = \frac{\sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2}$$

Suppose after k steps, the values started to reach a certain percentage, then we will take the k steps only.
For PCA, we use eigen value instead of singular value and thus the equivalent formula for PCA is as follows,

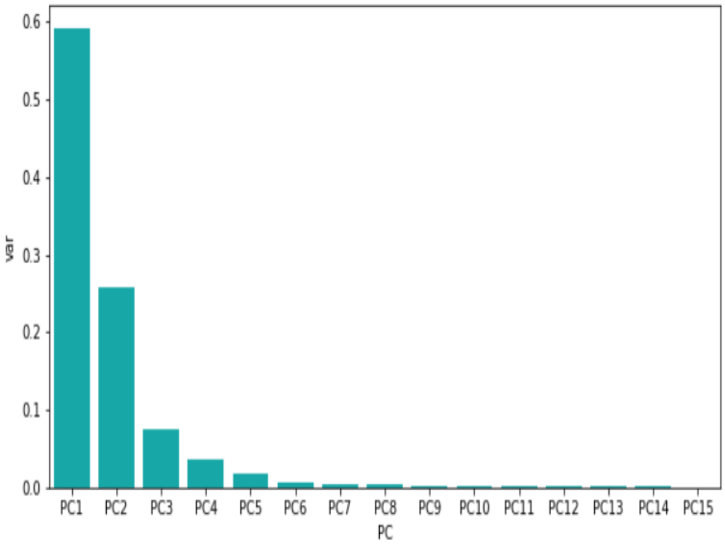
$$v_i = \frac{\lambda_i}{\sum_{j=1}^{r} \lambda_j}$$

The scree-plots of SVD and PCA will give a proper idea about the convergence.

For SVD, the scree-plot is as follows.



For PCA, the scree-plot is as follows.



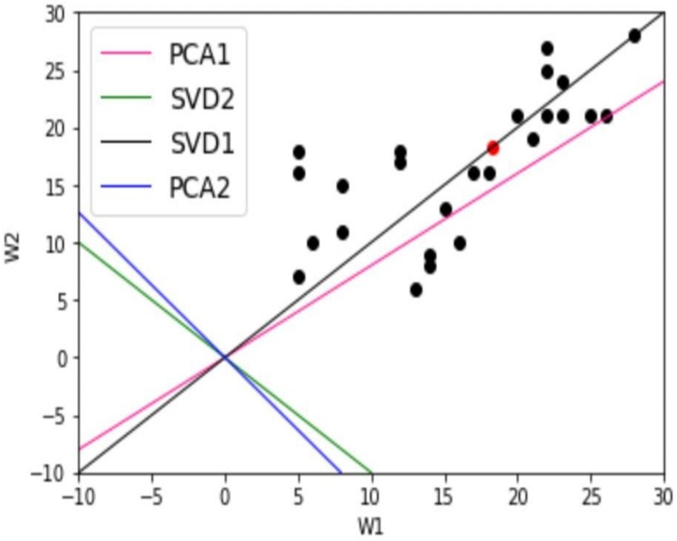
In this graph, observe that, after 5th dimension the total variance has almost started to converge at 0. Hence we can choose k as 5, although the choice of number of k depends on us.
But this method doesn't always work. Most of the time it gives accurate results. For instance, the author has worked with 10440 texhnical support documents in which 75% of the variance was associated with first 73 dimensions. But TM do not apply this method yet.

7 SVD in TM

SAS text miner uses SVD only for dimension reduction. PCA maps tens of thousands of variables to merely hundred variables, making it hard to interpret, whereas in SVD, after all the computation is complete we analyze the frequency of a particular term. In smaller cases, suppose in a collection of 20 documents PCA maps 20 dimensions to approximately 2-3 dimensions. But as the number increases it becomes really hard to make reasonable judgement in order to interpret the data set properly. But SVD analyse the frequency of terms after all the calculations are done and in this way it takes account of significance of all the variables accordingly their corresponding weights.

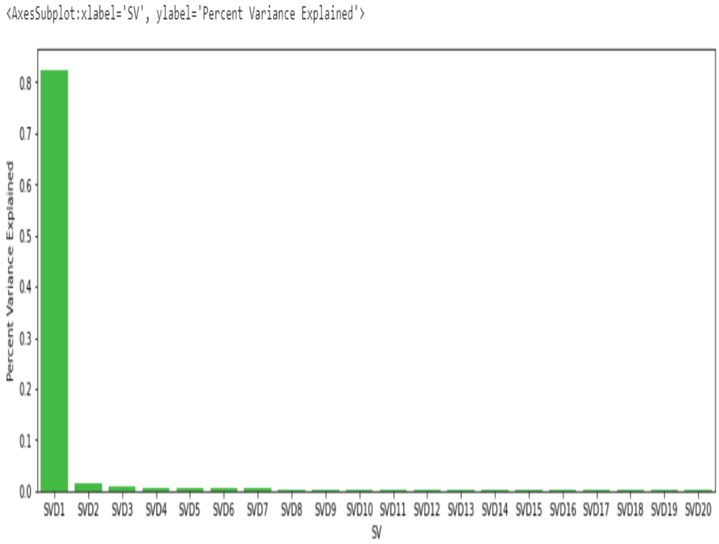
8 Observations

1. Suppose that we have documents with only two words 'W1' and 'W2' and we have plot that one the below figure. Now we draw SVD1 ,PCA1 and SVD2 ,PCA2. From the plot one can see that PCA1 and SVD1 are not the same line as we already said that PCA choose those dimension which has high variance and SVD go for the best fitting line in least square sense. New basis created by SVD and PCA are orthonormal basis which also reflected in this figure.



   The SVD1 line is affected by the shifting the value of W2 when W1 reamains the same while the PCA1 line is not influenced by this secondary variation.
   SVD and PCA line will be close or not, totally depends on the data and if we use mean adjusted matrix for SVD and PCA, we will get same result from both of these.

2. We had worked with a collection movie review data sets, where our computed first singular value has really high value compared to the other singular values. Hence all the document made a cloud of cluster around the first SVD line only. But while in real data sets, we saw that although some documents are really close together in a cluster, they were distinguishable on a finer level. The following scree-plot was obtained from applying SVD on the data set.



```
<AxesSubplot:xlabel='SV', ylabel='Percent Variance Explained'>
```

If the data is too imbalanced then the first few singular/eigen values will dominate the other singular/eigen values, but it may happen the other eigen/singular values are significant.

9 Conclusion

- SAS Text Miner uses the vector space model for representing text. We have formed a text-frequency matrix, where documents form columns and distinct terms form rows.
- There are two possible way to reduce dimensions, either eliminate terms, which is kind of trial and error approach and we cannot be certain that we will obtain correct result, or use SVD to generate new variables, which takes original variables and then form new variables from them, so it is more reliable.
- Formally SVD and PCA is computationally identical. The only difference is PCA chooses orthogonal dimensions that optimally account for the variance in the data and SVD simply selects orthogonal dimensions that reduce the sum of the perpendicular distances from the original observations to the new axis.

References

[1] R. Albright, J. A. Cox, K. Daly. *Skinning the Cat: Comparing Alternative Text Mining Algorithms for Categorization, Proceedings of the 2nd Data Mining Conference of DiaMondSUG*, Chicago, IL. DM Paper 113, 2001. Addison-Wesley, Reading, MA, 1999.

[2] I. T. Jolliffe, *Pincipal component analysis*. Springer, New York, 1986.

[3] StatQuest: Principal Component Analysis (PCA), Step-by-Step

[4] SVD - Singular value decomposition