IPRJ708P Project
Report

Experiments with Random Forest and LSA

(Feature Importance)

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Information Technology

Submitted by

Roll No	Names of Students
IIT2013194	Priyank Upadhya
IIT2013198	Anurag Das

Under the guidance of
Prof. Sudip Sanyal

Department of Information Technology
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
Allahabad, UP, India – 211012

Summer Semester 2016

Department of Information Technology

Indian Institute of Information Technology Allahabad

Certificate

This is to certify that this is a bonafide record of the project presented by the students whose names are given below during Seventh Semester and year 2016 in partial fulfilment of the requirements of the degree of Bachelor of Technology in Information Technology.

Roll No	Names of Students
IIT2013194	Priyank Upadhya
IIT2013198	Anurag Das

Prof. Sudip Sanyal
(Project Mentor)

Date:

Abstract

Latent Semantic Analysis is a Natural Language Processing technique for finding the hidden concepts and relationship between different words and documents. It is a dimension reduction algorithm that provides ways to cope up with classical problems such as polysemy and synonymy. Random Forest on the otherhand is an ensemble supervised machine learning algorithm used for classification and regression. It is amalgamation of various decision trees which are constructed using randomly selected features.
This report gives a description about an experiment conducted to find the correlation between the singular values obtained corresponding to the feature vectors in the reduced dimensional space after applying Latent Semantic Analysis and the feature importance after fitting the transformed data on Random Forest.

Contents

List of Figures

Chapter 1

Problem Definition

1. To find the relationship or correlation between the variance obtained corresponding to the feature vectors in the reduced dimensional space after applying Latent Semantic Analysis and the feature importance (feature in the reduced dimension space) obtained after fitting the transformed data on Random Forest and Mutual Information Model.

2. To check if the graph obtained by plotting the variance or singular value obtained using LSA and by plotting the feature importance after applying Random Forest and Mutual Information are similar.

3. Perform classification task for reduced features (for all reduced features and selected top k) and verify the accuracy obtained from the graphs obtained.

Chapter 2

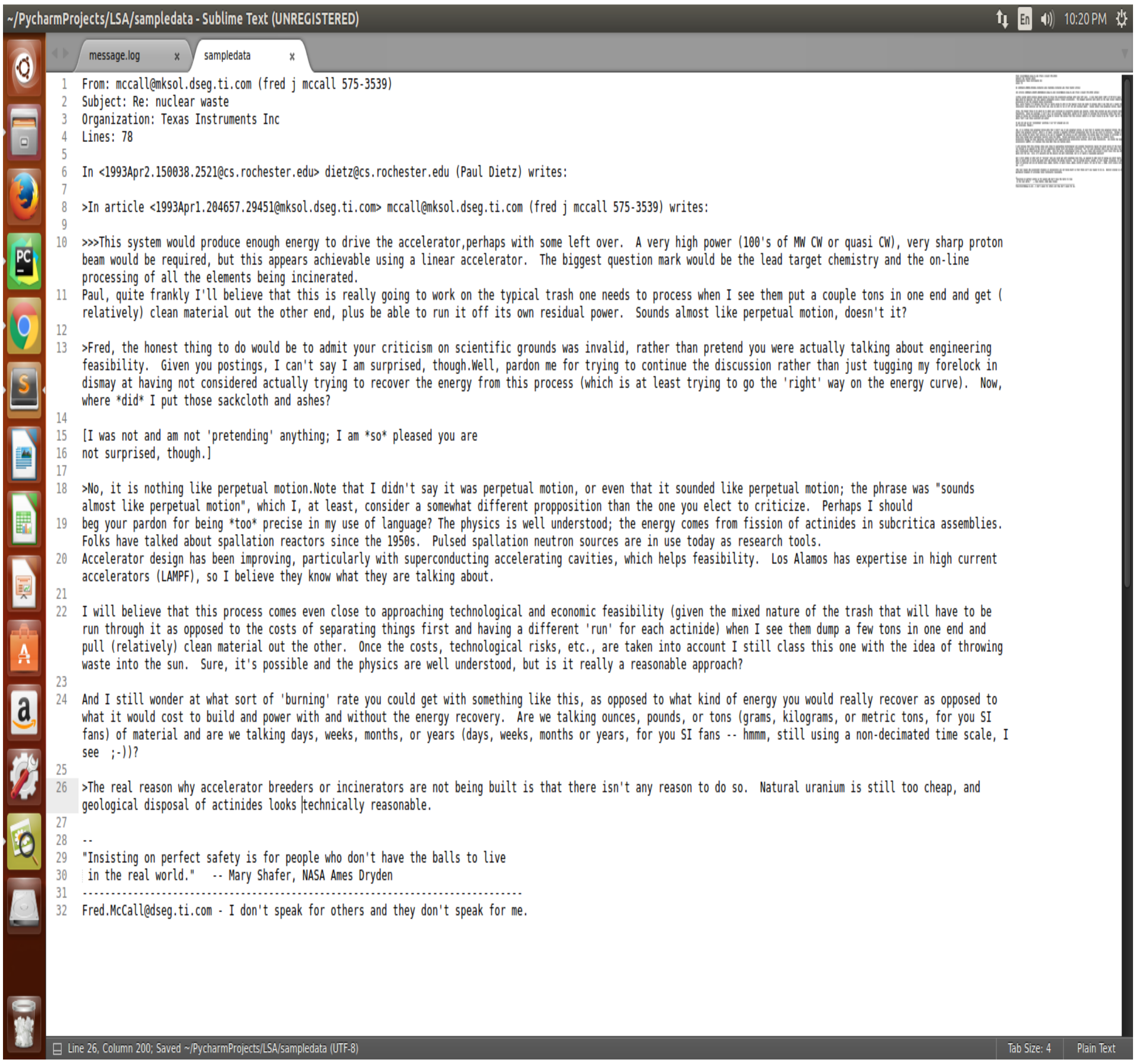
About the Dataset

The dataset used is the 20 newsgroup dataset. It comprises around 18000 newsgroups posts. The posts are on 20 topics which are split in two subsets: training data and testing data. It comes as a standard data with sklearn.datasets (datasets available with scikit learn).[8] The categories on which the data is classified are -

1. 'alt.atheism'
2. 'comp.graphics'
3. 'comp.os.ms-windows.misc'
4. 'comp.sys.ibm.pc.hardware'
5. 'comp.sys.mac.hardware'
6. 'comp.windows.x'
7. 'misc.forsale'
8. 'rec.autos'
9. 'rec.motorcycles'
10. 'rec.sport.baseball'
11. 'rec.sport.hockey'
12. 'sci.crypt'
13. 'sci.electronics'
14. 'sci.med'
15. 'sci.space'
16. 'soc.religion.christian'
17. 'talk.politics.guns'
18. 'talk.politics.mideast'
19. 'talk.politics.misc'
20. 'talk.religion.misc'

We have taken 4 categories for our purpose -
1. 'alt.atheism'
2. 'talk.religion.misc'
3. 'comp.graphics'
4. 'sci.space'

These four categories together comprises of 3387 documents.The total number of feature after applying tf-idf vectorizer is 10000. So the dataset on which Latent Semantic Analysis and Random Forest will be applied is a matrix of size 3387 * 10000. Sample document -




```
From: mccall@mksol.dseg.ti.com (fred j mccall 575-3539)
Subject: Re: nuclear waste
Organization: Texas Instruments Inc
Lines: 78

In <1993Apr2.150038.2521@cs.rochester.edu> dietz@cs.rochester.edu (Paul Dietz) writes:

>In article <1993Apr1.204657.29451@mksol.dseg.ti.com> mccall@mksol.dseg.ti.com (fred j mccall 575-3539) writes:

>>>This system would produce enough energy to drive the accelerator,perhaps with some left over.  A very high power (100's of MW CW or quasi CW), very sharp proton beam would be required, but this appears achievable using a linear accelerator.  The biggest question mark would be the lead target chemistry and the on-line processing of all the elements being incinerated.
Paul, quite frankly I'll believe that this is really going to work on the typical trash one needs to process when I see them put a couple tons in one end and get (relatively) clean material out the other end, plus be able to run it off its own residual power.  Sounds almost like perpetual motion, doesn't it?

>Fred, the honest thing to do would be to admit your criticism on scientific grounds was invalid, rather than pretend you were actually talking about engineering feasibility.  Given you postings, I can't say I am surprised, though.Well, pardon me for trying to continue the discussion rather than just tugging my forelock in dismay at having not considered actually trying to recover the energy from this process (which is at least trying to go the 'right' way on the energy curve).  Now, where *did* I put those sackcloth and ashes?

[I was not and am not 'pretending' anything; I am *so* pleased you are
not surprised, though.]

>No, it is nothing like perpetual motion.Note that I didn't say it was perpetual motion, or even that it sounded like perpetual motion; the phrase was "sounds almost like perpetual motion", which I, at least, consider a somewhat different propposition than the one you elect to criticize.  Perhaps I should
beg your pardon for being *too* precise in my use of language? The physics is well understood; the energy comes from fission of actinides in subcritica assemblies. Folks have talked about spallation reactors since the 1950s.  Pulsed spallation neutron sources are in use today as research tools.
Accelerator design has been improving, particularly with superconducting accelerating cavities, which helps feasibility.  Los Alamos has expertise in high current accelerators (LAMPF), so I believe they know what they are talking about.

I will believe that this process comes even close to approaching technological and economic feasibility (given the mixed nature of the trash that will have to be run through it as opposed to the costs of separating things first and having a different 'run' for each actinide) when I see them dump a few tons in one end and pull (relatively) clean material out the other.  Once the costs, technological risks, etc., are taken into account I still class this one with the idea of throwing waste into the sun.  Sure, it's possible and the physics are well understood, but is it really a reasonable approach?

And I still wonder at what sort of 'burning' rate you could get with something like this, as opposed to what kind of energy you would really recover as opposed to what it would cost to build and power with and without the energy recovery.  Are we talking ounces, pounds, or tons (grams, kilograms, or metric tons, for you SI fans) of material and are we talking days, weeks, months, or years (days, weeks, months or years, for you SI fans -- hmmm, still using a non-decimated time scale, I see  ;-))?

>The real reason why accelerator breeders or incinerators are not being built is that there isn't any reason to do so.  Natural uranium is still too cheap, and geological disposal of actinides looks technically reasonable.


--
"Insisting on perfect safety is for people who don't have the balls to live
 in the real world."   -- Mary Shafer, NASA Ames Dryden
------------------------------------------------------------------------------
Fred.McCall@dseg.ti.com - I don't speak for others and they don't speak for me.
```

Chapter 3

tf-idf vectorizer

In information retrieval, tf-idf [15] also known as term frequency - inverse document frequency is a statistical measure that depicts the importance of a word to a document. The tf-idf value is proportional to the number of times a word appears in a document and is inversely proportional to the number of times the word appears in the corpus.

Term Frequency:
Term Frequency measures that how frequently a word appears in a document and is proportional to it. For a term t in document d,

$$\mathrm{tf}(t,d) = 1 + \log(f_{t,d})$$

where $f_{t,d}$ is number of occurences of t in d

Figure 3.1: Logarithmic Term Frequency

Inverse Document Frequency:

Inverse document frequency is the measure of information provided by the word. The idf concept helps to take into account the fact that some words appear more frequently in general.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

N: total number of documents in the corpus $N = |D|$

$|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Figure 3.2: Inverse Document Frequency

Then tf-idf is calculated as

$$tfidf(t, d, D) = tf(t, D).idf(t, D)$$

Chapter 4

Random Forest

The first random forest algorithm was created by Tin Kam Ho while the extension of algorithm was given by Leo Breiman and Adele Cutler. Random forest is an ensemble machine learning algorithm used for classification and regression problems. It is a collection of many decision trees that work independently of each other on same data but with different features. It is based on the fact that a single decision tree might wrongly predict/classify the data but given N decision trees each of which works on same data but with different features, if X decision trees predict correctly and Y decision trees give wrong results ($N = X+Y$) then assuming that $X >> Y$, the forest of N trees will also give the same result as given by X trees. The features for each decision tree are selected randomly.[7]

The Random Forest Algorithm:

1. For b = 1 to B:

   (a) Draw a bootstrap sample Z* of size N from the training data.

   (b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, untill the minimum node of size n_{min} is reached.

      i. Select m variables at random from p variables.

      ii. Pick the best variable/split - point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}^B{}_1$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}^B_{rf}(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}^B_1$.

Figure 4.1: Random Forest Algorithm

There are two measures used for splitting a decision tree[7]:
1) Gini Index :-

$$Gini = \sum_{i \neq j} p(i)p(j)$$

$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

$$GiniGain = Gini - Gini(A)$$

Figure 4.2: gini index

$p(i)$ is the probability of occurence of that class i.
$p(i/v)$ is the probabilty of occurence of class i wrt class v.
GiniGain is the gain obtained after splitting at node A

2) Entropy -

$$I = -\sum_{c} p(c) \log_2 p(c)$$

$$I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

$$I(A) = -\sum_{v} p(v) \log_2(p(v))$$

$$GainRatio(A) = \frac{Gain(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

Figure 4.3: Entropy measure

$I(A)$ is the amount of information needed to determine the value of attribute A

I_{res} is the weighted sum for the amounts of information for the subsets after applying A

Chapter 5

Latent Semantic Analysis

Latent Semantic Analysis sometimes called Latent Semantic Indexing was patented in 1988 by Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer, Karen Lochbaum and Lynn Streeter. The main aim of Latent Semantic Analysis is to reduce the number of features and find the hidden relationship between different features and different documents.

Considering the dataset contains M documents, where each document contains several passages and each passage contains many words, let there be N distinct words. Each distinct word is assumed to be a feature. Since N can be a very large number, Latent Semantic Analysis tries to reduce this N that is it tries to reduce the feature space since it is mathematically as well as computationally costly to work on large N.[3][4]

The NxM term by document TF-IDF matrix is factorized using Singular value Decomposition.

[1]

$$X = USV^T$$

$$\begin{array}{ccccccc} & X & & U & S & & V^T \\ & (\mathbf{d}_j) & & & & & (\hat{\mathbf{d}}_j) \\ & \downarrow & & & & & \downarrow \\ (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = (\hat{\mathbf{t}}_i^T) \rightarrow & \Big[[\mathbf{u}_1] \dots [\mathbf{u}_l] \Big] & \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix} \end{array}$$

Figure 5.2: SVD decomposition process

Where, U is called as Left Singular Vectors of X, S is a diagonal matrix that contains Singular Values of X, sorted in descending order and V is called as Right Singular Vectors of X. Here U and V are orthogonal. U contains the Eigen vectors of XX^T and V contains the Eigen vectors of X^TX. Mathematically, the singular values of matrix X are the square root of the Eigen values of X^TX. Selecting $k << n$ top most singular values along with their respective left singular vectors and right singular vectors give us a matrix X' such that the variance covered by k features is maximized.[5]

$$X_k = U_k S_k {V^T}_k$$

Figure 5.3: Dimensionality reduction

Where, U_k is N x K matrix , S_k is a K x K matrix and V_k^T is a K x M matrix. Each column vector of U_k is a principal component (derived new axis) which is a linear combination of original features.

$$PC1 = \emptyset_1 x_1 + \emptyset_2 x_2 + \dots$$
$$PC2 = \emptyset_{11} x_1 + \emptyset_{22} x_2 + \dots$$
$$\vdots$$

Figure 5.4: Principal Components

Where x_1,x_2, are old features while PC_i is the i^{th} principal component (new axis). Let Y be a matrix containing new data points of the transformed data, then each new data point is derived by projecting the older data given by matrix X where each row was a feature (axis) and each column vector depicted a point on those axis, on the new plane where the axis's are given by principal components. [2]

$$Y = U^T X$$

Figure 5.5: Data Projection

Each row vector of Y will be a new point on the new reduced plane having only k features.

Chapter 6

Mutual Information

Mutual Information [10] of two random variables provides information about the mutual dependence of one variable on the other. It gives us an estimate about the uncertainty of second random variable given that we know the first one. We have used selectKBest [11] module of sklearn.feature_selection.mutual_info_classif for our work.

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t)P(C = e_c)}$$

Figure 6.1: mutual information

e(t) = 1 means that the document contains the term
e(c) = 1 means that the document belongs to class c
U is a random variable that takes values e(t) = 1 or e(t) = 0
C is a random variable that takes values e(c) = 1 or e(c) = 0

Chapter 7

Metrics used for Classification

We have used Support Vector Machine for classification purpose. The kernel is linear with error term C as 1.9 .Since the tf-idf weights were negative so we couldn't use Naive Baiyes as it required positive weights. The metrics [12] [13] for the classification used are -

Accuracy :
Accuracy refers to the closeness of a measured value to a standard or known value.

Precision :
Precision is the fraction of retrieved documents that are relevant to the query. It is the ratio of count of relevant-retrieved documents to the count of retrieved documents.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Figure 7.1: Precision

Recall :
Recall is the fraction of documents that are relevant to the query that are successfully retrieved. It is the ratio of count of relevant-retrieved documents to the count of relevant documents.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Figure 7.2: Recall

F-score :
Mathematically, it is the harmonic mean of the precision and recall. Since we cannot predict the performance with just precision or recall we need F-score which takes both of them into account.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Figure 7.3: F score

Chapter 8

Work Performed

The tasks performed are :-

1. Data cleaning
2. tf-idf vectorizing
3. Applying Latent Semantic Analysis
4. Applying Random Forest for feature selection
5. Applying mutual Information for feature selection
6. Compare the results obtained for feature importances
7. Applying Classification task

   a) On entire data without dimensionality reduction

   b) On dimensionally reduced data for top k for LSA, Random forest and Mutual Information for top k features
8. Compare the results obtained after classification task

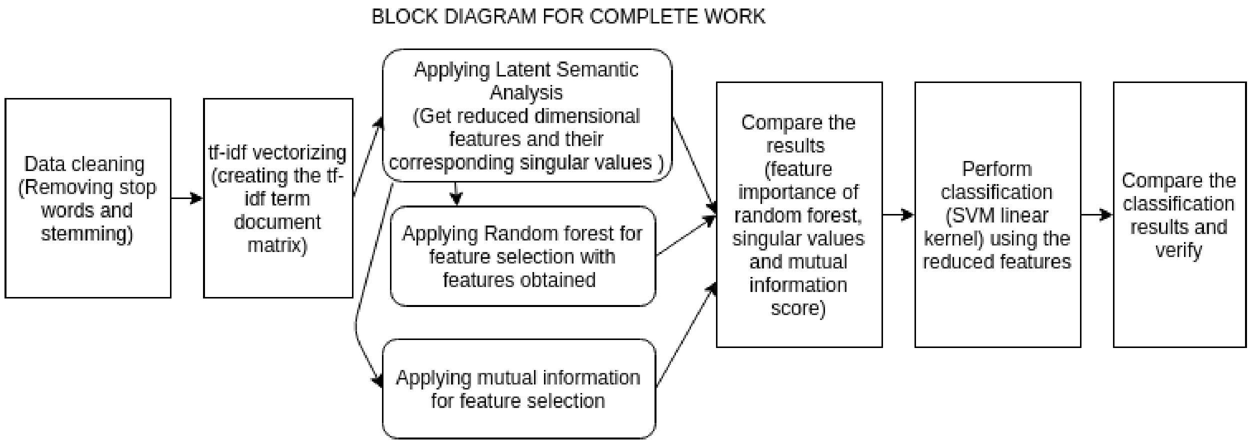


Figure 8.1: Workflow diagram

8.1 Data cleaning and tf-idf vectorizing

Steps :-

1. The stop words like a, an, the, or,was, have, etc are removed. The standard extended stop words of the scikit learn is used.
2. Snowball stemmer [14] is used which is an improvement of Porter stemmer to stem the words to their root form.
3. The term frequency (tf) and inverse document frequency (idf) is also calculated. (tf-idfvectorizer of scikit-learn is used for this purpose)
4. The term document matrix is prepared.

8.2 Applying Latent Semantic Analysis

Steps :-

1. Applying Single Value Decomposition on the input document-term matrix.
2. For SVD, svds() of scikit-learn from ARPACK library is used. (tf-idfvectorizer of scikit-learn is used for this purpose)
3. The matrix U, S, V are obtained where U is Document-Feature matrix, S is the singular value matrix, V is the feature-term matrix.
4. The V matrix is multiplied with initial X matrix to get the Document-feature matrix ie the data in transformed dimension.
5. The S matrix denotes the singular value or the variance of the corresponding features.

8.3 Applying Random Forest

Steps :-

1. The Document-feature matrix is obtained.
2. Corresponding target value is mapped with the Document feature matrix.
3. Random forest from scikit learn is applied on this data.
4. The feature importance is obtained for each feature.

8.4 Applying Mutual Information

Steps :-

1. For the Document Feature matrix obtained, target value (ie labels) are attached.
2. We perform feature selection (mutual information based) for the data
3. The Mutual Information score for each feature is obtained.

8.5 Compare the results obtained from graph

Steps :-

1. The S value obtained after applying LSA is plotted on a graph.
2. The feature importance after applying Random forest is plotted on another graph.
3. The Mutual information score for each feature is plotted.
4. All the three graphs are compared.
5. Correlation value is also obtained for (S (Singular Value) and feature importance) and (Random Forest and Mutual Information).
6. p-value is also obtained for (S(singular value) and feature importance) and (Random Forest and Mutual Information).

8.6 Applying Classification Task

Steps :-

1. For the Document Feature matrix obtained, target value (i.e labels) are attached.
2. For both (all reduced features and top k reduced) classification is performed with sklearn.svm.SVC with linear kernel.
3. The above classification is performed with various values of k and different numbers of total features.
4. The results obtained are compared.

Chapter 9

Result obtained

Sample Results

1. number of features = 600
2. reduced number of features = 300

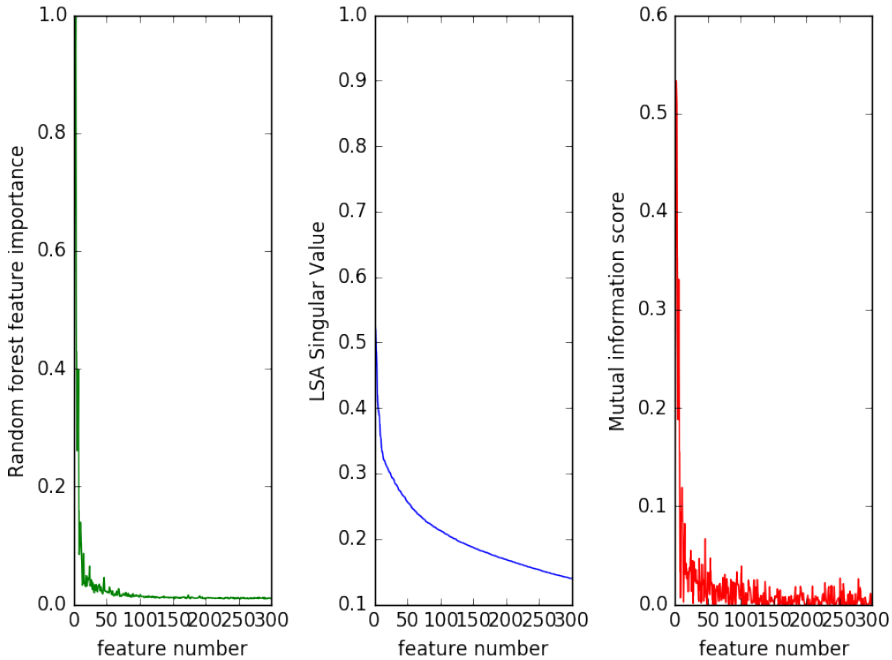


Figure 9.1: graph for features=600,reduced features=300

1. number of features = 500
2. reduced number of features = 50

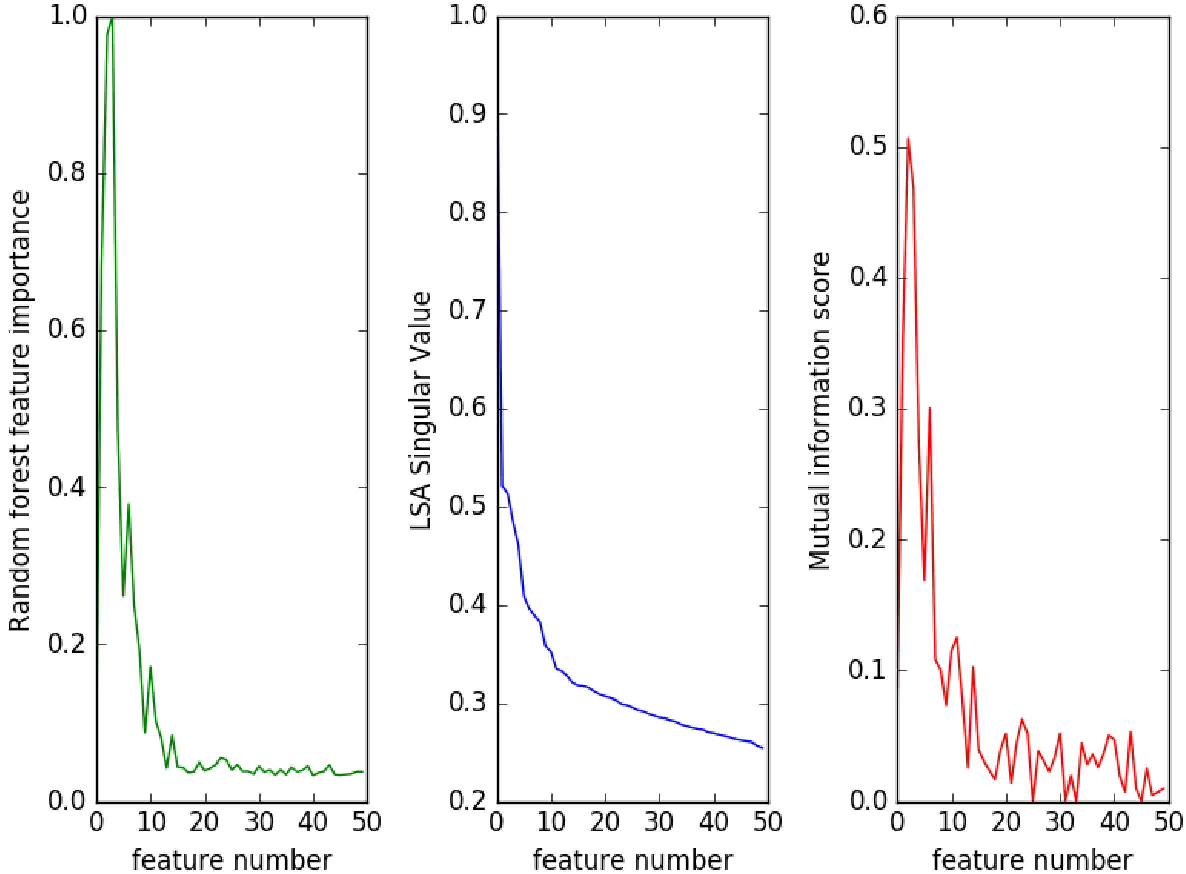


Figure 9.2: graph for features=500,reduced features=50

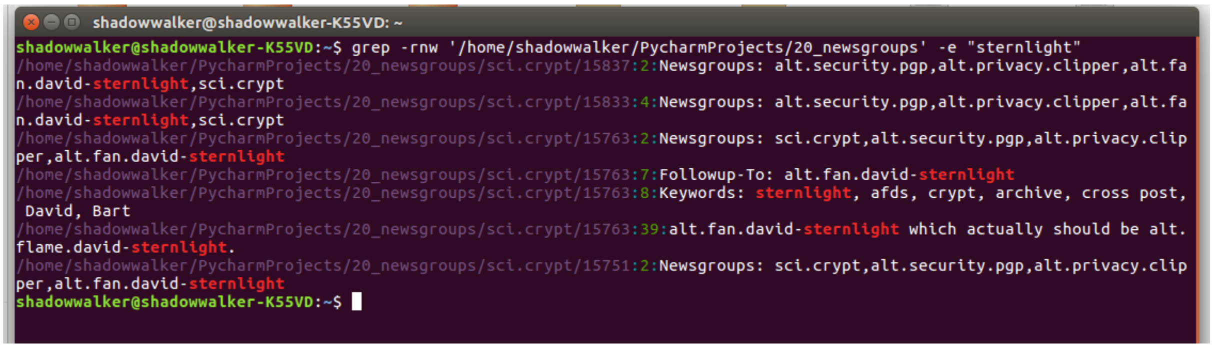
Some sample concepts obtained after LSA :
concept A
window 0.34157161196
file 0.188578790703
drive 0.177545849385
card 0.167210582507
dos 0.125476569751
driver 0.124476925917
program 0.121415497684
disk 0.118162468986
run 0.109035462494
softwar 0.0936654934395
scsi 0.0926694179264
pc 0.088907064117
mac 0.0873091416185
video 0.0775334339872
version 0.0772197854119
monitor 0.0766468380539
graphic 0.075073234221
color 0.0675863515945
sale 0.0674515153016
memori 0.0656276898966

concept B
game 0.376273311389
team 0.208473767716
player 0.143042929899
play 0.133235904621
hockey 0.110306611642
fan 0.102243491124
basebal 0.0974595301463
win 0.0965728788413
car 0.088490722566
espn 0.0862168939017
score 0.0815933742737
season 0.0784224537856
playoff 0.0705838372022
pitch 0.0692905601886
hit 0.0628684454784
nhl 0.0616729660241
leagu 0.060546834998

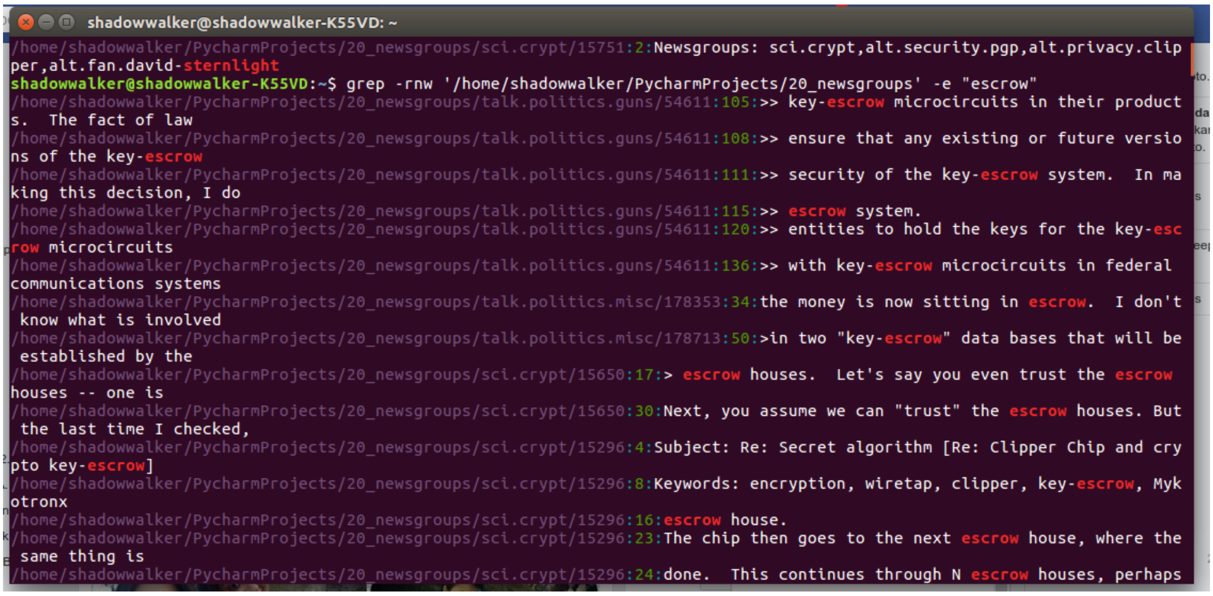
watch 0.0582026864992
toronto 0.0517079482021
leaf 0.051675110085

concept C
key 0.344255969473
chip 0.239851608704
encrypt 0.238839266148
clipper 0.216186159876
govern 0.170210472613
secur 0.127555953467
escrow 0.11450022608
phone 0.0959667869027
algorithm 0.0935421944315
law 0.080299709408
wiretap 0.0737021710668
nsa 0.0736291136529
public 0.0735072464978
gun 0.0694330710971
crypto 0.067163721029
des 0.0653044943662
secret 0.0632076838023
agenc 0.0608116379111
sternlight 0.0567156421054
enforc 0.0560589655335

We can see sternlight as ambiguity here since it means white light on a ship. But from below it can be verified that sternlight has been discussed a lot in documents of crypt, privacy on which the concept is based.



Similarly for escrow which means a bond, deed, or other document kept in the custody of a third party and taking effect only when a specified condition has been fulfilled, appears to be ambiguity. But from search we can find that it has been highly talked about in crypt topic.



Standard result obtained previously by various researchers on given dataset [9] for classification -

Paper	Model	Micro-ave accuracy	Notes
Lan, M, Tan, Chew-Lim, and Low, Hwee-Boon, 2006, Proposing a New Term Weighting Scheme for Text Categorization	SVM	0.808	
Larochelle, H and Bengio, Y, 2008, Classification using Discriminative Restricted Boltzmann Machines	hybrid discriminative RBM	0.762	Only 5000 most frequent tokens used as features
Li, B and Vogel, C, 2010, Improving Multiclass Text Classification with Error-Correcting Output Coding and Sub-class Partitions	ECOC Naive Bayes	0.818	
Rennie, Jason D M, 2003, On The Value of Leave-One-Out Cross-Validation Bounds	regularized least squares classifier	0.8486	Optimal regularization chosen post-hoc on test set

Figure 9.3: obtained from nlp.stanford

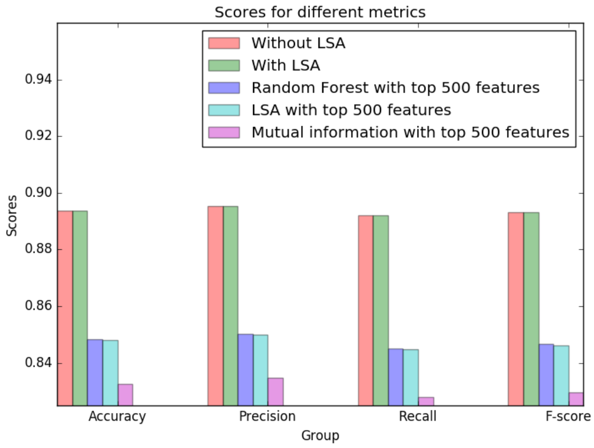
Results of classification after applying SVM with linear kernel
Total documents = 18846
Total features = 7837

	Accuracy	Precision	Recall	F-score
Without LSA	0.893633	0.895375	0.892079	0.893116
With LSA	0.893633	0.895375	0.892079	0.893116
Random Forest(top 500 features)	0.848275	0.850358	0.845015	0.846602
LSA (top 500 features)	0.848143	0.850043	0.844858	0.846287
Mutual information(top 500 features)	0.832493	0.83491	0.827857	0.829578

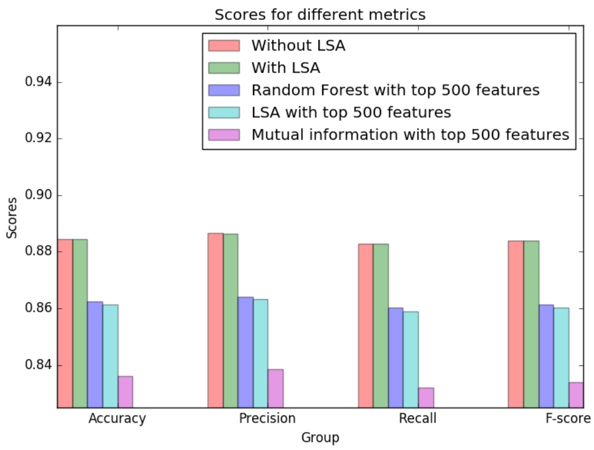
Total documents = 18846
Total features = 5916

	Accuracy	Precision	Recall	F-score
Without LSA	0.884482	0.88652	0.882875	0.884061
With LSA	0.8843501	0.88637	0.882745	0.883916
Random Forest(top 500 features)	0.862466	0.86407	0.860197	0.861419
LSA (top 500 features)	0.861273	0.863295	0.85893	0.860378
Mutual information(top 500 features)	0.836206	0.838534	0.832077	0.833902

The barplots of above classification metrics result :
Total documents = 18846
Total features = 7837



Total documents = 18846
Total features = 5916



Chapter 10

Conclusion

Thus from above observations we can conclude that it is not optimal to select top k features directly after dimensionality reduction using Singular Value Decomposition for classification or related task. Rather we can select top k features using random forest feature selection technique and can then use those reduced features.

Chapter 11

Comments and Suggestions

References

[1] tf-df vectorization
http://www.siam.org/meetings/sdm06/workproceed/Text%20Mining/antonellis21.pdf

[2] Vector Space Models of Semantics
https://www.jair.org/media/2934/live-2934-4846-jair.pdf

[3] Latent Semantic Analysis
http://lsa.colorado.edu/papers/dp1.LSAintro.pdf

[4] Latent Semantic Analysis
http://www.datascienceassn.org/sites/default/files/users/user1/lsa_presentation_final.pdf

[5] Representation in reduced space
http://www.rug.nl/research/portal/files/14692271/03c3.pdf

[6] Classification and regression using Random Forest
http://www.bios.unc.edu/~dzeng/BIOS740/randomforest.pdf

[7] Decison trees in Random forest
http://www.ijarcce.com/upload/2015/january/IJARCCE3L.pdf

[8] About the dataset
http://scikit-learn.org/stable/datasets/twenty_newsgroups.html

[9] Standard Classification Result
http://nlp.stanford.edu/wiki/Software/Classifier/20_Newsgroups

[10] Mutual information
http://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

[11] Mutual information scikit learn
http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html#sklearn.feature_selection.mutual_info_classif

[12] Precision, Recall, Fscore, Accuracy
http://xrce.fr/content/download/16594/118473/file/xrce_eval.pdf

[13] Scikit's classification metrics
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

[14] Snowball stemmer
http://www.nltk.org/api/nltk.stem.html

[15] scikit's tf-idf vectorizer
http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html