CSIC5011 Mini Project 1

A Dive into NIPS Words

GU Hanlin
hguaf@connect.ust.hk

HUANG Yifei
yhuangcc@connect.ust.hk

SUN Jiaze
jsunau@connect.ust.hk

Abstract

This paper aims to accomplish two tasks: we will first attempt to prune a high-dimensional term-document matrix by applying geometric reduction methods, before proceeding to explore the intrinsic relations and patterns in the low-dimensional data. For the first task, we selected 8 methods for analysis, which are Principle Component Analysis (PCA), Multi-Dimensional Scaling (MDS), ISOMAP, Locally Linear Embedding (LLE), Modified LLE, Hessian LLE, Spectral Embedding (also known as Laplacian Eigen Map), and Local Tangent Space Alignment (LTSA) [3]. In addition, we also seek to improve the previous results by performing *TFIDF* transform and Latent Dirichlet Analysis (LDA) to the data respectively before applying the 8 reduction methods. As for the second task, we will attempt to make predictions, based on results obtained from the first task, on things such as future topic trends, the relations between different topics, or even potential emergence of new interdisciplinary subjects. In particular, we will attempt to fit an ARIMA model based on the data obtained from LDA. Furthermore, we will explore simple topic relations by performing multivariate statistical analysis, providing visualization in the process.

1 Introduction

Using languages is perhaps one of humans' most extraordinary abilities. In recent years, machines have also begun to demonstrate remarkable potential in 'comprehending' and utilizing human's own natural languages, simply by mimicking the way in which a person acquires a new language, that is, making associations between words. In doing so, we create a sense of 'distance' between words, where words that have similar meanings are 'close', and drastically different words are 'far'. By representing words as vectors, or more commonly referred to as embeddings, we can mimic this abstract 'distance' between words, thus establishing the basis of letting machines understand word relations and even categorize documents.

The term-document[1] matrix that we used comes from the NIPS Word data set [2]. After removing 7 zero-columns, the matrix, denoted by X, is of size 11463×5804, where row i represents the i-th word, column j represents the j-th document, and the entry X_{ij} is the number of times word i appears in the j-th document. The vocabulary consists of words appearing in all NIPS conference papers published between 1987 and 2015, and is truncated by removing stopwords and words whose total number of appearances in all papers are no more than 50. Approximately 90% of the entries in X are zero, making it fairly sparse. In addition, the data set also includes the publish year for every paper. Figure 1 visualizes the vocabulary based on their appearances throughout all the years, with the help of a Python package called *wordcloud*.

[1] 'Word' and 'term' are used interchangeably throughout this paper, and the same also applies to 'paper' and 'document'.

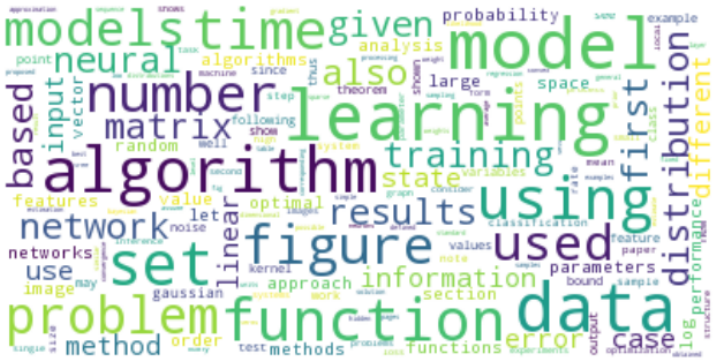

Figure 1: The word-cloud is generated based on the occurrences of words in all NIPS conference papers published between 1987 and 2015.

In this paper, we aim to address the problem of classifying such a data set, especially when applying reduction methods directly to the original data set might be unsatisfactory. Moreover, we combine various statistical methods to analyze the intrinsic relations and patterns in the data.

2 Basic Data Reduction and Visualization

2.1 On the Original Data Set

In this subsection, we apply the aforementioned 8 data reduction methods directly to the original data matrix. For simplicity of graphing, we only select 2 component for visualization. For each of the 8 methods, we applied it to both X and X^T to obtain embeddings of size 11463×2 and 5804×2 respectively, where the former correspond to the 11463 words, and the latter correspond to the 5804 documents. The plot of the embeddings are shown in Figure 2.

The results of PCA and MDS show that most data points are centered around one point, with the density decreasing steadily as one moves away from that point. ISOMAP can also roughly capture the same general pattern as seen in PCA and MDS, where the visible gaps and holes might be the result of distortion caused by actual holes in the data. Other manifold learning methods, however, do not seem to capture the pattern very well. One possible reason for this is that these methods assume the data lie in a low-dimensional manifold (in this case a 2-dimensional surface), while it is very likely that this might not be the case. In addition, extraneous tests shows that the variation captured by PCA is somewhat related to frequency of occurrence. For example, words that appear more frequently in all papers tend to be clustered on one side of the graph, while rare words tend to be on the other side. Similar phenomena were also observed for documents, but based on the number of words they have instead. However, this observation is purely empirical, and the 'components' produced by PCA or MDS in this case are difficult to interpret, so we shall end our analysis here.

2.2 After TFIDF transformation

The motivation for the TFIDF transformation is that the intrinsic pattern might not be truthfully reflected by the original data matrix. For example, the word 'abstract' and 'references' appear in almost every document, thus making little to no contribution to helping distinguish the documents from each other. In essence, the TFIDF transform assigns to each entry of X a weight that is inversely proportional to the number of documents in which they appear [1]. Figure 3 shows the results of the 8 reduction methods after TFIDF has been applied.

From the results, one can observe that applying TFIDF produces moderate improvements: PCA, ISOMAP and LTSA are able to capture the variation in a slightly more detailed manner. However, the LLE related algorithms are still unable to produce any discernible pattern. While it is possible that the data points are still not confined to a low-dimensional manifold even after the TFIDF transformation, another plausible explanation is the non-uniformity of the data. Methods like PCA and MDS take into account the global picture, while the manifold learning methods primarily focuses on the local pattern. Based on the observation from the former three methods, the density variation across the

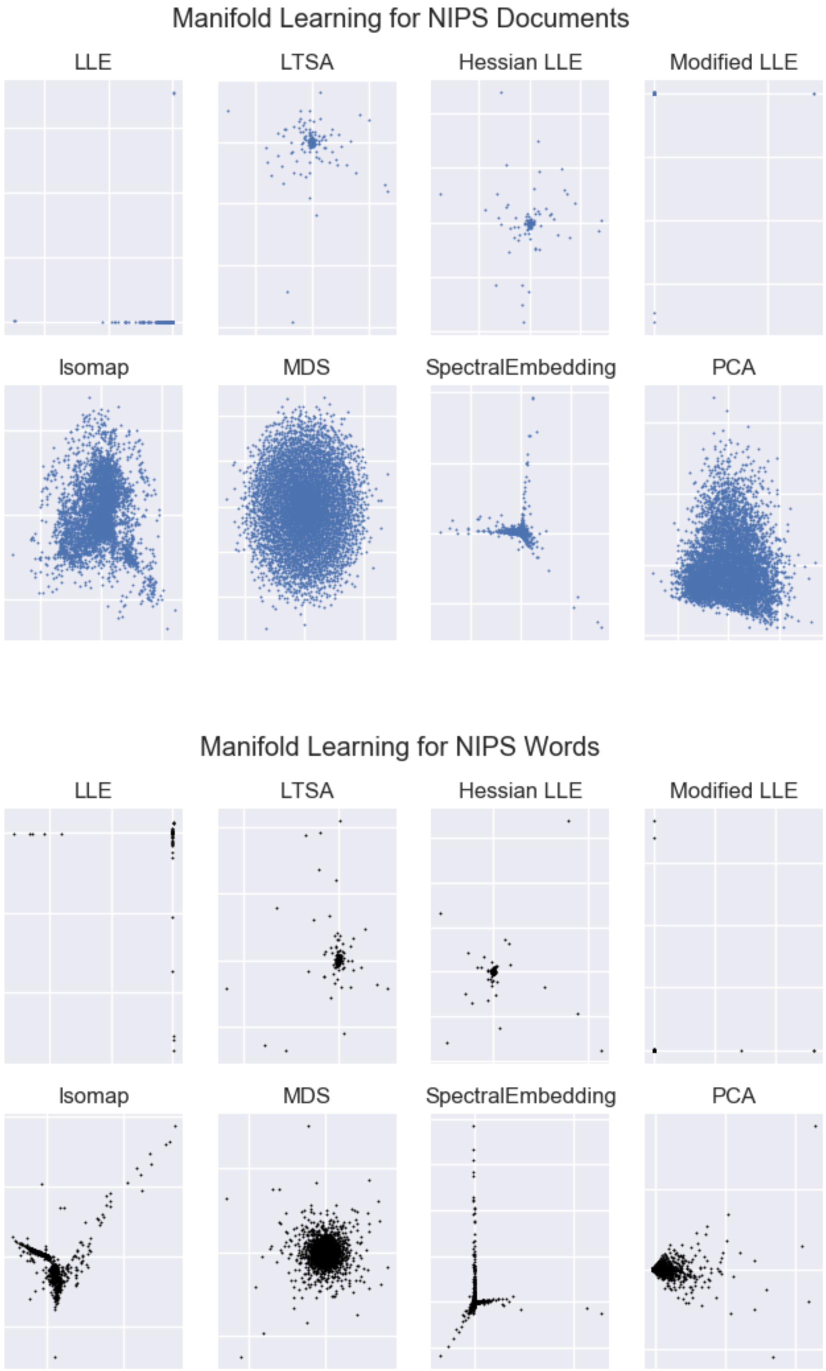

Figure 2: Visualization of the embeddings of documents (upper graph) and words (lower graph) obtained from 8 different reduction methods.

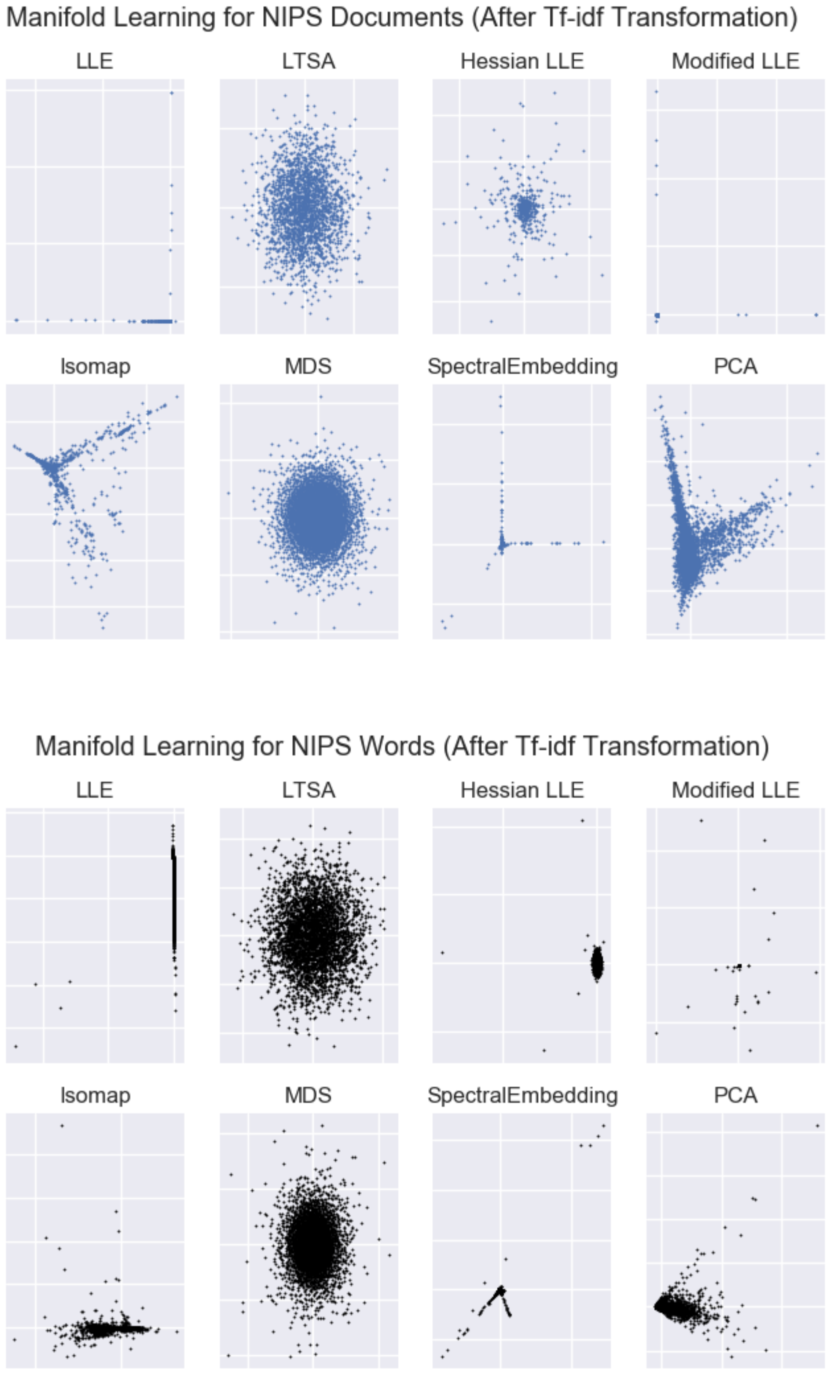

Figure 3: Visualization of the embeddings of documents (upper graph) and words (lower graph) obtained from 8 different reduction methods, after applying the TFIDF transformation.

board is huge, so it is unsurprising that applying LLE-related methods might lead to great distortion. To remedy this, a possible solution is to increase the number of neighboring points that are used in the calculation, but due to the high dimensionality of the data set, this would become too computationally costly for us to finish within reasonable time.

3 Latent Dirichlet Allocation (LDA)

The previous section shows that PCA and other manifold learning methods alone are not sufficient to analyze this type of data, because they do not take the intrinsic properties of the documents and the words into consideration. LDA, on the other hand, does take such matters into account. Essentially, LDA assumes that each document is a multinomial distribution over a set of topics; in turn, each topic is a multinomial distribution over words. Given a $t \times d$ term-document matrix X, LDA generates two matrices, *WORD_TOP* and *TOP_DOC*, each of size $t \times k$ and $k \times d$, where k represents the number of topics. Since each topic is a multinomial distribution over words, each column in *WORD_TOP* sums up to 1, and so do the columns of *TOP_DOC*. This way, we can regard these matrices as embeddings of documents and words in a k-dimensional 'topic' vector space, where we can then apply PCA or other manifold learning methods for further analysis.

A crucial step in applying LDA is to choose a suitable k. An excessively large k will cause the topics to be dependent on each other, while too small a value will not assign documents to their proper topics. In this section, we set $k = 5$, which is appropriate since almost all NIPS papers have an emphasis on machine learning, a relatively specialized discipline. In the next section we will discuss whether this choice is appropriate for this data. Thus, the *WORD_TOP* and *TOP_DOC* matrices are respectively of size 11463×5 and 5×5804.

3.1 Topics

For each column in *WORD_TOP*, we sort the words according to their probability in descending order, and obtain the 10 most common words for each topic, shown in Table 1.

Table 1: top 10 words of 5 topics

		Topics		
topic0	topic1	topic2	topic3	topic4
algorithm	learning	model	network	image
function	algorithm	data	neural	model
matrix	state	training	input	images
data	time	learning	time	figure
problem	policy	models	networks	object
learning	function	set	neurons	visual
set	value	using	learning	using
error	set	number	model	data
linear	problem	used	output	different
using	action	features	units	spatial

From the table we can roughly see that the 5 topics are somewhat dependent except for topic2, topic3 and topic4. And from each topic's words, we can roughly say what field each topic represents. For topic0, we can see that the word 'algorithm' is the most frequency word followed by 'function' and 'matrix', so we might say this topic is about algorithm and mainly focuses on the matrix. For topic1, with the same explanation we might say it is about algorithm but mainly focuses on the learning of the algorithm. For topic2, different from above two topics, the word 'model' is the most frequency word followed by 'data' and 'training', so we might say this topic is mainly concentrated on the model itself. For topic3, this together topic 4 are totally different from the other three topics, we can clearly see that the words in topic3 are about Neural Network, because the word 'neural' and 'network' occur in this topic in a very frequency rank. Also for topic4, which contains many words in Image Vision such as 'image', 'figure' and 'visual'.

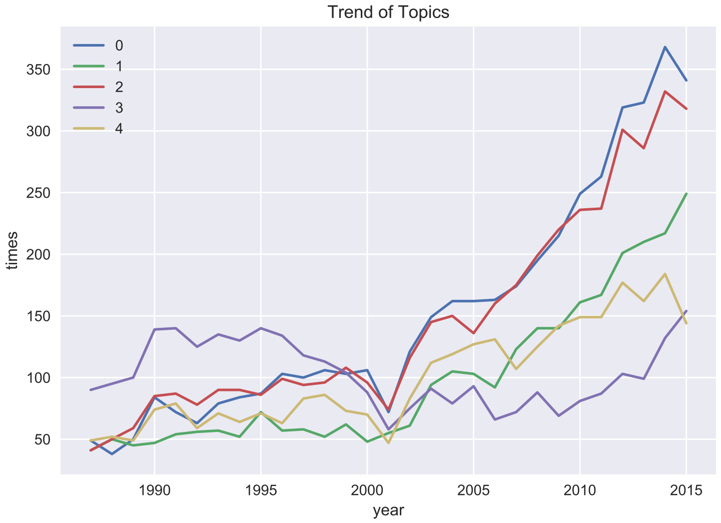

Figure 4: Topic trends by year.

Using the *TOP_DOC* matrix, if we count the top 3 topics for each document, then we can get the topic trends over the 29 years, as shown in Figure 4. From this figure, we can see that topic0 and topic2 have similar trend and both have increased in the recent years. The other 3 topics, albeit not as fast, also show signs of increasing in recent years, suggesting that they are becoming more popular, especially topic3, which is on Neural Network. Overall, the trends for all topics somewhat decreased form 1987 to 2001, after 2001 they all increased dramatically. The year 2001 is peculiar, since all topics decreased at the same time. As an interesting side note, the location of the NIPS Conference changed from America to Canada in 2001, which might or might not be related to this phenomenon.

3.2 Manifold learning for LDA

As mentioned before, we can also perform visualization by applying manifold method to *WORD_TOP* and *TOP_DOC* respectively, as shown in Figure 5. Now that the data is embedded in a low dimensional space, it is probable that manifold learning might become more relevant. For documents, comparing with the previous reduction results, Figure 2 on raw data, and Figure 3 by the *TFIDF* method, we can easily see that except for a few methods, almost all other reduction methods here outperform the previous ones. In particular, MDS and PCA produced strikingly regular patterns with faces and edges. Empirically, we can conjecture that the NIPS documents might indeed consist of several topics, where points on the edges can be regarded as documents focusing on a single subject, while points on the faces are of a more interdisciplinary nature. On the other hand, the results for words are roughly the same as the previous figures except for Spectral Embedding. This suggests that the structure of words is difficult to capture, as LLE and PCA seem to show that it is more of a radial-shape, with branches stemming out from a common center, rather than being a smooth, low-dimensional hyper-surface.

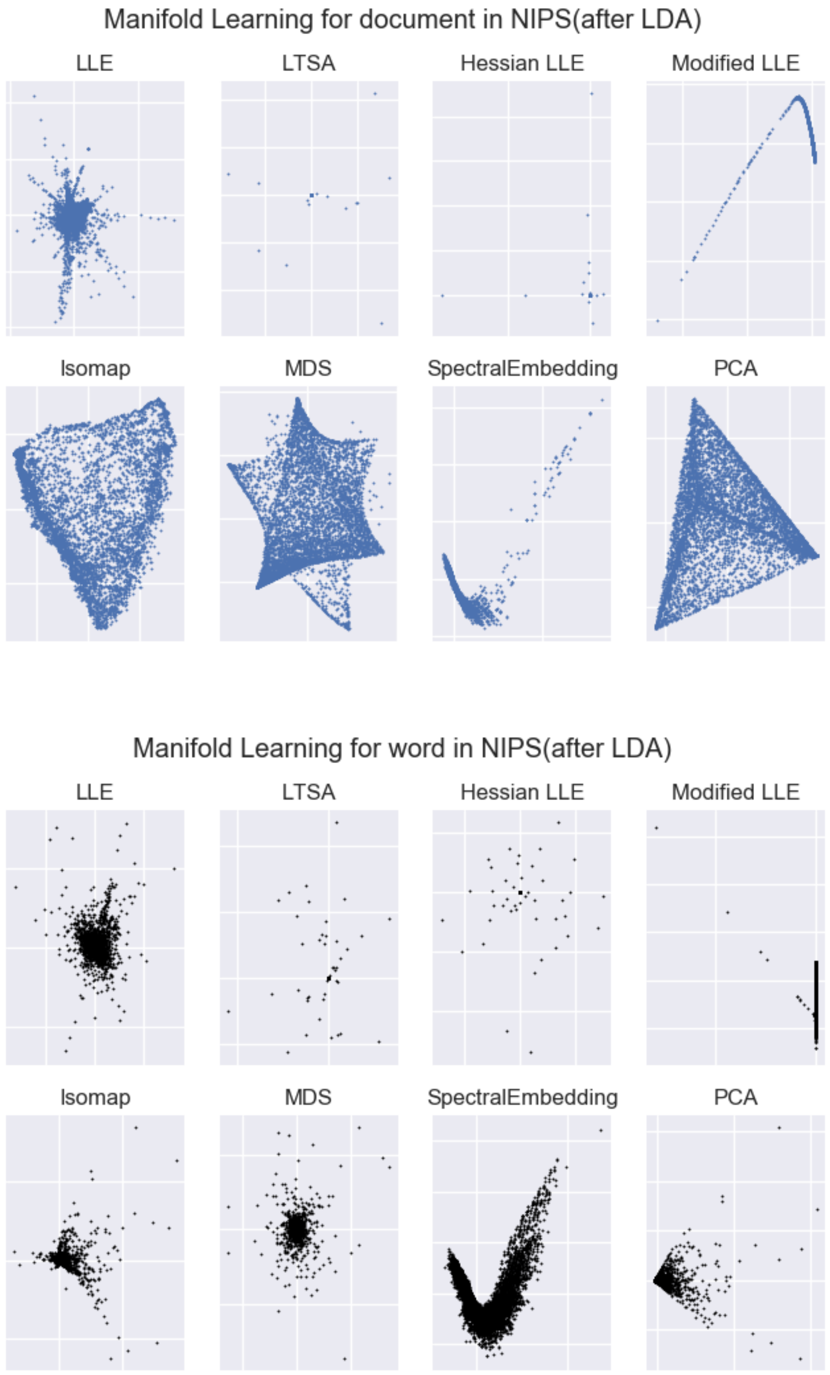

Figure 5: Visualization of the embeddings of documents (upper graph) and words (lower graph) obtained by applying the 8 different reduction methods to *TOP_DOC* and *WORD_TOP* respectively.

4 Regression and Prediction for Topics

From the discussion above, we have now acquired several topics based on the result of LDA. To do time series analysis on *TOP_DOC*, we individually sum up the columns of the same year and obtain a 5×29 matrix, corresponding to 5 topics over 29 years.

4.1 methods

In order to select a suitable model, stationarity of data need to be considered. After taking forward difference and drawing autocorrelation function (ACF) graph, we conclude there is no obvious difference of stationarity; ACF also decreases fast. Therefore, we select the ARIMA model for forecasting. As for the relations between topics, we fit a regression model for each one of the 5 topics over all the others. In addition, removal of multicollinearity and departure of data residual (such as homogeneity) is also required. For this, we process the data by performing Box-Cox transformation, and then we fit stepwise regression to the data.

4.2 forecast

First, based on the sequence diagram of topics, we observed that the time series was not stationary. We take the forward difference of the time series, and test for stationarity using the unit root test (ADF test). Then, we plot the ACF and PACF in order to determine the parameter p and q as in the ARIMA model, where p is the level truncation of ACF, and q is the level trailing of PACF. With the above results, the fitted ARIMA model is shown in Figure 6. The Ljung–Box test shows that the residue of the forecast is white noise, suggesting that the model is valid.

Using this model, we also attempted to forecast the topic trend over the succeeding 3 years, shown in Figure 6. Unsurprisingly, topic 0, 1, and 2, which respectively have keywords algorithm, model, and data, will increase rapidly. The model also shows that topic 3, which is on neural network, will start to gain traction in the future, albeit having been unpopular for a relatively long period. However, topic 4, which is on image, is projected to gradually lose popularity in the future.

4.3 regression

For regression, as we mentioned previously, we use one topic as the dependent variable Y and the other 4 topics as the independent variables. From the correlation matrix of data, we find topic 0, topic 2 and topic 3 are closely related. In order to evaluate the validity of the model, we should examine the following two conditions: residuals of data follow normal distribution; the multicollinearity of data is small. Taking stepwise regression and Box-Cox transformation (shown in Figure 7, we can see that the two conditions are satisfied. The fitted parameters are shown in Table 2.

By comparing the coefficients, we can see that topic 1 ,2 has close relation with topic 0; topic 4 also has great influence over topic 3; topic 4 have some relations with topic 1, 2, and 3. In reality, topic 0, 1, and 2 share similarities, as they are all based on data using algorithms or learning model; topic 4, which is image, is the one of the most important applications of neural network; topic 4 has cross section with topics on learning, model and neural network.

Table 2: linear relation among 5 topics

			Topics			
	topic0	topic1	topic2	topic3	topic4	intercept
topic0		1.3308	0.9638			-13.288
topic1	0.34390					4.23991
topic2	0.50526				0.37097	2.40942
topic3	-0.17021				-0.90152	70.14459
topic4		-0.32446	0.28006	-0.18978		28.13245

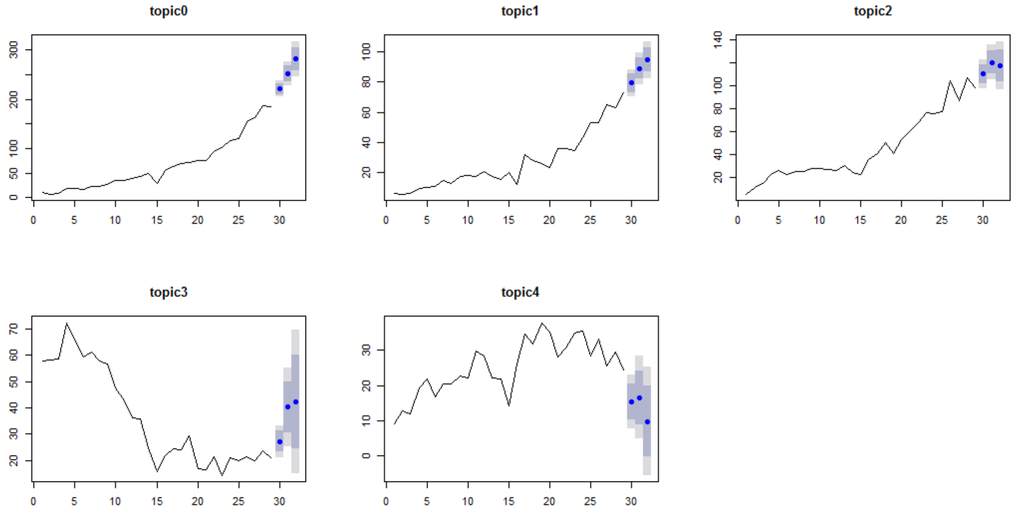

Figure 6: forecast in next three years in topics

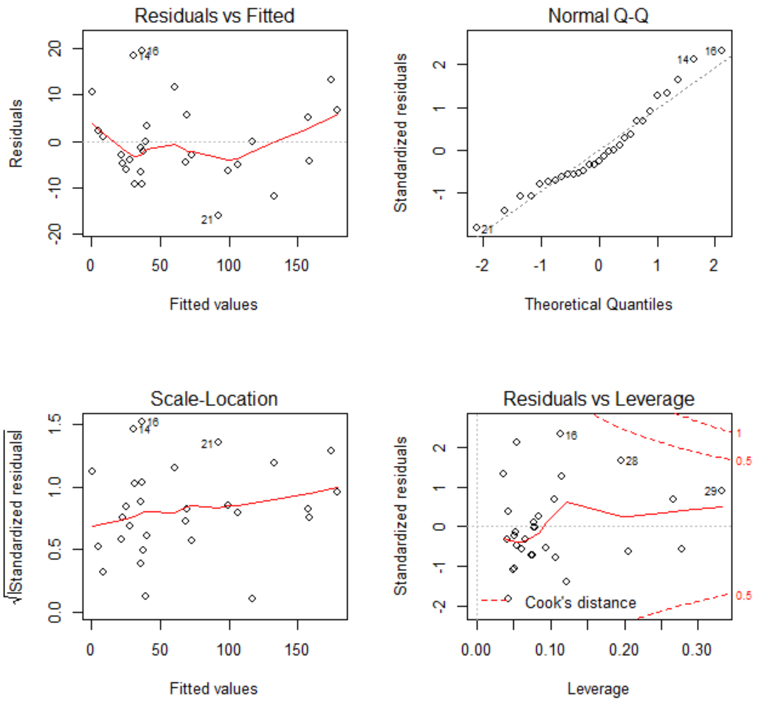

Figure 7: residue and qqplot

5 Individual Contribution

GU Hanlin focuses primarily on Section 4, which is on time series analysis and multivariate regression. He performed the said analyses via R.

Huang Yifei worked on Section 3, which is on LDA and its subsequent analyses. He performed said tasks using Python.

Sun Jiaze worked on Section 2, which is on basic geometric reduction and *TFIDF* transform. He performed these tasks via Python.

Everyone contributed equally to this project.

References

[1] D. JURAFSKY AND J. H. MARTIN, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (third edition draft)*. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf, 2017.

[2] V. PERRONE, P. A. JENKINS, D. SPANO, AND Y. W. TEH, *Poisson random fields for dynamic feature models*. https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015, 2016.

[3] Y. YAO, *A mathematical introduction to data science*. http://math.stanford.edu/~yuany/course/reference/book_datasci.pdf, 2017.