****

**Topic Modeling for Economic News Articles**

**Springboard Data Science Career Track Capstone Project II**

**Rueben Yang**

**December 2017**

1. **Introduction**

This project deals with the problem of building a machine learning model for news article classification – clustering the articles into a number of groups based on the underlying topic of each article. Therefore, the clients this project serves are the new agencies. The dataset used in this project is obtained from the *Data For Everyone Library*[[1]](#footnote-1), which contains 8000 news articles related to economics ranging from 1951 to 2014. The sources of the articles are Wall Street Journal and Washington Post. Topic modeling is used to classify the economic articles into different categories and determine the membership of them by discovering the structure in the corpus.

1. **Data Preprocessing**

Preprocessing the text data is the essential step in building a good topic model. The following steps are performed:

1. Remove numbers and special characters; all the remaining words are transformed to lowercase.
2. Remove superfluous content – at the beginning of some articles, there is author information (e.g. “*Author: Author Name*”), news source (e.g. “The Wall Street Journal Online”), or location information of the news (e.g. New York or Washington, which is often reiterated in the news content). The content as such is removed.
3. Lemmatize the tokens to group the inflected forms of a word into a single term. A lemmatizer is used in this case other than a stemmer since the words after lemmatization still remain readable.
4. In order to apply unigram in topic modeling, only the nouns are kept as they are more useful in revealing the subject of the articles compared with the other forms of words, for instance, verbs and adjectives.
5. The set of English stop words from *nltk.corpus* are filtered out from the list of tokens. After step (4), most of the stops were already removed, and this step ensures that the tokens contain no stop words.
6. To further refine the quality of the remaining tokens, the 50 most frequent words in the collection of these 8000 articles are reviewed:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Word | Occurrence | No. | Word | Occurrence | No. | Word | Occurrence |
| 1 | year | 10189 | 2 | market | 8498 | 3 | rate | 8151 |
| 4 | stock | 7669 | 5 | price | 5592 | 6 | economy | 4097 |
| 7 | interest | 3985 | 8 | bank | 3978 | 9 | month | 3794 |
| 10 | company | 3762 | 11 | week | 3526 | 12 | percent | 3488 |
| 13 | inflation | 3256 | 14 | investor | 3048 | 15 | time | 2988 |
| 16 | point | 2952 | 17 | government | 2831 | 18 | dollar | 2818 |
| 19 | president | 2806 | 20 | tax | 2723 | 21 | bond | 2690 |
| 22 | yesterday | 2621 | 23 | growth | 2581 | 24 | day | 2567 |
| 25 | reserve | 2463 | 26 | share | 2454 | 27 | fund | 2439 |
| 28 | state | 2399 | 29 | index | 2344 | 30 | business | 2326 |
| 31 | york | 2296 | 32 | quarter | 2194 | 33 | money | 2148 |
| 34 | increase | 2106 | 35 | job | 1984 | 36 | consumer | 1929 |
| 37 | fed | 1904 | 38 | report | 1894 | 39 | deficit | 1869 |
| 40 | policy | 1855 | 41 | sale | 1845 | 42 | economist | 1805 |
| 43 | trading | 1797 | 44 | analyst | 1768 | 45 | budget | 1751 |
| 46 | term | 1750 | 47 | today | 1713 | 48 | people | 1699 |
| 49 | security | 1646 | 50 | dow | 1639 |  |  |  |

**Table 1: Word Count for the Top 50 Words**

The following list of words from the 50 most popular words are excluded as they are considered superfluous:

*['day', 'today', 'yesterday', 'week', 'month', 'quarter', 'time', 'percent', 'rate', 'point', 'economy', 'economist', 'growth', 'increase', 'york', 'report', 'analyst', 'term', 'people']*

1. Eventually, the rare words that appear in less than 5 articles and the frequent words that appear in more than 50% of the articles are filtered out. On top of step (6), this step ensures the common words based document frequency are removed.

As the result of the above steps, there are 5293 unique tokens in the final dictionary created.

1. **Methodology**

David M. Blei, Andrew Y. Ng, and Michael I. Jordan proposed latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora[[2]](#footnote-2). LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics.

A high-level illustration is provided in this section. Suppose there are the following set of sentences:

1. NLF playoff picture: How the field looks right now
2. Projected 2018 NFL draft order: Browns take two-game lead in race for No. 1 pick
3. FIFA World Cup: Which teams do you think will make it to the round of 16?
4. The Senate tax bill would allow oil drilling in Alaskan wilderness
5. Trump slams DOJ and FBI in weekend tweetstorm

LDA is a way of discovering topics that these sentences contain. For the given sentences, if we ask for 2 topics, LDA might produce the following outcomes:

* **Sentences 1, 2, and 3**: 100% Topic A
* **Sentences 4 and 5**: 100% Topic B
* **Topic A** corresponds to words: NFL, playoff, draft, FIFA, World, Cup (one could interpret topic A to be sports)
* **Topic B** corresponds to words: Senate, Trump, DOJ, FBI (one could interpret topic B to be politics)

It’s worth noting that one has to figure out what the topics refer to as LDA only provides lists of words that covered by topics but does not automatically label the topics. To remedy the problem that topic models give no guaranty on the interpretability of their output, measuring coherence of topics has been recently studied by researchers. Michael Röde, Andreas Both, and Alexander Hinneburg has provided a comprehensive discussion on topic coherence[[3]](#footnote-3). In addition, Palmetto is a tool for measuring the quality of topics[[4]](#footnote-4), which provides the high-level descriptions for the popular coherence measures. In this project, the *Cv* and *Umass* are used to evaluate the topic coherence, and their high-level introductions are:

* *Cv* is based on a sliding window, a one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity. This coherence measure retrieves co-occurrence counts for the given words using a sliding window of size 110. The counts are used to calculated the NPMI of every top word to every other top word, thus, resulting in a set of vectors – one for every top word. The one-set segmentation of the top words leads to the calculation of the similarity between every top word vector and the sum of all top word vectors.
* *UMass* is based on document co-occurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure. The main idea of this coherence is that the occurrence of every top word should be supported by every top preceding top word. Thus, the probability of a top word to occur should be higher if a document already contains a higher order top word of the same topic. Therefore, for every word, the logarithm of its conditional probability is calculated using every other top word that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts.

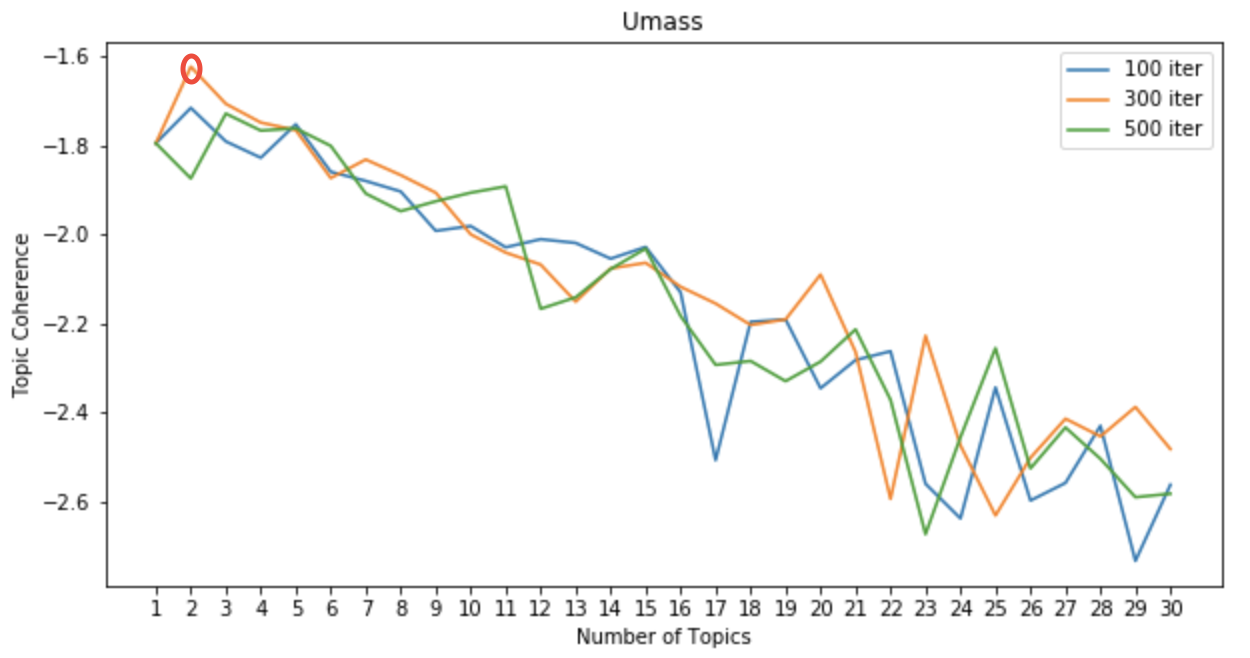
1. **Results**

This project applies *genism[[5]](#footnote-5)*, an open-source topic modeling toolkit implemented in Python.

Two parameters in *gensim*’s *LdaModel* are tuned – *num\_topics* (the number of requested latent topics to be extracted from the corpus) and *iterations* (the number of steps the variational inference is allowed without convergence). The candidates for *iterations* are 100, 300, and 500; and any integer in the closed interval [1, 30] for *num\_topics.* The entire dataset is used for training to produce classifications for all 8000 articles.

* 1. **Umass**

The *Umass* scores are plotted in **Figure 1** below.

**Figure 1: Topic Coherence based on Umass**

The maximum *Umass* is -1.623, which is obtained at *iteration* = 300 and *num\_topic* = 2. Hence, for the topic model based on *Umass*, the optimal parameters are 300 iterations and 2 topics. The top 10 words for each topic are shown in **Table 2**.

|  |  |
| --- | --- |
| Topic 1 | Topic 2 |
| market  stock  price  interest  investor  company  bond  share  dollar  index | tax  president  state  government  budget  job  house  deficit  bank  administration |

**Table 2: Top Words based on the Umass\_model**

There are no overlaps between the two sets top words. One possible interpretation would be that topic 1 is related to the financial market and topic 2 is related to macro economy.

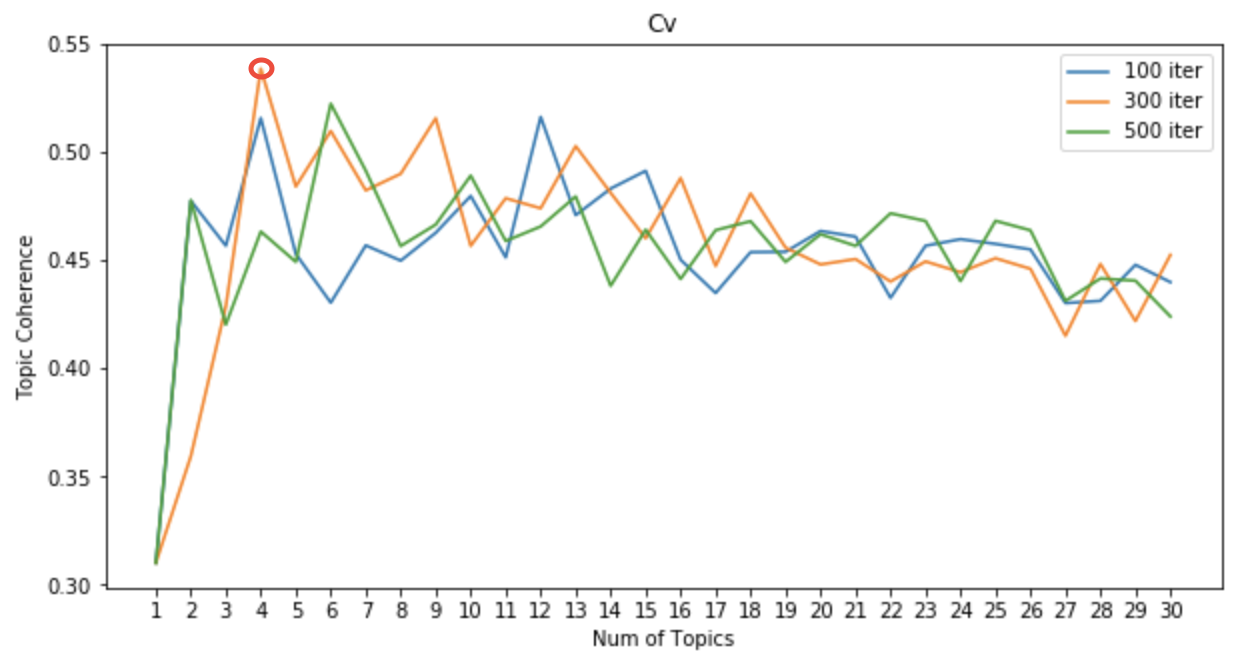
For each given document, the topic distribution as a list of tuples (*topic\_id, topic\_probability*) is provided by *genism*. Each article is labeled as the topic that yields the maximum topic probability for the article and the number of articles for each topic is shown in **Table 3**.

|  |  |  |
| --- | --- | --- |
| **Topic ID** | 1 | 2 |
| **Count** | 4363 | 3637 |

**Table 3: Classification of Articles based on the Umass\_model**

* 1. **Cv**

The *Cv* scores are plotted in **Figure 2**.



**Figure 2: Topic Coherence based on the Cv\_model**

The maximum *Cv* is 0.538, which is obtained at *iteration* = 300 and *num\_topic* = 4. Hence, for the topic model based on *Cv*, the optimal parameters are 300 iterations and 4 topics. The top 10 words for each topic are shown in **Table 4**.

|  |  |  |  |
| --- | --- | --- | --- |
| Topic 1 | Topic 2 | Topic 3 | Topic 4 |
| tax  state  president  job  budget  government  deficit  unemployment  house  administration | stock  market  price  index  investor  dollar  trading  dow  bond  share | company  sale  business  share  industry  firm  home  corp  service  inc | bank  interest  reserve  fed  fund  loan  market  money  treasury  mortgage |

**Table 4: Top Words based on the Cv\_model**

Topic 1 is quite similar to topic 2 from the*Umass\_*model, which is related to macro economy; topic 2 seems to be related to the financial market; for topic 3, one might consider it as corporate news; topic 4 could be related to the housing market and mortgage.

**Table 5** shows the number of articles for each topic.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topic ID** | 1 | 2 | 3 | 4 |
| **Count** | 2573 | 2689 | 1358 | 1380 |

**Table 5: Classification of Articles based on Cv\_model**

* 1. **Evaluation**

Given that topic modeling is unsupervised learning, the model evaluation becomes tricky as there are no expected labels that one can compare the predicted labels to.

In this project, a somewhat manual approach is used to evaluate the quality of the topic models:

* Randomly select one article from the dataset
* Read the picked article and list the 10 keywords from my own perspective
* Find the common words that appear both in my list of 10 keywords and the top 10 words of each topic obtained from the two models
* Tag the chosen article with the topic that has the most overlapping words with my list of keywords
* Compare the classifications of the picked article produced by the two models with the label assigned by my own perspective

The article with ID 2587 was selected from the dataset. The content of the article is shown as follows.

**Article #2587:**

A widely expected Federal Reserve increase in short-term rates sent bond prices falling and knocked stock prices down from their highs to mixed levels. The dollar was mixed. The bond market, practically dormant through much of the morning, began churning once the Fed said at midafternoon that it had raised interest rates. But as economists read through the Federal Open Market Committee statement on the move, bond prices began to slump, especially among shorter-term issues. Some economists noted that not only did the Fed increase the federal funds rate by 0.50%, but also the discount rate by the same margin. The fed funds rate is the bank overnight lending rate and the discount rate is the rate the Fed charges from its discount window. In addition to the rate boosts, the Fed termed signs of slowing as "tentative" and hinted that the economy remains quite vigorous, even as it raised rates for the seventh time in about one year. The language that accompanied the Fed's move, especially about inflation and capacity utilization, along with the unanimous vote, telegraphed a certain sense of concern to the bond market," said Frazier Evans, senior economist for Colonial Investment Services. In the stock market, a mild rally in cyclical issues rapidly escalated into a fierce argument between the two warring constituencies seeking control of the stock market. During the past two sessions, more than 800 million shares have traded hands as investors have wrestled with the recession question.

After reading the article, I came up with my list of the 10 keywords and it is shown in the leftmost column of **Table 6**. For the *Cv*-based model (300 iteration and 4 topics), the second topic has the most overlapping words (7 words) with my list; the other topics (first, third, and fourth topic) share 0, 1, and 4 words with my list, respectively. For the *Umass*-based model (300 iteration and 2 topics), the first topic has the most overlapping words (8 words), and its second topic doesn’t share any overlap with my list. Therefore, based on my opinion, the selected article should be labeled as topic 2 for the *Cv*-based and topic 1 for the *Umass*-based. The common words are highlighted in **Table 6**.

|  |  |  |
| --- | --- | --- |
| **My keywords** | ***Cv***  **(Topic 2)** | ***Umass***  **(Topic 1*)*** |
| federal  reserve  bond  price  stock  market  dollar  interest  inflation  share | stock  market  price  index  investor  dollar  trading  dow  bond  share | market  stock  price  interest  investor  company  bond  share  dollar  index |

**Table 6: My Keywords and Top Words from the Most Similar Topics**

The topic probabilities for Article #2587 produced by the two models are as follows (the maximum probabilities are highlighted):

|  |  |  |  |
| --- | --- | --- | --- |
| *Umass*-model (300 iteration and 2 topics) | | *Cv*-model (300 iteration and 4 topics) | |
| Topic ID | Probability | Topic ID | Probability |
| 1 | 0.894 | 1 | 0.051 |
| 2 | 0.106 | 2 | 0.521 |
|  |  | 4 | 0.424 |

**Table 7: Topic Probabilities**

Therefore, the classifications of Article #2587 based on the two models are identical with my own interpretation. More specifically, the *Umass*-model classifies Article #2587 as topic 1 with assurance as the probability for topic 1 is remarkably above the other topic, whereas *Cv*-model places the article under its topic 2 with less confidence as the probabilities it yields for topic 2 and topic 4 are not significantly distant from each other.

1. **Conclusion**

Based on the experiment, both topics models built in this project provide article classification that’s consistent with human interpretability, and hence, the two topics models show acceptable performance. Since the numbers of the topics (2 and 4) from the two models are significantly smaller than the number of possible words, the dimensionality of the articles is reduced materially. The optimal number of topics for the *Umass*-model is less than that of the *Cv*-model, so the topics produced by *Umass*-model are fairly exclusive to each other, while the words in the topics produced by *Cv*-model have some overlaps which could potentially lead to reduce the uniqueness of these topics.

1. **Recommendations for Clients**

Since both topic models are effective in classifying the news articles based on their underlying topics, it is recommended that news publishers, e.g. Wall Street Journal and Washington Post, consider using the approach presented in this project (data preprocessing steps, and *LDA* with *Umass* and *Cv*) in developing the machine leaning models for article classifications.

1. **Future Work**
2. The evaluation process could be implemented more comprehensively by asking multiple people reading various articles in the dataset and comparing the keywords that are collaboratively determined with the words from the topics produced by the models. This will strengthen the robustness of the evaluation process as it aims to reduce the bias. In addition, another potential approach would be finding the article that’s closest to the selected article from the perspective of Euclidean distance in topic probability distributions and reading the two articles to qualitatively assess their similarities.
3. There are other measures of topic coherence that could be used to train and evaluate the topic models, e.g. *Cp*, *UCI*, *NPMI*, etc.
4. Use n-gram so that the word order and phrases would be able to help capture the hidden topics of the articles.
5. Consider applying other algorithms in topic modeling, e.g. Latent Semantic Indexing (LSI) and Hierarchical Dirichlet Process (HDP).

1. https://www.crowdflower.com/data-for-everyone/ [↑](#footnote-ref-1)
2. D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research* *3* (2003) 993-1022. [↑](#footnote-ref-2)
3. M. Röde, A. Both, and A. Hinneburg. Exploring the Space of Topic Coherence Measures. [↑](#footnote-ref-3)
4. http://palmetto.aksw.org/palmetto-webapp/ [↑](#footnote-ref-4)
5. https://radimrehurek.com/gensim/ [↑](#footnote-ref-5)