Documentation

Reproducing the Mining Causal Topics in Text Data: Iterative Topic Modeling with Time Series Feedback

Darius Nguyen
Pritom Saha Akash
Trisha Das

1) An overview of the function of the code :

Many applications need textual topics to be studied together with external time series. This paper proposes a general text mining system for the discovery of this type of causal themes from the text. We implemented the algorithm presented in the paper in Python. Our implementation combines a given probabilistic topic model with time-series causal analysis to discover topics that are both coherent semantically and correlated with time-series data. As described in the paper, we iteratively refine the discovered topics to increase the correlation with the time series. Time series data provides feedback at each iteration by imposing prior distributions on parameters.

In our experiment, we examine the 2000 U.S. Presidential election campaign. The input text data is from New York Times articles from May through October of 2000. As a non-textual time-series input, we use prices from the Iowa Electronic Markets 2000 Presidential Winner-Takes-All Market. We also experimented with stock time-series data (AAMRQ vs. AAPL). Our implementation can determine causal topics efficiently.

We used PLSA as our topic modeling method. Granger Test is used to find out a set of candidate causal topics with lags. Pearson Correlation is used to find out word-level causality in our implementation. Though the authors of the paper used R programming language to do the Granger Causality test, we did it in Python to make the code compact and manageable. We used only one programming language to complete the full implementation.

2) Documentation of how the software is implemented:

The main parts of our code are as follows:

1. **Data Preparation:** We used text data from the 2000 U.S. Presidential election campaign. The input text data is from the New York Times articles from May through October of 2000. We filter them for keywords “Bush” and “Gore,” and use paragraphs mentioning one or both words. Also, we

scrapped time-series data from [2][3][4]. We used the “normalized” price of one candidate as a forecast probability of the election outcome: (Republic AvgPrice)/(Republic AvgPrice + Democratic AvgPrice). Other data cleansing steps were also taken. We also experimented with the High and Low prices of each party.

2. **Generating topics:** We used PLSA(Probabilistic Latent Semantic Analysis) topic modeling method to find out representative topics from the text data. This uses the Expectation-Maximization (EM) algorithm. We used a PLSA implementation package from [1]. This is many times faster than our previous implementation in MP3. That’s why we used this implementation in our program.

3. **Causal analysis with time series data:** We used the Granger test for measuring causality. We utilized the Python library `grangercausalitytests` from `statsmodels.tsa.stattools` for the Granger test. The output of this part gives significant causal topics with significance >95%. Also, each topic is associated with a corresponding time lag which can describe the causality of the corresponding topic the most.

4. **Word level causality**: We used the Pearson Correlation test for measuring the word level causality in our implementation. Each significant topic determined by the granger test is passed through this next level for finding word-level causality. Within each topic, the words with significant positive correlation and negative correlations are separated and grouped into two distributions. These distributions work as priors in the next iteration.

5. **Generating Topic Priors:** We generated topic priors for the causal topics and incorporated them into the next iteration PLSA.

3) Usage documentation:

Our program was built in a Jupyter notebook and ran on Google Colab. Please see the comments we added inline for instructions on running specific blocks of code. The code blocks can be run sequentially from beginning to end to see the results. Please find the Jupyter notebook on our GitHub repository:
(https://github.com/pritomsaha/CourseProject)

4) Participation:

netid	participation
huy2	● **Logistics:** ○ Create project in CMT & added project meta ○ Group coordination/planning ○ Contribute to proposal/progress report/documentation/presentation ● **Project work:** ○ Contribute to paper investigation, determining implementation steps, finding solution to roadblocks ○ Implemented stock time series cleansing and processing ○ Implement word-level causality modeling ○ Implement topic prior generation
paksash2	● **Logistics:** ○ Create the project on github and upload the project proposal and project progress reports. ○ Contribute to proposal/progress report/documentation/presentation ● **Project work:** ○ Contribute to paper investigation, determining implementation steps, finding solution to roadblocks ○ Extract the appropriate text data from new york time corpus [5]. ○ Preprocess and cleaning text data so that it can be used to train plsa model. ○ Finding out a working fast plsa model [1] and making necessary changes to the implementation of the plsa model so that it is appropriate for the topic modeling in the paper. ○ Making room for incorporating topic prior feedback to the plsa model. ○ Implementing the code for calculating topic coverage that is required in finding causal topics. ○ Combining all the modules (topic modeling, topic-level causality, word-level causality) to make it workable for running.
trishad2	● **Logistics** ○ Group coordination/planning ○ Contributed to the proposal, progress report, presentation, and documentation

	● **Project work**: ○ Contributed to the paper investigation, determining implementation steps, finding the solution to roadblocks ○ Stock time series cleansing and processing ○ Scraped time-series data(Iowa Electronic Markets (IEM)3 2000 Presidential Winner-Takes-All Market, AAMRQ, AAPL stock price data) from websites [2][3][4] ○ Implemented topic-level causality using Granger test ○ Worked on hyperparameter tuning ○ Wrote the documentation and created the PowerPoint presentation ○ Fixed bugs in code

References:

1. https://github.com/henryre/numba-plsa
2. https://iemweb.biz.uiowa.edu/pricehistory/pricehistory_SelectContract.cfm?market_ID=29
3. https://thestockmarketwatch.com/stock/stock-data.aspx?symbol=AAMRQ&action=showHistory&page=1&perPage=25&startMonth=4&startDay=1&startYear=2000&endMonth=9&endDay=30&endYear=2020&endDateLite=11%2F15%2F2020
4. https://finance.yahoo.com/quote/AAPL/history/
5. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T19