CMPT 733 – BIG DATA PROGRAMMING II

Project – Machine Learning Research Exploration

Contents

1. Motivation and Background

With the advancement in computation and data storage, data science has become a highly popular career choice. While many people are comfortable with usual data science techniques such as regression, decision tree, etc., some want to dive into more advanced machine learning topics, including deep learning, reinforcement learning and probabilistic graphical model. However, research in machine learning is fast pace environment with about 50 new publications in major machine learning journals each month. It would be difficult for a new machine learning practitioner to have an overview of and be updated with the research. Our project aims to identify topics in latest machine learning literatures using topic model [1], summarize the trend and explore interesting relationship among the topics. Based on that, we propose a potential data product with improvements over existing research database in the market by introducing similarity-based search and auto topic tagging.

2. Problem Statement

Specifically, our project starts with 3 research questions:

- What are recent topics in machine learning research?
- Is there any interesting relationship among these topics?
- Is it possible to find research document based on relevancy instead of simple keywords?

Answering these question poses a challenge for a human because it is not feasible to read thousands of documents. In additional, it often requires people with some level of expertise in machine learning to understand latest research. Thankfully, these research document are readily available online and we can apply machine learning algorithm to answer above questions with minimum requirement for human to look into the content.

3. Data Processing Pipeline

Our end-to-end solution is illustrated in Figure 3.1:

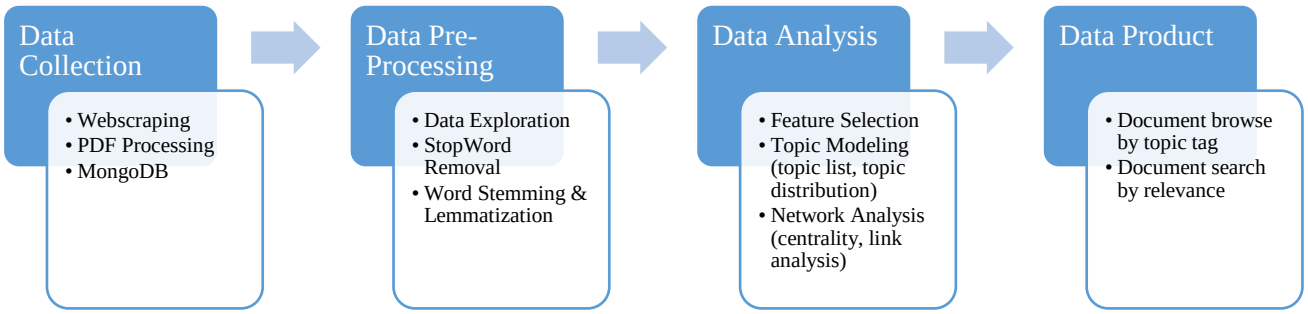


Figure 3.1 – End-to-End Data Processing Pipeline

3.1. Data Collection

Machine Learning journals and conferences would be an excellent source of data for our project. Mostly based on Journal impact factor, we decide to look at 5 major machine learning journals: Journal of Machine Learning Research (JMLR), Neural Information Processing Systems (NIPS), IEEE Pattern Recognition, SpringerLink Machine Learning and arXiV Machine Learning. We developed 5 Scrapy web crawlers to retrieve publications from these journals' websites in the past 3 years (about 7000 document - 4GBs data). The publication PDF files are processed with PDFminer to extract full-text. Then, the research publication details (title, abstract, url, full-text) are stored in MongoDB, a document-oriented NoSQL database.

3.2. Data Pre-Processing

Text data is processed through tokenization, stop-word removal and word lemmatization.

- Tokenization is performed to break text paragraph into words and exclude number/special characters.
- Stop-word removal helps remove undesired words from our corpus to facilitate analysis. In additional to standard Spark stop-word removal, we use wordcloud to visualize most common words in order to decide words to be removed. The benefits brought by stop-word removal are two folds: It helps eliminates words that are meaningless to our analysis (eg. the word "cid" appears in all document) and it reduces computation and memory resources.

- Word stemming and lemmatization helps merge words with the same symantic meaning (eg. “model”, “models”, “modeling”). Standford NLP library is used for this task.

3.3. Data Analysis

Topic Modeling is a type of statistical model for discovering the latent "topics" that occur in a collection of documents. In addition to the list of identified topics, Topic Model also provide topic distribution, i.e. what is contribution of each topic for a document. This topic distribution is the key to enable further network analysis and data product development.

We use network analysis to visualize and understand relationship among the topics. A network is constructed based on the occurrence of topics in document. Each vertex represents a topic, and each edge represent a certain association between two topics. We can look at nodes with high centrality as well as interesting associations and clusters to understand research trends.

3.4. Data Product

Although existing research database systems in the market have high-level categories, most of them do not have keyword/topic tag to enable fast and easy document browsing. In additional, search function provided by these systems are just simple keyword search to check whether certain words appear in a document, which might return document with little relevance. With topic distribution for each document, we propose a web-based data product which can address these short-coming of existing research database systems in the market.

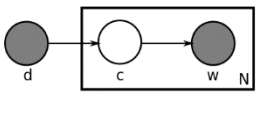
4. Methodology

4.1. Topic Model and Feature Selection

There are two common approach for topic modeling:

- Latent Semantic Analysis (LSA) replies on a matrix decomposition (Singular -Value Decomposition) on the term-document matrix to obtain a low rank approximation for such matrix.

- Latent Dirichlet Allocation (LDA) assumes that the occurrences of words in documents (w, d) that we observed are influenced by a certain number (k) of latent topics (c). Estimating probability distribution of word-topic and document-topic in LDA is computationally intractable. LDA replies on MCMC technique or variational approximation, which are still computational intensive, an ideal task to for Spark.



$$P(w,d) = \sum_c P(c)P(d|c)P(w|c)$$
$$= P(d)\sum_c P(c|d)P(w|c)$$

Although LSA model is much faster to train, LDA usually provides more meaningful topic identification [2]. Therefore, we decide to use LDA for our analysis.

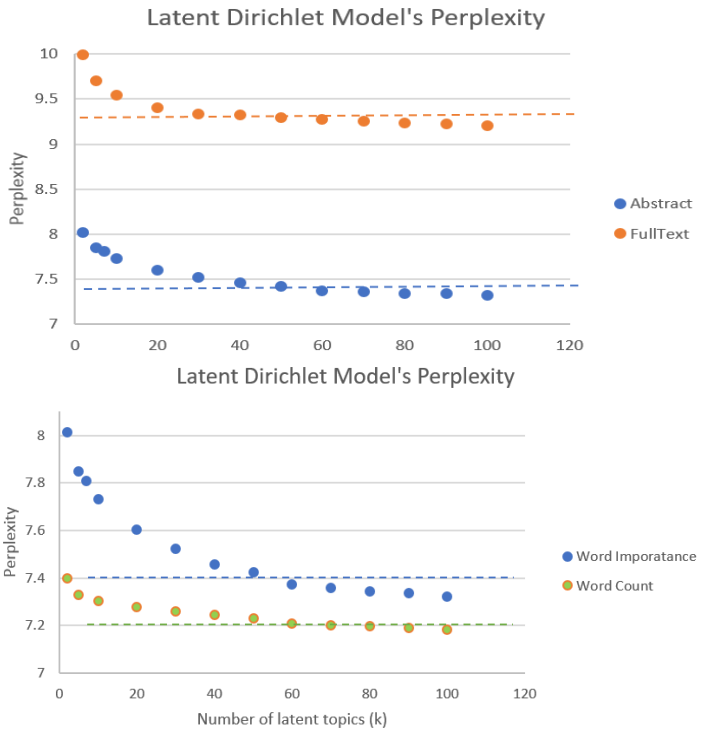
We can apply LDA on the document’s “abstract” or “full-text” (which contains abstract”) using simple word count (term-frequency) or word importance (term frequency–inverse document frequency). In theory, best feature and model can be selected by choosing the model with the lowest perplexity, which represent how impurity the distribution of topics is. However, in practice, models which achieve better predictive perplexity might have less interpretable latent spaces [3]. Therefore, we also need to look at “word intrusion”, i.e. whether topics inferred by a model are semantically “cohesive”.

There are several hyper-parameters we need to decide:

- Number of iterations to train LDA model
- Use “abstract” or “full-text” to derive feature
- Use word count vs word importance as feature

The first parameter, number of iterations to train LDA model, is easy to determine by looking at topic drift, i.e. is there alarming differences in important words in each topic across several runs? Its default value in Spark is 100, but it is not sufficient for LDA model to converge in our case. We settle with 500 iterations.

We run LDA model with “abstract” vs “full text” using word importance (term frequency–inverse document frequency) as feature. Using “abstract” as feature results in model with much lower perplexity. The result is unexpected because we hoped that “full text” would provide more information on topics covered by a document. Examining important words in each topic with number of topics k = 40, we do not notice any alarms in terms of topic meaning. Therefore, we decide to use “abstract” for our analysis.



Using “abstract” to derive feature, we run LDA model with word count vs word importance as feature. The result shows model with word count feature has marginally lower perplexity. However, topics identified using word count model are less meaningful and tend to include the same words with high frequency in corpus such as “model”, “learn”, “use”. We decide to keep word importance as feature.

With word importance based on “abstract” as feature, we can see that the model has little improvement in perplexity at around k = 50. We examine identified topics at k = 40, 45, 5, 55 to see which model gives the more meaningful topics and less unclear topics. We settle with number of topics k = 45. Most topics makes sense, but we do have 5 unclear/unknown topics out of 45 (roughly 12%). Details of the topic distribution and visualization of word importance for some topics are below:

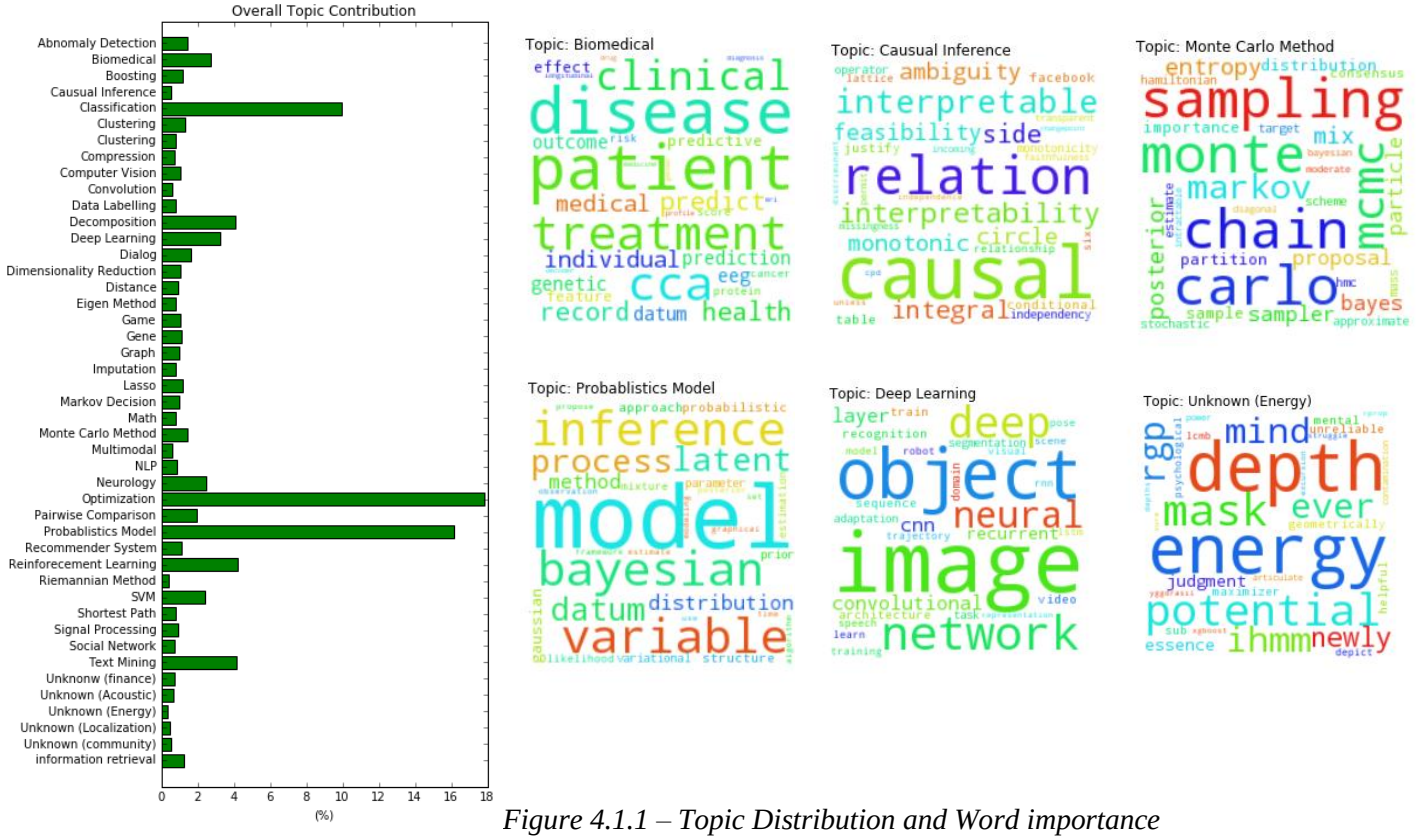
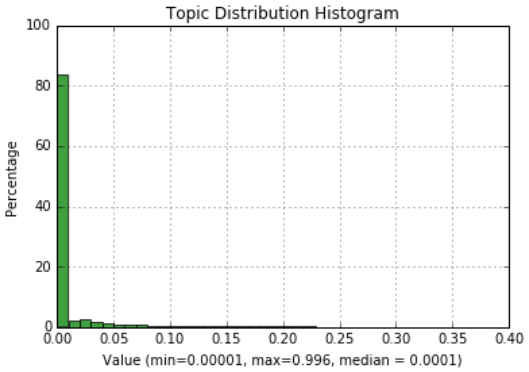


Figure 4.1.1 – Topic Distribution and Word importance

Looking at topic distribution over the whole document set, we can see that major machine learning research areas, in deceasing order, include Optimization (18%), Probabilistic Model (16%), Deep Learning (4%), Text Mining (4%), Biomedical (3%). That is interesting result because we expected Optimization topic to be a mature field and have less research, while Deep Learning is an very popular topic in recent years and should account for most research. Clearly, our impression is not what the data has shown.

4.2. Network Analysis

We use network analysis to explore relationship among topics. We need to construct a graph based on topic distribution for individual document, with each topic to be a vertex and each edge between two vertices to indicate if there is a document belongs to the two topics. Given that topic model provides topic distribution of all topics, i.e. document belongs to all topics to a certain degree, we need to filter out insignificant document-topic weight. As expected, most of the document-topic weights are very small (close to 0). We tried using different cut-off threshold. At cut-off value 0.0005 (80th percentile), the graph is still dense with almost all vertices linked to each other. We settle with cut-off value 0.0446 (90th percentile).



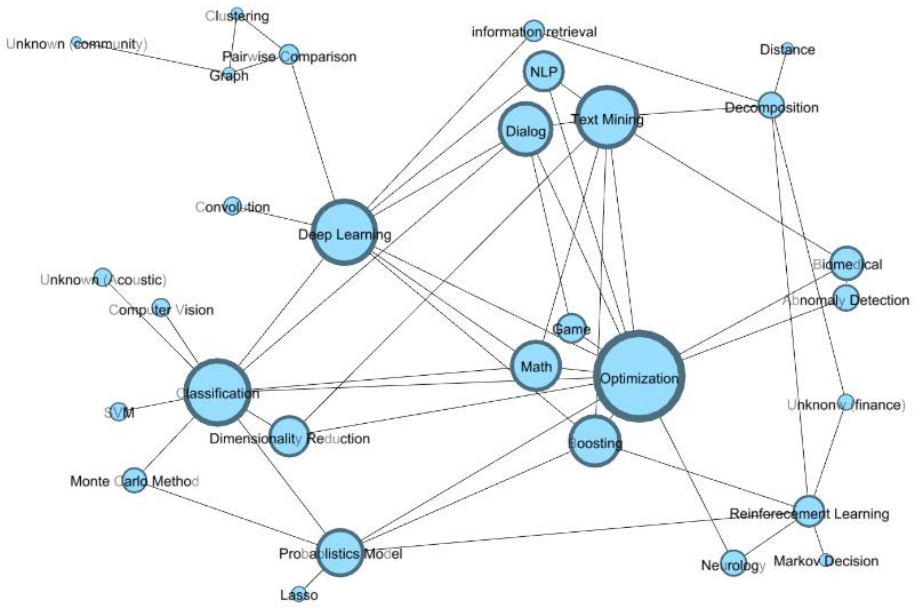
Even though two topics have the same document, their relationship might not be interesting for us to look at. For example, we expect that most research document in "Computer Vision" topic are also in "Convolution" topic. We can use Chi-square test to determine if an association between two topics is interesting [4]. We only keep topic relationship if the Chi-square test statistics indicates significant association at 99.99% confidence level.

	Yes B	No B	B Total
Yes A	YY	YN	YA
No A	NY	NN	NA
B Total	YB	NB	T

$$\chi^2 = T\frac{(YY * NN - YN * NY)^2}{YA * NA * YB * NB}$$

To measure topic importance in the whole network, we can look at degree centrality or Eigen centrality. Eigen centrality is preferred because it considers the importance of neighbours. In the above graph, vertex size represents Eigen centrality. As expected, "Optimization" topic has the highest centrality since it is probably a foundation topic that other topics rely on.

What is more interesting is to look at edges between topics and triangles formed by some topics since they might represent an interesting relationship or cluster. "Graph", "Clustering", "Pairwise Comparison" topics form a triangle representing a close relationship among the three topics as expected. Unexpected relationships include Deep Learning – Boosting and Biomedical – Text Mining. These relationships indicate a research trend to combine some of these topics together. For example, when we look at document belongs to Deep Learning and Boosting, we realize there are interesting techniques such as Deep Forest, in which the author proposed using decision tree instead of perceptron as a node a deep network.



5. Data Product

With topic distribution provided by LDA model, we can represent each document as a vector of topics indicating how much each topic contributes to the document. Based on that, we compute similarity across document or between a search query to document using Cosine Similarity. This is the enabler for our web-based data product:

- Auto-Tag: For an existing document or even a new submission, the system can automatically determine and suggest topic tags for the document
- Browse: Users can then conveniently browse research document using topic tags
- Search: Users can also key in an input query, which can be simple keywords or a full abstract of research paper, the system can return relevant document based on topic distribution similarity.

Some screenshots of our data product demo to demonstrate its capability are shown below:

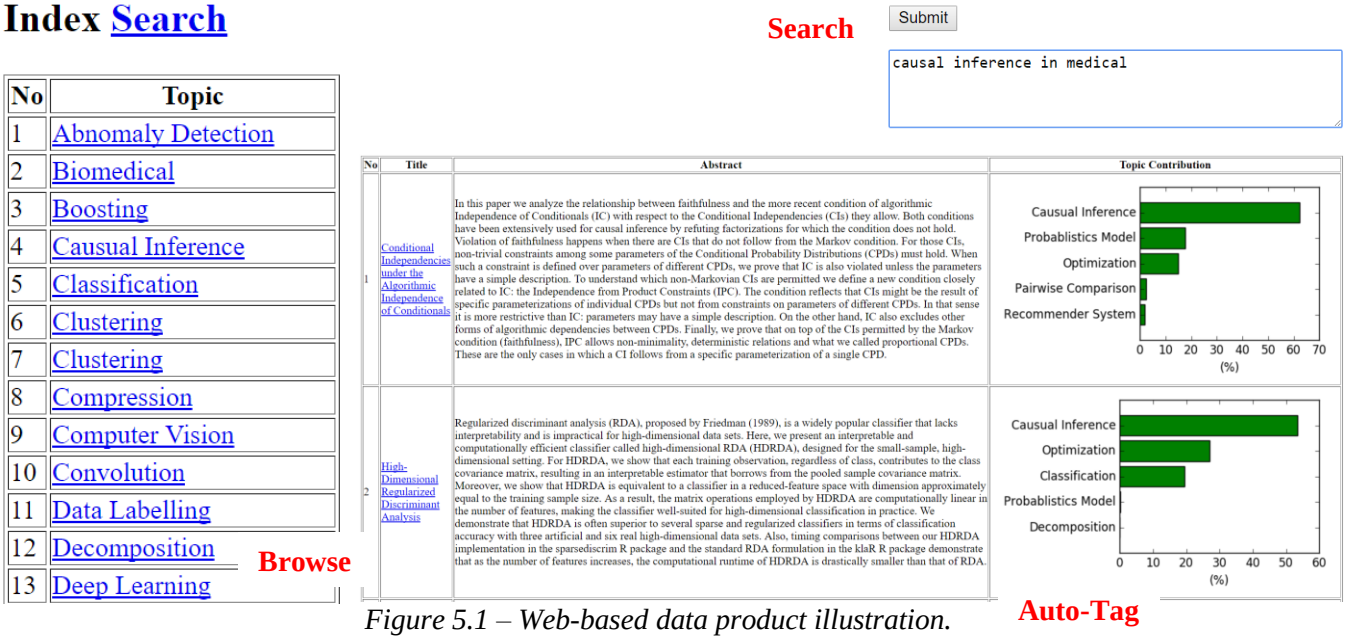


Figure 5.1 – Web-based data product illustration.

6. Lessons Learnt

In terms of methodology, we realize several important key lessons during our analysis and implementation:

- Initial data exploration can be very beneficial for model fitting and training time reduction. It helps eliminate noises that are not important to analytics model and reduce computation resources
- Lower loss/perplexity does not always mean better model. Sometimes, we tend to construct a problem as an optimization task, but we should always try to visualize the result and interpret it. A model can tell a coherent and sensible story even though it might have higher loss.
- A backup plan is never redundant. We developed our project codes to run on Spark 2.0, which was suddenly disabled one week before the project presentation. Luckily, we already finished all heavy computation and stored the result in Hadoop cluster, and the remaining tasks can be done locally.

In terms of technology, we learn to use Scrapy for web crawling, MongoDB for document storage, Spark-ML LDA for topic model, Spark-GraphX for network computation, wordcloud for text visualization, Gephi for network visualization, and Django for web-based prototype.

7. Summary

In summary, we perform web-scraping to retrieve document from major machine learning publications, perform data pre-processing and implement LDA model to identify latent topics in the document set. Major recent research topics in machine learning include Optimization, Probabilistic Model, Deep Learning, Text Mining, Biomedical. We use network analysis to explore interesting topic relationships, some of which include Deep Learning – Boosting and Biomedical – Text Mining. Finally, we compute document similarity based on topic distribution and enable our data product which provide users with auto-tag, browse and relevance – based search capabilities.

REFERENCES

1. *Blie et al., 2003,* Latent Dirichlet Allocation [link]
2. *Tuomo et al. 2006,* Applying Latent Dirichlet Allocation to Automatic Essay Grading [link]
3. *Chang et al. 2009,* Reading Tea Leaves: How Humans Interpret Topic Models [link]
4. *Sandy et al. 2015,* Advanced Analytics with Spark, Chapter 7 - Analyzing Co-occurrence Networks with GraphX [link]