

Old Dominion University

Department of Computer Science

Topic Modeling
NIPS Conference Papers

Fall, 2018

Pratik Navale

pnav001@odu.edu

TABLE OF CONTENTS

<u>CONTENT</u>	<u>PAGE</u>
ABSTRACT	2
INTRODUCTION	
About NIPS	2
Document classification	2
Topic modeling	3
TASKS	
Problem statement	3
Design and implementation	3
Overview of steps involved	4
Data pre-processing	5
Initial shot with supervised learning	7
Topic modeling: NMF and LDA	8
OUTPUT	
Topics	10
Top document titles grouped by topics	11
Evaluation metrics	12
Visualizing clusters	13
pyLDavis	13
REFERENCES	14

Abstract

Topic modeling is one of the most effective data mining methodologies in Natural Language Processing (NLP). Many researchers have done a significant work in this field using various techniques and algorithms. In this study, we explore the Neural Information Processing Systems (NIPS) dataset of conference papers to classify the documents. Primarily, two most famous topic modelling algorithms Non-negative Matrix Factorization (NMF) and Latent Dirichlet Allocation (LDA) have been used to analyze the data. Further, we focus on LDA model for our work since LDA outperforms NMF for this dataset. The results can be further explored and analyzed using pyLDAvis visualization.

Keywords: topic modeling, LDA, NIPS dataset, text classification

Introduction

About NIPS

Neural Information Processing Systems (NIPS) is one of the top machine learning conferences in the world. It covers topics ranging from deep learning and computer vision to cognitive science and reinforcement learning and about 2,500 papers are submitted every year. When it comes to separating the useful information from the irrelevant, or getting an overview of hundreds of these papers submitted, document classification is a worthwhile tool that can reduce the cost, time and essentially effort, of searching and retrieving the information that matters.

Document Classification

Document classification is the task of grouping documents into categories based upon their content such as words, phrases and word combinations. Document classification is a significant learning problem that is at the core of many information management and retrieval tasks. It performs an essential role in various applications that deals with organizing, classifying, searching and concisely representing a significant amount of information.

Automatic document classification can be broadly classified into two categories. These are supervised document classification and unsupervised document classification. In supervised document classification technique, some mechanism external to the classification model (generally human) provides information related to the correct document classification - such as labels. Thus, in case of supervised document classification, it becomes easy to test the accuracy of the document classification model. In unsupervised document classification, no information is provided by any external mechanism whatsoever. However, it can be achieved through clustering techniques. Some supervised algorithms that can be used for the purpose of document classification include Naive-Bayes, logistic regression, SVM, decision trees, etc. Unsupervised machine learning algorithms that can be used for document classification task

include clustering and dimensionality reduction methods, such as K-means, LDA (Latent Dirichlet Analysis), and NMF (Non-negative Matrix factorization).

Topic Modeling

Topic Modeling is an unsupervised learning approach to clustering documents, to discover topics based on their contents. It is very similar to how K-Means algorithm and Expectation-Maximization work. There are several scenarios when topic modeling can prove useful. Here are some of them:

Text classification – Topic modeling can improve classification by grouping similar words together in topics rather than using each word as a feature.

Recommender systems – Using a similarity measure we can build recommender systems. If our system would recommend articles for readers, it will recommend articles with a topic structure similar to the articles the user has already read.

Uncovering themes in texts – Useful for detecting trends in online publications for example.

Some well known topic modeling algorithms are LDA – Latent Dirichlet Allocation – the one we'll be focusing in this project. Its foundations are probabilistic graphical models. The central idea in LDA is to view each document as a mixture of topics, and try to learn these topics and words generated by each topic for each document. This model also provides a more compressed format to represent documents, which is very useful when handling large corpus. Latent topic models are statistical models for discovering the "topics" that occur in a collection of documents. Another topic modeling algorithm is NMF – Non-Negative Matrix Factorization which is based on linear algebra. We have also considered NMF in our project for modeling topics, however, we found LDA makes better topic models for our dataset.

Problem Statement



To put it in a sentence, our goal is to develop a topic model that effectively classifies NIPS conference papers into appropriate categories or topics with credible accuracy, and visualize the topics discovered by the model to get insights.

Design and Implementation

Dataset

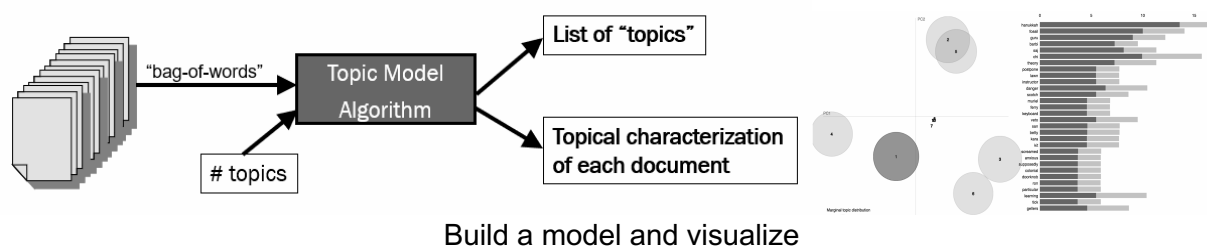
The dataset for this project includes the title, authors, abstracts, and extracted text for all NIPS papers to date (ranging from the first 1987 conference to the current 2017 conference). The paper text was extracted from raw PDF files of each paper and consolidated into both CSV files and as SQLite database. The data has been sourced from Ben Hamner, Kaggle Datasets.

We are interested in exploring the CSV file to determine what type of data we can use for the analysis and how it is structured. A research paper typically consists of a title, an abstract and the main text. Other data such as figures and tables were not extracted from the PDF files. Each paper discusses a novel technique or improvement. In this analysis, we will focus on analyzing these papers - hence further exploring papers.csv.

	# id	# year	A title	A event_type	A pdf_name	A abstract	A paper_text
			7241 unique values	Poster 30% Spotlight 2% Other (1) 68%	7241 unique values	Abstract Missing 46% We study the pow... 0% Other (3921) 54%	7237 unique values
1	1	1987	Self-Organization of Associative Database and Its Applications		1-self-organization-of-associative-database-and-its-applications.pdf	Abstract Missing	767 SELF-ORGANIZATION OF ASSOCIATIVE DATABASE AND ITS APPLICATIONS Hisashi Suzuki and Suguru Arimoto Osaka University, Toyonaka, Osaka 560, Japan ABSTRACT An efficient method of self-organizing assoc...
2	10	1987	A Mean Field Theory of Layer IV of Visual Cortex and Its Application to Artificial Neural Networks		10-a-mean-field-theory-of-layer-iv-of-visual-cortex-and-its-application-to-artificial-neural-networks.pdf	Abstract Missing	683 A MEAN FIELD THEORY OF LAYER IV OF VISUAL CORTEX AND ITS APPLICATION TO ARTIFICIAL NEURAL NETWORKS* Christopher L. Scofield Center for Neural Science and Physics Department Brown University Provi...
3	100	1988	Storing Covariance by the Associative Long-Term Potentiation and Depression of Synaptic Strengths in the Hippocampus		100-storing-covariance-by-the-associative-long-term-potentiation-and-depression-of-synaptic-strengths-in-the-hippocampus.pdf	Abstract Missing	394 STORING COVARIANCE BY THE ASSOCIATIVE LONG? TERM POTENTIATION AND DEPRESSION OF SYNAPTIC STRENGTHS IN THE HIPPOCAMPUS Patric K. Stanton?

A look into the NIPS papers dataset

Overview of Implementation Steps



Various implementation steps that went into building the model include:

- Data preprocessing
 - Data cleaning
 - Removing numbers
 - Converting text to lowercase
 - Tokenize text
 - Stemming/lemmatization
 - Removing extremely short words
 - Removing stopwords
 - Vector representation of text (bag-of-words model)
 - Feature selection and feature transformation
- Training the topic model algorithm (NMF and LDA)
- Obtain output matrices from the model
- List representation of topics and documents
- Visualize the topics

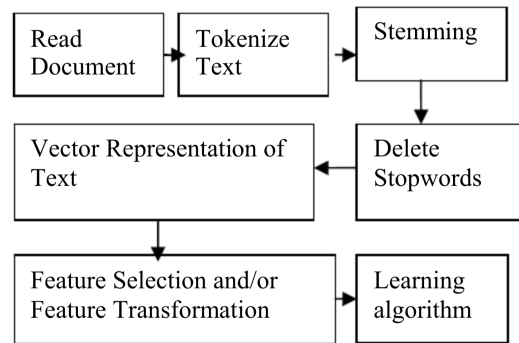
Data Preprocessing

Topic models do not have any actual semantic knowledge of the words, and so do not “read” the sentence. Instead, topic models use math. The tokens/words that tend to co-occur are statistically likely to be related to one another. However, that also means that the model is susceptible to “noise,” or falsely identifying patterns of co-occurrence if non-important but highly-repeated terms are used. As with most computational methods, “garbage in, garbage out.” Hence, we employ data cleaning techniques to feed “good” and meaningful data into the learning algorithm.

In order to make sure that the topic model is identifying interesting or important patterns instead of noise, the following preprocessing or “cleaning” steps were taken:

- Removing numbers/digits
- Converting text to lowercase
- Tokenization of text
- Stemming/lemmatization of text
- Removing extremely short words
- Removing stop words

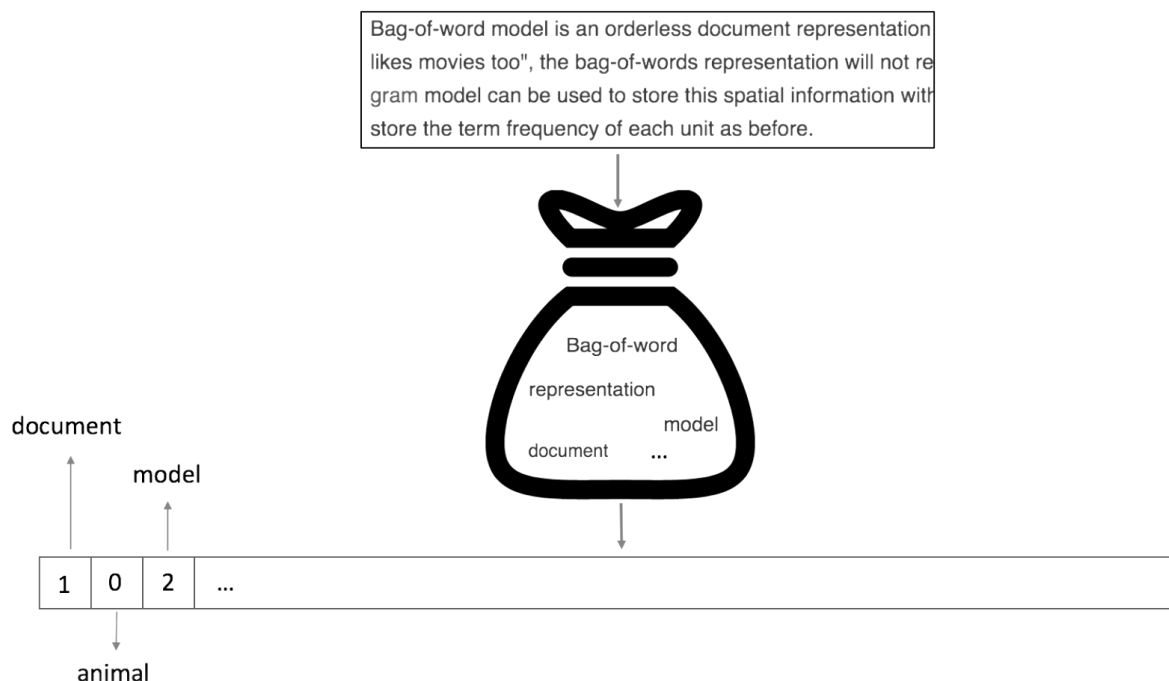
Tokenizing each sentence into a list of words helps in creating the bag-of-words model that topic model algorithm takes. Also, removing punctuations and unnecessary characters altogether help in generating meaningful topics. We do not want frequently appearing words in English such as ‘a’, ‘the’, ‘if’, ‘but’, etc, in our topics as they do not add meaning or context to the topics we are interested to discover from our data. Stemming is the process of cutting the words to its root words. For example, ‘networking’, ‘networks’ become ‘network’ and ‘generalize’, ‘generalization’ become general and ‘better’ and ‘best’ becomes ‘good’. The figure below gives an abstract idea about the preprocessing steps.



Abstract look into data preprocessing steps

Bag of words and vector representation

Traditionally, text documents are represented in NLP as a bag-of-words. This means that each document is represented as a fixed-length vector with length equal to the vocabulary size. Each dimension of this vector corresponds to the count or occurrence of a word in a document. Being able to reduce variable-length documents to fixed-length vectors makes them more amenable for use with a large variety of machine learning (ML) models and tasks.



Bag of words model for document analysis

The image above illustrates how a document is represented in a bag-of-word model: the word "document" has a count of 1, while the word "model" occurs twice in the text.

The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents.

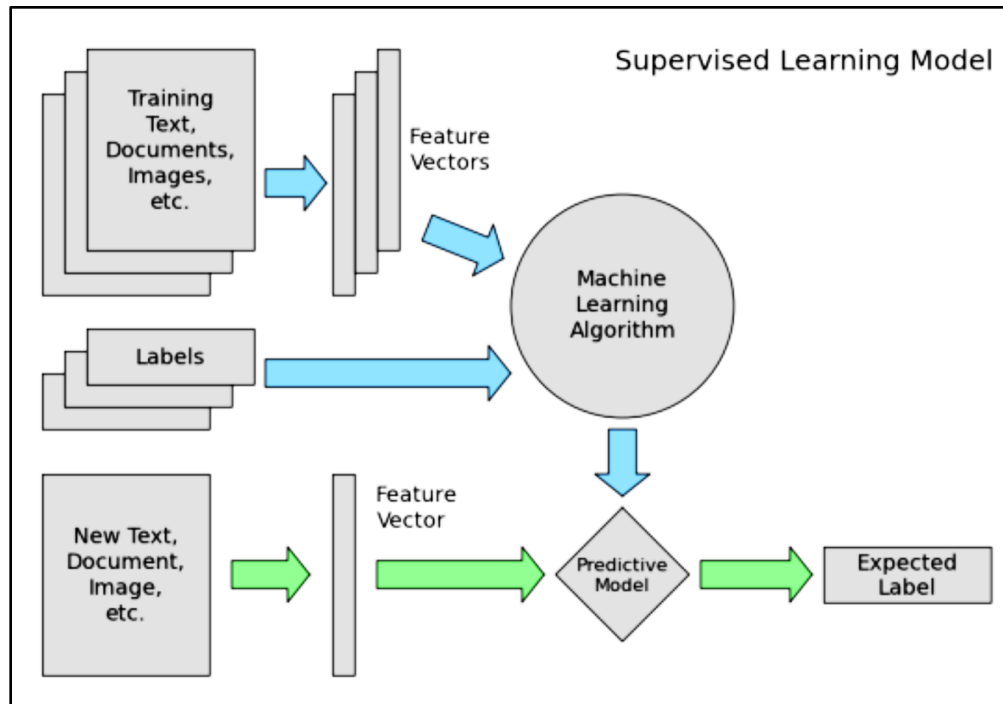
So, to create the doc-word matrix, first initialise the CountVectorizer class with the required configuration and then apply fit_transform was applied to actually create the matrix, which can be fed into LDA model. Similarly, tf-idf (term-frequency inverse document frequency) vectorizer was used for giving as input into NMF model as NMF can take it as input.

Initial Shot with Supervised Learning Algorithms

Text classification as a supervised machine learning task generally gives better accuracy since a labelled dataset containing text documents and their labels are used for training a classifier. After the preprocessing step, came the time to feed data into a machine learning model. We wanted to take a shot at supervised learning for which Naive-bayes classifier was considered. An end-to-end supervised text classification pipeline is composed of these main components:

1. **Dataset Preparation:** The first step is the dataset preparation step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then split into training and validation sets.
2. **Feature Engineering:** The next step is the feature engineering step in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data.
3. **Model Training:** The final step is the model building step in which a machine learning model is trained on a labelled dataset.
4. **Improve Performance of Text Classifier:** There are also different ways to improve the performance of text classifiers, which are done incrementally.

The figure below gives a better understanding of the steps that go into classifying documents the supervised way:



Supervised text classification model

For classifying documents the supervised way, we needed labels for our data. Generating labels for over 7000 documents that could serve as ground truth, was a very challenging task. Later on, after giving considerable thought into generating labels for our dataset, we decided to go with unsupervised learning model for text classification.

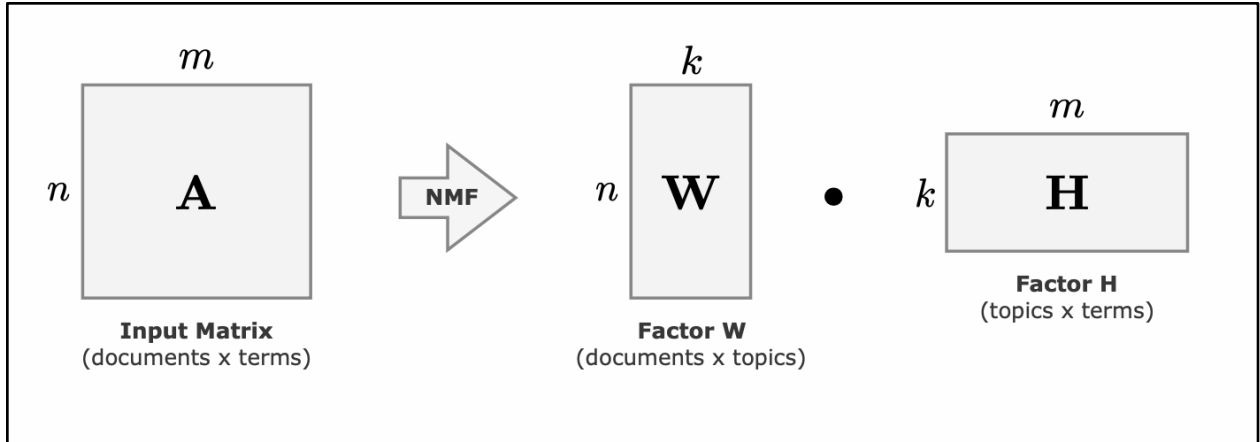
Topic modeling with NMF and LDA

Non-Negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) is a state of the art feature extraction algorithm. NMF is useful when there are many attributes and the attributes are ambiguous or have weak predictability. By combining attributes, NMF can produce meaningful patterns, topics, or themes.

NMF decomposes multivariate data by creating a user-defined number of features. Each feature is a linear combination of the original attribute set; the coefficients of these linear combinations are non-negative.

NMF decomposes a data matrix V into the product of two lower rank matrices W and H so that V is approximately equal to W times H . NMF uses an iterative procedure to modify the initial values of W and H so that the product approaches V . The procedure terminates when the approximation error converges or the specified number of iterations is reached.



NMF algorithm in terms of input and output matrices

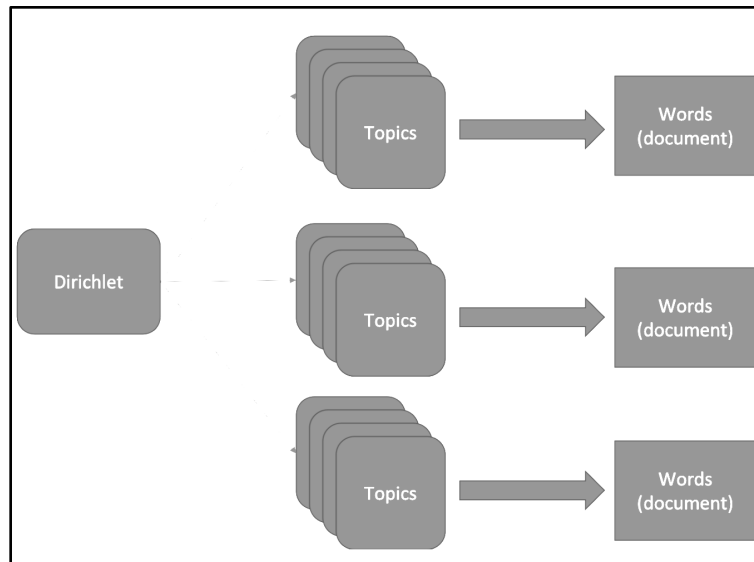
During model apply, an NMF model maps the original data into the new set of attributes (features) discovered by the model.

Latent Dirichlet Allocation (LDA)

LDA or Latent Dirichlet Allocation is a “generative probabilistic model” of a collection of composites made up of parts. In terms of topic modeling, the composites are documents and the parts are words and/or phrases (n-grams).

The probabilistic topic model estimated by LDA consists of two tables (matrices). The first table describes the probability or chance of selecting a particular part when sampling a particular topic (category). The second table describes the chance of selecting a particular topic when sampling a particular document or composite.

If you choose the number of topics to be less than the documents, using LDA is a way of reducing the dimensionality (the number of rows and columns) of the original composite versus part data set. With the documents now mapped to a lower dimensional latent/hidden topic/category space, you can now apply other machine learning algorithms which will benefit from the smaller number of dimensions. This is the main theme of generating topics with LDA. This is illustrated in the figure below.



LDA: discovering topics from words and documents

Parameters of LDA

Alpha and beta hyperparameters – Alpha represents document-topic density and Beta represents topic-word density. Higher the value of alpha, documents are composed of more topics and lower the value of alpha, documents contain fewer topics. On the other hand, higher the beta, topics are composed of a large number of words in the corpus, and with the lower value of beta, they are composed of few words.

Number of topics – Number of topics to be extracted from the corpus. We have tried various values for this parameter and found 7 topics as ideal, based on the topic formation and visual analysis of clusters formed.

Number of Topic Terms – Number of terms composed in a single topic. It is generally decided according to the requirement. If the problem statement talks about extracting themes or concepts, it is recommended to choose a higher number, if problem statement talks about extracting features or terms, a low number is recommended.

Number of iterations/passes – Maximum number of iterations allowed to LDA algorithm for convergence. We chose 10 iterations for different number of topics.

Output and Evaluation

We ran multiple iterations of both LDA and NMF models on our data with various values for 'number of topics'. The displayTopics() method in our python program outputs the top topics identified by the models as a list with top words from which topics can be inferred. Also, list of

relevant titles of papers grouped by each topic is revealed so that the papers can be identified and read to find relevance of those papers to their identified topics.

The output cluster formation with 5 topics gave reasonably well defined topics and the topic clusters were found to be well separated. The inferred topics, the titles of most relevant documents and topic visualization with pyLDavis (python library based on sklearn) are given below:

<p>NMF Topics</p> <p>Topic 0: algorithm kernel bound problem optim matrix learn method xi convex theorem loss data estim rank</p> <p>Topic 1: network imag train layer learn unit input featur neural output weight hidden recognit object deep</p> <p>Topic 2: polici action reward state agent learn reinforc regret algorithm control valu optim game time trajectori</p> <p>Topic 3: neuron spike cell synapt synaps activ respons stimulus input time circuit network neural signal model</p> <p>Topic 4: model data distribut infer estim posterior sampl gaussian latent prior likelihood bayesian mixtur variabl paramet</p>
<p>Top NMF documents' titles:</p> <p>Topic 0: FALKON: An Optimal Large Scale Kernel Method Estimation, Optimization, and Parallelism when Data is Sparse Accuracy at the Top Lower Bounds on Rate of Convergence of Cutting Plane Methods Stochastic Approximation for Canonical Correlation Analysis</p> <p>Topic 1: A Powerful Generative Model Using Random Weights for the Deep Image Representation Handwritten Digit Recognition with a Back-Propagation Network ImageNet Classification with Deep Convolutional Neural Networks Adaptive dropout for training deep neural networks Rapidly Adapting Artificial Neural Networks for Autonomous Navigation</p> <p>Topic 2: Adaptive Skills Adaptive Partitions (ASAP) Model-Free Least-Squares Policy Iteration A Convergent Form of Approximate Policy Iteration Experimental Results on Learning Stochastic Memoryless Policies for Partially Observable Markov Decision Processes Improved Switching among Temporally Abstract Actions</p> <p>Topic 3: Non-Boltzmann Dynamics in Networks of Spiking Neurons Attentional Processing on a Spike-Based VLSI Neural Network Temporal Coding using the Response Properties of Spiking Neurons Active dendrites: adaptation to spike-based communication Associative Memory in a Network of 'Biological' Neurons</p> <p>Topic 4: Hidden Markov Model Induction by Bayesian Model Merging Variational Inference for Bayesian Mixtures of Factor Analysers Bayesian Active Model Selection with an Application to Automated Audiometry Bayesian Sparse Factor Models and DAGs Inference and Comparison Dependent Multinomial Models Made Easy: Stick-Breaking with the Polya-gamma Augmentation</p>

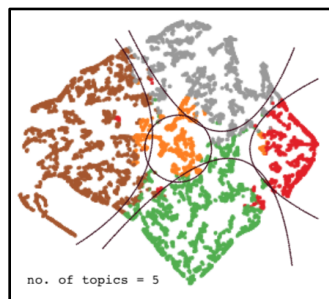
Topics and document titles inferred by NMF model

<p>LDA Topics</p> <p>Topic 0: algorithm matrix bound rank problem optim log learn theorem time ani follow regret sampl let</p> <p>Topic 1: learn algorithm method data kernel optim problem point error label xi train bound estim exampl</p> <p>Topic 2: graph algorithm node tree cluster problem network structur variabl model edg number approxim time xi</p> <p>Topic 3: network model imag train learn input neural featur figur neuron layer time object output perform</p> <p>Topic 4: model distribut state learn estim data sampl time paramet process observ polici gaussian valu infer</p>	
<p>Top LDA documents' titles:</p> <p>Topic 0: Fast and Memory Optimal Low-Rank Matrix Approximation Algorithms for Infinitely Many-Armed Bandits Prediction strategies without loss Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence Online Learning with Switching Costs and Other Adaptive Adversaries</p> <p>Topic 1: Rates of Convergence for Nearest Neighbor Classification Learning Non-Linear Combinations of Kernels Metric Learning with Multiple Kernels Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions Learning Kernels with Radiuses of Minimum Enclosing Balls</p> <p>Topic 2: Augmentative Message Passing for Traveling Salesman Problem and Graph Partitioning New Rules for Domain Independent Lifted MAP Inference Clusters and Coarse Partitions in LP Relaxations Approximating MAP by Compensating for Structural Relaxations Lifted Inference Rules With Constraints</p> <p>Topic 3: Context dependent amplification of both rate and event-correlation in a VLSI network of spiking neurons Extending Phase Mechanism to Differential Motion Opponency for Motion Pop-out Extending position/phase-shift tuning to motion energy neurons improves velocity discrimination A Computer Simulation of Olfactory Cortex with Functional Implications for Storage and Retrieval of Olfactory Information Do Deep Neural Networks Suffer from Crowding?</p> <p>Topic 4: Bayesian Hierarchical Reinforcement Learning Particle Gibbs for Infinite Hidden Markov Models The Infinite Gaussian Mixture Model Active Learning of Model Evidence Using Bayesian Quadrature Fisher Scoring and a Mixture of Modes Approach for Approximate Inference and Learning in Nonlinear State Space Models</p>	

Topics and document titles inferred by LDA model

After thorough analysis of inferred topics and their titles from both NMF and LDA models, we realized that the topics inferred by LDA model composed better word distribution throughout topics. Hence, we lean towards the LDA model and its output as it gives better results for our dataset.

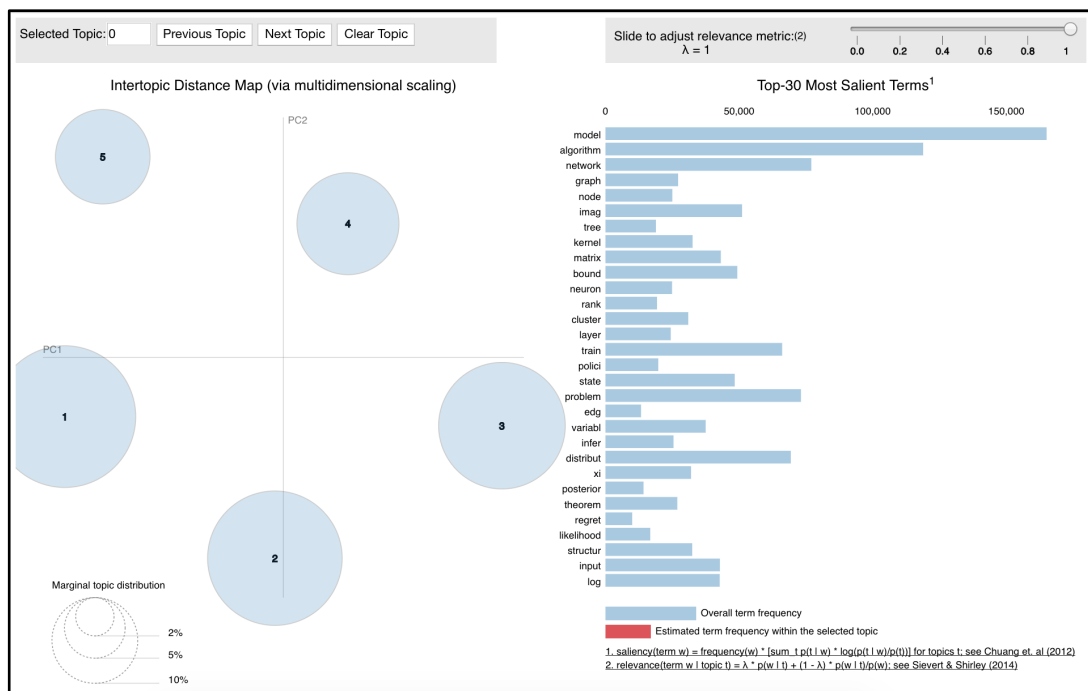
Further, the visualization of the cluster of documents by LDA for 5 topics is given below.



Cluster analysis: Visualizing document distribution by LDA

Visualizing Inferred Topics

Below is an illustration of 'pyLDavis', our interactive visualization for topic models fit using LDA. We can see the 5-topic model fit to the NIPS conference papers data.



Visualizing inferred topics by LDA for further analysis

We can see that the topics (circles on the left) are well defined and spread out across the distribution. Topics far from each other are contextually dissimilar. Closer the topics, similar their context. On the right hand side is a bar graph that visualizes top 30 words in the document corpus overall. Clicking on a topic (circle) changes the words on the right.

To further explore the visualization:

Click a circle in the left panel to select a topic, and the bar chart in the right panel will display the 30 most relevant terms for the selected topic, where we define the relevance of a term to a topic, given a weight parameter, $0 \leq \lambda \leq 1$, as $\lambda \log(p(\text{term} | \text{topic})) + (1 - \lambda) \log(p(\text{term} | \text{topic}) / p(\text{term}))$. The red bars represent the frequency of a term in a given topic, (proportional to $p(\text{term} | \text{topic})$), and the blue bars represent a term's frequency across the entire corpus, (proportional to $p(\text{term})$). Change the value of λ to adjust the term rankings -- small values of λ (near 0) highlight potentially rare, but exclusive terms for the selected topic, and large values of

λ (near 1) highlight frequent, but not necessarily exclusive, terms for the selected topic. Setting λ near 0.6 aids users in topic interpretation, although we expect this to vary across topics.

References

1. Bansal, S. (2016, August 24). Beginners Guide to Topic Modeling in Python. Analytics Vidhya. Retrieved from <https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/>
2. Bhaskar, A. (Spring 2017). *Document classification using machine learning*. Retrieved from https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1531&context=etd_projects
3. Brownlee, J. (2017, September 29). *How to Prepare Text Data for Machine Learning with scikit-learn*. Retrieved from <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/>
4. Edwin Chen, *Introduction to Latent Dirichlet Allocation*. Retrieved from <http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/>
5. Li, S. (2018, February 19). *Multi-Class Text Classification with Scikit-Learn*. Retrieved from <https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f>
6. Sarkar, T. (2018, October 29). *Machine learning with Python: Essential hacks and tricks*. Master machine learning, AI, and deep learning with Python. Retrieved from <https://opensource.com/article/18/10/machine-learning-python-essential-hacks-and-tricks>
7. Scikit-learn. *Working with text data*. Retrieved from https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
8. Shaikh, J. (2017, July 23). *Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK*. Retrieved from <https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a>