Latent Dirichlet Allocation Libraries

CS410 Text Information Systems

Jason Lundquist

jasondl3@illinois.edu

The following is an overview of the topic model known as Latent Dirichlet Allocation (LDA). LDA is a conventional model which can be applied to collections of discrete data in any area of interest, in this summary I will focus on LDA in the context of text modeling and analysis. Using this method for discovering topics among a given collection of text data this evaluation provides a comparison of libraries which implement LDA.

Overview

Topic modeling methods organize, understanding, search, and summarize text data; they are routinely used to discover hidden themes in text collections to classify documents into discovered themes, using these themes to organize, summarize, and search documents. Topic models evaluate relevance with respect to an input query without the need to evaluate the entire document. Therefore, by interpreting the document based on the topics predicted by the modeling method, the outcome is an optimized search process.

Latent Dirichlet Allocation is a generative probabilistic model for text data, a model where text is generated according to a specific set of rules involving probabilities. For each document, randomly assign each word in the document to one of *k* topics where *k* is chosen beforehand. For each document *d*, process each word *w* and compute: *p*(topic *t* | document *d*): the proportion of words in document *d* that are assigned to topic *t*. LDA establishes the words that belong to the topic *t* for a given document *d*. When many words from *d* belong to *t*, it is more probable that word *w* belongs to *t*, (#words in *d* with *t* +alpha / #words in *d* with any topic+ *k**alpha) *p*(word *w*| topic *t*): the proportion of assignments to topic *t* over all documents that come from this word *w*. LDA represents documents as a mixture of topics and a topic as a mixture of words. If a word has high probability of being in a topic, all the documents having *w* will be more strongly associated with *t* as well. Similarly, if *w* is not very probable to be in *t*, the documents which contain the *w* will have a low probability of being in *t* as the rest of the words in *d* will belong to another other topic and therefore *d* will have a higher probability for those topics.

LDA conducts text analysis well because it involves a set of topics, in this context intended to match the colloquial, human readable idea of a topic. In this overview I walk through the process of generating text data within the model, note that text data is viewed here as a “bag of words” where a single document as a collection of all its words. To begin, the model has a set of *k* topics, where a topic is defined as a probabilistic weighting of words. An example could be a weather-related topic where a relatively higher weighting of terms such as rain, temperature, or wind, in comparison to English text. Next, to generate a new document, a topic is determined for a new document, which is a relative weighting of the topics to be used for generation of this document, such as 30% sports and 70% Movies. Finally, this weighted mixture of topics is used to probabilistically generate the sequence of words comprising the new document; in this case, the model has gone through three distinct levels of probabilistic generation to

produce a document, this is purely a model of the text data. Visualizing that the text was generated according to this model does not yet inform the parameters of the model initially observing the text output of it. The purpose of defining this model is to use the observed text data to estimate the remaining unknown configuration of the model, i.e., the topics and the topic coverage. There exist different statistical inference approaches for estimating the model parameters, and the result is a set of *k* discovered topics shared among the text data, as well as the topic coverage of each document in the collection.

Libraries

Selection of LDA implementations.

Gensim

Gensim may be the most popular library used for LDA models; as a Python library which was initially released in 2009 by Radim Řehůřek as a hobby project but with the growth of Python in the space of machine learning and data science, the library has become a cornerstone of text analysis. Gensim includes the LDA functionality provided in MALLET, (discussed later in this text), which serves to bring the mature implementations into a more temporally relevant programming language, one of the major benefits of the library. Gensim is designed to process raw, unstructured digital texts “plain text” using unsupervised machine learning algorithms. The algorithms in Gensim, such as Word2Vec, FastText, Latent Semantic Indexing (LSI, LSA, LsiModel), Latent Dirichlet Allocation, automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised, and only need a corpus of plain text documents. Once these statistical patterns are found, any plain text documents can be concisely expressed in the new, semantic representation and queried for topical similarity against other documents.

Stanford Topic Modeling Toolbox

Stanford Topic Modeling Toolbox (TMT) is a resource developed by The Stanford Natural Language Processing Group. TMT "brings topic modeling tools to social scientists and others who wish to perform analysis on datasets that have a substantial textual component". TMT features the ability to import and manipulate texts, train topics models to create textual summaries, and generate compatible "outputs for tracking word usage across topics, time, and other groupings of data". TMT was written in 2009-2010 and uses an old version of Scala. The program is no longer being updated and The Stanford Natural Language Processing Group is no longer providing support for the users but "some people still use it and find it a friendly piece of software for LDA and Labeled LDA models". TMT is useful for running an ad hoc analysis of topics, as opposed to larger programmatic process that that use the model output. The toolbox takes text data stored in spreadsheets as input and then produces output in the form of txt and csv files. The software will preprocess the given text data according to the parameter choices, such as the tokenizer, which can be configured to prune non-alphanumeric strings, filter for a minimum length, prune out the top 30 most frequent words, prune out the top 5 least frequent words, and the number of desired topics. TMT also allows for the choice of LDA configurations with the main option being k, the number of topics to discover. Additionally, there is a choice between the numerical method of estimation: Collapsed Gibbs sampling (GibbsLDA) vs. Collapsed Variational Bayes approximation (CVB0LDA, the default). These are both implemented to take advantage of multi-threading and multi-

core machines. However, CVB0LDA requires fewer iterations as it has a faster rate of convergence, but GibbsLDA requires less memory during training. Finally, there are also two additional versions of LDA available: Labeled LDA and Partially Labeled LDA (PLDA).

GUI Topic Modelling Tool (GTMT)

The GUI Topic Modeling Tool (GTMT) is a graphical user interface for performing LDA, with its main strength running simple analysis. GTMT is built using the functionality of MALLET and is in other words simply a more user-friendly extension for less technical users. The output of running the estimation can be viewed in formatted HTML files, which provides for simple browsing and visualization of the topics. The Topic Modeling tool has three anchors:

Input anchor - Use the input anchor to connect the text data you want to analyze.
"D" anchor: Use the output anchor to pass the data you've analyzed downstream.
"R" anchor: Use the "R" anchor to view a report of the analysis.

The MALLET topic model package includes an extremely fast and highly scalable implementation of Gibbs sampling, efficient methods for document-topic hyperparameter optimization, and tools for inferring topics for new documents given trained models. Topic modeling can be easily compared to clustering, as in the number of topics, like the number of clusters, is a hyperparameter. By doing topic modeling you build clusters of words rather than clusters of texts. A text is thus a mixture of all the topics, each having a certain weight.

MALLET

MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text. MALLET includes sophisticated tools for document classification: efficient routines for converting text to features, a wide variety of algorithms (including Naïve Bayes, Maximum Entropy, and Decision Trees), and code for evaluating classifier performance using several commonly used metrics. In addition to classification, MALLET includes tools for sequence tagging for applications such as named-entity extraction from text. Algorithms include Hidden Markov Models, Maximum Entropy Markov Models, and Conditional Random Fields. These methods are implemented in an extensible system for finite state transducers. MALLET includes some built-in functionality for tokenization and text processing as well as numerical optimization, which are beneficial for topic modeling. MALLET was primarily written by Prof. Andrew McCallum, with the help of other graduate students, at the University of Massachusetts Amherst. For topic modeling MALLET offers Latent Dirichlet Allocation, Pachinko Allocation, and Hierarchical LDA. Estimation is performed using a performant multi-threaded implementation of Gibbs sampling. MALLET shared common features with other libraries configuring importing and processing of text data, as well as simple parameters such as the desired number of topics. Notably, MALLET allows for periodic optimization of the hyperparameters.

Conclusion

LDA is applied to text data decomposing the corpus document into a word matrix in two parts, the Document Topic Matrix, and the Topic Word, LDA is a matrix factorization technique. There exist different libraries, each with benefits and drawbacks observed in various implementations; summarized in this review are a few popular libraries of LDA. Of these, Gensim ranks among the highest as it is scalable and can easily process large volumes of corpora by using its incremental online training algorithms. It is scalable in nature, as there is no need for the whole input corpus to reside fully in Random Access Memory at any one time. All Gensim algorithms are memory-independent with respect

to the corpus size. Gensim is robust and has been in use in various systems and organizations for over a decade. Gensim is platform agnostic built using Python and runs on all the platforms that supports Python and Numpy. Additionally, efficient Multicore Implementations can speed up processing and retrieval on clusters, Gensim provides efficient multicore implementations of various popular algorithms like (LDA) and does not require costly annotations or hand tagging of documents because it uses unsupervised models. Gensim is licensed under the OSI-approved GNU LGPL license allowing for it to be used for both personal and commercial use for free. Any modifications made in Gensim are in turn open-sourced and has abundance of community support too.

References

"A Beginner's Guide to Latent Dirichlet Allocation (LDA)". https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2

Accessed 19 Oct. 2022.

"Topic Modeling and Latent Dirichlet Allocation (LDA) in Python". https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

Accessed 21 Oct. 2022.

"What is Gensim". https://radimrehurek.com/gensim/intro.html#what-is-gensim.

Accessed 01 Nov. 2022.

"Stanford Topic Modeling Toolbox". https://downloads.cs.stanford.edu/nlp/software/tmt/tmt-0.4/

Accessed 02 Nov. 2022.

"Topic Modeling Tool." Google Code, 22 Sept. 2011, code.google.com/archive/p/topic-modeling-tool/

Accessed 03 Nov. 2022.

"Mallet: MAchine Learning for LanguagE Toolkit". https://mimno.github.io/Mallet/

Accessed 04 Nov. 2022.