Tech Review: LDA for Topic Based Product Review Analysis

Benjamin Corn (bcorn2@illinois.edu)

Department of Computer Science: University of Illinois Urbana-Champaign

CS 410: Text Information Systems

Dr. ChengXiang Zhai

November 7, 2021

Overview

According to research performed by The Medill Spiegel Research Center (SRC) at Northwestern University, about 95% of customers read reviews before making a purchase [1] and 82% of customers actively seek out negative reviews when making a purchase decision [2]. For the product managers building products, they rely on customer feedback through product reviews to determine issues, new feature requests, and improvements to help inform priorities and roadmaps, but manually extracting meaningful insights from product and app reviews, on a recurring basis, can be time consuming, error prone, and lead to missing hidden insights, but these key insights can be the key to unlocking increased customer satisfaction and product growth. This paper reviews the unsupervised topic modeling approach of Latent Dirichlet Allocation (LDA) by providing an overview of the approach, popular implementations, and potential improvements over the standard LDA implementation. The review focuses on using LDA for the purposes of building and deploying applications to distill useful information from short text-based product reviews with a high number of written reviews (n > 1,000), into key topics (e.g., screen too small, too heavy, too slow) that product managers can use to discover hidden insights about their products. When evaluating these methods, the following criteria will be taken into consideration:

1. Product reviews are often short text (100-150 characters) [3].
2. Reviews can contain frequent grammar mistakes, slang, or specifications (e.g., model numbers, screen dimensions, resolution).
3. Deriving product review insights should not require manual tagging effort by a product manager.

Latent Dirichlet Allocation (LDA) Overview

Topic modeling is used to discover hidden topics in a set of documents. For the purposes of this review, the set of documents is assumed to be the set of written consumer reviews for a *single product* and the topics represent the *themes* found within reviews for the *single product*. Examples could include: “screen is too small”, “customer service is bad”, or “bad battery life”. It is also assumed that the set of reviews is too large for a product manager (or other stakeholder) to effectively discover themes, on a recurring basis.

LDA is a generative statistical model that utilizes an unsupervised learning algorithm to map documents to a fixed number (N) of topics (themes) by calculating the probability of words belonging to a topic. This is done by iterating over each document (d) and assigning each word (w) to one of (k) topics, (t) by calculating p(w | t) or p(t | d) * p(w | t)[5]. By iterating over each document and each word, the estimated probabilities will improve based on the information gained about the context of words. Once an LDA model has been generated and run on a document set, a set of topics is generated consisting of dominant keywords and keyword weights. Example output below:

- Topic 1: battery, short, charging, life, dying
- Topic 2: poor, quality, low, resolution, camera
- Topic 3: slow, update, fail, crashing
- Topic 4: installation, difficult, replacement, customer service

In general, the advantages of LDA include: providing good results for topic modeling of short-text reviews[14], is fast to run, and can predict the topics for new (unseen) documents. Disadvantages include the need to fine tune input documents to remove noise (e.g, technical words such as model numbers, resolutions, dimensions, etc.), performing multiple iterations to identify the ideal number of topics (k, a

hyperparameter to LDA), and the need for applying a secondary processing step to infer the topic from the topic keywords generated by LDA -- such as "summarizing" the topic vectors into an understandable label (e.g., for Topic 1 above, a label may be "poor battery life"). Improvements to LDA to address some of these disadvantages will be discussed below.

In terms of measuring the performance of a given topic model, two metrics are generally considered: model perplexity (statistical measure of how well a probability model predicts a sample) [6] and topic coherence (measure the degree of semantic similarity between high scoring keywords for a given topic) [7]. Lower perplexity measures given (k) number of topics generally represents a "better" model, however a model optimized for perplexity may not produce interpretable topics [7]. Topic Coherence attempts to solve this problem by measuring the "interpretability" of the generated model.

LDA Implementation Approach

There are several powerful libraries providing an optimized LDA implementation. Namely: Gensim and Sklearn. Regardless of the library used, the commonly accepted pre-processing steps of reviews (documents) include [11]:

1. Remove punctuation and apply lowercase to all words in a review
2. Remove stop words (e.g., the, to, a, is, are)
3. Lemmatization (converts word to meaningful base form)
4. Tokenization (divide the review into individual words)
5. Filter extremes based on number of occurrences
6. Generate bag of words or TF-IDF scores

From there, the steps are similar for Gensim and Sklearn: create the LDA model using the generated corpus from above steps and a word dictionary. Comparisons have been conducted between Gensim and Sklearn, however the results are not conclusive. Therefore, the specific advantages of each will be outlined.

- Gensim: primarily built for topic modeling applications and contains power pre-processing features such as punctuation, whitespaces, and numeric stripping, stopword removal, stemming, lemmatization, and bi-gram representations. Gensim also has out-of-the-box multi-core support for LDA parallelization that can improve performance on very large corpus jobs, and community support for CUDA GPU acceleration [8]. For model evaluation, Gensim provides a topic coherence pipeline [9] for coherence calculation. In general, Gensim provides a highly specialized API for topic modeling in Python with excellent community support and documentation.
- SKLearn: widely accepted library of well documented machine learning tools available in Python. SKLearn's LDA implementation provides fewer parameters, potentially offering a faster path to getting off the ground at the expense of fine tuning the LDA model. SKLearn has broad cloud support, such as simple end-to-end training and deployment of SKLearn models on Amazon SageMaker (AWS) and AzureML. In general, SKLearn may be easier to build and deploy an application with compared to Gensim, simply based on the widespread support of the toolkit. It should be noted that SKLearn does not provide as robust text pre-processing as Gensim, and does not support model coherence out-of-the-box.

Improving Topic Modeling Approach for Short-Text Reviews

As mentioned in a prior section, one of the pain points of LDA is generating a "theme" from the topic keyword vectors (e.g., battery, short, charging, life, dying). Ideally, this "extraction" is done in an automatic way vs. manually reviewing the vectors and coming up with an interpretation each time. To solve this problem, an approach outlined by Susan Li in a post titled '*Topic Modelling in Python with NLTK and Gensim'* [10] using Rapid Automatic Keyword Extraction (RAKE) to compare the output similarity of RAKE on a document title (in the case of product reviews, the review title or summary) to the keyword output of LDA. By doing so, an informative topic title can be produced (e.g., poor battery life) compared to the raw keyword vector output.

Another key pain point with LDA is determining the number of topics (k) for the LDA model. The process to identify (k) is iterative, selecting various (k) values and measuring model performance. For a scalable solution to be applied for product reviews of various subjects (e.g., electronics, furniture, home appliances), a better way to select (k) must be considered. Hierarchical Latent Dirichlet Allocation (hLDA) attempts to solve this problem by learning the correct (k) value rather than using one specified, however the accuracy of hLDA may take a hit in the process. [12]

Lastly, a supervised learning approach (i.e. text classification) using Logistic Regression Classification can be evaluated for the purpose of mapping reviews to topics for greater control over topic generation and accuracy of document to topic modeling, with the possibility of missing hidden topics due to the human bias in generating topics and tagging reviews with those topics. spaCy is a text classification library for Python that can enable building a text classification model based on a tagged set of reviews [13]. Further extending the text classification approach, a semi-supervised approach has been proposed of using the output topics of LDA as the inputs to a text classification model -- removing some of the human work required to tag documents and potentially discover hidden topics a human may not have.

Summary

LDA can be an effective mechanism for generating hidden topics from short-text product reviews for generating insights about product issues, new feature requests, and potential enhancements. Powerful libraries such as Gensim and SKLearn enable a relatively straightforward implementation of LDA, however generating useful results from the general implementation requires a "test and learn" approach using various (k) values as inputs, and providing LDA with a low-noise, well processed input dataset.

Topic keyword vectors generated by LDA may not be very helpful to product managers by themselves. Additionally processing must be done to extract the "theme" from the topic keywords, such as the RAKE approach or simply taking the highest weight topic. Providing product managers with the weighted topic keywords per topic and the frequency of those keywords in the reviews can help provide additional insights into what customers are saying in reviews.

To provide greater accuracy in mapping reviews to topics, a supervised learning approach should be considered. If the product has a large enough set of reviews and the product manager is able to tag the reviews, the outcome may produce more interesting and accurate insights about what customers value the most, find issues in the most, and are requesting the most. This approach requires the product manager to have existing insight into aspects of the product customers dislike, favor, or wish to have to generate the correct "labels" on each review.

References

[1] How Online Reviews Influence Sales. (2017). [E-book]. Spiegel Research Center.https://irp-cdn.multiscreensite.com/00a8b24b/files/uploaded/Spiegel_Online%20Review_eBook_Jun2017_Pv2.pdf

[2] *The Power of Reviews*. (2016). [E-book]. https://www.powerreviews.com/wp-content/uploads/2016/04/PowerofReviews_2016.pdf

[3] Woolf, M. (2014, June 17). *A Statistical Analysis of 1.2 Million Amazon Reviews*. Max Woolf's Blog. https://minimaxir.com/2014/06/reviewing-reviews/

[4] SNAP: Web data: Amazon reviews. (n.d.). Stanford Network Analysis Project. https://snap.stanford.edu/data/web-Amazon.html

[5] Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. *Frontiers in Artificial Intelligence*, *3*. https://doi.org/10.3389/frai.2020.00042

[6] Soltoff, B. (2021, September 1). *Topic modeling*. Computing for the Social Sciences. https://cfss.uchicago.edu/notes/topic-modeling/

[7] Kapadia, S. (2020, December 29). *Evaluate Topic Models: Latent Dirichlet Allocation (LDA)*. Medium. https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

[8] J. (n.d.). *GitHub - js1010/cusim: Superfast CUDA implementation of Word2Vec and Latent Dirichlet Allocation (LDA)*. GitHub. https://github.com/js1010/cusim

[9] *Gensim: topic modelling for humans*. (n.d.). Gensim. https://radimrehurek.com/gensim/models/coherencemodel.html

[10] *Gensim: topic modelling for humans*. (n.d.). Gensim. https://radimrehurek.com/gensim/parsing/preprocessing.html

[11] Li, S. (2018, July 6). *Topic Modelling in Python with NLTK and Gensim - Towards Data Science*. Medium. https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21

[12] *Inferring Label Hierarchies with hLDA*. (2018). Square Corner Blog. https://developer.squareup.com/blog/inferring-label-hierarchies-with-hlda/

[13] Navlani, A. (2020, April 21). *Tutorial: Text Classification in Python Using spaCy*. Dataquest. https://www.dataquest.io/blog/tutorial-text-classification-in-python-using-spacy/

[14] Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020b). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. *Frontiers in Artificial Intelligence*, *3*. https://doi.org/10.3389/frai.2020.00042