Extension of LDA

Topic modeling is a classic problem in [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval). Related models and techniques are, among others, [latent semantic indexing](https://en.wikipedia.org/wiki/Latent_semantic_indexing), [independent component analysis](https://en.wikipedia.org/wiki/Independent_component_analysis),[probabilistic latent semantic indexing](https://en.wikipedia.org/wiki/PLSI), [non-negative matrix factorization](https://en.wikipedia.org/wiki/Non-negative_matrix_factorization), and [Gamma-Poisson distribution](https://en.wikipedia.org/wiki/Gamma-Poisson_distribution).

The LDA model is highly modular and can therefore be easily extended. The main field of interest is modeling relations between topics. This is achieved by using another distribution on the simplex instead of the Dirichlet. The Correlated Topic Model[[9]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-9) follows this approach, inducing a correlation structure between topics by using the [logistic normal distribution](https://en.wikipedia.org/wiki/Logistic_normal_distribution) instead of the Dirichlet. Another extension is the hierarchical LDA (hLDA),[[10]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-10) where topics are joined together in a hierarchy by using the nested [Chinese restaurant process](https://en.wikipedia.org/wiki/Chinese_restaurant_process). LDA can also be extended to a corpus in which a document includes two types of information (e.g., words and names), as in the [LDA-dual model](https://en.wikipedia.org/w/index.php?title=LDA-dual_model&action=edit&redlink=1).[[11]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-11) Nonparametric extensions of LDA include the [Hierarchical Dirichlet process](https://en.wikipedia.org/wiki/Hierarchical_Dirichlet_process) mixture model, which allows the number of topics to be unbounded and learnt from data and the nested [Chinese Restaurant Process](https://en.wikipedia.org/wiki/Chinese_Restaurant_Process) which allows topics to be arranged in a hierarchy whose structure is learnt from data.

As noted earlier, PLSA is similar to LDA. The LDA model is essentially the Bayesian version of PLSA model. The Bayesian formulation tends to perform better on small datasets because Bayesian methods can avoid overfitting the data. For very large datasets, the results of the two models tend to converge. One difference is that PLSA uses a variable dto represent a document in the training set. So in PLSA, when presented with a document the model hasn't seen before, we fix \Pr(w \mid z)—the probability of words under topics—to be that learned from the training set and use the same EM algorithm to infer \Pr(z \mid d)—the topic distribution under d. Blei argues that this step is cheating because you are essentially refitting the model to the new data.

Variations on LDA have been used to automatically put natural images into categories, such as "bedroom" or "forest", by treating an image as a document, and small patches of the image as words;[[12]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-12) one of the variations is called [Spatial Latent Dirichlet Allocation](https://en.wikipedia.org/w/index.php?title=Spatial_Latent_Dirichlet_Allocation&action=edit&redlink=1).[[13]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-13)

Recently, LDA has been also applied to [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics) context.[[14]](https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#cite_note-14)

Source:PLDA: Parallel Latent Dirichlet Allocation for

Large-scale Applications

Yi Wang1, Hongjie Bai1, Matt Stanton2,

Wen-Yen Chen1, and Edward Y. Chang1

1 Google Beijing Research, Beijing, 100084, China

fwyi,hjbai,wenyen,edchangg@google.com

2 Computer Science, CMU, USA

mlstanto@cs.cmu.edu

LDA Performance Enhancement

The computation complexity of Gibbs sampling is K multiplied by the total number of

word occurrences in the training corpus. Prior work has explored multiple alternatives

for speeding up LDA, including both parallelizing LDA across multiple machines and

reducing the total amount of work required to build an LDA model. Relevant parallelization

efforts include:

– Nallapati and et al. [7] reported distributed computing of the VEM algorithm for

LDA [1].

– Newman and et al. [2] presented two synchronous methods, AD-LDA and HDLDA,

to perform distributed Gibbs sampling. AD-LDA is similar to distributed EM

[8] from a data-flow perspective; HD-LDA is theoretically equivalent to learning a

mixture of LDA models but suffers from high computation cost.

– Asuncion, Smyth and Welling [9] presented an asynchronous distributed Gibbs

sampling algorithm

Applications

LDA has been applied to extract topics from text documents.

For instance, Newman et al.[19] applied LDA to derive

400 topics such as “September 11 attacks”, “Harry Potter”,

“Basketball” and “Holidays” from a corpus of 330000

New York Times news articles and represent each news article

as a mixture of these topics. LDA has also been applied

for identification of topics in a number of different areas. For

instance, LDA has been used to find scientific topics from

abstracts of papers published in the proceedings of the national

academy of sciences [10] . McCallum et al. [18] have

proposed LDA to extract topics from social networks and

apply it to a collection of 250,000 Enron emails. A variation

on LDA has also been used by Steyvers et al. [22] to

analyze 160,000 abstracts from the “citeseer” computer science

collection. Recently, Zheng et al. [6] have applied LDA

to obtain various biological concepts from a protein related

corpus.

When facing massive data, namely Big Data, an efficient implementations over distributed system becomes urgent. There are already some implementation for LDA already proposed. Here we introduce two most adopted benchmark projects. One is Parallel Latent Dirichlet Allocation (PLDA) project developed by Google, which is based on Google’s MapReduce system. The other one is GraphLab, which focuses on distributed computation over graph. Researchers from AMPLab at Berkeley also implement LDA on GraphX, a Spark based abstraction for distributed graph computations. All these projects are using distributed computing techniques and are based on the computation resources from Amazon EC2.

Also, students from Prof. John Canny’s group implement LDA on GPU, which is another direction of research of making LDA faster. The technique used by Prof. John Canny’s group is focusing the parallel computing part.

With the help of distributed and parallel computing techniques, LDA can be applied to industry scale data and performs well in industrial applications.