CSE 250B - Project 3

Lund, Sigurd Stoeen
sigurlu@stud.ntnu.no

Wolff, Thomas
thomawo@stud.ntnu.no

This paper presents collapsed Gibbs sampling as a method of training latent Dirchlet allocation models for classifying documents.

Introduction

This paper describes a way to learn topics on a dataset. A dataset consists of many different documents that may belong to different categories. The goal is to be able to select meaningful and not redundant topics for the documents. A topic is described by the words in that topic with the highest probability.

The topics are learned with unsupervised learning using collapsed Gibbs sampling to train latent Dirchlet allocation (LDA) models.

Two datasets are used to test the implementation, a dataset of 400 documents called Classic400[1], and Reuters-21578[2] consisting of 19 043 news articles. Both will be tested with different hyperparameters to see which values separates the topics best and if the words in each topic gives meaning to a human.

The implementation is able to learn good topics, especially for the classic400 set. Classic400 gets best result with $K \leq 10$, but for Reuters-21578 the words describing the topics gives more meaning to a human with $K = 150$. Both datasets get best result with $\alpha = 1/K$ and $\beta = 200/V$.

Design and analysis of algorithms

First the learning problem is described, followed by a description of how Gibbs sampling is used to solve this problem.

Definition of the problem

Informally, the goal of learning in this problem is when presented with a collection of documents, decide which topics a document is about, and in some way describe what these topics are. Below follows a description of how the collection of documents are represented and some terminology. The next section describes how the documents are assumed to be generated and finally how collapsed Gibbs sampling is used to learn the model.

Representing the collection of documents

When presented with a collection of documents, there is a finite set of words used in this collection. This set is called the vocabulary of the collection of documents, and has size V. Steps are taken to reduce the size of the vocabulary. Words used only once or twice in the whole collection of documents are most likely to be misspellings, and hence they are omitted. Also, words that does not contribute to the understanding of what the documents are about, so called stop words, are not included in the vocabulary.

The collection of documents is represented as a vector w, containing all occurrences of words in all documents. The words need not occur in the order they do in the documents, as long as they occur in some known order.

The generative process assumed in LDA models

To be able to do unsupervised learning it is assumed that the documents follow a probabilistic process, and more specifically it is assumed here that each document follows a LDA model. Under the LDA model, it is assumed that each document is created using the following process[3]:

```
Given: Dirchlet distribution with parameter vector $\alpha$ of length $K$
Given: Dirchlet distribution with parameter vector $\beta$ of length $V$
For topic $1$ to $K$
    draw a multinomial with parameter vector $\phi_k$ according to $\beta$
for document number 1 to document number $M$
    draw a topic distribution, i.e a multinomial $\theta$ according to $\alpha$
    for each word in the document
        draw a topic $z$ according to $\theta$
        draw a word $w$ according to $\phi_z$
```

Where K is the number of topics, V is the size of the vocabulary, M is the number of documents.

In this manner, each document has a certain mixture of topics which follows that documents multinomial distribution θ. This distribution is drawn from the Dirchlet distribution with parameter vector α. Each word is created by first drawing a topic z according to that documents distribution over topics θ, followed by drawing a word from that topics multinomial distribution over words with parameter vector ϕ.

Learning the LDA model using collapsed Gibbs sampling

As mentioned, the goal of learning is to infer which topics a document is about, and to describe these topics in some way. In the light of the generative process described above, this is more formally to infer each documents θ vector and each topics distribution over words ϕ_z. Gibbs sampling will not infer these vectors directly, rather it will for each word w_i in every document in the collection, infer a topic z_i. This results in a vector z where z_i corresponds to the topic of word w_i in w. In this way, given a topic z for every word in every document in the collection, each document θ vector and each topic ϕ vector can be derived.

Gibbs sampling works by initially assigning some arbitrary topic z_i to word w_i for every word in w, and iteratively update the values for every z_i. When updating z_i it is assumed that the topic for every other position than i is correct, and a new value for z_i is drawn according to

$$p(z_i = j | z\prime, w) \propto \frac{q'_{jw_i} + \beta_{w_i}}{\sum_t q'_{jt} + \beta_t} \frac{n'_{mj} + \alpha_j}{\sum_k n'_{mk} + \alpha_k}$$

where $z\prime$ is z exept entry z_i. q'_{jw_i} is the number of times that word w_i occurs under the topic j in the whole collection of documents except the occurrence at position i, n'_{mj} is the number of times topic j occurs in document m except the occurrence at position i. The α and β terms are values from the parameter vectors of the two Dirchlet distributions that is used in the generative process. The summations in the denominators are the same expressions as in their respective numerators, only summed over all words and topics.

It can be shown that the distribution of topics for all words in the whole collection of documents converges to the correct distribution when doing an adequate number of iterations. That means that each z_i value does not converge to a specific value, but the distribution of z values over all words converges to the correct distribution. After Gibbs sampling is done, each word occurrence in the whole collection of documents has a topic assigned to it, and this topic is drawn from the distribution that the words in the documents is assumed to follow when generated. From these z-values the distribution over topics for each document can be inferred according to

$$\theta_{mz} = \frac{n_{mz}}{N_m}$$

I.e the probability of topic z within document m is how many times does topic z occur in document m out of how many words there are in document m, namely N_m. The distribution over words for each topic can be inferred according to

$$\phi_{zw_i} = \frac{q_{zw_i}}{\sum_t q_{zt}}$$

I.e the probability of word w_i in topic z is how many times does word w_i occur within the topic z out of all the times topic z occurs in the whole collection of documents.

Hyperparameters

α is a vector of length K (number of topics), and β is a vector of length V (size of vocabulary). Both are taken as input to the learning algorithm. β is a pseudocount of words that means that a word w_j occurs at least β_j times in each topic. This ensures that the probability for a word in a topic is never zero.

α_j describes how many pseudowords a document m_j contains. α_j is added initially to the word count of each document and topic.

Both vectors could be initialized with different values at each element, but that would require some automatically optimizing of the values, and are out of the scope of this paper. All β values is defined as a value divided by K, and all α values are divided by V. Both values should be less than 1.

Convergence criteria

The section ”Learning the LDA model using collapsed Gibbs sampling” stated that the distribution of topics over all words in the collection of documents converges if Gibbs sampling is done an adequate number of iterations. This section describes the convergence criteria used in this paper.

Since the distribution of topics over all words in the collection of documents converges, it is natural to use some sort of measure of change in the distribution between iterations, and stop Gibbs sampling when this change is below some threshold. The measure of change in the distribution used in this paper is the sum over squared differences between the probability for all words in the vocabulary in all topics in the previous iteration and the current iteration.

$$\sum_k \sum_t ||\phi_{kt(i-1)} - \phi_{kti}||^2$$

where ϕ_{kti} is the probability of word t in topic k after iteration i. A threshold of 5.0^{-6} is used.

Experiments

Design of experiments

The first dataset the implementation is tested on is called Classic400, containing 400 documents where each document has a category 1-3. The categories are not used for learning the topic model, but will be used for measuring the results. The data is taken from Elkan, 2006[4].

The second dataset is Reuters-21578 which contains 19 043 news articles from 672 categories. The vocabulary is build by excluding words that are of three letters or less, and digits are skipped. Stemming is also applied, resulting in a vocabulary of 10 227 words. The true categories is not used for measuring here.

Both datasets will be tested with different K, α and β values to see how these parameters affect the results.

The algorithm is runned until the difference in the z distribution is less than 5.0^-6 or a maximum number of epochs has ran. This value is set by watching when the difference in the distribution stops to drop. The maximum number of epochs is set to 200, since it usually does not drop much after that many iterations.

The results will be plotted in 3D space. For $K > 3$, a Principal Component Analysis (PCA) is used to reduce the dimensions to fit 3D space[5].

The result of the different values on α, β and K is primarily evaluated by a human by looking at the 3D plot how well the different topics are separated. It is also evaluated by looking at the 10 words with highest probability in each topic to see if the words makes sense for a human.

Results of Experiments

Classic400

As shown in Figures 1 - 5 this method is able to learn topics that divide the documents into clusters with their true labels. Each color belongs to a true category. In Figures 1 - 3 some changes due to the different α and β values can be seen. With lower α values the points lie more to the borders, than with a higher value.

Topic 0	Topic 1	Topic 2
system	patients	boundary
problems	ventricular	layer
research	fatty	wing
scientific	nickel	mach
retrieval	left	supersonic
development	cases	ratio
methods	acids	wings
language	aortic	velocity
field	blood	effects
science	normal	shock

Table 1: Classic400 - Most likely words in all topics - $K = 3$

For greater sizes of K, $K = 10$ is making more distinction between the categories than $K = 4$.

The words with highest probability in each topic makes sense to a human when $K = 3$, Table 1. For $K = 150$ the semantics of the words are not as easy to interpret, and it is probably because the learner is trying to categorize the text in a lot more topics than it really consists of since it is only 3 categories, Table 2.

Reuters-21578

It is easy for this dataset aswell to spot the different topics in the plot when $K = 3$, Figures 6 - 8, allthough here it is much more documents between the centroid of each topic. It is hard for a

Topic 1
airplane
loading
effects
calculating
outlined
external
load
planform
arbitrary
stiffness

Table 2: Classic400 - Most likely words in topic 1 - $K = 150$

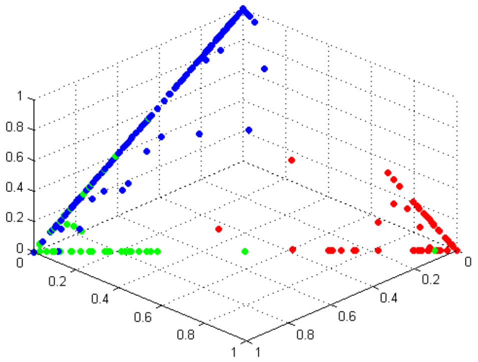

Figure 1: Classic400, $K = 3$, $\alpha = 0.05/K$, $\beta = 200/V$

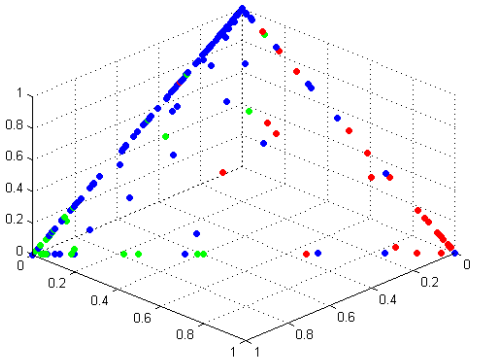

Figure 2: Classic400, $K = 3$, $\alpha = 0.05/K$, $\beta = 2000/V$

human to interpret the semantics of the words with highest probability, table 3.

As we can see in Topic 82 in Table 4, the words can make sense when $K = 150$, but that is not always the case as in Topic 102 where it is not as easy to interpret what the topic is.

Findings and lessons learned

Since no automatic method is used for testing with different α, β and K parameters, only a few combinations of these values are experimented with. The best results achieved was with $\alpha = 1/K$, $\beta = 200/V$ for both datasets. The topics for the Classic400 set is more separated, and that is natural because of the few documents and categories there. $K = 3$ works great for both datasets however, if we just look at the separation of the topics. For Reuters-21578

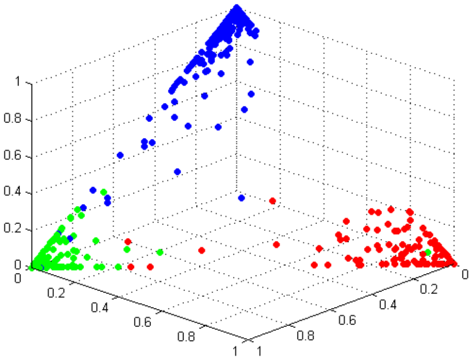

Figure 3: Classic400, $K = 3$, $\alpha = 1/K$, $\beta = 200/V$

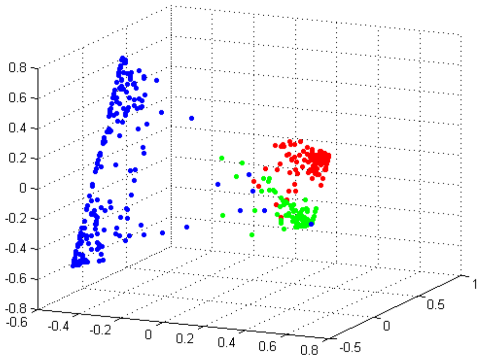

Figure 4: Classic400, $K = 4$, $\alpha = 1/K$, $\beta = 200/V$

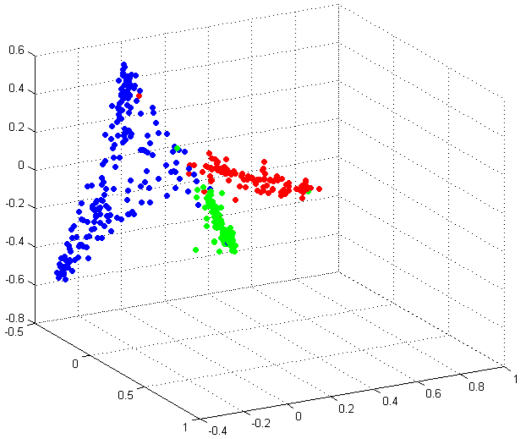

Figure 5: Classic400, $K = 10$, $\alpha = 1/K$, $\beta = 200/V$

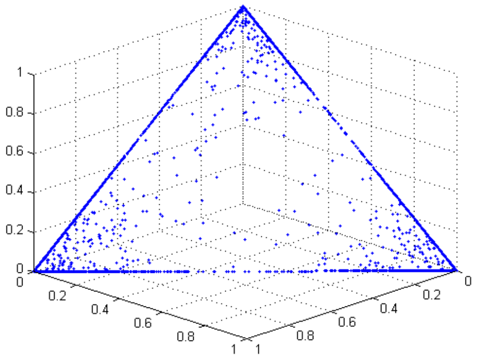

Figure 6: Reuters-21578, $K = 3$, $\alpha = 0.05/K$, $\beta = 200/V$

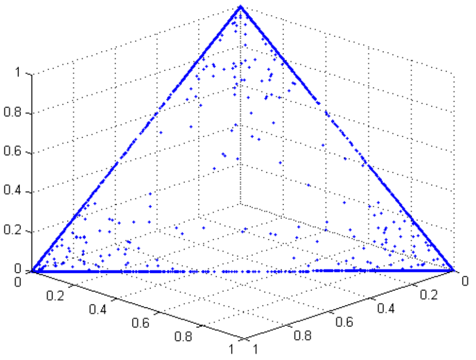

Figure 7: Reuters-21578, $K = 3$, $\alpha = 0.05/K$, $\beta = 2000/V$

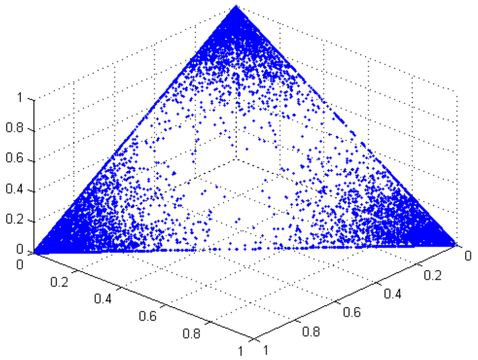

Figure 8: Reuters-21578, $K = 3$, $\alpha = 1/K$, $\beta = 200/V$

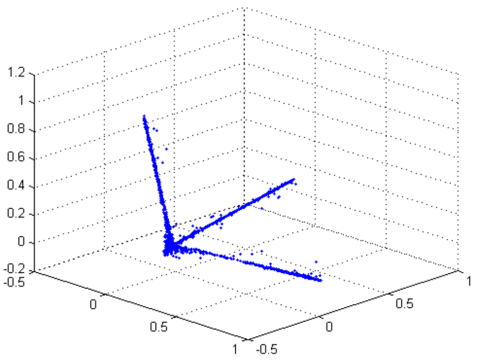

Figure 9: Reuters-21578, $K = 150$, $\alpha = 1/K$, $\beta = 200/V$

is it hard for a human to interpret the semantics in the words in a topic when $K = 3$. With $K = 150$ is it possible to find some topics that makes sense, but a lot of them are still very diffuse. Reuters-21578 consists of far more documents and categories than Classic400, and is definitely harder to find good topics for than Classic400.

How well the different topics are separated, could be computed numerically by summing the distance between each document from different topics. The best fit would be the one with the highest distance. This will give an exact value that describes the separation of each topic. It is more difficult to compute how the words in the topics makes sense to a human, and this will be approximately.

LDA can overfit the data by learning a model that is not general enough and it seems that this happens when K is big. By observing the difference from epoch to epoch over the distribution of z values, it follows that the model does not converge very well when $K = 150$. The difference

Topic 0	Topic 1	Topic 2
writedown	rada	roast
hawkey	citi	attorney
ongo	harm	wealthi
coupon	rest	sign
former	alfr	raton
ldbrinkman	signal	ratio
quit	housew	angra
nynex	danah	dilut
benjamin	jama	palac
hitest	redman	nomine

Table 3: Reuters-21578 - Most likely words in all topics - $K = 3$

Topic 82	Topic 102
dlrs	paralys
reuter	tonight
compani	winston
billion	angri
share	soda
nine	bonn
corp	rozich
sale	galleri
market	greenwich

Table 4: Reuters-21578 - Most likely words in topics 82 and 102 - $K = 150$

between each epoch is higher, and the reason for that could be that the model it is learning is too specialized, and the distribution varies more.

References

1. Classic400, http://cseweb.ucsd.edu/users/elkan/151/classic400.mat

2. Reuters-21578, Distribution 1.0, http://www.daviddlewis.com/resources/testcollections/reuters21578/

3. Charles Elkan, 2014, Text mining and topic models, http://cseweb.ucsd.edu/ elkan/250B/topicmodels.pdf

4. Charles Elkan, 2006, Clustering Documents with an Exponential-Family Approximation of the Dirichlet Compound Multinomial Distribution,

http://machinelearning.wustl.edu/mlpapers/paper_files/icml2006_Elkan06.pdf

5. Zito, T., Wilbert, N., Wiskott, L., Berkes, P. (2009). Modular toolkit for Data Processing (MDP): a Python data processing frame work, Front. Neuroinform. (2008) 2:8. doi:10.3389/neuro.11.008.2008