Topic modeling bla bla

May 30, 2017

Contents

1 Introduction 1

2 Theoretic background 1

2.1 The Multinomial Distribution 2

2.2 Generative process 3

2.3 Basic models in topic modeling 5

2.3.1 Latent Semantic Indexing (LSI) 5

2.3.2 Probabilistic Latent Semantic Indexing (PLSI) 6

2.3.3 Latent Dirichlet Allocation (LDA) 7

2.3.3.1 Dirichlet distribution 8

2.3.3.2 Model 9

2.3.3.3 Expectation Maximization 9

2.3.3.4 Variational Inference 9

3 Related work 9

4 Methodology 9

5 Analysis and results 9

6 Conclusion 9

7 Future work 9

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

1 Introduction

This is where you tell people why they should bother reading your article.

2 Theoretic background

In this chapter we will present some basic theoretic background about topic modeling. To deal with extensive text body, machine learning researchers have developed topic modeling: a suite of methods that analyze the words of the original text to discover the themes that run through the data and detect hidden semantic structures by analyzing a corpus. By marking up the documents using these topics and by using the resulting structure, the model can be used for organization, finding similar documents, or any other similar tasks. A topic is here defined as a probability distribution of words, meaning that certain words are more likely within a certain topic. Since the topics emerge from the analysis of the original texts, topic modeling algorithms do not require any prior annotations or labeling of the documents.

In order to apply a topic model to our data, the first question we must answer is how to represent documents. For understanding of natural language one must obviously preserve the order of the words in documents. However, for many large-scale data mining tasks, it is sufficient to use a simple representation that loses all information about word order. Given a collection of documents, the first task to perform is to identify the set of all words used at least once in at least one document. This set is called the vocabulary. Often, we reduce the size of the vocabulary by keeping only words that are used in at least a small percentage of the documents. Words that are found only once are often misspellings or other mistakes. Although the vocabulary is a set, we fix an arbitrary ordering for it so we can refer to word 1 through word m where m is the size of the vocabulary. Once the vocabulary has been fixed, each document is represented as a vector with integer entries of length m. If this vector is x then its j-th component x_j is the number of appearances of word j in the document. The length of the document is $n = \sum\limits_{j=1}^{m} x^j$.

Many applications of if topic modeling also eliminate from the vocabulary so-called stop words. These are words that are common in most documents and do not correspond to any particular subject matter. They include pronouns (you, he, it), connectives (and, because, however), prepositions (to, of, before), and auxiliaries (have, been, can, should). Stop words may also include generic nouns (amount, part, nothing) and verbs (do, have). A collection of documents is represented as a two-dimensional matrix where each row describes a document and each column corresponds to a word. Each entry in this matrix is an integer count; most entries are zero. It makes sense to view each column as a feature.

Some further common concepts and terms in topic modeling:

- A word is defined as an item from a vocabulary indexed from $1, ..., V$, where V is the size of the vocabulary. All words are represented as unit-basis vectors with one component equal to one and the rest equal to zero.

- A document is a collection of words denoted by $w = w_1, ..., w_N$, where w_n is the nth word and N is the total number of words in the collection.

- A corpus is a collection of documents in a dataset. It is denoted $D = w_1, ..., w_M$, where w_m is the mth document in the corpus and M is the total number of documents.

- Latent variables are variables that may not be directly observed, unlike observable variables. Latent variables can instead be inferred from other observable variables.

- Polysemy is the capacity for a word to have multiple related meanings. An example of this is the word plant which can mean a living organism of the kind exemplified by trees, herbs etc., or a place where an industrial or manufacturing process takes place.

- Synonymy is the capacity for several words to have similar meanings such as the words buy and purchase.

2.1 The Multinomial Distribution

Once we have a representation for individual documents, the natural next step is to select a model for a set of documents. It is important to understand the difference between a representation and a model. A representation is a way of encoding an entity as a data structure. A model is an abstraction of a set of entities, for example a probability distribution. Given a training set of documents, we will choose values for the parameters of a probabilistic model that make the training documents have high probability. Then, given a test document, we can evaluate its probability according to the model. The higher this probability is, the more similar the test document is to the training set.

The probability distribution that we use is the multinomial. Mathematically, this distribution is:

$$p(x;\theta) = \left(\frac{n!}{\prod_{j=1}^{m} x^j!} \right) \left(\prod_{j=1}^{m} \theta_j^{x_j} \right) \tag{1}$$

where the data x are a vector of non-negative integers and the parameters θ are a real-valued vector. Both vectors have the same length m. Intuitively, θ_j is the probability of word j while x_j is the count of word j. Each time word j appears in the document it contributes an amount θ_j to the total probability, hence the term $\theta_j^{x_j}$. The components of θ are non-negative and have unit $\sum_{j=1}^{m} \theta_j = 1$. A vector with these properties is called a unit vector.

Like any discrete distribution, a multinomial has to sum to one, where the sum is over all possible data points. Here, a data point is a document containing n words.

2.2 Generative process

Suppose that we have a collection of documents, and we want to find an organization for these, i.e. we want to do unsupervised learning. A common way to do unsupervised learning is to assume that the data were generated by some probabilistic process, and then to infer the parameters of this process. The generative process is a specification of a parametrized family of distributions. Learning is based on the principle of maximum likelihood, or some refinement of this principle such as maximum a posterior probability.

```
1. Fix a multinomial distribution with parameter vector $\phi$ of length $V$
2. for each word in the document:
     draw a word $w$ according to $\phi$
```

Above, step 1 sets up the probability distributions that are then used in step 2 to produce the observed training data. A single multinomial distribution can only represent a category of closely related documents. For a collection of documents of multiple categories/topics, a simple generative process is

```
1. Fix a multinomial $\alpha$ over categories 1 to $K$
   for category number 1 to category number $K$:
     fix a multinomial with parameter vector $\phi_k$
2. for document number 1 to document number $M$:
     draw a category $z$ according to $\alpha$
     for each word in the document:
         draw a word $w$ according to $\phi_z$
```

Note that z is an integer between 1 and K. For each document, the value of z is hidden, meaning that it exists conceptually, but it is never known, not even for training data. The generative process above gives the following global probability distribution

$$f(x) = \sum_{k=1}^{K} \alpha_k f(x; \phi_k) \tag{2}$$

where x is a document and ϕ_k is the parameter vector of the k-th multinomial. In general, x could be a data point of any type and ϕ_k could be the parameters of any appropriate distribution. K is called the number of components in the mixture model. For each k, $f(x; \theta_k)$ is the distribution of component number k. The scalar α_k is the proportion of component number k. A distribution like this is called a mixture distribution. In general, the components can be probability density functions, for example Gaussians, or probability mass functions, for example multinomials. We will see later how to do maximum likelihood estimation for mixture distributions, using a technique called expectation-maximization.

2.3 Basic models in topic modeling

First, well introduce some basic models for topic modeling. The Latent Semantic Indexing, or LSI, was presented by Deerwester et al. in 1990. The model manages to deal with the problem that multiple terms can refer to the same meaning, i.e. synonymy. However, it is not as successful regarding polysemy. The reason is that every term is represented as just one point in the so-called latent semantic space. Furthermore, a word that can mean two or more different things is represented as a weighted average of the different meanings.

After LSI was introduced, Hofmann presented the Probabilistic Latent Semantic Indexing, or PLSI model. PLSI is a so-called topic model where each word in a document is generated from a single topic which results in that each document in a corpus can be represented with a topic distribution. Later, in 2003 Blei et al. presented the Latent Dirichlet Allocation, or LDA. As opposed to PLSI, LDA is a statistically generative model for documents where each word in a document can be generated by all topics.

2.3.1 Latent Semantic Indexing (LSI)

Latent Semantic Indexing was one of the earliest methods for finding relationships between documents and the words that occur in them. In LSI a term-document matrix is created by analyzing the corpus, where the rows correspond to words and columns to documents. Each element in this sparse matrix describes the number of times a word occurs in a document but this term count can also be weighted with for instance TF-IDF. Letting each word represent a dimension in a very high dimensional space, a document can be seen as a vector with components corresponding to its weighted term counts.

A low-rank approximation of the term-document matrix is created using Single Value Decomposition, SVD, which creates new dimensions, called concepts, as linear combinations of the original words. This allows similarity measures and clustering methods by reducing the volume of the word space and thus making this space more densely populated.

The drawbacks with LSI are mainly the lack of a statistical foundation in the model. LSI is based on linear algebra instead of probabilistic modeling leaving it with a small toolbox for what can be achieved using the model.

2.3.2 Probabilistic Latent Semantic Indexing (PLSI)

Probabilistic Latent Semantic Indexing evolved from LSI and uses the same concept of finding a lower rank approximation of the term-document occurrence matrix. The difference is that instead of being based on linear algebra, PLSI is based on a mixture decomposition using a latent class model. PLSI associates an unobserved class variable z with each document-word observation pair (w, w). This z can be seen as a topic, since it is a probability distribution over words. As a generative model, PLSI can be defined in the following way for a corpus D:

1. Pick a document w with probability $p(w)$
2. For each of the N words in w:
    (a) Pick a topic z with probability $p(z|w)$
    (b) Generate a word w with probability $p(w|z)$

The result from PLSI is that every document is represented as mixing proportions for the topics, given by $p(z|w)$. Even though PLSI is generative for the analyzed corpus, it is not generative for new documents, which means is that there is no clear way of assigning probability to a document that is not part of the training data. Another problem is that the number of parameters in the model grows linearly with the number of documents.

2.3.3 Latent Dirichlet Allocation (LDA)

LDA is a topic model that was first presented as a graphical model for topic discovery by David Blei, Andrew Ng, and Michael I. Jordan in 2003. The basic idea is that documents exhibit multiple topics. LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's occurrence is attributable to one of the document's topics. This is similar to PLSI, except that in LDA the topic distribution is assumed to have a Dirichlet prior (see 2.x.x). In practice, this results in more reasonable mixtures of topics in a document. The reduced document description result in PLSI is a vector where each element describes the mixing proportion for a topic. A limitation with PLSI is that there is no generative (see 2.x) model for these proportions, making it difficult to handle unseen documents. The Latent Dirichlet Allocation model tries to solve this limitation by setting a Dirichlet prior on the topic distribution.

The basic idea in LDA is that we define a topic to be a Dirichlet distribution over a fixed vocabulary. Technically, the model assumes that the topics are generated first, before the documents. Now for each document in the collection, we generate the words in a two-stage process:

1. Randomly choose a distribution over topics.
2. For each word in the document:
   (a) Randomly choose a topic from the distribution over topics in step 1.

(b) Randomly choose a word from the corresponding distribution over the vocabulary.

This statistical model reflects the intuition that documents exhibit multiple topics. Each document exhibits the topics with different proportion (step 1); each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a). All the documents in the corpus share the same set of topics, but each document exhibits those topics with different proportion - thats the distinguishing characteristic of LDA. A more detailed explanation of the model follows in section 2.x.x.

2.3.3.1 Dirichlet distribution The Dirichlet distribution is a way to model random probability mass functions (PMFs)[1] for finite sets. It is also sometimes used as a prior in Bayesian statistics. The Dirichlet is the multivariate generalization of the beta distribution. It is an extension of the beta distribution for modeling probabilities for two or more events; when the result of the event has only 2 values, the Dirichlet distribution is equal to the beta distribution.

The Dirichlet distribution is a prior for the multinomial distribution. This means that if the prior distribution of the multinomial parameters is Dirichlet then the posterior distribution is also a Dirichlet distribution (with parameters different from those of the prior).
The Dirichlet process is a way to model randomness of a probability mass function (PMF) with unlimited options (e.g. an unlimited amount of dice in a bag). The process is similar Polyas Urn, only instead of having a set number of ball colors you have an unlimited amount:

[1] If you roll 1000 dice, the theoretical odds of any particular number showing up (i.e. a 1, 2, 3, 4, 5, or 6) are 1/6. However, you wont get that exact distribution in a real experiment due to manufacturing defects. If you have ten dice, each die will have its own probability mass function (PMF).

2.3.3.2 Model In LDA, each document is represented by a mixture of the topics, with weight 2w for topic z in document w. These weight have a Dirichlet prior, parameterized by a hyper-parameter . Each topic is a probability distribution over the vocabulary, with probability wz for word w in topic z.

2.3.3.3 Expectation Maximization bla

2.3.3.4 Variational Inference bla bla

3 Related work

4 Methodology

5 Analysis and results

6 Conclusion

7 Future work