

Chapter 2

A Few Words on Topic Modeling

Topic modeling techniques are used to discover the topics for (unlabeled) texts. As such, they are considered as unsupervised learning techniques which try to learn the patterns inside the text by considering words as observations. In this context, latent Dirichlet allocation (LDA) is a Bayesian topic modeling approach which has useful properties particularly for practical applications (Blei et al. 2003). In this section, we go through LDA by first reviewing the Dirichlet distribution, which is the basic distribution used in LDA.

2.1 Dirichlet Distribution

Dirichlet distribution is the *conjugate* prior for multinomial distribution likelihood (Kotz et al. 2000; Balakrishnan and Nevzorov 2003; Fox 2009). Specifically, the conjugate prior of a distribution has the property that after updating the prior, the posterior also has the same functional form as the prior (Hazewinkel 2002; Robert and Casella 2005). It has been shown that conjugate priors are found only inside the exponential families (Brown 1986).

2.1.1 Exponential Distributions

The density function of exponential distributions has a factor called *sufficient statistic*. The sufficient statistic is the sufficient function of the sample data (as reflected by its name) such that no other statistic that can be calculated from the sample

data provides any additional information than the sufficient statistic (Fisher 1922; Hazewinkel 2002). For instance, the maximum likelihood estimator in exponential families depends on the sufficient statistic but not all of observations.

The exponential families have the property that the dimension of sufficient statistic is bounded even if the size of observations goes to infinity, except a few member of exponential families such as uniform distribution. Moreover, the important property of exponential families is inside the theorems independently proved by Pitman (1936), Koopman (1936), and Darmois (1935) approximately at the same time. This property leads to efficient parameter estimation methods in exponential families. Examples of exponential families are the normal, Gamma, Poisson, multinomial, and Dirichlet distributions. In particular, the Dirichlet distribution is the *conjugate* prior for the multinomial distribution likelihood.

2.1.2 Multinomial Distribution

For the multinomial distribution, consider the trial of n events with observations $\mathbf{y} = (y_1, \dots, y_n)$ and the parameters $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$ where the observation of each event can take K possible values. For instance, in events of rolling a fair die n times, each observation y_i can take $K = 6$ values with equal probabilities ($\pi_1 = \frac{1}{6}, \dots, \pi_k = \frac{1}{6}$). Under such condition, this experiment is governed by a multinomial distribution. Formally, for the probability of having an observation $\mathbf{y} = (y_1, \dots, y_n)$ given the parameters $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$ we have:

$$p(\mathbf{y}|\boldsymbol{\pi}) = \frac{n!}{\prod_{i=1}^K n_i!} \prod_{i=1}^K \pi_i^{n_i},$$

where

$$n_i = \sum_{j=1}^n \delta(y_j, i)$$

in which $\delta(x, y)$ is the Kronecker delta function; $\delta(x, y) = 1$ if $x = y$, and zero otherwise. Moreover, it can be shown that in multinomial distribution, the expectation of number of times that the value i is observed over n trials is:

$$E(Y_i) = n\pi_i$$

and its variance is:

$$\text{Var}(Y_i) = n\pi_i(1 - \pi_i).$$

2.1.3 Dirichlet Distribution

For the conjugate prior of the likelihood of multinomial distribution, i.e., $p(\boldsymbol{\pi}|\mathbf{y})$, assume that the prior $p(\boldsymbol{\pi} = (\pi_1, \dots, \pi_k))$ is drawn from Dirichlet distribution with the hyper parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$ then the posterior $p(\boldsymbol{\pi}|\mathbf{y})$ is also drawn from Dirichlet distribution with the hyper parameters $(\alpha_1 + n_1, \dots, \alpha_k + n_k)$. Recall that n_i is the number of times the value i has been observed in the last trial, where $1 \leq i \leq K$.

This is the useful property of Dirichlet distribution which says that for updating the prior to get the posterior it suffices only to update the hyper parameters. That is, having a Dirichlet prior with the hyper parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$, after observing observations $(n_1, \dots, n_k)$ the posterior hyper parameters become $(\alpha_1 + n_1, \dots, \alpha_k + n_k)$. This property is discussed in the illustrative example further in this section.

Then, Dirichlet distribution for the parameter $\boldsymbol{\pi}$ with hyper parameter $\boldsymbol{\alpha}$ would be:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_i(\alpha_i))}{\prod_i(\Gamma(\alpha_i))} \prod_{i=1}^K \pi_i^{\alpha_i-1},$$

where $\Gamma(x)$ is the standard Gamma function. Note that Gamma function is an extension of factorial function. That is, for positive numbers Gamma function is the factorial function, i.e., $\Gamma(n) = n!$. Moreover, it can be shown that the expectation of Dirichlet prior $\boldsymbol{\pi}$ is:

$$E(\pi_i) = \frac{\alpha_i}{s} \quad (2.1)$$

and its variance is:

$$\text{Var}(\pi_i) = \frac{E(\pi_i)(1 - E(\pi_i))}{s + 1},$$

where $s = \alpha_1 + \dots + \alpha_k$ and is called the *concentration parameter*. The concentration parameter controls how concentrated the distribution is around its expected value (Sudderth 2006). The higher s is, the lower is the variance of the parameters. Moreover, given the concentration parameter s , the higher the hyper α_i is, the higher the expected value of π_i is. Therefore, the Dirichlet hyper parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_i)$ operate as a confidence measure.

Figure 2.1 plots 3 Dirichlet distributions with 3 values for s in three unit simplex (with 3 vertices). Note that $p(\boldsymbol{\pi})$ is a point in each simplex and $0 \leq \pi_i$, and $\sum_i^K \pi_i = 1$. Figure 2.1 shows that the higher the s is, the more concentration is around its expected value. In addition, the simplex in the middle has a high s whereas the one in the right has a lower s .

Neapolitan (2004) proved the useful property for the posterior of Dirichlet distribution. Suppose we are about to repeatedly perform an experiment with k

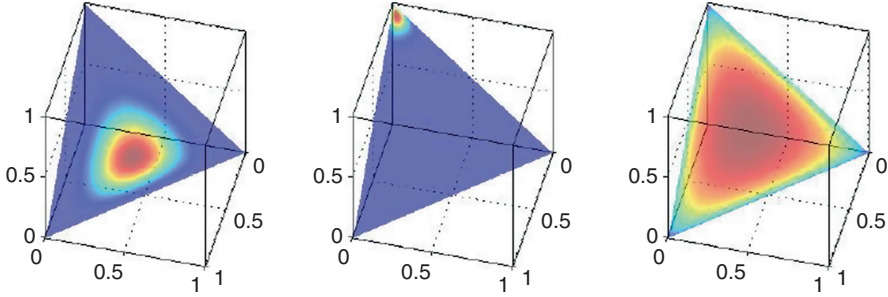


Fig. 2.1 The Dirichlet distribution for different values of the concentration parameter, taken from Huang (2005)

outcomes $x_1, x_2, \dots, x_k$. We assume exchangeable observations and present our prior belief concerning the probability of heads using a Dirichlet distribution with the parameters $\alpha = (\alpha_1, \dots, \alpha_k)$. Then, our prior probabilities become:

$$p(x_1) = \frac{\alpha_1}{m} \dots p(x_k) = \frac{\alpha_k}{m},$$

where $m = \alpha_1 + \dots + \alpha_k$.

After observing $x_1, \dots, x_k$ occurs, respectively, $n_1, \dots, n_k$ times in n trials where $n = n_1 + \dots + n_k$. Then, our posterior probabilities become as follows:

$$\begin{aligned} p(x_1|n_1, \dots, n_k) &= \frac{\alpha_1 + n_1}{s = m + n} \\ &\dots \\ p(x_k|n_1, \dots, n_k) &= \frac{\alpha_k + n_k}{s = m + n}. \end{aligned} \tag{2.2}$$

2.1.4 Example on the Dirichlet Distribution

Here, we present an illustrative example for the Dirichlet distribution, taken from Neapolitan (2009). Suppose we have an asymmetrical, six-sided die, and we have little idea of the probability of each side coming up. However, it seems that all sides are equally likely. So, we assign equal initial confidence about observing each number 1–6 appear by the die on the Dirichlet hyper parameters $\alpha = (\alpha_1, \dots, \alpha_k)$ as follows:

$$\alpha_1 = \alpha_2, \dots, \alpha_6 = 3.$$

Table 2.1 Dirichlet distribution example: the results of throwing a die 100 times

Outcome (x_i)	Number of occurrences (n_i)
1	10
2	15
3	5
4	30
5	13
6	27
n	100

Table 2.2 Dirichlet distribution example: the updated posterior probabilities

$p(1 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+10}{18+100} = 0.110$
$p(2 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+15}{18+100} = 0.153$
$p(3 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+5}{18+100} = 0.067$
$p(4 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+30}{18+100} = 0.280$
$p(5 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+13}{18+100} = 0.136$
$p(6 10, 15, 5, 30, 13, 27) = \frac{\alpha_1 + n_1}{s} = \frac{3+27}{18+100} = 0.254$

Table 2.3 Dirichlet distribution example: the updated hyper parameters

$\alpha_1 = \alpha_1 + n_1 = 3 + 10 = 13$
$\alpha_2 = \alpha_2 + n_2 = 3 + 15 = 18$
$\alpha_3 = \alpha_3 + n_3 = 3 + 5 = 8$
$\alpha_4 = \alpha_4 + n_4 = 3 + 30 = 33$
$\alpha_5 = \alpha_5 + n_5 = 3 + 13 = 16$
$\alpha_6 = \alpha_6 + n_6 = 3 + 27 = 30$

Then, we have $s = 3 \times 6 = 18$, and the prior probabilities are as follows:

$$p(1) = p(2) = \dots = p(6) = \frac{\alpha_i}{s} = \frac{3}{18} = 0.16667.$$

Next, suppose that we throw the die 100 times, with the following results shown in Table 2.1.

Using Eq. (2.2), the posterior probabilities can be updated as shown in Table 2.2.

Note in the example that the new value for the concentration parameter becomes $s = m + n$, where $m = 18$ ($\alpha_1 + \dots + \alpha_k$), and $n = 100$ (the number of observations). Moreover, the new values of hyper parameters become as shown in Table 2.3.

Using Eq. (2.1), $E(\pi_i) = \alpha_i/s$, the expected value of the parameters can be calculated as shown in Table 2.4.

Comparing the values in Table 2.4 to the ones in Table 2.2, we can see another important property of the Dirichlet distribution. That is, the number of observations directly reveals the confidence on the expected value of parameters.

In this section, we observed the Dirichlet distribution's useful properties:

1. The Dirichlet distribution is the conjugate prior for likelihood of multinomial distribution,

Table 2.4 Dirichlet distribution example: the expected value of hyper parameters

$E(\pi_1) = \alpha_1/s = 13/118 = 0.110$
$E(\pi_2) = \alpha_2/s = 18/118 = 0.153$
$E(\pi_3) = \alpha_3/s = 8/118 = 0.280$
$E(\pi_4) = \alpha_4/s = 33/118 = 0.067$
$E(\pi_5) = \alpha_5/s = 16/118 = 0.136$
$E(\pi_6) = \alpha_6/s = 30/118 = 0.254$

Table 2.5 The LDA example: the given text

1: I eat orange and apple since those are juicy.
2: The weather is so windy today.
3: The hurricane Catherine passed with no major damage.
4: Watermelons here are sweat because of the hot weather.
5: Tropical storms usually end by November.

2. For updating the posterior of multinomial distribution with Dirichlet prior, we need only to update the Dirichlet prior by adding the observation counts to the Dirichlet hyper prior, and
3. The number of observations directly reveals the confidence on the expected value of the parameters.

Because of these important properties, the Dirichlet distribution is applied largely in different applications. In particular, LDA assumes that the learned parameters follow the Dirichlet distribution. The following section describes the LDA method.

2.2 Latent Dirichlet Allocation

LDA is a latent Bayesian topic model which is used for discovering the hidden topics of documents (Blei et al. 2003). In this model, a document can be represented as a mixture of the hidden topics, where each hidden topic is represented by a distribution over words occurred in the document. Suppose we have the sentences shown in Table 2.5.

Then, the LDA method automatically discovers the topics that the given text contain. Specifically, given two asked topics, LDA can learn the two topics and the topic assignments to the given text. The learned topics are represented using the words and their probabilities of occurring for each topic as presented in Table 2.6. The topic representation for topic A illustrates that this topic is about *fruits*. And, the topic representation for topic B illustrates that this topic B is about the *weather*. Then, the topic assignment for each sentence can be calculated as presented in Table 2.7.

Formally, given a document in the form of $\mathbf{d} = (w_1, \dots, w_M)$ in a document corpus (set), $\mathbf{D}$, and given N asked topics, the LDA model learns two parameters:

1. The parameter $\boldsymbol{\theta}$ which is generated from the Dirichlet prior $\boldsymbol{\alpha}$.
2. The parameter $\boldsymbol{\beta}$ which is generated from Dirichlet prior $\boldsymbol{\eta}$.

Table 2.6 The LDA
example: the learned topics

Topic	A (%)	Topic	B (%)
Orange	20	Weather	30
Apple	20	Windy	10
Juicy	5	Hot	10
Sweat	1	Storm	9
...	...	...	...

Table 2.7 The LDA
example: the topic
assignments to the text

Sentence 1	Topic A	100 %	Topic B	0 %
Sentence 2	Topic A	0 %	Topic B	100 %
Sentence 3	Topic A	0 %	Topic B	100 %
Sentence 4	Topic A	65 %	Topic B	35 %
Sentence 5	Topic A	0 %	Topic B	100 %

The first parameter, θ , is a vector of size N for distribution of hidden topics, z . The second one, β , is a matrix of size $M \times N$ in which the column j stores the probability of each word given the topic z_j .

Figure 2.2 shows the LDA model in the plate notation in which the boxes are plates, that represents replicates. The shaded nodes are the observation nodes, i.e., the words w . The unshaded nodes z represent hidden topics. Then, the generative model of LDA performs as follows:

1. For each document, d , a parameter, θ , is drawn for the distribution of hidden topics based on multinomial distribution with the Dirichlet parameters α (cf. Dirichlet distribution in Sect. 2.1).
2. For each document set D , a parameter, β , is learned for the distribution of words given topics. Given each topic z , the vector β_z is drawn based on multinomial distribution with the Dirichlet parameters η .
3. Generate the j th word in the document i , $w_{i,j}$, as:
 - (a) Draw a topic $z_{i,j}$ based on the multinomial distribution with the parameter θ_i .
 - (b) Draw a word based on the multinomial distribution with the parameter $\phi_{z_{i,j}}$.

Comparison to Earlier Models

Blei et al. (2003) compared the LDA model to the related earlier models such as unigrams and mixture of unigrams (Bishop 2006; Manning and Schütze 1999), as well as probabilistic latent semantic analysis (PLSA) (Hofmann 1999). These three models are represented in Fig. 2.3.

Fig. 2.2 Latent Dirichlet allocation

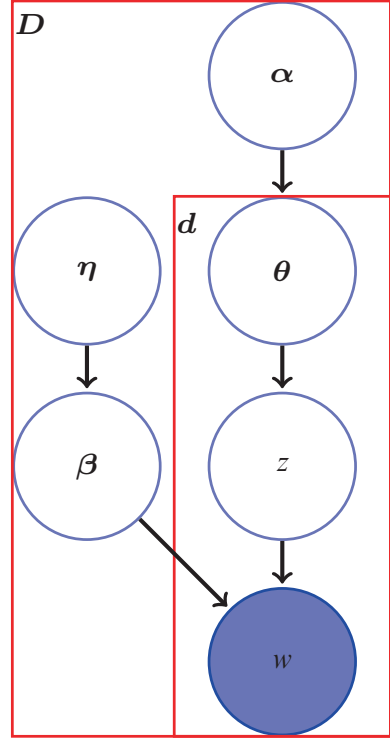


Figure 2.3a shows the unigram model. In unigrams, a document $\mathbf{d} = (w_1, \dots, w_n)$ is a mixture of words. So, the probability of having a document $\mathbf{d}$ is calculated as:

$$p(\mathbf{d}) = \prod_{w_i} p(w_i)$$

Then, in the mixture of unigrams in Fig. 2.3b, a word w is drawn from a topic z this time. Under this model, a document $\mathbf{d}$ is generated by:

1. Draw a hidden topic z .
2. Draw each word w based on the hidden topic z .

As such, in mixture of unigrams the probability of having the document $\mathbf{d}$ is calculated as:

$$p(\mathbf{d}) = \sum_z p(z) \prod_{w_i} p(w_i|z).$$

Notice that mixture of unigrams assumes that each document $\mathbf{d}$ includes only one hidden topic. This assumption is removed in PLSA model shown in Fig. 2.3c. In PLSA, a distribution θ is sampled and attached to each *observed* document

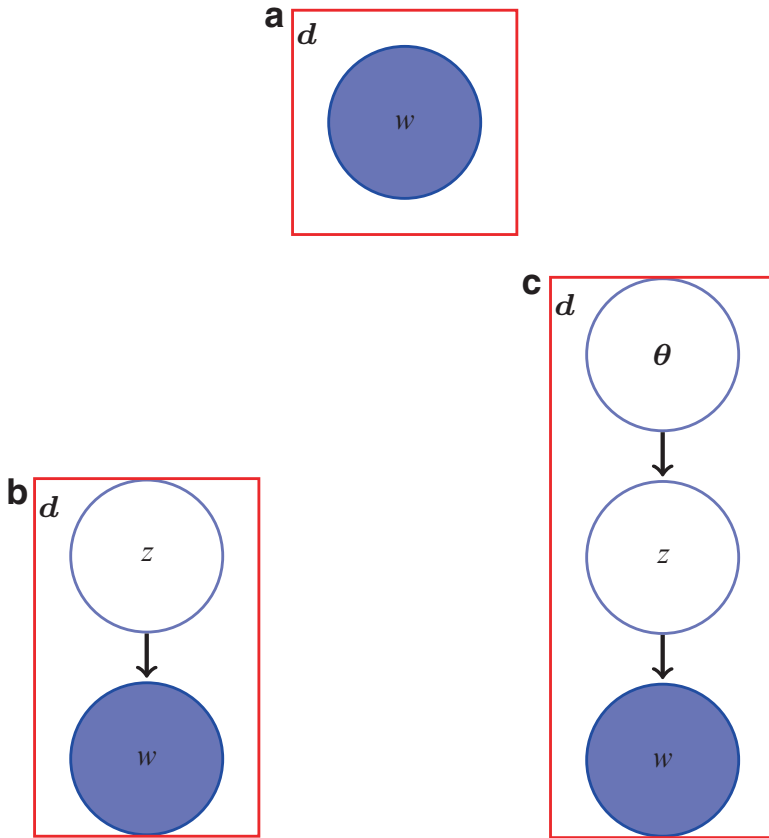


Fig. 2.3 (a) Unigram model. (b) Mixture of unigrams. (c) Probabilistic latent semantic analysis (PLSA)

for the distribution of hidden topics. Then, the probability of having a document $\mathbf{d} = (w_1, \dots, w_n)$ is calculated as:

$$p(\mathbf{d}) = \sum_z p(z|\theta) \prod_{w_i} p(w_i|z),$$

where θ is the distribution of hidden topics.

Note also that LDA is similar to PLSA in that both LDA and PLSA learn a parameter θ for the distribution of hidden topics of each document. Then, the probability of having a document $\mathbf{d} = (w_1, \dots, w_n)$ is calculated using:

$$p(\mathbf{d}) = \sum_z p(z|\theta) \prod_{w_i} p(w_i|z),$$

where θ is the distribution of hidden topics.

In contrast to PLSA, in LDA first a parameter α is generated which is used as the Dirichlet prior for the multinomial distribution θ of topics. In fact, Dirichlet prior can be used as a natural way to assign more probability to the random variables on which we have more confidence. Moreover, use of Dirichlet prior leads to interesting advantages of LDA over PLSA. First, as opposed to PLSA, LDA does not require to visit a document d to sample a parameter θ . But in LDA, the parameter θ is generated using the Dirichlet parameter α . As such, LDA is a well-defined generative model of documents which is able to assign probabilities to a previously *unseen* document of the corpus. Moreover, LDA is not dependent on the size of corpus and does not overfit as opposed to PLSA (Blei et al. 2003).

So, LDA is a topic modeling approach that considers mixture of hidden topics for documents, where documents are seen as bag of words. However, it does not consider the Markovian property among sentences. Later in this book, we introduce a variation of LDA that adds the Markovian property to LDA, for the topic transition from one sentence to the following one. In this context, hidden Markov models (HMMs) are used for modeling Markovian property particularly in texts. In the following section, we briefly review HMMs.

2.3 Hidden Markov Models

In Markovian domains the current environment's state depends on the state in the previous time step, similar to finite state machines. In fact, Markov models are generalized models of finite state machines in which the transitions are not deterministic. That is, in Markov models the current environment state depends on the previous state and the probability of landing to the current state, known as the transition probability (Manning and Schütze 1999).

In hidden Markov models (HMMs) (Rabiner 1990), as opposed to Markov models, states are not fully observable, but there is the idea of observations which give the current state of the model with only some probability. So, in HMMs there is an observation model besides the transition model. Similar to the Markov models, in HMMs the transition model is used for estimating the current state of the model with some probability, given the previous state. As such, we can state that an HMM with a deterministic observation model is equivalent to a Markov model, and that a Markov model with a deterministic transition model is equivalent to a finite state machine.

Figure 2.4 shows an HMM where hidden states $s_1, \dots, s_n$ are inside circles and observations $o_1, \dots, o_n$ are noted inside the shaded circles. The Markovian property in HMMs states that at each time step the state of the HMM depends on its previous state $p(s_t|s_{t-1})$, and the current observation depends on the current state $p(o_t|s_t)$.

Formally, an HMM is defined as a tuple (S, O, A, B, Π) :

- $S = s_1, \dots, s_N$ is a set of N states,
- The transition probability matrix A

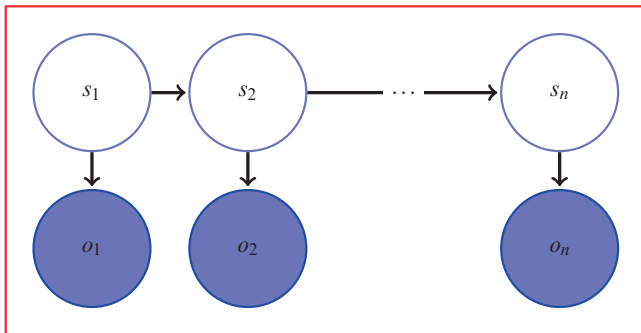


Fig. 2.4 The hidden Markov model, the *shaded nodes* are observations (o_i) used to capture hidden states (s_i)

$$A = \begin{pmatrix} a_{11}, \dots, a_{1n} \\ \dots \\ a_{n1}, \dots, a_{nn} \end{pmatrix}.$$

Each a_{ij} represents the probability of moving from state i to state j , s.t.

- $\sum_{j=1}^n a_{ij} = 1$,
- $O = o_1 o_2 \dots o_T$, is sequence of T observations, each one drawn from a vocabulary $V = v_1, v_2, \dots, v_V$,
- $B = b_i(o_i)$, is a sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation o_i being generated from a state i ,
- Π is the initial probability model which shows the probability that the model starts with each state in S .

Then, there are three fundamental questions that we want to answer in HMMs (Jurafsky and Martin 2009; Manning and Schütze 1999):

1. The first problem is to compute the *likelihood* of a particular observation sequence. Formally, we want to find out:

Given an HMM, $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $\Pr(O|\lambda)$.

2. Learning the most likely state sequence given a sequence of observations and the model. This problem is called *decoding*. This is interesting, for instance, in part-of-speech tagging where given a set of words as observations we would like to infer about the most probable tags of the words (Church 1988). Formally, we want to find out:

Given as input and HMM $\lambda = (A, B)$, and a sequence of observations $O = o_1, o_2, \dots, o_T$, find the most probable sequence of states, i.e., $(s_1, \dots, s_T)$.

That is, we want to find out *the state sequence that best explains the observations*.

3. HTMM training, i.e., learning the HMM parameters. Given a sequence of observations what the most probable model parameters are:

$$\operatorname{argmax}_{\lambda} p(o_1, \dots, o_n | \lambda). \quad (2.3)$$

This problem is called *parameter estimation*.

Note that there is no analytical solution for the maximization of parameter estimation in Eq.(2.3). This problem is tackled with a well-known algorithm named as *Baum–Welch* or *Forward–Backward* algorithm (Welch 2003), which is an *Expectation Maximization* (EM) algorithm.

In fact, EM is a class of algorithms for learning unknown parameters of a model. The basic idea is to pretend that the parameters of the model are known and then to infer the probability that each observation belongs to each model (Russell and Norvig 2010). Then, the model refits to the observations. Specifically, each model is fitted to all observations, and each observation is weighted by the probability that it belongs to that model. This process iterates until convergence.

EM algorithms start with a random parameter, and calculate the probability of observations. Then, they observe in the calculations to find which state transitions and observation probabilities have been used most, and increase the probability of those. This process leads to an updated parameter which gives higher probability to the observations. Then, the following two steps are iterated until convergence: calculating the probabilities of observations given a parameter (expectation) and updating the parameter (maximization).

Formally, an EM algorithm works as follows. Assuming the set of parameter Θ , hidden variables Z and observations X . First, the function Q is defined as (Dempster et al. 1977):

$$Q(\Theta | \Theta^t) = E[\log p(X, Z | \Theta) | X, \Theta^t]. \quad (2.4)$$

Then, in the expectation and maximization steps the following calculations are performed:

1. Expectation: $Q(\Theta | \Theta^t)$ is computed.
2. Maximization: $\Theta^{t+1} = \operatorname{argmax}_{\Theta} Q(\Theta | \Theta^t)$

That is, the parameter Θ^{t+1} is set to the Θ that maximizes $Q(\Theta | \Theta^t)$.

For instance, in Baum–Welch algorithm the expectation and maximization steps are as follows:

1. In the expectation the following two calculations are done:
 - Calculating the expected number of times that observation o has been observed from state s for all states and observations, given the current parameter of the model.
 - Calculating the expected number of times that state transitions from state s_i to state s_j is done, given the current parameters of the model.

2. In the maximization step the parameters A , B , and Π are set to the parameters which maximize the expectations above.

More specifically, the Expectation and Maximization step for HMM parameter learning can be derived as described in Jurafsky and Martin (2009):

1. Expectation:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\Pr(O|\gamma)} \quad \forall t \text{ and } j$$

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)} \quad \forall t, i, \text{ and } j,$$

where α_t is known as *forward path probability*:

$$\alpha_t(j) = \Pr(o_1, o_2, \dots, o_t, s_t = j | \lambda)$$

and $\beta_t(j)$ is known as *backward path probability*:

$$\beta_t(i) = \Pr(o_{t+1}, o_{t+2}, \dots, o_T | s_t = i, \lambda)$$

2. Maximization:

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi(i, j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \xi(i, j)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1, s.t. O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}.$$

In this section, we introduced the basic methods used in topic modeling. In particular, we studied the LDA method and HMMs, the background for hidden topic Markov model (HTMM). The HTMM approach adds Markovian property to the LDA method, and is introduced in Chap. 4. In the following chapter, we introduce the sequential decision making domain and its application on spoken dialogue systems.

Building Dialogue POMDPs from Expert Dialogues

An end-to-end approach

Chinaei, H.; Chaib-draa, B.

2016, VII, 119 p. 22 illus., 21 illus. in color., Softcover

ISBN: 978-3-319-26198-0