Parameter estimation for text analysis

Gregor Heinrich

gregor@arbylon.net

Abstract. This primer presents parameter estimation methods common in Bayesian statistics and apply them to discrete probability distributions, which commonly occur in text modeling. Presentation starts with maximum likelihood and a posteriori estimation approaches and the full Bayesian approach. This presentation is completed by an overview of Bayesian networks, a graphical language to express probabilistic models. As an application, the model of latent Dirichlet allocation is explained and a full derivation of an approximate inference algorithm given based on Gibbs sampling.

1 Introduction

The motivation for these overview notes is to explain the basics in Bayesian parameter estimation necessary to understand the inner workings of topic-based text analysis approaches like probabilistic latent semantic analysis (PLSA) [1], latent Dirichlet allocation (LDA) [2] or methods of the same family. It appears that there is no common book or introductory paper that fills this role: Most known texts use examples from the Gaussian domain, which look rather different and much more complicated compared to applications that use count data. On the other hand, the examples of the latter are missing in most literature but are crucial when it comes to understanding text analysis approaches.

We therefore will systematically introduce the basic concepts of parameter estimation with a couple of simple examples on binary data in Section 2. We then will introduce the concept of conjugacy along with a refresher on the most common probability distributions needed in the text domain in Section 3. The joint presentation of conjugacy with associated real-world conjugate pairs directly justifies the choice of distributions introduced. Section 4 will introduce Bayesian networks as a graphical language to describe systems via their probabilistic models.

With these basic concepts, we present the idea of latent Dirichlet allocation (LDA) in Section 5, a flexible model to estimate the properties of text. On the example of LDA, the usage of Gibbs sampling is shown as a straight-forward means of approximate inference in Bayesian networks.

2 Parameter estimation approaches

We face two inference problems, (1) to estimate values for a set of distribution parameters ϑ that can best explain a set of observations $\mathcal{X}$ and (2) to calculate

the probability of new observations $\tilde{x}$ given previous observations, i.e., to find $p(\tilde{x}|\mathcal{X})$. We will refer to the former problem as the estimation problem and to the latter as the prediction problem.

The data set $\mathcal{X} \triangleq \{x_i\}_{i=1}^{|\mathcal{X}|}$ can be considered a sequence of independent and identically distributed (i.i.d.) realisations of a random variable (r.v.) X. The parameters ϑ are dependent on the distributions considered, e.g., for a Gaussian, $\vartheta = \{\mu, \sigma^2\}$.

For these data and parameters, a couple of probability functions are ubiquitous in Bayesian statistics. They are best introduced as parts of Bayes' rule, which is[1]:

$$p(\vartheta|\mathcal{X}) = \frac{p(\mathcal{X}|\vartheta) \cdot p(\vartheta)}{p(\mathcal{X})}, \tag{1}$$

and we define the corresponding terminology:

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}. \tag{2}$$

In the next paragraphs, we will show different estimation methods that start from simple maximisation of the likelihood, then show how prior belief on parameters can be incorporated by maximising the posterior and finally use Bayes' rule to infer a complete posterior distribution.

Maximum likelihood (ML) estimation tries to find parameters that maximise the likelihood,

$$L(\vartheta|\mathcal{X}) \triangleq p(\mathcal{X}|\vartheta) = \bigcap_{x \in \mathcal{X}} \{X = x|\vartheta\} = \prod_{x \in \mathcal{X}} p(x|\vartheta), \tag{3}$$

i.e., the probability of the joint event that X generates the data $\mathcal{X}$. Because of the product in Eq. 3, it is often simpler to use the log likelihood, $\mathcal{L} \triangleq \log L$. The ML estimation problem then can be written as:

$$\hat{\vartheta}_{\text{ML}} = \underset{\vartheta}{\operatorname{argmax}}\, \mathcal{L}(\vartheta|\mathcal{X}) = \underset{\vartheta}{\operatorname{argmax}} \sum_{x \in \mathcal{X}} \log p(x|\vartheta). \tag{4}$$

The common way to obtain the parameter estimates is to solve the system:

$$\frac{\partial \mathcal{L}(\vartheta|\mathcal{X})}{\partial \vartheta_k} \stackrel{!}{=} 0 \quad \forall \vartheta_k \in \vartheta. \tag{5}$$

The probability of a new observation $\tilde{x}$ given the data $\mathcal{X}$ can now be found using the approximation[2]:

$$p(\tilde{x}|\mathcal{X}) = \int_{\vartheta \in \Theta} p(\tilde{x}|\vartheta)\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta \tag{6}$$

$$\approx \int_{\vartheta \in \Theta} p(\tilde{x}|\hat{\vartheta}_{\text{ML}})\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta = p(\tilde{x}|\hat{\vartheta}_{\text{ML}}), \tag{7}$$

[1] Derivation: $p(\vartheta|\mathcal{X}) \cdot p(\mathcal{X}) = p(\mathcal{X}, \vartheta) = p(\mathcal{X}|\vartheta) \cdot p(\vartheta)$.

[2] The ML estimate $\hat{\vartheta}_{\text{ML}}$ is considered a constant, and the integral over the parameters given the data is the total probability that integrates to one.

that is, the next sample is anticipated to be distributed with the estimated parameters $\hat{\vartheta}_{\mathrm{ML}}$.

As an example, consider a set $\mathcal{C}$ of N Bernoulli experiments with unknown parameter p, e.g., realised by tossing a deformed coin. The Bernoulli density function for the r.v. C for one experiment is:

$$p(C{=}c|p) = p^c\,(1-p)^{1-c} \triangleq \mathrm{Bern}(c|p) \tag{8}$$

where we define $c{=}1$ for heads and $c{=}0$ for tails[3].

Building an ML estimator for the parameter p can be done by expressing the (log) likelihood as a function of the data:

$$\begin{aligned}\mathcal{L} &= \log \prod_{i=1}^{N} p(C{=}c_i|p) = \sum_{i=1}^{N} \log p(C{=}c_i|p) &(9)\\ &= n^{(1)} \log p(C{=}1|p) + n^{(0)} \log p(C{=}0|p)\\ &= n^{(1)} \log p + n^{(0)} \log(1-p) &(10)\end{aligned}$$

where $n^{(c)}$ is the number of times a Bernoulli experiment yielded event c. Differentiating with respect to (w.r.t.) the parameter p yields:

$$\frac{\partial \mathcal{L}}{\partial p} = \frac{n^{(1)}}{p} - \frac{n^{(0)}}{1-p} \stackrel{!}{=} 0 \quad \Leftrightarrow \quad \hat{p}_{\mathrm{ML}} = \frac{n^{(1)}}{n^{(1)}+n^{(0)}} = \frac{n^{(1)}}{N}, \tag{11}$$

which is simply the ratio of heads results to the total number of samples. To put some numbers into the example, we could imagine that our coin is strongly deformed, and after 20 trials, we have $n^{(1)}{=}12$ times heads and $n^{(0)}{=}8$ times tails. This results in an ML estimation of of $\hat{p}_{\mathrm{ML}} = 12/20 = 0.6$.

Maximum a posteriori (MAP) estimation is very similar to ML estimation but allows to include some a priori belief on the parameters by weighting them with a prior distribution $p(\vartheta)$. The name derives from the objective to maximise the posterior of the parameters given the data:

$$\hat{\vartheta}_{\mathrm{MAP}} = \operatorname*{argmax}_{\vartheta} p(\vartheta|\mathcal{X}). \tag{12}$$

By using Bayes' rule (Eq. 1), this can be rewritten to:

$$\begin{aligned}\hat{\vartheta}_{\mathrm{MAP}} &= \operatorname*{argmax}_{\vartheta} \frac{p(\mathcal{X}|\vartheta)p(\vartheta)}{p(\mathcal{X})} \qquad \Big|\; p(\mathcal{X}) \neq f(\vartheta)\\ &= \operatorname*{argmax}_{\vartheta} p(\mathcal{X}|\vartheta)p(\vartheta) = \operatorname*{argmax}_{\vartheta} \{\mathcal{L}(\vartheta|\mathcal{X}) + \log p(\vartheta)\}\\ &= \operatorname*{argmax}_{\vartheta} \Big\{ \sum_{x\in\mathcal{X}} \log p(x|\vartheta) \;+\; \log p(\vartheta) \Big\}. &(13)\end{aligned}$$

[3] The notation in Eq. 8 is somewhat peculiar because it makes use of the values of c to "filter" the respective parts in the density function and additionally uses these numbers to represent disjoint events.

Compared to Eq. 4, a prior distribution is added to the likelihood. In practice, the prior $p(\vartheta)$ can be used to encode extra knowledge as well as to prevent overfitting by enforcing preference to simpler models, which is also called Occam's razor[4].

With the incorporation of $p(\vartheta)$, MAP follows the Bayesian approach to data modelling where the parameters ϑ are thought of as r.v.s. With priors that are parametrised themselves, i.e., $p(\vartheta) := p(\vartheta|\alpha)$ with hyperparameters α, the belief in the anticipated values of ϑ can be expressed within the framework of probability[5], and a hierarchy of parameters is created.

MAP parameter estimates can be found by maximising the term $\mathcal{L}(\vartheta|\mathcal{X}) + \log p(\vartheta)$, similar to Eq. 5. Analogous to Eq. 7, the probability of a new observation, $\tilde{x}$, given the data, $\mathcal{X}$, can be approximated using:

$$p(\tilde{x}|\mathcal{X}) \approx \int_{\vartheta\in\Theta} p(\tilde{x}|\hat{\vartheta}_{\mathrm{MAP}})\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta = p(\tilde{x}|\hat{\vartheta}_{\mathrm{MAP}}). \tag{14}$$

Returning to the simplistic demonstration on ML, we can give an example for the MAP estimator. Consider the above experiment, but now there are values for p that we believe to be more likely, e.g., we believe that a coin usually is fair. This can be expressed as a prior distribution that has a high probability around 0.5. We choose the beta distribution:

$$p(p|\alpha,\beta) = \frac{1}{\mathrm{B}(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \triangleq \mathrm{Beta}(p|\alpha,\beta), \tag{15}$$

with the beta function $\mathrm{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$. The function $\Gamma(x)$ is the Gamma function, which can be understood as a generalisation of the factorial to the domain of real numbers via the identity $x! = \Gamma(x+1)$. The beta distribution supports the interval [0,1] and therefore is useful to generate normalised probability values. For a graphical representation of the beta probability density function (pdf), see Fig. 1. As can be seen, with different parameters the distribution takes on quite different pdfs.

In our example, we believe in a fair coin and set $\alpha = \beta = 5$, which results in a distribution with a mode (maximum) at 0.5. The optimisation problem now becomes (cf. Eq. 11):

$$\frac{\partial}{\partial p}\mathcal{L} + \log p(p) = \frac{n^{(1)}}{p} - \frac{n^{(0)}}{1-p} + \frac{\alpha-1}{p} - \frac{\beta-1}{1-p} \stackrel{!}{=} 0 \tag{16}$$

$$\Leftrightarrow \quad \hat{p}_{\mathrm{MAP}} = \frac{n^{(1)}+\alpha-1}{n^{(1)}+n^{(0)}+\alpha+\beta-2} = \frac{n^{(1)}+4}{n^{(1)}+n^{(0)}+8} \tag{17}$$

This result is interesting in two aspects. The first one is the changed behaviour of the estimate $\hat{p}_{\mathrm{MAP}}$ w.r.t. the counts $n^{(c)}$: their influence on the estimate is

[4] Pluralitas non est ponenda sine necessitate = Plurality should not be posited without necessity. Occam's razor is also called the principle of parsimony.

[5] Belief is not identical to probability, which is one of the reasons why Bayesian approaches are disputed by some theorists despite their practical importance.

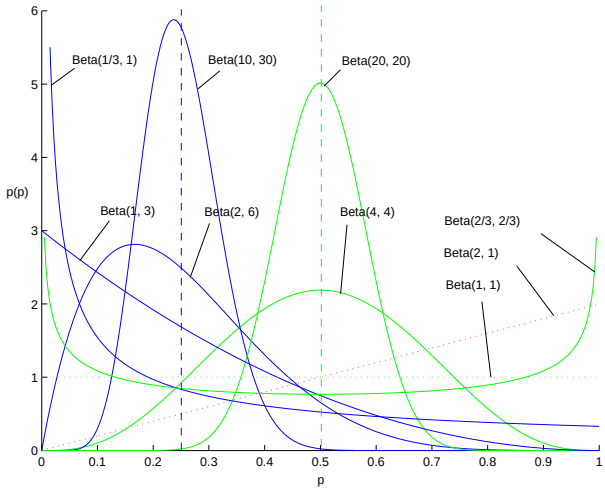

Fig. 1. Density functions of the beta distribution with different symmetric and asymmetric parametrisations.

reduced by the additive values that "pull" the value towards $\hat{p}_{\mathrm{MAP}} = 4/8 = 0.5$. The higher the values of the hyperparameters α and β, the more actual observations are necessary to revise the belief expressed by them. The second interesting aspect is the exclusive appearance of the sums $n^{(1)}+\alpha-1$ and $n^{(0)}+\beta-1$: It is irrelevant whether the counts actually derive from actual observations or prior belief expressed as hypervariables. This is why the hyperparameters α and β are often referred to as pseudo-counts. The higher pseudo-counts exist, the sharper the beta distribution is concentrated around its maximum. Again, we observe in 20 trials $n^{(1)}$=12 times heads and $n^{(0)}$=8 times tails. This results in an MAP estimation of $\hat{p}_{\mathrm{MAP}} = 16/28 = 0.571$, which in comparison to $\hat{p}_{\mathrm{ML}} = 0.6$ shows the influence of the prior belief of the "fairness" of the coin.

Bayesian estimation extends the MAP approach by allowing a distribution over the parameter set ϑ instead of making a direct estimate. Not only encodes this the maximum (a posteriori) value of the data-generated parameters, but it also incorporates expectation as another parameter estimate as well as variance information as a measure of estimation quality or confidence. The main step in this approach is the calculation of the posterior according to Bayes' rule:

$$p(\vartheta|\mathcal{X}) = \frac{p(\mathcal{X}|\vartheta) \cdot p(\vartheta)}{p(\mathcal{X})}. \tag{18}$$

As we do not restrict the calculation to finding a maximum, it is necessary to calculate the normalisation term, i.e., the probability of the "evidence", $p(\mathcal{X})$, in

Eq. 18. Its value can be expressed by the total probability w.r.t. the parameters[6]:

$$p(\mathcal{X}) = \int_{\vartheta \in \Theta} p(\mathcal{X}|\vartheta)\, p(\vartheta)\, \mathrm{d}\vartheta. \tag{19}$$

As new data are observed, the posterior in Eq. 18 is automatically adjusted and can eventually be analysed for its statistics. However, often the normalisation integral in Eq. 19 is the intricate part of Bayesian estimation, which will be treated further below.

In the prediction problem, the Bayesian approach extends MAP by ensuring an exact equality in Eq. 14, which then becomes:

$$p(\tilde{x}|\mathcal{X}) = \int_{\vartheta \in \Theta} p(\tilde{x}|\vartheta)\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta \tag{20}$$

$$= \int_{\vartheta \in \Theta} p(\tilde{x}|\vartheta)\, \frac{p(\mathcal{X}|\vartheta)p(\vartheta)}{p(\mathcal{X})}\, \mathrm{d}\vartheta \tag{21}$$

Here the posterior $p(\vartheta|\mathcal{X})$ replaces an explicit calculation of parameter values ϑ. By integration over ϑ, the prior belief is automatically incorporated into the prediction, which itself is a distribution over $\tilde{x}$ and can again be analysed w.r.t. confidence, e.g., via its variance.

As an example, we build a Bayesian estimator for the above situation of having N Bernoulli observations and a prior belief that is expressed by a beta distribution with parameters $(5, 5)$, as in the MAP example. In addition to the maximum a posteriori value, we want the expected value of the now-random parameter p and a measure of estimation confidence. Including the prior belief, we obtain[7]:

$$p(p|\mathcal{C}, \alpha, \beta) = \frac{\prod_{i=1}^{N} p(C{=}c_i|p)\; p(p|\alpha, \beta)}{\int_0^1 \prod_{i=1}^{N} p(C{=}c_i|p)\; p(p|\alpha, \beta)\, \mathrm{d}p} \tag{22}$$

$$= \frac{p^{n^{(1)}} (1-p)^{n^{(0)}}\; \frac{1}{\mathrm{B}(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}}{Z} \tag{23}$$

$$= \frac{p^{[n^{(1)}+\alpha]-1}(1-p)^{[n^{(0)}+\beta]-1}}{\mathrm{B}(n^{(1)}+\alpha, n^{(0)}+\beta)} \tag{24}$$

$$= \mathrm{Beta}(p|n^{(1)}+\alpha, n^{(0)}+\beta) \tag{25}$$

The $\mathrm{Beta}(\alpha, \beta)$ distribution has mean, $\mathsf{E}\{p|\alpha, \beta\} = \alpha(\alpha+\beta)^{-1}$, and variance, $\mathsf{V}\{p|\alpha, \beta\} = \alpha\beta(\alpha+\beta+1)^{-1}(\alpha+\beta)^{-2}$. Using these statistics, our estimation

[6] This marginalisation is why evidence is also refered to as "marginal likelihood". The integral is used here as a generalisation for continuous and discrete sample spaces, where the latter require sums.

[7] The marginal likelihood Z in the denominator is simply determined by the normalisation constraint of the beta distribution.

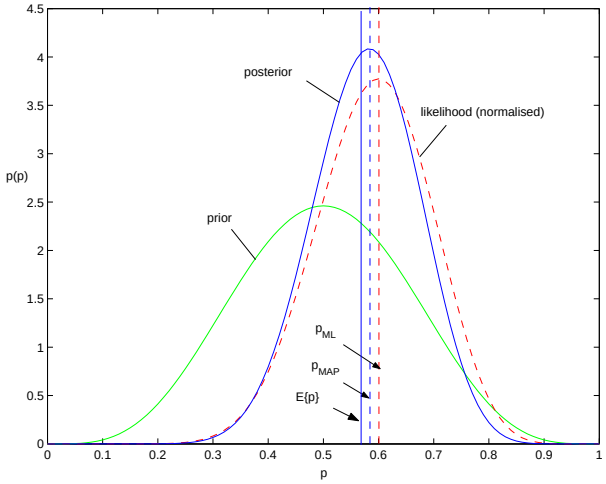

Fig. 2. Visualising the coin experiment.

result is:

$$\mathsf{E}\{p|\mathcal{C}\} = \frac{n^{(1)}+\alpha}{n^{(1)}+n^{(0)}+\alpha+\beta} = \frac{n^{(1)}+5}{N+10} \tag{26}$$

$$\mathsf{V}\{p|\mathcal{C}\} = \frac{(n^{(1)}+\alpha)(n^{(0)}+\beta)}{(N+\alpha+\beta+1)(N+\alpha+\beta)^2} = \frac{(n^{(1)}+5)(n^{(0)}+5)}{(N+11)(N+10)^2} \tag{27}$$

The expectation is not identical to the MAP estimate (see Eq. 17), which literally is the maximum and not the expected value of the posterior. However, if the sums of the counts and pseudo-counts become larger, both expectation and maximum converge. With the 20 coin observations from the above example ($n^{(1)}$=12 and $n^{(0)}$=8), we obtain the situation depicted in Fig. 2. The Bayesian estimation values are $\mathsf{E}\{p|\mathcal{C}\} = 17/30 = 0.567$ and $\mathsf{V}\{p|\mathcal{C}\} = 17 \cdot 13/(31 \cdot 30^2) = 0.0079$.

3 Conjugate distributions

Calculation of Bayesian models often becomes quite difficult, e.g., because the summations or integrals of the marginal likelihood are intractable or there are unknown variables. Fortunately, the Bayesian approach leaves some freedom to the encoding of prior belief, and a frequent strategy to facilitate model inference is to use *conjugate prior* distributions.

A conjugate prior, $p(\vartheta)$, of a likelihood, $p(x|\vartheta)$, is a distribution that results in a posterior distribution, $p(\vartheta|x)$ with the same functional form as the prior (but different parameters). The last example (Eq. 25 and above) illustrates this: The posterior turned out to be a beta distribution like the prior, and the determination of the normalising term $1/Z$ was simple.

In addition to calculational simplifications, conjugacy often results in meaningful interpretations of hyperparameters, and in our beta–Bernoulli case, the resulting posterior can be interpreted as the prior with the observation counts $n^{(c)}$ added to the pseudo-counts α and β (see Eq. 25).

Moreover, conjugate prior-likelihood pairs often allow to marginalise out the likelihood parameters in closed form and thus express the likelihood of observations directly in terms of hyperparameters. For the beta–Bernoulli case, this looks as follows[8]:

$$p(\mathcal{C}|\alpha,\beta) = \int_0^1 p(\mathcal{C}|p)\; p(p|\alpha,\beta)\,\mathrm{d}p \tag{28}$$

$$= \int_0^1 p^{n^{(1)}}(1-p)^{n^{(0)}}\; \frac{1}{\mathrm{B}(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}\,\mathrm{d}p \tag{29}$$

$$= \frac{1}{\mathrm{B}(\alpha,\beta)} \int_0^1 p^{n^{(1)}+\alpha-1}(1-p)^{n^{(0)}+\beta-1}\,\mathrm{d}p \qquad \Big|\ \text{Beta}\int \tag{30}$$

$$= \frac{\mathrm{B}(n^{(1)}+\alpha, n^{(0)}+\beta)}{\mathrm{B}(\alpha,\beta)} = \frac{\Gamma(n^{(1)}+\alpha)\Gamma(n^{(0)}+\beta)}{\Gamma(n^{(1)}+n^{(0)}+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}. \tag{31}$$

This result can be used to make predictions on the distribution of future Bernoulli trials without explicit knowledge of the parameter p but from prior observations. This is expressed with the predictive likelihood for a new observation[9]:

$$p(\tilde{c}{=}1|\mathcal{C},\alpha,\beta) = \frac{p(\tilde{c}{=}1,\mathcal{C}|\alpha,\beta)}{p(\mathcal{C}|\alpha,\beta)} = \frac{\frac{\Gamma(n^{(1)}+1+\alpha)}{\Gamma(n^{(1)}+1+n^{(0)}+\alpha+\beta)}}{\frac{\Gamma(n^{(1)}+\alpha)}{\Gamma(n^{(1)}+n^{(0)}+\alpha+\beta}} \tag{32}$$

$$= \frac{n^{(1)}+\alpha}{n^{(1)}+n^{(0)}+\alpha+\beta}. \tag{33}$$

There are a couple of important prior–likelihood pairs that can be used to simplify Bayesian inference as described above. One important example related to the beta distribution is the binomial distribution, which gives the probability that exactly $n^{(1)}$ heads from the N Bernoulli experiments with parameter p are observed:

$$p(n^{(1)}|p,N) = \binom{N}{n^{(1)}} p^{n^{(1)}}(1-p)^{n^{(0)}} \triangleq \mathrm{Bin}(n^{(1)}|p,N) \tag{34}$$

As the parameter p has the same meaning as with the Bernoulli distribution, it comes not as a surprise that the conjugate prior on the parameter p of a binomial distribution is a beta distribution, as well. Other distributions that count Bernoulli trials also fall into this scheme, such as the negative-binomial distribution.

[8] In the calculation, the identity of the beta integral, $\int_0^1 x^a(1-x)^b\,\mathrm{d}x = \mathrm{B}(a+1,b+1)$ is used, also called Eulerian integral of the first kind.

[9] Here the identity $\Gamma(x+1) = x\Gamma(x)$ is used.

Multivariate case. The distributions considered so far handle outcomes of a binary experiment. If we generalise the number of possible events from 2 to a finite integer K, we can obtain a K-dimensional Bernoulli or *multinomial* experiment, e.g., the roll of a die. If we repeat this experiment, we obtain a multinomial distribution of the counts of the observed events (faces of the die), which generalises the binomial distribution:

$$p(\vec{n}|\vec{p}, N) = \binom{N}{\vec{n}} \prod_{k=1}^{K} p_k^{n^{(k)}} \triangleq \text{Mult}(\vec{n}|\vec{p}, N) \tag{35}$$

with the multinomial coefficient $\binom{N}{\vec{n}} = \frac{N!}{\prod_k n^{(k)}!}$. Further, the elements of $\vec{p}$ and $\vec{n}$ follow the constraints $\sum_k p_k = 1$ and $\sum_k n^{(k)} = N$ (cf. the terms $(1-p)$ and $n^{(1)} + n^{(0)} = N$ in the binary case).

The multinomial distribution governs the multivariate variable $\vec{n}$ with elements $n^{(k)}$ that count the occurrences of event k within N total trials, and the multinomial coefficient counts the number of configurations of individual trials that lead to the total.

A single multinomial trial generalises the Bernoulli distribution to a discrete categorical distribution:

$$p(\vec{n}|\vec{p}) = \prod_{k=1}^{K} p_k^{n^{(k)}} = \text{Mult}(\vec{n}|\vec{p}, 1) \tag{36}$$

where the count vector $\vec{n}$ is zero except for a single element $n^{(z)}$=1. Hence we can simplify the product and replace the multivariate count vector by the index of the nonzero element z as an alternative notation:

$$p(z|\vec{p}) = p_z \triangleq \text{Mult}(z|\vec{p}), \tag{37}$$

which is identical to the general discrete distribution $\text{Disc}(\vec{p})$. Introducing the multinomial r.v. C, the likelihood of N repetitions of a multinomial experiment (cf. Eq. 9), the obervation set $\mathcal{C}$, becomes:

$$p(\mathcal{C}|\vec{p}) = \prod_{n=1}^{N} \text{Mult}(C{=}z_i|\vec{p}) = \prod_{n=1}^{N} p_{z_i} = \prod_{k=1}^{K} p_k^{n^{(k)}}, \tag{38}$$

which is just the multinomial distribution with a missing normalising multinomial coefficient. This difference is due to the fact that we assume a sequence of outcomes of the N experiments instead of getting the probability of a particular multinomial count vector $\vec{n}$, which could be generated by $\binom{N}{\vec{n}}$ different sequences $\mathcal{C}$.[10] In modelling text observations, this last form of a repeated multinomial experiment is quite important. For the parameters $\vec{p}$ of the multinomial

[10] In a binary setting, this corresponds to the difference between the observations from a repeated Bernoulli trial and the probability of (any) $n^{(1)}$ successes, which is described by the binomial distribution.

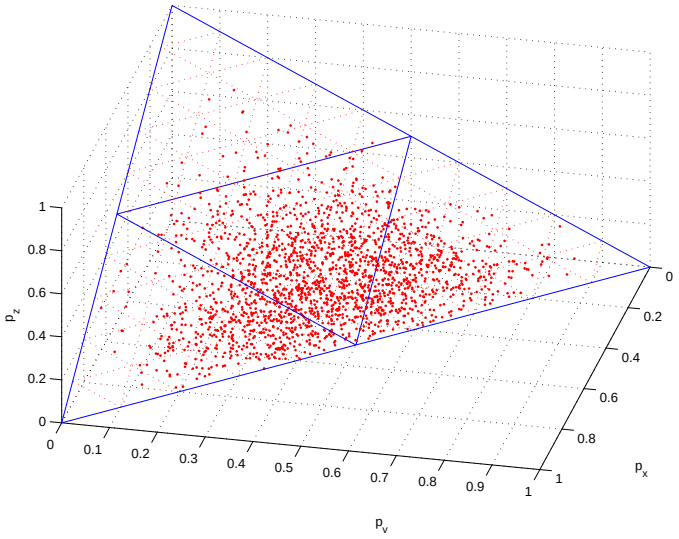

Fig. 3. 2000 samples from a Dirichlet distribution $\mathrm{Dir}(4, 4, 2)$. The plot shows that all samples are on a simplex embedded in the three-dimensional space, due to the constraint $\sum_k p_k = 1$.

distribution, the conjugate prior is the Dirichlet distribution, which generalises the beta distribution from 2 to K dimensions:

$$p(\vec{p}|\vec{\alpha}) = \mathrm{Dir}(\vec{p}|\vec{\alpha}) \triangleq \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{39}$$

$$\triangleq \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{K} p_k^{\alpha_k - 1}, \quad \Delta(\vec{\alpha}) = \frac{\prod_{k=1}^{\dim \vec{\alpha}} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{\dim \vec{\alpha}} \alpha_k)}, \tag{40}$$

with parameters $\vec{\alpha}$ and the "Dirichlet delta function" $\Delta(\vec{\alpha})$, which we introduce for notational convenience[11]. An example of a Dirichlet distribution can be seen in Fig. 3. In many applications, a symmetric Dirichlet distribution is used, which

[11] The function $\Delta(\vec{\alpha})$ can be seen as a multidimensional extension to the beta function: $\mathrm{B}(\alpha_1, \alpha_2) = \Delta(\{\alpha_1, \alpha_2\})$. It comes as a surprise that this notation is not used in the literature, especially since $\Delta(\vec{\alpha})$ can be shown to be the Dirichlet integral of the first kind for the summation function $f(\Sigma x_i)$=1: $\Delta(\vec{\alpha}) = \int_{\Sigma x_i = 1} \prod_i^N x_i^{\alpha_i - 1} \, \mathrm{d}^N \vec{x}$, analogous to the beta integral: $\mathrm{B}(\alpha_1, \alpha_2) = \int_0^1 x^{\alpha_1 - 1}(1 - x)^{\alpha_2 - 1} \, \mathrm{d}x$.

is defined in terms of a scalar parameter $\alpha = \sum \alpha_k / K$ and the dimension K:

$$p(\vec{p}|\alpha, K) = \mathrm{Dir}(\vec{p}|\alpha, K) \triangleq \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} p_k^{\alpha-1} \tag{41}$$

$$\triangleq \frac{1}{\Delta_K(\alpha)} \prod_{k=1}^{K} p_k^{\alpha-1}, \quad \Delta_K(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(K\alpha)}. \tag{42}$$

Modelling text. Consider a set $\mathcal{W}$ of N i.i.d. draws from a multinomial random variable W. This can be imagined as drawing N words w from a vocabulary $\mathcal{V}$ of size V. The likelihood of these samples is simply:

$$L(\vec{p}|\vec{w}) = p(\mathcal{W}|\vec{p}) = \prod_{t=1}^{V} p_t^{n^{(t)}}, \quad \sum_{t=1}^{V} n^{(t)} = N, \quad \sum_{t=1}^{V} p_t = 1, \tag{43}$$

where $n^{(t)}$ is the number of times term t was observed as a word[12]. This example is the unigram model, which assumes a general distribution of terms of a vocabulary $\mathcal{V}$, $\mathrm{Mult}(t \in \mathcal{V}|\vec{p})$, where $\vec{p}$ is the probability that term t is observed as word w in a document. The unigram model assumes just one likelihood for the entire text considered, which is for instance useful for general assumptions about a language or corpus but does not differentiate between any partial sets, e.g., documents. In addition, it is a perfect basis to develop more complex models.

Assuming conjugacy, the parameter vector $\vec{p}$ of the vocabulary can modelled with a Dirichlet distribution, $\vec{p} \sim \mathrm{Dir}(\vec{p}|\vec{\alpha})$. Analogous to Eq. 25, we obtain the important property of the Dirichlet posterior to merge multinomial observations $\mathcal{W}$ with prior pseudo-counts $\vec{\alpha}$:

$$p(\vec{p}|\mathcal{W}, \vec{\alpha}) = \frac{\prod_{n=1}^{N} p(\vec{w}|\vec{p})\ p(\vec{p}|\vec{\alpha})}{\int_{\mathcal{P}} \prod_{n=1}^{N} p(\vec{w}|\vec{p})\ p(\vec{p}|\vec{\alpha})\, \mathrm{d}\vec{p}} \tag{44}$$

$$= \frac{1}{Z} \prod_{t=1}^{V} p^{n^{(t)}}\ \frac{1}{\Delta(\vec{\alpha})} p^{\alpha_t - 1} \tag{45}$$

$$= \frac{1}{\Delta(\vec{\alpha} + \vec{n})} \prod_{t=1}^{V} p^{\alpha_t + n^{(t)} - 1} \tag{46}$$

$$= \mathrm{Dir}(\vec{p}|\vec{\alpha} + \vec{n}). \tag{47}$$

Here the likelihood of the words $\prod_{n=1}^{N} p(\vec{w}|\vec{p})$ was rewritten to that of repeated terms $\prod_{t=1}^{V} p(w{=}t|\vec{p})^{n^{(t)}}$ and the known normalisation of the Dirichlet distribution used. The pseudo-count behaviour of the Dirichlet corresponds to the important Pólya urn scheme: An urn contains W balls of V colours, and for

[12] Term refers to the element of a vocabulary, and word refers to the element of a document, respectively. We refer to terms if the category in a multinomial is meant and to words if a particular observation or count is meant. Thus a term can be instantiated by several words in a text corpus.

each sample of a ball $\tilde{w}$, the ball is replaced and an additional ball of the same colour added (sampling with over-replacement). That is, the Dirichlet exhibits a "rich get richer" or clustering behaviour.

It is often useful to model a new text in terms of the term counts from prior observations instead of some unigram statistics, $\vec{p}$. This can be done using the Dirichlet pseudo-counts hyperparameter and marginalising out the multinomial parameters $\vec{p}$:

$$p(\mathcal{W}|\vec{\alpha}) = \int_{\vec{p}\in\mathcal{P}} p(\mathcal{W}|\vec{p})\; p(\vec{p}|\vec{\alpha})\,\mathrm{d}^V\vec{p} \tag{48}$$

Compared to the binary case in Eq. 30, the integration limits are not [0,1] any more, as the formulation of the multinomial distribution does not explicitly include the probability normalisation constraint $\sum_k p_k{=}1$. With this constraint added, the integration domain $\mathcal{P}$ becomes a plane $(K-1)$-simplex embedded in the K-dimensional space that is bounded by the lines connecting points $p_k{=}1$ on the axis of each dimension k – see Fig. 3 for three dimensions.[13]

$$p(\mathcal{W}|\vec{\alpha}) = \int_{\vec{p}\in\mathcal{P}} \prod_{n=1}^{N} \mathrm{Mult}(W{=}w_n|\vec{p},1)\;\mathrm{Dir}(\vec{p}|\vec{\alpha})\,\mathrm{d}\vec{p} \tag{49}$$

$$= \int_{\vec{p}\in\mathcal{P}} \prod_{v=1}^{V} p_v^{n^{(v)}} \frac{1}{\Delta(\vec{\alpha})} \prod_{v=1}^{V} p_v^{\alpha_v-1}\,\mathrm{d}^V\vec{p} \tag{50}$$

$$= \frac{1}{\Delta(\vec{\alpha})} \int_{\vec{p}\in\mathcal{P}} \prod_{v=1}^{V} p_v^{n^{(v)}+\alpha_v-1}\,\mathrm{d}^V\vec{p} \qquad \Big|\ \text{Dirichlet}\int \tag{51}$$

$$= \frac{\Delta(\vec{n}+\vec{\alpha})}{\Delta(\vec{\alpha})}, \quad \vec{n} = \{n^{(v)}\}_{v=1}^{V} \tag{52}$$

Similar to the beta–Bernoulli case, the result states a distribution over terms observed as words given a pseudo-count of terms already observed, without any other statistics. More importantly, a similar marginalisation of a parameter is central for the formulation of posterior inference in LDA further below. The distribution in Eq. 52 has also been called the Dirichlet–multinomial distribution or Pólya distribution.

4 Bayesian networks and generative models

Bayesian networks (BNs) are a formal graphical language to express the behaviour (the joint distribution) of a system model in terms of random variables

[13] In the calculation, we use the Dirichlet integral of the first kind (over simplex $\mathcal{T}$):

$$\int_{\vec{t}\in\mathcal{T}} f(\sum_i^N t_i) \prod_i^N t_i^{\alpha_i-1}\,\mathrm{d}^N\vec{t} = \underbrace{\frac{\prod_i^N \Gamma(\alpha_i)}{\Gamma(\sum_i^N \alpha_i)}}_{\Delta(\vec{\alpha})} \int_0^1 f(\tau)\tau^{(\sum_i^N \alpha_i)-1}\,\mathrm{d}\tau$$

and their dependencies. They are a special case of Graphical Models, an important methodology in machine learning [3].

By only considering the most relevant dependency relations, inference calculations are considerably simplified – compared to assuming dependency between all variables, which is exponentially complex w.r.t. their number.

A Bayesian network forms a directed acyclical graph (DAG) with nodes that correspond to random variables and edges that correspond to conditional probability distributions, where the condition variable at the origin of an edge is called a parent node and the dependent variable at the end of the edge a child node. Bayesian networks distinguish between observed variables, which correspond to evidence nodes, and hidden variables, which correspond to hidden nodes. In Fig. 4, two BNs are presented.

Independence between variables can be determined from the topology of the graph. However, the relevant indepencence property is *conditional* independence: Two variables X and Y are conditionally independent given a condition Z, symbolically $X \perp Y|Z$, if $p(X, Y|Z) = p(X|Z) \cdot p(Y|Z)$. A verbal explanation of this conditional independence is that knowing Z, any information about the variable X does not add to the information about Y and vice versa. Here information can consist either of observations or parameters.

In a Bayesian network, the general rule is that a variable is conditionally independent of all variables outside of its Markov blanket, a subgraph of the BN defined as the set of a node's parents, its children, and its children's parents (co-parents). Within the Markov blanket of a node, conditional independence can be determined by assessing the criterion of new information from above. Alternatively, there is a method called "Bayes ball" that formalises these considerations [3].[14]

In many models, conditionally independent replications of nodes exist that share parents and/or children, e.g., to account for multiple values or mixture components. In this case, conditional independence is equivalent to exchangeability, which is the property of a set of nodes n_i to be invariant to permutation of their indices i. As a graphical representation, so-called plates exist, which surround the subset of exchangeable nodes and have a replication count.

All elements of the graphical language can be seen in the Dirichlet–multinomial model shown in the last section whose corresponding BN is shown in Fig. 4(a). The double circle around the variable $\vec{w}=\{w_{m,n}\}$ denotes an evidence node, i.e., an observed variable, and the surrounding plate indicates the N i.i.d. trials. The unknown variables $\vec{p}$ and $\vec{\alpha}$ can be distinguished into a multivariate parameter $\vec{\alpha}$, which is to be estimated, and a hidden variable $\vec{p}$, which is

[14] We can summarise rules equivalent to Bayes ball for common settings: Consider adding a new node to a set of independent nodes. Then, looking at the new structure, conditional dependence given the new node is introduced (1) for the parents of a new evidence node, (2) for the children of a new hidden node and (3) for the neighbours of a new hidden transitive node (with a parent and a child node). For the same situations, conditional independence given the new node is preserved if we exchange between new node status "evidence" and "hidden".

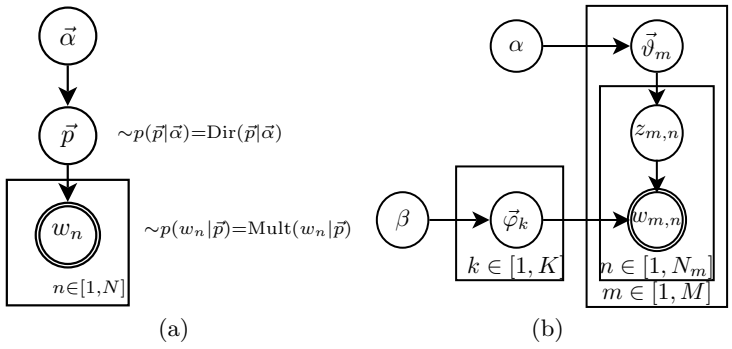

Fig. 4. Bayesian networks (a) of the Dirichlet–multinomial unigram model, and (b) of latent Dirichlet allocation.

to be handled in some way. The network shows that the observations $\vec{w}$ and hyperparameters $\vec{\alpha}$ are conditionally independent given the parameters $\vec{p}$.

Generative models. The advantage of Bayesian networks is that they provide an often intuitive description of an observed phenomenon as a so-called generative model, which states how the observations could have been generated by sampling values and propagating these along the directed edges of the network. Variable dependencies and edges can often be justified by causal relationships which re-enact a real phenomenon or are used as a artificial variables.

For the simple case of the Dirichlet–multinomial model, the generative model of a unigram (word) looks as follows:

$$\vec{p} \sim \mathrm{Dir}(p|\alpha) \tag{53}$$

$$w \sim \mathrm{Mult}(w|\vec{p}) \tag{54}$$

This means, a vector of parameters $\vec{p}$ is sampled from a Dirichlet distribution, and afterwards a word w is sampled from the multinomial with parameters $\vec{p}$. The task of Bayesian inference is to "invert" generative models and "generate" parameter values from given observations, trying to cope with any hidden variables. For the example model, this has been shown in Eq. 52, where the hidden variable $\vec{p}$ was handled by integrating it out. However, only in special cases is it possible to derive the complete posterior this way, and in the next section we will see how inference in a more complex model like LDA can be done.

5 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) by Blei et al. [2] is a probabilistic generative model that can be used to estimate the properties of multinomial observations by unsupervised learning. With respect to text modelling, LDA is a method to perform so-called latent semantic analysis (LSA). The intuition behind LSA is

to find the latent structure of "topics" or "concepts" in a text corpus, which captures the meaning of the text that is imagined to be obscured by "word choice" noise. The term latent semantic analysis has been coined by Deerwester et al. [4] who empirically showed that the co-occurrence structure of terms in text documents can be used to recover this latent topic structure, notably without any usage of background knowledge. In turn, latent-topic representations of text allow modelling of linguistic phenomena like synonymy and polysemy. This allows information retrieval systems to represent text in a way suitable for matching user needs (queries) with content items on a meaning level rather than by lexical congruence.

LDA is a model closely linked to the probabilistic latent semantic analysis (PLSA) by Hofmann [1], an application of the latent aspect method to the latent semantic analysis task. More specifically, LDA extends PLSA method by defining a complete generative model [2], and Girolami and Kaban showed that LDA with a uniform prior Dir(1) is a full Bayesian estimator for the same model for which PLSA provides an ML or MAP estimator [5].

Mixture modelling. LDA is a mixture model, i.e., it uses a convex combination of a set of component distributions to model observations. A convex combination is a weighted sum whose weighting proportion coefficients sum to one. In LDA, a word w is generated from a convex combination of topics z. In such a mixture model, the probability that a word w instantiates term t is:

$$p(w{=}t) = \sum_k p(w{=}t|z{=}k)p(z{=}k), \quad \sum_k p(z{=}k) = 1 \tag{55}$$

where each mixture component $p(w{=}t|z{=}k)$ is a multinomial distribution over terms (cf. the unigram model above) that corresponds to one of the latent topics $z{=}k$ of the text corpus. The mixture proportion consists of the topic probabilities $p(z{=}k)$. However, LDA goes a step beyond a global topic proportion and conditions the topic probabilities on the document a word belongs to. Based on this, we can formulate the main objectives of LDA inference: to find (1) the term distribution $p(t|z{=}k) = \vec{\varphi}_k$ for each topic k and (2) the topic distribution $p(z|d{=}m) = \vec{\vartheta}_m$ for each document m. The estimated parameter sets $\underline{\Phi} = \{\vec{\varphi}_k\}_{k=1}^K$ and $\underline{\Theta} = \{\vec{\vartheta}_m\}_{m=1}^M$ are the basis for latent-semantic representation of words and documents.

Generative model. To derive an inference strategy, we view LDA as a generative model. Consider the Bayesian network of LDA shown in Fig. 4(b). This can be interpreted as follows: LDA generates a stream of observable words $w_{m,n}$, partitioned into documents $\vec{w}_m$. For each of these documents, a topic proportion $\vec{\vartheta}_m$ is drawn, and from this, topic-specific words are emitted. That is, for each word, a topic indicator $z_{m,n}$ is sampled according to the document-specific mixture proportion, and then the corresponding topic-specific term distribution $\vec{\varphi}_{z_{m,n}}$ used to draw a word. The topics $\vec{\varphi}_k$ are sampled once for the entire corpus.

Because LDA leaves flexibility to assign a different topic to every observed word (and a different proportion of topics for every document), the model is

⊡ "topic plate"
for all topics $k \in [1, K]$ **do**
  sample mixture components $\vec{\varphi}_k \sim \mathrm{Dir}(\vec{\beta})$
end for
⊡ "document plate":
for all documents $m \in [1, M]$ **do**
  sample mixture proportion $\vec{\vartheta}_m \sim \mathrm{Dir}(\vec{\alpha})$
  sample document length $N_m \sim \mathrm{Poiss}(\xi)$
  ⊡ "word plate":
  for all words $n \in [1, N_m]$ in document m **do**
    sample topic index $z_{m,n} \sim \mathrm{Mult}(\vec{\vartheta}_m)$
    sample term for word $w_{m,n} \sim \mathrm{Mult}(\vec{\varphi}_{z_{m,n}})$
  end for
end for

Fig. 5. Generative model for latent Dirichlet allocation

not only referred to as a mixture model, but in fact as an admixture model. In genetics, admixture refers to a mixture whose components are itself mixtures of different features. Bayesian modelling of admixture for discrete data was notably done by Pritchard et al. [6] to model population genetics before LDA was proposed for text.

The complete (annotated) generative model [2] is presented in Fig. 14. Fig. 5 gives a list of all involved quantities.

M number of documents to generate (const scalar).
K number of topics / mixture components (const scalar).
V number of terms t in vocabulary (const scalar).
$\vec{\alpha}$ hyperparameter on the mixing proportions (K-vector or scalar if symmetric).
$\vec{\beta}$ hyperparameter on the mixture components (V-vector or scalar if symmetric).
$\vec{\vartheta}_m$ parameter notation for $p(z|d{=}m)$, the topic mixture proportion for document m. One proportion for each document, $\underline{\Theta} = \{\vec{\vartheta}_m\}_{m=1}^{M}$ ($M \times K$ matrix).
$\vec{\varphi}_k$ parameter notation for $p(t|z{=}k)$, the mixture component of topic k. One component for each topic, $\underline{\Phi} = \{\vec{\varphi}_k\}_{k=1}^{K}$ ($K \times V$ matrix).
N_m document length (document-specific), here modelled with a Poisson distribution [2] with constant parameter ξ.
$z_{m,n}$ mixture indicator that chooses the topic for the nth word in document m.
$w_{m,n}$ term indicator for the nth word in document m.

Fig. 6. Quantities in the model of latent Dirichlet allocation

Likelihoods. According to the model, the probability that a word $w_{m,n}$ instantiates a particular term t given the LDA parameters is:

$$p(w_{m,n}{=}t|\vec{\vartheta}_m, \underline{\Phi}) = \sum_{k=1}^{K} p(w_{m,n}{=}t|\vec{\varphi}_k) p(z_{m,n}{=}k|\vec{\vartheta}_m), \tag{56}$$

which is just another formulation of the mixture model in Eq. 55 and corresponds to one iteration on the word plate of the Bayesian network. From the topology of the Bayesian network, we can further specify the complete-data likelihood of a document, i.e., the joint distribution of all known and hidden variables given the hyperparameters:

$$p(\vec{w}_m, \vec{z}_m, \vec{\vartheta}_m, \underline{\Phi}|\vec{\alpha}, \vec{\beta}) = \overbrace{\underbrace{\prod_{n=1}^{N_m} p(w_{m,n}|\vec{\varphi}_{z_{m,n}}) p(z_{m,n}|\vec{\vartheta}_m)}_{\text{word plate}} \cdot p(\vec{\vartheta}_m|\vec{\alpha})}^{\text{document plate (1 document)}} \cdot \underbrace{p(\underline{\Phi}|\vec{\beta})}_{\text{topic plate}} . \tag{57}$$

To specify this distribution is often simple and useful as a basis for other derivations. So we can obtain the likelihood of a document $\vec{w}_m$, i.e., of the joint event of all words occurring, as one of its marginal distributions by integrating out the distributions $\vec{\vartheta}_m$ and $\underline{\Phi}$ and summing over $z_{m,n}$:

$$p(\vec{w}_m|\vec{\alpha}, \vec{\beta}) = \iint p(\vec{\vartheta}_m|\vec{\alpha}) \cdot p(\underline{\Phi}|\vec{\beta}) \cdot \prod_{n=1}^{N_m} \sum_{z_{m,n}} p(w_{m,n}|\vec{\varphi}_{z_{m,n}}) p(z_{m,n}|\vec{\vartheta}_m) \, \mathrm{d}\underline{\Phi} \, \mathrm{d}\vec{\vartheta}_m \tag{58}$$

$$= \iint p(\vec{\vartheta}_m|\vec{\alpha}) \cdot p(\underline{\Phi}|\vec{\beta}) \cdot \prod_{n=1}^{N_m} p(w_{m,n}|\vec{\vartheta}_m, \underline{\Phi}) \, \mathrm{d}\underline{\Phi} \, \mathrm{d}\vec{\vartheta}_m \tag{59}$$

Finally, the likelihood of the complete corpus $\mathcal{W} = \{\vec{w}_m\}_{m=1}^{M}$ is determined by the product of the likelihoods of the independent documents:

$$p(\mathcal{W}|\vec{\alpha}, \vec{\beta}) = \prod_{m=1}^{M} p(\vec{w}_m|\vec{\alpha}, \vec{\beta}). \tag{60}$$

Inference via Gibbs sampling. Although latent Dirichlet allocation is still a relatively simple model, exact inference is generally intractable. The solution to this is to use approximate inference algorithms, such as mean-field variational expectation maximisation [2], expectation propagation [7], and Gibbs sampling [8,9,6].

Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC) simulation [10,11] and often yields relatively simple algorithms for approximate inference in high-dimensional models such as LDA. Therefore we select this approach and present a derivation that is more complete than the original one by

Griffiths and Steyvers [8,9]. An alternative approach to Gibbs sampling in an LDA-like model is due Pritchard et al. [6] that actually pre-empted LDA in its interpretation of admixture modelling and formulated a direct Gibbs sampling algorithm for a model comparable to Bayesian PLSA [15].

MCMC methods can emulate high-dimensional probability distributions $p(\vec{x})$ by the stationary behaviour of a Markov chain. This means that one sample is generated for each transition in the chain after a stationary state of the chain has been reached, which happens after a so-called "burn-in period" that eliminates the influence of initialisation parameters. Gibbs sampling is a special case of MCMC where the dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, which we denote $\vec{x}_{\neg i}$. The algorithm works as follows:

1. choose dimension i (random or by permutation[16])
2. sample x_i from $p(x_i|\vec{x}_{\neg i})$.

To build a Gibbs sampler, the univariate conditionals (or full conditionals) $p(x_i|\vec{x}_{\neg i})$ must be found, which is possible using:

$$p(x_i|\vec{x}_{\neg i}) = \frac{p(\vec{x})}{\int p(\vec{x})\,\mathrm{d}x_i} \text{ with } \vec{x} = \{x_i, \vec{x}_{\neg i}\} \tag{61}$$

For models that contain hidden variables $\vec{z}$, their posterior given the evidence, $p(\vec{z}|\vec{x})$, is a distribution commonly wanted. With Eq. 61, the general formulation of a Gibbs sampler for such latent-variable models becomes:

$$p(z_i|\vec{z}_{\neg i}, \vec{x}) = \frac{p(\vec{z}, \vec{x})}{\int_Z p(\vec{z}, \vec{x})\,\mathrm{d}z_i}, \tag{62}$$

where the integral changes to a sum for discrete variables. With a sufficient number of samples $\tilde{\vec{z}}_r$, $r \in [1, R]$, the latent-variable posterior can be approximated using:

$$p(\vec{z}|\vec{x}) \approx \frac{1}{R} \sum_{r=1}^{R} \delta(\vec{z} - \tilde{\vec{z}}_r), \tag{63}$$

with the Kronecker delta $\delta(\vec{u}) = \{1 \text{ if } \vec{u}{=}0;\ 0 \text{ otherwise}\}$.

Gibbs sampler derivation. To develop a Gibbs sampler for LDA, we apply the hidden-variable method from above. The hidden variables in our model are $z_{m,n}$, i.e., the topics that appear with the words of the corpus $w_{m,n}$. We do not need to include the parameter sets $\underline{\Theta}$ and $\underline{\Phi}$ because they are just the statistics of the associations between the observed $w_{m,n}$ and the corresponding $z_{m,n}$, the state variables of the Markov chain.

[15] This work is lesser known in the text modelling field due to its application in genetics, which uses different notation and terminology.

[16] Liu [11] calls these variants random-scan and systematic-scan Gibbs samplers.

⊡ initialisation
zero all count variables, $n_m^{(z)}, n_m, n_z^{(t)}, n_z$
for all documents $m \in [1, M]$ **do**
  for all words $n \in [1, N_m]$ in document m **do**
    sample topic index $z_{m,n} \sim \text{Mult}(1/K)$
    increment document–topic count: $n_m^{(z)} + 1$
    increment document–topic sum: $n_m + 1$
    increment topic–term count: $n_z^{(t)} + 1$
    increment topic–term sum: $n_z + 1$
  end for
end for
⊡ Gibbs sampling over burn-in period and sampling period
while not finished **do**
  for all documents $m \in [1, M]$ **do**
    for all words $n \in [1, N_m]$ in document m **do**
      ⊡ for the current assignment of z to a term t for word $w_{m,n}$:
      decrement counts and sums: $n_m^{(z)} - 1; n_m - 1; n_z^{(t)} - 1; n_z - 1$
      ⊡ multinomial sampling acc. to Eq. 79 (decrements from previous step):
      sample topic index $\tilde{z} \sim p(z_i | \vec{z}_{\neg i}, \vec{w})$
      ⊡ use the new assignment of z to the term t for word $w_{m,n}$ to:
      increment counts and sums: $n_m^{(\tilde{z})} + 1; n_m + 1; n_{\tilde{z}}^{(t)} + 1; n_{\tilde{z}} + 1$
    end for
  end for
  ⊡ check convergence and read out parameters
  if converged and L sampling iterations since last read out **then**
    ⊡ the different parameters read outs are averaged.
    read out parameter set $\underline{\Phi}$ according to Eq. 84
    read out parameter set $\underline{\Theta}$ according to Eq. 85
  end if
end while

Fig. 7. Gibbs sampling algorithm for latent Dirichlet allocation

We first show the complexity of calculating $p(\vec{z}|\vec{w})$. This distribution is directly proportional to the joint distribution:

$$p(\vec{z}|\vec{w}) = \frac{p(\vec{w}, \vec{z})}{\sum_z p(\vec{w}, \vec{z})} \tag{64}$$

where we use unindexed vectors $\vec{w}$ and $\vec{z}$ to denote the complete corpus and omitted the hyperparameters. The distribution covers a large space of discrete random variables, and the difficult part for evaluation is its denominator, which represents a summation over K^W terms [9]. At this point, the Gibbs sampling procedure comes into play. In our setting, the desired Gibbs sampler runs a Markov chain that uses the full conditional $p(z_i|\vec{z}_{\neg i}, \vec{w})$ in order to simulate $p(\vec{z}|\vec{w})$. We can obtain the full conditional via the hidden-variable approach by

evaluating Eq. 62, which requires to formulate the joint distribution. In LDA, this joint distribution can be factored:

$$p(\vec{w}, \vec{z}|\vec{\alpha}, \vec{\beta}) = p(\vec{w}|\vec{z}, \vec{\beta})p(\vec{z}|\vec{\alpha}), \tag{65}$$

because the first term is independent of $\vec{\alpha}$ due to the conditional independence of $\vec{w}$ and α given z, and the second term is independent of $\vec{\beta}$. Both elements of the joint distribution can now be handled separately. The first term, $p(w|z)$, can be derived from a multinomial on the observed word counts given the associated topics:

$$p(\vec{w}|\vec{z}, \underline{\Phi}) = \prod_{i=1}^{W} \varphi_{z_i, w_i}. \tag{66}$$

That is, the W words of the corpus are observed according to independent multinomial trials with parameters conditioned on the topic indices z_i. We can now split the product over words into one product over topics and one over the vocabulary, separating the contributions of the topics:

$$p(\vec{w}|\vec{z}, \underline{\Phi}) = \prod_{z=1}^{K} \prod_{t=1}^{V} \varphi_{z,t}^{n_z^{(t)}}, \tag{67}$$

where we use the notation $n_z^{n^{(t)}}$ to denote the number of times that term t has been observed with topic z. The target distribution $p(\vec{w}|\vec{z}, \vec{\beta})$ is obtained by integrating over $\underline{\Phi}$, which can be done componentwise using Dirichlet integrals within the product over z:

$$p(\vec{w}|\vec{z}, \vec{\beta}) = \int p(\vec{w}|\vec{z}, \underline{\Phi})\ p(\underline{\Phi}|\vec{\beta})\ \mathrm{d}\underline{\Phi} \tag{68}$$

$$= \int \prod_{z=1}^{K} \frac{1}{\Delta(\vec{\beta})} \prod_{t=1}^{V} \varphi_{z,t}^{n_z^{(t)}+\beta_t-1} \mathrm{d}\vec{\varphi}_z \tag{69}$$

$$= \prod_{z=1}^{K} \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})}, \quad \vec{n}_z = \{n_z^{(t)}\}_{t=1}^{V}. \tag{70}$$

This can be interpreted as a product of K Dirichlet–multinomial models (cf. Eq. 52), representing the corpus by K separate "topic texts".

Analogous to $p(\vec{w}|\vec{z}, \vec{\beta})$, the topic distribution $p(\vec{z}|\vec{\alpha})$ can be derived, starting with the conditional and rewriting its parameters into two products, separating the contributions of the documents:

$$p(\vec{z}|\underline{\Theta}) = \prod_{i=1}^{W} \vartheta_{d_i, z_i} = \prod_{m=1}^{M} \prod_{z=1}^{K} \vartheta_{m,z}^{n_m^{(z)}}, \tag{71}$$

where the notation d_i refers to the document a word i belongs to and $n_m^{(z)}$ refers to the number of times that topic z has been observed with a word of document

m. Integrating out $\underline{\Theta}$, we obtain:

$$p(\vec{z}|\vec{\alpha}) = \int p(\vec{z}|\underline{\Theta})\, p(\underline{\Theta}|\vec{\alpha})\ \mathrm{d}\underline{\Theta} \tag{72}$$

$$= \int \prod_{m=1}^{M} \frac{1}{\Delta(\vec{\alpha})} \prod_{z=1}^{K} \vartheta_{m,z}^{n_m^{(z)}+\alpha_z-1}\, \mathrm{d}\vec{\vartheta}_m \tag{73}$$

$$= \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}, \quad \vec{n}_m = \{n_m^{(z)}\}_{z=1}^{K}. \tag{74}$$

The joint distribution therefore becomes:

$$p(\vec{z}, \vec{w}|\vec{\alpha}, \vec{\beta}) = \prod_{z=1}^{K} \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}. \tag{75}$$

From this, we can derive the update equation for the hidden variable:[17]

$$p(z_i|\vec{z}_{\neg i}, \vec{w}) = \frac{p(\vec{z}, \vec{w})}{p(\vec{z}_{\neg i}, \vec{w})} = \frac{p(\vec{w}|\vec{z})}{p(\vec{w}|\vec{z}_{\neg i})} \cdot \frac{p(\vec{z})}{p(\vec{z}_{\neg i})} \tag{76}$$

$$= \frac{\Delta(\vec{n}_z + \vec{\beta}) \cdot \Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{n}_{z,\neg i} + \vec{\beta}) \cdot \Delta(\vec{n}_{m,\neg i} + \vec{\alpha})} \tag{77}$$

$$= \frac{\frac{\Gamma(n_z^{(t)}+\beta_t)}{\Gamma(\sum_{v=1}^{V} n_z^{(v)}+\beta_v)} \cdot \frac{\Gamma(n_m^{(z)}+\alpha_z)}{\Gamma(\sum_{z=1}^{K} n_m^{(z)}+\alpha_z)}}{\frac{\Gamma(n_z^{(t)}-1+\beta_t)}{\Gamma([\sum_{t=1}^{V} n_z^{(t)}+\beta_t]-1)} \cdot \frac{\Gamma(n_m^{(z)}-1+\alpha_z)}{\Gamma([\sum_{z=1}^{K} n_m^{(z)}+\alpha_z]-1)}} \tag{78}$$

$$= \frac{n_{z,\neg i}^{(t)} + \beta_t}{[\sum_{v=1}^{V} n_z^{(v)} + \beta_v] - 1} \cdot \frac{n_{m,\neg i}^{(z)} + \alpha_z}{[\sum_{z=1}^{K} n_m^{(z)} + \alpha_z] - 1} \tag{79}$$

where the counts $n_{\cdot,\neg i}^{(\cdot)}$ indicate that the word or topic with index i is excluded.

Finally, we need to relate the required parameters $\underline{\Theta}$ and $\underline{\Phi}$ to the statistics of the Markov chain. We can do this by predicting the distribution of topics for new observations based on the current state of the Markov chain. More specifically, we evaluate the distribution of a new topic–word pair ($\tilde{z}{=}k, \tilde{w}{=}t$) that is observed

[17] Considering the distribution in Eq. 75, only the terms of the products over m and z remain that contain the index i, all others cancel out. Further, the identity $\Gamma(x) = (x-1)\Gamma(x-1)$ is used.

in a document $d(\tilde{w})=d(\tilde{z})=m$, given the state $(\vec{z},\vec{w})$:[18]

$$p(\tilde{z}=k,\tilde{w}=t|\vec{z},\vec{w}) = p(\tilde{w}=t|\tilde{z}=k;\vec{z},\vec{w})\cdot p(\tilde{z}=k|\vec{z},\vec{w}) \tag{80}$$

$$= \frac{p(\tilde{z}=k,\tilde{w}=t;\vec{z},\vec{w})}{p(\vec{z},\vec{w})} \tag{81}$$

$$= \frac{\frac{\Gamma(n_k^{(t)}+1+\beta_t)}{\Gamma(\sum_{v=1}^{V} n_k^{(v)}+1+\beta_v)}\cdot\frac{\Gamma(n_m^{(k)}+1+\alpha_k)}{\Gamma(\sum_{z=1}^{K} n_m^{(z)}+1+\alpha_z)}}{\frac{\Gamma(n_k^{(t)}+\beta_t)}{\Gamma(\sum_{v=1}^{V} n_k^{(v)}+\beta_v)}\cdot\frac{\Gamma(n_m^{(k)}+\alpha_k)}{\Gamma(\sum_{z=1}^{K} n_m^{(z)}+\alpha_z)}} \tag{82}$$

$$= \frac{n_k^{(t)}+\beta_t}{\sum_{v=1}^{V} n_k^{(v)}+\beta_v}\cdot\frac{n_m^{(k)}+\alpha_k}{\sum_{z=1}^{K} n_m^{(z)}+\alpha_z}. \tag{83}$$

With the definitions $\underline{\Phi} = p(\tilde{w}|\tilde{z})$ and $\underline{\Theta} = p(\tilde{z}|d(\tilde{z}))$, we can interpret the factor $p(\tilde{w}=t|\tilde{z}=k;\vec{z},\vec{w})$ as the parameter $\varphi_{k,t}$ and the factor $p(\tilde{z}=k|\vec{z},\vec{w})$ as the parameter $\vartheta_{m,k}$, where m is implicitly given as the document $d(\tilde{z})$:

$$\varphi_{k,t} = \frac{n_k^{(t)}+\beta_t}{\sum_{v=1}^{V} n_k^{(v)}+\beta_v}, \tag{84}$$

$$\vartheta_{m,k} = \frac{n_m^{(k)}+\alpha_k}{\sum_{z=1}^{K} n_m^{(z)}+\alpha_z}. \tag{85}$$

Using Eqs. 79, 84 and 85, the Gibbs sampling procedure in Fig. 30 can be run. The procedure itself uses only five larger data structures, the count variables $n_m^{(z)}$ and $n_z^{(t)}$, which have dimension $M\times K$ and $K\times V$ respectively, their row sums n_m and n_z with dimension M and K, as well as the state variable $z_{m,n}$ with dimension W.[19] The Gibbs sampling algorithm runs over the three periods initialisation, burn-in and sampling. However, to determine the required lengths of the burn-in is one of the drawbacks with MCMC approaches. There are several criteria to check that the Markov chain has converged (see [11]), and we manually check how well the parameters cluster semantically related words and documents for different corpora and use these values as estimates for comparable settings. When reading out the parameters, there are several approaches. One is to just use only one read out, another is to average a number of samples, and often it is desirable to leave an interval of L iteration between subsequent read-outs to obtain decorrelated states of the Markov chain. This interval is often called "thin interval" or sampling lag.

6 Analysing topic models

Topic models such as LDA estimate soft associations between latent topics and observed entities, i.e., words, documents, but in model extensions also authors

[18] Cf. Eq. 79. Here only the counts of the current topic–word and document–topic associations (k,t) and (m,k) create terms that don't cancel out.

[19] The sum n_m is just the document length.

etc. These associations are tbe basis for a number of operations relevant to information processing and language modelling. We will in this section outline methods to analyse the topic structure in order (1) to estimate the topic structure of unseen documents (querying), (2) to estimate the quality of the clustering implied by the estimated topics and (3) to infer new associations on the basis of the estimated ones, e.g., the similarity between words or between documents or their authors. For this, the examplary case of LDA is used, which provides information about the topics present in documents – the parameter set $\underline{\Theta}$ –, the content of topics in terms of words – the parameter set $\underline{\Phi}$.

Querying can be considered topic estimation of unknown documents (queries) and comparing these topic distributions to those of the known documents. Appropriate similarity measures permit ranking. A query is simply a vector of words $\tilde{\vec{w}}$, and we can find matches with known documents by estimating the posterior distribution of topics $\tilde{\vec{z}}$ given the word vector of the query $\tilde{\vec{w}}$ and the LDA model $L(\underline{\Theta}, \underline{\Phi})$: $p(\tilde{\vec{z}}|\tilde{\vec{w}}, L) = p(\tilde{\vec{z}}|\tilde{\vec{w}}, \vec{w}, \vec{z})$. Considering $\tilde{\vec{w}}$ a document $\tilde{m}$, this is the same as the right term of the predictive likelihood in Eq. 83 and consequently Eq. 85 for document $\tilde{m}$. In order to find the required counts for a complete new document, we can follow the approach of [1] or [12] to run the inference algorithm on the new document exclusively, similar to Eq. 79. We first initialise the algorithm by randomly assigning topics to words and then perform a number of loops through the Gibbs sampling update (locally for the words i of $\tilde{m}$):[20]

$$p(\tilde{z}_i{=}k|\tilde{\vec{z}}_{\neg_i}, \tilde{\vec{w}}; \vec{z}_{\neg i}, \vec{w}) = \\ \frac{n_k^{(t)} + \tilde{n}_{k,\neg i}^{(t)} + \beta_t}{[\sum_{v=1}^{V} n_k^{(v)} + \tilde{n}_k^{(v)} + \beta_v] - 1} \cdot \frac{n_{\tilde{m}}^{(k,\neg i)} + \alpha_k}{[\sum_{z=1}^{K} n_{\tilde{m}}^{(z)} + \alpha_z] - 1} \tag{86}$$

where the new variable $\tilde{n}_k^{(t)}$ counts the observations of term t and topic k in the unseen document. This equation gives a colourful example of the workings of Gibbs posterior sampling: High estimated word–topic associations $n_k^{(t)}$ will dominate the multinomial masses compared to the contributions of $\tilde{n}_k^{(t)}$ and $n_{\tilde{m}}^{(k)}$, which are chosen randomly and therefore unlikely to be clustered. Consequently, on repeatedly sampling from the distribution and updating of $n_{\tilde{m}}^{(k)}$, the masses of topic–word associations are propagated into document–topic associations. Note the smoothing influence of the Dirichlet hyperparameters.

Applying Eq. 85 gives the topic distribution for the unknown document:

$$\vartheta_{\tilde{m},k} = \frac{n_{\tilde{m}}^{(k)} + \alpha_k}{\sum_{z=1}^{K} n_{\tilde{m}}^{(z)} + \alpha_z}. \tag{87}$$

This querying procedure is applicable for complete collections of unknown documents, which is done by letting $\tilde{m}$ range over the unknown documents.

[20] The formulation in Eq. 86 can as well be related to the parameters of the model, $\underline{\Phi}$ and $\underline{\Theta}$, using Eqs. 84 and 85.

As the distribution over topics $\vartheta_{\tilde{m}}$ now is in the same form as the definition of $\underline{\Theta}$, we can compare the query term to the documents of the corpus. A simple measure is the Kullback-Leibler divergence [13], which is defined between two discrete random variables X and Y, as[21] $\mathsf{KL}(X||Y) = \sum_{n=1}^{N} P(X = n)\,[\log_2 P(X = n) - \log_2 P(Y = n)]$.

Clustering. Often it is of importance to cluster documents or terms. As mentioned above, the LDA model already provides a soft clustering of the documents and of the terms of a corpus by associating them to topics. To use this clustering information requires the evaluation of similarity, and in the last section, the similarity between a query document and the corpus documents was computed using the Kullback Leibler divergence. This measure can be applied to the distributions of words over topics as well as to the distribution of topics over documents in general, which reveals the internal similarity pattern of the corpus according to its latent semantic structure.

In addition to determining similarities, the evaluation of clustering quality is of particular interest for topic models like LDA. In principle, evaluation can be done by subjective judgement of the estimated word and document similarities. A more objective evaluation, however, is the comparison of the estimated model to an a priori categorisation for a given corpus as a reference[22]. Among the different methods to compare clusterings, we will show the Variation of Information distance (VI-distance) that is able to calculate the distance between soft or hard clusterings of different numbers of classes and therefore provides maximum flexibility of application.

The VI distance measure has been proposed by Meila [14], and it assumes two distributions over classes for each document: $p(c{=}j|d_m)$ and $p(z{=}k|d_m)$ with class labels (or topics) $j \in [1, J]$ and $k \in [1, K]$. Averaging over the corpus yields the class probabilities $p(c{=}j) = 1/M \sum_m p(c{=}j|d_m)$ and $p(z{=}k) = 1/M \sum_m p(z{=}k|d_m)$.

Similar clusterings tend to have co-occurring pairs $(c{=}j, z{=}k)$ of high probability $p(\cdot|d_m)$. Conversely, dissimilarity corresponds to independence of the class distributions for all documents, i.e., $p(c{=}j, z{=}k) = p(c{=}j)p(z{=}k)$. To find the degree of similarity, we can now apply the Kullback-Leibler divergence between the real distribution and the distribution that assumes independence. In information theory, this corresponds to the mutual information of the random variables C and Z that describe the event of observing classes with documents in the two clusterings [14,15]:

$$\begin{aligned} I(C, Z) &= \mathsf{KL}\{p(c, z)||p(c)p(z)\} \\ &= \sum_{j=1}^{J}\sum_{k=1}^{K} p(c{=}j, z{=}k)[\log_2 p(c{=}j, z{=}k) - \log_2 p(c{=}j)p(z{=}k)] \end{aligned} \tag{88}$$

[21] KL is not a distance measure proper because it is not symmetric. Thus alternatively, a smoothed, symmetrised extension, the Jensen-Shannon divergence, can be used: $\mathsf{JS}(\vec{\xi}_i||\vec{\vartheta}_j) = \frac{1}{2}(\mathsf{KL}(\vec{\xi}_i||\vec{\mu}_i) + \mathsf{KL}(\vec{\vartheta}_i||\vec{\mu}_i))$ with $\vec{\mu}_{ij} = \frac{1}{2}(\vec{\vartheta}_i + \vec{\xi}_i)$.

[22] It is immediately clear that this is only as objective as the reference categorisation.

where the joint probability refers the corpus-wide average co-occurrence of class pairs in documents, $p(c{=}j, z{=}k) = \frac{1}{M}\sum_{m=1}^{M} p(c{=}j|d_m)p(z{=}k|d_m)$.

The mutual information between two random variables becomes 0 for independent variables. Further, $I(C,Z) \leq \min\{H(C), H(Z)\}$ where $H(C){=}-\sum_{j=1}^{J} p(c{=}j)\log_2 p(c{=}j)$ is the entropy of C. This inequality becomes an equality $I(C,Z){=}H(C){=}H(Z)$ if and only if the two clusterings are equal. Meila used these properties to define the Variation of Information cluster distance measure:

$$\mathsf{VI}(C,Z) = H(C) + H(Z) - 2I(C,Z) \tag{89}$$

and shows that $\mathsf{VI}(C,Z)$ is a true metric, i.e., is always non-negative, becomes zero if and only if $C{=}Z$, symmetric, and observes the triangle inequality, $\mathsf{VI}(C,Z){+}\mathsf{VI}(Z,X){\geq}\mathsf{VI}(C,X)$[14]. Further, the VI metric only depends on the proportions of cluster assciations with data items, i.e., it is invariant to the absolute numbers of data items.

An application of the VI distance to LDA has been shown in [15], where the document–topic associations $\underline{\Theta}$ of a corpus of between 20000 news stories are compared to IPTC categories assigned manually to them.

Perplexity. A common criterion of clustering quality that does not require a priori categorisations is perplexity, originally used in language modelling [16]. Perplexity is a measure of the ability of a model to generalise to unseen data. It is defined as the reciprocal geometric mean of the likelihood of a test corpus given the model:

$$\mathsf{P}(\tilde{\mathcal{W}}) = \prod_{m=1}^{M} p(\tilde{\vec{w}}_m)^{-\frac{1}{N}} = \exp -\frac{\sum_{m=1}^{M}\log p(\tilde{\vec{w}}_m)}{\sum_{m=1}^{M} N_m} \tag{90}$$

The predictive likelihood of a word vector can in principle be calculated by integrating out all parameters from the joint distribution of the word observations in a document. For LDA, the likelihood of a text document of the test corpus $p(\tilde{\vec{w}}_m)$ can be directly expressed as a function of the parameters:

$$p(\tilde{\vec{w}}_m) = \prod_{n=1}^{N_d}\sum_{k=1}^{K} p(w_n{=}t|z_n{=}k)\cdot p(z_n{=}k|d{=}m) \tag{91}$$

$$= \prod_{v=1}^{V}\left(\sum_{k=1}^{K}\varphi_{k,t}\cdot\vartheta_{m,k}\right)^{n_m^{(v)}} \tag{92}$$

where $n_m^{(v)}$ is the number of times term t has been observed in document m.

The common method to evaluate perplexity in topic models is to hold out test data from the corpus to be trained and then test the estimated model on the held-out data[23]. Higher values of perplexity indicate a higher misrepresentation of the words of the test documents by the trained topics.

[23] This is often referred to as cross-validation, and a common enhancement of this method is to choose mutually exclusive subsets of the corpus as hold-out data and average over the results.

Retrieval performance. Other standard quality metrics view topic models as information retrieval approaches, which requires that it be possible to rank items for a given query, i.e., an unknown document (see above). The most prominent retrieval measures are precision and recall [17]. Recall is defined as the ratio between the number of retrieved relevant items to the total number of existing relevant items. Precision is defined as the ratio between the number of relevant items and the total of retrieved items. The goal is to maximise both, but commonly they have antagonistic behaviour, i.e., trying to increase recall will likely reduce precision. To compare different systems, combinations of precision P and recall R metrics have been developed, such as the F_1 measure, $F_1 = 2PR/(P + R)$, which can also be generalised to a weighted F_1 measure, $F_w = (\lambda_P + \lambda_R)PR/(\lambda_P P + \lambda_R R)$. With the given weightings, the preferences to precision or recall can be adjusted. A direct relation between precision and recall to perplexity and language models has been given by Azzopardi et al. [16].

7 Conclusion

We have introduced the basic concepts of probabilistic estimation, such as the ML, MAP and full Bayesian parameter estimation approaches and have shown their behaviour in the domain of discrete data. We have further introduced the principle of conjugate distributions as well as the graphical language of Bayesian networks. With these theoretical preliminaries, we have presented the model of latent Dirichlet allocation (LDA) and a complete derivation of approximate inference via Gibbs sampling.

These models can be considered the basic building blocks of a general framework of probabilistic modeling of text and be used to develop more sophisticated and application-oriented models, such as hierarchical models, models that combine content and relational data (such as social networks) or models that include multimedia features that are modeled in the Gaussian domain.

References

1. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI'99, Stockholm (1999)
2. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. In: Advances in Neural Information Processing Systems 14, Cambridge, MA, MIT Press (2002)
3. Murphy, K.: An introduction to graphical models. Web (2001)
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
5. Girolami, M., Kaban, A.: On an equivalence between PLSI and LDA. In: Proc. of ACM SIGIR. (2003)
6. Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. Genetics **155** (2000) 945–959
7. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proc. UAI. (2002)

8. Griffiths, T.: Gibbs sampling in the generative model of Latent Dirichlet Allocation. Technical report, Stanford University (2002)
9. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences **101** (2004) 5228–5235
10. MacKay, D.J.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)
11. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer (2001)
12. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004)
13. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22** (1951) 79–86
14. Meila, M.: Comparing clusterings. In: Proc. 16th Ann. Conf. on Learn. Theory. (2003)
15. Heinrich, G., Kindermann, J., Lauth, C., Paaß, G., Sanchez-Monzon, J.: Investigating word correlation at different scopes—a latent concept approach. In: Workshop Lexical Ontology Learning at Int. Conf. Mach. Learning. (2005)
16. Azzopardi, L., Girolami, M., van Risjbergen, K.: Investigating the relationship between language model perplexity and IR precision-recall measures. In: Proc. SIGIR. (2003)
17. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press & Addison-Wesley (1999)