5.6. The Gibbs sampler is a correct sampler: given the conditional distributions of the target distribution, running it will converge to the target distribution. For two random variables X and Y, this means that the conditional distributions $p(X \mid Y)$ and $p(Y \mid X)$ uniquely identify the joint distribution $p(X, Y)$. Prove that analytically (i.e., show that $p(X, Y)$ can be expressed in terms of the conditional distributions). Hint: can you express $p(X)$ (or $p(Y)$) in terms of the conditionals?

CHAPTER 6

Variational Inference

In the previous chapter, we described some of the core algorithms used for drawing samples from the posterior, or more generally, from a probability distribution. In this chapter, we consider another approach to approximate inference—variational inference.

Variational inference treats the problem of identifying the posterior as an *optimization problem*. When this optimization problem is solved, the output is an approximate version of the posterior distribution. This means that the objective function that variational inference aims to optimize is a function over a family of distributions. The reason this is an *approximate* inference is that this family of distributions is usually not inclusive of the true posterior, and makes strong assumptions about the form of the posterior distribution.

The term "variational" here refers to concepts from mathematical analysis (such as the calculus of variations) which focus on the maximization and minimization of functionals (mappings from a set of functions to real numbers). This kind of analysis has been used frequently in physics (e.g., quantum mechanics). Very commonly, it is used in the context of minimizing energy through a functional that describes the state of physical elements.

Section 6.1 begins the discussion of variational inference in this chapter, by describing the basic variational bound used in variational inference. We then discuss mean-field variational inference, the main type of variational inference used in Bayesian NLP (Sections 6.2–6.3). We continue with a discussion of empirical Bayes estimation with variational approximations (Section 6.4). In the next section (Section 6.5), we discuss various topics related to variational inference in Bayesian NLP, covering topics such as initialization of variational inference algorithms, convergence diagnosis, variational inference decoding, the relationship between variational inference and KL minimization, and finally, online variational inference. We conclude with a summary (Section 6.6).

6.1 VARIATIONAL BOUND ON MARGINAL LOG-LIKELIHOOD

Consider a typical scenario in which the observations are represented by the random variables $X^{(1)}, \ldots, X^{(n)}$. These observations are (deterministic or probabilistic) functions of the latent structure $Z^{(1)}, \ldots, Z^{(n)}$. These latent structures are the targets for prediction.

On top of the latent structure and the observations, there is a prior $p(\theta|\alpha)$ over the parameters θ such that $\alpha \in A$ is the hyperparameter. This prior is a top-level prior (Section 3.5), but

Input: Observed data $x^{(1)}, \ldots, x^{(n)}$, a partition of the latent variables into $U_1, \ldots, U_p$ and a set of possible distributions for $U_1, \ldots, U_p$: $\mathcal{Q}_1, \ldots, \mathcal{Q}_p$.
Output: Factorized approximate posterior $q(U_1, \ldots, U_p)$.

Initialize $q^*(U_i)$ from $\mathcal{Q}_i$ for $i = 1, \ldots, p$
$q^*(U_1, \ldots, U_p) \leftarrow \left(\prod_{i=1}^{p} q^*(U_i)\right)$
repeat
  for $i \in \{1, \ldots, p\}$ **do**
    Set $\mathcal{Q}^* = \{q^*(U_1)\} \times \ldots \times \mathcal{Q}_i \times \ldots \times \{q^*(U_p)\}$
    $q^*(U_i) \leftarrow$ the factor $q(U_i)$ in

$$\arg\max_{q \in \mathcal{Q}^*} \mathcal{F}(q, x^{(1)}, \ldots, x^{(n)} | \alpha) \tag{6.7}$$

  end for
  $q^*(U_1, \ldots, U_p) \leftarrow \left(\prod_{i=1}^{p} q^*(U_i)\right)$
until the bound $\mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n)} | \alpha)$ converged
return q^*

Algorithm 6.1: The mean-field variational inference algorithm. Its input are observations, a partition of the random variables that the inference is done on, and a set of distribution families, one set per element in the partition. The algorithm then iterates (with iterator i) through the different elements in the partition, each time maximizing the variational bound for the observations with respect to $\mathcal{Q}_i$, while holding $q^*(U_j)$ for $j \neq i$ fixed.

The optimization problem in Equation 6.7 is not always easy to solve, but fortunately, the solution has a rather general formula. It can be shown that $q^*(U_i)$ maximizing Equation 6.7 equals:

$$q^*(U_i) = \frac{\exp\left(E_{q_{-i}}[\log p(\boldsymbol{X}, U_1, \ldots, U_p)]\right)}{Z_i}, \tag{6.8}$$

where q_{-i} is a distribution over U_{-i} defined as:

$$q_{-i}(U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_p) = \prod_{j \neq i} q(U_j),$$

and Z_i is a normalization constant that integrates or sums the numerator in Equation 6.8 with respect to U_i. For example, if U_i is discrete, then $Z_i = \sum_u E_{q_{-i}}[\log p(\boldsymbol{X}, U_1, \ldots, U_i = u, \ldots, U_p)]]$. This general derivation appears in detail in Bishop (2006), but in Section 6.3.1 we derive a specific case of this formulation for the Dirichlet-Multinomial family.

The following decisions need to be made by the modeler when actually implementing Algorithm 6.1.

- **Partitioning of the latent variables** This issue is discussed at length in Chapter 5 and in Section 6.2. The decision that the modeler has to make with respect to this issue is how to carve up the random variables into a set of random variables that have minimal interaction, or that offer some computational tractability for maximizing the variational bound with respect to each of members of this set of random variables.
- **Choosing the parametrization of each factor ($\mathcal{Q}_i$)** Determining the parametrization of each of the factors requires a balance between richness of the parametrization (so we are able to get a tighter bound) and tractability (see below). It is often the case that even when $\mathcal{Q}_i$ is left nonparametric (or when it includes the set of all possible distributions over the sample space of U_i), the solution to the coordinate ascent step is actually a distribution from a parametric family. Identifying this parametric family can be done as part of the derivation of the variational EM algorithm, so that the variational distributions can be represented computationally (and optimized with respect to their parameters, also referred to as "variational parameters").
- **Optimizing the bound at each step of the coordinate ascent** At each step of the mean-field variational inference algorithm, we have to find the factor in q that maximizes the variational bound while maintaining all other factors fixed according to their values from the previous iterations. If the parametrization of each factor is chosen carefully, then sometimes closed-form solutions for these mini-maximization problems are available (this is especially true when the prior is conjugate to the likelihood). It is also often the case that a nested optimization problem needs to be solved using optimization techniques such as gradient descent or Newton's method. Unfortunately, sometimes the nested optimization problem itself is a non-convex optimization problem.

Recent work, such as the work by Kucukelbir et al. (2016), tries to minimize the decision making which is not strictly related to modeling and data collection. The work by Kucukelbir et al. proposes an automatic variational inference algorithm, which uses automatic differentation, integrated into the Stan programming language Carpenter et al. (2015).

6.3.1 DIRICHLET-MULTINOMIAL VARIATIONAL INFERENCE

The following example demonstrates the decisions that the modeler has to make when deriving a variational inference algorithm. We derive in this section a mean-field variational inference algorithm which is commonly used for Dirichlet-Multinomial models.

Consider the case where the likelihood is a collection of multinomials, which is often the case with NLP models such as probabilistic context-free grammars and hidden Markov models.

In such a case, the model family is parametrized by $\theta = (\theta^1, \ldots, \theta^K)$ such that each θ^k is in the probability simplex of dimension $N_k - 1$ for some natural number N_k:

$$\theta_i^k \geq 0 \qquad \forall k \in \{1, \ldots, K\}, \forall i \in \{1, \ldots, N_k\}$$
$$\sum_{i=1}^{N_k} \theta_i^k = 1 \qquad \forall k \in \{1, \ldots, K\}.$$

For example, in the case of PCFGs, K would be the number of nonterminals in the grammar, N_k would be the number of rules for each nonterminal and θ_i^k would correspond to the probability of the ith rule for the kth nonterminal. See Section 8.2 for more details about this formulation. Let $f_i^k(x, z)$ be a function that counts the times that event i from multinomial k fires in (x, z).

The most common choice of a conjugate prior for this model is a product-of-Dirichlet distribution, such that:

$$p(\theta|\alpha) \propto \prod_{k=1}^{K} \prod_{i=1}^{N_k} (\theta_i^k)^{\alpha_i^k - 1},$$

with $\alpha = (\alpha^1, \ldots, \alpha^K)$ and $\alpha^k \in \mathbb{R}^{N_k}$ such that $\alpha_i^k \geq 0$ for all i and k.

Assuming a top-level prior, and with $X^{(1)}, \ldots, X^{(n)}$ being the observed random variables and $Z^{(1)}, \ldots, Z^{(n)}$ being the latent structures, the likelihood is:

$$\prod_{j=1}^{n} p(x^{(j)}, z^{(j)}|\theta) = \prod_{j=1}^{n} \prod_{k=1}^{K} \prod_{i=1}^{N_k} (\theta_i^k)^{f_i^k(x^{(j)}, z^{(j)})}$$
$$= \prod_{k=1}^{K} \prod_{i=1}^{N_k} (\theta_i^k)^{\sum_{j=1}^{n} f_i^k(x^{(j)}, z^{(j)})},$$

with $f_i^k(x, z)$ being the count of event i from multinomial k in the pair (x, z). We denote in short:

$$f_{k,i} = \sum_{j=1}^{n} f_i^k(x^{(j)}, z^{(j)}).$$

We will be interested in mean-field variational inference such that q remains nonparametric (for now), factorized into $q(\theta)$ and $q(\boldsymbol{Z})$. That, in essence, is the tightest approximation one can use, while assuming independent parameters and latent structures for the approximate posterior

family. In that case, the functional $\mathcal{F}(q, x^{(1)}, \ldots, x^{(n)}|\alpha)$, following Equation 6.3 (which gives the bound on the marginal log-likelihood), looks like the following:

$$\mathcal{F}(q, x^{(1)}, \ldots, x^{(n)}|\alpha)$$
$$= E_q\left[\log\left(p(\theta|\alpha) \times \prod_{k=1}^{K} \prod_{i=1}^{N_k} (\theta_i^k)^{f_{k,i}}\right)\right] - E_q[\log q(\theta)] - E_q[\log q(\boldsymbol{Z})]$$
$$= \sum_{k=1}^{K} \sum_{i=1}^{N_k} E_q[(f_{k,i} + \alpha_i^k - 1) \times \log(\theta_i^k)] + H(q(\theta)) + H(q(\boldsymbol{Z})),$$

where $H(q(\theta))$ denotes the entropy of the distribution $q(\theta)$, and $H(q(\boldsymbol{Z}))$ denotes the entropy of the distribution $q(\boldsymbol{Z})$ (for the definition of entropy, see Appendix A).

If we consider Algorithm 6.1 for this case, then we iterate between stages: (a) assuming $q(\theta)$ is fixed, we optimize the bound in the above equation with respect to $q(\boldsymbol{Z})$ and (b) assuming $q(\boldsymbol{Z})$ is fixed, we optimize the bound in the above equation with respect to $q(\theta)$.

Assume the case where $q(\theta)$ is fixed. In that case, $f_{k,i}$ depends only on the latent assignments $z^{(1)}, \ldots, z^{(n)}$ and not on the parameters, and therefore it holds that:

$$\mathcal{F}(q, x^{(1)}, \ldots, x^{(n)}|\alpha) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} E_q[(f_{k,i} + \alpha_i^k - 1) \times \psi_i^k] + H(q(\boldsymbol{Z})) + \text{const}$$
$$= \sum_{k=1}^{K} \sum_{i=1}^{N_k} E_q[\psi_i^k f_{k,i} - \log A(\psi)] + H(q(\boldsymbol{Z})) + \text{const}, \tag{6.9}$$

with ψ having the same vector structure like θ and α such that

$$\psi_i^k = E_{q(\theta)}[\log(\theta_i^k)],$$

and

$$\log A(\psi) = \sum_{z^{(1)}} \cdots \sum_{z^{(n)}} \exp\left(\sum_{k=1}^{K} \sum_{i=1}^{N_k} \psi_i^k f_{k,i}\right).$$

Note that the term $\log A(\psi)$ can be added to Equation 6.9 because it does not depend on the latent structures, since we sum them out in this term. It does, however, depend on $q(\theta)$, but it is assumed to be fixed. If we carefully consider Equation 6.9, we note that it denotes the negated KL-divergence (Appendix A) between $q(\boldsymbol{Z})$ and a log-linear model over $\boldsymbol{Z}$ with sufficient statistics $f_{k,i}$ and parameters ψ_i^k. Therefore, when $q(\theta)$ is fixed, the functional $\mathcal{F}$ is maximized when we choose $q(\boldsymbol{Z})$ to be a log-linear distribution with the sufficient statistics $f_{k,i}$ and parameters $\psi_i^k = E_{q(\theta)}[\log(\theta_i^k)]$.

The meaning of this is that even though we *a priori* left $q(\boldsymbol{Z})$ to be in a nonparametric family, we discovered that the tightest solution for it resides in a parametric family, and this family has a very similar form to the likelihood (the main difference between the approximate posterior family and the likelihood is that with the approximate posterior family we also require normalization through $\log Z(\psi)$ because ψ does not necessarily represent a collection of multinomial distributions).

What about the opposite case, i.e., when $q(\boldsymbol{Z})$ is fixed and $q(\theta)$ needs to be inferred? In that case, it holds that:

$$\mathcal{F}(q, x^{(1)}, \ldots, x^{(n)}|\alpha) \propto \sum_{k=1}^{K}\sum_{i=1}^{N_k}\left(E_q[f_{k,i}] + \alpha_i^k - 1\right) \times E_{q(\theta)}[\log \theta_i^k] - H(q(\theta)). \quad (6.10)$$

If we carefully consider the equation above, we see that it is proportional to the KL-divergence between $q(\theta)$ and a product of Dirichlet distributions (of the same form as the prior family) with hyperparameters $\beta = (\beta^1, \ldots, \beta^K)$ such that $\beta_i^k = E_q[f_{k,i}] + \alpha_i^k$. This is again a case where we leave $q(\theta)$ nonparametric, and we discover that the tightest solution has a parametric form. In fact, not only is it parametric, it also has the same form as the prior family.

The final variational inference algorithm looks like this:

- Initialize in some way $\beta = (\beta^1, \ldots, \beta^K)$.
- Repeat until convergence:
  - Compute $q(z^{(1)}, \ldots, z^{(n)})$ as the log-linear model mentioned above with parameters $\psi_i^k = E_{q(\theta)}[\log(\theta_i^k)|\beta]$.
  - Compute $q(\theta)$ as a product of Dirichlet distributions with hyperparameters $\beta_i^k = E_q[f_{k,i}|\psi] + \alpha_i^k$.

Consider the computation of $E_{q(\theta)}[\log(\theta_i^k)|\beta]$ and $E_q[f_{k,i}|\psi]$. It is known that for a given Dirichlet distribution, the expected log value of a single parameter can be expressed using the digamma function, meaning that:

$$E_{q(\theta)}[\log(\theta_i^k)|\beta] = \Psi(\beta_i^k) - \Psi(\sum_{i=1}^{N_k}\beta_i^k),$$

with Ψ representing the digamma function. The digamma function cannot be expressed analytically, but there are numerical recipes for finding its value for a given parameter. See Appendix B for more details about the digamma function and its relationship to the Dirichlet distribution.

On the other hand, computing $E_q[f_{k,i}|\psi]$ can be done using an algorithm that heavily depends on the structure of the likelihood function. For PCFGs, for example, this expectation can be computed using the inside-outside algorithm. For HMMs, it can be done using the forward-backward algorithm. See Chapter 8 for more details. Note that this expectation is computed for each observed example separately, i.e., we calculate $E_q[f_i^k(x^{(j)}, z^{(j)})|\psi]$ for $j \in \{1, \ldots, n\}$ and then aggregate all of these counts to get $E_q[f_{k,i}|\psi]$.

Whenever we are simply interested in the posterior over $q(\boldsymbol{Z})$, the above two update steps collapse to the following update rule for the variational parameters of $q(\boldsymbol{Z})$:

$$(\psi_i^k)^{\text{new}} \leftarrow \Psi(E_q[f_{k,i}|\psi^{\text{old}}] + \alpha_i^k) - \Psi(\sum_{i=1}^{N_k} E_q[f_{k,i}|\psi^{\text{old}}] + \alpha_i^k). \quad (6.11)$$

Note that for these updates, the variational parameters ψ_i^k need to be initialized first. Most often in the Bayesian NLP literature, when variational inference is used, the final update rule forms such as the one above are described. The log-linear model parametrized by ψ_i^k can be re-parameterized using a new set of parameters $\mu_i^k = \exp(\psi_i^k)$ for all k and i. In this case, the update becomes:

$$(\mu_i^k)^{\text{new}} \leftarrow \frac{\exp\left(\Psi(E_q[f_{k,i}|\mu^{\text{old}}] + \alpha_i^k)\right)}{\exp\left(\Psi(\sum_{i=1}^{N_k} E_q[f_{k,i}|\mu^{\text{old}}] + \alpha_i^k)\right)}. \quad (6.12)$$

Note that now we have a similar update to the EM algorithm, where we compute expected counts, and in the M-step, we normalize them. The main difference is that the counts are passed through the filter of the exp-digamma function, $\exp(\Psi(x))$. Figure 6.1 plots the exp-digamma function and compares it against the function $x - 0.5$. We can see that as x becomes larger, the two functions get closer to each other. The main difference between the two functions is that at values smaller than 0.5, for which the exp-digamma function returns positive values which are very close to 0, while $x - 0.5$ returns negative values. Therefore, one way to interpret the update Equation 6.12 is as the truncation of low expected counts during the E-step (lower than 0.5). Higher counts are also subtracted a value of around 0.5, and the higher the count is in the E-step, the less influential this decrease will be on the corresponding μ parameter.

6.3.2 CONNECTION TO THE EXPECTATION-MAXIMIZATION ALGORITHM

The variational inference algorithm in the previous section, in spirit, resembles the expectation-maximization (EM) algorithm of Dempster et al. (1977), which is set up in the frequentist setting. The goal of the EM algorithm is to estimate the parameters of a given model from incomplete data.

The EM algorithm iterates between two steps: the E-step, in which the posterior over the latent structures is computed, and the M-step, in which a new set of parameters is computed,

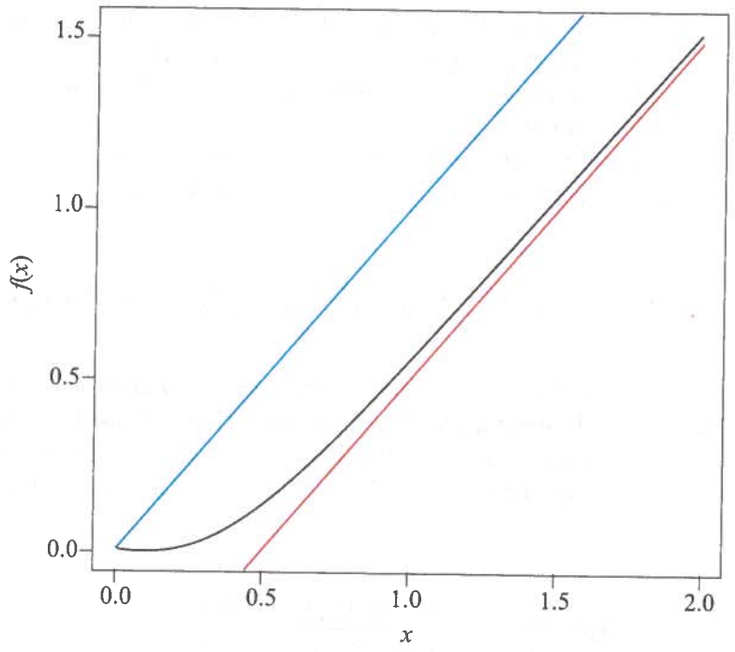

Figure 6.1: A plot of the function $f(x) = \exp(\Psi(x))$, the exp-digamma function (in the middle in black) compared to the functions $f(x) = x$ (at the top in blue) and $f(x) = x - 0.5$ (at the bottom in red). Adapted from Johnson (2007b).

until the marginal log-likelihood converges. It can be shown that the EM algorithm finds a local maximum of the marginal log-likelihood function. The M-step is performed by maximizing the expected log-likelihood of all variables in the model. The expectation is taken with respect to a product distribution: the product of the empirical distribution over the observed data and the posterior induced in the E-step. For more detailed information about the expectation-maximization algorithm, see Appendix A.

There is actually a deeper connection between the EM algorithm and the variational inference algorithm presented in Algorithm 6.1. The variational inference algorithm reduces to the EM algorithm when the inputs to the variational inference algorithm and the prior in the model are chosen carefully.

Consider the case where the set of latent variables is partitioned into two random variables: in terms of Algorithm 6.1, U_1 corresponds to a random variable over the parameters, and U_2 corresponds to a variable over the set of all latent structures in the model (usually, it would be $Z^{(1)}, \ldots, Z^{(n)}$). Hence, the posterior has the form $q(\theta, \boldsymbol{Z}) = q(\theta)q(\boldsymbol{Z})$.

Consider also $\mathcal{Q}_1$ to represent the set of all distributions that place their whole probability mass on a single point on the parameter space. This means that $\mathcal{Q}_1$ includes the set of all the distributions $q(\theta|\mu)$ (parameterized by $\mu \in \Theta$) such that

$$q(\theta|\mu) = \begin{cases} 1, & \text{if } \theta = \mu \\ 0 & \text{otherwise,} \end{cases}$$

$\mathcal{Q}_2$, on the other hand, remains nonparametric, and just includes the set of all possible distributions over the latent structures. Last, the prior chosen in the model is chosen to be $p(\theta) = c$ (for some constant c) for all $\theta \in \Theta$, i.e., a uniform non-informative prior (possibly improper).

The functional $\mathcal{F}$ now in essence depends on the assignment μ (selecting $q(\theta|\mu)$) and the $q(\boldsymbol{Z})$. We will express this functional as:

$$\mathcal{F}(q(\boldsymbol{Z}), \mu, x^{(1)}, \ldots, x^{(n)}) = E_{q(\boldsymbol{Z})}\left[\log\left(\frac{p(\mu|\alpha)p(\boldsymbol{Z}, \boldsymbol{X} = (x^{(1)}, \ldots, x^{(n)})|\mu)}{q(\boldsymbol{Z})}\right)\right].$$

If we assume a non-informative constant prior, then maximizing the bound with respect to $q(\boldsymbol{Z})$ and μ can be done while ignoring the prior:

$$\mathcal{F}(q(\boldsymbol{Z}), \mu, x^{(1)}, \ldots, x^{(n)}) \propto E_{q(\boldsymbol{Z})}\left[\log\left(\frac{p(\boldsymbol{Z}, \boldsymbol{X} = (x^{(1)}, \ldots, x^{(n)})|\mu)}{q(\boldsymbol{Z})}\right)\right].$$

This functional is exactly the same bound that the expectation-maximization algorithm maximizes. Maximizing the right-hand side with respect to $q(\boldsymbol{Z})$ while keeping μ fixed yields the posterior $q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X} = (x^{(1)}, \ldots, x^{(n)}), \mu)$, which in turn yields the E-step in the EM algorithm. On the other hand, maximizing the right-hand side with respect to μ yields the M-step—doing so maximizes the bound with respect to the parameters, which keeps $q(\boldsymbol{Z})$ fixed. See Appendix A for a derivation of the EM algorithm.

6.4 EMPIRICAL BAYES WITH VARIATIONAL INFERENCE

In the empirical Bayes setting (Section 4.3), parameters are drawn for each observed instance. There, the typical approach to mean-field variational inference would be to use an approximate posterior family such that all latent structures and all parameter sets are independent of each other (see Equation 6.6).

The variational inference algorithm (Algorithm 6.1) in this case actually separately solves each pair of problems for each instance i, finding the posterior $q(\theta^{(i)})$ and $q(Z^{(i)})$. Therefore, when using this kind of mean-field approximation, we require an additional estimation step, which integrates all the solutions for these sub-problems into a re-estimation step of the prior.

Input: Observed data $x^{(1)}, \ldots, x^{(n)}$, the bound $\mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n)}|\alpha)$.
Output: Factorized approximate posteriors $q(\theta^{(i)})$ and $q(Z^{(i)})$ for $i \in \{1, \ldots, n\}$ and an estimated hyperparameter α.

1: Initialize α'
2: **repeat**
3: Maximize $\mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n)}|\alpha')$ with respect to q^*
4: using Algorithm 6.1 with factorization as in Equation 6.6
5: $\alpha' \leftarrow \arg\max_{\alpha'} \mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n)}|\alpha')$
6: **until** the bound $\mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n)}|\alpha')$ converges
7: **return** (α', q^*)

Algorithm 6.2: The mean-field variational expectation-maximization algorithm (empirical Bayes).

This is the main idea behind the variational EM algorithm. Variational EM is actually an expectation-maximization algorithm, in which the hyperparameters for a prior family are estimated based on data, and in which the E-step is an *approximate* E-step that finds a posterior based on a variational inference algorithm, such as the one introduced in Algorithm 6.1. The approximate posterior is identified over $\boldsymbol{Z}$ and θ, while the M-step maximizes the marginal log-likelihood with respect to the hyperparameters.

The variational EM algorithm, with mean-field variational inference for the E-step, is given in Algorithm 6.2.

6.5 DISCUSSION

We turn now to a discussion about important issues regarding the variational inference algorithms presented in this chapter—issues that have a crucial effect on the performance of these algorithms, but do not have well-formed theory.

6.5.1 INITIALIZATION OF THE INFERENCE ALGORITHMS

The need to properly initialize the variational parameters in variational inference is a problematic issue, mostly because it does not have a well-established theory, yet it has been shown that initialization may greatly affect the results when using variational inference.

For example, with the Dirichlet-multinomial family mean-field variational inference algorithm in Section 6.3.1, one has to decide how to initialize β (or alternatively, if the algorithm is started on the second step in the loop instead of the first step, when computing the expectations of the features $f_{k,i}$, one would have to decide how to initialize the parameters of the log-linear model $q(z^{(1)}, \ldots, z^{(n)})$).

The variational bound that $\mathcal{F}$ represents, for a general model, is a non-convex function (with respect to the approximate posterior q), and therefore Algorithm 6.1 does not have any guarantees in terms of converging to the global maximum of the variational bound. One approach for tackling this issue is quite similar to the solution that is often used for the EM algorithm in the form of random restarts. The variational inference algorithm is run repetitively from different starting points, and we eventually choose the run that gives the maximal value to the variational bound.

This method will not necessarily lead to optimal results with respect to the evaluation metric used. The aim of maximizing the variational bound is to obtain higher log-likelihood for the observed data. Log-likelihood here is used as a proxy for the underlying evaluation metric, such as the parsing evaluation metric (Black et al., 1991) or part-of-speech accuracy. However, being just a proxy, it does not fully correlate with the evaluation metric. Even if we were able to globally maximize the log-likelihood, this problem would persist.

For this reason, random restarts, which aim at maximizing the variational bound, are sometimes replaced with a more specific initialization technique which is based on some intuition that the modeler has about the relationship between the data and the model. For example, a common technique for unsupervised dependency parsing, is to initialize EM (Klein and Manning, 2004) or variational inference (Cohen et al., 2009) with parameters that tend to prefer attachments for words that are close, in their position in the text, to each other. This is a very useful bias for unsupervised dependency parsing in general (Eisner and Smith, 2005, Spitkovsky et al., 2010).

Other initialization techniques include initialization based on a simpler model, sometimes a model which induces a concave log-likelihood function (Gimpel and Smith, 2012). Many of the techniques used to initialize EM for various models can also be used effectively for variational inference.

6.5.2 CONVERGENCE DIAGNOSIS

Checking for convergence of the variational inference algorithm (or more specifically, the bound $\mathcal{F}(q^*, x^{(1)}, \ldots, x^{(n}|\alpha)$ in Algorithm 6.1 or Algorithm 6.2) is relatively easy, since all quantities in the bound $\mathcal{F}$ are computable. However, it is important to note that unlike EM, variational inference does not guarantee an increase in the log-likelihood of the data after each iteration. While both EM and variational inference are coordinate ascent algorithms, EM finds a local maximum for the log-likelihood function, and variational inference finds a local maximum for the variational bound only. (Both algorithms use a similar bounding technique, based on Jensen's inequality, but the EM's bound for the log-likelihood is tight because no assumptions are made for the approximate posterior family. This means that the bound for EM equals the log-likelihood at its maximal value.)

6.5.3 THE USE OF VARIATIONAL INFERENCE FOR DECODING

There are several ways to use the output of variational inference and variational EM in order to actually predict or estimate parameters. In the non-empirical-Bayes variational inference setting, once $q(\theta)$ is estimated, this posterior can be summarized as a point estimate following the techniques in Chapter 4. Afterward, decoding can proceed using this point estimate. In addition, one can follow maximum *a posteriori* decoding directly using $q(\boldsymbol{Z})$ and identify

$$(z^{(1)}, \ldots, z^{(n)}) = \arg \max_{(z^{(1)}, \ldots, z^{(n)})} q(\boldsymbol{Z} = (z^{(1)}, \ldots, z^{(n)})).$$

A similar route can be followed in the empirical Bayesian setting, decoding $z^{(i)}$ by computing $\arg\max_z q(Z^{(i)} = z)$.

With variational EM, the hyperparameters α that are being eventually estimated can be used to get a summary for the parameter's point estimate. For example, given these hyperparameters α, one can use the mean value of the posterior over the parameters as a point estimate θ^*

$$\theta^* = E[\theta|\alpha] = \int_\theta \theta p(\theta|\alpha) d\theta,$$

or alternatively, $\theta^* = \arg\max_\theta p(\theta|\alpha)$ (corresponding to maximum *a posteriori* estimate). See Chapter 4 for a discussion. If the hyperparameters α have the same structure as the parameters (i.e., for each hyperparameter in the ith coordinate, α_i, maps directly to a parameter θ_i), then the hyperparameters themselves can be used as a point estimate. The hyperparameters may not adhere, perhaps, to constraints on the parameter space (i.e., it could be the case that $\alpha \notin \Theta$), but they often do yield *weights*, which can be used in decoding the underlying model.

Cohen and Smith (2010b) used this technique, and estimated the hyperparameters of a collection of logistic normal distributions for grammar induction. The Gaussian means were eventually used as parameters for a weighted grammar they used in decoding.

The above approach is especially useful when there is a clear distinction between a training set and a test set, and the final performance measures are reported on the test set, as opposed to a setting in which inference is done on all of the observed data.

When this split between a training and test set exists, one can use a different approach to the problem of decoding with variational EM. Using the hyperparameters estimated from the training data, an extra variational inference step can be taken on the test set, thus identifying the posterior over latent structures for each of the training examples (using mean-field variational inference). Based on these results, it is possible to follow the same route mentioned in the beginning of this section, finding the highest scoring structure according to each of the posteriors and using these as the predicted structure.

6.5.4 VARIATIONAL INFERENCE AS KL DIVERGENCE MINIMIZATION

Consider Equation 6.3 again, restated below:

$$\begin{aligned}
&\log p(\boldsymbol{X}|\alpha) = \\
&= E_q \left[\log \left(\frac{p(\theta|\alpha) \left(\prod_{i=1}^n p(Z^{(i)}|\theta) p(x^{(i)}|Z^{(i)}, \theta)\right)}{q(\theta, \boldsymbol{Z})} \right) \middle| \alpha \right] \\
&= \mathcal{F}(q, x^{(1)}, \ldots, x^{(n)}|\alpha).
\end{aligned}$$

The bound $\mathcal{F}$ actually denotes the Kullback-Leibler (KL) divergence (see Appendix A) between q and the posterior. As mentioned in the beginning of this chapter, finding an approximate posterior is done by minimizing $\mathcal{F}$. Therefore, minimization of the bound $\mathcal{F}$ corresponds to finding a posterior q from the family of posteriors $\mathcal{Q}$ which minimizes $\mathrm{KL}(q, p)$.

KL divergence is not a symmetric function, and unfortunately, this minimization of the KL divergence is done in the "reverse direction" from what is desirable. In most, "more correct," KL divergence minimization problems (such as maximum likelihood estimation), the free distribution that is optimized should represent the second argument to the KL divergence, while the "true" distribution (the true posterior, in the case of variational inference), should represent the first argument. In the reverse direction, $\min_q \mathrm{KL}(q, p)$, one could find solutions that are not necessarily meaningful. Still, with this approach, the KL divergence would get its minimum when $p = q$ (and then it would be 0), which is a desirable property.

A discussion regarding KL divergence minimization direction for variational inference, with graphical models, is given by Koller and Friedman (2009).

6.5.5 ONLINE VARIATIONAL INFERENCE

Standard variational inference and variational EM algorithms work in a batch mode. This means that the available learning data is pre-determined, statistics are then computed from all datapoints (E-step), and finally an update to the parameters is made (M-step). Expectation-maximization works in a similar way.

An alternative to these batch algorithms is online algorithms. With online algorithms, each datapoint is first processed, then followed by an update to the parameters. The motivation behind online algorithms is a scenario in which an "infinite" stream of datapoints is fed to an algorithm, and the inference algorithm needs to update its internal state. This internal state can be used to make predictions until more data in the infinite stream arrives. This setting is especially appealing for large-scale data and real-world applications, in which a statistical model is continuously updated as more data is being presented to it (for example, from the web).

Variational inference and variational EM can be converted into an online algorithm as well, relying on ideas from the literature regarding online EM (Cappé and Moulines, 2009, Liang and Klein, 2009, Neal and Hinton, 1998, Sato and Ishii, 2000). The idea behind this conversion is

to make an update to the parameters once the posterior of an example is computed. The current parameters are then interpolated (with some mixture coefficient $\lambda \in [0, 1]$) using the statistics computed for the example.

Batch algorithms and online algorithms, as described above are two extremes in a wide spectrum of approaches. As a middle ground, one can use a so-called "mini-batch" online algorithm, in which several examples are processed at a time using the current set of parameters, and only then followed by an update to the parameters.

Online variational inference that relies on such ideas has been used, for example, for the LDA model (Hoffman et al., 2010), as well as for unsupervised learning of syntax (Kwiatkowski et al., 2012a). It has also been used for nonparametric models, such as the hierarchical Dirichlet process (Wang et al., 2011). Such models are described in Chapter 7.

6.6 SUMMARY

Variational inference is just one of the workhorses used in Bayesian NLP inference. The most common variant of variational inference used in NLP is that of mean-field variational inference, the main variant discussed in this chapter.

Variational expectation-maximization can be used in the empirical Bayes setting, using a variational inference sub-routine for the E-step, while maximizing the variational bound with respect to the hyperparameters in the M-step.

6.7 EXERCISES

6.1. Consider the model in Example 5.1. Write down a mean-field variational inference algorithm to infer $p(\theta|x^{(1)}, \ldots, x^{(n)})$.

6.2. Consider again the model in Example 5.1, only now change it such that there are multiple parameter draws, and $\theta^{(1)}, \ldots, \theta^{(n)}$ drawn for each example. Write down a mean-field variational inference algorithm to infer $p(\theta^{(1)}, \ldots, \theta^{(n)}|x^{(1)}, \ldots, x^{(n)})$.

6.3. Show that Equation 6.8 is true. Also, show that Equation 6.10 is true.

6.4. Let $\theta_1, \ldots, \theta_K$ represent a set of K parameters, where K is fixed. In addition, let $p(X|\theta_i)$ represent a fixed distribution for a random variable X over sample space Ω such that $|\Omega| < \infty$. Assume that $p(x|\theta_i) \neq p(x|\theta_j)$ for any $x \in \Omega$ and $i \neq j$. Define the following model, parametrized by μ:

$$p(X|\mu, \theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \mu_k \, p(X|\theta_k),$$

where $\sum_{k=1}^{K} \mu_k = 1$ and $\mu_k \geq 0$. This is a mixture model, with mixture components that are fixed.

What is the log-likelihood for n observations, $x^{(1)}, \ldots, x^{(n)}$, with respect to the parameters μ? Under what conditions is the log-likelihood convex, if at all (with respect to μ)?

6.5. Now assume that there is a symmetric Dirichlet prior over μ, hyperparametrized by $\alpha > 0$. Compute the marginal log-likelihood n observations, $x^{(1)}, \ldots, x^{(n)}$, integrating out μ. Is this a convex function with respect to α?