An Introduction to Bayesian Inference Via Variational Approximations: Supplemental Notes

Justin Grimmer

July 1, 2010

This document contains the supplemental material for "An Introduction to Bayesian Inference Via Variational Approximations"

1 Deriving the General Variational Approximation Algorithm

1.1 The Tractability-Fit Tradeoff in Variational Approximations

The goal of a variational approximation is to approximate a posterior, $p(\boldsymbol{\beta}|\boldsymbol{Y})$ by making an approximating distribution, $q(\boldsymbol{\beta})$, as close as possible to the true posterior (Bishop, 2006). We search over the space of approximating distributions in order to find the particular distribution with the minimum KL-divergence with the actual posterior. Formally, we search over the set of approximating distributions $q(\boldsymbol{\beta})$ to minimize

$$\text{KL}(q(\boldsymbol{\beta})||p(\boldsymbol{\beta}|\boldsymbol{Y})) \equiv \text{KL}(q||p) \quad = \quad -\int q(\boldsymbol{\beta}) \log \left\{ \frac{p(\boldsymbol{\beta}|\boldsymbol{Y})}{q(\boldsymbol{\beta})} \right\} d\boldsymbol{\beta}. \tag{1.1}$$

If we make no assumptions about the factorized distribution, then Equation 1.1 is minimized when $q(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|\boldsymbol{Y})$ (because $\log 1 = 0$). Of course, this is not particularly helpful, because the posterior is generally intractable. To make manipulation of the approximating distribution possible, we introduce additional assumptions into the approximating distribution, with the hope that this will make inference tractable, while also still providing a close approximation to the true posterior.

This presents the inherent tension when approximating a posterior with a variational approximation, while also illustrating one of the major advantages of variational approximations. On the one hand, using a more general approximating distribution will lead to a better approximation of the posterior distribution: we make the approximation *arbitrarily close* by reducing the number of assumptions made in the approximating distribution. But fewer assumptions result in the approximating distribution being less tractable, limiting the usefulness of the approximation. On the other hand, introducing more restrictive assumptions to the approximating distribution, results in a poorer approximation (or a greater KL-divergence between the approximating distribution and true posterior). But if the additional assumptions imposed upon the approximation are useful, then the approximating distribution will be easier to manipulate for inference. Therefore, the goal (and indeed, the art) of applying variational approximations is to identify a set of approximating distributions that lead to tractable inferences, while also providing a close fit to the true posterior distribution. This tension is actually an advantage of the variational approximation over other approximation methods, because we can control the quality of our approximation by controlling the assumptions made in the approximating distributions.

Following a large literature in computer science and machine learning, we use a very general form of approximating distributions (eg Jordan et al. 1999; Blei et al. 2003; Bishop 2006). We focus upon approximating distributions that contain additional independence than in the true posterior, but we make no *other assumption about the particular parametric form of the approximating distribution*. Rather, the distributional form for the approximating distribution will be *estimated*. We choose this approximating distribution also because it has been rigorously proven to perform well when applied to a large class of models. Wang and Titterington (2004) demonstrate that, given a sufficient number of observations from the data, this family of approximating distributions will correctly characterize the posterior mean, a guarantee not possible for sampling based approaches to inference. [1]

Following Bishop (2006), we call this a *factorized* approximation, because the independence assumption results in the approximating distribution being divided into a set of factors (or blocks of parameters). Similar to the Gibbs sampler, we first partition $\boldsymbol{\beta}$, into a set of K blocks, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_K)$. Then we will restrict attention to approximating distributions that have the form,

$$q(\boldsymbol{\beta}) \quad = \quad \prod_{k=1}^{K} q(\boldsymbol{\beta}_k). \tag{1.2}$$

The variational algorithm will identify (rather than assume) the specific parametric families that constitute each component of the factorized distribution.

1.2 Maximizing a Lower Bound to Minimize the KL-Divergence

Working directly with Equation 1.1 can be challenging (Jordan et al., 1999; Bishop, 2006), so we instead solve an equivalent problem: maximizing a lower bound on the log-probability of the data (Bishop, 2006).[2] To obtain this bound, we first write the log-marginal probability of the data

$$\log p(\boldsymbol{Y}) \quad = \quad \log \int p(\boldsymbol{Y}, \boldsymbol{\beta}) d\boldsymbol{\beta}$$

Next, we introduce the approximating distribution by multiplying by $\frac{q(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} = 1$,

$$\log p(\boldsymbol{Y}) \quad = \quad \log \int p(\boldsymbol{Y}, \boldsymbol{\beta}) \frac{q(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} d\boldsymbol{\beta}$$

Then apply Jensen's inequality to move the logarithm inside the integration (Blei et al., 2003; Bishop, 2006)

$$\log p(\boldsymbol{Y}) \quad \geq \quad \underbrace{\int q(\boldsymbol{\beta}) \log \frac{p(\boldsymbol{\beta}, \boldsymbol{Y})}{q(\boldsymbol{\beta})} d\boldsymbol{\beta}}_{\mathcal{L}(q)} \tag{1.3}$$

The right-hand side of Inequality 1.3 represents a lower-bound on the marginal probability of the data. To see that maximizing $\mathcal{L}(q)$ is equivalent to minimizing the KL-divergence between the true posterior and the

[1] This guarantee is very general, but requires sufficient sample size and is proven for exponential family models or mixtures of exponential family models. Blei and Lafferty (2006) applies variational approximations to a non-exponential family model, noting that there are fewer guarantees, but the approximation appeared to perform well in their application.

[2] The derivation in this section is standard in the variational approximation literature and will at times follow the arguments advanced in Jordan et al. (1999); Bishop (1999, 2006).

approximating distribution, we rely upon a well-known identity (Bishop, 2006)[3],

$$\log p(\boldsymbol{Y}) \quad = \quad \mathcal{L}(q) + \mathrm{KL}(q||p). \tag{1.4}$$

$\log p(\boldsymbol{Y})$ is a fixed number that is unchanged as we modify $q(\boldsymbol{\beta})$. Further, KL-divergences are always non-negative and therefore as $\mathcal{L}(q)$ increases, $\mathrm{KL}(q||p)$ must decrease. It follows directly that when $\mathcal{L}(q)$ is at a maximum, $\mathrm{KL}(q||p)$ must be at a minimum.

The dual relationship between the lower-bound on the data and the KL-divergence between the approximating distribution reveals further the tractability-fit tradeoff. On the one hand, we can obtain an arbitrarily good fit on the lower bound of the marginal probability of the data (and equivalently reduce the KL-divergence between the true and approximating distributions to zero) by allowing for arbitrary distributions in the posterior–which of course makes the problem intractable and renders the variational approach useless. As we add assumptions to the approximating distribution, it will be easier to manipulate. But, the additional assumptions imply that the lower bound on the data will be loose–or that our estimate of the posterior will be lower quality.

1.3 An Iterative Algorithm for Maximizing $\mathcal{L}(q)$

We now derive the update steps used to maximize $\mathcal{L}(q)$ with respect to the approximating distribution. Before we perform this derivation, please note that this section is technical and justifies the steps we use throughout the paper to apply variational approximations. The remainder of the paper, however, is intelligible without following the derivations here and therefore can be skipped by the uninterested reader.[4]

The algorithm begins by first initializing each of the factors of the approximation, $q(\boldsymbol{\beta}_1)^{\mathrm{old}}, \ldots q(\boldsymbol{\beta}_K)^{\mathrm{old}}$ and we consider each term of the lower bound iteratively.[5] At the moment, initializing the functions may seem abstract, but in our particular examples it will be clear exactly how to initialize the distributions.

First, focus on updating one block of parameters, $\boldsymbol{\beta}_k$. We can rewrite the lower bound to focus on the terms in $q(\boldsymbol{\beta}_k)$ as,

$$\begin{aligned}\mathcal{L}(q)_{\boldsymbol{\beta}_k} \quad = \quad & -\int q(\boldsymbol{\beta}_k) \underbrace{\left(\int \prod_{j \neq k} \log p(\boldsymbol{\beta}, \boldsymbol{Y}) q(\boldsymbol{\beta}_j)^{\mathrm{old}} d\boldsymbol{\beta}_j\right)}_{\mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta}, \boldsymbol{Y})]} d\boldsymbol{\beta}_k \\ & - \int q(\boldsymbol{\beta}_k) \log q(\boldsymbol{\beta}_k) d\boldsymbol{\beta}_k - \underbrace{\sum_{j \neq k} \int q(\boldsymbol{\beta}_j)^{\mathrm{old}} \log q(\boldsymbol{\beta}_j)^{\mathrm{old}} d\boldsymbol{\beta}_j}_{\text{Constants}}\end{aligned} \tag{1.5}$$

Where we have collected terms that do not depend on $q(\boldsymbol{\beta}_k)$. The first underbrace collects

$$\mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta}, \boldsymbol{Y})] \quad = \quad \int \prod_{j \neq k} \log p(\boldsymbol{\beta}, \boldsymbol{Y}) q(\boldsymbol{\beta}_j)^{\mathrm{old}} d\boldsymbol{\beta}_j \tag{1.6}$$

[3]This argument is well known in the literature on variational approximations. See Bishop (2006) for an extended explication.

[4]The derivation in this section is standard in the variational approximation literature, see Jordan et al. (1999) and Bishop (2006).

[5]Bishop (2006) shows that the lower bound is convex in the factors, so long as the posterior distribution arises from an exponential family distributions (or exponential family distributions with missing values). We will restrict our attention to that family of posterior distributions here.

which is equal to the expected value of the log posterior with the expectation taken over the variational distribution using all blocks but $\boldsymbol{\beta}_k$ (Bishop, 2006). Further, $-\sum_{j\neq k}\int q(\boldsymbol{\beta}_j)\log q(\boldsymbol{\beta}_j)d\boldsymbol{\beta}_j$ is a collection of constants, with respect to $q(\boldsymbol{\beta}_k)$. We will collect the terms that do not depend directly on $\boldsymbol{\beta}_k$ into the distribution $\log\tilde{p}(\boldsymbol{\beta}_k) = \mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta},\boldsymbol{Y})] + \text{constants}$. Inserting this distribution into Equation 1.5 and collecting terms we have (Bishop, 2006),

$$\mathcal{L}(q)_{\boldsymbol{\beta}_k} = \int q(\boldsymbol{\beta}_k)\log\left\{\frac{\tilde{p}(\boldsymbol{\beta}_k)}{q(\boldsymbol{\beta}_k)}\right\}d\boldsymbol{\beta}_k. \tag{1.7}$$

Equation 1.7 is the negative KL-Divergence between $\tilde{p}(\boldsymbol{\beta}_k)$ and $q(\boldsymbol{\beta}_k)$ and therefore will be maximized when $q(\boldsymbol{\beta}_k) = \tilde{p}(\boldsymbol{\beta}_k)$. Applying a logarithm to both sides shows that this distribution is easily calculated as,

$$\begin{aligned}\log q(\boldsymbol{\beta}_k)^{\text{new}} &= \log\tilde{p}(\boldsymbol{\beta}_k)\\ &= \mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta},\boldsymbol{Y})] + \text{constants}.\end{aligned}$$

Our final step to obtaining the form of $q(\boldsymbol{\beta}_k)$ will be to exponentiate each side and ensure that we have obtained the proper constants (Bishop, 2006),

$$q(\boldsymbol{\beta}_k)^{\text{new}} = \frac{\exp(\mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta},\boldsymbol{Y})])}{\int\exp(\mathrm{E}_{j\neq k}[\log p(\boldsymbol{\beta},\boldsymbol{Y})])d\boldsymbol{\beta}_k}$$

In practice, we will avoid carrying out the final integral by inspecting the form of $\log q(\boldsymbol{\beta}_k)^{\text{new}}$ (this will be clear in the specific examples). If we repeat this argument for each of the K factors, we have an algorithm to describe the optimization.

The update equations for each step are iterative and depend on values for the other equations.[6] In theory, convergence to a global maximum is guaranteed and easily monitored by evaluating the change in $\mathcal{L}(q)$. $\mathcal{L}(q)$ is a single number and therefore assessing convergence of any one application of the model is automatic (avoiding the complicated convergence diagnostics prevalent in applications of sampling based approaches to inference).

To summarize, a typical iteration of the variational approximation algorithm contains the following steps,

$$\begin{aligned}q(\boldsymbol{\beta}_1)^{\text{new}} &= \frac{\exp\left(\mathrm{E}_{j\neq 1}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)}{\int\exp\left(\mathrm{E}_{j\neq 1}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)d\boldsymbol{\beta}_1}\\ q(\boldsymbol{\beta}_2)^{\text{new}} &= \frac{\exp\left(\mathrm{E}_{j\neq 2}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)}{\int\exp\left(\mathrm{E}_{j\neq 2}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)d\boldsymbol{\beta}_2}\\ \vdots\quad &\vdots\quad \vdots\\ q(\boldsymbol{\beta}_K)^{\text{new}} &= \frac{\exp\left(\mathrm{E}_{j\neq K}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)}{\int\exp\left(\mathrm{E}_{j\neq K}[\log p(\boldsymbol{\beta},\boldsymbol{Y})]\right)d\boldsymbol{\beta}_K}\end{aligned}$$

There is an interesting comparison between the steps of a Gibbs sampler and the steps of a variational approximation (Blei and Jordan, 2006). In the Gibbs sampler, we condition upon the sampled values of the parameters, then draw from the corresponding distribution. In a variational approximation, we integrate (average) over the current estimates of the posterior distribution to obtain the approximating distribution.

[6] This iterative approach is referred to as *coordinate ascent* in the optimization literature (Bishop, 2006).

2 Bayesian Probit Regression

2.1 Gibbs Sampler

The Gibbs sampler for the Bayesian probit model proceeds in two steps and relies upon a data-augmentation (Albert and Chib, 1993; Gelman et al., 1995; Jackman, 2000). We divide the parameters into two blocks: the regression coefficients and the latent propensities and start each iteration with the current draw of the regression coefficients, $\boldsymbol{\beta}^t$. The first step is to draw the latent propensities. Define $\mu_i = \boldsymbol{X_i}'\boldsymbol{\beta}^t$,

$$Y_i^{*,t} \sim \left\{ \begin{array}{l} \text{Normal}_{[0,\infty)}(\mu_i, 1) \text{ if } Y_i = 1 \\ \text{Normal}_{(-\infty,0]}(\mu_i, 1) \text{ if } Y_i = 0, \end{array} \right. .$$

where $\text{Normal}_{[0,\infty)}(\mu_i, 1)$ is a normal distribution truncated to positive numbers. After drawing the latent propensity vector, $\boldsymbol{Y}^{*,t}$ we draw the updated parameter vector, $\boldsymbol{\beta}^{t+1}$. A standard derivation shows that,

$$\boldsymbol{\beta}^{t+1} \quad \sim \quad \text{Multivariate Normal}(\bar{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$$

where,

$$\begin{aligned} \bar{\boldsymbol{\beta}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \frac{1}{\sigma^2}\boldsymbol{I}\right)^{-1} \left(\boldsymbol{X}'\boldsymbol{Y}^*\right) \\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \frac{1}{\sigma^2}\boldsymbol{I}\right) \end{aligned}$$

2.2 EM Algorithm

To describe the EM algorithm for the Bayesian probit regression, we begin each iteration t with the current estimate for the regression parameters $\boldsymbol{\beta}^t$. The augmented data for the EM algorithm are the latent propensities for a positive response. Inspection shows that the posterior distribution for the augmented data, $p(Y_i^*|Y_i, \boldsymbol{X}_i, \boldsymbol{\beta})$, is a truncated-normal distribution, with expected value $E[Y_i] = \boldsymbol{X}_i\boldsymbol{\beta} + M_i$, where $M_i = \frac{-\phi(-\boldsymbol{X}_i\boldsymbol{\beta})}{\Phi(-\boldsymbol{X}_i\boldsymbol{\beta})}$ if $Y_i = 0$ and $M_i = \frac{\phi(-\boldsymbol{X}_i\boldsymbol{\beta})}{1-\Phi(-\boldsymbol{X}_i\boldsymbol{\beta})}$ if $Y_i = 1$, where ϕ is the normal density and Φ is the cumulative distribution function (Jackman, 2000). Collect these expected values into the vector $\text{E}[\boldsymbol{Y}^*]^t$.

During the maximization step, we average over the latent distribution and then maximize to obtain the updated values of $\boldsymbol{\beta}^{t+1}$,

$$\boldsymbol{\beta}^{t+1} \quad = \quad (\boldsymbol{X}'\boldsymbol{X} + \frac{1}{\sigma^2}\boldsymbol{I})^{-1} \left(\boldsymbol{X}'\text{E}[\boldsymbol{Y}^*]^t\right)$$

These steps are repeated until the sequence of regression coefficients converge.

3 Deriving Update Steps for Bayesian Probit Regression

This section provides the update steps for the Bayesian probit regression.

Update for $q(Y_i^*)^{\text{new}}$ Recall, we will compute the expected value of the log-posterior distribution, with the expectation taken over the distribution on the regression parameters $\boldsymbol{\beta}^{\text{var}}$. Writing out the terms that depend upon $\boldsymbol{Y}_i^*$, (and suppose that $Y_i = 1$),

$$\log q(\boldsymbol{Y}_i^*) \quad = \quad I(Y_i^* > 0)\left[-\frac{1}{2}\text{E}_{\boldsymbol{\beta}}\left(Y_i^* - \boldsymbol{X}_i^*\boldsymbol{\beta}\right)^2\right]$$

Completing the square reveals that this is a Truncated normal distribution, with $\mu_i = \boldsymbol{X}_i \mathrm{E}[\boldsymbol{\beta}]$. To complete the update step, we will first need to determine the functional form of $q(\boldsymbol{\beta})$.

Update for $q(\boldsymbol{\beta})^{\textbf{new}}$ To complete this update, we take the expectation of the log-posterior over the latent-propensities, using the approximating distribution. Collect the latent propensities into the $N \times 1$ vector $\boldsymbol{Y}^*$. Carrying out the expectations,

$$\begin{aligned}
\log q(\boldsymbol{\beta}) &= -\frac{1}{2}\mathrm{E}_{\boldsymbol{Y}^*}\left[(\boldsymbol{Y}^* - \boldsymbol{X}\boldsymbol{\beta})^{'}(\boldsymbol{Y}^* - \boldsymbol{X}\boldsymbol{\beta})\right] - \frac{1}{2\sigma^2}\boldsymbol{\beta}^{'}\boldsymbol{\beta} + \text{constants} \\
&= -\frac{1}{2}\underbrace{\mathrm{E}_{\boldsymbol{Y}^*}(\boldsymbol{Y}^*)^{'}(\boldsymbol{Y}^*)}_{\text{constant}} - \mathrm{E}_{\boldsymbol{Y}^*}[\boldsymbol{Y}^*]^{'}\boldsymbol{X}^{'}\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{\beta}^{'}\boldsymbol{X}^{'}\boldsymbol{X}\boldsymbol{\beta} - \frac{1}{2\sigma^2}\boldsymbol{\beta}^{'}\boldsymbol{\beta}
\end{aligned}$$

Which is easily recognizable as a quadratic form, so $q(\boldsymbol{\beta}) = \text{Multivariate Normal}(\boldsymbol{\beta}^{\text{var}}, \boldsymbol{\Sigma}^{\text{var}})$, with variance-covariance matrix $\boldsymbol{\Sigma}^{\text{var}} = (\boldsymbol{X}^{'}\boldsymbol{X} + \frac{1}{\sigma^2}\boldsymbol{I})^{-1}$, and mean parameter $\boldsymbol{\beta}^{\text{var}} = (\boldsymbol{X}^{'}\boldsymbol{X} + \frac{1}{\sigma^2}\boldsymbol{I})^{-1}(\boldsymbol{X}^{'}\mathrm{E}[\boldsymbol{Y}^*])$. Applying one final step to assess the expected values of the latent propensities (Jackman, 2000),

$$\mathrm{E}[Y_i^*] = \begin{cases} \mu_i + \frac{\phi_i}{1-\Phi_i} \text{ if } Y_i = 1 \\ \mu_i - \frac{\phi_i}{\Phi_i} \text{ if } Y_i = 0, \end{cases} \quad . \tag{3.1}$$

where ϕ_i is equal to $\phi(-\mu_i)$ where ϕ is the normal density and $\Phi_i = \Phi(-\mu_i)$ where Φ is the cumulative normal density.

Finishing the Latent Parameter Updates To finish the updates for the latent parameters, we note that $\mathrm{E}[\boldsymbol{\beta}] = \boldsymbol{\beta}^{\text{var}}$, so $\mu_i = \boldsymbol{X}_i\boldsymbol{\beta}^{\text{var}}$.

To assess convergence, we monitor change in the lower-bound. To obtain the bound for a particular model, it is helpful to write it as $\mathcal{L}(q) = \mathrm{E}[\log p(\boldsymbol{Y}^*, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})] - \mathrm{E}[\log q(\boldsymbol{\beta}, \boldsymbol{Y}^*]$ (Blei et al., 2003), or the expected value of the log-posterior (with the expectation taken over the variational distribution, averaging over both $\boldsymbol{\beta}$ and $\boldsymbol{Y}^*$) minus the expected value of the logged-approximating distribution, which is the entropy of the approximating distribution. Using the independence in the approximating distribution and the posterior we find that,

$$\mathcal{L}(q) = \mathrm{E}_{\boldsymbol{Y}^*,\boldsymbol{\beta}}\left[\log p(\boldsymbol{Y}^*|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\right] + \mathrm{E}_{\boldsymbol{Y}^*,\boldsymbol{\beta}}\left[\log p(\boldsymbol{\beta}|\boldsymbol{\beta}^{\text{prior}}, \frac{1}{\sigma^2}\boldsymbol{I})\right] - \mathrm{E}_{\boldsymbol{Y}^*,\boldsymbol{\beta}}\left[\log q(\boldsymbol{Y}^*)\right] - \mathrm{E}_{\boldsymbol{Y}^*,\boldsymbol{\beta}}\left[q(\boldsymbol{\beta})\right]$$

To carry out the remaining expectations, we rely upon well-known properties of normal distributions, but provide it to obtain property here: the entropy of a random variable, y distributed according to a truncated normal distribution with center μ and unit-variance is $\text{Entropy(Trunc. Normal)} = -\frac{1}{2}\left(\mathrm{E}[y^2] - 2\mu\mathrm{E}[y] + \mu^2 + \log(2\pi)\right) - I(y > 0)\left(\log(1 - \Phi(-\mu))\right) - I(y < 0)\left(\log(\Phi(-\mu))\right)$. Carrying out the expectations provides the lower bound

$$\begin{aligned}
\mathcal{L}(q) = &\underbrace{-\frac{1}{2}\text{Tr}\left(\boldsymbol{X}^{'}\boldsymbol{X}\left[\boldsymbol{\beta}^{\text{var}}(\boldsymbol{\beta}^{\text{var}})^{'} + \boldsymbol{\Sigma}^{\text{var}}\right]\right) + \sum_{i=1}^{N}\left[\frac{1}{2}\mu_i^2 + \log(1 - \Phi(-\mu_i))^{I(y_i=1)}\log(\Phi(-\mu_i))^{I(y_i=0)}\right]}_{\mathrm{E}[\log p(\boldsymbol{Y}^*|\boldsymbol{\beta},\boldsymbol{X},\boldsymbol{Y})] - \mathrm{E}[\log q(\boldsymbol{Y}^*)]} \\
&\underbrace{-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta}^{\text{var}})^{'}\boldsymbol{\beta}^{\text{var}} + \text{Tr}(\boldsymbol{\Sigma}^{\text{var}})\right] - \frac{M}{2}(\log 2\pi + \log\sigma^2)}_{\mathrm{E}[\log p(\boldsymbol{\beta})]} + \underbrace{\frac{1}{2}\log|\boldsymbol{\Sigma}^{\text{var}}| + \frac{1}{2}(1 + \log 2\pi)}_{\mathrm{E}[\log q(\boldsymbol{\beta})]}
\end{aligned}$$

3.1 Code for Variational Probit Model

This subsection provides example code for the variational probit regression

```
##The first code provides the lower bound on the data

##betas are the current posterior means
##of the approximating distribution on the coefficients
##X is the vector of covariates
##stars is the expected value of the
##augmented posterior approximation for each observation (E[y], y~q(y^{*}))
##sigs is the covariance matrix for
##the posterior approximating the coefficients
## draws are the dependent variable

len<- length
lower.bound2<- function(betas, X, stars, sigs, draws){

##the code is a straightforward implementation
##of the lower bound, divided into parts

ab<- t(X)%*%X%*%(betas%*%t(betas) + sigs)

part1<- sum(diag(ab))/2

part2<- t(betas)%*%betas + sum(diag(sigs))

part2<- part2*(beta.mat.prior[1,1])

part2<- part2/2 + (1/2)*log(det(solve(beta.mat.prior))) + (len(betas)/2)*log(2*pi)

part3<- t(stars)%*%stars

part4<- len(betas)/2 + (1/2)*log(det(sigs)) + (len(betas)/2)*log(2*pi)

bounds<-  part1  +  part2 + part3/2 + part4

parts<- c(-part1, -part2, part3/2, part4)

##we will return the lower bound and the constituent
##parts, useful for monitoring convergence of the algorithm

bounds<- list(bounds, parts)

names(bounds)<- c('bounds', 'parts')
```

```
return(bounds)
}


##we will use the lower-bound to monitor convergence,
##stopping the model when the lower-bound drops below 1e-8

func.reg<- function(X, draws){

##the variance covariance for the beta priors,
##can easily be added as an argument

beta.mat.prior<- diag(1/100, ncol(X))

##for teaching purposes, it is useful to store each updated
##mean vector for the beta approximating distribution

beta.VA<- matrix(NA, nrow=1000, ncol=ncol(X))

##we begin with random values for the augmented data

require(msm)

ystars<- rep(0, nrow(X))
for(j in 1:nrow(X)){
ystars[j]<- ifelse(draws[j]==1, rtnorm(1, mean=0.5, sd=1, lower=0, upper=Inf),
rtnorm(1, mean=-0.5, sd=1, lower=-Inf, upper=0) )
}
##we will store the progress of the lower bound on the model

bounds<- c()


zz<- 0

##this stores the parts of the lower bound

parts<- matrix(NA, nrow=1000,ncol=4)

j<- 0

##creating a while loop

while(zz==0){
j<- j + 1
```

```
##updating the beta parameters

beta.VA[j,]<- solve(t(X)%*%X + beta.mat.prior)%*%t(X)%*%ystars
##this does not need to be in the loop
##(it doesn't change over observations)
##but is placed here for teaching purposes

sigs<- solve(t(X)%*%X + beta.mat.prior)

##computing the inner product of the
##covariates current estimates of the coefficients

stars<- X%*%beta.VA[j,]

denom1<- pnorm(-stars)

num1<- dnorm(-stars)

##now, computing the expected value for each
##individual's augmented data, given
##current estimates of the approximating
##distribution on the coefficients

ystars[which(draws==0)]<- stars[draws==0] +
-num1[which(draws==0)]/denom1[which(draws==0)]

ystars[which(draws==1)]<- stars[draws==1] +
num1[which(draws==1)]/(1 - denom1[which(draws==1)])

##calculating the lower bound

trial<- lower.bound2(beta.VA[j,], X, ystars, sigs, draws)

bounds[j]<- trial$bounds

parts[j,]<- trial$parts

if(j>1){

##observing convergence

ab<-abs(bounds[j]-  bounds[j-1])

if(ab<1e-8){
```

```
zz<-1}

}

}

##the information to be returned, after convergence

stuff<- list(bounds, beta.VA[j,], sigs)

names(stuff)<- c('bound','betas', 'sigma')

return(stuff)

}

example.run<- func.reg(X, draws) ##where X are the covariates and draws are the dep
```

4 Derivation for the Voting Bloc Model

Update for $q(\boldsymbol{\tau}_i)^{\textbf{new}}$ Writing out the components of the log-posterior that depend upon $\boldsymbol{\tau}_i$ and taking the appropriate expectations,

$$\log q(\boldsymbol{\tau})_i^{\text{new}} = \sum_{k=1}^{K} \tau_{ik} \left[\text{E}[\log \pi_k] + \sum_{j=1}^{J} \left(V_{ij}\text{E}[\log \theta_{kj}] + (1 - V_{ij})\text{E}[\log(1 - \theta_{kj})] \right) \right] + \text{const.} \quad (4.1)$$

Define $\log \rho_{ik} = \text{E}[\log \pi_k] + \sum_{j=1}^{J}(V_{ij}\text{E}[\log \theta_{kj}] + (1 - V_{ij})\text{E}[\log(1 - \theta_{kj})$ and $r_{ik} = \frac{\rho_{ik}}{\sum_{k=1}^{K} \rho_{ik}}$. Exponentiating both sides of Equation 4.1 shows that $q(\boldsymbol{\tau})_i$ is a multinomial distribution, with the probability of senator i belong to bloc k given by r_{ik}. We will finish the update step for r_{ik} once we have obtained the distributional form for $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\theta}_{kj})$.

Update for $q(\boldsymbol{\pi})^{\textbf{new}}$ Writing out the components of the log-posterior that depend upon $\boldsymbol{\pi}$, we obtain

$$\log q(\boldsymbol{\pi})^{\text{new}} = \sum_{k=1}^{K} (\alpha_k - 1) \log \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} \text{E}[\tau_{ik}] \log \pi_k + \text{const.}$$

and the previous section showed that $\text{E}[\tau_{ik}] = r_{ik}$. This is the kernel of the Dirichlet distribution and therefore, $q(\boldsymbol{\pi}) \equiv$ Dirichlet($\boldsymbol{\lambda}$) distribution, with typical element $\lambda_k = \alpha_k + \sum_{i=1}^{N} r_{ik}$.

Update for $q(\boldsymbol{\theta}_{kj})^{\textbf{new}}$ Focusing on the terms in the posterior that depend upon θ_{kj},

$$\log q(\theta_{kj})^{\text{new}} = (\gamma_1 - 1) \log \theta_{kj} + (\gamma_2 - 1) \log(1 - \theta_{kj}) + \sum_{i=1}^{N} \text{E}[\tau_{ik}] \left[V_{ij} \log \theta_{kj} + (1 - V_{ij})(1 - \log \theta_{kj}) \right] + \text{const.}$$

This is the kernel of the Beta distribution. So, $q(\theta_{kj}) \equiv \text{Beta}(\eta_{kj1}, \eta_{kj2})$ where,

$$\eta_1 = \gamma_1 + \sum_{i=1}^{N} r_{ik} V_{ij} \quad ; \quad \eta_2 = \gamma_2 + \sum_{i=1}^{N} r_{ik}(1 - V_{ij}).$$

Completing $q(\boldsymbol{\tau}_i)^{\text{new}}$ To complete the algorithm, we compute,

$$\text{E}[\log \pi_k] = \Psi(\lambda_k) - \Psi(\sum_{z=1}^{K} \lambda_z)$$

(where $\Psi(\cdot)$ is the digamma function), and

$$\text{E}[\log \theta_{kj1}] = \Psi(\eta_{kj1}) - \Psi(\eta_{kj1} + \eta_{kj2}) \quad ; \quad \text{E}[\log \theta_{kj2}] = \Psi(\eta_{kj2}) - \Psi(\eta_{kj1} + \eta_{kj2}.$$

To summarize an iteration of the algorithm, we start with the values of $\boldsymbol{\lambda}^{\text{old}}$ and $\boldsymbol{\eta}_{kj}^{\text{old}}$ for all k and all j from a previous iteration (or initialize the values if this is the first step). We then obtain $q(\boldsymbol{\tau}_i)^{\text{new}}$ for all i, or equivalently update,

$$r_{ik}^{\text{new}} \quad \propto \quad \exp\left[\text{E}[\log \pi_k] + \sum_{j=1}^{J} \{V_{ij}\,[\text{E}[\log \theta_{kj1}]] + (1 - V_{ij})\,[\text{E}[\log \theta_{kj1}]]\}\right].$$

We then obtain $q(\boldsymbol{\pi})^{\text{new}}$, which is equivalent to updating the parameters

$$\lambda_k^{\text{new}} \quad = \quad \alpha_k + \sum_{i=1}^{N} r_{ik}^{\text{new}}$$

And finally we obtain $q(\boldsymbol{\theta})^{\text{new}}$, which is equivalent to updating the parameters,

$$\begin{aligned} \theta_{kj1}^{\text{new}} &= \gamma_1 + \sum_{i=1}^{N} r_{ik}^{\text{new}} V_{ij} \\ \theta_{kj2}^{\text{new}} &= \gamma_2 + \sum_{i=1}^{N} r_{ik}^{\text{new}}(1 - V_{ij}) \end{aligned}$$

I implemented and applied the voting bloc model to roll call data from the US Senate during the 110th Congress (2007-2008), setting $\gamma_1 = \gamma_2 = 1$ and $\alpha_k = 1$ for all k. Figure **??** shows that the algorithm quickly converges to the posterior (where convergence is monitored using the lower-bound on the log-probability of the data.

We now provide the lower-bound. To do so, we rely upon the following factorization of the posterior (implicit in the model described above),

$$p(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{V}|\boldsymbol{\alpha}, \boldsymbol{\gamma}) \quad = \quad p(\boldsymbol{\theta}|\boldsymbol{\gamma})p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(\boldsymbol{\tau}|\boldsymbol{\pi})p(\boldsymbol{V}|\boldsymbol{\tau}, \boldsymbol{\theta})$$

The lower-bound is given by,

$$\begin{aligned}
\mathcal{L}(q) \quad = \quad & \underbrace{\log\Gamma(\sum_{m=1}^{2}\gamma_m) - \sum_{m=1}^{2}\log\Gamma(\gamma_m) + \sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{z=1}^{2}\left\{(\gamma_z - 1)\left[\Psi(\eta_{kjz}) - \Psi(\sum_{m=1}^{2}\eta_{kjm})\right]\right\}}_{\mathrm{E}[\log p(\boldsymbol{\theta}|\boldsymbol{\lambda})]} \\
& \underbrace{+\log\Gamma(\sum_{k=1}^{K}\alpha_k) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k - 1)\left[\lambda_k - \Psi(\sum_{z=1}^{K}\lambda_z)\right]}_{\mathrm{E}[\log p(\boldsymbol{\pi}|\boldsymbol{\alpha})]} \\
& \underbrace{+\sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}(\Psi(\lambda_k) - \Psi(\sum_{z=1}^{K}\lambda_z))}_{\mathrm{E}[\log p(\boldsymbol{\tau}|\boldsymbol{\pi})]} \\
& \underbrace{+\sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\sum_{j=1}^{J}(V_{ij}[\Psi(\eta_{kj1}) - \Psi(\eta_{kj1}+\eta_{kj2})] + (1 - V_{ij})[\Psi(\eta_{kj2}) - \Psi(\eta_{kj1}+\eta_{kj2})])}_{\mathrm{E}[\log p(\boldsymbol{V}|\boldsymbol{\pi},\boldsymbol{\tau})]} \\
& \underbrace{-\sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\log r_{ik}}_{\mathrm{E}[\log q(\boldsymbol{\tau})]} \underbrace{-\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{z=1}^{2}(\eta_{kjz} - 1)[\Psi(\eta_{kjz}) - \Psi(\eta_{kj1}+\eta_{kj2})]}_{\mathrm{E}\log q(\boldsymbol{\theta})} \underbrace{-\sum_{k=1}^{K}(\lambda_k - 1)(\Psi(\lambda_k) - \Psi(\sum_{z=1}^{K}\lambda_z))}_{\mathrm{E}[\log q(\boldsymbol{\pi})]}
\end{aligned}$$

4.1 Code for Voting Bloc Model

```
##the code to implement the lower bound
##again, this is straightforward implementation
##with parameters that are identical to the text
##with votes comprising the fixed dependent variable
##in the analysis

len<- length
lower.bound<- function(lambda, taus, etas){
digam.lam<- lambda
digam.lam<- digamma(digam.lam) - digamma(sum(digam.lam))
part1<- 0
for(i in 1:nrow(taus)){
part1<- part1 + taus[i,]%*%digam.lam}
part1<- c(part1)
part2<- 0
digam.array<- etas
for(m in 1:k){
digam.array[,,m]<- digamma(etas[,,m]) - digamma(apply(etas[,,m], 1, sum))
}
```

```
for(i in 1:nrow(votes)){
for(m in 1:k){
part2<- part2 + taus[i,m]*(sum(votes[i,]*digam.array[,1,m]
+ (1- votes[i,])*digam.array[,2,m], na.rm=T))
}
}
part3<- -lgamma(sum(lambda))
part4<-  sum(lgamma(lambda))
part5<- - sum((lambda-1)*digam.lam)
part6<- 0
for(j in 1:k){
part6<- part6 + sum(lgamma(apply(etas[,,j], 1, sum))) - sum(lgamma(etas[,,j]))
}
part6<- part6*-1
part7<- -sum(digam.array*etas)
taus.smooth<- taus
for(j in 1:102){
taus.smooth[j,]<- taus.smooth[j,] + 1e-06
taus.smooth[j,]<- taus.smooth[j,]/sum(taus.smooth[j,])
}
part8<- taus.smooth*log(taus.smooth)
part8<- -sum(part8)
part9<- lgamma(k) - lgamma(1)*k; part10 <- lgamma(2)*k*j
l.bound<- part1+ part2 + part3 + part4 +
 part5 + part6 + part7 + part8 + part9 + part10
return(l.bound)
}


##this next section uses MCMCpack
library(MCMCpack)
func.block<- function(k, alpha, gamma1, gamma2, votes){
##creating location to store parameters
taus<- matrix(NA, nrow = nrow(votes), ncol=k)
lambda<- rgamma(k, shape=1, rate=1)
etas<- array(NA, dim=c(ncol(votes), 2, k))
for(m in 1:k){
for(j in 1:nrow(etas)){
etas[j,,m]<- rgamma(2, shape=1, rate=1)
}
}
bound<- c()
z<- 0
counter<- 0
##while loop for estimation
```

```
while(z ==0){
counter<- counter + 1
if(counter>1){
taus.old<- taus}
digam.lam<- digamma(lambda)- digamma(sum(lambda))
digam.array<- etas
for(m in 1:k){
digam.array[,,m]<- digamma(etas[,,m]) - digamma(apply(etas[,,m], 1, sum))
}
for(i in 1:nrow(votes)){
part1<- digam.lam
 for(m in 1:k){
part1[m]<- part1[m] + sum(votes[i,]*digam.array[,1,m], na.rm=T) +
sum((1-votes[i,])*digam.array[,2,m], na.rm=T)
}
part1<- part1 - max(part1)
taus[i,]<- exp(part1)/sum(exp(part1))
}
lambda<- alpha + apply(taus, 2, sum)
for(j in 1:k){
for(m in 1:ncol(votes)){
etas[m,1,j]<- gamma1 + sum(taus[,j]*votes[,m], na.rm=T)
etas[m,2,j]<- gamma2 + sum(taus[,j]*(1-votes[,m]), na.rm=T)
}
}
##storing the current value of the lower bound
bound[counter]<- lower.bound(lambda, taus, etas)
if(counter>1){
diff<- abs(bound[counter] - bound[counter-1])
if(diff<1e-10){
z<- 1}
}
}
output<- list(taus, lambda, etas, bound)
names(output)<- c('taus', 'lambda', 'etas','bound')
return(output)
}

##example run

example.run<-  func.block(k=10, alpha=1,
\begin{flushright}

\end{flushright}gamma1=1, gamma2=1, votes=votes)
```

5 Update Steps for Dirichlet Process Prior

This section describes the update steps for the nonparametric topic model, developed in Section 6.

Before starting, note that we can write $p(\boldsymbol{\tau}_i|\boldsymbol{v})$ as $p(\boldsymbol{\tau}_i|\boldsymbol{v}) = \prod_{j=1}^{\infty} \left[(1-v_j)^{\mathrm{I}(\tau_i>j)} v_j^{\mathrm{I}(\tau_i=j)}\right]$ (Blei and Jordan, 2006).

Update for $q(\boldsymbol{\tau})^{\text{new}}$ Taking the logarithm of the posterior and taking expectations we find that

$$\begin{aligned}\log q(\boldsymbol{T})^{\text{new}} &= \sum_{i=1}^{N}\sum_{k=1}^{K} \tau_{i,k} \left(\mathrm{E}[\log v_k] + \mathrm{E}[\log(1-v_k)]\right) \\ &\quad + \sum_{i=1}^{N}\sum_{k=1}^{K} \tau_{i,k} y_{i,k} \mathrm{E}[\log \boldsymbol{\theta}_k] + \text{constants}.\end{aligned}$$

Define $\log \zeta_{i,k} = \mathrm{E}[\log v_k] + \mathrm{E}[\log(1-v_k)] + y_{i,k}\mathrm{E}[\log \boldsymbol{\theta}_k]$ and $r_{i,k} = \frac{\zeta_{i,k}}{\sum_{k=1}^{K} \zeta_{i,k}}$. Taking the exponential of both sides we find that $q(\boldsymbol{\tau})$ is a multinomial distribution. To complete the update steps, we first determine the functional forms of $q(\boldsymbol{v})$ and $q(\boldsymbol{\theta})$.

Update for $q(\boldsymbol{v})^{\text{new}}$ Taking the logarithm of the posterior and taking the appropriate expectations we find that

$$\log q(\boldsymbol{v}_k)^{\text{new}} = (\mathrm{E}[\alpha]-1)\log(1-v_k) + \sum_{i=1}^{N} \log(1-v_k) \sum_{j=k+1}^{K} r_{i,j} + \sum_{i=1}^{N} r_{i,k} v_k$$

which is the kernel of a Beta distribution (Blei and Jordan, 2006). Define $\gamma_{k,1} = 1 + \sum_{i=1}^{N} r_{i,k}$ and $\gamma_{k,2} = \mathrm{E}(\alpha) + \sum_{i=1}^{N}\sum_{j=k+1}^{K} r_{i,j}$. Then $q(\boldsymbol{v}_k) \equiv \text{Beta}(\gamma_{k,1}, \gamma_{k,2})$.

Update for $q(\boldsymbol{\theta}_k)$ Taking the logarithm and carrying out expectations we find

$$\log q(\boldsymbol{\theta}_k^{\text{new}}) = \sum_{j=1}^{w} (\lambda_j - 1)\log \theta_{k,j} + \sum_{i=1}^{N}\sum_{j=1}^{w} r_{i,k} y_{i,j} \log \theta_{k,j}.$$

This the kernel of a Dirichlet distribution. Define $\boldsymbol{\eta}_k = \boldsymbol{\lambda} + \sum_{i=1}^{N} r_{i,k}\boldsymbol{y}_i$. Then, we have $q(\boldsymbol{\theta}_k) = \text{Dirichlet}(\boldsymbol{\eta}_k)$.

Update for $q(\alpha)$ Taking the logarithm and expectations we have

$$\log q(\alpha)^{\text{new}} = (\alpha-1)\log s_1 + s_1 \log s_2 - \alpha s_2 - \log\Gamma(\alpha) + \sum_{k=1}^{K-1} (\alpha-1)\left[\Psi(\gamma_{k,2}) - \Psi(\gamma_{k,1}+\gamma_{k,1})\right].$$

This is a Gamma(w_1, w_2) distribution , where $w_1 = s + K - 1$ and $w_2 = s_2 - \sum_{k=1}^{K-1}\left[\Psi(\gamma_{k,2}) - \Psi(\gamma_{k,1}+\gamma_{k,1})\right]$ and therefore $\mathrm{E}[\alpha] = \frac{w_1}{w_2}$.

Completing updates for $q(\boldsymbol{\tau})^{\text{new}}$ Carrying out the expectations, we have that

$$\begin{aligned} \text{E}[\log v_k] &= \Psi(\gamma_{1,k}) - \Psi(\gamma_{1,k} + \gamma_{2,k}) \\ \text{E}[\log(1 - v_k)] &= \Psi(\gamma_{2,k}) - \Psi(\gamma_{1,k} + \gamma_{2,k}) \\ \text{E}[\log \boldsymbol{\theta}_k] &= \sum_{j=1}^{w} \Psi(\eta_{k,j}) - \Psi(\sum_{m=1}^{w} \eta_{k,m}) \end{aligned}$$

We repeat the update steps until convergence, which is evaluated by evaluating the lower-bound $\mathcal{L}(q)$.

6 Preprocessing the Documents

In this section I describe how the 64,033 press releases are translated into count vectors. The first preprocessing step discards the order of words in the press release, leaving an unordered set of words remaining (Quinn et al., 2010; Hopkins and King, 2007). While one might expect the order of words to be crucial to understanding the sentiment expressed in a text, identifying the topic of a press release should be invariant to permutations of word order. Certain topics, such as the Iraq war, should result in specific words appearing with high frequency (`troop, war, iraqi`) irrespective of whether the senator supports or opposes the war.

After discarding word order, a series of steps are performed to group words together that differ only due to the way they are used in the document. First, all the words are placed into lower case and all punctuation is removed. Then, I applied the Porter stemming algorithm to each word (Porter, 1980). The stemming algorithm takes as an input a word and returns the word's basic building block, or *stem*. For example, the stemming algorithm takes the words `family`, `families` and returns `famili`.

After stemming the words in each document, I counted the number of occurrences of each word in the *corpus*, the total set of press releases. All words that do not occur in at least 0.5 % of addresses were removed (Quinn et al., 2010). Finally, I removed all *stop* words (eg, `around, whereas, why, whether`).

References

Albert, J.H. and S. Chib. 1993. "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association* 88(422):669–679.

Bishop, Christopher. 1999. "Variational Principal Components." *Proceedings Ninth International Conference on Artificial Neural Networks* 1:509–514.

Bishop, Christopher. 2006. *Pattern Recognition and Machine Learning*. Springer.

Blei, David and John Lafferty. 2006. "Dynamic Topic Models." *Proceedings of the 23rd International Conference on Machine Learning* 23.

Blei, David and Michael Jordan. 2006. "Variational Inference for Dirichlet Process Mixtures." *Journal of Bayesian Analysis* 1(1):121–144.

Blei, David et al. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning and Research* 3:993–1022.

Gelman, Andrew, John Carlin, Hal Stern and Donal Rubin. 1995. *Bayesian Data Analysis*. Chapman & Hall.

Hopkins, Daniel and Gary King. 2007. "Extracting Systematic Social Science Meaning from Text.".

Jackman, Simon. 2000. “Estimation and Inference via Bayesian Simulation: An Introduction to Markov Chain Monte Carlo.” *American Journal of Political Science* 44(2):375–404.

Jordan, Michael et al. 1999. “An Introduction to Variational Methods for Graphical Models.” *Machine Learning* 37:183–233.

Porter, Martin. 1980. “An Algorithm for Suffix Stripping.” *Program* 14(3):130–137.

Quinn, Kevin et al. 2010. “How to Analyze Political Attention with Minimal Assumptions and Costs.” *American Journal of Political Science* 54(1):209–228.

Wang, Bo and DM Titterington. 2004. “Convergence and Asymptotic Normality of Variational Bayesian Approximations for Exponential Family Models With Missing Values.” *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* 20:577–584.