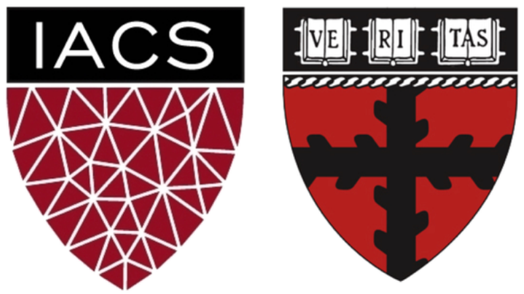
Lecture #10: Bayesian Latent Variable Models and Variational Inference

AM 207: Advanced Scientific Computing

Stochastic Methods for Data Analysis, Inference and Optimization

Fall, 2020


IACS
VE RI TAS

Outline

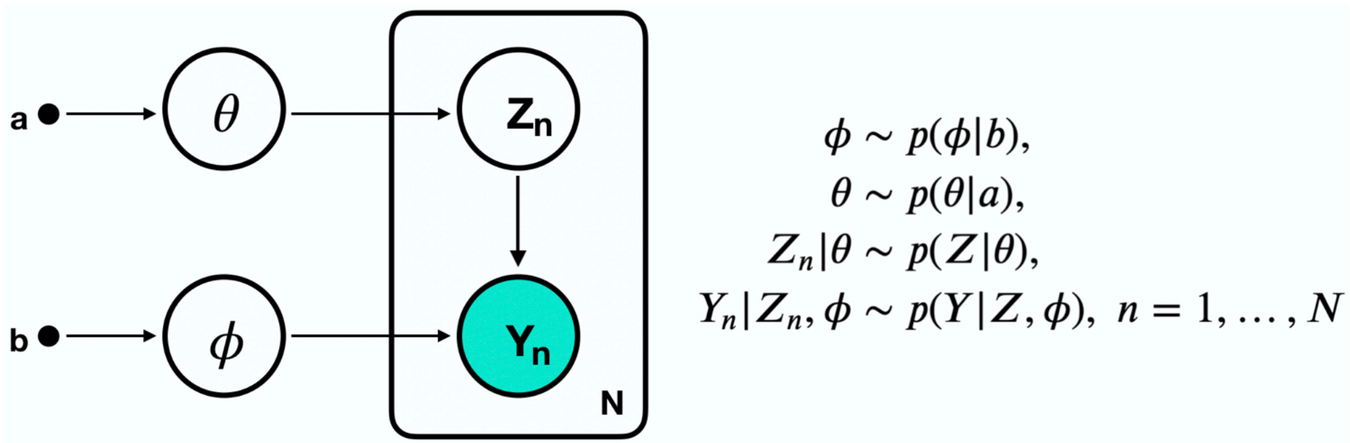
1. Bayesian Latent Variable Models
2. Coordinate Ascent Variational Inference
3. Bayesian Gaussian Mixture Models

Bayesian Latent Variable Models

Bayesian Latent Variable Models

Overfitting is an always a concern when using MLE model parameters. We can mitigate the effect of outliers in the data on the model we learn by treating the parameters as random variables and placing priors on them.

In a latent variable model, maximum liklihood inference treats parameters θ, ϕ as unknown constants and produces point-estimates for them. In a Bayesian latent variable model, θ, ϕ are random variables and we derive the posterior distribution over them.



$$
\begin{aligned}
\phi &\sim p(\phi|b), \\
\theta &\sim p(\theta|a), \\
Z_n|\theta &\sim p(Z|\theta), \\
Y_n|Z_n, \phi &\sim p(Y|Z, \phi), \ n = 1, \ldots, N
\end{aligned}
$$

That is, we want to infer

$$
p(\theta, \phi, Z_1, \ldots, Z_N|Y_1, \ldots, Y_N, a, b) = \frac{p(\theta|a)p(\phi|b)\prod_n p(Y_n|Z_n, \phi)p(Z_n|\theta)}{\prod_n p(Y_n)}.
$$

Challenges in Bayesian Inference

Unfortunately, most Bayesian models with multiple types of random variables (like Bayesian latent variable models) have complex posteriors that do not match known distributions. ***Exact inference*** is not possible.

Sampling from the posterior may not always be the best option because:

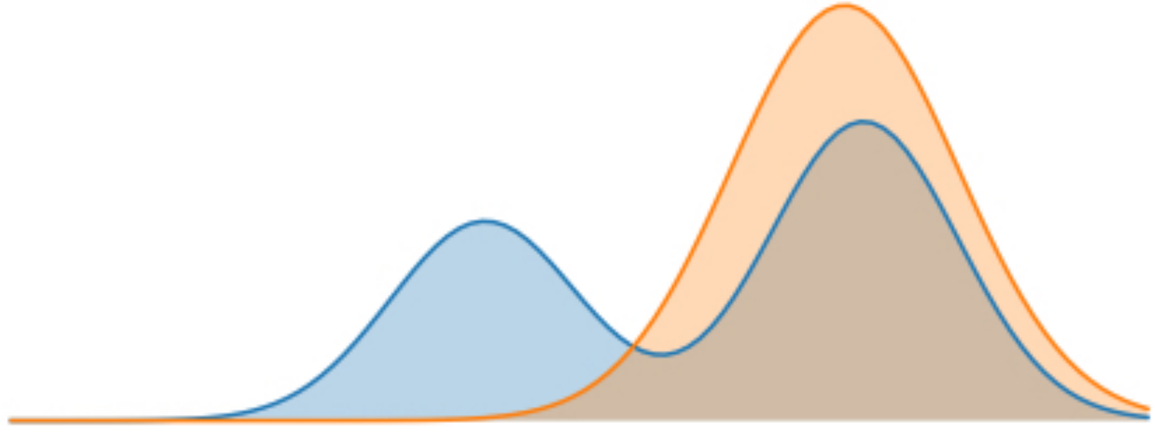
1. Convergence of samplers may be slow (due to high dimensionality of the distribution or multimodality)

2. Samplers like Metropolis-Hastings requires evaluating the liklihood $\prod_n p(Y_n|Z_n, \phi)$ in each iteration, if the observed data is large (N is in the millions), this computation is expensive.

The Idea of Variational Inference

Idea: (Approximate Inferenec) Approximate the hard posterior $p(\theta, \phi, Z_1, \ldots, Z_N | Y_1, \ldots, Y_N)$ with a distribution q that is easy to sample from (like a Gaussian). Any computation involving the posterior can now be done with q.

This approximation of $p(\theta, \phi, Z_1, \ldots, Z_N | Y_1, \ldots, Y_N)$ with a distribution q is called ***variational inference***.



The Design of the Variational Objective

Goal: given a target posterior distribution $p(\psi|Y_1, \ldots, Y_N), \psi \in \mathbb{R}^I$ we want to find a distribution $q(\psi|\lambda^*)$ in a family of distributions $Q = \{q(\psi|\lambda)|\lambda \in \Lambda\}$ such that $q(\psi|\lambda^*)$ best approximates p.

Design Choices: we need to choose:

A. ***(Variational family)*** a family Q of candidate distributions for approximating p. The members of Q are called the ***variational distributions***.

Our Choice: we assume that the joint $q(\psi)$ factorizes completely over each dimension of ψ, i.e. $q(\psi) = \prod_{i=1}^{I} q(\psi_i|\lambda_i)$. This is called the ***mean field assumption***. What can go wrong with this design choice?

B. ***(Divergence measure)*** a divergence measure to quantify the difference between p and q.

Our Choice:

$$D_{\mathrm{KL}}(q(\psi|\lambda)\|p(\psi|Y_1, \ldots, Y_N)) = \mathbb{E}_{\psi\sim q(\psi|\lambda)}\left[\log\left(\frac{q(\psi|\lambda)}{p(\psi|Y_1, \ldots, Y_N)}\right)\right]$$

What can go wrong with this design choice?

Variational Inference as Optimization

We now formalize variational inference for a target $p(\psi)$: find $q(\psi|\lambda^*)$ where

$$\begin{aligned}\lambda^* &= \operatorname*{argmin}_{\lambda} \; D_{\mathrm{KL}}(q(\psi|\lambda)\|p(\psi|Y_1, \ldots, Y_N))) \\ &= \operatorname*{argmin}_{\lambda} \; \mathbb{E}_{\psi\sim q(\psi|\lambda)} \left[\log\left(\frac{q(\psi|\lambda)}{p(\psi|Y_1, \ldots, Y_N))}\right)\right]\end{aligned}$$

Recall that for EM, we had proved that minimizing the KL is equivalent to maximizing the ELBO (for which it is easier to compute the gradient). We will do the same here:

$$\begin{aligned}\min_{\lambda} D_{\mathrm{KL}}(q(\psi|\lambda)\|p(\psi|Y_1, \ldots, Y_N))) &\overset{\text{equiv}}{\equiv} \max_{\lambda} -D_{\mathrm{KL}}(q(\psi|\lambda)\|p(\psi|Y_1, \ldots, Y_N))) \\ &= \max_{\lambda} -\mathbb{E}_{\psi\sim q(\psi|\lambda)} \left[\log\left(\frac{q(\psi|\lambda)}{p(\psi|Y_1, \ldots, Y_N))}\right)\right] \\ &= \max_{\lambda} \underbrace{\mathbb{E}_{\psi\sim q(\psi|\lambda)} \left[\log\left(\frac{p(\psi, Y_1, \ldots, Y_N))}{q(\psi|\lambda)}\right)\right]}_{ELBO(\lambda)} \\ &\quad - \log p(Y_1, \ldots, Y_N).\end{aligned}$$

Thus, the variational objective can be rephrased as maximizing the $ELBO$.

Gradients of the ELBO

Unfortunately, the ELBO for variational inference of the posterior does not have easy gradients,

$$\nabla_\lambda \underbrace{\mathbb{E}_{\psi \sim q(\psi|\lambda)}\left[\log\left(\frac{p(\psi, Y_1, \ldots, Y_N)}{q(\psi|\lambda)}\right)\right]}_{ELBO(\lambda)}.$$

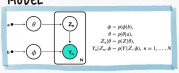
In particular, the issue is that the gradient taken is with respect to the parameter ψ of the distribution over which we are taking the expectation - i.e. we cannot push the gradient into the expectation.

Today we will maximize the $ELBO$ using coordinate ascent (just as in the case of EM). But you'll see that ***coordinate ascent variational inference*** requires that we perform model specific computations (often in closed form). This restrict the class of Bayesian models for which we can perform variational inference.

Two of the major development we will cover later in the semester address how to estimate this gradient **efficiently and without bias**.

Coordinate Ascent Variational Inference

MODEL



Let $\Psi = \{\psi, \theta, z_1 \ldots z_N$

Variational family $Q = \{q(\Psi|\lambda) \mid \lambda \in \Delta\}$

Goal: $\lambda^* = \arg\min_\lambda D_{KL}[p(\Psi|\text{Data}) \,\|\, q(\Psi|\lambda)]$

Problem: we don't know $q(\Psi|\lambda)$ and we can't evaluate it!

Solution: instead of min. D_{KL} we max ELBO

$$\arg\max_\lambda \underbrace{\mathop{\mathbb{E}}_{\Psi\sim q(\Psi|\lambda)}\left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right]}_{\text{ELBO}}$$

Why? ELBO involves the joint:

$p(\Psi, \text{Data}) = p(\text{Data}|\Psi)\,p(\Psi)$

The joint is tractable while posterior is not.

Assumptions: mean-field

$$q(\Psi|\lambda) = \prod_{i=1} q_i(\psi_i|\lambda_i)$$

How: right now we can only optimize this analytically, so we do coordinate-ascent on ELBO, because this is easier!

iterate:

$$\max_{\lambda_i} \text{ELBO}(\psi_i|\psi_{-i})$$

Claim: $\mathop{\mathbb{E}}_{\Psi\sim q(\Psi|\lambda)}[\cdots] = \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\cdots]\right]$

$$\begin{aligned}
\mathop{\mathbb{E}}_{\Psi\sim q(\Psi|\lambda)}\left[\log\left(\frac{p(\Psi|\text{Data})}{q(\Psi|\lambda)}\right)\right] &= \int_\Psi \left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right] q(\Psi|\lambda)\,d\Psi \\
&= \int_{\psi_i}\int_{\Psi_{-i}} \left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right] q(\Psi|\lambda)\,d\Psi_{-i}\,d\psi_i \\
&= \int_{\psi_i}\int_{\Psi_{-i}} \left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right] q(\Psi|\lambda)\,d\Psi_{-i}\,d\psi_i \\
&= \int_{\psi_i}\int_{\Psi_{-i}} \left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right] q(\Psi_{-i}|\lambda_{-i})\,q(\psi_i|\lambda_i)\,d\Psi_{-i}\,d\psi_i \\
&= \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}\left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right]\right]
\end{aligned}$$

In fact, with a bit more algebra, we can show that:

$$\max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}\left[\log\left(\frac{p(\Psi|\text{Data})}{q(\Psi|\lambda)}\right)\right]\right] \equiv \max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}\left[\log\left(\frac{p(\Psi_i,\text{Data}|\Psi_{-i})}{q(\psi_i|\lambda_i)}\right)\right]\right]$$

Claim: $\max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}[\cdots] \equiv \min_{\lambda_i} D_{KL}[\cdots]$

$$\max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}\left[\log\left(\frac{p(\Psi_i,\text{Data}|\Psi_{-i})}{q(\psi_i|\lambda_i)}\right)\right]\right] \overset{\text{algebra}}{=} \max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})] - \log q(\psi_i|\lambda_i)\right]$$

$$\overset{\text{algebra}}{=} \max_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\log\left(\exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})] - \log q(\psi_i|\lambda_i)\right\}\right)\right]$$

$$\equiv \min_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\log\left(\frac{q(\psi_i|\lambda_i)}{\exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})]\right\}}\right)\right]$$

looks like D_{KL} but denominator is not a pdf!

$$= \min_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\log\left(\frac{q(\psi_i|\lambda_i)}{z\exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})]\right\}}\right)\right] \quad \text{where } z = \sum$$

make denominator into pdf with normalizing constant z!

$$= \min_{\lambda_i} \mathop{\mathbb{E}}_{\psi_i\sim q(\psi_i|\lambda_i)}\left[\log\left(\frac{q(\psi_i|\lambda_i)}{z\exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})]\right\}}\right)\right] + \log(z)$$

$$\equiv \min_{\lambda_i} D_{KL}\left[q(\psi_i|\lambda_i) \,\|\, z\exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})]\right\}\right]$$

Lesson: we solve for $\lambda^* = \arg\max_\lambda \underbrace{\mathop{\mathbb{E}}_{\Psi\sim q(\Psi|\lambda)}\left[\log\left(\frac{p(\Psi,\text{Data})}{q(\Psi|\lambda)}\right)\right]}_{\text{ELBO}}$ by coordinate ascent

each time updating λ_i by

$$\lambda_i^* = \arg\max_{\lambda_i} \text{ELBO}(\lambda_i, \lambda_{-i})$$

$$q_i(\psi_i|\lambda_i^*) \propto \exp\left\{\mathop{\mathbb{E}}_{\psi_{-i}\sim q(\psi_{-i}|\lambda_{-i})}[\log p(\Psi_i,\text{Data}|\Psi_{-i})]\right\}$$

Maximizing the ELBO via Coordinate Ascent

The coordinate ascent algorithm maximizes an objective function $ELBO(\lambda)$ by iteratively maximizing over λ_i, holding constant $\lambda_{-i} = [\lambda_1 \ \ldots \ \lambda_{i-1} \ \lambda_{i+1} \ \ldots \ \lambda_I]$.

The ***coordinate ascent variational inference algorithm***:

1. **Initialization:** pick an intial value $\lambda^{(0)}$
2. **Coordinate-wise maximization:**

   Repeat for $j = 1, \ldots, J$ iterations:

   Cycle thru $i = 1, \ldots, I$ coordinates:

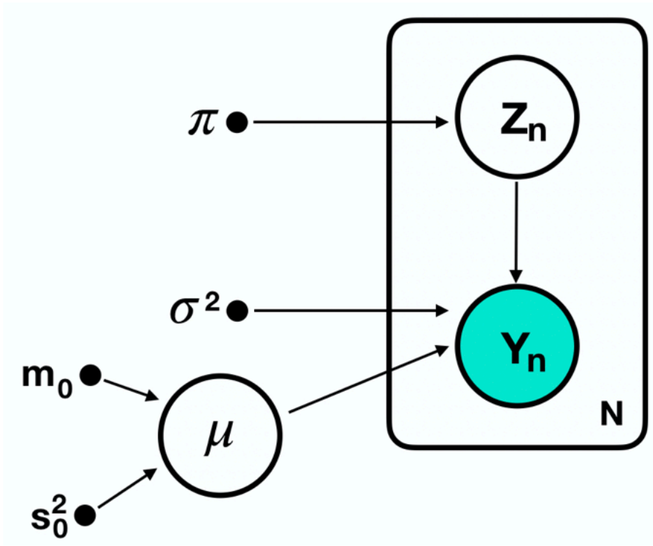
$$q(\psi_i | \lambda_i^{\text{new}}) \propto \exp\left\{ \mathbb{E}_{\psi_{-i} \sim q(\psi_{-i} | \lambda_1^{\text{new}}, \ldots, \lambda_{i-1}^{\text{new}}, \lambda_{i+1}^{\text{old}}, \ldots, \lambda_I^{\text{old}})} \left[\log p(Y_1, \ldots, Y_N, \psi) \right] \right\}.$$

where $\psi_{-i} = [\psi_1 \ \ldots \ \psi_{i-1} \ \psi_{i+1} \ \ldots \ \psi_I]$.

Bayesian Gaussian Mixture Models

Variational Inference for Bayesian Gaussian Mixture Models

We consider a Bayesian model for a mixture of K number of univariate Gaussians:



$$
\begin{aligned}
\mu_k &\sim \mathcal{N}(0, \sigma_0^2) \\
Z_n &\sim Cat\left(\left[\frac{1}{K}, \ldots, \frac{1}{K}\right]\right) \\
Y_n | Z_n, \mu &\sim \mathcal{N}(\mu_{Z_n}, \sigma^2 = 1) \\
n &= 1, \ldots, N, \\
k &= 1, \ldots, K, \\
\mu &= [\mu_1, \ldots, mu_K]
\end{aligned}
$$

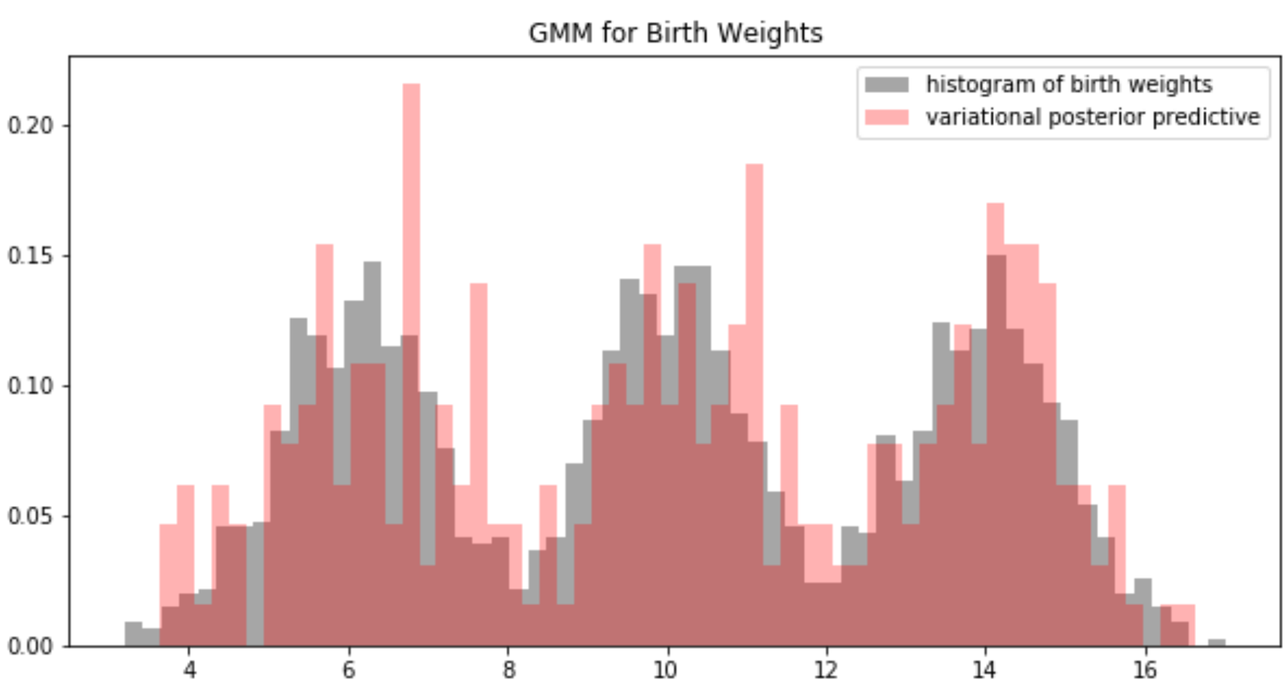
The ***hyperparameters*** of the models are π, σ^2, m_0, s_0^2, which are constants that must be selected prior to inference. For example, to simplify our computations we selected $\pi = [1/K, \ldots, 1/K], m_0 = 0, \sigma = 1$.

We make the mean field assumption -- that our variational posterior factorizes completely:

$$
q(Z, \mu | m, s^2, \phi) = \prod_{k=1}^{K} q(\mu_k | m_k, s_k^2) \prod_{n=1}^{N} q(Z_n | \phi_n).
$$

Implemenation of CAVI for Bayesian GMM

In [10]:
```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax.hist(y, bins=60, density=True, color='gray', alpha=0.7, label='histogram of b
irth weights')
posterior_predictive_samples = posterior_predictive_sampling(m_current, s_sq_cur
rent, 100)
ax.hist(posterior_predictive_samples, bins=60, density=True, color='red', alpha=
0.3, label='variational posterior predictive')
ax.set_title('GMM for Birth Weights')
ax.legend(loc='best')
plt.show()
```



Sanity Check: ELBO During Training

Remember that ploting the posterior predictive against actual data is not always an option (e.g. high-dimensional data).

A sanity check for that your CAVI algorithm has been implemented correctly is to plot the ELBO (or alternatively, the observed data log-likelihood) over the iterations of the algorithm:

$$ELBO(\phi, m, s^2) = \mathbb{E}_{Z,\mu \sim q(Z,\mu|\phi,m,s^2)} \left[\log \left(\frac{p(Y_1, \ldots, Y_N, Z_1, \ldots, Z_N, \mu)}{q(Z, \mu|\phi, m, s^2)} \right) \right]$$

In [12]:
```
fig, ax = plt.subplots(1, 1, figsize=(10, 2))
ax.plot(range(len(ELBOs)), ELBOs, color='red', alpha=0.5)
ax.set_title('ELBO over iterations of CAVI')
plt.show()
```