

Approximating The Posterior

Expectation Maximization and Friends

DS6040 Summer 2024
Teague R. Henry



SCHOOL *of* DATA SCIENCE

Outline

- Review
- Why Approximate?
- Unsupervised Clustering as Mixture Models
- The Expectation-Maximization Algorithm
- Variational Inference Writ Large

Review

We want to obtain the posterior $P(\theta|X)$:

- If we have conjugacy, we already *know* the posterior distribution.
 - It'll be the same distribution as the prior
 - It's parameters are a function of the data and the priors' hyperparameters
- If we don't have conjugacy, we can sample:
 - Samplers crawl over the posterior space until they find the posterior
 - Gibbs Samplers are for setups with proper conditionals
 - Metropolis/Metropolis Hastings Samplers are for any model setup
 - Hamiltonian MCMC is a modern way of sampling from high dimensional space.
Its like metropolis, but it uses that gradient information to guide it in high dimensional space.

the joint posterior might not have a known proper conditions. Here, the joint posterior is not properly defined

But what are some problems with samplers?

The conceptual understanding of what these samplers do will be on the midterm

Problems with Samplers

In order to sample from the posterior, a sampler has to find the posterior first.

- If your start values are really far away from the posterior, this might take some time.
- If you've got a really odd likelihood surface, you might not be able to traverse it with a sampler
- Categorical parameters (like class membership) are very finicky to sample from.
 - And the issues get much larger very quickly with increasing amounts of data.

To help with these issues, we can try mode approximation.

Mode Approximation

Instead of trying to sample from the posterior, we can try to find the mode of the posterior and use that to guide our sampler.

- Useful for determining starting values for a sampler

this is valuable b/c if we start really far away from our posterior, we might never know where our posterior is. We can use mode approximation to help

We also might have so much data, we don't care about our estimates of uncertainty. In that case, we can use the mode as a point estimate of the parameters.

- In this case, the mode of a posterior can be viewed as a penalized maximum likelihood estimate, with the penalty being the log of the prior distribution...

it turns out that you can do lasso, ridge, elastic net but setting a penalty

Our goal is to find the mode of the posterior

Optimizers!

When the model is simple, you can use your standard optimizers to find the mode of the posterior

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$

Newton-Raphson: this is an optimizer

Use gradient information of the log of the posterior to climb to the mode.

Eventually, this will converge to the mode of our posterior distribution

$$L(\theta|X) = \log(P(\theta|X))$$
$$\theta^t = \theta^{t-1} - [L''(\theta^{t-1})]^{-1} L'(\theta^{t-1})$$

You need to know what the first and second derivative of the log posterior is...

Teague learns that when the math becomes too hard, throw computing power at it.

- Not always easy to know!
- But you can use numerical derivatives to get it.

Mixture Models and Latent Variables

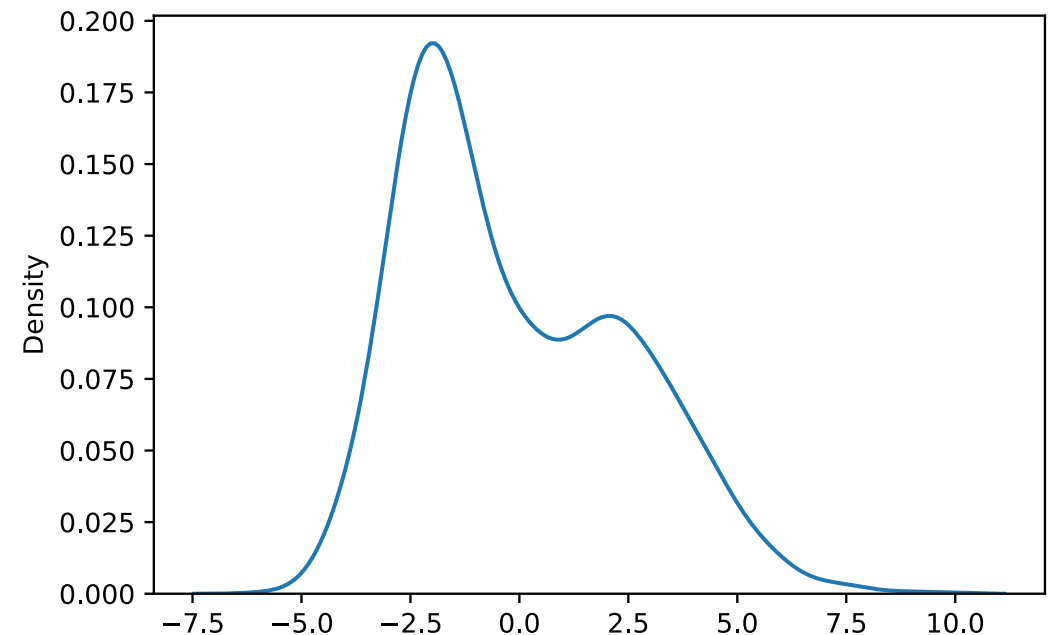
Data is rarely “nicely” distributed...

- Assumption – Once external factors are controlled for, the data is nicely distributed...

We rarely (never) have all possible important features.

- Sometimes this is ignorable...

How can we model data with features we need but don't have?



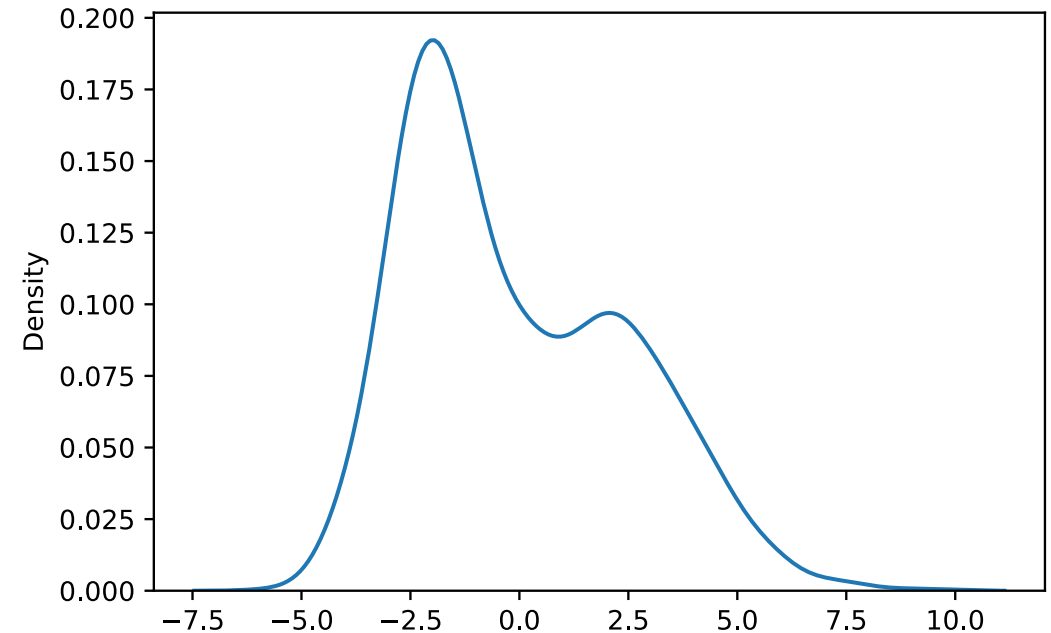
we can use these mixture models. A mixture model is a parameterized cluster model

Cases to Approximate the Mode

Consider a mixture model (a parameterized cluster model)

- The class membership of individual observations are all parameters...
- The parameters that govern the clusters depend on the observations in the clusters.
- The membership of observations depends on the membership of other observations!

we don't know what these clusters are, so our member ship depends on how these clusters are defined



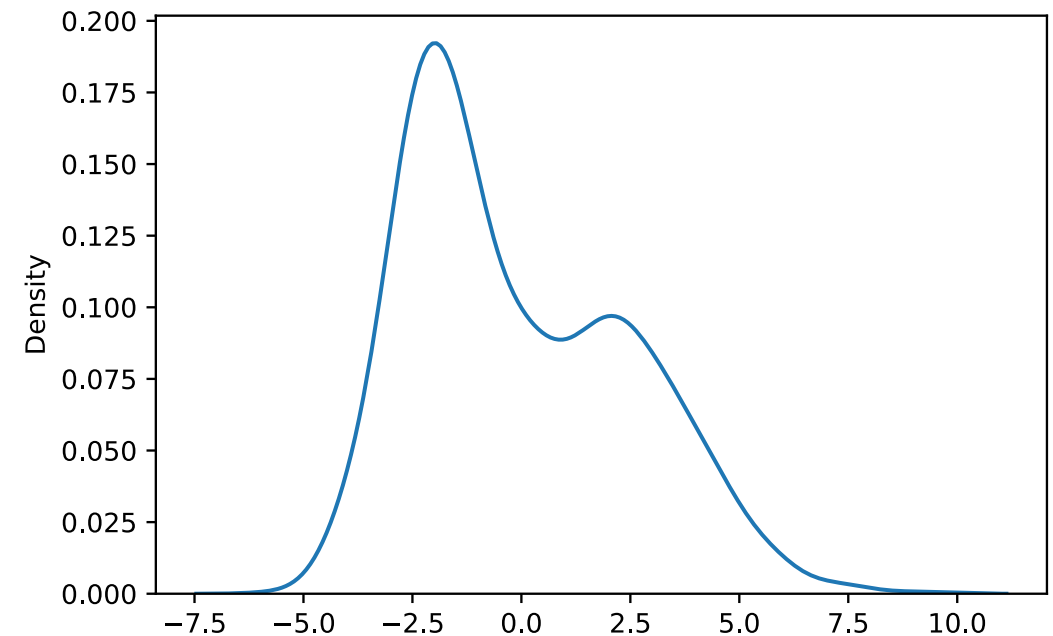
Very tricky to directly sample from!

Mixture Models and Latent Variables

Data Augmentation – Propose unobserved (latent) variables to explain your data...

Latent variables are defined by the relations you impose between them observed data.

Example – Mixture membership...



Simple Gaussian Mixture Model

Observed data - X_i

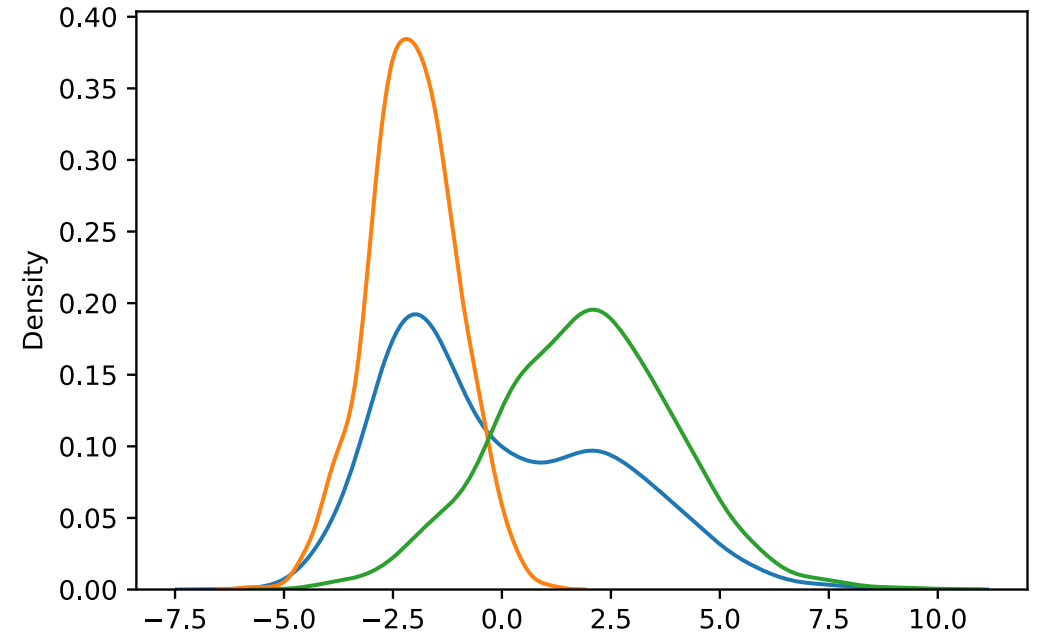
Model - $X_i \sim N(\mu_{z_i}, \sigma_{z_i}^2)$

What is z_i ?

- Not a parameter...
- Not observed data...

z_i is a latent variable we made up.

- We are making a strong claim here, that X_i is normal, conditional on z_i .



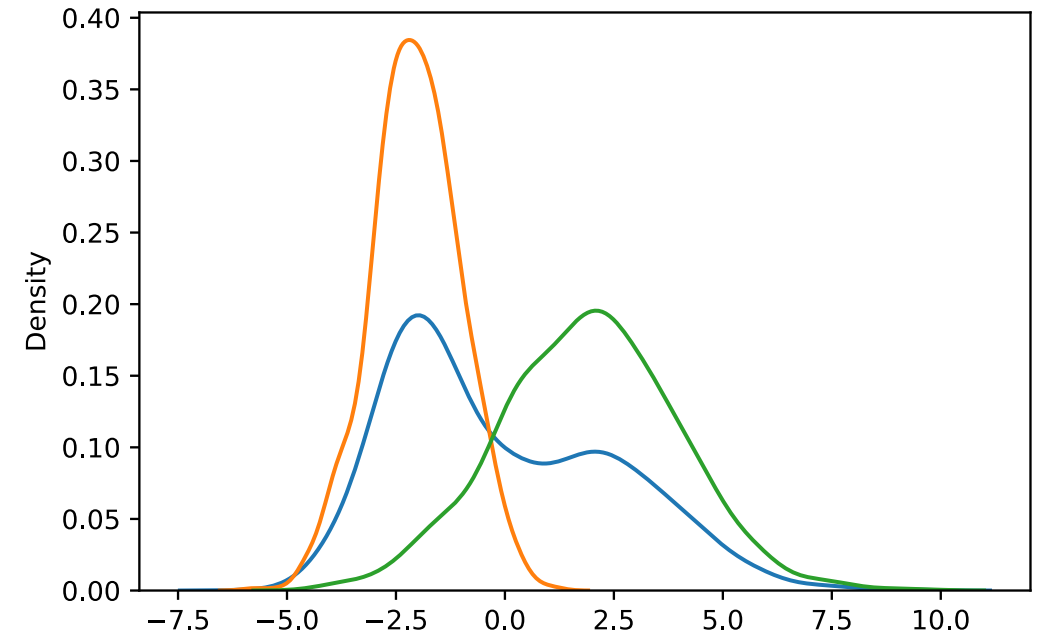
A Word of Warning

Beware clustering, mixture models and unsupervised learning methods...

Reification Fallacy -

Just because you find a cluster / component, doesn't mean it's a real thing...

Sometimes a weird distribution is just that, a weird distribution...



be careful to assign a mean to these clusters unless you have a good reason to these clusters should be well defined (have good separation). If you can circle clusters with a pencil, you can attempt to explain them.

The EM Algorithm

To determine the parameters of the mixture distributions, we need to know z_i , but to know z_i we need to determine the parameters of the mixture...

- What does this sound like? (Hint: Starts with “G” and ends with “ibbs Sampler”)

The Expectation Maximization Algorithm (Dempster, Laird, & Rubin (1976), 63963 *cites!*)

- Let \mathbf{X} be observed data, \mathbf{Z} be latent/missing data, and $\boldsymbol{\theta}$ be parameters.
- Likelihood: $p(\mathbf{X}|\boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})d\mathbf{Z} = \int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{X}|\boldsymbol{\theta})d\mathbf{Z}$ our issue is that we must remove \mathbf{Z} b/c we don't know it
- **Expectation Step:** Using your current estimate of θ_t , calculate $E[\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_t] = \mathbf{Z}_t$
- **Maximization Step:** Using your current estimate of \mathbf{Z}_t as observed, estimate MLE solution for $\boldsymbol{\theta}_{t+1}$ this is a generalized version of k-means

The EM Algorithm

why are we talking about mode approximation again? It's unlikely that we are doing these methods in practice. We can use these methods to examine the posterior space.

The Expectation Maximization Algorithm (Dempster, Laird, & Rubin (1976), 63963 *cites!*)

- Let \mathbf{X} be observed data, \mathbf{Z} be latent/missing data, and $\boldsymbol{\theta}$ be parameters.
- Likelihood: $p(\mathbf{X}|\boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})d\mathbf{Z} = \int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{X}|\boldsymbol{\theta})d\mathbf{Z}$
- Expectation Step: Using your current estimate of $\boldsymbol{\theta}_t$, calculate $E[\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}_t] = \mathbf{Z}_t$
- Maximization Step: Using your current estimate of \mathbf{Z}_t as observed, estimate MLE solution for $\boldsymbol{\theta}_{t+1}$

Advantages –

- By adding in “fake” data, you can make an intractable optimization problem tractable.
- EM algorithm is guaranteed to converge to a (local) minima. It just works, and works well.

Disadvantages –

- Uncertainty is underestimated, as you are feeding in \mathbf{Z}_t as known in each step.

Variational Inference

As we know, any “interesting” Bayesian model has an improper posterior.

Samplers – Construct a Markov Chain that samples from the posterior...

- Still approximate, but we are sampling from the full posterior.

only approximate b/c we have a limited number of samplers

Variational Inference

- Instead of trying to sample from an improper posterior, why don't we just... not?
- Let's instead approximate the posterior using a much simpler set of proper distributions
- That way, if we get a good approximation, we can calculate our EAP, MAP, and credible intervals directly.

Variational Inference – Technical Deets

Let $p(\theta|X)$ be our standard posterior distribution.

- Assume hard to sample from, and we don't have the analytic expression for it.

there are many ways that a posterior distribution would be hard to sample from. Ex. when you have to crunch large amounts of data down for each step and you have to do something many times.

Goal: Find a distribution $q(\theta)$ that approximates $p(\theta|X)$

- Specifically $q^*(\theta) = \operatorname{argmin}_{q(\theta) \in \mathcal{Q}} KL(q(\theta) || p(\theta|X))$
- We are looking for the distribution $q(\theta)$ that minimizes the KL divergence between it and the posterior distribution.
- But... (All expectations taken over $q(\theta)$)

KL divergence is a measure of distance between two distr. it is from information theory and basically says if KL is 0, distr. are the same. If anything is above this, then they are different

$$KL(q(\theta) || p(\theta|X)) = E[\log q(\theta)] - E[\log p(\theta \cap X)] + \log p(X)$$

The marginal strikes again!

boo marginal

Variational Inference – Technical Deets


$$KL(q(\theta)||p(\theta|X)) = E[\log q(\theta)] - E[\log p(\theta \cap X)] + \log(p(X))$$

Getting closer, but we still have that pesky marginal hanging around...

Rearranging

$$\log(p(X)) = KL(q(\theta)||p(\theta|X)) + E[\log p(\theta \cap X)] - E[\log q(\theta)]$$

KL is strictly positive, so the **Evidence Lower Bound** is defined as:


$$ELBO(q) = E[\log p(\theta \cap X)] - E[\log q(\theta)]$$

Using some probability rules:

$$ELBO(q) = E[\log p(X|\theta)] + E[\log p(\theta)] - E[\log q(\theta)]$$

Just uses the likelihood!

Just uses the priors!

Uses our variational distribution

approximate distributor

Variational Inference – Technical Deets

$$ELBO(q) = E[\log p(X|\theta)] + E[\log p(\theta)] - E[\log q(\theta)]$$

Maximizing the ELBO is equivalent to minimizing the KL divergence between $q(\theta)$ and $p(\theta|X)$.

To Recap:

- Approximate $p(\theta|X)$ using $q(\theta)$, the variational distribution.
- Choose $q(\theta)$ so that $ELBO(q)$ is maximized.
- Once you have your ELBO maximizing $q(\theta)$, you can treat $q(\theta)$ as your best approximation to $p(\theta|X)$
- How do you choose what $q(\theta)$ looks like?

Mean Field Approximation

How do you chose what $q(\theta)$ looks like?

we use a mean field approximation. Theta has its own set of governing principles.

- First, note that $q(\theta)$ refers to a distribution of θ , which has its own governing parameters η .
- The goal is to determine both the family of distributions and the values of η that maximize the ELBO...

Consider the situation when θ is **multidimensional** (i.e. there are more than one parameter we need a posterior for.)

Mean Field Approximation

Consider the situation when θ is multidimensional (i.e. there are more than one parameter we need a posterior for.)

- $p(\theta|\mathbf{X})$ is a complex distribution with dependencies between each marginal distribution.

Mean Field Approximation –

assumes parameters are not related to each other

$$q(\theta|\eta) = \prod_{i=1}^p q(\theta_i|\eta_i) \quad \text{the product of individual parameters}$$

The variational distributions of all parameters are independent from one another.

Mean Field Approximation

Mean Field Approximation –

$$q(\boldsymbol{\theta}|\boldsymbol{\eta}) = \prod_{i=1}^p q(\theta_i|\eta_i)$$

Advantages –

- Independent distributions are simpler to optimize η_i

all we're doing now is matching the location of these parameters independently

Disadvantages –

- Cannot capture dependency between posterior dimensions

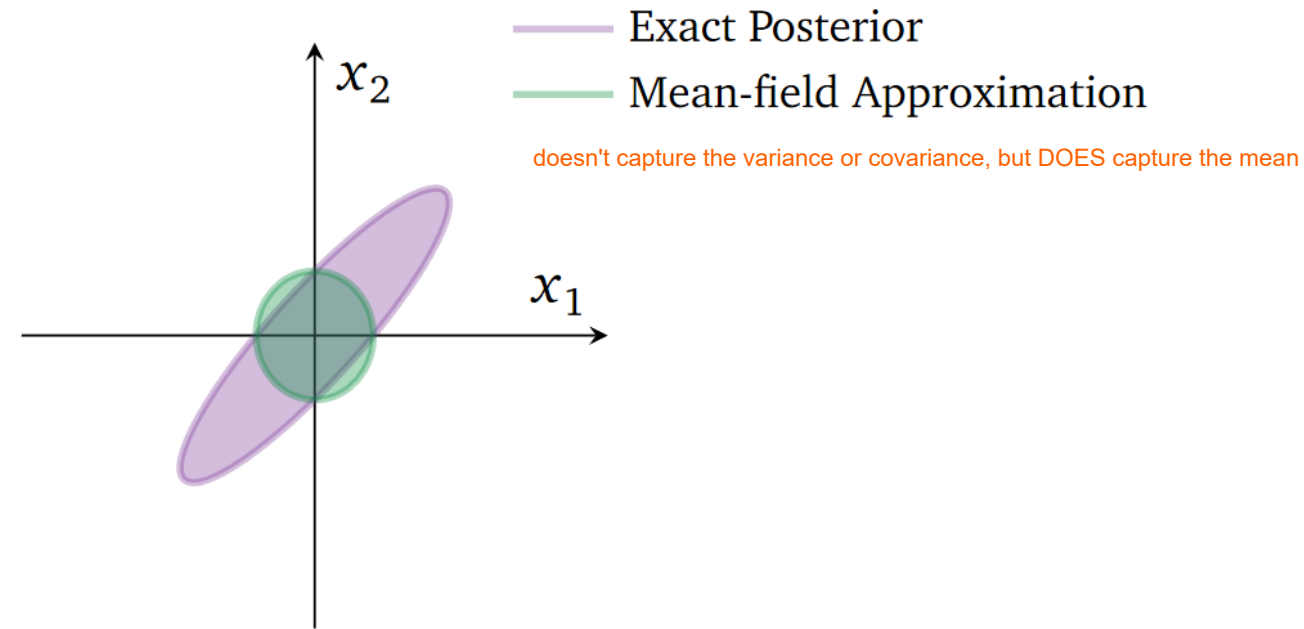


Figure from: <https://arxiv.org/pdf/1601.00670.pdf>
An excellent overview of variational inference ^

Determining $q^*(\theta_i)$

$$q^*(\theta_i) \propto \exp\{E_{-i}[\log p(\theta_i|X, \boldsymbol{\theta}_{-i})]\}$$

$$q^*(\theta_i) \propto \exp\{E_{-i}[\underbrace{\log p(X|\theta_i, \boldsymbol{\theta}_{-i})}_{\text{Just your likelihood!}} + \underbrace{\log p(\theta_i|\boldsymbol{\theta}_{-i})}_{\text{Given by your prior setup}} + \underbrace{\log p(\boldsymbol{\theta}_{-i})}_{\text{Constant with respect to } \theta_i}]\}$$

$$\log q^*(\theta_i) \propto E_{-i}[\log p(X|\theta_i, \boldsymbol{\theta}_{-i}) + \underbrace{\log p(\theta_i|\boldsymbol{\theta}_{-i})}_{\text{This is inside an expectation over } \boldsymbol{\theta}_{-i}}]$$

$$\log q^*(\theta_i) \propto E_{-i}[\log p(X|\theta_i, \boldsymbol{\theta}_{-i})] + \underbrace{\log p(\theta_i)}_{\text{Just the prior on } \theta_i}$$

The optimal $q^*(\theta_i)$ (using mean field approximation) is proportional to a function of the data (which is fixed) and fixed values of $\boldsymbol{\theta}_{-i}$

Optimizing ELBO using CAVI

$$\log q^*(\theta_i) \propto E_{-i}[\log p(X|\theta_i, \boldsymbol{\theta}_{-i})] + \log p(\theta_i)$$

This equation tells you the family of your variational distribution, and how the parameters η_i are a function of the data, priors and $\boldsymbol{\theta}_{-i}$

Coordinate Ascent Variational Inference (CAVI) – fixes the problem by finding optimal set of parameters

- For $i \in [1, \dots, p]$:
 - Find the optimal η_i using $E_{-i}[\log p(X|\theta_i, \boldsymbol{\theta}_{-i})] + \log p(\theta_i)$, holding $\boldsymbol{\theta}_{-i}$ fixed
- Repeat until the ELBO has converged.
- Similar in theory to a Gibbs Sampler, and identical to EM.

Summary

Variational Inference –

- Approximate $p(\boldsymbol{\theta}|X)$ using $q(\boldsymbol{\theta})$
- Assume that $q(\boldsymbol{\theta}) = \prod_{i=1}^p q(\theta_i)$ (Mean Field Approximation)
- Do math to determine optimal family and variational parameters η_i
 - $\log q^*(\theta_i) \propto E_{-i}[\log p(X|\theta_i, \boldsymbol{\theta}_{-i})] + \log p(\theta_i)$
- Update the optimal η_i using X and $\boldsymbol{\theta}_{-i}$ as fixed values.
- Continue to update until ELBO converges.

At the end of this you will have $q^*(\boldsymbol{\theta}|\boldsymbol{\eta}^*)$ as your approximation to $p(\boldsymbol{\theta}|\mathbf{X})$

- You can then directly calculate EAP, MAP, and credible intervals because $q^*(\boldsymbol{\theta}|\boldsymbol{\eta}^*)$ is, by design, a proper set of distributions (which are nicely independent of one another)

Next Week

Introduction to Stan!

- Fundamentals of probabilistic programming
- Basic syntax
- Running Stan code in R

Then, more Stan!

- Complex models in Stan
- Diagnostics
- Results interpretation