Admixture models and coordinate ascent variational inference

In this lecture we will derive a **coordinate ascent variational inference (CAVI) algorithm** to do approximate posterior inference in an **admixture model.**

More specifically, the model will be **latent Dirichlet allocation (LDA)**, the most well-known form of admixture model introduced independently by Pritchard, Stephens, Donnelly (2000) for statistical genetics and Blei, Ng, Jordan (2003) for topic modeling.

Admixture model

The data consist of "documents", or more generally groups of data points. For "document" d=1....D:

$$w_d \equiv (w_{d1} \cdots w_{dN_d})$$

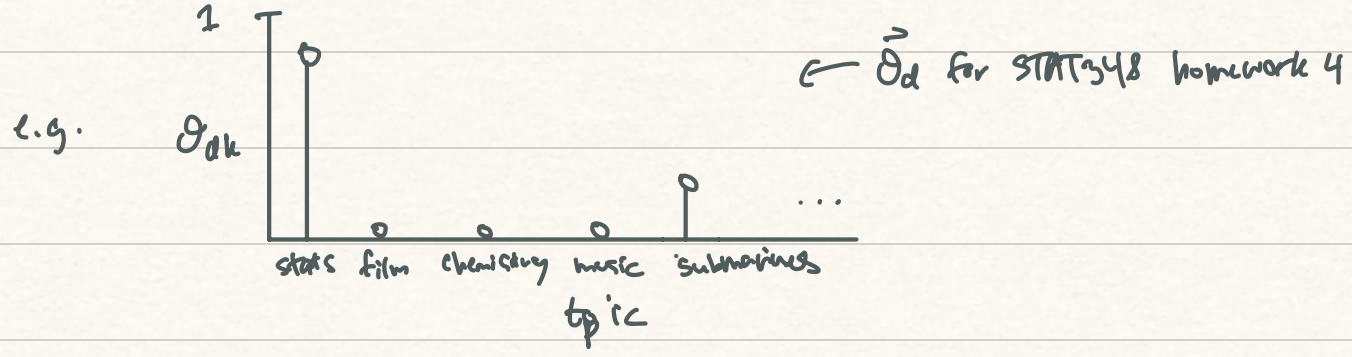
Each document is a group of Nd "word tokens", or more generally categorical data points. Each token i in document d is:

$$w_{di} \in \{1, \cdots, V\}$$

↑ vocabulary

We assume that each document can be described by a **mixture of K topics:**

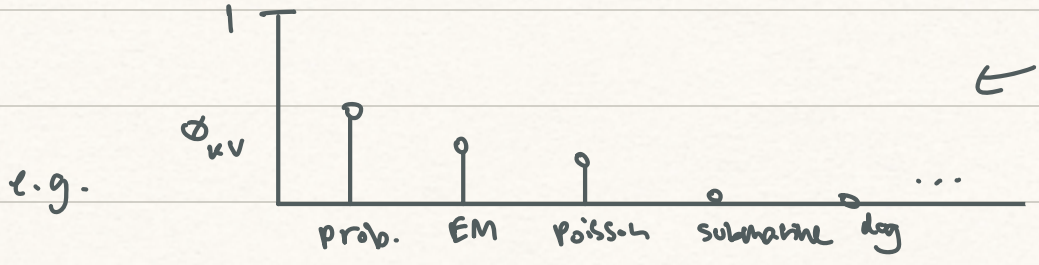
$$\vartheta_d \equiv (\vartheta_{d1} \cdots \vartheta_{dK}), \quad \sum_k \vartheta_{dk} = 1$$



← $\vec{\vartheta}_d$ for STAT 348 homework 4

Each "topic" is then defined by a distribution over the vocabulary:

$$\phi_k \equiv (\phi_{k1} \cdots \phi_{kV}), \quad \sum_v \phi_{kv} = 1$$



← $\vec{\phi}_k$ for "stats" topic

The generative process is as follows. We assume each word token was drawn from some topic:

for $d = 1 \dots D$:

for $i = 1 \dots N_d$:

$$z_{di} \sim \text{Categorical}(\vartheta_d)$$

$$w_{di} \sim \text{Categorical}(\varphi_{z_{di}})$$

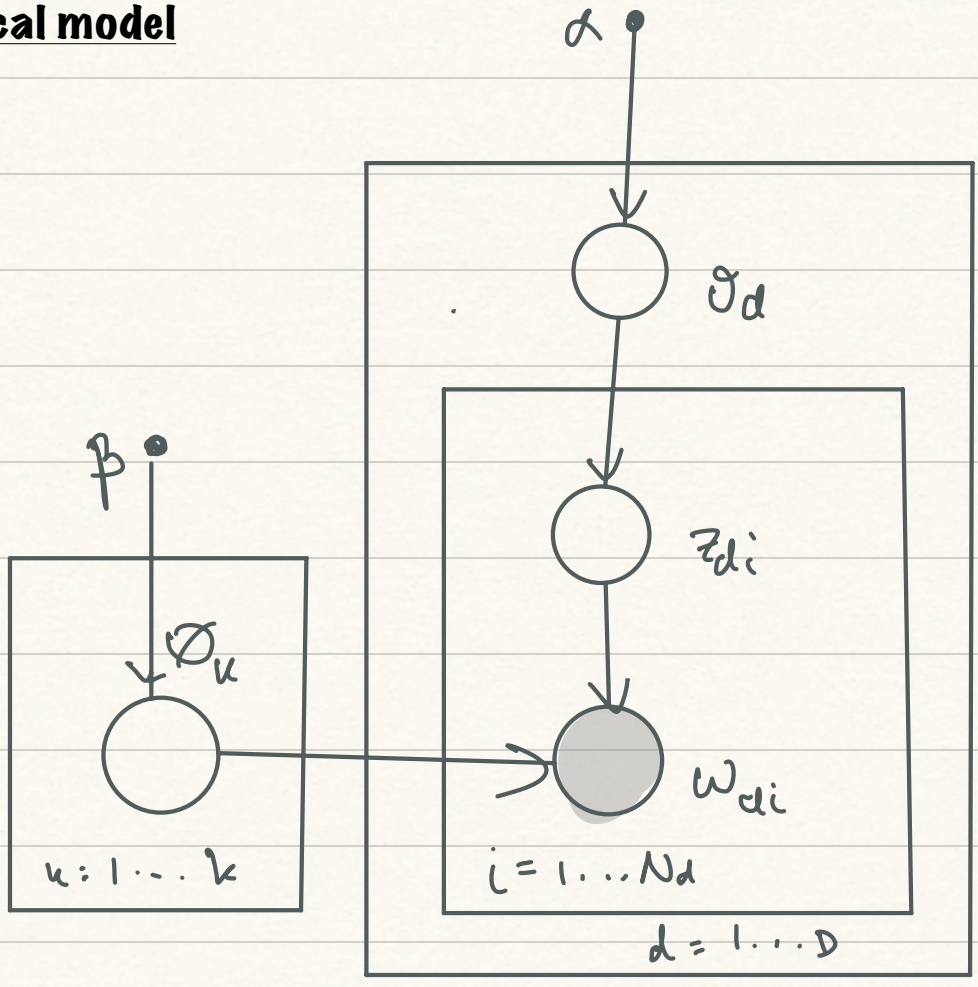
Each document is modeled by a mixture model, with mixture weights theta_d, but the whole **corpus** of documents is then said to be modeled by an **admixture model** or **mixed-membership mixture model.**

To complete a Bayesian specification of the model, we place priors over the parameters:

$$\vartheta_d \overset{iid}{\sim} \text{Dir}(\alpha_1 \dots \alpha_K), \quad d = 1 \dots D$$

$$\varphi_k \overset{iid}{\sim} \text{Dir}(\beta_1 \dots \beta_V), \quad k = 1 \dots K$$

The graphical model



Applications of LDA

- **Topic modeling:** the data is a corpus of text documents. The goal is to infer a description of the **thematic structure** of the corpus via the topics.
- **Population genetics:** each "document" is an individual. At each genetic locus, they can have 0, 1 or 2 mutations. We assume that an individual's genome can be described as an admixture over K **ancestral populations ("topics").**
- **Inference of cell types:** each cell has been profiled and is associated with a collection of "reads", where each read corresponds to a genetic locus that has been expressed. We assume that cells can be described as a mixture over cell archetypes, each of which is a distribution over genes.
- **Radioactive material detection:** each document is a censor, which is a collection of "reads" of different levels of rays. We assume each censor corresponds to a mixture over topics of rays, each of which corresponds to a different kind of material.

Complete conditionals, sufficient stats, "bag of words"

$$p(z_{di} = k \mid w_{di} = v, -) \propto p(z_{di} = k \mid \vartheta_d)\, p(w_{di} = v \mid z_{di} = k, \Phi)$$

$$\propto \vartheta_{dk} \phi_{kv}$$

$$= \frac{\vartheta_{dk} \phi_{kv}}{\sum_j \vartheta_{dj} \phi_{jv}}$$

$$p(\vartheta_d \mid -) \propto p(\vartheta_d \mid \alpha) \prod_i \prod_k p(z_{di} = k \mid \vartheta_d)^{1(z_{di}=k)}$$

$$\propto \mathrm{Dir}(\vartheta_d ; \alpha) \prod_i \prod_k \vartheta_{dk}^{1(z_{di}=k)}$$

$$\propto \prod_k \vartheta_{dk}^{\alpha_k - 1} \cdot \prod_k \vartheta_{dk}^{\sum_i 1(z_{di}=k)}$$

$$= \mathrm{Dir}(\vartheta_d ; \alpha_1 + y_{d1}, \cdots, \alpha_K + y_{dK})$$

where $y_{dj} = \sum_i 1(z_{di} = j)$

$$p(\phi_k \mid -) \propto \mathrm{Dir}(\phi_k; \beta) \prod_d \prod_i \prod_v \phi_{kv}^{1(z_{di}=k)\,1(w_{di}=v)}$$

$$\propto \prod_v \phi_{kv}^{\beta_v - 1} \cdot \prod_v \phi_{kv}^{\sum_d \sum_i \cdots}$$

$$= \mathrm{Dir}(\phi_k; \beta_1 + y_{k1}, \ldots, \beta_V + y_{kV})$$

where $y_{kv} = \sum_d \sum_i 1(z_{di}=k)\,1(w_{di}=v)$

Notice that the sufficient statistics for the Dirichlet vectors are just the counts of indicators Z. These counts form two matrices:

$$(y_{kv})_{k,v} \in \mathbb{N}^{K \times V} \qquad (y_{dk})_{d,k} \in \mathbb{N}^{D \times K}$$

Consider how we might sample the Z's in a Gibbs sampler, and then compute these statistics:

```
for d = 1 ... D:
    for v = 1 ... V:
        for i = 1 ... N_d:
            if w_di = v:
                z_di ~ Cat(p_dv)
```

where $p_{dv} = (p_{dv1} \cdots p_{dvK})$, $p_{dvk} = \frac{\theta_{dk}\phi_{kv}}{\sum_j \theta_{dj}\phi_{jv}}$

The inner loop over the vocabulary v = 1...V is just a way to show that all indicators z_di for tokens w_di=v are iid from the same complete conditional, described by the probability vector p_dv.

The number of tokens in document d that take value v are:

$$y_{dv} = \sum_{i=1}^{N_d} 1(w_{di}=v)$$

Therefore, equivalently, we can sample the sums of indicators from a multinomial.

$$\text{for } d = 1 \dots D$$

$$\text{for } v = 1 \dots V$$

$$(y_{dv1} \cdots y_{dvK}) \sim \text{Multinomial}(y_{dv}, p_{dv})$$

$$\text{where } y_{dvk} = \sum_i 1(w_{di} = v)\, 1(z_{di} = k)$$

"The number of tokens in document d of word type v assigned to topic k"

The sufficient statistics for the parameters are then just contractions of this 3-dimensional array:

$$y_{dk} = \sum_v y_{dvk} \qquad y_{kv} = \sum_d y_{kv}$$

What have we learned? To fit LDA to a corpus of word tokens,
we only need the document-word counts:

$$p(\theta, \Phi, z \mid (w_d)_d) = p(\theta, \Phi, z \mid (y_{dv})_{d,v})$$

In other words, the generative process encodes the "**bag of words**" assumption. The order or the unique identity of the different word tokens does not matter. All that matters is how many tokens of each word type v are in each document d: y_dv.

This is a **very** bad assumption about how text is actually generated. Nevertheless, when applied to documents, LDA learns interpretable latent structure that can be useful at a high level.

CAVI for LDA

We will make the simplifying "**mean-field assumption**" that the q-distribution is a **factorized family**. This will facilitate the optimization problem:

$$q(\Theta, \Phi, z) \approx p(\Theta, \Phi, z \mid w)$$

We will make the simplifying "**mean-field assumption**" that the q-distribution is a **factorized family**. This will facilitate the optimization problem:

$$q(\Theta, \Phi, z) = \prod_d q(\theta_d) \prod_k q(\phi_k) \prod_d \prod_i q(z_{di})$$

Recall from last time the ELBO objective function we will seek to maximize:

$$\underset{q}{\operatorname{argmin}}\ KL(q(\cdots) \,\|\, p(\cdots \mid w)) = \underset{q}{\operatorname{argmax}}\ \underbrace{\mathbb{E}_q\left[\log \frac{p(\cdots)}{q(\cdots)}\right]}_{= ELBO(q)}$$

Since q factorizes, it will be convenient to maximize the ELBO coordinate-wise (i.e., updating each factor, holding all other factors fixed).

$$\text{e.g.}\quad q^*(\theta_d) = \underset{q(\theta_d)}{\operatorname{argmax}}\ ELBO(q)$$

For any factor, the optimal update takes the following form, using theta_d as an example:

$$q^*(\theta_d) \propto \exp\left(\mathbb{E}_{q_{\setminus \theta_d}}[\log p(\theta_d \mid -)]\right)$$

where the expectation is with respect to all factors in q **except** the given one, and the term inside is the complete conditional of the given variable under the true model.

This general form for the optimal update is given by Bishop (2006). Let's confirm that it is true. Consider the KL from any **other** setting of the density q(theta_d) to the (supposedly) optimal one:

$$\begin{aligned} KL(q(\theta_d) \,\|\, q^*(\theta_d)) &= \mathbb{E}_{q(\theta_d)}[\log q(\theta_d) - \log q^*(\theta_d)] \\ &\propto \mathbb{E}_{q(\theta_d)}[\log q(\theta_d)] - \mathbb{E}_{q(\theta_d)}\left[\mathbb{E}_{q_{\setminus \theta_d}}[\log p(\theta_d \mid -)]\right] \\ &\propto -H(q(\theta_d)) - \mathbb{E}_q[\log p(\Theta, \Phi, z \mid w)] \\ &\propto -ELBO(q) \end{aligned}$$

So indeed to maximize the ELBO wrt q(theta_d) you minimize the KL to the stated optimal update.

As we've seen before with EM, the optimal update takes the form of a geometric mean. This means that if the complete conditional is an exponential family, the optimal update to the factor is also a member of that same exponential family:

$$\text{if} \quad p(\vartheta_d \mid -) = f(\vartheta_d ; \eta_d(\cdots)) \propto h(\vartheta_d) \exp\left(t(\vartheta_d)^\top \eta_d(\cdots)\right)$$

where the complete conditional's natural parameter is potentially a function of data and other latent variables.

$$\text{then} \quad q^*(\vartheta_d) = f\left(\vartheta_d ; \mathbb{E}_q[\eta_d(\cdots)]\right)$$

where the natural parameter of this factor is the expectation under the variational distribution (not including this factor) of the complete conditional's natural parameter.

To be clear, this says two things: 1) that the optimal factor is in the same exponential family as the complete conditional, and 2) that its natural parameter takes the form above. To be explicit about this we will introduce notation for the **variational parameters** (which until now have been implicit):

$$\text{①} \quad q^*(\vartheta_d) \equiv q(\vartheta_d ; \tilde{\eta}_d) \equiv f(\vartheta_d ; \tilde{\eta}_d)$$

$$\text{②} \quad \tilde{\eta}_d = \mathbb{E}_q[\eta_d(\cdots)]$$

we are using a tilde to distinguish the **variational parameter** from the parameter of the complete conditional in the true model.

In the case, the complete conditional is Dirichlet. The natural parameter of the Dirichlet is just its shape parameters (the usual parameters), so:

$$\begin{aligned} q^*(\vartheta_d) &= \text{Dir}(\vartheta_d ; \alpha_1 + \mathbb{E}_q[y_{d1}], \cdots, \alpha_K + \mathbb{E}_q[y_{dK}]) \\ &\equiv \text{Dir}(\vartheta_d ; \tilde{\alpha}_{d1}, \cdots \tilde{\alpha}_{dK}) \end{aligned}$$

Let's take a look at the expectations:

$$\mathbb{E}_q[y_{dk}] \equiv \mathbb{E}_{q_{\setminus \vartheta_d}}\left[\sum_i 1(z_{id} = k)\right]$$

$$= \sum_{i=1}^{N_d} q(z_{id} = k)$$

So we see that this depends on the factors q(z_id).

$$q^*(z_{id}=k) \propto \exp\left(\mathbb{E}_q\left[\log p(z_{id}=k \mid w_{id}=v, -)\right]\right)$$

$$\propto \exp\left(\mathbb{E}_q\left[\log \frac{\vartheta_{dk}\varphi_{kv}}{\sum_j \vartheta_{dj}\varphi_{jv}}\right]\right)$$

The normalizer of the probability does not depend on k (the assumed value of z_id:

$$\propto \exp\left(\mathbb{E}_q\left[\log \vartheta_{dk}\varphi_{kv}\right]\right)$$

$$\propto \exp\left(\mathbb{E}_{q(\vartheta_d)}\left[\log \vartheta_{dk}\right] + \mathbb{E}_{q(\varphi_{kv})}\left[\varphi_{kv}\right]\right)$$

$$\propto G_q[\vartheta_{dk}] \cdot G_q[\varphi_{kv}]$$

Here we are introducing new notation for the geometric expected value.

where $G_q[x] \triangleq \exp\left(\mathbb{E}_{x\sim q}[\log x]\right)$

Putting it altogether:

$$q^*(z_{id}=k) \equiv \tilde{p}_{dvk} = \frac{G_q[\vartheta_{dk}]\,G_q[\varphi_{kv}]}{\sum_j G_q[\vartheta_{dj}]\,G_q[\varphi_{jv}]}$$

Do we know these geometric expected values? In this case, the relevant q-distribution is a Dirichlet (as we showed in the last derivation). Both the arithmetic and geometric expected values under a Dirichlet are known in closed form:

$$(X_1 \cdots X_k) \sim \mathrm{Dir}(\alpha_1 \cdots \alpha_k)$$

$$\mathbb{E}[X_k] = \frac{\alpha_k}{\sum_j \alpha_j}, \qquad G[X_k] = \frac{\exp(\psi(\alpha_k))}{\sum_j \exp(\psi(\alpha_j))}$$

where $\psi(\cdot)$ is digamma function.

CAVI for LDA:

While ELBO not converged:

$q(\theta_d)$ ||| $\tilde{\alpha}_{dk} = \alpha_k + \mathbb{E}_q[Y_{dk}] \quad \forall\, d,k$

$C_q[\theta_{dk}] = \exp(\psi(\tilde{\alpha}_{dk}) - \sum_j \psi(\alpha_{dj})) \quad \forall\, d,k$

$q(\beta_k)$ ||| $\tilde{\beta}_{kv} = \beta_v + \mathbb{E}_q[Y_{kv}] \quad \forall\, k,v$

$C_q[\beta_k] = \exp(\psi(\tilde{\beta}_{kv}) - \sum_j \psi(\tilde{\beta}_{jv})) \quad \forall\, k,v$

$q(z_{id})$ ||| $\tilde{p}_{dvk} \propto C_q[\theta_{dk}] C_q[\beta_{kv}] \quad \forall\, d,v,k$

$\mathbb{E}_q[Y_{dk}] = \sum_v Y_{dv} \tilde{p}_{dvk}$

$\mathbb{E}_q[Y_{kv}] = \sum_d Y_{dv} \tilde{p}_{dvk}$

compute ELBO(q)

Computing the ELBO:

$$\mathbb{E}_q\left[\log \frac{p(\theta,\beta,z,w)}{q(\theta,\beta,z)}\right]$$

$$= \mathbb{E}_q[\log p(w \mid -)] + \mathbb{E}_q\left[\log \frac{p(\theta,\beta,z)}{q(\theta,\beta,z)}\right]$$

$$= \mathbb{E}_q[\log p(w \mid -)] + \mathbb{E}_q\left[\log \frac{p(\theta)}{q(\theta)} \cdot \frac{p(\beta)}{q(\beta)} \cdot \frac{p(z \mid \theta)}{q(z)}\right]$$

$$\mathbb{E}_q\left[\log \frac{p(\theta)}{q(\theta)}\right] = \sum_d \mathbb{E}_q\left[\log \frac{p(\theta_d)}{q(\theta_d)}\right]$$

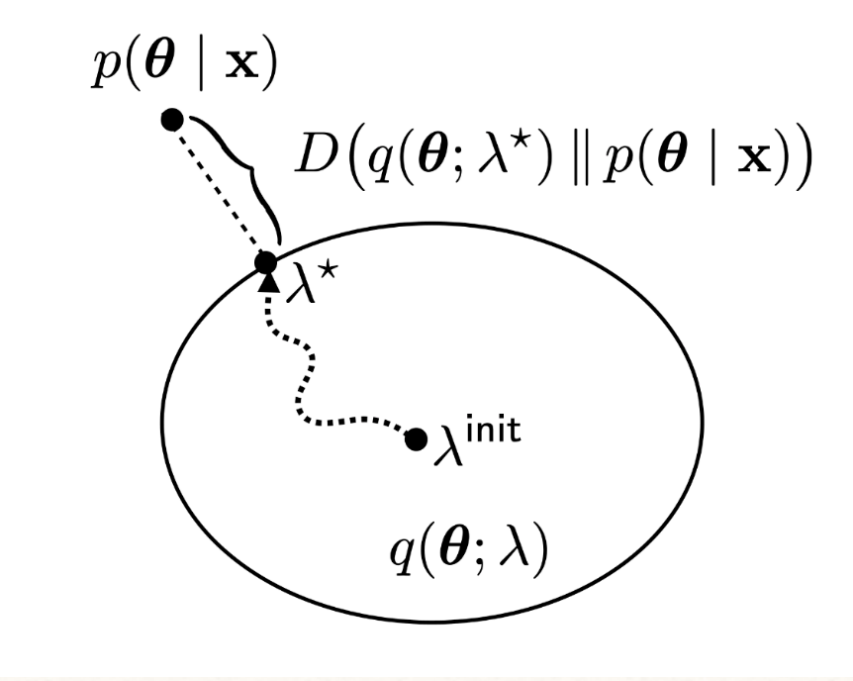
$$= -\sum_d KL(q(\theta_d) \,||\, p(\theta_d))$$

$$= -\sum_d KL(Dir(\theta_d; \tilde{\alpha}_d) \,||\, Dir(\theta_d; \alpha))$$

The KL between two Dirichlets is known in closed form. The other terms can be simplified in similar ways and will end up involving simple expectations under the variational distribution.

What is CAVI doing?

Popping up a level, CAVI is finding the member of a simple parametric family that is closest in KL to the exact posterior. It does so by maximizing the ELBO coordinate-wise. Here's the picture:



In this case, theta represents all latent variables parameters, x is the data, and lambda are the variational parameters. Lambda* represents the value of the variational parameters that brings q(...) closest in KL to the exact posterior. Since q is a factorized family though, there may always be a gap.

Stochastic variational inference (SVI)

For very large data sets, even CAVI can be too slow to run. In this setting, when the model has exponential family complete conditionals we can turn a CAVI algorithm into a scalable **stochastic variational inference (SVI)** algorithm.

To explain this, we will move to a much more general description of a model. **The notation will be different**—e.g., Z in this section will not be the same as Z in the previous sections.

Here's a very general description that many models can be written as:

$$p(x, z, \eta \mid \alpha) = p(\eta \mid \alpha) \prod_{i=1}^{n} p(z_i, x_i \mid \eta)$$

data : $x_1 \cdots x_n$

local latent variables : $z_1 \cdots z_n$

global latent variables : η
aka "parameters"

hyperparameters : α

The **complete data likelihood** for data point i is:

$$p(z_i, x_i \mid \eta) = h_\ell(z_i, x_i) \exp\left(\eta^T t_\ell(z_i, x_i) - a_\ell(\eta)\right)$$

The conjugate prior is:

$$p(\eta \mid \alpha) = f(\eta; \alpha) \; \alpha_\eta \; h_c(\eta) \exp\left(\alpha^T t_c(\eta)\right)$$

$$t_c(\eta) = [\eta, -a_\ell(\eta)]^T$$

The complete conditional is then:

$$p(\eta \mid -) = f(\eta, \alpha_n) \; \alpha_\eta \; h_c(\eta) \exp\left(\alpha_n^T t_c(\eta)\right)$$

where the natural parameter of the complete conditional of the parameter is:

$$\alpha_n = \left[\alpha_1 + \sum_{i=1}^{n} t_\ell(z_i, x_i), \quad \alpha_2 + n\right]^T$$

We will setup a variational distribution that factorizes:

$$q(z, \eta) = q(\eta; \lambda) \prod_{i=1}^{n} q(z_i; \gamma_i)$$

where lambda and gamma_i are the variational parameters.

As we saw in the last section, the optimal CAVI update for the factor over eta would be

$$q^*(\eta; \lambda) \propto_\eta \exp\left(\mathbb{E}_{q_{\setminus \eta}}[\log p(\eta \mid -)]\right)$$

$$\propto_\eta h_c(\eta) \exp\left(\mathbb{E}_q[\alpha_n]^T t_c(\eta)\right)$$

$$= f(\eta, \underbrace{\mathbb{E}_q[\alpha_n]}_{= \lambda})$$

Although we derived this in another way, we could think about this in terms of **gradients of the ELBO** (or KL divergence). Consider taking the following gradient with respect to the variational parameter lambda:

$$\nabla_\lambda \text{ELBO} = \nabla_\lambda \mathbb{E}_q\left[\log \frac{p(x, z, \eta \mid \alpha)}{q(\eta, z \mid \lambda, \gamma)}\right]$$

Terms not involving lambda or eta will drop. We can also add a constant into the numerator to obtain:

$$= \nabla_\lambda \mathbb{E}_{q(\eta, z \mid \lambda, \gamma)}\left[\log \frac{p(\eta \mid \alpha)}{q(\eta \mid \lambda)}\right]$$

Now assume that q(...) is the same exponential family as the complete conditional

$$= \nabla_\lambda \mathbb{E}_q\left[\log \frac{f(\eta; \alpha_n)}{p(\eta; \lambda)}\right]$$

$$= \nabla_\lambda \left(\mathbb{E}_q\left[(\alpha_n - \lambda)^T t_c(\eta)\right] - \left[a_c(\alpha_n) - a_c(\lambda)\right]\right)$$

The expectation is over both global and local latents, and the natural parameter of the complete conditional is a function of local latents:

$$= \nabla_\lambda \left(\mathbb{E}_q[\alpha_n] - \lambda\right)^T \underbrace{\mathbb{E}_q[t_c(\eta)]}_{\equiv \nabla_\lambda a_c(\lambda)} - \nabla_\lambda a_c(\lambda)$$

The expectation of the sufficient statistics is the **mean parameter** of any expfam, equal to the gradient of the log normalizer.

$$= \underbrace{\nabla_\lambda^2 a_c(\lambda)}_{\text{Hessian}} \left(\mathbb{E}_q[\alpha_n] - \lambda\right) + \cancel{\nabla_\lambda a_c(\lambda)} - \cancel{\nabla_\lambda a_c(\lambda)}$$

$$\hookrightarrow \left[\nabla_\lambda^2 a(\lambda)\right]_{ij} = \frac{\partial}{\partial \lambda_i} \frac{\partial}{\partial \lambda_j} a(\lambda)$$

The term out front is then the Hessian of the log-normalizer (matrix of second-order derivatives). It's immediately clear from the gradient that the optimal setting of the variational parameter is:

$$\lambda^* = \mathbb{E}_q[\alpha_n]$$

which we already knew.

Unpacking this expectation, we see that it involves a sum over n data points:

$$\mathbb{E}_q[\alpha_n] = \mathbb{E}_q\left[\begin{array}{l} \alpha_1 + \sum_{i=1}^n t_1(x_i, z_i) \\ \alpha_2 + n \end{array}\right]$$

$$= \left[\begin{array}{l} \alpha_1 + \sum_{i=1}^n \mathbb{E}_{q(z_i;\gamma_i)}[t(x_i, z_i)] \\ \alpha_2 + n \end{array}\right]$$

If there are many data points, it may be prohibitive to update all the local factors q(zi).

Instead, we will derive an algorithm that sub-samples our data. To do so, we will re-cast variational inference in terms of **stochastic optimization**.

Consider the following **stochastic gradient ascent** algorithm

for iteration t:

$$\lambda_t = \lambda_{t-1} + \epsilon_t \nabla_\lambda \widehat{ELBO}_t$$

where epsilon_t is the **step-size** at iteration t and we are following the gradient of an **estimate** of the ELBO, which might be different at each iteration.

If our estimator of the ELBO is **unbiased**—i.e.,

$$\mathbb{E}[\nabla_\lambda \widehat{ELBO}_t] = \mathbb{E}[\nabla_\lambda ELBO]$$

and if the sequences of step-sizes follows the **Robins-Monroe conditions**—i.e.m

$$\sum_t \epsilon_t = \infty \quad \text{and} \quad \sum_t \epsilon_t^2 < \infty, \quad \text{e.g. } \epsilon_t = t^{-k}, \; k \in (1/2, 1]$$

then the parameter will eventually converge to a **local mode** of the ELBO:

$$\lim_{t \to \infty} \lambda_t \in \text{local modes of ELBO}$$

Why does this help us? Because unbiased estimators of gradients can often be derived by **sub-sampling** the data. Consider a gradient of the following form:

$$\nabla_\lambda \text{objective} = \sum_{i=1}^{n} d_i(x_i, z_i, \lambda)$$

Then we could sub-sample the data points uniformly at random to construct the following unbiased estimator:

$$i \sim \text{uniform}(1 \ldots n)$$

$$\widehat{\nabla_\lambda \text{objective}} = n\, d_i(x_i, z_i, \lambda)$$

Unfortunately the **Euclidean gradient** of the ELBO does not have this form...

...however the **pre-conditioned Euclidean gradient** does:

$$[\nabla_\lambda^2 a_c(\lambda)]^{-1} \nabla_\lambda ELBO = \lambda - \mathbb{E}_q[\alpha_n]$$

- $[\nabla_\lambda^2 a_c(\lambda)]^{-1}$: pre-conditioning matrix
- $\nabla_\lambda ELBO$: Euclidean gradient
- $\lambda - \mathbb{E}_q[\alpha_n]$: natural gradient (as we will show)

It turns out that pre-conditioning the Euclidean gradient in this particular way (with the inverse of the Hessian) gives us the **natural gradient** when the complete conditionals are all exponential families.

What is a **natural gradient**? Amari (1998) introduced the idea of natural gradient ascent for fitting probabilistic models. The natural gradient is defined by the Euclidean gradient pre-conditioned by the **inverse Fisher information** matrix, which in our setting would be:

$$g(\lambda) \triangleq I(\lambda)^{-1} \nabla_\lambda \text{ELBO}$$

where the Fisher information matrix is defined as:

$$I(\lambda) = -\nabla_\lambda^2 \mathop{\mathbb{E}}_{\eta|\lambda}[\log q(\eta;\lambda)]$$

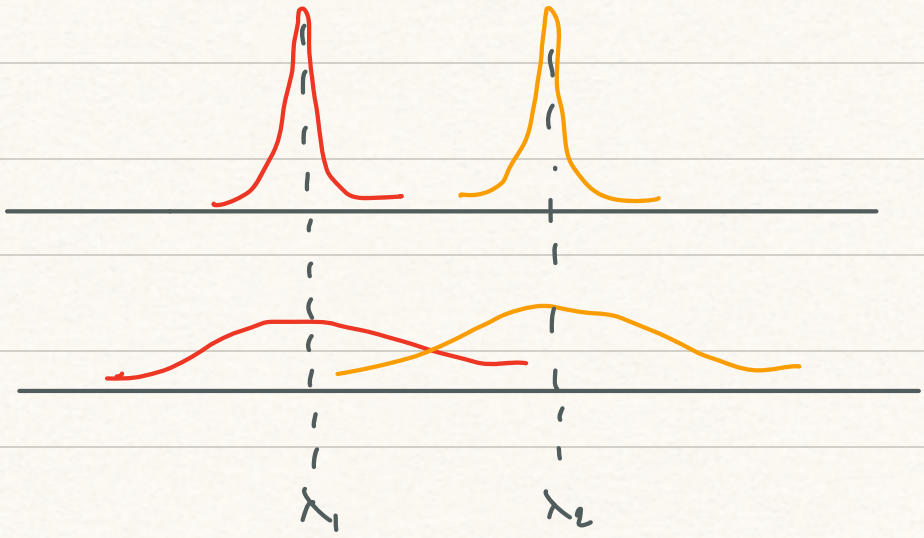
For exponential families, the Fisher information is:

$$= -\nabla_\lambda^2 \left(\mathbb{E}[t(\eta)]^\top \lambda - a(\lambda) \right)$$

$$= \nabla_\lambda^2 a(\lambda)$$

So the Hessian of the log-normalizer is the Fisher information for exponential families.

The natural gradient is a **local re-scaling** of the Euclidean gradient to account for the fact that distances between probability distributions is not equal to the Euclidean distance between their parameters. Here's a simple example:



In both plots, the Euclidean distance between the parameters of the orange and red distributions is the same. However the KL-divergence between the two distributions is much higher in the top than in the bottom.

Consider then the following **natural gradient step:**

$$\lambda_t = \lambda_{t-1} + \zeta_t \, g(\lambda_{t-1})$$

In our settings, the natural gradient equals:

$$= \lambda_{t-1} + \zeta_t \left(\mathbb{E}_q[\alpha_n] - \lambda_{t-1} \right)$$

Re-arranging terms:

$$= (1 - \zeta_t)\lambda_{t-1} + \zeta_t \mathbb{E}_q[\alpha_n]$$

Recall that the expectation of the posterior natural parameter equals

$$\mathbb{E}_q[\alpha_n] = \mathbb{E}_q\left[\begin{bmatrix} \alpha_1 + \sum_{i=1}^{n} t(z_i, x_i) \\ \alpha_2 + n \end{bmatrix}\right] = \begin{bmatrix} \alpha_1 + \sum_i \mathbb{E}_q[t(z_i, x_i)] \\ \alpha_2 + n \end{bmatrix}$$

So we can replace this expectation with an unbiased estimator of the expectation:

$$\hat{\mathbb{E}}_q[\alpha_n] = \begin{bmatrix} \alpha_1 + n\,\mathbb{E}_q[t(z_i, x_i)] \\ \alpha_2 + n \end{bmatrix}, \quad i \sim \text{Uniform}(1 \dots n)$$

Putting it altogether, here is the **stochastic VI** algorithm, which performs stochastic natural gradient ascent via data sub-sampling:

for iteration t:

subsample a data point $\quad i \sim \text{Uniform}(1 \dots n)$

update the factor for the local latent variable $\quad q^*(z_i) = \dots$

take a natural gradient step in the global latent variable

$$\lambda_t = (1 - \zeta_t)\lambda_{t-1} + \zeta_t \begin{bmatrix} \alpha_1 + n\,\mathbb{E}_q[t(z_i, x_i)] \\ \alpha_2 + n \end{bmatrix}$$