CSC 412/2506:
Probabilistic Learning and Reasoning

Week 6: HMMs and Variational Inference I

Ohad Shamir

University of Toronto

Overview: First hour

- Hidden Markov Models
- Forward / Backward Algorithm
- Viterbi Algorithm

Announcements:

- Assignment 2 due next Monday; TA office hours on website
- Will post assignment 1 solution on quercus
- Mid-term exam after reading week. Will cover first 6 weeks of course; practice exam+solutions will be published

Sequential data

Recall last week: Data being i.i.d. is sometimes a poor assumption:

- Sequential data is common in time-series modelling (e.g. stock prices, speech, video analysis) or ordered (e.g. textual data, gene sequences).
- Recall the general joint factorization via the chain rule

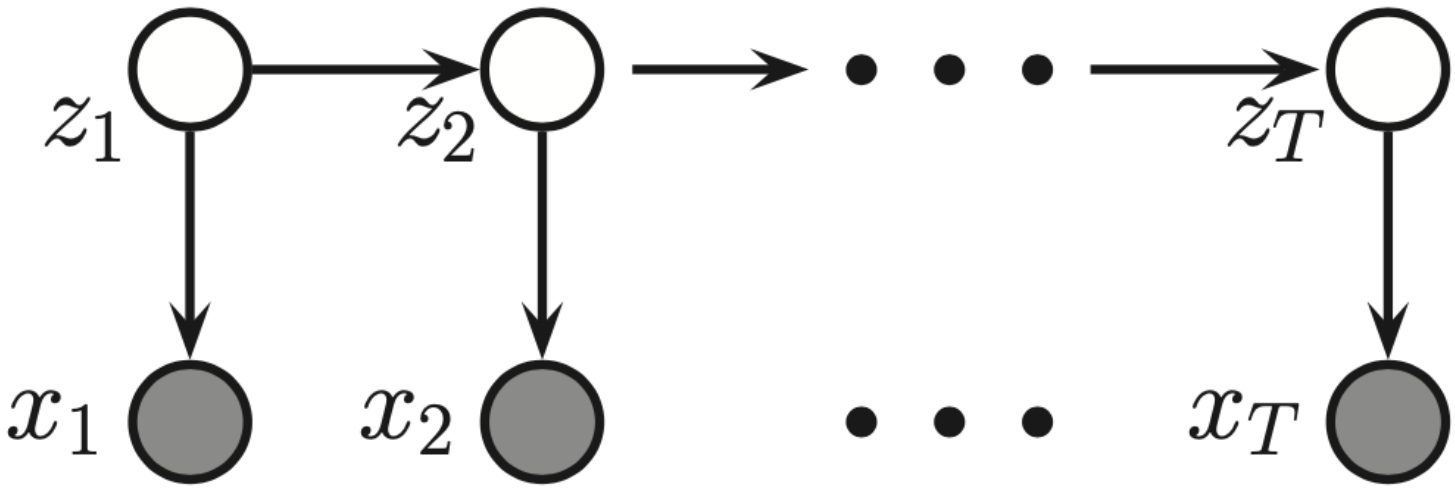
$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t-1}, ..., x_1) \quad \text{where } p(x_1|x_0) = p(x_1).$$

- This quickly becomes intractable for high-dimensional data
- So we **made** the simplifying assumption that our data can be modeled as a **first-order Markov chain**

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$$

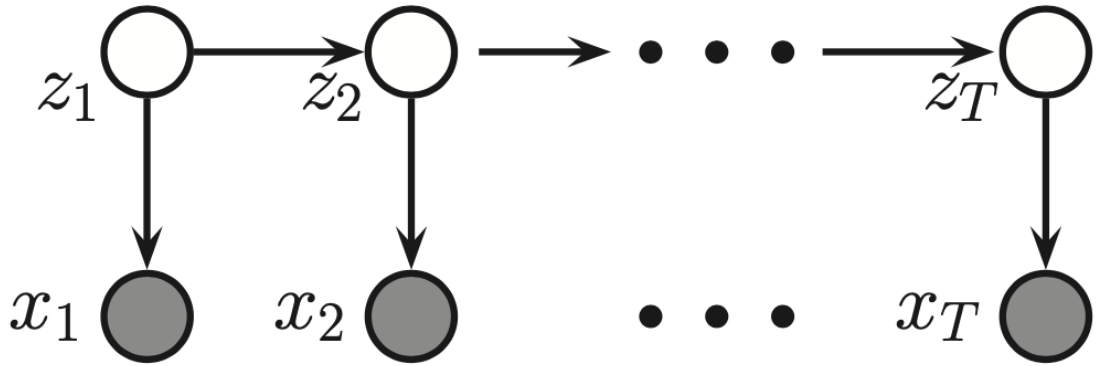
Sequential Data

- In certain cases, Markov chain assumption is also restrictive, e.g. when we have latent variables which are not observed.
- Motivates *Hidden Markov Models*



Hidden Markov Models (HMMs)

- HMMs hide the temporal dependence by keeping it in the unobserved state.
- For each observation x_t, we associate a corresponding unobserved hidden/latent variable z_t



- The joint distribution of the model becomes

$$p(x_{1:T}, z_{1:T}) = p(z_1) \prod_{t=2}^{T} p(z_t|z_{t-1}) \prod_{t=1}^{T} p(x_t|z_t)$$

Hidden Markov Models (HMMs)

Assuming we have a homogeneous model, we only have to know three sets of distributions

1. **Initial distribution**: $\pi(i) = p(z_1 = i)$. The probability of the first hidden variable being in state i
2. **Transition distribution**:
   $\Psi(i, j) = p(z_{t+1} = j | z_t = i) \quad i \in \{1, ..., k\}$. The probability of moving from hidden state i to hidden state j.
3. **Emission probability**: $\psi_t(i) = p(x_t | z_t = i)$. The probability of an observed random variable x given the state of the hidden variable that "emitted" it.

HMMs: Objectives

We consider the following objectives:

1. Compute the probability of a latent variable given an observation sequence.
   That is, we want to be able to compute $p(z_t|x_{1:T})$. This is achieved with the **Forward-Backward algorithm**.
2. Infer the most likely sequence of hidden states (MAP inference).
   That is, we want to be able to compute

$$z^\star = \underset{z_{1:T}}{\text{argmax}}\ p(z_{1:T}|x_{1:T}).$$

   This is achieved using the **Viterbi algorithm**.

Forward algorithm

- Let us begin with the easier goal of computing $p(z_t|x_{1:t})$ for any t (where $x_{1:t}$ are given)
- Define the probability vector

$$\alpha_t(j) = p(z_t = j|x_{1:t})$$

- We will show that α_t can be recursively computed as follows:

$$\alpha_1 \propto \psi_1 \odot \pi \quad \text{and for all } t > 1, \quad \alpha_t \propto \psi_t \odot (\Psi^T \alpha_{t-1})$$

where $\pi(i) = p(z_1 = i), \psi_t(j) = p(x_t|z_t = j), \Psi(i,j) = p(z_t = j|z_{t-1} = i)$ as before, and $\odot$ is the Hadamard (entrywise) product.

- This will be part of the full forward-backward algorithm later.

Forward algorithm: Derivation

- $t = 1$: By Bayes' rule, for any state j,

$$\alpha_1(j) \;=\; p(z_1 = j \mid x_1) \;=\; \frac{p(x_1 \mid z_1 = j)\, p(z_1 = j)}{p(x_1)} \;\propto\; \psi_1(j)\pi(j).$$

- $t > 1$: Again by Bayes' rule, for any state j,

$$\begin{aligned}
\alpha_t(j) &= p(z_t = j \mid x_{1:t}) \\
&\propto p(x_t \mid z_t = j)\, p(z_t = j \mid x_{1:t-1}) \\
&= \psi_t(j) \sum_i p(z_t = j \mid z_{t-1} = i, x_{1:t-1})\, p(z_{t-1} = i \mid x_{1:t-1}) \\
&= \psi_t(j) \sum_i \underbrace{p(z_t = j \mid z_{t-1} = i)}_{\Psi(i,j)} \underbrace{p(z_{t-1} = i \mid x_{1:t-1})}_{\alpha_{t-1}(i)} \\
&= \psi_t(j) \sum_i \Psi(i,j)\alpha_{t-1}(i).
\end{aligned}$$

Forward-Backward algorithm

- Forward algorithm gives $p(z_t|x_{1:t})$ for all t
- What we often actually want:

$$\gamma_t = p(z_t|x_{1:T}) \quad \forall t$$

  assuming that we know π, Ψ, ψ_t as before.
- Given by

$$\gamma_t \propto \alpha_t \odot \beta_t$$

where

$$\beta_t(i) = p(x_{t+1:T}|z_t = i)$$

given by the *backward recursion*

$$\forall i \ , \beta_T(i) = 1, \quad \text{and} \quad \beta_t = \Psi(\psi_{t+1} \odot \beta_{t+1}) \quad \forall t < T$$

So algorithm overall: Compute all α_t by forward recursion; all β_t by backward recursion; and compute any γ_t.

Forward-Backward algorithm: Derivation

$\gamma_t \propto \alpha_t \odot \beta_t$ follows from

$$\begin{aligned}
\gamma_t = p(z_t|x_{1:T}) &= p(z_t, x_{1:T})/p(x_{1:T}) \\
&\propto p(z_t, x_{1:T}) \\
&= p(z_t, x_{1:t})p(x_{t+1:T}|z_t, x_{1:t}) \\
&= p(z_t, x_{1:t})p(x_{t+1:T}|z_t) \\
&\propto \alpha_t \odot \beta_t
\end{aligned}$$

Backward recursion for β_t:

$$\begin{aligned}
\beta_t(i) &= p(x_{t+1:T}|z_t = i) \\
&= \sum_j p(z_{t+1} = j, x_{t+1:T}|z_t = i) \\
&= \sum_j p(x_{t+2:T}|z_{t+1} = j, z_t = i, x_{t+1})p(x_{t+1}|z_{t+1} = j, z_t = i)p(z_{t+1} = j|z_t = i) \\
&= \sum_j p(x_{t+2:T}|z_{t+1} = j)p(x_{t+1}|z_{t+1} = j)p(z_{t+1} = j|z_t = i) \\
&= \sum_j \beta_{t+1}(j)\psi_{t+1}(j)\Psi(i, j)
\end{aligned}$$

Viterbi algorithm

- The Viterbi algorithm (Viterbi 1967) is used to compute the most probable sequence.

$$\hat{z} = \arg\max_{z_{1:T}} \; p(z_{1:T}|x_{1:T})$$

Viterbi algorithm: Derivation

- Define the probability of the most likely path ending at state j:

$$\delta_t(j) = \max_{z_{1:t-1}} p(z_{1:t-1}, z_t = j, x_{1:t}).$$

- **Recursive step ($t > 1$):** Using the HMM assumptions,

$$\begin{aligned}
\delta_t(j) &= \max_{z_{1:t-1}} p(z_{1:t-1}, z_t = j, x_{1:t}) \\
&= \max_{z_{1:t-1}} p(z_{1:t-1}, x_{1:t-1})\, p(z_t = j \mid z_{t-1})\, p(x_t \mid z_t = j) \\
&= p(x_t \mid z_t = j) \max_i \left[p(z_t = j \mid z_{t-1} = i) \max_{z_{1:t-2}} p(z_{1:t-2}, z_{t-1} = i, x_{1:t-1}) \right] \\
&= p(x_t \mid z_t = j) \max_i \left[\delta_{t-1}(i)\, p(z_t = j \mid z_{t-1} = i) \right] \\
&= \boxed{\psi_t(j) \max_i \left[\delta_{t-1}(i) \Psi(i, j) \right]}.
\end{aligned}$$

Viterbi algorithm: Pseudocode

- **Initialization** ($t = 1$)**:**

$$\delta_1(j) = p(z_1 = j, x_1) = \pi(j)\psi_1(j).$$

- **Forward recursion** ($t = 2, \ldots, T$)**:** For each state j, compute

$$\boxed{\delta_t(j) = \psi_t(j) \max_i \left[\delta_{t-1}(i)\Psi(i,j)\right]}$$

and store the maximizing previous state:

$$\theta_t(j) = \arg\max_i \left[\delta_{t-1}(i)\Psi(i,j)\right].$$

- **Termination:** Find the final state of the most probable path:

$$z_T^* = \arg\max_j \delta_T(j).$$

- **Traceback:** Recover the preceding states, working backward:

$$\boxed{z_t^* = \theta_{t+1}(z_{t+1}^*), \qquad t = T-1, \ldots, 1.}$$

Summary: HMMs

- HMMs is a latent Markov chain with observations.
- Forward-backward algorithm can be used to find probability of latent variables given observations.
- Viterbi algorithm can be used to do MAP.
- Key takeaway: We can use the structure of the HMM to do efficient computations via recursions

- Next (more fun!): Variational inference.

Overview: part 2

- Variational Inference
- M-projection
- I-projection
- Naive mean-field approach

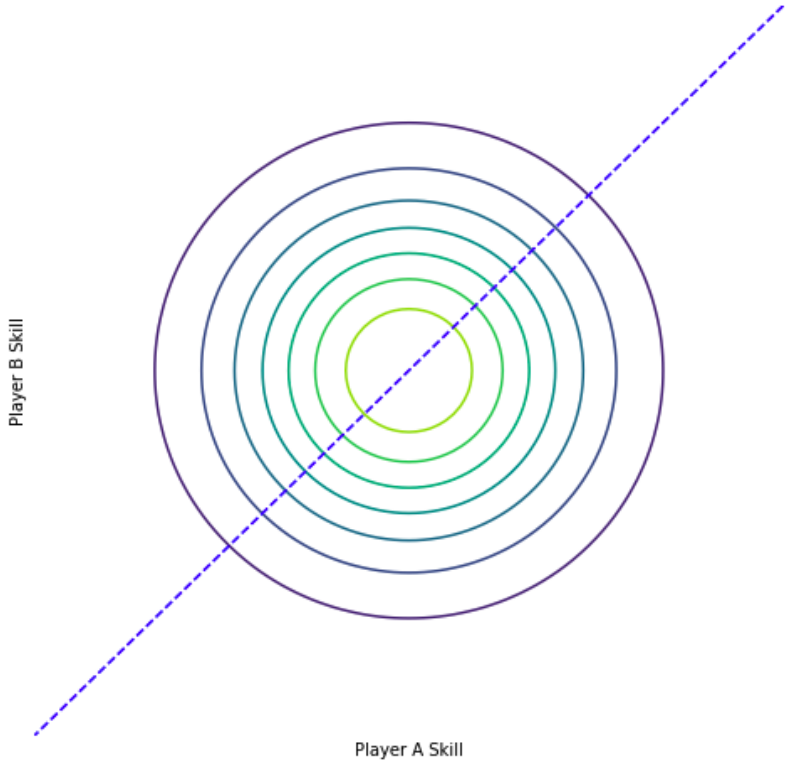
Posterior Inference for Latent Variable Models

We've worked with a few latent variable models, such as the generative image model and the trueskill model.
These models have a factorization $p(x, z) = p(z)p(x|z)$ where

- x are the observations or data,
- z are the unobserved (latent) variables
- $p(z)$ is usually called the **prior**
- $p(x|z)$ is usually called the **likelihood**
- The conditional distribution of the unobserved variables given the observed variables (aka the **posterior**) is

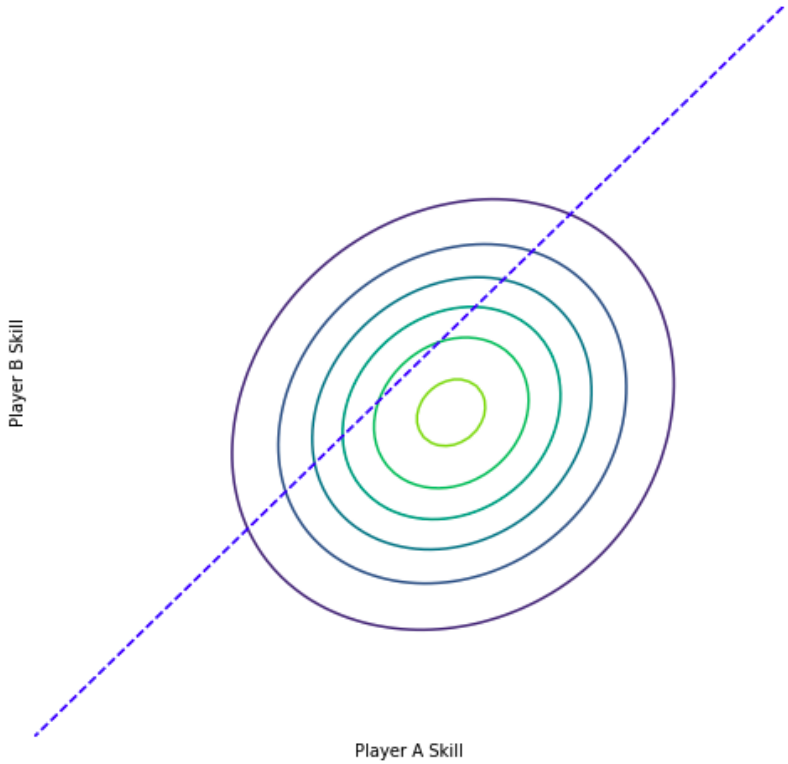
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x, z)dz}$$

Prior:



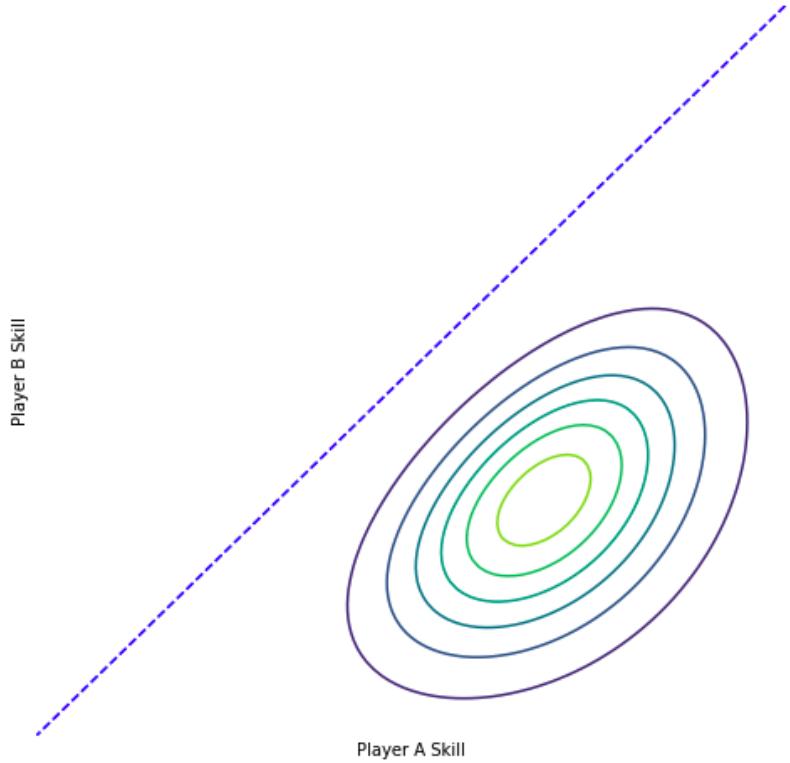
Says we're very uncertain about both player's skill.

Posterior:



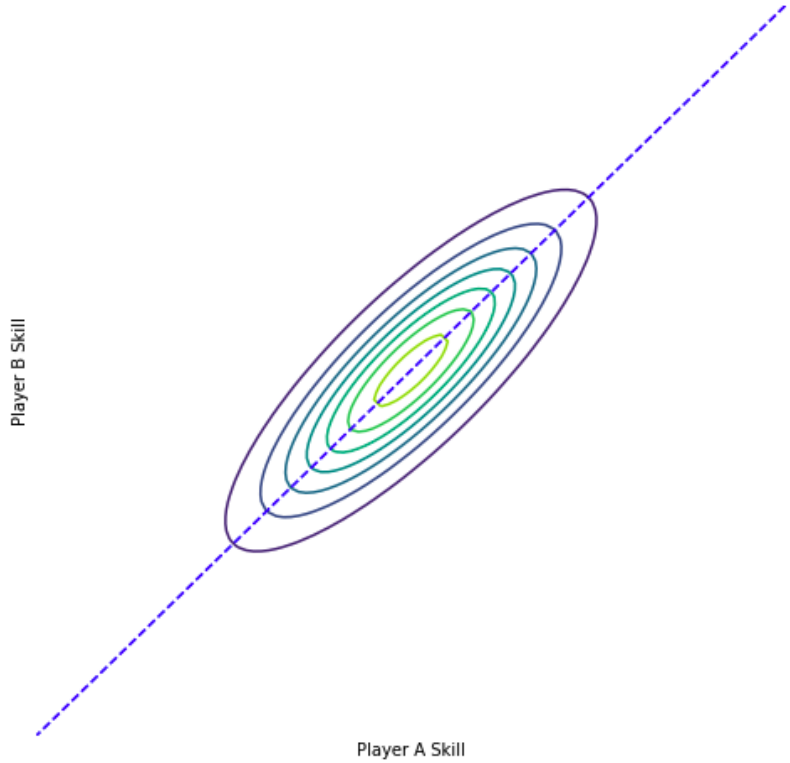
The posterior isn't Gaussian anymore.

Posterior after A beats B 10 times:



Now the posterior is certain that A is better than B.

Posterior after both beat each other 10 times:



Now the posterior is certain that neither player is much better than the other, but is uncertain how good they both are in an absolute sense.

What is hard to compute about the posterior?

- The integral $p(x) = \int p(x, z)dz$ is intractable whenever z is high dimensional. This makes evaluating or sampling from the normalized posterior $p(z|x)$ for a given x and z also intractable.

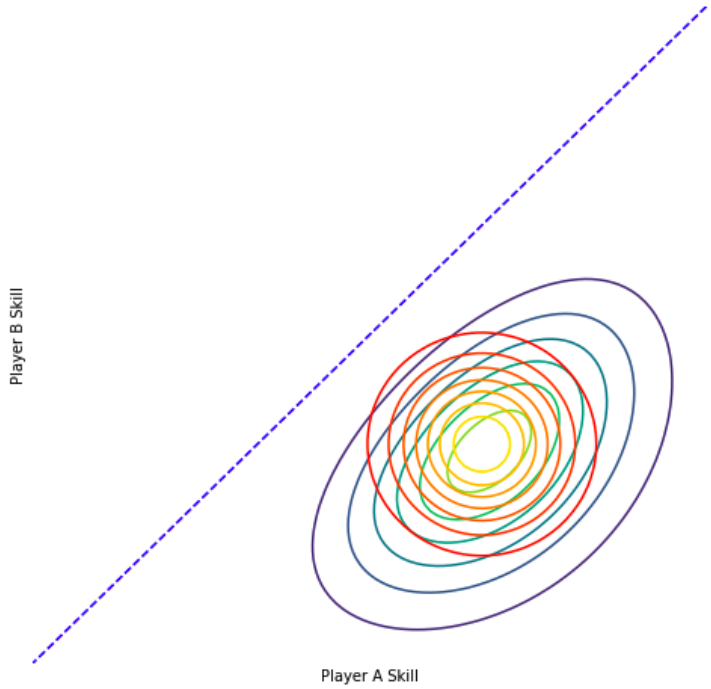
Variational methods

- Variational inference is closely related to the calculus of variations, developed in the 1700s by Euler, Lagrange.
- Variational inference is an approximate inference method where we seek a tractable (e.g., factorized) approximation to the target intractable distribution.

Variational methods

To be more formal, variational inference works as follows:

- Choose a tractable distribution $q(z) \in Q$ from a feasible set Q. This distribution will be used to approximate $p(z|x)$.
  - For example, $q(z) = \mathcal{N}(z|\mu, \Sigma)$. The idea is that we'll try choose a Q that makes $q(z)$ a good approximation of the true posterior $p(z|x)$.
- Encode some notion of "difference" between $p(z|x)$ and q that can be effciently estimated. Usually we will use the KL divergence.
- Minimize this difference. Usually we will use an iterative optimization method.



- Whatever feasible set we choose for Q, it's usually not the case that there is any $q \in Q$ that exactly matches the true posterior.
- But computing the true posterior is intractable, so we have to take a shortcut somewhere.

How to measure closeness: KL divergence

We will measure the difference between q and p using the **Kullback-Leibler divergence**

$$KL(q(z)||p(z|x)) = \int q(z) \log \frac{q(z)}{p(z|x)} dz$$
$$= \mathop{\mathbb{E}}_{z\sim q} \log \frac{q(z)}{p(z|x)}$$

Properties of the KL Divergence

- $KL(q||p) \geq 0$
- $KL(q||p) = 0 \Leftrightarrow q = p$
- $KL(q||p) \neq KL(p||q)$
- KL divergence is not a metric, since it's not symmetric

Which direction of KL to use? $KL(q||p)$ vs $KL(p||q)$

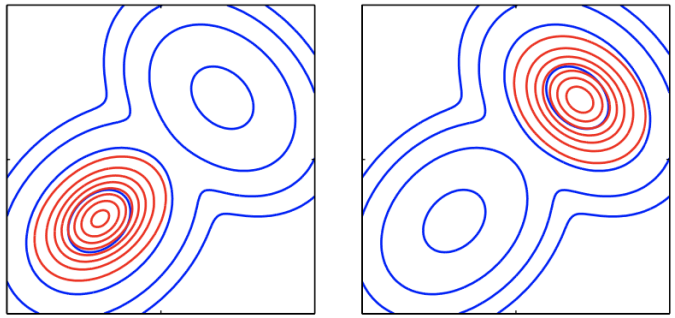
- We could minimize $KL(q||p)$ or $KL(p||q)$
- Which one to choose?
- As always, we will go with the tractable one.

Option 1: Information (I-)Projection:

I-projection: $q^* = \arg\min_{q \in Q} KL(q||p) = \mathbb{E}_{x \sim q(x)} \log \frac{q(x)}{p(x)}$:

- $p \approx q \implies KL(q||p)$ small
- I-projection underestimates support: $KL(q||p)$ penalizes q if q has mass where p doesn't, but not vice versa
- Also, does not yield the correct moments.

$p(x)$ is mixture of **two** 2D Gaussians and Q is the set of all 2D Gaussian distributions (with arbitrary covariance matrices)



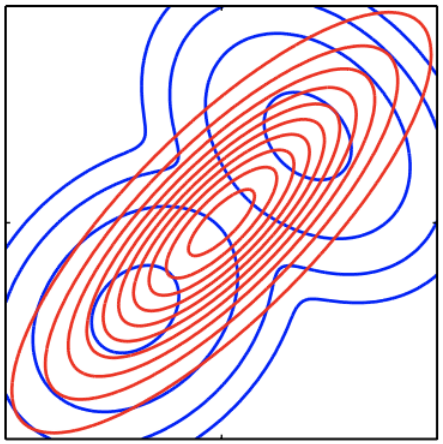
p=Blue, q^*=Red (two equivalently good solutions!)

Option 2: Moment (M-)projection

M-projection: $q^* = \arg\min_{q \in Q} KL(p||q) = \mathbb{E}_{x \sim p(x)} \log \frac{p(x)}{q(x)}$:

- $p \approx q \implies KL(p||q)$ small
- $KL(p||q)$ penalizes q missing mass where p has some.
- M-projection yields a distribution $q(x)$ with the correct mean and covariance.

$p(x)$ is mixture of **two** 2D Gaussians and Q is the set of all 2D Gaussian distributions (with arbitrary covariance matrices)



p=Blue, q^*=Red

Which option to choose?

- Most variational inference algorithms use **I-projection.**
  - Reason: M-projection involves expectation over (intractable) p

- As we will see, I-projection closely related to maximizing **entropy**

$$H(q) = -\mathbb{E}_{x\sim q(x)} \log q(x)$$

measuring the uncertainty in the distribution q.

- **Note:** There is a theorem saying that

$$\begin{aligned} &\text{maximize}_q \; H(q) \\ &\text{subject to } \mathbb{E}_{x\sim q(x)}[f_i(x)] = t_i \quad \text{for } i = 1, .., k \end{aligned}$$

is always maximized by an exponential family.

Mean-field approach

- Say we have an arbitrary MRF:

$$p(x|\theta) = \exp\left\{\sum_{c\in\mathcal{C}} \phi_c(x_c) - \log Z(\theta)\right\}$$

- We find an approximate distribution $q(x) \in Q$ by performing I-projection to $p(x)$.

$$q^* = \arg\min_{q\in Q} KL(q||p) = \mathbb{E}_{x\sim q(x)} \log \frac{q(x)}{p(x|\theta)}$$

$$\arg\min_{q\in Q} KL(q||p) = \mathbb{E}_{x\sim q(x)}\left[\log q(x) - \sum_{c\in\mathcal{C}} \phi_c(x_c) + \log Z(\theta)\right]$$

$$= \arg\max_{q\in Q} \sum_{c\in\mathcal{C}} \mathbb{E}_q[\phi_c(x_c)] + H(q)$$

- For tractability, we need a nice set Q.

Naive Mean-Field

- One way to proceed is the mean-field approach where we assume:

$$q(x) = \prod_{i \in V} q_i(x_i)$$

  the set Q is composed of distributions with this factorization.
- Using this in the maximization problem, we can simplify things

$$q^* = \arg\max_{q \in Q} \sum_{c \in \mathcal{C}} \sum_{x_c} q(x_c)\phi_c(x_c) + H(q)$$

- We notice $q(x_c) = \prod_{i \in c} q_i(x_i)$ and also

$$
\begin{aligned}
H(q) =& \mathbb{E}_q[-\log q(x)] = -\sum_x q(x) \log q(x) \\
&= -\sum_x q(x) \Big[\sum_i \log q_i(x_i) \Big] \\
&= -\sum_i \sum_x \Big[q_i(x_i) \log q_i(x_i) \Big] \frac{q(x)}{q_i(x_i)} \\
&= -\sum_i \sum_{x_i} \Big[q_i(x_i) \log q_i(x_i) \Big] \sum_{x \setminus x_i} \frac{q(x)}{q_i(x_i)} \\
&= -\sum_i \sum_{x_i} \Big[q_i(x_i) \log q_i(x_i) \Big] \\
&= \sum_i H(q_i)
\end{aligned}
$$

Example: Pairwise MRF

- Thus the final optimization problem reduces to

$$q^* = \arg\max_q \sum_{c \in \mathcal{C}} \sum_{x_c} \phi_c(x_c) \prod_{i \in c} q_i(x_i) + \sum_i H(q_i)$$

$$\text{subject to: } q_i(x_i) \geq 0 \text{ and } \sum_{x_i} q_i(x_i) = 1.$$

- Let's further simplify the setting and assume that we have a pairwise MRF. Then the optimization problem becomes

$$q^* = \arg\max_q \sum_{(i,j) \in \mathcal{E}} \sum_{x_i, x_j} \phi_{ij}(x_i, x_j) q_i(x_i) q_j(x_j) - \sum_i \sum_{x_i} q_i(x_i) \log(q_i(x_i))$$

$$\text{subject to: } q_i(x_i) \geq 0 \text{ and } \sum_{x_i} q_i(x_i) = 1.$$

Coordinate maximization

This problem is hard as it has many local maxima! But we can still try to optimize, using (say) block coordinate ascent.

- Initialize $\{q_i(x_i)\}_{i \in V}$ uniformly
- Iterate over $i \in V$
  - Greedily maximize the objective over $q_i(x_i)$
  - This is equivalent to: $q_i(x_i) \propto \exp\left\{ \sum_{j \in N(i)} \sum_{x_j} q_j(x_j) \phi_{ij}(x_i, x_j) \right\}$
    - which follows from: write the Lagrangian, take the derivative, set to zero, and solve
  - Repeat until convergence.

This is guaranteed to converge but can converge to local optima.

Summary

- Approximate the complex (intractable) distribution with a simpler (tractable) distribution
- I-projection & M-projection measure the distance to true posterior
- Mean field approximation is a way to simplify the set of distributions
- More variational inference next week (then reading week + midterm).