From Variational Inference to Variational Auto-Encoder

Shichang Zhang

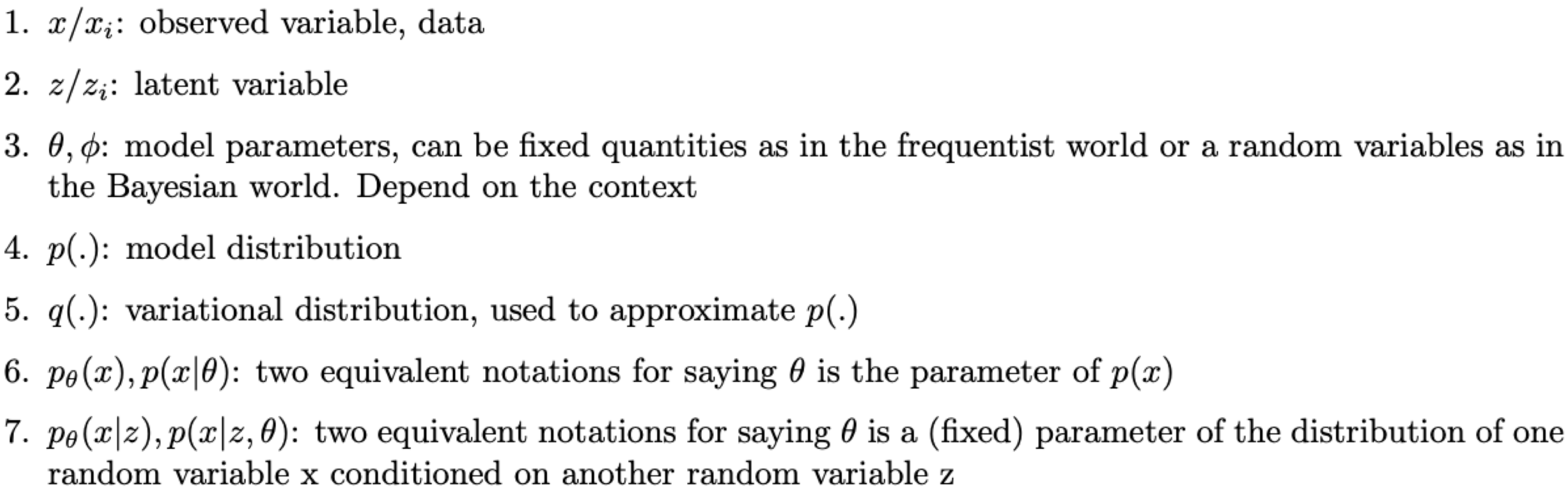
Some slides adopted from Dmitry Vetrov, Deep Bayes summer school 2019, and Sergey Levine, Deep Reinforcement Learning 2019

Agenda

- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder

As Sergey Levine pointed out in lecture, this topic is related to but not about reinforcement learning. We will see connections here and there

Notation Clarification



1. x/x_i: observed variable, data
2. z/z_i: latent variable
3. θ, ϕ: model parameters, can be fixed quantities as in the frequentist world or a random variables as in the Bayesian world. Depend on the context
4. $p(.)$: model distribution
5. $q(.)$: variational distribution, used to approximate $p(.)$
6. $p_\theta(x), p(x|\theta)$: two equivalent notations for saying θ is the parameter of $p(x)$
7. $p_\theta(x|z), p(x|z, \theta)$: two equivalent notations for saying θ is a (fixed) parameter of the distribution of one random variable x conditioned on another random variable z

Agenda

- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder

Approximate inference

Probabilistic model: $p(x, \theta) = p(x \mid \theta) p(\theta)$

Variational Inference

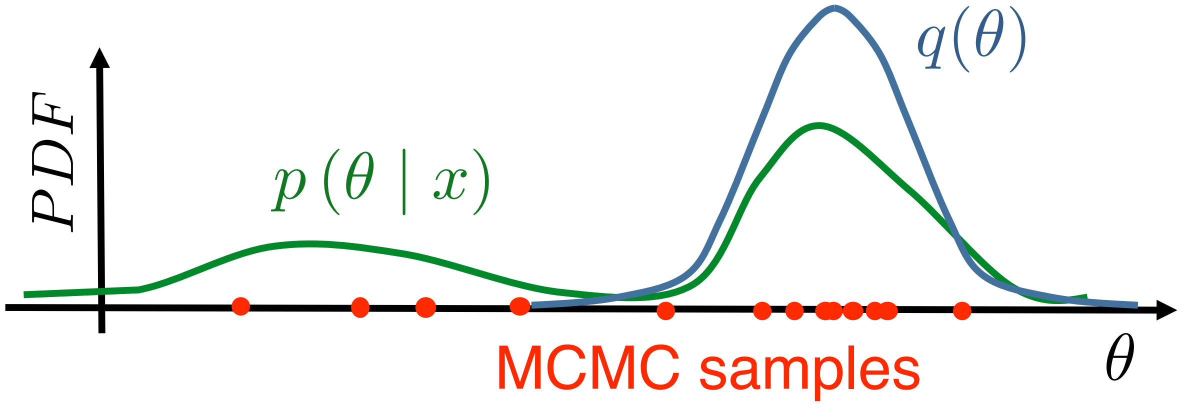
Approximate $p(\theta \mid x) \approx q(\theta) \in \mathcal{Q}$

- Biased
- Faster and more scalable

MCMC

Samples from unnormalized $p(\theta \mid x)$

- Unbiased
- Need a lot of samples



Variational inference

Probabilistic model: $p(x, \theta) = p(x \mid \theta)p(\theta)$

Main idea: find posterior approximation $p(\theta \mid x) \approx q(\theta) \in \mathcal{Q}$, using the following criterion function:

$$F(q) := KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$

Kullback-Leibler divergence

a good mismatch measure between two distributions over the **same domain**

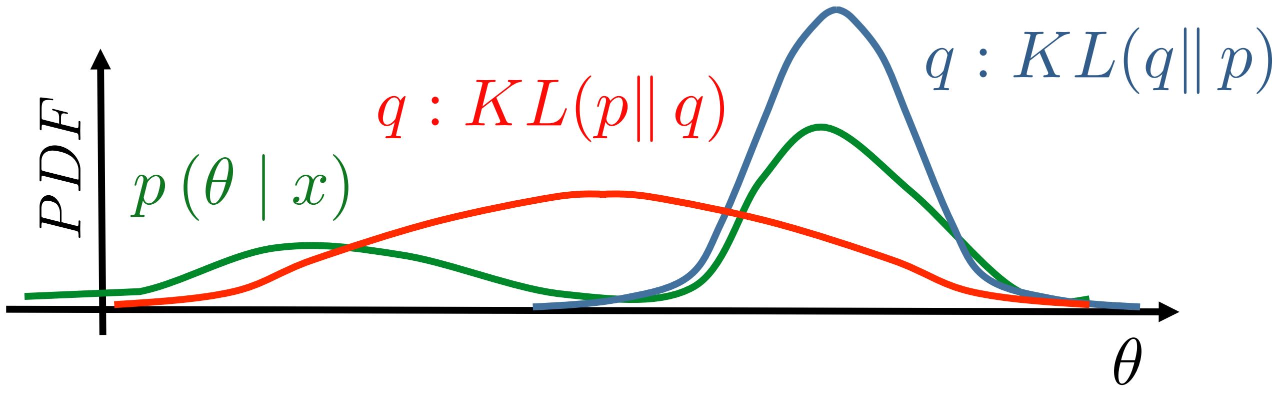
Kullback-Leibler divergence

A good mismatch measure between two distributions over the **same domain**

$$KL(q(\theta) \| p(\theta \mid x)) = \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta$$

Properties:

- $KL(q \| p) \geq 0$
- $KL(q \| p) = 0 \iff q = p$
- $KL(q \| p) \neq KL(p \| q)$



Variational inference

Probabilistic model: $p(x, \theta) = p(x \mid \theta)p(\theta)$

Main idea: find posterior approximation $p(\theta \mid x) \approx q(\theta) \in \mathcal{Q}$, using the following criterion function:

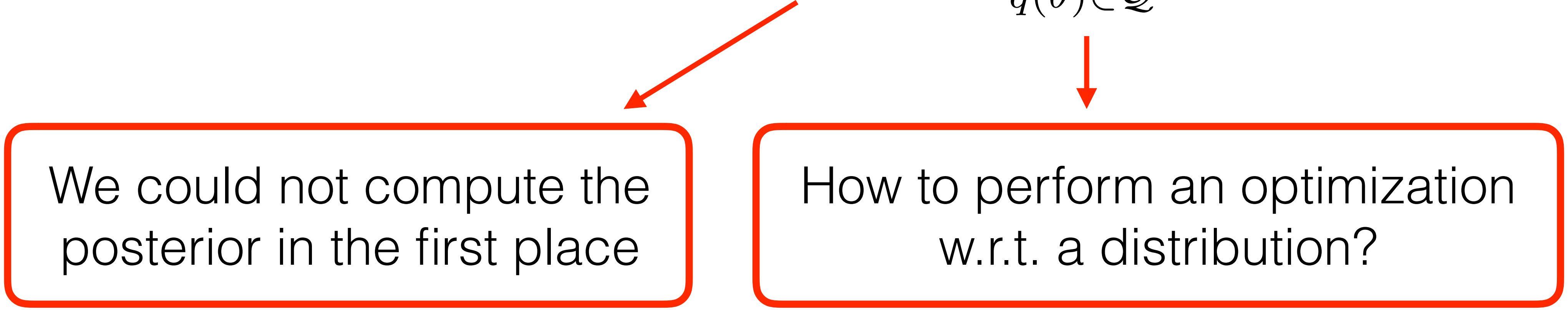
$$F(q) := KL(q(\theta) \| \, p(\theta \mid x)) \rightarrow \min_{q(\theta) \in \mathcal{Q}}$$

Variational inference

Probabilistic model: $p(x, \theta) = p(x \mid \theta)p(\theta)$

Main idea: find posterior approximation $p(\theta \mid x) \approx q(\theta) \in \mathcal{Q}$, using the following criterion function:

$$F(q) := KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$



We could not compute the posterior in the first place

How to perform an optimization w.r.t. a distribution?

Mathematical magic

$$\log p(x)$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta = \int q(\theta) \log \frac{p(x,\theta)}{p(\theta \mid x)} d\theta =$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta = \int q(\theta) \log \frac{p(x, \theta)}{p(\theta \mid x)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x, \theta) q(\theta)}{p(\theta \mid x) q(\theta)} d\theta =$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta = \int q(\theta) \log \frac{p(x, \theta)}{p(\theta \mid x)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x, \theta) q(\theta)}{p(\theta \mid x) q(\theta)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x, \theta)}{q(\theta)} d\theta + \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta =$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta = \int q(\theta) \log \frac{p(x,\theta)}{p(\theta \mid x)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x,\theta) q(\theta)}{p(\theta \mid x) q(\theta)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta + \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta =$$

$$= \mathcal{L}(q(\theta)) + KL(q(\theta) \| p(\theta \mid x))$$

Mathematical magic

$$\log p(x) = \int q(\theta) \log p(x) d\theta = \int q(\theta) \log \frac{p(x, \theta)}{p(\theta \mid x)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x, \theta) q(\theta)}{p(\theta \mid x) q(\theta)} d\theta =$$

$$= \int q(\theta) \log \frac{p(x, \theta)}{q(\theta)} d\theta + \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid x)} d\theta =$$

$$= \boxed{\mathcal{L}(q(\theta))} + \boxed{KL(q(\theta) \| p(\theta \mid x))}$$

Evidence lower bound (ELBO)

KL-divergence we need for VI

ELBO = Evidence Lower Bound

$$\log p(x) = \mathcal{L}(q(\theta)) + KL(q(\theta) \| p(\theta \mid x))$$

Evidence:

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)} = \frac{p(x \mid \theta)p(\theta)}{\int p(x \mid \theta)p(\theta)d\theta} = \frac{\text{Likelihood} \ \times \ \text{Prior}}{\text{Evidence}}$$

Evidence of the probabilistic model shows the total probability of observing the data.

Lower Bound: KL is non-negative → $\log p(x) \geq \mathcal{L}(q(\theta))$

Variational inference

Optimization problem with intractable posterior distribution:

$$F(q) := KL(q(\theta) \| \, p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$

Variational inference

Optimization problem with intractable posterior distribution:

$$F(q) := KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$

Let’s use our magic:

$$\log p(x) = \mathcal{L}(q(\theta)) + KL(q(\theta) \| p(\theta \mid x))$$

Variational inference

Optimization problem with intractable posterior distribution:

$$F(q) := KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$

Let’s use our magic:

$$\log p(x) = \mathcal{L}(q(\theta)) + KL(q(\theta) \| p(\theta \mid x))$$

does not depend on q

depend on q

Variational inference

Optimization problem with intractable posterior distribution:

$$F(q) := KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}}$$

Let's use our magic:

$$\log p(x) = \mathcal{L}(q(\theta)) + KL(q(\theta) \| p(\theta \mid x))$$

does not depend on q

depend on q

$$KL(q(\theta) \| p(\theta \mid x)) \to \min_{q(\theta) \in \mathcal{Q}} \quad \Leftrightarrow \quad \mathcal{L}(q(\theta)) \to \max_{q(\theta) \in \mathcal{Q}}$$

Variational inference

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta \to \max_{q(\theta) \in \mathcal{Q}}$$

Variational inference: ELBO interpretation

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x, \theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x \mid \theta) p(\theta)}{q(\theta)} d\theta =$$

Variational inference: ELBO interpretation

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x \mid \theta) p(\theta)}{q(\theta)} d\theta =$$

$$= \int q(\theta) \log p(x \mid \theta) d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta =$$

Variational inference: ELBO interpretation

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x \mid \theta) p(\theta)}{q(\theta)} d\theta =$$

$$= \int q(\theta) \log p(x \mid \theta) d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta =$$

$$= \mathbb{E}_{q(\theta)} \log p(x \mid \theta) - KL(q(\theta) \| \, p(\theta))$$

Variational inference: ELBO interpretation

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x \mid \theta) p(\theta)}{q(\theta)} d\theta =$$

$$= \int q(\theta) \log p(x \mid \theta) d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta =$$

$$= \underbrace{\boxed{\mathbb{E}_{q(\theta)} \log p(x \mid \theta)}}_{\text{data term}} - \underbrace{\boxed{KL(q(\theta) \| p(\theta))}}_{\text{regularizer}}$$

Variational inference: ELBO interpretation 2

Final optimization problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x|\theta)p(\theta)}{q(\theta)} d\theta$$

$$= \int q(\theta) \log p(x|\theta) d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta$$

$$= \mathbb{E}_{q(\theta)}[\log p(x|\theta)] + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta$$

$$= \mathbb{E}_{q(\theta)}[\log p(x|\theta)] + \int q(\theta) \log p(\theta) d\theta - \int q(\theta) \log q(\theta) d\theta$$

$$= \mathbb{E}_{q(\theta)}[\log p(x|\theta) + \log(p(\theta))] + \mathcal{H}(q(\theta))$$

Variational inference

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta \to \max_{q(\theta) \in \mathcal{Q}}$$

How to perform an optimization w.r.t. a distribution?

Variational inference

Final optimisation problem:

$$\mathcal{L}(q(\theta)) = \int q(\theta) \log \frac{p(x, \theta)}{q(\theta)} d\theta \to \max_{q(\theta) \in \mathcal{Q}}$$

How to perform an optimization w.r.t. a distribution?

Mean field approximation

Factorized family

$$q(\theta) = \prod_{j=1}^{m} q_j\left(\theta_j\right), \quad \theta = [\theta_1, \ldots, \theta_m]$$

Parametric approximation

Parametric family

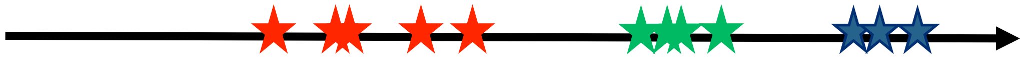
$$q(\theta) = q(\theta \mid \lambda)$$

Agenda

- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder

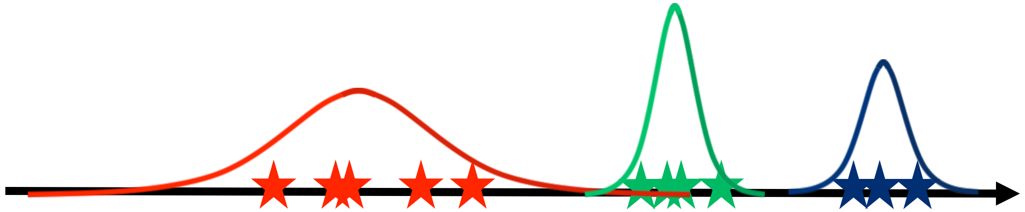
Latent variable modeling: example

- Now suppose we’re given several sets of points from different gaussians
- We need to estimate the parameters of those gaussians and their weights



Latent variable modeling: example

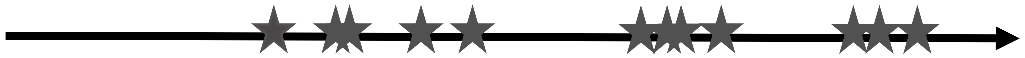
- Now suppose we’re given several sets of points from different gaussians
- We need to estimate the parameters of those gaussians and their weights



- The problem is as easy if we know what objects were generated from each gaussian

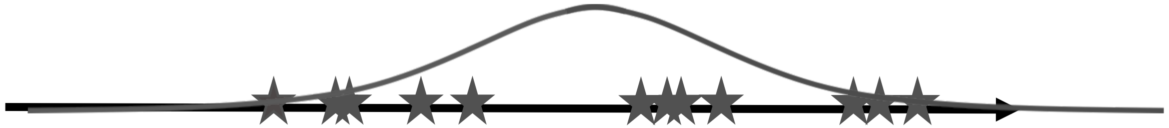
Latent variable modeling: example

- Now what if we do not know what objects were generated by each gaussian
- Of course we could still try to use a single gaussian model…



Latent variable modeling: example

- Now what if we do not know what objects were generated by each gaussian
- Of course we could still try to use a single gaussian model…
- … but there is a better way: latent variable model!



Mixture of gaussians

- For each object x_i we establish additional latent variable z_i which denotes the index of gaussian from which i-th object was generated

- Then our model is

$$p(X, Z|\theta) = \prod_{i=1}^{n} p(x_i, z_i|\theta) = \{\text{Product rule}\} = \prod_{i=1}^{n} p(x_i|z_i, \theta)p(z_i|\theta) = \prod_{i=1}^{n} \pi_{z_i}\mathcal{N}(x_i|\mu_{z_i}, \sigma^2_{z_i})$$

Mixture of gaussians

- For each object x_i we establish additional latent variable z_i which denotes the index of gaussian from which i-th object was generated

- Then our model is

$$p(X,Z|\theta) = \prod_{i=1}^{n} p(x_i, z_i|\theta) = \{\text{Product rule}\} = \prod_{i=1}^{n} p(x_i|z_i,\theta)p(z_i|\theta) = \prod_{i=1}^{n} \pi_{z_i}\mathcal{N}(x_i|\mu_{z_i}, \sigma^2_{z_i})$$

- Here $\pi_j = p(z_i = j)$ are prior probability of j-th gaussian and $\theta = \{\mu_j, \sigma_j, \pi_j\}_{j=1}^{K}$ are the parameters to be estimated

- If we know both X and Z we obtain explicit ML-solution:

$$\theta_{ML} = \arg\max_\theta p(X,Z|\theta) = \arg\max_\theta \log p(X,Z|\theta)$$

Latent variable model objective

- When z is unknown. We need to maximize the incomplete log likelihood (sum over z) for the mixture of Gaussians model

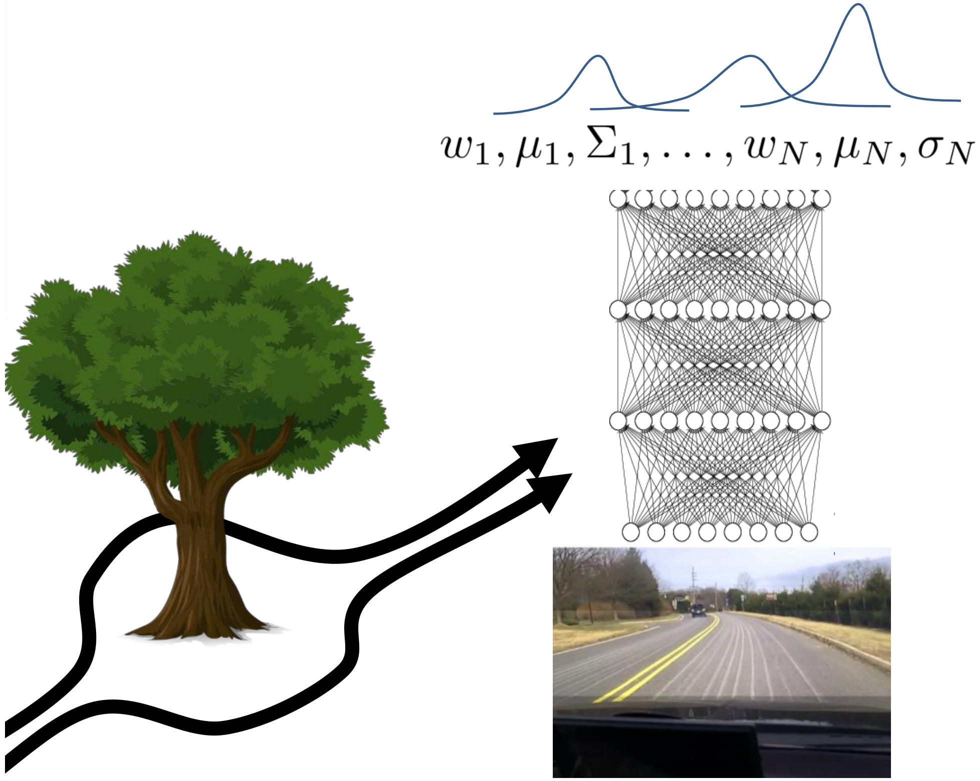
$$\log p_\theta(x) = \log \sum_z p_\theta(x|z)p(z)$$

- For general latent variable z, when z can be continues, we use integral instead of summation

$$\log p_\theta(x) = \log \int p_\theta(x|z)p(z)dz$$

Latent variable model in RL

- Generate Multi-modal policies



How do we train latent variable models?

the model: $p_\theta(x)$

the data: $\mathcal{D} = \{x_1, x_2, x_3, \dots, x_N\}$

maximum likelihood fit:

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \log p_\theta(x_i)$$

$$p(x) = \int p(x|z)p(z)dz$$

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \log \left(\int p_\theta(x_i|z)p(z)dz \right)$$

How do we train latent variable models?

the model: $p_\theta(x)$

$$p(x) = \int p(x|z)p(z)dz$$

the data: $\mathcal{D} = \{x_1, x_2, x_3, \dots, x_N\}$

maximum likelihood fit:

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \log p_\theta(x_i)$$

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \log \left(\int \underset{\uparrow}{p_\theta(x_i|z)p(z)dz} \right)$$

completely intractable

Optimize the lower bound

Rewrite the objective

$$\log p(x_i) = D_{\mathrm{KL}}(q_i(\ z\ )\|p(z|x_i)) + \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \overbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p, q_i)}$$

Optimize the lower bound

Rewrite the objective

$$\log p(x_i) = D_{\mathrm{KL}}(q_i(\ z\ )\|p(z|x_i)) + \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \overbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p, q_i)}$$

How many quantities are we optimizing over?

Optimize the lower bound

Rewrite the objective

$$\log p(x_i) = D_{\mathrm{KL}}(q_i(z)\|p(z|x_i)) + \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \mathcal{L}_i(p, q_i)$$

$$\log p(x_i) \geq \overbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p, q_i)}$$

How many quantities are we optimizing over?

What are we maximizing when the lower bound is tight?

Estimating the log-likelihood

alternative: *expected* log-likelihood:

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i E_{z \sim p(z|x_i)}[\log p_\theta(x_i, z)]$$

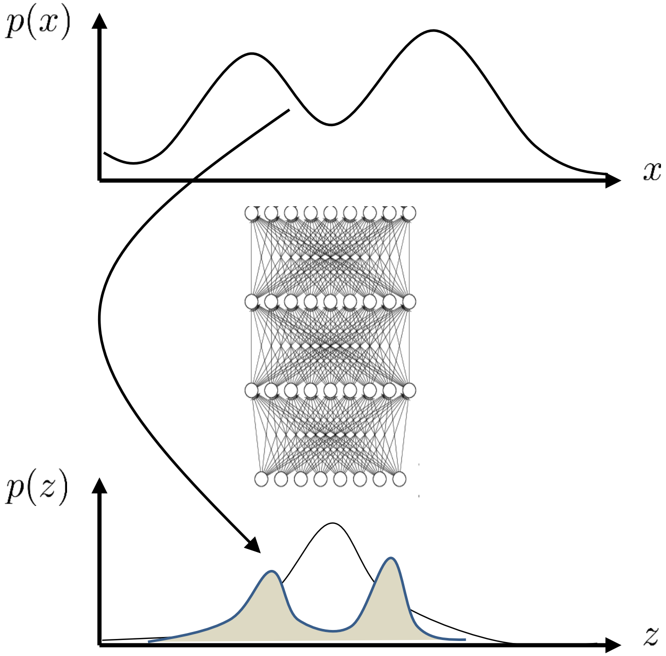
Estimating the log-likelihood

alternative: *expected* log-likelihood:

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i E_{z \sim p(z|x_i)}[\log p_\theta(x_i, z)]$$

intuition: "guess" most likely z given x_i, and pretend it's the right one

...but there are many possible values of z so use the distribution $p(z|x_i)$



How do we use this?

$$\log p(x_i) \geq \overbrace{E_{z \sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p, q_i)}$$

$$\cancel{\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \log p_\theta(x_i)}$$

$$\theta \leftarrow \arg\max_\theta \frac{1}{N} \sum_i \mathcal{L}_i(p, q_i)$$

How do we use this?

$$\log p(x_i) \geq \overbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p,q_i)}$$

$$\cancel{\theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \log p_\theta(x_i)}$$

$$\theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \mathcal{L}_i(p,q_i)$$

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p,q_i)$:
        sample $z \sim q_i(z)$
        $\nabla_\theta \mathcal{L}_i(p,q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p,q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p,q_i)$
```

How do we use this?

$$\log p(x_i) \geq \overbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}^{\mathcal{L}_i(p, q_i)}$$

$$\cancel{\theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \log p_\theta(x_i)} \qquad \theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \mathcal{L}_i(p, q_i)$$

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i( z )$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

how?

How do we use this?

$$\log p(x_i) \geq \underbrace{E_{z\sim q_i(z)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_i)}_{\mathcal{L}_i(p, q_i)}$$

$$\cancel{\theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \log p_\theta(x_i)} \qquad \theta \leftarrow \arg\max_\theta \frac{1}{N}\sum_i \mathcal{L}_i(p, q_i)$$

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i(z)$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

how?

let's say $q_i(z) = \mathcal{N}(\mu_i, \sigma_i)$

use gradient $\nabla_{\mu_i} \mathcal{L}_i(p, q_i)$ and $\nabla_{\sigma_i} \mathcal{L}_i(p, q_i)$

gradient ascent on μ_i, σ_i

What’s the problem?

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i( z )$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

How many parameters are there?

let’s say $q_i(z) = \mathcal{N}(\mu_i, \sigma_i)$

use gradient $\nabla_{\mu_i} \mathcal{L}_i(p, q_i)$ and $\nabla_{\sigma_i} \mathcal{L}_i(p, q_i)$

gradient ascent on μ_i, σ_i

What’s the problem?

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i(\, z\, )$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

let’s say $q_i(z) = \mathcal{N}(\mu_i, \sigma_i)$

use gradient $\nabla_{\mu_i} \mathcal{L}_i(p, q_i)$ and $\nabla_{\sigma_i} \mathcal{L}_i(p, q_i)$

gradient ascent on μ_i, σ_i

How many parameters are there? $|\theta| + (|\mu_i| + |\sigma_i|) \times N$

Review

- What have we done so far?
  - We saw variational inference and latent variable model
  - We use variational inference to change the training objective of latent variable model from an intractable integration to a tractable lower bound
  - The problem of optimizing this lower bound is that there are too many parameters

Review

- What have we done so far?
  - We saw variational inference and latent variable model
  - We use variational inference to change the training objective of latent variable model from an intractable integration to a tractable lower bound
  - The problem of optimizing this lower bound is that there are too many parameters
- Now let’s go from the classic era to deep era

Agenda

- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder

What’s the problem?

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i(\, z \,)$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

let’s say $q_i(z) = \mathcal{N}(\mu_i, \sigma_i)$

use gradient $\nabla_{\mu_i} \mathcal{L}_i(p, q_i)$ and $\nabla_{\sigma_i} \mathcal{L}_i(p, q_i)$

gradient ascent on μ_i, σ_i

How many parameters are there? $|\theta| + (|\mu_i| + |\sigma_i|) \times N$

What’s the problem?

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}_i(p, q_i)$:
        sample $z \sim q_i( z )$
        $\nabla_\theta \mathcal{L}_i(p, q_i) \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_i(p, q_i)$
    update $q_i$ to maximize $\mathcal{L}_i(p, q_i)$
```

let’s say $q_i(z) = \mathcal{N}(\mu_i, \sigma_i)$

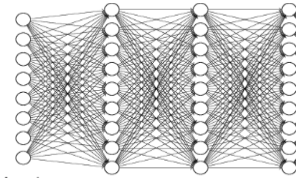
use gradient $\nabla_{\mu_i} \mathcal{L}_i(p, q_i)$ and $\nabla_{\sigma_i} \mathcal{L}_i(p, q_i)$

gradient ascent on μ_i, σ_i

How many parameters are there? $|\theta| + (|\mu_i| + |\sigma_i|) \times N$

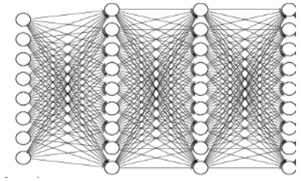
intuition: $q_i(z)$ should approximate $p(z|x_i)$ what if we learn a *network* $q_i(z) = q(z|x_i) \approx p(z|x_i)$?

z



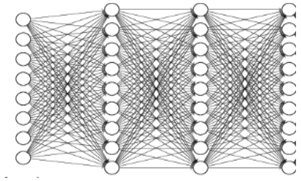
$p_\theta(x|z)$

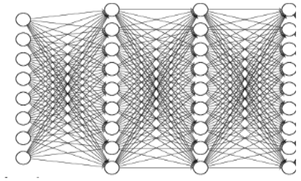
x



$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$

Amortized variational inference

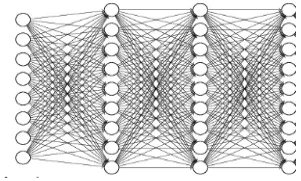
z  $p_\theta(x|z)$

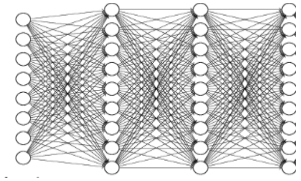
x  $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

$$\log p(x_i) \geq \overbrace{E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_\phi(z|x_i))}^{\mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))}$$

Amortized variational inference

z  $p_\theta(x|z)$

x  $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$

$$\log p(x_i) \geq \overbrace{E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_\phi(z|x_i))}^{\mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))}$$

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

how do we calculate this?

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\boxed{\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

$$\mathcal{L}_i = E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_\phi(z|x_i))$$

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

look up formula for entropy of a Gaussian

$$\mathcal{L}_i = E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_\phi(z|x_i))$$

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

look up formula for entropy of a Gaussian

$$\mathcal{L}_i = E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)] + \mathcal{H}(q_\phi(z|x_i))$$

Non-trivial, different from θ

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\boxed{\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

look up formula for entropy of a Gaussian

$$\mathcal{L}_i = \underbrace{E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)]}_{J(\phi) = E_{z\sim q_\phi(z|x_i)}[r(x_i, z)]} + \mathcal{H}(q_\phi(z|x_i))$$

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

look up formula for entropy of a Gaussian

$$\mathcal{L}_i = \underbrace{E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)]}_{J(\phi) = E_{z \sim q_\phi(z|x_i)}[r(x_i, z)]} + \mathcal{H}(q_\phi(z|x_i))$$

can just use policy gradient!

$$\nabla J(\phi) \approx \frac{1}{M} \sum_j \nabla_\phi \log q_\phi(z_j|x_i) r(x_i, z_j)$$

Direct policy differentiation

$$\theta^\star = \arg\max_\theta \underbrace{E_{\tau\sim p_\theta(\tau)}\left[\sum_t r(\mathbf{s}_t,\mathbf{a}_t)\right]}_{J(\theta)}$$

a convenient identity

$$\underline{\pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)} = \pi_\theta(\tau)\frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} = \underline{\nabla_\theta \pi_\theta(\tau)}$$

$$J(\theta) = E_{\tau\sim\pi_\theta(\tau)}[\underbrace{r(\tau)}_{\sum_{t=1}^{T} r(\mathbf{s}_t,\mathbf{a}_t)}] = \int \pi_\theta(\tau) r(\tau) d\tau$$

$$\nabla_\theta J(\theta) = \int \underline{\nabla_\theta \pi_\theta(\tau)} r(\tau) d\tau = \int \underline{\pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)} r(\tau) d\tau = E_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) r(\tau)]$$

Amortized variational inference

```
for each $x_i$ (or mini-batch):
    calculate $\nabla_\theta \mathcal{L}(p_\theta(x_i|z), q_\phi(z|x_i))$:
        sample $z \sim q_\phi(z|x_i)$
        $\nabla_\theta \mathcal{L} \approx \nabla_\theta \log p_\theta(x_i|z)$
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
    $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{L}$
```

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

look up formula for entropy of a Gaussian

$$\mathcal{L}_i = \underbrace{E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z) + \log p(z)]}_{} + \mathcal{H}(q_\phi(z|x_i))$$

$$J(\phi) = E_{z \sim q_\phi(z|x_i)}[r(x_i, z)]$$

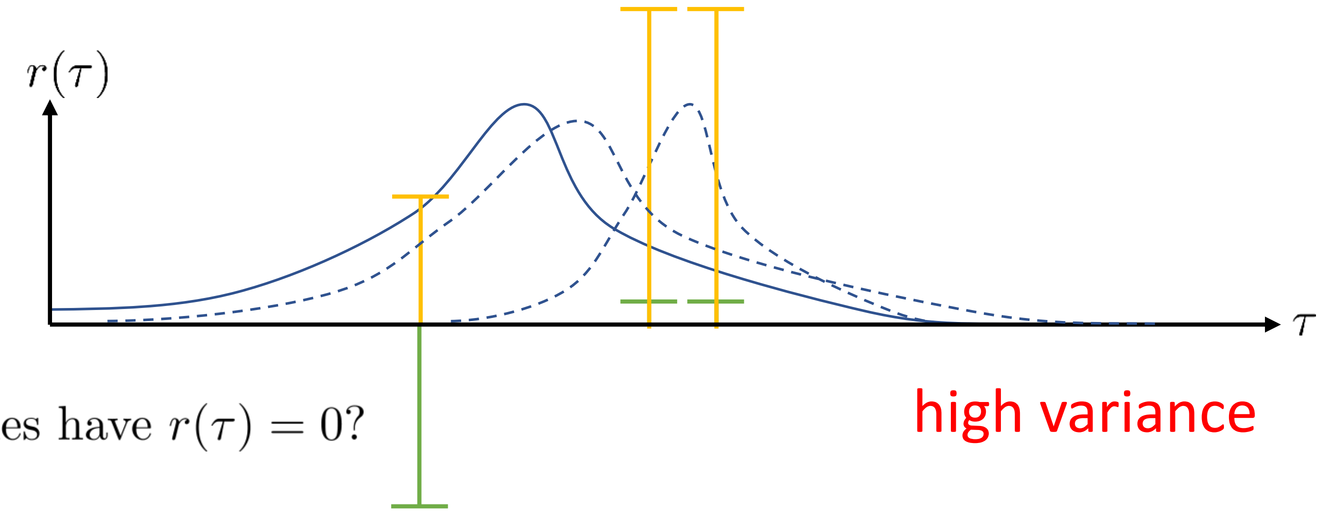
can just use policy gradient!

What's wrong with this gradient?

$$\nabla J(\phi) \approx \frac{1}{M} \sum_j \nabla_\phi \log q_\phi(z_j|x_i) r(x_i, z_j)$$

What is wrong with the policy gradient?

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \log \pi_\theta(\tau) r(\tau)$$



even worse: what if the two "good" samples have $r(\tau) = 0$?

high variance

The reparameterization trick

Is there a better way?

The reparameterization trick

Is there a better way?

$$J(\phi) = E_{z \sim q_\phi(z|x_i)}[r(x_i, z)]$$

$$= E_{\epsilon \sim \mathcal{N}(0,1)}[r(x_i, \mu_\phi(x_i) + \epsilon\sigma_\phi(x_i))]$$

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

$$z = \mu_\phi(x) + \epsilon\sigma_\phi(x)$$

The reparameterization trick

Is there a better way?

$$J(\phi) = E_{z \sim q_\phi(z|x_i)}[r(x_i, z)]$$

$$= E_{\epsilon \sim \mathcal{N}(0,1)}[r(x_i, \mu_\phi(x_i) + \epsilon\sigma_\phi(x_i))]$$

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

$$z = \mu_\phi(x) + \epsilon\sigma_\phi(x)$$

$$\epsilon \sim \mathcal{N}(0,1)$$

independent of ϕ!

The reparameterization trick

Is there a better way?

$$J(\phi) = E_{z \sim q_\phi(z|x_i)}[r(x_i, z)]$$

$$= E_{\epsilon \sim \mathcal{N}(0,1)}[r(x_i, \mu_\phi(x_i) + \epsilon \sigma_\phi(x_i))]$$

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$

$$z = \mu_\phi(x) + \epsilon \sigma_\phi(x)$$

$\epsilon \sim \mathcal{N}(0,1)$

independent of ϕ!

estimating $\nabla_\phi J(\phi)$:

sample $\epsilon_1, \ldots, \epsilon_M$ from $\mathcal{N}(0,1)$ (a single sample works well!)

$$\nabla_\phi J(\phi) \approx \frac{1}{M} \sum_j \nabla_\phi r(x_i, \mu_\phi(x_i) + \epsilon_j \sigma_\phi(x_i))$$

Reparameterization trick vs. policy gradient

- Policy gradient
  - Can handle both discrete and continuous latent variables
  - High variance, requires multiple samples & small learning rates
- Reparameterization trick
  - Only continuous latent variables
  - Very simple to implement
  - Low variance

Correct: Gumbel Softmax extends reparameterization to discrete variables

$$\nabla_\phi J(\phi) \approx \frac{1}{M} \sum_j \nabla_\phi \log q_\phi(z_j|x_i) r(x_i, z_j)$$

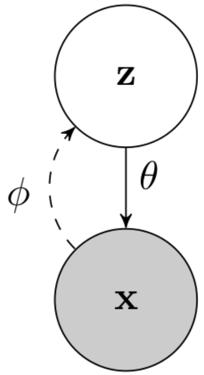
$$\nabla_\phi J(\phi) \approx \frac{1}{M} \sum_j \nabla_\phi r(x_i, \mu_\phi(x_i) + \epsilon_j \sigma_\phi(x_i))$$

Agenda

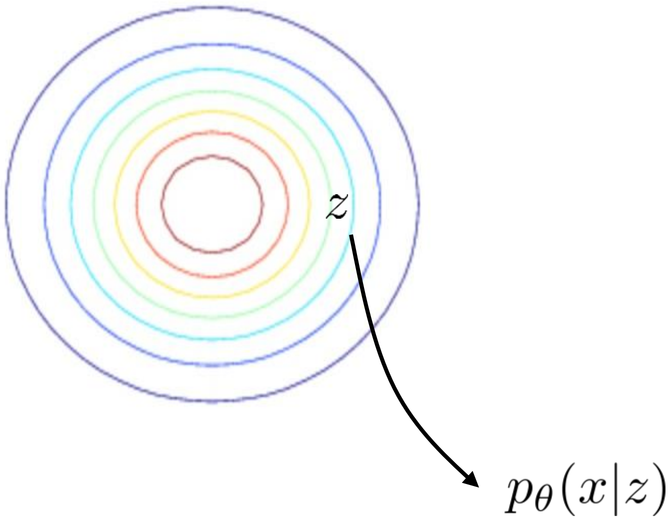
- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder

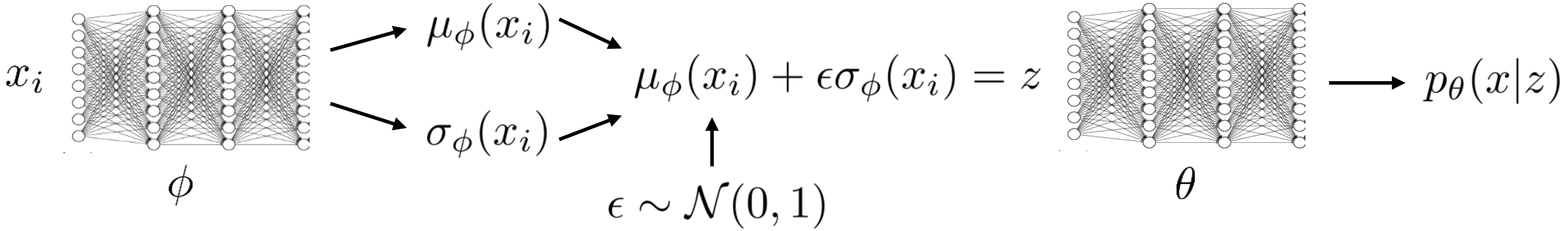
The *variational* autoencoder

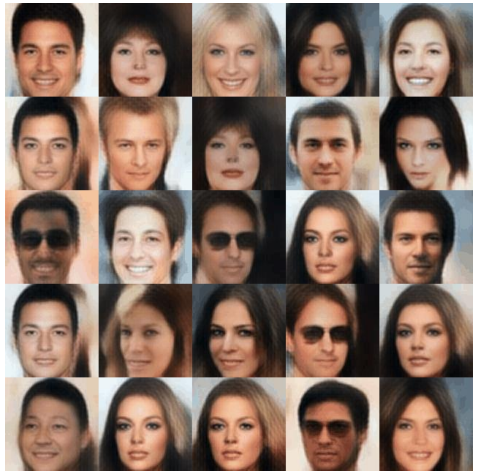
$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$



$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma_\theta(z))$



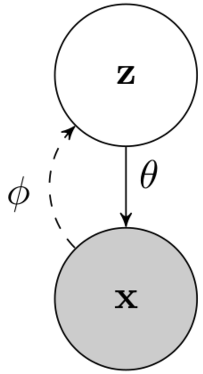

x_i
ϕ
$\mu_\phi(x_i)$
$\sigma_\phi(x_i)$
$\mu_\phi(x_i) + \epsilon\sigma_\phi(x_i) = z$
$\epsilon \sim \mathcal{N}(0,1)$
θ
$p_\theta(x|z)$



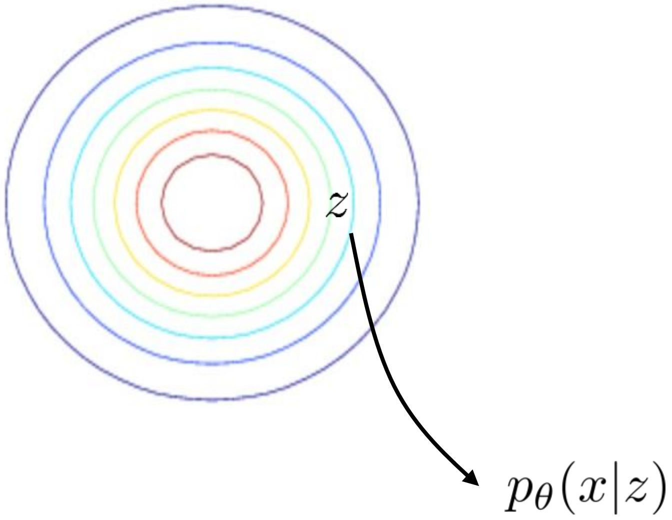
$$\max_{\theta,\phi} \frac{1}{N} \sum_i \log p_\theta(x_i|\mu_\phi(x_i) + \epsilon\sigma_\phi(x_i)) - D_{\mathrm{KL}}(q_\phi(z|x_i)\|p(z))$$

Using the variational autoencoder

$$q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x))$$



$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma_\theta(z))$$



$$p(x) = \int p(x|z)p(z)dz$$

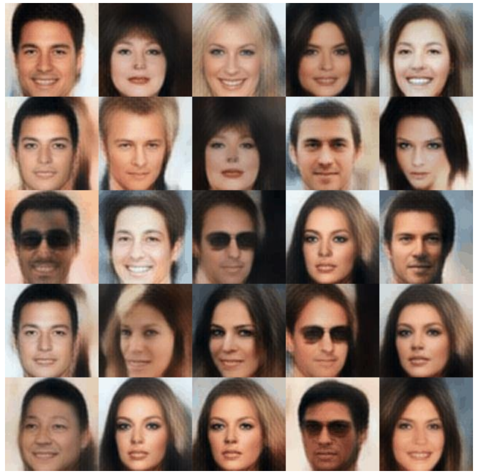
why does this work?

sampling:

$$z \sim p(z)$$

$$x \sim p(x|z)$$

$$\mathcal{L}_i = E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\text{KL}}(q_\phi(z|x_i)\|p(z))$$



Agenda

- Review Variational Inference
- Latent Variable Models
- Amortized Variational Inference and The Reparameterization Trick
- Variational Auto-Encoder
- VAE Variants

β-VAE

- Idea: we have two terms in the VAE loss function. We can add an additional parameter to balance them

$$\mathcal{L}_i = E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\mathrm{KL}}(q_\phi(z|x_i) \| p(z))$$

β-VAE

- Idea: we have two terms in the VAE loss function. We can add an additional parameter to balance them

$$\mathcal{L}_i = E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\mathrm{KL}}(q_\phi(z|x_i)\|p(z))$$

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\ D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

β-VAE

- Idea: we have two terms in the VAE loss function. We can add an additional parameter to balance them

$$\mathcal{L}_i = E_{z \sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\mathrm{KL}}(q_\phi(z|x_i)||p(z))$$

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \; D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

- More flexibility
- For $\beta > 1$, it encourages conditional independence, which leads to disentangled representations
- Not a valid lower bound of the incomplete log-likelihood anymore

β-VAE as a constraint optimization problem

- Consider the optimization problem

$$\max_{\phi,\theta} \mathbb{E}_{x\sim\mathbf{D}}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right] \quad \text{subject to } \; D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon$$

β-VAE as a constraint optimization problem

- Consider the optimization problem

$$\max_{\phi,\theta} \mathbb{E}_{x\sim\mathbf{D}} \left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right] \qquad \textbf{subject to} \;\; D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon$$

- Rewrite as a Lagrangian

$$\mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\left(D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \epsilon\right)$$

β-VAE as a constraint optimization problem

- Consider the optimization problem

$$\max_{\phi,\theta} \mathbb{E}_{x\sim\mathbf{D}} \left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right] \quad \text{subject to } \; D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon$$

- Rewrite as a Lagrangian

$$\mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\,(D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \epsilon)$$

$$\mathcal{F}(\theta, \phi, \beta; \mathbf{x}, \mathbf{z}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \; D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

VAE prior

- Idea: other than a simple isotropic normal distribution N(0, I), what is a more reasonable prior distribution of latent variable z. Especially when we want z to be multimodal

Variational Deep Embedding (VaDE)

- Use mixture of Gaussian as the prior. There will be one more layer of latency, a discrete latent random variable c for the latent variable z

Variational Deep Embedding (VaDE)

- Use mixture of Gaussian as the prior. There will be one more layer of latency, a discrete latent random variable c for the latent variable z

VAE

$$\log p_\theta(x) = \log \int p_\theta(x|z)p(z)dz \qquad \mathcal{L}_i = E_{z\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)] - D_{\mathrm{KL}}(q_\phi(z|x_i)\|p(z))$$

$$\log p(\mathbf{x}) = \log \int_{\mathbf{z}} \sum_{c} p(\mathbf{x}, \mathbf{z}, c)d\mathbf{z} \qquad \mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) = E_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{KL}(q(\mathbf{z}, c|\mathbf{x})||p(\mathbf{z}, c))$$

VaDE

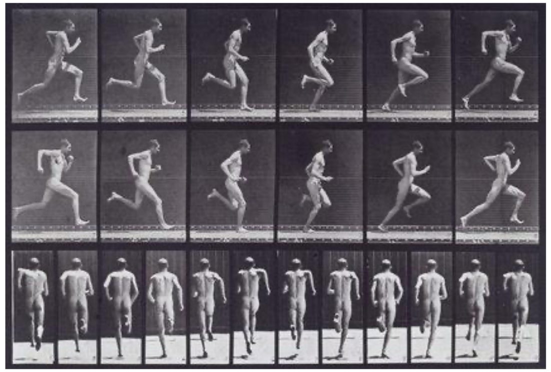
VaDE assumption: $q(\mathbf{z}, c|\mathbf{x}) = q(\mathbf{z}|\mathbf{x})q(c|\mathbf{x}).$

Paper Reference

- Higgins, Irina, et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." *Iclr* 2.5 (2017): 6
- Jiang, Zhuxi, et al. "Variational deep embedding: An unsupervised and generative approach to clustering." *arXiv preprint arXiv:1611.05148* (2016).
- Tomczak, Jakub M., and Max Welling. "VAE with a VampPrior." *arXiv preprint arXiv:1705.07120* (2017).
- Jang, Eric, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax."

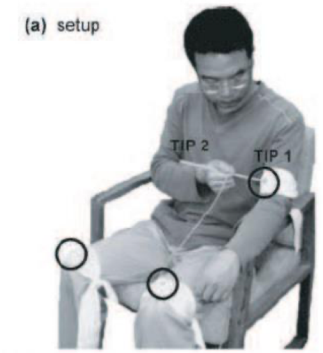
We’ll see more of this for…

Using RL/control + variational inference to model human behavior



Muybridge (c. 1870)

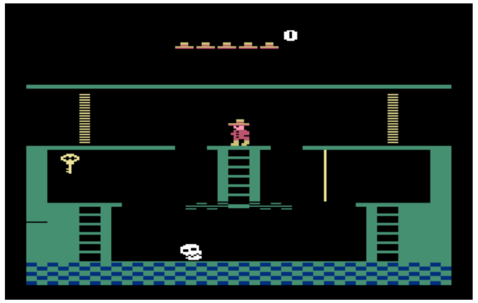
Mombaur et al. ‘09



Li & Todorov ‘06

Ziebart ‘08

Using generative models and variational inference for exploration



Thanks!

Q & A