# Latent Variable Models

Dr. Zulqarnain Khan

(Slides from: Stefano Ermon)

# So far…

- Autoregressive models:
  - Chain rule based factorization is fully general
  - Compact representation via conditional independence and/or neural parameterizations
- Autoregressive models Pros:
  - Easy to evaluate likelihoods
  - Easy to train
- Autoregressive models Cons:
  - Requires an ordering
  - Generation is sequential
  - Cannot learn features in an unsupervised way

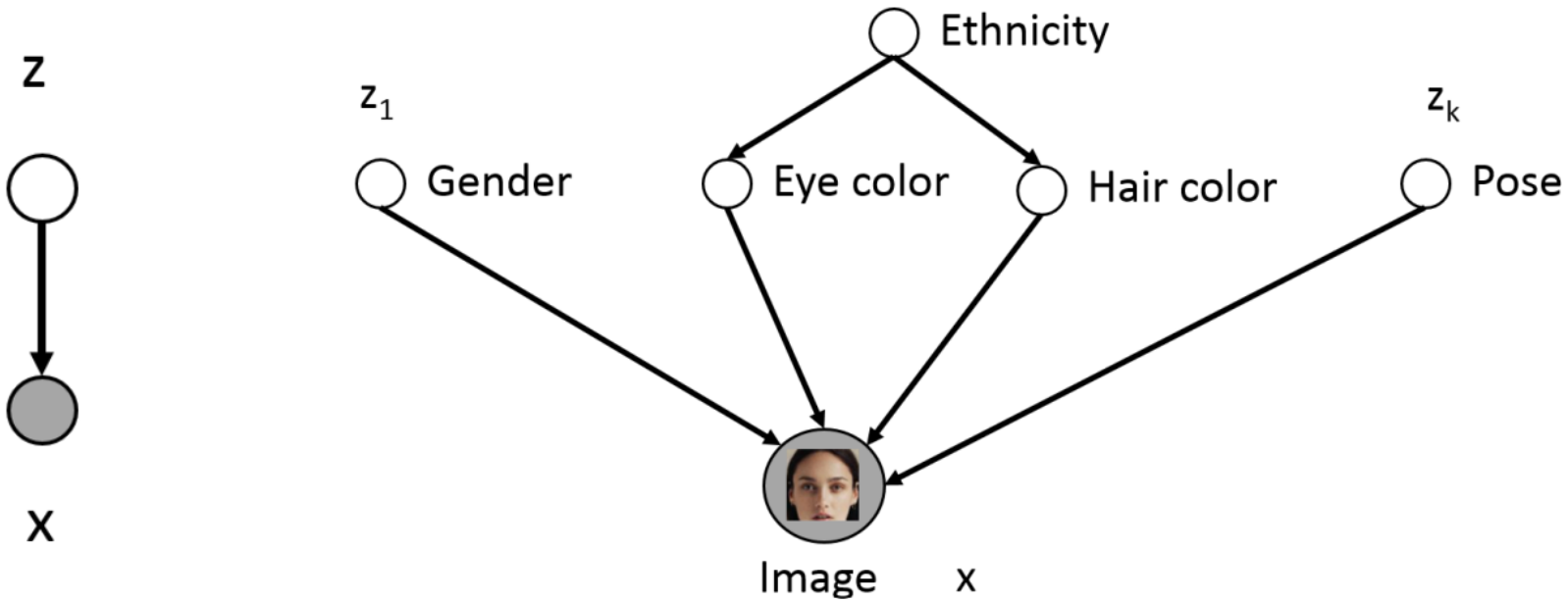
# Latent Variable Models

- Mixture Models
- Variational Autoencoder
- Variational Inference and Learning

# Motivation

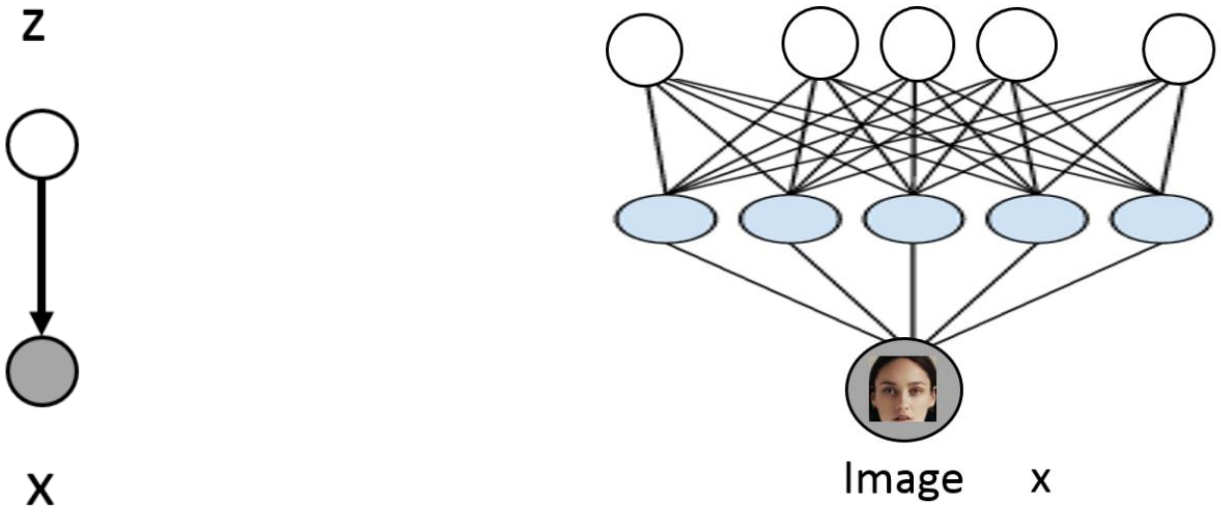
- Lots of variability in images **x** due to gender, eye color, hair color, pose, etc. However, unless images are annotated, these factors of variation are not explicitly available (latent).
- Idea: explicitly model these factors using latent variables **z**

# Motivation




- Only shaded variables **x** are observed in the data (pixel values)
- Latent variables **z** correspond to high level features
  - If **z** chosen properly, p(**x**|**z**) could be much simpler than p(**x**)
  - If we had trained this model, then we could identify features via p(**z** | **x**), e.g., p(EyeColor = Blue|**x**)
- Challenge: Very difficult to specify these conditionals by hand

# Deep Latent Variable Models

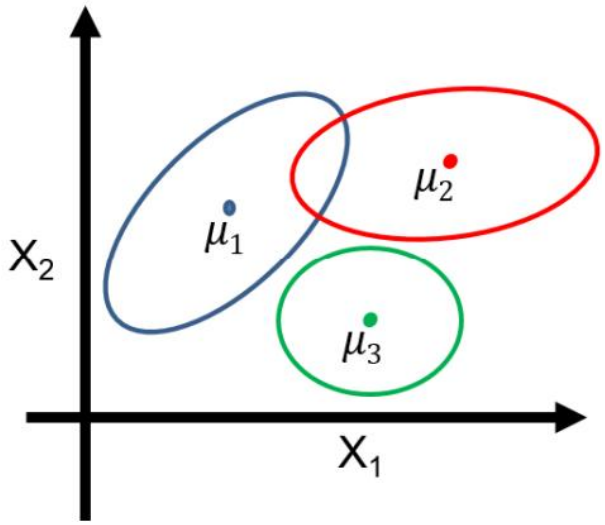



- Use neural networks to model the conditionals (deep latent variable models):
  - $\mathbf{z} \sim N(0, I)$
  - p(x | z) = $N\ (\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta, \Sigma_\theta$ are neural networks
- Hope that after training, z will correspond to meaningful latent factors of variation (features). Unsupervised representation learning. As before, features can be computed via p(z | x)

# A Shallow Latent Variable Model

Mixture of Gaussians. Bayes net: $z \rightarrow x$

- $z \sim Categorical(1, \ldots, K)$
- $p(x|z = k) = N(\mu_k, \Sigma_k)$

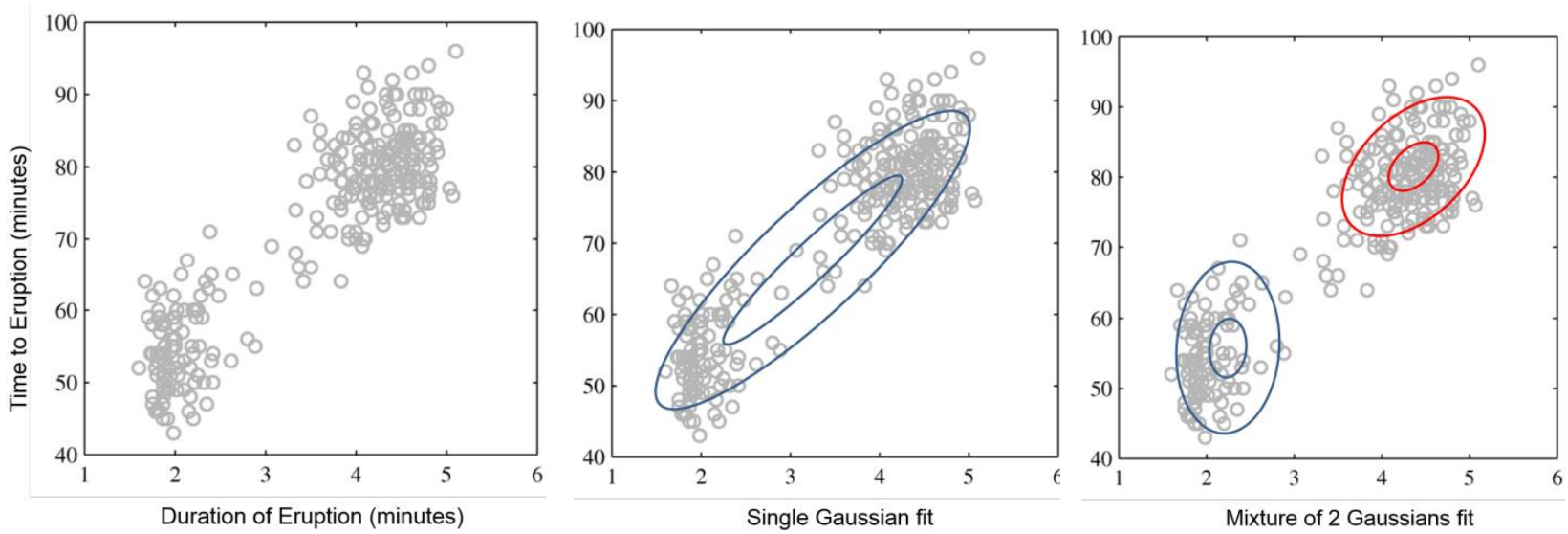



Generative process

- Pick a mixture component $k$ by sampling $z$
- Generate a data point by sampling from that Gaussian

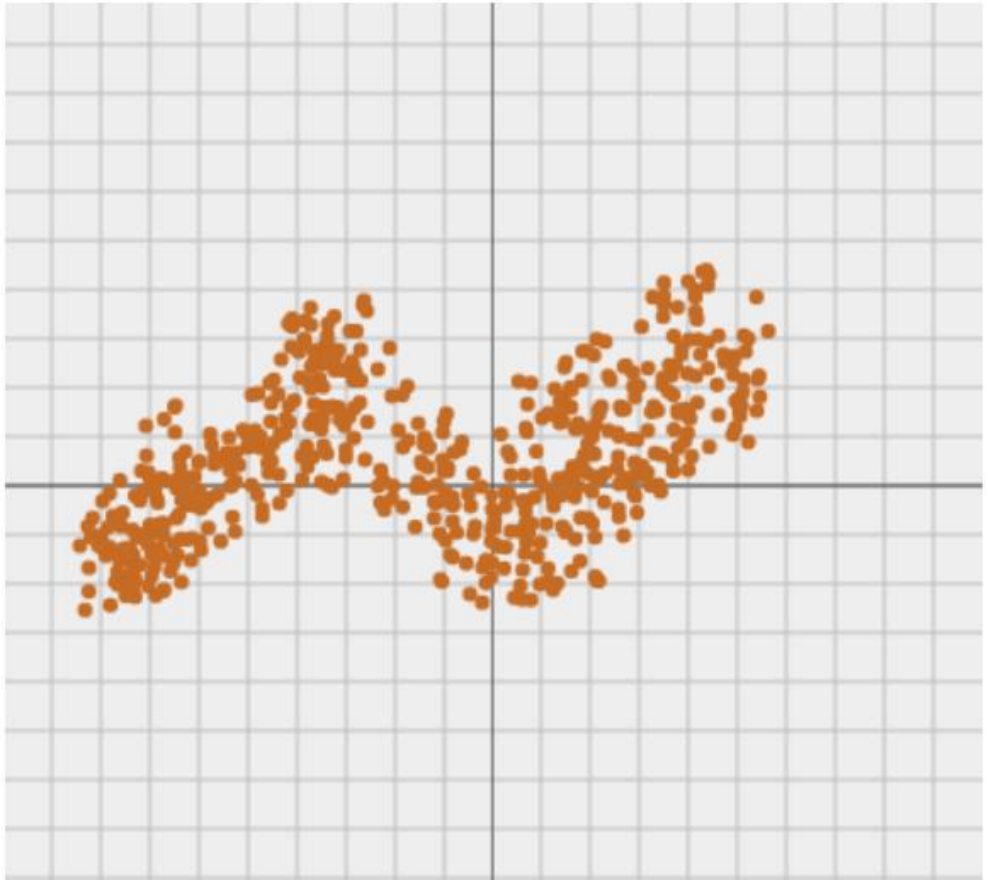
# Mixture of Gaussians

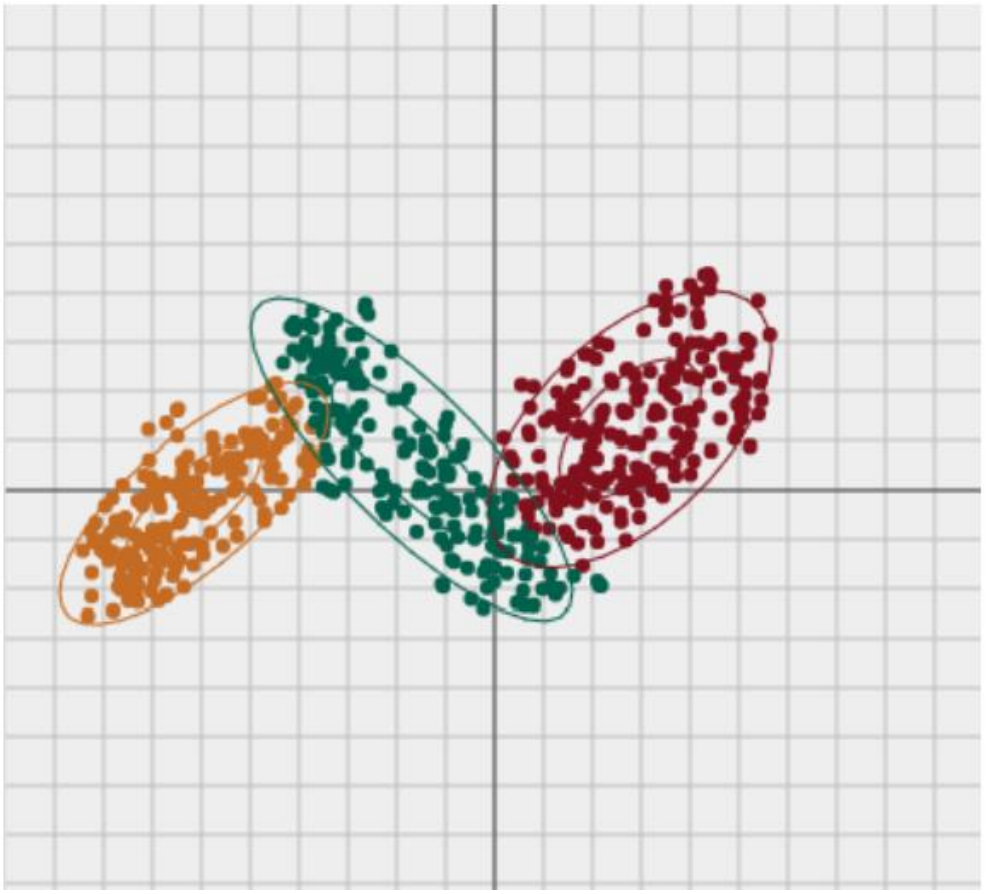
- $z \sim Categorical(1, \ldots, K)$
- $p(x|z = k) = N(\mu_k, \Sigma_k)$




- Clustering: The posterior p(z | x) identifies the mixture component
- Unsupervised learning: We are hoping to learn from unlabeled data (ill-posed problem)

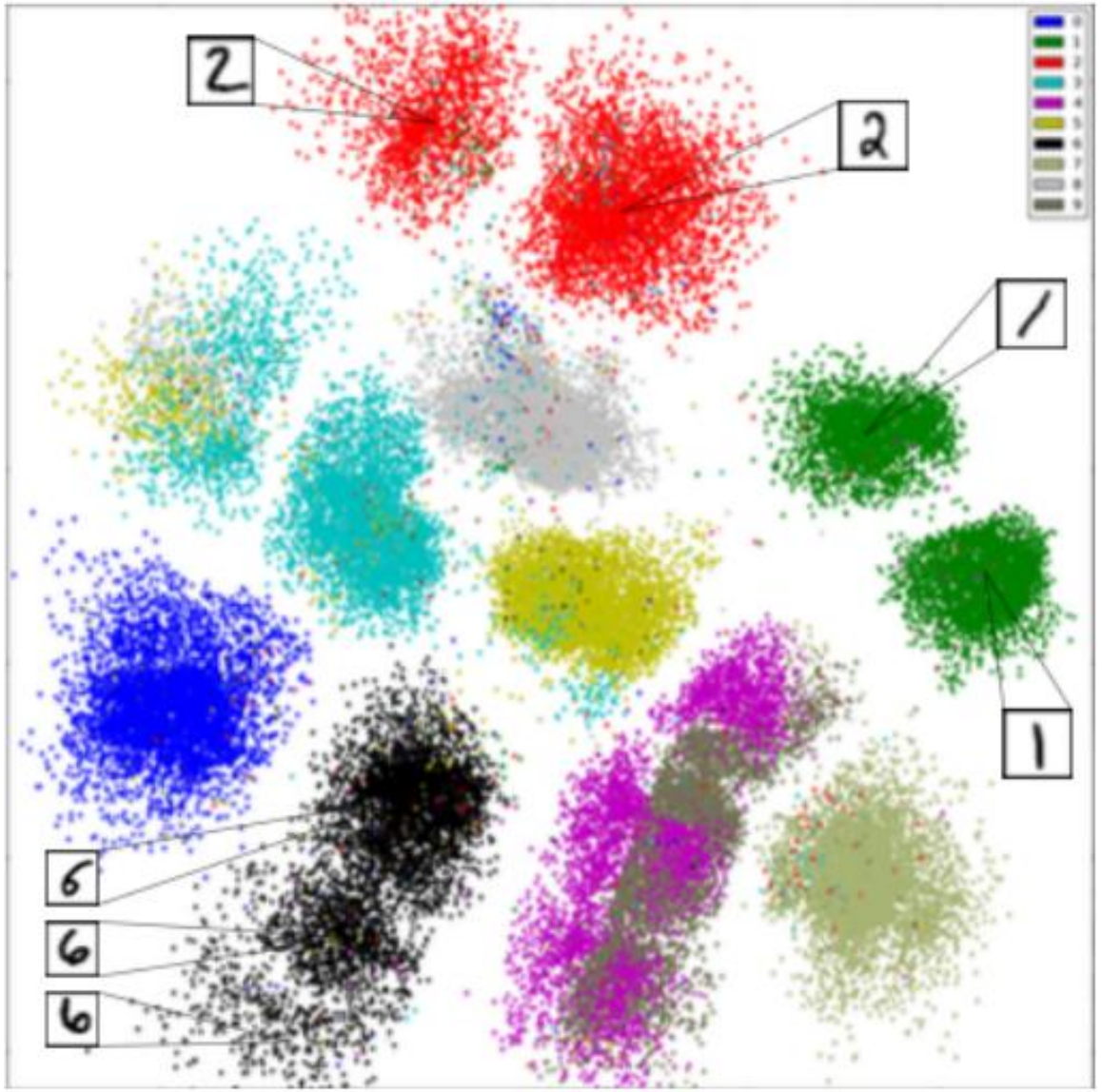
# Unsupervised Learning





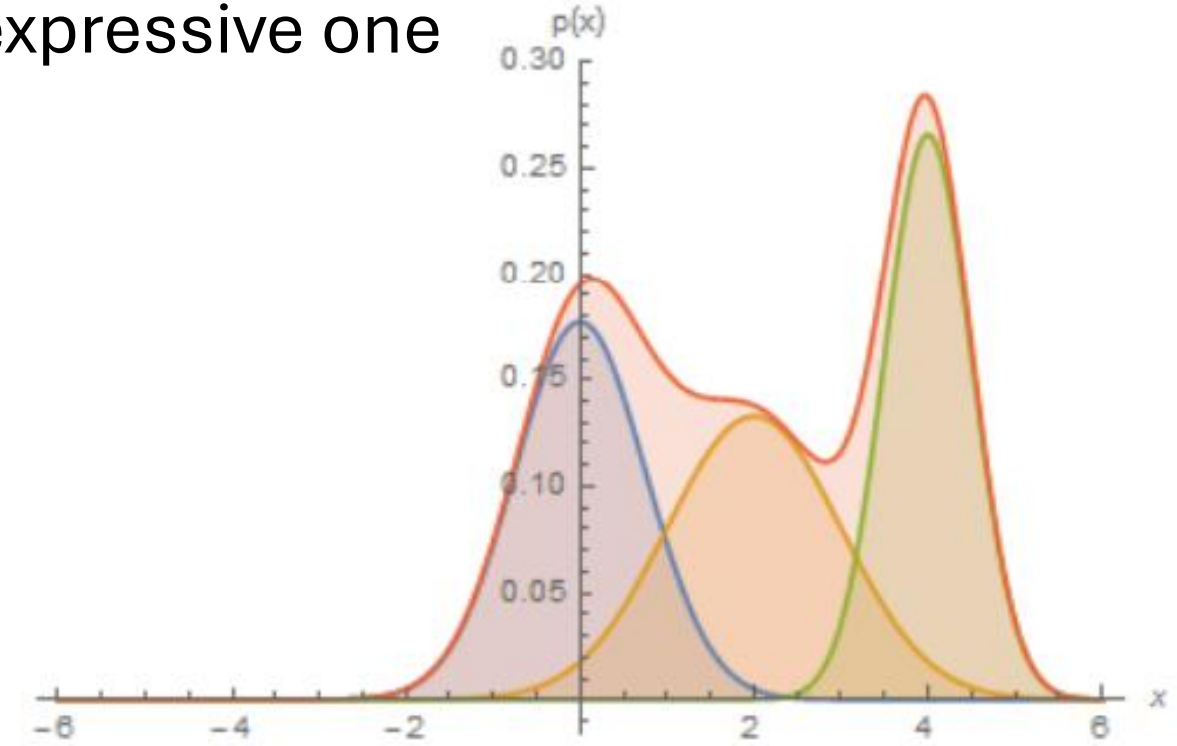
- Shown is the posterior probability that a data point was generated by the i-th mixture component, P(z = i|x)

# Unsupervised learning



# Mixture models

- Alternative motivation: Combine simple models into a more complex and expressive one

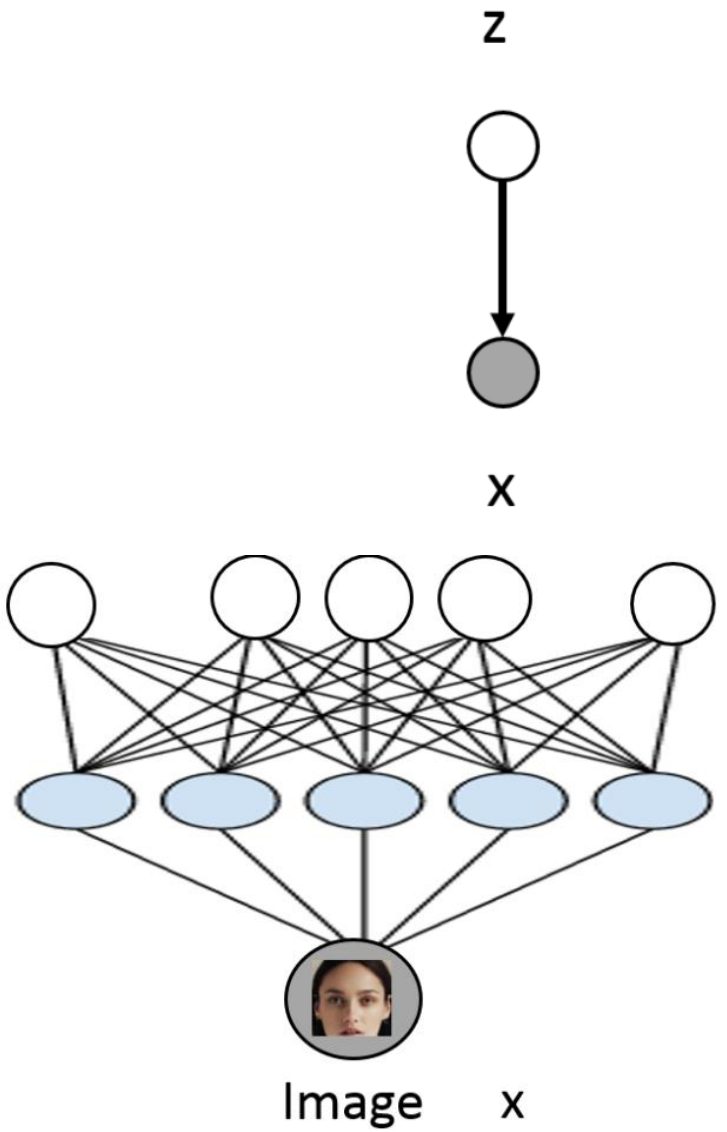



$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x} \mid \mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{z} = k) \underbrace{\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)}_{\text{component}}$$

# Motivation

A mixture of infinite number of Gaussians:

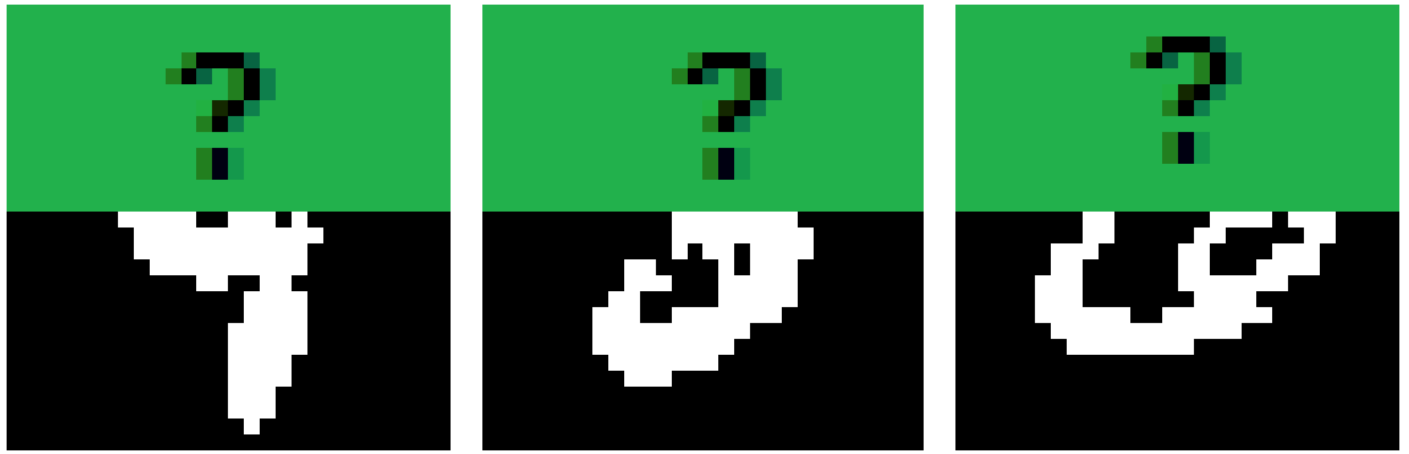
- $z \sim N(0, I)$
- $p(x|z) = N(\mu_\theta(z), \Sigma_\theta(z))$, where $\mu_\theta, \Sigma_\theta$ come from NNs.
  - $\mu_\theta(z) = \sigma(Az + c)$
  - $\Sigma_\theta(z) = diag(\exp(\sigma(Bz + d))$
  - $\theta = A, B, c, d$
- Even though $p(x|z)$ is simple, the marginal $p(x)$ is very complex/flexible.



# So far…

- Latent Variable models:
  - Allow us to define complex models p(x) in terms of simpler building blocks p(x|z)
  - Natural for unsupervised learning tasks (clustering, unsupervised representation learning)
  - No free lunch: much more difficult to learn compared to fully observed autoregressive models

# Marginal Likelihood



- Suppose some pixel values are missing at train time (e.g., top half)
- Let X denote observed random variables, and Z the unobserved ones (hidden/latent)
- Suppose we have a model for the joint distribution (e.g., PixelCNN) $p(X, Z; \theta)$

What is the probability $p(X = \bar{x}; \theta)$ of observing a training data point $\bar{x}$?

$$\sum_{\mathbf{z}} p(\mathbf{X} = \bar{\mathbf{x}}, \mathbf{Z} = \mathbf{z}; \theta) = \sum_{\mathbf{z}} p(\bar{\mathbf{x}}, \mathbf{z}; \theta)$$

- Need to consider all possible ways to complete the image (fill green part)

# Variational Autoencoder Marginal Likelihood

A mixture of infinite number of Gaussians:

- $z \sim N(0, I)$
- $p(x|z) = N(\mu_\theta(z), \Sigma_\theta(z))$, where $\mu_\theta, \Sigma_\theta$ are NNs
- Z are unobserved at train time (hidden/latent)
- Suppose we have a model for the joint distribution. What is the probability $p(X = \bar{x};\ \theta)$ of observing a training data point $\bar{x}$?

$$\int_{\mathbf{z}} p(\mathbf{X} = \bar{\mathbf{x}}, \mathbf{Z} = \mathbf{z}; \theta) d\mathbf{z} = \int_{\mathbf{z}} p(\bar{\mathbf{x}}, \mathbf{z}; \theta) d\mathbf{z}$$

# Partially observed data

- Suppose that our joint distribution is $p(X, Z; \theta)$
- We have a dataset D, where for each datapoint the X variables are observed (e.g., pixel values) and the variables Z are never observed (e.g., cluster or class id.). $D = \{x^{(1)}, \cdots, x^{(M)}\}$.
- Maximum likelihood learning:

$$\log \prod_{\mathbf{x} \in \mathcal{D}} p(\mathbf{x}; \theta) = \sum_{\mathbf{x} \in \mathcal{D}} \log p(\mathbf{x}; \theta) = \sum_{\mathbf{x} \in \mathcal{D}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)$$

- Evaluating $\log \sum_z p(x, z; \theta)$ can be intractable. Suppose we have 30 binary latent features, $z \in \{0, 1\}^{30}$. Evaluating $\sum_z p(x, z; \theta)$ involves a sum with $2^{30}$ terms. For continuous variables, $\log \int_z p(x, z; \theta) dz$ is often intractable. Gradients $\nabla_\theta$ also hard to compute.
- Need approximations. One gradient evaluation per training data point $x \in D$, so approximation needs to be cheap.

# First attempt: Naive Monte Carlo

- Likelihood function $p_\theta(x)$ for Partially Observed Data is hard to compute:

$$p_\theta(\mathbf{x}) = \sum_{\text{All values of } \mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z}) = |\mathcal{Z}| \sum_{\mathbf{z} \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} p_\theta(\mathbf{x}, \mathbf{z}) = |\mathcal{Z}| \mathbb{E}_{\mathbf{z} \sim Uniform(\mathcal{Z})} [p_\theta(\mathbf{x}, \mathbf{z})]$$

We can think of it as an (intractable) expectation. Monte Carlo to the rescue:

- Sample z (1) , · · · , z (k) uniformly at random
- Approximate expectation with sample average

$$\sum_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z}) \approx |\mathcal{Z}| \frac{1}{k} \sum_{j=1}^{k} p_\theta(\mathbf{x}, \mathbf{z}^{(j)})$$

Works in theory but not in practice. For most z, $p_\theta(x, z)$ is very low (most completions don't make sense). Some completions have large $p_\theta(x, z)$ but we will never "hit" likely completions by uniform random sampling. Need a clever way to select $z^{(j)}$

# Evidence Lower Bound

- Likelihood function $p_\theta(x)$ for Partially Observed Data is hard to compute:

$$\log\left(\sum_{\mathbf{z}\in\mathcal{Z}} p_\theta(\mathbf{x},\mathbf{z})\right) = \log\left(\sum_{\mathbf{z}\in\mathcal{Z}} \frac{q(\mathbf{z})}{q(\mathbf{z})} p_\theta(\mathbf{x},\mathbf{z})\right) = \log\left(\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]\right)$$

- Idea: Use Jensen's Inequality for concave functions $\mathrm{E}[f(X)] \geq f[E(X)]$ . Log is a concave function

$$\log\left(\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]\right) \geq \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\left[\log\left(\frac{p_\theta(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right)\right]$$

- This is called Evidence Lower BOund (ELBO). $p_\theta(x)$ is evidence.

# Variational Inference

- Suppose q(z) is any probability distribution over the hidden variables
- Evidence lower bound (ELBO) holds for any q

$$\begin{aligned}\log p(\mathbf{x};\theta) &\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log\left(\frac{p_\theta(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right) \\ &= \sum_{\mathbf{z}} q(\mathbf{z}) \log p_\theta(\mathbf{x},\mathbf{z}) \underbrace{- \sum_{\mathbf{z}} q(\mathbf{z}) \log q(\mathbf{z})}_{\text{Entropy } H(q) \text{ of } q} \\ &= \sum_{\mathbf{z}} q(\mathbf{z}) \log p_\theta(\mathbf{x},\mathbf{z}) + H(q)\end{aligned}$$

- Equality holds if $q = p(z|x;\theta)$

$$\log p(\mathbf{x};\theta) = \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{z},\mathbf{x};\theta) + H(q)$$

# Connection to KL-Divergence

- Note that

$$D_{KL}(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x};\theta)) = -\sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{z},\mathbf{x};\theta) + \log p(\mathbf{x};\theta) - H(q) \geq 0$$

- Rearranging, we re-derived the Evidence lower bound (ELBO)

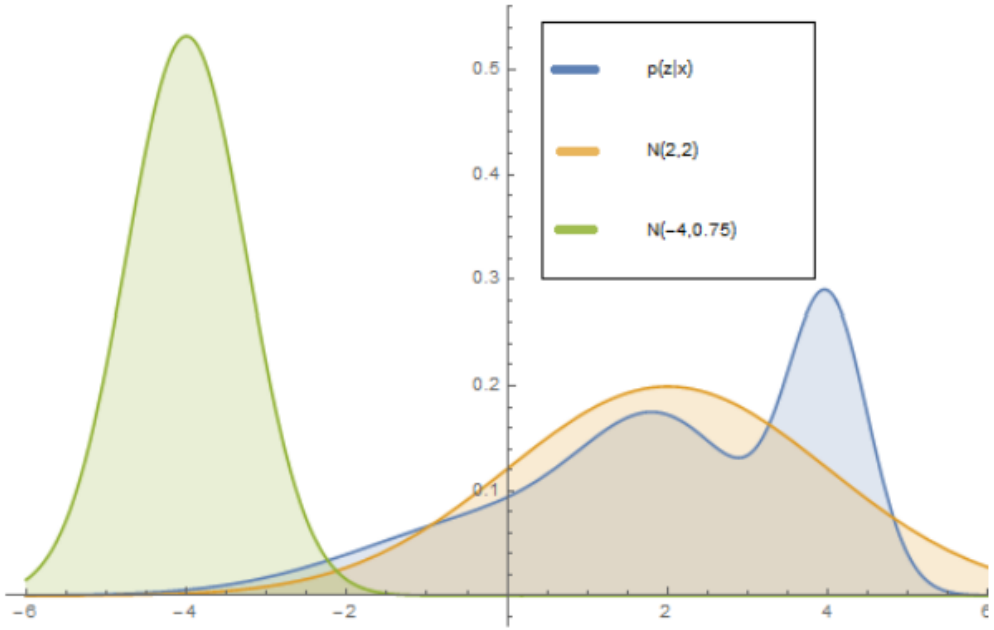
$$\log p(\mathbf{x};\theta) \geq \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{z},\mathbf{x};\theta) + H(q)$$

- Equality holds if $q = p(z|x;\theta)$ because $D_{KL}(q(z) \parallel p(z|x;\theta))$=0

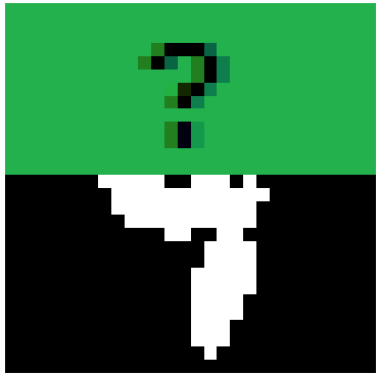
- In general $\log p(x;\theta) = ELBO + D_{KL}(q(z)||p(z|x;\theta)$. The closer $q(z)$ is to $p(z|x;\theta)$, the closer the ELBO is to the true log-likelihood

# How to choose q(z)?

- Variational inference:
  - Parameterize $q(z)$ by parameters $\phi$.
  - Pick $\phi$ so that $q(z;\ \phi)$ is as close as possible to the often intractable $p(z|x;\theta)$. For example $q(z;\phi) = N(\phi_1, \phi_2)$
  - In the figure, the posterior $p(z|x;\theta)$ (blue) is better approximated by $N(2,2)$ (orange) than $N(-4, 0.75)$ (green)



# A variational approximation to the posterior

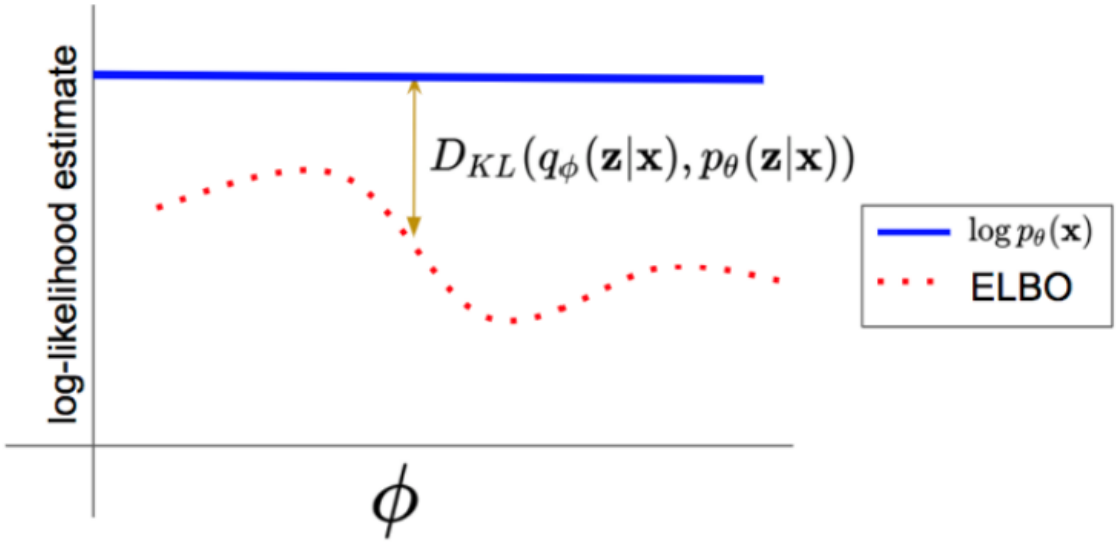


- Assume $p(x^{top}, x^{bottom}; \theta)$ assigns high probability to images that look like digits. In this example, we assume $z = x^{top}$ are unobserved (latent)
- Suppose $q(x^{top}; \phi)$ is a (tractable) probability distribution over the hidden variables (missing pixels in this example) $x^{top}$ parameterized by $\phi$ (variational parameters). Let's assume it is a joint Bernoulli distribution

$$q(\mathbf{x}^{top}; \phi) = \prod_{\text{unobserved variables } \mathbf{x}_i^{top}} (\phi_i)^{\mathbf{x}_i^{top}} (1 - \phi_i)^{(1-\mathbf{x}_i^{top})}$$

- Is $\phi_i = 0.5 \; \forall i$ a good approximation to the posterior $p(x^{top}|x^{bottom}; \theta)$?
- Is $\phi_i = 1 \; \forall i$ a good approximation to the posterior?
- Is $\phi_i \approx 1$ for pixels i corresponding to the top part of digit 9 a good approximation?

# Why ELBO?




$$\log p(\mathbf{x};\theta) \geq \sum_{\mathbf{z}} q(\mathbf{z};\phi)\log p(\mathbf{z},\mathbf{x};\theta) + H(q(\mathbf{z};\phi)) = \underbrace{\mathcal{L}(\mathbf{x};\theta,\phi)}_{\text{ELBO}}$$

$$= \mathcal{L}(\mathbf{x};\theta,\phi) + D_{KL}(q(\mathbf{z};\phi)\|p(\mathbf{z}|\mathbf{x};\theta))$$

- The better $q(z;\phi)$ can approximate the posterior p(z|x; θ), the smaller $D_{KL}(q(z;\phi) \parallel p(z|x;\theta))$ we can achieve, the closer ELBO will be to $\log p(x;\ \theta)$. Next: jointly optimize over θ and ϕ to maximize the ELBO over a dataset