Probabilistic Graphical Models

Brown University CSCI 2950-P, Spring 2013
Prof. Erik Sudderth

Lecture 8:
Inference & Learning for Exponential Families, Expectation Maximization (EM) Algorithm

Some figures courtesy Michael Jordan's draft textbook,
An Introduction to Probabilistic Graphical Models

Exponential Families of Distributions

$$
\begin{aligned}
p(\mathbf{x}|\boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta})} h(\mathbf{x}) \exp[\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})] \\
&= h(\mathbf{x}) \exp[\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}) - A(\boldsymbol{\theta})]
\end{aligned}
\qquad
\begin{aligned}
Z(\boldsymbol{\theta}) &= \int_{\mathcal{X}^m} h(\mathbf{x}) \exp[\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})] d\mathbf{x} \\
A(\boldsymbol{\theta}) &= \log Z(\boldsymbol{\theta})
\end{aligned}
$$

$\phi(x) \in \mathbb{R}^d \longrightarrow$ fixed vector of *sufficient statistics* (features), specifying the family of distributions

$\theta \in \Theta \longrightarrow$ unknown vector of *natural parameters*, determine particular distribution in this family

$Z(\theta) > 0 \longrightarrow$ normalization constant or *partition function*, ensuring this is a valid probability distribution

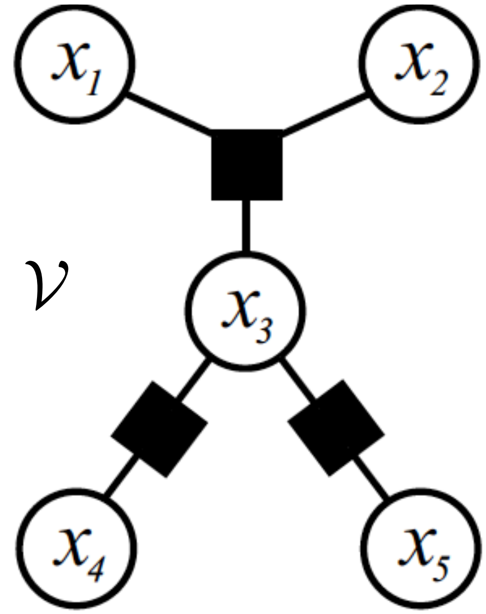
$h(x) > 0 \longrightarrow$ *reference measure* independent of parameters (for many models, we simply have $h(x) = 1$)

To ensure this construction is valid, we take

$$\Theta = \{\theta \in \mathbb{R}^d \mid Z(\theta) < \infty\}$$

Factor Graphs & Exponential Families

$$p(x) = \frac{1}{Z(\theta)} \prod_{f \in \mathcal{F}} \psi_f(x_f \mid \theta_f)$$



$\mathcal{F} \longrightarrow$ set of hyperedges linking subsets of nodes $f \subseteq \mathcal{V}$

$\mathcal{V} \longrightarrow$ set of N nodes or vertices, $\{1, 2, \ldots, N\}$

$Z \longrightarrow$ normalization constant (partition function)

- A *factor graph* is created from non-negative potential functions
- To guarantee non-negativity, we typically define potentials as

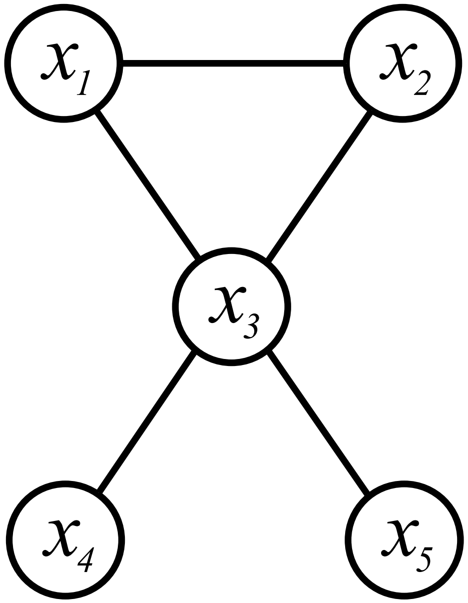
$$\psi_f(x_f \mid \theta_f) = \nu_f(x_f) \exp\left\{ \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) \right\}$$

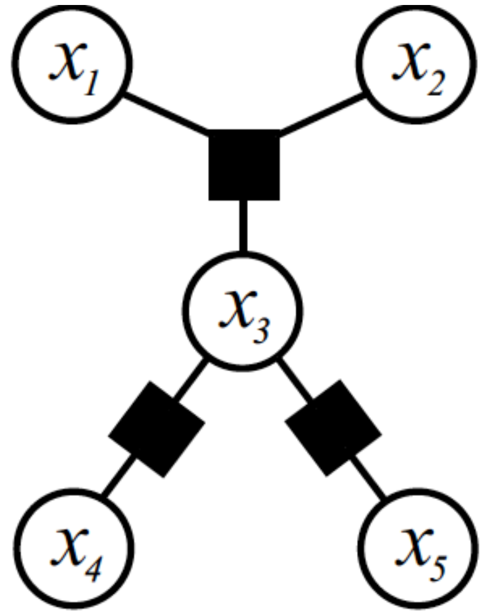
Local exponential family:

$$\theta_f \triangleq \{\theta_{fa} \mid a \in \mathcal{A}_f\}$$

$$p(x \mid \theta) = \left(\prod_{f \in \mathcal{F}} \nu_f(x_f) \right) \exp\left\{ \sum_{f \in \mathcal{F}} \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) - \Phi(\theta) \right\} \qquad \Phi(\theta) = \log Z(\theta)$$

Undirected Graphs & Exp. Families





$$p(x \mid \theta) = \left(\prod_{f \in \mathcal{F}} \nu_f(x_f) \right) \exp \left\{ \sum_{f \in \mathcal{F}} \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) - \Phi(\theta) \right\} \qquad \Phi(\theta) = \log Z(\theta)$$

- Pick features to define an exponential family of distributions
- Use factor graph to represent structure of chosen statistics
- Create undirected graph with a clique for every factor node
- *Result:* Visualization of Markov properties of your family

Generalized Linear Models

- General framework for modeling non-Gaussian data with linear prediction, using exponential families:
  - Construct instance-specific natural parameters:

$$\theta_i = w^T \phi(x_i)$$

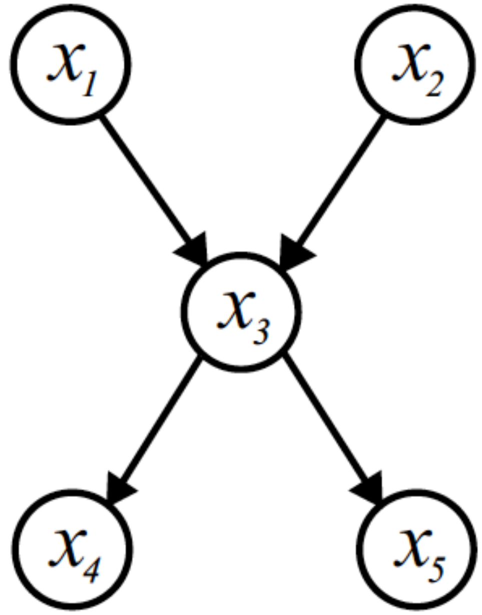
  - Observation comes from exponential family:

$$p(y_i \mid x_i, w) = \exp\{y_i \theta_i - A(\theta_i)\}$$

- Special cases: linear regression and logistic regression
- ML and MAP estimation is generally straightforward
- Many possible extensions:
  - *Multivariate responses* with more parameters (biggest difficulty is notation and indexing)
  - *Link functions* to allow more flexibility in how $(w, x_i) \rightarrow \theta_i$

Directed Graphs & Exp. Families

$$p(x) = \prod_{i=1}^{N} p(x_i \mid x_{\Gamma(i)}, \theta_i)$$



$$p(x_i \mid x_{\Gamma(i)}, \theta_i) = \exp\left\{ x_i \theta_i^T \phi(x_{\Gamma(i)}) - A(\theta_i^T \phi(x_{\Gamma(i)})) \right\}$$

- For each node, pick an appropriate exponential family
- Pick features of parent nodes relevant to child variable
  Most generally, indicators for all joint configurations of parents.
- Child parameters are a (learned) linear func. of parent features
- *Result:* Node-specific generalized linear models

Inference versus Learning

- **Inference:** Given a model with known parameters, estimate or find marginals of "hidden" variables for some data instance
- **Learning:** Given multiple data instances, find (often ML/MAP) estimates of parameters for a graphical model of their structure
  - Training instances may be *completely* or *partially* observed

Example: Expert systems for medical diagnosis

- **Inference:** Given observed symptoms for a particular patient, infer probabilities that they have contracted various diseases
- **Learning:** Given a database of many patient diagnoses, learn the relationships between diseases and symptoms

Example: Markov random fields for semantic image segmentation

- **Inference:** What object category is depicted at each pixel?
- **Learning:** How do objects relate to low-level image features?

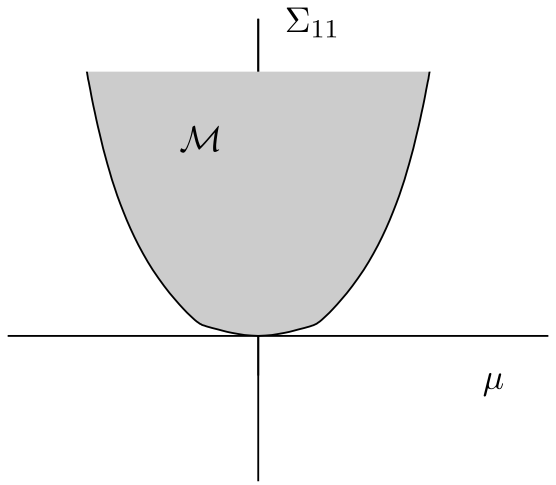
Mean Parameter Spaces

$$p(x \mid \theta) = \exp\{\theta^T \phi(x) - A(\theta)\}$$

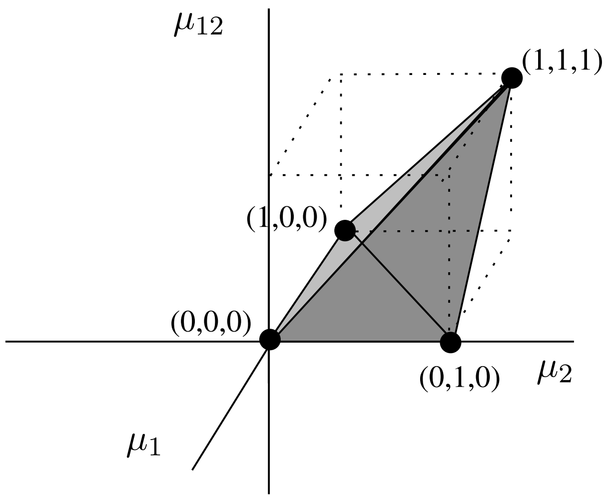
$$\mu_a = \mathbb{E}_p[\phi_a(x)] = \int \phi_a(x) p(x) \, dx$$

$$\mathcal{M} \triangleq \{\mu \in \mathbb{R}^d \mid \exists \, p \text{ such that } \mathbb{E}_p[\phi(x)] = \mu\}$$

- For a given collection of sufficient statistics, what is the set of all *realizable* mean parameters?



Scalar Gaussian



Pair of Binary Variables

- The set of realizable parameters is always *convex*. Why?

Preview: Inference and Learning

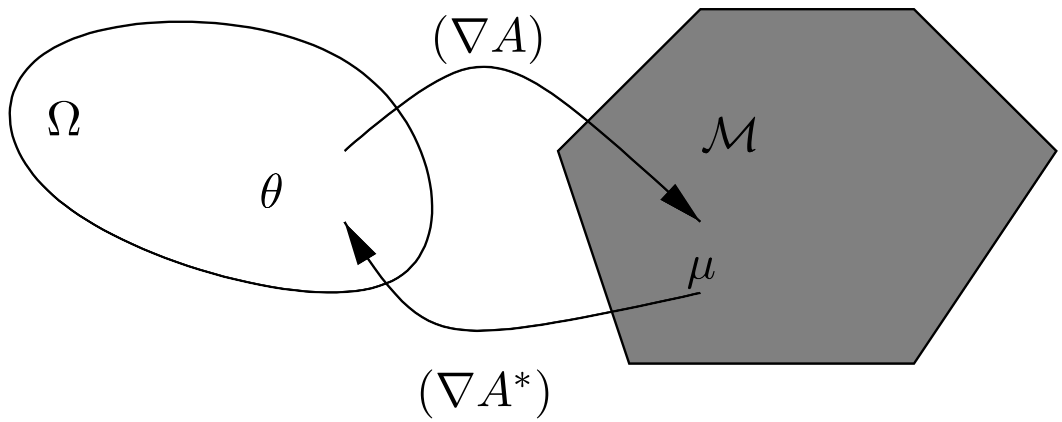
$$p(x \mid \theta) = \exp\{\theta^T \phi(x) - A(\theta)\}$$

$$A(\theta) = \log \int_{\mathcal{X}} \exp\{\theta^T \phi(x)\}\, dx$$

$$\Omega = \{\theta \in \mathbb{R}^d \mid A(\theta) < +\infty\}$$

$$\mu_a = \mathbb{E}_p[\phi_a(x)] = \int \phi_a(x) p(x)\, dx$$

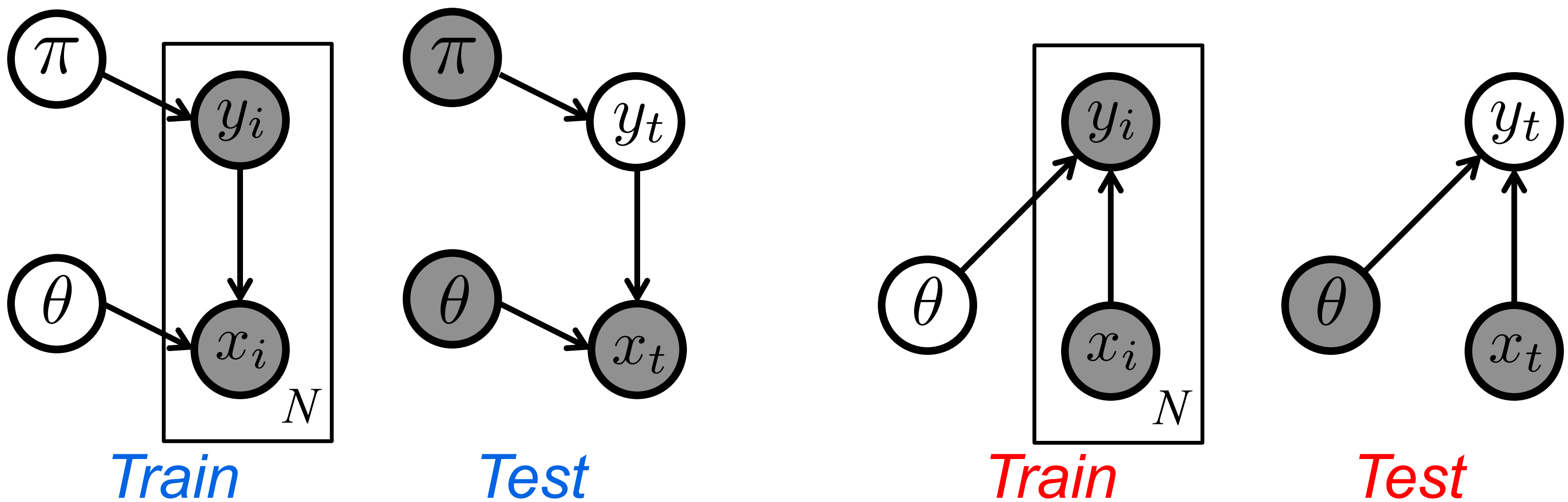
$$\mathcal{M} \triangleq \{\mu \in \mathbb{R}^d \mid \exists\ p \text{ such that } \mathbb{E}_p[\phi(x)] = \mu\}$$



Supervised Learning

Generative ML or MAP Learning:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \left[\log p(y_i \mid \pi) + \log p(x_i \mid y_i, \theta)\right]$$



Discriminative ML or MAP Learning:

$$\max_{\theta} \; \log p(\theta) + \sum_{i=1}^{N} \log p(y_i \mid x_i, \theta)$$

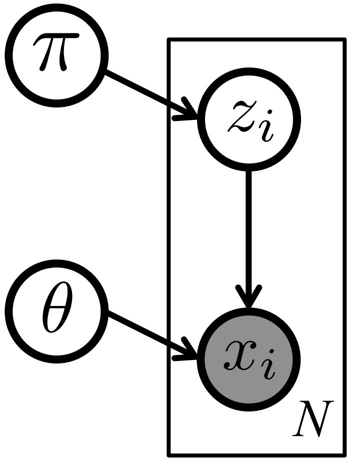
Unsupervised Learning

Clustering:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\sum_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \right]$$

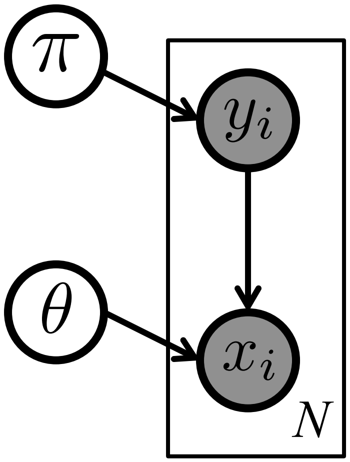
Dimensionality Reduction:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\int_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \; dz_i \right]$$

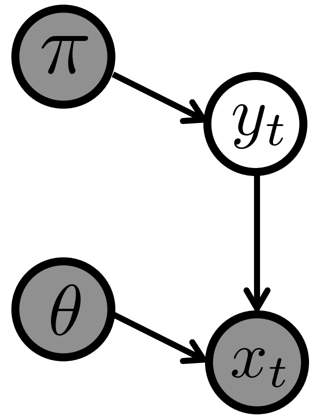


- No notion of training and test data: labels are *never* observed
- As before, *maximize* posterior probability of model parameters
- For hidden variables associated with each observation, we *marginalize* over possible values rather than estimating
  - Fully accounts for uncertainty in these variables
  - There is one hidden variable per observation, so cannot perfectly estimate even with infinite data
- Must use generative model (discriminative degenerates)

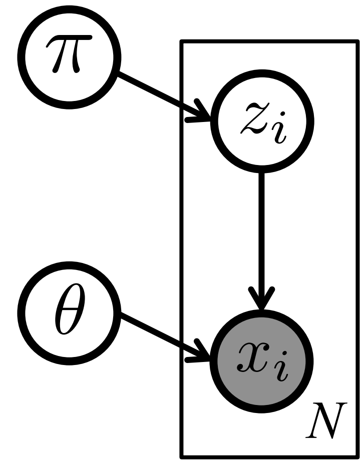
Unsupervised Learning Algorithms



Supervised Training



Supervised Testing



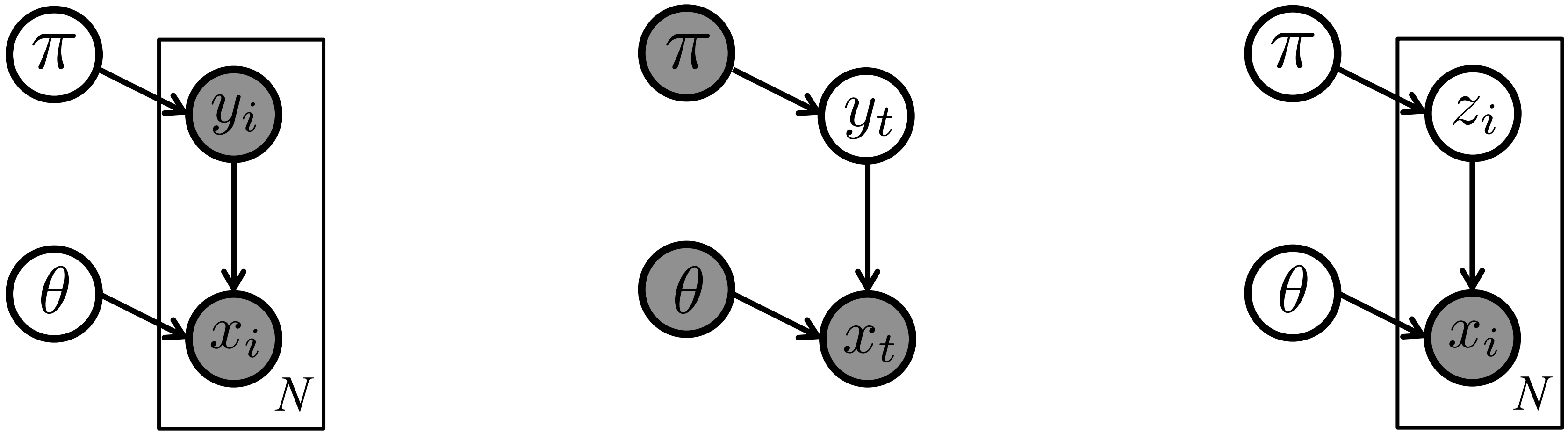
Unsupervised Learning

π, θ ⟶ *parameters (shared across instances)*

$z_1, \ldots, z_N$ ⟶ *hidden data (unique to particular instances)*

- **Initialization:** Randomly select starting parameters
- **Estimation:** Given parameters, infer likely hidden data
  - Similar to *testing* phase of supervised learning
- **Learning:** Given hidden & observed data, find likely parameters
  - Similar to *training* phase of supervised learning
- **Iteration:** Alternate estimation & learning until convergence

Expectation Maximization (EM)



Supervised Training | *Supervised Testing* | *Unsupervised Learning*

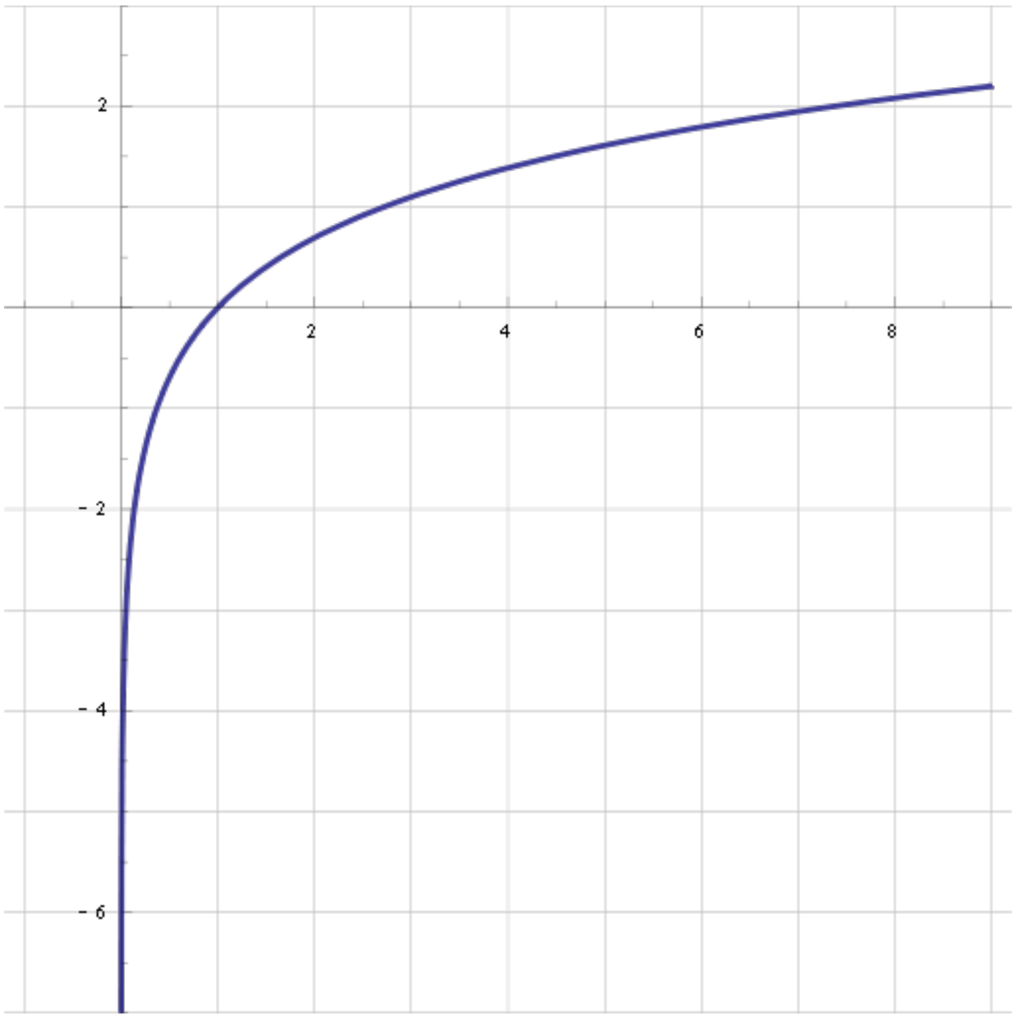
π, θ ⟶ *parameters (shared across observations)*

$z_1, \ldots, z_N$ ⟶ *hidden data (unique to particular instances)*

- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden data
  - Equivalent to test inference of full posterior distribution
- **M-Step:** Given posterior distributions, find likely parameters
  - Distinct from supervised ML/MAP, but often still tractable
- **Iteration:** Alternate E-step & M-step until convergence

Concavity & Jensen's Inequality

$$\ln(\mathbb{E}[X]) \geq \mathbb{E}[\ln(X)]$$



EM as Lower Bound Maximization

$$\ln p(x \mid \theta) = \ln \left(\sum_z p(x, z \mid \theta) \right)$$

$$\ln p(x \mid \theta) \geq \sum_z q(z) \ln \left(\frac{p(x, z \mid \theta)}{q(z)} \right)$$

$$\ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta)$$

- **Initialization:** Randomly select starting parameters $\theta^{(0)}$
- **E-Step:** Given parameters, find posterior of hidden data

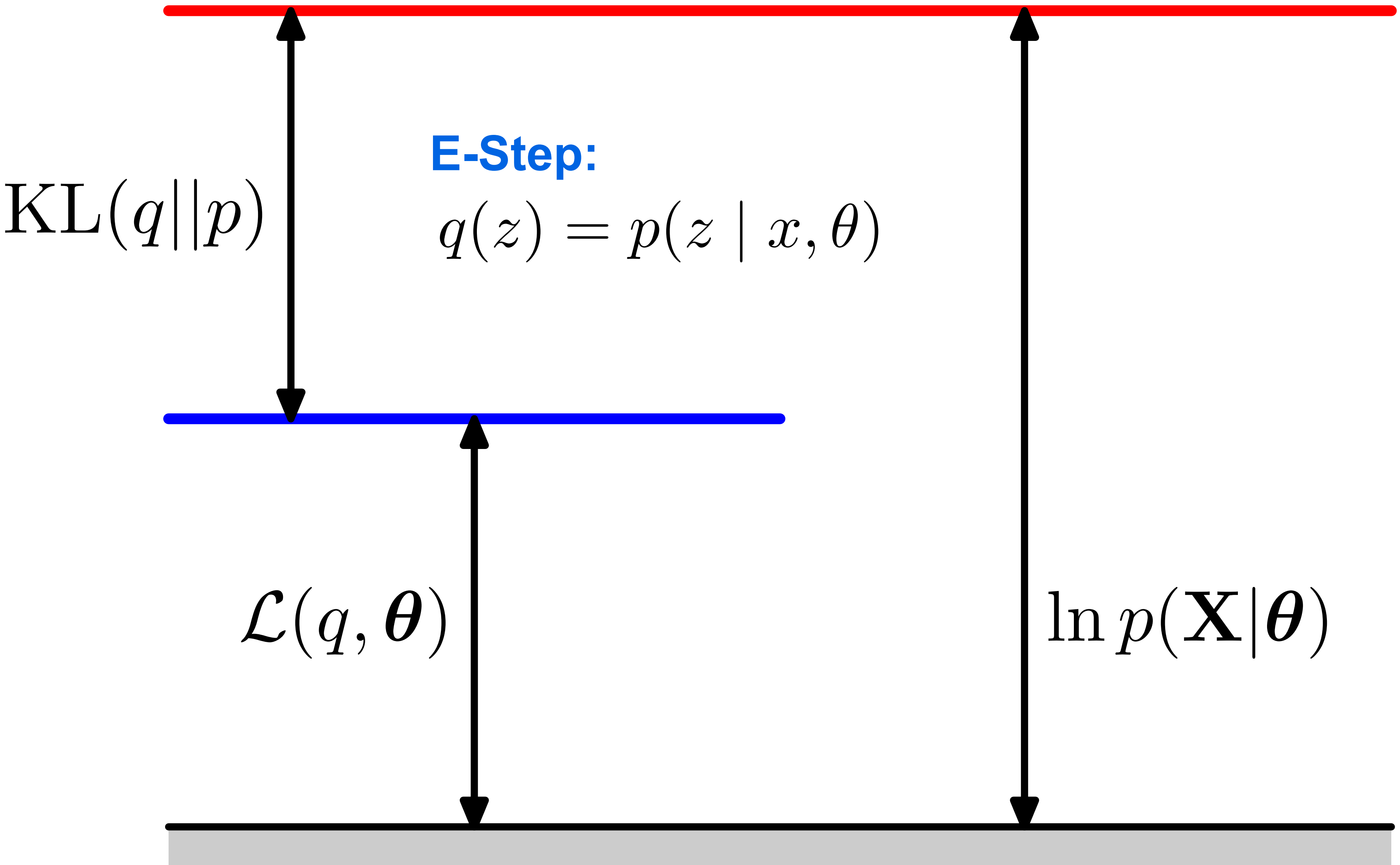
$$q^{(t)} = \arg\max_q \mathcal{L}(q, \theta^{(t-1)})$$

- **M-Step:** Given posterior distributions, find likely parameters

$$\theta^{(t)} = \arg\max_\theta \mathcal{L}(q^{(t)}, \theta)$$

- **Iteration:** Alternate E-step & M-step until convergence

Lower Bounds on Marginal Likelihood



C. Bishop, Pattern Recognition & Machine Learning

EM: Expectation Step

$$\ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta)$$

$$q^{(t)} = \arg\max_q \mathcal{L}(q, \theta^{(t-1)})$$

- General solution, for any probabilistic model:

$$q^{(t)}(z) = p(z \mid x, \theta^{(t-1)})$$

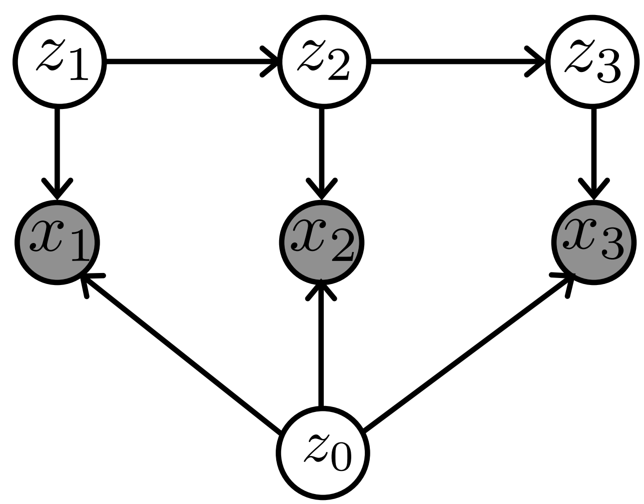
posterior distribution given current parameters

- For a directed graphical model:

$\theta \longrightarrow$ *fixes conditional distributions of every child node, given parents*

$x \longrightarrow$ *observed nodes (training data)*

$z \longrightarrow$ *unobserved nodes (hidden data)*



Inference: Find summary statistics of posterior needed for following M-step