
Intro to Machine Learning Review
May 11, 2012

Probabilistic Model

- $z_i \sim Multinomial(\pi)$
- $x_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i})$

Log Likelihood

$$\mathcal{L}(x_1, ..., x_n) = \log p(\pi) + \sum_{k=1}^{K} \log p(\mu_k) + \sum_{k=1}^{K} \log p(\Sigma_k)$$
$$+ \sum_{i=1}^{N} \log \sum_{k=1}^{K} p(x_i|z_k)p(z_k|\pi)$$

The Problem with Gaussian Mixture Models

Even with a simplified Probabilistic Model ...

- $\mu \sim \mathcal{N}(0, 1)$
- $z_i \sim \mathbb{U}$
- $x_i \sim \mathcal{N}(\mu_{z_i}, I)$

Log Likelihood FAIL

$$\mathcal{L}(x_1, ..., x_n) = C - \frac{1}{2}\sum_{k=1}^{K}\mu_k^2 + \sum_{i=1}^{N}\log\sum_{k=1}^{K}e^{-\frac{(x_i-\mu_k)^2}{2}}$$

$$\frac{\partial\mathcal{L}(x_1, ..., x_n)}{\partial\mu_k} = -\mu_k + \sum_{i=1}^{N}\frac{(x_i-\mu_k)e^{-\frac{(x_i-\mu_k)^2}{2}}}{\sum_{k=1}^{K}e^{-\frac{(x_i-\mu_k)^2}{2}}}$$

The Steps

- E - find $q(z)$ that is closest to $p(z|x,\theta)$
- M - maximize $\mathbb{E}_q[\mathcal{L}]$

Easy Math Level : 99

$$r_{ik} = q(z_i = k)$$

$$\mathbb{E}_q\left[\mathcal{L}(x_1^n, z_1^n)\right] = C - \frac{1}{2}\sum_{k=1}^{K}\mu_k^2 - \frac{1}{2}\sum_{i=1}^{N}\mathbb{E}_q\left[\sum_{k=1}^{K}\delta(z_i,k)(x_i-\mu_k)^2\right]$$

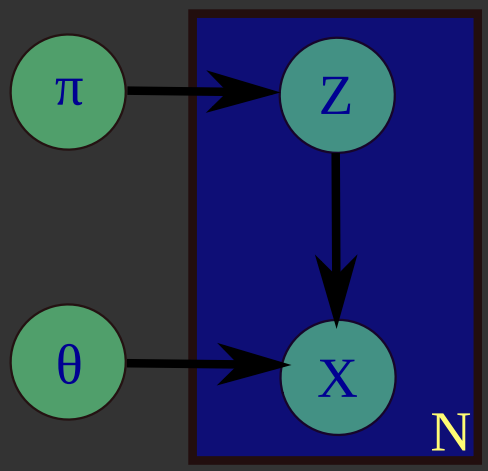
$$= C - \frac{1}{2}\sum_{k=1}^{K}\mu_k^2 - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}(x_i-\mu_k)^2$$

$$\frac{\partial \mathbb{E}_q[\mathcal{L}(x_1^n, z_1^n)]}{\partial \mu_k} = -\mu_k - \sum_{i=1}^{N} r_{ik}(\mu_k - x_i)$$

EM Mathematically

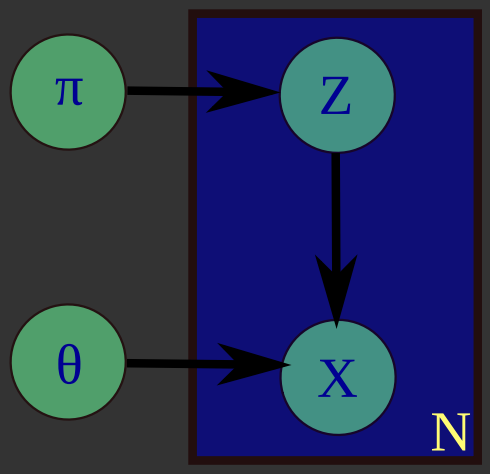
$$
\begin{aligned}
\log p(x|\theta) &= \log \sum_{z \in Z} \frac{q(z)}{q(z)} p(z|\theta) p(x|z,\theta) \\
&= \log \mathbb{E}_q \left[\frac{p(z|\theta) p(x|z,\theta)}{q(z)} \right] \\
&\geq \mathbb{E}_q \left[\log \frac{p(z|\theta) p(x|z,\theta)}{q(z)} \right] \\
&= \mathbb{E}_q [\log p(z|\theta) p(x|z,\theta)] - \mathbb{E}_q [\log q(z)]
\end{aligned}
$$

Graphically



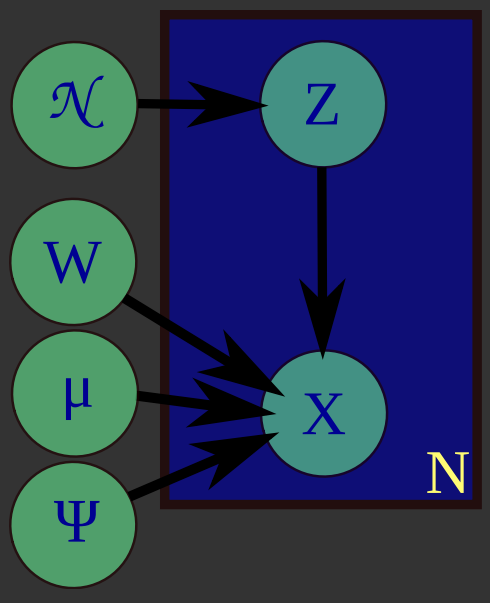
- θ, Z not independent given X
- If we knew Z, it would be easy
- K means - pick the best z_i
- EM - use $p(z|x, \theta)$ to soft assign
- MCMC - sample from $p(z|x, \theta)$

Sampling



- Sample variables given their Markov Blanket - parents, children, and co-parents (nodes that share children)
- Must sample everything (z,θ,π)
- We need prior β on θ and α on π
- Gibbs sampling
  $p(z|\theta, \pi, \mathbf{x})$
  $p(\theta|\theta^{-}, \mathbf{z}, \mathbf{x}, \beta)$
  $p(\pi|\mathbf{z}, \alpha)$

Factor Analysis



- $z_i \sim \mathcal{N}(0, \mathcal{I})$
- $x_i \sim \mathcal{N}(Wz_i + \mu, \Psi)$

$$\begin{pmatrix} z_i \\ x_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \mu \end{pmatrix}, \begin{pmatrix} \mathcal{I} & W^T \\ W & WW^T + \Psi \end{pmatrix}\right)$$

- $p(x_i) = \mathcal{N}(\mu, WW^T + \Psi)$, so Ψ must be diagonal
- $\Psi = \alpha\mathcal{I} \rightarrow$ Probabilistic PCA

Facts

- Both FA and PPCA are unhindered by data translation
- FA is immune to scalings (important for uncomparable units)
- PPCA is immune to data rotations
- PPCA finds the same solution as PCA (using matrix SVD)

Definition (Multinomial Emissions)

- S states, and one extra boundary state $*$
- V words in vocuabulary
- a multinomial of size V for each $i \in S$ defining emission $P(i \Rightarrow w)$ for $w \in V$
- a matrix of size $S + 1 x S + 1$ for each pair $i, j \in (S \cup *)^2$ defining transition $P(i \rightarrow j)$

Hidden Markov Models - Sum Product

Forward Probability α_i^t

The summed probability of all paths that pass through state i at time t, until and incuding the decision to emit x_t

$$\alpha_j^{t+1} = \left(\sum_{i \in S} \alpha_i^t P(i \to j) \right) P(j \Rightarrow x_t)$$

Backward Probability β_i^t

The summed probability of all paths that pass through state i at time t, after the decision to emit x_t

$$\beta_i^{t-1} = \sum_{j \in S} P(i \to j) P(j \Rightarrow x_t) \beta_j^t$$

Hidden Markov Models - Updates

Transitions

Collect expected counts ($\mathbb{EC}$) of transition events
to get $\mathbb{EC}[i \to j]$ add up $\alpha_j^t P(i \to j) P(j \Rightarrow x_{t+1}) \beta_j^{t+1}$ for every possible transition from i at time t to j at time $t+1$, normalized by that sequence's total probability

$$\hat{P}(i \to j) = \frac{\mathbb{EC}[i \to j]}{\mathbb{EC}[i \to \bullet]}$$

Emissions (Multinomial)

Collect expected counts ($\mathbb{EC}$) of emission events
to get $\mathbb{EC}[i \Rightarrow w]$ add up $\alpha_i^t \beta_i^t$ normalized by that sequence's total probability for every instance of $x_t = w$

$$\hat{P}(i \Rightarrow w) = \frac{\mathbb{EC}[i \Rightarrow w]}{\mathbb{EC}[i \Rightarrow \bullet]}$$

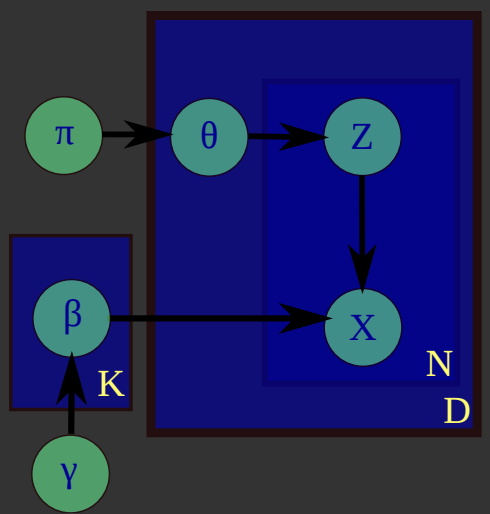
Viterbi

- Use a dynamic programming table T that stores pointers to cells in the previous column and probabilities.
- Initialize row i in the first column with (NULL,$P(* \rightarrow i)$)
- Compute column $k+1$ by filling cell j with ($argmax_i f(i)$,$f(i)$) where $f(k) = T(k,i)\,(P(i \rightarrow j)P(j \Rightarrow x_t))$

Max Prob Tag

- Compute Forward and Backward Probabilities
- For each time t choose $argmax_i(\alpha_i^t \beta_i^t)$
- Corresponds better to many NLP evaluation metrics

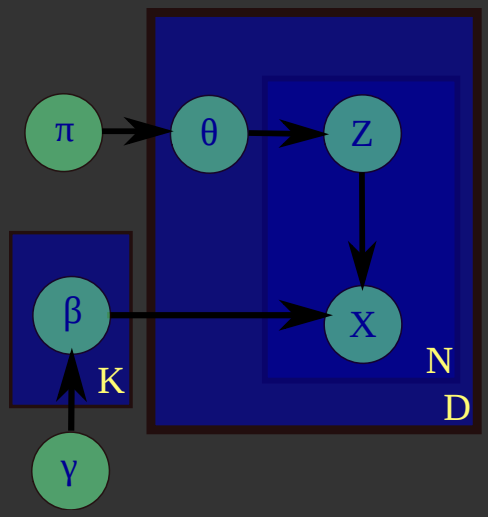
Latent Dirichlet Allocation



- $\theta_j \sim Dirichlet(\pi)$
- $z_{ij} \sim Multinomial(\theta_j)$
- $\beta_k \sim Dirichlet(\gamma)$
- $x_{ij} \sim Multinomial(\beta_{z_{ij}})$

Avoid point estimates of θ and β by integrating over all possible values

Latent Dirichlet Allocation - EM doesn't work

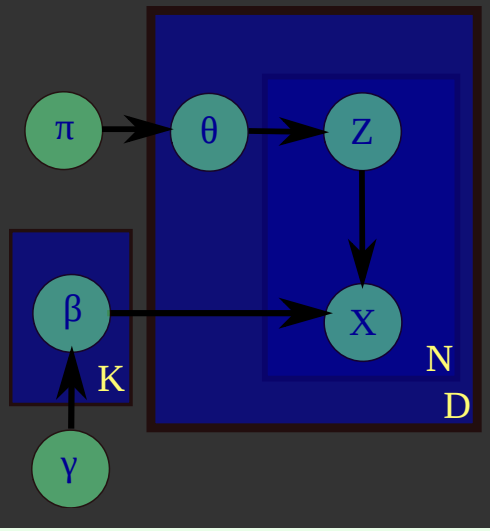


- $\theta_j \sim Dirichlet(\pi)$
- $z_{ij} \sim Multinomial(\theta_j)$
- $\beta_k \sim Dirichlet(\gamma)$
- $x_{ij} \sim Multinomial(\beta_{z_{ij}})$

This integral is intractable, and necessary for $p(\theta_i, z_i | x_i, \pi, \gamma)$

$$p(x_i|\gamma,\pi) = \int Dir(\theta_i) \prod_{n=1}^{N} \sum_{k=1}^{K} \prod_{j=1}^{V} (\theta_{ik}\beta_{ij})^{x_i^n} \partial\theta_i$$

Latent Dirichlet Allocation - Gibbs Sampling



- $\theta_j \sim Dirichlet(\pi)$
- $z_{ij} \sim Multinomial(\theta_j)$
- $\beta_k \sim Dirichlet(\gamma)$
- $x_{ij} \sim Multinomial(\beta_{z_{ij}})$

- Sample from $p(z_{ij}|\mathbf{z}^{-}, \gamma, \pi)$ by integrating out θ and β