CSci 8980: Advanced Topics in Graphical Models

Infinite Mixture Models, Indian Buffet Process

Instructor: Arindam Banerjee

October 30, 2007

Finite Mixture Models

- Prior of cluster assignment is independent

$$P(\mathbf{c}|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{N} \theta_{c_i}$$

Finite Mixture Models

- Prior of cluster assignment is independent

$$P(\mathbf{c}|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{N} \theta_{c_i}$$

- The mixture model is given by

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(x_i|c_i = k)\theta_k$$

Finite Mixture Models

- Prior of cluster assignment is independent

$$P(\mathbf{c}|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{N} \theta_{c_i}$$

- The mixture model is given by

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(x_i|c_i = k)\theta_k$$

- Define a (symmetric) Dirichlet prior over θ

$$D\left(\frac{\alpha}{K}, \cdots, \frac{\alpha}{K}\right) = \frac{\Gamma(\frac{\alpha}{K})^K}{\Gamma(\alpha)}$$

Finite Mixture Models

- Prior of cluster assignment is independent

$$P(\mathbf{c}|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{N} \theta_{c_i}$$

- The mixture model is given by

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(x_i|c_i = k)\theta_k$$

- Define a (symmetric) Dirichlet prior over θ

$$D\left(\frac{\alpha}{K}, \cdots, \frac{\alpha}{K}\right) = \frac{\Gamma(\frac{\alpha}{K})^K}{\Gamma(\alpha)}$$

- The prior model

$$\begin{aligned} \theta|\alpha &\sim \text{Dirichlet}\left(\frac{\alpha}{K}, \cdots, \frac{\alpha}{K}\right) \\ c_i|\theta &\sim \text{Discrete}(\theta) \end{aligned}$$

Finite Mixture Models (Contd.)

- The marginal probability of assignment vector **c**

$$
\begin{aligned}
P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{N} P(c_i|\theta)p(\theta)d\theta \\
&= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}
\end{aligned}
$$

Finite Mixture Models (Contd.)

- The marginal probability of assignment vector **c**

$$
\begin{aligned}
P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{N} P(c_i|\theta)p(\theta)d\theta \\
&= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}
\end{aligned}
$$

- Note that $m_k = \sum_{i=1}^{N} \delta(c_i = k)$

Finite Mixture Models (Contd.)

- The marginal probability of assignment vector **c**

$$
\begin{aligned}
P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{N} P(c_i|\theta)p(\theta)d\theta \\
&= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}
\end{aligned}
$$

- Note that $m_k = \sum_{i=1}^{N} \delta(c_i = k)$
- Individual assignments are exchangeable, not independent

Finite Mixture Models (Contd.)

- The marginal probability of assignment vector **c**

$$
\begin{aligned}
P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{N} P(c_i|\theta)p(\theta)d\theta \\
&= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}
\end{aligned}
$$

- Note that $m_k = \sum_{i=1}^{N} \delta(c_i = k)$
- Individual assignments are exchangeable, not independent
- Distribution is over a partitioning

Finite Mixture Models (Contd.)

- The marginal probability of assignment vector **c**

$$
\begin{aligned}
P(\mathbf{c}) &= \int_{\Delta_K} \prod_{i=1}^{N} P(c_i|\theta)p(\theta)d\theta \\
&= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}
\end{aligned}
$$

- Note that $m_k = \sum_{i=1}^{N} \delta(c_i = k)$
- Individual assignments are exchangeable, not independent
- Distribution is over a partitioning
- Have to assume K to be the maximum number of partitions

Infinite Mixture Models

- Assume infinitely many classes

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(x_i|c_i = k)\theta_k$$

Infinite Mixture Models

- Assume infinitely many classes

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(x_i|c_i = k)\theta_k$$

- One approach is to use a Dirichlet Process to get $P(\mathbf{c})$

Infinite Mixture Models

- Assume infinitely many classes

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(x_i|c_i = k)\theta_k$$

- One approach is to use a Dirichlet Process to get $P(\mathbf{c})$
- Alternatively, one can compute $\lim_{K\to\infty} P(\mathbf{c})$

Infinite Mixture Models

- Assume infinitely many classes

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(x_i|c_i = k)\theta_k$$

- One approach is to use a Dirichlet Process to get $P(\mathbf{c})$
- Alternatively, one can compute $\lim_{K\to\infty} P(\mathbf{c})$
- Let K_+ be the number of classes with $m_k > 0$, $K = K_+ + K_0$

Infinite Mixture Models

- Assume infinitely many classes

$$P(X|\theta) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(x_i|c_i = k)\theta_k$$

- One approach is to use a Dirichlet Process to get $P(\mathbf{c})$
- Alternatively, one can compute $\lim_{K\to\infty} P(\mathbf{c})$
- Let K_+ be the number of classes with $m_k > 0$, $K = K_+ + K_0$
- Using $\Gamma(x) = (x-1)\Gamma(x-1)$, we have

$$P(\mathbf{c}) = \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{j=1}^{m_k - 1} \left(j + \frac{\alpha}{K}\right) \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}$$

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent
  - Induce the same partitioning, the label values do not matter

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent
  - Induce the same partitioning, the label values do not matter
  - Denote the partitioning induced by $\mathbf{c}$ as $[\mathbf{c}]$

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent
  - Induce the same partitioning, the label values do not matter
  - Denote the partitioning induced by $\mathbf{c}$ as $[\mathbf{c}]$
- With $K = K_+ + K_0$ classes, $[\mathbf{c}]$ has $K!/K_0!$ assignment vectors

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent
  - Induce the same partitioning, the label values do not matter
  - Denote the partitioning induced by $\mathbf{c}$ as $[\mathbf{c}]$
- With $K = K_+ + K_0$ classes, $[\mathbf{c}]$ has $K!/K_0!$ assignment vectors
- The probability of each assignment vector is the same, so

$$P([\mathbf{c}]) = \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{j=1}^{m_k - 1} \left(j + \frac{\alpha}{K}\right) \right) \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)}$$

Infinite Mixture Models (Contd.)

- As $K \to \infty$, $P(\mathbf{c}) \to 0$ for any particular $\mathbf{c}$
- However, $K_+ \leq N$, hence finitely many equivalence classes
  - Assignments $\{1, 1, 2\}$ and $\{2, 2, 1\}$ are equivalent
  - Induce the same partitioning, the label values do not matter
  - Denote the partitioning induced by $\mathbf{c}$ as $[\mathbf{c}]$
- With $K = K_+ + K_0$ classes, $[\mathbf{c}]$ has $K!/K_0!$ assignment vectors
- The probability of each assignment vector is the same, so

$$P([\mathbf{c}]) = \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K^+} \prod_{j=1}^{m_k - 1} \left(j + \frac{\alpha}{K} \right) \right) \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)}$$

- Taking limits as $K \to \infty$, we have

$$\lim_{K \to \infty} P([\mathbf{c}]) = \alpha^{K_+} \left(\prod_{k=1} (m_k - 1)! \right) \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)}$$

Chinese Restaurant Process

- CRP gives a prior over partitions

$$P(c_i = k | c_1, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{otherwise} \end{cases}$$

Chinese Restaurant Process

- CRP gives a prior over partitions

$$P(c_i = k | c_1, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{otherwise} \end{cases}$$

- With N objects, the probability of a particular partition $[\mathbf{c}]$ is

$$\alpha^{K_+} \left(\prod_{k=1} (m_k - 1)! \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}$$

Chinese Restaurant Process

- CRP gives a prior over partitions

$$P(c_i = k | c_1, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{otherwise} \end{cases}$$

- With N objects, the probability of a particular partition $[\mathbf{c}]$ is

$$\alpha^{K_+} \left(\prod_{k=1} (m_k - 1)! \right) \frac{\Gamma(\alpha)}{\Gamma(N + \alpha)}$$

- Intuitive means of specifying a prior for infinite mixture models

Chinese Restaurant Process

- CRP gives a prior over partitions

$$P(c_i = k | c_1, \dots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & \text{otherwise} \end{cases}$$

- With N objects, the probability of a particular partition $[\mathbf{c}]$ is

$$\alpha^{K_+} \left(\prod_{k=1} (m_k - 1)! \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}$$

- Intuitive means of specifying a prior for infinite mixture models
- Sequential process to generate exchangeable class assignments

Latent Feature Models

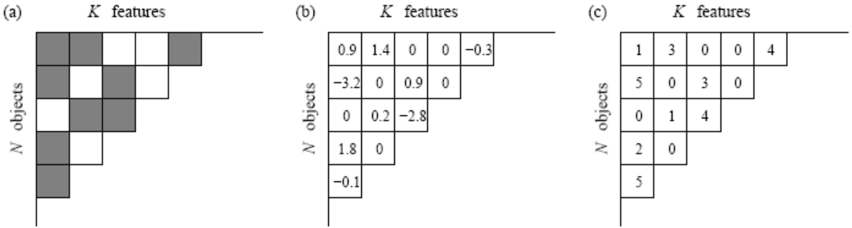


Figure 3: Feature matrices. A binary matrix Z, as shown in (a), can be used as the basis for sparse infinite latent feature models, indicating which features take non-zero values. Elementwise multiplication of Z by a matrix V of continuous values gives a representation like that shown in (b). If V contains discrete values, we obtain a representation like that shown in (c).

Latent Feature Models (Contd.)

- A latent feature has two components

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features
  - A distribution $P(X|F)$ relating observations and features

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features
  - A distribution $P(X|F)$ relating observations and features
- Consider $F = Z \otimes V$ with $P(F) = P(Z)P(V)$ where

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features
  - A distribution $P(X|F)$ relating observations and features
- Consider $F = Z \otimes V$ with $P(F) = P(Z)P(V)$ where
  - Z is a binary matrix, indicating which features are on

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features
  - A distribution $P(X|F)$ relating observations and features
- Consider $F = Z \otimes V$ with $P(F) = P(Z)P(V)$ where
  - Z is a binary matrix, indicating which features are on
  - V is a matrix containing feature values

Latent Feature Models (Contd.)

- A latent feature has two components
  - A distribution $P(F)$ over features
  - A distribution $P(X|F)$ relating observations and features
- Consider $F = Z \otimes V$ with $P(F) = P(Z)P(V)$ where
  - Z is a binary matrix, indicating which features are on
  - V is a matrix containing feature values
- Z determines the effective dimensionality of the model

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$
- An object contains feature k with Bernoulli probability π_k

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$
- An object contains feature k with Bernoulli probability π_k
- The probability of a binary matrix Z

$$P(Z|\pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1 - \pi_k)^{N - m_k}$$

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$
- An object contains feature k with Bernoulli probability π_k
- The probability of a binary matrix Z

$$P(Z|\pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k}(1-\pi_k)^{N-m_k}$$

- Define a Beta prior $B(r,s)$ over π_k

$$p(\pi_k) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \pi_k^{r-1}(1-\pi_k)^{s-1}$$

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$
- An object contains feature k with Bernoulli probability π_k
- The probability of a binary matrix Z

$$P(Z|\pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1-\pi_k)^{N-m_k}$$

- Define a Beta prior $B(r, s)$ over π_k

$$p(\pi_k) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \pi_k^{r-1} (1-\pi_k)^{s-1}$$

- With $r = \alpha/K, s = 1$, we have $p(\pi_k) = \alpha/K \pi_k^{\alpha/K-1}$

Finite Feature Models

- Consider N objects and K features, Z is $N \times K$
- An object contains feature k with Bernoulli probability π_k
- The probability of a binary matrix Z

$$P(Z|\pi) = \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1 - \pi_k)^{N - m_k}$$

- Define a Beta prior $B(r, s)$ over π_k

$$p(\pi_k) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} \pi_k^{r-1} (1 - \pi_k)^{s-1}$$

- With $r = \alpha/K, s = 1$, we have $p(\pi_k) = \alpha/K \pi_k^{\alpha/K - 1}$
- Generative model

$$\begin{aligned} \pi_k|\alpha &\sim \text{Beta}(\alpha/K, 1) \\ z_{ik}|\pi_k &\sim \text{Bernoulli}(\pi_k) \end{aligned}$$

Finite Feature Models (Contd.)

- The marginal distribution of Z

$$\begin{aligned} P(Z) &= \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik}|\pi_k) \right) p(\pi_k) d\pi_k \\ &= \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \end{aligned}$$

Finite Feature Models (Contd.)

- The marginal distribution of Z

$$P(Z) = \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik}|\pi_k) \right) p(\pi_k) d\pi_k$$

$$= \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

- The expected number of non-zeroes is bounded for any K

Finite Feature Models (Contd.)

- The marginal distribution of Z

$$\begin{aligned} P(Z) &= \prod_{k=1}^{K} \int \left(\prod_{i=1}^{N} P(z_{ik}|\pi_k) \right) p(\pi_k) d\pi_k \\ &= \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \end{aligned}$$

- The expected number of non-zeroes is bounded for any K
- Since each column is independent

$$E[1^T Z1] = KE[1^T z_k] = K \sum_{i=1}^{N} E(z_{ik}) = KN \frac{\alpha/K}{1 + \alpha/K} \leq N\alpha$$

Equivalence Classes

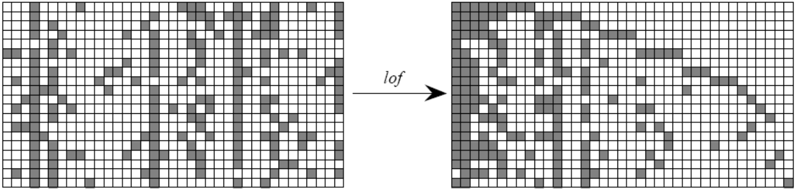


Figure 4: Binary matrices and the left-ordered form. The binary matrix on the left is transformed into the left-ordered binary matrix on the right by the function $lof(\cdot)$. This left-ordered matrix was generated from the exchangeable Indian buffet process with $\alpha = 10$. Empty columns are omitted from both matrices.

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$
  - History h is the decimal equivalent of the column z_k

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$
  - History h is the decimal equivalent of the column z_k
  - K_h denote the number of features having *history h*

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$
  - History h is the decimal equivalent of the column z_k
  - K_h denote the number of features having *history h*
  - K_0 denote the number of features having $m_k = 0$

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$
  - History h is the decimal equivalent of the column z_k
  - K_h denote the number of features having *history* h
  - K_0 denote the number of features having $m_k = 0$
  - Then, $K_+ = \sum +h = 1^{2^N-1} K_h$ and $K = K_+ + K_0$

Equivalence Classes (Contd.)

- Left-ordering defines an equivalence class $[Z]$
  - Two matrices are equivalent if $lof(Z) = lof(Y)$
  - Inference w.r.t. *lof* is appropriate for models unaffected by feature ordering
  - All linear models belong to this category
- How to compute cardinality of $[Z]$
  - History h is the decimal equivalent of the column z_k
  - K_h denote the number of features having *history h*
  - K_0 denote the number of features having $m_k = 0$
  - Then, $K_+ = \sum + h = 1^{2^N - 1} K_h$ and $K = K_+ + K_0$
- Then, the cardinality of $[Z]$ is

$$\binom{K}{K_0 \; \ldots \; K_{2^N - 1}} = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!}$$

Infinite Feature Models

- The marginal probability of an equivalence class

$$P([Z]) = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

Infinite Feature Models

- The marginal probability of an equivalence class

$$P([Z]) = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

- Taking $K \to \infty$, with $H_N = \sum_{j=1}^{N} 1/j$, we get

$$\lim_{K\to\infty} P([Z]) = \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha H_N) \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}$$

Infinite Feature Models

- The marginal probability of an equivalence class

$$P([Z]) = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

- Taking $K \to \infty$, with $H_N = \sum_{j=1}^{N} 1/j$, we get

$$\lim_{K\to\infty} P([Z]) = \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha H_N) \prod_{k=1}^{K_+} \frac{(N-m_k)!(m_k-1)!}{N!}$$

- Exchangeable distribution, only depending on m_k and K_h

Infinite Feature Models

- The marginal probability of an equivalence class

$$P([Z]) = \frac{K!}{\prod_{h=0}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}$$

- Taking $K \to \infty$, with $H_N = \sum_{j=1}^{N} 1/j$, we get

$$\lim_{K\to\infty} P([Z]) = \frac{\alpha^{K_+}}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha H_N) \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}$$

- Exchangeable distribution, only depending on m_k and K_h
- The probability does not change by re-ordering objects

Indian Buffet Process

- Consider Indian restaurant with infinite dishes

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The $i^{th} customer$ moves along the buffet

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The i^{th}*customer* moves along the buffet
    - Let m_k be the number of previous customers who tried disk k

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The $i^{th} customer$ moves along the buffet
    - Let m_k be the number of previous customers who tried disk k
    - Samples popular dishes with probability $\frac{m_k}{i}$

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The i^{th}*customer* moves along the buffet
    - Let m_k be the number of previous customers who tried disk k
    - Samples popular dishes with probability $\frac{m_k}{i}$
    - Samples Poisson($\frac{\alpha}{i}$) new dishes

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The i^{th}*customer* moves along the buffet
    - Let m_k be the number of previous customers who tried disk k
    - Samples popular dishes with probability $\frac{m_k}{i}$
    - Samples Poisson($\frac{\alpha}{i}$) new dishes
- The process generates a binary matrix sequentially

Indian Buffet Process

- Consider Indian restaurant with infinite dishes
- Each customer chooses dishes following a sequential process
- The generative process
  - First customer takes the first Poisson(α) dishes
  - The i^{th}*customer* moves along the buffet
    - Let m_k be the number of previous customers who tried disk k
    - Samples popular dishes with probability $\frac{m_k}{i}$
    - Samples Poisson($\frac{\alpha}{i}$) new dishes
- The process generates a binary matrix sequentially
- The lof equivalence class has the distribution $P([Z])$

Indian Buffet Process (Contd.)

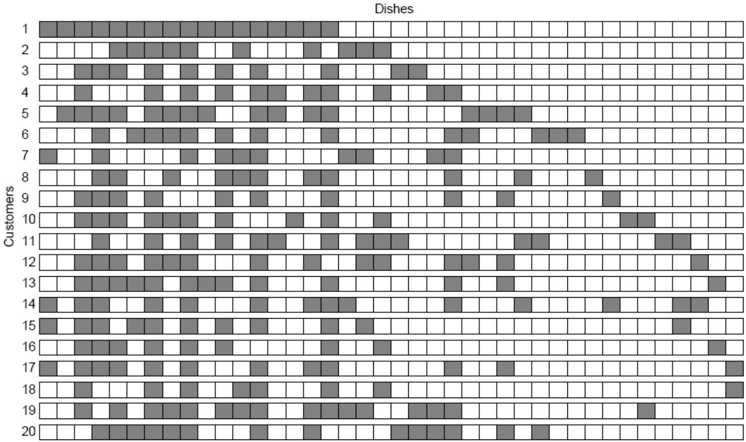


Figure 5: A binary matrix generated by the Indian buffet process with $\alpha = 10$.

Inference by Gibbs Sampling

- For a finite latent feature model, the full conditional

$$P(z_{ik} = 1 | Z_{-(i,k)}, X) \propto P(z_{ik} = 1 | Z_{-(i,k)}) P(X|Z)$$

Inference by Gibbs Sampling

- For a finite latent feature model, the full conditional

$$P(z_{ik} = 1 | Z_{-(i,k)}, X) \propto P(z_{ik} = 1 | Z_{-(i,k)}) P(X|Z)$$

- For the Beta-Bernoulli model

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \int_0^1 P(z_{ik} | \pi_k) P(\pi_k | z_{-i,k}) d\pi_k = \frac{m_{-i,k} + \alpha/K}{N + \alpha/K}$$

Inference by Gibbs Sampling

- For a finite latent feature model, the full conditional

$$P(z_{ik} = 1 | Z_{-(i,k)}, X) \propto P(z_{ik} = 1 | Z_{-(i,k)}) P(X|Z)$$

- For the Beta-Bernoulli model

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \int_0^1 P(z_{ik}|\pi_k) P(\pi_k | z_{-i,k}) d\pi_k = \frac{m_{-i,k} + \alpha/K}{N + \alpha/K}$$

- Only depends on the assignments for feature k, since columns are independent

Inference by Gibbs Sampling

- For a finite latent feature model, the full conditional

$$P(z_{ik} = 1 | Z_{-(i,k)}, X) \propto P(z_{ik} = 1 | Z_{-(i,k)}) P(X|Z)$$

- For the Beta-Bernoulli model

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \int_0^1 P(z_{ik}|\pi_k) P(\pi_k | z_{-i,k}) d\pi_k = \frac{m_{-i,k} + \alpha/K}{N + \alpha/K}$$

- Only depends on the assignments for feature k, since columns are independent
- For the infinite case, for $m_k > 0$

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \frac{m_{-i,k}}{N}$$

Inference by Gibbs Sampling

- For a finite latent feature model, the full conditional

$$P(z_{ik} = 1 | Z_{-(i,k)}, X) \propto P(z_{ik} = 1 | Z_{-(i,k)}) P(X|Z)$$

- For the Beta-Bernoulli model

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \int_0^1 P(z_{ik} | \pi_k) P(\pi_k | z_{-i,k}) d\pi_k = \frac{m_{-i,k} + \alpha/K}{N + \alpha/K}$$

- Only depends on the assignments for feature k, since columns are independent
- For the infinite case, for $m_k > 0$

$$P(z_{ik} = 1 | \mathbf{z}_{-i,k}) = \frac{m_{-i,k}}{N}$$

- New features should be drawn from Poisson$(\frac{\alpha}{N})$

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model
  - Gaussian distribution with mean $\mathbf{z}_i A$ and covariance $\Sigma_X = \sigma_X^2 I$

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model
  - Gaussian distribution with mean $\mathbf{z}_i A$ and covariance $\Sigma_X = \sigma_X^2 I$
  - $\mathbf{z}_i$ is a $1 \times K$ binary vector, A is $K \times D$ matrix

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model
  - Gaussian distribution with mean $\mathbf{z}_i A$ and covariance $\Sigma_X = \sigma_X^2 I$
  - $\mathbf{z}_i$ is a $1 \times K$ binary vector, A is $K \times D$ matrix
- In matrix notation $E[X] = ZA$, so that

$$P(X|Z, A, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{ND/2}} \exp\left\{-\frac{1}{2\sigma_X^2}\mathrm{tr}((X - ZA)^T(X - ZA))\right\}$$

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model
  - Gaussian distribution with mean $\mathbf{z}_i A$ and covariance $\Sigma_X = \sigma_X^2 I$
  - $\mathbf{z}_i$ is a $1 \times K$ binary vector, A is $K \times D$ matrix
- In matrix notation $E[X] = ZA$, so that

$$P(X|Z, A, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{ND/2}} \exp\left\{-\frac{1}{2\sigma_X^2}\mathrm{tr}((X - ZA)^T(X - ZA))\right\}$$

- Bayesian model with Gaussian prior over A

$$P(A|\sigma_A) = \frac{1}{(2\pi\sigma_A^2)^{KD/2}} \exp\left\{-\frac{1}{\sigma_A^2}\mathrm{tr}(A^T A)\right\}$$

Finite Linear Gaussian Model

- Observation $\mathbf{x}_i \in \mathbb{R}^d$ is generated from a latent model
  - Gaussian distribution with mean $\mathbf{z}_i A$ and covariance $\Sigma_X = \sigma_X^2 I$
  - $\mathbf{z}_i$ is a $1 \times K$ binary vector, A is $K \times D$ matrix
- In matrix notation $E[X] = ZA$, so that

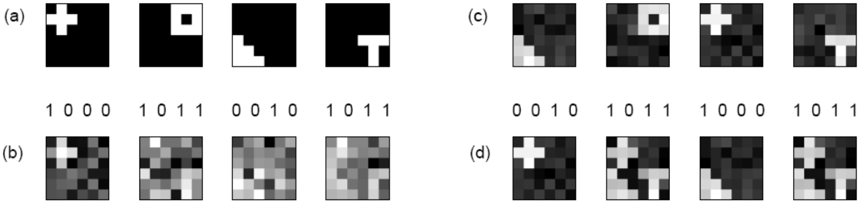
$$P(X|Z, A, \sigma_X) = \frac{1}{(2\pi\sigma_X^2)^{ND/2}} \exp\left\{-\frac{1}{2\sigma_X^2}\mathrm{tr}((X - ZA)^T(X - ZA))\right\}$$

- Bayesian model with Gaussian prior over A

$$P(A|\sigma_A) = \frac{1}{(2\pi\sigma_A^2)^{KD/2}} \exp\left\{-\frac{1}{\sigma_A^2}\mathrm{tr}(A^T A)\right\}$$

- The model remains well defined when $K \to \infty$

Results



Results (Contd.)