RWTH AACHEN UNIVERSITY

Advanced Machine Learning
Lecture 16

Beta Processes

19.12.2012

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

Announcement

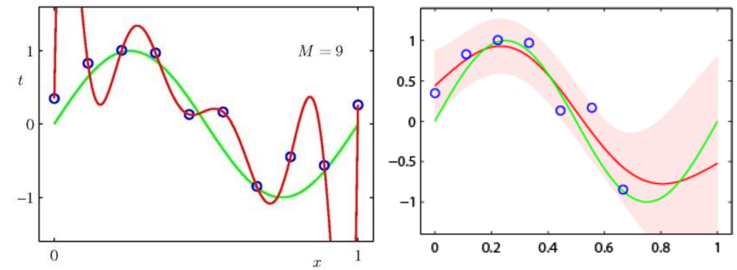
- Exercise sheet 3 will be made available tonight
  - Dirichlet Process Mixture Models
  - Gibbs Sampling
  - Finite Mixtures
  - DPMM Sampling

RWTH AACHEN UNIVERSITY

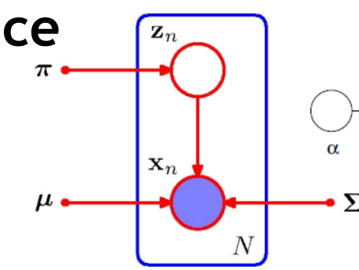
This Lecture: *Advanced Machine Learning*

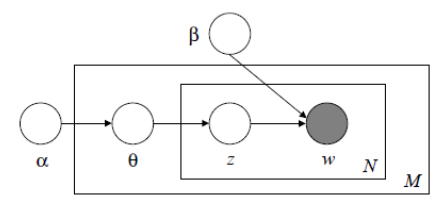
- Regression Approaches
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
  - Gaussian Processes

$$f : \mathcal{X} \rightarrow \mathbb{R}$$



- Bayesian Estimation & Bayesian Non-Parametrics
  - Prob. Distributions, Approx. Inference
  - Mixture Models & EM
  - Dirichlet Processes
  - Latent Factor Models
  - Beta Processes





- SVMs and Structured Output Learning
  - SV Regression, SVDD
  - Large-margin Learning

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Topics of This Lecture

- Latent Factor Models
  - Recap
- Towards Infinite Latent Factor Models
  - General formulation
  - Priors on binary matrices
  - Finite latent feature model
  - Left-ordered binary matrices
  - Indian Buffet Process
- Beta Processes
  - Properties
  - Stick-Breaking construction
  - Efficient Inference
  - Applications

Recap: Latent Factor Models

- **Mixture Models**
  - Assume that each observation was generated by *exactly* one of K components.
  - The uncertainty is just about which component is responsible.

- **Latent Factor Models**
  - Each observation is influenced by *each* of K components (factors or features) in a different way.
  - Sparse factor models: only a small subset of factors is active for each observation.

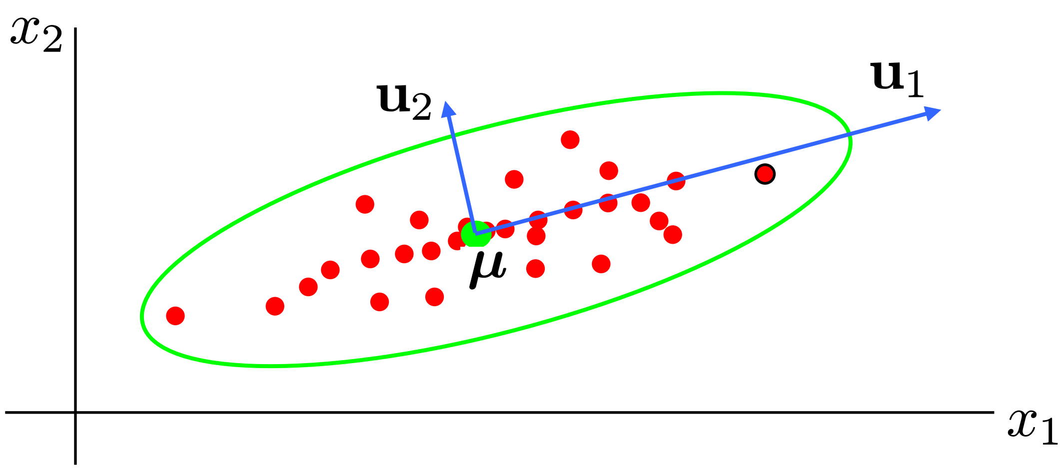
Recap: Principal Component Analysis

- **Find the projection that maximizes the variance**
  - **Covariance matrix of the data**

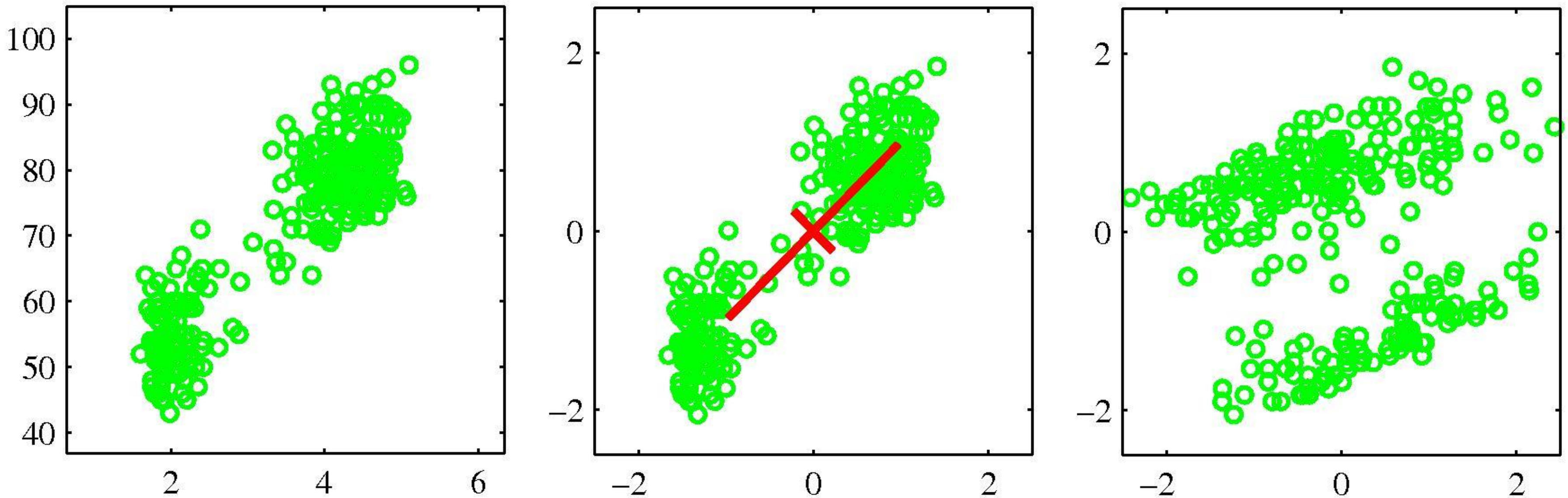
$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

  - **Optimal linear projection into a K-dimensional space is given by the first K eigenvectors $\mathbf{u}_1,\ldots,\mathbf{u}_K$ of $\mathbf{S}$.**

$$\mathbf{y}_n = \mathbf{U}_{1..K}\mathbf{x}_n$$



Recap: PCA for Whitening



Original data | Principal axes | Whitened data

- **Whitening procedure**
  - Define for each data point the transformed value as

$$\mathbf{y}_n = \mathbf{L}^{-1/2}\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})$$

$$\begin{aligned} \mathbf{L} &= \mathrm{diag}\{\lambda_i\} \\ \mathbf{U} &= [\mathbf{u}_1, \ldots, \mathbf{u}_D] \end{aligned}$$

$\Rightarrow$ The transformed set $\{\mathbf{y}_n\}$ has zero mean and unit covariance.

Image source: C.M. Bishop

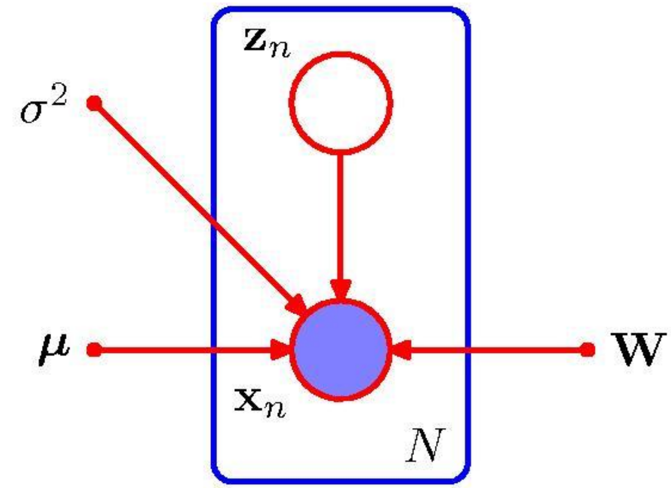
Recap: Probabilistic PCA

- Graphical Model
  - Introduce an explicit latent variable $\mathbf{z}$ corresponding to the principal component subspace.
  - Define a Gaussian prior distribution

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$$

  - Conditional distribution also Gaussian

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$



  - Because of this linear-Gaussian model, the marginal distribution will also be Gaussian

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}), \qquad \mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$$

  - Posterior distribution (again Gaussian)

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x}-\boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}\right), \quad \mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$$

Image source: C.M. Bishop

Recap: Interpretation of Probabilistic PCA

- Analysis
  - Marginal distribution: $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}),$
  - Covariance matrix: $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$

  $\Rightarrow$ The columns of $\mathbf{W}$ define the principal subspace of PCA.

  - Maximum Likelihood estimates

$$\boldsymbol{\mu}_{\mathrm{ML}} = \bar{\mathbf{x}}$$

$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_K(\mathbf{L}_K - \sigma^2\mathbf{I})^{1/2}\mathbf{R}$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{D-K}\sum_{i=K+1}^{D}\lambda_i$$

  $\Rightarrow$ The model correctly captures the variance of the data along the principal axes and approximates the variance in all remaining directions by σ^2, the average of the discarded eigenvalues.

Recap: Examples of Latent Factor Models

- Probabilistic PCA (pPCA)
  - Linear-Gaussian model with isotropic covariance

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

- Factor Analysis (FA)
  - Same linear-Gaussian model, but with diagonal covariance

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \qquad \boldsymbol{\Psi} = \mathrm{diag}\{\psi_i\}$$

- Independent Component Analysis (ICA)
  - Observed variables are related linearly to the latent variables, but the latent distribution is non-Gaussian.
  - Assumption: latent variables z_j are independent.

$$p(\mathbf{z}) = \prod_{j=1}^{K} p(z_j)$$

Topics of This Lecture

- Latent Factor Models
  - Recap
- Towards Infinite Latent Factor Models
  - General formulation
  - Priors on binary matrices
  - Finite latent feature model
  - Left-ordered binary matrices
  - Indian Buffet Process
- Beta Processes
  - Properties
  - Stick-Breaking construction
  - Efficient Inference
  - Applications

Recap: General Latent Factor Models

- General formulation
  - Assume that the data are generated by noisy weighted combination of latent factors

$$\mathbf{x}_n = \mathbf{F}\mathbf{y}_n + \epsilon$$

  - Mixture Models: DPs enforce that the main part of the probability mass is concentrated on few cluster components.
  - Latent Factor Models: enforce that each object is represented by *a sparse subset* of an unbounded number of features.

- Incorporating sparsity
  - Decompose $\mathbf{F}$ into the product of two components: $\mathbf{F} = \mathbf{Z} \otimes \mathbf{W}$, where $\otimes$ is the Hadamard product (element-wise product).
    - z_{mk} is a binary mask variable indicating whether factor k is "on".
    - w_{mk} is a continuous weight variable.

  $\Rightarrow$ Enforce sparsity by restricting the non-zero entries in $\mathbf{Z}$.

Priors on Latent Factor Models

- Defining suitable priors
  - We will focus on defining a prior on $\mathbf{Z}$, since the effective dimensionality of the latent feature model is determined by $\mathbf{Z}$.
  - Assuming that $\mathbf{Z}$ is sparse, we can define a prior for infinite latent feature models by defining a distribution over infinite binary matrices.
- Desiderata for such a distribution
  - Objects should be exchangeable.
  - Inference should be tractable.
- Procedure
  - Start with a model that assumes a finite number of features and consider the limit as this number approaches infinity.

A Finite Feature Model

- **Modeling assumptions**
  - We have N objects and K features.
  - Binary variables z_{nk} indicates that object n possesses feature k.
  - Each object possesses feature k with probability π_k and features are generated independently.

$\Rightarrow$ The probability of a matrix $\mathbf{Z}$ given $\boldsymbol{\pi} = \{\pi_1,\ldots,\pi_k\}$ is given by a Binomial distribution

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \prod_{n=1}^{N} p(z_{nk}|\pi_k) = \prod_{k=1}^{K} \pi_k^{m_k} (1-\pi_k)^{N-m_k}$$

where $m_k = \sum_{n=1}^{N} z_{nk}$ is the number of objects possessing feature k.

A Finite Feature Model

- **Defining a prior**
  - Define a prior on π by assuming that each π_k follows a Beta distribution (conjugate to the binomial):

$$p(\pi_k) = \mathrm{Beta}(\pi_k; r, s) = \frac{\pi_k^{r-1}(1-\pi_k)^{s-1}}{B(r,s)}$$

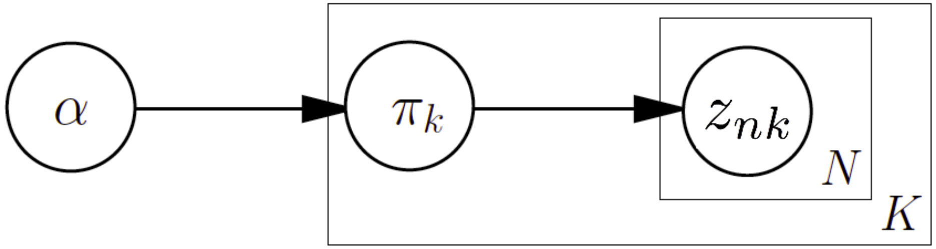
where $B(r,s)$ is the beta function

$$B(r,s) = \int_0^1 \pi_k^{r-1}(1-\pi_k)^{s-1} \mathrm{d}\pi_k = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)}$$

  - We set $r = \alpha/K$ and $s = 1$, so this equation becomes

$$B(\frac{\alpha}{K}, 1) = \frac{\Gamma(\frac{\alpha}{K})}{\Gamma(1+\frac{\alpha}{K})} = \frac{K}{\alpha}$$

A Finite Feature Model



- **Resulting probability model**
  - **Finite Beta-Bernoulli model**

$$\begin{aligned} \pi_k \mid \alpha &\sim \mathrm{Beta}(\frac{\alpha}{K}, 1) \\ z_{nk} \mid \pi_k &\sim \mathrm{Bernoulli}(\pi_k) \end{aligned}$$

  - **Each z_{nk} is independent of all other assignments conditioned on π_k and the π_k are generated independently.**

Image source: Erik Sudderth

A Finite Feature Model

- **We can now marginalize out π**
  - **Marginal probability of the matrix $\mathbf{Z}$:**

$$
\begin{aligned}
p(\mathbf{Z}) &= \prod_{k=1}^{K} \int \left(\prod_{n=1}^{N} p(z_{nk}|\pi_k) \right) p(\pi_k) \mathrm{d}\pi_k \\
&= \prod_{k=1}^{K} \frac{B(m_k + \frac{\alpha}{K}, N - m_k + 1)}{B(\frac{\alpha}{K}, 1)} \\
&= \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})}
\end{aligned}
$$

conjugacy b/w binomial and beta

$\Rightarrow$ **This distribution depends only on the counts m_k.**

$\Rightarrow$ **It is therefore exchangeable.**

Important Property

- **Bound on the number of entries**
  - **Expectation of the number of non-zero entries in $\mathbf{Z}$:**

$$\mathbb{E}\left[\mathbf{1}^T \mathbf{z}_k \mathbf{1}\right] = \mathbb{E}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\right] = K\mathbb{E}\left[\mathbf{1}^T \mathbf{z}_k\right]$$

(columns of Z are in-dependent)

$$= K\sum_{n=1}^{N}\mathbb{E}[z_{nk}] = K\sum_{n=1}^{N}\underbrace{\int_0^1 \pi_k p(\pi_k)\mathrm{d}\pi_k}$$

Expectation of a $\mathrm{Beta}(r,s)$ random variable is $r/(r+s)$

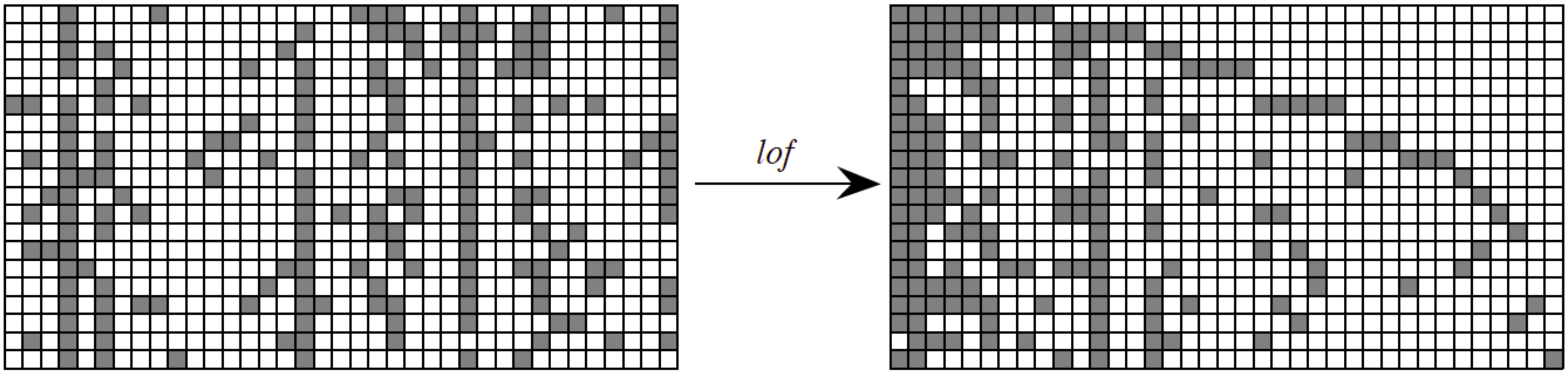
$$= KN\frac{\frac{\alpha}{K}}{1+\frac{\alpha}{K}} = \frac{N\alpha}{1+\alpha/K}$$

$\Rightarrow$ **For any K, the expectation of this number is bounded by $N\alpha$.**

Topics of This Lecture

- Latent Factor Models
  - Recap
- Towards Infinite Latent Factor Models
  - General formulation
  - Priors on binary matrices
  - Finite latent feature model
  - Left-ordered binary matrices
  - Indian Buffet Process
- Beta Processes
  - Properties
  - Stick-Breaking construction
  - Efficient Inference
  - Applications

Equivalence Class of Binary Matrices



- **Equivalence class of binary matrices**
  - **Define a function $\mathrm{lof}(\mathbf{Z})$ that maps binary matrices into left-ordered binary matrices by ordering the columns of $\mathbf{Z}$ by the magnitude of the binary number expressed by that column.**
  - **There is a unique left-ordered form for every binary matrix.**
  - **Two matrices $\mathbf{Y}$ and $\mathbf{Z}$ are equivalent iff $\mathrm{lof}(\mathbf{Y}) = \mathrm{lof}(\mathbf{Z})$.**
  - **The lof-equivalence class of $\mathbf{Z}$ is denoted $[\mathbf{Z}]$.**

Image source: Zoubin Ghahramani

Equivalence Class of Binary Matrices

- **What is the cardinality of $[\mathbf{Z}]$?**
  - Columns of a binary matrix are not guaranteed to be unique:
  - Since an object can possess multiple features, it is possible for two features to be possessed by exactly the same set of objects.
  - The cardinality of $[\mathbf{Z}]$ is therefore reduced if $\mathbf{Z}$ contains identical columns

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

where K_h is the number of columns with binary number h.

Towards Infinite Feature Models

- **Taking the limit $K \to \infty$**
  - **Probability of a** lof**-equivalence class of binary matrices**

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

  - **Reordering the columns such that $m_k > 0$ if $k \leq K_+$, we can derive (after several intermediate steps)**

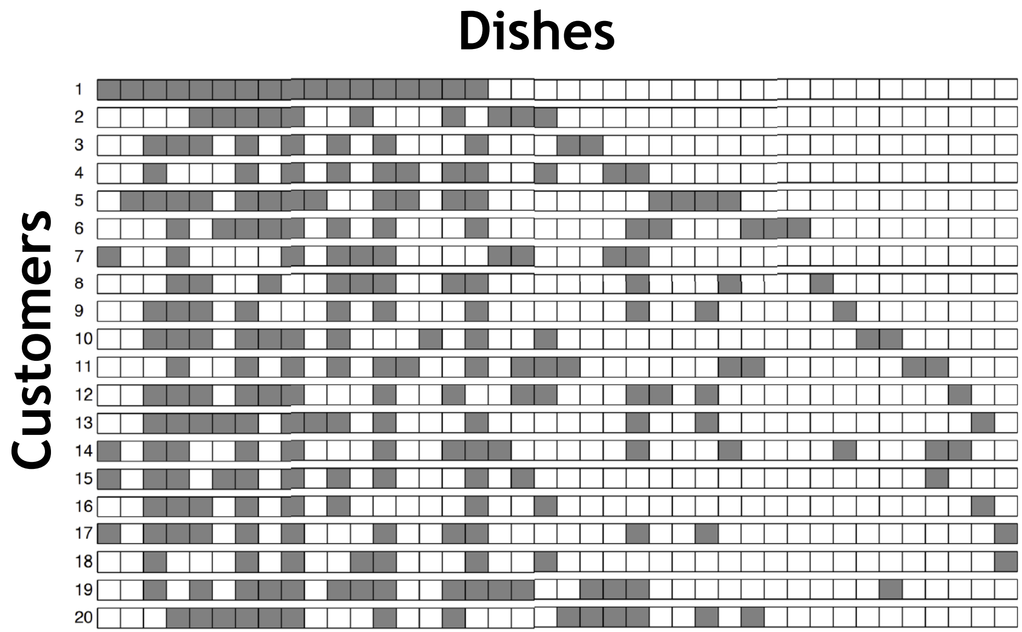
$$\lim_{K\to\infty} p([\mathbf{Z}]) = \frac{\alpha^{K_+}}{\prod_{h=0}^{2^N-1} K_h!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}$$

  - **where H_N is the N^{th} harmonic number $H_N = \sum_{j=1}^{N} 1/j$.**

Topics of This Lecture

- Latent Factor Models
  - Recap
- Towards Infinite Latent Factor Models
  - General formulation
  - Priors on binary matrices
  - Finite latent feature model
  - Left-ordered binary matrices
  - Indian Buffet Process
- Beta Processes
  - Properties
  - Stick-Breaking construction
  - Efficient Inference
  - Applications

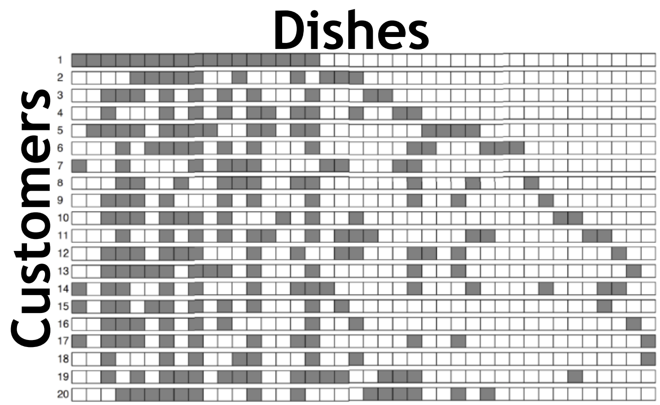
The Indian Buffet Process



"Many Indian restaurants in London offer lunchtime buffets with an apparently infinite number of dishes"

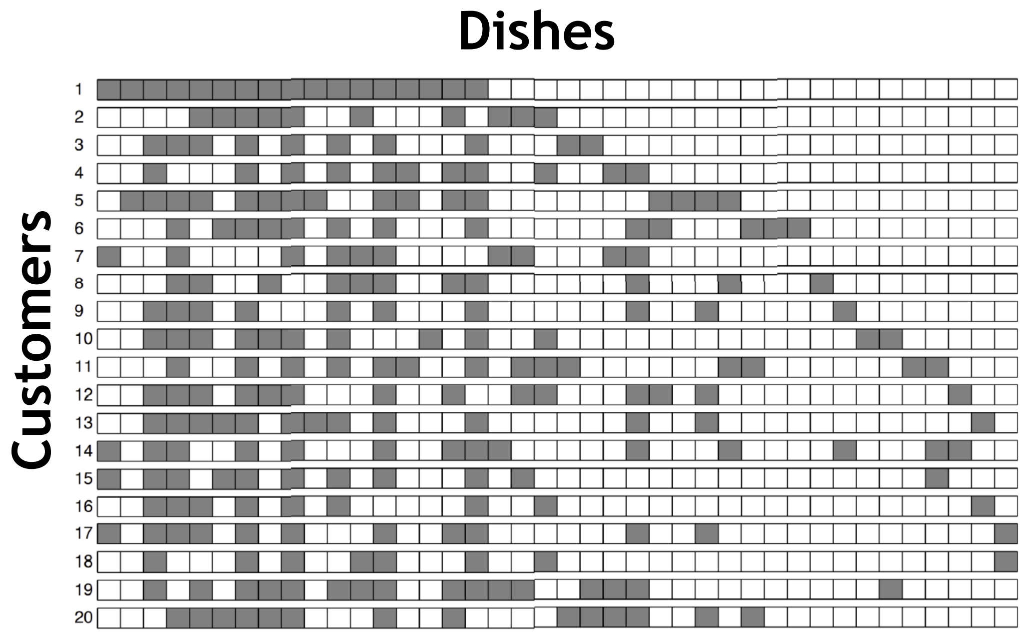
[Zoubin Ghahramani]

The Indian Buffet Process



- **Analogy to Chinese Restaurant Process**
  - Visualize feature assignment as a sequential process of customers sampling dishes from an (infinitely long) buffet
  - 1st customer starts at the left of the buffet, and takes a serving from each dish, stopping after a $\mathrm{Poisson}()$ number of dishes as her plate becomes overburdened.
  - The n^{th} customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself with probability m_k/n, and trying a $\mathrm{Poisson}(\alpha/n)$ number of new dishes.
  - The customer-dish matrix is our feature matrix, $\mathbf{Z}$.

The Indian Buffet Process (IBP)



- **Properties of the IBP**
  - Generative process to create samples from an infinite latent feature model.
  - The IBP is exchangeable, up to a permutation of the order with which dishes are listed in the feature matrix.
  - The number of features sampled at least once is $\mathcal{O}(\alpha \log N)$.

to be continued in 2013

References and Further Reading

- Tutorial papers for infinite latent factor models
  - A good introduction to the topic
    - Z. Ghahramani, T.L. Griffiths, P. Sollich, "Bayesian Nonparametric Latent Feature Models", Bayesian Statistics, 2006.