Nonparametric Probabilistic Modelling

Zoubin Ghahramani

Department of Engineering
University of Cambridge, UK

zoubin@eng.cam.ac.uk
http://learning.eng.cam.ac.uk/zoubin/

Signal processing and inference in the physical sciences

Royal Society 2012

Probabilistic Modelling

- A model describes data that one could observe from a system
- If we use the mathematics of probability theory to express all forms of uncertainty and noise associated with our model...
- ...then *inverse probability* (i.e. Bayes rule) allows us to infer unknown quantities, adapt our models, make predictions and learn from data.

Bayesian Modelling

Everything follows from two simple rules:

Sum rule: $P(x) = \sum_y P(x, y)$

Product rule: $P(x, y) = P(x)P(y|x)$

$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta, m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D}, m)$	posterior of θ given data $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

Bayesian Occam's Razor and Model Comparison

Compare model classes, e.g. m and m', using posterior probabilities given $\mathcal{D}$:

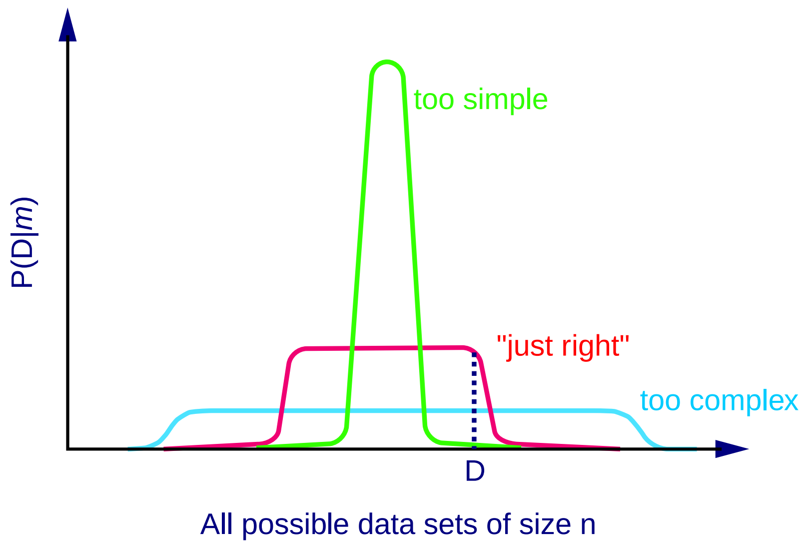
$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)\, p(m)}{p(\mathcal{D})}, \qquad p(\mathcal{D}|m) = \int p(\mathcal{D}|\boldsymbol{\theta}, m)\; p(\boldsymbol{\theta}|m)\; d\boldsymbol{\theta}$$

Interpretations of the Marginal Likelihood ("model evidence"):

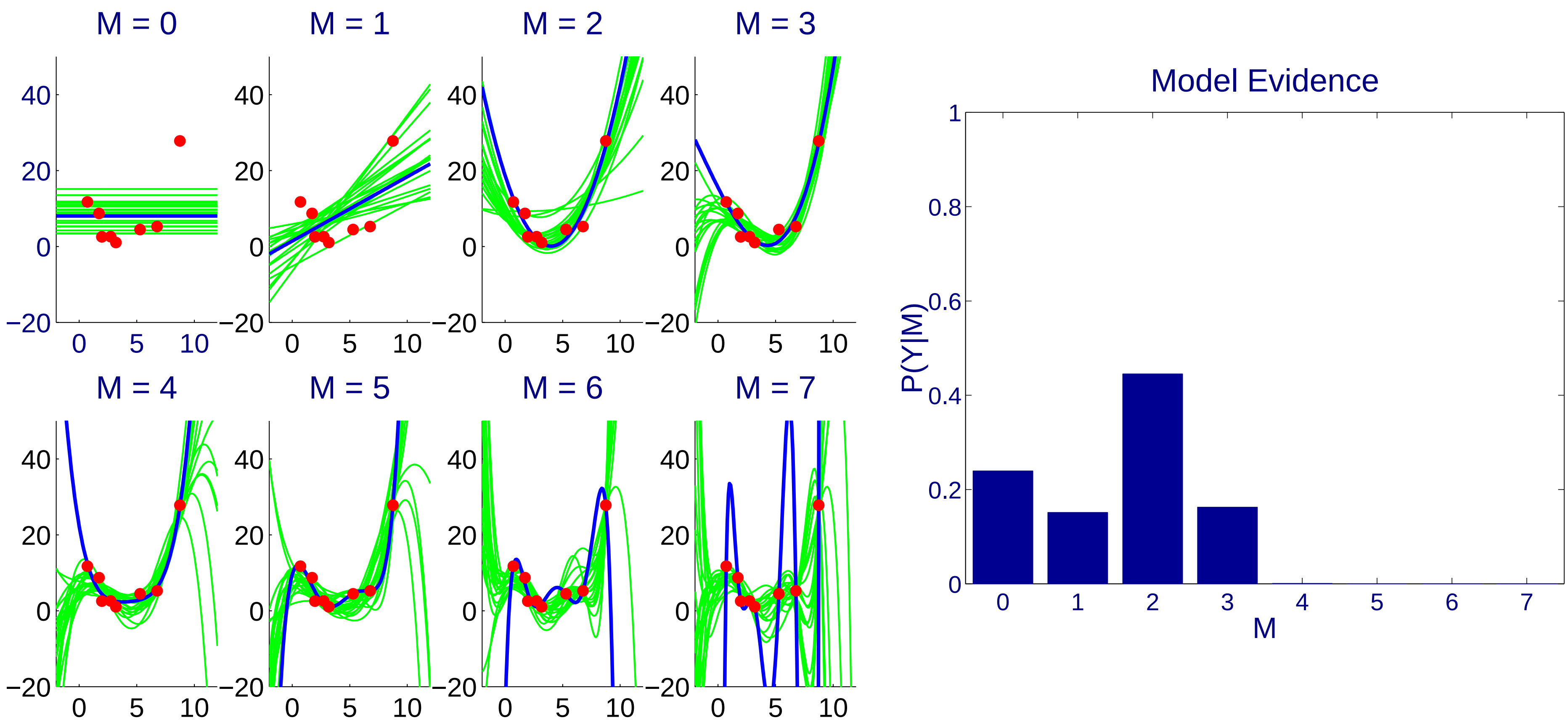
- The probability that *randomly selected* parameters from the prior would generate $\mathcal{D}$.
- Probability of the data under the model, *averaging* over all possible parameter values.
- $\log_2 \left(\frac{1}{p(\mathcal{D}|m)} \right)$ is the number of *bits of surprise* at observing data $\mathcal{D}$ under model m.

Model classes that are too simple are unlikely to generate the data set.

Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.



Bayesian Model Comparison: Occam's Razor at Work

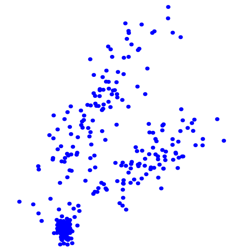


For example, for quadratic polynomials ($m = 2$): $y = a_0 + a_1 x + a_2 x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and parameters $\boldsymbol{\theta} = (a_0\ a_1\ a_2\ \sigma)$

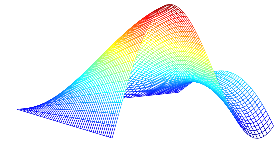
```
demo:  polybayes
```

Learning Model Structure

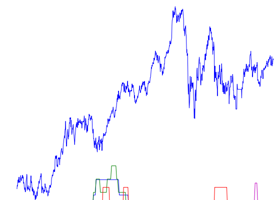
How many clusters in the data?



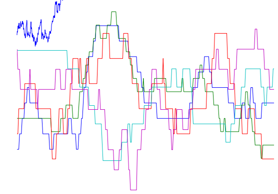
What is the intrinsic dimensionality of the data?



Is this input relevant to predicting that output?

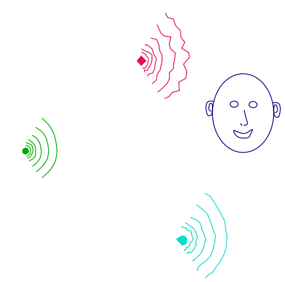


What is the order of a dynamical system?

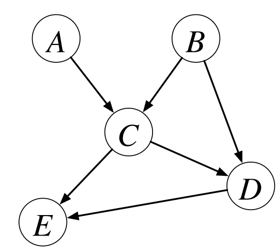


How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY



How many auditory sources in the input?



What is the structure of a graphical model?

Approximate Inference

$$
\begin{aligned}
P(x|\mathcal{D},m) &= \int P(x|\theta,\mathcal{D},m)P(\theta|\mathcal{D},m)d\theta \\
P(\mathcal{D}|m) &= \int P(\mathcal{D}|\theta,m)P(\theta|m)\,d\theta
\end{aligned}
$$

How do we compute these integrals in practice?

- Laplace Approximation
- Bayesian Information Criterion (BIC)
- Variational Bayesian approximations
- Expectation Propagation (and loopy belief propagation)
- Markov chain Monte Carlo
- Sequential Monte Carlo
- ...

Bayesian Nonparametrics

Why...

- **Why Bayesian?**

  Simplicity (of the framework)

- **Why nonparametrics?**

  Complexity (of real world phenomena)

Parametric vs Nonparametric Models

- *Parametric models* assume some finite set of parameters θ. Given the parameters, future predictions, x, are independent of the observed data, $\mathcal{D}$:

$$P(x|\theta, \mathcal{D}) = P(x|\theta)$$

  therefore θ capture everything there is to know about the data.

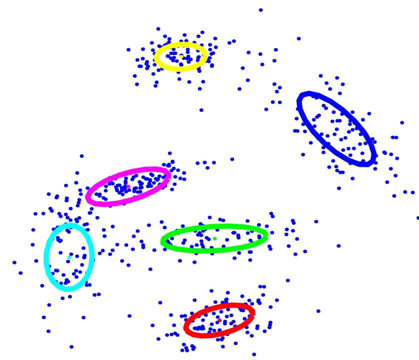
- So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.

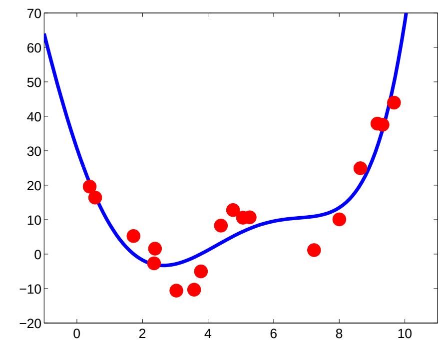
- *Non-parametric models* assume that the data distribution cannot be defined in terms of such a finite set of parameters. But they can often be defined by assuming an *infinite dimensional* θ. Usually we think of θ as a *function*.

- The amount of information that θ can capture about the data $\mathcal{D}$ can grow as the amount of data grows. This makes them more flexible.

Why nonparametrics?

- flexibility
- better predictive performance
- more realistic





All successful methods in machine learning are essentially nonparametric[1]:

- kernel methods / SVM / GP
- deep networks / large neural networks
- k-nearest neighbors, ...

[1] or highly scalable!

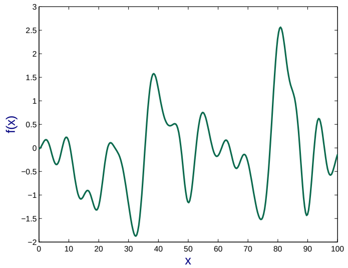
Overview of nonparametric models and uses

Bayesian nonparametrics has many uses.

Some modelling goals and *examples* of associated nonparametric Bayesian models:

Modelling goal	Example process
Distributions on functions	Gaussian process
Distributions on distributions	Dirichlet process
	Polya Tree
Clustering	Chinese restaurant process
	Pitman-Yor process
Hierarchical clustering	Dirichlet diffusion tree
	Kingman's coalescent
Sparse binary matrices	Indian buffet processes
Survival analysis	Beta processes
Distributions on measures	Completely random measures
...	...

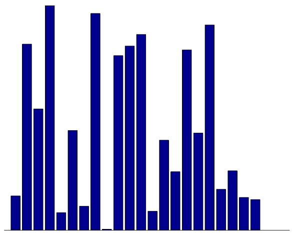
Gaussian and Dirichlet Processes



- Gaussian processes define a distribution on functions

$$f \sim \mathrm{GP}(\cdot|\mu, c)$$

where μ is the mean function and c is the covariance function.
We can think of GPs as "infinite-dimensional" Gaussians



- Dirichlet processes define a distribution on distributions

$$G \sim \mathrm{DP}(\cdot|G_0, \alpha)$$

where $\alpha > 0$ is a scaling parameter, and G_0 is the base measure.
We can think of DPs as "infinite-dimensional" Dirichlet distributions.

Note that both f and G are infinite dimensional objects.

Outline

Bayesian nonparametrics applied to models of other structured objects:

- Time Series

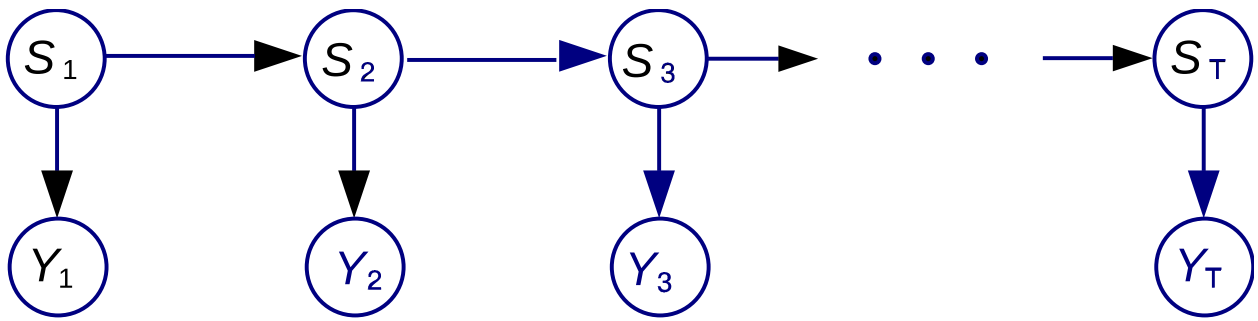
- Sparse Matrices

- Networks

Time Series

Hidden Markov Models

Hidden Markov models (HMMs) are widely used sequence models for speech recognition, bioinformatics, biophysics, text modelling, video monitoring, etc.



In an HMM, the sequence of observations $\mathbf{y}_1, \ldots, \mathbf{y}_T$ is modelled by assuming that it was generated by a sequence of discrete hidden states $s_1, \ldots, s_T$ with Markovian dynamics.

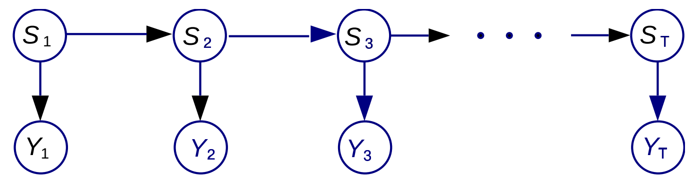
If the HMM has K states ($s_t \in \{1, \ldots K\}$) the transition matrix has $K \times K$ elements.

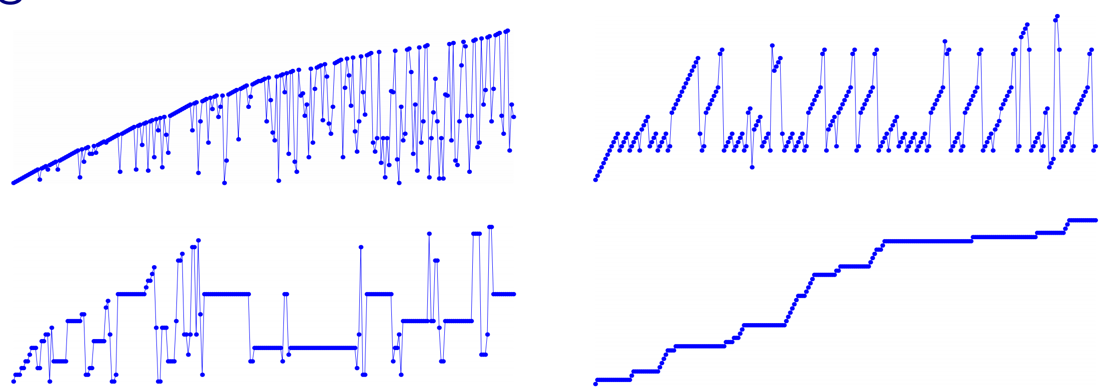
HMMs can be thought of as *time-dependent mixture models*.

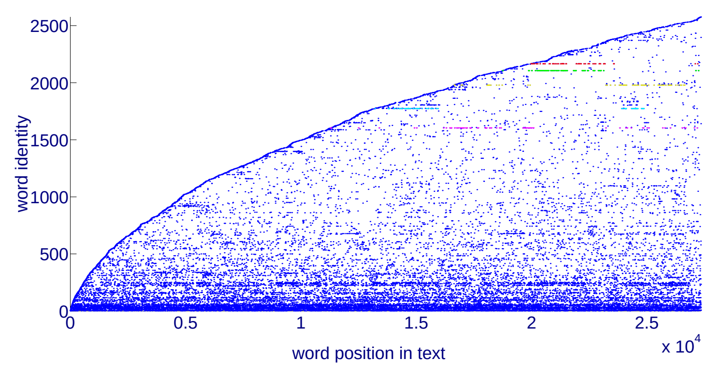
Infinite hidden Markov models (iHMMs)

Let the number of hidden states $K \to \infty$.

Here are some typical state trajectories for an iHMM. Note that the number of states visited grows with T.

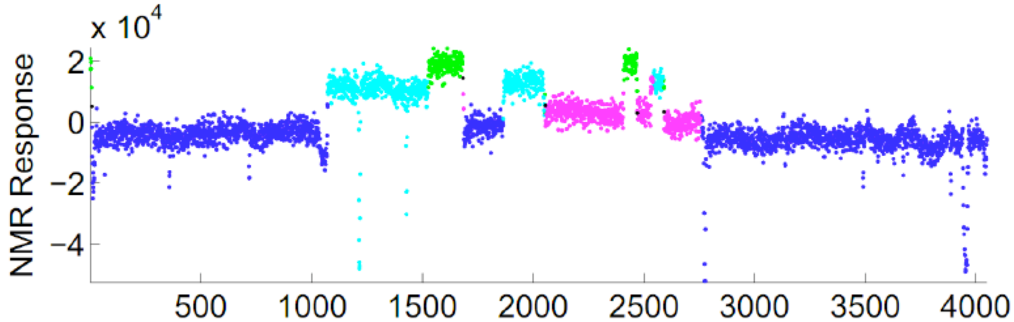


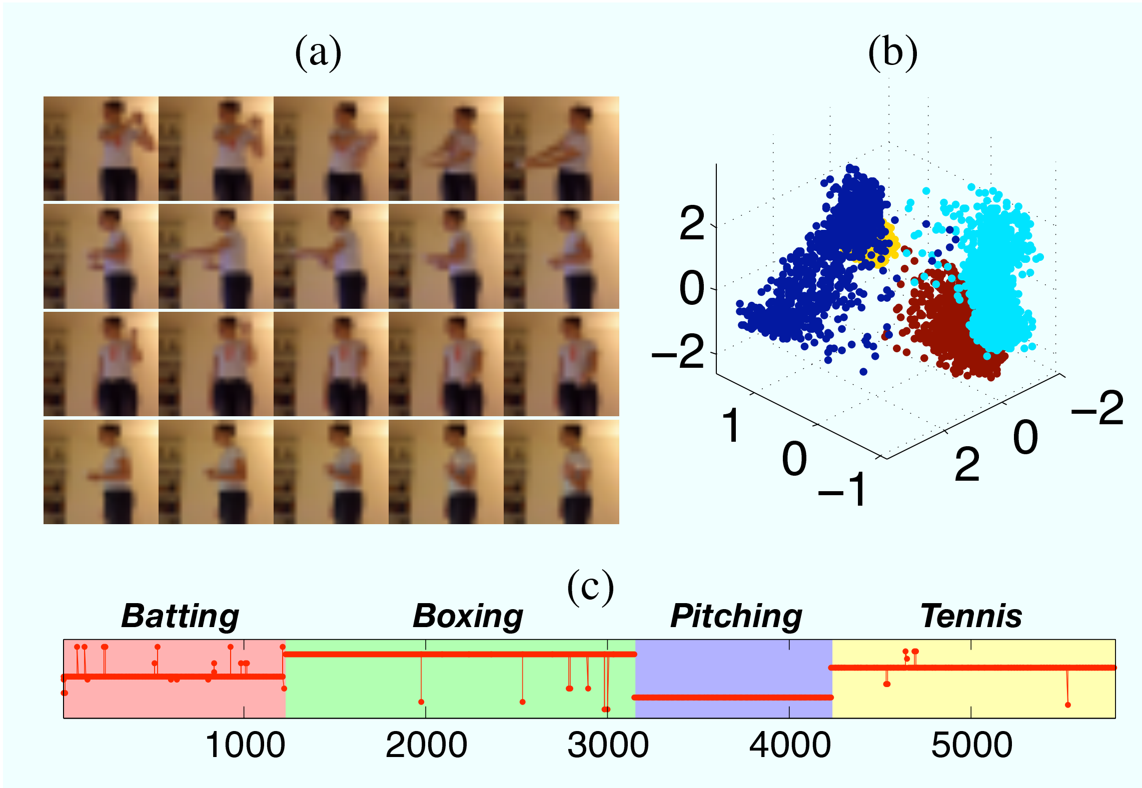




- Introduced in (Beal, Ghahramani and Rasmussen, 2002).
- Teh, Jordan, Beal and Blei (2005) showed that iHMMs can be derived from hierarchical Dirichlet processes, and provided a more efficient Gibbs sampler.
- We have recently derived a much more efficient sampler based on Dynamic Programming (Van Gael, Saatci, Teh, and Ghahramani, 2008). `http://mloss.org/software/view/205/`
- And we have parallel (.NET) and distributed (Hadoop) implementations (Bratieres, Van Gael, Vlachos and Ghahramani, 2010).

Infinite HMM: Changepoint detection and video segmentation

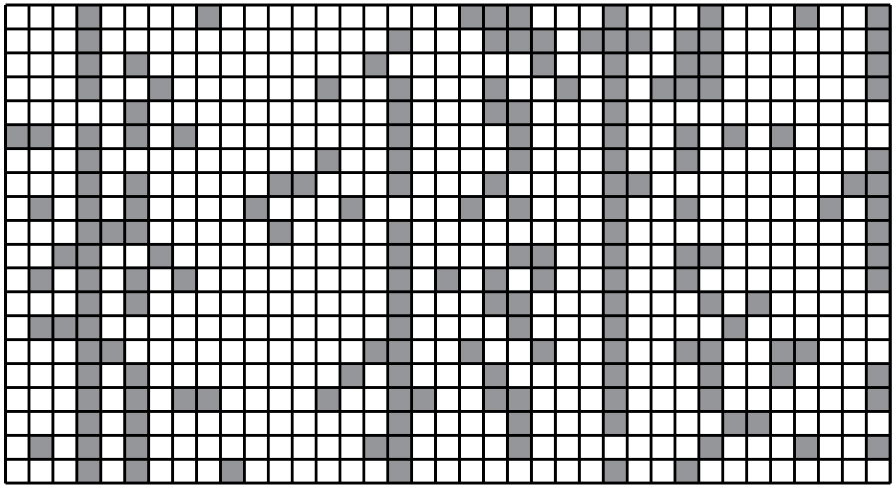




(w/ Tom Stepleton, 2009)

Sparse Matrices

From finite to infinite sparse binary matrices



$z_{nk} = 1$ means object n has feature k:

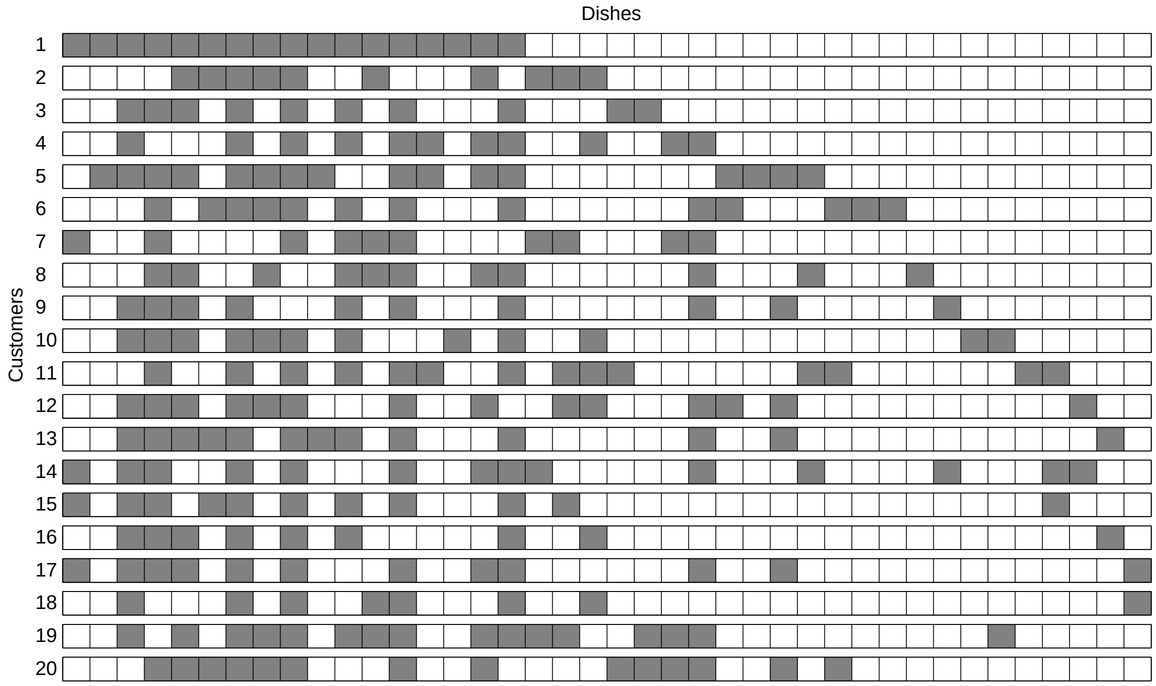
$$z_{nk} \sim \mathrm{Bernoulli}(\theta_k)$$

$$\theta_k \sim \mathrm{Beta}(\alpha/K, 1)$$

- Note that $P(z_{nk} = 1|\alpha) = E(\theta_k) = \frac{\alpha/K}{\alpha/K+1}$, so as K grows larger the matrix gets sparser.
- So if $\mathbf{Z}$ is $N \times K$, the expected number of nonzero entries is $N\alpha/(1+\alpha/K) < N\alpha$.
- Even in the $K \to \infty$ limit, the matrix is expected to have a finite number of non-zero entries.
- $K \to \infty$ results in an Indian buffet process (IBP)[2]

[2]Naming inspired by analogy to "Chinese restaurant process" (CRP) from probability theory.

Indian buffet process



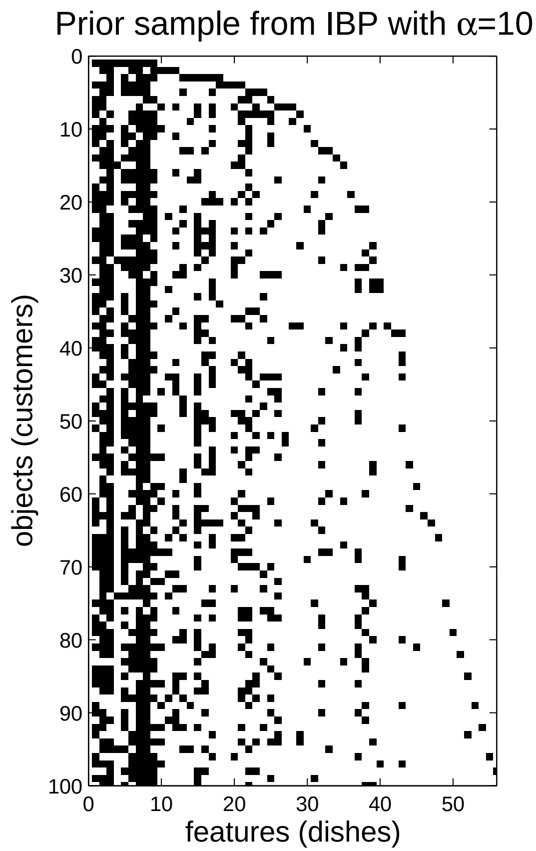
"Many Indian restaurants in London offer lunchtime buffets with an apparently infinite number of dishes"

- First customer starts at the left of the buffet, and takes a serving from each dish, stopping after a Poisson(α) number of dishes as his plate becomes overburdened.
- The n^{th} customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself dish k with probability m_k/n, and trying a Poisson(α/n) number of new dishes.
- The customer-dish matrix, $\mathbf{Z}$, is a draw from the IBP.

(w/ Tom Griffiths 2006; 2011)

Properties of the Indian buffet process

$$P([\mathbf{Z}]|\alpha) = \exp\big\{-\alpha H_N\big\} \frac{\alpha^{K_+}}{\prod_{h>0} K_h!} \prod_{k \le K_+} \frac{(N-m_k)!(m_k-1)!}{N!}$$



Shown in (Griffiths and Ghahramani 2006, 2011):

- It is infinitely exchangeable.
- The number of ones in each row is Poisson(α)
- The expected total number of ones is αN.
- The number of nonzero columns grows as $O(\alpha \log N)$.

Additional properties:

- Has a stick-breaking representation (Teh, et al 2007)
- Has as its de Finetti mixing distribution the Beta process (Thibaux and Jordan 2007)
- More flexible two and three parameter versions exist (w/ Griffiths & Sollich 2007; Teh and Görür 2010)

Posterior Inference in IBPs

$$P(\mathbf{Z}, \alpha|\mathbf{X}) \propto P(\mathbf{X}|\mathbf{Z})P(\mathbf{Z}|\alpha)P(\alpha)$$

Gibbs sampling: $P(z_{nk} = 1|\mathbf{Z}_{-(nk)}, \mathbf{X}, \alpha) \propto P(z_{nk} = 1|\mathbf{Z}_{-(nk)}, \alpha)P(\mathbf{X}|\mathbf{Z})$

- If $m_{-n,k} > 0$, $P(z_{nk} = 1|\mathbf{z}_{-n,k}) = \frac{m_{-n,k}}{N}$
- For infinitely many k such that $m_{-n,k} = 0$: Metropolis steps with truncation* to sample from the number of new features for each object.
- If α has a Gamma prior then the posterior is also Gamma $\rightarrow$ Gibbs sample.

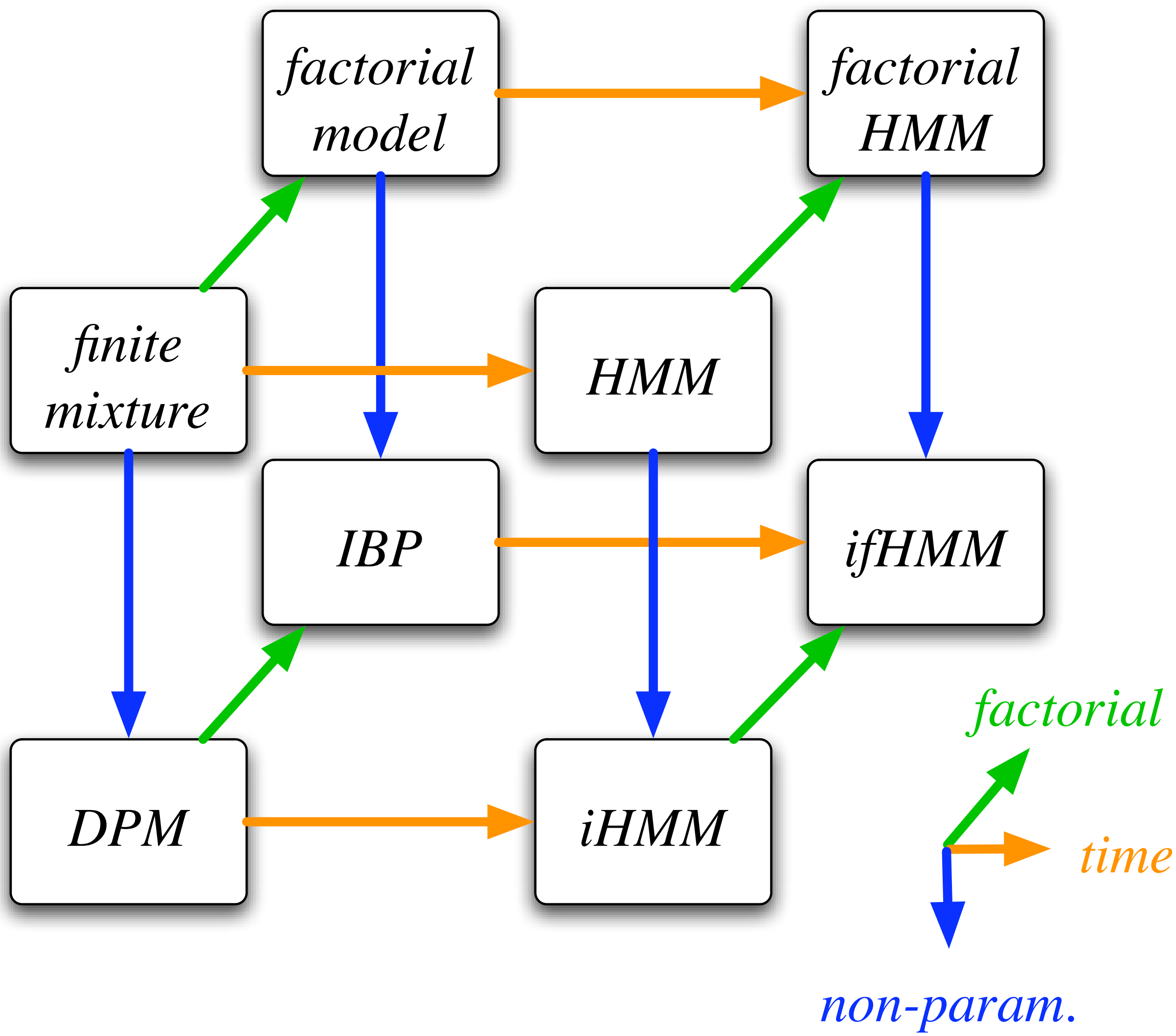
Conjugate sampler: assumes that $P(\mathbf{X}|\mathbf{Z})$ can be computed.

Non-conjugate sampler: $P(\mathbf{X}|\mathbf{Z}) = \int P(\mathbf{X}|\mathbf{Z}, \theta)P(\theta)d\theta$ cannot be computed, requires sampling latent θ as well (e.g. approximate samplers based on (Neal 2000) non-conjugate DPM samplers).

Slice sampler: works for non-conjugate case, is not approximate, and has an *adaptive truncation level* using an IBP stick-breaking construction (Teh, et al 2007) see also (Adams et al 2010).

Deterministic Inference: variational inference (Doshi et al 2009a) parallel inference (Doshi et al 2009b), beam-search MAP (Rai and Daume 2011), power-EP (Ding et al 2010)

The Big Picture:
Relations between some models



Modelling Data with Indian Buffet Processes

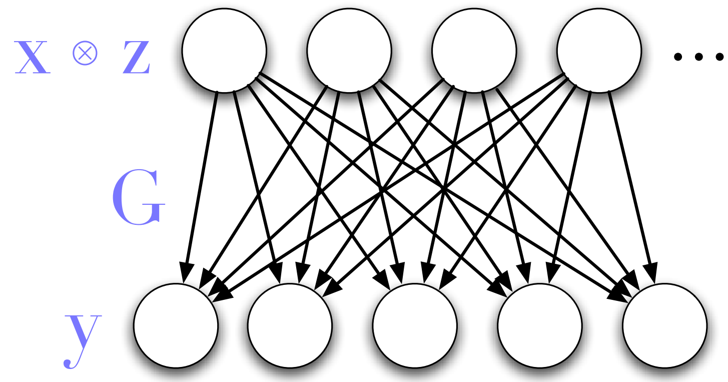
Latent variable model: let $\mathbf{X}$ be the $N \times D$ matrix of observed data, and $\mathbf{Z}$ be the $N \times K$ matrix of sparse binary latent features

$$P(\mathbf{X}, \mathbf{Z}|\alpha) = P(\mathbf{X}|\mathbf{Z})P(\mathbf{Z}|\alpha)$$

By combining the IBP with different likelihood functions we can get different kinds of models:

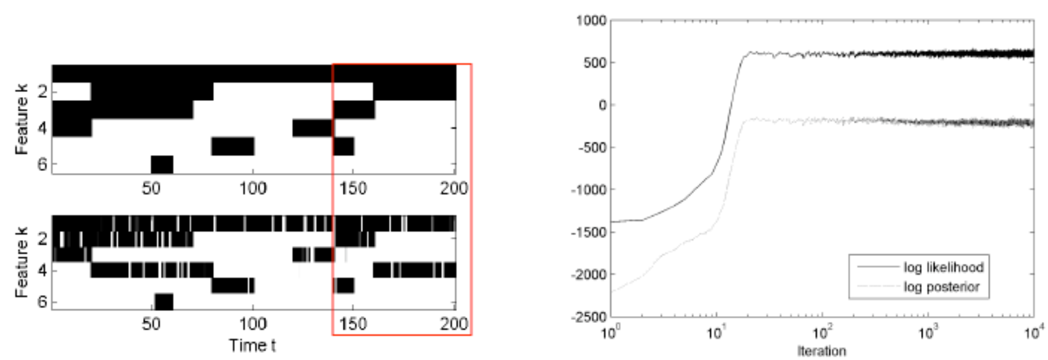
- Models for graph structures (w/ Wood, Griffiths, 2006; w/ Adams and Wallach, 2010)
- Models for protein complexes (w/ Chu, Wild, 2006)
- Models for choice behaviour (Görür & Rasmussen, 2006)
- Models for users in collaborative filtering (w/ Meeds, Roweis, Neal, 2007)
- Sparse latent trait, pPCA and ICA models (w/ Knowles, 2007, 2011)
- Models for overlapping clusters (w/ Heller, 2007)

Infinite Independent Components Analysis



Model: $\mathbf{Y} = \mathbf{G}(\mathbf{Z} \otimes \mathbf{X}) + \mathbf{E}$

where $\mathbf{Y}$ is the data matrix, $\mathbf{G}$ is the mixing matrix $\mathbf{Z} \sim \mathrm{IBP}(\alpha, \beta)$ is a mask matrix, $\mathbf{X}$ is heavy tailed sources and $\mathbf{E}$ is Gaussian noise.



(a) *Top:* True $\mathbf{Z}$. *Bottom:* Inferred $\mathbf{Z}$. Red box denotes test data.

(b) Plot of the log likelihood and posterior for the duration of the iICA_2 run.

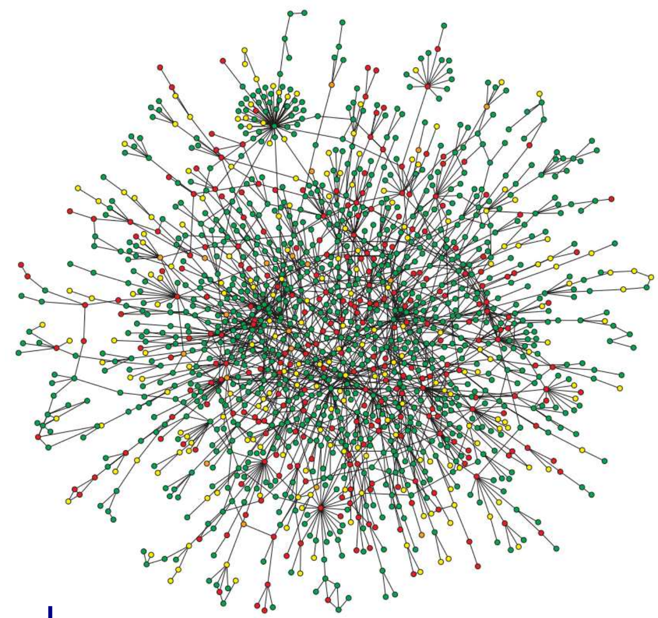
Fig. 1. True and inferred $\mathbf{Z}$ and algorithm convergence.

(w/ David Knowles, 2007, 2011)

Networks

Modelling Networks

We are interested in modelling networks.



Biological networks: protein-protein interaction networks

Social networks: friendship networks; co-authorship networks

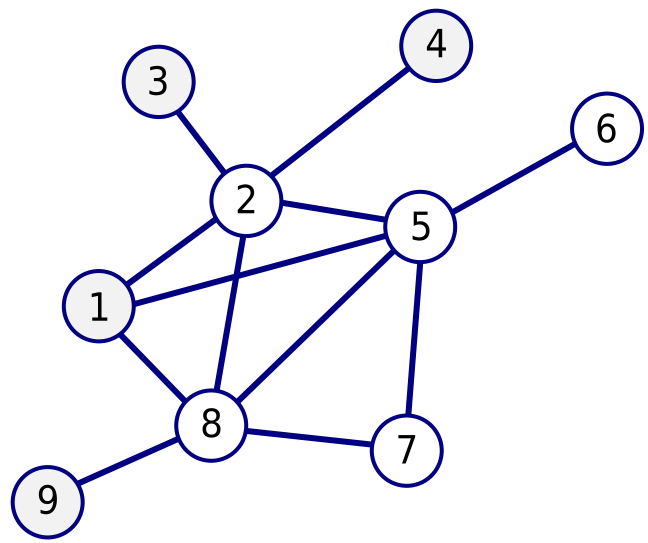
We wish to have models that will be able to

- predict missing links,
- infer latent properties or classes of the objects,
- generalise learned properties from smaller observed networks to larger networks.

Figure from Barabasi and Oltvai 2004: A protein-protein interaction network of budding yeast

What is a network?

- A set $\mathcal{V}$ of entities (nodes, vertices) and
- A set $\mathcal{Y}$ of pairwise relations (links, edges) between the entities



We can represent this as a graph with a binary adjacency matrix $\mathbf{Y}$ where element $y_{ij} = 1$ represents a link between nodes v_i and v_j

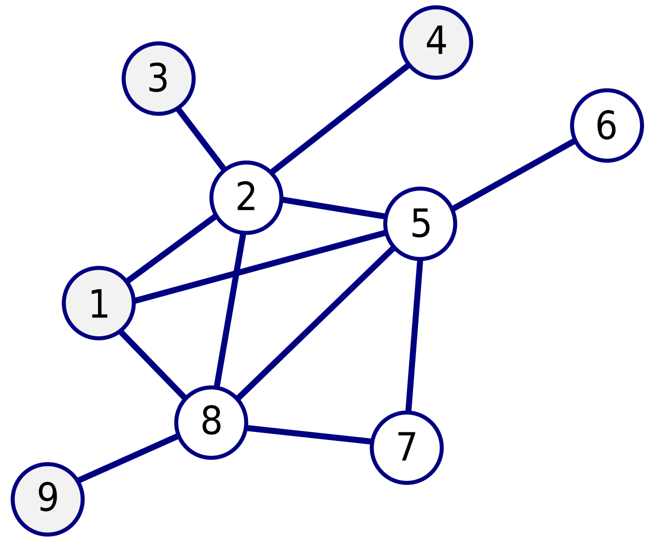
We'll focus on undirected graphs (i.e. networks of symmetric relations) but much of what is discussed extends to more general graphs.

What is a model?

Descriptive statistics: identify interesting properties of a network (e.g. degree distribution)

Predictive or generative model: A model that could generate random networks and predict missing links, etc.

Erdös-Rényi Model

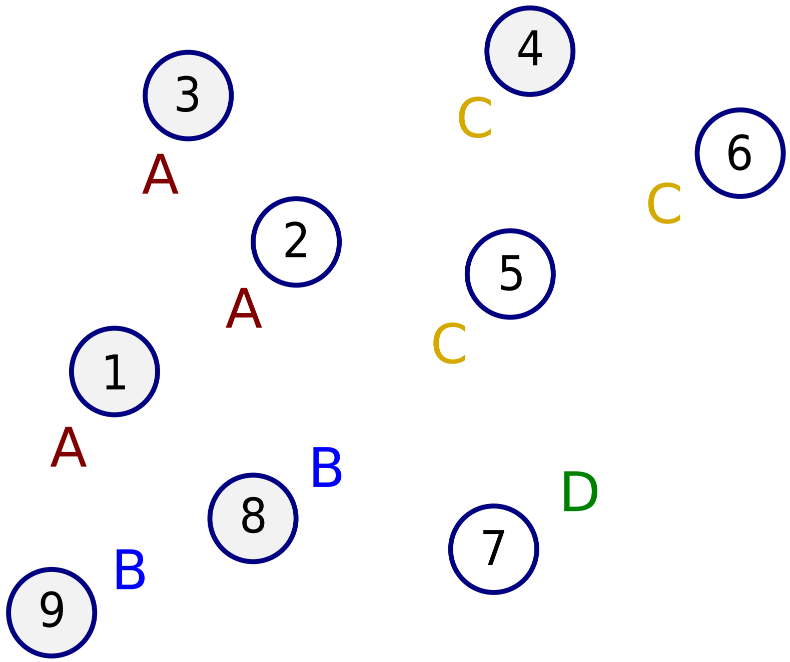


A very simple model that assumes each link is independent, and present with probability $\pi \in [0, 1]$

$$y_{ij} \sim \mathrm{Bern}(\pi)$$

This model is easy to analyse but does not have any interesting structure or make any nontrivial predictions. The only thing one can learn from such a model is the average density of the network.

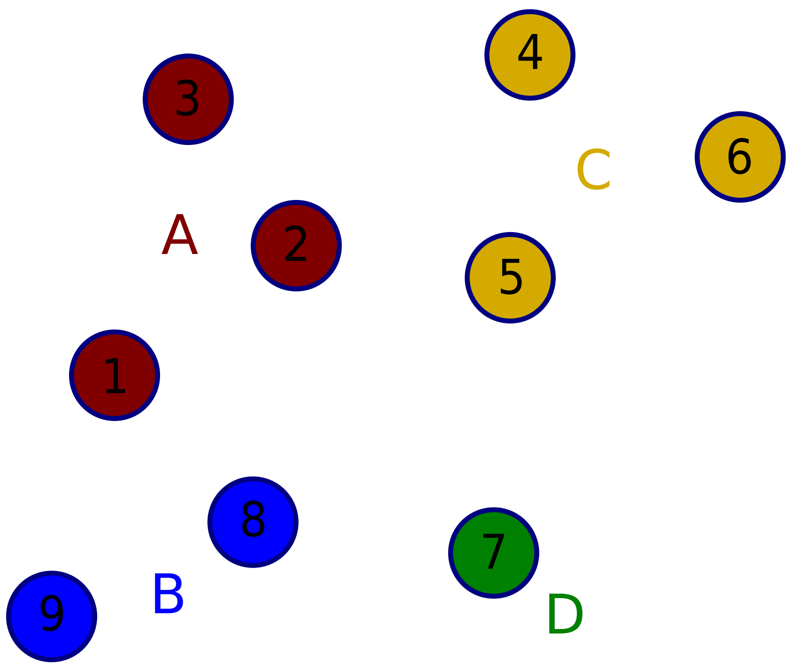
Latent Class Models



The basic idea is to posit that the structure of the network arises from latent (or hidden) variables associated with each node.

We can think of latent class models as having a single discrete hidden variable associated with each node.

Latent Class Models

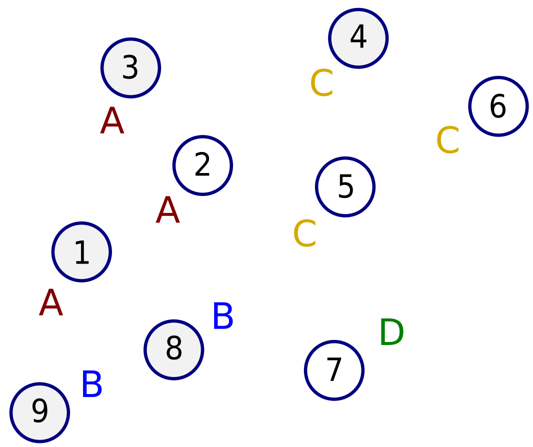


This corresponds to a *clustering* of the nodes.
Such models can be used for *community detection*.

For example, the discrete hidden variables might correspond to the political views of each individual in a social network.

Latent Class Models

Stochastic Block Model (Nowicki and Snijders, 2001)



Each node v_i has a hidden class from a set of K possible classes: $c_i \in \{1, \ldots, K\}$

For all i:

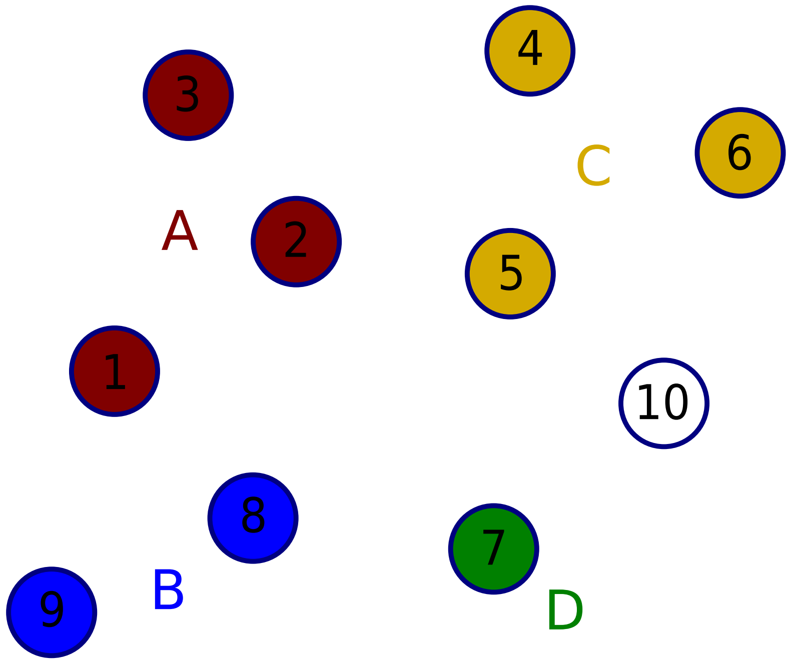
$$c_i \sim \mathrm{Discrete}(p_1, \ldots p_K)$$

The probability of a link between two nodes v_i and v_j depends on their classes:

$$P(y_{ij} = 1 | c_i = k, c_j = \ell) = \rho_{k\ell}$$

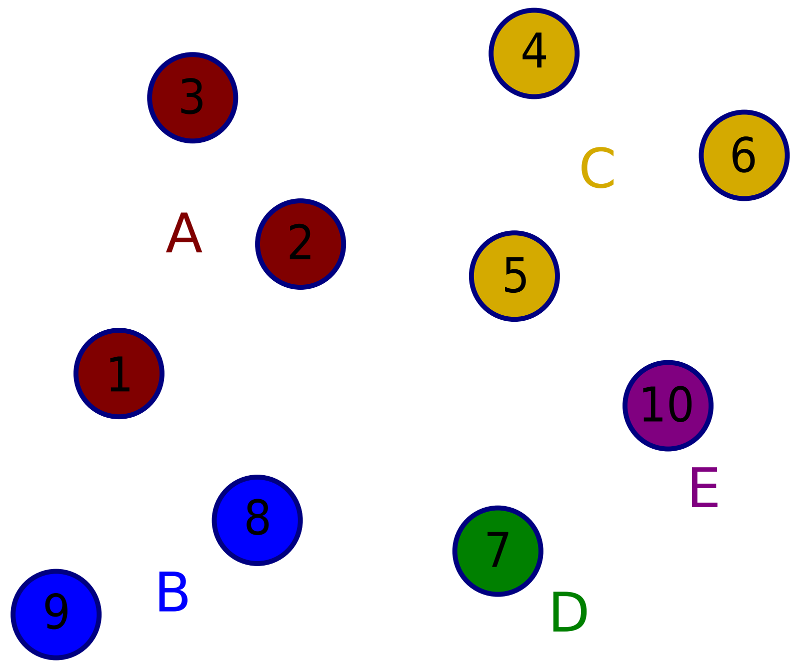
The parameters of the model are the $K \times 1$ class proportion vector $\mathbf{p} = (p_1, \ldots, p_K)$ and the $K \times K$ link probabillity matrix $\boldsymbol{\rho}$ where $\rho_{k\ell} \in [0, 1]$.

Latent Class Models



If we observe a new node, which class do we assign it to?

Nonparametric Latent Class Models



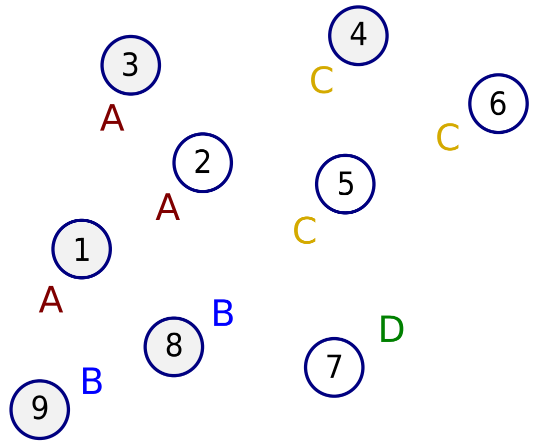
The new node could belong to one of the previously observed classes, but might also belong to an as yet unobserved class.

This motivates *nonparametric* models, where the number of observed classes can grow with the number of nodes.[3]

[3]Nonparametric models are sometimes called *infinite* models since they allow infinitely many classes, features, parameters, etc.

Nonparametric Latent Class Models

Infinite Relational Model (Kemp et al 2006)



Each node v_i has a hidden class $c_i \in \{1, \ldots, \infty\}$

For all i:

$$c_i | c_1, \ldots, c_{i-1} \sim \mathrm{CRP}(\alpha)$$

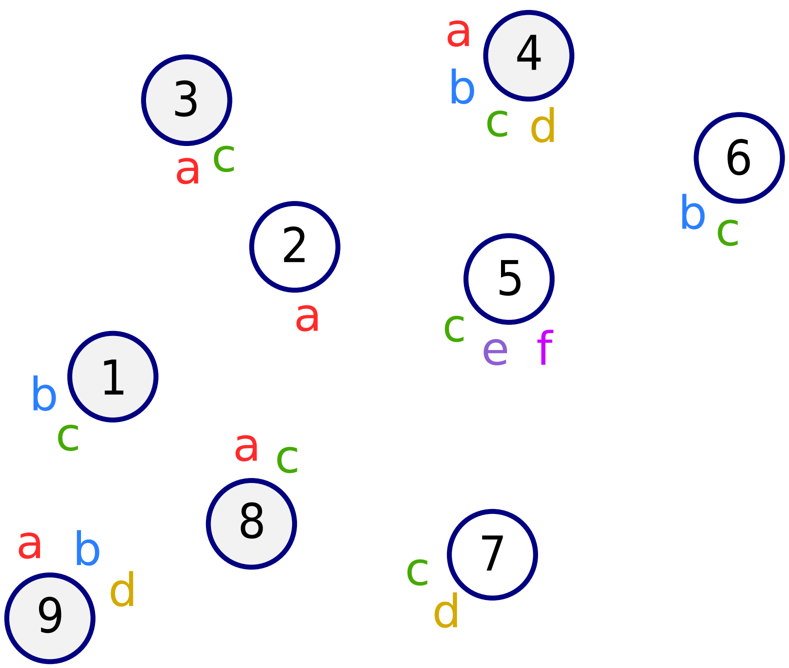
As before, probability of a link between two nodes v_i and v_j depends on their classes:

$$P(y_{ij} = 1 | c_i = k, c_j = \ell) = \rho_{k\ell}$$

Note that $\boldsymbol{\rho}$ is an infinitely large matrix, but if we give each element a beta prior we can integrate it out.

Inference done via MCMC. Fairly straightforward to implement.

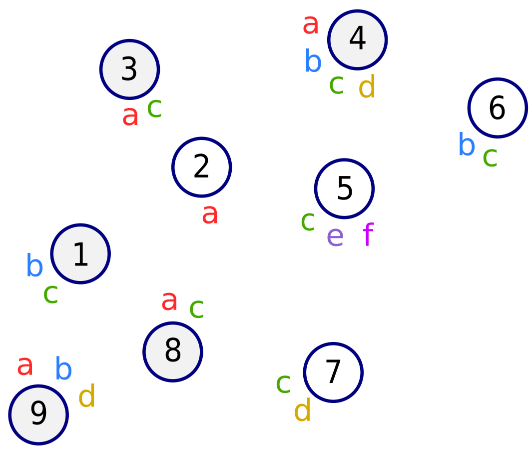
Latent Feature Models



- Each node posses some number of latent features.
- Alternatively we can think of this model as capturing *overlapping clusters or communities*
- The link probability depends on the latent features of the two nodes.
- The model should be able to accommodate a potentially unbounded (infinite) number of latent features.

Latent Feature Models

Nonparametric Latent Feature Relational Model (Miller et al 2010)



Let $z_{ik} = 1$ denote whether node i has feature k

The latent binary matrix $\mathbf{Z}$ is drawn from an IBP distribution: $\mathbf{Z}|\alpha \sim \mathrm{IBP}(\alpha)$

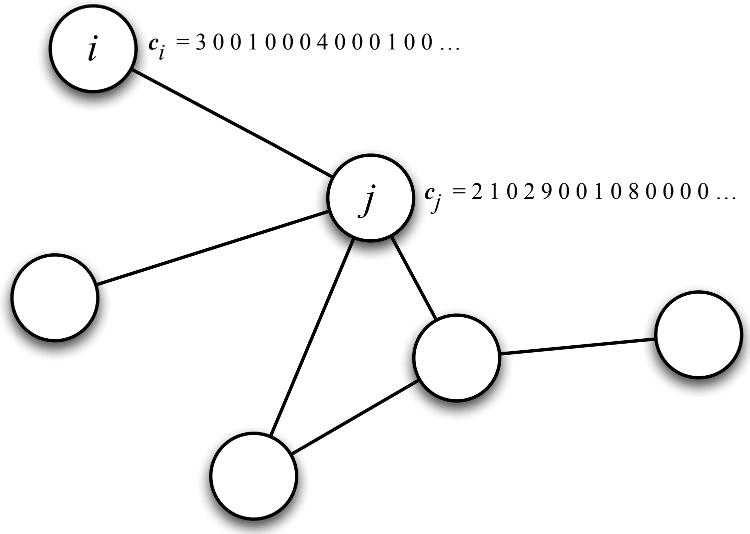
The elements of the parameter matrix $\mathbf{W}$ are drawn iid from: $w_{k\ell} \sim \mathrm{N}(0, \sigma^2)$

The link probability is:

$$P(y_{ij} = 1|\mathbf{W}, \mathbf{Z}) = \sigma\left(\sum_{k,\ell} z_{ik}\, z_{j\ell}\, w_{k\ell}\right)$$

where $\sigma(\cdot)$ is the logistic (sigmoid) function.

Infinite Latent Attribute model for network data



- Each object has some number of latent attributes
- Each attribute can have some number of discrete values
- Probability of a link between object i and j depends on the attributes of i and j:

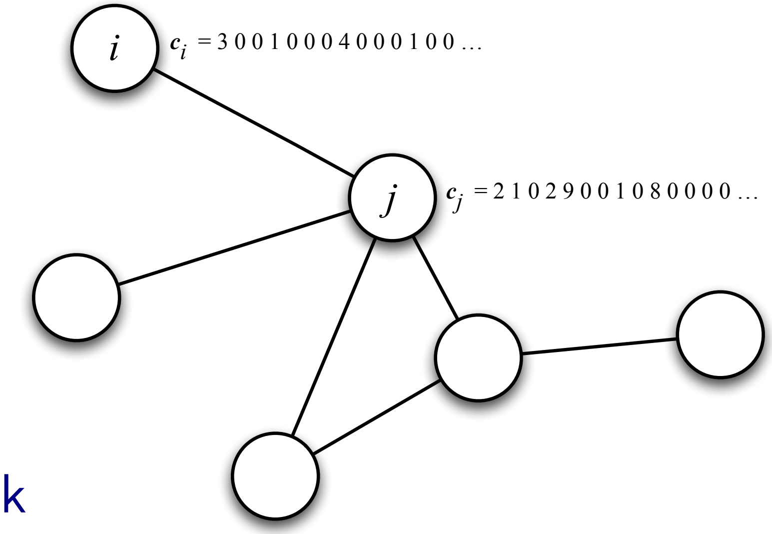
$$P(y_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j, \mathbf{C}, \mathbf{W}) = \sigma\Big(\sum_m z_{im} z_{jm} w^{(m)}_{c_i^m c_j^m} + s\Big)$$

- Potentially unbounded number of attributes, and values per attribute[4]
- Generalises both the IRM and the NLFRM.

(w/ Konstantina Palla, David Knowles, 2012)

[4] An IBP is used for the attribute matrix, $\mathbf{Z}$ and a CRP for the values of each attribute, $\mathbf{C}$

Infinite Latent Attribute model for network data



Example: a student friendship network

- Each student might be involved in some activities or have some features:
  `person_i` has attributes `(College, sport, politics)`
  `person_j` has attributes `(College, politics, religion, music)`

- Each attribute has some values:
  `person_i` = `(College=Trinity, sport=squash, politics=LibDem)`
  `person_j` = `(College=Kings, politics=LibDem, religion=Catholic, music=choir)`

- Prob. of link between person i and j depends on their attributes and values.
- The attributes and values are *not observed*—they are learned from the network.

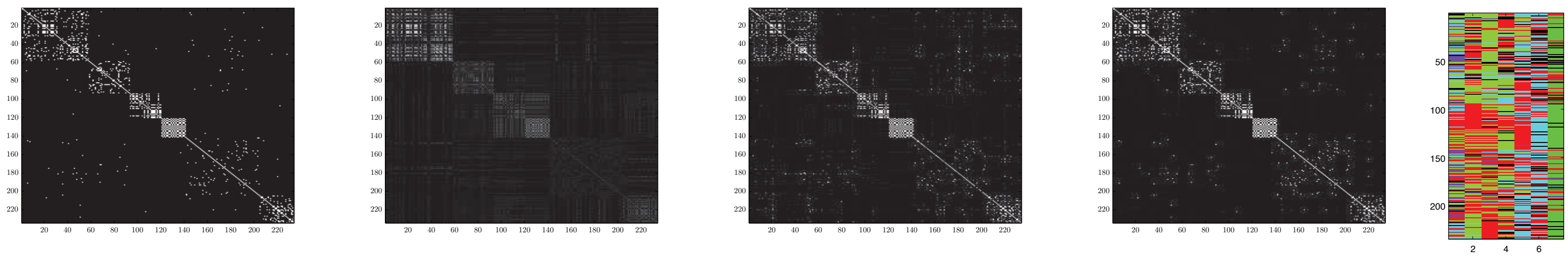
Infinite Latent Attribute: Results

Table 1. NIPS coauthorship network results. The best results are highlighted in bold where statistically significant.

	IRM	LFIRM	ILA ($M = 6$)	ILA ($M = \infty$)
Train error	0.0427 ± 0.0009	0.0197 ± 0.0052	0.0086 ± 0.0005	$\mathbf{0.0058 \pm 0.0005}$
Test error	0.0440 ± 0.0014	0.0228 ± 0.0041	0.0141 ± 0.0012	$\mathbf{0.0106 \pm 0.0007}$
Test log likelihood	-0.0859 ± 0.0043	-0.0547 ± 0.0079	$\mathbf{-0.0322 \pm 0.0058}$	$\mathbf{-0.0318 \pm 0.0094}$

Table 2. Gene interaction network results. The best results are highlighted in bold where statistically significant.

	IRM	LFIRM	ILA ($M = 6$)	ILA ($M = \infty$)
Train error	0.3562 ± 0.0008	0.2603 ± 0.0098	0.2044 ± 0.0066	$\mathbf{0.0248 \pm 0.0010}$
Test error	0.3608 ± 0.0031	0.2661 ± 0.0086	0.2284 ± 0.0077	$\mathbf{0.0735 \pm 0.0047}$
Test log likelihood	-0.4669 ± 0.0097	-0.4223 ± 0.0147	-0.3596 ± 0.0156	$\mathbf{-0.2654 \pm 0.0447}$



(a) True links (b) IRM (c) LIFRM (d) ILA (e) ILA

Figure 3. Predictions for the three models on the NIPS 1-17 coauthorship dataset. In (a), white denotes that two people wrote a paper together, while in (b)-(d), the lighter the entry, the more confident the model is that the corresponding authors would collaborate. In (e), we present the clusters recovered by ILA in the 7 corresponding features. Different colors denote the different subcluster assignments.

Summary

- Probabilistic modelling and Bayesian inference are two sides of the same coin
- Bayesian machine learning treats learning as a probabilistic inference problem
- Bayesian methods work well when the models are flexible enough to capture relevant properties of the data
- This motivates non-parametric Bayesian methods, e.g.:
  - Gaussian processes for **regression and classification**
  - Infinite HMMs for **time series** modelling
  - Indian buffet processes for **sparse matrices** and latent feature modelling
  - Infinite latent attibute model for **network modelling**

Thanks to

Tom Griffiths

Konstantina Palla

David Knowles

Creighton Heaukulani

`http://learning.eng.cam.ac.uk/zoubin`

`zoubin@eng.cam.ac.uk`

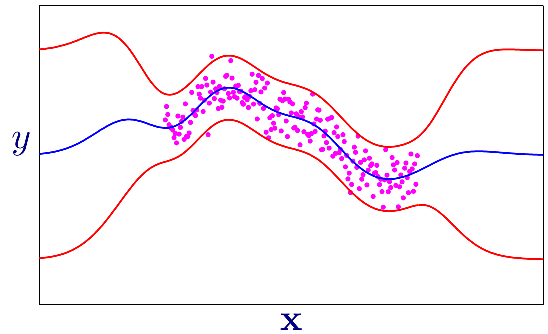
Some References

- Beal, M. J., Ghahramani, Z. and Rasmussen, C. E. (2002) The infinite hidden Markov model. NIPS **14**:577–585.
- Bratieres, S., van Gael, J., Vlachos, A., and Ghahramani, Z. (2010) Scaling the iHMM: Parallelization versus Hadoop. International Workshop on Scalable Machine Learning and Applications (SMLA-10), 1235–1240.
- Griffiths, T.L., and Ghahramani, Z. (2006) Infinite Latent Feature Models and the Indian Buffet Process. NIPS **18**:475–482.
- Griffiths, T.L., and Ghahramani, Z. (2011) The Indian buffet process: An introduction and review. *Journal of Machine Learning Research* **12**(Apr):1185–1224.
- Knowles, D. and Ghahramani, Z. (2007) Infinite Sparse Factor Analysis and Infinite Independent Components Analysis. In *7th International Conference on Independent Component Analysis and Signal Separation (ICA 2007)*. Lecture Notes in Computer Science Series (LNCS) **4666**:381–388.
- Knowles, D.A. and Ghahramani, Z. (2011) Nonparametric Bayesian Sparse Factor Models with application to Gene Expression modelling. *Annals of Applied Statistics* **5**(2B):1534-1552.
- Meeds, E., Ghahramani, Z., Neal, R. and Roweis, S.T. (2007) Modeling Dyadic Data with Binary Latent Factors. NIPS **19**:978–983.
- Stepleton, T., Ghahramani, Z., Gordon, G., Lee, T.-S. (2009) The Block Diagonal Infinite Hidden Markov Model. AISTATS 2009, 552–559.
- van Gael, J., Saatci, Y., Teh, Y.-W., and Ghahramani, Z. (2008) Beam sampling for the infinite Hidden Markov Model. ICML 2008, 1088-1095.
- van Gael, J and Ghahramani, Z. (2010) Nonparametric Hidden Markov Models. In Barber, D., Cemgil, A.T. and Chiappa, S. *Inference and Learning in Dynamic Models*. CUP.

Appendix

Nonlinear regression and Gaussian processes

Consider the problem of nonlinear regression:
You want to learn a function f with error bars from data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$



A Gaussian process defines a distribution over functions $p(f)$ which can be used for Bayesian regression:

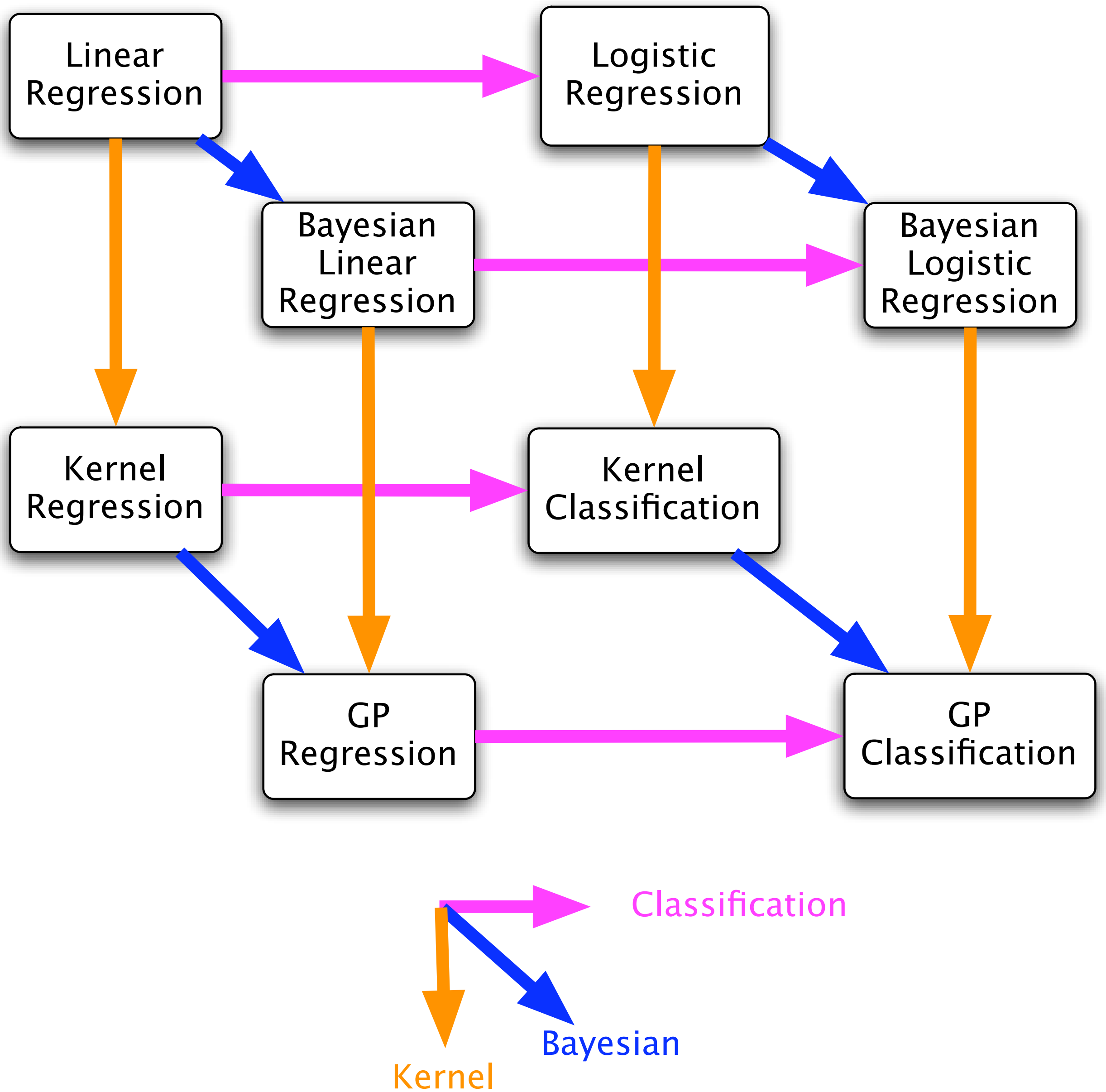
$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

Let $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))$ be an n-dimensional vector of function values evaluated at n points $x_i \in \mathcal{X}$. Note, $\mathbf{f}$ is a random variable.

Definition: $p(f)$ is a Gaussian process if for *any* finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, the marginal distribution over that subset $p(\mathbf{f})$ is multivariate Gaussian.

Excellent textbook: Rasmussen and Williams (2006) and easy to use Matlab code:
`http://www.gaussianprocess.org/gpml/code/`

A picture


Linear Regression
Logistic Regression
Bayesian Linear Regression
Bayesian Logistic Regression
Kernel Regression
Kernel Classification
GP Regression
GP Classification
Classification
Bayesian
Kernel

Nonparametric Binary Matrix Factorization

genes × patients

users × movies

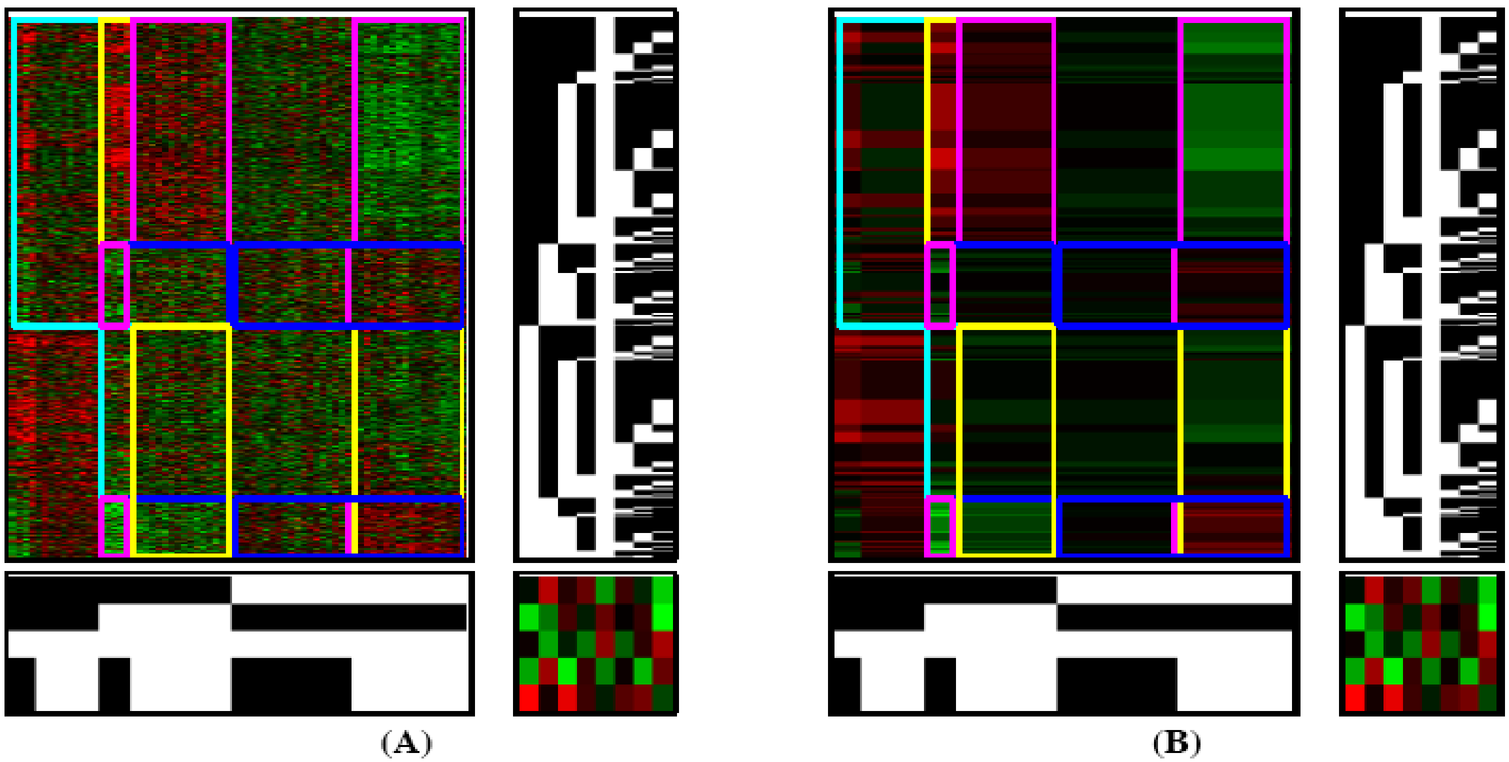


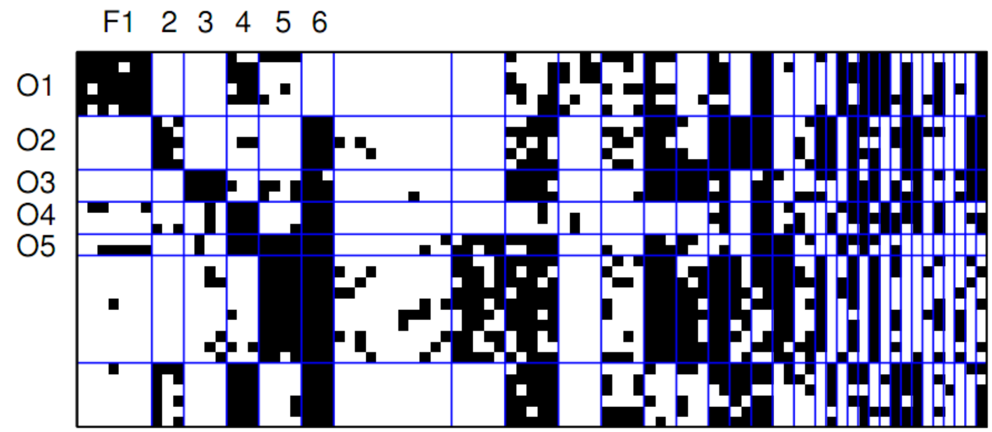
Figure 5: Gene expression results. **(A)** The top-left is $\mathbf{X}$ sorted according to contiguous features in the final $\mathbf{U}$ and $\mathbf{V}$ in the Markov chain. The bottom-left is $\mathbf{V}^\top$ and the top-right is $\mathbf{U}$. The bottom-right is $\mathbf{W}$. **(B)** The same as **(A)**, but the expected value of $\mathbf{X}$, $\hat{\mathbf{X}} = \mathbf{UWV}^\top$. We have hilighted regions that have both u_{ik} and v_{jl} on. For clarity, we have only shown the (at most) two largest contiguous regions for each feature pair.

Meeds et al (2007) Modeling Dyadic Data with Binary Latent Factors.

Nonparametric Latent Class Models

O1 killer whale, blue whale, humpback, seal, walrus, dolphin
O2 antelope, horse, giraffe, zebra, deer
O3 monkey, gorilla, chimp
O4 hippo, elephant, rhino
O5 grizzly bear, polar bear

F1 flippers, strain teeth, swims, arctic, coastal, ocean, water
F2 hooves, long neck, horns
F3 hands, bipedal, jungle, tree
F4 bulbous body shape, slow, inactive
F5 meat teeth, eats meat, hunter, fierce
F6 walks, quadrapedal, ground



Taken from Kemp et al., 2006. Animal clusters, feature clusters, and a sorted matrix showing the relationships between them. The matrix includes seven of the twelve animal clusters and all of the feature clusters.

Network Modelling: Extensions

- Directed networks
- Networks with multiple kinds of relations (edges)
- Scaling to large network datasets
- Using auxiliary information (e.g. observed features of nodes)
- Dynamic networks that evolve over time

We are currently working on many of the above and would welcome potential collaborations.