Modern Bayesian Nonparametrics: Beyond Dirichlet and Gaussian processes

Zoubin Ghahramani

Department of Engineering
University of Cambridge, UK

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

NIPS Workshop
Modern Nonparametric Methods in Machine Learning
Lake Tahoe 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:

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

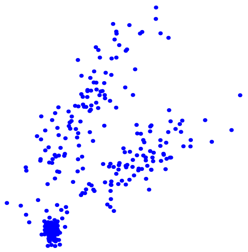
Model Comparison:

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

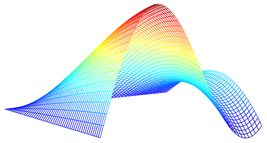
$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

Learning Model Structure

How many clusters in the data?

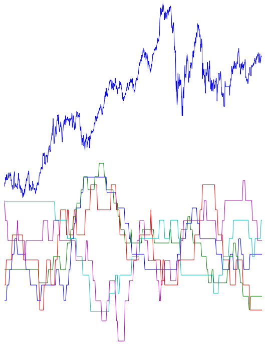


What is the intrinsic dimensionality of the data?



Variable selection: is some variable relevant to predicting another?

What is the order of a dynamical system?

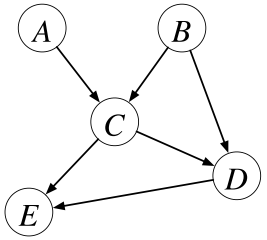


How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY

How many hidden sources in the input?

What is the structure of a graphical model?



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.

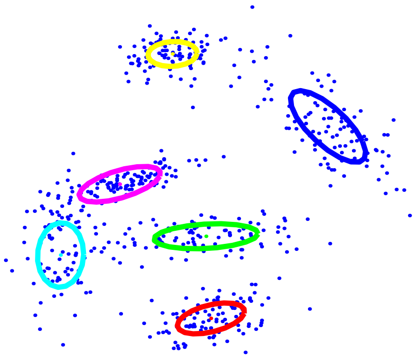
- So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.

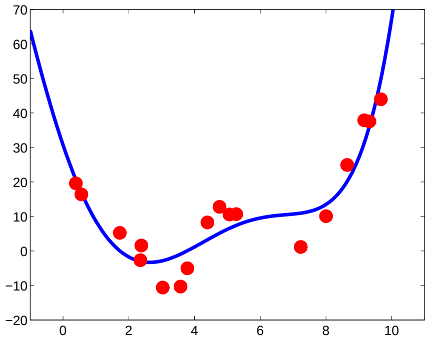
- *Non-parametric models* assume that the data distribution cannot be defined in terms of such a finite set of parameters. But it can often be defined by assuming an *infinite dimensional* θ. Usually we think of θ as a *function*.

- 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, ...

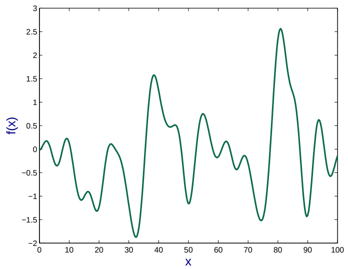
[1] or highly scalable!

Examples of non-parametric models

Parametric	Non-parametric	Application
polynomial regression	Gaussian processes	function approx.
logistic regression	Gaussian process classifiers	classification
mixture models, k-means	Dirichlet process mixtures	clustering
hidden Markov models	infinite HMMs	time series
factor analysis / pPCA / PMF	infinite latent factor models	feature discovery
...		

Gaussian and Dirichlet Processes

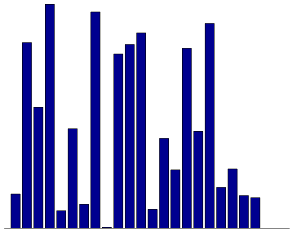
- Gaussian processes define a distribution on functions



$$f \sim \text{GP}(\cdot|\mu, K)$$

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

- Dirichlet processes define a distribution on distributions



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

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.

Gaussian Processes and SVMs

Support Vector Machines and Gaussian Processes

We can write the SVM loss as: $$\min_{\mathbf{f}} \ \frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} + C\sum_i (1 - y_i f_i)_+$$

We can write the negative log of a GP likelihood as: $$\frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} - \sum_i \ln p(y_i|f_i) + c$$

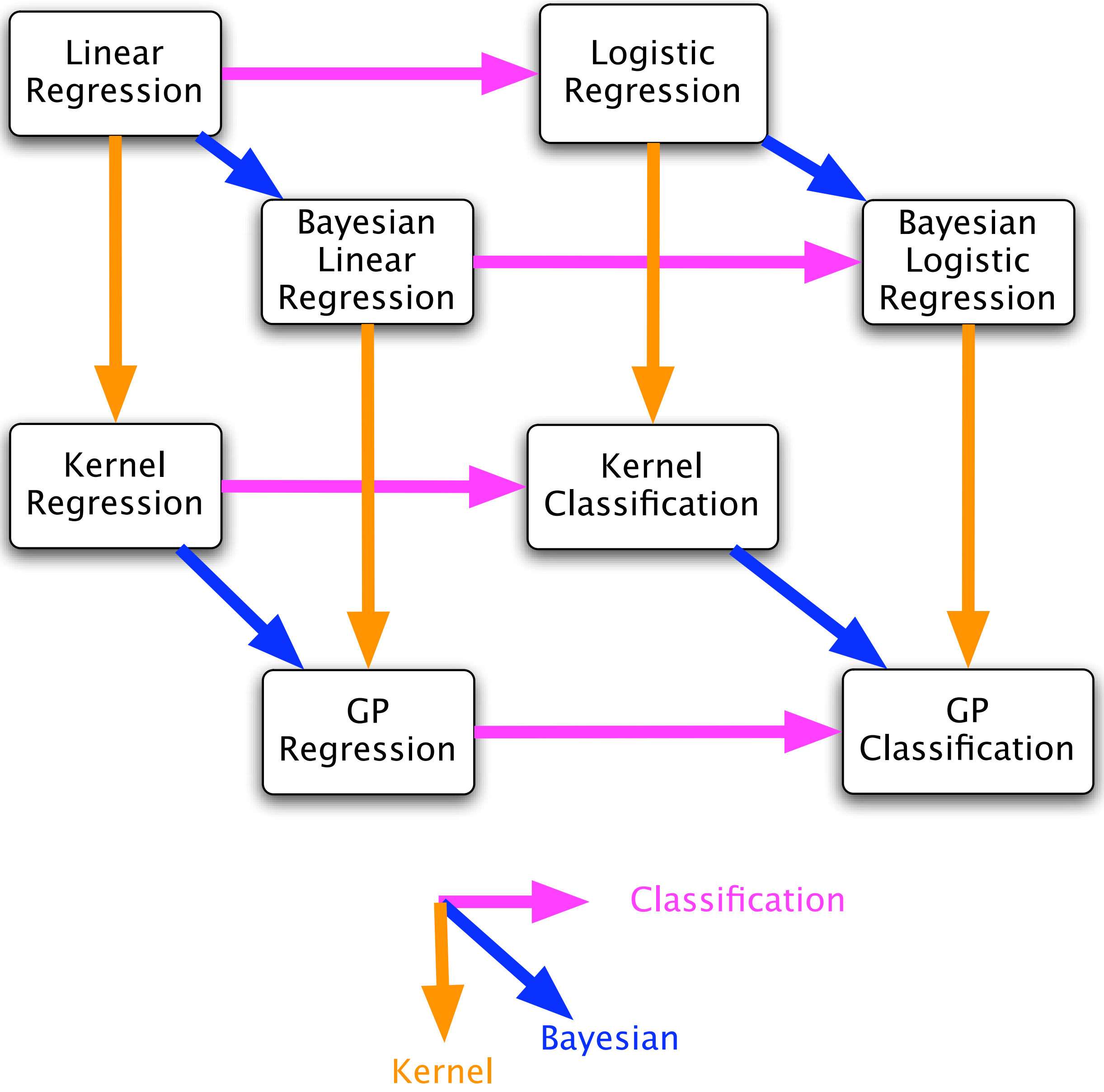
Equivalent? No.

With Gaussian processes we:

- Handle **uncertainty** in unknown function $\mathbf{f}$ by averaging, not minimization.
- Compute $p(y = +1|\mathbf{x}) \neq p(y = +1|\hat{\mathbf{f}}, \mathbf{x})$.
- Can **learn the kernel parameters** automatically from data, no matter how flexible we wish to make the kernel.
- Can **learn the regularization parameter** C without cross-validation.
- Can incorporate **interpretable** noise models and priors over functions, and can sample from prior to get intuitions about the model assumptions.
- We can combine **automatic feature selection** with learning using ARD.

Easy to use Matlab code: `http://www.gaussianprocess.org/gpml/code/`

A picture

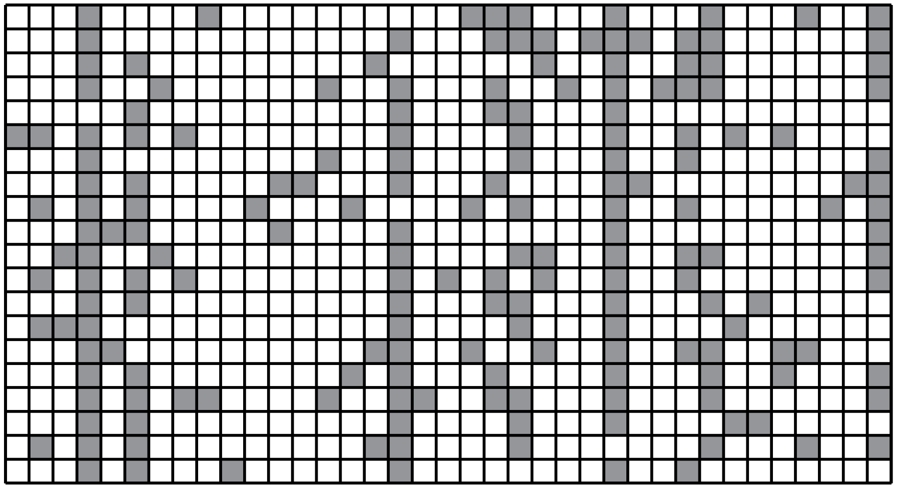

Linear Regression
Logistic Regression
Bayesian Linear Regression
Bayesian Logistic Regression
Kernel Regression
Kernel Classification
GP Regression
GP Classification
Classification
Bayesian
Kernel

Moving beyond GPs and DPs

Bayesian nonparametrics applied to models of other structured objects:

- Sparse Matrices
- Overlapping clusters
- Networks
- Exchangeable Arrays
- Covariances
- Hierarchies

Sparse binary matrices and the Indian buffet process



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

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

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

- 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)

Nonparametric Binary Matrix Factorization

genes × patients
users × movies

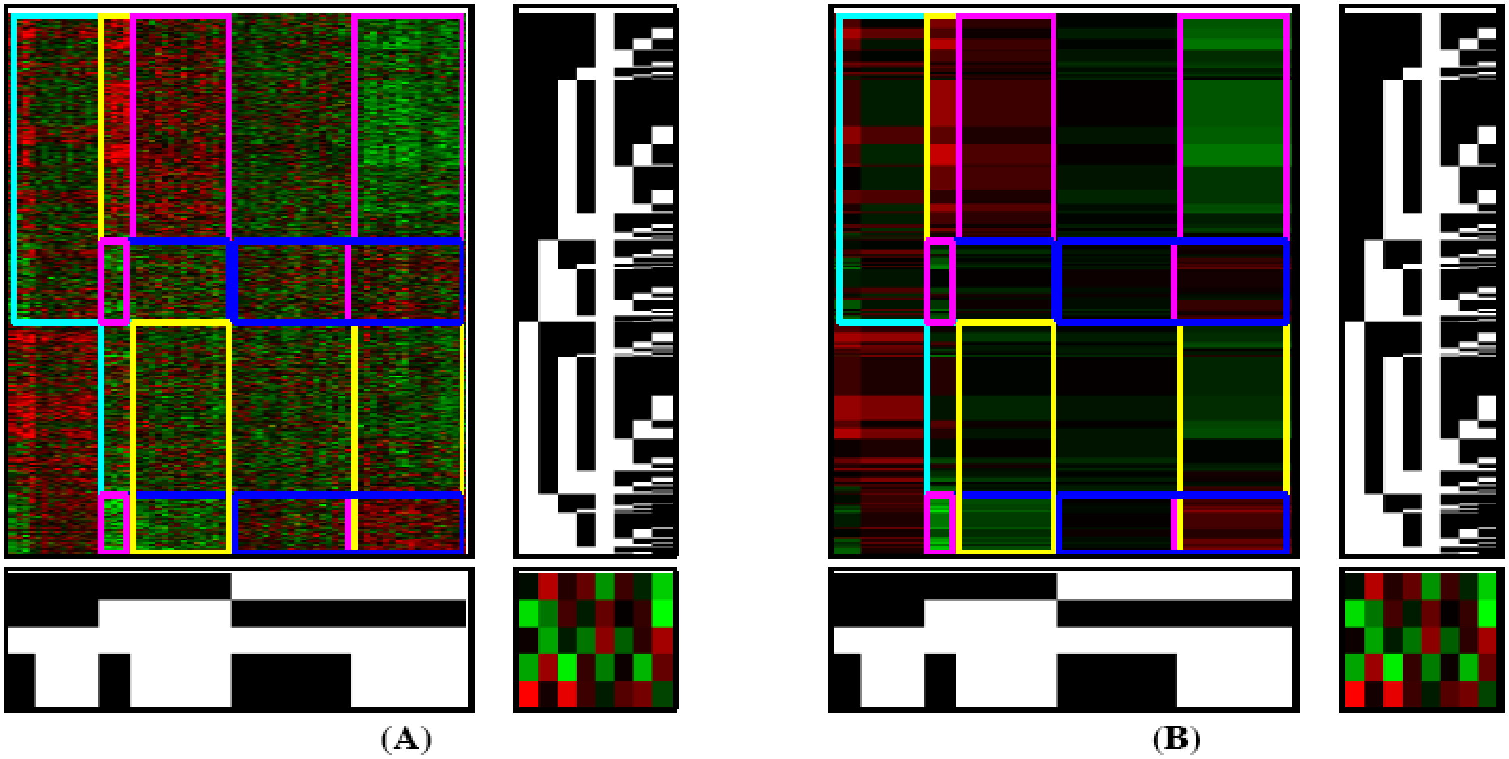
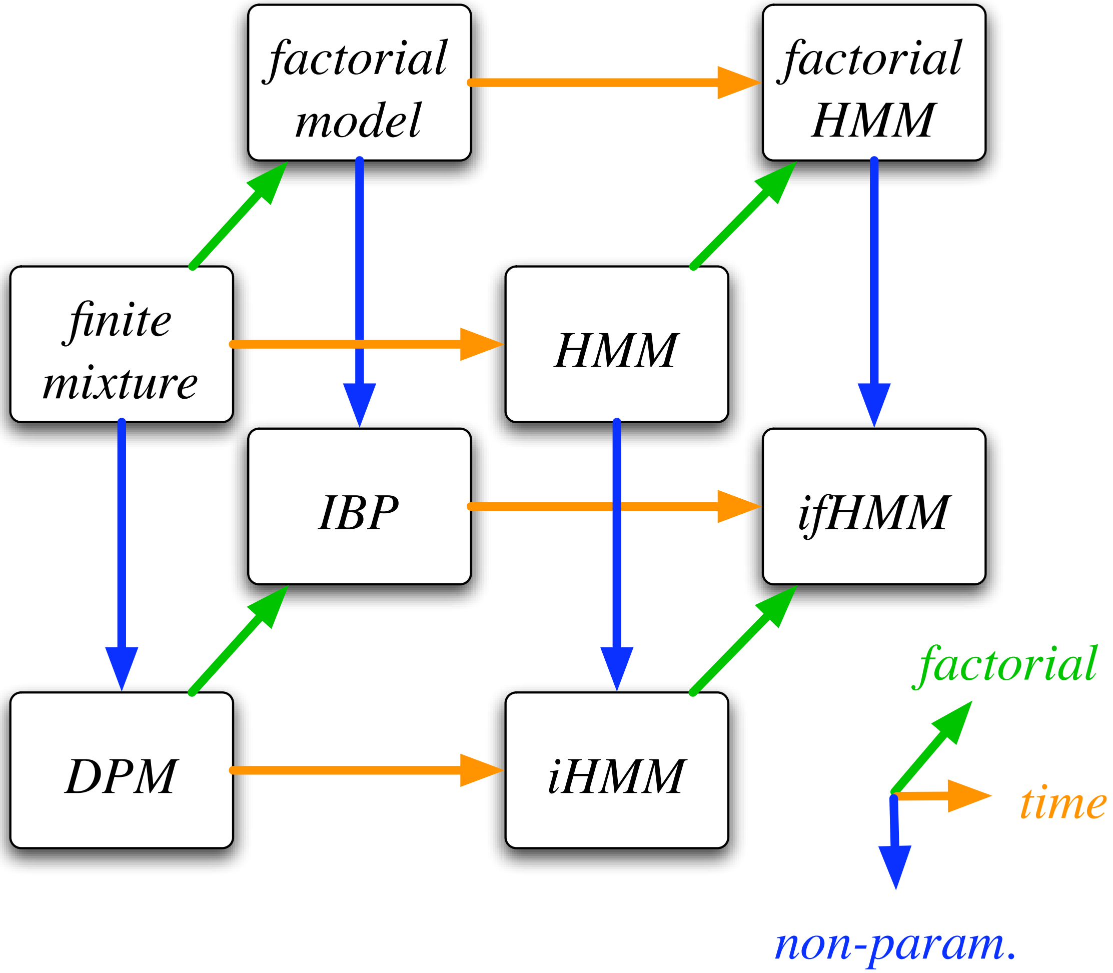


Figure 5: Gene expression results. **(A)** The top-left is $\mathbf{X}$ sorted according to contiguous features in the final $\mathbf{U}$ and $\mathbf{V}$ in the Markov chain. The bottom-left is $\mathbf{V}^\top$ and the top-right is $\mathbf{U}$. The bottom-right is $\mathbf{W}$. **(B)** The same as **(A)**, but the expected value of $\mathbf{X}$, $\hat{\mathbf{X}} = \mathbf{U}\mathbf{W}\mathbf{V}^\top$. 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.

The Big Picture:
Relations between some models



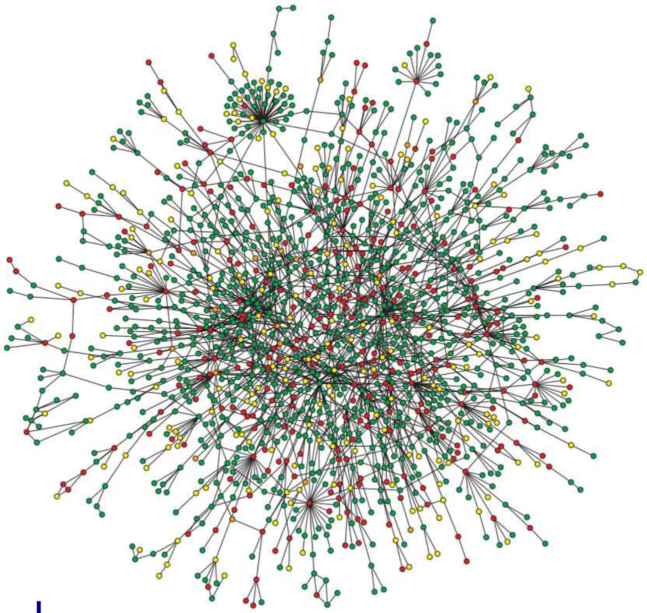
Factorial models allow data points to belong to multiple overlapping clusters simultaneously, or equivalently have a factored state space.

Networks

[very brief, as I am speaking in the Social Network and Social Media Workshop this afternoon]

Modelling Networks

We are interested in modelling networks.



Biological networks: protein-protein interaction networks

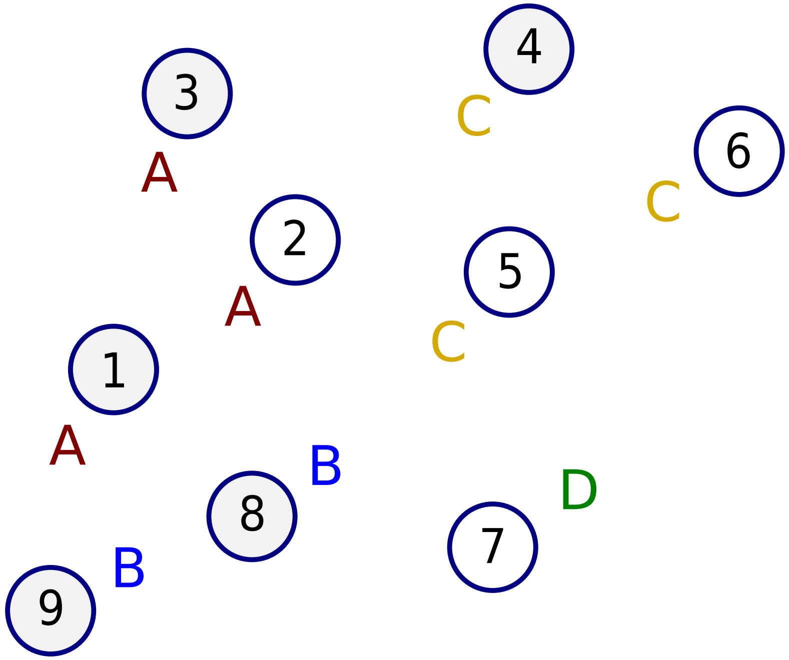
Social networks: friendship networks; co-authorship networks

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

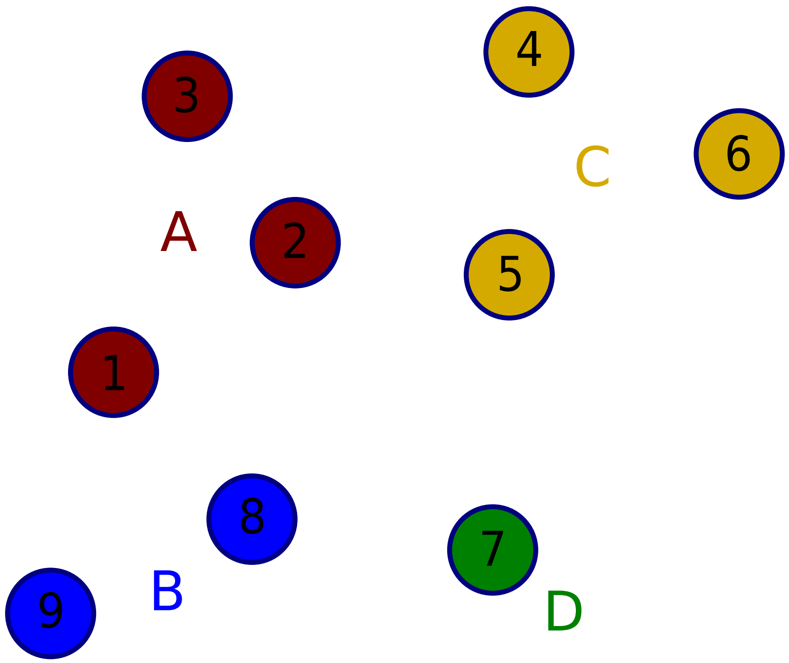
Latent Class Models



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

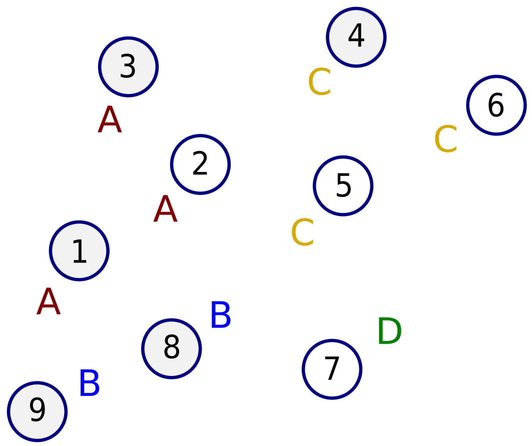


This corresponds to a *clustering* of the nodes.
Such models can be used for *community detection*.

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

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:

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

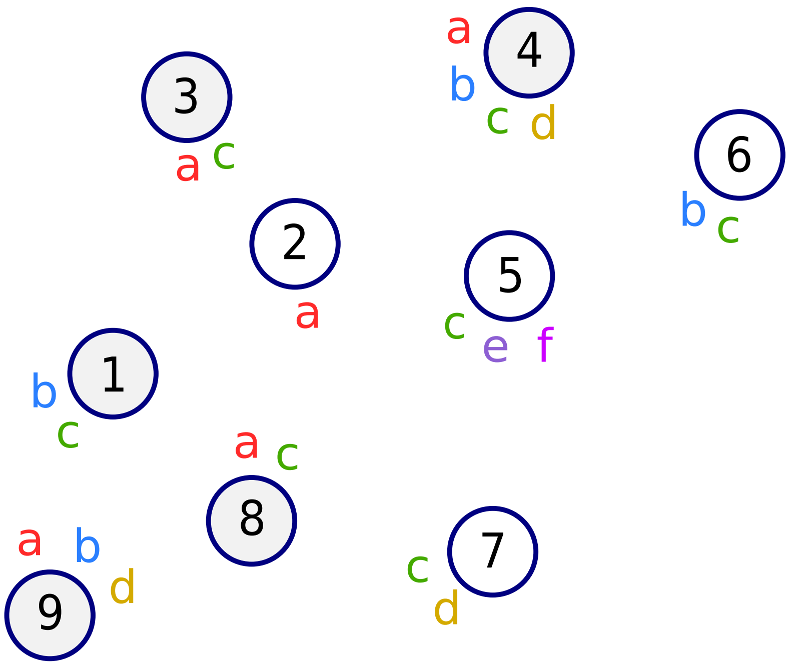
As before, probability of a link between two nodes v_i and v_j depends on their classes:

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

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



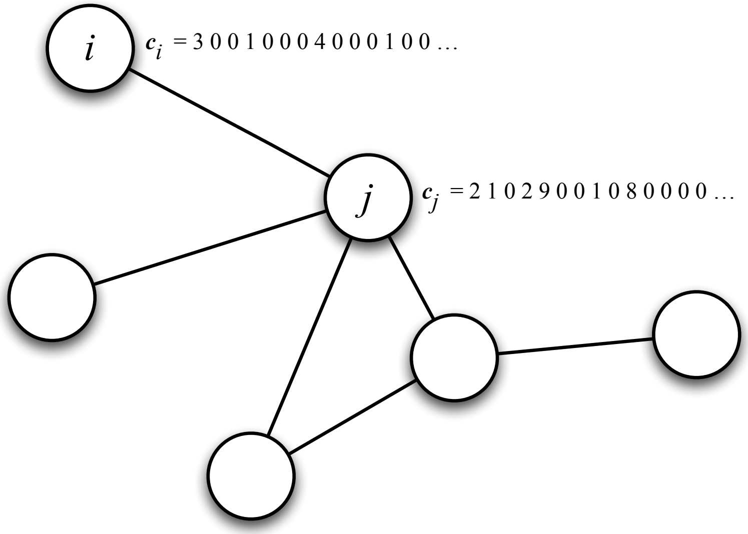
- Each node posses some number of latent features.

- 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.

Infinite Latent Attribute model for network data



i
c_i = 3 0 0 1 0 0 0 4 0 0 0 1 0 0 ...
j
c_j = 2 1 0 2 9 0 0 1 0 8 0 0 0 0 ...

- 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:

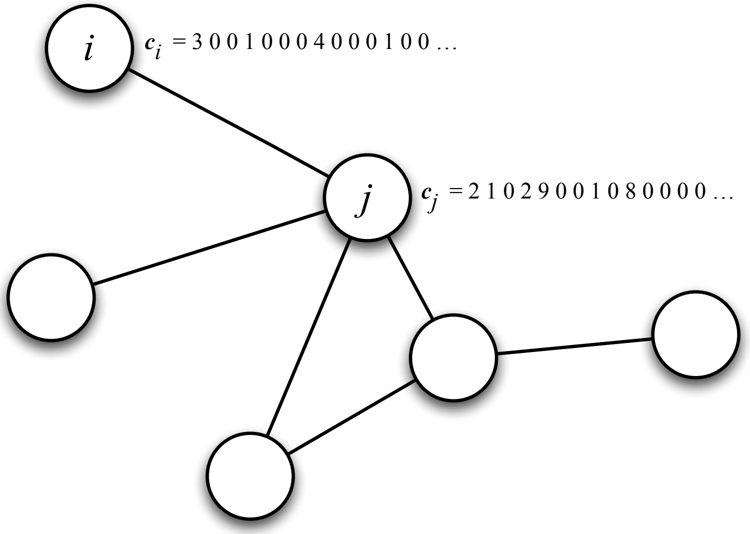
$$P(y_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j, \mathbf{C}, \mathbf{W}) = \sigma\Big(\sum_m z_{im} z_{jm} w^{(m)}_{c_i^m c_j^m} + s \Big)$$

- Potentially unbounded number of attributes, and values per attribute[2]
- Generalises both the IRM and the NLFRM.

(w/ Konstantina Palla, David Knowles, ICML 2012)

[2] 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


i
$c_i = 3\,0\,0\,1\,0\,0\,0\,4\,0\,0\,0\,1\,0\,0\ldots$
j
$c_j = 2\,1\,0\,2\,9\,0\,0\,1\,0\,8\,0\,0\,0\,0\ldots$

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.

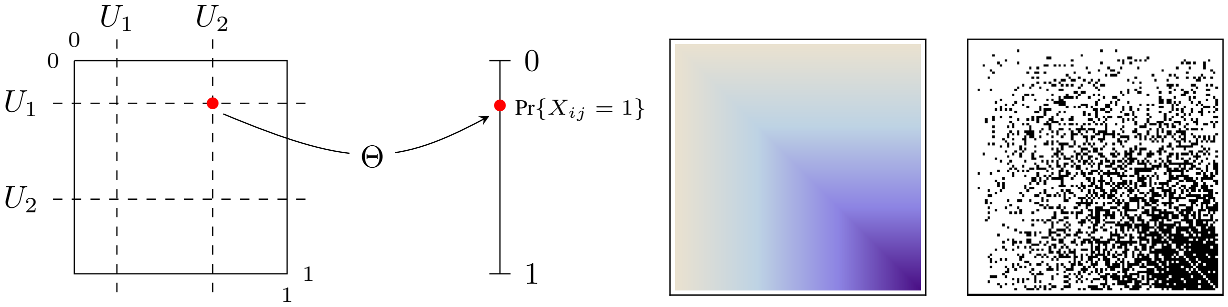
	IRM	LFIRM	ILA ($M = 6$)	ILA ($M = \infty$)
Train error	0.3562 ± 0.0008	0.2603 ± 0.0098	0.2044 ± 0.0066	$\mathbf{0.0248 \pm 0.0010}$
Test error	0.3608 ± 0.0031	0.2661 ± 0.0086	0.2284 ± 0.0077	$\mathbf{0.0735 \pm 0.0047}$
Test log likelihood	-0.4669 ± 0.0097	-0.4223 ± 0.0147	-0.3596 ± 0.0156	$\mathbf{-0.2654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

IRM: (Kemp et al 2006)
LFIRM: (Miller et al 2010)

Exchangeable Arrays

Exchangeable arrays: An array $X = (X_{ij})_{i,j \in \mathbb{N}}$ is called an exchangeable array if $(X_{ij}) \overset{d}{=} (X_{\pi(i)\pi(j)})$ for every $\pi \in S_\infty$.

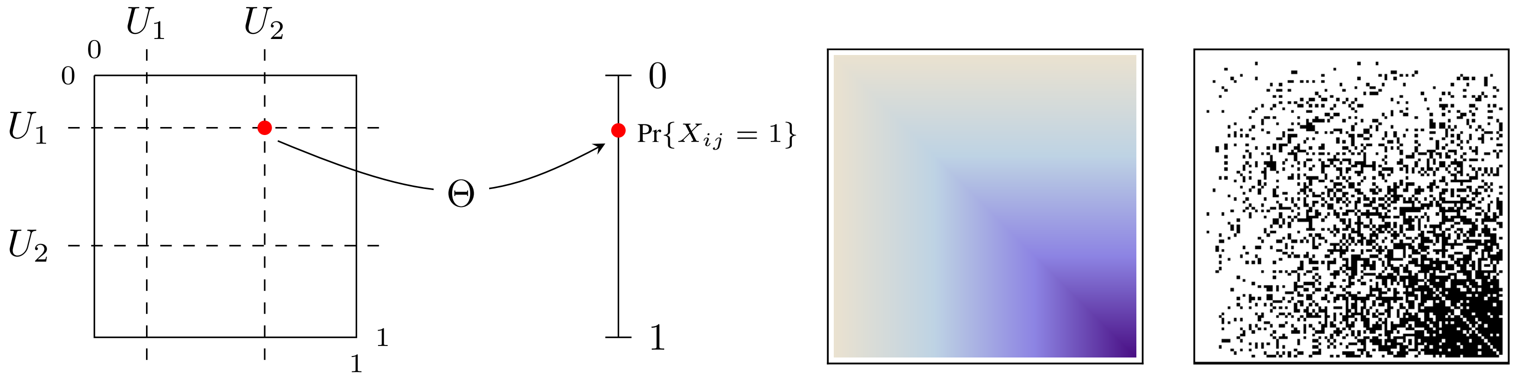
Aldous-Hoover Theorem:
A random matrix (X_{ij}) is exchangeable if and only if there is a random (measurable) function $F : [0,1]^3 \to X$ such that $(X_{ij}) \overset{d}{=} (F(U_i, U_j, U_{ij}))$ for every collection $(U_i)_{i \in \mathbb{N}}$ and $(U_{ij})_{i \leq j \in \mathbb{N}}$ of i.i.d. Uniform$[0,1]$ random variables, where $U_{ji} = U_{ij}$ for $j < i \in \mathbb{N}$.



Interpretation:
Any model of matrices, arrays (or graphs) where the order of rows and columns (nodes) is irrelevant can be expressed by assuming *latent variables* associated with each row and column, and a *random function* mapping these latent variables to the observations.

Random Function Model

We develop a nonparametric probabilistic model for arrays and graphs that makes explicit the Aldous Hoover representation:



$$\begin{aligned} \Theta &\sim \mathrm{GP}(0, \kappa) && (1) \\ U_1, U_2, \ldots &\overset{\text{iid}}{\sim} \mathrm{Uniform}[0, 1] && (2) \\ W_{ij} &= \Theta(U_i, U_j) && (3) \\ X_{ij} &\sim P[\cdot | W_{ij}] && (4) \end{aligned}$$

(w/ James Lloyd, Dan Roy, Peter Orbanz, NIPS 2012)

Random Function Model

The random function model can be related to a number of existing models for matrices, arrays/tensors, and graphs.

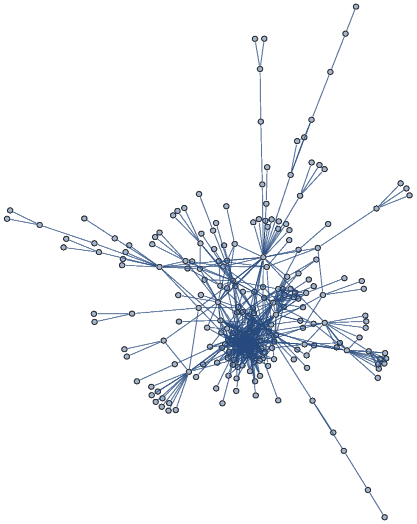
Graph data

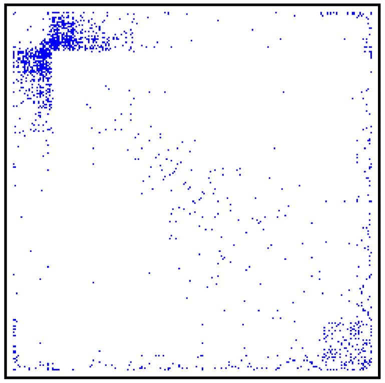
Random function model	Θ	$\sim$	$\mathcal{GP}\left(0, \kappa\right)$
Latent class	W_{ij}	$=$	$m_{U_i U_j}$ where $U_i \in \{1, \ldots, K\}$
IRM	W_{ij}	$=$	$m_{U_i U_j}$ where $U_i \in \{1, \ldots, \infty\}$
Latent distance	W_{ij}	$=$	$-\lvert U_i - U_j \rvert$
Eigenmodel	W_{ij}	$=$	$U_i' \Lambda U_j$
LFRM	W_{ij}	$=$	$U_i' \Lambda U_j$ where $U_i \in \{0, 1\}^\infty$
ILA	W_{ij}	$=$	$\sum_d \mathbb{I}_{U_{id}} \mathbb{I}_{U_{jd}} \Lambda^{(d)}_{U_{id} U_{jd}}$ where $U_i \in \{0, \ldots, \infty\}^\infty$
SMGB	Θ	$\sim$	$\mathcal{GP}\left(0, \kappa_1 \otimes \kappa_2\right)$

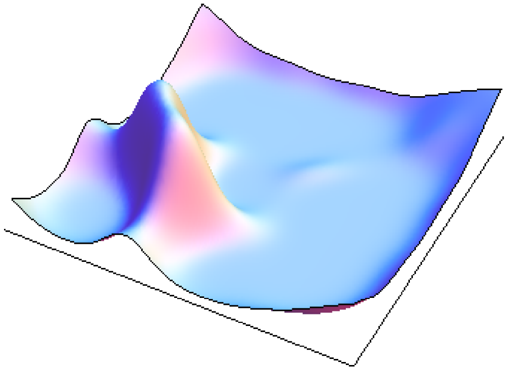
Real-valued array data

Random function model	Θ	$\sim$	$\mathcal{GP}\left(0, \kappa\right)$
Mondrian process based	Θ	$=$	piece-wise constant random function
PMF	W_{ij}	$=$	$U_i' V_j$
GPLVM	Θ	$\sim$	$\mathcal{GP}\left(0, \kappa \otimes \delta\right)$

Random Function Model: Results





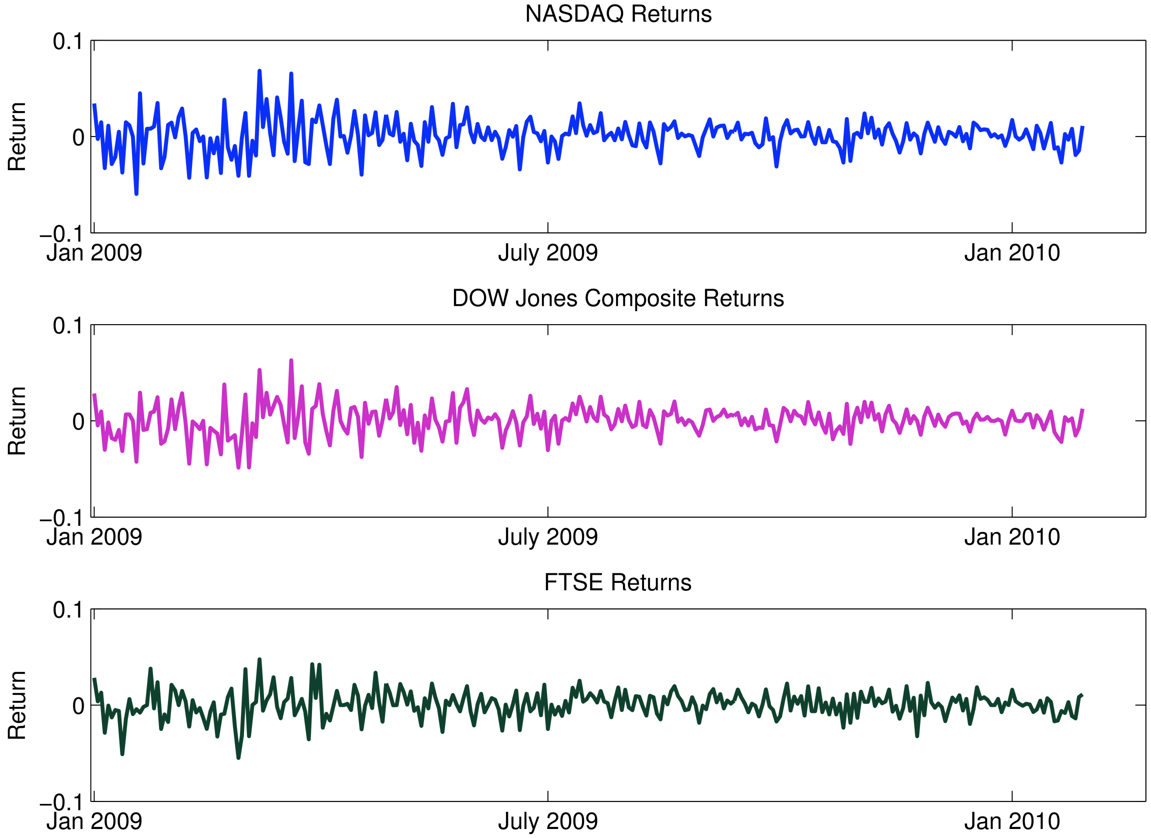


AUC results

Data set	High school			NIPS			Protein		
Latent dimensions	1	2	3	1	2	3	1	2	3
PMF	0.747	0.792	0.792	0.729	0.789	0.820	0.787	0.810	0.841
Eigenmodel	0.742	0.806	0.806	0.789	0.818	0.845	0.805	0.866	0.882
GPLVM	0.744	0.775	0.782	0.888	0.876	0.883	0.877	0.883	0.873
RFM	**0.815**	**0.827**	**0.820**	**0.907**	**0.914**	**0.919**	**0.903**	**0.910**	**0.912**

Covariance Matrices

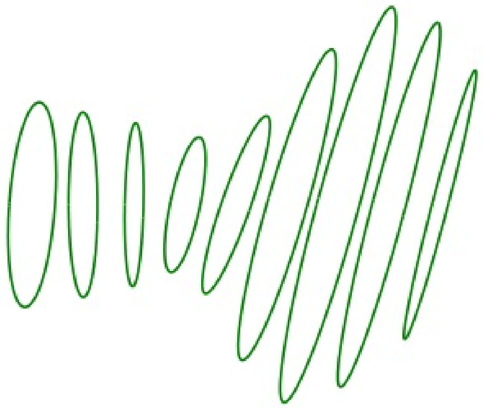
Consider the problem of modelling a covariance matrix Σ that can change as a function of time, $\Sigma(t)$, or other input variables $\Sigma(x)$. This is a widely studied problem in *Econometrics*.



Models commonly used are multivariate GARCH, and multivariate stochastic volatility models, but these only depend on t, and generally don't scale well.

Generalised Wishart Processes for Covariance modelling

Modelling time- and spatially-varying covariance matrices. Note that covariance matrices have to be symmetric positive (semi-)definite.



If $\mathbf{u}_i \sim \mathcal{N}$, then $\Sigma = \sum_{i=1}^{\nu} \mathbf{u}_i \mathbf{u}_i^\top$ is s.p.d. and has a Wishart distribution.

We are going to generalise Wishart distributions to be dependent on time or other inputs, making a nonparametric Bayesian model based on Gaussian Processes (GPs).

So if $\mathbf{u}_i(t) \sim \mathrm{GP}$, then $\Sigma(t) = \sum_{i=1}^{\nu} \mathbf{u}_i(t) \mathbf{u}_i(t)^\top$ defines a **Wishart process**.

This is the simplest form, many generalisations are possible.
Also closely linked to **Copula processes**.

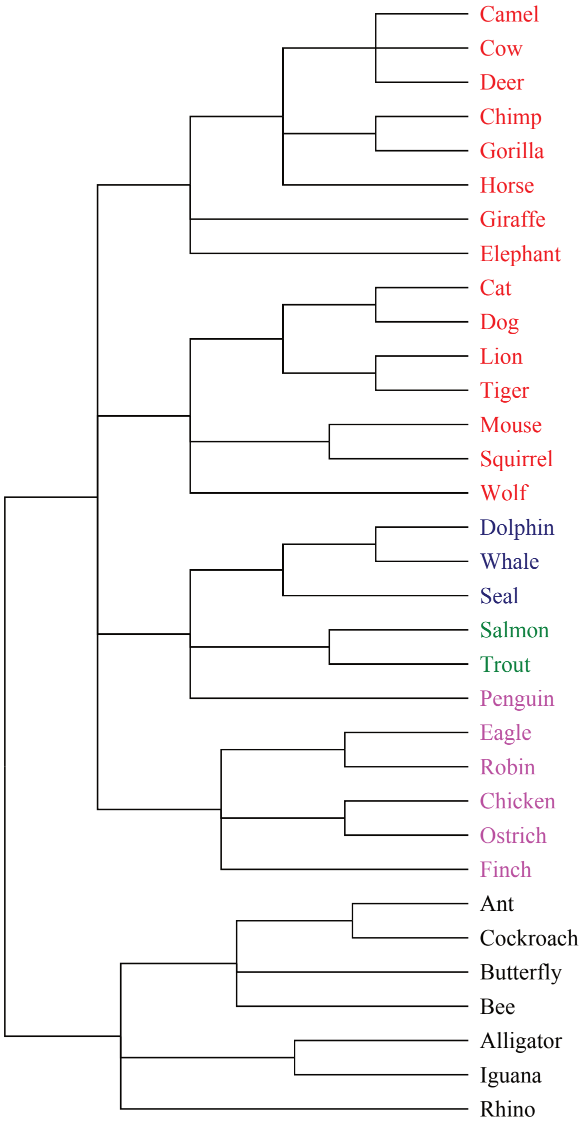
(w/ Andrew Wilson, NIPS 2010, UAI 2011)

Hierarchies

- true hierarchies

- parameter tying

- visualisation and interpretability



Dirichlet Diffusion Trees (DDT)

(Neal, 2001)

In a DPM, parameters of one mixture component are independent of other components – this lack of structure is potentially undesirable.

A DDT is a generalization of DPMs with hierarchical structure between components.

To generate from a DDT, we will consider data points $x_1, x_2, \ldots$ taking a random walk according to a Brownian motion Gaussian diffusion process.

- $x_1(t) \sim$ Gaussian diffusion process starting at origin $(x_1(0) = 0)$ for unit time.
- $x_2(t)$ also starts at the origin and follows x_1 but diverges at some time τ, at which point the path followed by x_2 becomes independent of x_1's path.
- $a(t)$ is a divergence or hazard function, e.g. $a(t) = 1/(1-t)$. For small dt:

  $$P(x_i \,\text{diverges at time}\, \tau \in (t, t+dt)) = \frac{a(t)dt}{m}$$

  where m is the number of previous points that have followed this path.
- If x_i reaches a branch point between two paths, it picks a branch in proportion to the number of points that have followed that path.

Dirichlet Diffusion Trees (DDT)

Generating from a DDT:

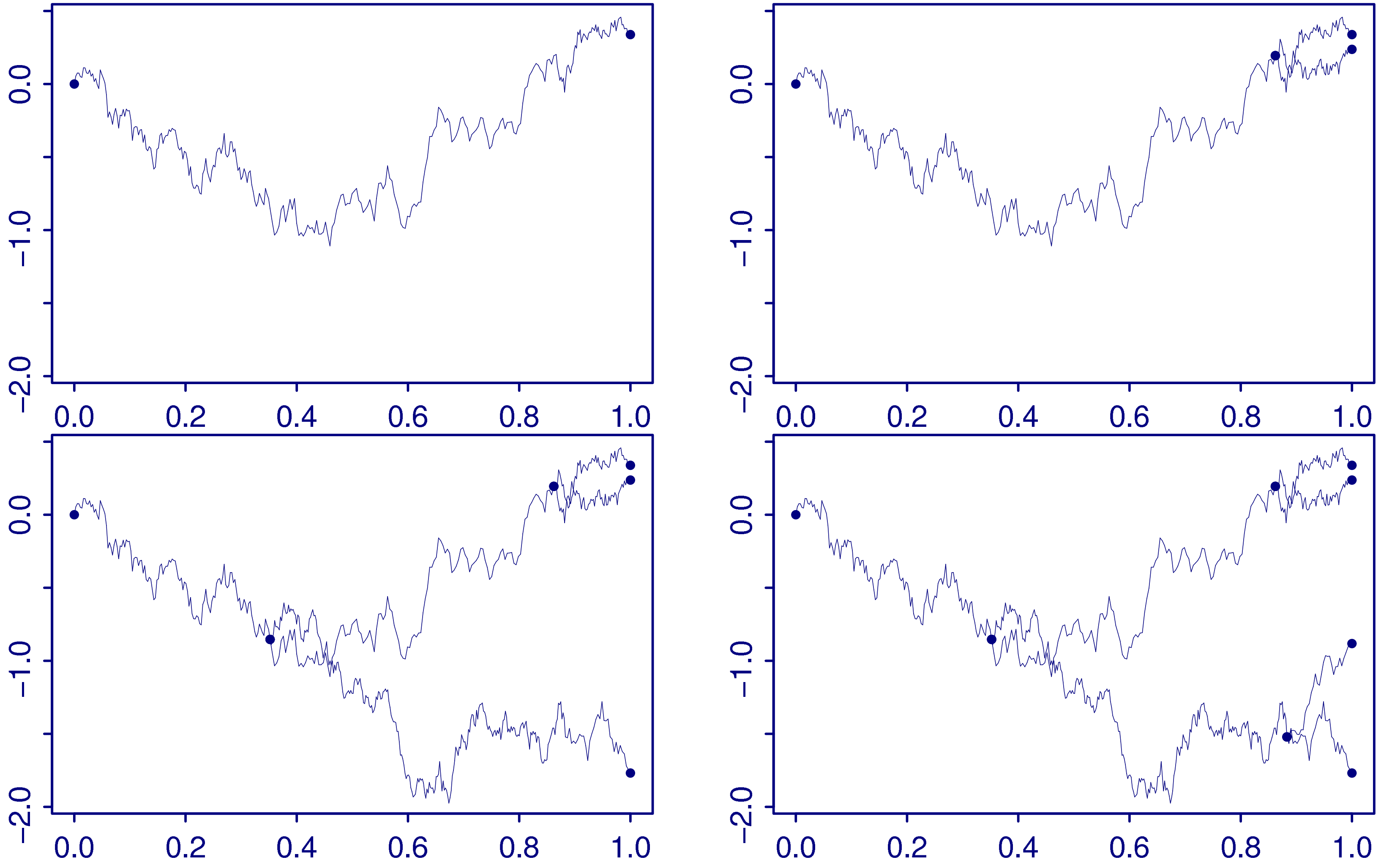


Figure from (Neal 2001)

Pitman-Yor Diffusion Trees

Generalises a DDT, but at a branch point, the probability of following each branch is given by a Pitman-Yor process:

$$P(\text{following branch } k) = \frac{b_k - \alpha}{m + \theta},$$
$$P(\text{diverging}) = \frac{\theta + \alpha K}{m + \theta},$$

to maintain exchangeability the probability of diverging also has to change.

- naturally extends DDTs ($\theta = \alpha = 0$) to arbitrary non-binary branching
- infinitely exchangeable over data
- prior over structure is the most general Markovian consistent and exchangeable distribution over trees (McCullagh et al 2008)

(w/ Knowles UAI 2011)

Pitman-Yor Diffusion Tree: Results

$N_{\text{train}} = 200$, $N_{\text{test}} = 28$, $D = 10$ Adams et al. (2008)

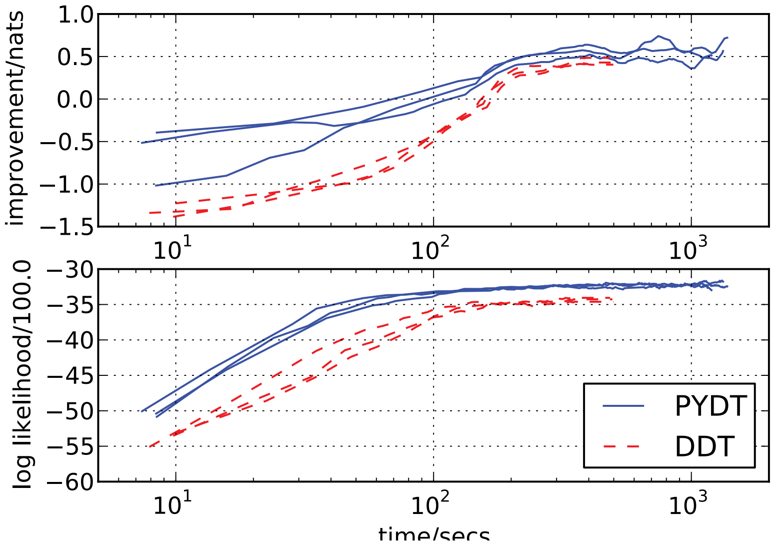
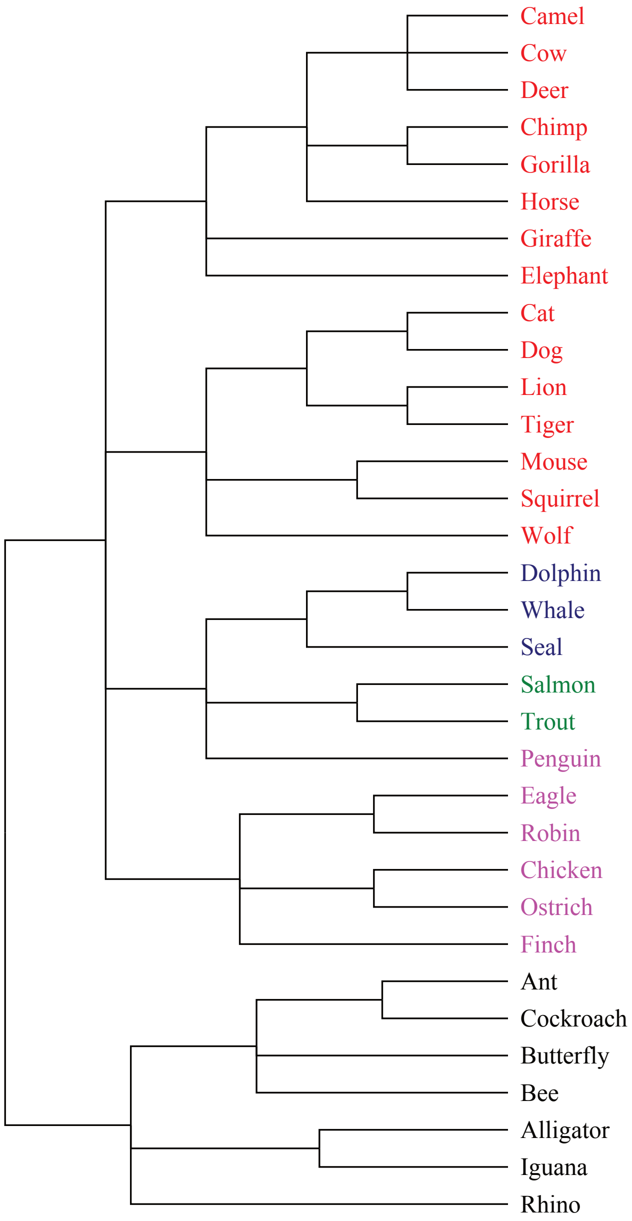


Figure: Density modeling of the $D = 10$, $N = 200$ macaque skull measurement dataset of Adams et al. (2008). *Top*: Improvement in test predictive likelihood compared to a kernel density estimate. *Bottom*: Marginal likelihood of current tree. The shared x-axis is computation time in seconds.



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**
  - Dirichlet process mixtures for **clustering**
  - Indian buffet processes for **sparse matrices** and latent feature modelling
  - Infinite latent attribute model for **network modelling**
  - Aldous-Hoover random function model for **exchangeable arrays**
  - Wishart processes for **covariance modelling**
  - Pitman-Yor diffusion trees for **hierarchical clustering**

Open Challenge:
Bridging Bayesian and classical nonparametrics

- We need more classical theory (consistency, convergence rates, etc) for modern Bayesian nonparametric models

- Some problems are easier to handle in one framework than in the other:
  - Consider density estimation: Kernel density estimation is easy, Dirichlet process mixture modelling is harder
  - On the other hand, for complex modelling problems, Bayesian methods are easier to compose, and naturally avoid the overfitting that can occur where the number of parameters grows with the data.

- We should translate ideas from one framework to the other where possible

- We need more empirical and theoretical comparisons.

Thanks to

Konstantina Palla

David Knowles

Andrew Wilson

Peter Orbanz

Dan Roy

James Lloyd

`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.
- Bru, M. (1991). Wishart processes. *Journal of Theoretical Probability* 4(4):725751.
- Griffiths, T.L., and Ghahramani, Z. (2011) The Indian buffet process: An introduction and review. *Journal of Machine Learning Research* **12**(Apr):1185–1224.
- Heller, K.A., and Ghahramani, Z. (2005) Bayesian Hierarchical Clustering. *International Conference on Machine Learning* (ICML 2005), 297–304.
- Kemp, C., J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. (2006) Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*.
- Knowles, D.A. and Ghahramani, Z. (2011) Pitman-Yor Diffusion Trees. In *Uncertainty in Artificial Intelligence (UAI 2011)*.
- Meeds, E., Ghahramani, Z., Neal, R. and Roweis, S.T. (2007) Modeling Dyadic Data with Binary Latent Factors. NIPS **19**:978–983.
- Miller, K.T., T. L. Griffiths, and M. I. Jordan. (2010) Nonparametric latent feature models for link predictions. In *Advances in Neural Information Processing Systems 22*.
- Neal, R.M. (2000) Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

- Nowicki, K. and Snijders, T. A. B. (2001) Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087.
- Orbanz, P. (2010) Construction of nonparametric Bayesian models from parametric Bayes equations. In *Advances in Neural Information Processing Systems 22*, 2010.
- Palla, K., Knowles, D.A., and Ghahramani, Z. (2012) An infinite latent attribute model for network data. In ICML 2012.
- Stepleton, T., Ghahramani, Z., Gordon, G., Lee, T.-S. (2009) The Block Diagonal Infinite Hidden Markov Model. AISTATS 2009, 552–559.
- Teh, Y.W., Jordan, M.I, Beal, M. and Blei, D. (2004) Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.
- Wilson, A.G., and Ghahramani, Z. (2010, 2011) Generalised Wishart Processes. arXiv:1101.0240v1. and UAI 2011
- Wilson, A.G. and Ghahramani, Z. (2010). Copula Processes. In NIPS 2010.
- Wilson, A.G., Knowles, D.A., and Ghahramani, Z. (2011, 2012). Gaussian process regression networks. arXiv (2011) and ICML 2012.
- 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.
- Xu, Y., Heller, K.A., and Ghahramani, Z. (2009) Tree-Based Inference for Dirichlet Process Mixtures. *AISTATS 2009*, 623–630.