Relational Learning and Network Modelling Using Infinite Latent Attribute Models

Konstantina Palla, David A. Knowles, and Zoubin Ghahramani

Abstract—Latent variable models for network data extract a summary of the relational structure underlying an observed network. The simplest possible models subdivide nodes of the network into clusters; the probability of a link between any two nodes then depends only on their cluster assignment. Currently available models can be classified by whether clusters are disjoint or are allowed to overlap. These models can explain a "flat" clustering structure. Hierarchical Bayesian models provide a natural approach to capture more complex dependencies. We propose a model in which objects are characterised by a latent feature vector. Each feature is itself partitioned into disjoint groups (subclusters), corresponding to a second layer of hierarchy. In experimental comparisons, the model achieves significantly improved predictive performance on social and biological link prediction tasks. The results indicate that models with a single layer hierarchy over-simplify real networks.

Index Terms—Machine learning, unsupervised learning, network models

1 Introduction

Network data encoding pairwise relations between objects appears in many fields. For instance, in biology, a protein network connects interacting partners, while in a social network, links between individuals indicate relationships such as friendship. We focus on the most common type of network data—sets of observations represented as an unweighted, undirected graph—in the ensuing discussion. The motivation behind the analysis of these networks is two fold. First, there is a desire to understand the latent structure responsible for the network; what are the features of the proteins that account for the observed interactions and what is the mechanism behind the links or non-links among groups of people. Second, the prediction of "missing" links in the network arises as an important challenge; how likely is it that a pair of proteins interact or that two social network members are friends. A prominent theme in machine learning is the use of latent variable methods, which approach this problem by extracting a simplified summary of the graph and predicting the presence or absence of links based on this latent representation. Latent class and latent feature models are the two most common categories found in the literature.

Latent class models assume that there are a number of clusters (classes) and that each node belongs to a single cluster. Under these models, the link probability between two objects depends only on their cluster assignments. Early work in this category includes the stochastic block model (SB) proposed in Nowicki and Snijders [16]. Instead of assuming a fixed number of clusters, the Infinite Relational Model (IRM) and the Infinite Hidden Relational Model [9], [22] use the Chinese restaurant process [18] to allow a potentially infinite number of clusters. The Mixed Membership Stochastic Block Model [[1], MMSB] increases the expressiveness of the latent class models by allowing mixed membership, associating each object with a distribution over clusters.

Latent feature models increase the flexibility of the generative process by letting each object possess a vector of features and determine the link probabilities based on interactions among the features. In Hoff et al. [7] the link probability between two objects is determined by the similarity of their real-valued feature vectors. Miller et al. [12] uses a vector of binary features which can be interpreted as allowing objects to belong to multiple clusters at the same time. Their model, the Latent Feature Infinite Relational Model (LFRM), assumes that the number of clusters is not known a priori and uses the Indian Buffet Process [[6], IBP] to determine the number of latent clusters.

The limitation of a single cluster membership makes the latent class models less flexible than the latent feature models. As an intuitive example, consider a network of individuals at a collegiate University, in which a link denotes friendship or acquaintance. Here there will be multiple *types* of cluster, for instance colleges, departments and sports teams. A person might be a member of more than one cluster and his cluster-memberships determine his interaction with others. To capture this structure a single membership model, such as the IRM, must introduce a cluster for each possible combination of the types of cluster, which in our example would be to introduce clusters such as 'Gryffindor college, Department of Mathematics, Football'. This results in an exponential explosion of clusters, making learning, inference and generalisation difficult. Latent feature models, e.g., the LFRM, can instead use the feature vector representation to implicitly account for the possible combination of clusters. Though powerful, these models only account for a flat clustering of the objects. In the context of the

- *K. Palla and Z. Ghahramani are with the University of Cambridge, Cambridge, England, United Kingdom. E-mail: konstantina.palla@gmail.com, zoubin@eng.cam.ac.uk.*
- *D. A. Knowles is with Stanford University, Stanford, CA, USA. E-mail: knowles84@gmail.com.*

Manuscript received 16 Sep. 2012; revised 28 Apr. 2014; accepted 3 May 2014. Date of publication 15 May 2014; date of current version 14 Jan. 2015. Recommended for acceptance by R.P. Adams, E. Fox, E. Suddert, and Y.W. Teh. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2014.2324586

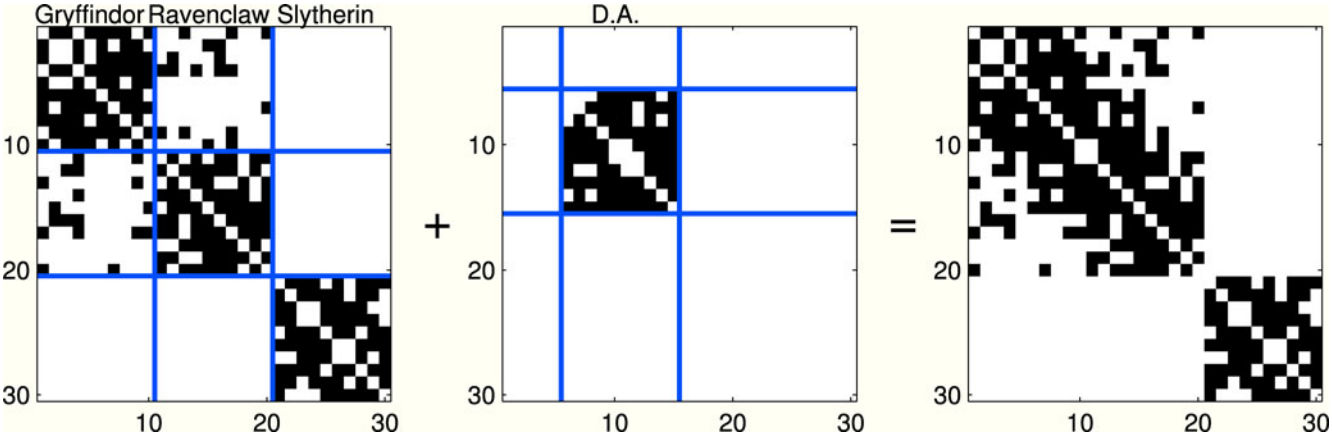


Fig. 1. An illustrative example of the structure modelled by ILA which explains the friendship network at Hogwarts school. *Left:* each student at Hogwarts has the 'house' feature, and belongs to one of three subclusters (houses): Gryffindor, Ravenclaw or Slytherin. These subclusters are disjoint. Students within a house are very likely (0.8) to be friends. Students from Gryffindor are sometimes friends with Ravenclaws (0.2), but Slytherins are never friends with either of the other houses. *Middle*: a second feature with only one subcluster is membership of Dumbledore's Army, which contains Ravenclaws and Gryffindors, and has an even higher probability (0.9) of resulting in a friendship link than being a member of the same house. *Right:* the final friendship network is an element-wise OR of the links resulting from either of the two features (see Section 2.1).

University social network, the 'college' feature might be divided into many different subclusters, such as 'Slytherin college', 'Gryffindor college' etc. The same for 'sport', with subclusters like 'basketball', 'tennis', etc. The LFRM must represent each cluster with a new feature, which will result in feature vectors of greater size with a cost in interpretability. Allowing an explicit representation of the partitioning of each general class into subclasses would provide a more structured representation of the data.

Towards this end, we develop a new nonparametric latent feature model. We use a binary feature vector to indicate the features that an object has. If an object has a particular feature, then the object belongs to a particular subcluster of this feature. Equivalently, we can think of objects having several attributes (features) which have discrete values (the subcluster assignments). Following our university example, a person might have the 'college' attribute and belong to the 'Gryffindor college' subcluster, but cannot simultaneously be a member of another college. We denote our model by ILA for Infinite Latent Attribute model. We use a nonparametric Bayesian approach to simultaneously infer the number of features and number of subclusters inside each feature, while at the same time inferring what features are active for each object, which subcluster it belongs to and how subcluster membership influences the observed interactions. We emphasise that throughout this paper only latent, unobserved attributes are considered: the extension to the case where side information for each node is available is left to future work. We illustrate the concept of latent attributes in Fig. 1 using a hypothetical friendship network at Hogwarts School of Witchcraft and Wizardry. Similar attributes could be envisaged for other domains such as the coauthorship and gene networks considered in the results section. For coauthorship one might expect attributes corresponding to geography, institution, research field or seniority (professors publish more often with postdocs/graduate students than each other). For genetic interaction networks attributes might be gene type (transciption factor, enzyme encoding, micro-RNA encoding) or pathway membership.

The paper is arranged as follows. In Section 2 we describe the generative process for our nonparametric model. Section 3 explains the relationship of our model to several recently proposed models. In Section 4 we derive an algorithm for performing approximate posterior inference, parameter estimation and link prediction. Section 5 discusses the computational cost of our proposed model relative to others. In Section 7 we study our model's performance on one synthetic and three real data sets. Section 8 investigates the convergence and mixing properties of our sampler empirically and Section 9 concludes. In the Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2324586, we show the correctness of our sampler using joint distribution tests, and present further convergence analysis. An earlier version of this paper was Palla et al. [17].

2 Model Description

Let $\mathbf{R}$ be an $N \times N$ binary matrix that contains the links among the objects. In ILA, each object $i = 1, \ldots, N$, is represented by a binary vector of latent feature values, $\mathbf{z}_i$. If there are M features, then $\mathbf{Z}$ is a $N \times M$ binary matrix indicating which features each object has active, with $z_{im} = 1$ if the ith object has feature m and $z_{im} = 0$ otherwise. Let $\mathbf{C}$ be a set of vectors, $\mathbf{C} = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(M)}\}$, that describe the subcluster assignments within each feature, such that $\mathbf{c}^{(m)}$ is a vector of length N where $c_i^{(m)}$ denotes the subcluster the $\mathrm{i\text{-}th}$ object belongs to in the $\mathrm{m\text{-}th}$ feature ($c_i^{(m)}$ is set to 0 if object i does not have feature m). The number of subclusters present in the $\mathrm{m\text{-}th}$ feature, which is also not known a priori, is denoted as $K^{(m)}$, so that $c_i^{(m)} \in \{0, 1, \ldots, K^{(m)}\}$. Finally, let $\mathbf{W}$ be a set of M real-valued weight matrices of size $K^{(m)} \times K^{(m)}$ each, where $w_{kk'}^{(m)}$ is the weight that affects the probability of there being a link from object i to object j, given that object i belongs to subcluster k and object j belongs to subcluster k' of the $\mathrm{m\text{-}th}$ feature.

Given the feature matrix $\mathbf{Z}$, the set of the subcluster assignments $\mathbf{C}$, and the set of the weight matrices $\mathbf{W}$, the

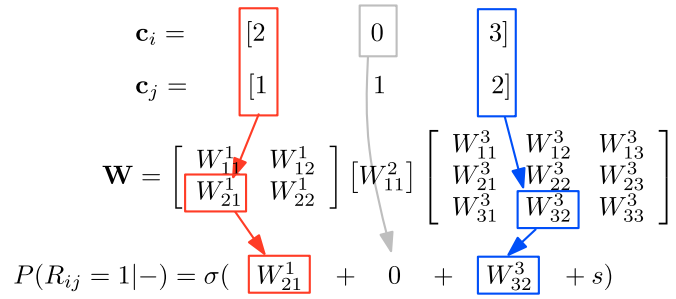


Fig. 2. Diagram of the ILA model. $\mathbf{c}_i$ and $\mathbf{c}_j$ are the subcluster assignments for objects i and j respectively, shown here with $M = 3$ features. $c_i^{(2)}$ being zero corresponds to the absence of feature 2 for object i, so this feature contributes no weight. For the two features which are active for both i and j, namely features 1 and 3, the subcluster assignments dictate which element of the feature's weight matrix should be chosen for each feature. Finally the weights are summed and passed through a sigmoid function to give the probability of a link between i and j.

probability that there is a link from object i to object j is given by

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

where the sum ranges over all M features, s is a bias term, and $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid (logistic) function that maps the input arguments from $(-\infty, +\infty)$ to $(0, 1)$, ensuring that the result is a valid probability. Under this model, only features that are on for both objects influence the probability of a link between them. For these common features, the appropriate weight values are summed up, depending on the subcluster assignments of i and j. The weight values are continuous variables which can be positive or negative allowing pairs of subclusters to encourage or discourage links between them correspondingly. We assume that given the $\mathbf{Z}$, $\mathbf{C}$ and $\mathbf{W}$, the probability of each link is independent and the likelihood is therefore as follows:

$$\Pr(\mathbf{R} \mid \mathbf{Z}, \mathbf{C}, \mathbf{W}) = \prod_{i,j} \Pr(r_{ij} \mid \mathbf{z}_i, \mathbf{z}_j, \mathbf{C}, \mathbf{W}). \quad (2)$$

In order to allow flexible inference of the latent structure from data, we set the number of possible features M and the number of subclusters in each feature $K^{(m)}$ to infinity by using an IBP prior on $\mathbf{Z}$ and CRP priors on the $\mathbf{c}$'s. The hierarchical generative model is then

$$\begin{aligned}
\mathbf{Z} \mid \alpha &\sim IBP(\alpha) \\
\mathbf{c}^{(m)} \mid \gamma &\sim CRP(\gamma) \\
w^{(m)}_{kk'} \mid \sigma_w &\sim N\big(0, \sigma_w^2\big) \\
r_{ij} \mid \mathbf{Z}, \mathbf{C}, \mathbf{W} &\sim Bernoulli\Big(\sigma\Big(\sum_m z_{im} z_{jm} w^{(m)}_{c_i^m c_j^m} + s\Big)\Big).
\end{aligned}$$

The ILA model is illustrated in Fig. 2.

The IBP parameter, α, affects the number of represented features, whereas the CRP parameter, γ, controls the number of subclusters inside each feature. To improve the flexibility of our model, we put Gamma priors on α and γ, and a Gaussian prior on the bias term s as follows:

$$\alpha \sim \mathcal{G}(1, 1), \qquad \gamma \sim \mathcal{G}(1, 1), \qquad s \sim \mathcal{N}\big(\mu_s, \sigma_s^2\big),$$

where μ_s and σ_s are the mean and standard deviation hyperparameters for the bias (we use $\mu_s = -1, \sigma_s = 1$ unless otherwise stated).

2.1 Noisy-OR Likelihood

An alternative likelihood function is a noisy-OR:

$$\Pr(r_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j, \mathbf{C}, \mathbf{W}) = 1 - (1 - \sigma) \prod_m \Big(1 - w^{(m)}_{c_i^m c_j^m}\Big)^{z_{im} z_{jm}}, \quad (3)$$

where $\sigma \in [0, 1]$ is a noise level, and $w^{(m)}_{st} \in [0, 1]$ are again weights. This can be interpreted as saying that two nodes will have a link between them if they have an interaction resulting from any of the features, or anyway with some small probability, σ. A feature can only result in an interaction if both nodes have that feature active, in which case their subcluster assignments determine the probability of having an interaction in this feature. To specify the full model we put beta priors on the weights and noise:

$$w^{(m)}_{st} \sim \text{Beta}(\alpha_w, \beta_w), \qquad \sigma \sim \text{Beta}(\alpha_\sigma, \beta_\sigma), \quad (4)$$

where unless otherwise stated we set $(\alpha_w, \beta_w) = (0.5, 0.5)$ and $(\alpha_\sigma, \beta_\sigma) = (1, 3)$. We show in Section 6 that this likelihood has computational advantages in that it allows the computational complexity of the method to scale as the number of links in the network rather than the total possible number of links, N^2. A similar approach has been applied to scaling inference in the LFRM [13]. Since in real world networks the number of links often grows more slowly than N^2 (known both informally and in a more technical sense as "sparse" graphs), this can be a significant saving. One significant difference to the logistic-Gaussian likelihood model is that the noisy-OR cannot represent features with *negative* weights that reduce the probability of interaction: additional features can never result in decreasing the probability of a link.

3 Related Work

Here we examine three models that are closely related to ILA. The IRM of Kemp and Tenenbaum [9] and the LFRM of Miller et al. [12] both use nonparametric Bayesian approaches to account for potentially infinite number of clusters in the data. In the IRM, the link probability between two objects depends only on the clusters they are assigned to:

$$\Pr(r_{ij} = 1 \mid c_i, c_j, \eta) = \eta_{c_i c_j}, \quad (5)$$

where the link probabilities for each pair of clusters, $\{\eta_{kk'} : k, k' = 1, \ldots, K\}$ are given independent Beta priors, and the cluster assignments, c are given a CRP prior. ILA and the LFRM on the other hand put a logistic-normal prior on the between feature and subcluster link probabilities. More specifically, the LFRM defines the link probability as

$$\Pr(r_{ij} = 1 \mid \mathbf{Z}, \mathbf{W}) = \sigma\Big(\sum_{kl} z_{ik} z_{jl} w_{kl} + s\Big), \quad (6)$$

where $\mathbf{W}$ is a $K \times K$ real valued weight matrix (with K being the number of features), given an elementwise

Gaussian prior, and $\mathbf{Z}$ is an $N \times K$ matrix of binary feature vectors drawn from an IBP. Comparing Equations (6) and (1) for ILA, we see how the two models differ. The LFRM defines a weight value for each possible pair of features, while ILA defines a weight *matrix* for each feature, whose elements correspond to every pair of subclusters in that feature. The link probability in LFRM depends on all the possible pairs of features that are on for both objects, while in the ILA model, the link probability is contributed to only by features that are *simultaneously* on for both objects. While subclusters within a feature can interact in ILA, subclusters from different features do not interact.

Unlike the IRM, the ILA model does not partition the objects into a set of non-overlapping clusters; although it specifies non-overlapping subclusters for each feature, it also allows each object to have multiple features, thus accounting for multiple membership. ILA is more expressive than LFRM because it associates each feature with a set of subclusters.

Interestingly, both the IRM and LFRM can be thought of as special cases of our model. If only one column of $\mathbf{Z}$ is switched on in ILA (i.e., there is only one feature which is on for every object) then this is equivalent to the IRM. In this case the ILA logistic-Gaussian likelihood becomes

$$\Pr(r_{ij} = 1 \mid \mathbf{Z} = \mathbf{1}, \mathbf{C}, \mathbf{W}) = \sigma\Big(w^{(1)}_{c_i^1 c_j^1} + s\Big). \tag{7}$$

Contrasting this to Equation (5) the ILA has a logistic-normal prior on the between subcluster link probabilities rather than a Beta prior, but this is a relatively minor difference. With $\sigma = 0$ the ILA noisy-OR likelihood in this case (i.e., $\mathbf{Z} = \mathbf{1}$) is identical to the IRM.

If the LFRM is constrained to have a weight matrix $\mathbf{W}$ with only diagonal non-zero elements, then its link probability becomes

$$\Pr(r_{ij} = 1 \mid \mathbf{Z}, \mathbf{W}) = \sigma\Big(\sum_k z_{ik} z_{jk} w_{kk} + s\Big).$$

This is then equivalent to ILA in the case when there is only one subcluster in each feature, since the ILA logistic link probability is then

$$\Pr(r_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j, \mathbf{C} = 1, \mathbf{W}) = \sigma\Big(\sum_m z_{im} z_{jm} w^{(m)}_{11} + s\Big).$$

In [13] a version of LFRM using the noisy-OR likelihood,

$$\Pr(r_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j, \mathbf{W}) = 1 - (1 - \sigma) \prod_{kl} (1 - w_{kl})^{z_{ik} z_{jl}}, \tag{8}$$

was proposed. This is equivalent to ILA with the noisy-OR likelihood under the same conditions as for LFRM: if the model of Equation (8) has diagonal W and ILA has only one subcluster per feature.

ILA can also be seen as a extension of the Multiplicative Attribute Graph (MAG) model proposed in Kim and Leskovec [10], where the link probability is

$$\Pr(r_{ij} = 1 \mid \mathbf{C}, \eta) = \prod_m \eta^{(m)}_{c_i^m c_j^m},$$

where η is a set of M two by two matrices of probabilities with elementwise independent Beta priors, and the $\mathbf{c}$'s are equivalent to our subcluster assignment variables but constrained to takes values in $\{1, 2\}$. We extend this model in three ways: 1) we learn the number of subclusters in each feature, rather than fixing it to two, 2) we learn the number of features M, and 3) we incorporate additional sparsity, in that an object need not have a particular feature active at all. We parameterise our model in terms of real valued weights which contribute to the log odds of a link being on, rather than with probabilities that are multiplied together, but this entails no loss of flexibility. In fact this may be advantageous to ILA since the MAG suffers from each new feature decreasing all link probabilities.

There are several models that have been proposed for discovering hierarchical structure in relational data [4], [20]. In [4], each object is still a member of one out of many non-overlapping clusters. In [20] each object can belong to multiple overlapping classes, which are nested in a hierarchy. Our model is distinct in using a factorial structure to allow each object to be a member of many subclusters as long as these subclusters are in different features.

4 Inference

In the following, we present a method for inferring the latent variables of the model: the infinite binary feature matrix, $\mathbf{Z}$, the subcluster assignments, $\mathbf{c}^{(m)}$ for each feature m, and the weight matrices, $\mathbf{W}^{(m)}$. Simultaneously we recover the number of features and the number of subclusters within each feature. As with many other Bayesian models, exact inference is intractable so we employ Markov Chain Monte Carlo (MCMC), and follow an iterative procedure that achieves posterior inference over the latent variables.

4.1 Sampling the Feature Matrix, Z

We Gibbs sample each element of $\mathbf{Z}$ in succession. For each object i, the sampler makes the following decisions: which of the current M available features should be turned on/off, and how many new features should be turned on. However, when turning on a feature the sampler must also sample a new subcluster assignment and, in case of adding a new subcluster, the related new weights.

By exchangeability of the rows of $\mathbf{Z}$ we can assume that the ith object is the last to be added to $\mathbf{Z}$ after $N - 1$ rows have already been added. For all the M features currently present in $\mathbf{Z}$, the conditional posterior probability of an entry z_{im}, $m = 1, \ldots, M$ follows a Bernoulli distribution:

$$\begin{aligned}\Pr(z_{im} = 1 \mid \mathbf{Z}_{-im}, \mathbf{C}_{-im}, \mathbf{W}, \mathbf{R}) \\ \propto \frac{n_{-im}}{N} \Pr(\mathbf{R} \mid z_{im} = 1, \mathbf{Z}_{-im}, \mathbf{C}_{-im}, \mathbf{W}),\end{aligned} \tag{9}$$

where $\mathbf{Z}_{-im}$ is the $\mathbf{Z}$ matrix excluding the $Z(i, m)$ element, n_{-im} is the number of times feature m is present in $\mathbf{Z}_{-im}$ and $\mathbf{C}_{-im}$ excludes the subcluster assignment $c_i^{(m)}$. To compute the probability in Equation (9), we need to sum over $c_i^{(m)}$, the space of the possible subclusters that the ith object may be assigned to if z_{im} is to be turned on. This also includes integration over a possible new subcluster. However, the prior over the parameters $\mathbf{W}^{(m)}$ related to a new subcluster is not conjugate for either likelihood function (logistic or noisy-OR), and thus the likelihood term cannot

be computed exactly. To overcome this problem, we use the auxiliary variable approach proposed in Neal [14] (Algorithm 8), both to facilitate the integration required in Equation (9), and to decide which subcluster to assign the ith object to in the mth feature if z_{im} is turned on.

We must also sample the number of new features unique to the i-th row, $M_{\text{new}}^{(i)}$. Instead of considering these features separately, we calculate the conditional posterior over $M_{\text{new}}^{(i)}$, using the fact that under the IBP the prior distribution over M_{new} for the last row is Poisson(α/N). Combining the Poisson prior with the likelihood, we obtain the conditional posterior over $M_{\text{new}}^{(i)}$. However, to obtain the required likelihood term we need values for $\mathbf{C}^{(m)}$ and $\mathbf{W}^{(m)}$ for the proposed new features. Clearly $c_i^{(m)} = 1$ for any new features, since a feature active for only one object can only have one subcluster. Integrating over the weights is not straightforward because the prior over $\mathbf{W}^{(m)}$ is not conjugate to the likelihood. We therefore employ a Metropolis Hastings step, proposing values for $w_{11}^{(m)}$ from the prior so that the acceptance ratio becomes simply the likelihood ratio for including the new features and associated $\mathbf{C}^{(m)}$ and $\mathbf{W}^{(m)}$ values in the model versus not including them.

4.2 Sampling the Subcluster Assignments, $\mathbf{C}$

We may choose to resample each $\mathbf{C}^{(m)}$ in succession as a second step, again using Algorithm 8 of Neal [14]. In practice we found this unnecessary since $\mathbf{C}$ is sampled in the process of sampling $\mathbf{Z}$.

4.3 Sampling the Weights, $\mathbf{W}$

Given $\mathbf{Z}$ and $\mathbf{C}$, the sampler successively resamples each of the weights $\{w_{kk'}^{(m)} : k, k' = 1, \ldots, K^{(m)}, m = 1, \ldots, M\}$. Since we do not have conjugacy (due to the logistic link function), we cannot sample directly from the posterior over $w_{kk'}^{(m)}$. To overcome this problem we used both Metropolis Hastings and slice sampling [15] but found the latter resulted in faster mixing. For the noisy-OR likelihood the weights are constrained to lie in the unit interval. We again use slice sampling but find that using the logit reparameterisation, $t = \log(w/[1-w])$ gives somewhat improved performance.

4.4 Hyperparameters

We use slice sampling the CRP concentration parameter, γ and the bias, s. Having put a gamma prior on the IBP hyperparameter, α, conjugacy facilitates exact sampling from a gamma posterior.

4.5 IRM Implementation

Our implementation of the IRM of Kemp and Tenenbaum [9] uses standard Gibbs sampling. The IRM is not naturally a good model for sparse real world networks, a problem we attempt to alleviate by putting an asymmetric beta prior Beta(β_1, β_2) on the weights and a log-normal prior over β_1 and β_2 with $\mu = -1$ and $\sigma = 1$. In the IRM we are able to integrate out the weights η analytically due to conjugacy, so we need only sample the cluster assignments and the hyperparameters: the CRP concentration parameter, γ and the weight hyperparameters β_0 and β_1. We use slice sampling for the hyperparameters.

4.6 LFRM Implementation

For the LFRM of Miller et al. [12], we Gibbs sample the IBP matrix $\mathbf{Z}$ and slice sample each element of the weight matrix $\mathbf{W}$ sequentially, followed by the IBP concentration parameter.

4.7 Sequential Initialisation

The Gibbs updates described above are the simplest moves we could make in a MCMC inference procedure for the ILA model. However, these updates are quite incremental, since only a single variable is updated at a time. Due to the extremely large number of possible configuration states, $\prod_{m=1}^{M}(K^{(m)}+1)^N$, the sampler can suffer from local modes and have somewhat slow mixing. Non-incremental moves, like splitting and merging features in the $\mathbf{Z}$ matrix or subcluster assignments in $\mathbf{C}$ can produce major changes in the configuration state in a single iteration and can help the sampler explore more efficiently. Split-merge sampling in the IBP has been previously described in Meeds et al. [11]. However, we found that a sequential initialization of the sampler improved the performance, guiding the sampler closer to neighborhoods of higher probability.

To sequentially initialise all parameters the nodes are first randomly permuted and then added to the model as follows. Initially two nodes are added to the model with no features active. Then a few (typically three) iterations of the MCMC sampler are run. Then the next node is added, with no features turned on, and another three iterations of the sampler are run. This procedure is iterated until all nodes have been added. The sampler will naturally grow the number of features and subclusters within each feature as more data is added. The advantage of this method is that the initialisation is appropriate for the model, the sampler is very fast initially due to the small number of nodes, and the search space is small initially so it is easier for the Markov chain to find a relatively high probability region of parameter space. We also used sequential initialisation for our implementation of LFRM, but not for IRM where empirically we did not find it helpful perhaps due to the simpler nature of the model.

4.8 Prediction

A principled way to evaluate a generative model is by its ability to predict missing data values given some observations. For ILA we collect T samples $\{\{\mathbf{Z}_{(1)}, \mathbf{C}_{(1)}, \mathbf{W}_{(1)}\}, \ldots, \{\mathbf{Z}_{(T)}, \mathbf{C}_{(T)}, \mathbf{W}_{(T)}\}\}$ from the posterior and estimate the predictive distribution of a missing link as the average of the predictive distributions for each of the collected samples. Assuming that we want to predict the missing link r_{ij} between objects i and j, the approximate predictive distribution is

$$\Pr(r_{ij} = 1 \,|\, \mathbf{R}_{\text{train}}) \approx \frac{1}{T}\sum_{t=1}^{T} \Pr(r_{ij} = 1 \,|\, \mathbf{Z}_{(t)}, \mathbf{C}_{(t)}, \mathbf{W}_{(t)}).$$

5 Computational Complexity

In general, the computational cost of latent feature models scales quadratically in the number of objects. In the LFRM, computing the likelihood has a complexity of $\mathcal{O}(M^2N^2)$,

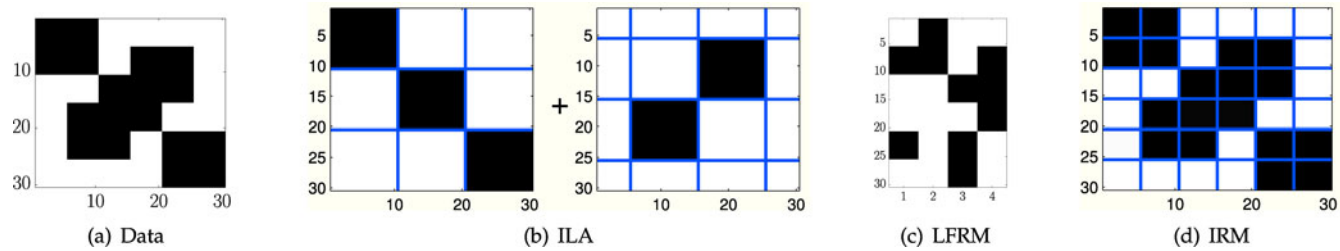


Fig. 3. Synthetic data example. (a) Observed synthetic 30×30 link matrix. White corresponds to zero, black to one. (b) ILA (logistic) solution. Contribution to the adjacency matrix from the two features found, the first has three subclusters, and the second two dis-associative subclusters. (c) LFRM solution: $\mathbf{Z}$ matrix. White corresponds to zero, black to one (active feature). (d) IRM solution. Six clusters are found.

where M and N is the number of represented features[1] and the number of objects correspondingly. For ILA, the link probability between two objects given by Equation (1), results in computational cost $\mathcal{O}(MN^2)$ when calculating the likelihood across all pairs. The computational cost of the IRM scales linearly in the number of links in the network, $L = \sum_{ij} r_{ij}$, because the likelihood, with the link probabilities η integrated out, can be written as

$$\Pr(\mathbf{R} \,|\, \mathbf{c}) = \prod_{a,b} \frac{\mathrm{Beta}(n(a,b)+\beta, \overline{n}(a,b)+\beta)}{\mathrm{Beta}(\beta,\beta)},$$

where $n(a,b)$ is the number of pairs of objects (i,j) where $i \in a$ and $j \in b$ and $R(i,j) = 1$, $\overline{n}(a,b)$ is the number of such pairs where $R(i,j) = 0$, and $\mathrm{Beta}(\cdot,\cdot)$ is the Beta function. The computational cost of computing the likelihood in the IRM is therefore $\mathcal{O}(K^2 L)$.

The fact that the computational cost for ILA grows quadratically in N motivates using a different likelihood, the noisy-OR, that allows computation to scale with the number of observed links, as described in the next section (Section 6). This allows improved scalability on typical sparse real world networks where the number of links is much smaller than the number of non-links. This likelihood comes with the restriction of only being able to have positive weights between clusters (homophily).

6 Noisy OR Likelihood Computation

Using ideas from Mørup et al. [13] we show here how the noisy-OR likelihood enables us to calculate the complete likelihood with computational complexity that scales linearly with the number of links in the graph, as opposed to $O(N^2)$ for the Gaussian-logistic likelihood. This is beneficial for real world networks where the number of observed links typically grows much more slowly than N^2 and results in sparse networks. We introduce the indicator variables

$$\xi_{is}^m = z_{im} \mathbb{I}\big[c_i^m = s\big],$$

so that we can rewrite Equation (3) as

$$\Pr(r_{ij} = 1 \,|\, \mathbf{z}_i, \mathbf{z}_j, \mathbf{C}, \mathbf{W}) = 1 - (1-\sigma)\exp\Big(\sum_{mst} P_{st}^{(m)} \xi_{is}^m \xi_{jt}^m\Big),$$

where we define

$$P_{st}^{(m)} := \log\big(1 - w_{st}^{(m)}\big).$$

1. M is potentially unbounded, but in practice the model will use some finite number of features to model any finite data set.

Assuming $\sigma = 0$ (the computations go through with little modification for $\sigma > 0$) the overall likelihood is now

$$\prod_{(i,j)\in\mathcal{Y}_1} 1 - \exp\Big(\sum_{mst} P_{st}^{(m)} \xi_{is}^m \xi_{jt}^m\Big) \times \prod_{(i,j)\in\mathcal{Y}_0} \exp\Big(\sum_{mst} P_{st}^{(m)} \xi_{is}^m \xi_{jt}^m\Big), \tag{10}$$

where $\mathcal{Y}_1$ and $\mathcal{Y}_0$ denote the sets of links and non-links in the training data respectively. The product over the non-links can be written

$$\begin{aligned}\prod_{(i,j)\in\mathcal{Y}_0} \exp\Big(\sum_{mst} P_{st}^{(m)} \xi_{is}^m \xi_{jt}^m\Big) &= \exp \sum_{(i,j)\in\mathcal{Y}_0} \sum_{mst} P_{st}^{(m)} \xi_{is}^m \xi_{jt}^m \\ &= \exp \sum_{mst} P_{st}^{(m)} \sum_{(i,j)\in\mathcal{Y}_0} \xi_{is}^m \xi_{jt}^m.\end{aligned} \tag{11}$$

We can write the inner sum efficiently as

$$S_{mst} = \sum_{(i,j)\in\mathcal{Y}_0} \xi_{is}^m \xi_{jt}^m = \sum_i \xi_{is}^m \sum_j \xi_{jt}^m - \sum_{(i,j)\in\mathcal{Y}_0^c} \xi_{is}^m \xi_{jt}^m,$$

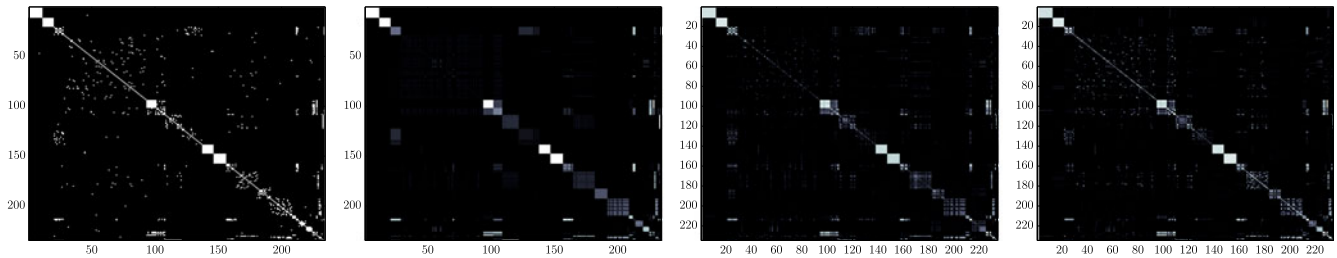
where the first term is an efficient, $O(N)$, means of calculating the sum over the complete graph, and the second term is a sum over the complement $\mathcal{Y}_0^c$, which is typically small, consisting of the training links, missing and test edges.

7 Results

We present results on a toy synthetic data set and on three real world data sets: the NIPS coauthorship network, a novel gene interaction network and a larger curated gene pathway network.

7.1 Synthetic Data

We first explored the ability of our model to recover the underlying structure of a network using synthetic data. We considered one simple $N = 30$ synthetic data set (Fig. 3a) hand-constructed to have an unambiguous most parsimonious solution under each model. Under ILA (logistic) this is the feature matrix shown in Fig. 3b with two features. The first feature has three homophilic subclusters (i.e., individuals tend to have links if they are in the same cluster), whereas the second feature has two heterophilic subclusters (i.e., individuals tend to link if they are in different clusters). We ran ILA for 200 MCMC iterations following sequential initialisation. The sample with the lowest energy (highest log probability under the posterior) corresponds exactly to the



(a) True links (b) IRM (c) LFRM (d) ILA (logistic)

Fig. 4. Predictions for the three models on the NIPS 1-17 coauthorship data set. In (a), white denotes that two people wrote a paper together, while in (b), (c), and (d), the lighter the entry, the more confident the model is that the corresponding authors would collaborate.

TABLE 1
NIPS Coauthorship Network Results

	IRM (simple)	LFRM	ILA logistic	ILA noisy-OR
Train error (0-1 loss)	0.0262 ± 0.0046	0.0221 ± 0.0009	0.0190 ± 0.0013	$\mathbf{0.0179 \pm 0.0029}$
Test error (0-1 loss)	0.0289 ± 0.0041	0.0251 ± 0.0012	0.0242 ± 0.0016	$\mathbf{0.0227 \pm 0.0032}$
Test log likelihood	-0.0484 ± 0.0056	-0.0306 ± 0.0034	-0.0305 ± 0.0031	$\mathbf{-0.0292 \pm 0.0042}$
AUC	0.9305 ± 0.0123	0.9169 ± 0.0136	$\mathbf{0.9360 \pm 0.0081}$	0.9058 ± 0.0194

expected "true" structure, as shown in Fig. 3b. The MAP sample found using LFRM is shown in Fig. 3c. Again this is a passable explanation of the data but it is considerably more convoluted than the simple, interpretable but rich solution found using ILA. Note that running 2,000 iterations (following sequential initialisation) of LFRM no better solution was found. In contrast the IRM finds the flat clustering of six clusters shown in Fig. 3d, which is an acceptable solution but does not capture the rich structure that ILA is able to.

7.2 NIPS Coauthorship Network

We compare the performance of the IRM, LFRM and ILA on the NIPS coauthorship data set [5], where a link corresponds to two individuals being coauthors of a paper at one of the first 17 NIPS conferences (see Fig. 4a). Following Miller et al. [12] we use only the 234 most connected authors. We run 10 repeats, each time holding out a different 20% of the data (links and non-links) and using a different random initialisation. We run two versions of ILA: the first with the logistic likelihood, and the second with the noisy-OR likelihood. We run 1,000 iterations for each method and calculate evaluation metrics averaged over the last 300 samples. The results are shown in Table 1, and Figs. 5 and 6 which shows predictive performance as a function of MCMC iteration number and computation time respectively.

We confirm the finding in Miller et al. [12] that LFRM outperforms the IRM on this data set. However, across all three evaluation metrics one of the two versions of ILA significantly outperforms LFRM (for example, the t-test between the test error for LFRM and ILA noisy-OR shows the means to be significantly different with a p-value of 10^{-5}). Which likelihood model appears best for ILA depends on the choice of evaluation metric. Under the ILA posterior M is concentrated between 10 and 14 features, with typically two to four subclusters per feature. Figs. 5 and 6 show that while ILA (even the noisy-OR version) has somewhat higher computational cost per MCMC iteration,

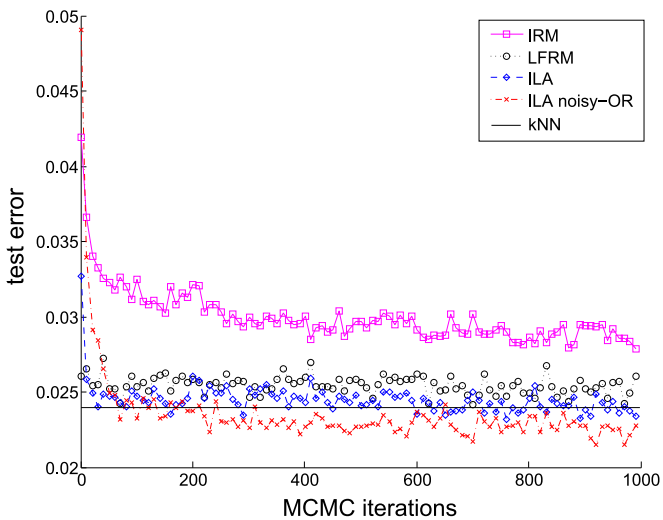


Fig. 5. Average test error as a function of MCMC iteration on the NIPS data set. Because of sequential initialisation even at iteration 1 much of the predictive performance has already been achieved.

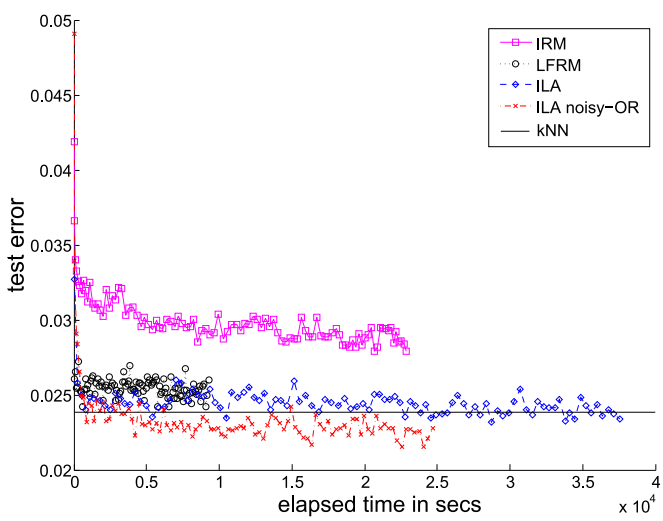


Fig. 6. Average test error as a function of time on the NIPS data set. While ILA has a higher per iteration cost than the IRM or LFRM we see that ILA would still perform best even given the same computation budget.

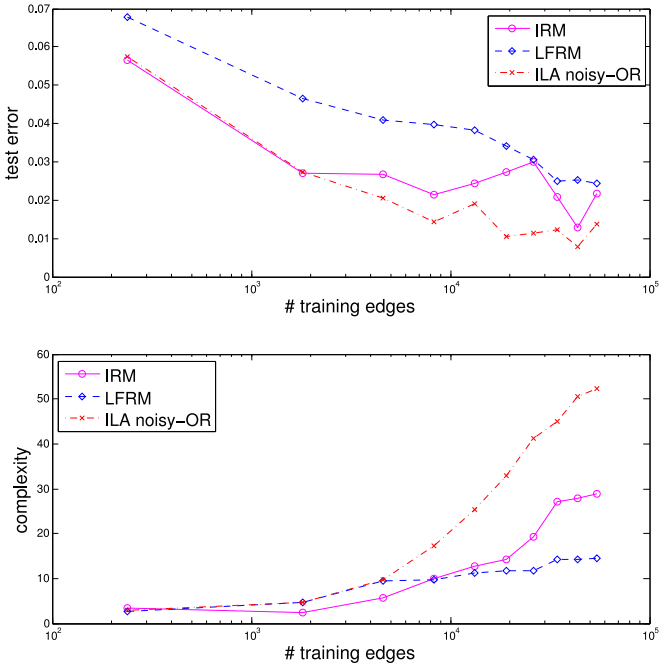


Fig. 7. Predictive performance (top) and model complexity (bottom) for varying amounts of training data on the NIPS data set. Points shown are averages from three repeats. Complexity is number of clusters for IRM, number of features for LFRM and total number of subclusters across all features for ILA (noisy-OR).

good predictive performance could in fact be achieved with considerably fewer iterations than we used. In particular, our conclusions would be the same if we stopped all four models when LFRM finished its 1,000 iterations. In order to compare to a non-model based approach we also predicted links using k-nearest neighbour (kNN) imputation as implemented in the R package 'imputation', which includes cross-validation for choosing k. The resulting test error is 0.0239 ± 0.002, which is beaten only by the noisy-OR version of ILA. While the non-parametric nature of kNN allows it to achieve good predictive performance, it lacks the interpretability of the model based methods we focus on in this paper.

In Fig. 4 the link predictions for each of the three models are presented. Figs. 4b, 4c, and 4d visualize the belief of each model that there should be a link between each pair of authors. The link matrices were constructed after running the three models on the NIPS 1-17 data set for 1,000 iterations, using the same random seed and averaging over the last 300 samples. To facilitate interpretability, we ordered the authors by the clusters found by the IRM. It can be clearly seen that both the LFRM and ILA (logistic) models outperform the IRM by appearing more confident and reproducing the corresponding network more faithfully. Considering Figs. 4c and 4d, LFRM and ILA appear comparable, with ILA being slightly more confident. Quantitatively however, Table 1 shows that the ILA solution is superior.

Using this data set we also investigate predictive performance and model complexity as a function of the amount of training data provided to the various models. A subset of 23 nodes (i.e. 10% of the $N = 234$ data set) were chosen as test nodes, and half of the edges between these nodes were used as test edges. Initially only these 23 nodes were trained on, and then 10% more of the nodes were added sequentially until all $N = 234$ nodes were included, keeping the test edges the same. The results are shown in Fig. 7, where complexity is measured as the number of clusters for IRM, the number of features for LFRM and the total number of subclusters across all features for ILA. Here we find IRM outperforms LFRM, contrary to our conclusion from Table 1: this is likely to be a result of the very different holdout pattern for this experiment. However, we again see ILA outperforming the other two models.

7.3 Gene Interaction Network

We present results on a subset of the interaction data presented in Jonikas et al. [8].[2] This is an example of a new class of high throughput gene interaction assays, in this case using the yeast *S. cerevisiae*. A range of "deletion" strains are created, each of which has a single gene deleted. Some phenotypic response is measured during the growth of each strain, in this case unfolded protein response (UPR), a measure of how badly the cell is doing at correctly folding its membrane proteins. "Double mutants" with two distinct genes deleted are then screened. Based on the single deletion strains, the expected UPR response for these double mutants can be predicted (see Jonikas et al. [8] for details) assuming no interaction between the two deleted genes. If the observed UPR response is significantly different from this predicted value then the genes must interact in some way, so we consider this as an edge in the network. We use the 156 genes with the least missing data. We run 10 repeats with a different 10% of the observed data heldout each time, and perform 1,000 MCMC iterations for all models. Again we find ILA significantly outperforms LFRM and the simple IRM (see Table 2). ILA typically finds around $M = 30$ features with three to five subclusters per feature. We find significantly more features are associated with particular properties of the genes as defined by Gene Ontology classes[3] than would be expected by chance ($p < 10^{-3}$ calculated by permutation testing), for example the three subclusters of one particular feature have very different proportions of ligand binding genes ($10/41, 21/27$ and $2/20$ respectively).

In Fig. 8 the link predictions for each of the three models for the gene data set are presented. As in the NIPS data set, the link matrices were constructed after running each model for 1,000 iterations, using the same seed and averaging over the last 300 samples. The clusters found by the IRM were used to order the genes. Both the LFRM and ILA (logistic) models outperform the IRM, but ILA is able to capture more structure in the data.

7.4 Cancer Gene Map

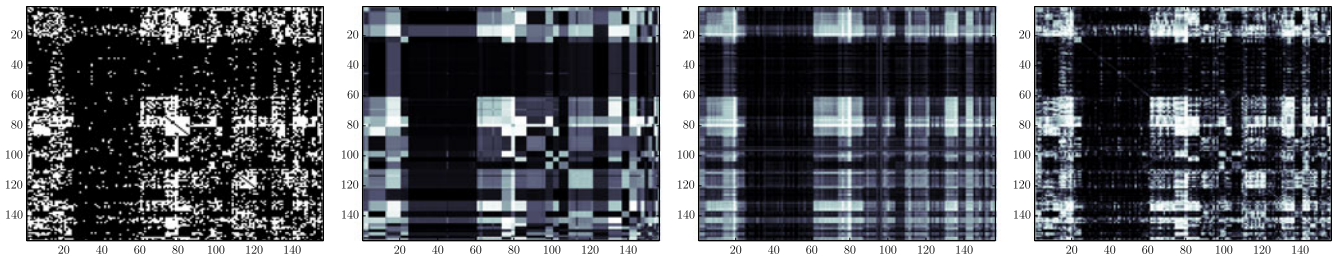
We apply the noisy-OR implementation of ILA to modelling the cancer cell map[4] network, a hand curated network of pathway interactions between human genes. The data set includes 1,978 interactions among 1,139 genes, thereby representing the largest network we analyse here. However, we are able to leverage the efficient noisy-OR likelihood

2. http://weissmanlab.ucsf.edu/upremap/.
3. http://www.geneontology.org/.
4. http://cancer.cellmap.org.

TABLE 2
Gene Interaction Network Results

	IRM	LFRM	ILA logistic	ILA noisy-OR
Train error (0-1 loss)	0.2110 ± 0.0022	0.2526 ± 0.0135	$\mathbf{0.1573 \pm 0.0065}$	0.1822 ± 0.0046
Test error (0-1 loss)	0.2425 ± 0.0059	0.2681 ± 0.0113	$\mathbf{0.2269 \pm 0.0059}$	0.2322 ± 0.0051
Test log likelihood	-0.2463 ± 0.0066	-0.2123 ± 0.0120	-0.2218 ± 0.0189	$\mathbf{-0.2100 \pm 0.0099}$
AUC	0.8701 ± 0.0135	0.8433 ± 0.0186	$\mathbf{0.8849 \pm 0.0118}$	0.8788 ± 0.0099

All values are averages over the test edges. The best results are highlighted in bold.



(a) True interactions (b) IRM (c) LFRM (d) ILA (logistic)

Fig. 8. Predictions for the three models on the gene interaction network data set. In (a), white denotes that two genes interact, while in (b), (c), and (d), the lighter the entry, the more confident the model is that the corresponding genes would interact. Note that a lot more of the structure in the data is captured by ILA.

computation described in Section 6 due to the sparsity of the graph. We are able to run $1{,}000$ iterations, following sequential initialisation, of ILA on a standard PC in just under eight hours. Our implementation is in MATLAB and the code is not optimised: using a lower level programming language such as C and more efficient data structures we expect this time could be greatly reduced since significant book-keeping is required, but this was not the focus of our work. For comparison, running IRM or LFRM for the same number of iterations takes around 5 hours. ILA finds between 8 and 14 reproducible features on this data set, with up to six subclusters per feature, whereas IRM finds between 50 and 80 clusters, and LFRM between 20 and 25 features.

In order to assess whether the structure found by ILA is more biologically meaningful than that found by IRM or LFRM we again look for association with the Gene Ontology (GO) terms. We use only the 467 GO biological process terms belonging to at least 10 of the $N = 1{,}139$ genes in the data set. We run ten MCMC chains for $1{,}000$ iterations for each model, and investigate the final sample for each. For each group, i.e., cluster (IRM), feature (LFRM) or subcluster (ILA), with at least 10 member genes, we test for association using a chi-squared test of independence. The resulting p-values are used in the Benjamini Hochberg procedure to control the global False Discovery Rate (FDR) at 0.01. The proportion of groups with at least one[5] GO term significant at this level is shown in Fig. 9, where we see that ILA consistently finds more biologically meaningful groupings of genes than IRM or LFRM.

It is interesting to look in detail at the structure found in a specific run. For one ILA chain, we investigate the most active feature, which contains four subclusters. Two of these subclusters show a high level of dis-associativity: while none of the genes within either subcluster interact, 32% of the possible intercluster links are present, a very high proportion in such a sparse network. The nodes in these two subclusters, and their interactions are shown in Fig. 10. The subcluster assignments are shown in pink and cyan. Using a straightforward visualisation, without a model based approach like ILA, it would be difficult to spot the strong pattern in this sub-network. Fig. 11 shows the interactions with one subcluster on the left and the other on the right. The six genes in the smaller subcluster, REL, NFKB1, NFKB2, RELA, NFKBIB and NFKBIA are all in fact closely linked biologically, despite this information not being present in the cancer cell map. NFKB1 or NFKB2 binds REL or RELA to form a transcription factor known as NFKB, which is inhibited by NFKBIA and NFKBIB [21]. Thus we see that exploratory analysis using ILA is able to uncover biologically meaningful groupings of the genes in the network.

8 Convergence Diagnostic Tests

Here we test the ability of the method to explore the posterior over the latent variables in the model. In the Supplementary Material, available online, we show the correctness of sam-

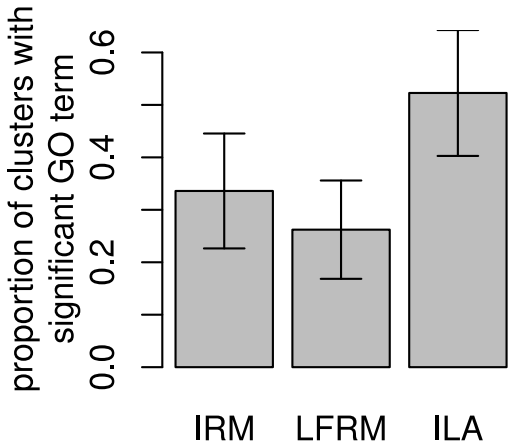


Fig. 9. The proportion of clusters (IRM), features (LFRM) or subclusters (ILA) with a significant association to a Gene Ontology term across 10 repeats. ILA finds more meaningful groups of genes than IRM or LFRM.

5. It is not useful to look at how many GO terms a group is associated with since there is significant redundancy between GO terms.

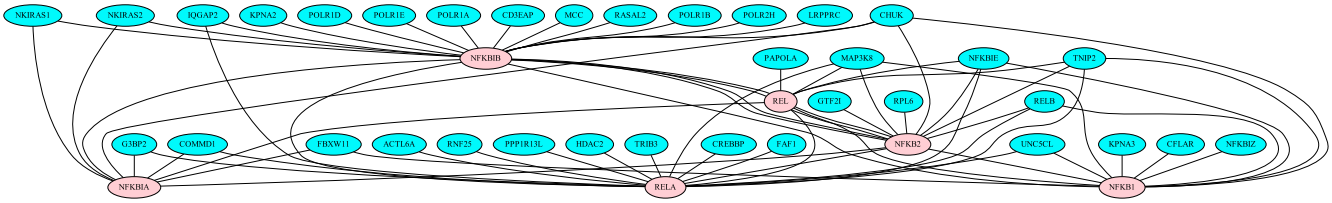


Fig. 10. A subnetwork of the cancer cell map with high disassociavity. These nodes belong to two subclusters of a particularly active four subcluster feature.

pler using the joint distribution testing methodology of [3], and we evaluate both the Raftery and Lewis [19] and the Gelman and Rubin [2] convergence diagnostics.

In order to examine the burn-in time and mixing rate of our sampler, we used a range of techniques that assess different aspects of the chain. For the following tests, we used a synthetic data set of size $N = 90$ as seen in Fig. 12 and ran the sampler for approximately $8{,}000$ iterations. For completeness, we also performed convergence analysis on the $N = 234$ NIPS coauthorship data set (see Section 7.2).

8.1 Traceplots and Running Mean Plots

Trace and running mean plots of the IBP hyperparameter, α, CRP hyperparameter, γ and bias parameter are shown in Fig. 13 for two random initialisations of the ILA MCMC chain. In both cases, the chains converge to a reasonable mode of the distribution relatively quickly (in just a few hundred iterations) but take considerably longer to move between modes, often staying in a single mode for over a thousand iterations. Two modes are easily noticeable in the second chain: one at $(\alpha, \gamma) \approx (0.6, 1.5)$ and one at $(1.2, 0.7)$. The later uses more features and fewer subclusters than the former mode. It is not surprising that moving between these modes is challenging given the incremental nature of our Gibbs sampler, but it encouraging that the chain does eventually succeed in doing so. Autocorrelation plots for the two chains of Fig. 13 are included in the Supplementary Material, available online. For the first chain the autocorrelation decays to a negligible level in just tens of iterations, but by looking at the traceplots of Fig. 13 together with the autocorrelations for the second run we conclude that this is an optimistic view resulting from the fact that the first run stayed in a single mode of a multimodal posterior throughout. The second run managed to move, albeit slowly, between two qualitatively different modes of the posterior, which results in non-negligible autocorrelation even at a delay of over 100. These differences between these two runs highlight the difficulty in assessing MCMC convergence: looking only at the first run would suggest we are exploring the entire posterior effectively, whereas the results of the second chain suggests there are multiple modes never visited by the first chain. For the more complex realworld NIPS data set, traceplots are shown in Fig. 14. Here we see less distinct modes in the posterior, and the autocorrelation of the bias decays the slowest, taking around 100 iterations. This mixing rate is quite reasonable, with the caveat that the chain might be missing other importants modes (this concern is addressed by the methodology in Section 8.2 and Section 3 of the Supplementary Material, available online). In the Supplementary Material, available online, we include the corresponding plots for one run of IRM and LFRM. The IRM appears to require a longer burning than ILA or LFRM, but has no parameter whose autocorrelation stays as high for as long as the bias parameter in ILA or LFRM. This can be justified by the nature of the models: the combinatorial nature of IBP results in multimodal posterior distributions as opposed to the simpler IRM.

8.2 Convergence from Extreme Initial States

Similarly we test the ability of our sampler to converge to the same stationary distribution from specific, rather than random, extreme initialisations:

1) no features,
2) one feature for each node, active for only that node (and therefore one subcluster per node), i.e., $\mathbf{Z} = \mathbf{I}_N$,

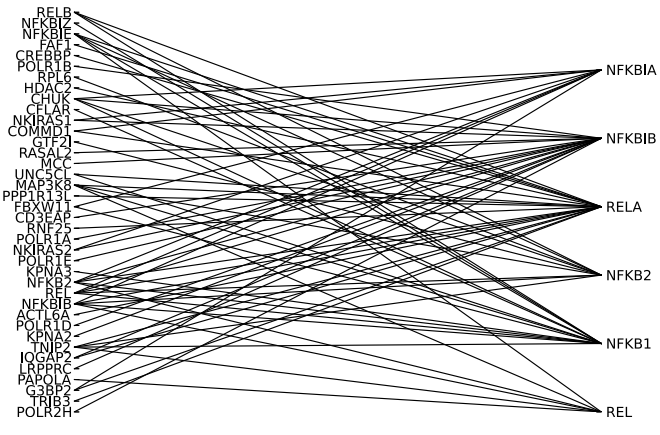


Fig. 11. The same subnetwork as Fig. 10, but organised to emphasise the disassociative nature of the two subclusters.

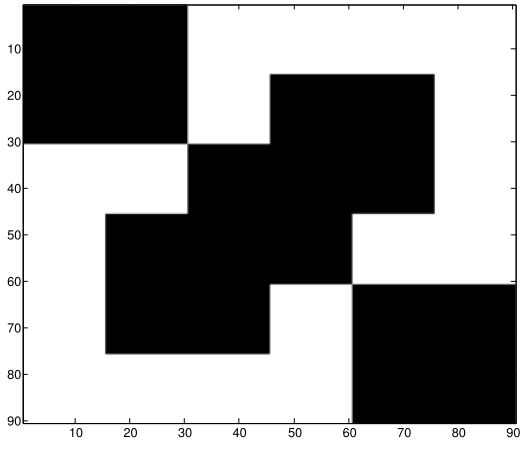
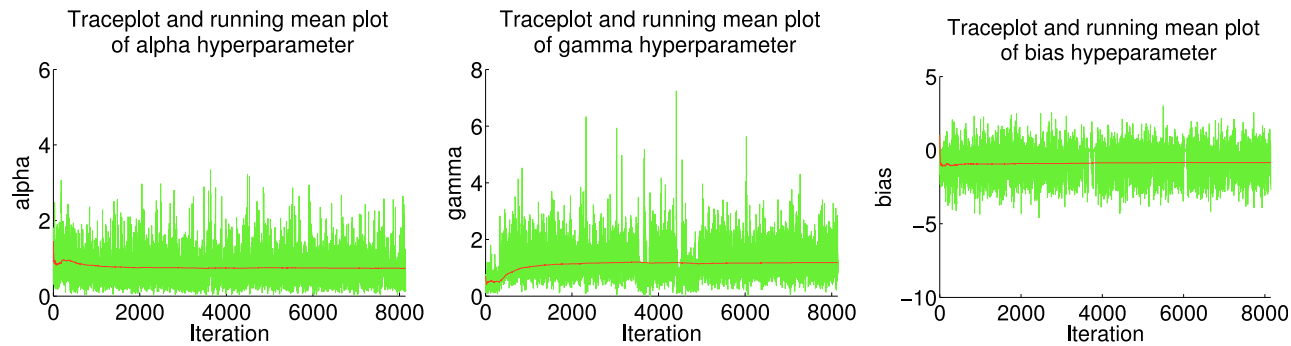
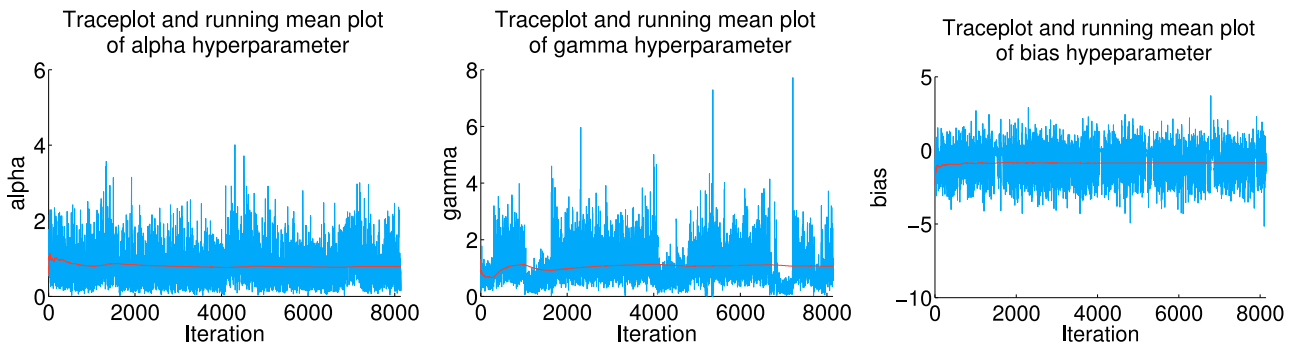


Fig. 12. Synthetic link matrix used for the convergence tests ($N = 90$). Black denotes link.



(a) IBP concentration, α (b) CRP concentration, γ (c) Bias



(d) IBP concentration, α (e) CRP concentration, γ (f) Bias

Fig. 13. Traceplots and running mean plots (red line) of the ILA logistic α, γ and bias parameters for the synthetic data set with $N = 90$. The two rows represent different random initialisation. Convergence to a posterior mode is rapid, but moving between modes takes many iterations.

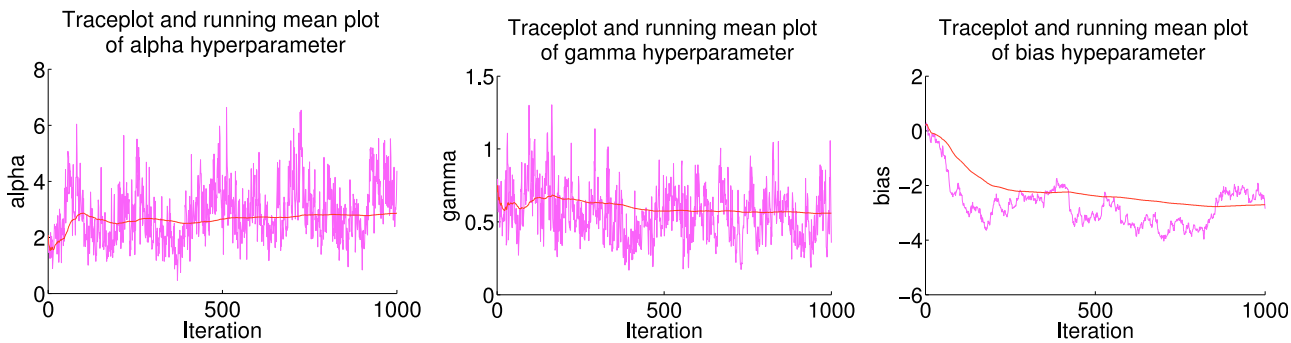


Fig. 14. Traceplots plots of the α, γ and bias parameters of ILA for the NIPS data set ($N = 234$). On this more complex real world data set distinct modes are not apparent but there is significant autocorrelation, particularly for the bias parameter, for up to 100 iterations.

3) one feature active for all nodes, all nodes in the same subcluster, and
(4) one feature active for all nodes, each node in its own subcluster.

We use the NIPS coauthorship data set ($N = 234$, see Section 7.2), and run 1,000 MCMC iterations of the sampler from these four very different initialisations. Our main concern is whether the complexity of the solution found will be the same in each case, which we assess in Fig. 15 by showing the total number of subclusters at each iteration for each chain. Encouragingly we see that after around 200 iterations the number of subclusters is similar for all four chains, although initialisation 2 (which starts with N features) does continue to have slightly more subclusters than the other chains throughout the 1,000 iterations.

In conclusion, the ILA MCMC sampler does not seem to have significantly greater mixing problems than IRM or LFRM, despite the somewhat increased complexity of the model. The empirical analysis presented here suggests that for all three models convergence to a posterior mode is quite rapid, and while moving between modes is more challenging, it does occur in a computationally feasible number of MCMC iterations. A final note to make is that while MCMC convergence diagnostics are important, if the goal is to obtain good predictive performance a single MCMC run can be viewed as a speed-accuracy tradeoff (see Section 7).

9 Conclusion

The evaluation and convergence tests indicate that our proposed MCMC sampler for ILA is capable of rapidly reaching a posterior mode, and eventually moving between isolated posterior modes. While more elaborate sampling techniques, such as split-merge proposals for the IBP

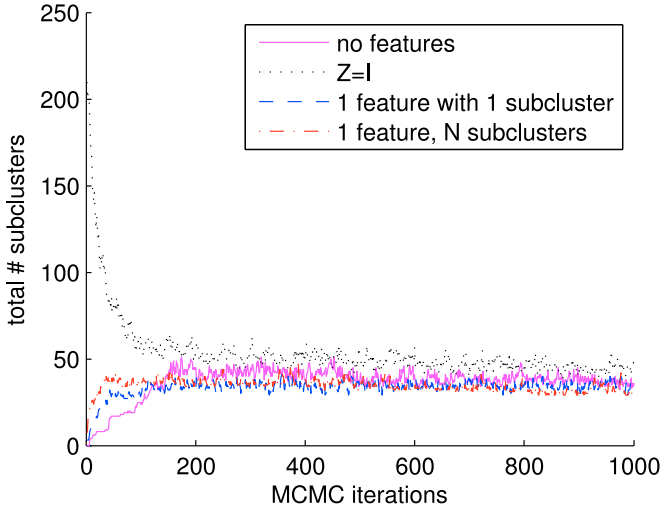


Fig. 15. Convergence properties of the ILA (logistic) MCMC sampler assessed on the NIPS coauthorship data set ($N = 234$) by using extreme initialisations. Despite very different initialisations each chain converges after around 200 MCMC iterations to a very similar posterior mode measured in terms of the total number of subclusters.

features could be used allowing more global, rather than incremental, moves, the model's strong empirical results at the link prediction task suggest that even the current, simple sampler finds meaningful latent structure.

Using an alternative noisy-OR likelihood model allows us to derive an MCMC implementation which scales linearly with the number of links in the observed network. This quantity grows more slowly than the *potential* number of links, N^2, in the "sparse" graph structures typical of real networks. As a result, using this likelihood we are able to run ILA on a data set with $N = 1{,}139$ nodes, and find biologically meaningful subclusters of interest using the inferred structure.

With the exception of Roy et al. [20] the latent variable models proposed to date for network data extract only a flat clustering, be it overlapping or not, of the the nodes in the network. Our experimental results on two very different data sets suggest that such models fail to capture the complex nature of real world networks. ILA, however, is able to naturally represent this complexity using overlapping features which are divided into subclusters, with corresponding gains in empirical performance.

Acknowledgments

The authors would like to thank Peter Orbanz for helpful discussion and suggestions for improving the manuscript. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant Numbers EP/I036575/1 and EP/H019472/1, and the Microsoft Research Roger Needham Scholarship of Wolfson College, Cambridge.

References

[1] E. M. Airoldi, D. M. Blei, E. P. Xing, and S. E. Fienberg, "Mixed membership stochastic block models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 33–40.

[2] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statist. Sci.*, vol. 7, pp. 457–511, 1992.

[3] J. Geweke, "Getting it right," *J. Amer. Statist. Assoc.*, vol. 99, no. 467, pp. 799–804, 2004.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, pp. 7821–7826, 2002.

[5] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *J. Mach. Learn. Res.*, vol. 8, pp. 2265–2295, 2007.

[6] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 475–482.

[7] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Amer. Statist. Assoc.*, vol. 97, pp. 1090–1098, 2001.

[8] M. C. Jonikas, S. R. Collins, V. Denic, E. Oh, E. M. Quan, V. Schmid, J. Weibezahn, B. Schwappach, P. Walter, J. S. Weissman, and M. Schuldiner, "Comprehensive characterization of genes required for protein folding in the endoplasmic reticulum," *Science*, vol. 323, pp. 1693–1697, 2009.

[9] C. Kemp and J. B. Tenenbaum, "Learning systems of concepts with an infinite relational model," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 381–388.

[10] M. Kim and J. Leskovec, "Modeling social networks with node attributes using the multiplicative attribute graph model," in *Proc. Annu. Conf. Uncertainty Artif. Intell.*, 2011, pp. 400–409.

[11] E. Meeds, Z. Ghahramani, R. M. Neal, and S. T. Roweis, "Modeling dyadic data with binary latent factors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 475–482.

[12] K. Miller, T. Griffiths, and M. Jordan, "Nonparametric latent feature models for link prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1276–1284.

[13] M. Mørup, M. N. Schmidt, and L. K. Hansen, "Infinite multiple membership relational modeling for complex networks," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2011, pp. 1–6.

[14] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Comput. Graph. Statist.*, vol. 9, pp. 249–265, 2000.

[15] Radford M. Neal, "Slice sampling," *Ann. Statist.*, vol. 31, pp. 705–767, 2003.

[16] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," *J. Amer. Statist. Assoc.*, vol. 96, pp. 1077–1087, 2001.

[17] K. Palla, D. A. Knowles, and Z. Ghahramani, "An infinite latent attribute model for network data," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 572–580.

[18] J. Pitman, "Combinatorial stochastic processes," Dept. Statist., Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep. 621, 2002.

[19] A. E. Raftery and S. Lewis, "How many iterations in the Gibbs sampler?" in *Proc. Bayesian Stat. 4: 4th Valencia Int. Meeting* 1992, pp. 763–773.

[20] D. M. Roy, C. Kemp, V. Mansinghka, and J. B. Tenenbaum, "Learning annotated hierarchies from relational data," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1185–1192.

[21] S. M. Ruben, J. F. Klement, T. A. Coleman, M. Maher, C. H. Chen, and C. A. Rosen, "I-Rel: A novel rel-related protein that inhibits NF-kappa B transcriptional activity," *Genes Develop.*, vol. 6, no. 5, pp. 745–760, 1992.

[22] Z. Xu, V. Tresp, K. Yu, and H. P. Kriegel, "Infinite hidden relational models," in *Proc. 22nd Conf. Annu. Conf. Uncertainty Artif. Intell.*, 2006, pp. 544–551.

Konstantina Palla received the undergraduate degree in electrical and computing engineering from the National and Technical University of Athens and the master's degree in informatics from the University of Edinburgh. She is currently working toward the PhD degree in the Engineering Department at the University of Cambridge. She is a member of the Machine Learning Group under the supervision of Professor Zoubin Ghahramani. Her current research interest includes nonparametric Bayesian methods.

David A. Knowles received the undergraduate degree from the University of Cambridge which comprised two years of physics before switching to engineering to complete the MEng degree with Professor Ghahramani, the master's degree in bioinformatics and systems biology from the Imperial College London, and the PhD degree with Zoubin Ghahramani in the Machine Learning Group of the Cambridge University Engineering Department, during which he worked part-time at Microsoft Research Cambridge developing Infer.NET. He is a postdoctoral researcher with Daphne Koller in the Computer Science Department at Stanford University. His research involves both the development of novel machine learning methods and their application to data analysis problems in biology.

Zoubin Ghahramani studied computer science and cognitive science at the University of Pennsylvania, received the PhD degree from MIT in 1995, and was a postdoctoral fellow at the University of Toronto. He is a professor of information engineering at the University of Cambridge. His academic career includes concurrent appointments at the Gatsby Computational Neuroscience Unit in London, and as a faculty member of CMU's Machine Learning Department for more than 10 years. His current research focuses on nonparametric Bayesian modelling and statistical machine learning. He has more than 200 publications in computer science, statistics, engineering, and neuroscience. He has served on the editorial boards of several leading journals in the field, including *Journal of Machine Learning Research*, *Journal of Artificial Intelligence Research*, *Annals of Statistics, Machine Learning, Bayesian Analysis*, and was an associate editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*.