Int. J. Mach. Learn. & Cyber. (2016) 7:479–489
DOI 10.1007/s13042-015-0439-1

ORIGINAL ARTICLE

Structure learning for weighted networks based on Bayesian nonparametric models

Xiaojuan Jiang[1] · **Wensheng Zhang**[1]

Received: 10 February 2015 / Accepted: 30 September 2015 / Published online: 23 October 2015

Abstract With the increase of availability and scope of complex networks, structure learning for networks has received an enormous amount of interest in many fields, including physics, computer and information sciences, biology and the social sciences. To extract compact and flexible representations for weighted networks, we propose a new Bayesian nonparametric model to learn from both the existence and weight of interactions between nodes. Our model adopts Dirichlet process prior to automatically infer the partition over nodes in weighted networks without specifying the number of clusters. This is vital for structure discovery in complex networks, especially for novel domains where we have little prior knowledge. We develop a mean-field variational algorithm to efficiently approximate the model's posterior distribution over infinite latent clusters. Conducting extensive experiments on synthetic data set and four popular data sets, we demonstrate that our model can effectively capture the latent structure for complex weighted networks.

Keywords Structure learning · Clustering · Probabilistic graph models · Bayesian nonparametric models · Variational inference

✉ Wensheng Zhang
wensheng.zhang@ia.ac.cn

Xiaojuan Jiang
xiaojuan.jiang@ia.ac.cn

[1] Institute of Automation, University of Chinese Academy of Sciences, Beijing 100190, People's Republic of China

1 Introduction

Statistical analysis of complex networks has been an active area of research for decades, and is becoming an increasingly important challenge in pattern recognition and machine learning [12, 30]. Consisting of pairwise measurements, such as existence or absence of links between pairs of objects, networks have been used to analyze interpersonal social relationships [30], communication networks [31], academic paper co-authorships and citations [33], protein interactions [25], gene regulatory patterns [17], and much more [12]. Unlike traditional attribute data collected from individual objects, the observations in networks are no longer independent or exchangeable because objects are pairwise related. Independence or exchangeability is a key assumption made in machine learning and statistics for traditional attribute data [3, 6]. This intrinsic difference in structure requires special treatments for network data.

A central problem in the network literature is to uncover the latent structure based on the observed pairwise interactions between objects [11, 12]. Among all the statistical models proposed for this end, Stochastic Block Model (SBM) [15, 31, 41] is an elegant probabilistic graph model of block structure in unweighted networks. Probabilistic graph models [39] are perfect integration of probability theory and graph theory. They provide a natural tool for dealing with uncertainty that occurs throughout applied mathematics and engineering. Apart from probabilistic graph models, a variety of different approaches exist to process uncertainty; see, for example, [40, 42–44].

SBM assumes that there are a number of clusters such that each object in the network belongs to a single cluster. Objects in the same cluster are structurally equivalent, which means that their connectivity with other objects is

similar. Under this assumption, the link probability between two objects depends only on their cluster assignments. It is notable that the partition of objects induced by SBM is based on the similarity of interactions between objects, and this similarity can be viewed as a generalization of similar measure defined for traditional attribute data [6, 47]. With further assumptions on inter-block and intra-block connectivity, SBM has been successfully used for modeling assortative network structure [28], disassortative structure [30] and bipartite structure [23]. And SBM also has been generalized for count-valued data, degree correction [19] and categorical values [13]. The Mixed Membership Stochastic Block Model [2] further increases the expressiveness of SBM by allowing mixed membership, associating each object with a distribution vector over clusters.

Most of these models share a basic assumption that the networks are unweighted, where the interaction existence or absence is represented as a binary variable. However, most real-world networks contain information about link weights. For instance, in social networks the weights represent the strengths of social ties between people [30], while in biological networks such as the connectome (i.e., networks formed by neurons connections to each other) weights can code the number of links that exist between neurons [45]. A common technique, employed to conduct analysis on weighted networks, is transforming the data into the binary framework via thresholding [29]. But the potential loss of information caused by thresholding may lead to obscuration or distortion in recovering underlying structure [1, 38].

Instead of thresholding, to directly learn the latent structure of weighted networks, an extension of SBM with Poisson likelihood [19, 24] was considered for count-valued pairwise interactions. Recently, a generalization of SBM, called Weighted Stochastic Block Model (WSBM), was introduced in [1] to learn the latent structure by combining the link-existence and link-weight information.

Although very powerful, all these models require one to specify the number of latent clusters (or blocks), which may be very difficult to access for real-world networks. Usually, this parameter is tuned via a computational expensive model selection procedure, such as minimum description length [34], or Bayes factors [1, 14]. To relax the finite-cardinality assumption on the latent clusters, the Infinite Relational Model (IRM) [20] and the Infinite Hidden Relational Model [46] use the Dirichlet process prior to define a nonparametric relational model for unweighted networks.

In this paper, we introduce the Weighted Infinite Relational Model (WIRM), a Bayesian nonparametric model that can learn a potentially infinite number of clusters from both the existence and weight of links. We treat each weighted link as a draw from a parametric exponential distribution family. The exponential families include many of the most common distributions, which enables us to directly use the weight information in recovering the latent block structure. Moreover, WIRM uses a nonparametric Bayesian approach to simultaneously infer the number of latent clusters, the cluster membership of each object, and how cluster membership influences the observed weighted interactions.

The paper is arranged as follows. We first describe the generative process of our model in Sect. 2. Section 3 explains the relationship of our model to two popular models. We then derive a variational inference algorithm for performing approximate posterior inference and parameter estimation in Sect. 4. Section 5 compares the performance of the WIRM to alternative methods for structure learning and two link prediction tasks, and analyzes the results. Finally, Sect. 6 concludes the paper.

2 Weighted infinite relational model

Consider a directed relational network of N objects. Let A be a $N \times N$ matrix that contains links information among objects. The direction information is contained in the matrix $A = [A_{ij}]$. For example, A_{ij} represents the directed link information from object i to object j, so A_{ij} is not necessarily equal to A_{ji}. Here, we assume that there are two types of information in the link observations: information about existence (presence or absence of links) and information about weights (the weighted values). To specify these two types of information in the network, we can take the adjacency matrix $\mathbf{A}$ as a binary-valued matrix or a real-valued (or count-valued) matrix. Our goal is to partition the set of objects into clusters, so that the relationships between objects can be predicted by their cluster assignments. The number of latent clusters present in the network, which is not known a prior, is denoted by K, so that the cluster assignment variable of object i is $z_i \in \{1, 2, \ldots, K\}$.

2.1 Modeling observed link information

Suppose we are given the cluster assignment vector $\mathbf{Z} = \{z_1, \ldots, z_N\}$ that represents a partition of the N objects into K clusters. For each pair of clusters (kk'), we can model the 'bundle' of links from objects in cluster k to those in cluster k', using an exponential distribution family parameterized by $\theta_{kk'}$. That is, for object i with cluster assignment $z_i = k$ and object j with $z_j = k'$, the likelihood of observing a link A_{ij} from object i to object j is given by

$$P(A_{ij}|Z, \theta) \propto \exp\left[T(A_{ij}) \cdot \eta(\theta_{z_i z_j})\right] \quad (1)$$

where T is the vector valued function of sufficient statistics, and η is the vector valued function of natural parameter.

Exponential distribution family [6, 39] provides a powerful uniform representation for different probability distributions. Indeed, by choosing different function pairs (T, η), we can use the exponential family to express many continuous or discrete distributions, including the Bernoulli, the Multinoulli, the Gaussian, the Poisson, the Gamma, the Geometric, the NegBinomial, etc. Members of the exponential family have some important properties. For example, the exponential family is the only family of distributions for which conjugate priors exist; this simplifies the computation of posteriors. Moreover, the exponential family is at the core of variational inference; by exploiting the conjugate duality between the cumulant function and the entropy for exponential families, a wide variety of variational representations for different probabilistic inference problems were developed in [39].

To model binary existence information of the links, the Bernoulli distribution (with success rate p) could be a good choice, and we can set the sufficient statistics and the natural parameters pair of the exponential family as: $T = (x, 1)$ and $\eta = (\log[p/(1-p)], \log[1-p])$. For count-valued existence information of links, we may choose the Poisson distributions with mean parameter λ, and the corresponding function pair can be set as: $T = (x, 1)$ and $\eta = (\log \lambda, -\lambda)$. To model real-valued weight information of links, we may choose the Gaussian distribution with mean μ and variance σ^2, which has sufficient statistics $T = (x, x^2, 1)$ and natural parameters $\eta = (\mu/\sigma^2, -1/(2\sigma^2), -\mu^2/(2\sigma^2))$.

Let $A^{(e)}$ be the link-existence observation and $A^{(w)}$ be the link-weight observation. If the pair (T_e, η_e) denotes the family of link-existence distributions and (T_w, η_w) denotes the family of link-weight distributions, then we may incorporate these two types of information into the likelihood function via a simple relative importance parameter $c \in [0, 1]$:

$$\log P(A_{ij}^{(e)}, A_{ij}^{(w)} | Z, \theta) \propto c T_e(A_{ij}^{(e)}) \cdot \eta_e(\theta_{z_i z_j}^{(e)}) + (1-c) T_w(A_{ij}^{(w)}) \cdot \eta_w(\theta_{z_i z_j}^{(w)}). \quad (2)$$

2.2 Nonparametric prior on cluster assignment

In order to allow flexible inference of the latent structure of data, we set the number of possible clusters K to be infinity by using the Dirichlet process prior. The Dirichlet process, introduced by Ferguson [10], is the underlying random measure of the Chinese restaurant process (CRP) [3, 35], which is widely used as a nonparametric prior for latent class models. A distinguishing characteristic of the prior is that conditioned on data, we examine the posterior distribution of Z to obtain a data-dependent distribution of how many clusters are needed.

The CRP metaphor gives the intuition. Imagine a restaurant with infinite number of tables, each with infinite number of seats. The customers enter the restaurant one after another, and each chooses a table at random. In the CRP with parameter α, each customer chooses an occupied table with probability proportional to the number of occupants, and chooses the next vacant table with probability proportional to α. This process continues until all customers have seats, defining a distribution over allocations of people to tables, and more generally, objects to clusters. It is known that the joint probability of final assignment is not affected by the order of customers getting into the restaurant, which is called exchangeability [35].

The Chinese restaurant construction of Dirichlet process directly leads itself to a Gibbs sampler; whereas for the variational inference of Dirichlet process, we turn to the stick-breaking construction of Sethuraman [37], which provides a concrete set of hidden variables on which to place an approximate posterior [8, 21, 22]. The stick-breaking representation of the cluster assignment $z_i \in \{1, 2, \ldots\}$ is defined as follows:

$$\begin{aligned} &v_k : (1, \alpha), \quad k = 1, 2, \ldots \\ &\pi_k(v) = v_k \prod_{l=1}^{k-1} (1 - v_k), \quad k = 1, 2, \ldots \\ &z_i : (\pi_k(v)), \quad i = 1, \ldots, N \end{aligned} \quad (3)$$

2.3 The full Bayesian model

To perform fully Bayesian inference, we now introduce the prior for link bundle parameter θ. For Bayesian models, if we use conjugate priors, the inference analysis would be considerably simplified, because the posterior distributions have the same functional form as the priors. Given exponential family likelihood (1) with link bundle parameter θ, the standard conjugate prior [6, 39] is

$$p(\theta) = \frac{1}{Z(\tau)} \exp[\tau \cdot \eta(\theta)], \quad (4)$$

where τ parameterizes the prior and $Z(\tau)$ is a normalizing factor.

For notational convenience, we let r index the $K \times K$ link-bundles between clusters; hence $\theta = (\theta_1, \ldots, \theta_r)$. When we update the prior based on the observed links in a given link bundle r, the posterior's parameter becomes $\tau_r = \tau + T_r$, where T_r is the sufficient statistics of the observed links; see Sect. 4.3 for details. Thus we see that the parameter τ can be interpreted as an effective number of pseudo-observations in the prior, which push the

likelihood function away from the degenerate cases so that every link bundle produces a valid and reasonable parameter estimate.

Now, we can summarize the whole generative process of the Weighted Infinite Relational Model:

- For each object i, assign a cluster membership z_i as in (3).
- For each pair of clusters (kk'), draw a link bundle parameter $\theta_{kk'}$ according to (4).
- For each pair of objects with index i and j, draw the link observation A_{ij} from the exponential family in (1).

Note that we can take (2) instead of (1) as the likelihood function in the generative process of WIRM, to represent the existence and weight observations at the same time.

3 Related work

Here, we examine two models that are closely related to WIRM. Using Bernoulli likelihood, the Infinite Relational Model (IRM) [20] previously adapted the Dirichlet process to define a nonparametric model for network modeling. More specifically, the link-existence probability between two objects is

$$P(A_{i,j} = 1|Z, W) = W_{z_i z_j}$$

where the link probabilities for each pair of clusters, $\{W_{kk'} : k, k' = 1, \ldots, K\}$, are given independent Beta priors, and the cluster assignment Z follows a CRP construction of the Dirichlet process. It is notable that IRM is a special case of our model, with $c = 1$ in likelihood defined as (2). This implies that IRM ignores link-weight information, and fits only to the link-existence information. Moreover, the approximate inference of our model is conducted using variational methods, but inference for IRM follows a Markov chain Monte Carlo (MCMC) sampling procedure.

On the other hand, WIRM can also be seen as a nonparametric extension of the WSBM proposed in [1], where the number of clusters K is chosen before the model can be applied to data. WSBM is a generative model that can learn from both the presence and weight of links, using combined likelihood of both types of link information. But the prior on cluster assignment Z is a multinomial distribution over a fixed, finite number of clusters. Compared to WSBM, a distinctive feature of the WIRM is its ability to infer from the observed data that how many latent clusters there are, and learn increasingly complex representations when more observations are encountered. This is vital when we have little prior knowledge about the number of clusters, especially for applications in novel domains.

4 Inference

The WIRM posits a generative probabilistic process of network data that includes hidden structure. Given link observations, our goal is to uncover the underlying structure of the weighted network by inferring the posterior distribution of the latent variables. However, like many interesting Bayesian nonparametric models [4, 20], the posterior distribution of the latent variables under a Dirichlet process Prior is not available in closed form. Here, we apply the variational method to infer the latent variables of WIRM.

Variational inference is a wide-used approach to approximating the posterior in graph models [5, 18]. This approach is based on the idea of approximating the posterior with a simpler family of distributions and searching for the member of that family that is closest to the posterior. This problem is (approximately) solved by optimizing a function equal up to a constant to the KL divergence of the approximate distribution from the true posterior. Compared to the sampling methods based on MCMC [16, 27], variational methods are deterministic, usually more efficient and they have an objective to monitor the convergence behavior [8, 21, 22].

We now represent a mean-field variational algorithm for WIRM with likelihood function defined in (1). For the general case with likelihood defined in (2), the inference algorithm follows with minor modifications, and here we omit the redundant details. To derive the variational optimization, we first propose the truncated mean-field variational distributions in Sect. 4.1, then variational objective function is derived in Sect. 4.2; finally, in Sect. 4.3, we will present an explicit coordinate ascent algorithm for optimizing the objective function with respect to the variational parameters.

4.1 Truncated variational distributions

The hidden variables that we are interested in are the auxiliary stick-breaking variables $V = \{v_1, \ldots, v_K\}$, the cluster assignment vector $Z = \{z_1, \ldots, z_N\}$, and the link parameters $\theta = \{\theta_1, \cdots, \theta_r\}$. We use the truncated stick-breaking representation for variational distributions [8]. By setting $q(v_K = 1) = 1$ for a fixed K, we enforce the proportions $\pi_k(v)$ in (3) to be zero for $k > K$. It is a remarkable fact that our model follows a full Dirichlet process prior which is not truncated; only the variational posterior is truncated. Moreover, the truncation level K is a variational parameter which can be freely set; it is not a part of the prior model specification. If K is large enough, the fitted approximate posterior will exhibit fewer than K clusters.

We use the following fully factorized variational distribution for mean-field variational inference:

$$q(V,Z,\theta)=\prod_{k=1}^{K-1}q(v_k;\gamma_k)\prod_{i=1}^{N}q(z_i;\phi_i)\prod_{r}q(\theta_r;\tau_r), \tag{5}$$

where $q(v_k;\gamma_k)$ are beta distributions, $q(z_i;\phi_i)$ are multinomial distributions, and $q(\theta_r;\tau_r)$ are exponential family distributions with natural parameters τ_r and sufficient statistics $\eta(\theta_r)$.

4.2 Lower bound on the marginal likelihood

Using the standard variational theory, we have the lower bound for marginal log likelihood of the observed data:

$$\begin{aligned}\log p(A)&\geq E_q[\log p(A,V,Z,\theta)]-E_q[\log q(V,Z,\theta)]\\&=E_q[\log p(V|\alpha)]+E_q[\log p(Z|V)]+E_q[\log p(\theta|\tau_0)]\\&\quad+E_q[\log p(A|V,Z,\theta)-E_q[\log q(V,Z,\theta)]\triangleq\mathcal{L}(q),\end{aligned} \tag{6}$$

here and elsewhere in the paper we omit the variational parameters when using q in (5) as a subscript of an expectation.

Now we expand the lower bound $\mathcal{L}(q)$ in (6) with the approximate posterior q in (5). To simplify notation, let $\langle T\rangle_r$ and $\langle\eta\rangle_r$ be the expected values of the sufficient statistics T_r and natural parameters η_r under the approximation distribution q, that is,

$$\langle T\rangle_r=\sum_{i,j}\sum_{(z_i,z_j)=r}\phi_{i,z_i}\phi_{j,z_j}T(A_{i,j}),$$

$$\langle\eta\rangle_r=\frac{\partial}{\partial\tau}\log Z(\tau)|_{\tau=\tau_r}.$$

By substituting q and the conjugate prior p in (6), and evaluating all the expectations, we have:

$$\begin{aligned}\mathcal{L}(q)&=\sum_r(\langle T\rangle_r+\tau_0-\tau_r)\cdot\langle\eta\rangle_r+\sum_r\log\frac{Z(\tau_r)}{Z(\tau_0)}\\&\quad+(K-1)\log\alpha-\sum_{k=1}^{K-1}\log\frac{\Gamma(\gamma_{k,1}+\gamma_{k,2})}{\Gamma(\gamma_{k,1})\Gamma(\gamma_{k,2})}\\&\quad-\sum_{k=1}^{K-1}\{(\gamma_{k,1}-1)E_q[\log v_k]+(\gamma_{k,2}-\alpha)E_q[\log(1-v_k)]\}\\&\quad+\sum_{i=1}^{N}\sum_{k=1}^{K-1}\left\{\sum_{l=k+1}^{K}\phi_{i,l}E_q[\log(1-v_k)]+\phi_{i,k}E_q[\log v_k]\right\}\\&\quad-\sum_{i=1}^{N}\sum_{k=1}^{K}\phi_{i,k}\log\phi_{i,k},\end{aligned} \tag{7}$$

where

$$E_q[\log v_k]=\Psi(\gamma_{k,1})-\Psi(\gamma_{k,1}+\gamma_{k,2}),$$

$$E_q[\log(1-v_k)]=\Psi(\gamma_{k,2})-\Psi(\gamma_{k,1}+\gamma_{k,2}).$$

The digamma function, denoted by Ψ, arises from the derivative of the log normalization factor in the Beta distribution.

4.3 Coordinate ascent algorithm

Now, we present an explicit coordinate ascent algorithm for optimizing the bound (7). We will seek a consistent solution by first initializing all of the factors in q appropriately, and then iteratively optimize the variational lower bound with respect to each factor in turn. Convergence is guaranteed [6, 7] because the bound $\mathcal{L}(q)$ is convex with respect to each of the factors in the variational distribution q.

The details of the iteration are as follows:

Update for the link bundle parameter θ_r The variational distribution for the link bundle parameter θ_r is exponential family with sufficient statistic $\eta(\theta_r)$ and natural parameter τ_r. Coordinate ascent update equation for the variational parameter τ_r is

$$\tau_r=\tau_0+\langle T\rangle_r.$$

Update for the cluster assignment z_i The variational parameter for the cluster assignment z_i is $\{\phi_{i,k}\}_k$, and the update equation for $\{\phi_{i,k}\}_k$ is

$$\phi_{i,k}\propto\exp\left\{E_q[\log v_k]+\sum_{l=1}^{k-1}E_q[\log(1-v_l)]+\sum_r\frac{\partial\langle T\rangle_r}{\partial\phi_{i,k}}\cdot\langle\eta\rangle_r\right\},$$

where $\frac{\partial\langle T\rangle_r}{\partial\phi_{i,k}}=\sum_{(k,l)=r}\sum_{j\neq i}T(A_{ij})\phi_{j,l}$.

Update for the auxiliary stick-breaking variable v_k The variational distribution for the auxiliary stick-breaking variable v_k is a beta distribution parameterized with the shape parameters $(\gamma_{k,1},\gamma_{k,2})$. Coordinate ascent update equations for these free variational parameters are

$$\gamma_{k,1}=1+\sum_i\phi_{i,k},\gamma_{k,2}=\alpha+\sum_i\sum_{l=k+1}^{K}\phi_{i,l}.$$

Algorithm 1 represents coordinate ascent algorithm for variational inference of WIRM. It is pointed out by Blei and Jordan [8] that, although the algorithm yields a bound for any starting value of the variational parameters, poor initialization can lead to local maxima that yields poor bounds. In practice, we run the algorithm multiple times with random initializations and choose the final parameter settings that give the best bound on the marginal likelihood in (7). To further improve the performance, we may apply a sequential initialization scheme [8] (which can be viewed as a variational version of sequential importance sampling). Under this scheme, we initialize the variational distribution by incrementally updating the parameters according to a random permutation of the objects in the network.

Algorithm 1 Coordinate ascent algorithm for variational inference of WIRM

Input: Link-weighted network A, Hyper parameters c, α, Prior parameter τ_0 and Function pairs (T, η)

Output: Variational parameters τ, ϕ, γ

Initialize ϕ

repeat

  for $r = 1, \cdots, K^2$ **do**

    Set $\langle T \rangle_r = \sum_{i,j} \sum_{(z_i, z_j)=r} \phi_{i,z_i} \phi_{j,z_j} T(A_{i,j})$ and $\langle \eta \rangle_r = \frac{\partial}{\partial \tau} \log Z(\tau)\big|_{\tau=\tau_r}$.

    Update $\tau_r = \tau_0 + \langle T \rangle_r$.

  end for

  for $k = 1, \cdots, K$ **do**

    Update $\gamma_{k,1} = 1 + \sum_i \phi_{i,k}$ and $\gamma_{k,2} = \alpha + \sum_i \sum_{l=k+1}^{K} \phi_{i,l}$.

    Compute $E_q[\log v_k] = \Psi(\gamma_{k,1}) - \Psi(\gamma_{k,1} + \gamma_{k,2})$ and $E_q[\log(1 - v_k)] = \Psi(\gamma_{k,2}) - \Psi(\gamma_{k,1} + \gamma_{k,2})$.

  end for

  for $i = 1, \cdots, N$ **do**

    Compute $\frac{\partial \langle T \rangle_r}{\partial \phi_{i,k}} = \sum_{(k,l)=r} \sum_{j \neq i} T(A_{ij}) \phi_{j,l}$.

    Update $\phi_{i,k} \propto \exp\left\{ E_q[\log v_k] + \sum_{l=1}^{k-1} E_q[\log(1 - v_l)] + \sum_r \frac{\partial \langle T \rangle_r}{\partial \phi_{i,k}} \cdot \langle \eta \rangle_r \right\}$.

  end for

until τ, ϕ, γ converge

Return τ, ϕ, γ

5 Experiments

In this section, we evaluate the performance of WIRM on a synthetic data and four real-world networks. Experiments were conducted for two purposes. First, we generate synthetic data to explore the ability of our model to infer the number of latent clusters using both link-existence and link-weight information. Second, for two prediction tasks on real-world datasets, we compare the performance of our model with several state-of-the-art network models.

5.1 Synthetic data

We generate a simple $N = 100$ synthetic dataset with 4 known equal-size clusters; see Fig. 1. The weights and existences of each link bundle are drawn from Gaussian distributions and Bernoulli distributions, respectively, with different bundle-specific parameters. This dataset is carefully designed, so that the bundle-specific parameters are shared in a subtle manner. Specifically, if we only consider the weight information, the nodes can be naturally separated into two equal-size sup-clusters: one is the cluster comprised of nodes indexed by {1,…,50}, the other is comprised of nodes indexed by {51,…,100}; the link weights between objects in different clusters are drawn from $\mathcal{N}(1, 1)$, and the weights between objects among the same cluster are drawn from $\mathcal{N}(10, 1)$; see Fig. 1a. Considering the existence information (ignoring the weights) leads to a different cluster assignment: one cluster is comprised of nodes indexed by {1,…,25,51,…,75}, the other one is comprised of nodes indexed by {26,…,50,76,…,100}; the probability that there is a link between objects in different clusters is 0.1, and the probability that there is a link between objects among the same cluster is 0.9; see Fig. 1b.

To analyze this network, we set the truncation level to be 20 and fit our model with pure weight information by setting $c = 0$, pure existence information by $c = 1$, and mixed information by $c = 0.5$, respectively. The posterior cluster assignments over 20 possible clusters learned by WIRM are shown in Fig. 2. Examining the results, we can see that the latent structure learned by WIRM with $c = 0$ exactly recovers the partition underlying the link-weight information (Fig. 2a), and with $c = 1$ reveals the partition underlying the existence information (Fig. 2b). Moreover, Fig. 2c demonstrates the ability of the fitted model with $c = 0.5$ to recover the ground-truth partition with 4 equal-size clusters, regarding the combination of both information types.

The approximate posterior distribution q in (5) is truncated at $K = 20$. Although we hold K fixed in our simulations, it is possible to optimize K with respect to the approximate lower bound. Indeed, Fig. 3 shows how the optimal bounds on log likelihood for different variants of WIRM change as functions of the truncation level K. We can see that the optimal bounds are relatively stable for different value of $K > 4$. This implies that WIRM can function well, when the truncation level K is reasonably chosen.

We assess the structure exploration performance of WSBM [1] and IRM [20] on this dataset. It is notable that K denotes the truncation level in our model, which can be seen as an upper bound for the true number of clusters; while in WSBM, K stands for the presumed value of the number of clusters. For this synthetic dataset, WSBM with $K = 2$ recovers the latent structure for any single type information of links, and WSBM with $K = 4$ captures the ground-truth partition using the combination of both information types. But the performance of WSBM is very sensitive to K, and the cluster assignment diverges as K increase. On the other hand, IRM can automatically discover the partition underlying the existence information, using the truncation-free MCMC inference procedure; however, it can't reveal the latent structure concerning weight information.

Furthermore, we compare the performance of WIRM, WSBM and IRM in three noisy situations. The first is that we randomly discard a portion of the interactions when training different models (called "salt & pepper"-like noise). The second is that we randomly arrange a portion of the interactions that are randomly chosen from the network, and keep the rest interactions untouched (called "randomly arranging" noise). In these two situations, the noise varies in density (the portion of interactions being discarded or randomly arranged), specifically, {0.05, 0.10, 0.15, 0.20};

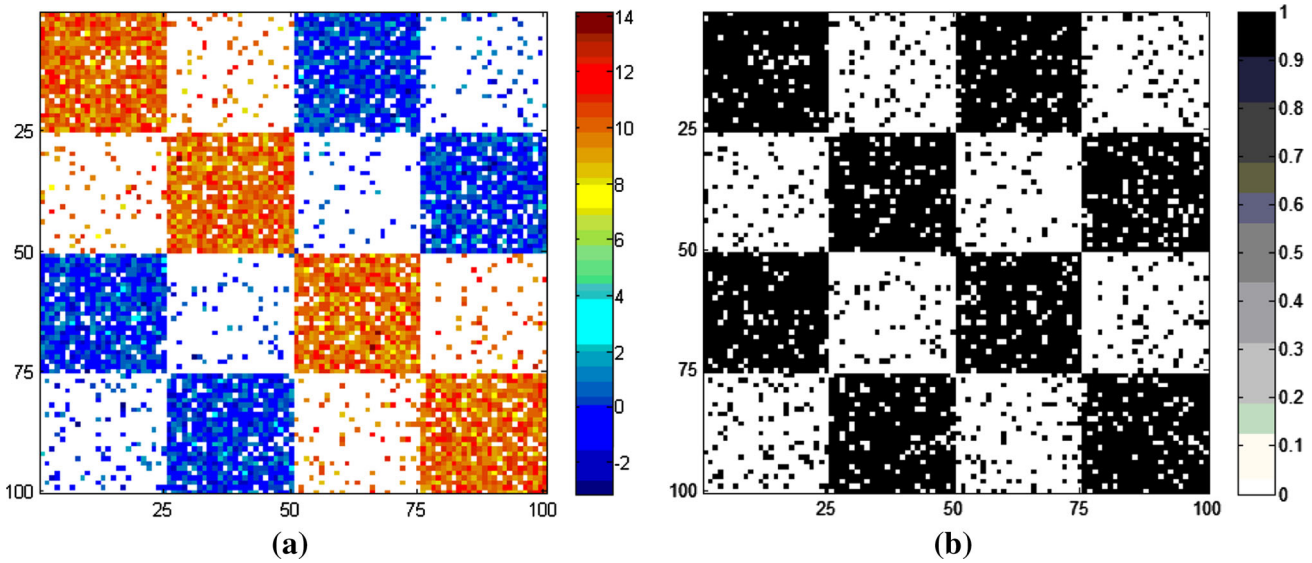

Fig. 1 Observed synthetic data example. **a** Observed synthetic 100 × 100 link-weight matrix. *White* corresponds to unobserved interaction. **b** Observed synthetic 100 × 100 link-existence matrix. *White* corresponds to zero, *black* to one

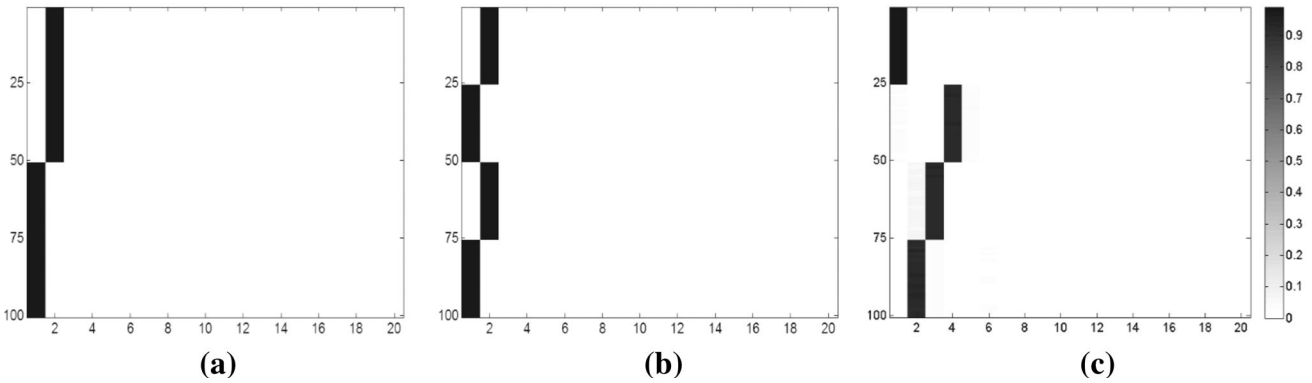

Fig. 2 Results for WIRM. **a** Posterior cluster assignments learned from link-weight information. **b** Posterior cluster assignments learned from link-existence information. **c** Posterior cluster assignments learned from both types of information

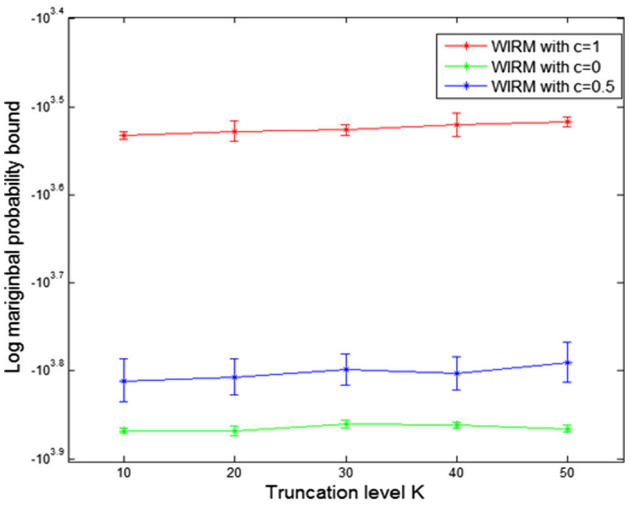

Fig. 3 The optimal bound on the log probability as function of the truncation level

we can see both weight and existence information are affected by the noise. The third situation is simulated by adding Gaussian white noise with standard deviations {1, 10, 20, 30} to the original observed link weights, which doesn't change the existence information.

For each noisy situation and each model, we run 25 independent trials and evaluate the performance using the normalized mutual information (NMI) [26] between the partition learned and the ground-truth partition. We omit the experiment details for saving space, and summarize the findings to our interests as follows. Firstly, for all noisy cases, WIRM with truncation level $K = 20$ performs comparable or slightly worse than WSBM (with carefully hand-chosen parameter K) and IRM. Secondly, the latent structures learned from the existence information are more stable than the structures underlying the weight information, and IRM is more stable than WIRM and WSBM to learn existence information. Moreover, for Gaussian white noise, we found

that those models learning from both weight and existence information perform moderately better than those only fitting weight information (here we only compare those models that can learn from the weight information or both types of information, because the Gaussian noise doesn't affect the existence information). Finally, the results of all models for "salt & pepper"-like noise are relatively better than "randomly arranging" noise. Indeed, in the latter case, by randomly arranging a portion of the interactions which are randomly chosen from the network, the information contained in these interactions is discarded and some misleading information is added; however, the "salt & pepper"-like noise just causes information loss.

5.2 Real-world networks

We now compare our model to several other network models for predicting the existence or the weight of some unobserved interactions on four real-world networks. The weighted networks used for the comparison are given as follows:

Collaboration [33] Nodes represent 226 nations on Earth, and each of the 20,616 links is weighted by a normalized count of academic papers whose author lists include that pair of nations.

Congress [36] Nodes represent the 163 committees in the 102nd US Congress, and each of the 26,569 edges is weighted by the pairwise normalized 'interlock' value of shared members.

Airport [9] This is a network of the 500 busiest commercial airports in the United States, and each of the 5960 directed links is weighted by the number of passengers traveling from one airport to another.

Forum [32] The student social network at UC Irvine includes 1899 users that sent or received at least one message, and each of the 20,291 directed links is weighted by the number of messages sent between users.

We evaluate the following variants of our model: the 'pure' WIRM (pWIRM), using only weight information ($c = 0$), the 'balanced' WSBM (bWIRM), using both link existence and weight information ($c = 0.5$), and the 'non-' WIRM (nWIRM), using only link existence information ($c = 1$). We use Gaussian distribution to model the weight of link interactions, and Bernoulli distribution to model the existence information. A comparative study with the other typical models, (namely, WSBM [1], SBM [31], and IRM [20]), is also performed. The training details of WSBM are same as in [1], and the number of latent clusters for WSBM is fixed at $K = 4$ as therein. We use the publicly available source codes and adopt the original implementations of WSBM provided by the authors. They described an approach for choosing K based on Bayes factors, i.e., choosing K with largest marginal log-likelihood [1]. However, this approach requires to repeatedly train WSBM for different K, and they didn't conduct this scheme on real datasets. We will further discuss this model selection issue in the end of this section.

In both prediction tasks, we treat all networks as directed, and fit each model on 80 % of N^2 interactions, and use the remaining 20 % for test. For all datasets, the truncation level for our model is fixed at $K = 50$. The interactions between two objects are predicted based on their cluster assignments learned. Specifically, for object i with cluster assignment $z_i = k$ and object j with $z_j = k'$, the link A_{ij} follows an exponential distribution family parameterized by $\theta_{kk'}$, moreover, if object i and j are assigned to the same cluster, i.e. $z_i = z_j = k$, then the distribution of A_{ij} is parameterized by $\theta_{kk'}$. For those models that were initially established for unweighted networks (nWIRM, SBM and IRM), we take their partitions and compute the sample mean weight for each of the induced link bundles in the weighted network, and take this value as predictor for the weight of any missing link in that bundle.

For each model and each dataset, we run 25 repeats, each time with a different 80/20 cross-validation split and using a different random initialization, and then compute the average mean-squared error (MSE) on the particular prediction task. Specifically, let $T = \{(i,j)\}$ be the set of links which is hold out for test. The mean squared error is defined as follows:

$$MSE = \frac{1}{|T|} \sum_{(i,j)\in T} \left(A_{ij}^{pred} - A_{ij}\right)^2$$

where $|T|$ is the cardinality of set T, i.e. the total number of hold-out links, and A_{ij}^{pred} is the predicted value of link from object i to object j.To compare the results across different datasets, we normalized link-weights to the interval $[-1, 1]$ after applying a logarithmic transform.

The average running time for a single iteration of the coordinate ascent algorithm (see Algorithm 1) for WIRM is shown in Table 1. We find that the average running time for a single iteration increases almost linearly as the number of nodes in the networks increases, which implies the efficiency of our algorithm. Moreover, we represent the average training time for all the models in Table 2, in order to provide a fair comparison. We can see that the whole

Table 1 Average running time (in seconds) of single iteration for different variants of WIRM

	pWIRM	bWIRM	nWIRM
Collaboration	0.0119	0.0182	0.0145
Congress	0.0952	0.0198	0.0137
Airport	0.0258	0.0548	0.0309
Forum	0.2691	0.6228	0.3486

Table 2 Average training time (in seconds) for different models

	pWIRM	bWIRM	nWIRM	pWSBM	bWSBM	SBM	IRM
Collaboration	1.87	2.19	0.79	1.01	1.68	0.47	13.39
Congress	0.76	2.03	0.70	0.25	0.68	0.42	10.35
Airport	8.53	34.46	7.24	3.81	12.77	2.58	48.90
Forum	18.89	47.08	22.86	4.56	23.05	10.26	225.90

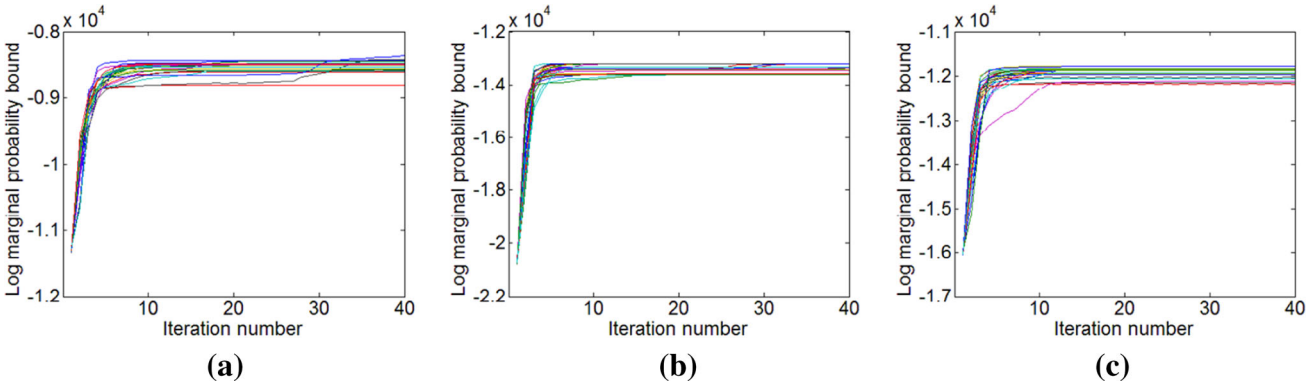

Fig. 4 Log marginal probability bound during iterations for pWIRM in (**a**), nWIRM in (**b**) and bWIRM in (**c**) on Collaboration dataset with 25 randomly initialized runs

Table 3 Average (std) of the (expected) number of clusters learned by different models

	pWIRM	bWIRM	nWIRM	IRM
Collaboration	17.4 (1.1)	18.3 (0.9)	17.0 (0.8)	16.9 (1.0)
Congress	30.2 (1.7)	32.4 (1.4)	29.9 (1.6)	29.6 (1.4)
Airport	11.0 (0.8)	11.8 (0.7)	10.8 (0.6)	10.9 (0.7)
Forum	23.4 (1.6)	25.1 (1.5)	23.2 (1.4)	22.2 (1.6)

training procedure of WIRM usually takes two to three times as that of WSBM, and IRM is more computationally expensive than others because MCMC algorithms take long time to converge.

To demonstrate the efficiency and stability of our approximate inference algorithm, we examine the change of the log marginal probability bound during the iterations. The results on Collaboration dataset is shown in Fig. 4. The results on the other three datasets are similar, and we omit it. It is found that on all datasets, the bound converges within several iterations, and then keeps relatively stable in the following iterations.

Both WIRM and IRM are Bayesian nonparametric models, which assume that the number of clusters is not known a priori and use Dirichlet Process to determine the number of latent clusters. In Table 3, we list the numbers of latent clusters learned by WIRM and IRM. It is also notable that the estimated numbers of clusters are similar for all methods, and by incorporating two types of link information, bWIRM usually divides networks into more clusters. Moreover, the numbers of clusters learned by WIRM on all datasets are not affected by the fixed truncation level K.

We now report the prediction results. Tables 4 and 5 represent the results for predicting link-existences and link-weights, respectively. The bolded values denote the best MSE across all models, and parentheses indicate the uncertainty (standard error) in the last digit. Clearly, for the link-existence prediction, nWIRM and IRM outperform WSBM and SBM by using nonparametric priors. And pWIRM performs well for the link-weight prediction, as it is designed to learn only from weight information. We also notice that, bWIRM is very competitive on both tasks, which reveals its capability to learn both types of information simultaneously without confusing each other.

Finally, it is important to highlight, that WIRM avoids the model selection procedure by using a nonparametric Bayesian approach. Given Dirichlet process prior, we can simultaneously infer the number of latent clusters, the cluster assignment for each object, and how cluster assignment influences the observed interactions. Although the training procedure for WIRM usually takes two to three times as that for WSBM with $K = 4$, it may take much longer if we conduct the whole model selection scheme to tune K for WSBM. We also try fitting WSBM with the estimated number of clusters by WIRM, and the results show that its prediction performance is quite similar to WIRM. This is because WIRM is the nonparametric extension of the WSBM, the latent structure learned by

Table 4 Average MSE on link existence prediction in 25 randomly initialized trials

	pWIRM	bWIRM	nWIRM	pWSBM	bWSBM	SBM	IRM
Collaboration	0.0734 (3)	0.0691 (2)	**0.0680 (4)**	0.1446 (3)	0.1167 (3)	0.1138 (3)	**0.0683 (4)**
Congress	0.1315 (9)	0.1280 (7)	**0.1204 (3)**	0.1765 (4)	0.1648 (4)	0.1640 (5)	**0.1252 (5)**
Airport	0.0100 (1)	0.0085 (2)	**0.0069 (1)**	0.0202 (1)	0.0156 (1)	0.0158 (1)	**0.0064 (3)**
Forum	0.00546 (1)	0.00514 (1)	**0.00503** (2)	0.00560 (1)	0.00535 (1)	0.00535 (1)	0.00522 (1)

Table 5 Average MSE on link weight prediction in 25 randomly initialized trials

	pWIRM	bWIRM	nWIRM	pWSBM	bWSBM	SBM	IRM
Collaboration	0.0413 (2)	0.0461 (2)	0.0547 (2)	**0.0407** (1)	0.0462 (1)	0.0497 (3)	0.0789 (4)
Congress	**0.0412 (6)**	0.0428 (3)	0.0451 (4)	0.0571 (4)	0.0594 (4)	0.0634 (6)	0.0432 (5)
Airport	**0.0158** (7)	0.0180 (4)	0.0223 (2)	0.0486 (6)	0.0543 (5)	0.0632 (8)	0.0222 (6)
Forum	**0.0490** (5)	0.0504 (4)	0.0516 (3)	0.0726 (3)	0.0845 (3)	0.0851 (4)	0.0543 (4)

WSBM with the estimated value of K by WIRM, is probably the same as that by WIRM. On the other hand, WIRM can learn both existence and weight information of links, using an efficient variational inference procedure, while the nonparametric model IRM fits only to the existence information by a time-consuming MCMC sampling scheme.

6 Conclusions

In this paper, we propose a novel Bayesian nonparametric model to generalize the classic infinite relation model to the important case of weighted networks. This model adopts Dirichlet Process prior, in order to learn the number of latent clusters and the cluster assignment from the data. An efficient variational inference algorithm is developed to approximate the posterior distributions. The empirical results show that our model can efficiently capture the complex latent structure of weighted networks, and accurately predict the missing interactions and their weights. Our future work is to extend our study to some more complicated networks, including dynamic networks, and bipartite networks, etc.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Grant No. 61472423, 61432008, 61532006, U1135005). We thank the reviewers for their helpful comments and constructive suggestions which improved the paper greatly.

References

1. Aicher C, Jacobs AZ, Clauset A (2014) Learning latent block structure in weighted networks. J Complex Netw. doi:10.1093/comnet/cnu026
2. Airoldi EM et al (2008) Mixed membership stochastic block models. J Mach Learn Res 9:1981–2014
3. Aldous DJ (1985) Exchangeability and related topics. Lect Notes Math 1117:1–198
4. Antoniak CE (1974) Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. Ann Stat 2(6):1152–1174
5. Attias H (2000) A variational Bayesian framework for graphical models. In: Advances in neural information processing systems. MIT Press, Cambridge, MA, pp 209–215
6. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
7. Blei DM, Jordan MI (2006) Variational inference for Dirichlet process mixtures. Bayesian Anal 1(1):121–143
8. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, New York
9. Colizza V, Pastor-Satorras R, Vespignani A (2007) Reaction–diffusion processes and metapopulation models in heterogeneous networks. Nat Phys 3(4):276–282
10. Ferguson TS (1973) A Bayesian analysis of some nonparametric problems. Ann Stat 209–230
11. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826
12. Goldenberg A et al (2010) A survey of statistical network models. Found Trends Mach Learn 2(2):129–233
13. Guimerà R, Sales-Pardo M (2013) A network inference method for large-scale unsupervised identification of novel drug-drug interactions. PLoS Comput Biol 9(12):e1003374
14. Hofman JM, Wiggins CH (2008) Bayesian approach to network modularity. Phys Rev Lett 100(25):258701
15. Holland PW, Laskey KB, Leinhardt S (1983) Stochastic block models: first steps. Soc Netw 5(2):109–137
16. Jain S, Neal RM (2004) A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. J Comput Graph Stat 13(1):158–182
17. Jonikas MC et al (2009) Comprehensive characterization of genes required for protein folding in the endoplasmic reticulum. Science 323(5922):1693–1697
18. Jordan MI et al (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233
19. Karrer B, Newman MEJ (2011) Stochastic block models and community structure in networks. Phys Rev E 83(1):016107
20. Kemp C et al (2006) Learning systems of concepts with an infinite relational model. In: Proceedings of the 21 national

conference on artificial intelligence (AAAI), Boston, Massachusetts, 16–20 July 2006
21. Kurihara K, Welling M, Vlassis N A (2006) Accelerated variational Dirichlet process mixtures. In: Proceedings of neural information processing systems (NIPS)
22. Kurihara K, Welling M, Teh Y W (2007) Collapsed variational Dirichlet process mixture models. In: IJCAI. Morgan Kaufmann Publishers, Burlington, Massachusetts, pp 2796–2801
23. Larremore DB, Clauset A, Jacobs AZ (2014) Efficiently inferring community structure in bipartite networks. Phys Rev E 90(1):012805
24. Mariadassou M, Robin S, Vacher C (2010) Uncovering latent structure in valued graphs: a variational approach. Ann Appl Stat 4(2):715–742
25. Middendorf M, Ziv E, Wiggins CH (2005) Inferring network mechanisms: the Drosophila melanogaster protein interaction network. Proc Natl Acad Sci USA 102(9):3192–3197
26. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press, Cambridge
27. Neal RM (2000) Markov chain sampling methods for Dirichlet process mixture models. J Comput Graph Stat 9(2):249–265
28. Newman MEJ (2003) Mixing patterns in networks. Phys Rev E 67(2):026126
29. Newman MEJ (2004) Analysis of weighted networks. Phys Rev E 70(5):056131
30. Newman MEJ (2010) Networks: an introduction. Oxford University Press, New York
31. Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic block structures. J Am Stat Assoc 96(455):1077–1087
32. Opsahl T, Panzarasa P (2009) Clustering in weighted networks. Soc Netw 31(2):155–163
33. Pan RK, Kaski K, Fortunato S (2012) World citation and collaboration networks: uncovering the role of geography in science. Sci Rep. doi:10.1038/srep00902
34. Peixoto TP (2013) Parsimonious module inference in large networks. Phys Rev Lett 110(14):148701
35. Pitman J (2002) Combinatorial stochastic processes. In: Technical report 621. Dept. Statistics, UC Berkeley
36. Porter MA, Mucha PJ, Newman MEJ et al (2005) A network analysis of committees in the US House of Representatives[J]. Proc Natl Acad Sci USA 102(20):7057–7062
37. Sethuraman J (1991) A constructive definition of Dirichlet priors. Stat Sin 4:639–650
38. Thomas AC, Blitzstein JK (2011) Valued ties tell fewer lies: why not to dichotomize network edges with thresholds. arXiv:1101.0788
39. Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Found Trends Mach Learn 1(1–2):1–305. doi:10.1561/2200000001
40. Wang X, Dong C (2009) Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy. IEEE Trans Fuzzy Syst 17(3):556–567
41. Wang YJ, Wong GY (1987) Stochastic block models for directed graphs. J Am Stat Assoc 82(397):8–19
42. Wang X, Dong L, Yan J (2012) Maximum ambiguity based sample selection in fuzzy decision tree induction. IEEE Trans Knowl Data Eng 24(8):1491–1505
43. Wang X, He Y, Wang D (2014) Non-naive Bayesian classifiers for classification problems with continuous attributes. IEEE Trans Cybern 44(1):21–39
44. Wang X, Xing H, Li Y et al (2014) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst. doi:10.1109/TFUZZ.2014.2371479
45. White JG et al (1986) The structure of the nervous system of the nematode Caenorhabditis elegans: the mind of a worm. Phil Trans R Soc Lond 314:1–340
46. Xu Z et al (2006) Infinite hidden relational models. In: Proceedings of the 22nd conference on uncertainty in artificial intelligence (UAI), MA, USA, 13–16 July 2006
47. Yeung DS, Wang XZ (2002) Improving performance of similarity-based clustering by feature weight learning. IEEE T Pattern Anal 24(4):556–561