Nonparametric BLOG

Peter Carbonetto, Jacek Kisynski, Nando de Freitas and David Poole
Dept. of Computer Science
University of British Columbia
Vancouver, BC, Canada V6T 1Z4

Abstract

The BLOG language was recently developed for defining first-order probability models over worlds with unknown numbers of objects. It handles important problems in AI, including data association and population estimation. This paper extends the expressiveness of the BLOG language by adopting generative processes over function spaces — known as nonparametrics in the Bayesian literature. We introduce syntax for reasoning about arbitrary collections of objects, and their properties, in an intuitive manner. By exploiting exchangeability, distributions over unknown objects and their attributes are cast as Dirichlet processes, which resolve difficulties in model selection and inference caused by varying numbers of objects. We demonstrate these concepts with applications to air traffic control and citation matching.

1 Introduction

Probabilistic first-order logic has played a prominent role in recent attempts to develop more expressive models in artificial intelligence [3, 4, 6, 8, 9, 12, 16, 17, 18]. Among these, the Bayesian logic (BLOG) approach stands out for its ability to handle unknown numbers of objects and data association in a coherent fashion, and it does not assume unique names and domain closure.

A BLOG model specifies a probability distribution over possible worlds of a typed first-order language. That is, it defines a probability model over objects and functions. A model structure corresponds to a possible world, which is obtained by extending each object type and interpreting each function symbol. BLOG allows one to define guaranteed objects or to specify functions for generating unknown objects. For example, in the aircraft tracking domain (Sec. 5.2), times and radar blips are known, and we infer the unknown aircraft objects. BLOG is therefore a generative probabilistic language where one samples values of functions, objects and the number of objects of each type.

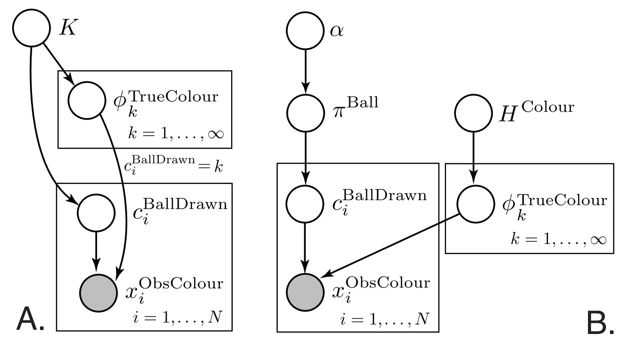

Figure 1: The BLOG directed graphical model model *(a)* and the Dirichlet process mixture (DPM) *(b)* for the ball-and-urn example. In *(a)*, the number of balls is sampled from the Poisson and the choice among the available balls is made uniformly, yielding a sample of x_i^{ObsColor}. In model *(b)*, one samples from an infinite set of colours, but observes only a finite set, according the the prior and data.

In this paper, we introduce Nonparametric BLOG (NP-BLOG), a language which extends the original framework developed in [12]. NP-BLOG is distinguished by its ability to handle object attributes, and collections of attributes, which depend on unbounded sets of objects. We extend the BLOG language by adopting Bayesian nonparametrics, which are probabilistic models with infinitely many parameters [1]. Nonparametric models specify distributions over function spaces, which is precisely what we need in probabilistic first-order inference. Let us illustrate this hypothesis starting with a simple example, and then following up with a more interesting domain.

Fig. 1A shows a simple BLOG model for a variation of the problem explored in [11]. An urn contains a set of balls of various colours. We draw several balls with replacement and observe their colour, corrupted by noise in the sampling process. Since the BLOG model is generative it is possible to sample from the priors and weight the samples by the likelihood in order to answer two important ques-

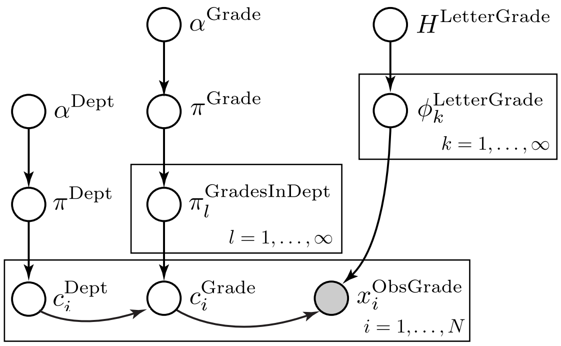

Figure 2: The directed graphical model for the grades and departments domain. We highlight the fact that $\pi_l^{\text{GradesInDept}}$ specifies a distribution over the unknown grades in each department l, and π^{Grade} is a global distribution over grades. Given that we know department to which a grade x_i belongs, $c_i^{\text{Dept}} = l$, the model generates the referring grade $c_i^{\text{GradesInDept}} \sim \phi_l^{\text{GradesInDept}}$. NP-BLOG can automatically generate this graphical model.

tions: How many coloured balls are there in the urn? Do two draws correspond to the same ball in urn?

Let us now consider a nonparametric model, known as a Dirichlet process mixture, for the ball-and-urn example (shown in Fig. 1B). This model, which is explained in detail in Sec. 3, also estimates the number of balls and carry out data association in infinite spaces. Significantly, it also enables us to estimate the probability that a future, unseen draw will be of colour, say Red, given our previous observations. That is, one can infer $P(x_{N+1}^{\text{ObsColour}} = \mathsf{Red} | x_{1:N}^{\text{ObsColour}})$ even if none of the balls 1 to N were observed to be Red. In BLOG, one cannot trivially talk about the probability of observing a colour that hasn't been previously observed.

To further stress the advantage of nonparametrics, we now consider the hierarchical Dirichlet process model described in Fig. 2. The objective is to model marks handed out by computer science departments at different universities for the purposes of, say, graduate admissions. This nonparametric model can infer the distribution of marks $\pi_l^{\text{GradesInDept}}$ for each department l, without having to specify the number of CS departments beforehand. In other words, if one didn't know about the McGill School of CS a priori (it is one of the unknown objects), BLOG cannot address the distribution of grades in this department, but Dirichlet process models can. Moreover, we can talk about the probability that a student that received an A in a course at MIT will receive the same standing at UBC. The grades and departments model is an example of a more general theme that appears throughout this paper: nonparametrics allows us to infer distributions characteristic to specific objects, even when we don't know whether those objects exist beforehand.

We formulate a language that allows one to specify nonparametric models in an intuitive manner. We focus on an important class of nonparametric methods, the Dirichlet process, because it handles distributions over unbounded sets of objects, as long as the objects themselves are exchangeable. In Sec. 2, we formalize the notion of object exchangeability, and argue that the presupposition of exchangeable unknown objects encompasses a wide range of interesting problems. Additionally, the nonparametric nature of DPs makes them suitable for solving model selection problems that arise when there is uncertainty about the number of objects. Models based on DPs have been shown to be capable of solving a variety of difficult tasks, such as topic document retrieval [2, 20]. When provided expert knowledge, our approach can tackle all the applications cited here, and others.

Sec. 4 formalizes our proposed language extension as a set of rules that map logical formulae to a nonparametric generative process. Sec. 5 demonstrates these concepts on two domains. We conduct experiments in Sec. 6, evaluating an NP-BLOG model for citation matching, demonstrating accurate and efficient probabilistic inference on a real-world problem. We stress that NP-BLOG is an extension to the BLOG language, so it retains all the functionality specified in [12].

2 Exchangeability of unknown objects

Unknown objects are those objects which are not guaranteed by the BLOG model. Unknown objects are generated by the model, and thus may not exist in all possible worlds. In this section, we formalize some important properties of generated objects. In particular, we adopt the notion of exchangeability [5] to objects in first-order probabilistic logic.

Let $\mathcal{X}$ be a set of objects. Given $x_i \in \mathcal{X}$, $\phi^f(x_i)$ denotes a random variable associated with the function symbol f, where the range of $\phi^f(x_i)$ is well-defined.[1] We use the bold notation $\boldsymbol{\phi}(x_i)$ to denote the mappings for all function symbols associated with object x_i. Reasoning about functions for multiple unknown objects is just a matter of defining $\mathcal{X}$ to be the space of unknown object tuples.

Definition 1. *Let $\beta_1, \beta_2, \ldots, \beta_K$ be a partition of $\mathcal{X}$ with finite K, and let $g(\cdot)$ be a permutation of the integers from 1 to K. The set of objects $\mathcal{X}$ is exchangeable if and only if*

$$P\left(\boldsymbol{\phi}(\beta_1), ..., \boldsymbol{\phi}(\beta_K)\right) = P\left(\boldsymbol{\phi}(\beta_{g(1)}), ..., \boldsymbol{\phi}(\beta_{g(K)})\right),$$

where $\boldsymbol{\phi}(\beta_{g(k)})$ is shorthand for $\boldsymbol{\phi}(x_i \in \beta_{g(k)})$.

In the case when $\mathcal{X}$ is finite, the concept of exchangeability is intuitive: the ordering of the object symbols is irrelevant, since the possible worlds remain equally likely. Exchangeability is useful for reasoning about conditional probability

[1] Throughout the rest of the paper, we use the equivalent notation $\phi_{x_i}^f$, but it is particularly cumbersome here.

densities (CPDs) involving unknown sets of objects. From Definition 1, we have the following result.

Proposition 1. *Let $P(\phi^f(x_1), \ldots, \phi^f(x_K), K|e_1, \ldots, e_N)$ be a CPDs over the number of objects K and the random variables $\phi^f(x_1)$ associated with the function symbol f (or multiple symbols) conditioned on evidence $e_1, \ldots, e_N$. It is possible to define P such that the sequence of objects $(x_1, \ldots, x_K)$, $x_k \in \mathcal{X}$, is exchangeable if and only if the evidence does not contain any statements referring to a particular x_k (the evidence may, on the other hand, refer to all of the objects using universal or existential quantifiers).*

For example, the distribution of the hair colours of two people, Eric and Mike, is not exchangeable given evidence that Eric is the father of Mike. What about sets of objects that form sequences, such as time? As long as we do not set the predecessor function beforehand, then any sequence is legally exchangeable. One could use the above proposition to define unknown objects. Note that objects may be unknown even if there is a fixed number of them.

In this paper, models are restricted to exchangeable unknown objects. Therefore, the order of unknown objects is not important, and we can reason about set of objects rather than sequences. While there are many domains in which one would like to infer the presence of objects that are not exchangeable, this constraint leaves us open to modeling a wide range of interesting and challenging domains.

By the above definitions, exchangeable objects are distinguished by their attributes. Unknown objects are assigned *non-rigid designators*; a symbol in different possible worlds does not necessarily refer to the same object, and so it does not make sense to assign it a rigid label. This consideration imposes a constraint: we cannot ask what is the position of aircraft a at time t, but rather, what is the position at time t of the aircraft that generated radar blip b (see Sec. 5.2). We can handle involving non-rigid designators with macros [12]. While we cannot form a query that addresses a specific unknown object, or subset of objects, we can pose questions about aircraft using existential and universal quantifiers (resolved using Skolemization, for instance). We could ask, for example, how many aircraft have traveled faster than the speed of sound.

3 Dirichlet processes

A Dirichlet process $G \,|\, \alpha, H \sim DP(\alpha, H)$, with parameter α and base measure H, is a unique probability measure defined on the space of all probability measures $G \in \mathcal{M}(\Phi)$ on $(\Phi, \mathcal{B})$ satisfying

$$(G(\beta_1), \ldots, G(\beta_K)) \sim DP(\alpha H(\beta_1), \ldots, \alpha H(\beta_K)) \quad (1)$$

for every measurable partition $\beta_1, \ldots, \beta_K$ of Φ, so it is a well-defined random probability distribution. The base measure defines the expectation of each partition and α is a precision parameter. One can consider the DP as a generalization of the Dirichlet distribution to infinite spaces.

In the previous section, we established properties of exchangeability for unknown objects. In order to explain the connection between exchangeability and the Dirichlet process, it is instructive to construct DPs with the Pólya urn scheme [5] (which should not be confused with the ball-and-urn example in the introduction). Consider an urn with balls of K possible colours, in which the probability of the first ball ϕ_1 being colour k is given by the normalized base measure H_k. We draw a ball from the urn, observe its colour, then return it to the urn. We then make another draw, observing its colour with probability $p(\phi_2 = k|\phi_1) = (\alpha H_k + \delta(\alpha_1 = k))/(\alpha + 1)$. This process continues, and after N observations ϕ_i the colour k of the next ball is distributed as

$$P(\phi_{N+1} = k|\phi_{1:N}) = \frac{\alpha H_k}{\alpha + N} + \frac{\sum_{i=1}^{N} \delta(\phi_i = k)}{\alpha + N}.$$

The marginal of this process $P(\phi_{1:N})$, which is obtained by applying the chain rule to the successive predictive distributions, can be shown to satisfy the following infinite mixture representation:

$$P(\phi_{1:N}) = \int_{\mathcal{M}(\Phi)} \left(\prod_{k=1}^{K} \pi_k^{\sum_{i=1}^{N} \delta(\phi_i = k)} \right) DP_{\alpha,H}(d\pi)$$

where the π_k are multinomial success rates of each colour k [5]. This result, which is a manifestation of de Finetti's theorem, establishes the existence and uniqueness of the DP prior for the Pólya urn scheme [5]. In Pólya urn scenario, observations ϕ_i are exchangeable and independently distributed given the measure G.

Analogously, if the urn allows for infinitely many colours, then for any measurable interval β of Ψ we have

$$p(\phi_{N+1} \in \beta|\phi_{1:N}) = \frac{\alpha H(\beta)}{\alpha + N} + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta(\phi_i \in \beta).$$

The first term in this expansion corresponds to prior knowledge and the second term corresponds to the empirical distribution. Larger values of α indicate more confidence on the prior measure H. Note that, as N increases, most of the colours will be repeated. Asymptotically, one ends up sampling colours from a large but finite set of colours, achieving a clustering effect. Nonetheless, there is always some probability of generating a new cluster.

DPs are essential building blocks in our formulation of nonparametric first-order logic. In the literature, these blocks are used to construct more flexible models, such as Dirichlet process mixtures (DPMs) and hierarchical or nested DPs [2, 20]. Since observations from a DP are provably discrete, DPMs add an additional layer $x_i \,|\, \phi_i \sim P(x_i|\phi_i)$ in order to model continuous draws x_i drawn from discrete mixture components ϕ_i.

In the Pólya urn scheme, G is integrated out and the ϕ_i's are sampled directly from H. Many ϕ_i's are repeated due to the clustering effect, so each x_i is sampled from a mixture of distributions with each component corresponding a group of ϕ_i's. Most algorithms for sampling DPs are based on this scheme [2, 14, 20]. In the DP hierarchies constructed by our language, however, we need an explicit representation of the measure G. This compels us to use the stick-breaking construction [19], which establishes that i.i.d. sequences $w_k \,|\, \alpha \sim \mathrm{Beta}(1, \alpha)$ and $\phi_k \,|\, H \sim H$ can be used to construct the equivalent empirical distribution $G(\,\cdot\,) = \sum_{k=1}^{\infty} \pi_k \delta(\phi_k)$, where the stick-breaking weights $\pi_k = w_k \prod_{j=1}^{k-1}(1 - w_j)$ satisfy $\sum_{k=1}^{\infty} \pi_k = 1$ with probability one. This shows that G is an infinite sum of discrete values. In this setting, the distribution over the weights π_k is equivalent to the Dirichlet with symmetric weights α/K. The DPM due to the stick-breaking construction is given by

$$\begin{aligned} \phi_i \,|\, H &\sim H & \pi \,|\, \alpha &\sim \mathrm{Dirichlet}(\alpha/K, \ldots, \alpha/K) \\ c_i \,|\, \pi &\sim \pi & x_i \,|\, \phi_i, c_i &\sim p(x_i|\phi_{c_i}), \end{aligned} \tag{2}$$

where $c_i = k$ indicates that sample x_i belongs to cluster k. The ball-and-urn (Fig. 1) is in fact an example of a DPM, where Φ is the unknown set of colours. By grounding on the support of the observations, the true number of colours K is finite. Therefore, the DP easily infers the true number of objects. At the same time, it permits us to be open about seeing new colours as new balls are drawn. In the NP-BLOG setting, the unknown objects are the clusters.

The NP-BLOG semantics in Sec. 4 define arbitrary hierarchical mixtures of Dirichlet processes. By the stick-breaking construction (2), every random variable x_i has a countable set of K ancestors (the unknown objects), hence DPMs preserve the well-definedness of BLOG models.

To infer the hidden variables of our models, we employ the efficient blocked Gibbs sampling algorithms developed in [7]. Since our models can consist of hierarchies of DPs, the algorithm presented in [7] often consists of a single step in the overall Gibbs sampler. One complication to inference stems from the fact that a product of Dirichlet distributions is difficult to simulate. Teh [20] provides a solution using an auxiliary variable sampling scheme.

4 Syntax and semantics

The objective of this section is to formalize the NP-BLOG language. We need to specify a procedure that takes a set of statements $\mathcal{L}_\Psi$ in the language and returns a model Ψ. Sections 4.1 and 4.2 are largely devoted to defining notation so that we can properly elaborate on NP-BLOG semantics (Sec. 4.3). We emphasize that our language retains all the functionality of BLOG. All unknown objects must be exchangeable, but this also an implicit assumption in BLOG.

4.1 Conditional probability densities

In BLOG, a model type is specified by an extension. For example $[\mathsf{Aircraft}] = \{a_1, \ldots, a_n\}$, where the logical variable a_i connotes an aircraft object, n is the number of aircraft objects, and $[\tau]$ denotes the extension of type τ. Types have extensions that can vary over possible worlds ω. Guaranteed objects exist in all possible worlds, and unknown objects are not guaranteed. The domain and range of a function is specified by its type signature $f : [\tau_1] \times \cdots \times [\tau_N] \mapsto [\tau_0] \cup \{\mathsf{null}\}$. For example, the function symbol State declares inputs to be of type Aircraft and Time and returns an object of type R6Rector, so $\phi^{\mathrm{State}}(a_i, t_j) = x_k$ (Sec. 5.2).

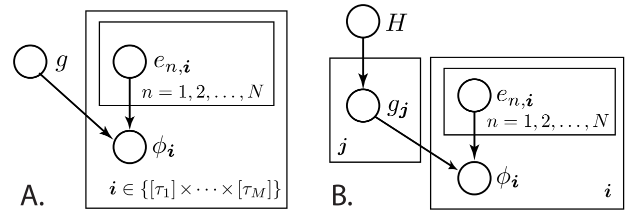

Figure 3: *(a).* Directed graph of a CPD g with evidence e. *(b).* CPDs $g_{\boldsymbol{j}}$ with evidence e and prior H.

Consider a measure g that induces a mapping from the domain $[\tau_0] \times [\tau_1] \times \cdots \times [\tau_N]$ to the probability simplex on $[\tau_0]$. This defines a conditional probability density (CPD) $g(\phi|e_1, e_2, \ldots, e_N)$, where each e_n is a piece of evidence. For instance, e_1 determines whether a fire occurred in a particular house, e_2 determines whether someone (presumably a burglar) broke into the house, ϕ indicates whether the electronic alarm notifies security, then $g(\phi|e_1, e_2)$ is the probability that the alarm goes off given the presence or absence of a fire and burglar. (Note that exchangeability does not apply until we consider distributions over unknown objects, in Sec. 4.3.) Thus, the random variable ϕ is a mapping from possible worlds to the alarm event. There might be several different houses, so we need to index the random variable ϕ according to the choice of a particular house. We need to index the evidence as well — a fire is evidence local to a house. Suppose $[\tau] = \{x_1, x_2, \ldots, x_{n(\tau)}\}$, where $n(\tau)$ is the number of objects of type τ, then $i \in \{1, 2, \ldots, n(\tau)\}$ indexes the objects in the extension of type τ. We denote $\boldsymbol{i} \in \{[\tau_1] \times \cdots \times [\tau_M]\}$ to be the selection of several objects from the power set, where M is the number of object types indexing the random variable ϕ. With this notation in place, we have an assignment $\phi_{\boldsymbol{i}}$ for every choice $\boldsymbol{i}$ drawn from the CPD $g(\phi_{\boldsymbol{i}}|e_{1,\boldsymbol{i}}, \ldots, e_{N,\boldsymbol{i}})$. This induces the directed graphical model in Fig. 3A. It is important to keep in mind that the indices range over the set of natural numbers, so there is an infinite series of plates in Fig. 3A. We don't denote this explicitly, but only a subset of the evidence variables need be indexed by the set of types in question.

4.2 Nonparametrics

Implicitly, the probability measure g is given by a fixed set of parameters, but we might want the parameters to depend on the choice of object. For example, the alarms of each house may have different makes, and they may be more effective at detecting fires or burglars. This seems to be no different than the framework developed in Sec. 4.1, since we can already correctly model this situation with a parameter indexed by the house object. However, when we are agnostic with regards to the number of houses, this technique is problematic since the indices can range over an infinite set, hence requiring an infinite set of parameters. Nonparametric models provide a solution to this problem.

As before, we introduce $\boldsymbol{j}$ that indexes the parameters of CPD g, ranging over the power set $\{[\tau_{M+1}] \times \cdots \times [\tau_{M+J}]\}$. We denote the varying parameterizations by $g_{\boldsymbol{j}}(\phi_{\boldsymbol{i}}|e_{1,\boldsymbol{i}},\ldots,e_{N,\boldsymbol{i}})$, and J is the number of types that index the parameters of $g_{\boldsymbol{j}}$. If $g_{\boldsymbol{j}}$ is defined as in 4.1, then the $g_{\boldsymbol{j}}$'s are defined on the parameter space $\mathcal{M}([\tau_0] \times [\tau_1] \times \cdots \times [\tau_N])$. We also introduce a nonparametric prior H over the choice of densities, in one-to-one correspondence with the objects. The graphical model with a nonparametric prior is depicted in Fig. 3B.

4.3 Dependency statements

The dependency statement is the key ingredient in the specification of a generative process Ψ over possible worlds $\omega \in \Omega_\Psi$. In particular, it defines the generation of a random variable $\phi_{\boldsymbol{i}}$ associated with a function symbol f and a tuple of logical variables $\mathsf{x}_1,\ldots,\mathsf{x}_{\mathsf{J+M}}$ associated with types $\tau_1,\ldots,\tau_{J+M}$. For example, the dependency statement $\mathsf{TrueColour(b)} \sim \mathsf{ColourDist()}$ means that each $\phi_{b_i}^{\mathrm{TrueColour}}$ is drawn from a distribution specified by ColourDist(). In order to treat nonparametrics, we introduce function symbols that refer to probability densities, and curly braces to index them. The logical variables $\{\mathsf{x}_1,\ldots,\mathsf{x}_\mathsf{J}\}$ determine the choice of measure g (or the choice of its parameters).

In general, a dependency statement looks like

$$\begin{aligned} &\mathsf{f}\{\mathsf{x}_1,\ldots,\mathsf{x}_\mathsf{J}\}(\mathsf{x}_{\mathsf{J+1}},\ldots,\mathsf{x}_{\mathsf{J+M}}) \\ &\quad \sim \mathsf{g}\{\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{L}\}(\mathsf{t}_{\mathsf{L+1}},\ldots,\mathsf{t}_{\mathsf{L+N}}); \end{aligned} \tag{3}$$

where f and g are function symbols, and $\mathsf{t}_1,\ldots,\mathsf{t}_{\mathsf{L+N}}$ are terms or formulae of the language in which the logical symbols $\mathsf{x}_1,\ldots,\mathsf{x}_{\mathsf{J+M}}$ can appear. For this to be a valid statement, both f and g must be defined on the same range $\mathcal{X}$. This covers the possibility that function symbol f refers to a probability measure over $\mathcal{X}$, in which case the probability measure g corresponding to symbol g must be defined on $\mathcal{M}_f(\mathcal{X})$, where $\mathcal{M}_f$ is the parameter space of f. (Note if f is associated with a function rather than a probability density, it follows that J is 0.) The first L terms inside the curly braces determine the choice of measure g, and the terms inside the parentheses are inputs to g. According to the semantics rules in Sec. 4.2, the dependency statement defines a nonparametric CPD, where logical variables index instances of $\phi_{\boldsymbol{i}}^f$.

BLOG also includes contingencies in dependency statements, but these can be subsumed within our formal framework by defining a new measure $g'_{\boldsymbol{j}} = \sum_t \delta(c_{\boldsymbol{i}} = \mathrm{cond}_t) g_{t,\boldsymbol{j}}(\phi_{\boldsymbol{i}}|e_{1,\boldsymbol{i}},\ldots,e_{N,\boldsymbol{i}})$, where $\delta(\cdot)$ is the indicator function, $c_{\boldsymbol{i}} = \mathrm{cond}_t$ specifying the condition which must be satisfied in order to sample from the density $g_{t,\boldsymbol{j}}$, and the summation is over the number of conditions. Infinite contingencies and their connection to graphical models are discussed in [13].

Consider a set of D dependency statements, such that their respective type signatures all declare a single argument with common type τ_1, and $[\tau_1]$ varies over possible worlds ω. Suppose then the logical variable x_1 is associated with this common type. We reason about collections of properties defined by dependency statements for the unknown objects $x_i \in [\tau_1]$ because they are jointly drawn from a Dirichlet process. Leaving out the curly braces from (3) for ease of presentation, we have

$$\begin{aligned} \mathsf{f}_1(\mathsf{x}_1,\mathsf{x}_{1,2},\ldots,\mathsf{x}_{1,\mathsf{M}_1}) &\sim \mathsf{g}_1(\mathsf{t}_{1,1},\ldots,\mathsf{t}_{1,\mathsf{N}_1}); \\ \vdots \qquad & \qquad \vdots \\ \mathsf{f}_\mathsf{D}(\mathsf{x}_1,\mathsf{x}_{\mathsf{D},2},\ldots,\mathsf{x}_{\mathsf{D},\mathsf{M}_\mathsf{D}}) &\sim \mathsf{g}_\mathsf{D}(\mathsf{t}_{\mathsf{D},1},\ldots,\mathsf{t}_{\mathsf{D},\mathsf{N}_\mathsf{D}}); \end{aligned} \tag{4}$$

where M_d and N_d are the number of arguments to function symbols f_d and g_d, respectively. Some of the input arguments are unknown while others may be guaranteed. As before, each $\mathsf{f_d}$ is associated with the random variables $\phi_{\boldsymbol{i}}^{f_d}$, where $\boldsymbol{i}$ ranges over the power set $\{[\tau_1] \times [\tau_{d,1}] \times \cdots \times [\tau_{d,N_d}]\}$. The distribution over $\phi_{\boldsymbol{i}}^{f_d}$ depends on g_d and the evidence terms, as illustrated in Fig. 3B. The set of statements (4) defines the generative process

$$\pi^{\tau_1} \sim \mathrm{Dirichlet}\left(\alpha^{\tau_1}/n(\tau_1),\ldots,\alpha^{\tau_1}/n(\tau_1)\right) \tag{5}$$

$$\phi_{\boldsymbol{i}}^{f_d} \sim H^{g_d}(\,\cdot\,|e_{1,\boldsymbol{i}},\ldots,e_{N_d,\boldsymbol{i}}), \tag{6}$$

where $n(\tau_1)$ is the number of unknown objects of type τ_1, H^{g_d} is the probability measure paired with symbol g_d, α^{τ_1} is the user-defined Dirichlet process concentration parameter, and π^{τ_1} is a multinomial distribution such that each success rate parameter $\pi_k^{\tau_1}$ determines the probability of choosing a particular object $x_1 \in [\tau_1]$. For multiple unknown objects $\mathsf{x}_1,\mathsf{x}_2,\ldots$, it is a matter of defining a tuple of types $\boldsymbol{\tau}$ and multinomial distribution $\pi^{\boldsymbol{\tau}}$ over the respective tuples of unknown objects.

There is one important exception to rule (6). If we have declared a function symbol f with a return type τ ranging over a set of unknown objects, then there exists the default generating process

$$\phi_{\boldsymbol{i}}^f \sim \pi^\tau. \tag{7}$$

Rule (7) automatically specifies a distribution over unknown objects. For example, the NP-BLOG model in Sec. 5.1 constructs a distribution over publications π^{Pub}, and hence ReferringPub(c) for every citation object is automatically drawn from π^{Pub}. We note that the semantics of unknown objects defined here do not preclude the need for number statements defines in [12], since there are certainty situations in which one would like to reason about the number of objects without taking to account their properties. In the aircraft tracking model (Sec. 5.2), for example, the number of blips at each time step is specified by a number statement.

NP-BLOG allows for the definition of a symbol g that corresponds to a multinomial distribution over the set of objects $[\tau]$. This exhibits the default prior

$$\phi_{\boldsymbol{i}}^{g} \sim \mathrm{Dirichlet}(\alpha^{g}\pi^{\tau}), \tag{8}$$

analogous to (7). This is useful for modeling collections of objects such as the authors of a publication or the grades in each department. In both cases (7) and (8), one can override the defaults by including appropriate dependency statements for f and g, in which case we get $\phi_{\boldsymbol{i}}^{f} \sim g$ following the rule (6).

The generative process (5-6) is a stick-breaking construction over the objects $i \in [\tau_1]$ and their properties $\phi_i^{f_d}$. When the number of unknown objects $n(\tau_1)$ tends to the limit, (5-6) is equivalent to the Dirichlet process

$$\begin{aligned} G^{g_1} \times \cdots \times G^{g_D} &\sim DP(\alpha^{\tau_1}, H^{g_1} \times \cdots \times H^{g_D}) \\ \theta_{\boldsymbol{i}}^{f_d} &\sim G^{g_d}, \;\; d = 1, 2, \ldots, D, \end{aligned} \tag{9}$$

where each $\theta_{\boldsymbol{i}}^{f_d}$ is an instance of the symbol f_d. We adopt the slightly different notation here due to the following: if $\theta_{\boldsymbol{i}}^{f_d}$ is a distribution over unknown objects τ_2, then $\theta_{\boldsymbol{i}}^{f_d}$ is the product measure $G^{g'_1} \times G^{g'_2} \times \cdots$, where $g'_1, g'_2, \ldots$ are function symbols taking objects of type τ_2 as input.

Since BLOG is a typed, free language, we need to allow for the null assignment to $\phi_{\boldsymbol{i}}^{f}$. We permit the clause

$$\mathsf{f}(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_{\mathsf{J+M}}) \text{ if cond then null;} \tag{10}$$

which defines the distribution $\delta(c_{\boldsymbol{i}} = \mathrm{cond})\delta(\mathsf{null}) + \delta(c_{\boldsymbol{i}} \neq \mathrm{cond})\pi^{\boldsymbol{\tau}}$. This statement is necessary to take care of the situation when an object's source can be of different types, as in the aircraft tracking domain with false alarms in Sec. 5.2.

The set of rules (5-8,10), combined with the number statements [12], maps a language $\mathcal{L}_\Psi$ to a model Ψ, which is a set of symbols and a distribution over possible worlds $\omega \in \Omega_\Psi$. The rules of semantics assemble models that are arbitrary hierarchies of DPs.

5 Applications

We now demonstrate the application of NP-BLOG to two domains, citation matching and aircraft tracking. These domains illustrate how NP-BLOG captures complex models of unknown objects in an elegant manner, and provides functionality beyond that of BLOG.

5.1 Citation matching

One of the main challenges in developing an automatic citation matching algorithm is the resolution identity uncertainty: two dissimilar citations might refer to the same publication. Pasula *et al.* incorporate unknown objects and identity uncertainty into a probabilistic relational model [15]. There has also been some recent work on modeling identity uncertainty through dependence relations in conditional random fields [21]. We take a generative approach as in [11], but the key difference with our model is that collections of unknown objects are drawn from DPs. The model infers the number of true publications and authors given observations in the form of citations extracted from research papers.

```
01 type Author; type Pub;
02 type Citation; type AuthorAsCited;
03 String Name(Author);
04 String Title(Pub);
05 Author PubAuthorsDist(Pub);
06 Pub ReferringPub(Citation);
07 String CitedTitle(Citation);
08 Citation CitedIn(AuthorAsCited);
09 Author ReferringAuthor(AuthorAsCited);
10 String CitedName(AuthorAsCited);
11 Name(a) ~ NameDist();
12 Title(p) ~ TitleDist();
13 CitedTitle(c) ~ TitleStrDist(Title(ReferringPub(c)));
14 ReferringAuthor(u)
       ~ PubAuthorsDist{ReferringPub(CitedIn(u))};
15 CitedName(u) ~ NameStrDist(Name(ReferringAuthor(u)));
```

Figure 4: NP-BLOG model for citation matching.

The model for the citation matching domain is shown in Fig. 4. Lines 1-2 declare the object types. Lines 3-10 specify the type signatures of the function symbols. Lines 11-15 are dependency statements, which we explain in more detail here. The terms to the right of the "$\sim$" refer to CPDs which generate the random variables corresponding to function symbols.

All publication objects (and the mappings of ReferringPub(c)) are drawn independently from a distribution hidden from the user, according to rule (7). Each publication object p has two attributes: a Title(p) drawn from TitleDist(), and a collection of authors, PubAuthorsDist{p}, implicitly drawn from the distribution of publications following (8). Given that we know ReferringPub(c), we generate the observed title. We also generate the author objects according to the author population of the referring publication, PubAuthorsDist{p}, which defines a probability measure over authors indexed by each publication object p. The important point is that NP-BLOG infers a distribution over collections of unknown publications, authors, and groups of authors in publications, all handled through DPs. We reinforce this point in Sec. 6, were we simulate this model on the CiteSeer database.

By tracing the rules of semantics, one should see that only

```
01 type UFO; type Aircraft; type Blip; type Time;
02 R6Vector StateTransDist(UFO);
03 UFO KindOfAircraft(Aircraft);
04 R6Vector State(Aircraft,Time);
05 R3Vector ApparentPos(Blip);
06 Boolean IsFalseAlarm(Blip);
07 Aircraft BlipSource(Blip);
08 Time BlipTime(Blip);
09 StateTransDist{u} ~ StateTransitionPrior();
10 State(a,t) if t = 0 then ~ InitState() else
   ~ StateTransDist{KindOfAircraft(a)}(State(a,Pred(t)));
11 #Blip: ((BlipTime) → (t)) ~ NumBlips();
12 FalseAlarm(b) ~ IsFalseAlarmDist();
13 if FalseAlarm(b) = true then BlipSource(b) = null;
14 ApparentPos(b) if IsFalseAlarm(b) = true then
   ~ FalseAlarmDist() else
   ~ ObsDist(State(BlipSource(b),BlipTime(b)));
```

Figure 5: NP-BLOG model for the aircraft domain.

thing the model does not generate is values for CitedIn(u), and hence they must be provided by the data. The training set can still provide observations from any number of object attributes, for example CitedTitle(c) and CitedName(u).

Both BLOG and NP-BLOG can answer the following queries: Is the referring publication of citation c the same as the referring publication of citation d? How many authors are there in the given citation database? How many citations refer to the publication referenced by citation c? What are the names of the authors of the publication referenced by citation c? How many publications contain the author a, where a is one of the authors in the publication referenced by citation c? And what are the titles of those publications? However, only NP-BLOG can answer the following query: what group of researchers do we expect to be authors in a future, unseen publication?

5.2 Aircraft tracking

Milch *et al.* present the aircraft tracking in [12], in which aircraft in flight appear as blips on a radar screen, and the objective is to infer the number of aircraft and their flight paths. Radars are susceptible to noise so some blips might not represent any aircraft. Conversely, a single airplane can produce multiple detections. We assume that the aircraft remain within the range of the radar. We add a new dimension by modeling different kinds of aircraft, for example blimps, jets and gliders. Names of the kinds of aircraft are members of the type UFO. We differentiate UFOs according to their flight patterns — blimps hardly move at all, jets can reach high speeds, and gliders circle around a lot. NP-BLOG is able to handle this situation quite well: we would like the unknown UFO objects to possess distributions over flight patterns, which we cannot do in BLOG.

The NP-BLOG model for the aircraft domain, with false alarms and UFOs, is given in Fig. 5. We assume that default types, such as real vectors, have been previously declared. On line 11, the model generates a certain number of blips at every time step. Line 12 determines whether or not a blip is a false alarm. If it is not a false alarm, we need to generate BlipSource(b) from the hidden aircraft distribution. Both aircraft and UFOs are drawn from their respective distributions, modeled using DPs. Lines 2 and 9 specify that StateTransDist{u} is a distribution over state transitions indexed by unknown UFO objects u, which in turn are generate according to a nonparametric prior StateTransitionPrior(). The states of an aircraft at a given time t follow the CPD of StateTransDist{u}, given evidence KindOfAircraft(a) and State(a,Pred(t)).

The aircraft model is written in a high-level fashion, yet captures sophisticated properties such as the flight patterns of unbounded kinds of aircraft. The NP-BLOG model can answer queries that BLOG cannot: for instance, at what speed do we expect an unseen aircraft to be traveling, given that it is the same kind as the aircraft that generated blip b?

	Face	Reinforce.	Reason.	Constraint
Num. citations	349	406	514	295
Num. papers	246	149	301	204
Phrase matching	0.94	0.79	0.86	0.89
RPM+MCMC	0.97	0.94	0.96	0.93
CRF-Seg ($N = 9$)	0.97	0.94	0.94	0.95
NP-BLOG	0.93	0.84	0.89	0.86

Table 1: Citation matching results for the NP-BLOG, Phrase Matching [10], PRM [15], and CRF-Seg [21]. Performance is measured by counting the number of publication clusters that are recovered perfectly. The NP-BLOG column reports is an average over 1000 samples of possible worlds.

6 Experiment

The purpose of this experiment is to show that the language NP-BLOG we have described realizes probabilistic inference on an real-world example. We simulate the citation matching model presented in Sec. 5.1 on the CiteSeer data set [10], which consists of collections of citations from four research areas in AI. We use Markov Chain Monte Carlo (MCMC) to simulate possible worlds from the model posterior, given evidence in the form of cited authors and titles. A short description of the inference engine is provided in Sec. 3. Table 1 compares the performance of the NP-BLOG model to [15, 21] and the greedy agglomerative clustering method [10] implemented by [15]. We achieve respectable matching accuracy, even though the specification of the model required only a few lines in NP-BLOG. Whereas [21] use as many as 9 different citation fields and [15] train priors from US Census data and BibTeX bibliographies, we use the Jaro metric for author surnames and the standard TF-IDF information retrieval metric for dis-

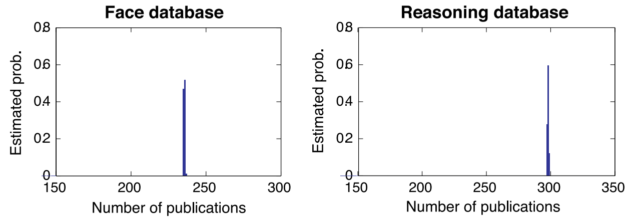

Figure 6: Estimated distribution of the number of publications for the Face and Reasoning data sets, generated from 1000 MCMC samples. The true number of publications is Face = 246 and Reasoning = 295.

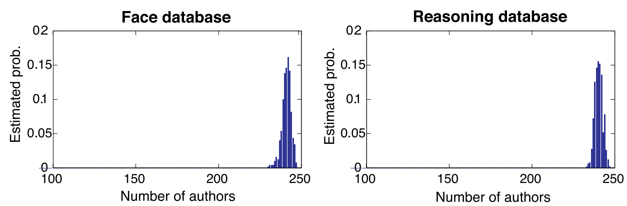

Figure 7: Estimated distribution of the hidden number of authors for the Face and Reasoning data sets.

tances between titles. Due to errors in the labeled citation strings, we expect there to be slight inconsistencies in the evaluation. Our NP-BLOG Gibbs sampling scheme is efficient since 1000 iterations are more than sufficient for training sets with as many as 500 observations.

In Figures 6 and 7, we plot the Monte Carlo estimate of the numbers of publication and author clusters for two data sets. The posteriors over the number of publications are highly peaked, and they closely match the ground truth.

7 Conclusions

This paper presented a novel set of semantics for modeling collections of objects and their properties in arbitrary hierarchies by extending the BLOG first-order probabilistic language. We demonstrated that NP-BLOG handles complex domains while obscuring the complicated implementation details from the user. We adopted Bayesian nonparametric methods, and notably Dirichlet processes, for defining distributions over collections of unknown, exchangeable objects and their properties. We showed that NP-BLOG can handle queries that cannot be answered by parametric representations. Significantly, Dirichlet processes handle model selection of unbounded sets of objects in first-order probabilistic inference in a cohesive and efficient fashion.

There is much future work on this topic. For one, an important direction is the development of more efficient inference algorithms for hierarchies of Dirichlet processes (e.g. using sequential Monte Carlo methods). As well, we would like to get around type constraints by introducing uncertainty about object types.

Acknowledgements

This paper wouldn't have happened without the help of Brian Milch. Also, thanks to Gareth Peters, Mike Klaas and Mikhail Bilenko for their assistance.

References

[1] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. 1994.

[2] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 2004.

[3] P. Domingos and M. Richardson. Markov logic: A unifying framework for statistical relational learning. In *ICML Wkshp. on Stat. Rel. Learning*, 2004.

[4] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In Dzeroski and Lavrac, editors, *Relational Data Mining*. 2001.

[5] J. K. Ghosh and R. V. Ramamoorthi. *Bayesian Nonparametrics*. 2003.

[6] D. Heckerman, C. Meek, and D. Koller. Probabilistic models for relational data. Technical report, MSR, 2004.

[7] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *JASA*, 96:161–173, 2001.

[8] M. Jaeger. Relational bayesian networks: a survey. *Electronic Articles in Comp. and Inf. Sci.*, 6, 2002.

[9] K. B. Laskey. First-order bayesian logic. Technical report, GMU Dept. of Sys. Eng. and Op. Res., 2005.

[10] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32:67–71, 1999.

[11] B. Milch, B. Marthi, and S. Russell. BLOG: Relational modeling with unknown objects. In *ICML Wkshp. on Stat. Rel. Learning*, 2004.

[12] B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *IJCAI*, 2005.

[13] B. Milch, B. Marthi, D. Sontag, S. Russell, D. L. Ong, and A. Kolobov. Approximate inference for infinite contingent Bayesian networks. In *AI-Stats*, 2005.

[14] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

[15] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *NIPS*, 2003.

[16] H. Pasula and S. Russell. Approximate inference for first-order probabilistic languages. In *IJCAI*, 2001.

[17] A. Pfeffer. Ibal: An integrated bayesian agent language. In *IJCAI*, 2001.

[18] D. Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[19] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.

[20] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. Technical report, Dept. of Statistics, University of California at Berkeley, 2004.

[21] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *UAI*, 2004.