
Building Graph Structures from the Beta Process

Noel Welsh

School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
nhw@cs.bham.ac.uk

Jeremy Wyatt

School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
j1w@cs.bham.ac.uk

Abstract

We show how a variety of directed and undirected graphs can be modelled via the hierarchical Beta process [7], extending the domain of non-parametric methods to an important new class of problems. Experimental results for the task of inferring topological maps using data collected from robot sensors validate the practical utility of our model.

1 Introduction

Graph structures are ubiquitous in Computer Science. For example, in computer vision one might represent an object as an graph of two-dimensional views, a structure known as an aspect graph (e.g. [3]). In robotics, a robot might use a topological map, another graph structure, to direct exploration of an environment (e.g. [2, 5]). The question arises how we can infer these graphs from data. Answering this question within the framework of Bayesian non-parametrics is the main contribution of this paper. This approach offers several advantages: an explicit representation of uncertainty, a large number of well-known tools can be applied to the inference task, and the non-parametric approach does not constrain the graph to be a fixed size.

In Section 2 we define directed and undirected graphs. In Section 3 we describe the hierarchical Beta process [7]. We show how to use the hierarchical Beta process to create a distribution over directed graphs in Section 4, over undirected graphs in Section 5, and finally how to extend the model to include transition probabilities in Section 6. Experimental results for inferring topological maps are given in Section 7.

2 Graphs

An *undirected graph* G is a set of nodes N and a set of edges E connecting nodes. Edges can be represented as a set of two elements i, j , indicating that nodes i and j are connected. Alternatively we can represent the edges as binary variables, so that $E_{ij} = 1$ if nodes i and j are connected, and $E_{ij} = 0$ otherwise. Note that this relationship is symmetric; $E_{ij} = E_{ji}$.

Using the binary representation of edges we write E_i to indicate the binary vector of edges from node i . Similarly, we write E for the symmetric binary matrix of edges for the entire graph.

A *directed graph* is exactly the same as an undirected graph except edges now have a direction. This means that if i is connected to j , j is not necessarily connected to i . We can now represent edges as ordered pairs (i, j) of nodes, or continue with the binary representation but remove the constraint that $E_{ij} = E_{ji}$.

3 The hierarchical Beta process

All our graph models are built on the hierarchical Beta process (HBP; [7]). The hierarchical Beta process is itself built from the Beta process and the Bernoulli process. The Beta and Bernoulli processes form a conjugate pair, and together provide a distribution over infinite binary vectors. The alert reader will recall that we can represent edges E_i as binary vectors, and hence see that the Beta-Beroulli process is a possible building block for distributions over graph.

Space prevents us from including a full description of the HBP; please see [7]. In this section we give an informal description, showing what the Beta and Bernoulli processes do and motivating the HBP.

The Beta process is a distribution on an infinite vector of reals in $[0, 1]$. We can interpret this as the probability of an edge. It has two parameters. The first is called the *concentration parameter* and the second the *base measure*. The base measure is the more important of the two. It is a distribution defined over the indices of the vector and gives a weight to each index. The larger the weight the larger the real is the vector is likely to be. The Beta process is written $BP(c, B_0)$, with concentration parameter c and base measure B_0 .

The Bernoulli process is a distribution on an infinite vector of binary numbers 0, 1. It is parameterised by a sample from a Beta process. We write $BeP(B)$ for a Bernoulli process with parameter B .

What is a node? Remember the nodes of the graph generate observations. We need some probability distribution over observations, so we make a node a set of parameters for this distribution. A node is also an index. We have one Beta-Beroulli process per node and we need each process to share indices. This is why we introduce the HBP. We have a top-level Beta process in the hierarchy whose job is to choose indices (nodes) and create a vector of weights for these indices. Then each node-level Beta process is parameterised by the same sample from the top-level process, and so they all share the same set of indices (nodes).

4 The infinite directed graph

The infinite directed graph (IDG) is our first graph model, and it follows directly from the HBP. As the name suggests the model is a directed graph, so we need to define distributions over the set of nodes and edges. The model is as follows:

- We assume observations are drawn from a distribution F which is parameterised by a sample $\theta \sim \Theta$. That is, for an observation x and some parameters θ , $Pr(x) = F(\theta)$.
- $N \sim BP(c, B_0)$
Nodes are drawn from a Beta process. The Beta process is parameterised by some concentration parameter c , and a measure over Θ .
- $C_i \sim BP(c_j, N)$
Each node i has an infinite connectivity vector C_i , which gives the probability of an edge from i to all other nodes. C_i is drawn from a Beta process that is parameterised by the node set N and a node specific concentration parameter c_i (which is normally the same for all nodes).
- $E_i \sim BeP(C_i)$
For each node i the binary vector of edges E_i is drawn from a Bernoulli process parameterised by C_i .

This model is just the HBP with a two-level hierarchy. We have applied this model, with a minor extension, to the problem of inferring topological maps. Further details are given in Section 7.

5 The infinite undirected graph

In many situations an undirected graph is preferable. For example, maps are usually undirected – if you can get from A to B you can get from B to A – so an undirected graph halves the number of

parameters. We can extend the IDG to an undirected model which we call the infinite undirected graph (IUG).

The IUG is a constrained version of the IDG. If there is a connection from node i to j there must be a connection from j to i . This means that the edge vectors E_i are not independent. To deal with this constraint we order the nodes and say E_{ji} is given by E_{ij} for $i < j$. So the connectivity for a node E_j is dependent on all its predecessors.

6 The infinite sparse hidden Markov model

The hidden Markov model (HMM) has found many applications, and has been extended to the non-parametric setting as the infinite hidden Markov model (IHMM) [1, 6]. We can define an extension of the IDG that also has transition probabilities, making it a HMM variant. It differs from the IHMM in being sparsely connected. That is, states only have a connection to a few others; in the IHMM there is a non-zero probability of a transition to any state. Hence we call our model the infinite sparse HMM (ISHMM). We hypothesise that this property means the ISHMM has simpler inference and is applicable to a wider range of problems than the IHMM.

To go from the IDG to the ISHMM we need only define the transition probabilities T_{ij} , which give the probability of transitioning from state i to j given $E_{ij} = 1$. We require the transition probability sum to one, but we don't know which edges exist until we sample them. To get around this we sample unnormalised weights in $[0, 1]$, and then normalise when the edges are known.

More concretely we define transition weights W_i for node i as a sample from a Beta process: $W_i \sim BP(c_i, N)$. Then the probability of a transition T_{ij} is given by $T_{ij} = \frac{W_{ij}}{\sum_i E_{ij} W_{ij}}$. Since there are only a finite number of edges we can perform this calculation.

We can see two reasons for using the ISHMM. Firstly, we can use the exact forward-backward algorithm for inference as there are only a finite number of edges from each node. Thus we expect ISHMM samplers to mix faster than those for the IHMM. Furthermore we can constrain the ISHMM in interesting ways. For example, we could say each node must be a corridor (with two edges) or an office (with one edge) and use rejection sampling to draw samples that meet these criteria. This allows use to create structured models without having to derive a new distribution.

7 Applications

A topological map is a map that only retains essential information. The London Underground map, a fragment of which is shown in Figure 1, is a well known example. We can represent such maps as graphs, and hence apply the IDG to the task of inferring topological maps from data.

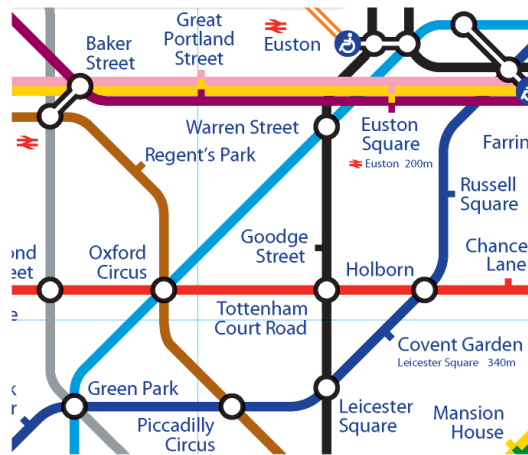


Figure 1: A fragment of the London Underground map, an example of a topological map.

To use the IDG we add a small extension that tells us how the agent moves around the map – necessary for sampling. We decide that all transitions are equally likely. Given this choice we can implement a simple collapsed Gibbs sampler.

In Figure 2 we show a high probability sample from the posterior of an experiment using the Austin data set prepared by Patrick Beeson for the Robotics Data Set Repository (Radish) [4]. The data consists of laser scans which we augment with artificial odometry readings. The Figure shows the location of the odometry readings (the crosses) and the inferred nodes and edges (circles and lines respectively). This inferred map is essentially identical to the true map.

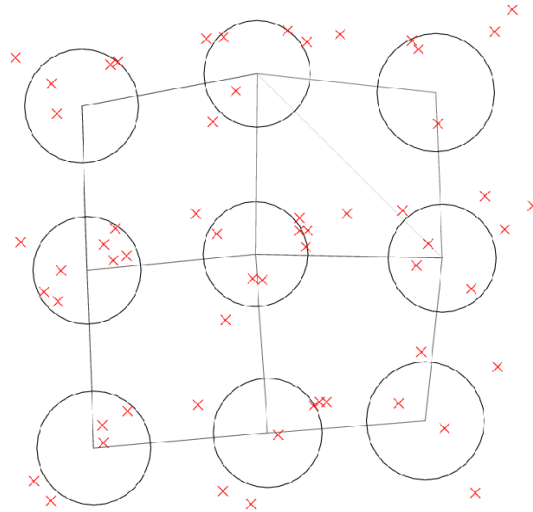


Figure 2: A map inferred from a mixture of odometry and laser range finder data. Crosses indicate where the location of the odometry readings. The inferred model is very close to the true map.

In Figure 3 we show an artificial data set called Two Squares. For this data set the observations are odometry and obstacle sensors, which we model as Gaussian/Gaussian and Multinomial/Dirichlet pairs respectively.

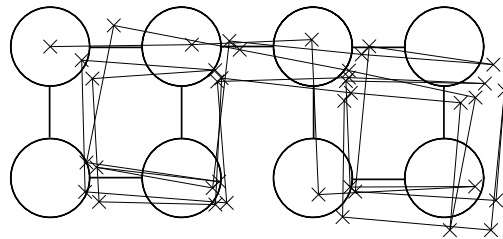


Figure 3: The Two Squares data set.

In Figure 4 we show how samples evolve with increasing amounts of data. The size of the circles (nodes) indicates the uncertainty in their true location. As the data increases the samples get closer to the true map. It is apparent that we also become more certain of node locations as data increases and similarly the posterior probability (not shown) becomes increasingly peaked. We should hope this would happen with a Bayesian model but demonstrating this effect is particularly important for models used by autonomous robots. Truly autonomous robots will need to quantify uncertainty in their environment model, and make use of this knowledge when planning actions. For example, a exploratory robot should be capable of identifying a gap in its model and then plan and execute actions to fill that gap. Because our model allows us to represent a potentially unbounded number of unknown locations and connections it is able to support reasoning and exploration methods that explore in order to test such hypotheses. Such work would be an interesting use of this model to support exploration for SLAM.

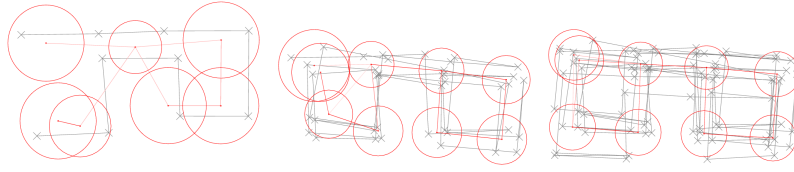


Figure 4: Evolution of the posterior for the Two Squares data set as the amount of data increases. The samples are getting closer and closer to the true map.

8 Conclusions and future work

We have shown how to construct a variety of graph structures using the hierarchical Beta process. Our results in the domain of topological mapping show the infinite directed graph is practical and useful. Future work involves implementing the IUG and ISHMM, and new sampling algorithms.

Acknowledgments

We gratefully acknowledge the support of the European Commission funded project CogX: Cognitive Systems that Self-Understand and Self-Extend, FP7 project IST-215181.

References

- [1] Matthew Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 14*, pages 577–585, Cambridge, MA, 2002. MIT Press.
- [2] Patrick Beeson, Joseph Modayil, and Benjamin Kuipers. Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy. *International Journal of Robotics Research*, 2010. In Press.
- [3] Christopher M. Cyr and Benjamin B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. *Int. J. Comput. Vision*, 57(1):5–22, 2004. ISSN 0920-5691. doi: <http://dx.doi.org/10.1023/B:VISI.0000013088.59081.4c>.
- [4] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2003. URL <http://radish.sourceforge.net/>.
- [5] Ananth Ranganathan and Frank Dellaert. Inference in the space of topological maps: An MCMC-based approach. In *IROS*. IEEE, 2004. URL <http://www.cc.gatech.edu/~dellaert/pub/Ranganathan04iros.pdf>.
- [6] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- [7] Romain Thibaux and Michael I. Jordan. Hierarchical beta processes and the indian buffet process. Technical Report 719, Department of Statistics, University of California, Berkeley, November 2006.