Hybrid methods for Bayesian inference

DanielWinterstein and Hannu Rajaniemi
ThinkTank Mathematics Ltd, ETTC, King's Buildings, Mayfield Road, Edinburgh, EH9 3JL

Abstract

We study a novel approach to inference on Bayesian graphical models based on combining deterministic and non-deterministic approximation methods. The theory we develop has many potential applications - including robot localisation, agent tracking, and reasoning about an enemy agent's goals

1. Introduction

Bayesian approaches are one of the dominant techniques for inference and learning. They seamlessly combine prior knowledge with data, provide error bar or confidence intervals for estimation and prediction and offer mechanisms for model and feature selection. One of the major barriers for Bayesian approaches is computational complexity. This project investigates a new technique - a hybrid message passing algorithm - for tackling this issue.

There are a number of existing algorithms for this task. There is no 'top' algorithm. The disparate existing algorithms, such as Monte-Carlo approaches or Kalman filters, are miles apart with quite different strengths and weaknesses. One size does not fit all. Unfortunately a particular scenario may have several aspects that make it unsuitable for any one algorithm. In practice, techniques are often combined - but in a crude way. Each algorithm is treated as a black box with some custom 'plumbing code' to connect them.

The hybrid approach which we describe here offers considerable benefits. It is more flexible than a 'one size fits all' approach. Yet it still provides a solid theoretical basis, and should achieve better accuracy than the 'black boxes plumbed together with custom piping' approach. It also has potential for developing better user interfaces, including producing coherent explanations for the actions of autonomous systems.

Our aim is to combine sophisticated mathematical theory with pragmatic concerns. The chief strength of the hybrid framework is its flexibility. Most research develops one technique. By contrast, the hybrid approach allows different techniques to be used as appropriate. The theory we develop has many potential applications - including robot localisation, agent tracking, and reasoning about an enemy agent's goals.

2. Bayesian Graphical Models (BGMs) and Factor Graphs

Bayesian Graphical Models, also known as *belief networks*, are a graphical notation for describing the structure of dependencies between random variables.

If the distributions include unknown parameters, *hyperpriors* (prior distributions for the parameters) can be introduced as additional variable nodes in the graph. This allows reasoning about the choice of distributions to be encoded in the structure of the graphical model itself.

A BGM implicitly encodes the factorisation structure of a probability distribution. The formalism of *factor graphs* [1] makes this factorisation explicit by introducing additional nodes for the factors themselves. That is, some nodes represent variables whilst others represent factors in the joint

probability distribution. Figure 1 shows a simple example.

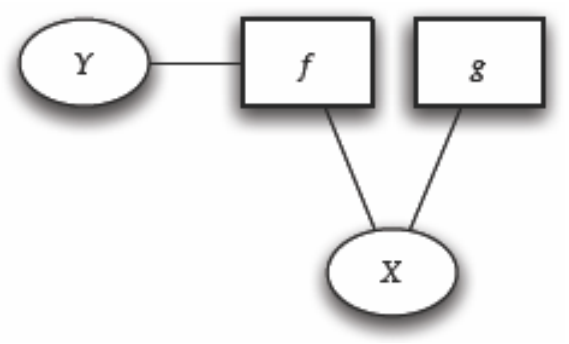

Figure 1: A factor graph encoding $\mathbb{P}(X, Y) = f(X, Y)\, g(X)$

Factor graphs are less intuitive than BGMs and so not ideal for defining and interacting with statistical models. However, they provide a natural setting for defining and analysing inference algorithms.

For a fuller treatment of BGMs and factor graphs, see [2, 3, 4].

3. Inference and Message Passing Algorithms

The real utility of Bayesian graphical models lies in efficient ways of performing inference, i.e. computing the posterior distributions of a variable node or node subsets given the observed values of a number of other nodes. The graphical structure of the model can be exploited to find efficient algorithms for inference. In particular, many algorithms can be expressed in terms of propagating local *messages* around the graph. This gives dramatic improvements in performance. For example, naive filtering in a Markov chain has complexity $O(s^n)$ - where s is the number of states and n the chain length - but only $O(ns^2)$ using message passing.

Message passing algorithms are closely related to energy minimisation, and several of the ideas in this field have come from statistical physics. The nodes in a graph can be thought of as particles, with the entropy of the probability distributions equating to heat energy.

Even using message passing inference can be intractable. The computational complexity depends on the algorithm used, the complexity of the individual factors, and the size and sparsity of the graph.

3.1 The Sum-Product Algorithm

The *sum-product* algorithm, also known as *belief propagation* (BP) is the starting point for message passing algorithms. The forward-backward algorithm used in Markov chains is a particular case of the sum-product algorithm. We briefly define the algorithm below. For a fuller treatment and proof of convergence, see [2,1].

The sum-product algorithm works as follows:

- Convert the BGM into a factor graph[1].
- Let X be a variable node connected to factor nodes f_i, $i = 1. \; . \; .m$. Define the message from X to f_i to be:

$$(X \to f_i) = \prod_{j \neq i} (f_j \to X) \qquad (1)$$

  - that is, the product of all messages to X from nodes other than f_n. We use $\prod \emptyset = 1$ for the empty product, which means that if X is a leaf node and f is its only connection, then $(X \to f) = 1$.

- Let f be a factor node connected to variable nodes X, Y. Define the message from f to X to be $(f \to X) = \int_Y f(x, y).(Y \to f)(y)dy$. The effect of this is that $(f \to X)$ is the distribution for X once the variables Y and Z have been removed through marginalisation.

  More generally, if f is connected to variable nodes X, Y_1, ... , Y_n, then:

$$(f \to X) = \int_{Y_1 \ldots Y_n} f(x, y_1, \ldots, y_n) \prod_i (Y_i \to f)(y_i) \qquad (2)$$

  If f is a leaf node and X is its only

[1] Belief propagation was originally formulated for BGMs by Pearl [3], but we prefer to work with factor graphs in this report.

connection, then $(f \rightarrow X) = f$ using $\prod_{\emptyset} = 1$. If the variables are discrete, then the integrals are replaced by sums over the possible values - hence the name sum-product.

- A message $(A \rightarrow B)$ from node A to its neighbour B is ready to transmit if A has received messages from all its neighbours other than B (this is true for both variable and factor nodes).
- The algorithm starts with the leaf nodes, which are automatically ready to transmit.
- If the graph is a tree, the algorithm will terminate when each node has transmitted a message to, and received a message from, each of its neighbours. At this point, the marginal distribution for a node will be given by the product of all incoming messages. That is, $\mathbb{P}(X = x) = \prod_i (f_i \rightarrow X)(x)$ where f_i are the neighbours of X. If the original graph was a BGM - as we are considering - then the distribution will be normalised. Otherwise, a normalisation step is required.

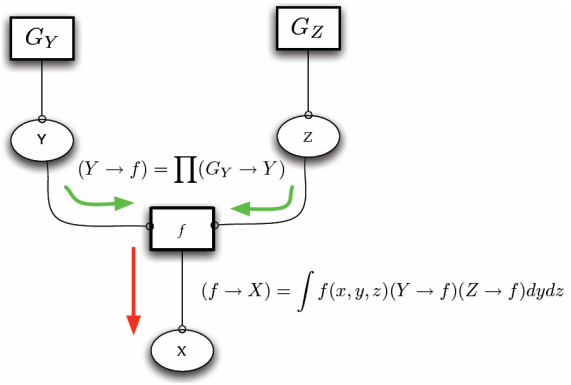

Figure 2: Messages in the sum-product algorithm

The sum-product algorithm is an exact inference algorithm. However, it can be computationally intractable, and convergence is only proven for acyclic graphs.

3.2 Related Algorithms

3.2.1 Loopy Belief Propagation

Loopy belief propagation [5] is simply the use of the sum-product algorithm on graphs with loops. In the rest of this report we will use 'BP' to refer to this more general use. The loops cause BP to become an iterative algorithm. Messages may be updated (i.e. re-sent) several times as information spreads through the graph.

Belief propagation is more efficient than the junction tree algorithm. However it is not precise - the answers found are approximations - and convergence is not guaranteed. Performance is poor if there are many tight loops with conflicting interactions and weak evidence [5].

3.2.2 The Kalman Filter as an Approximate Method

The Kalman filter assumes that transitions and observations are linear functions of the system's state, and that noise is Gaussian. In practice, both of these assumptions are often broken, yet the filter often remains useful. In these cases, the Kalman filter becomes an approximate method. Non-linear transitions are approximated by linear ones, and non-Gaussian sources of uncertainty are approximated by Gaussian noise.

Viewed as a message passing algorithm, this is equivalent to linearising the factor nodes and approximating all messages with the closest Gaussian message.

3.2.3 Generalised Belief Propagation

Generalised Belief Propagation (GBP) uses clusters of nodes. Messages are sent between both normal nodes and the clusters. The famous Junction Tree algorithm is an exact form of GBP. Considering clusters of nodes allows for correlations between nodes to be transmitted.

GBP is more expensive than BP - how expensive depends on the level of clustering used. It can work on problems where BP fails [6]. An important open question is how the nodes should be clustered.

3.3 Expectation Propagation (EP)

Expectation propagation (EP) is an iterative technique for approximating complex distributions. It is not a message passing technique, although in special cases it can be related to message passing. Where it is suitable, EP can achieve more accurate estimation with comparable or less complexity than other techniques [7].

In EP, we approximate each factor with a distribution chosen from a restricted family, typically Gaussians when working with continuous variables. The name *expectation propagation* refers to the 'typical' case where the updates involve messages that propagate the mean (a.k.a the expectation) and variance (the 2nd expectation).

Note that EP approximates (usually multivariable) factors - *not* (single-variable) messages. This is closely related to the cluster-based messages of Generalised Belief Propagation, and indeed the two approaches can be shown to be formally connected [8]. If we restrict the approximation to a product of single-variable functions, that is we only consider fully-factorised distributions, then EP is equivalent to BP [7].

3.4 Monte-Carlo Methods

Integration over the possible states is a key step in statistical inference. Straightforward numerical integration is not tractable for high-dimensional state spaces - the computational cost is exponentially expensive relative to the dimensionality. Monte Carlo methods (i.e. sampling based methods) provide a practical way of calculating such integrals. Monte Carlo methods are a powerful and flexible tool. However they do require considerable computational resources.

It is often not possible to sample from a distribution using analytic methods. Markov-chain Monte Carlo (MCMC) provides a solution to this. MCMC methods use a random walk (guided by tractable 'local' calculations of the distribution) to 'get inside' the distribution and draw a sample. The idea is that the desired distribution is approximated as the limiting distribution of a Markov chain (see [9] for details). *Gibbs sampling* is a form of MCMC well suited to BGMs – including loopy graphs.

3.4.1 Particle Filters

Particle filtering [9] is a powerful application of importance sampling to state estimation in hidden Markov models. Adapting particle filters to work in general factor graphs formed part of this project.

A number of weighted samples (the *particle cloud* or just *the particles*) are generated for the root variable of the chain (i.e. these are samples for the hidden value at time 0). The particles are then propagated through the chain re-weighted at each step. The particles approximately define a distribution in which variables earlier in the chain have been marginalised out.

4. Approximate Graphs

The computational cost of a message passing algorithm is roughly proportional to the number of edges. If there are too many edges, inference becomes intractable. It is quite possible to construct BGMs which involve a vast number of nodes and edges.

The solution is to drop edges. If the graph becomes disconnected, then the sections are independent. Disconnected sub-graphs can be dropped entirely for a particular calculation. The question is which edges to

drop. Clearly it should be those that have little effect, i.e. those which reflect weak correlations. Several schemes are possible [10,11].

Ignoring a message is the same as approximating it with a constant function, i.e. using $(A \to B)(x) \approx 1$. This is a good approximation if the message did not carry much information. Such messages simply vanish from products. This allows us to treat graph approximations as a form of message type.

5. Hybrid Message Passing

The key idea behind the sum-product algorithm is the creation of messages which marginalise regions of the graph. These messages can be approximated in various ways, for example by sampling (Monte Carlo) or by using simpler functions. Different approximations are compatible and can be mixed. We have developed a framework based on message passing that allows hybrid algorithms to be built (described below). We call this approach *hybrid message passing* (HMP). Hybrid message passing is not an algorithm per se, but a way to plug together different algorithms that is both theoretically and pragmatically sound. The advantage of this approach is that it gives a great deal of flexibility. It covers exact messages, EP, Gaussian approximations, mixture models and particle messages for high-dimensional non-Gaussian variables.

A particular inference would use a mixture of deterministic and Monte-Carlo calculations. We hope this will allow us to work with genuinely high-dimensional problems without sacrificing speed and accuracy.

Combining different types of message is key to the functioning of hybrid message passing. In many cases there are fast analytic solutions to the product and integral of messages. These include most cases involving Gaussian messages or particle messages. Where analytic solutions do not exist, sampling-and-fitting allows us to move between representations.

6. Evaluation

We have conducted theoretical evaluations of HMP by considering several scenarios. We intend to follow these with empirical testing. Here we outline some of our findings.

6.1 Scenario #1: Gaussian Observations

Consider tracking an agent via a series of position measurements with Gaussian noise. We wish to estimate the current position.

The Kalman Filter is the optimal filter. It is both fast and accurate. Monte-Carlo methods will work but are a poor choice here. They are both less efficient and less accurate than the Kalman filter. Approximate message passing is almost identical to Kalman filtering.

6.2 Scenario #2: Proximity Detector Observations

Consider two agents ‘Alice’ and ‘Bob’ moving in a 1km square area. Let A and B be the random variables representing their positions. Both Alice and Bob have Gaussian priors. Alice is fitted with a proximity detector that has a fixed range of 10m. If Bob comes in range, the detector will be triggered. There is also an alarm which *should* be triggered if we infer that Alice and Bob are within 10m. I.e. both the detector and the alarm have uniform distributions in ||B−A|| given by:

$$\mathbb{P}(\text{detection}|\,\mathrm{A},\mathrm{B}) = \begin{cases} 1 & \text{if } ||\,\mathrm{B}-\mathrm{A}\,|| < 10, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbb{P}(\text{alarm}|\,\mathrm{A},\mathrm{B}) = \begin{cases} 1 & \text{if } ||\,\mathrm{B}-\mathrm{A}\,|| < 10, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that the proximity detector has been triggered, and let $f(a, b) = \mathbb{P}(\text{detection}|\, a,b)$. We are interested in two questions: What is the position of the agents? And will

the alarm be triggered as it should? Although simple, this scenario can be problematic in a couple of ways.

The Kalman filter and related algorithms are inappropriate here. $\mathbb{P}$(detection) is non-linear, so it does not fit into the Kalman filter. It can be treated using the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) [12]. On paper these appear to solve the non-linear problem, but in fact their performance is poor for this scenario.

A Monte-Carlo approach can solve both the position and the alarm questions, but it needs considerable computational power. Suppose Alice detects she is near Bob, but there is a good deal of uncertainty on either's location. We sample for Alice and Bob simultaneously. There is only a $\frac{1}{100^2}$ chance of drawing a sample where Alice and Bob are close. Most samples from the prior will place Alice and Bob far apart and hence have 0 posterior probability. We need a good number of nonzero probability samples to get a reliable approximation to the posterior, say 1,000. To get 1,000 non-zero probability samples for the posterior will require roughly 100 *million samples* from the prior.

Loopy Belief Propagation has no problem solving the position question. The messages from the proximity observation translate information about Bob's position into information about Alice's (and vice versa). BP has difficulty solving the alarm question though. Suppose we do not know where either Alice or Bob are (i.e. they have priors with high variance). The true posterior in this case is that the positions of Alice and Bob remain unknown, but the alarm is triggered. However using BP messages, *the alarm will not fire*.

For the position question, EP performs similarly to BP in terms of both accuracy and speed. EP can also solve the alarm question, even in the case where both Alice and Bob's position are unknown (where BP fails).

6.3 Scenario #3: Uncertain Identity

Suppose we are tracking multiple agents using video or image data. When an observation is made, the identity of the observed agent may be uncertain. Consider a simple detector which reports location and identity. It has high locational accuracy (e.g. with noise $\mathcal{N}(0, 1)$), but can mistake identity with probability ½. This leads to multimodal posteriors.

The particle filter naturally handles multimodal distributions, as does a mixture-model approach.

Any approach based on a uni-modal model such as a Gaussian will be unable to handle this form of uncertainty. This includes the Kalman filter, BP, and EP. The best that can be done is to ignore the mistaken identity and increase the observation noise to compensate.

6.4 A More Complex Scenario: Tracking Multiple Bots

Consider tracking *n* agents in a 2D arena. We have a map of the arena showing walls. The agents use two types of sensor:

1. Intermittent location detection with Gaussian noise.
2. Proximity detection between agents.

The agents are not all the same. Some agents have both sensors, some none (e.g. enemy or neutral agents). Moreover some agents are more important, and tracking them should be given priority.

This scenario can be modelled as a form of coupled HMM. Hybrid message passing allows us to use different methods depending on the importance of the agents. The less-important agents can be tracked using Gaussian messages. The more important agents can be tracked using

particle filters or mixture models. The hybrid framework allows for these two approaches to interact effectively.

One problem with message passing will be the size of the graph, which is $O(n^2.t)$. By limiting the amount of history stored and identifying uninformative events, we can get an approximate graph that grows linearly with respect to the number of agents and is constant with respect to time.

Overall the combination of Gaussian message passing, particle filtering and the ability to incorporate limited history should give improved performance.

7. Conclusions

In this report we have developed a novel class of hybrid inference algorithms, which we call hybrid message passing. This framework allows various deterministic and non-deterministic approximation methods to be combined in a systematic way. The resulting hybrid methods have the potential of dealing efficiently with high-dimensional probability distributions and complex problems without sacrificing accuracy.

This work builds upon existing inference algorithms for Bayesian graphical models (BGMs). As we have seen, many inference problems can be naturally expressed in this setting. BGMs can be converted into factor graphs, and message passing on factor graphs provides a convenient formalism for developing algorithms.

Hybrid message passing extends both Kalman filters and particle filters, so it is obviously suitable anywhere that either of these techniques are used. This covers a wide range of applications, including object tracking.

There has been surprisingly little prior work on combining deterministic and non-deterministic Bayesian inference methods. Dawid et al. [13] discuss hybrid methods in the context of junction trees. However, the paper focuses mainly on working with discrete probability distributions. More recently, Dauwels [14] discussed constructing hybrid message-passing algorithms by combining sampling methods e.g. with Expectation Maximisation, with some applications to signal processing.

Examining a number of simple scenarios (including tracking multiple agents with noisy observations and proximity detection) demonstrates that the hybrid approach offers significant advantages over traditional techniques.

The scenarios above show that each of the basic methods is the best for some situations – but will perform badly in others. By selecting the appropriate method, hybrid message passing can perform well in all the scenarios.

We believe that hybrid message passing is well suited for inference in autonomous systems. This includes collaboration between multiple agents, e.g. a swarm of robots. Message-passing on a graph translates naturally into message-passing between agents. The ability to balance accuracy requirements with computational resources also makes it attractive for robot platforms operating in complex environments. We hope to explore the potential of this approach in future work with the SEAS DTC.

8. References

[1] F. Kschischang, B. Frey, and H. Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Transactions on Information Theory 47 (2001), no. 2, 498–519.

[2] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] K. Murphy, *A Brief Introduction to Graphical Models and Bayesian Networks*, Available on-line at http://www. cs. ubc. ca/murphyk/Bayes/bnintro. html (1998).

[5] B. Frey and D. MacKay, *A Revolution: Belief Propagation in GraphsWith Cycles*, Advances in Neural Information Processing Systems 10 (1998).

[6] J. S. Yedidia,W. T. Freeman, and Y. Weiss, *Constructing free-energy approximations and generalized belief propagation algorithms*, Information Theory, IEEE Transactions on 51 (2005), no. 7, 2282–2312.

[7] T. P. Minka, *Expectation propagation for approximate Bayesian inference*, Uncertainty in Artificial Intelligence 17 (2001) 362–369.

[8] M. Welling, T. Minka, and Y. Teh, *Structured Region Graphs: Morphing EP into GBP*, Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05) 609.

[9] D. J. C. Mackay, *Information theory, inference, and learning algorithms*. Cambridge University Press New York, 2003.

[10] A. Choi and A. Darwiche, *An Edge Deletion Semantics for Belief Propagation and Its Practical Impact on Approximation Quality*, Proceedings of the National Conference on Artificial Intelligence 21 (2006), no. 2, 1107.

[11] A. Choi, H. Chan, and A. Darwiche, *On Bayesian Network Approximation by Edge Deletion*

[12] S. J. Julier and J. K. Uhlmann., *A new extension of the kalman filter to nonlinear systems*., AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II (1997).

[13] P. A. Dawid, U. Kjaerulff, and S. L. Lauritzen, *Hybrid propagation in junction trees*, in IPMU, pp. 87–97. 1994.

[14] J. Dauwels, *On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation*. PhD thesis, Swiss Federal Institute of Technology, 2006.

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.