Conditional Random Fields for Reinforcement Learning

Xinhua Zhang*, Douglas Aberdeen, S.V.N Vishwanathan
National ICT Australia
Locked Bag 8001
Canberra, Australia
firstname.lastname@nicta.com.au

Distributed reinforcement learning (RL) involves a collection of nodes, or *agents*, choosing actions to maximise a long-term reward measure. Examples of such domains are traffic routing for roads or networks, sensor networks, pursuer-evader problems, and job-shop scheduling. The simplest algorithms assume all agents are independent, learning to cooperate only through a shared reward function. More advanced algorithms explicitly share information about state, or factor the global reward into local rewards [1]. But in all these cases each node chooses its action independently. A naïve fix is to make decisions sequentially, allowing nodes to condition their actions on decisions that earlier nodes made.

However, we would much prefer that the nodes choose the optimal *joint* set of actions, taking into account the actions of all other relevant nodes. We use conditional random fields (CRFs) to efficiently model the conditional dependencies between agents. The same inference methods used for CRFs can be used to sample node actions from a *joint* stochastic policy. We also show how to optimise this joint policy by estimating the gradients of the long-term average reward with respect to the policy parameters. Moreover, similar methods could be used for RL policies based on arbitrary graphical models.

CRFs are traditionally used to model $P(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\theta})$, which is the probability of a set of labels $\mathbf{y}$, conditioned on observable variables $\mathbf{x}$ and the CRF parameters $\boldsymbol{\theta}$ [4]. CRF training iterates through sets of training instances $\{\mathbf{x}, \mathbf{y}\}$, finding $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|X, Y)$. To predict labels for a novel observation $\mathbf{x}'$ we select labels $\mathbf{y}' = \arg\max_{\mathbf{y}} P(\mathbf{y} \,|\, \mathbf{x}'; \boldsymbol{\theta}^*)$. To extend CRFs to online temporal processes Dynamic Bayesian Networks (DBN) have been used, unfolding the CRF model over time. Another interpretation of our work is that we show how CRF parameters can be adapted online for time-series prediction, and control, without needing DBN models.

Our RL framework is that of distributed partially observable Markov decisions processes. Each RL agent is represented by a node in the CRF. The input vector $\mathbf{x}$ represents the total set of observations/features presented to all the agents. Actions are equivalent to hidden labels $\mathbf{y}$, each element in $\mathbf{y}$ representing a single node's action. The optimisation task is to find the CRF parameters $\boldsymbol{\theta}$ such that sampling joint actions $\mathbf{y}_{(t)}$ from $P(\cdot \,|\, \mathbf{x}_{(t)}; \boldsymbol{\theta})$ maximises the long-term average reward $R(\theta) = \lim_{T\to\infty} 1/T \sum_{t=1}^{T} r_{(t)}$ (a discounted model may also be used).

The CRF/policy distribution is represented as an exponential family $P(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\theta}) = \exp(\langle \phi(\mathbf{x}, \mathbf{y}), \boldsymbol{\theta} \rangle - z(\boldsymbol{\theta}|\, \mathbf{x}))$. Here ϕ is the *sufficient statistic*, a vector of features for nodes and edges; and z is the log partition function $z(\boldsymbol{\theta}|\, \mathbf{x}) := \ln \sum_{y \in \mathcal{Y}} \exp(\langle \phi(\mathbf{x}, y), \boldsymbol{\theta} \rangle)$. Node features represent the observation of state available at each node. The edge features encode the communication between nodes about their actions and features. It is worth noting that this exponential family representation, with a dot product between features and parameters, implements exactly the *soft-max* stochastic policy with linear feature combination commonly encountered in RL applications. Only the edge features prevent trivial factorisation of the distribution into independent agents.

Thus the policy distribution is complex to evaluate, however using CRFs allows the clique decomposition theorem to come into play, decomposing the distribution into terms over maximal cliques $c \in \mathcal{C}$ of the CRF graph so that $P(\mathbf{x} \,|\, \mathbf{y}; \boldsymbol{\theta}) = \exp(\sum_{c \in \mathcal{C}} \langle \phi_c(\mathbf{x}, \mathbf{y}_c), \boldsymbol{\theta}_c \rangle - z(\boldsymbol{\theta}|\, \mathbf{x}))$.

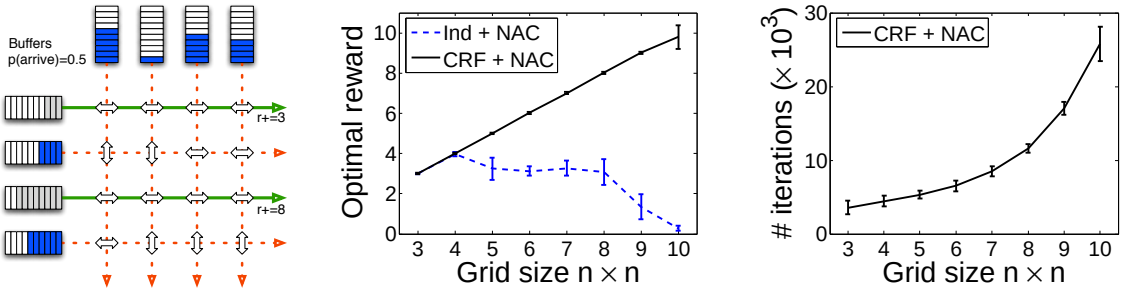
For example, in a 1D CRF (chain) the cliques are the set of all adjacent nodes i and j. An often useful clique sufficient statistic in this case is $\phi_{ij}(\mathbf{x}, y_i, y_j) = [\mathbf{x}, 1]^\top$ for connected nodes i, j if $y_i = y_j$, and $[\mathbf{x}, 0]^\top$ otherwise, encoding whether neighbouring nodes are selecting the same action.

*__Topic:__ graphical models, control __Preference:__ Oral/Poster

To evaluate and sample from this still complex distribution we enlist methods from graphical model inference. For 1D CRFs dynamic programming can be used to efficiently compute $P(\mathbf{x} \mid \mathbf{y}; \boldsymbol{\theta})$. For 2D CRFs, representing a mesh of RL agents, we employed the tree MCMC sampler [3].

But this does not solve the problem of optimisation. Policy-gradient (PG) algorithms are appealing here because they directly optimise stochastic policies, rather than needing to infer values of actions. We used the recent natural actor critic algorithm [5]. Such algorithms estimate $\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta})$ by generating trajectories through the POMDP, following the current stochastic policy. At each step $\nabla_{\boldsymbol{\theta}} \log P(\mathbf{y}_{(t)} \mid \mathbf{x}_{(t)}; \boldsymbol{\theta})$ is computed and added to a discounted eligibility trace $\mathbf{e}_{(t)}$. In [2] it was shown that the one step estimate of $\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = r_{(t)}\, \mathbf{e}_{(t)}$. Stochastic gradient ascent can then be used to adjust the parameters at each step. This is the core process of many policy gradient algorithms, including the natural actor-critic.

Our preliminary experiments are on an abstract traffic domain, illustrating a case where independent RL nodes fail to find the optimal policy, despite a global reward. Nodes are arranged on an $n \times n$ grid. Each node is responsible for a gate that allows traffic to flow vertically or horizontally. If *all* the gates along a row or column of the graph align, then all buffered traffic at dummy boundary nodes will flow through the graph in that step. A single misaligned gate blocks traffic, causing the length 10 buffer to fill up as traffic arrives, possibly dropping traffic. Traffic units arrive with Pr=0.5 per time step per boundary node. The reward is how many traffic units passed through the graph. Two features per node indicate the number of traffic units waiting for the node to align vertically, and horizontally. Additionally, each edge in the graph generates a feature of 1 if the two nodes agree on an alignment. The optimal policy is for the *all* gates to align in the orientation of the most waiting traffic, but since each node only knows how many traffic units are waiting for it, it must "negotiate" with neighbours on which way they wish to align. The optimal reward is the grid size n. The left graph shows the CRF RL approach compared to a naïve implementation with indepenent agents that do not use edge features (labelled NN-NAC). Averaged over 100 runs per grid size, the CRF approach obtains the optimal reward all the way to grid size 10 (100 nodes), at which point some runs fail to reach the optimal policy. The right graph shows the number of learning iterations.



References

[1] J. Andrew Bagnell and Andrew Y. Ng. On local rewards and scaling distributed reinforcement learning. In *Proc. NIPS'2005*, volume 18, 2006.

[2] J. Baxter and P.L. Bartlett. Infinite-horizon policy-gradient estimation. *JAIR*, 15:319–350, 2001.

[3] Firas Hamze and Nando de Freitas. From fields to trees. In *UAI*, 2004.

[4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*. Morgan Kaufmann, 2001.

[5] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *ECML 16, Porto, Portugal*, pages 280–291. Springer, 2005.