Application of new temporal difference learning methods for approximate solution of large linear systems

Hamid R. Maei, Richard S. Sutton, Csaba Szepesvári
Department of Computing Science, University of Alberta

Bertsekas and Yu (2009) recently proposed simulation based temporal difference (TD) learning methods that can be used to approximate the solution of very large linear systems of equations, whose analytic solution is intractable. This is a potential new, large, important class of applications of reinforcement learning ideas. Borrowing the idea used in TD methods with linear function approximation, the key feature of their proposal is to project the original systems of equations into a lower n-dimensional subspace (feature space) to obtain the approximate solution in an iterative manner. However, the limitation of conventional TD methods with $O(n)$ complexity is that they only converge for a narrow class of linear systems.

Here, we propose the application of new temporal difference algorithms based on gradient descent, GTD, (Sutton et al. 2009) in approximating large linear systems. GTD methods are guaranteed to converge for any arbitrary linear systems of equations. Such algorithms are gradient descent version of TD and previously have been proposed in reinforcement learning problems (Sutton et al. 2009).

Large linear systems of equations appear in the context of dynamic programming, where the problem is to evaluate the value of each state given the model of the environment. For such a large systems of equations each feature brings a value and practitioners are loath to give any of them up, therefore when the dimensionality of the feature space is still large (e.g. 10^6 in computer Go, Silver et al. 2007), second order methods whose storage and computational complexity is $O(n^2)$ are impractical. To find an approximate solution, bootstrapping reinforcement learning techniques with $O(n)$ complexity, such as temporal difference learning TD algorithms (e.g. TD(0)) have been proposed in conjunction with linear function approximation. Indeed one of the methods proposed by Bertsekad and Yu (2009), is a simulated based extension of TD(0) used in conjunction with linear function approximation. However, the TD(0) algorithm only converges for narrow class of problems whose projection matrix in low dimensional space has contraction properties.

Several new TD methods based on gradient descent, GTD-2(0) and TD-C(0), (Sutton et al. 2009) do not require such contraction properties for convergence

guarantee. Just like TD(0), these algorithms are O(n), incremental and are suitable for online use. However, unlike TD(0), they are guaranteed to converge under fewer conditions and thus can be used to approximate the solution of any arbitrary large systems of linear equations.

Equation approximation using GTD-2(0) and TD-C(0): Suppose we want to solve systems of equations of the form $x = Ax + b$, where $x \in \Re^N$, $b \in \Re^N$ and A is $N \times N$ matrix. The approximate solution of this equation in n-dimensional space ($n \ll N$), $x \approx \Phi\theta$, where $\theta \in \Re^n$ and Φ is $N \times n$ feature matrix whose rows are the feature vectors with size n.

To avoid matrix-vector computation, we use sampling technique. To do this, let's generate a sequence of transitions between indices, $\{(i_0, j_0), (i_1, j_1), ...\}$ using a Markov chain transition matrix P with desirable property of $p_{i_k j_k} \neq 0$ if and only if $a_{i_k j_k} \neq 0$, where $a_{i_k j_k} \equiv [A]_{i_k j_k}$ and $p_{i_k j_k} = [P]_{i_k j_k}$, for kth sample. Let's assume $(\phi(i_k), b_{i_k}, \phi(j_k))$ represents the kth observed sample, where $\phi(i_k)$ and b_{i_k} indicate the feature vector of index i_k and i_kth element of vector b respectively. Given such samples, here we consider GTD-2(0) and TD-C(0) algorithms:

GTD-2(0) algorithm The parameter update for GTD-2(0) algorithm is: $\theta_{k+1} = \theta_k + \alpha_k(\phi(i_k) - \frac{a_{i_k j_k}}{p_{i_k j_k}}\phi(j_k))w_k^\top \phi(i_k)$, where w_k update is: $w_{k+1} = w_k + \beta_k(\delta_k - w_k^\top \phi(i_k))\phi(i_k)$. Here, α_k and β_k are step size parameters and, $\delta_k = b_{i_k} + \frac{a_{i_k j_k}}{p_{i_k j_k}}\theta^\top \phi(j_k) - \theta^\top \phi(i_k)$, represents TD error.

TD-C(0) algorithm: Known as TD(0) with gradient term correction, updates the parameters θ as: $\theta_{k+1} = \theta_k + \alpha_k \delta_k \phi(i_k) - \alpha_k \frac{a_{i_k j_k}}{p_{i_k j_k}}\phi(j_k)w_k^\top \phi(i_k)$, where w_k update is similar to GTD-2(0) algorithm.

With standard step-size assumptions, Sutton et al. (2009), have shown that the above algorithms converge to TD(0) solution; that is, approximate solution of large linear systems in the form $x = Ax + b$.

References

D. P. Bertsekas and H. Yu. (2009) Projected equation methods for approximate solution of large linear systems, J. Computational and Applied Mathematics, Volume 227 , Issue 1, Pages 27-50

Silver, D., Sutton, R. S., and Mueller, M. (2007) Reinforcement Learning of Local Shape in the Game of Go, Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07).

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvari, Cs., Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada.