An Accelerated Gradient Method for Distributed Multi-Agent Planning with Factored MDPs

Sue Ann Hong
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
sahong@cs.cmu.edu

Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
ggordon@cs.cmu.edu

Abstract

We study optimization for collaborative multi-agent planning in factored Markov decision processes (MDPs) with shared resource constraints. Following past research, we derive a distributed planning algorithm for this setting based on Lagrangian relaxation: we optimize a convex *dual function* which maps a vector of resource prices to a bound on the achievable utility. Since the dual function is not differentiable, the most common method for optimizing it is subgradient descent. This method is appealing, since we can compute the subgradient by asking each agent to plan independently of the others using the current resource prices; however, subgradient descent unfortunately requires $O(\epsilon^{-2})$ iterations to achieve accuracy ϵ, and therefore the overall Lagrangian relaxation algorithm can have trouble scaling to realistic domains. So, instead, we propose to optimize a smoothed version of the dual function via a fast proximal gradient algorithm. By trading the error caused by smoothing against the faster convergence of the proximal gradient method, we demonstrate that we can obtain faster ($O(\epsilon^{-1})$) convergence of the overall Lagrangian relaxation. Furthermore, we propose a particular smoothing method, based on maximum causal entropy, for which the subgradient calculation remains simple and efficient.

1 Introduction

Multi-agent planning is often naturally formulated as a factored combinatorial optimization problem with shared resources: each agent has its own local state, constraints, and objectives, while agents interact by competing for scarce, shared resources. Such a problem is usually intractable as a centralized optimization problem, since the joint state over all agents is exponential in the number of agents, and therefore standard centralized optimization techniques (such as simplex or log barrier methods in the case of linear programming) are not typically applicable. Hence, given the natural factorization over agents, it is beneficial to seek a distributed solution, where each agent solves its individual local planning problem with a fast single-agent planning algorithm. In addition, planning is a fundamental subroutine in learning utility functions from data (e.g. inverse reinforcement learning), and therefore efficient planning methods are essential for learning.

Here we consider multiagent planning in factored MDPs with linear rewards, where each agent's sub-problem is represented as an MDP, and agents interact with one another through constraints on shared resources. Guestrin et al. [1] give a distributed planning algorithm in this setting based on Benders decomposition, which can be considered a form of Lagrangian relaxation. As we will see, Lagrangian relaxation decomposes the overall planning problem such that each agent's individual planning problem can be solved independently from the others, based on dual values associated with the shared resource constraints. The dual values act as *prices* on the shared resources, thereby balancing the resource usage by the agents. Consequently, our goal becomes to find the optimal dual values.

A natural way to optimize the dual values for factored MDPs is to use a gradient-based method such as subgradient descent, or more preferably, an accelerated gradient method such as FISTA [2], which provides faster convergence. (Computing the subgradient of the dual function amounts to planning in each agent's sub-problem independently, using the current resource prices.) However, accelerated gradient methods are not immediately applicable to planning in factored linear MDPs, as the Lagrangian dual function does not have a Lipschitz continuous gradient. To address this limitation, we propose instead to optimize the dual function for a strongly convex penalized version of the optimization problem.

While the penalty term gives us a strongly convex objective function (and therefore ensures that the dual function has a Lipschitz continuous gradient, so that the accelerated gradient algorithm will work), it introduces an error to the desired objective. However, the strength of the penalty term determines a bound on the error as well as a bound on the convergence rate of the accelerated gradient algorithm. So, given a computational budget, we can optimize for the penalty strength which minimizes total error, by balancing error due to the perturbed objective against error due to incomplete convergence.

The above trick of optimizing the strength of a strongly convex penalty is often used in combination with accelerated gradient methods, and has been successful in other optimization problems related to machine learning. However, in the case of factored MDPs, we face an additional difficulty: if we add a strongly convex penalty to the MDP objective, we can no longer use fast MDP planning algorithms such as value iteration. If we were forced to back off to general convex optimization methods when solving the MDP subproblems, we would lose much of the benefit of the accelerated gradient algorithm: our iteration count would be lower, but each iteration would be much more expensive. To address this issue, we propose to use a maximum causal entropy penalty [3]: with this penalty, we can still use a slight modification of value iteration to solve each individual MDP, while gaining the benefit of faster overall convergence.

In the following, we formally present our problem setting and the algorithm, then show experimental results in a congested path planning domain that suggest that, with the correct balance, we can achieve much faster overall convergence without sacrificing the solution quality.

2 Problem Definition

In this section, we formulate the individual MDP planning problems and the shared constraints as a mathematical program, which is used to derive the Lagrangian relaxation algorithm in Section 3.

Let random variables A_{it} and S_{it} represent the action and state of agent i at time t respectively. For agent i, given the distribution over the initial states $P(S_{i1})$, the transition probabilities $P(S_{i,t+1}|A_{it}, S_{it})$, and the rewards $r_{S_{it}}$, the goal of linear reward MDP planning is to find a policy, $P(A_{it}|S_{it})$, that optimizes:

$$\max_{P(A_{it}|S_{it}),P(S_{it})} \sum_{t=1}^{T}\sum_{S_{it}} r_{S_{it}} P(S_{it}) \tag{1}$$

s.t.

$$P(S_{it}) = \sum_{S_{i,t-1}}\sum_{A_{i,t-1}} P(S_{it}|A_{i,t-1}, S_{i,t-1})P(A_{i,t-1}|S_{i,t-1}) \tag{2}$$

$$\sum_{S_{it}} P(S_{it}) = 1, \quad \sum_{A_{it}} P(A_{it}|S_{it}) = 1. \tag{3}$$

We augment the objective function with causal entropy, a common regularizer,

$$\frac{1}{\beta}\sum_{t} H(A_{it}||S_{it})$$

to obtain a strongly convex objective, where

$$H(A_{it}||S_{it}) = -\sum_{A_{it},S_{it}} P(A_{it}, S_{it}) \log P(A_{it}|S_{it})$$

and $\frac{1}{\beta}$ controls the contribution of the causal entropy term in the objective.

We also assume shared constraints among the agents that are piecewise-linear in $P(S_{it})$. We denote matrices $D_1, ..., D_n$ such that D_i defines the linear resource usage $D_{ijst}P(S_{it})$ of agent i of resource j at time t. Then the total resource usage of resource j at time t over all agents is $\sum_i D_{ij}P(S_{it})$. We impose resource constraints using a hinge loss $f_j(x) = \max(0, \alpha_j(x - c_j))$ on the total resource consumption[1].

Finally, our overall multi-agent optimization problem can be written as:

$$\max_{P(A_{it}|S_{it}),P(S_{it})} \sum_{i=1}^{n} \frac{1}{\beta} H(A_{it}||S_{it}) + r_i^{\mathrm{T}} P(S_{it}) - \sum_{j=1}^{r} f_j(\sum_i D_{ij} P_i(S_t)) \text{ s.t. } \forall i, C_i \tag{4}$$

where C_i denotes the constraint set defined by equations (2)–(3) for each agent i. It is easy to see that our problem mostly contains independent parts: the objective function with the exception of the hinge loss is decomposable, and the constraints C_i are only over agent i's policy. We show in the next section how Lagrangian relaxation leads to a naturally distributed algorithm where each agent's planning may be performed in parallel.

3 Lagrangian Relaxation and an Accelerated Gradient Method

For simplicity, let us denote as vector x_i the policy probabilities $P(A_{it}|S_{it})$, and y_i as the state probabilities $P(S_{it})$. We can write the Lagrangian dual of (4) as

$$\inf_{\lambda} V_\beta(\lambda) = \inf_{\lambda} \max_{x,y} \sum_{i=1}^{n} \frac{1}{\beta} H(x_i) + r_i^{\mathrm{T}} y_i + \sum_{j=1}^{r} [f_j^*(\lambda_j) - \lambda_j \sum_i D_{ij} y_i] \text{ s.t. } \forall i, C_i,$$

which provides an upper bound on the value of (4). Expanding the dual function $f_j^*(\lambda_j) = \sup_x[\lambda_j x - f_j(x)]$, we obtain

$$V_\beta(\lambda) = \max_{x,y} \sum_{i=1}^{n} \frac{1}{\beta} H(x_i) + r_i^{\mathrm{T}} y_i + \sum_{j=1}^{r} [I(0 \le \lambda_j \le \alpha_j) + \lambda_j (c_j - \sum_i D_{ij} y_i)] \text{ s.t. } \forall i, C_i,$$

where c_j, α_j are the threshold and slope of hinge loss f_j, and the indicator function $I(Z)$ equals ∞ if condition Z is not satisfied, and equals 0 otherwise.

Note that $V_\beta(\lambda)$ is differentiable for λ s.t. $\lambda_j \in (0, \alpha_j), \forall j$. For such λ, it can be shown that a subgradient of $V_\beta(\lambda)$ is $c - \sum_i D_i y_i^*$, where

$$(x_i^*, y_i^*) = \arg\max_{x_i, y_i} \frac{1}{\beta} H(x_i) + (r_i - \lambda D_i)^{\mathrm{T}} y_i.$$

This fact gives us a subgradient projection method to optimize over λ, where to compute the gradient y_i^* can be calculated by each agent i in a distributed fashion given the current λ. In particular, for our objective, we can use softmax value iteration to find y_i^*.[2]

Furthermore, since $\sum_{i=1}^{n} \frac{1}{\beta} H(x_i)$ is strongly convex in x_i, y_i,

$$\max_{x,y} \sum_{i=1}^{n} \frac{1}{\beta} H(x_i) + r_i^{\mathrm{T}} y_i + \sum_{j=1}^{r} \lambda_j (c_j - \sum_i D_{ij} y_i) \text{ s.t. } \forall i, C_i$$

has a Lipschitz continuous gradient with respect to λ, allowing us to apply FISTA [2] to obtain the accelerated gradient Lagrangian relaxation algorithm (AGLR), shown in Algorithm 1.

Note that the number of iterations required for FISTA to obtain an ϵ-optimal solution is bounded by $O(\sqrt{L}/\sqrt{\epsilon})$, where L is a Lipschitz constant of $\nabla V_\beta(\lambda)$; hence the error at iteration T is bounded

[1] Any piece-wise linear penalty function will work, but we assume hinge loss for concreteness.

[2] Note that the gradient of $V_\beta(\lambda)$ with respect to λ_j is exactly the excess usage of resource j, which gives rise to the price interpretation of the dual variables.

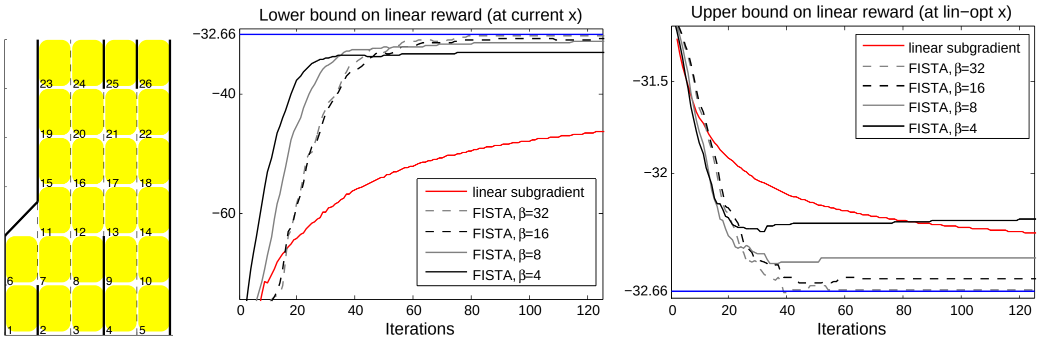
Algorithm 1 The accelerated gradient Lagrangian relaxation algorithm

```
Input: $L$, a Lipschitz constant of $\nabla V_\beta(\lambda)$.
$\lambda_0 \leftarrow 0$
$\nu_1 \leftarrow \lambda_0$
$d_0 \leftarrow 1$
for $t \leftarrow 1, 2, ...T$
    • $\forall i, x_i^*, y_i^* \leftarrow \arg\max_{x_i, y_i} \frac{1}{\beta} H(x_i) + (r_i - \lambda D_i)^{\mathrm{T}} y_i$ (solved by softmax value iteration)
    • $\forall j, \lambda_{tj} \leftarrow \nu_{tj} - \frac{1}{L}(c_j - \sum_i D_{ij} y_i^*)$
    • $\forall j, \lambda_{tj} \leftarrow \max(0, \min(\lambda_{tj}, \alpha_j))$
    • $d_{t+1} \leftarrow \frac{1+\sqrt{1+4d_t^2}}{2}$
    • $\nu_{t+1} \leftarrow \lambda_t + (\frac{d_t - 1}{d_{t+1}})(\lambda_t - \lambda_{t-1})$
```

by $O(\beta/T^2)$. We can also show that the suboptimality of the solution obtained by maximizing the causal entropy objective (4) is bounded by $(T_H \log |A|)/\beta$, where $|A|$ denotes the size of the set of actions in the MDP, and T_H the time horizon. Hence, given a budget of T for the number of iterations of FISTA, we can optimally balance the two sources of error by setting $\beta = O(T)$, yielding a total error of $O(1/T)$.

4 Experiments

We studied the convergence rate and quality of AGLR on a path planning domain of X-shaped bridges and interchanges, where vehicles are forced to change lanes and contend for shared road space. Such interchanges are a common culprit in heavy traffic, as agents are often trying to cross many lanes in a short section of a road. In Figure 1(a) we depict an example problem instance, where states (physical locations of vehicles) are represented by yellow rectangles labeled with state IDs. In our experiments, each agent starts at one of the first five states (1–5, i.e., entering the bridge) and must reach its goal state (randomly picked as one of 23–26, i.e., exiting the bridge). In each state, agents are allowed 4 actions, corresponding to staying in the current state or moving in each forward direction (moving one state to the left-forward, straight, or right-forward), but cannot cross the solid lines.



(a) (b) Convergence of lower bound (primal) (c) Convergence of upper bound (dual)

Figure 1: (a) the planning domain, an X-shaped bridge. (b)–(c) convergence comparisons.

The experimental results show how the choice of the smoothing parameter β can balance the trade off between convergence and optimality to achieve accelerated convergence while maintaining solution quality comparable to directly optimizing the linear reward function (despite the bias introduced by the causal entropy term). Figure 1(b) shows the convergence of AGLR with different values of smoothing parameter β, in terms of the linear primal values (without the causal entropy term), computed at the solution to the regularized objective at each iteration. Hence these values correspond to the solution available to be used at each iteration, and also serve as lower bounds to the optimal solution. The lower bounds demonstrate that in all cases optimizing the regularized objective leads to improved convergence albeit at the expense of optimality with respect to the unregularized objective. A lower value of β means the objective has a greater weight on causal entropy, and unsurprisingly, runs with lower β values converge more quickly, but to worse linear objective values. However, the subgradient Lagrangian relaxation directly optimizing the linear objective function (the solid red line), while eventually reaching the optimum solution (whereas running FISTA with a large β may not), converges extremely slowly; at iteration 1000 it achieves a lower bound of -38.543, which FISTA with $\beta = 4$ reaches at iteration 20.

In Figure 1(c) we plot upper bounds of the optimal primal value of the same AGLR runs; at each iteration, we plot $V_\infty(\lambda)$, i.e., the dual value of the *linear* objective at the current λ. The plot shows the trade-off between convergence speed and quality very similar to that in the lower bound plot. Moreover, it shows that with a sufficiently high value of β, we can approach the optimum very quickly; the blue horizontal line in both Figures 1(b)-1(c) represent the same value that is bounded and closely approached by the $\beta = 32$ curves in both plots. This shows that with a high enough value of β, we may recover the optimal solution with substantially fewer iterations than subgradient descent.

In this example setting, $\beta = 32$ appears sufficient to achieve a good solution and very fast convergence. However, it would be interesting to see if any gain would come from adaptively tuning β over the gradient descent iterations to achieve even faster convergence while maintaining good solution quality.

5 Conclusion

We presented an efficient distributed algorithm for planning with factored MDPs based on Lagrangian relaxation and FISTA. Preliminary results are promising and show the superior convergence of the algorithm compared to the standard subgradient algorithm. We plan to further study the algorithm on larger and more complex problems, and to extend the algorithm to adaptively control the weight of the causal entropy smoothing term to further speed up convergence.

Acknowledgements

The authors gratefully acknowledge support from ONR MURI grant number N00014-09-1-1052, and from a grant from Intel Corporation.

References

[1] Carlos Guestrin and Geoffrey J. Gordon. Distributed planning in hierarchical factorered MDPs. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 2002.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. on Imaging Sciences*, 2(1):183–202, 2009.

[3] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proc. of the International Conference on Machine Learning*, pages 1255–1262, 2010.