Programming Assignment

Approximate Dynamic Programming & Reinforcement Learning

Chair for Data Processing
TUM Department of Electrical and Computer Engineering
Technical University of Munich

Student: Petar Bevanda
ID: 03700814

January 30, 2019

Probabilistic Formulation of the Problem

The problem in the naive implementation for all possible combinations of the state, action, next state triplets is that the size of the problem might become much larger than necessary. In this implementation there was a list of actions possible given a state. And the probability transition matrix along with one-step-costs is represented inside a nested dictionary with the elements being lists of tuples of action probability, next state, and cost.

Policy

The policy is represented as a 2D array with row number representing the state number and the list in that row having five representing numbers for each of the five actions. The best action in the state is given a cost of one and all others zero. This could be also further simplified by using a dictionary but I kept my previous representation as it has not much impact on performance in problems of this size.

Solution with Dynamic Programming

The ”linearization” of the state space is done by mapping row-wise the admissible state's number position on the grid to the respective element of the cost vector.

Determining the DP algorithm convergence is done by looking if the maximal cost function change across any state is less than a set up threshold (very small number near zero). Also, policy iteration (PI) and value iteration (VI) generate the same policy. Varying the discount factor has an effect on the policy as it determines how heavily will we weight the previous cost in the state, meaning the smaller it gets the more it is evaluating the just the one stage forward.

A Study of Algorithm Properties

PI is actually slower by looking at the time needed because it requires VI in every iteration for the policy evaluation. But on the other hand, looking at the number of iterations, PI is much faster. When wanting performance while having the computing power PI is the way to go.

Used are α values of $0.8, 0.7, 0.6$ for error plots of VI and PI under both one step costs. Those were chosen as they produce the policy seen in the $\alpha = 0.9$. Going to lower discount factors such as $\alpha = 0.01$ produces non optimal behaviour. For the cost function g_1 the algorithms have relatively comparable performance. VI takes more iterations to converge that PI but both get in the close vicinity of the fixed point in similar number of iterations.

The biggest difference can be observed actually in the case of the cost g_2, where it is evident what policy iteration is a better/faster than value iteration, as the policy more often than not converges faster than a value function. The key is that PI pursues the finding of the optimal behaviour/policy which can happen before we get the optimal value function and only then extracting the optimal policy from it as it is done in VI.

The performance can be boosted by having a good guess in the initial policy of PI that is usually randomly initialized. In my case i had it "empty" to have nicely scaled error plots of the VI and PI compared to ground truths. Also worth noting is that with the g_1 one step cost the difference to the ground truths for the chosen discounts is not big as in the case of g_2 (Fig. 5, Fig. 6). The reason for that lies in the fact that for cost g_2 getting to the goal is being "rewarded" as well as the self loop as the cost zero is the smallest in that setting. The implies one has to look further (more long-term) to get to the get the optimal policy which explains why discount factors smaller than 0.99 converge to a much bigger error with respect to the ground truth cost.

Also worth noting is that going the longer way around because of the trap seen in the ground truth case with cost g_1 is non-existent in the case of g_2 (Fig. 1). In that case idling in the terminal state is "rewarded" and along with a big discount factor leads to the agent's "fearless" behaviour near the trap.

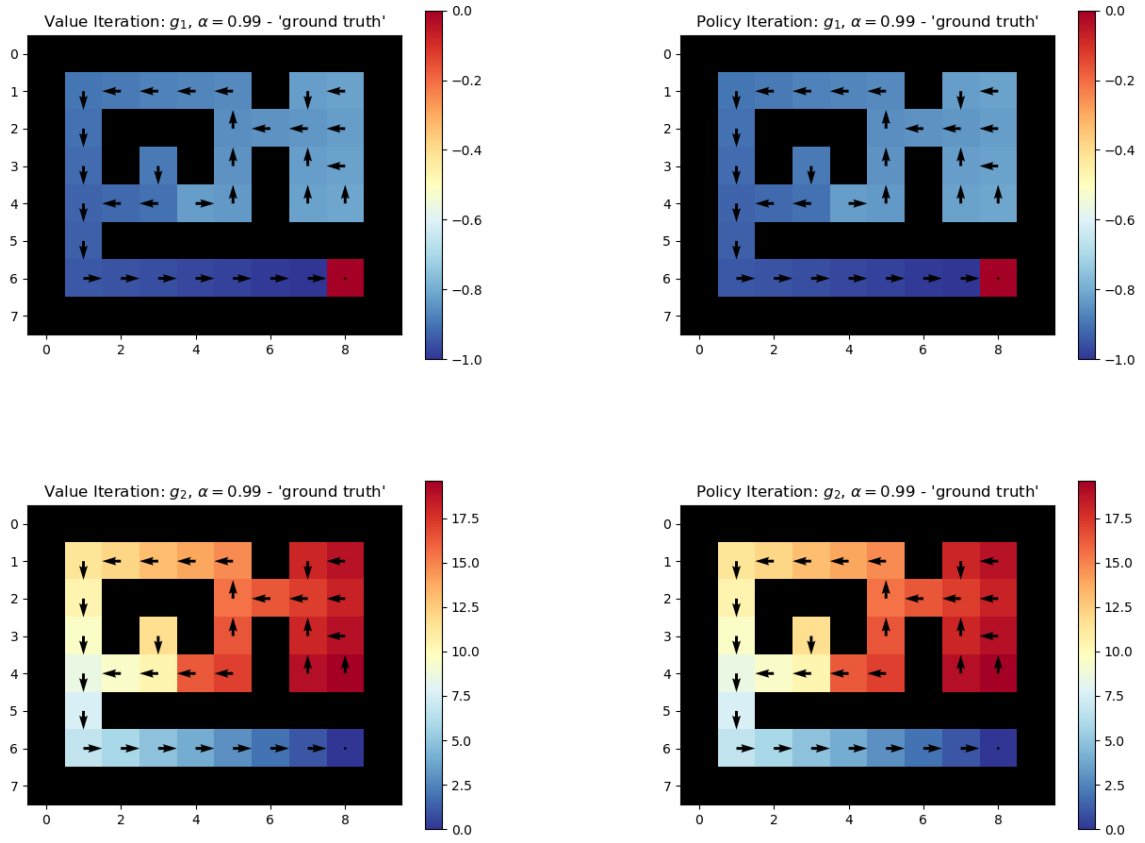

Figure 1: Ground truth costs and policies

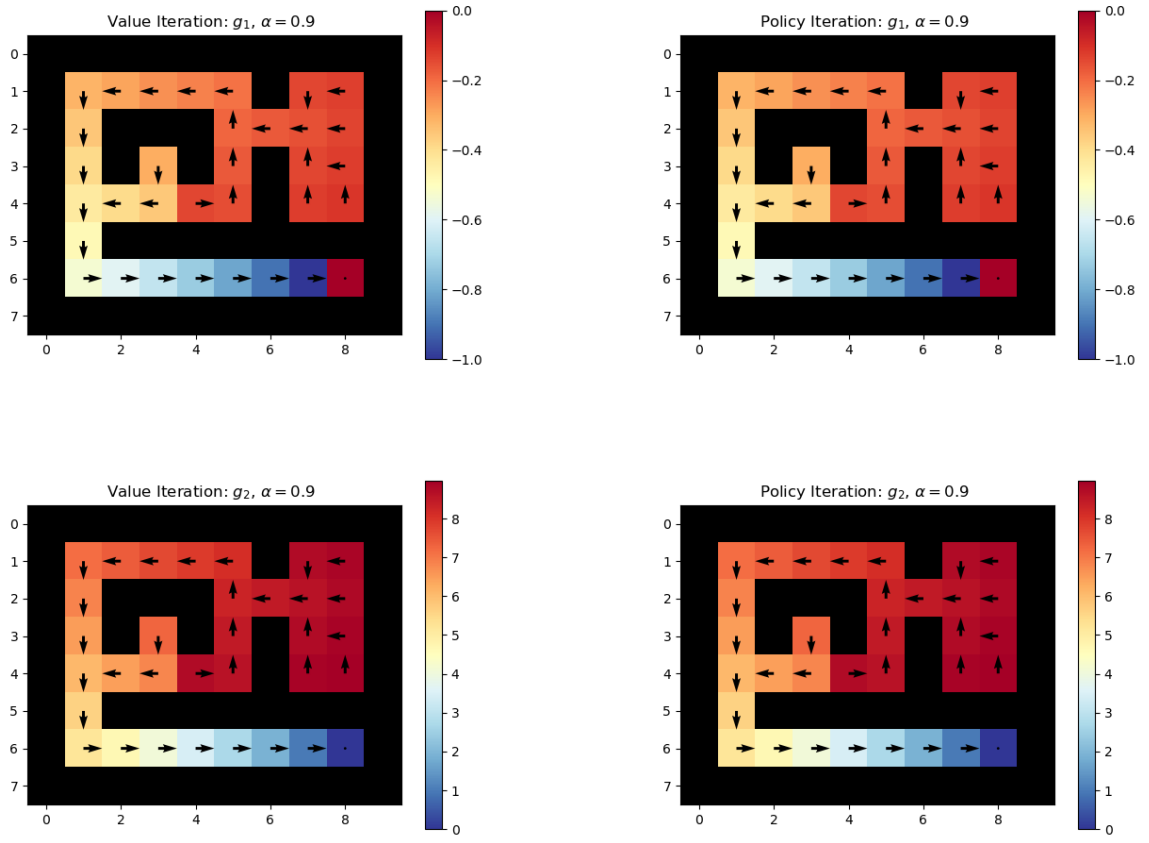

Figure 2: Cost and policy plots for $\alpha = 0.9$

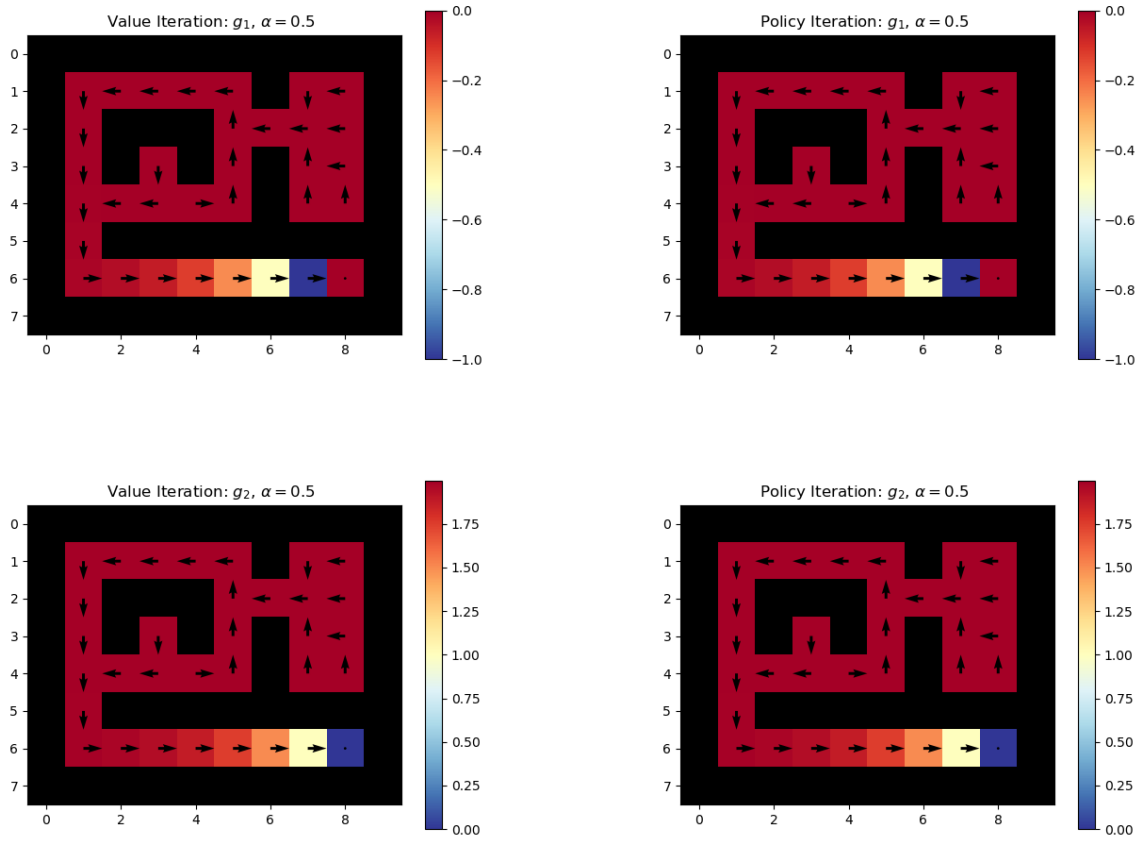

Figure 3: Cost and policy plots for $\alpha = 0.5$

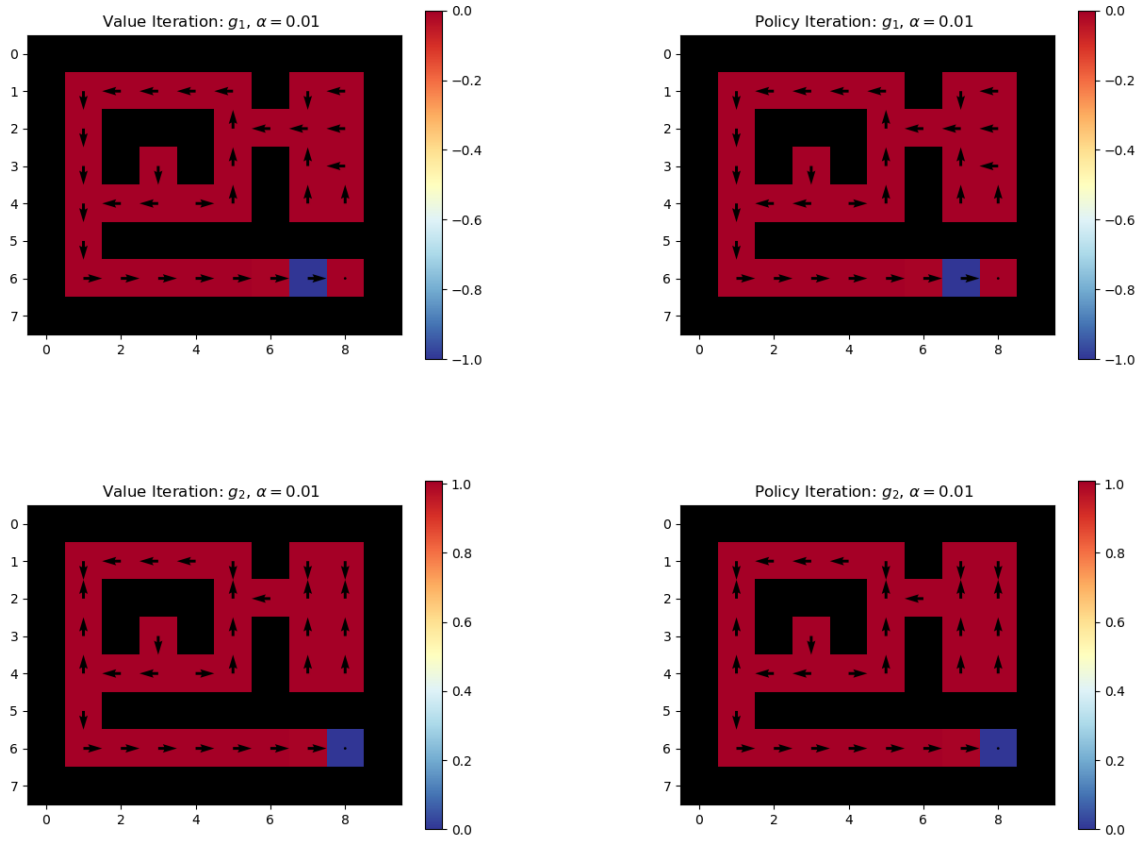

Figure 4: Cost and policy plots for $\alpha = 0.01$

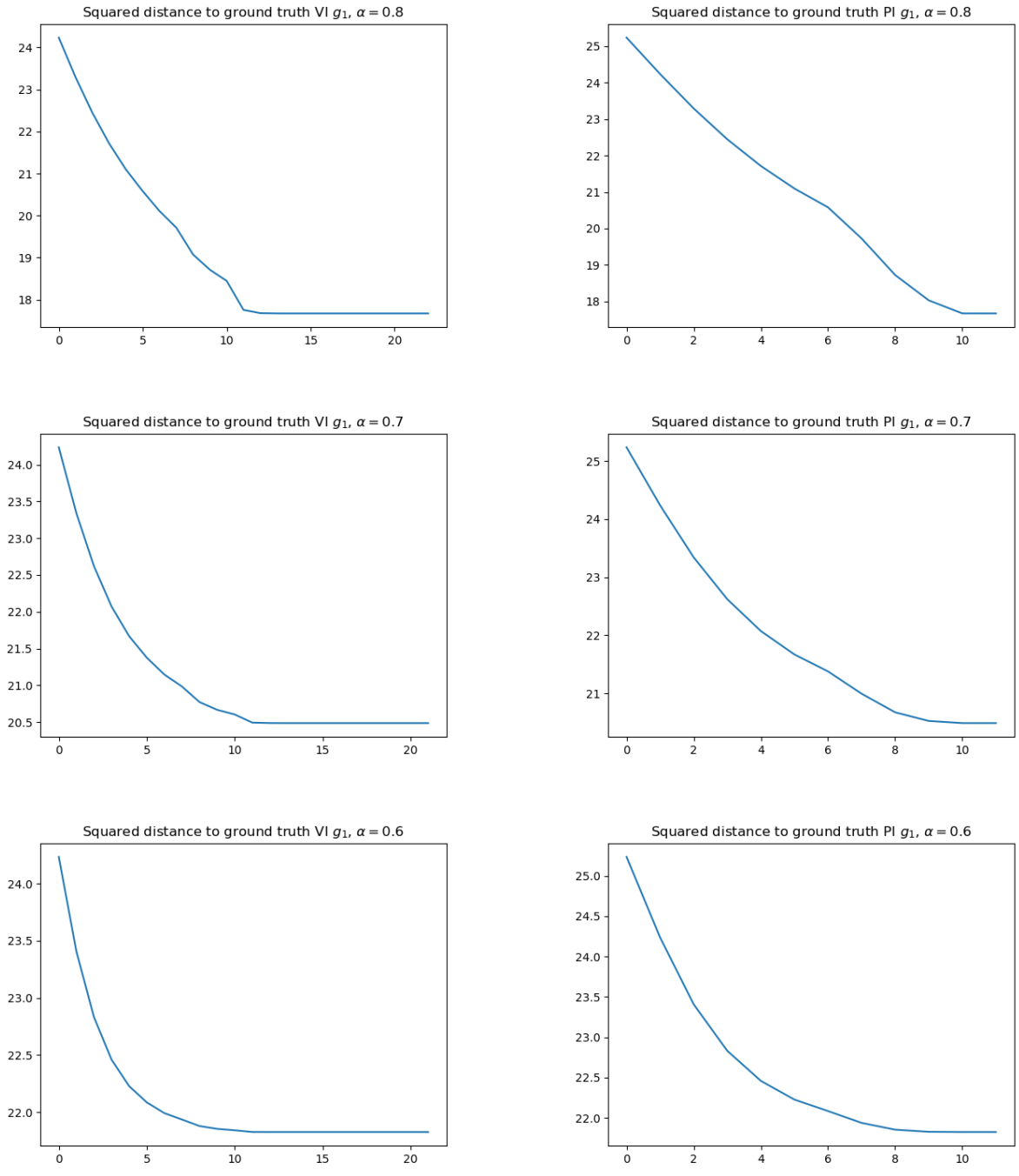

Figure 5: Error plots for g_1 and chosen discount factors

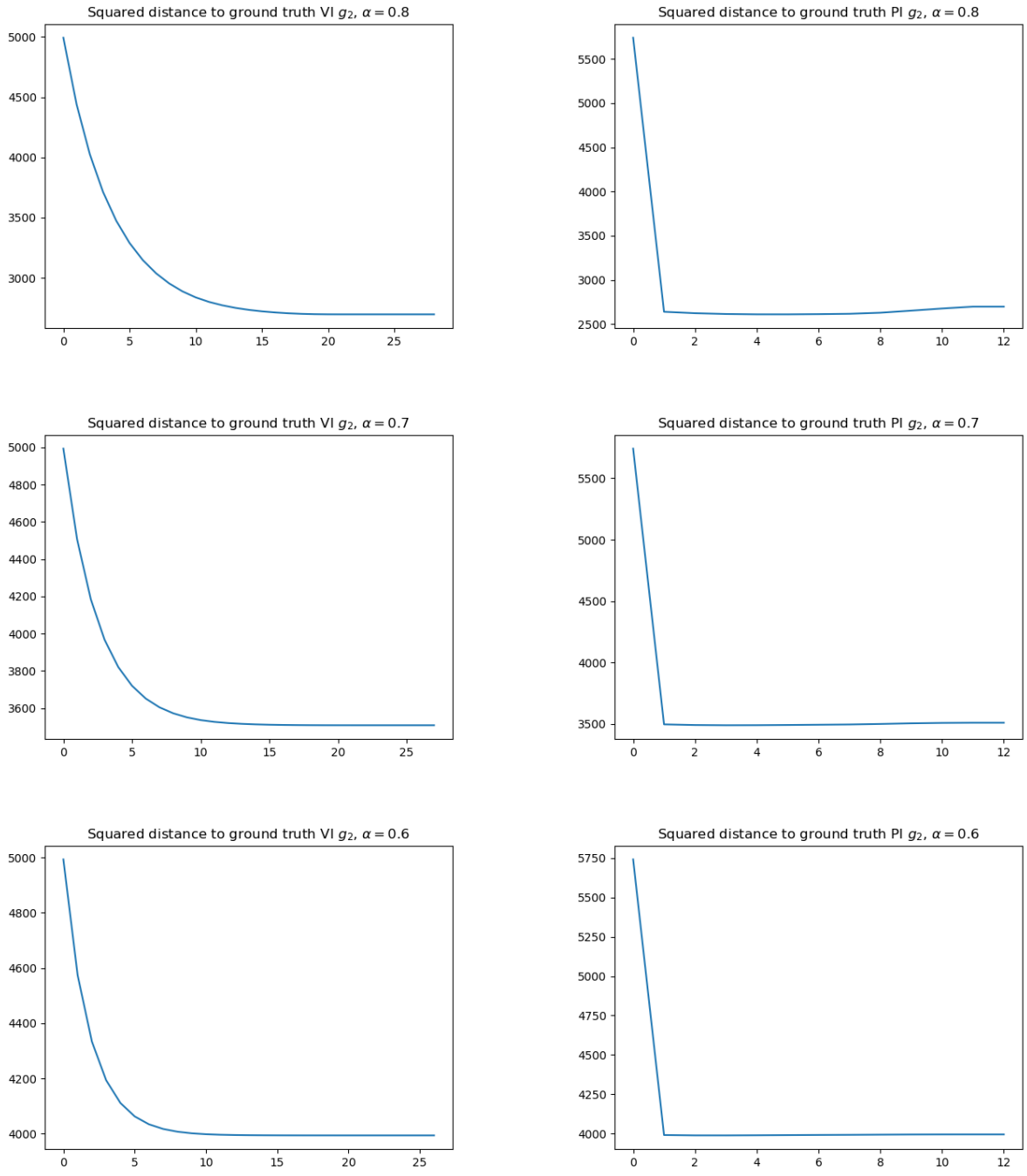

Figure 6: Error plots for g_2 and chosen discount factors

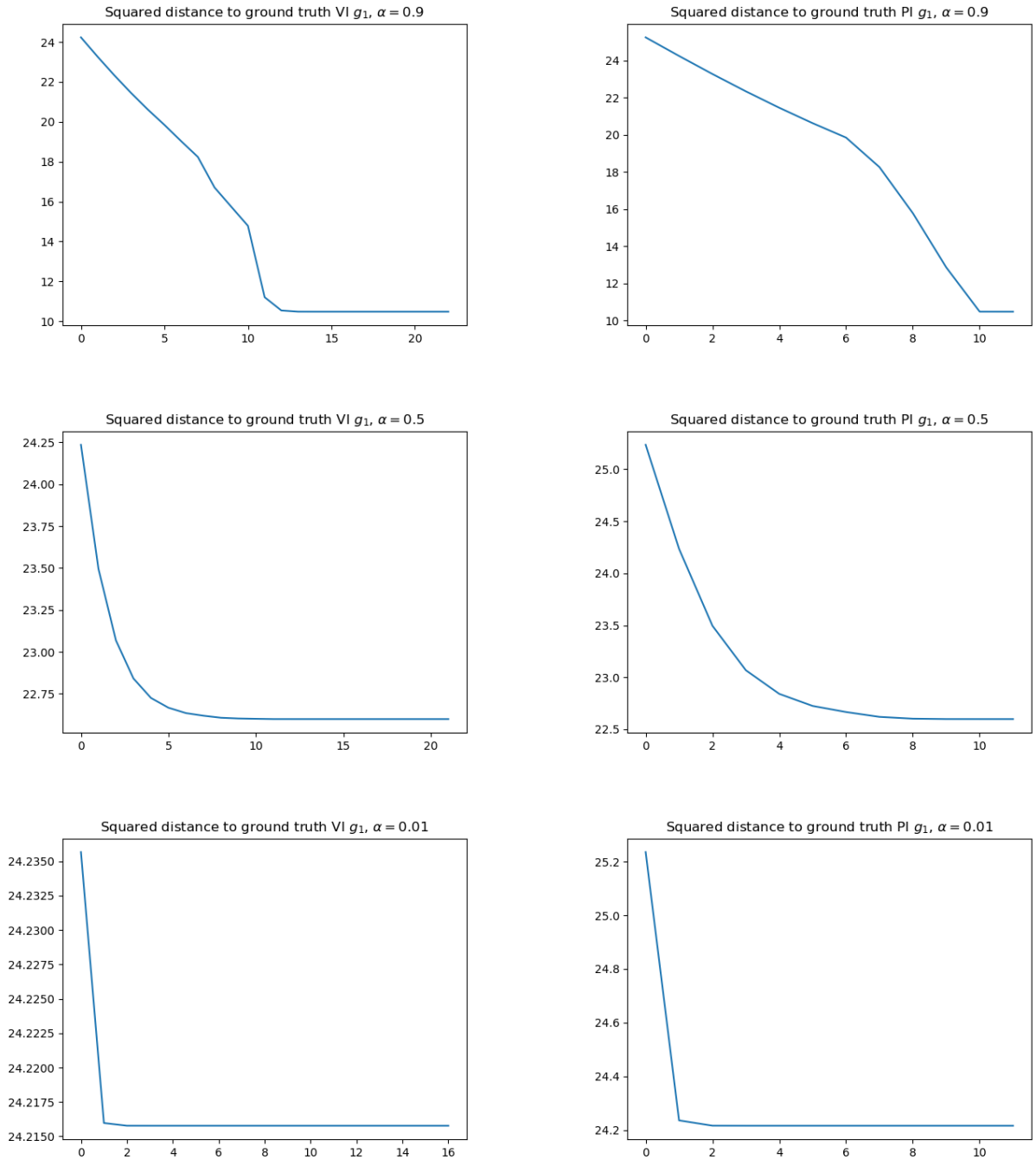

Figure 7: Error plots for g_1 and given discount factors

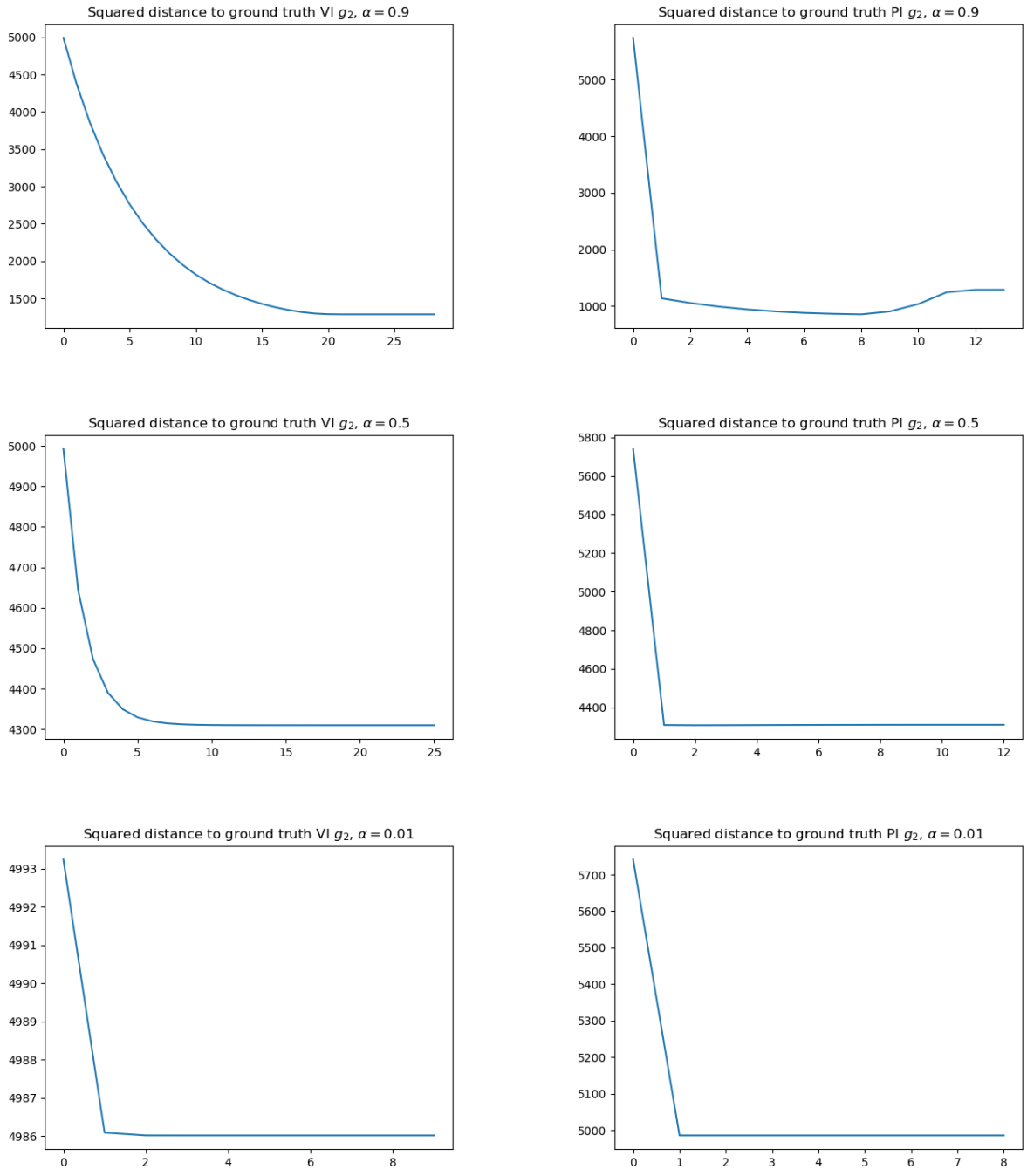

Figure 8: Error plots for g_2 and given discount factors

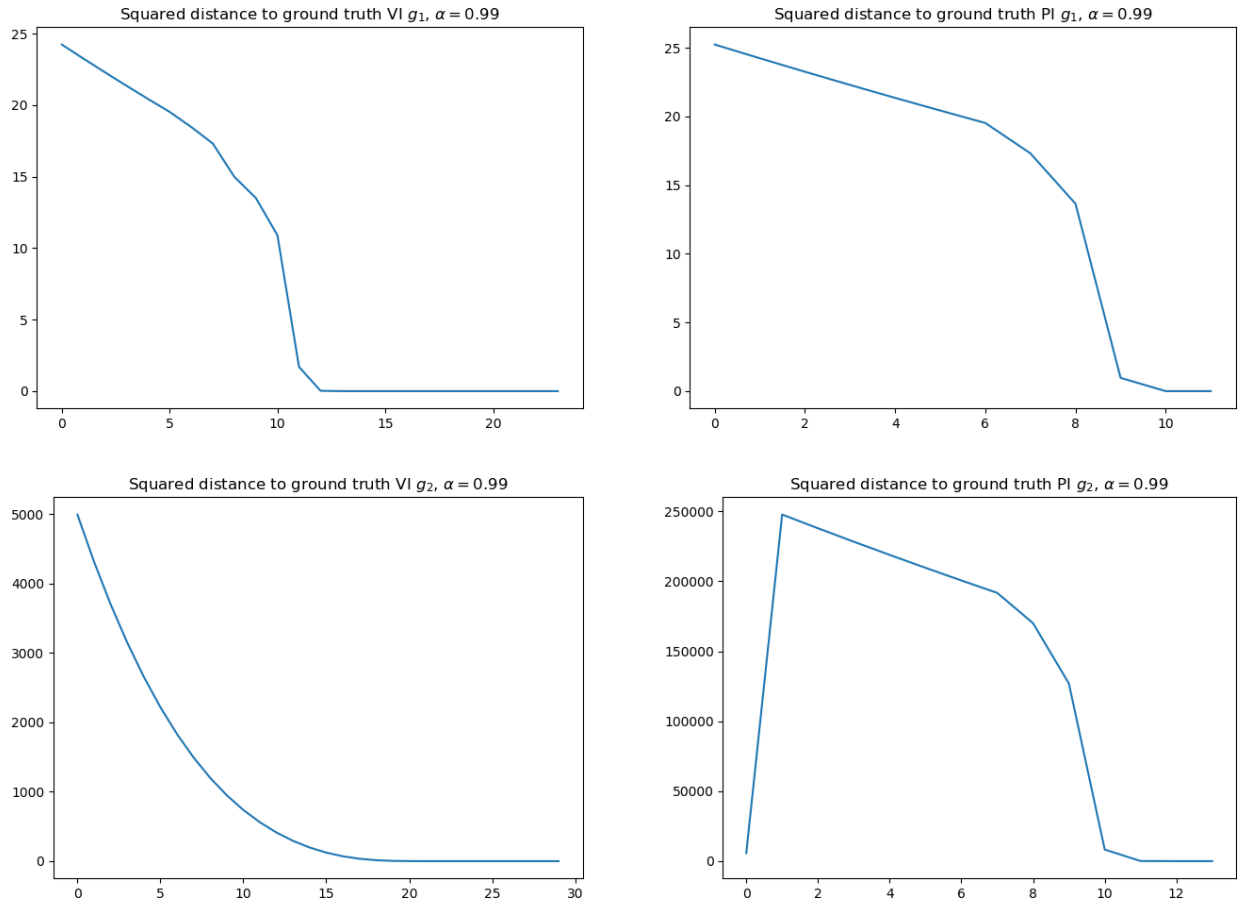

Figure 9: Error plots for given costs and discount factor $\alpha = 0.99$