Partially Observable Markov Decision Processes

General Form of a POMDP solution

We now summarize and generalize the ideas presented in the above example. The lines in the two state value functions become hyper-planes in the general case. Since we can represent a hyper-plane with a vector of its coefficients, our value function can (and will) be represented with a set of vectors.

To define some notation, let `V'` be the set of vectors we desire for the next stage of value iteration and `V` be the set of vectors we have from the previous stage. Here the stages of value iteration simply correspond to succeeding horizon lengths.

As we saw in the example, each vector in `V'` is constructed from a combination of vectors in `V`. More precisely each vector is constructed from the immediate rewards and the transformation (using the `S()` functions) of `V`.

A vector in `V` has a particular strategy associated with it. Each vector at a stage represents the value of acting according to the particular current and future strategy for that vector. Selecting a vector at a stage is the same as selecting a particular course of action at a stage and a particular future action strategy. The future strategy is contingent upon the particular observations that are received, but for a particular observation sequence the future strategy is fixed.

When we are given the set of vectors, it is easy to decide how to act for a given belief state; just select the action associated with the vector that gives the highest value at that point. Our problem is to construct those vectors, `V'`, given that we know the best ways to act in the previous stage, which is what `V` tells us.

Each action we perform has some immediate and some future consequences. The immediate consequences are directly available from the `POMDP` problem formulation. The future consequences are completely contained in the set `V`. However, things change when we perform an action, so all the possible future consequences must be weighted by the probability that we will incur those consequences. These probabilities come directly from the `POMDP` formulation and, more specifically, depend upon our current belief, the immediate action we take and the observation that the world provides us.

Again without going into the formulas, suffice it to say that we can take the set `V` and transform it into a set `S(a,z)` for each action, `a`, and observation, `z`. Think of each of these sets as the effect of choosing the the various courses of action for the vectors in `V` after having performed action `a` and having seen observation `z`. Constructing each of the sets `S(a,z)` is a simple transformation using the set `V` and the `POMDP` model parameters.

Once transformed into S(a,z) it is easy to see which vector in V has the best future course of action, given that our immediate action was a and immediate observation was z. In fact, the vectors in S(a,z) also form a piecewise linear convex value function over belief space. Note that if we look at the partition of S(a,z) and the partition of V they are not usually the same.

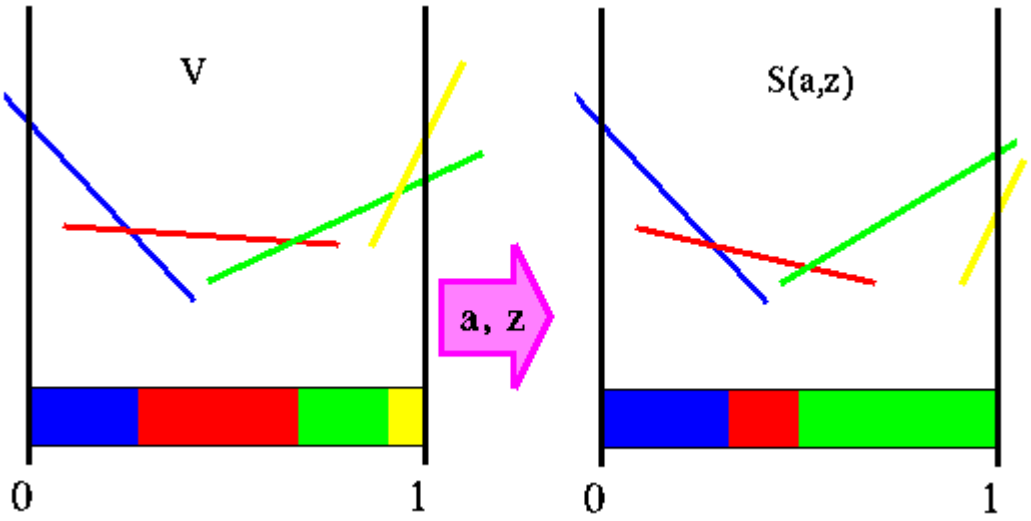
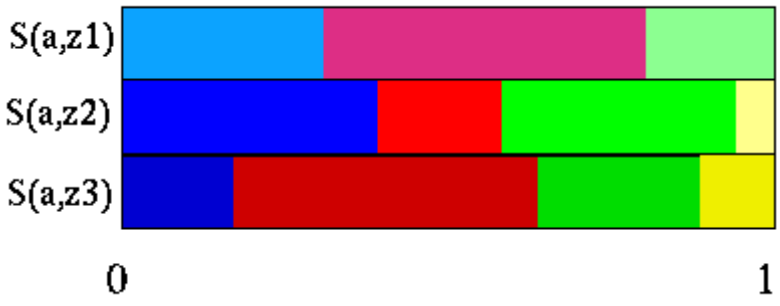


Figure 4: Transformed V for a and z

Note that a point in the function V is how good it is to follow that strategy if our belief was this point, whereas a point in the transformed value function is how good it is to follow that strategy, given we started at that point, performed action a and saw observation z.

We now come back to our main problem: how to construct the vectors in V' from the set V. One key point to many of the solution procedures is that given a particular belief point, we can easily construct the vector in V' for that belief point. This may not be completely obvious without any formulas, but it actually easy to show how with our partition view of the sets of vectors.

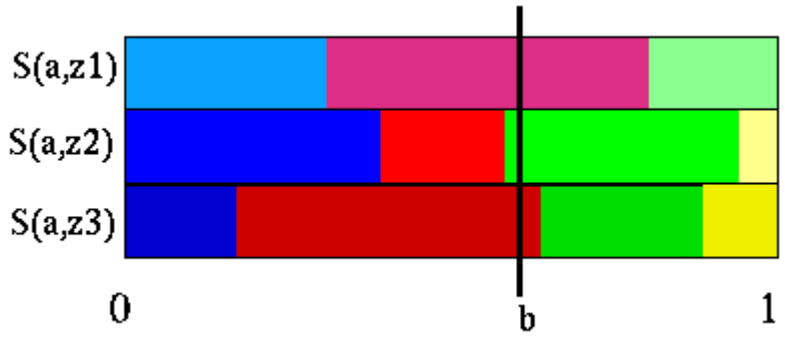
Below is a picture of the S(a,z) partitions of belief space for a particular action, with each observation, one on top of the other.



Adjacent belief partitions for transformed value function

For each action we can perform, we could see any of the possible observations after doing this. For a given action, we need to look at the immediate effects of the action and select a future course of action for each possible observation we can get. Ignoring the immediate effects for the moment, we actually have the information we need to select a course of action for each observation; the sets S(a,z).

For a given observation, the choice of action is obvious at a given belief point. The vector that dominates is the one to choose and this is already worked out for us in the partitioning diagram, because it shows where each vector dominates. For all the observations we do the same and it amounts to drawing a vertical line corresponding to a belief point and selecting all the vectors whose partition it intersects.



Point through the transformed value function's partitions construction

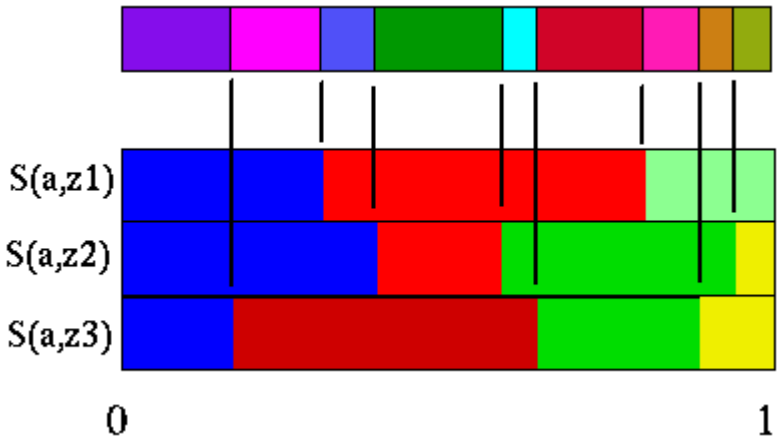
This selection process is giving us a kind of contingency strategy. It tells us which `V` vector to choose for each observation, which amounts to telling us which future course of action is best.

So for a particular action, given a belief point the vector we are interested in is comprised of the immediate effects of the action plus the sum of all of the chosen vector from each of the sets `S(a,z)`. We must also add the immediate reward effects for that particular action.

However, the vector constructed this way may or may not be in `V'`. We actually need to construct a vector at this point for each action and determine which one is the dominant one. The dominant one belongs to the set `V'` and its course of action is associated with it. The strategy that that vector represents is that immediate action, plus the future strategies from the `V` vectors chosen. Since we selected a different vector for each observation, the future strategy is contingent upon the observation we get.

If it were possible to enumerate all the belief points, we could construct all the vectors this way. Many belief points would produce duplicate vectors, but we could discard those and be left with the set we wanted. Since enumerating belief points is not possible, we must be smarter about generating these vectors. Since we know there are only a finite number of vectors, there exists a finite set of points that would allow us to generate each vector. The problem is that we don't immediately have access to this set.

Displaying the partitioning of the transformed `V` actually make it easy for us to pick out the vectors by eye. The figure below shows how just need to inspect the partitions to see where there is a different combination of vectors we could choose. We look at each observation and select a color, the number of different combination are the number of vectors that will be in the value function for this particular action `a`.



Making a new partition from S(a,z) partitions

This is easy when we have a simple two state POMDP and these nice partitioning pictures. To do this automatically for a POMDP with a lot of states is not so easy. We can visually inspect the partition diagrams and pick out the number of regions, and we know immediately how to combine the transformed V vectors to create the new vectors. However, in higher dimensional space it will be very hard to apply this same sort of visual inspection. All we really need is a single point in each of the regions, but knowing the regions or the set of points we need is not such as easy task.

Continue

Back | POMDP Tutorial | Next

Last modified: Fri Nov 7 15:29:42 CST 2003