

## Parametric Policy Iteration

- This algorithm is basically the same as the infinite dimensional version of the policy iteration algorithm we just saw, except that we approximately solve each policy valuation step by approximating the solution  $V_\alpha$  as a linear combination of  $k$  basis functions  $(\rho_1, \dots, \rho_k)$ .

Suppose we set  $V_\alpha(s) = \sum_{i=1}^k \theta_i \rho_i(s)$ . Then the equation for  $V_\alpha$

$$V_\alpha(s) = u(s, \alpha(s)) + \beta \int V_\alpha(s') p(s'|s, \alpha(s)) ds' \quad (1)$$

is transformed into a linear equation with  $k$  unknown parameters  $\theta = (\theta_1, \dots, \theta_k)$ :

$$\sum_{i=1}^k \theta_i \rho_i(s) = u(s, \alpha(s)) + \beta \int \sum_{i=1}^k \theta_i \rho_i(s') p(s'|s, \alpha(s)) ds' \quad (2)$$

- Suppose we evaluate the above equation at a set of  $N$  points in  $S$ , with  $N \geq k$ . Then define the  $N$  by  $k$  matrices  $P$  and  $EP$  with elements  $P_{j,k}$  and  $EP_{j,k}$ , given by

$$P_{j,k} = \rho_k(s_j) \quad (3)$$

$$EP_{j,k} = \int \rho_k(s') p(s'|s_j, \alpha(s_j)). \quad (4)$$

Further define the  $N$  by 1 vector  $y$  with  $j_{th}$  element  $y_j$  given by

$$y_j = u(s_j, \alpha(s_j)), \quad (5)$$

and let the  $N$  by  $k$  matrix  $X$  be given by

$$X = (P - \beta EP). \quad (6)$$

Then the system of equations in (2) can be written in matrix form as

$$y = X\theta. \quad (7)$$

If  $N = k$  and  $X$  is invertible the solution for  $\theta$  is simply

$$\theta = y/X = X^{-1}y. \quad (8)$$

If  $N > k$  we have an overdetermined system and in general there is no  $\theta \in R^k$  that allows us to exactly solve  $y = X\theta$ . However, we can form an approximate solution using the ordinary least squares estimator (OLS). Meaning the value of  $\theta$  that minimizes the distance  $\|y - X\theta\|^2$  and is given by

$$\theta = (X'X)^{-1}X'y. \quad (9)$$

- The parametric policy iteration algorithm is the same as the infinite dimensional policy iteration algorithm above, except that the policy valuation step is solved by approximating  $V_\alpha$  as a linear combination of basis functions using the OLS estimator given above.
- In general we will not be able to exactly integrate the basis functions and must use a quadrature rule to approximate the elements in  $EP$ . Therefore the PPI algorithm requires the following choices:
  1. The quadrature rule for computing the elements of  $EP$ .
  2. The sample points  $(s_1, \dots, s_N)$  at which  $P$  and  $EP$  are evaluated at.
  3. The set of basis functions  $(\rho_1, \dots, \rho_k)$ .
- Note that the policy improvement step would only be done on the same  $N$  points  $(s_1, \dots, s_N)$ . Thus only  $NK$  numerical integration and  $N$  maximizations are required for each policy valuation and policy improvement step, so if  $N$  and  $k$  can be chosen to be small, it is possible to find an approximation solution to the DP problem with amazingly few computations.
- This is true provided the basis functions  $(\rho_1(s), \dots, \rho_k(s))$  are sufficiently easy to evaluate at each  $s \in S$ . The resulting solution is defined by a parameter vector  $\theta^*$  that enables us to evaluate  $V(s) = \sum_{k=1} \theta_i^* \rho_i(s)$  very rapidly at any  $s \in S$ .
- Evaluating the corresponding decision  $\alpha(s)$  at that point would require an approximate solution to

$$\alpha(s) = \max_{a \in A(s)} \left[ u(s, a) + \beta \int \sum_{i=1}^k \theta_i^* \rho_i(s') p(s'|s, a) ds' \right]. \quad (10)$$

- In many cases this can be done quite rapidly, or alternatively, using the values  $[\alpha(s_j)]$ , for  $j = 1, \dots, N$  from the last step of policy iteration, it might be possible to interpolate values  $\alpha(s)$  for  $s \in (s_1, \dots, s_N)$  if the decision rule is sufficiently smooth.