Model-based approaches for continuous state

estimating a model & then using to compute a V or policy
often considered more sample efficient than model free estimates
likely to be true for model free algorithms that only use data
tuples (s, a, r, s') once like Q-learning
but when use experience replay or multiple iterations through
the data, less clear

as in value function approximation, many ways to approximate a
model

∘ Approximate Linear Models & Connection to LSPI

assume set of indep features for representing transition & reward models

$\Phi(s) = [\phi_1(s) \ldots \phi_K(s)]^T$ k features

1^{st} consider uncontrolled / fixed policy case — for a fixed π

$P(s'|s,a) \rightarrow P(s'|s)$ shorthand because only one action per s

define
$$\Phi(s'|s) \overset{s' \sim P(s'|s)}{=} [\phi_1(s') \ldots \phi_K(s')]$$

want to compute a $k \times k$ matrix P_Φ (like w for value func approx) that

1) predicts expected next feature vectors — $k \times 1$
$$P_\Phi^T \Phi(s) \approx E_{s' \sim P(s'|s)} \{\Phi(s'|s)\}$$

2) minimizes the expected feature-prediction error
$$P_\Phi = \arg\min_{P_K} \sum_s \| P_K^T \phi(s) - E\{\Phi(s'|s)\} \|_2^2$$

solving the optimization in (2)

compute the expected next state directly as $P\Phi$ ← true transition model

$P\Phi$ is a $n \times k$ matrix

i^{th} row is expected value of features on next step after starting in s_i

Φ is a $n \times k$ matrix of feature values for each state

i^{th} row of ΦP_Φ is P_Φ's predic of next feature values for state i

$\Phi P_\Phi \approx P\Phi$

least squares soln : $\Phi^T P_\Phi = \Phi^T P \Phi$
$$P_\Phi = (\Phi^T\Phi)^{-1} \Phi^T P \Phi$$

yields $\approx$ next feature vec $\widehat{P\Phi} = \Phi r_\phi$

predict reward using same features, project (using least squares)
reward into space of features

$r_\phi = (\Phi^T\Phi)^{-1} \Phi^T R$

$\hat{R} = \Phi r_\phi$

now prove linear fixed point soln for Φ (resulting ω) identical to soln for value get using P_Φ & r_Φ

- uncontrolled setting
  let x be any k-length vector (represents a state)
  Bellman eqn
  $$V[x] = r_\Phi^T x + \gamma V[P_\Phi^T x]$$
  $$= \sum_{i=0}^{\infty} \gamma^i r_\Phi^T (P_\Phi^i)^T x$$
  can rewrite in terms of original state space as
  $$V = \underbrace{\Phi \sum_{i=0}^{\infty} \gamma^i P_\Phi^i r_\Phi}_{\text{some linear combo of } \Phi\text{'s columns}}$$
  implies can express $V = \Phi \omega'$ for some vector ω
  $$V = \hat{R} + \gamma \widehat{P \Phi} \omega'$$
  $$\Phi \omega' = \hat{R} + \gamma \widehat{P\Phi} \omega'$$
  $$\Phi \omega' = \Phi r_\Phi + \gamma \Phi P_\Phi \omega \qquad \text{sub in expr for } \widehat{P\Phi} \text{ \& } \hat{R}$$
  $$\omega' = \underbrace{(I - \gamma P_\Phi)^{-1} r_\Phi}$$
  ↖ identity matrix
  well defined if P_Φ has a spectral radius $< 1/\gamma$
  (if $> 1/\gamma$ implies value of some states is unbounded)

thm: for any Markov reward process (MRP) ← define and a set of features Φ the linear model soln & linear fixed point soln are identical

proof:
  soln for linear model is
  $$\omega' = (I - \gamma P_\Phi)^{-1} r_\Phi$$
  $$= (I - \gamma (\Phi^T \Phi)^{-1} \Phi^T P \Phi)^{-1} (\Phi^T \Phi)^{-1} \Phi^T R$$
  $$= \omega_\Phi \text{ for fixed point}$$ □

⇒ for a given set of features Φ, get exact same soln if compute approx linear model & then use to compute value function, approx linear fixed point value func as directly computing approx linear fixed point value func

- controlled case: LSPI
  $Q^\pi(s,a)$. define basis functions over $s \times a$ space
  $$\hat{Q} = \sum_{i=1}^{k} w_i \phi_i(s,a)$$
  policy eval: use LSTDQ to compute $\hat{Q}$ of a fixed π
  a MDP with a fixed π is equal to a MRP whose state space is $S \times A$
  LSTDQ = LSTD executed on this equiv MRP
  by thm above, LSTDQ soln = approx linear models w/smallest L_2 error