Reinforcement Learning and Optimal Control

ASU, CSE 691, Winter 2019

Dimitri P. Bertsekas
dimitrib@mit.edu

Lecture 12

Outline

1. Review of Aggregation Frameworks

2. Definition of the Aggregate Problem

3. Simulation-Based Solution of the Aggregate Problem

4. Variants of Aggregation

Aggregation within the Approximation in Value Space Framework

Approximate minimization

First Step "Future"

$$\min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha \tilde{J}(j)\big)$$

Approximations:

Replace $E\{\cdot\}$ with nominal values (certainty equivalence)

Adaptive simulation

Monte Carlo tree search

Computation of $\tilde{J}$:

Problem approximation

Rollout

Approximate PI

Parametric approximation

Aggregation

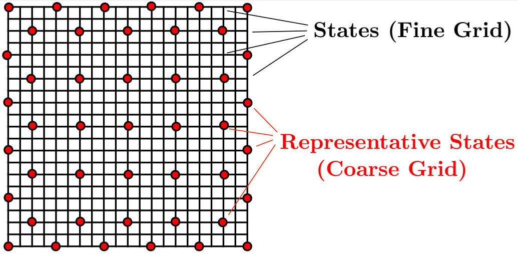
ONE-STEP LOOKAHEAD

MULTISTEP LOOKAHEAD IS SIMILAR - WE WILL DISCUSS LATER

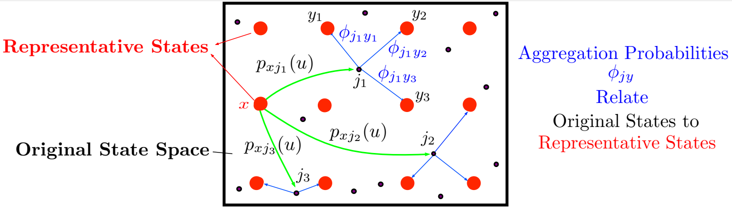
Some important differences from alternative schemes:

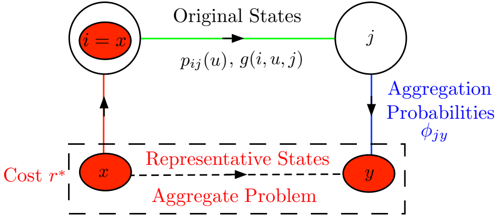
- In aggregation, $\tilde{J}$ aims to approximate J^*, not the cost function J_μ of a policy μ, like rollout or approximate PI.
- $\tilde{J}$ converges to J^* as the aggregation becomes finer, i.e., as the number of representative states or features increases.
- Key factor for good performance: Choose properly the rep. features so that the number needed for good performance is not excessive.

A classical example.



Original states are related to representative states with interpolation coefficients called aggregation probabilities.





Original cost approximation by interpolation

$$\hat{p}_{xy}(u) = \sum_{j=1}^{n} p_{xj}(u)\phi_{jy}, \quad \hat{g}(x,u) = \sum_{j=1}^{n} p_{xj}(u) g(x,u,j), \qquad \tilde{J}(j) = \sum_{y \in \mathcal{A}} \phi_{jy} r_y^*$$

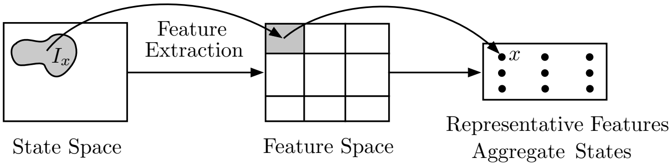
Exact methods

Once the aggregate model is computed (i.e., its transition probs. and cost per stage), any exact DP method can be used: VI, PI, optimistic PI, or linear programming.

Model-free (simulation-based) methods

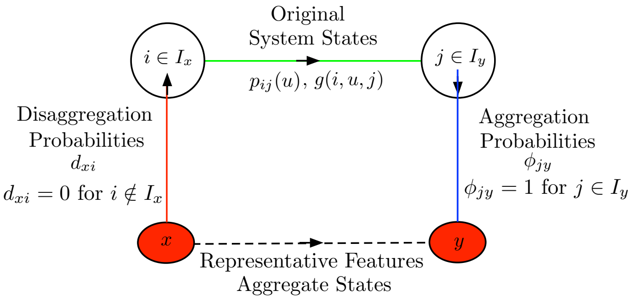
Given a simulator for the original problem, we can obtain a simulator for the aggregate problem. Then use an (exact) model-free method to solve the aggregate problem.

Feature-Based Aggregation - Discretize the Feature Space

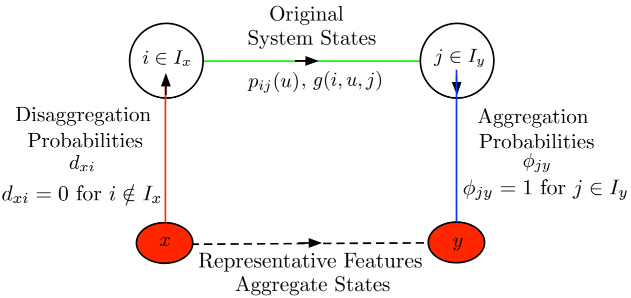


Representative features formation - Guiding ideas:

- Feature map F: States i with similar $F(i)$ should have similar $J^*(i)$.
- Footprint I_x of feature x: States i in I_x should have feature $F(i) \approx x$.



A Simple but Flawed Version of the Aggregate Problem



Patterned after the simpler representative states model

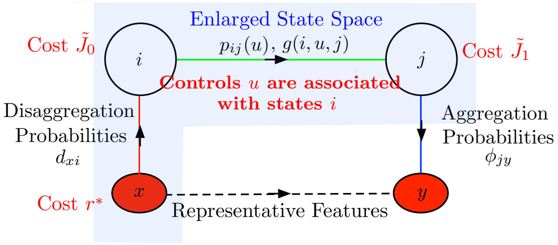
Aggregate dynamics and costs

- Aggregate dynamics: Transition probabilities between representative features x, y

$$\hat{p}_{xy}(u) = \sum_{i \in I_x} d_{xi} \sum_{j=1}^{n} p_{ij}(u)\phi_{jy}$$

- Expected cost per stage:

$$\hat{g}(x,u) = \sum_{i \in I_x} d_{xi} \sum_{j=1}^{n} p_{xj}(u) g(x,u,j)$$



Bellman equations for the enlarged problem

$$r_x^* = \sum_{i=1}^{n} d_{xi}\tilde{J}_0(i), \qquad x \in \mathcal{A},$$

$$\tilde{J}_0(i) = \min_{u\in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha\tilde{J}_1(j)\big), \qquad i = 1,\ldots,n,$$

$$\tilde{J}_1(j) = \sum_{y\in\mathcal{A}} \phi_{jy} r_y^*, \qquad j = 1,\ldots,n$$

r^* solves uniquely the composite Bellman equation $r^* = Hr^*$:

$$r_x^* = (Hr^*)(x) = \sum_{i=1}^{n} d_{xi} \min_{u\in U(i)} \sum_{j=1}^{n} p_{ij}(u)\left(g(i,u,j) + \alpha \sum_{y\in\mathcal{A}} \phi_{jy}\, r_y^*\right), \qquad x \in \mathcal{A}$$

Error Bound

Approximation error for the piecewise constant case ($\phi_{jy} = 0$ or 1 for all j, y)

Consider the footprint sets

$$S_y = \{j \mid \phi_{jy} = 1\}, \qquad y \in \mathcal{A}$$

The $(J^* - \tilde{J})$ error is small if J^* varies little within each S_y. In particular,

$$\left| J^*(j) - r_y^* \right| \leq \frac{\epsilon}{1-\alpha}, \qquad j \in S_y, \ y \in \mathcal{A},$$

where $\epsilon = \max_{y \in \mathcal{A}} \max_{i,j \in S_y} \left| J^*(i) - J^*(j) \right|$ is the max variation of J^* within S_y.

Implication

Choose representative features x so that J^* varies little over the footprint of x.

This is a generally valid qualitative guideline

Holds for the more general piecewise linear interpolation case.

Simulation-Based Asynchronous Value Iteration for the Aggregate Problem

A sampled version of VI for solving $r^* = Hr^*$: $r^{k+1} \approx (1-\gamma^k)r^k + \gamma^k H(r^k)$ with

$$(Hr)(x) = \sum_{i=1}^{n} d_{xi} \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u) \left(g(i,u,j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy}\, r_y \right), \qquad x \in \mathcal{A}$$

Note that H is a contraction.

At time k iterate for a single rep. feature x_k, and keep all other r_x^k unchanged:

$$r_{x_k}^{k+1} = (1-\gamma^k) r_{x_k}^k + \gamma^k \min_{u \in U(i)} \sum_{j=1}^{n} p_{i_k j}(u) \left(g(i_k,u,j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy} r_y^k \right)$$

where i_k is a sample from I_{x_k} selected according to $d_{x_k i}$, and γ^k is a stepsize.

Convergence result [Tsitsiklis and Van Roy (1995)]

With $\gamma^k \to 0$ and other technical conditions, this iteration converges to the unique solution r^*. Some similarity with (exact) Q-learning proofs.

Simulation-Based Policy Iteration

Uses policy evaluation/policy improvement to generate policy/cost pairs $\{(\mu^k, r^k)\}$.
Converges monotonically ($r^{k+1} \leq r^k$) and finitely ($r^k = r^*$ for sufficiently large k).

Policy evaluation of current policy μ^k

Solve the (linear) composite Bellman equation $r^k = H_{\mu^k} r^k$ for μ^k, where

$$(H_{\mu^k} r)(x) = \sum_{i=1}^{n} d_{xi} \sum_{j=1}^{n} p_{ij}\big(\mu^k(i)\big) \left(g(i, \mu^k(i), j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy}\, r_y \right), \qquad x \in \mathcal{A}$$

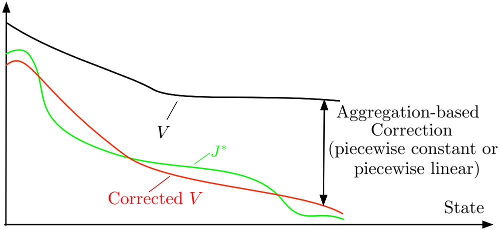
Two possibilities:

- Iteratively: Using a sampled version of VI with sampling for both i and for j.
- By matrix inversion: Write the equation $r^k = H_{\mu^k} r^k$ in matrix form as $r^k = A^k r^k + b^k$. Evaluate A^k and b^k by simulation, and set $r^k = (I - A^k)^{-1} b^k$.

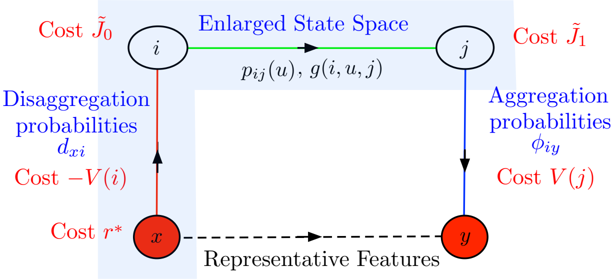
Policy improvement by one-step lookahead

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u) \left(g(i, u, j) + \alpha \sum_{y \in \mathcal{A}} \phi_{jy} r_y^k \right), \qquad i = 1, \ldots, n$$

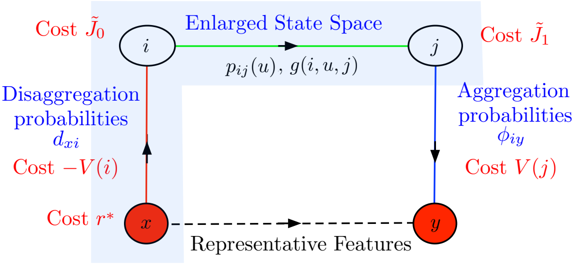
Biased Aggregation - Suppose we Know a Good Approximation $V \approx J^*$; How do we Correct it?



We add a "bias" function V to the cost structure of the enlarged aggregate problem

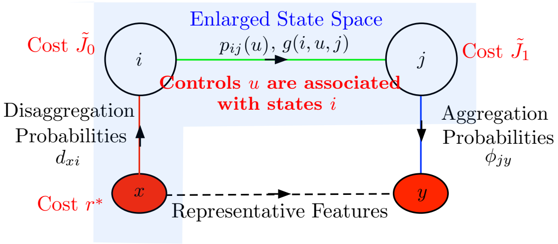


Some Results for Biased Aggregation



Let $(r^*, \tilde{J}_0, \tilde{J}_1)$ be the solution [note that $\tilde{J}_1(j) = V(j) + \sum_{y\in\mathcal{A}} \phi_{jy} r^*_y$]

- When $V = J^*$ then $r^* = 0$, $\tilde{J}_0 = \tilde{J}_1 = J^*$, and any optimal policy for the aggregate problem is optimal for the original problem.
- When $V = J_\mu$ for some policy μ, the policy produced by aggregation is a rollout policy based on μ, when there is a single rep. feature. Suggests that with multiple rep. features the aggregation/rollout policy should be much better than rollout.
- Error bounds similar to the ones for the case $V = 0$ suggest to choose rep. features and footprint sets within which $V - J^*$ varies little.
- We do not know J^*, but we may use $T^k V$ (k value iterations on V) as an approximation. Then use $V - T^k V$ as a scoring function to form rep. features.



How do VI and PI benefit from the problem being deterministic?

- VI form: $r_{x_k}^{k+1} = (1-\gamma^k) r_{x_k}^k + \gamma^k \min_{u\in U(i)} \sum_{j=1}^n p_{i_k j}(u) \Big(g(i_k,u,j) + \alpha \sum_{y\in\mathcal{A}} \phi_{jy} r_y^k \Big)$
- Policy evaluation: Solve the composite Bellman equation $r^k = H_{\mu^k} r^k$, where

$$(H_{\mu^k} r)(x) = \sum_{i=1}^n d_{xi} \sum_{j=1}^n p_{ij}\big(\mu^k(i)\big) \left(g\big(i,\mu^k(i),j\big) + \alpha \sum_{y\in\mathcal{A}} \phi_{jy}\, r_y \right), \qquad x \in \mathcal{A}$$

- Policy improvement: $\mu^{k+1}(i) = \arg\min_{u\in U(i)} \sum_{j=1}^n p_{ij}(u) \Big(g(i,u,j) + \alpha \sum_{y\in\mathcal{A}} \phi_{jy} r_y^k \Big)$
- How about using representative states? Possibility of multistep lookahead?

Deterministic Problems - Aggregation with ℓ-Step Lookahead

For a deterministic problem, the simulation-based VI and PI are simplified

- The sampled version of VI has the form

$$r_{x_k}^{k+1} = (1-\gamma^k) r_{x_k}^k + \gamma^k \min_{u \in U(i)} \left(g(i_k, u) + \alpha \sum_{y \in \mathcal{A}} \phi_{f(i_k,u)y} r_y^k \right)$$

- No expectation over j is required.
- If representative states are used, there is no need for sampling according to the probabilities $d_{x_k i}$ to obtain i_k (so $\gamma^k \equiv 1$).

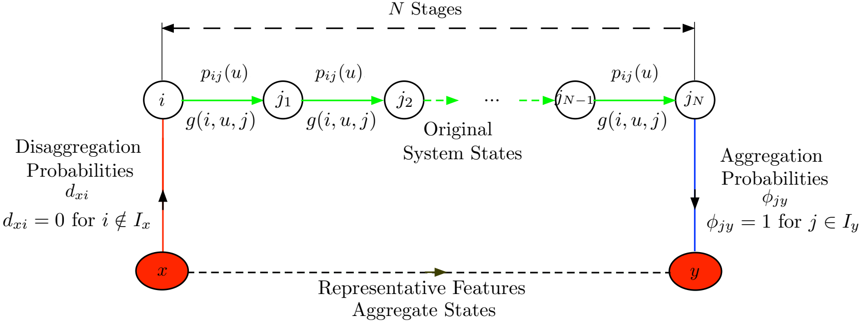
Given r^*, consider ℓ-step lookahead minimization

- At state i_0 we find

$$(u_0^*, \ldots, u_{N-1}^*) \in \arg \min_{(u_0,\ldots,u_{\ell-1})} \left(\sum_{k=0}^{\ell-1} \alpha^k g(i_k, u_k) + \alpha^\ell \sum_{y \in \mathcal{A}} \phi_{i_\ell y} r_y^* \right)$$

  and apply $\tilde{\mu}(i_0) = u_0^*$.
- This is a shortest path problem, and its solution on-line may be fast.

N-Step Feature-Based Aggregation



- The composite system consists of $N+2$ stochastic Bellman equations.
- Simulation-based version of VI is hard to implement.
- Simulation-based version of PI is possible, but policies are multistep.

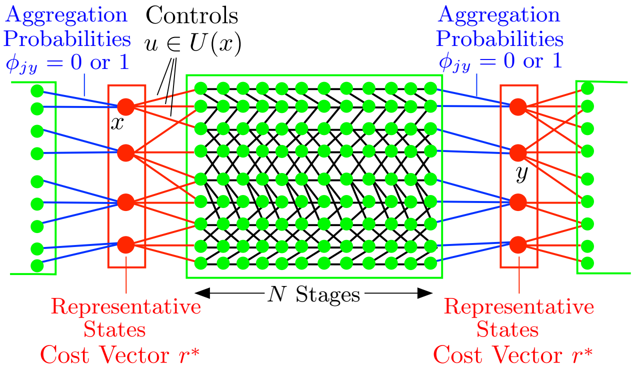
A simpler case: Deterministic problem and representative states (no features)

- Then each VI iteration involves solution of an N-stage deterministic DP (shortest path) problem: $r^{k+1} = H_N(r^k)$, where H_N is the N-stage DP operator.
- This algorithm embodies the idea of aggregation in both space and time.

Plan 5-day auto travel from Boston to San Francisco - How would you do it?

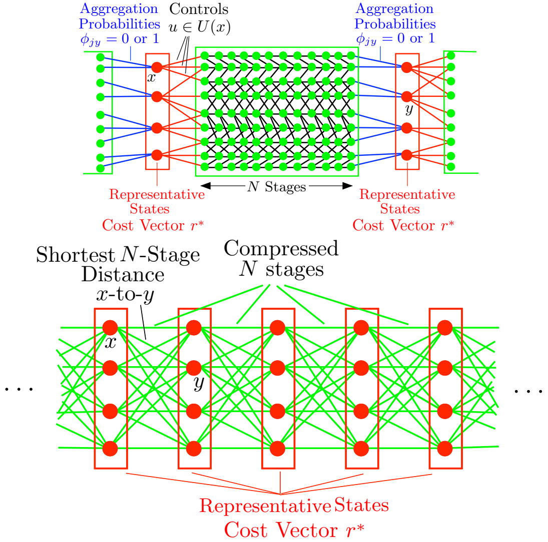
- Select major stops/cities (New York, Chicago, Salt Lake City, Phoenix, etc).
- Select major stopping times (times to stop for sleep, rest, etc).
- Decide on space and time schedules at a coarse level. Optimize the details later.
- We may view this as an example of reduction of a very large-scale shortest path problem to a manageable problem by spacio-temporal aggregation.

Deterministic Problems - N-Stage Aggregation with Representative States and Aggregation Probabilities $\phi_{jy} = 0$ or 1

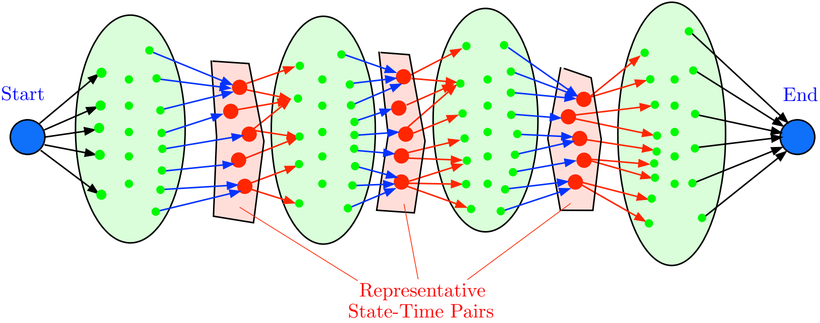


An example of spacio-temporal aggregation

- The infinite horizon discounted aggregate problem decomposes into a sequence of (identical) N-stage shortest path problems.
- Compute shortest path from each rep. state x to each rep. state y.
- Construct a low-dimensional deterministic infinite horizon DP problem (the states are just the representative states).



- Each N-stages block is “compressed" into an all-to-all shortest path problem.
- The compressed problem is a low-dimensional deterministic DP problem.



Deterministic shortest path and finite horizon extensions

- Consider the space-time tube of a deterministic shortest path problem.
- Introduce space-time barriers, i.e., subsets of representative state-time pairs that “separate past from future" (think of the Boston-San Francisco travel).
- “Compress" the portions of the space-time tube between two successive barriers into shortest path problems between each state-time pair of the left barrier to each state-time pair of the right barrier.
- Form a “master" shortest path problem of low dimension that involves only the representative state-time pairs.

About the Next and Final Lecture

WE WILL GIVE AN OVERVIEW OF THE ENTIRE COURSE