Project Reminders

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Presentations in class, Dec 5 and 7. Maximum of 10 minutes for your presentation. Make sure you practice!

Grade based on class survey which will count for **1/3 of participation component** (5% of final grade overall); Survey question:

“This presentation was clear, interesting, and informative: 1, 2, 3, 4, 5”.

Resources for presentations:

http://www.nature.com/scitable/ebooks/english-communication-for-scientists-14053993/giving-oral-presentations-14239332

http://www.pcworld.idg.com.au/slideshow/366369/world-worst-powerpoint-presentations/

Project Presentation Schedule

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Dec 5 Presentations

- Isaac, Adam, Shuzhe: Ventilation in Intensive Care Units
- Huiwen, Siyu: Treasure hunting
- Suyanpeng, Liyang: Pac Man
- Seok-Joo, Aditi, Ryan: Blackjack
- Andrew, Valerie, Anna: Fantasy Football Draft
- Derek, Chandra, Sajan: Robot Navigation

Dec 7 Presentations

- Weiyu, Sijia: Uber
- Junhong, Luze, Mohammad: Tennis
- Darshan, Ikka, Srinivas: Navigating conferences
- Zhanren, Palaniapan, Dhanush: IOE Master's Course Selection
- Chien-Yi, Yixuan, Zheng: Safety Stock Planning

Recap

Methods we have covered for solving infinite horizon MDPs include:

- Value Iteration
- Policy Iteration
- Modified Policy Iteration
- LP Formulation of an MDP

Last class: Partially observable MDPs

Monotone Policies

Policies with a simple structure are appealing

- Easier for decision makers to understand
- Easier to implement
- Easier to evaluate computationally

A common example is a **control limit policy**

$$a(s) = \begin{cases} a_1, if\ s < s^* \\ a_2, if\ s \geq s^* \end{cases}$$

where a_1 and a_2 are alternative actions and s^* is a control limit.

Question: What conditions guarantee the existence of a control limit policy for an infinite horizon problem?

Monotonicity: Infinite Horizon MDPs

The following theorem provides a means to prove existence of a monotone policy for infinite horizon MDPs

Theorem (≈6.11.1 Puterman) Suppose the following hold:

1. $v \in V^{\sigma}$ implies $\mathrm{L}v \in V^{\sigma}$

2. $v \in V^{\sigma}$ implies there exists a $\mathrm{d}' \in D^{\sigma} \cap argmax_{d \in D} L_d v$.

3. For any convergent sequence $\{\mathrm{v}^{\mathrm{n}}\} \subset V^{\sigma}$, $\lim_{n \to \infty} v^n \in V^{\sigma}$.

Then there exists an optimal stationary policy in Π^{σ}

Proof: See Puterman pp 255-256

Monotonicity: Infinite Horizon MDPs

In English Theorem 6.11.1 says:

> If the value function and decision rule at each iteration of the value iteration algorithm have a special property (e.g. monotonicity) and that property is preserved through applications of operator L then there is an optimal policy with the property.

Monotonicity: Infinite Horizon MDPs

Theorem 6.11.1 is a powerful result that can be used to prove a number of properties of infinite horizon MDPs including the following that is analogous to Theorem 4.7.4 in the finite horizon context

Theorem (6.11.6 Puterman) Suppose for $t = 1, \ldots, N-1$

1. $r(s,a)$ is nondecreasing in s for all $a \in A$.
2. $q(k|s,a)$ is nondecreasing in s for all $k \in S, a \in A$.
3. $r(s,a)$ is superadditive (subadditive) on $S \times A$.
4. $q(k|s,a)$ is superadditive (subadditive) on $S \times A$, $\forall k$

Then there exist optimal stationary decision rules, $d^*(s)$, which are nondecreasing (nonincreasing) in s.

New Topic: Limits of DP

Realistic applications often lead to DPs that cannot be solved exactly.

Often the state space becomes extremely large resulting in at least two potential problems for using MDPs:

- Difficulty in computing the value function
  - "The curse of dimensionality"
- Difficulty in parameter estimation
  - "The curse of modeling"

Example

You are developing a decision support system to manage inventory for a retail outlet selling 100 different types of shoes. Weekly you must decide whether and how much to order of each shoe.

Example

You are a transportation engineer developing an intelligent traffic control system for downtown Ann Arbor.

All of these problems have in common that the state space becomes extremely large.

Approximate Dynamic Programming

Many MDPs cannot be solved exactly due to the curse of dimensionality. However, many options exist to find “good” solutions:

- Heuristics
- Approximate dynamic programming methods
  - Deterministic
  - Sampling (Simulation) based

The ideal approach is problem dependent.

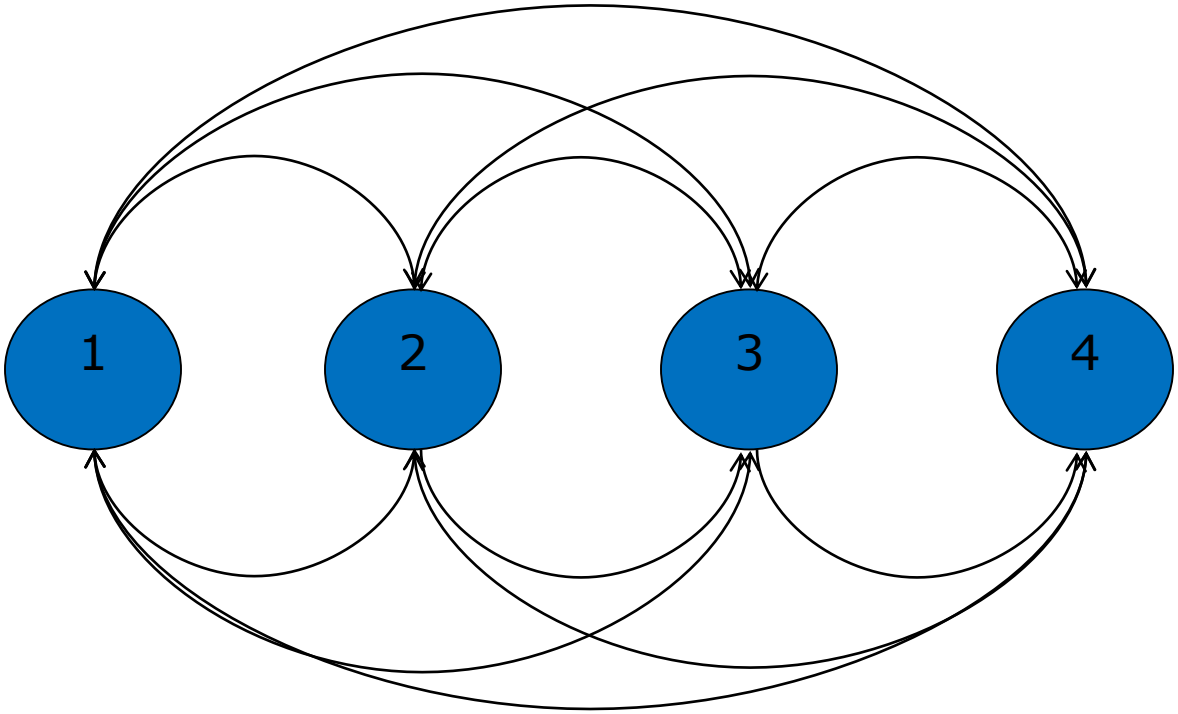
Approximate DP Approaches

- Deterministic aggregation methods
  - State aggregation
  - Basis function expansion
- Sampling based methods

Approximation: Reducing the State Set

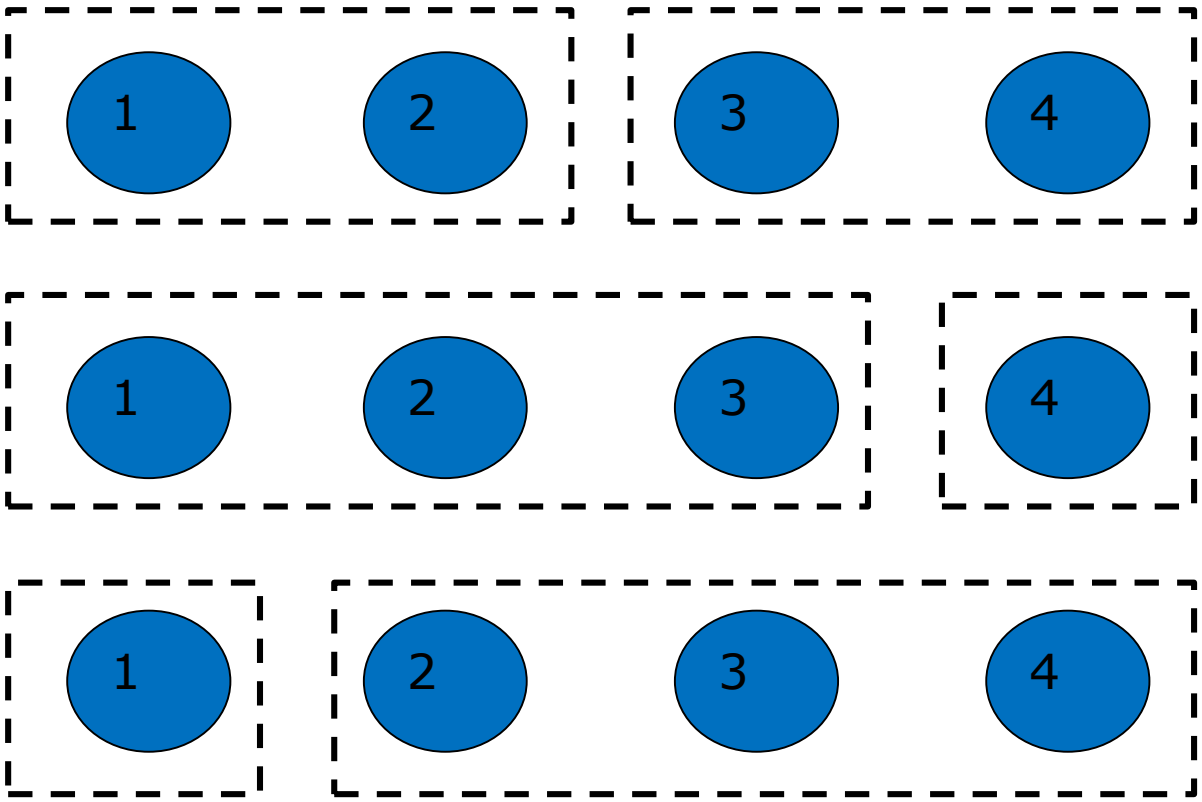
The number of state is often the most prohibitive characteristic of large-scale DPs. Aggregating states is one approximation method for reducing the number of states:

4 State Example:



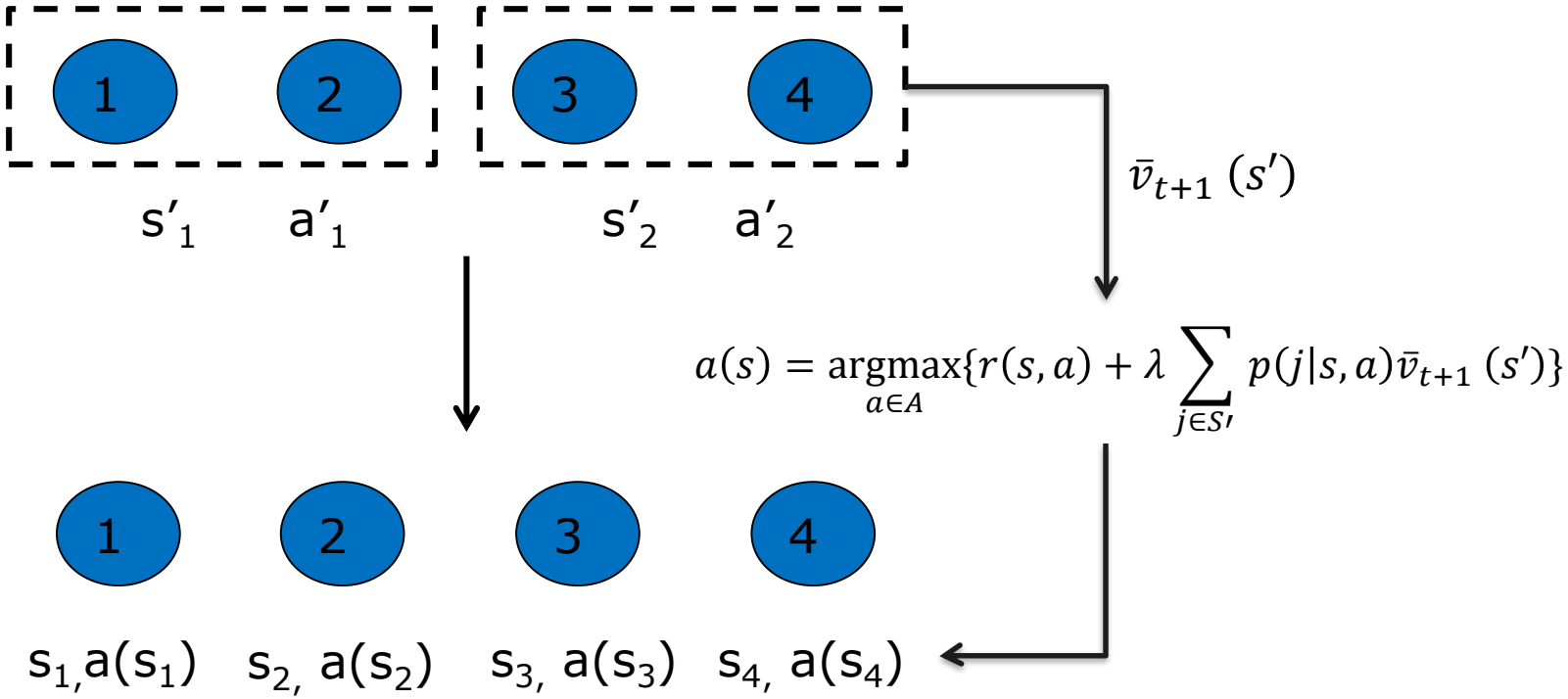
Approximations:State Aggregation

Aggregation involves generating a **partition** that is a “reasonable” approximation



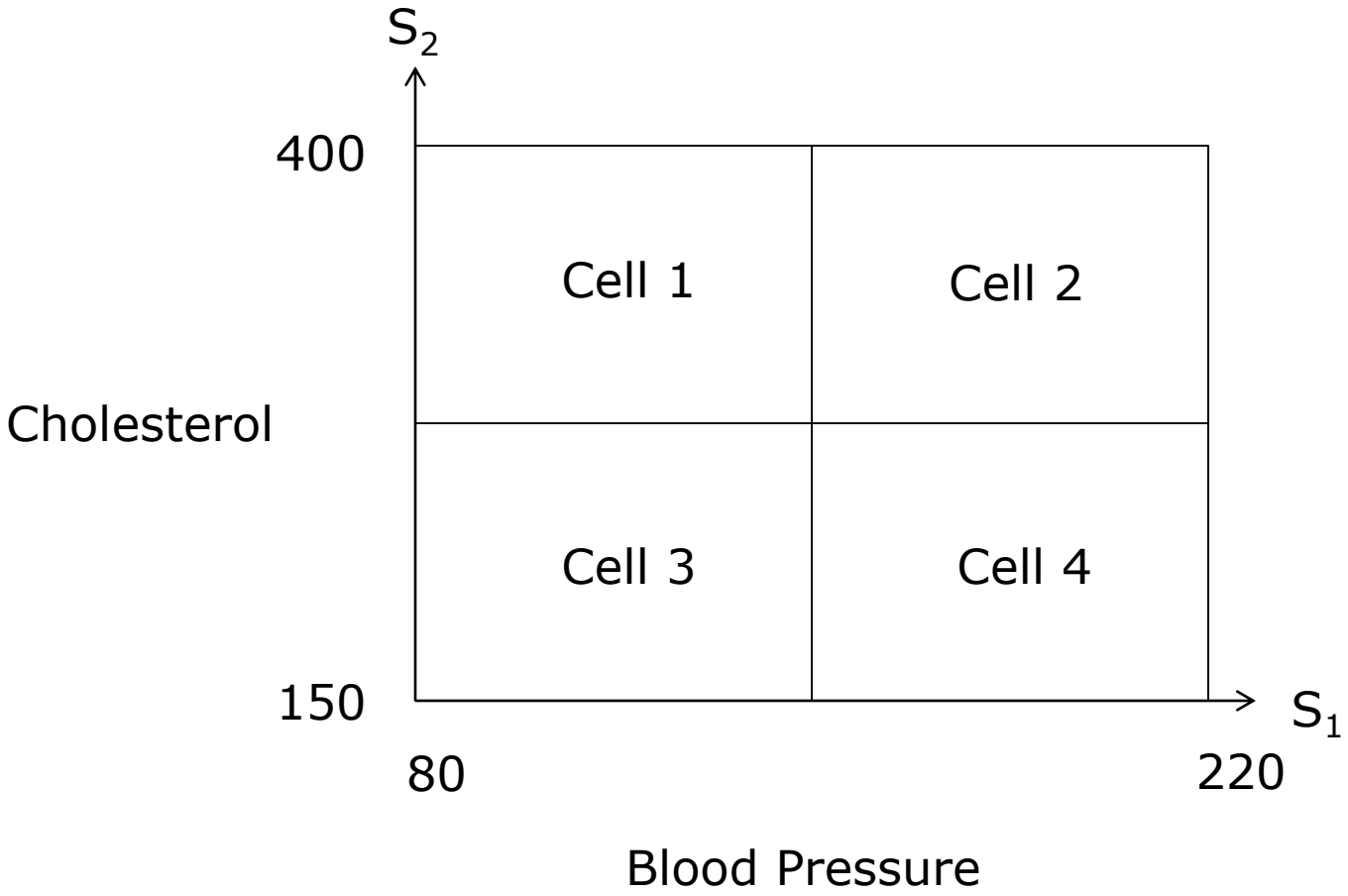
Approximations:State Disaggregation

After solving the DP the resulting policy must be disaggregated



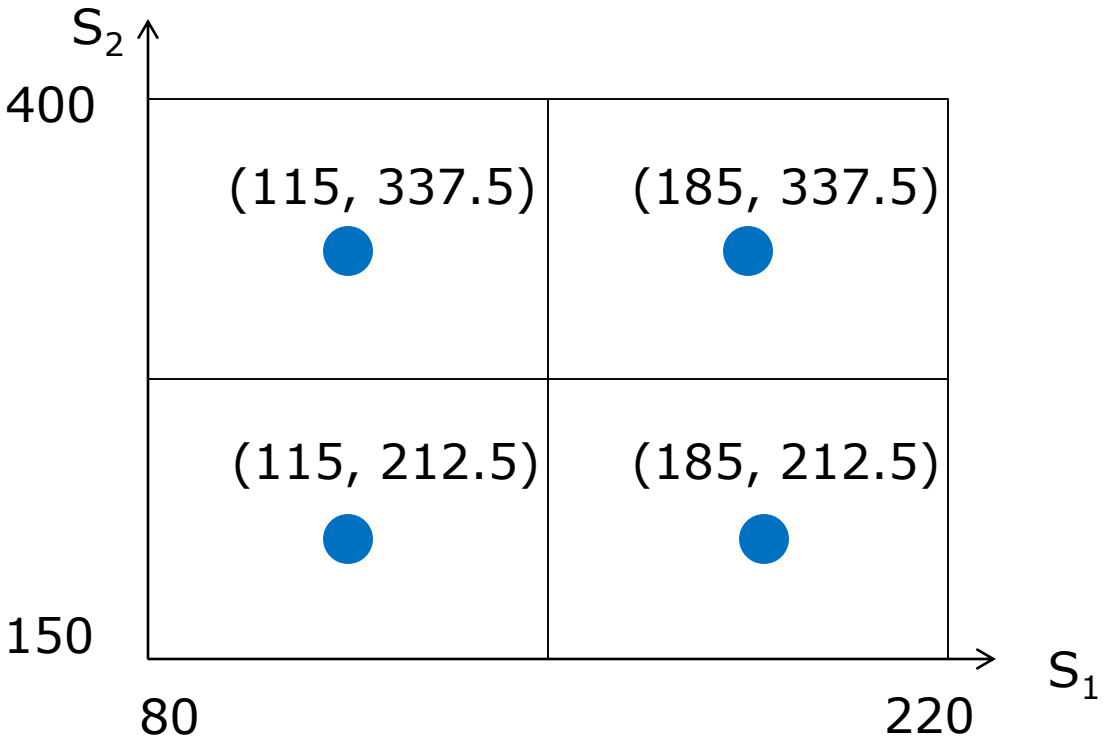
Aggregation in Multiple Dimensions

Consider the following 2-dimensional example. States are S_1 and S_2 which can vary continuously in some range. For example the states in a drug treatment model could be cholesterol and blood pressure.



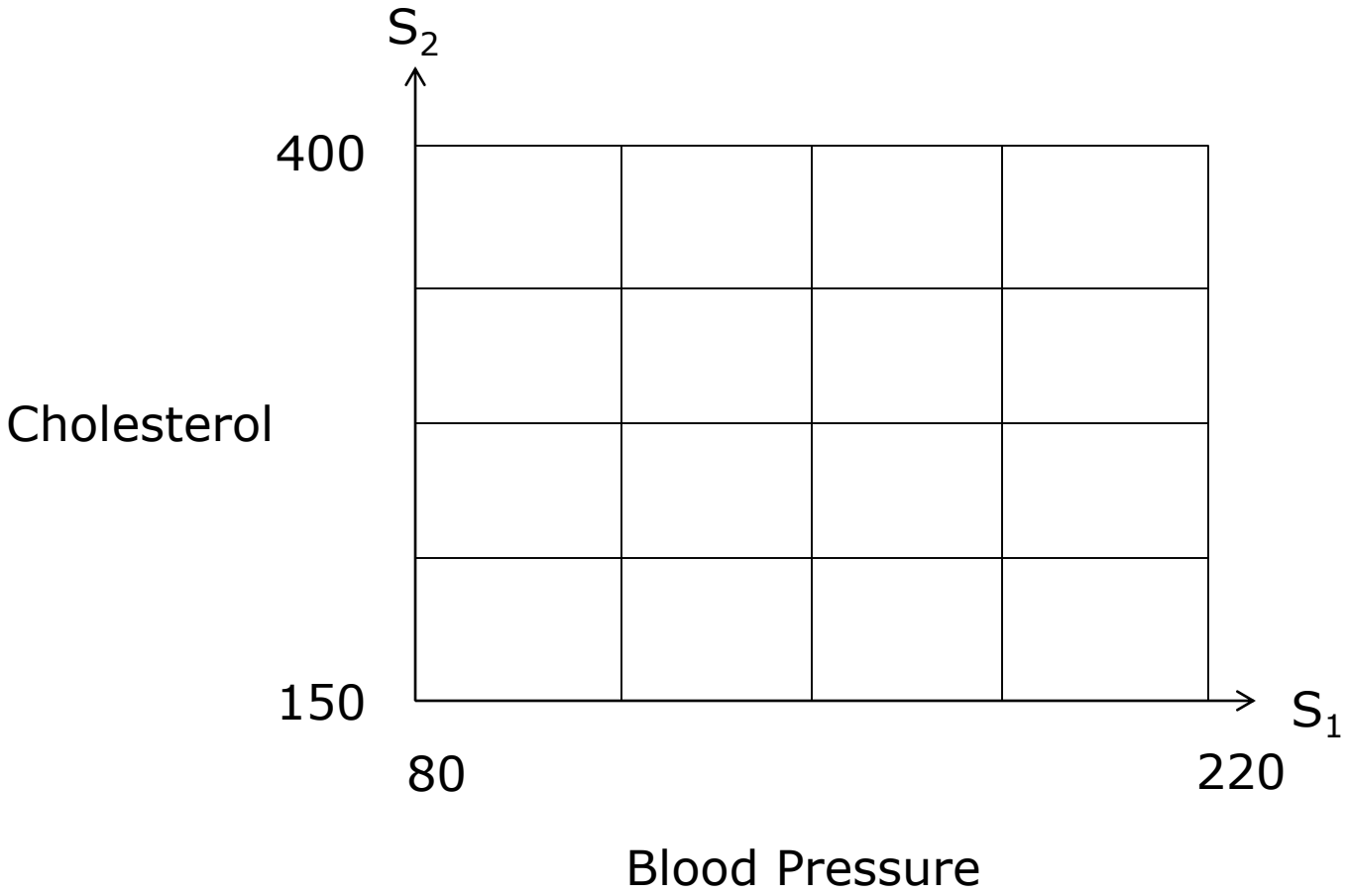
Aggregation in Multiple Dimensions

4 State Example: Each **cell** of the partition is represented by a **discrete state** (e.g. conditional mean)



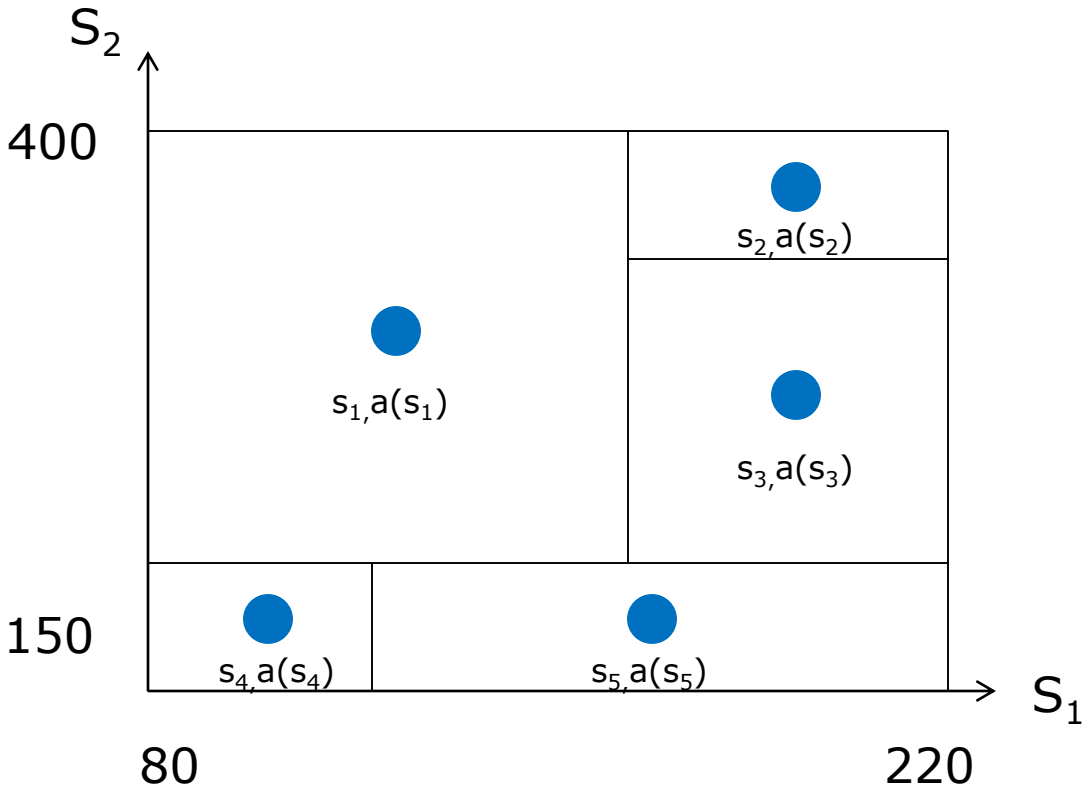
Aggregation in Multiple Dimensions

Consider the following 2-dimensional example. States are S_1 and S_2 which can vary continuously in some range. For example the states in a drug treatment model could be cholesterol and blood pressure.



Linear Growth Partitioning

The following is an alternative partitioning method in which three partitioning decisions are made: (a) cell, (b) direction, (c) position



Dimension Reduction Methods

Basis function approximations approximate the optimal value function as a weighted sum of basis functions:

For certain types of functions the value function can be expressed exactly → $V(s) = \sum_{k=1}^{\infty} r_k \phi_k(s)$

Realistically a finite set of K functions must be selected → $V(s) \approx \tilde{V}(s) = \sum_{k=1}^{K} r_k \phi_k(s)$

1. Select Basis functions $\phi_1, \ldots, \phi_K$, which are functions of the original state characteristics (e.g. blood pressure, cholesterol)

2. Compute weights $r_1, \ldots, r_K$ so that $\sum_{k=1}^{K} r_k \phi_k(s) \approx V(s)$

Dimension Reduction Methods

Basis function approximations seek to approximate the optimal value function with a lower dimensional function:

$$\tilde{V}(s) = \sum_{k=1}^{K} r_k \phi_k(s)$$

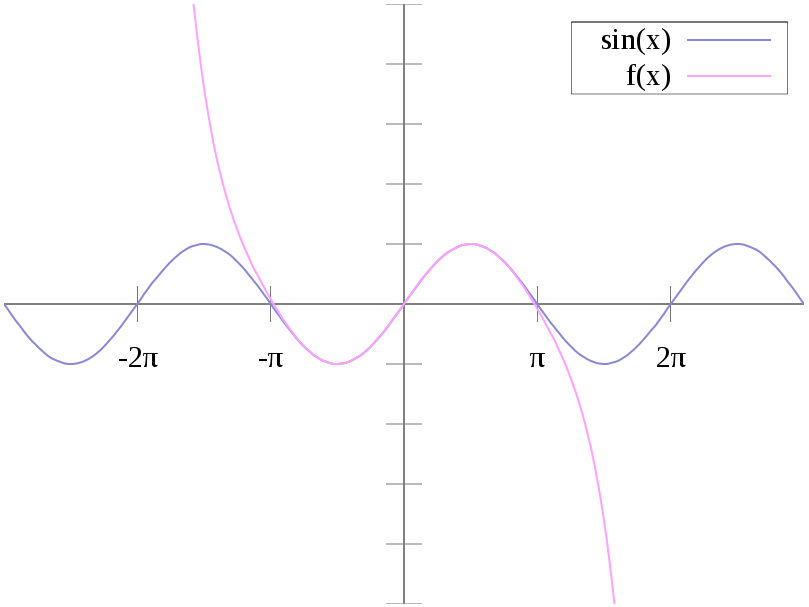
1. Select Basis functions $\phi_1, \ldots, \phi_K$, which are function of the original state characteristics

2. Compute weights $r_1, \ldots, r_K$ so that $\sum_{k=1}^{K} r_k \phi_k(s) \approx V(s)$

Approximation Examples

Taylor series:

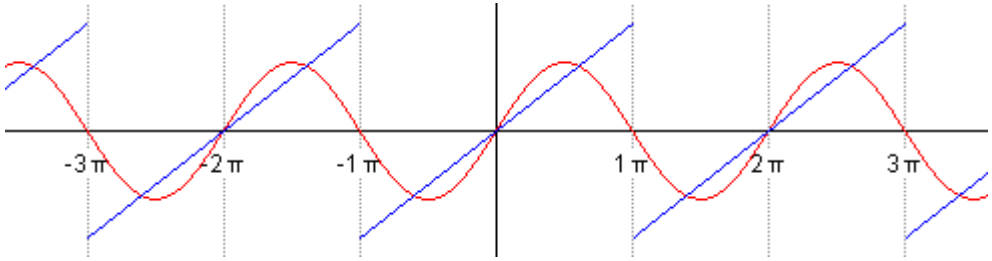
$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$



Linear regression:

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

Examples of Basis Functions



Examples of common basis functions:

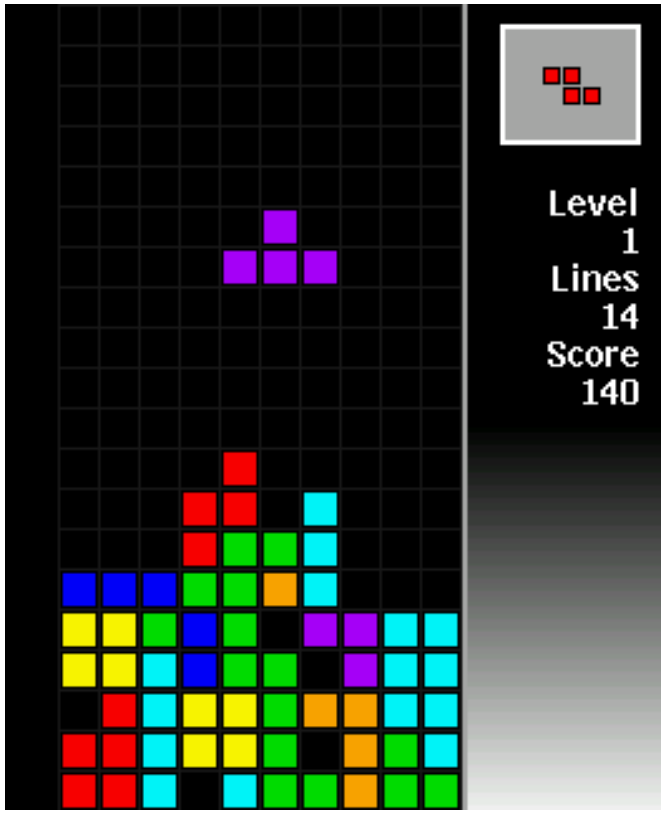
Fourier Series

Radial Basis Functions

Legendre Polynomials

Wavelets

Example: Tetris



$\phi_1, \ldots, \phi_{10}$: Basis functions mapping the state to the height h_k of each of the ten columns

$\phi_{11}, \ldots, \phi_{19}$: Basis functions mapping the state to the absolute difference between heights of successive columns: $|h_{k+1} - h_k|, k = 1, \ldots, 9$

ϕ_{20}: Basis function maps state to the maximum column height: $\max_k h_k$

ϕ_{21}: Basis function maps state to number of "holes" in the wall

ϕ_{22}: Basis function is equal to one for every state

Approximate Linear Program

The linear programming formulation of an MDP can be used as a way to estimate basis function coefficients.

$$\min \sum_{j \in S} \alpha_j v(j)$$

s.t.

$$v(s) - \lambda \sum_{j \in S} p(j|s,a) v(j) \geq r(s,a), \quad \forall a \in A, s \in S$$

$$v(s) \text{ urs}, \quad \forall s \in S$$

This LP has many decision variables and constraints

Replacing $v(j)$ with $\tilde{v}(j) = \sum_{k=1}^{K} r_k \phi_k(s)$

$$\min \sum_{j \in S} \alpha_j \sum_{k=1}^{K} r_k \phi_k(j)$$

s.t.

$$\sum_{k=1}^{K} r_k \phi_k(s) - \lambda \sum_{j \in S} p(j|s,a) \sum_{k=1}^{K} r_k \phi_k(s) \geq r(s,a), \quad \forall a \in A, s \in S$$

$$r_k \text{ urs}, \quad \forall k = 1, \ldots, K$$

This LP has K decision variables and many constraints

How good are these approximations?

Example: Tetris

- Using a linear program to set the basis function coefficient resulted in policies with an average of 4274 points per game (human experts average about 3200 points per game)

- Other more advanced methods have lead to improvements averaging 5500 points per game

Want to learn more?

Powell WB. Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley; 2007.

Schweitzer PJ, Seidmann A. Generalized Polynomial Approximations in Markovian Decision Processes. Journal of Mathematical Analysis and Applications. 1985;110:568–582.

De Farias DP, Van Roy B. The Linear Programming Approach to Approximate Dynamic Programming. Operations Research. 2003;51(6):850–865.