CS475/CS675
Lecture 24: July 21, 2016

Open problems

Two Open Problems

- **Kernel methods:** how to solve linear systems of equation in less than cubic time
- **Markov decision processes:** how to evaluate factored policies in less than exponential time

Kernel Methods

- Class of **non-parametric** Machine Learning techniques that scale with the amount of data
- Examples:
  - **Gaussian processes**
  - Support vector machines
  - Kernel logistic regression
  - Kernel principal component analysis
  - Kernel perceptron

Gaussian Process

- Quick recall:
  - Non-parametric regression
  - Infinite dimensional Gaussian
- Picture:

Kernel

- Covariance function is a kernel function

$$k(x, x') = \phi(x)^T \phi(x')$$

- Where $\phi(x)$ is the feature function that defines the kernel
- Popular kernels with infinitely many features:

Gaussian kernel: $\frac{e^{\left|\left|x-x'\right|\right|_2^2}}{\sigma}$

Exponential kernel: $\frac{e^{\left|\left|x-x'\right|\right|_2}}{\sigma}$

Common problem

- In all kernel methods, a linear system of equations must be solved:

$$(K + cI)w = b$$

- K is an instantiation of the kernel function called the Gram matrix, i.e. $K_{x,x'} = k(x, x') \quad \forall x, x' \in dataset$
- c is a constant positive scalar
- b is constant vector
- w is the vector of unknowns

Challenge

- K is an $n \times n$ matrix where n is the number of data points in the dataset
- Linear system takes $O(n^3)$ time to solve
- This does not scale to large datasets, i.e., millions or billions of data points.
- How can we reduce the time to $O(n^2)$ or less?

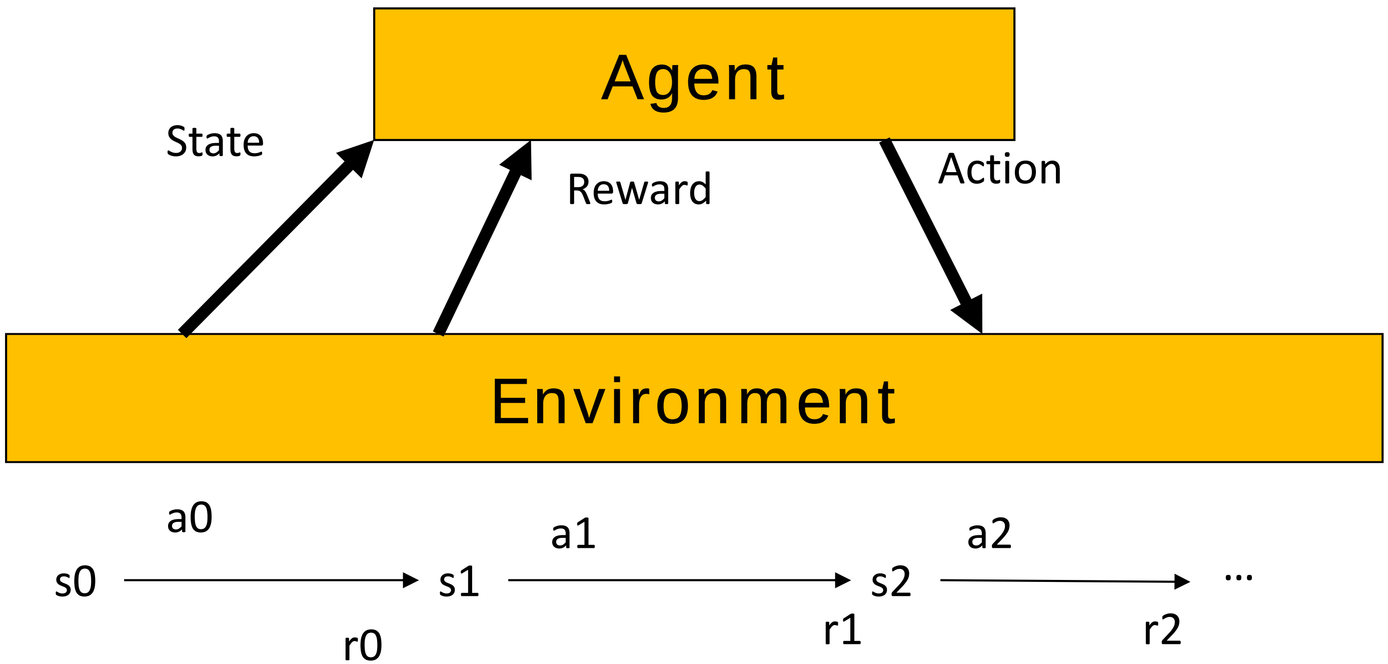
Properties

- Gram matrix K is
  - Symmetric
  - Positive semi-definite
  - We also know the feature function $\phi(x)$ that is used to create $k(x, x') = \phi(x)^T \phi(x')$

- Can you exploit those properties to reduce the solution complexity to $O(n^2)$ or less?

Markov Decision Processes

- Popular model in Operations Research and Artificial Intelligence for decision-theoretic planning



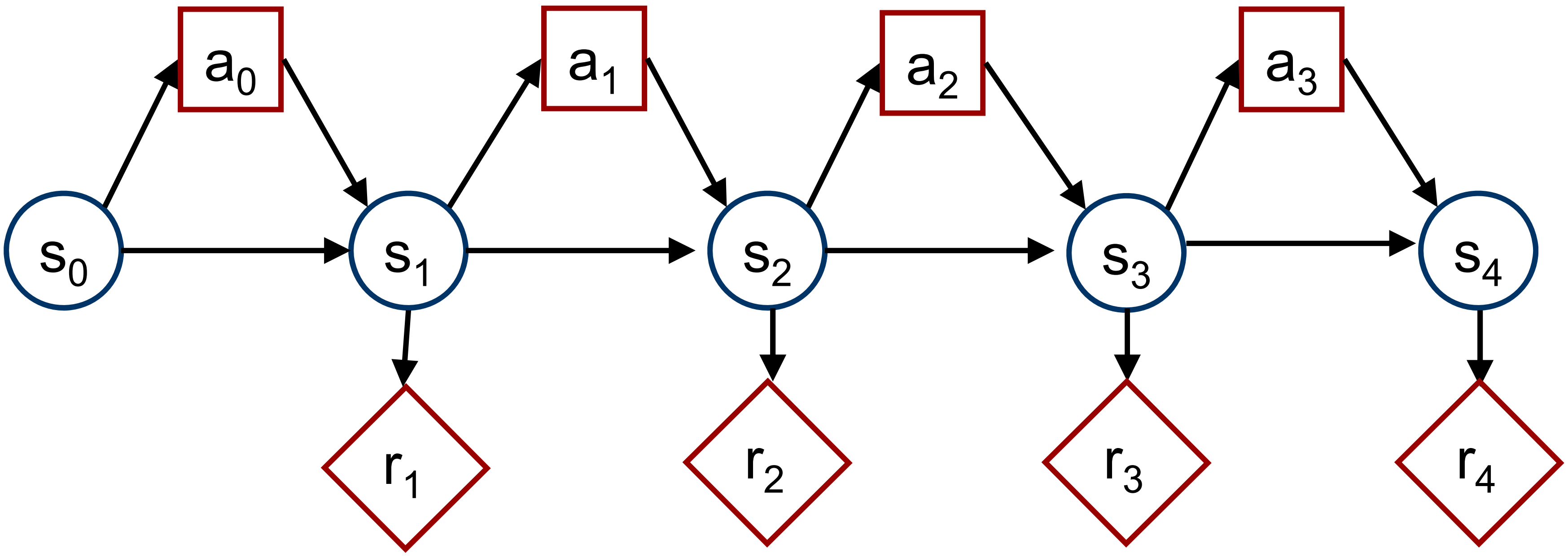
Markov Decision Processes

Formally: $< S, A, T, R, \gamma >$

Set of states S, set of actions A, discount $\gamma \in (0,1)$

Transition function $T(s, a, s') = \Pr(s'|s, a)$

Reward function $R(s, a) \in \mathcal{R}$



Policy

- Policy $\pi: S \rightarrow A$ (mapping from states to actions)

- Let n be the number of states
- Transition matrix: T^{π} $(n \times n)$

- Reward vector: R^{π} $(n \times 1)$

Value Function

- Value $V^{\pi}(s_0)$ of a policy π at state s_0:

$$
\begin{aligned}
V^{\pi}(s_0) = {} & \mathrm{R}^{\pi}(\mathrm{s}_0) \\
& +\gamma \sum_{s_1} \Pr(s_1|s_0,\pi)\, R^{\pi}(s_1) \\
& +\gamma^2 \sum_{s_1} \Pr(s_1|s_0,\pi) \sum_{s_2} \Pr(s_2|s_1,\pi)\, R^{\pi}(s_2) \\
& +\gamma^3 \sum_{s_1} \Pr(s_1|s_0,\pi) \sum_{s_2} \Pr(s_2|s_1,\pi) \sum_{s_3} \Pr(s_3|s_2,\pi)\, R^{\pi}(s_3) \\
& + \cdots
\end{aligned}
$$

Bellman's Equation

- Recursive formula:

$$V^{\pi}(s_0) = \mathrm{R}^{\pi}(\mathrm{s}_0) + \gamma \sum_{s_1} \Pr(s_1|s_0,\pi)\, V^{\pi}(s_1)$$

- Matrix form:

$$V^{\pi} = R^{\pi} + \gamma T^{\pi} V^{\pi}$$

- Solution: system of linear equation

$$(I - \gamma T^{\pi})V^{\pi} = R^{\pi}$$

Problem

- Let n be the number of states
- Transition matrix T^{π} is $n \times n$
- Time $O(n^3)$ which is prohibitive for large state spaces

Factored MDP

- Let k be the number of binary features
- Each state corresponds to all combinations of binary features
- This yields $n = 2^k$ states
- Time $O(2^{3k})$ which is exponential in the number of features
- Challenge: can we reduce the solution to be polynomial in k?

Factored MDP

- Factored transition matrix

$$T(f_1', f_2', \dots, f_k' | f_1, f_2, \dots, f_k) = \prod_{i=1}^{k} \Pr(f_i' | parents(f_i'))$$

- Additive reward function

$$R^\pi(f_1, f_2, \dots, f_k) = \sum_{i=1}^{k} R^\pi(f_i)$$

Properties

- Factored MDP
  - Rows of T^{π} sum to 1
  - Largest eigenvalue of T^{π} is 1
  - T^{π} is factored and R^{π} is additive

- Can you exploit those properties to reduce the time complexity to be polynomial in k?