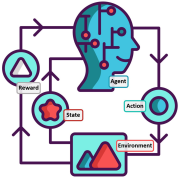
Unsupervised Learning, Large-scale Machine Learning, Reinforement Learning, Final Review

Dana Golden, Lilia Maliar



Data Science and Machine Learning - **November 30, 2024**

Presentation Outline

1. Introduction and Background
2. Unsupervised Learning
3. Large-scale ML
4. RL
5. Final Review
6. Finding Data

Review of Classification

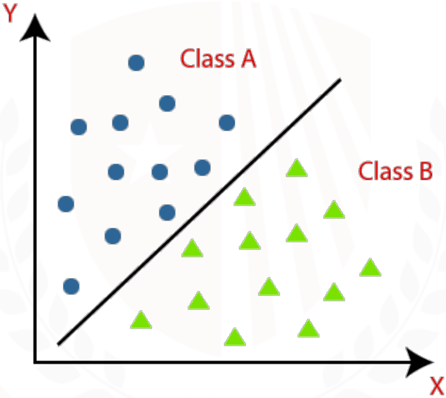


Figure 1: Classification Machine Learning

Unsupervised Learning

- No labels
- Finds patterns and creates groups in the underlying structure of the data
- Useful for anomaly detection or creating groups where previously missing, market segmentation

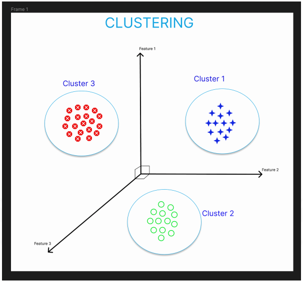


Figure 2: Clustering

Large-scale Machine Learning

- Real-world machine learning is highly dependent on massive amounts of data
- This means cloud computing, databases, efficient algorithms, and GPUs

Large-scale Machine Learning

- Real-world machine learning is highly dependent on massive amounts of data
- This means cloud computing, databases, efficient algorithms, and GPUs
- Companies don't want you to analyze data anymore, they want you to build things that interact with users and exist in the real world

Reinforcement Learning

- Reinforcement learning shows remarkable promise for synergy with economics in the future
- Reinforcement learning is the topic in machine learning most likely to have a major impact on the economics profession
- Multi-agent reinforcement learning is actually highly similar to game theory

K-means Overview

- An unsupervised learning algorithm for dividing highly dimmensional datapoints into clusters
- Computationally intensive but easy due to heuristic algorithms
- Great starting place to find patterns in data

K-means Visualized

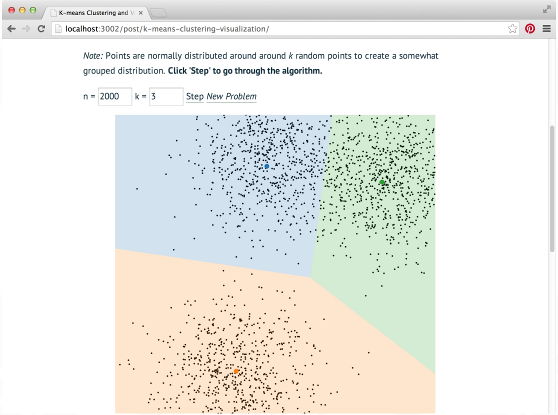


Figure 3: K-means visualized

K-means Objective Function

- Given a set of n d-dimensional observations $(x_1, x_2, ...x_n)$, K-means clustering partitions them into $l \leq n$ sets $S = \{S_1, S_2...S_k\}$
- K-means minimizes within-cluster sum of squares/ variance

$$arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 = arg\min_S \sum_{i=1}^{k} |S_i| Var(S_i) \quad (1)$$

K-means Algorithm

- Random initialization: A set of random points chosen for initialization
- Assignment step: Each point is assigned to cluster with closest center

$$S_i^{(t)} = \{x_p : \left\|x_p - m_i^{(t)}\right\|^2 \leq \left\|x_p - m_j^{(t)}\right\|^2 \forall j, 1 \leq j \leq k\} \qquad (2)$$

- Center of cluster reset to mean of cluster

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \qquad (3)$$

- Termination: Algorithm terminates when within cluster sum of squares stops decreasing or assignments no longer change
- Different results every time, always converges but not necessarily to global optimum
- How do we know it always converges?

K-medoids Algorithm

- Greedily select k of N datapoints to minimize cost
- Associate datapoints to closest medoid by any distance metric
- Consider every non-mediod point in the medoid and pick the new centers based on ones that minimize costs
- Reassign points
- Continue to do this until termination
- If cost stops decreasing, terminate

K-medoids Algorithm

- Greedily select k of N datapoints to minimize cost
- Associate datapoints to closest medoid by any distance metric
- Consider every non-mediod point in the medoid and pick the new centers based on ones that minimize costs
- Reassign points
- Continue to do this until termination
- If cost stops decreasing, terminate
- Minimizes pairwise distances rather than variance, so more robust to outliers

Elbow Curve

- Choose number of clusters based on inflection point of Elbow curve

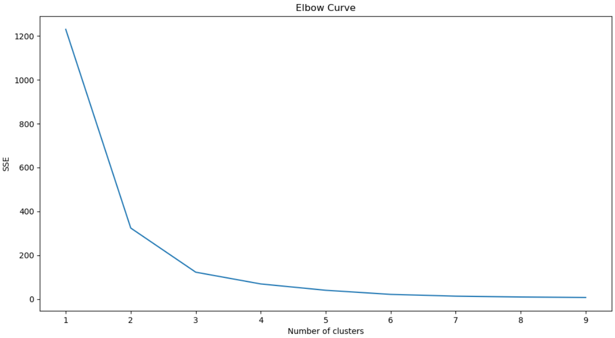


Figure 4: Elbow Curve

Cloud Computing

- Cloud computing resources pool the power of many individual computing resources simultaneously
- Cloud computing runs many algorithms in parallel on large amounts of data
- Useful for creating large models that are put into operation

Seawulf

- Seawulf is SBU's research cloud computing system
- Seawulf allows many machines to be used to run tasks in parallel
- Valuable way to train large models

Graphical Processing Units

- Graphical processing units (GPUs) speed up training of neural networks
- GPUs are very good at performing certain tasks repetitively and train networks much faster

Graphical Processing Units

- Graphical processing units (GPUs) speed up training of neural networks
- GPUs are very good at performing certain tasks repetitively and train networks much faster
- The best neural networks are trained on servers of GPUs
- You can access GPU's through Google Colab

Reinforcement Learning Overview

- Reinforcement learning learns from the environment using punishments and rewards
- An agent observes the state, takes an action, moves to the next state, and observes a reward

Reinforcement Learning Overview

- Reinforcement learning learns from the environment using punishments and rewards
- An agent observes the state, takes an action, moves to the next state, and observes a reward
- Very similar to many economic setups
- The current forefront of machine learning, very similar to human learning
- Very useful for learning complex environments

Example Problem

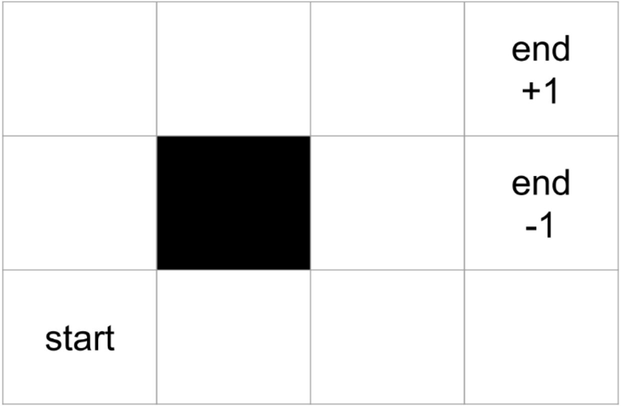
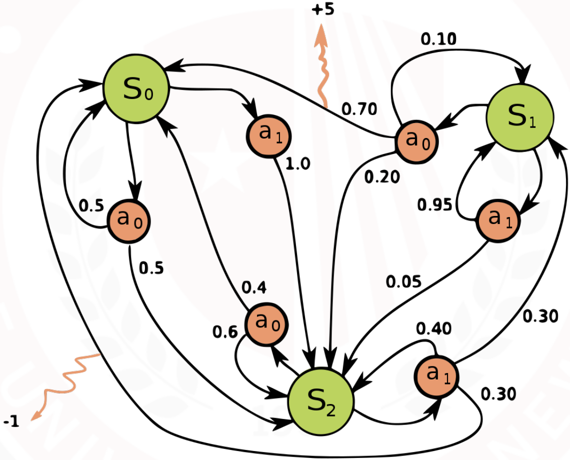


Figure 5: Grid World Example

Markov Decision Process

- A Markov Decision Process consists of a 4-tuple (S, A, P_a, R_a)
- S is the state space
- A is the action space $P_a(s, s')$ is the conditional probability that action a in state s at time t will lead to state s' at time $t + 1$, generally assumed to have Markov property
- $R_a(s, s')$ is the reward received from transitioning from state s to s' due to action a

MDP Visualized



Dynamic Programming

- Value iteration is common in both dynamic economics and reinforcement learning
- Value iteration is a model-based method to find the value of being in a particular state

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

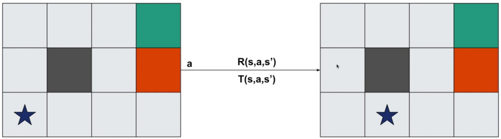


Figure 7: Value Iteration

Q-learning

- New Q-value is equal to the former Q-value plus the learning rate times the reward plus the discounted maximum Q value of the next state

$$Q(s,a)' = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \tag{4}$$

Q-learning

- New Q-value is equal to the former Q-value plus the learning rate times the reward plus the discounted maximum Q value of the next state

$$Q(s,a)' = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s,a)) \tag{4}$$

- How do you learn new states?

Multi-armed Bandits

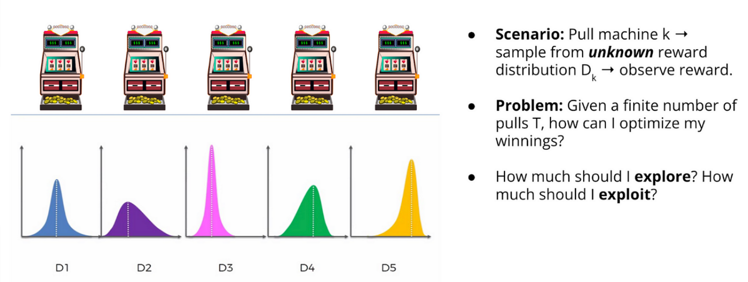


Figure 8: Multi-armed Bandits

Exploration Exploitation Tradeoff

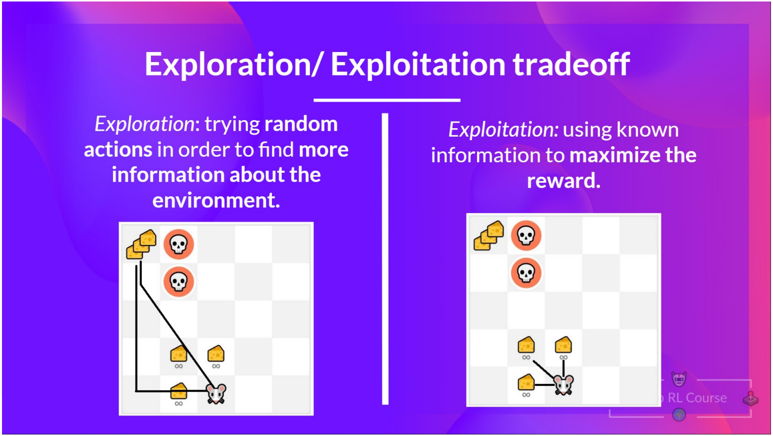
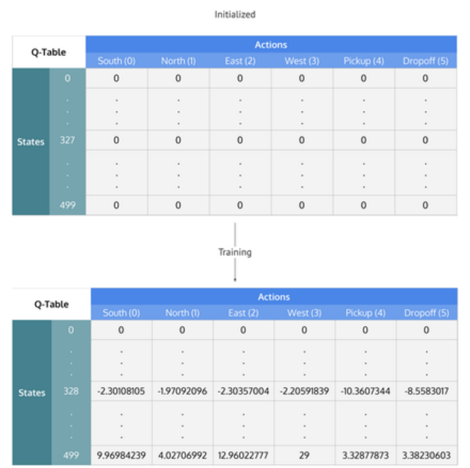


Figure 9: Caption

Q-table



Initialized

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	327	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	0	0	0	0	0	0

Training

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

Figure 10: Q-table Visualized

Gridworld with Q-values

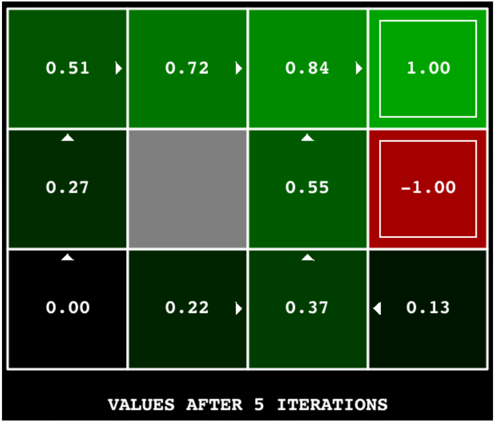


Figure 11: Grid World after iterations

Deep Q-learning

- Deep Q-learning is the forefront of reinforcement learning
- Deep Q-learning combines neural networks with Q-learning to more efficiently map the Q-function
- Allows for significantly more dimensional state spaces

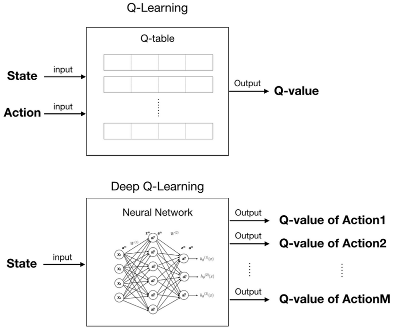


Figure 12: Deep Q-learning

Reinforcement Learning and Economics

- Game theory: Algorithmic decision making
- IO: Algorithmic pricing
- Macro: Dynamic optimization, HANK models

Final Review

- What questions do you have?

Open-source Data

- FRED
- ERS
- EIA
- NASS
- Census
- OECD
- IEA
- World Bank
- BEA
- BLS
- IRS: Statistics of Income

IPUMS

- IPUMS

WRDS

- Compustat
- IRI
- CRSP

Common Surveys

- SIPP
- NLSY
- CEX
- SCF
- PSID

Thank You So Much!