Unsupervised Learning, Recommenders, Reinforcement Learning

Week 1: Unsupervised learning

Learning Objectives:

- Implement the k-means clustering algorithm
- Implement the k-means optimization objective
- Initialize the k-means algorithm
- Choose the number of clusters for the k-means algorithm
- Implement an anomaly detection system
- Decide when to use supervised learning vs. anomaly detection
- Implement the centroid update function in k-means
- Implement the function that finds the closest centroids to each point in k-means

Clustering is an unsupervised machine learning technique used to group similar data points together.

The **K-means** algorithm is a method to automatically cluster similar data points together. Each cluster is represented by its centroid, which is the mean of the points in the cluster. The algorithm iteratively assigns data points to clusters and updates the centroids until convergence.

Anomaly detection is the process of identifying unusual patterns that do not conform to expected behavior, often referred to as outliers.

Supervised learning vs. anomaly detection

Methods for Transformation Non-Gaussian Features to Gaussian

1. Log Transformation:
   y=log(x+1) (adding 1 to avoid log of zero)

2. Box-Cox Transformation:

Formula: $$y = \frac{(x^\lambda - 1)}{\lambda} \text{ if } \lambda \neq 0; y = \log(x) \text{ if } \lambda = 0.$$

Gaussian distribution is given by

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean and σ^2 is the variance.

Week 2: Recommender systems

Learning Objectives:

- Implement collaborative filtering recommender systems in TensorFlow
- Implement deep learning content based filtering using a neural network in TensorFlow
- Understand ethical considerations in building recommender systems

Collaborative filtering is a technique used by recommender systems to predict the interests of a user by collecting preferences from many users.

User-based collaborative filtering: Finds users who are similar to the active user and recommends items they liked.

Item-based collaborative filtering: Finds items similar to the ones the active user liked and recommends them.

The **cold start problem** occurs in recommender systems when there is insufficient data to make accurate recommendations.

Content-based filtering: Recommends items based on their similarity to items the user has liked in the past. It uses features of items (like genre, author, description) and the user's history to make recommendations.

Week 3: Reinforcement learning

Learning Objectives:

- Understand key terms such as return, state, action, and policy as it applies to reinforcement learning
- Understand the Bellman equations
- Understand the state-action value function
- Understand continuous state spaces
- Build a deep Q-learning network

Bellman's equation

$$Q_{i+1}(s, a) = R + \gamma \max_{a'} Q_i(s', a')$$

Return: The total accumulated reward an agent receives over time.
State: The current situation or configuration of the environment.
Action: A decision or move made by the agent that affects the state.
Policy: A strategy that maps states to actions, guiding the agent's behavior.

Experience Replay: A technique where the agent stores past experiences in a buffer and randomly samples from it to train, improving learning stability and efficiency.

In **continuous state spaces**, the set of possible states is infinite and not discrete.