Introduction to Machine Learning

Clustering
Goals of Supervised Learning

Learn a hypothesis $h$ from labeled dataset $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ that has low error on unseen data.
Goals of Reinforcement Learning

Find a “policy” $\pi : S \rightarrow A$ that maximizes reward in an environment.

- **plant:**
  - rich soil $\rightarrow$ poor soil with probability 0.9
  - poor soil $\rightarrow$ rich soil with probability 0.1
  - $P(S_t = \text{poor} \mid S_{t-1} = \text{rich}, A_{t-1} = \text{plant}) = 0.9$ (probability of transitioning from rich soil to poor soil)
  - $T(\text{rich, plant, poor}) = 0.1$ (probability of staying in rich soil)

- **fallow:**
  - rich soil $\rightarrow$ poor soil with probability 0.98
  - poor soil $\rightarrow$ rich soil with probability 0.02
  - $P(\text{rich, following, poor}) = 0.95$ (probability of transitioning from rich soil to poor soil)
  - $T(\text{poor, Following, rich}) = 0.05$ (probability of staying in rich soil)

Rewards:
- $R(\text{rich, plant}) = 100$
- $R(\text{poor, plant}) = 10$
- $R(\text{rich, fallow}) = 0$
- $R(\text{poor, fallow}) = 0$
Goals of Unsupervised Learning?

Clustering

Cluster Dendrogram

Goals of Unsupervised Learning?

Representation Learning

(Lab 7)
The ML Landscape

- **Supervised Learning**: Train a model that performs well (e.g., high accuracy) on unseen data.

- **Reinforcement Learning**: Learn a policy that maximizes expected reward in some environment.

- **Unsupervised Learning**:
  - Extract useful insights from data
  - Learn features for downstream tasks
Clustering

- Find a mapping from each data point to a cluster.
- Modeling choices:
  - How many clusters?
  - How do we define “close”?
  - How do we know if we have succeeded?
K-Means Clustering Objective

\[
\arg \min_{\mu, y} \sum_{i=1}^{n} \sum_{j=1}^{k} 1\{y^{(i)} = j\} \|x^{(i)} - \mu^{(j)}\|^2_2
\]

Find K cluster assignments and cluster means such that across all data points, the squared Euclidean distance between the data point and the cluster mean of its assigned cluster is minimized.
K-Means Algorithm

Input:

1. Data points \( \{x^{(i)}\}_{i=1}^{n} \)
2. Number of clusters \( k \)
3. Number of iterations

Output:

1. An assignment of each point to a cluster.
2. A “centroid” of each cluster with which to assign new points.
K-Means Algorithm

Input:
1. Data points \( \{x^{(i)}\}_{i=1}^{n} \)
2. Number of clusters \( k \)
3. Number of iterations \( \tau \)

Output:
1. An assignment of each point to a cluster.
2. A “centroid” of each cluster with which to assign new points.

\[
\text{k-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
\quad \text{y}_{\text{old}} = y \\
\quad \text{for } i = 1 \text{ to } n \\
\quad \quad y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|^2_2 \\
\quad \quad \text{for } j = 1 \text{ to } k \\
\quad \quad \quad \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \\
\quad \quad \text{if } y = y_{\text{old}} \\
\quad \quad \quad \text{break} \\
\quad \text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
K-Means Algorithm

\[
k\text{-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
y_{old} = y \\
\text{for } i = 1 \text{ to } n \\
y^{(i)} = \arg \min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\text{for } j = 1 \text{ to } k \\
\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}} \\
\text{if } y = y_{old} \\
\text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
K-Means Algorithm

\[ \text{k-means}(k, \tau) \]
\[
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
\quad y_{\text{old}} = y \\
\quad \text{for } i = 1 \text{ to } n \\
\quad \quad y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_{2}^{2} \\
\quad \text{for } j = 1 \text{ to } k \\
\quad \quad \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \\
\quad \text{if } y = y_{\text{old}} \\
\quad \quad \text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \]
K-Means Algorithm

\[
k\text{-means}(k, \tau)
\]
\[
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
\[
\text{for } t = 1 \text{ to } \tau
\]
\[
y^{(i)}_{\text{old}} = y^{(i)}
\]
\[
\text{for } i = 1 \text{ to } n
\]
\[
y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2
\]
\[
\text{break}
\]
\[
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]

Some options:
1. Choose \( k \) data points uniformly at random, without replacement
2. Choose uniformly at random within the span of the data

K = 5
K-Means Algorithm

\[
\text{k-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^k, \{y^{(i)}\}_{i=1}^n \\
\text{for } t = 1 \text{ to } \tau \\
y_{\text{old}} = y \\
\text{for } i = 1 \text{ to } n \\
y^{(i)} = \arg \min_j \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\text{for } j = 1 \text{ to } k \\
\mu^{(j)} = \frac{\sum_{i=1}^n 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^n 1\{y^{(i)} = j\}} \\
\text{if } y = y_{\text{old}} \\
\text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^k, \{y^{(i)}\}_{i=1}^n
\]
k-means\( (k, \tau) \)

\[
\begin{align*}
\text{Init } & \{\mu^{(j)}\}^{k}_{j=1}, \{y^{(i)}\}^{n}_{i=1} \\
\text{for } & t = 1 \text{ to } \tau \\
& y_{\text{old}} = y \\
\text{for } & i = 1 \text{ to } n \\
& y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\text{for } & j = 1 \text{ to } k \\
& \mu^{(j)} = \frac{\sum^{n}_{i=1} 1\{y^{(i)} = j\} x^{(i)}}{\sum^{n}_{i=1} 1\{y^{(i)} = j\}} \\
\text{if } & y = y_{\text{old}} \\
& \text{break} \\
\text{return } & \{\mu^{(j)}\}^{k}_{j=1}, \{y^{(i)}\}^{n}_{i=1}
\end{align*}
\]
K-Means Algorithm

\[ \text{k-means}(k, \tau) \]
\[
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
\[ \text{for } t = 1 \text{ to } \tau \]
\[ y_{\text{old}} = y \]
\[ \text{for } i = 1 \text{ to } n \]
\[ y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|^{2} \]
\[ \text{for } j = 1 \text{ to } k \]
\[ \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\}x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \]
\[ \text{if } y = y_{\text{old}} \]
\[ \text{break} \]
\[ \text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \]

Assign each data point to closest “centroid”
K-Means Algorithm

\[ \text{k-means}(k, \tau) \]
\[ \text{Init } \{\mu(j)\}_{j=1}^{k}, \{y(i)\}_{i=1}^{n} \]
\[ \text{for } t = 1 \text{ to } \tau \]
\[ y_{old} = y \]
\[ \text{for } i = 1 \text{ to } n \]
\[ y(i) = \arg \min_j \|x(i) - \mu(j)\|_2 \]
\[ \text{for } j = 1 \text{ to } k \]
\[ \mu(j) = \frac{\sum_{i=1}^{n} 1\{y(i) = j\}x(i)}{\sum_{i=1}^{n} 1\{y(i) = j\}} \]
\[ \text{if } y = y_{old} \]
\[ \text{break} \]
\[ \text{return } \{\mu(j)\}_{j=1}^{k}, \{y(i)\}_{i=1}^{n} \]

Update centroid as the average of all the points assigned to it.
K-Means Algorithm

\[
\text{k-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
\quad y_{\text{old}} = y \\
\quad \text{for } i = 1 \text{ to } n \\
\quad \quad y^{(i)} = \arg \min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\quad \text{for } j = 1 \text{ to } k \\
\quad \quad \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \\
\text{if } y = y_{\text{old}} \\
\quad \text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]

- *i* for loop: update the assignments
- *j* for loop: update the cluster centers

Observe: these two loops are not nested
K-Means Algorithm

k-means(k, τ)
Init \( \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \)

for \( t = 1 \) to \( \tau \)

\( y_{old} = y \)

for \( i = 1 \) to \( n \)

\( y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \)

for \( j = 1 \) to \( k \)

\( \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \)

if \( y = y_{old} \)
break

return \( \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \)
**K-Means Algorithm**

\[
\text{k-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
\quad y_{old} = y \\
\text{for } i = 1 \text{ to } n \\
\quad y^{(i)} = \arg \min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\text{for } j = 1 \text{ to } k \\
\quad \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \\
\text{if } y = y_{old} \\
\quad \text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
K-Means Algorithm

\[
k\text{-means}(k, \tau)
\]
Init \( \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \)

for \( t = 1 \) to \( \tau \)

\[
y_{\text{old}} = y
\]

for \( i = 1 \) to \( n \)

\[
y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2
\]

for \( j = 1 \) to \( k \)

\[
\mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}}
\]

if \( y = y_{\text{old}} \)

break

return \( \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \)
K-Means Algorithm

\[
k\text{-means}(k, \tau)
\]
\[
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
\[
\text{for } t = 1 \text{ to } \tau
\]
\[
y_{\text{old}} = y
\]
\[
\text{for } i = 1 \text{ to } n
\]
\[
y^{(i)} = \text{arg min}_{j} \| x^{(i)} - \mu^{(j)} \|_2^2
\]
\[
\text{for } j = 1 \text{ to } k
\]
\[
\mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}}
\]
\[
\text{if } y = y_{\text{old}}
\]
\[
\text{break}
\]
\[
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
K-Means Algorithm

\[
k\text{-means}(k, \tau) \\
\text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \\
\text{for } t = 1 \text{ to } \tau \\
\quad y_{\text{old}} = y \\
\quad \text{for } i = 1 \text{ to } n \\
\quad \quad y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \\
\quad \text{for } j = 1 \text{ to } k \\
\quad \quad \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \\
\quad \text{if } y = y_{\text{old}} \\
\quad \quad \text{break} \\
\text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
\]
K-Means Algorithm

\[ \text{k-means}(k, \tau) \]
\[ \text{Init } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \]
\[ \text{for } t = 1 \text{ to } \tau \]
\[ y_{\text{old}} = y \]
\[ \text{for } i = 1 \text{ to } n \]
\[ y^{(i)} = \arg \min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2 \]
\[ \text{for } j = 1 \text{ to } k \]
\[ \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}} \]
\[ \text{if } y = y_{\text{old}} \]
\[ \text{break} \]
\[ \text{return } \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n} \]

Use centroid to cluster new data
K-Means Algorithm

K-means is sensitive to initialization!

K-means is guaranteed to converge with iterations, but not necessarily to the global minimum.
K-Means Algorithm

K-means is sensitive to initialization! And number of clusters

Why can’t we just increase K?
K-Means Algorithm

K-means is sensitive to initialization!
And number of clusters
And choice of distance metric

What are some issues with this “distance”?

```python
k-means(k, \tau)
Init \{\mu(j)\}_{j=1}^{k}, \{y(i)\}_{i=1}^{n}
for t = 1 to \tau
  y_{\text{old}} = y
  for i = 1 to n
    y^{(i)} = \arg \min_j \|x^{(i)} - \mu^{(j)}\|_2
  for j = 1 to k
    \mu^{(j)} = \frac{\sum_{i=1}^{n} 1\{y^{(i)} = j\}x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = j\}}
  if y = y_{\text{old}}
  break
return \{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}
```
K-Means Algorithm

unsupervised learning

clustering

k-means clustering

using k-means algorithm
K-Means Algorithm

K-means algorithm is just one way of optimizing the K-means objective

$$L(\mu) = \sum_{i=1}^{n} \min_{j} \| x^{(i)} - \mu^{(j)} \|_{2}^{2}$$

$$\arg\min_{\mu, y} \sum_{i=1}^{n} \sum_{j=1}^{k} 1\{y^{(i)} = j\} \| x^{(i)} - \mu^{(j)} \|_{2}^{2}$$

Can also just do gradient descent!
When to use K-Means?

K-means works well when:
- Data “circular”
When to use K-Means?

K-means works well when:
- Data “circular”
- Clusters have roughly the same size
When to use K-Means?

K-means works well when:
- Data “circular”
- Clusters have roughly the same size
- Clusters are well separated