CS 4700: Foundations of Artificial Intelligence

Spring 2020
Prof. Haym Hirsh

Lecture 34
May 4, 2020
Naïve Bayes: Laplace Smoothing

Laplace (or “Additive”) Smoothing

Estimate \( P(\tilde{x}_{\text{test},j} = v_a \mid c_l) \) using

\[
\frac{n_{jat} + \alpha}{N_l + \alpha |V_j|}
\]

\( \alpha > 0: \) “smoothing parameter”

(Often \( \alpha \leq 1 \))
Naïve Bayes: Laplace Smoothing

Attributes from discrete sets: Assign label $c$

$$c = \arg \max_{c \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{ja_jl} + \alpha}{N_l + \alpha |V_j|} \right]$$

Binary attributes case: Assign label $c$

$$c = \arg \max_{c \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \left( \frac{n_{j1l} + \alpha}{N_l + \alpha |V_j|} \right)^{x_j} \left( 1 - \frac{n_{j1l} + \alpha}{N_l + \alpha |V_j|} \right)^{1-x_j} \right]$$
Naïve Bayes: Laplace Smoothing

Attributes from discrete sets: Assign label $c$

$$c = \arg\max_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j a_j l} + \alpha}{N_l + \alpha |V_j|} \right]$$

Binary attributes case: Assign label $c$

$$c = \arg\max_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \left( \frac{n_{j1l} + \alpha}{N_l + 2\alpha} \right)^{x_j} \left( 1 - \frac{n_{j1l} + \alpha}{N_l + 2\alpha} \right)^{1-x_j} \right]_{|V_j|=2}$$
Machine Learning

• Most common approaches:
  • Supervised learning
  • Unsupervising learning
  • Reinforcement learning
Machine Learning

• Most common approaches:
  • Supervised learning
  • Unsupervised learning
  • Reinforcement learning ✔ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
  • Unsupervised learning
  • Reinforcement learning ✓ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
  • Unsupervised learning
  • Reinforcement learning ✓ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  • Unsupervised learning
  • Reinforcement learning ✓ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
  Chapter 21: Deep Learning
  https://en.wikipedia.org/wiki/Backpropagation
  • Unsupervised learning – Dimensionality reduction
  • Reinforcement learning ✓ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  • Unsupervised learning – Dimensionality reduction
  • Reinforcement learning ✓ - Chapters 17 and 22
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
  Chapter 21: Deep Learning
  https://en.wikipedia.org/wiki/Backpropagation
  • Unsupervised learning – Clustering
  • Reinforcement learning ✓ - Chapters 17 and 22
Clustering

(Somewhat) more formally:

Given:
• a set D of N examples D={\(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\)}
  where each \(\vec{x}_i\) is in some \(\mathcal{X}\) (often \(\mathcal{X} = \mathbb{R}^n\))

Find:
• k sets \(\{S_1, \ldots, S_k\}\) such that each \(S_i \subseteq D\) and \(\bigcup_{1}^{k} S_i = D\)
  (often: and for \(1 \leq i \neq j \leq k\), \(S_i \cap S_j = \emptyset\))

where \(\sum_{i=1}^{k} \sum_{\vec{x}_1, \vec{x}_2 \in S_i} D(\vec{x}_1, \vec{x}_2)\) is “small” and
\(\sum_{1 \leq i \neq j \leq k} \sum_{\vec{x}_1 \in S_i, \vec{x}_2 \in S_j} D(\vec{x}_1, \vec{x}_2)\) is “big”

Assign items in D to k (possibly disjoint) subsets of D so that items in the same set are “close” and items in two different sets are “far”
Machine Learning

• Most common approaches:
  • Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  • Unsupervised learning – Clustering
  • Reinforcement learning ✓ - Chapters 17 and 22
Looking Ahead

• Wednesday, May 6: Natural Language Processing
  Prof. Claire Cardie

• Friday, May 8: Computer Vision
  Grant Van Horn, Ph.D. (Lab of O)

• Monday, May 11: Societal implications of AI
  - What should you worry about
Clustering

• Reflects a human cognitive skill that we might like machines to emulate
Clustering

• Reflects a human cognitive skill that we might like machines to emulate
• Useful tool in image processing, machine learning, data analytics (“cluster analysis”), ...
Sample Applications

- Image processing – color encodings, image representations
- Market analysis – partition customers into coherent segments
- E-commerce – product categories, user categories
- Social network analysis – identify “communities”
- Natural language processing – lexical ambiguity
- Evolutionary biology – inferring phylogenetic trees
- Linguistics – inferring language families
Facebook stops letting advertisers target people interested in 'pseudoscience'

The move comes after an investigation found that Facebook was allowing advertisers to target ads at 78 million users interested in the topic.
Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
Clustering

• Partition points into $k$ sets such that
  • Any two points in the same set are similar
  • Any two points in different sets are dissimilar

• $k$ can be an external parameter or a value determined by the algorithm
Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm
- Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)
Clustering

• Partition points into k sets such that
  • Any two points in the same set are similar
  • Any two points in different sets are dissimilar

• k can be an external parameter or a value determined by the algorithm

• Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)

• Clusters can be disjoint, overlapping, or hierarchical
Clustering

• Partition points into k sets such that
  • Any two points in the same set are similar
  • Any two points in different sets are dissimilar

• k can be an external parameter or a value determined by the algorithm

• Partitions can be categorical (in/out) ("hard clustering") or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) ("soft clustering")

• Clusters can be disjoint, overlapping, or hierarchical

• Typically presumes the existence of a measure $D(\bar{x}_1, \bar{x}_2)$ giving a "distance" between two points $D(\bar{x}_1, \bar{x}_2)$ – such as Euclidean distance
Clustering

• Partition points into k sets such that
  • Any two points in the same set are similar
  • Any two points in different sets are dissimilar
• k can be an external parameter or a value determined by the algorithm
• Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)
• Clusters can be disjoint, overlapping, or hierarchical
• Typically presumes the existence of a measure $D(\overline{x}_1, \overline{x}_2)$ giving a “distance” between two points $D(\overline{x}_1, \overline{x}_2)$ – such as Euclidean distance
Clustering

(Somewhat) more formally:

Given:

• a set D of N examples D={x_1, x_2, ..., x_N}
  where each x_i is in some X (often X = R^n)

Find:

• k sets {S_1, ..., S_k} such that each S_i ⊆ D and \( \bigcup_{1}^{k} S_i = D \)
  (often: and for \( 1 \leq i \neq j \leq k \), \( S_i \cap S_j = \emptyset \))

where \( \sum_{i=1}^{k} \sum_{x_1, x_2 \in S_i} D(x_1, x_2) \) is “small” and

\( \sum_{1 \leq i \neq j \leq k} \sum_{x_1 \in S_i, x_2 \in S_j} D(x_1, x_2) \) is “big”
Clustering

(Somewhat) more formally:

Given:

• a set $D$ of $N$ examples $D=\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N\}$
  where each $\bar{x}_i$ is in some $\mathcal{X}$ (often $\mathcal{X} = \mathbb{R}^n$)

Find:

• $k$ sets $\{S_1, \ldots, S_k\}$ such that each $S_i \subseteq D$ and $\bigcup_{i=1}^{k} S_i = D$
  (often: and for $1 \leq i \neq j \leq k$, $S_i \cap S_j = \emptyset$)

where

$$\sum_{i=1}^{k} \sum_{\bar{x}_1, \bar{x}_2 \in S_i} D(\bar{x}_1, \bar{x}_2) \text{ is “small” and}$$

$$\sum_{1 \leq i \neq j \leq k} \sum_{\bar{x}_1 \in S_i, \bar{x}_2 \in S_j} D(\bar{x}_1, \bar{x}_2) \text{ is “big”}$$

Assign items in $D$ to $k$
(possibly disjoint) subsets of $D$ so that items in the same set are “close” and items in two different sets are “far”
Clustering Approaches

• Hierarchical
Clustering Approaches

• Hierarchical
  • Top down
Clustering Approaches

• Hierarchical
  • Top down
Clustering Approaches

• Hierarchical
  • Top down
  • Bottom up / "agglomerative"
Clustering Approaches

• Hierarchical
  • Top down
  • Bottom up
Clustering Approaches

- Hierarchical
  - Top down
  - Bottom up
- Distributional/centroid-based
  - k-means clustering
k-Mean Clustering

General idea
(k is given)

1. Pick $k$ points at random from $D$ - call them $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_k$ (“c” for “centroid”)
k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_k$ ("c" for "centroid")

2. Define $S_i = \{ \bar{x} \in D \mid \text{argmin}_i D(\bar{x}, \bar{c}_i) \}$ for each $1 \leq i \leq k$
   - this sets each $S_i$ to all points in D that are closer to $\bar{c}_i$ than any other centroid
k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_k$ (“c” for “centroid”)
2. Define $S_i = \{ \bar{x} \in D \mid \text{argmin}_i D(\bar{x}, \bar{c}_i) \}$ for each $1 \leq i \leq k$
   - this sets each $S_i$ to all points in D that are closer to $\bar{c}_i$ than any other centroid
3. Define $\bar{c}_i = “\text{average}”$ of all $s \in S_i$ for each $1 \leq i \neq j \leq k$
   - this resets each $\bar{c}_i$ to the center (“centroid”) of all of the points in $S_i$
k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $c_1, c_2, ..., c_k$ (“c” for “centroid”)
2. Define $S_i = \{ \bar{x} \in D \mid \text{argmin}_i D(\bar{x}, c_i) \}$ for each $1 \leq i \leq k$
   - this sets each $S_i$ to all points in D that are closer to $c_i$ than any other centroid
3. Define $\bar{c}_i = “\text{average”} \ of \ all \ s \in S_i \ for \ each \ 1 \leq i \neq j \leq k$
   - this resets each $\bar{c}_i$ to the center (“centroid”) of all of the points in $S_i$
4. If the $c_i$’s have changed from the previous iteration, go to 2, otherwise end
k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $c_1, c_2, ..., c_k$ ("c" for "centroid")
2. Define $S_i = \{ \bar{x} \in D \mid \text{argmin}_i D(\bar{x}, \bar{c}_i) \}$ for each $1 \leq i \leq k$
   - this sets each $S_i$ to all points in D that are closer to $\bar{c}_i$ than any other centroid
3. Define $\bar{c}_i = \text{"average"}$ of all $s \in S_i$ for each $1 \leq i \neq j \leq k$
   - this resets each $\bar{c}_i$ to the center ("centroid") of all of the points in $S_i$
4. If the $c_i$'s have changed from the previous iteration, go to 2, otherwise end

Example of “Expectation Maximization” (EM Algorithm)
k-means Clustering Algorithm

\text{For } i = 1 \text{ to } k
\quad \overline{c_i} = \text{random } \overline{x} \in D \text{ not already selected;}
\text{Repeat}
\quad \text{For } i = 1 \text{ to } k
\quad \quad S_i = \{ \overline{x} \in D \mid \text{argmin distance}(\overline{c_i}, \overline{x}) \} \\
\text{For } i = 1 \text{ to } k
\quad c_i = \frac{\sum_{x \in S_i} \overline{x}}{|S_i|}
\text{Until <stopping condition> } \text{[example: centroids don’t change]}
Initial Seeding

Initial Centroids
Initial Seeding

Data points closest to blue centroid

Data points closest to red centroid

Initial Centroids
Initial Seeding

Data points closest to blue centroid

Data points closest to red centroid

Initial Centroids

Compute new centroids for each cluster

https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805
**Initial Seeding**

Data points closest to blue centroid

Initial Centroids

Data points closest to red centroid

**After Round 1**

Compute new centroids for each cluster
**Initial Seeding**

- Initial Centroids
- Data points closest to blue centroid
- Data points closest to red centroid

**After Round 1**

- Compute new centroids for each cluster
- Data points closest to blue centroid
- Data points closest to red centroid

https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805
**Initial Seeding**

- **Initial Centroids**
- **Data points closest to blue centroid**
- **Data points closest to red centroid**

**After Round 1**

- **Data points closest to blue centroid**
- **Data points closest to red centroid**

*Compute new centroids for each cluster*
K-means clustering algorithm

An example 2-cluster run is shown with the clusters.

**Initial Seeding**
- Initial Centroids
- Data points closest to blue centroid
- Data points closest to red centroid

**After Round 1**
- Compute new centroids for each cluster
- Data points closest to blue centroid
- Data points closest to red centroid

**After Round 2**
- Compute new centroids for each cluster
K-means clustering algorithm: An example 2-cluster run is shown with the clusters.

Initial Seeding:
- Initial Centroids
- Data points closest to blue centroid
- Data points closest to red centroid

After Round 1:
- Compute new centroids for each cluster
- Data points closest to blue centroid
- Data points closest to red centroid

After Round 2:
- Compute new centroids for each cluster
- Data points closest to blue centroid
- Data points closest to red centroid

https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805
K-means clustering algorithm: An example 2-cluster run is shown with the clusters.
K-means clustering algorithm: An example 2-cluster run is shown.

1. **Initial Seeding**
   - Initial Centroids
   - Data points closest to blue centroid
   - Data points closest to red centroid

2. **After Round 1**
   - Compute new centroids for each cluster
   - Data points closest to blue centroid
   - Data points closest to red centroid

3. **After Round 2**
   - Compute new centroids for each cluster
   - Data points closest to blue centroid
   - Data points closest to red centroid

4. **After Round 3**
   - Compute new centroids for each cluster
   - Data points closest to blue centroid
   - Data points closest to red centroid

K-means Clustering Algorithm: An example 2-cluster run is shown with the clusters.

**Initial Seeding**
- Initial Centroids
- Data points closest to blue centroid
- Data points closest to red centroid

**After Round 1**
- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid
- Compute new centroids for each cluster

**After Round 2**
- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid

**After Round 3**
- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid

[Graph showing the process of K-means clustering with data points and centroid updates.]

[Source: https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805]
**Initial Seeding**

Initial Centroids

Data points closest to blue centroid

Data points closest to red centroid

**After Round 1**

Compute new centroids for each cluster

Data points closest to blue centroid

Data points closest to red centroid

Compute new centroids for each cluster

**After Round 2**

Data points closest to blue centroid

Data points closest to red centroid

Compute new centroids for each cluster

**After Round 3**

Data points closest to blue centroid

Data points closest to red centroid

Compute new centroids for each cluster

---

https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805
**Initial Seeding**

- Data points closest to blue centroid
- Data points closest to red centroid
- Initial Centroids

**After Round 1**

- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid
- Compute new centroids for each cluster

**After Round 2**

- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid
- Compute new centroids for each cluster

**After Round 3**

- Data points closest to blue centroid
- Compute new centroids for each cluster
- Data points closest to red centroid
- Unchanged

---

[Link: https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-isShown-with-the-clusters_fig3_268880805]
K-means clustering algorithm: An example 2-cluster run is shown with the clusters.
k-Mean Clustering

Many refinements:
• Picking starting points
• Convergence (random restarts)
• Probabilistic membership
  • Example: Data selected from k Gaussian distributions
    Figure out mean and S.D. of each distribution