Unsupervised learning

- So far we have reviewed some fundamentals, discussed Maximum Likelihood Estimation (MLE) for probabilistic models, and neural networks/backprop SGD
Unsupervised learning

• So far we have reviewed some fundamentals, discussed Maximum Likelihood Estimation (MLE) for probabilistic models, and neural networks/backprop SGD

• We have mostly considered *supervised* settings (implicitly) although the above methods are general; we will shift focus to *unsupervised* learning for a few weeks
Unsupervised learning

• So far we have reviewed some fundamentals, discussed Maximum Likelihood Estimation (MLE) for probabilistic models, and neural networks/backprop SGD

• We have mostly considered *supervised* settings (implicitly) although the above methods are general; we will shift focus to *unsupervised* learning for a few weeks

• Both the probabilistic and neural perspectives will continue to be relevant here — and we will consider the former explicitly for clustering next week
Clustering
Clustering

Unsupervised learning (no labels for training)
Group data into similar classes that
• Maximize *inter-cluster* similarity
• Minimize *intra-cluster* similarity
Clustering

Unsupervised learning (no labels for training)
Group data into similar classes that
• Maximize *inter-cluster* similarity
• Minimize *intra-cluster* similarity
What is a natural grouping?

Choice of clustering criterion can be task-dependent
What is a natural grouping?

Choice of clustering criterion can be task-dependent

Simpson’s Family

School Employees
What is a natural grouping?

Choice of clustering criterion can be task-dependent
Defining Distance Measures

Dissimilarity/distance: \( d(x_1, x_2) \)
Similarity: \( s(x_1, x_2) \)
Proximity: \( p(x_1, x_2) \)
Defining Distance Measures

Dissimilarity/distance: \( d(x_1, x_2) \)
Similarity: \( s(x_1, x_2) \)

Proximity: \( p(x_1, x_2) \)
Defining Distance Measures

Dissimilarity/distance: \( d(x_1, x_2) \)

Similarity: \( s(x_1, x_2) \)

Proximity: \( p(x_1, x_2) \)
Distance Measures

- Euclidean Distance
  \[ \sqrt{k \sum_{i=1}^{k} (x_i - y_i)^2} \]
Distance Measures

- **Euclidean Distance**
  \[ \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \]

- **Mahattan Distance**
  \[ \sum_{i=1}^{k} |x_i - y_i| \]
Distance Measures

- **Euclidean Distance**: \[ \sqrt{\left( \sum_{i=1}^{k} (x_i - y_i)^2 \right)} \]
- **Mahattan Distance**: \[ \sum_{i=1}^{k} |x_i - y_i| \]
- **Minkowski Distance**: \[ \left( \sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{\frac{1}{q}} \]

Classification Model: K-Nearest Neighbor (KNN)
Similarity over \textit{functions of inputs}

- The preceding measures are distances defined on the original input space $X$

- A better representation may be some function of these features $\phi(x)$
Similarity: *Kernels*

- **Linear (inner-product)**
  \[ k(x, x') = (\langle x, x' \rangle + c) \]

- **Polynomial**
  \[ k(x, x') = (\langle x, x' \rangle + c)^m \]

- **Radial Basis Function (RBF)**
  \[ k(x, x') = \exp\left(-\frac{1}{2}\gamma^{-2}\|x-x'\|^2\right) \]
Figure 12.10 SVM with different kernels. Note that while the decision boundary is nonlinear, the underlying problem being solved is for a linear separating hyperplane (albeit with a nonlinear kernel).

First feature
Second feature
(a) SVM with linear kernel

First feature
Second feature
(b) SVM with RBF kernel

First feature
Second feature
(c) SVM with polynomial (degree 2) kernel

First feature
Second feature
(d) SVM with polynomial (degree 3) kernel

There is a unique reproducing kernel Hilbert space associated with every kernel (Aronszajn, 1950; Berlinet and Thomas-Agnan, 2004). In this unique association, \((x) = k(·, x)\) is called the canonical feature map. The generalization from an inner product to a kernel function (12.52) is known as the kernel trick (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004), as it hides away the explicit non-linear feature map.

The matrix \(K_{2R^N \times R^N}\), resulting from the inner products or the application of \(k(·, ·)\) to a dataset, is called the Gram matrix, and is often just referred to as the kernel matrix. Kernels must be symmetric and positive semidefinite functions so that every kernel matrix \(K\) is symmetric and positive semidefinite (Section 3.2.3):

\[
\langle z \rangle_{R^N} > K \langle z \rangle_{R^N} > 0.
\] (12.53)

Some popular examples of kernels for multivariate real-valued data \(x_i \in \mathbb{R}^D\) are the polynomial kernel, the Gaussian radial basis function kernel, and the rational quadratic kernel (Schölkopf and Smola, 2002; Rasmussen).
Why kernels?

“The key insight in kernel-based learning is that you can rewrite many linear models in a way that doesn’t require you to ever explicitly compute $\phi(x)$

- Daume, CIML
Similarities vs Distance Measure

**Distance Measure**

- \( D(A, B) = D(B, A) \)  
  \( \text{Symmetry} \)
Similarities vs Distance Measure

*Distance Measure*

- $D(A, B) = D(B, A)$  
  *Symmetry*
- $D(A, A) \geq 0$  
  *Reflexivity*
Similarities vs Distance Measure

**Distance Measure**

- \( D(A, B) = D(B, A) \)  
  *Symmetry*
- \( D(A, A) \geq 0 \)  
  *Reflexivity*
- \( D(A, B) = 0 \iff A = B \)  
  *Positivity (Separation)*
Distance Measure

• $D(A, B) = D(B, A)$  \hspace{2cm} Symmetry
• $D(A, A) \geq 0$  \hspace{2cm} Reflexivity
• $D(A, B) = 0 \text{ iff } A = B$  \hspace{2cm} Positivity (Separation)
• $D(A, B) \leq D(A, C) + D(B, C)$  \hspace{2cm} Triangular Inequality
Similarities vs Distance Measure

**Distance Measure**

- \( D(A, B) = D(B, A) \) \quad Symmetry
- \( D(A, A) \geq 0 \) \quad Reflexivity
- \( D(A, B) = 0 \text{ iff } A = B \) \quad Positivity (Separation)
- \( D(A, B) \leq D(A, C) + D(B, C) \) \quad Triangular Inequality

**Similarity functions**

- Less formal; encodes some notion of similarity but not necessarily well defined
- Can be negative
- May not satisfy triangular inequality
Cosine similarity

\[
similarity(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}
\]
Four Types of Clustering

1. *Centroid-based (K-means, K-medoids)*
Four Types of Clustering

2. Connectivity-based (Hierarchical)

Notion of Clusters: Cut off dendrogram at some depth
Four Types of Clustering

3. Density-based (DBSCAN, OPTICS)

Notion of Clusters: Connected regions of high density
Four Types of Clustering

4. Distribution-based (Mixture Models)

Notion of Clusters: Distributions on features
K-Means clustering (board)
### K-means Algorithm

**Input:**

\[ X = \{x_1, x_2, \ldots, x_N\} \]

Number of clusters \( K \)

**Initialize:**

\( K \) random centroids \( \mu_1, \mu_2, \ldots, \mu_K \)
K-means Algorithm

Input: $X = \{x_1, x_2, \ldots, x_N\}$
Number of clusters $K$

Initialize: $K$ random centroids $\mu_1, \mu_2, \ldots, \mu_K$

Repeat Until Convergence

1. For $i = 1, \ldots, K$ do
   \begin{equation*}
   C_i = \{x \in X | i = \arg\min_{1 \leq j \leq K} \| x - \mu_j \|^2 \}
   \end{equation*}

Output: $C_1, C_2, \ldots, C_K$
**K-means Algorithm**

**Input:** \( X = \{x_1, x_2, \ldots, x_N\} \)
Number of clusters \( K \)

**Initialize:** \( K \) random centroids \( \mu_1, \mu_2, \ldots, \mu_K \)

**Repeat Until Convergence**

1. For \( i = 1, \ldots, K \) do
   \[
   C_i = \{x \in X | i = \arg\min_{1 \leq j \leq K} \| x - \mu_j \|^2\}
   \]

2. For \( i = 1, \ldots, K \) do
   \[
   \mu_i = \arg\min_z \sum_{x \in C_i} \| z - x \|^2
   \]
K-means Algorithm

Input: \( X = \{x_1, x_2, \ldots, x_N\} \)
Number of clusters \( K \)

Initialize: \( K \) random centroids \( \mu_1, \mu_2, \ldots, \mu_K \)

Repeat Until Convergence

1. For \( i = 1, \ldots, K \) do
   \[ C_i = \{x \in X | i = \text{arg min}_{1 \leq j \leq K} \| x - \mu_j \|^2\} \]

2. For \( i = 1, \ldots, K \) do
   \[ \mu_i = \text{arg min} \sum_{z \in C_i} \| z - x \|^2 \]

Output: \( C_1, C_2, \ldots, C_K \)
K-means Clustering

Randomly initialize $K$ centroids $\mu_k$
K-means Clustering

Assign each point to closest centroid, then update centroids to average of points
K-means Clustering

Assign each point to closest centroid, then update centroids to average of points
K-means Clustering

Repeat until convergence
(no points reassigned, means unchanged)
K-means Clustering

Repeat until convergence
(no points reassigned, means unchanged)
K-means Algorithm

Input: \(X = \{x_1, x_2, \ldots, x_N\}\)
Number of clusters \(K\)

Initialize: \(K\) random centroids \(\mu_1, \mu_2, \ldots, \mu_K\)

Repeat Until Convergence

1. For \(i = 1, \ldots, K\) do
   \(C_i = \{x \in X | i = \arg \min_{1 \leq j \leq K} ||x - \mu_j||^2\}\)

2. For \(i = 1, \ldots, K\) do
   \(\mu_i = \arg \min_{x \in C_i} \sum_{z} ||z - x||^2\)

Output: \(C_1, C_2, \ldots, C_K\)

- K-means: Set \(\mu\) to mean of points in \(C\)
- K-medoids: Set \(\mu=x\) for point in \(C\) with minimum SSE
Let's see some examples in Python
“Good” Initialization of Centroids

Iteration 1

Iteration 2

Iteration 3

Iteration 4

Iteration 5

Iteration 6
“Bad” Initialization of Centroids

Iteration 1

Iteration 2

Iteration 3

Iteration 4

Iteration 5
Example: 10 Clusters

Iteration 4

5 pairs of clusters, two initial points in each pair
Example: 10 Clusters

5 pairs of clusters, two initial points in each pair
Importance of Initial Centroids

Initialization tricks

• Use multiple restarts

• Initialize with hierarchical clustering

• Select more than K points, keep most widely separated points
Choosing K

K=1, SSE=873

K=2, SSE=173

K=3, SSE=134
Choosing K

We can plot the cost function values for $K$ equals 1 to 6...

The abrupt change at $K = 2$, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as "elbow finding" or "knee finding".

“Elbow finding” (a.k.a. “knee finding”)
Set $K$ to value just above “abrupt” increase
K-means Limitations: Differing Sizes

Original Points

K-means (3 clusters)
K-means Limitations: Different Densities

Original Points

K-means (3 clusters)
K-means Limitations: Non-globular Shapes

Original Points

K-means (2 clusters)
Overcoming K-means Limitations

Intuition: “Combine” smaller clusters into larger clusters

- One Solution: Hierarchical Clustering
- Another Solution: Density-based Clustering
K-means in action: Download the notebook starter for today from blackboard (and CSV file)
Density-based Clustering
DBSCAN

arbitrarily shaped clusters

oise

A density-based algorithm for discovering clusters in large spatial databases with noise.

M Ester, HP Kriegel, J Sander, X Xu - Kdd, 1996 - aaai.org

Abstract Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to ...

Cited by 8901 Related articles All 70 versions Cite Save More

(one of the most-cited clustering methods)
**Intuition**

- A *cluster* is a region of *high* density
- *Noise* points lie in regions of *low* density
Defining “High Density”

Naïve approach

For each point in a cluster there are at least a minimum number (MinPts) of points in an Eps-neighborhood of that point.
Defining “High Density”

Eps-neighborhood of a point p

\[ N_{Eps}(p) = \{ q \in D \mid \text{dist} (p, q) \leq Eps \} \]
Defining “High Density”

- In each cluster there are two kinds of points:
  - points inside the cluster (core points)
  - points on the border (border points)

An Eps-neighborhood of a border point contains significantly less points than an Eps-neighborhood of a core point.
Defining “High Density”

Better notion of cluster

For every point \( p \) in a cluster \( C \) there is a point \( q \in C \), so that

1. \( p \) is inside of the \( \varepsilon \)-neighborhood of \( q \)

and

2. \( N_{\varepsilon}(q) \) contains at least \( \text{MinPts} \) points.

Core points = high density

Border points are connected to core points

Parameter: \( \text{MinPts} = 5 \)

\( |N_{\varepsilon}(p)| = 6 \geq 5 = \text{MinPts} \) (core point condition)

\( |N_{\varepsilon}(q)| = 4 < 5 = \text{MinPts} \) (core point condition)
Density Reachability

Definition

A point $p$ is **directly density-reachable** from a point $q$ with regard to the parameters $Eps$ and $MinPts$, if

1) $p \in N_{Eps}(q)$  \hspace{1cm} \text{(reachability)}

2) $|N_{Eps}(q)| \geq MinPts$  \hspace{1cm} \text{(core point condition)}

Remark

Directly density-reachable is symmetric for pairs of core points.

It is not symmetric if one core point and one border point are involved.
Density Reachability

**Definition**

A point $p$ is **directly density-reachable** from a point $q$ with regard to the parameters $\text{Eps}$ and $\text{MinPts}$, if

1) $p \in N_{\text{Eps}}(q)$  
   (reachability)

2) $|N_{\text{Eps}}(q)| \geq \text{MinPts}$  
   (core point condition)

**Parameter:** $\text{MinPts} = 5$

$p$ directly density reachable from $q$

$p \in N_{\text{Eps}}(q)$

$|N_{\text{Eps}}(q)| = 6 \geq 5 = \text{MinPts}$  
(core point condition)

$q$ **not** directly density reachable from $p$

$|N_{\text{Eps}}(p)| = 4 < 5 = \text{MinPts}$  
(core point condition)

**Note:** This is an asymmetric relationship
Density Reachability

**Definition**

A point $p$ is **density-reachable** from a point $q$ with regard to the parameters $Eps$ and $MinPts$ if there is a chain of points $p_1, p_2, \ldots, p_s$ with $p_1 = q$ and $p_s = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$ for all $1 < i < s-1$.

- $MinPts = 5$
- $|N_{Eps}(q)| = 5 = MinPts$ (core point condition)
- $|N_{Eps}(p_1)| = 6 \geq 5 = MinPts$ (core point condition)
Density Connectivity

Definition (density-connected)

A point \( p \) is density-connected to a point \( q \) with regard to the parameters Eps and MinPts if there is a point \( v \) such that both \( p \) and \( q \) are density-reachable from \( v \).

\[
\text{MinPts} = 5
\]

Remark: Density connectivity is a symmetric relation.

Note: This is a symmetric relationship.
Definition of a Cluster

A cluster with regard to the parameters Eps and MinPts is a non-empty subset $C$ of the database $D$ with

1) For all $p, q \in D$:
   
   If $p \in C$ and $q$ is density-reachable from $p$ with regard to the parameters Eps and MinPts, then $q \in C$.  
   
   (Maximality)

2) For all $p, q \in C$:

   The point $p$ is density-connected to $q$ with regard to the parameters Eps and MinPts.
   
   (Connectivity)
Definition of Noise

Let $C_1, \ldots, C_k$ be the clusters of the database $D$ with regard to the parameters $Eps_i$ and $MinPts_i$ ($i=1, \ldots, k$).

The set of points in the database $D$ not belonging to any cluster $C_1, \ldots, C_k$ is called noise:

$$\text{Noise} = \{ p \in D \mid p \notin C_i \text{ for all } i = 1, \ldots, k \}$$
(1) Start with an arbitrary point p from the database and retrieve all points density-reachable from p with regard to Eps and MinPts.
DBSCAN Algorithm

(1) Start with an arbitrary point $p$ from the database and retrieve all points density-reachable from $p$ with regard to Eps and MinPts.

(2) If $p$ is a core point, the procedure yields a cluster with regard to Eps and MinPts and all points in the cluster are classified.
DBSCAN Algorithm

(1) Start with an arbitrary point \( p \) from the database and retrieve all points density-reachable from \( p \) with regard to Eps and MinPts.

(2) If \( p \) is a core point, the procedure yields a cluster with regard to Eps and MinPts and all points in the cluster are classified.

(3) If \( p \) is a border point, no points are density-reachable from \( p \) and DBSCAN visits the next unclassified point in the database.
DBSCAN Algorithm

Original Points

Point types: core, border and noise

When DBSCAN Works Well

- Resistant to Noise
- Can handle clusters of different shapes and sizes
$\text{DBSCAN strengths}$

- Resistant to noise
- Can handle arbitrary shapes
DBSCAN Weaknesses

- Varying densities
- High-dimensional data

Sensitive to hyperparameters
Let’s see what it does with Trump’s tweets...