Today

- Unsupervised learning
- Definition and properties
- A few clustering models
- Other instantiations of unsupervised learning
Experience (E)

- What data does $f$ experience?
  - (Focus on algorithms that experience whole datasets)
  - **Unsupervised.** Examples alone $\{X_i\}_{i=0}^n$
  - **Supervised.** Examples come with labels $\{(X_i, Y_i)\}_{i=0}^n$
1. Unsupervised

\[ \{x_i\}_{i=0}^{n} \]

- Experience examples alone
- Learn “useful properties of the structure of the data”
- E.g., clustering, density modeling (p(x)), PCA, FA.
Different tasks

- Finding patterns
- Clustering
- Dimensionality reduction
- Density modelling
- ...
Clustering

\[ f : X \rightarrow \{1, \ldots, K\} \]

- Task: Assign each point to one of K clusters
  - Cluster: a set of similar points (a group)
  - Alternatively: Divide the space into K regions. Assign points to a cluster based on the region they lie in
  - Similar to classification.
Clustering

- Desideratum: group similar points together
- Similarity is often defined as being close according to some similarity or distance function
- E.g., in Euclidean space

\[
\text{distance}(x_i, x_j) = \| x_i - x_j \|^2
\]
K-means clustering

- A particular clustering model (and accompanying algorithm)

- For a particular number of clusters (K), find cluster centers $\mu_k$ that minimize the within cluster distance

Objective $:= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2$

$$r = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix}_{N \times K}$$
K-means clustering

- A particular clustering model (and accompanying algorithm)
- For a particular number of clusters (K), find cluster centers $\mu_k$ that minimize the within cluster distance

$$\text{Objective} := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2$$

- Algorithm to minimize the objective:
  - Initialize the cluster centers
  - Until convergence:
    - Update $r$
    - Update cluster centers $\mu_k \ \forall k$
K-means clustering

• A particular clustering model (and accompanying algorithm)

• For a particular number of clusters (K), find cluster centers $\mu_k$ that minimize the within cluster distance

$$
\text{Objective} := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2
$$

• Algorithm to minimize the objective:

  • Initialize the cluster centers
  
  • Until convergence:

    • Update $r$

    • Update cluster centers $\mu_k \ \forall k$

\[ r = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix}_{N \times K} \]
K-means clustering

- A particular clustering model (and accompanying algorithm)

- For a particular number of clusters (K), find cluster centers $\mu_k$ that minimize the within cluster distance

$$\text{Objective} := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2$$

- Algorithm to minimize the objective:
  - Initialize the cluster centers
  - Until convergence:
    - Update $r$
    - Update cluster centers $\mu_k$ $\forall k$
K-means clustering

- A particular clustering model (and accompanying algorithm)

- For a particular number of clusters (K), find cluster centers $\mu_k$ that minimize the within cluster distance

  \[
  \text{Objective} := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2
  \]

- Algorithm to minimize the objective:
  - Initialize the cluster centers
  - Until convergence:
    - Update $r$
    - Update cluster centers $\mu_k \ \forall k$
Figure 9.1 from PRML
• Initialize the cluster centers

• Until convergence:

1. Update r

2. Update cluster centers
K-means on images

Each pixel is a datum

\[ x_i = (r_i, g_i, b_i) \]
K-means on images

Each pixel is a datum

\[ x_i = (r_i, g_i, b_i) \]

- We cluster pixels by color
- I.e., pixels of similar color will be belong to the same cluster
- We re-draw the image by replacing the color of each pixel by the color of its cluster
A few failure cases

- Only works with squared Euclidean distance
- Not robust to outliers
- What about non-continuous data?

K-medoids Objective:\n\[ \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} d(x_i, \mu_k) \]

- Tends to result in relatively uniform cluster sizes
- Hard assignments

[http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html]
<table>
<thead>
<tr>
<th>MiniBatchKMeans</th>
<th>AffinityPropagation</th>
<th>MeanShift</th>
<th>SpectralClustering</th>
<th>Ward</th>
<th>AgglomerativeClustering</th>
<th>DBSCAN</th>
<th>Birch</th>
<th>GaussianMixture</th>
</tr>
</thead>
<tbody>
<tr>
<td>.01s</td>
<td>4.58s</td>
<td>.10s</td>
<td>1.56s</td>
<td>15s</td>
<td>.14s</td>
<td>.01s</td>
<td>.06s</td>
<td>.01s</td>
</tr>
<tr>
<td>.02s</td>
<td>5.01s</td>
<td>.07s</td>
<td>2.99s</td>
<td>22s</td>
<td>.15s</td>
<td>.01s</td>
<td>.06s</td>
<td>.01s</td>
</tr>
<tr>
<td>.03s</td>
<td>2.94s</td>
<td>.13s</td>
<td>.39s</td>
<td>1.04s</td>
<td>.78s</td>
<td>.02s</td>
<td>.06s</td>
<td>.02s</td>
</tr>
<tr>
<td>.02s</td>
<td></td>
<td>.10s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.02s</td>
<td>2.20s</td>
<td>.07s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.02s</td>
<td>2.13s</td>
<td>.10s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html]
Unsupervised learning

- In supervised learning you have:
  - Clear metric (e.g., mean squared error, accuracy, etc.)
  - Procedures for comparing different models

- Unsupervised learning
  - Unclear what the right metric is
  - How to compare different models?

- Data dimensions can have a big impact on solution
  - You should carefully select the input
A probabilistic approach to k-means clustering

Objective := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2
A probabilistic approach to k-means clustering

Objective := \[ \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2 \]
A probabilistic approach to k-means clustering

Objective := \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2
A probabilistic approach to k-means clustering

Objective: \[ \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| \mathbf{x}_i - \mu_k \|^2 \]

where:

- \( r \) is an indicator matrix:
  \[ r = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ \vdots \\ 0 & 1 \end{bmatrix}_{N \times K} \]
  
- \( r_{i.} \) is the i'th line:
  \[ r_{i.} = \begin{bmatrix} 0 & 1 \end{bmatrix}_{1 \times K} \]
A probabilistic approach to k-means clustering

Objective: \[ \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2 \]

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\vdots & \\
0 & 1
\end{bmatrix}_{N \times K}
\]

the i'th line \[ r_{i:} = \begin{bmatrix} 0 & 1 \end{bmatrix}_{1 \times K} \]

\[ P(r_{i:}) = \text{Categorical}(\pi) \]

\[ = \prod_{k=1}^{K} \pi_{ik}^r \]
A probabilistic approach to k-means clustering

Objective := \( \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \parallel x_i - \mu_k \parallel^2 \)

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\vdots \\
0 & 1
\end{bmatrix}_{N \times K}
\]

the i'th line \( r_{i} = [0 \ 1]_{1 \times K} \)

\[
P(r_{i}) = \text{Categorical}(\pi)
\]

= \( \prod_{k=1}^{K} \pi_{ik} \)}
A probabilistic approach to k-means clustering

Objective := \( \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|^2 \)

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\vdots \\
0 & 1
\end{bmatrix}_{N \times K}
\]

the i'th line

\[ r_{i.} = \begin{bmatrix} 0 & 1 \end{bmatrix}_{1 \times K} \]

\[ x_i = \begin{bmatrix} 0.2 & 0.4 \end{bmatrix} \]

\[ \mu_k = \begin{bmatrix} 0.3 & 0.1 \end{bmatrix} \]

\[ P(r_{i.}) = \text{Categorical}(\pi) \]

\[ P(x_i | r_{ik} = 1) = \mathcal{N}(x_i | \mu_k, \Sigma_k) \]
A probabilistic approach to k-means clustering

\[ P(r_i) = \text{Categorical}(\pi) \]

\[ P(x_i \mid r_{ik} = 1) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \]

\[
P(x_i \mid \theta) = \sum_{r_i} P(r_i \mid \theta)P(x_i \mid r_i, \theta)
\]

\[
= \sum_{k} \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \quad \theta := \{\mu_k, \Sigma_k, \pi_k\}, \forall k
\]

\[
P(r_i = 1 \mid x)
\]
A probabilistic approach to k-means clustering

\[
P(r_i) = \text{Categorical}(\pi)
\]
\[
P(x_i \mid r_{ik} = 1) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k)
\]

\[
P(x_i \mid \theta) = \sum_{r_i} P(r_i \mid \theta)P(x_i \mid r_i, \theta)
\]

\[
= \sum_k \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \quad \theta := \{\mu_k, \Sigma_k, \pi_k\}, \forall k
\]

\[
0 \leq \pi \leq 1
\]
\[
\sum_k \pi_k = 1
\]

\[
P(r_i = 1 \mid x)
\]
A probabilistic approach to k-means clustering

\[ P(r_i) = \text{Categorical}(\pi) \]
\[ P(x_i \mid r_{ik} = 1) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \]

\[ P(x_i \mid \theta) = \sum_{r_i} P(r_i \mid \theta) P(x_i \mid r_i, \theta) \]
\[ = \sum_k \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \quad \theta := \{\mu_k, \Sigma_k, \pi_k\}, \forall k \]

1) Soft clustering

Each cluster \(k\) explains \(\pi_k\) of each datapoint

The posterior over \(r\) (\(P(r_i = 1 \mid x)\)) makes it explicit

\[ 0 \leq \pi \leq 1 \]
\[ \sum_k \pi_k = 1 \]
A probabilistic approach to k-means clustering

\[ P(r_i) = \text{Categorical}(\pi) \]
\[ P(x_i \mid r_{ik} = 1) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \]

\[ P(x_i \mid \theta) = \sum_{r_i} P(r_i \mid \theta) P(x_i \mid r_i, \theta) \]
\[ = \sum_k \pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \quad \theta := \{\mu_k, \Sigma_k, \pi_k\}, \forall k \]

1) **Soft clustering**
Each cluster \( k \) explains \( \pi_k \) of each datapoint
The posterior over \( r \) (\( P(r_i = 1 \mid x) \)) makes it explicit

2) **Mixture of Gaussians (GMMs)**
Convex combination of Gaussian distributions

\[ 0 \leq \pi \leq 1 \]
\[ \sum_k \pi_k = 1 \]
A probabilistic approach to k-means clustering

- Estimation of the parameters: Maximum likelihood estimate
  - Could do it jointly ("a la" neural network)
  - Often in two separate steps (similar as the non-probabilistic version)
    - This leads to the Expectation-Maximization (EM) algorithm
Initial cluster centers

Final solution

[Figure 9.8 from PRML]
K-means

Similar

GMM better

GMM better

Similar

Similar

GMMs

Similar

GMM better

GMM better

Similar

Similar
Comparing K-means to GMMs

- GMMs learns covariance matrix
- Per cluster variance
- Covariance terms
- GMMs has many more parameters
  - Covariance matrix (MxM)
Beyond Clustering
Beyond Clustering

- Clustering is the classical task
Beyond Clustering

• Clustering is the classical task

• What’s the usefulness of unsupervised learning?
Beyond Clustering

• Clustering is the classical task

• What’s the usefulness of unsupervised learning?

  • Data exploration
Beyond Clustering

- Clustering is the classical task
- What’s the usefulness of unsupervised learning?
  - Data exploration
  - Dimensionality reduction
Beyond Clustering

- Clustering is the classical task
- What’s the usefulness of unsupervised learning?
  - Data exploration
  - Dimensionality reduction
  - Visualization
Beyond Clustering

- Clustering is the classical task

- What's the usefulness of unsupervised learning?
  - Data exploration
  - Dimensionality reduction
  - Visualization
  - Pre-processing
Beyond Clustering

• Clustering is the classical task

• What’s the usefulness of unsupervised learning?
  • Data exploration
  • Dimensionality reduction
    • Visualization
  • Pre-processing
  • Learn useful representations
Beyond Clustering

- Clustering is the classical task
- What’s the usefulness of unsupervised learning?
  - Data exploration
  - Dimensionality reduction
    - Visualization
  - Pre-processing
  - Learn useful representations
  - Pre-training / semi-supervised learning
Autoencoders

• A type of neural network for dimensionality reduction*

• Non-linear PCA
Autoencoders

• A type of neural network for dimensionality reduction*
  
  • Non-linear PCA
  
  • Intuition: let’s learn to copy the data

\[ x' = f(x) \]
Autoencoders

• A type of neural network for dimensionality reduction*

• Non-linear PCA

• Intuition: let’s learn to copy the data

\[ x' = f(x) \]

• We force a “bottleneck”

\[ z = f_1(x) \]
\[ x' = f_2(z) \]
\[ \text{dim}(z) < \text{dim}(x) \]
Autoencoders

- A neural network architecture for unsupervised learning
Unsupervised learning as supervised learning

- Examples:
  - Auto-encoders “predict” their inputs
  - Language models “predict” the next word
- Create a target from the x’s (or a subset)
- Find a task to ensure that you learn something useful
- Self-supervised learning
Some thoughts on probabilistic approaches

- Many “equivalences” from known models/algorithms to probabilistic formulations
  - K-means -> Mixture of Gaussians
  - Linear Regression -> MLE in Gaussian model
  - Logistic Regression -> MLE in Bernoulli model
- Allows to think of models in a common framework
  - Build a joint distribution from a marginal and a conditional
    \[ P(x, z) = P(x \mid z)P(z) \]
- Learning procedures
  - Well motivated
  - Can be shared across problems
Unsupervised learning takeaways

• Most data (in the world) is unlabeled

• Useful tasks: clustering, density estimation, dimensionality reduction

• K-means and Gaussian mixtures (GMMs)

• Performance is harder to judge (measure)

• Note that all our examples were in 2D

• Can be used in downstream tasks