Recap: Classification

- Classification systems:
  - Supervised learning
  - Make a rational prediction given evidence
  - We’ve seen several methods for this
  - Useful when you have labeled data (or can get it)

  E.g. Perceptron, Neural Net, Nearest-Neighbor, Support Vector Machine

Clustering

- Clustering systems:
  - Unsupervised learning
  - Detect patterns in unlabeled data
    - E.g. group emails or search results
    - E.g. find categories of customers
    - E.g. detect anomalous program executions
  - Useful when you don’t know what you’re looking for
  - Requires data, but no labels
  - Often get gibberish

K-Means

- An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest mean
    - Assign each mean to the average of its assigned points
  - Stop when no points’ assignments change

K-Means Example
Example: K-Means

- [web demos]

K-Means as Optimization

- Consider the total distance to the means:
  \[ \phi(x_i, a_i, c_k) = \sum_i \text{dist}(x_i, c_k) \]
- Each iteration reduces \( \phi \)
  - Two stages each iteration:
    - Update assignments: fix means \( c \), change assignments \( a \)
    - Update means: fix assignments \( a \), change means \( c \)

Phase I: Update Assignments

- For each point, re-assign to closest mean:
  \[ a_i = \arg \min_k \text{dist}(x_i, c_k) \]
- Can only decrease total distance \( \phi \)!
  \[ \phi(x_i, a_i, c_k) = \sum_i \text{dist}(x_i, c_{a_i}) \]

Phase II: Update Means

- Move each mean to the average of its assigned points:
  \[ c_k = \frac{1}{|\{i : a_i = k\}|} \sum_{i : a_i = k} x_i \]
  - Also can only decrease total distance… (Why?)
- Fun fact: the point \( y \) with minimum squared Euclidean distance to a set of points \( \{x\} \) is their mean

Initialization

- K-means is non-deterministic
  - Requires initial means
  - It does matter what you pick!
  - What can go wrong?
  - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics

K-Means Getting Stuck

- A local optimum:

Why doesn’t this work out like the earlier example, with the purple taking over half the blue?
K-Means Questions

• Will K-means converge?
  • To a global optimum?

• Will it always find the true patterns in the data?
  • If the patterns are very very clear?

• Will it find something interesting?
  • Do people ever use it?
  • How many clusters to pick?

Clustering for Segmentation

• Quick taste of a simple vision algorithm

  • Idea: break images into manageable regions for visual
    processing (object recognition, activity detection, etc.)

Representing Pixels

• Basic representation of pixels:
  • 3 dimensional color vector <r, g, b>
  • Range r, g, b in [0, 1]
  • What will happen if we cluster the pixels in an
    image using this representation?

• Improved representation for segmentation:
  • 5 dimensional vector <r, g, b, x, y>
  • Range x in [0, M], y in [0, N]
  • Bigger M, N makes position more important
  • How does this change the similarities?

  • Note: real vision systems use more sophisticated
    encodings which can capture intensity, texture,
    shape, and so on.

K-Means Segmentation

• Results depend on initialization!
  • Why?

  • Note: best systems use graph segmentation algorithms

Other Uses of K-Means

• Speech recognition: can use to quantize wave slices
  into a small number of types (SOTA: work with
  multivariate continuous features)

• Document clustering: detect similar documents on
  the basis of shared words (SOTA: use probabilistic
  models which operate on topics rather than words)

EM Algorithm

• Like soft K-means

• Don’t make hard assignments of points to
  clusters: Now have distribution over
  different clusters

• Update of means is weighted
  by distribution

<table>
<thead>
<tr>
<th>Data</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>0.05</td>
<td>0.01</td>
<td>0.9</td>
<td>0.88</td>
<td>0.08</td>
<td>0.02</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>0.9</td>
<td>0.85</td>
<td>0.93</td>
<td>0.02</td>
<td>0.07</td>
<td>0.08</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>0.05</td>
<td>0.05</td>
<td>0.07</td>
<td>0.1</td>
<td>0.85</td>
<td>0.9</td>
</tr>
</tbody>
</table>
**Intuition of EM**

- **E-step**: Compute a distribution on the labels of the points, using current parameters.
- **M-step**: Update parameters using current guess of label distribution.

**Uses of EM**

- Used in many machine learning settings
- Learning transition & emission probabilities in HMMs
- In object recognition in computer vision
- Text document analysis

**Agglomerative Clustering**

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters
- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there’s only one cluster left
- Produces not one clustering, but a family of clusterings represented by a dendrogram

**Collaborative Filtering**

- Ever wonder how online merchants decide what products to recommend to you?
- Simplest idea: recommend the most popular items to everyone
  - Not entirely crazy! (Why)
  - Can do better if you know something about the customer (e.g., what they’ve bought)
- Better idea: recommend items that similar customers bought
  - A popular technique: collaborative filtering
  - Define a similarity function over customers (e.g., how similar are they?)
  - Look at purchases made by people with high similarity
  - Trade-off: relevance of comparison set vs confidence in predictions
  - How can this go wrong?