ECE 5424: Introduction to Machine Learning

Topics:
- Unsupervised Learning: Kmeans, GMM, EM

Readings: Barber 20.1-20.3

Stefan Lee
Virginia Tech
Tasks

Supervised Learning

- **Classification** (x $\rightarrow$ y, Discrete)
- **Regression** (x $\rightarrow$ y, Continuous)

Unsupervised Learning

- **Clustering** (x $\rightarrow$ c, Discrete ID)
- **Dimensionality Reduction** (x $\rightarrow$ z, Continuous)
Unsupervised Learning

- Learning only with X
  - Y not present in training data

- Some example unsupervised learning problems:
  - Clustering / Factor Analysis
  - Dimensionality Reduction / Embeddings
  - Density Estimation with Mixture Models
New Topic: Clustering
Synonyms

- Clustering
- Vector Quantization
- Latent Variable Models
- Hidden Variable Models
- Mixture Models

- Algorithms:
  - K-means
  - Expectation Maximization (EM)
Some Data
K-means

1. Ask user how many clusters they’d like.
   *(e.g. \( k=5 \))
K-means

1. Ask user how many clusters they’d like.  
   \textit{(e.g. k=5)}

2. Randomly guess k cluster Center locations
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it’s closest to. (Thus each Center “owns” a set of datapoints)
**K-means**

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it’s closest to.

4. Each Center finds the centroid of the points it owns
K-means

1. Ask user how many clusters they’d like.  
   *(e.g. k=5)*

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it’s closest to.

4. Each Center finds the centroid of the points it owns

5. …Repeat until terminated!
K-means

• Randomly initialize $k$ centers
  - $(0) = 1^{(0)}, \ldots, k^{(0)}$

• Assign:
  - Assign each point $i \in \{1, \ldots, n\}$ to nearest center:
    - $C(i) \leftarrow \arg\min_j ||x_i - \mu_j||^2$

• Recenter:
  - $\mu_j$ becomes centroid of its points
K-means

• Demo
  – http://mlehman.github.io/kmeans-javascript/
What is K-means optimizing?

• Objective $F(\mu, C)$: function of centers and point allocations $C$:

  $F(\mu, C) = \sum_{i=1}^{N} ||x_i - \mu_{C(i)}||^2$

  1-of-$k$ encoding $F(\mu, a) = \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||x_i - \mu_j||^2$

• Optimal K-means:

  $\min \min_{a} F(\mu, a)$
Coordinate descent algorithms

• Want: \( \min_a \min_b F(a, b) \)

• Coordinate descent:
  – fix a, minimize b
  – fix b, minimize a
  – repeat

• Converges!!!
  – if F is bounded
  – to a (often good) local optimum
    • as we saw in applet (play with it!)

• K-means is a coordinate descent algorithm!
K-means as Co-ordinate Descent

• Optimize objective function:

\[
\min_{\mu_1, \ldots, \mu_k} \min_{a_1, \ldots, a_N} F(\mu, a) = \min_{\mu_1, \ldots, \mu_k} \min_{a_1, \ldots, a_N} \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||x_i - \mu_j||^2
\]

• Fix \(a\), optimize \(\mu\) (or \(C\))
K-means as Co-ordinate Descent

- Optimize objective function:

\[
\min_{\mu_1, \ldots, \mu_k} \min_{a_1, \ldots, a_N} F(\mu, a) = \min_{\mu_1, \ldots, \mu_k} \min_{a_1, \ldots, a_N} \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||x_i - \mu_j||^2
\]

- Fix a (or C), optimize
One important use of K-means

- Bag-of-word models in computer vision
Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.
Interest Point Features

Detect patches

[Mikojaczyk and Schmid '02]
[Matas et al. '02]
[Sivic et al. '03]

Compute SIFT descriptor

[Lowe'99]

Normalize patch

Slide credit: Josef Sivic
dictionary formation
Clustering (usually k-means)
Clustered Image Patches

Fei-Fei et al. 2005
Image representation

frequency

codewords

Fei-Fei Li
(One) bad case for k-means

- Clusters may overlap
- Some clusters may be “wider” than others
- GMM to the rescue!
Recall Multi-variate Gaussians
Hidden Data Causes Problems #1

- Fully Observed (Log) Likelihood factorizes
- Marginal (Log) Likelihood doesn’t factorize
- All parameters coupled!
GMM vs Gaussian Joint Bayes Classifier

• On Board
  – Observed Y vs Unobserved Z
  – Likelihood vs Marginal Likelihood
Hidden Data Causes Problems #2

Figure Credit: Kevin Murphy
Hidden Data Causes Problems #2

- Identifiability

Figure Credit: Kevin Murphy

(C) Dhruv Batra

Figure Credit: Kevin Murphy

35
Hidden Data Causes Problems #3

• Likelihood has singularities if one Gaussian “collapses”
Special case: spherical Gaussians and hard assignments

• If \( P(X|Z=k) \) is spherical, with same for all classes:

\[
P(x_i | z = j) \propto \exp \left[ -\frac{1}{2\sigma^2} \|x_i - \mu_j\|^2 \right]
\]

• If each \( x_i \) belongs to one class \( C(i) \) (hard assignment), marginal likelihood:

\[
\prod_{i=1}^{N} \sum_{j=1}^{k} P(x_i, y = j) \propto \prod_{i=1}^{N} \exp \left[ -\frac{1}{2\sigma^2} \|x_i - \mu_{C(i)}\|^2 \right]
\]

• \( \text{M(M)LE same as K-means!!!} \)
The K-means GMM assumption

- There are k components
- Component $i$ has an associated mean vector $\mu_i$
The K-means GMM assumption

- There are $k$ components
- Component $i$ has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:
The K-means GMM assumption

- There are $k$ components
- Component $i$ has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:

1. Pick a component at random:
   Choose component $i$ with probability $P(y=i)$
The K-means GMM assumption

- There are k components
- Component i has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:

1. Pick a component at random: Choose component i with probability $P(y=i)$
2. Datapoint $\sim N(\mu_i, \sigma^2 I)$
The **General GMM assumption**

- There are k components
- Component i has an associated mean vector $m_i$
- Each component generates data from a Gaussian with mean $m_i$ and covariance matrix $\Sigma_i$

Each data point is generated according to the following recipe:

1. Pick a component at random:
   Choose component i with probability $P(y=i)$
2. Datapoint $\sim N(m_i, \Sigma_i)$