Clustering
K-means

Machine Learning – CSE546
Emily Fox
University of Washington
November 4, 2013

Clustering images

Set of Images

organize into coherent “themes”

key: no labels given

©Carlos Guestrin 2005-2013

[Goldberger et al.]
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**

- Randomly initialize k centers
  - \( \mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)} \) (or “smartly”)

- **Classify**: Assign each point \( j \in \{1, \ldots, N\} \) to nearest center:
  - \( C^{(t)}(j) \leftarrow \arg \min_i ||\mu_i - x_j||^2 \)

- **Recenter**: \( \mu_i \) becomes centroid of its point:
  - \( \mu_i^{(t+1)} \leftarrow \arg \min_j \sum_{x : C(j) = i} ||\mu - x_j||^2 \)
  - Equivalent to \( \mu_i \leftarrow \text{average of its points!} \)

©Carlos Guestrin 2005-2013
Mixtures of Gaussians

Machine Learning – CSE546
Emily Fox
University of Washington
November 4, 2013

(One) bad case for k-means

- Clusters may overlap
- Some clusters may be “wider” than others

So centers alone don’t tell the whole story
Density as Mixture of Gaussians

Approximate density with a mixture of Gaussians

\[ p(x^i | \pi, \mu, \Sigma) = \sum_{k=1}^{K} \pi_k N(x^i | \mu_k, \Sigma_k) \]

Clustering our Observations

Imagine we have an assignment of each \( x^i \) to a Gaussian

Complete data labeled by true cluster assignments

Our actual observations

C. Bishop, *Pattern Recognition & Machine Learning*
Clustering our Observations

Imagine we have an assignment of each $x^i$ to a Gaussian.

Introduce latent cluster indicator variable $z^i$.

Then we have

$$p(x^i | z^i=k, \pi, \mu, \Sigma) = \mathcal{N}(x^i | \mu_k, \Sigma_k)$$

We must infer the cluster assignments from the observations.

Posterior probabilities of assignments to each cluster given model parameters:

$$r_{ik} = p(z^i = k | x^i, \pi, \mu, \Sigma) = \frac{\pi_k \mathcal{N}(x^i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x^i | \mu_j, \Sigma_j)}$$

motivates an iterative alg.
Unsupervised Learning: not as hard as it looks

- Sometimes easy
- Sometimes impossible
- and sometimes in between

Summary of GMM Concept

- Estimate a density based on $x^1, \ldots, x^N$

$$p(x^i | \pi, \mu, \Sigma) = \sum_{z^i = 1}^{K} \pi_{z^i} N(x^i | \mu_{z^i}, \Sigma_{z^i})$$

- Complete data labeled by true cluster assignments
- Surface Plot of Joint Density, Marginalizing Cluster Assignments
Summary of GMM Components

- Observations: \( x^i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N \)
- Hidden cluster labels: \( z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N \)
- Hidden mixture means: \( \mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K \)
- Hidden mixture covariances: \( \Sigma_k \in \mathbb{R}^{d \times d}, \quad k = 1, 2, \ldots, K \)
- Hidden mixture probabilities: \( \pi_k, \quad \sum_{k=1}^{K} \pi_k = 1 \)

Gaussian mixture marginal and conditional likelihood:

\[
p(x^i | \pi, \mu, \Sigma) = \sum_{z^i = 1}^{K} \pi_{z^i} \ p(x^i | z^i, \mu, \Sigma) \]

\[
p(x^i | z^i, \mu, \Sigma) = \mathcal{N}(x^i | \mu_{z^i}, \Sigma_{z^i}) \]
What if we want to do density estimation with multimodal or clumpy data?

But we don’t see class labels!!!

- MLE:
  - $\arg\max \prod_i P(z^i, x^i)$

- But we don’t know $z^i$
- Maximize marginal likelihood:
  - $\arg\max \prod_i P(x^i) = \arg\max \prod_i \sum_{k=1}^{K} P(z^i=k, x^i)$
Special case: spherical Gaussians and hard assignments

If $P(X|z=k)$ is spherical, with same $\sigma$ for all classes:

$$P(x^i | z^i = k) \propto \exp\left[-\frac{1}{2\sigma^2} \|x^i - \mu_k\|^2\right]$$

If each $x^i$ belongs to one class $C(i)$ (hard assignment), marginal likelihood:

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

Same as K-means!!!
Generic Mixture Models

- Observations:
- Parameters:
- Likelihood:

Ex. $z^i = \text{country of origin}$, $x^i = \text{height of } i^{th} \text{ person}$
- $k^{th}$ mixture component = distribution of heights in country $k$

ML Estimate of Mixture Model Params

- Log likelihood
  \[ L_x(\theta) \triangleq \log p(\{x^i\} | \theta) = \sum_i \log \sum_{z^i} p(x^i, z^i | \theta) \]

- Want ML estimate
  \[ \hat{\theta}^{ML} = \]

- Neither convex nor concave and local optima
If “complete” data were observed...

- Assume class labels $z^i$ were observed in addition to $x^i$

$$L_{x,z}(\theta) = \sum_i \log p(x^i, z^i | \theta)$$

- Compute ML estimates
  - Separates over clusters $k$

- Example: mixture of Gaussians (MoG) $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$

Iterative Algorithm

- Motivates a coordinate ascent-like algorithm:
  1. Infer missing values $z^i$ given estimate of parameters $\hat{\theta}$
  2. Optimize parameters to produce new $\hat{\theta}$ given “filled in” data $z^i$
  3. Repeat
- Example: MoG (derivation soon... + HW)
  1. Infer “responsibilities”

$$r_{ik} = p(z^i = k | x^i, \hat{\theta}(t-1))$$

  2. Optimize parameters

$$\max_{\pi_k} \pi_k$$

$$\max_{\mu_k, \Sigma_k} \mu_k, \Sigma_k$$
E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. \( \Rightarrow \) convergence to a local optimum guaranteed

- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data
After first iteration

After 2nd iteration
After 3rd iteration

After 4th iteration
After 20th iteration

Some Bio Assay data
GMM clustering of the assay data

Resulting Density Estimator
Expectation Maximization (EM) – Setup

- More broadly applicable than just to mixture models considered so far

- Model: $x$ observable – “incomplete” data
  $y$ not (fully) observable – “complete” data
  $\theta$ parameters

- Interested in maximizing (wrt $\theta$):
  $$p(x \mid \theta) = \sum_y p(x, y \mid \theta)$$

- Special case:
  $$x = g(y)$$

---

Expectation Maximization (EM) – Derivation

- Step 1
  - Rewrite desired likelihood in terms of complete data terms
  $$p(y \mid \theta) = p(y \mid x, \theta)p(x \mid \theta)$$

- Step 2
  - Assume estimate of parameters $\hat{\theta}$
  - Take expectation with respect to $p(y \mid x, \hat{\theta})$
Expectation Maximization (EM) – Derivation

Step 3
- Consider log likelihood of data at any $\theta$ relative to log likelihood at $\hat{\theta}$
  
  $L_x(\theta) - L_x(\hat{\theta})$

Aside: Gibbs Inequality

$E_p[\log p(x)] \geq E_p[\log q(x)]$

Proof:

Step 4
- Determine conditions under which log likelihood at $\theta$ exceeds that at $\hat{\theta}$

Using Gibbs inequality:

If

Then

$L_x(\theta) \geq L_x(\hat{\theta})$
Motivates EM Algorithm

- Initial guess:
- Estimate at iteration $t$:

  **E-Step**
  Compute

  **M-Step**
  Compute

Example – Mixture Models

- **E-Step** Compute $U(\theta, \hat{\theta}^{(t)}) = E[\log p(y \mid \theta) \mid x, \hat{\theta}^{(t)}]$ 
- **M-Step** Compute $\hat{\theta}^{(t+1)} = \arg \max_{\theta} U(\theta, \hat{\theta}^{(t)})$

- Consider $y^i = \{z^i, x^i\}$ i.i.d.

  $p(x^i, z^i \mid \theta) = \pi_z, p(x^i \mid \phi_{z^i}) =$

  $E_{q_{\theta}}[\log p(y \mid \theta)] = \sum_i E_{q_{\theta}}[\log p(x^i, z^i \mid \theta)] =$
Coordinate Ascent Behavior

- Bound log likelihood:
  \[ L_x(\theta) = U(\theta, \hat{\theta}^{(t)}) + V(\theta, \hat{\theta}^{(t)}) \]
  \[ \geq L_x(\hat{\theta}^{(t)}) = U(\hat{\theta}^{(t)}, \hat{\theta}^{(t)}) + V(\hat{\theta}^{(t)}, \hat{\theta}^{(t)}) \]

Comments on EM

- Since Gibbs inequality is satisfied with equality only if \( p=q \), any step that changes \( \hat{\theta} \) should strictly increase likelihood.

- In practice, can replace the M-Step with increasing \( U \) instead of maximizing it (Generalized EM).

- Under certain conditions (e.g., in exponential family), can show that EM converges to a stationary point of \( L_x(\theta) \).

- Often there is a natural choice for \( y \) … has physical meaning.

- If you want to choose any \( y \), not necessarily \( x=g(y) \), replace \( p(y \mid \theta) \) in \( U \) with \( p(y, x \mid \theta) \).
Initialization

- In mixture model case where \( y^i = \{ z^i, x^i \} \) there are many ways to initialize the EM algorithm

- Examples:
  - Choose K observations at random to define each cluster. Assign other observations to the nearest "centroid" to form initial parameter estimates
  - Pick the centers sequentially to provide good coverage of data
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed

- Can be quite important to convergence rates in practice

What you should know

- K-means for clustering:
  - algorithm
  - converges because it’s coordinate ascent

- EM for mixture of Gaussians:
  - How to “learn” maximum likelihood parameters (locally max. like.) in the case of unlabeled data

- Be happy with this kind of probabilistic analysis

- Remember, E.M. can get stuck in local minima, and empirically it DOES

- EM is coordinate ascent