Clustering images

Set of Images

Cluster images

organize data into themes

given no labels

Clustering K-means

Machine Learning – CSE546
Carlos Guestrin
University of Washington

November 4, 2014

©Carlos Guestrin 2005-2014
K-means

- Randomly initialize $k$ centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

- 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: $\mu_i^{(t)}$ becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg \min_{\mu} \sum_{j : C^{(t)} = i} \|\mu - x_j\|^2$
  - $\mu_i^{(t+1)} = \frac{\sum_{j : C^{(t)} = i} x_j}{\sum_{j : C^{(t)} = i}}$

Repeat until convergence: no point change, cluster membership

Mixtures of Gaussians

Machine Learning – CSE546
Carlos Guestrin
University of Washington
November 4, 2014
(One) bad case for k-means

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

Density Estimation

- Estimate a density based on $x_1, \ldots, x^N$

$X \sim p$

Learn $p$ from data

Combined population

US
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

**Mixture of 3 Gaussians**

**Contour Plot of Joint Density**

Gaussians in $d$ Dimensions

$$P(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\Sigma}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]$$
Density as Mixture of Gaussians

Approximate density with a mixture of Gaussians:

\[ p(x | \pi, \mu, \Sigma) = \sum_{i=1}^{K} \pi_i N(x | \mu_i, \Sigma_i) \]

Our actual observations

C. Bishop, Pattern Recognition & Machine Learning
Imagine we have an assignment of each $x^i$ to a Gaussian. Introduce latent cluster indicator variable $z^i$. Then we have:

$$p(x^i | z^i = j) = \pi_j$$

$$p(z^i = j) = \pi_j$$

Then we have:

$$p(x^i | z^i, \pi, \mu, \Sigma) \propto N(\mu_j, \Sigma_j)$$

If we had observed the $z^i$, estimating parameters would be easy.
Clustering our Observations

We must infer the cluster assignments from the observations.

\[ r_{ik} = \frac{p(z_i = k | x_i, \pi, \mu, \Sigma)}{\sum_{j=1}^{K} p(z_i = j | x_i, \mu_j, \Sigma_j)} \]

Posterior probabilities of assignments to each cluster *given* model parameters:

\[ p(z_i = k | x_i, \pi, \mu, \Sigma) = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i | \mu_j, \Sigma_j)} \]

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

Gaussian mixture marginal and conditional likelihood:

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

$$p(x^i | z^i, \mu, \Sigma) = N(x^i | \mu_{z^i}, \Sigma_{z^i})$$
What if we want to do density estimation with multimodal or clumpy data?
But we don’t see class labels!!!

- MLE:
  \[ \arg\max_{\pi, \mu, \Sigma} \prod_{i} P(z_i, x_i) \]

  But we don’t know \( z_i \)

  Maximize marginal likelihood:
  \[ \arg\max_{\mu, \Sigma} \prod_{i} P(x_i) = \arg\max_{\mu, \Sigma} \prod_{i} \sum_{k=1}^{K} P(z_i = k | x_i) \]

Special case: spherical Gaussians and hard assignments

If \( P(X|z=k) \) is spherical, with same \( \sigma \) for all classes:
\[ P(x' \mid z = k) \propto \exp \left( -\frac{1}{2 \sigma^2} \| x' - \mu_k \|^2 \right) \]

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!!!
EM: “Reducing” Unsupervised Learning to Supervised Learning

- If we knew assignment of points to classes → Supervised Learning!

- Expectation-Maximization (EM)
  - Guess assignment of points to classes
  - In standard ("soft") EM: each point associated with prob. of being in each class
  - Recompute model parameters
  - Iterate

Generic Mixture Models

- Observations: \(x^1, \ldots, x^N\) with \(x^i \in \mathbb{R}^q\)

- Parameters:
  - \(\pi = \{\pi_1, \ldots, \pi_k\}\) mix weights
  - \(\phi = \{\phi, \ldots, \phi_k\}\) mix component params
  - \(\phi_k = \{\mu_k, \Sigma_k\}\)
  - \(\theta = \{\pi, \phi\}\)

- Likelihood:
  \[
  p(x^i | \theta) = \sum_{k=1}^{K} \pi_k \cdot p(x^i | \phi_k) \approx N(x^i | \mu_k, \Sigma_k)
  \]

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

MoG Example:
ML Estimate of Mixture Model Params

- Log likelihood
  \[ L_x(\theta) = \log p(x_i \mid \theta) = \sum_{i=1}^{m} \log \sum_{z_i} p(x_i, z_i \mid \theta) \]

- Want ML estimate
  \[ \hat{\theta}^{ML} = \arg\max_{\theta} L_x(\theta) \]

- Neither convex nor concave and local optima

If “complete” data were observed...

- Assume class labels \( z_i \) were observed in addition to \( x_i \)
  \[ L_{x,z}(\theta) = \sum_{i=1}^{m} \log p(x_i, z_i \mid \theta) = \sum_{i=1}^{m} \log \sum_{i=1}^{k} \pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \]

- Compute ML estimates
  - Separates over clusters \( k \)

- Example: mixture of Gaussians (MoG)
  \[ \theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K} \]
  \[ \hat{\mu}_k = \frac{\sum_{z_i = k} x_i}{N_k} \text{, similarly for } \hat{\Sigma}_k \]
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...)
  1. Infer “responsibilities”
     $$r_{ik} = p(z^i = k | x^i, \hat{\theta}^{(t-1)}) = \frac{\prod_k p(x^i | \phi_k^{(t-1)})}{\sum_j \prod_k p(x^i | \phi_j^{(t-1)})}$$
  2. Optimize parameters
     $$\max \text{ w.r.t. } \pi_k :$$
     $$\max \text{ w.r.t. } \mu_k, \Sigma_k :$$

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
Gaussian Mixture Example: Start

Guess some $\Theta^{(0)}$

After first iteration

re-estimate $\pi, \mu, \sigma$

re-weight $r_{ik}$
After 2nd iteration

After 3rd iteration
After 4th iteration

After 5th iteration
After 6th iteration

After 20th iteration

looks pretty good.
Some Bio Assay data

GMM clustering of the assay data
E.M.: The General Case

- E.M. widely used beyond mixtures of Gaussians
  - The recipe is the same…

- Expectation Step: Fill in missing data, given current values of parameters, $\theta^{(t)}$
  - If variable $y$ is missing (could be many variables)
  - Compute, for each data point $x_i$, for each value $i$ of $y$:
    - $P(y=i|x, \theta^{(t)})$

- Maximization step: Find maximum likelihood parameters for (weighted) "completed data”:
  - For each data point $x_i$, create $k$ weighted data points
  - Set $\theta^{(t+1)}$ as the maximum likelihood parameter estimate for this weighted data

- Repeat
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 "centriped" 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 quality of solution 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

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

- EM is coordinate ascent
### 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 $\theta$):
  $$ \max_{\theta} \prod_{x} p(x \mid \theta) = \sum_{y} p(x, y \mid \theta) $$
- **Special case:**
  $$ x = g(y) $$

### Expectation Maximization (EM) – Derivation

**Step 1**
- Rewrite desired likelihood in terms of complete data terms
  $$ p(y \mid \theta) = p(y \mid x, \theta)p(x \mid \theta) $$
- $$ = \log p(x \mid \theta) = \log p(y \mid x) - \log p(y \mid x, \theta) $$
- $$ \Rightarrow \frac{\log p(x \mid \theta)}{L(x(\theta))} $$

**Step 2**
- Assume estimate of parameters $\hat{\theta}$
- Take expectation with respect to $p(y \mid x, \hat{\theta})$
  $$ L_x(\theta) = \frac{\text{E}[-\log p(y \mid x, \hat{\theta})]}{\text{E}[\log p(y \mid x, \hat{\theta})]} $$
  $$ + \frac{\text{E}[-\log p(y \mid x, \hat{\theta})]}{\text{E}[\log p(y \mid x, \hat{\theta})]} $$
Expectation Maximization (EM) – Derivation

- **Step 3**
  - Consider log likelihood of data at any $\theta$ relative to log likelihood at $\hat{\theta}$
  - $L_x(\theta) - L_x(\hat{\theta}) = [U(\theta, \hat{\theta}) - U(\hat{\theta}, \hat{\theta})] + [V(\theta, \hat{\theta}) - V(\hat{\theta}, \hat{\theta})]$  
  - Focus on this term

- **Aside: Gibbs Inequality**
  - $E_p[\log p(x)] \geq E_p[\log q(x)]$
  - Proof:

Expectation Maximization (EM) – Derivation

- **Step 4**
  - Determine conditions under which log likelihood at $\theta$ exceeds that at $\hat{\theta}$
  - Using Gibbs inequality:
    - $U(\theta, \hat{\theta}) \geq E[-\log p(y|x, \hat{\theta}) | x, \hat{\theta}] > E[-\log p(y|x, \hat{\theta}) | x, \hat{\theta}] = U(\hat{\theta}, \hat{\theta})$
  - If $U(\theta, \hat{\theta}) > U(\hat{\theta}, \hat{\theta})$
    - Then $L_x(\theta) > L_x(\hat{\theta})$
      - Choose $\theta$ such that $U$ increases
Motivates EM Algorithm

- Initial guess: $\hat{\theta}^{(0)}$
- Estimate at iteration $t$: $\hat{\theta}^{(t)}$

**E-Step**

Compute

$$U(\theta, \hat{\theta}^{(t)}) = E[\log p(y | \theta) | x, \hat{\theta}^{(t)}]$$

**M-Step**

Compute

Example – Mixture Models

- **E-Step** Compute

$$U(\theta, \hat{\theta}^{(t)}) = E[\log p(y | \theta) | x, \hat{\theta}^{(t)}]$$

- **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.

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

$$E_{q_\theta}[\log p(y | \theta)] = \sum_i E_{q_\theta}[\log p(x^i, z^i | \theta)] =$$
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 "centriod" 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