Introduction to Machine Learning

Lecture 13: EM, Clustering and More

Nir Ailon
Hidden (Latent) Variables

- Disease → Symptom
- Word → Sound
- Movie Experience → Rating
MLE With Hidden Variables

• Assume $X, Y \in \mathcal{X} \times \mathcal{Y}$ random variables distribution with unknown parameter $\Theta$

• Given iid draw $(X_1, Y_1), \ldots, (X_m, Y_m)$
  • We see the $X_i'$s only!!!!!!

• $\hat{\Theta} = \hat{\Theta}_{MLE} = \arg\max_{\Theta} LL(X_1 \ldots X_m; \Theta)$

• By independence:

$$LL(X_1 \ldots X_m; \Theta) = \sum_{i=1}^{m} \log \Pr_{\Theta}[X_i] = \sum \log \sum_{y \in \mathcal{Y}} \Pr_{\Theta}[X_i, y]$$

• Can be a difficult optimization problem in many seemingly simple cases
EM (Expectation-Maximization) Algorithm

• The idea is that if we knew for each example $X_i$ the conditional distribution on $y$:
  \[ \Pr[Y = y|X_i] \]

• ...then instead of optimizing the log-likelihood
  \[ LL(X_1 \ldots X_m; \Theta) = \sum LL(X_i; \Theta) \]
  we could optimize the expected log-likelihood, where the expectation is with respect to a distribution on the unobserved variable:
  \[ \sum_i \sum_y \Pr[Y = y|X_i] \log [X_i, y] \]

• This is not the Log-Likelihood... will get back to this problem shortly

• But we don’t have $\Pr[Y = y|X_i]$...
EM (Expectation-Maximization) Algorithm

- Define new variables $Q_{iy} \geq 0$ for $(i, y) \in [m] \times Y$
- Constraint: $\sum_y Q_{iy} = 1$ for all $i$
- New objective function:

$$F(Q, \Theta) = \sum_i \sum_y Q_{iy} \log \Pr[\Theta[X_i, y]$$

**(E)xpectation step:**

$$Q_{iy}^{(t+1)} = \Pr[\Theta^{(t)}[Y = y|X_i]]$$

**(M)aximization step:**

$$\Theta^{(t+1)} = \arg\max_{\Theta} F(Q^{(t+1)}, \Theta)$$

*Note: (M) step clearly improves $F$. It can be shown that (E) step improves something related to $F$*
EM Guarantee

• Theorem:

\[ LL(X_1 \ldots X_m | \Theta^{(t+1)}) \geq LL(X_1 \ldots X_m | \Theta^{(t)}) \]

• Proof in book...
EM For Clustering (Soft k-means)

• $Y \in \mathcal{Y} = [k]$ cluster id from unknown distribution
  $c_y = \Pr[Y = y]$

• $X \in \mathcal{X} = \mathbb{R}^n$
  \[ X|Y \sim N(\mu_y, \Sigma) \]

• $\Theta = (c_1 \ldots c_k, \mu_1 \ldots \mu_k)$

\[
\begin{align*}
LL(X_1 \ldots X_m; \Theta) &= \sum_i \log \sum_y c_y e^{-\frac{(X_i-\mu_y)^T \Sigma^{-1}(X_i-\mu_y)}{2}} + m \cdot \alpha \\
\end{align*}
\]

• Hard to optimize exactly
• (Not convex)

• *Note: $\Sigma$ must be assumed to be known*
EM For Clustering
(Soft k-means)

\[ F(Q, \Theta) = \sum_i \sum_y Q_{i,y} \log \Pr[ X_i, y ] = \frac{(X_i - \mu_y)^T \Sigma^{-1} (X_i - \mu_y)}{2} + \alpha = \sum_i \sum_y Q_{i,y} \left( \log c_y - \frac{(X_i - \mu_y)^T \Sigma^{-1} (X_i - \mu_y)}{2} + \alpha \right) \]

- (E) step:
  \[ Q_{i,y}^{(t+1)} = \frac{\Pr_{\Theta(t)}[X_i, y]}{\sum_y \Pr_{\Theta(t)}[X_i]} \]

- (M) step:
  \[ \Theta^{(t+1)} = \arg \max_{\Theta} F(Q^{(t+1)}, \Theta) \]
(M) Step for Soft k-Means

\[ F(Q^{(t+1)}, \Theta) = \sum_i \sum_y Q_{iy}^{(t+1)} \log \Pr[X_i, y] = \]
\[ \sum \sum Q_{iy}^{(t+1)} \left( \log c_y e^{-\frac{(x_i-\mu_y)^T \Sigma^{-1}(x_i-\mu_y)}{2}} + \alpha \right) = \]
\[ \sum \sum Q_{iy}^{(t+1)} \left( \log c_y - \frac{(x_i-\mu_y)^T \Sigma^{-1}(x_i-\mu_y)}{2} + \alpha \right) \]

• Optimal \( \Theta \) for:

\[ c_y = \frac{\sum_i Q_{iy}}{\sum_y, \sum_i Q_{iy}} = \frac{\sum_i Q_{iy}}{m} \]
\[ \mu_y = \frac{\sum_i Q_{iy}X_i}{\sum_i Q_{iy}} \]
Data Clustering
Why Cluster?

• Data clustering: A good way to get intuition about data
  • Retailers cluster clients based on profiles
  • Biologists cluster genes based on expression similarity
  • Social network researchers may cluster people (nodes in a graph) based on
    • Profile similarity
    • Graph neighborhood similarity
    • Etc
Clustering is Tricky to Define

• Intuitive definition:
  Given set of points $X_1 \ldots X_m$, distance measure $d(X_i, X_j) \geq 0$ and number $k$, partition set to $k$ subsets such that points within each subset are close (similar) to each other, and points between any two subsets are distant (dissimilar) from each other.

  (Can define using similarity function instead)
  (Can define without need to provide $k$)

• How do we define ``similar” and ``dissimilar’’?
  • Multiple criteria to choose from

• Clustering induces a transitive (equivalence) relation, while similarity is not transitive
  • If x similar to y and y similar to z, x is not necessarily similar to z
Clustering Can be Tricky to Define

Close (similar) points should not be separated
Clustering can be Tricky to Define

Far (dissimilar) points should be separated
Linkage Based Clustering

• Start by setting each point $X_i$ to be its own cluster
• At each step, unite the closest two clusters
• Need some stopping condition
Linkage Based Clustering

• How to define distance between clusters? Various options:
  
  • \( d(C_1, C_2) = \min_{X_i \in C_1, X_j \in C_2} d(X_i, X_j) \)
  
  • \( d(C_1, C_2) = \max_{X_i \in C_1, X_j \in C_2} d(X_i, X_j) \)
  
  • \( d(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{X_i \in C_1, X_j \in C_2} d(X_i, X_j) \)

• How to decide on stopping condition?
  
  • When we have \( k \) clusters
  
  • When all distances (between clusters) larger than \( r \)
  
  • Don’t stop! Use the ``history’’ as the solution
    
    • More informative than a single clustering
    
    • Contains hierarchical information
Linkage Based Clustering: Dendogram Output
Cost Minimization Clustering

- Minimize objective function $G(C_1, \ldots, C_k)$
  
  - $G_{kmeans} = \min_{\mu_1 \ldots \mu_k} \sum_{j=1}^{k} \sum_{i:X_i \in C_j} d(X_i, \mu_j)^2$
  
  - $G_{kmedoids} = \min_{\mu_1 \ldots \mu_k \in \{x_1 \ldots x_m\}} \sum_{j=1}^{k} \sum_{i:X_i \in C_j} d(X_i, \mu_j)^2$
  
  - $G_{kmedian} = \min_{\mu_1 \ldots \mu_k} \sum_{j=1}^{k} \sum_{i:X_i \in C_j} d(X_i, \mu_j)$
  
  - $G_{SOD} = \sum_{j=1}^{k} \sum_{X,X' \in C_j} d(X, X')$
  
  - ....
Spectral Clustering

- Assume "similarity function" $w(X_i, X_j)$ view
  - If you have a distance function, can define eg
    $$w(X_i, X_j) = e^{-d(X_i,X_j)^2 / \sigma^2}$$

- View $X_1, \ldots, X_m$ and $w$ as weighted undirected graph

- Clustering objective = Cut cost
  (Total weight of all edges crossing clusters)
  $$G_{cut}(C_1 \ldots C_k) = \sum_j \sum_{x \in C_j, x' \notin C_j} w(X, X')$$

Poly-Time optimizable for fixed $k$
Spectral Clustering

- Problem is that $G_{cut}$ often prefers very small clusters (even singletons)
- Can "punish" small clusters:

$$G_{RatioCut}(C_1 \ldots C_k) = \sum_j \frac{1}{|C_j|} \sum_{x \in C_j, x' \not\in C_j} w(x, x')$$

- NP-Hard for $k = 2$
- Equivalent problem formulation:
  - Clustering viewed as matrix $H \in \mathbb{R}^{m \times k}$:
    $$H_{ij} = \frac{1}{\sqrt{|C_j|}} \begin{cases} 1 & X_i \in C_j \\ 0 & \text{otherwise} \end{cases}$$
  - $H$ orthonormal
  - $G_{RatioCut} = \text{trace } H^T L H$
    $$L = \text{(Unnormalized) Graph Laplacian}$$
  - Instead of optimizing over discrete $H$, optimize over all $H$ with orthonormal columns
- Closed solution:
  $$\tilde{H} = k \text{ eigenvectors of } L \text{ (as columns)}$$
Spectral Clustering Rounding

• How to ``round’’ $\tilde{H}$ to discrete $H$?

• Common practice:
  • Define new geometric embedding $\phi(X_i) = i$’th row of $\tilde{H}$
  • Clustering now hopefully easy (e.g. with k-means)

• Note: This generates embedding in $\mathbb{R}^k$ even if original distance/similarity didn’t come from embedding in real vector space!
When we see unlabeled data, we conclude that it is generated from distribution with two connected components (each with a different label).

(Semi)Supervised Learning

- Clustering is considered an "unsupervised" learning scenario
  - We don’t use any label. We just find simple geometric structure in features of unlabeled data.

- Can we combine both supervised and unsupervised learning?

What’s going on here?

When we see unlabeled data, we conclude that it is generated from distribution with two connected components (each with a different label)
(Semi)Supervised Learning

• Graph Based
  • Prefer decision boundary that has low density of labeled + unlabeled points

• Manifold based
  • Assuming the data is contained in a low dimensional manifold (or close to it), use labeled+unlabeled data to learn the structure of the manifold, then learn on lower dimensional space

• Generative modeling approach
  • View labels of unlabeled data as latent variables. Incorporate in parameter estimation (e.g. using EM)

• Many heuristics...