Expectation-Maximization and Clustering

Machine Learning

Overview

1. Expectation-Maximization Framework.
2. Clustering framework.
3. Expectation-Maximization Clustering.
4. Clustering and Dimensionality Reduction.

Expectation-Maximization Algorithm

- TL;DR: An algorithm to efficiently iteratively find maximum likelihood estimators.
- **Given:**
  - A family of distributions $\mathcal{M} = \{\mu_\lambda : \lambda \in \Lambda\}$ over $\mathcal{Z}$ with known densities $p_{Z|\Lambda}(z \mid \lambda)$.
  - A dataset $s_{X;\text{train}} \sim \mu_{\lambda_\star;X}$ with hidden labels.
- **Idea:** Use $\mathbb{E}_y\left[\log\left(p_{Z|\Lambda}(x, y \mid \lambda)\right)\right]$ as a proxy for $\log\left(p_{X|\Lambda}(x \mid \lambda)\right)$. (Provably good.)
- "Simple" alternating optimization scheme:

Input: A distribution family $\mathcal{M}$, a dataset $s_{X;\text{train}}$, a large integer horizon T.

Output: An estimate $\widehat{\lambda}$ for $\lambda_\star$.

$\lambda_1 \leftarrow$ any element of Λ.

for $t \in [T]$ **do**

$Q(\lambda \mid \lambda_t) \leftarrow \sum_{i=1}^{n} \mathbb{E}_{y \sim \mu_{Y|X,\Lambda}(\cdot|x_i,\lambda_t)}\left[\log\left(p_{Z|\Lambda}(x_i, y \mid \lambda)\right) \,\middle|\, x_i, \lambda\right]$ ▷ Expectation.

$\lambda_{t+1} \leftarrow \operatorname{argmax}_{\lambda \in \Lambda} Q(\lambda \mid \lambda_t)$ ▷ Maximization.

return λ_{T+1}

Clustering

Clustering Setting

- Have data $s_{\text{train};X} \sim \mu_X$.
- Want to place data into **clusters** (sets of data points).
- Then we can say things about:
  - Existing data, based on which cluster it is in.
  - New data, by placing it into a cluster and making the same inference.

- Useful for:
  - Prediction (i.e. predict based on average prediction in the cluster)
  - Multi-modal statistics (learning the modes of the distribution in an unsupervised way).

EM For Mixture Model Clustering

- Let's introduce a hidden "label" such that y_i denotes the cluster of x_i.
- Let each μ_λ be a distribution with known density. Normal distribution is popular.
- Furthermore, let's let $\mathcal{M}$ be a family of mixtures of k distributions:

$$\mathcal{M} = \left\{ \sum_{i=1}^{k} \alpha_i \mu_{\lambda_i} : \lambda_1, \ldots, \lambda_k \in \Lambda, \alpha_1, \ldots, \alpha_k \in [0,1], \sum_{i=1}^{k} \alpha_i = 1 \right\}.$$

  If $x \sim \sum_{i=1}^{k} \alpha_i \mu_{\lambda_i}$ then $p_Y(y_i) = \alpha_i$ and $p_{X|Y}(x \mid y_i) \sim \mu_{\lambda_i}$. Thus we jointly estimate the parameters α and λ. This is a **mixture model**.
- Apply EM algorithm to recover the estimated $\widehat{\alpha}, \widehat{\lambda}$ that allows us to estimate a new y for a given x by maximizing $p_{Y|X}(y \mid x, \widehat{\alpha}, \widehat{\lambda})$.

Clustering and Dimensionality Reduction

Curse of Dimensionality

A phenomenon where in high-dimensional space, not all regions of space are represented, even by a huge data set. The amount of training data to ensure "good" generalization is often exponential in the feature dimension.

- Hence, EM clustering might fail to work well (Gaussian/Laplace dimensions have density as a monotone function of distance in $\mathbb{R}^d$).
- Solution: Do smart, data-driven dimensionality reduction, such as PCA.
  - Random projections fail because most likely we get a random direction with sparse variation.
- Then, cluster the projections onto the principal components.