Maximum Likelihood and Bayesian Methods

Shyue Ping Ong

University of California, San Diego

NANO281

Overview

1. Preliminaries
2. Maximum Likelihood Inference
3. Bayesian Methods
4. Expectation Maximization

Preliminaries

- Thus far, we have discussed ML models in the context of minimizing a loss function, typically a sum of squares.
- These approaches have a probabilistic interpretation and are instances of the maximum likelihood (ML) approach to fitting.

Maximum Likelihood Inference

- Consider a probability distribution/mass function for the observations:

$$z_i \sim g_\theta(z)$$

- Also, known as a parametric model for Z, with θ being unknown parameter that govern the distribution.
- Let us assume that Z has a normal distribution ($\theta = (\mu, \sigma^2)$):

$$g_\theta = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

- The likelihood of the observed data under model g_θ is then given by the *likelihood function*:

$$L(\theta; \mathrm{Z}) = \prod_{i=1}^{N} g_\theta(z_i)$$

Maximum Likelihood Inference, cont.

- Consider the log of L:

$$\ell(\theta; \mathbf{Z}) = \sum_{i=1}^{N} \log g_\theta(z_i)$$

- *Score function* $\dot{\ell}(\theta; \mathbf{Z}) = 0$ at maximum (dot means derivative).
- *Information matrix*:

$$\mathbf{I}(\theta) = -\sum_{i=1}^{N} \frac{\partial^2 \ell(\theta; z_i)}{\partial \theta \partial \theta^T}$$

- *Observed information* $= \mathbf{I}(\hat{\theta})$ ($\hat{\theta}$ is the estimated parameter)
- *Fisher information* $\mathbf{i}(\theta) = E_\theta[\mathbf{I}(\theta)]$, widely used in optimal experimental design.

Linear Regression Revisited as MLE

- $Y = \beta_0 + \beta_1 X + \varepsilon$
- $\varepsilon \sim N(0, \sigma^2)$, i.e., independent Gaussian noise.

$$
\begin{aligned}
L(\beta_0, \beta_1, \sigma; \mathrm{y}) &= \prod_{i=1}^{N} P(y_i | x_i; \hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}) \\
&= \prod_{i=1}^{N} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{2\hat{\sigma}^2}} \\
\ell(\beta_0, \beta_1, \sigma; \mathrm{y}) &= \sum_{i=1}^{N} \log \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{2\hat{\sigma}^2}} \\
&= -\frac{N}{2} \log \hat{\sigma^2} 2\pi - \sum_{i=1}^{N} \frac{(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{2\hat{\sigma}^2}
\end{aligned}
$$

- Clearly, MLE in this case is equivalent to minimizing least squares.

Bayesian Methods

- *Posterior* distribution given by:

$$P(\theta|\mathrm{Z}) = \frac{P(\mathrm{Z}|\theta)P(\theta)}{\int P(\mathrm{Z}|\theta)P(\theta)d\theta}$$

- Updated knowledge about θ after seeing data.

Expectation Maximization

- Popular approach for solving MLE problems.
- Algorithm (example for two-component Gaussian mixture):
  1. Start with initial guesses for parameters (e.g, two random y_i).
  2. Expectation step: Compute the *responsibilities*.

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, i = 1, 2, ..., N$$

  3. Maximization step: Compute weighted means and variances.

$$\hat{\mu_1} = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \hat{\sigma_1^2} = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu_1})^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)}$$

$$\hat{\mu_2} = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \hat{\sigma_2^2} = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu_2})^2}{\sum_{i=1}^N \hat{\gamma}_i}$$

     with mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.
  4. Iterate until convergence.

Gaussian mixture using EM in scikit-learn

```
from sklearn.mixture import GaussianMixture
estimator = GaussianMixture(n_components=n_classes,
                            covariance_type="spherical",
                            max_iter=20,
                            random_state=0)
estimator.means_init = np.array([x_train[y_train == i].mean(axis=0)
                                 for i in range(n_classes)])
estimator.fit(x_train)
```

The End