MLT Week-4

Sherry Thomas
21f3001449

Contents

Abstract

The week introduces estimators, and delves deeper into topics like Maximum Likelihood Estimator and Bayesian Estimator. Later, it goes into Gaussian Mixture Models and its implementation.

Introduction to Estimation

Estimators in machine learning are algorithms or models used to estimate unknown parameters or predict outcomes based on data. The aim of the method is to find/predict the unknown parameters describing the distribution of the data.

Let $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be a dataset where $\mathbf{x}_i \in \{0,1\}^d$. We assume that the datapoints are independent and identically distributed.

Independence means $P(\mathbf{x}_i|\mathbf{x}_j) = P(\mathbf{x}_i)$. Identically distributed means $P(\mathbf{x}_i) = P(\mathbf{x}_j) = p$.

Maximum Likelihood Estimation

Fisher's Principle of Maximum Likelihood

Fisher's principle of maximum likelihood is a statistical method used to estimate the parameters of a statistical model by choosing the parameter values that maximize the likelihood function, which measures how well the model fits the observed data.

Likelihood Estimation on Bernoulli Distributions

Applying the likelihood function on the above dataset, we get

$$\begin{aligned}\mathcal{L}(p;\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\}) &= P(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n;p)\\ &= p(\mathbf{x}_1;p)p(\mathbf{x}_2;p)\dots p(\mathbf{x}_n;p)\\ &= \prod_{i=1}^{n} p^{\mathbf{x}_i}(1-p)^{1-\mathbf{x}_i}\end{aligned}$$

$$\therefore \log(\mathcal{L}(p;\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\})) = \arg\max_p \log\left(\prod_{i=1}^{n} p^{\mathbf{x}_i}(1-p)^{1-\mathbf{x}_i}\right)$$

Differentiating wrt p, we get

$$\therefore \hat{p}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

Likelihood Estimation on Gaussian Distributions

Let $\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\}$ be a dataset where $\mathbf{x}_i \sim \mathcal{N}(\mu,\sigma^2)$. We assume that the datapoints are independent and identically distributed.

$$\begin{aligned}\mathcal{L}(\mu,\sigma^2;\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\}) &= f_{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n}(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n;\mu,\sigma^2)\\ &= \prod_{i=1}^{n} f_{\mathbf{x}_i}(\mathbf{x}_i;\mu,\sigma^2)\\ &= \prod_{i=1}^{n}\left[\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(\mathbf{x}_i-\mu)^2}{2\sigma^2}}\right]\\ \therefore \log(\mathcal{L}(p;\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\})) &= \sum_{i=1}^{n}\left[\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{(\mathbf{x}_i-\mu)^2}{2\sigma^2}\right]\end{aligned}$$

Differentiating wrt μ and σ, we get

$$\hat{\mu}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

$$\hat{\sigma^2}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\mu)^T(\mathbf{x}_i-\mu)$$

Bayesian Estimation

Bayesian estimation is a statistical method that updates the estimates of model parameters by combining prior knowledge or beliefs with observed data to calculate the posterior probability distribution of the parameters.

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be a dataset where $\mathbf{x}_i$ belongs to a distribution with parameters θ. We assume that the datapoints are independent and identically distributed. We also assume that θ is also from a probability distribution.

Our goal is to update the parameters using the data.

i.e.

$$P(\theta) \Rightarrow P(\theta|\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\})$$

where, using Bayes' Law, we get

$$P(\theta|\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}) = \left(\frac{P(\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}|\theta)}{P(\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\})}\right) * P(\theta)$$

Bayesian Estimation for a Bernoulli Distribution

Let $\{x_1, x_2, \dots, x_n\}$ be a dataset where $x_i \in \{0, 1\}$ with parameter θ. What distribution can be suited for $P(\theta)$?

A commonly used distribution for priors is the Beta Distribution.

$$f(p; \alpha, \beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{z} \qquad \forall p \in [0, 1]$$

where z is a normalizing factor

Hence, using the Beta Distribution as the Prior, we get,

$$\begin{aligned} P(\theta|\{x_1, x_2, \dots, x_n\}) &\propto P(\theta|\{x_1, x_2, \dots, x_n\}) * P(\theta) \\ f_{\theta|\{x_1, x_2, \dots, x_n\}}(p) &\propto \left[\prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i}\right] * [p^{\alpha-1}(1-p)^{\beta-1}] \\ f_{\theta|\{x_1, x_2, \dots, x_n\}}(p) &\propto p^{\sum_{i=1}^{n} x_i + \alpha - 1}(1-p)^{\sum_{i=1}^{n}(1-x_i)+\beta-1} \end{aligned}$$

i.e. we get,

$$\text{BETA PRIOR } (\alpha, \beta) \xrightarrow[Bernoulli]{\{x_1, x_2, \dots, x_n\}} \text{BETA POSTERIOR } (\alpha + n_h, \beta + n_t)$$

$$\therefore \hat{p_{\text{ML}}} = \mathbb{E}[\text{Posterior}] = \mathbb{E}[\text{Beta}(\alpha + n_h, \beta + n_t)] = \frac{\alpha + n_h}{\alpha + n_h + \beta + n_t}$$

Gaussian Mixture Models

Gaussian Mixture Models are a type of probabilistic model used to represent complex data distributions by combining multiple Gaussian distributions.

The procedure is as follows:

- Step 1: Generate a mixture component among $\{1, 2, \dots, K\}$ where $z_i \in \{1, 2, \dots, K\}$. We get,

$$P(z_i = k) = \pi_k \qquad \left[\sum_{i=1}^{K} \pi_i = 1 \quad 0 \leq \pi_i \leq 1 \quad \forall i\right]$$

- Step 2: Generate $\mathbf{x}_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2_{z_i})$

Hence, there are $3K$ parameters. Let these parameters be represented by θ.

However, since $\sum_{i=1}^{K} \pi_i = 1$, the number of parameters to be estimated becomes $3K - 1$ for a GMM with K components.

Likelihood of GMM's

$$\begin{aligned}
\mathcal{L}\left(\begin{array}{c}\mu_1, \mu_2, \dots, \mu_K \\ \sigma_1^2, \sigma_2^2, \dots, \sigma_K^2 \\ \pi_1, \pi_2, \dots, \pi_K\end{array} ; \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\right) &= \prod_{i=1}^{n} f_{\text{mix}}\left(\mathbf{x}_i; \begin{array}{c}\mu_1, \mu_2, \dots, \mu_K \\ \sigma_1^2, \sigma_2^2, \dots, \sigma_K^2 \\ \pi_1, \pi_2, \dots, \pi_K\end{array}\right) \\
&= \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k * f_{\text{mix}}(\mathbf{x}_i; \mu_k, \sigma_k)\right] \\
\therefore \log \mathcal{L}(\theta) &= \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k * \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i - \mu_k)^2}{2\sigma_k^2}}\right]
\end{aligned}$$

To solve the above equation, we need to understand convexity.

Convexity and Jensen's Inequality

Convexity is a property of a function or set that implies a unique line segment can be drawn between any two points within the function or set. For a concave function, this property can be expressed as,

$$f\left(\sum_{k=1}^{K} \lambda_k a_k\right) \geq \sum_{k=1}^{K} \lambda_k f(a_k)$$

where

$$\sum_{k=1}^{K} \lambda_k = 1$$

a_k are points of the function

This is also known as **Jensen's Inequality**.

Estimating the Parameters

As log is a concave function, using the above equation, we can also approximate the likelihood function for GMM's as follows,

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k * \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i - \mu_k)^2}{2\sigma_k^2}} \right]$$

Introduce for data point $\mathbf{x}_i$, the parameters $\{\lambda_1^i, \lambda_2^i, ..., \lambda_k^i\}$ s.t. $\forall i, k \sum_{k=1}^{K} \lambda_k^i = 1; 0 \leq \lambda_k^i \ le 1$.

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \lambda_k^i \left(\pi_k * \frac{1}{\lambda_k^i \sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i - \mu_k)^2}{2\sigma_k^2}} \right) \right]$$

Using Jensen's Inequality, we get

$$\log \mathcal{L}(\theta) \geq \text{modified\_log}\mathcal{L}(\theta)$$

$$\therefore \text{modified\_log}\mathcal{L}(\theta) = \sum_{i=1}^{n} \sum_{i=k}^{K} \lambda_k^i \log \left(\pi_k * \frac{1}{\lambda_k^i \sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i - \mu_k)^2}{2\sigma_k^2}} \right) \tag{1}$$

Note that the modified-log likelihood function gives a lower bound for the true log likelihood function at θ. Finally, to get the parameters, we do the following:

- To get θ: Fix λ and maximize over θ.

$$\max_{\theta} \sum_{i=1}^{n} \sum_{i=k}^{K} \lambda_k^i \log \left(\pi_k * \frac{1}{\lambda_k^i \sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i - \mu_k)^2}{2\sigma_k^2}} \right)$$

Differentiate w.r.t. $\mu, \sigma^2,$ and π to get the following

$$\hat{\mu}_k^{\text{MML}} = \frac{\sum_{i=1}^{n} \lambda_k^i \mathbf{x}_i}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\hat{\sigma}_k^{2^{\text{MML}}} = \frac{\sum_{i=1}^{n} \lambda_k^i (\mathbf{x}_i - \hat{\mu}_k^{\text{MML}})^2}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\hat{\pi}_k^{\text{MML}} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}$$

- To get λ: Fix θ and maximize over λ. For any i:

$$\max_{\lambda_1^i,\lambda_2^i,\ldots,\lambda_k^i} \sum_{k=1}^{K} \left[\lambda_k^i \log \left(\pi_k * \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i-\mu_k)^2}{2\sigma_k^2}} \right) - \lambda_k^i \log(\lambda_k^i) \right] \quad s.t. \quad \sum_{k=1}^{K} \lambda_k^i = 1; 0 \leq \lambda_k^i \leq 1$$

  Solving the above constrained optimization problem analytically, we get

$$\hat{\lambda}_k^{i\text{MML}} = \frac{\left(\frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i-\mu_k)^2}{2\sigma_k^2}} \right) * \pi_k}{\sum_{k=1}^{K} \left(\frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(\mathbf{x}_i-\mu_k)^2}{2\sigma_k^2}} \right) * \pi_k}$$

EM Algorithm

The EM (Expectation-Maximization) algorithm is a popular method for estimating the parameters of statistical models with incomplete data by iteratively alternating between expectation and maximization steps until convergence to a stable solution.

The algorithm is as follows:

- Initialize $\theta^0 = \left\{ \begin{array}{c} \mu_1, \mu_2, \ldots, \mu_K \\ \sigma_1^2, \sigma_2^2, \ldots, \sigma_K^2 \\ \pi_1, \pi_2, \ldots, \pi_K \end{array} \right\}$ using Lloyd's algorithm.
- Until convergence ($||\theta^{t+1} - \theta^t|| \leq \epsilon$ where ϵ is the tolerance parameter) do the following:

$$\lambda^{t+1} = \arg\max_{\lambda} \text{ modified\_log}(\theta^t, \lambda) \quad \rightarrow \text{ Expectation Step}$$

$$\theta^{t+1} = \arg\max_{\theta} \text{ modified\_log}(\theta, \lambda^{t+1}) \quad \rightarrow \text{ Maximization Step}$$

EM algorithm produces soft clustering. For hard clustering using EM, a further step is involved:

- For a point $\mathbf{x}_i$, assign it to a cluster using the following equation:

$$z_i = \arg\max_k \lambda_k^i$$

Credits:

Professor Arun Rajkumar: The content as well as the notations are from his slides and lecture.