Supervised Learning

Christophe Troalen

July 2024

Disclaimer: These notes were originally created for personal reference during my studies at Imperial College London and Télécom SudParis. As such, they may not be exhaustive or fully accurate, serving mainly as quick notes rather than comprehensive explanations.

1 Classification

1.1 Generative Approach

The generative approach models the class-conditional densities $p(x \mid y = k)$ as well as the class priors $p(y = k)$, and uses these to compute posterior probabilities $p(y = k \mid x)$ with *Bayes' theorem*. We can use this approach to generate typical data from the model by drawing samples from $p(x \mid y = k)$.

Examples of generative classifiers include:

- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Naive Bayes

1.2 Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)

These classifiers partition $X = \mathbb{R}^d$ into regions with consistent class predictions through separating hyperplanes. The conditional densities are modeled as multivariate normal for each class k:

$$X \sim N(\mu_k, \Sigma_k)$$

Discriminant functions are defined as:

$$g_k(X) = \log(P(X \mid Y = k)) + \log(P(Y = k))$$

For two-class problems, the optimal classifier is:

$$f^* : x \mapsto 2 \cdot \mathbb{1}\{g_1(x) - g_{-1}(x) > 0\} - 1$$

- **LDA** assumes equal covariance across classes ($\Sigma_k = \Sigma$).
- **QDA** allows different covariances in each class, creating quadratic decision boundaries.

1.3 Naive Bayes Classifier

This classifier selects the class that maximizes the posterior probability given the observation:

$$\text{argmax}_k\{p(y = k \mid x)\} = \text{argmax}_k\{p(x \mid y = k)p(y = k)\}$$

Naive Bayes simplifies the computation by assuming conditional independence among features:

$$p(x \mid y = k) = \prod_{i=1}^{d} p(x_i \mid y = k, \theta_{ik})$$

Examples of Naive Bayes models based on feature types:

- **Gaussian distribution** for real-valued features:

$$p(x \mid y = k; \theta) = \prod_{j=1}^{d} \Phi(x \mid \mu, \sigma^2)$$

- **Bernoulli distribution** for binary features:

$$p(x \mid y = k; \theta) = \prod_{j=1}^{d} Ber(x_j \mid \theta_{jk})$$

- **Multinomial distribution** for count data.

1.4 Discriminative Approach

Instead of modeling class-conditional densities, the discriminative approach models the conditional probability $p(y = k \mid x)$ directly, often using a generalized linear model framework.

Examples of discriminative classifiers include:

- Logistic Regression
- K-nearest Neighbors (KNN)
- Support Vector Machines (SVMs)
- Perceptrons

1.5 Logistic Regression

Aims to predict the label $Y \in \{0, 1\}$ based on features $X \in \mathbb{R}^d$. Logistic regression explicitly models the distribution of Y given X.

$$P(Y = 1 \mid X) = \sigma(\langle w, X \rangle + b)$$

where $w \in \mathbb{R}^d$ is a vector of model weights and $b \in \mathbb{R}$ is the intercept, and σ is the sigmoid function defined as:

$$\sigma : z \mapsto \frac{1}{1 + e^{-z}}$$

We define the log-odd ratio as:

$$\log(P(Y = 1 \mid X)) - \log(P(Y = 0 \mid X)) = \langle w, X \rangle + b$$

Thus, we have:

$$P(Y = 1 \mid X) \geq P(Y = 0 \mid X) \iff \langle w, X \rangle + b$$

defining our classification rule (linear classification rule), which requires estimating w and b. These parameters can be estimated by Maximum Likelihood estimation.

For a multi-label classification, we can extend logistic regression. The objective is to predict the label $Y \in \{1, \ldots, M\}$ based on $X \in \mathbb{R}^d$. Softmax regression models the distribution of Y given X.

For all $1 \leq m \leq M$,

$$z_m = \langle w_m, X \rangle + b_m$$

and

$$P(Y = m \mid X) = \text{softmax}(z)_m$$

where $z \in \mathbb{R}^M$, $w_m \in \mathbb{R}^d$ is a vector of model weights and $b_m \in \mathbb{R}$ is an intercept, and

$$\text{softmax}(z_m) = \frac{\exp(z_m)}{\sum_{j=1}^{M} \exp(z_j)}$$

One neuron is a multi-class extension of the logistic regression model.

2 Regression

2.1 Multidimensional Framework

We now assume that the model is given by:

$$Y_i = X_i^T \beta_* + \mathcal{E}_i$$

where $X_i \in \mathbb{R}^d$, $\beta_* \in \mathbb{R}^d$, and where $(\mathcal{E}_i)_{1 \leq i \leq n}$ are independent, identically distributed, and centered.

We can write the model:

$$Y = X\beta_* + \mathcal{E}$$

where:

$$\mathcal{E} = \begin{pmatrix} \mathcal{E}_1 \\ \vdots \\ \mathcal{E}_n \end{pmatrix}, \quad Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_1^T \\ \vdots \\ X_n^T \end{pmatrix} \in \mathbb{R}^{n \times d}$$

The least squares estimator is then:

$$\hat{\beta}_n \in \text{Argmin}_{\beta \in \mathbb{R}^d} \|Y - X\beta\|_2^2$$

2.2 Proposition

If $X^T X$ is invertible, then:

$$\hat{\beta}_n = (X^T X)^{-1} X^T Y$$

Moreover, if $\mathbb{V}[\mathcal{E}_i] = \sigma_*^2$, then:

$$\mathbb{V}[\hat{\beta}_n] = \frac{\sigma_*^2}{n} (X^T X)^{-1}$$

2.3 Proof

For all $\beta \in \mathbb{R}^d$, we have:

$$\|Y - X\beta\|_2^2 = \|Y\|_2^2 + \beta^T X^T X \beta - 2\beta^T X^T Y$$

Hence,

$$\nabla_\beta \|Y - X\beta\|_2^2 = 2X^T X\beta - 2X^T Y$$

The gradient of the squared norm is zero when:

$$\nabla_\beta \|Y - X\beta\|_2 = 0 \Leftrightarrow \hat{\beta} = (X^T X)^{-1} X^T Y$$

Moreover, the variance of $\hat{\beta}_n$ is given by:

$$\mathbb{V}[\hat{\beta}_n] = (X^T X)^{-1} X^T \mathbb{V}[Y] X (X^T X)^{-1}$$

Since $\mathbb{V}[Y] = \sigma_*^2 I_n$, we have:

$$\mathbb{V}[\hat{\beta}_n] = \sigma_*^2 (X^T X)^{-1}$$