Machine Learning in Communications
Lecture 4: Statistical Estimation and its Role in Machine Learning (Classification)

Harpreet S. Dhillon

Wireless@VT, Department of Electrical & Computer Engineering
Virginia Tech, Blacksburg, VA

https://www.dhillon.ece.vt.edu
hdhillon@vt.edu

JTG/IEEE Information Theory Society Summer School
IIT Kanpur

Lecture Objectives

In this lecture, we will complete our discussion of statistical estimation by covering the classification class. The specific topics are:

- The idea of Bayes classifier.
- The idea of a Naïve Bayes classifier.
- Logistic regression and its underlying generative model.
- Connection between logistic regression and Naïve Bayes classifier.

Bayes Classifier

- Remember from Lecture 1 that $h(x) = E[Y|X = x]$ minimizes $E[(Y - h(X))^2]$.
- We now do a similar calculation for the classification case (with 0/1 loss model).
- As before, we assume $(\boldsymbol{x}, y) \sim p(\boldsymbol{x}, y)$.
- Multi-class classification problem: $y \in \{1, 2, \cdots, k\}$.
- We are interested in finding a function $g(\cdot)$ that minimizes the following expected loss:

$$\begin{aligned} E\left[\text{Loss}\right] &= E_{p(\boldsymbol{x},y)}\left[L(y, g(\boldsymbol{x}))\right] \\ &= E_{p(\boldsymbol{x})}\left[\sum_{y=1}^{k} L(y, g(\boldsymbol{x}))p\left(y|\boldsymbol{x}\right)\right]. \end{aligned}$$

  - Note that the function $g(\cdot)$ maps $\boldsymbol{x}$ to the set $\{1, 2, \ldots k\}$.

Bayes Classifier

$$E\left[\text{Loss}\right] = E_{p(\boldsymbol{x})}\left[\sum_{y=1}^{k} L(y, g(\boldsymbol{x}))p\left(y|\boldsymbol{x}\right)\right].$$

- Because of the assumption of the 0/1 loss function, $L(y, g(\boldsymbol{x}))$ will be 0 for one term (for which $y = g(\boldsymbol{x})$) and 1 for all the others.
- Therefore, the above expression can be written as

$$E\left[\text{Loss}\right] = E_{p(\boldsymbol{x})}\left[1 - p(g(\boldsymbol{x})|\boldsymbol{x})\right].$$

- As we did before, we can again minimize this expression point wise to arrive at

$$\hat{y} = \hat{g}(\boldsymbol{x}) = \arg\max_{g} p(g(\boldsymbol{x})|\boldsymbol{x}).$$

- This is a Bayes classifier. Note that we are effectively maximizing the posterior here.
- This is what k-NN directly tries to approximate.

Bayes Classifier

- So, are we done?
- Not so fast! Since we do not have the true distribution, we cannot implement the Bayes classifier directly.
- We need to estimate it. Here are two approaches we will study:
  - Naïve Bayes: First estimate $p(\boldsymbol{x}|y)$ and $p(y)$, and then apply Bayes rule to determine $p(y|\boldsymbol{x})$. It is a *generative* approach. Naïve because it approximates $p(\boldsymbol{x}|y)$.
  - Logistic regression: We directly estimate $p(y|\boldsymbol{x})$. This is a discriminative approach.
- Question: Why do we call the first approach "generative"?
- Let's start with the generative approach.

Why Naïve?

- In order to understand this, let's consider a simple case: $x_i \in \{0,1\}, \forall i$ and $y \in \{1,2,\cdots,k\}$.
- In generative approach, we need to approximate $p(y)$ and $p(\boldsymbol{x}|y)$. Let's see how many parameters do we need to estimate these:
  - Estimating $p(Y=y)$: we need to estimate $k-1$ parameters, i.e., $\{p_1, p_2, \cdots, p_{k-1}\}$, since p_k will be simply $1-\sum_{m=1}^{k-1} p_m$.
  - Estimating $p\left(X_1 = x_1, X_2 = x_2, \cdots, X_d = x_d | Y = y\right)$: we need to estimate $\left(2^d - 1\right) k$ to characterize this distribution. In particular, for every y, we need to learn $2^d - 1$ parameters. This is clearly not feasible even for small values of d.

Naïve Bayes

- Naïve Bayes makes the following conditional independence assumption:

$$p(X_1 = x_1, X_2 = x_2, \cdots, X_d = x_d | Y = y) = \prod_{i=1}^{d} p(X_i = x_i | Y = y).$$

- We observe from above that $p(X_i = x_i | Y = y)$ needs to be estimated. Since $p(X_i = x_i | Y = y)$ is binary distribution, it can be characterized by estimating just one parameter.
- Thus, we need to estimate d parameters for each $Y = y$, and hence the total number of parameters to be estimated is kd, which seems to be doable compared to $(2^d - 1)k$.
- We will revisit Naïve Bayes when we explore its connection with Logistic regression shortly.
- Let's first introduce Logistic regression.

Logistic Regression: Setup

- For logistic regression, we consider the following problem setting:
  - The features vector: $\boldsymbol{x} \in \mathbb{R}^d$. Dataset of feature vectors: $\mathbf{X}$.
  - The output: $y \in \{0, 1\}$.
  - The distribution of the output conditioned on the features vector: $y|\boldsymbol{x} \sim \mathrm{Ber}(\theta_x)$.
- The objective is to characterize $p(y|\boldsymbol{x})$, i.e., we need to estimate θ_x for every x.
- Can we directly use θ_x as $\hat{\theta}_x = \boldsymbol{\beta}^T \boldsymbol{x}$? *Clearly no. Not confined to* $[0, 1]$*.*
- This can be achieved by using the sigmoid function: $\sigma(z) = \frac{1}{1+\exp(-z)}$.
- The distribution of the output conditioned on the features vector is now given by: $y|\boldsymbol{x} \sim \mathrm{Ber}\left(\sigma\left(\boldsymbol{\beta}^T \boldsymbol{x}\right)\right)$.

Logistic Regression

- Using the sigmoid function, we get the following form for the posterior

$$p(y=1|\boldsymbol{x}) = \hat{\theta}_x = \sigma(\boldsymbol{\beta}^T\boldsymbol{x}) = \frac{1}{1+\exp\left(-\boldsymbol{\beta}^T\boldsymbol{x}\right)},$$

$$p(y=0|\boldsymbol{x}) = 1-\hat{\theta}_x = 1-\sigma(\boldsymbol{\beta}^T\boldsymbol{x}) = \frac{1}{1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}\right)},$$

  where our objective reduces to the estimation of the parameters $\boldsymbol{\beta}$ from the data.
- For prediction, we just need to know which probability in larger, i.e., $p(y=1|\boldsymbol{x})$ or $p(y=0|\boldsymbol{x})$.

Logistic Regression

- Under this setup, our predicted output $\hat{y}$ will be 1 if the following condition holds:

$$
\begin{aligned}
&\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x}} \geq 1 \\
\Rightarrow &\log\left[\frac{\frac{1}{1+\exp(-\boldsymbol{\beta}^T\boldsymbol{x})}}{\frac{\exp(-\boldsymbol{\beta}^T\boldsymbol{x})}{1+\exp(-\boldsymbol{\beta}^T\boldsymbol{x})}}\right] \geq 0 \\
\Rightarrow &\log\left[\frac{1}{\exp\left(-\boldsymbol{\beta}^T\boldsymbol{x}\right)}\right] \geq 0 \\
\Rightarrow &\boldsymbol{\beta}^T\boldsymbol{x} \geq 0.
\end{aligned}
$$

- We get a linear classifier.

Logistic Regression: Learning Parameters

Let us recall that we have the following problem setting:

- Model: $y|\boldsymbol{x} \sim \mathrm{Ber}(\theta_x)$.
- Dataset: $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$.
- $p(y=1|\boldsymbol{x}) = \theta_x = \frac{1}{1+\exp(-\boldsymbol{\beta}^T\boldsymbol{x})}$.
- First goal: $\hat{\boldsymbol{\beta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\beta}} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta})$.
- Let's first write the likelihood function:

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^{n} \theta_{\boldsymbol{x}_i}^{y_i} (1-\theta_{\boldsymbol{x}_i})^{1-y_i}$$

Logistic Regression: Log Likelihood

The log likelihood can be expressed as

$$
\begin{aligned}
LL(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \left[y_i \log\left(\theta_{\boldsymbol{x}_i}\right) + (1-y_i)\log\left(1-\theta_{\boldsymbol{x}_i}\right) \right] \\
&= \sum_{i=1}^{n} \left[y_i \log\left(\frac{1}{1+\exp\left(-\boldsymbol{\beta}^T\boldsymbol{x}_i\right)}\right) + (1-y_i)\log\left(\frac{1}{1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)}\right) \right] \\
&= \sum_{i=1}^{n} \left[y_i \log\left(\frac{\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)}{1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)}\right) + (1-y_i)\log\left(\frac{1}{1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)}\right) \right] \\
&= \sum_{i=1}^{n} \left[y_i \log\left(\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)\right) - \log\left(1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)\right) \right] \\
&= \sum_{i=1}^{n} \left[y_i \boldsymbol{\beta}^T\boldsymbol{x}_i - \log\left(1+\exp\left(\boldsymbol{\beta}^T\boldsymbol{x}_i\right)\right) \right].
\end{aligned}
$$

Concave function in $\boldsymbol{\beta}$. Use gradient descent on $-LL(\boldsymbol{\beta})$.

Logistic Regression: MAP Case

- After completing ML estimator, our next step is to do MAP estimator.
- The MAP estimator can be obtained as follows

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\beta}} \log\left(p\left(\boldsymbol{\beta}|\mathbf{y},\mathbf{X}\right)\right) \\ &= \arg\max_{\boldsymbol{\beta}} \left[\log\left(p\left(\mathbf{y}|\boldsymbol{\beta},\mathbf{X}\right)\right) + \log\left(p(\boldsymbol{\beta})\right)\right],\end{aligned}$$

where we use Gaussian prior $\boldsymbol{\beta} \sim \mathcal{N}\left(0, \sigma_o^2 I\right)$, i.e., we have

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_o^2}} \exp\left(\frac{-\beta_j^2}{2\sigma_o^2}\right).$$

Logistic Regression: MAP Case

This gives us:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log\left(p\left(y_i|\boldsymbol{\beta}, \boldsymbol{x}_i\right)\right) + \sum_{j=0}^{d} \log\left(p(\beta_j)\right) \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \log\left(p\left(y_i|\boldsymbol{\beta}, \boldsymbol{x}_i\right)\right) + \sum_{j=0}^{d} \left[\frac{-\beta_j^2}{2\sigma_o^2} - \underbrace{\frac{1}{2}\log\left(2\pi\sigma_o^2\right)}_{\text{Not function in } \boldsymbol{\beta}}\right] \\
&= \arg\max_{\boldsymbol{\beta}} LL(\boldsymbol{\beta}) - \frac{1}{2\sigma_o^2}\left\|\boldsymbol{\beta}\right\|_2^2.
\end{aligned}
$$

As in the case of linear regression, we recover a regularization term.

Nonlinear Decision Boundaries with Logistic Regression

Remember our discussion on polynomial regression.
Let's construct a similar example for logistic regression too.

Connection Between Gaussian NB and Logistic Regression

Now, let's connect Gaussian Naïve Bayes to logistic regression. We consider the following setting:

- $Y = y \in \{0, 1\}$.
- $p(Y = 1) = \theta$ and $p(Y = 0) = 1 - \theta$.
- Naïve Bayes: $p(\boldsymbol{x}|y) = \prod_{j=1}^{d} p(x_j|y)$.
- The distribution of the j^{th} feature x_j conditioned on the i^{th} label y_i is normal with mean μ_{ji} and variance σ_j^2, i.e., $p\left(x_j|Y = y_i\right) = \mathcal{N}\left(\mu_{ji}, \sigma_j^2\right)$.

Connection Between Gaussian NB and Logistic Regression

For this setting, we derive $p(y=1|\boldsymbol{x})$ as follows

$$\begin{aligned} p(y=1|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y=1)p(y=1)}{p(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x}|y=1)p(y=1)}{\sum_y p(\boldsymbol{x}|y)p(y)} \\ &= \frac{1}{1+\frac{p(\boldsymbol{x}|y=0)p(y=0)}{p(\boldsymbol{x}|y=1)p(y=1)}}. \end{aligned}$$

Connection Between Gaussian NB and Logistic Regression

Taking the exp log for the term in the denominator of the above expression, we get

$$
\begin{aligned}
p(y=1|\boldsymbol{x}) &= \frac{1}{1+\exp\left[\log\left(\frac{p(\boldsymbol{x}|y=0)p(y=0)}{p(\boldsymbol{x}|y=1)p(y=1)}\right)\right]} \\
&= \frac{1}{1+\exp\left[\log\left(\frac{p(\boldsymbol{x}|y=0)}{p(\boldsymbol{x}|y=1)}\right)+\log\left(\frac{p(y=0)}{p(y=1)}\right)\right]} \\
&= \frac{1}{1+\exp\left[\log\left(\prod_{j=1}^{d}\underbrace{\frac{p(x_j|y=0)}{p(x_j|y=1)}}_{\tau}\right)+\log\left(\frac{1-\theta}{\theta}\right)\right]}.
\end{aligned}
$$

Connection Between Gaussian NB and Logistic Regression

Now let's look at the term τ carefully:

$$\tau = \frac{p(x_j|y=0)}{p(x_j|y=1)} = \frac{\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\left(\frac{-(x_j-\mu_{j0})^2}{2\sigma_j^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\left(\frac{-(x_j-\mu_{j1})^2}{2\sigma_j^2}\right)}.$$

Connection Between Gaussian NB and Logistic Regression

Taking the log of the above expression, we get

$$\begin{aligned}
\log(\tau) &= \frac{-\left(x_j - \mu_{j0}\right)^2}{2\sigma_j^2} + \frac{\left(x_j - \mu_{j1}\right)^2}{2\sigma_j^2} \\
&= \frac{-x_j^2 - \mu_{j0}^2 + 2x_j\mu_{j0} + x_j^2 + \mu_{j1}^2 - 2x_j\mu_{j1}}{2\sigma_j^2} \\
&= \underbrace{\left(\frac{2\mu_{j0} - 2\mu_{j1}}{2\sigma_j^2}\right) x_j}_{\text{Linear}} + \underbrace{\frac{\mu_{j1}^2 - \mu_{j0}^2}{2\sigma_j^2}}_{\text{Constant}} \\
&= -\underbrace{\left(\frac{2\mu_{j1} - 2\mu_{j0}}{2\sigma_j^2}\right)}_{\beta_j} x_j - \text{Constant}.
\end{aligned}$$

Using this, we recover the logistic regression form:

$$p(y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{\beta}^T \boldsymbol{x}\right)}.$$

Summary

This concludes our discussion on statistical estimation and its role in machine learning.

Today's lecture focused on the estimation problem. Specifically, we covered:

- The idea of Bayes classifier.
- The idea of a Naïve Bayes classifier.
- Logistic regression and its underlying generative model.
- Connection between logistic regression and Naïve Bayes classifier.