MA2823: Foundations of Machine Learning
Chapter 4: Bayesian Learning Theory

Lecturer: Chloé-Agathe Azencott

Scribes: LOH Mathias
KIM Taehyun
MEHDI Tomas

---

1 Introduction

In this chapter, we aim to *apply Bayes rule* for simple inference and decision problems and explain the correlation between *Bayes decision rule, empirical risk minimization, maximum a priori and maximum likelihood.* In doing so, express conditional independence among random variables and finally, we aim to apply *Naive Bayes algorithm*.

2 Bayes Decision Theory

2.1 Probability and inference

Definition. We define a random variable $X : \Omega \to E$ an application from the set of possible outcome Ω to some set E.

Although X is usually a real-valued function ($E = \mathbb{R}$), it *does not* return a probability. Rather, X describes some numerical property that outcomes in Ω may have - e.g, the number of heads in a random collection of coin flips. The probability that X takes value ≤ 3 is the measure of the set of outcomes $\{\omega \in \Omega : X(\omega) \leq 3\}$, denoted $P(x \leq 3)$

Example (Coin toss). Let us suppose the result of tossing a coin is $x \in \{heads, tails\}$. Because this is a random process, we have *incomplete information about it*. Since the result is either heads or tails, we can define this as a random variable X to follow that of a *Bernoulli distribution*, defined by

$$X = \begin{cases} 1, & \text{outcome is heads} \\ 0, & \text{outcome is tails} \end{cases}$$

To find the probability of obtaining x, we just have to find the probability $P(X = x)$

2.2 Classification

Definition. We define $C = (C_1, ..., C_k) \in \mathbb{R}^k$ the family of classes in our classification problem and a family of random variables $\boldsymbol{x} = (X_1, ..., X_k)$ observable data which we believe is sufficient in giving us ample information about each of our classes C_i.

Example (Credit-scoring). In banks, according to their past transactions, some customers are *low-risk* in that they paid back their loans and the bank profited from them and other customers are *high-risk* in that they defaulted. The bank would thus like to learn the class "high-risk customer"so that in the future, when there is a new application for a loan, we can check that the person obeys the class description or not and thus accept or reject the application.

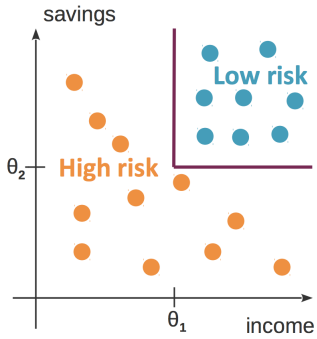

Figure 1: A classification problem with two pieces of observable information *income* and *savings*

For the simplicity of this problem, let us suppose that only two pieces of information are observable (and we observe them because we have reason to believe they give us an idea bout their credibility). These two information are *income* and *savings*, which we represent by two random variables X_1 and X_2. The credibility of a customer is denoted by a Bernoulli random variable C conditioned on the observable $\boldsymbol{x} = [X_1, X_2]^T$ where $C = 1$ indicates a high risk customer and $C = 0$ indicates a low risk customer. Thus if we know $P(C|X_1, X_2)$, when a new application arrives with $X_1 = x_1$ and $X_2 = x_2$, we can

$$\text{Choose} \begin{cases} C = 1, & \text{if } P(C = 1|x_1, x_2) > 0.5 \\ C = 0, & \text{otherwise} \end{cases}$$

or equivalently

$$\text{Choose} \begin{cases} C = 1, & \text{if } P(C = 1|x_1, x_2) > P(C = 0|x_1, x_2) \\ C = 0, & \text{otherwise} \end{cases}$$

2.3 Decision rules

2.3.1 Bayes decision rule

Theorem. *Bayes Theorem. We say the probability of obtaining a class C given knowledge of $\boldsymbol{x}$*

$$P(C|\boldsymbol{x}) = \frac{P(C)p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$$

where

- $P(C|\boldsymbol{x})$ *defined as the posterior*
- $P(C)$ *defined as the prior,*
- $p(\boldsymbol{x}|C)$ *defined as the likelihood*
- $p(x)$ *defined as the evidence*

Indeed, we have in fact already seen Bayes' decision rule for a simplified case. Bringing us back to the problem of credit-scoring, the *Bayes' decision rule* is simply

$$\text{Choose} \begin{cases} C = 1, & \text{if } P(C = 1|\boldsymbol{x} = (x_1, x_2)) > P(C = 0|\boldsymbol{x} = (x_1, x_2)) \\ C = 0, & \text{otherwise} \end{cases}$$

2.3.2 MAP decision rule

For this decision rule, we want to pick the hypothesis that is the most probable, i.e., maximize the posterior $P(C|\boldsymbol{x}) = \frac{P(C)p(\boldsymbol{x}|C)}{p(\boldsymbol{x})}$. Again, in a 2 classifier problem, we denote $\Lambda_{\text{MAP}}(\boldsymbol{x}) = \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})}$. The MAP decision rule thus state

$$\text{Choose } \begin{cases} C = 1, & \text{if } \Lambda_{\text{MAP}}(\boldsymbol{x}) > 1 \\ C = 0, & \text{otherwise} \end{cases}$$

2.3.3 Likelihood ratio test (LRT)

For this decision rule, we would want to test whether the likelihood ratio $\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C=1)}{p(\boldsymbol{x}|C=0)}$ is larger than or equal to ratio of priors $\frac{P(C=0)}{P(C=1)}$, i.e.

$$\text{Choose } \begin{cases} C = 1, & \text{if } \frac{P(\boldsymbol{x}|C=1)}{P(\boldsymbol{x}|C=0)} > \frac{P(C=0)}{P(C=1)} \\ C = 0, & \text{otherwise} \end{cases}$$

Example (LRT with Gaussian). *Suppose we have $P(\boldsymbol{x}|C = 1) \sim N(4, 1)$ and $P(\boldsymbol{x}|C = 0) \sim N(10, 1)$ two Gaussian likelihoods with equal priors.* ***Derive a decision rule based on the LRT.***

We first note that priors are equal, our decision rule hence becomes

$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C = 1)}{p(\boldsymbol{x}|C = 0)} = \frac{e^{-(x-4)^2/2}}{e^{-(x-10)^2/2}} > \frac{P(C = 0)}{P(C = 1)} = 1$$

The condition that determines the decision boundary is therefore $\frac{e^{-(x-4)^2/2}}{e^{-(x-10)^2/2}} > 1$. By taking the logarithm on the above equation, we thus obtain:

$$log(\Lambda(\boldsymbol{x})) = 84 - 12x > 0$$

Which gives the decision boundary $x < 7$, i.e.

$$\text{Choose } \begin{cases} C = 1, & \text{if } x < 7 \\ C = 0, & \text{otherwise} \end{cases}$$

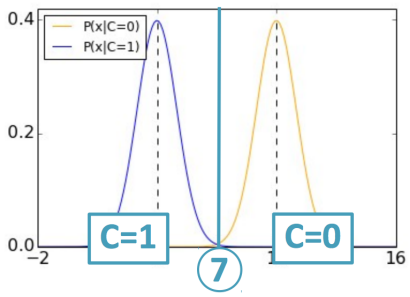

Figure 2: Graphical solution of the LRT with Gaussian.

2.3.4 Bayes' rule for multiple classes

In the general case, we have K mutually exclusive and exhaustive classes; $C_i, i = 1, ..., K$. We have prior probabilities satisfying

$$P(C_i) \geq 0 \quad \text{and} \quad \sum_{i=1}^{K} P(C_i) = 1$$

$p(\boldsymbol{x}|C_i)$ is the probability of seeing $\boldsymbol{x}$ as the input when it is known to belong to class C_i. The posterior probability of class C_i can be calculated as

$$P(C_k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|C_k)P(C_k)}{\sum_{1 \leq l \leq K} P(\boldsymbol{x}|C_l)P(C_l)}$$

and for minimum error, the *Bayes' classifier* chooses the class with the highest posterior probability; that is, we

$$\text{Choose } C_i \text{ if } P(C_i|\boldsymbol{x}) = \max_k P(C_k|\boldsymbol{x})$$

2.4 Losses and risks

It may be the case that decisions that we make are not equally good or costly. A financial firm for example, must take into account the potential gain and loss when making a decision for a loan applicant. An accepted low-risk appplicant increases profit, while a rejected high-risk applicant decreases loss. In the following sections we could consider these losses and risks and we consider that there exists K different classes classes, noted $C_i, i = 1, ..., K$.

2.4.1 Penalization

Definition. Let us define action α_i as the decision to assign the input to class C_i and λ_{kl} as the *quantity of loss* for taking action α_k when the input actually belongs to C_l. The *expected risk* is the sum of the risks for all the classes that the point could belong to times the probability that it belongs to the given class, that is,

$$R(\alpha_k|\boldsymbol{x}) = \sum_{l=1}^{K} \lambda_{kl} P(C_l|\boldsymbol{x})$$

We note that this becomes the case of choosing the *actions with minimal risks*, i.e., we choose α_i if $R(\alpha_i|\boldsymbol{x}) = \min_k R(\alpha_k|\boldsymbol{x})$ or alternatively

$$R(\alpha_k|\boldsymbol{x}) = \arg\min_R (\alpha_k|\boldsymbol{x})$$

2.4.2 0/1 Loss

Definition. Let us define K actions $\alpha_i, i = 1, ..., K$. In the special case of the 0/1 *loss case* where

$$\lambda_{kl} = \begin{cases} 0, & \text{if } k = l \\ 1, & \text{if } k \neq l \end{cases}$$

where all correct decisions have no loss and all errors are equally costly. The risk of taking action α_i is

$$R(\alpha_i|\boldsymbol{x}) = 1 - P(C_i|x)$$

Thus to minimize risk, we choose the most probable class. Note that this is a *special case* and rarely do applications have a symmetric 0/1 loss. This however serves as a simple model for analysis in this course!

2.4.3 Reject

In some applications, wrong decisions may have very high cost, and it is generally required that a more complex (e.g, manual classification) be made if automation dictates a *low certainty of its decision*. We thus create an additional *action of reject* α_{K+1} which is the option to not take any decision. A possible loss function is thus (following the example of 0/1 loss),

$$\lambda_{kl} = \begin{cases} \lambda \text{ if } k = K+1 \\ 0 \text{ if } k = l \\ 1 \text{ otherwise} \end{cases}$$

The decision is therefore taken according to:

$$\begin{cases} \text{Choose } C_l, & \text{if } R(\alpha_l|\boldsymbol{x}) < R(\alpha_k|\boldsymbol{x})\, \forall k \neq l \textbf{ and } R(\alpha_l|\boldsymbol{x}) < R(\alpha_{K+1}|\boldsymbol{x}) \\ \text{reject} & \text{otherwise} \end{cases}$$

We choose $0 \leq \lambda \leq 1$, otherwise $\lambda = 0$ will always reject whereas $\lambda = 1$ will never reject!

2.4.4 Bayes Risk

We have thus far seen the different ways we could construct the risks of our problems. In this section, we introduce the notion of the total expected risk that we should expect in a classification problem.

Definition. We call **Bayes' risk** the overall expected risk, the sum of all risks of taking any action α_k:

$$R(\boldsymbol{x}) = \sum_{k=1}^{K} R(\alpha_k|\boldsymbol{x}) = \sum_{k=1}^{K}\sum_{l=1}^{K} \lambda_{kl} P(C_l|\boldsymbol{x})$$

2.4.5 Discriminant functions

Definition. Classification can also be seen as implementing a set of *discriminating functions*, $f_i(\boldsymbol{x}), i = 1, ..., k$, such that we choose C_i if $f_i(\boldsymbol{x}) = \max_k f_k(\boldsymbol{x})$. We represent the Bayes' classifier by setting:

$$f_k(\boldsymbol{x}) = -R(\alpha_k|\boldsymbol{x})$$

Furthermore, we can define these discriminant functions into K *decision regions* $\mathcal{R}_1, ..., \mathcal{R}_K$, which separated by *decision boundaries* which separates one class from another. These decision regions are given by

$$R_k = \{x : f_k(\boldsymbol{x}) = \max_l f_l(\boldsymbol{x})\}$$

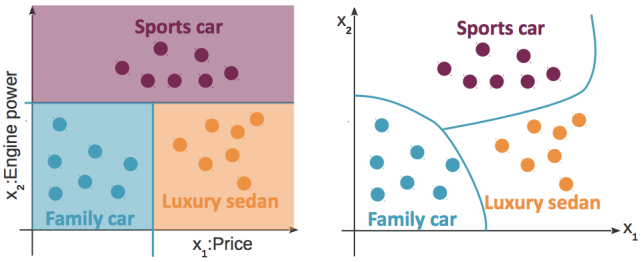

Figure 3: Example of 3 decision boundaries

2.4.6 Supplement: losses for regression

We introduce the following losses commonly seen and used in regression

- **Square loss**: $L(f(\boldsymbol{x}), y) = (f(\boldsymbol{x}) - y)^2$.
  This loss functions has the problem of penalizing heavily outliers, which results in increasing the numerical error with unclean data. The two next loss functions seek to solve that problem.
- **ϵ - insensitive loss**: $L(f(\boldsymbol{x}), y) = (|f(\boldsymbol{x}) - y| - \epsilon)_+$
- **Huber Loss**: $L_\delta(f(\boldsymbol{x}), y) = \begin{cases} \frac{1}{2}(y - f(\boldsymbol{x}))^2, & \text{if } |y - f(\boldsymbol{x})| \leq \delta \\ \delta|y - f(\boldsymbol{x})| - \delta^2/2, & \text{otherwise} \end{cases}$
  However, these two functions are not smooth, which can cause problems in algorithms which seek to minimize them.

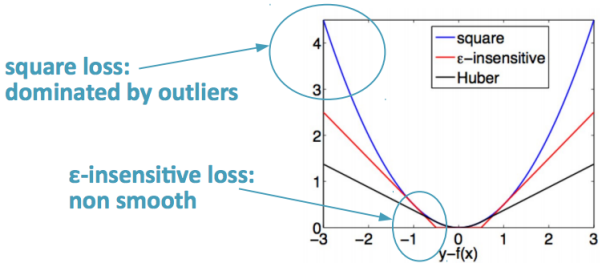

Figure 4: Summary of regression errors

2.5 Empirical risk minimization

The main idea is that the loss function $L(f(\boldsymbol{x}), y)$ is small when $f(\boldsymbol{x})$ predicts y well. We have

- Expected risk: $R = \mathbb{E}[L(f(\boldsymbol{x}), y)]$
- Empirical risk: $R_n = \frac{1}{n}\sum_{i=1}^n L(f(\boldsymbol{x}^i), y^i)$
- ERM estimator of the functional class F is the solution, when it exists, of $\hat{f}_n = \arg\min_{f \in \mathcal{F}} R_n(f)$

However ERM is ill posed and it can be that an infinite number of solutions minimize the empirical risk to zero. ERM is also not statistically consistent. Law of large numbers is only true if capacity (VC-dimension) of f is not too large.

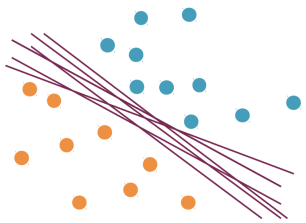

Figure 5: Ill posed problem: infinite solutions!

3 Multivariate classification: Naive Bayes

3.1 Naive Bayes

Naive Bayes is a *multivariate classification*, which is to say, $\boldsymbol{x}$ is multidimensional. We also have an assumption that the variables $\boldsymbol{x} = (x_1, x_2, ..., x_p)$ are conditionally independent, i.e.,

$$p(x_i|C_k, x_j) = p(x_i|C_k)$$

By applying this property to Bayes' rule that we have seen previously,

$$\begin{aligned} P(C_k|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|C_k)P(C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)} \\ &= \frac{P(C_k)p(x_1|C_k)p(x_2|C_k)\cdots p(x_p|C_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)} \\ &= \frac{1}{Z}P(C_k)p(x_1|C_k)p(x_2|C_k)\cdots p(x_p|C_k) \end{aligned}$$

Where we note $\frac{1}{Z}$ a scaling factor independent of C_k.

3.2 Algorithms

3.2.1 Maximum a posteriori (MAP) estimation

We recall in a *MAP decision rule*, we pick the hypothesis that is most probable (Homework 3!). For Naive Bayes, the *MAP decision rule* is defined by

$$\hat{y} = \arg \max_{k=1,..,K} p(C_k)\Pi_{i=1}^{n} p(\boldsymbol{x}^i|C_k)$$

3.2.2 Bernoulli Naive Bayes

In the case of a *binary classification problem* (2 classes), we can define each sample as the outcome of p Bernoulli trials, with the assumption that each trials are *independent* from the others. Then the decision rule for Bernoulli Naive Bayes is based on

$$p(x_j|C_k) = p_j^{x_j}(1 - p_j)^{(1-x_j)}$$

3.2.3 Gaussian Naive Bayes

We further assume that $p(x_j|C_k)$ is a *univariate Gaussian* and the decision rule is defined as

$$p(x_j|C_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

where μ and σ are estimated using maximum likelihood.

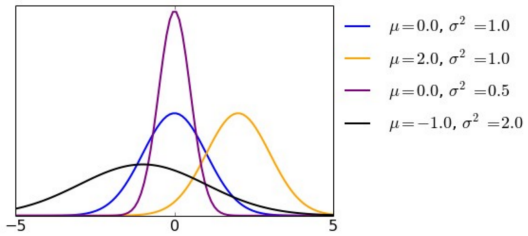

Figure 6: Naive Gaussian examples with varying μ and σ

3.3 Bayesian model selection

The key idea of this section is that we can also take priors on models, and we denote it as $p(\text{model})$. Similar to previous sections, we can thus define a decision rule

$$p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})p(\text{model})}{p(\text{data})}$$

In addition, if we take the log on both sides of the equation, we obtain,

$$\log p(\text{model}|\text{data}) = \log p(\text{data}|\text{model}) + \log p(\text{model}) - \log p(\text{data})$$

where we can further define $\log p(\text{data}|\text{model})$ as the *training error* and $\log p(\text{model})$ the *complexity* of our model. We can also take the maximum a posteriori (MAP) which is similar to minimizing

$$E' = \text{empirical error } + \boldsymbol{\lambda} \text{ model complexity}$$