STAT 525

10/17/23

Lecture 6

Classifiers:

- Generative vs discriminative
- Loss functions
  empirical loss, Bayes loss
- Bias & Variance
  - for parameter estimation
  - what's different for prediction

Q1 Thursday
beginning of class

HW2 – due Monday
10/23 11:59pm

Lecture II: Prediction – Basic concepts

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

October, 2023

Parametric vs non-parametric ✓

Generative and discriminative models for classification ←

- Generative classifiers
- Discriminative classifiers
- Generative vs discriminative classifiers

Loss functions ←

- Bayes loss

Variance, bias and complexity ←

Reading HTF Ch.: 2.1–5,2.9, 7.1–4 bias-variance tradeoff, Murphy Ch.: 1., 8.6[1], Bach Ch.:

[1] Neither textbook is close to these notes except in a few places; take them as alternative perspectives or related reading

The “learning” problem

- **Given**
- a problem (e.g. recognize digits from $m \times m$ gray-scale images)
- a **sample** or (**training set**) of **labeled data**

  $$\mathcal{D} = \{(x^1, y^1), (x^2, y^2), \ldots (x^n, y^n)\}$$

  drawn i.i.d. from an unknown P_{XY}
- **model class** $\mathcal{F} = \{f\}$ = set of predictors to choose from

- **Wanted**
- a predictor $f \in \mathcal{F}$ that performs well on future samples from the same P_{XY}

  - “choose a predictor $f \in \mathcal{F}$” = training/learning
  - “performs well on future samples” (i.e. f **generalizes** well) – how do we measure this? how can we “guarantee” it?
  - choosing $\mathcal{F}$ is the **model selection problem** – about this later

A zoo of predictors

- Linear regression ✓
- Logistic regression
- Linear Discriminant (LDA) ✓
- Quadratic Discriminant (QDA) ✓
- CART (Decision Trees) Discriminative
- K-Nearest Neighbors — " —
- Nadaraya-Watson (Kernel regression) ← NW classifier ?
- Naive Bayes ✓ generative
- Neural networks/Deep learning D
- Support Vector Machines D
- Monotonic Regression

Generative classifiers

One way to define a classifier is to assume that each class is generated by a distribution $g_y(X) = P(X|Y = y)$. If we know the distributions g_y and the class probabilities $P(Y = y)$, we can derive the *posterior probability* distribution of Y for a given x. This is

$$P(Y = y|X) \;=\; \frac{P(Y = y)g_y(X)}{\sum_{y'} P(Y = y')g_{y'}(X)} \;=\; \frac{P(Y = y)g_y(X)}{P(X)} \tag{1}$$

The "best guess" for $Y(X)$ (i.e. the decision rule) is

$$f(X) = \mathrm{argmax}_y P(Y = y|x) = \mathrm{argmax}_y P(Y = y)g_y(x) \tag{2}$$

- (1) amounts to a likelihood ratio test for Y.
- The functions $g_y(x)$ are known as **generative models** for the classes y.
  Therefore, the resulting classifier is called a **generative classifier**.
  Examples: LDA, QDA, Naive Bayes.
- In contrast, a classifier defined directly in terms of $f(x)$ (or $P_{Y|X}$), like the linear, quadratic, decision tree is called a **discriminative classifier**.
- In practice, we may not know the functions $g_y(x)$, in which case we estimate them from the sample $\mathcal{D}$.

Generative classifiers (Binary)

given $\mathcal{D} = \{ (x^i, y^i), i=1:n \}$ $\quad y^i \in \{\pm 1\}$

1. fit g_+ "=" $P_{X|y=+}$ ← generative models

g_- "=" $P_{X|y=-}$

↑ density or PMF

Ex: classify objects by images $\quad g_{car} = f(\text{image} \mid \text{car})$ ← density estimation

$g_{person} = f(\text{-''-} \mid \text{person})$

2. $p_\pm = \frac{n_\pm}{n}$ $\quad n_\pm = |\{x^i \mid y^i = \pm\}|$

counts

3. Bayes' rule

$$P[y=+ \mid x] = \frac{\overset{\text{prior}}{p_+}\ \overset{\text{likelihood}}{g_+(x)}}{p_+ g_+(x) + p_- g_-(x)} \Rightarrow \hat{y}(x) = \underset{y}{\operatorname{argmax}} \underbrace{P[y|x]}_{\text{confidence}}$$

Ex: • Extends to any $y \in \{1, 2, \dots m\}$

multiclass

Generative classifier and the likelihood ratio

$$P(Y = y|X) = \frac{P(Y = y)g_y(X)}{\sum_{y'} P(Y = y')g_{y'}(X)} = \frac{P(Y = y)g_y(X)}{P(X)}$$

$$f(x) = \mathrm{argmax}_y P(Y = y|x) = \mathrm{argmax}_y g_y(x)P(Y = y)$$

Likelihood Ratio test (for $y \in \{\pm 1\}$)

$$\frac{g_+(x)P(Y = +)}{g_-(x)P(Y = -)} = \frac{P[+|x]}{P[-|x]}$$

Example (Fisher's LDA in one dimension)

Assume $Y = \pm 1$, $g_y(x) = N(x, \pm\mu, \sigma^2 I)$, i.e each class is generated by a Normal distribution with the same spherical covariance matrix, but with a different mean. Let $P(Y = 1) = p \in (0, 1)$. Then, the posterior probability of Y is

$$P(Y = 1|x) \propto pe^{-||x-\mu||^2/(2\sigma^2)} \quad P(Y = -1|x) \propto (1-p)e^{-||x+\mu||^2/(2\sigma^2)} \tag{3}$$

and $f(x) = 1$ iff $\ln P(Y = 1|x)/P(Y = -1|x) \geq 0$, i.e iff

$$\ln\frac{p}{1-p} - \frac{1}{2\sigma^2}[||x^2|| - 2\mu^T x + ||\mu||^2 - ||x^2|| - (2\mu)^T x - ||\mu||^2] = \left(\frac{2\mu}{\sigma^2}\right)^T x + \ln\frac{p}{1-p} \geq 0 \tag{4}$$

Hence, the classifier $f(x)$ turns out to be a linear classifier. The decision boundary is perpendicular to the segment connecting the centers $\mu, -\mu$. This classifier is known as **Fisher's Linear Discriminant**. [**Exercises** Show that if the generative models are normal with different variances, then we obtain a quadratic classifier. What happens if the models g_y have the same variance, but it is a full covariance matrix Σ?]

$$g_{\pm} = N(\mu_{\pm}, \sigma^2 I)$$

same covariance

Discriminative classifiers

- Defined directly in terms of $f(x)$ or (almost) equivalently, in terms of the decision boundary $\{f(x) = 0\}$
- Can be classified by the shape of the decision boundary (if it's simple)
  - linear, polygonal, quadratic, cubic,. . .

The ambiguity of "linear classifier"

Does it mean $f(x) = \beta^T x$ OR $\{f(x) = 0\}$ is a hyperplane ?
If we talk about **classification** and the domain of x is $\mathbb{R}^d$, then "linear" refers to decision boundary. Otherwise it refers to the expression of $f(x)$. Exercise Find examples when the two definitions are not equivalent

- Can be grouped by model class (obviously)
  - Neural network, K-nearest neighbor, decision tree, . . .
    Exercise Is logistic regression a generative or discriminative classifier?
- By method of training (together with model class)
  - For example, Perceptron algorithm, Logistic Regression, (Linear) Support Vector Machine (see later), Decision Tree with 1 level are all linear classifiers, but usually produce different decision boundaries give a $\mathcal{D}$

A comparison of generative and discriminative classifiers

Advantages of generative classifiers

- Generative classifiers are statistically motivated ✓
- Generative classifiers are *asymptotically optimal* ✓

$g_y = P(x|y)$

$g_y \longrightarrow$ True $P_{x|y}$

Theorem

If $Y \in \{\pm 1\}$*, the model class* G_y *in which we are estimating* g_y *contains the true distributions* $P(X|Y = y)$ *for every* y*, and* $g_y = P(X|Y), P(Y = y)$ *are estimated by Maximum Likelihood then the expected loss*[2] *of the generative classifier* f_g *given by (2) tends to the Bayes loss when* $n \to \infty$*, i.e* $\lim_{n\to\infty} L_{01}(f_g) \leq \min_{f \in \mathcal{F}} L_{01}(f)$*. Here* $\mathcal{F}$ *is the class of likelihood ratio classifiers obtainable from* g_y*'s in* $\mathcal{G}_y$*.*

- The log-likelihood ratio $\ln \frac{P(Y=1|x)}{P(Y=-1|x)}$ is a natural confidence measure for the label at $f_g(x)$. The further away from 0 the likelihood ratio, the higher the confidence that the chosen y is correct.
- Generative classifiers extend naturally to more than two classes. If a new class appears, or the class distribution $P(Y)$ changes, updating the classifier is simple and computationally efficient.
- Often it is easier to pick a (parametric) model class for g_y than an f directly. Generative models are generally more intuitive, while often representing/visualizing decision boundaries between more than two classes is tedious.

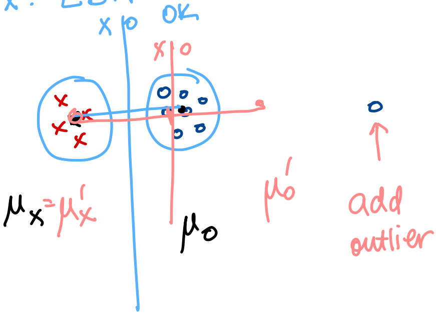
[2] Loss, Bayes loss, $L01$ are defined in the next section.

Advantages of discriminative classifiers

- Generative models offer no guarantees if the true g_y aren't in the chosen model class, whereas for many classes of discriminative models there are guarantees.
- Many discriminative models have performance guarantees for any sample size n, while generative models are only guaranteed for large enough n
- Discriminative classifiers offer many more choices (but one must know how to pick the right model)
- Generative models do not use data optimally in the non-asymptotic regime (when $n \ll \infty$). This has been confirmed practically many times, as discriminative classifiers have been very successful for limited sample sizes

Exercise LDA vs Logistic regression: Experiment with LDA vs LR when data comes from 2 Normal distributions, with outliers. What outliers affect which method more? Experiment also on a toy data set like the one in the lecture notes.

Ex: LDA



LR

$$\beta^T x = f(x) = \ln \frac{P[1|x]}{P[-1|x]}$$

models this

not these or $P_{x|y}$ separately

⇒ LR not generative

Loss functions

The **loss function** represents the cost of error in a prediction problem. We denote it by L, where

$$L(\overset{\text{true}}{y}, \underset{\text{predicted}}{\hat{y}}) = \text{the cost of predicting } \hat{y} \text{ when the actual outcome is } y$$

Note that sometimes the loss depends on x directly. Then we would write it as $L(y, \hat{y}, x)$.

As usually $\hat{y} = f(x)$ or $\operatorname{sgn} f(x)$, we will typically abuse notation and write $L(y, f(x))$.

We assume $L \geq 0$ always

Least Squares (LS) loss

Regression

The **Least Squares (LS)** (or **quadratic**) loss function is given by

$$L_{LS}(y, f(x)) = \frac{1}{2}(y - f(x))^2 \tag{5}$$

This loss is commonly associated with regression problems.

Example: L_{LS} is the log-likelihood of a regression problem (linear or not) with Gaussian noise.

- $L_1(y,\hat{y}) = |y - \hat{y}|$ ← "induces sparsity" $\|\beta\|_1$

Robust

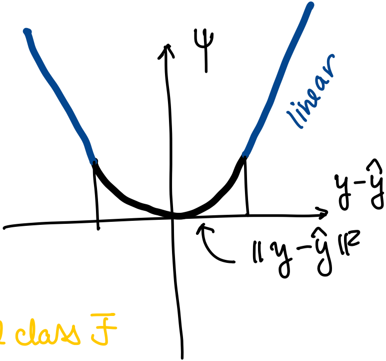
- Huber

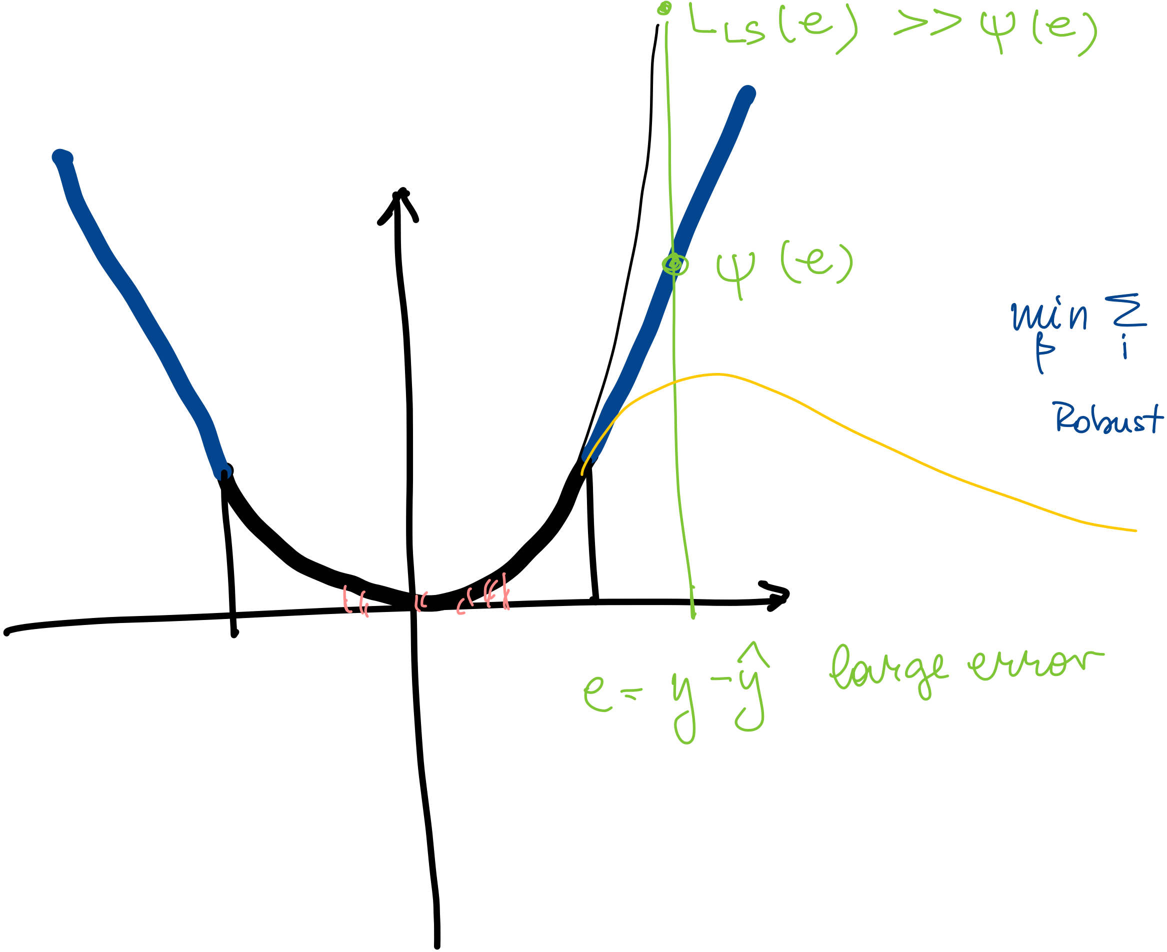
$L_H = \psi(y - \hat{y})$

- $-$Log likelihood

$L_{\log L} = -\ln P(y|x)$

in model class $\mathcal{F}$





$$\min_{\beta} \sum_{i} \psi(y^i - \beta^T x)$$

Robust linear regression

Loss functions for classification

For classification, a natural loss function is the **misclassification error** (also called **0-1 loss**)

$$L_{01}(y, f(x)) \;=\; 1_{[y \neq f(x)]} \;=\; \begin{cases} 1 & \text{if } y \neq f(x) \\ 0 & \text{if } y = f(x) \end{cases} \quad \text{mistake} \tag{6}$$

Sometimes different errors have different costs. For instance, classifying a HIV+ patient as negative (**a false negative** error) incurs a much higher cost than classifying a normal patient as HIV+ (**false positive** error). This is expressed by **asymmetric misclassification costs**. For instance, assume that a false positive has cost one and a false negative has cost 100. We can express this in the matrix

$f(x)$:	+	−
true :+	0	100 ← miss
−	1	0

↑ false detection | false alarm

In general, when there are p classes, the matrix $L = [L_{kl}]$ defines the loss, with L_{kl} being the cost of misclassifying as l an example whose true class is k.

Multiclass

$y \in \{1, .. m\}$

$$L = \begin{array}{cc|c|c|c|c|} & \hat{y} & 1 & 2 & \cdot & m \\ \hline \text{True } y & 1 & 0 & & & \\ \hline & 2 & & 0 & & \\ \hline & \cdot & & & 0 & \\ \hline & m & & & & 0 \\ \hline \end{array}$$

$L_{y\hat{y}}$ = cost of guessing $\hat{y}$ when y true

Expected loss and empirical loss

- Objective of prediction = to minimize expected loss on future data, i.e.

$$\text{minimize } \underbrace{L(f)}_{f\in\mathcal{F}} = \underbrace{E_{P(X,Y)}}_{\text{nature}}[\overset{\text{problem}}{\underline{L(Y,f(X))}}] \text{ over } f \in \mathcal{F} \text{ model class} \qquad (7)$$

We call $L(f)$ above **expected loss**.

Example (Misclassification error $L_{01}(f)$)

$L_{01}(f)$ = probability of making an error on future data.

$$L_{01}(f) = P[\underbrace{Yf(X)}_{\operatorname{sgn} f(x) \neq y \text{ mistake}} < 0] = E_{P_{XY}}[1_{[Yf(X)<0]}] \qquad (8)$$

Expected loss and empirical loss

- Objective of prediction = to minimize expected loss on future data, i.e.

$$\text{minimize } L(f) \;=\; E_{P(X,Y)}[L(Y, f(X)] \text{ over } f \in \mathcal{F} \tag{7}$$

  We call $L(f)$ above **expected loss**.

- $L(f)$ cannot be minimized or even computed directly, because we don't know the data distribution P_{XY}.
  Therefore, in training predictors, one uses the **empirical** data distribution given by the sample $\mathcal{D}$.

- The empirical loss (or **empirical error** or **training error**) is the average loss on $\mathcal{D}$

averaged over $\mathcal{D}$

$$\boxed{\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y^i, f(x^i))} \qquad \hat{L}_{\mathcal{D}}(f) \;=\; \frac{1}{n}\sum_{i=1}^{n} 1_{[y^i f(x^i) < 0]} \tag{8}$$

- Finally, the value of the **optimal expected loss** for our model class (this is the loss value we are aiming for) is denoted by $L(\mathcal{F})$.

best $L(f)$ in $\mathcal{F}$

$$L(\mathcal{F}) \;=\; \min_{f \in \mathcal{F}} \boxed{E_{P(X,Y)}[L(Y, f(X))]} \quad L(f) \tag{9}$$

Note that of all the quantities above, we can only know $\hat{L}(f)$ for a finite number of f's in $\mathcal{F}$.

$\hat{L}(f)$ = empirical Loss

1-NN

Training — $\min_{\mathcal{F}} \hat{L}(f)$.

linear regression (L_{LS})

logistic regression

$$\min_{\mathcal{F}} \Big[\underbrace{\hat{L}(f)}_{\uparrow\, \mathcal{D}} + \lambda \underbrace{R(f)}_{\uparrow\, \text{regularization indep of } \mathcal{D}} \Big]$$

SVM, Lasso regression

Ridge regression

other (LDA)

K-NN

NW

Bayes loss

problem → L

nature → P_{XY}

- How small can the expected loss $L(f)$ be?
  It is clear that

$$L(\mathcal{F}) \;=\; \min_{f\in\mathcal{F}} L(f) \;\geq\; \boxed{\min_{f} L(f) \;=\; L^*} \qquad (10)$$

  where L^* is taken over all possible functions f that take values in $\mathcal{Y}$.
- L^* is the absolute minimum loss for the given P_{XY} and it is called the **Bayes loss**.
- The Bayes loss is usually not zero

I $\quad y = f^*(x)$ deterministic $\Rightarrow L^* = 0$

II $\quad P_{Y|X}$ not deterministic $\Rightarrow L^* > 0$

given x :

$P_{Y|X}$ known

choose: $\hat{y}$ $\qquad E_{P_{Y|X}}[L(y,\hat{y})] = $

$= \ell(x,\hat{y})$

$$\boxed{f^*(x)} = y^*(x) = \operatorname*{argmin}_{\hat{y}}$$

Bayes loss for (binary) classification

- Fix x and assume $P_{Y|X}$ known. Then:
  - Label y will have probability $P_{Y|X}(y|x)$ at this x.
  - No deterministic guess $f(x)$ for y will make the classification error $E_{P_{Y|X=x}}[L_{01}(y, f(x))]$ (unless $P_{Y|X=x}$ is itself deterministic)
  - Best guess minimizes the probability of being wrong. This is achieved by chosing the most probable class

$$y^*(x) = \operatorname*{argmax}_y P_{Y|X}(y|x) \qquad \pm \tag{11}$$

most probable

  - The probability of being wrong if we choose $y^*(x)$ is $1 - p^*(x)$, where $p^*(x) = \max_y P_{Y|X}(y|x)$.
- The **Bayes classifier** is $y^*(x)$ as a function of x and its expected loss is the Bayes loss

$$\underset{\longrightarrow}{L^*_{01}} = E_{P_X}[\underbrace{1 - p^*(X)}_{\Pr[err]}] = E_{P_X}[1 - \max_y P[Y|X]] \tag{12}$$

This shows that the Bayes loss is a property of the problem, via L and P_{XY}, and not of any model class or learning algorithm.

Example

In a classification problem where the class label depends deterministically of the input, the Bayes loss is 0. For example, classifying between written English and written Japanese has (probably) zero Bayes loss.

Example

Consider the least squares loss and the following data distribution: $P_{Y|X} \sim N(g(X), \sigma^2)$. In other words, the Y values are normally distributed around a deterministic function $g(X)$. In this case, optimal least squares predictor is the mean of Y given X, which is equal to $g(X)$. The Bayes loss is the expected squared error around the mean, which is σ^2. Exercise what is the expression of the Bayes loss if $P_{Y|X} \sim N(g(X), \sigma(X)^2)$?

Exercise What is the Bayes loss if (1) $P(Y|X) \sim N((\beta^*)^T X, \sigma^2 I)$ and the loss is L_{LS}; (2) $P(X|Y = \pm 1) \sim N(\mu_\pm, \sigma^2 I)$ and the loss is L_{01} (for simplicity, assume $X \in \mathbb{R}, \mu_{pm} = \pm 1, \sigma = 1$); (3) give a formula for the Bayes loss if we know $P(X|Y = \pm 1), P(Y)$, $Y \in \{\pm 1\}$ and the loss is L_{01}. (4) Give an example of a situation when the Bayes loss is 0.

Bias & Variance for parameter estimation

Pb $X^{1:n} \sim N(\mu, \sigma^2)$ ← assume true model

Max Likelihood $\hat{\mu} = \dots$

$\hat{\sigma}^2 = \dots$

$E[\hat{\mu} - \mu] = 0$ r.v. unbiased

$\mathcal{D}_n \equiv P_X^n$ n samples

$\operatorname{Var}\hat{\mu} = \frac{\sigma^2}{n}$

$E[\hat{\sigma}^2 - \sigma^2] = -\frac{\sigma^2}{n}$ biased

$E[\hat{\sigma}^2] = \sigma^2 \frac{n-1}{n}$

$\operatorname{Var}\hat{\sigma}^2 = \dots$

Prediction

- estimating $f(x)$ a function !!

  P_{XY} distribution

- no f^{true} (sometimes)