Statistical Foundations of Learning

Debarghya Ghoshdastidar

School of Computation, Information and Technology
Technical University of Munich

Statistical Learning Problem

Outline

- How do we formally pose a ML problem?
  - Terminology
  - Assumptions + Probabilistic framework
  - Loss / risk and risk minimisation
  - Bayes risk (optimal solution for learning problem)

Formal setup of supervised learning

- Domain or feature space $\mathcal{X}$
  - Space containing features of data
  - Example: Car specification (max power, fuel type etc.) ; All 600×800 car images
- Label set $\mathcal{Y}$
  - Set containing all possible outcomes of our prediction task
  - Example: $\mathcal{Y} = \{0, 1\}$ for binary classification (city / sports car)
  - Example: $\mathcal{Y} = [0, \infty)$ for regression (CO_2 emission)
- Goal: Find a predictor $h : \mathcal{X} \to \mathcal{Y}$ ... will write as $h \in \mathcal{Y}^{\mathcal{X}}$

Learning algorithm

- Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
  - $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, and we write $S \in (\mathcal{X} \times \mathcal{Y})^m$
- Learner or learning algorithm $\mathcal{A} : \bigcup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{Y}^{\mathcal{X}}$
  - $\mathcal{A}$ takes a training sample S of any size $m \geq 1$
  - $\mathcal{A}(S) : \mathcal{X} \to \mathcal{Y}$ is a predictor

Probabilistic framework for data

- Learning is difficult if training and test data are not '*similar*'
  - Every training car is sports car, but test car is city car
  - No training car has 6 doors, but we need to predict for a car with 6 doors
- Key assumptions in (most of) statistical learning:
  - There is a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$
  - Every training / test data $(x, y) \sim \mathcal{D}$
  - Training data $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ are i.i.d. samples ... we write $S \sim \mathcal{D}^m$

Simple case: True predictor is deterministic function

- Let (x, y) such that $x \sim \mathcal{D}_{\mathcal{X}}$ and $y = f(x)$
- Example of car:
  - $x = (x^f, x^s)$ with $x^f \in \{\text{gas, diesel, electric}\}$ and $x^h \in [0, \infty)$ is horsepower
  - $\mathcal{D}_{\mathcal{X}}$ such that $x^f \sim \text{Uniform}(\{\text{gas, diesel, electric}\})$, and $x^h \in \text{Exp}(500)$ independent
  - $y = f(x) \in \{\text{city, sports}\}$ could be as follows:

	x^f		
x^h	gas	diesel	electric
< 500	city	sports	city
≥ 500	sports	sports	city

General case: True label is also random

- Assuming deterministic label is restrictive
  - Two cars with close specifications could have different purpose
  - We assume that true label of any x is random
- Write $\mathcal{D} = \mathcal{D}_{\mathcal{X}} \times \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$ ... joint distribution of (x, y)
  - $\mathcal{D}_{\mathcal{X}} =$ marginal distribution over $\mathcal{X}$
  - $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x) =$ probability that the label is y, given that x is observed

Loss function

- Loss $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$
  - Measures difference between predicted and true labels / values
- Example: 0-1 loss for binary classification
  - Let $y =$ true label, and $h(x) =$ prediction
  - $\ell(h(x), y) = \mathbf{1}\{h(x) \neq y\}$

... more losses to be discussed later

- Example: squared loss for regression
  - $\ell(h(x), y) = (h(x) - y)^2$

Risk

- Risk / generalisation error of predictor h with respect to distribution $\mathcal{D}$

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\ell(h(x), y)\big]$$

  - We sample $(x, y) \sim \mathcal{D}$, and measure the expected error of h

- **Verify:** $L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$ for 0-1 loss

- Generalisation error of learner $\mathcal{A}$ w.r.t. $\mathcal{D}$ given training sample S

  - Given S, the learned predictor is $\widehat{h}_S = \mathcal{A}(S)$

$$L_{\mathcal{D}}(\widehat{h}_S) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\ell(\widehat{h}_S(x), y)\big]$$

  - Expected test error of the predictor learned by $\mathcal{A}$

Empirical risk

- Empirical risk of h on sample S

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), y_i)$$

  - $L_S(h) =$ training error of h w.r.t. sample S
  - For $S \sim \mathcal{D}^m$, sample average is an estimate of $L_{\mathcal{D}}(h)$

- **Verify:** $\mathbb{E}_{S \sim \mathcal{D}^m}[L_S(h)] = L_{\mathcal{D}}(h)$ for any fixed h

Empirical risk minimisation

- Goal of supervised learning: Find predictor with low test error (risk)

$$\underset{h \in \mathcal{Y}^{\mathcal{X}}}{\text{minimise}} \; L_{\mathcal{D}}(h)$$

  - We cannot compute $L_{\mathcal{D}}(h)$ without knowledge of $\mathcal{D}$

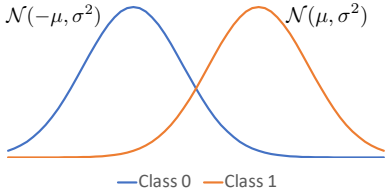
- Empirical risk minimisation (ERM)

$$\underset{h \in \mathcal{Y}^{\mathcal{X}}}{\text{minimise}} \; L_{S}(h)$$

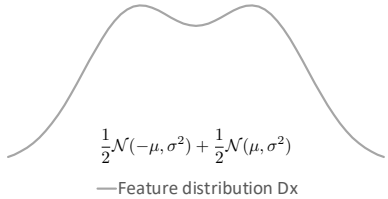
  - Replace $L_{\mathcal{D}}(h)$ by its estimate computed on training sample S

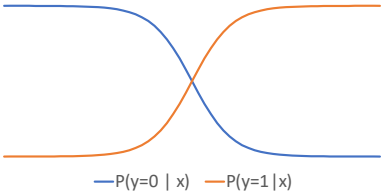
- Question: Why do we assume all of S is training data? Where is test data?

Example: Two equally likely Gaussian-distributed classes



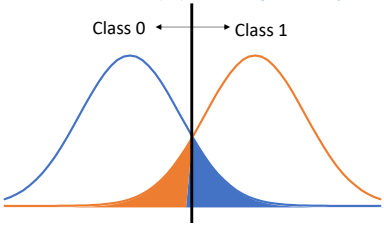
How do we formally state the problem? What are $\mathcal{D}_{\mathcal{X}}$ and $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$?



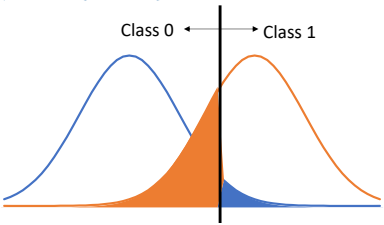


Example (continued)

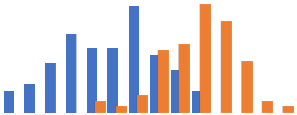
Predictor $h(x) = \mathbf{1}\{x > 0\}$



$h(x) = \mathbf{1}\{x > t\}$, makes less error on one class



ERM tries to find this optimal threshold based on samples S



Empirical risk minimisation

- Solving ERM directly can lead to poor solutions. Why?
- Example:
  - Let $x \sim \text{Uniform}([0,1])$ and $y = \mathbf{1}\{x > 0.1\}$
  - Given $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
  - Find $\widehat{h}$ such that $L_S(\widehat{h}) = 0$, but $L_{\mathcal{D}}(\widehat{h}) = 0.9$
  - Solution:

$$\widehat{h}(x) = \begin{cases} 1 & \text{if } x = x_i \in \{x_1, \ldots, x_m\} \text{ and } y_i = 1 \\ 0 & \text{if } x = x_i \in \{x_1, \ldots, x_m\} \text{ and } y_i = 0 \\ 0 & \text{if } x \notin \{x_1, \ldots, x_m\} \end{cases}$$

Risk minimisation

- Goal of supervised learning: Find predictor with low test error (risk)

$$\underset{h \in \mathcal{Y}^{\mathcal{X}}}{\text{minimise}} \; L_{\mathcal{D}}(h)$$

  - We cannot compute $L_{\mathcal{D}}(h)$ without knowledge of $\mathcal{D}$

- Assume $\mathcal{D}$ is known and $\mathcal{Y} = \{0, 1\}$
  - Recall $\mathcal{D} = \mathcal{D}_{\mathcal{X}} \times \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y|x)$
  - Define $\eta(x) = \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1|x)$ ... probability that x has label 1
  - $\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 0|x) = 1 - \eta(x)$

Computing $L_{\mathcal{D}}(h)$ for binary classification

Theorem Risk.1 (Risk of a binary classifier)

For any deterministic h, the risk with respect to 0-1 loss is

$$\begin{aligned} L_{\mathcal{D}}(h) &= \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(h(x) \neq y \mid x) \right] \\ &= \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\eta(x) \cdot \mathbf{1}\{h(x) = 0\} + (1 - \eta(x)) \cdot \mathbf{1}\{h(x) = 1\} \right] \end{aligned}$$

Computing $L_{\mathcal{D}}(h)$ for binary classification: Proof

Recall definition of 0-1 loss: $\ell(h(x), y) = \mathbf{1}\{h(x) \neq y\}$

$$\begin{aligned} L_{\mathcal{D}}(h) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbf{1}\{h(x) \neq y\}\right] \\ &= \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\left[\mathbf{1}\{h(x) \neq y\} \,|\, x\right]\right] = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(h(x) \neq y \mid x)\right] \end{aligned}$$

Can also write

$$L_{\mathcal{D}}(h) = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\mathbb{E}_{\mathcal{Y}|\mathcal{X}}\left[\mathbf{1}\{h(x) = 0\}\,\mathbf{1}\{y = 1\} + \mathbf{1}\{h(x) = 1\}\,\mathbf{1}\{y = 0\} \,\middle|\, x\right]\right]$$

Note h is deterministic. Given x, $h(x)$ is independent of y

$$L_{\mathcal{D}}(h) = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\mathbf{1}\{h(x) = 0\}\,\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 1|x) + \cdot\mathbf{1}\{h(x) = 1\}\,\mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = 0|x)\right]$$

Lower bound on $L_{\mathcal{D}}(h)$

For any deterministic h

$$\begin{aligned} L_{\mathcal{D}}(h) &= \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\eta(x)\cdot\mathbf{1}\left\{h(x)=0\right\}+(1-\eta(x))\cdot\mathbf{1}\left\{h(x)=1\right\}\right] \\ &\geq \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\min\left\{\eta(x),1-\eta(x)\right\}\right] \end{aligned}$$

Reason: For every x, we incur cost of $\eta(x)$ or $(1-\eta(x))$ (so, at least minimum of both)

Bayes risk

Bayes classifier:

$$h^*(x) = \begin{cases} 1 & \text{if } \eta(x) \geq 0.5 \\ 0 & \text{if } \eta(x) < 0.5 \end{cases} \qquad \text{that is, } 1 - \eta(x) \text{ is smaller}$$

Note: $\mathbf{1}\{h^*(x) = 0\}$ when $\eta(x)$ is smaller, and $\mathbf{1}\{h^*(x) = 1\}$ when $1 - \eta(x)$ is smaller.

$$\eta(x) \cdot \mathbf{1}\{h^*(x) = 0\} + (1 - \eta(x)) \cdot \mathbf{1}\{h^*(x) = 1\} \;=\; \min\{\eta(x), 1 - \eta(x)\}$$

Theorem Risk.2 (Bayes risk = Generalisation error of Bayes classifier)

If $\ell =$ 0-1 loss, then

$$L^*_{\mathcal{D}} \;=\; \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\big[\min\{\eta(x), 1 - \eta(x)\}\big] \;=\; L_{\mathcal{D}}(h^*) \;=\; \min_{h \in \{0,1\}^{\mathcal{X}}} L_{\mathcal{D}}(h)$$

Example of Bayes classifier

Predicting time of crash in a software installation

- $x \in [0, 1]$: fraction of completion of the installation
- $\eta(x) =$ probability that system crashes after x-fraction completion
- Ground truth: $\eta(x) = |1 - 2x|$
- Exercise: Derive h^* and $L^*_{\mathcal{D}}$, assuming $x \sim \text{Uniform}[0, 1]$
- $h^*(x) = \begin{cases} 1 \text{ for } x \in \left[0, \frac{1}{4}\right] \cup \left[\frac{3}{4}, 1\right], \\ 0 \text{ for } x \in \left(\frac{1}{4}, \frac{3}{4}\right) \end{cases}$ and $L^*_{\mathcal{D}} = L_{\mathcal{D}}(h^*) = \frac{1}{4}$
- Implement any learning algorithm and check if you can get better test error (computed over large sample)

Questions

- While computing $L_{\mathcal{D}}(h)$, we assume h is a deterministic function of x
  - Which step fails if this is not true?
- Derive the Bayes classifier and Bayes risk for k-class classification
  - You need to define $\eta_1(x), \ldots, \eta_k(x)$ where $\eta_i(x) = \mathbb{P}_{\mathcal{Y}|\mathcal{X}}(y = i|x)$
  - $\sum_{i=1}^{k} \eta_i(x) = 1$

Bayes risk for regression

- Assume $\mathcal{Y} = \mathbb{R}$ and squared loss $\ell(h(x), y) = (h(x) - y)^2$
- For any distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{R}$ and regressor $h : \mathcal{X} \to \mathbb{R}$

$$\text{Risk:} \qquad L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(h(x) - y)^2\right]$$

- ERM: $\underset{h\in\mathcal{Y}^{\mathcal{X}}}{\text{minimise}}\ L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2$

  ... least square regression (Ordinary least square if h is linear)

- Can we characterise the Bayes classifier: $h^*(x) = \underset{h\in\mathcal{Y}^{\mathcal{X}}}{\arg\min}\ L_{\mathcal{D}}(h)$?

Bayes risk for regression

Verify for any deterministic regressor $h : \mathcal{X} \to \mathbb{R}$

$$L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(h(x) - y)^2\right] = \mathbb{E}_x\left[(h(x))^2 + \mathbb{E}[y^2|x] - 2 \cdot h(x) \cdot \mathbb{E}[y|x]\right]$$

- The above cannot be simplified without further assumptions
- Assume $y = f(x) + \epsilon$, where f is "true" function and ϵ is "noise"
  Also assume ϵ is independent of x, and $\mathbb{E}[\epsilon] = 0, \mathbb{E}[\epsilon^2] = \sigma^2$
- Verify that $L_{\mathcal{D}}(h) = \mathbb{E}_x\left[(h(x) - f(x))^2\right] + \sigma^2$
- Hence, Bayes regressor $h^*(x) = f(x)$ and Bayes risk $L^*_{\mathcal{D}} = \sigma^2$

Up next

- Can a learner achieve Bayes risk if it has unlimited training data?
- We will analyse k-nearest neighbour classifier
  - If k is large, then k-NN achieves Bayes risk asymptotically (as $m \to \infty$)
- Later, we will see this is difficult to achieve in general for $m < \infty$