Statistical models for prediction

COMS 4771 Fall 2023

Goals of prediction

General statistical model for prediction:

- Regard outcome that we want to predict as a random variable Y, and corresponding feature vector we observe as a random vector X
- Joint distribution P of (X, Y) is the "full population" of interest

Problem: Create a program $f : \mathcal{X} \to \mathcal{Y}$ that, given X, returns a prediction of Y

Usually these programs are called predictors or prediction functions

How to measure how good/bad a prediction is?

Loss function $\mathrm{loss}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ measures how bad $\hat{y}$ is as a prediction of the outcome y

$$\mathrm{loss}(\hat{y}, y)$$

(Loss is usually non-negative, and smaller loss is better)

Example: zero-one loss (usually for classification problems)

$$\mathrm{loss}_{0/1}(\hat{y}, y) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise} \end{cases}$$

Example: squared error, a.k.a. square loss (for $\mathcal{Y} \subseteq \mathbb{R}$)

$$\mathrm{loss}_{\mathrm{sq}}(\hat{y}, y) = (\hat{y} - y)^2$$

X and Y are random variables, so $\text{loss}(f(X), Y)$ **is also a random variable**!

Standard "average-case" benchmark: expected value of the loss, a.k.a. risk:

$$\text{Risk}[f] = \mathbb{E}[\text{loss}(f(X), Y)]$$

Expectation integrates $\text{loss}(f(x), y)$ with respect to joint distribution of (X, Y)

Standard loss functions are usually simplifications of application-specific loss

Example: spam filtering, $\mathcal{Y} = \{\text{ham}, \text{spam}\}$

- Mildly annoying if spam email is erroneous put in the inbox
- But very bad if real (important) email is put in spam folder
- Zero-one loss treats both types of mistakes equally
- Perhaps better to use $\text{loss}(\hat{y}, y)$ given by

	$y = \text{ham}$	$y = \text{spam}$
$\hat{y} = \text{ham}$	0	10
$\hat{y} = \text{spam}$	1	0

This is an example of a cost-sensitive loss function

Optimal predictions of binary outcomes

Suppose you want to **predict binary outcome** Y where $\mathrm{range}(Y) = \{0, 1\}$ to minimize the risk under zero-one loss (i.e., error rate)

$X =$ side-information, potentially informative about distribution of Y

Example:

- Y is outcome of coin toss
- X is initial position of the coin, angle at which thumb hits the coin, current wind conditions, ...

If you **ignore** X, then the best (constant) prediction of Y is

$$y^\star = \begin{cases} \underline{\hspace{2em}} & \text{if } \Pr(Y = 1) < 1/2 \\ \underline{\hspace{2em}} & \text{if } \Pr(Y = 1) > 1/2 \\ \underline{\hspace{6em}} & \text{if } \Pr(Y = 1) = 1/2 \end{cases}$$

Note that $y^\star$ depends on the marginal distribution of Y:

$$\Pr(Y = 1) = \sum_x \Pr(Y = 1 \wedge X = x)$$

If you **observe** X, it may be possible to do better

- Best prediction given $X = x$ is

$$f^\star(x) = \begin{cases} \underline{\hspace{2em}} & \text{if } \underline{\hspace{14em}} \\ \underline{\hspace{2em}} & \text{if } \underline{\hspace{14em}} \\ \underline{\hspace{6em}} & \text{if } \underline{\hspace{14em}} \end{cases}$$

- $f^\star(x)$ depends on the conditional distribution of Y given $X = x$

Role of training data

Difficulty: **optimal predictions/predictors depend on distribution of** (X, Y)

- E.g., if distribution (X, Y) corresponds to entire human population, the need to poll entire human population to calculate optimal prediction / predictors

Training data can help, under certain assumptions

Assumption: **training data is "representative" sample of population**

Usual interpretation: training data $(X^{(1)}, Y^{(1)}), \ldots, (X^{(n)}, Y^{(n)})$ form independent and identically distributed (i.i.d.) sample from distribution of (X, Y)

Notation:

$$((X^{(i)}, Y^{(i)}))_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} (X, Y)$$

or

$$((X^{(i)}, Y^{(i)}))_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} P$$

(if P is the distribution of (X, Y))

Example: **guess optimal prediction** $y^\star$ using training data

- Let $\hat{Y}$ be the majority value among $Y^{(1)}, \ldots, Y^{(n)}$, i.e.,

$$\hat{Y} = \begin{cases} 0 & \text{if more 0s than 1s in } Y^{(1)}, \ldots, Y^{(n)} \\ 1 & \text{if more 1s than 0s in } Y^{(1)}, \ldots, Y^{(n)} \\ \text{either 0 or 1} & \text{if equal number of 0s and 1s} \end{cases}$$

- What's the probability that $\hat{Y} = y^\star$?

Example: **guess optimal predictor** $f^\star$ using training data (for finite $\mathrm{range}(X)$)

- Let $\hat{f}(x)$ be the majority value among all $Y^{(i)}$ such that $X^{(i)} = x$
  - If no such examples exist, then set $\hat{f}(x)$ arbitrarily

- Same as previous example, except with $D = |\mathrm{range}(X)|$ "coins", and as few as n/D training data pertinent to some coins

Some ways training data can help when $\mathrm{range}(X)$ **is large/infinite**

- Assume/leverage "local regularity"
  - Prediction at x benefits from training data $(X^{(i)}, Y^{(i)})$ for with $X^{(i)}$ nearby x
- Assume/leverage "global structure"
  - Prediction at x benefits from all training data $(X^{(i)}, Y^{(i)})$

Why i.i.d. assumption? Consider some gross violations:

- Distribution of training data has nothing to do with distribution of (X, Y)

- Suppose $(X^{(1)}, Y^{(1)}) \sim (X, Y)$, and then we define $(X^{(i)}, Y^{(i)}) = (X^{(1)}, Y^{(1)})$ for all $i = 2, \ldots, n$

Role of test data

Assumption: test data $(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), \ldots, (\tilde{X}^{(m)}, \tilde{Y}^{(m)}) \overset{\text{i.i.d.}}{\sim} (X, Y)$, all independent of training data

Suppose we have created a classifier $\hat{f} \colon \mathcal{X} \to \mathcal{Y}$ using training data, and we would like to know how good it is

- (True) error rate is $\mathrm{err}[\hat{f}] = \mathbb{E}[\mathrm{loss}_{0/1}(\hat{f}(X), Y)]$
- To calculate $\mathrm{err}[\hat{f}]$, we need to know the distribution of (X, Y)
- Using test data, we estimate $\mathrm{err}[\hat{f}]$ by

$$\widetilde{\mathrm{err}}[\hat{f}] = \frac{1}{m} \sum_{i=1}^{m} \mathrm{loss}_{0/1}(\hat{f}(\tilde{X}^{(i)}), \tilde{Y}^{(i)})$$

This is the test error rate

Test error rate: $\widetilde{\mathrm{err}}[\hat{f}] = \dfrac{S}{m}$ where

$$S = \sum_{i=1}^{m} \mathbb{1}\{\hat{f}(\tilde{X}^{(i)}) \neq \tilde{Y}^{(i)}\}$$

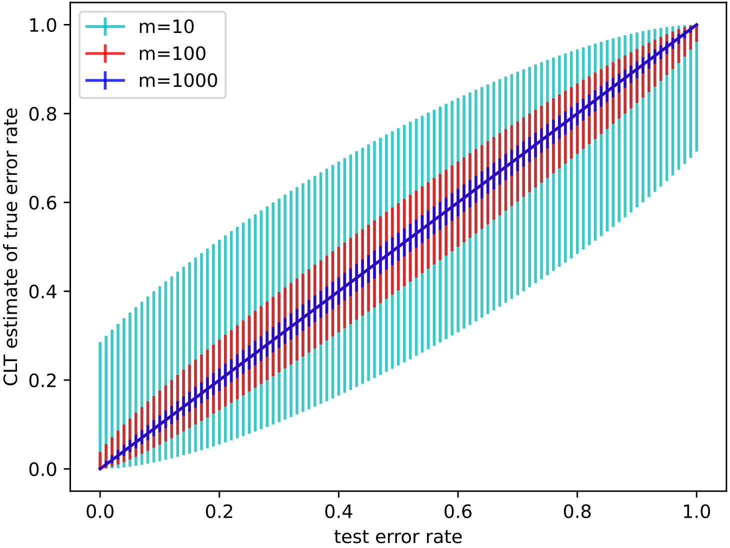
is sum of m i.i.d. $\mathrm{Bernoulli}(\theta)$ random variables where $\theta = \mathrm{err}[\hat{f}]$

Distribution of S is Binomial with m trials and success probability θ

- Notation: $S \sim \mathrm{Binomial}(m, \theta)$

Facts about $S \sim \mathrm{Binomial}(m, \theta)$

- $\mathbb{E}(S) = m\theta$
- $\mathrm{var}(S) = m\theta(1-\theta)$
- $\dfrac{S - m\theta}{\sqrt{m\theta(1-\theta)}} \longrightarrow \mathrm{N}(0,1)$ as $m \to \infty$ (by Central Limit Theorem)


m=10
m=100
m=1000
1.0
0.8
0.6
0.4
0.2
0.0
CLT estimate of true error rate
0.0
0.2
0.4
0.6
0.8
1.0
test error rate

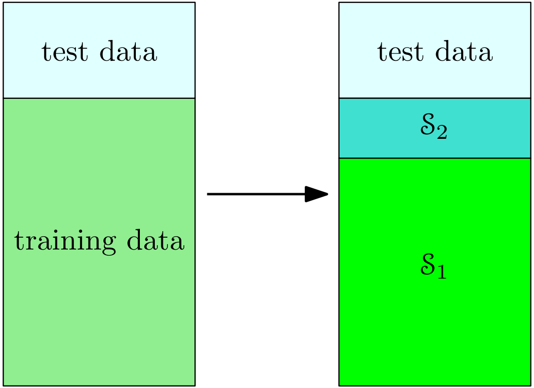
Why should test data be independent of training data?
Why doesn't previous argument apply with i.i.d. training data?

Cross validation

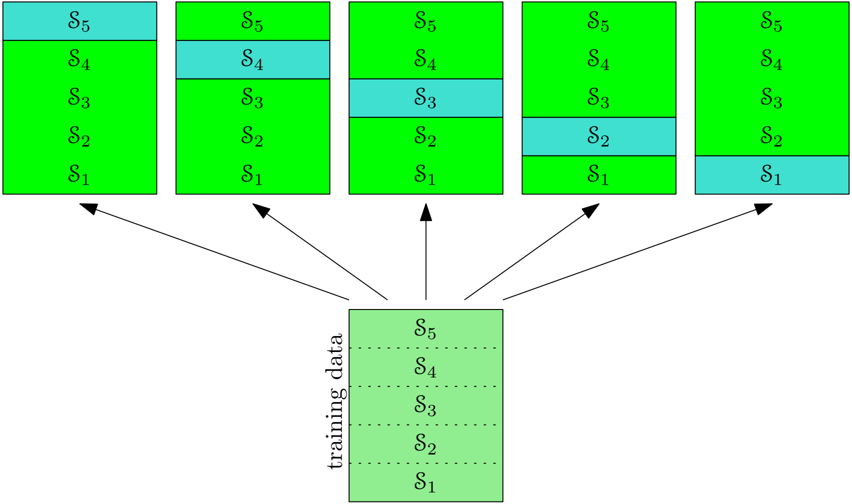
Common practice: split dataset into three parts

1. Training data: provided as input to learning algorithms
2. Validation data (a.k.a. development data, held-out data): used to evaluate experimentation with models, tweaks to learning algorithm, etc.
3. Test data: only used after you have settled on the learning algorithm/hyperparameters/etc., to evaluate the final predictor

(Hold-out) cross validation: simulate splitting dataset into training + test data
. . . all done only using training data



K-fold cross validation



Leave one out cross validation (LOOCV): K-fold cross validation with $K = n$

Optimal predictions of real-valued outcomes

Suppose you are to predict the real-valued outcome Y where $\mathrm{range}(Y) \subseteq \mathbb{R}$ so as to minimize risk under square loss (i.e., minimize MSE)

- If you ignore X, then best (constant) prediction of Y is $y^* = \mathbb{E}(Y)$

- If you observe X, then best prediction given $X = x$ is

$$\eta(x) = \mathbb{E}(Y \mid X = x)$$

  Here, $\eta \colon \mathcal{X} \to \mathbb{R}$ is the conditional mean function

Dartmouth students' (first-year) college GPA vs high school GPA