Statistical Learning Theory

Based on David Rosenberg and He He's materials

Tal Linzen

CDS, NYU

Jan 25, 2022

Decision theory

- In data science problems, we generally need to:
    - Make a decision:
        - Move email to spam folder?

Decision theory

- In data science problems, we generally need to:
    - Make a decision:
        - Move email to spam folder?
    - Take an action:
        - In a self-driving car, make a right turn
        - Reject the hypothesis that $\theta = 0$ (classical statistics)

Decision theory

- In data science problems, we generally need to:
  - Make a decision:
    - Move email to spam folder?
  - Take an action:
    - In a self-driving car, make a right turn
    - Reject the hypothesis that $\theta = 0$ (classical statistics)
  - Produce some output:
    - Whose face is it in the image?
    - The Hindi translation of a Japanese input sentence

Decision theory

- In data science problems, we generally need to:
    - Make a decision:
        - Move email to spam folder?
    - Take an action:
        - In a self-driving car, make a right turn
        - Reject the hypothesis that $\theta = 0$ (classical statistics)
    - Produce some output:
        - Whose face is it in the image?
        - The Hindi translation of a Japanese input sentence
    - Predicting where a storm will be in an hour (what forms of output are possible here?)

Decision theory

- In data science problems, we generally need to:
  - Make a decision:
    - Move email to spam folder?
  - Take an action:
    - In a self-driving car, make a right turn
    - Reject the hypothesis that $\theta = 0$ (classical statistics)
  - Produce some output:
    - Whose face is it in the image?
    - The Hindi translation of a Japanese input sentence
  - Predicting where a storm will be in an hour (what forms of output are possible here?)
  - An **action** is the generic term for what is produced by our system.

Inputs

We make our decision based on context:

- Inputs [ML]
- Covariates [Statistics]

Examples of inputs

- A picture
- The location of the storm in the last 24 hours, other weather-related measurements
- A search query

Outcome

Inputs are often paired with **outputs** or **labels**.

Examples of outcomes/outputs/labels

- Whether or not the picture actually contains an animal
- The storm's location one hour after they query
- Which, if any, of the suggested URLs were selected

Evaluation Criterion

Decision theory is about finding "optimal" actions, under various definitions of optimality.

Examples of Evaluation Criteria

- Is the classification correct?
- Does the transcription exactly match the spoken words?
  - Should we give partial credit (for getting only some of the words right)? How?
- How far is the storm from the predicted location? (If we're producing a point estimate)
- How likely is the storm's actual location under the predicted distribution? (If we're doing density prediction)

Typical Sequence of Events

Many problem domains can be formalized as follows:

1. Observe input x.
2. Take action a.
3. Observe outcome y.
4. Evaluate action in relation to the outcome.

Three spaces:

- Input space: $\mathcal{X}$
- Action space: $\mathcal{A}$
- Outcome space: $\mathcal{Y}$

Formalization

Prediction Function

A **prediction function** (or **decision function**) gets input $x \in \mathcal{X}$ and produces an action $a \in \mathcal{A}$:

$$\begin{aligned} f: \quad \mathcal{X} &\to \mathcal{A} \\ x &\mapsto f(x) \end{aligned}$$

Formalization

Prediction Function

A **prediction function** (or **decision function**) gets input $x \in \mathcal{X}$ and produces an action $a \in \mathcal{A}$:

$$\begin{aligned} f : \quad \mathcal{X} &\to \mathcal{A} \\ x &\mapsto f(x) \end{aligned}$$

Loss Function

A **loss function** evaluates an action in the context of the outcome y.

$$\begin{aligned} \ell : \quad \mathcal{A} \times \mathcal{Y} &\to \mathbf{R} \\ (a, y) &\mapsto \ell(a, y) \end{aligned}$$

Evaluating a Prediction Function

Goal: Find the optimal prediction function.

Intuition: If we can evaluate how good a prediction function is, we can turn this into an optimization problem.

- The loss function ℓ evaluates a *single* action
- How do we evaluate the prediction function *as a whole*?
- We will use the standard **statistical learning theory** framework.

Statistical Learning Theory

Define a space where the prediction function is applicable

- Assume there is a **data generating distribution** $P_{\mathcal{X} \times \mathcal{Y}}$.
- All input/output pairs (x, y) are generated i.i.d. from $P_{\mathcal{X} \times \mathcal{Y}}$.

One common desideratum is to have a prediction function $f(x)$ that “does well on average”:

$$\ell(f(x), y) \text{ is usually small, in some sense}$$

How can we formalize this?

Risk

Definition

The **risk** of a prediction function $f : \mathcal{X} \to \mathcal{A}$ is

$$R(f) = \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}} \left[\ell(f(x), y) \right].$$

In words, it's the **expected loss** of f over $P_{\mathcal{X} \times \mathcal{Y}}$.

We can't actually compute the risk function:

Since we don't know $P_{\mathcal{X} \times \mathcal{Y}}$, we cannot compute the expectation.
But we can **estimate** it.

The Bayes Prediction Function

Definition

A **Bayes prediction function** $f^*: \mathcal{X} \to \mathcal{A}$ is a function that achieves the *minimal risk* among all possible functions:

$$f^* \in \operatorname*{arg\,min}_f R(f),$$

where the minimum is taken over all functions from $\mathcal{X}$ to $\mathcal{A}$.

- The risk of a Bayes prediction function is called the **Bayes risk**.
- A Bayes prediction function is often called the "**target function**", since it's the best prediction function we can possibly produce.

Example: Multiclass Classification

- Spaces: $\mathcal{A} = \mathcal{Y} = \{1, \ldots, k\}$
- 0-1 loss:

$$\ell(a, y) = 1(a \neq y) := \begin{cases} 1 & \text{if } a \neq y \\ 0 & \text{otherwise.} \end{cases}$$

Example: Multiclass Classification

- Spaces: $\mathcal{A} = \mathcal{Y} = \{1, \dots, k\}$
- 0-1 loss:

$$\ell(a, y) = 1(a \neq y) := \begin{cases} 1 & \text{if } a \neq y \\ 0 & \text{otherwise.} \end{cases}$$

- Risk:

$$\begin{aligned} R(f) &= \mathbb{E}\left[1(f(x) \neq y)\right] \quad = 0 \cdot \mathbb{P}(f(x) = y) + 1 \cdot \mathbb{P}(f(x) \neq y) \\ &= \mathbb{P}(f(x) \neq y), \end{aligned}$$

  which is just the misclassification error rate.

- The Bayes prediction function returns the most likely class:

$$f^*(x) \in \underset{1 \leqslant c \leqslant k}{\arg\max}\, \mathbb{P}(y = c \mid x)$$

But we can't compute the risk!

- Can't compute $R(f) = \mathbb{E}\left[\ell(f(x), y)\right]$ because we **don't know** $P_{\mathcal{X}\times\mathcal{Y}}$.

But we can't compute the risk!

- Can't compute $R(f) = \mathbb{E}\left[\ell(f(x), y)\right]$ because we **don't know** $P_{\mathcal{X}\times\mathcal{Y}}$.
- One thing we can do in ML/statistics/data science is **estimate** it:

But we can't compute the risk!

- Can't compute $R(f) = \mathbb{E}[\ell(f(x), y)]$ because we **don't know** $P_{\mathcal{X}\times\mathcal{Y}}$.
- One thing we can do in ML/statistics/data science is **estimate** it:

Assume we have sample data:

Let $\mathcal{D}_n = ((x_1, y_1), \ldots, (x_n, y_n))$ be drawn i.i.d. from $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$.

But we can't compute the risk!

- Can't compute $R(f) = \mathbb{E}\left[\ell(f(x), y)\right]$ because we **don't know** $P_{\mathcal{X}\times\mathcal{Y}}$.
- One thing we can do in ML/statistics/data science is **estimate** it:

Assume we have sample data:

Let $\mathcal{D}_n = ((x_1, y_1), \dots, (x_n, y_n))$ be drawn i.i.d. from $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$.

- We draw inspiration from the strong law of large numbers:
  If $z_1, \dots, z_n$ are i.i.d. with expected value $\mathbb{E}z$, then

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} z_i = \mathbb{E}z,$$

with probability 1.

The Empirical Risk

Let $\mathcal{D}_n = ((x_1, y_1), \ldots, (x_n, y_n))$ be drawn i.i.d. from $\mathcal{P}_{\mathcal{X} \times \mathcal{Y}}$.

Definition

The **empirical risk** of $f : \mathcal{X} \to \mathcal{A}$ with respect to $\mathcal{D}_n$ is

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

By the strong law of large numbers,

$$\lim_{n \to \infty} \hat{R}_n(f) = R(f),$$

almost surely.

Empirical Risk Minimization

Definition

A function $\hat{f}$ is an **empirical risk minimizer** if

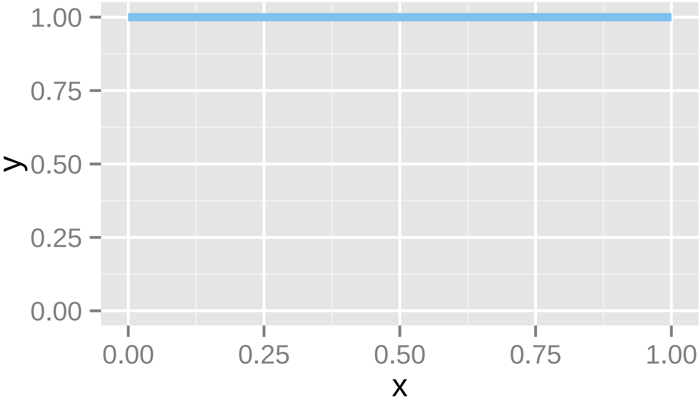
$$\hat{f} \in \underset{f}{\arg\min}\, \hat{R}_n(f),$$

where the minimum is taken over all functions $f : \mathcal{X} \to \mathcal{A}$.

- In an ideal world we'd want to find the risk minimizer.
- Is the empirical risk minimizer close enough?
- In practice, we always only have a finite sample...

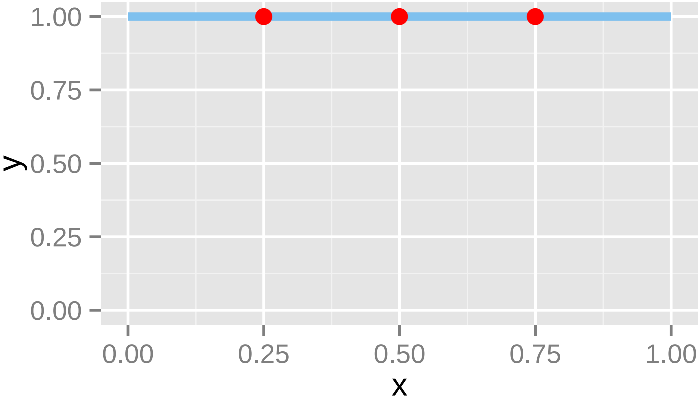
Empirical Risk Minimization

- $P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).
- A plot of $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$:



Empirical Risk Minimization

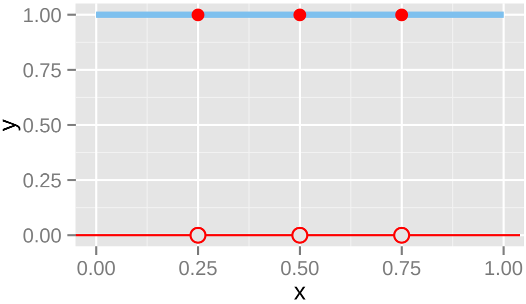
$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).



A sample of size 3 from $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$.

Empirical Risk Minimization

$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).

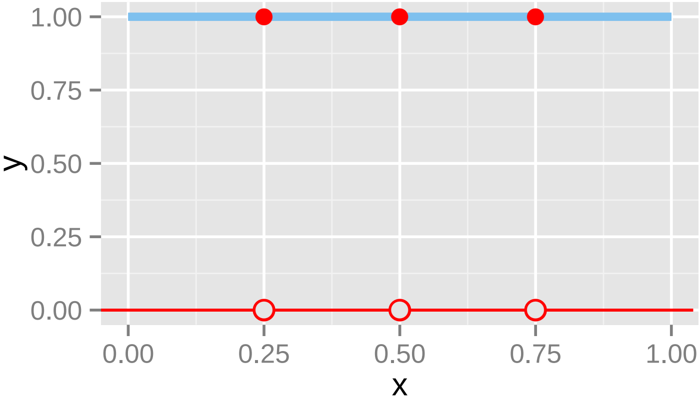


A proposed prediction function:

$$\hat{f}(x) = 1(x \in \{0.25, 0.5, 0.75\}) = \begin{cases} 1 & \text{if } x \in \{0.25, .5, .75\} \\ 0 & \text{otherwise} \end{cases}$$

Empirical Risk Minimization

$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).



Under either the square loss or the 0/1 loss, $\hat{f}$ has Empirical Risk = 0 and Risk = 1.

Empirical Risk Minimization

- In this case, ERM led to a function f that just memorized the data.
- How can we improve **generalization** from the training inputs to new inputs?
- We need to smooth things out somehow!
  - A lot of modeling is about spreading and extrapolating information from one part of the input space $\mathcal{X}$ into unobserved parts of the space.
- One approach is **constrained ERM**:
  - Instead of minimizing empirical risk over *all* prediction functions,
  - We constrain our search to a particular subset of the space of functions, called a **hypothesis space**.

Hypothesis Spaces

Definition

A **hypothesis space** $\mathcal{F}$ is a set of prediction functions $\mathcal{X} \to \mathcal{A}$ that we consider when applying ERM.

Desirable properties of a hypothesis space:

- Includes only those functions that have the desired "regularity", e.g. smoothness, simplicity
- Easy to work with (e.g., we have efficient algorithms to find the best function within the space)

Most applied work is about designing good hypothesis spaces for specific tasks.

Constrained Empirical Risk Minimization

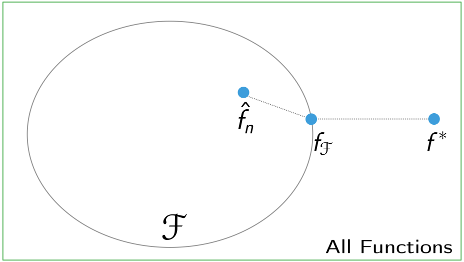
- Given a hypothesis space $\mathcal{F}$, a set of prediction functions mapping $\mathcal{X} \to \mathcal{A}$,
- An **empirical risk minimizer** (ERM) in $\mathcal{F}$ is a function $\hat{f}_n$ such that

$$\hat{f}_n \in \operatorname*{arg\,min}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

- A **Risk minimizer** in $\mathcal{F}$ is a function $f^*_{\mathcal{F}} \in \mathcal{F}$ such that

$$f^*_{\mathcal{F}} \in \operatorname*{arg\,min}_{f \in \mathcal{F}} \mathbb{E}\left[\ell(f(x), y)\right].$$

Excess Risk Decomposition



$$f^* = \arg\min_{f} \mathbb{E}\left[\ell(f(x), y)\right]$$

$$f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(f(x), y)\right]$$

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

- **Approximation error** (of $\mathcal{F}$) $= R(f_{\mathcal{F}}) - R(f^*)$
- **Estimation error** (of $\hat{f}_n$ in $\mathcal{F}$) $= R(\hat{f}_n) - R(f_{\mathcal{F}})$

Excess Risk Decomposition for ERM

Definition

The **excess risk** compares the risk of f to the Bayes optimal f^*:

$$\textbf{Excess Risk}(f) = R(f) - R(f^*)$$

- Can excess risk ever be negative?

The excess risk of the ERM $\hat{f}_n$ can be decomposed:

$$\begin{aligned}\textbf{Excess Risk}(\hat{f}_n) &= R(\hat{f}_n) - R(f^*) \\ &= \underbrace{R(\hat{f}_n) - R(f_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{R(f_{\mathcal{F}}) - R(f^*)}_{\text{approximation error}} .\end{aligned}$$

- There is a tradeoff between estimation error and approximation error

Approximation Error

Approximation error $R(f_{\mathcal{F}}) - R(f^*)$ is

- a property of the class $\mathcal{F}$
- the penalty for restricting to $\mathcal{F}$ (rather than considering all possible functions)

Bigger $\mathcal{F}$ mean *smaller* approximation error.

Concept check: Is approximation error a random or non-random variable?

Estimation Error

Estimation error $R(\hat{f}_n) - R(f_{\mathcal{F}})$

- is the performance hit for choosing f using finite training data
- is the performance hit for minimizing empirical risk rather than true risk

With *smaller* $\mathcal{F}$ we expect *smaller* estimation error.

Under typical conditions: "With infinite training data, estimation error goes to zero."

Concept check: Is estimation error a random or non-random variable?

ERM in Practice

- What have we been glossing over by writing “argmin”?

ERM in Practice

- What have we been glossing over by writing "argmin"?
- In practice, we need a method to find $\hat{f}_n \in \mathcal{F}$: this can be very difficult!

ERM in Practice

- What have we been glossing over by writing “argmin”?
- In practice, we need a method to find $\hat{f}_n \in \mathcal{F}$: this can be very difficult!
- For nice choices of loss functions and classes $\mathcal{F}$, we can get arbitrarily close to the exact minimizer
    - But that takes time – is it always worth it?

ERM in Practice

- What have we been glossing over by writing “argmin”?
- In practice, we need a method to find $\hat{f}_n \in \mathcal{F}$: this can be very difficult!
- For nice choices of loss functions and classes $\mathcal{F}$, we can get arbitrarily close to the exact minimizer
  - But that takes time – is it always worth it?
- For some hypothesis spaces (e.g. neural networks), we don’t know how to find $\hat{f}_n \in \mathcal{F}$.

Optimization Error

- In practice, we don't find the ERM $\hat{f}_n \in \mathcal{F}$.
- We find $\tilde{f}_n \in \mathcal{F}$ that we hope is good enough.
- **Optimization error:** If $\tilde{f}_n$ is the function our optimization method returns, and $\hat{f}_n$ is the empirical risk minimizer, then

$$\text{Optimization Error } = R(\tilde{f}_n) - R(\hat{f}_n).$$

Error Decomposition in Practice

- Excess risk decomposition for function $\tilde{f}_n$ returned by an optimization algorithm in practice:

$$\begin{aligned} \textbf{Excess Risk}(\tilde{f}_n) &= R(\tilde{f}_n) - R(f^*) \\ &= \underbrace{R(\tilde{f}_n) - R(\hat{f}_n)}_{\text{optimization error}} + \underbrace{R(\hat{f}_n) - R(f_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{R(f_{\mathcal{F}}) - R(f^*)}_{\text{approximation error}} \end{aligned}$$

- It would be nice to observe the error decomposition for a practical $\tilde{f}_n$!
- How would we address each type of error?
- Why is this usually impossible?
- But we could constuct an artificial example, where we know $P_{\mathcal{X}\times\mathcal{Y}}$ and f^* and $f_{\mathcal{F}}$...

ERM Overview

- Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbf{R}$,

ERM Overview

- Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbf{R}$,
- Choose a hypothesis space $\mathcal{F}$.

ERM Overview

- Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbf{R}$,
- Choose a hypothesis space $\mathcal{F}$.
- Use an optimization method to find an empirical risk minimizer $\hat{f}_n \in \mathcal{F}$:

$$\hat{f}_n = \underset{f \in \mathcal{F}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

ERM Overview

- Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbf{R}$,
- Choose a hypothesis space $\mathcal{F}$.
- Use an optimization method to find an empirical risk minimizer $\hat{f}_n \in \mathcal{F}$:

$$\hat{f}_n = \underset{f \in \mathcal{F}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

- (Or find a $\tilde{f}_n$ that comes close to $\hat{f}_n$)

ERM Overview

- Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbf{R}$,
- Choose a hypothesis space $\mathcal{F}$.
- Use an optimization method to find an empirical risk minimizer $\hat{f}_n \in \mathcal{F}$:

$$\hat{f}_n = \underset{f \in \mathcal{F}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

- (Or find a $\tilde{f}_n$ that comes close to $\hat{f}_n$)
- The data scientist's job:
  - Choose $\mathcal{F}$ that balances approximation and estimation error.
  - As we get more training data, we can use a bigger $\mathcal{F}$.