

Intro to Machine Learning

Supervised Learning

Learn to make predictions from data.

Setup:

$$\text{Data } D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\} \subseteq \mathbb{R}^d \times C$$

\mathbb{R}^d : d -dimensional feature space

\vec{x}_i : input vector of i -th input sample

y_i : label of i -th sample

C : label space

Multiple scenarios for C :

- binary classification $C = \{0, 1\}$ or $C = \{-1, +1\}$
 $|C| = 2$

example: spam filtering. An email is either spam (+1) or not spam (-1)

- multi-class classification: $C = \{1, \dots, K\}$
 $|C| > 2$

example: face classification. A person can be exactly one of K identities
(e.g. 1 = "Barack Obama", 2 = "George W. Bush")

- Regression $C = \mathbb{R}$
 $C = \mathbb{R}$

example: predict temperature or height of a person

The goal of supervised learning is to find a function $h: \mathbb{R}^d \rightarrow C$ such that $h(x_i) = y_i$ for $(\vec{x}_i, y_i) \in D$ (training)
and $h(x_+) = y_+$ for $(\vec{x}_+, y_+) \notin D$ (testing)

2 We call \vec{x}_i a feature vector and the d dimensions the features describing the i -th sample.

Examples of feature vectors:

- Patient data in a hospital ($d \leq 100$) dense)

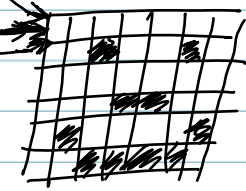
$$\vec{x}_i = \begin{pmatrix} 1 \\ 183 \\ 67 \\ \vdots \end{pmatrix} \begin{array}{l} \leftarrow \text{male/female} \\ \leftarrow \text{height in cm} \\ \leftarrow \text{age in years} \\ \end{array}$$

- Text document in bag-of-words format: ($d \approx 100,000 - 10M$ sparse)

$$\vec{x}_i = \begin{pmatrix} 10 \\ 0 \\ \vdots \\ 2 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \leftarrow \text{occurrences of word "a"} \\ \leftarrow \text{ "ant"} \\ \leftarrow \text{ "like"} \\ \leftarrow \text{ "zynga"} \end{array}$$

(we call a feature vector sparse if it consists of mostly zeros.)

- Image: ($d \approx 800,000 - 10M$) dense)

$$\vec{x}_i = \begin{pmatrix} 0.1 \\ 0.7 \\ 0.9 \end{pmatrix} \begin{array}{l} \leftarrow \text{red value} \\ \leftarrow \text{green value} \\ \leftarrow \text{blue value} \end{array}$$


A 7MP camera pic results in $7 \times 3 = 21$ million features.

3 Loss functions (aka risk function)

We want to find the best $h()$ for a data set D .

How do we quantify "best"?

There are many choices. Famous examples:

- 0/1 loss: $\frac{1}{n} \sum_{i=1}^n \mathbb{1}_{h(\vec{x}_i) \neq y_i}$ \leftarrow returns error rate on $\left\{ \begin{array}{l} 1 \text{ if } h(\vec{x}_i) \neq y_i \\ 0 \text{ o/w} \end{array} \right.$

- squared loss $\frac{1}{n} \sum_{i=1}^n (h(\vec{x}_i) - y_i)^2$
- absolute loss $\frac{1}{n} \sum_{i=1}^n |h(\vec{x}_i) - y_i|$ } Typically used for regression

Generalization

If you find a function $h()$ with low loss on your data D , how do you know it will still get examples right that are not in D ?

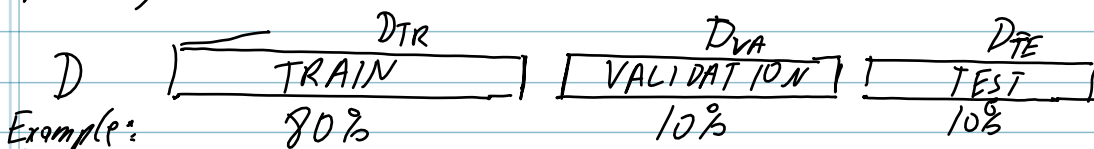
Bad example "memorizer" $h()$:

$$h(x) = \begin{cases} y_i & \text{if } \exists (\vec{x}_i, y_i) \in D \text{ s.t. } \vec{x}_i = \vec{x} \\ 0 & \text{o/w} \end{cases}$$

Gets perfect 0% error on training data D , but does horribly with samples not in D .

Train / test splits:

Split your data into train, validation and test splits.



QUIZ:
Why do we need D_{VA} ?

- Choose $h()$ on TRAIN (D_{TR}) and evaluate on D_{TE} .

4 How should you split the data?

- By time if there is a temporal component
- uniformly at random if data is i.i.d.

The test error (or test loss) approximates the true generalization error/loss.

Let's put it all together:

| <u>Learning</u> | <u>Evaluation</u> | <u>Generalization</u> |
|---|--|---|
| $h = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{ D_{TR} } \sum_{(\tilde{x}, y) \in D_{TR}} l(\tilde{x}, y)$ | $E_{TE} = \frac{1}{ D_{TE} } \sum_{(\tilde{x}, y) \in D_{TE}} l(\tilde{x}, y)$ | $\epsilon = E[l(\tilde{x}, y)]_{(\tilde{x}, y) \sim P}$ |
| hypothesis class (set of all possible classifiers) | training loss | testing loss |
| | | generalization loss (unattainable) |

Quiz: Why does $E_{TE} \xrightarrow{|D_{TE}| \rightarrow \infty} \epsilon$?

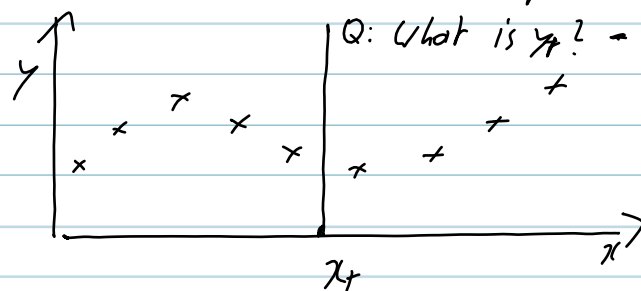
No Free lunch (Every ML algorithm makes assumptions. Always know which?)

Which hypothesis class \mathcal{H} should you choose?

It depends on the data. It encodes your assumptions about the data set / distribution P .

There is no ^{one} perfect \mathcal{H} for all problems (no free lunch theorem)

Example:



Q: What is x_t ? - A: Impossible to know without assumptions.