Machine Learning
Lecture 1

Instructors

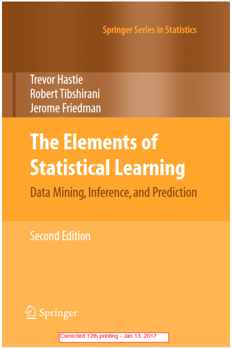
Lecturer Anton Selitskiy

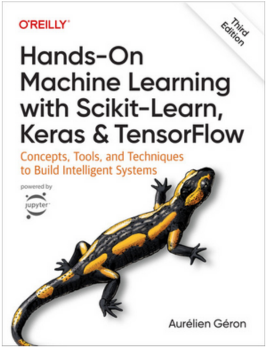
- **Office:** GOL-3701
- **Webpage:** `https://www.cs.rit.edu/~ams/`
- **Email:** amsvcs@rit.edu
- **Office Hours:**
  - Tuesday/Thursday 11am — 12:30pm and 2:30 — 3:30pm
  - Any time I'm in the office
  - By appointment

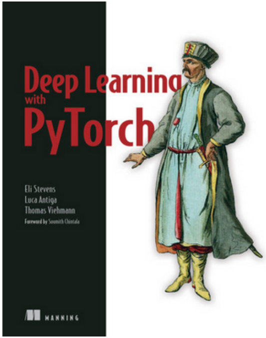
Class Assistants

Sec. 2: Holden Lalumiere
Sec. 3: Ryan Haver, David Millard

Literature







(HTF) T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning. Second Edition. 2017

(AG) Au. Geron. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Third Edition. 2022

(AVS) L. Antiga, Th. Viehmann, E. Stevens. Deep Learning with PyTorch. 2020

Some Problems

1. Transform hours into minutes:

$$f(x) = 60x$$

2. Sentiment Analysis:

"Well, that movie was just fantastic. I couldn't stop laughing the entire time. The plot was so riveting, and the acting was top-notch. Definitely worth my time. Not."

- Positive
- Neutral
- Negative

Typical Problems

If x is a text, we want to predict $f(x) = 1$, 0, or -1. But what does $f(x)$ mean?

- Not clear dependencies
- No exact formula
- We have some examples
- We are OK with an approximate solution
- We can use examples for predictions!

Notation

x — example/sample/features of the data. If it has a numerical representation $x = (x_1,\ x_2,\ \ldots,\ x_d)$, it is considered as a random vector X on $\mathbb{R}^d$.
X — all possible examples (sample space)
y — response/target
Y — target space

Training data: If we have several examples of the data, then we'll use a superscript: $\left(x^{(i)},\ y^{(i)}\right)$, $i = 1,\ 2, \ldots,\ N$. This notation is convenient for multidimensional data: $x^{(i)} = \left(x_1^{(i)},\ x_2^{(i)},\ \ldots,\ x_d^{(i)}\right)$

ML Model (or Algorithm)

Example. We want to predict the best place for a new restaurant based on the revenue. We have information about the distance to the nearest restaurant and average traffic on the street. The simplest model is the linear model:

$$a(x) = w_0 + w_1 x + w_2 x_2.$$

ML model/algorithm: The goal of ML algorithm is for given characteristics/features of the data predict the target: $a(x) \approx y$. **Training/Learning** is the process of identifying of the coefficients w_0 and w_1 (weights in front the features) from the available data.

Prediction: Let's say we trained our linear model and found the values of w_0, w_1, and w_2. If the nearest restaurant is in 1 mile and in average 5 cars/minute pass the place, the predicted revenue is given by

$$a = w_0 + w_1 + 2w_2.$$

Loss Function

Loss function is a function of two arguments $Loss(a(x),\ y)$ taking values in $[0,\ \infty)$, such that $Loss(a(x),\ y) = 0$ if an only if $a(x) = y$.

1. 0-1 loss (counting or Hamming distance)

$$Loss(a(x),\ y) = [a(x) \neq y]$$

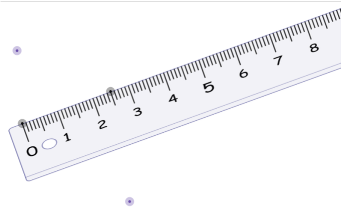
2. L_2 loss (Euclidean distance)

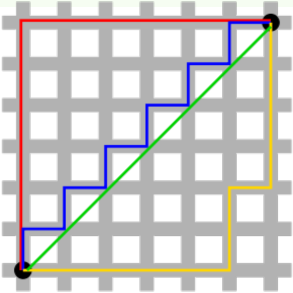
$$Loss(a(x),\ y) = \sqrt{(a_1 - y_1)^2 + \ldots + (a_d - y_d)^2}$$

3. L_1 loss (Manhattan distance)

$$Loss(a(x),\ y) = |a_1 - y_1| + \ldots + |a_d - y_d|$$

ℓ_p Distance





Euclidean (or ℓ_2 distance) $d\left(x^{(1)},\ x^{(2)}\right) = \sqrt{\sum_{j=1}^{d}\left(x_j^{(1)} - x_j^{(2)}\right)^2}$

Manhattan Distance (or ℓ_1) $d\left(x^{(1)},\ x^{(2)}\right) = \sum_{j=1}^{d}\left|x_j^{(1)} - x_j^{(2)}\right|$

ℓ_p distance $d\left(x^{(1)},\ x^{(2)}\right) = \sqrt[p]{\sum_{j=1}^{d}\left|x_j^{(1)} - x_j^{(2)}\right|^p}$

Image source: 1) https://teacher.desmos.com/activitybuilder/custom/60527c839f5bc7445f2ac793?collections=5f2c6c2c3a7f6a21c8c5d7e8. 2) https://en.wikipedia.org/wiki/Taxicab_geometry

Expected Loss and Training/Learning the Model

Expected Loss: usually the distribution of the data $(X, \ Y)$ is unknown and empirical distribution, i.e., uniform with probability of every sample equal to $1/N$, is used:

$$\mathrm{E}[Loss(a(X), \ Y)] = \int\limits_{\mathbf{X}\times\mathbf{Y}} Loss(a(x), \ y)p(x, \ y)dxdy$$
$$\approx \frac{1}{N}\sum_{i=1}^{N} Loss\big(a(x^{(i)}), \ y^{(i)}\big).$$

If $a(x) = w_0 + w_1 x_1 + \ldots + w_d x_d$, then w can be found from the optimization problem

$$\frac{1}{N}\sum_{i=1}^{N} Loss\big(a(x^{(i)}), \ y^{(i)}\big) \to \min_{w}.$$

Quality Metrics Examples

Example 1.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(a\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Example 2.

$$Error = \frac{1}{N}\sum_{i=1}^{N}\left[a\left(x^{(i)}\right) \neq y^{(i)}\right] = \frac{\#incorrect\ \ predictions}{N}$$

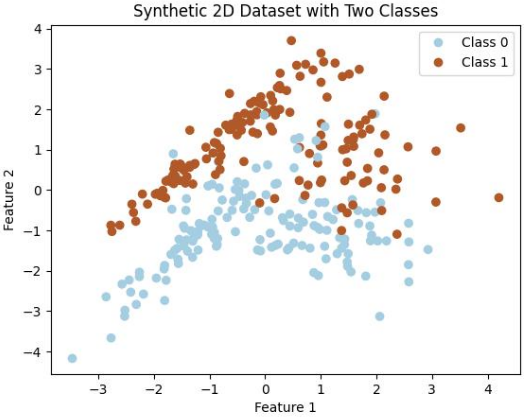
$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\left[a\left(x^{(i)}\right) = y^{(i)}\right] = \frac{\#correct\ \ predictions}{N}$$

Features

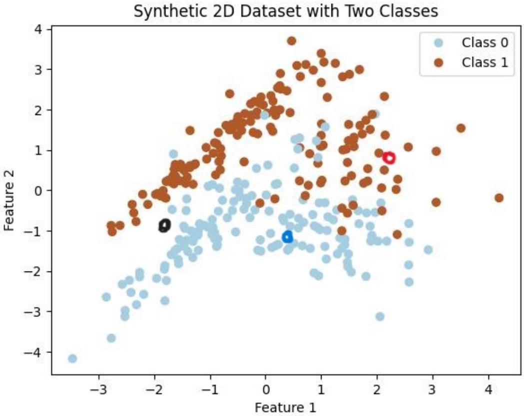
- Binary/Boolean 0, 1
- Numerical
- Categorical: ordinal and nominal
- Textual
- Datetime
- Geospatial
- Image
- Audio

PROBLEM SOLVING

Classification Problem



Classification Problem and Similarity Principle



Classification Problem and Similarity Principle

We will assume that if the features are similar, then the objects are from the same class

k Nearest Neighbors (kNN): training

- We are given $\left(x^{(i)},\ y^{(i)}\right)$, $i = 1,\ 2, \ldots,\ N$.
- $\mathbb{Y} = \{1,\ 2,\ \ldots,\ C\}$ (classification problem).
- training $=$ memorizing of the given data.

k Nearest Neighbors (kNN): prediction

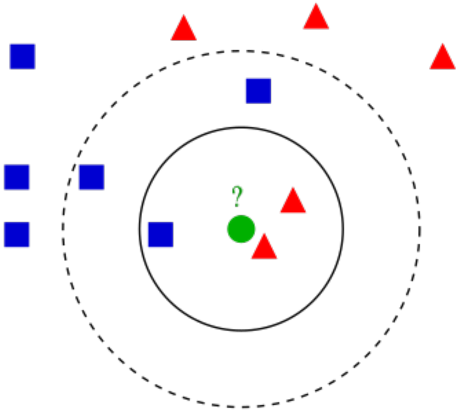
- We have a new feature x.
- Define the distance between new feature and $x^{(i)}$: $d\left(x,\ x^{(i)}\right)$.
- Rearrange the objects by the closeness to x :

$$d\left(x,\ x^{(i_1)}\right) \leq d\left(x,\ x^{(i_2)}\right) \leq \ldots \leq d\left(x,\ x^{(i_N)}\right).$$

- Look at the first k labels and assign the class with the highest number of representatives:

$$a(x) = \underset{c \in \mathbb{Y}}{argmax} \sum_{s=1}^{k} \left[y^{(i_s)} = c\right].$$

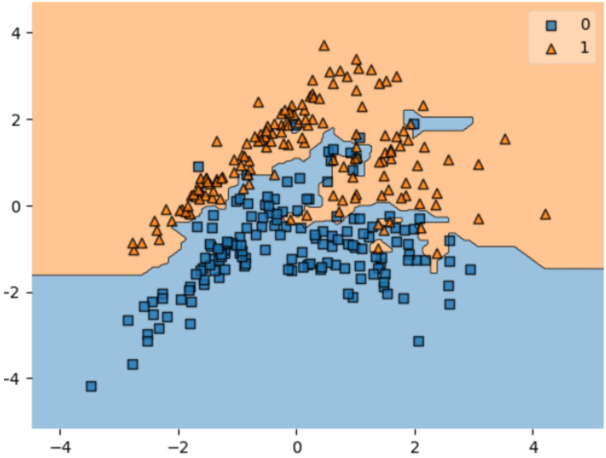
k Nearest Neighbors (kNN): prediction



https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

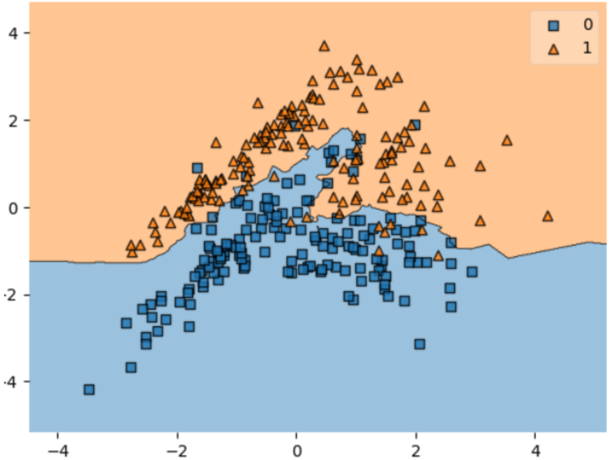
k Nearest Neighbors (kNN): prediction

$$k = 1,\ p = 1$$



k Nearest Neighbors (kNN): prediction

$$k = 5,\ p = 2$$

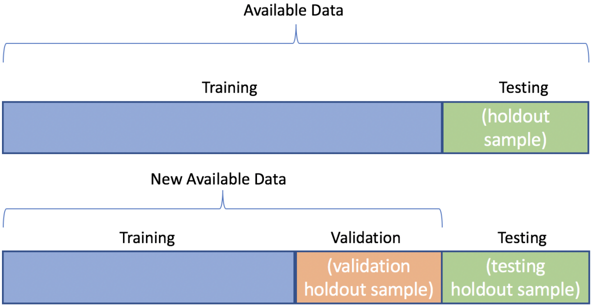


Model Comparison

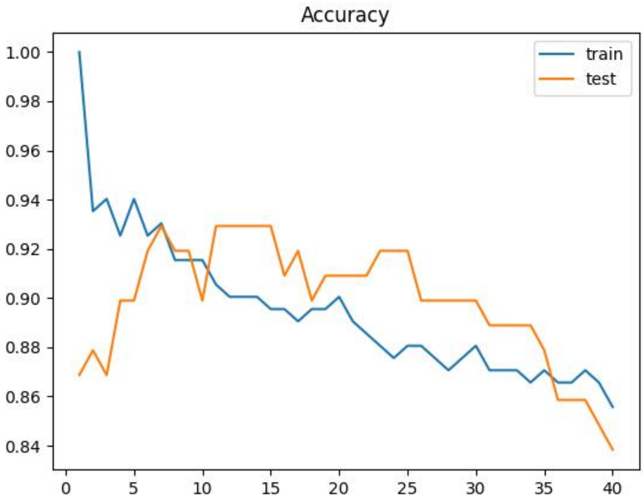
- How to compare models?
- How to choose k?

Hyper parameters

k can not be tuned on the training data!



How to choose k?



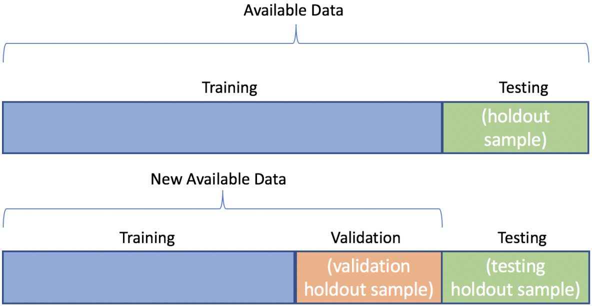
Overfitting and Generalization

Exam preparation:

Memorized lectures

Understand material

Typical Split



- Train set: build the model
- Validation set: tune hyper parameters
- Test set: evaluate quality of your model