CS 221: Section #1

Foundations

Roadmap

1. Probability
2. Linear Algebra
3. Python Tips
4. Recurrence

Machine Learning

Machine Learning 101

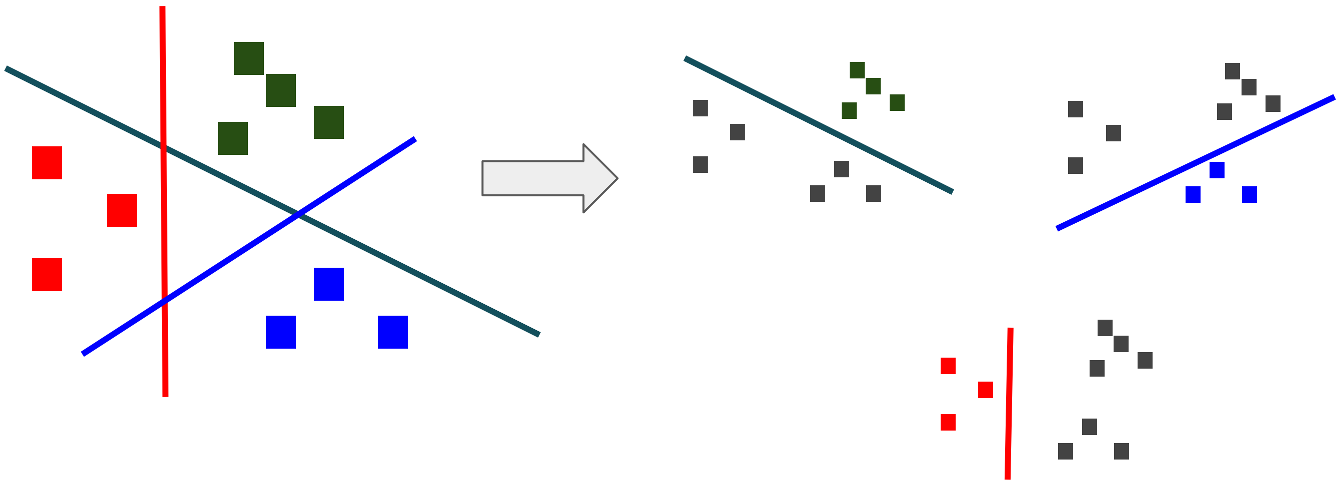
- Representation of our **data**
- Some **target** value
- Want to find a **predictor** or estimator
- Best possible predictor minimizes a **loss function**

Binary Classification

$$\boldsymbol{x} \Rightarrow \boxed{f_w} \Rightarrow y \in \{-1, 1\}$$

Multiclass Classification

- Extension of binary
- Example: Classify if something is red, green or blue



Loss functions

- Estimator or predictor from a parameterized family f_w
- How to choose our estimator f_w or pick our parameter w?
- “Best possible” estimator minimizes unhappiness on training data

Loss functions

- Ideal is a 0-1 loss:

$$loss_{0-1}(x, y, w) = \begin{cases} 1 & if\ \hat{y} = y \\ 0 & otherwise \end{cases}$$

- Problem?

Loss functions

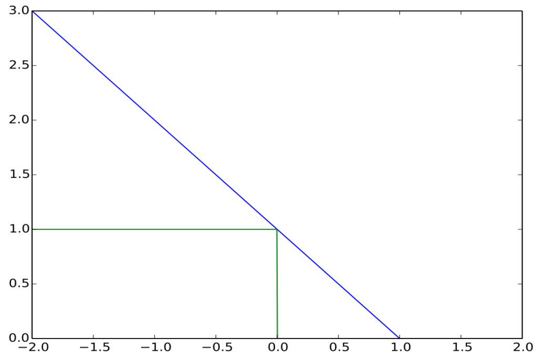


Photo taken from https://en.wikipedia.org/wiki/Hinge_loss

- How to select optimal w?
- Continuous approximation of 0-1 loss
- Example: Hinge loss

$$loss_{hinge}(x, y, w) = max\{1 - (w \cdot \phi(x))y, 0\}$$

- Example: Logistic regression

Probability

Random Variables

- Discrete: $\mathbb{P}(X = a)$ OR $p_X(a)$
- Example: Rolling a dice. Outcomes {1, 2, 3, 4, 5, 6}

- Continuous: $\mathbb{P}(X \leq a) = \int_{-\infty}^{a} f_X(u) du$
- Example: Uniform random variable in [0, 1]

Conditional Probability

- What is the probability that event A occurs given that event B has occurred.
- Denoted $\mathbb{P}(A|B)$

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Example

	$\mathbf{A = 0}$	$\mathbf{A = 1}$	$\mathbf{A = 2}$	$\mathbf{A = 3}$
$\mathbf{B = 0}$	0.1	0.25	0.1	0.05
$\mathbf{B = 1}$	0.15	0	0.15	0.2

- What is $\mathbb{P}(A = 2)$
- What is $\mathbb{P}(A = 2 \mid B = 1)$

Independence

- A random variable ***X*** (event *A)* is independent of a random variable ***Y*** (event *B*) if the realization of ***Y*** (or B) does not affect the probability distribution of ***X*** (or *A*).
- Example: Suppose we toss a coin and roll a die. What is the probability that 5 appears on the die given that heads appeared on the coin?

Expectation

$$\mathbb{E}[A] = \sum_a a\,\mathbb{P}[A = a]$$

$$\mathbb{E}[A] = \int a f_A(a)\,da$$

Example

	$\mathbf{A=0}$	$\mathbf{A=1}$	$\mathbf{A=2}$	$\mathbf{A=3}$
$\mathbf{B=0}$	0.1	0.25	0.1	0.05
$\mathbf{B=1}$	0.15	0	0.15	0.2

- Are A and B independent?
- What are $\mathbb{E}[A]$, $\mathbb{E}[B]$, $\mathbb{E}[A+B]$

Example

	A = 0	**A = 1**	**A = 2**	**A = 3**
B = 0	0.1	0.25	0.1	0.05
B = 1	0.15	0	0.15	0.2

- Are A and B independent?
- What are $\mathbb{E}[A]$, $\mathbb{E}[B]$, $\mathbb{E}[A+B]$

Linearity of Expectation: $\mathbb{E}[A+B] = \mathbb{E}[A] + \mathbb{E}[B]$

True even when A and B are dependent!

Example

Suppose n hatted people toss their hats into the air and pick up one hat at random

In expectation, how many people get their own hats back?

Hint: linearity of expectation

Linear Algebra

Useful Properties

$$\boldsymbol{v}^2 = ||\boldsymbol{v}||_2^2 = \boldsymbol{v}^T\boldsymbol{v}$$

$$(\boldsymbol{A} + \boldsymbol{B})^T = \boldsymbol{A}^T + \boldsymbol{B}^T$$

$$(\boldsymbol{A}\boldsymbol{B})^T = \boldsymbol{B}^T\boldsymbol{A}^T$$

Mean Squared Error:

$$L = \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Gradient of the weights:

$$\frac{\partial L}{\partial w} = \frac{2}{n} \sum_{i=1}^{n} (y_i - w^T x_i) x_i$$

Mean Squared Error:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2$$

Gradient of the label:

$$\frac{\partial L}{\partial y_i} = \frac{2}{n}(y_i - w^T x_i)$$

$$\frac{\partial L}{\partial \boldsymbol{y}} = \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \vdots \\ \frac{\partial L}{\partial y_n} \end{bmatrix} = \frac{2}{n} \begin{bmatrix} (y_1 - w^T x_1) \\ \vdots \\ (y_n - w^T x_n) \end{bmatrix}$$

EXAMPLE PROBLEM 1:
Binary classification, stochastic gradient descent

[White board]

Python Tips

Recurrences

Leveraging recursion

- Overlapping subproblems
- Optimal substructure
- Convert the given problem into a smaller (easier) one.

Example: Edit distance (In more detail)

- Question we are trying to answer is: What is the minimum number of edits do we need to make to transform word **a** into word **b?**
- (Also known as Levenshtein distance)