Perceptron Learning Algorithm

ROHU00120 “Data and text mining based on artificial intelligence research applied supporting accounting and financial decision-making (using R statistics and Python)”

János Szenderák

These slides are based on Wolfgang Ertel's (2024). Introduction to artificial intelligence. Springer Nature, 356 pages.

The material presented here is used solely for educational purposes.

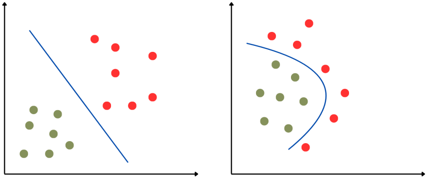
What is AI and Machine Learning?

- Elaine Rich: "Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better".
- Machine learning is a subfield of artificial intelligence (AI) whereby algorithms “learn” patterns in data to perform specific tasks.
- How much data we have/we need?
- How much computational power we have/we need?

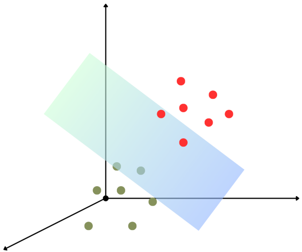
UNIVERSITY *of* DEBRECEN

The main ideas

Which one is a linearly separable dataset?



The main ideas



How to separate a set of points?

- Draw a straight line.
- If training data can be separated by a straight line, it is linearly separable.
- In n dimensions, a hyperplane is needed for separation.
- Hyperplane = linear subspace of dimension $n-1$.

$$\text{The } (n-1)\text{-dimensional hyperplane in } \mathbb{R}^n\text{:} \quad \sum_{i=1}^{n} a_i x_i = \theta$$

Equation of a Line

$$\sum_{i=1}^{n} a_i x_i = \theta$$

- This is the general equation of a hyperplane.

Linear Separability

Two sets $M_1 \subset \mathbb{R}^n$ and $M_2 \subset \mathbb{R}^n$ are linearly separable if real numbers $a_1, \ldots, a_n, \theta$ exist such that:

$$\sum_{i=1}^{n} a_i x_i > \theta \quad \forall x \in M_1,$$

$$\sum_{i=1}^{n} a_i x_i \leq \theta \quad \forall x \in M_2$$

$$\theta = \text{threshold}$$

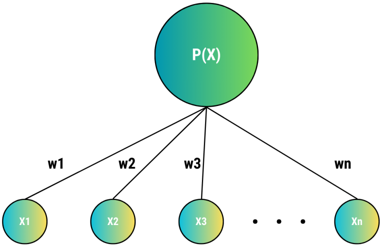
A Simple Learning Algorithm

Let $w = (w_1, \ldots, w_n) \in \mathbb{R}^n$ be a weight vector and $x \in \mathbb{R}^n$ an input vector. A perceptron represents a function $P : \mathbb{R}^n \to \{0, 1\}$:

$$P(x) = \begin{cases} 1 & \text{if } w \cdot x = \sum_{i=1}^{n} w_i x_i > 0 \\ 0 & \text{else} \end{cases}$$

Perceptron = two-layer neural network with activation.

Graphical Representation



Graphical representation of a perceptron as a two-layer neural network.

Classification Rule

$$\sum_{i=1}^{n} w_i x_i > 0$$

- Points x above hyperplane $\sum w_i x_i = 0$ are positive ($P(x) = 1$).
- Other points are negative ($P(x) = 0$).
- Separating hyperplane goes through origin ($\theta = 0$).

UNIVERSITY *of* DEBRECEN

Learning Rule

$$M_+ = \text{positive training patterns}, \quad M_- = \text{negative training patterns}$$

$$\text{Perceptron Learning}[M_+, M_-]$$

- $w =$ arbitrary vector of reals.
- Repeat:
  - For all $x \in M_+$: if $wx \leq 0$, then $w = w + x$.
  - For all $x \in M_-$: if $wx > 0$, then $w = w - x$.
- Until all $x \in M_+ \cup M_-$ are correctly classified.

Output: $P(x) = 1$ for all $x \in M_+$.

Weight Vector Update

- If $wx > 0 \implies P(x) = 1$.
- If not, update weight vector:

$$w \leftarrow w + x$$

- Repeated updates $\Rightarrow$ wx eventually positive.
- For negative training data:

$$(w - x)x = wx - x^2$$

Example

Training sets:

$$M_+ = \{(0, 1.8), (2, 0.6)\}, \quad M_- = \{(-1.2, 1.4), (0.4, -1)\}$$

Initial weight vector:

$$w = (1, 1)$$

Decision boundary:

$$wx = x_1 + x_2 = 0$$

Convergence

Let classes M_+ and M_- be linearly separable by hyperplane $wx = 0$.

- The perceptron learning algorithm converges for every initialization of w.
- The perceptron P with weight vector w divides classes:

$$P(x) = 1 \Leftrightarrow x \in M_+, \quad P(x) = 0 \Leftrightarrow x \in M_-$$

Bias Term

- If θ is missing, perceptron divides sets only by line through origin.
- Trick: add constant input $x_n = 1$.
- Weight $w_n = -\theta$ acts as threshold.
- $x_n = 1$ is called the bias unit $\Rightarrow$ shifts hyperplane.

Perceptron with Threshold

$$P_\theta(x_1, \ldots, x_n) = \begin{cases} 1 & \sum_{i=1}^{n} w_i x_i > \theta \\ 0 & \text{else} \end{cases}$$

- Any arbitrary threshold can be simulated by perceptron $P : \mathbb{R}^n \to \{0, 1\}$ with $\theta = 0$ and bias unit.

Summary

- A function $f : \mathbb{R}^n \to \{0, 1\}$ can be represented by a perceptron iff positive and negative sets are linearly separable.
- Perceptron is the simplest neural network model.
- Equivalent to Naïve Bayes: the simplest Bayesian network.

The Nearest Neighbor Method

The Nearest Neighbor Method

- For perceptron, knowledge available in training data is saved in weights w_i.
- If the system should generalize to new data, generalization is time-intensive.
- Goal: compact representation of data in the form of a function that classifies data well.
- Nearest Neighbor Method (NNM) uses memorization of all data by simply saving them.

UNIVERSITY of DEBRECEN

Problem: Applying Knowledge

- How to apply saved knowledge to new data?
- Example: Skiing
  - Collecting and saving data is not enough.
  - Practice is needed.
  - Represented by w_i.
- Medical example:
  - During diagnosis of a difficult case, remembering similar past examples helps.
  - Memorization enables generalization.
  - Need for a similarity measurement.

Similarity

- Smaller distance in feature space $\Rightarrow$ more similar two examples.
- Distance measure: Euclidean distance

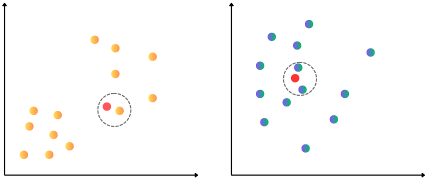
$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Weighted version:

$$d_w(x, y) = |x - y| = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}$$

The main ideas

The 1-nearest neighbor and the 3-nearest neighbor example (k = 1 and k = 3)



Nearest Neighbor Algorithm

$$NN[M_+, M_-, s]$$

$$t = \arg \min_{x \in M_+ \cup M_-} d(s, x)$$

$$\text{If } t \in M_+ \text{ then return } (+), \quad \text{else return } (-)$$

- $t =$ training data
- $s =$ new example

Voronoi Diagram

- Visualization tool for nearest neighbor classification.
- Divides space into regions around training points.
- Each region corresponds to the set of points closest to a training example.

NN vs Perceptron

- NN does not generate a line.
- NN is significantly more powerful than perceptron.
- Can correctly realize arbitrarily complex dividing lines.
- A single erroneous point can lead to a very bad classification.

K-NN Smoothing

- To smooth decision surface, use k-nearest neighbors:

$$V = \{k \text{ nearest neighbors in } M_+ \cup M_-\}$$

- If $|M_+ \cap V| > |M_- \cap V|$, return $+$.
- If $|M_+ \cap V| < |M_- \cap V|$, return $-$.
- Else, return random$(+, -)$.

Example

- Example of K-NN classification on board as a class exercise.
- Training points influence the classification based on the majority of nearest neighbors.

Distance Measurement Importance

- In large datasets, there are more neighbors at a large distance than at a small distance.
- K-nearest neighbors average function value:

$$\hat{f}(x) = \frac{1}{k} \sum_{i=1}^{k} f(x_i)$$

- As k grows, estimate dominated by far-away neighbors.
- To fix this, weight neighbors:

$$w_i = \frac{1}{1 + \alpha d(x, x_i)^2}$$

- Weighted average:

$$f(x) = \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

- Problem: selecting α.

Bayes's Theoream and the Naiv Bayes Estimator

Source: Think Bayes, 2nd Edition by Allen B. Downey, O'Reilly Media, Inc. 335 p.

UNIVERSITY of DEBRECEN

Bayes's Theorem and Naive Bayes Estimation

- Necessary concept: conditional probability. Is there a difference?
- Example:
  - A Hungarian person will have a heart attack.
  - A Hungarian man, who is unhealthy, smokes and works in finance, will get a heart attack?
- Probability = number between 0 and 1 representing degree of belief:
  - 1 = certainty fact is true
  - 0 = certainty fact is false
  - Intermediate values = degrees of certainty (different from frequentist interpretation).

UNIVERSITY of DEBRECEN

Conditional Probability

- Heart attack in Hungary: 15,000 cases/year.
- $15,000/9.5$ million $\approx 0.15\%$ if chosen randomly.
- Usual notation:

$$P(A|B) = \text{probability of event A given B is true}$$

- Example:

$$P(\text{heart attack}|\text{smoker}) > P(\text{heart attack}|\text{non-smoker})$$

Conjoint Probability

- Probability that two events are true:

$$P(A \wedge B) = P(A \text{ and } B)$$

- If A and B are independent:

$$P(A \wedge B) = P(A)P(B)$$

- Knowing one event's outcome does not help predict the other.

Conjoint Probability

Non-independent events:

- A: rains today
- B: rains tomorrow
- $P(B|A) > P(A)$
- Probability of conjunction:

$$P(AB) = P(A)P(B|A) \quad \text{or} \quad P(AB) = P(B)P(A|B)$$

Bayes's Theorem

- Put them together:

$$P(AB) = P(A)P(B|A) = P(B)P(A|B)$$

- Divide through by $P(B)$:

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

- Surprisingly powerful – but why? (Hot-cold game)

Diachronic Interpretation

- Another way: The Bayes theorem updates the probability of hypothesis H in light of the data D.
- Diachronic = something happens over time.

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)}$$

Bayes's Theorem

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)}$$

- $P(H)$: prior probability (before data).
- $P(H|D)$: posterior probability (after data).
- $P(D|H)$: likelihood (probability of data under hypothesis).
- $P(D)$: normalizing constant.

Bayes's Theorem and Hypotheses

- Simplify by specifying hypotheses that are:
  - Mutually exclusive: at most one can be true.
  - Collectively exhaustive: at least one must be true.
- If we compute $P(D)$, we can use law of total probability:

$$P(D) = P(B_1)P(D|B_1) + P(B_2)P(D|B_2)$$

Law of Total Probability: Example

Events:

- A: person is late for work
- B_1: person travels by bus
- B_2: person travels by train

$$P(B_1) = 0.6, \quad P(B_2) = 0.4$$

$$P(A|B_1) = 0.2, \quad P(A|B_2) = 0.3$$

$$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) = 0.6 \cdot 0.2 + 0.4 \cdot 0.3 = 0.24$$

UNIVERSITY of DEBRECEN

Law of Total Probability: Extreme Example

Events:

- A: person is late for work
- B_1: travels by train
- B_2: travels by car

$$P(B_1) = 0.1, \quad P(B_2) = 0.9$$

$$P(A|B_1) = 0.9, \quad P(A|B_2) = 0.1$$

$$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) = 0.1 \cdot 0.9 + 0.9 \cdot 0.1 = 0.18$$

UNIVERSITY of DEBRECEN

Naive Bayes Setup

- Classes: $w \in \{\text{Risky}, \text{Invest}\}$.
- Features:
  - x_1: Growth status (No growth / High growth)
  - x_2: Credit score (Good = + / Bad = -)
- Data: 12 companies
  - 7 classified as Risky
  - 5 classified as Invest

Prior Probabilities

$$P(\text{Risky}) = \frac{7}{12} \approx 0.583, \quad P(\text{Invest}) = \frac{5}{12} \approx 0.417$$

The prior reflects the baseline likelihood of encountering a risky vs.

investable company before observing any features.

Class-Conditional Probabilities: Risky

Within the 7 risky companies:

- Growth:

$$P(\text{No Growth} \mid \text{Risky}) = \tfrac{5}{7}, \quad P(\text{High Growth} \mid \text{Risky}) = \tfrac{2}{7}$$

- Credit:

$$P(+ \mid \text{Risky}) = \tfrac{4}{7}, \quad P(- \mid \text{Risky}) = \tfrac{3}{7}$$

Class-Conditional Probabilities: Invest

Within the 5 investable companies:

- Growth:

$$P(\text{No Growth} \mid \text{Invest}) = \tfrac{3}{5}, \quad P(\text{High Growth} \mid \text{Invest}) = \tfrac{2}{5}$$

- Credit:

$$P(+ \mid \text{Invest}) = \tfrac{4}{5}, \quad P(- \mid \text{Invest}) = \tfrac{1}{5}$$

Prediction Example A

New firm: High growth, Good credit $(x_1 = \text{High},\ x_2 = +)$

$$P(\text{Risky} \mid x) \propto \frac{7}{12} \cdot \frac{2}{7} \cdot \frac{4}{7} = \frac{4}{42} \approx 0.095,$$
$$P(\text{Invest} \mid x) \propto \frac{5}{12} \cdot \frac{2}{5} \cdot \frac{4}{5} = \frac{8}{60} \approx 0.133.$$

$$P(\text{Risky} \mid x) \approx 0.417, \quad P(\text{Invest} \mid x) \approx 0.583$$

Prediction: Invest

UNIVERSITY of DEBRECEN

Prediction Example B

New firm: No growth, Bad credit ($x_1 = \text{No}, \; x_2 = -$)

$$P(\text{Risky} \mid x) \propto \frac{7}{12} \cdot \frac{5}{7} \cdot \frac{3}{7} = \frac{15}{84} \approx 0.179,$$
$$P(\text{Invest} \mid x) \propto \frac{5}{12} \cdot \frac{3}{5} \cdot \frac{1}{5} = \frac{3}{60} = 0.05.$$

$$P(\text{Risky} \mid x) \approx 0.782, \quad P(\text{Invest} \mid x) \approx 0.218$$

Normalize by the total of $(0.179 + 0.05)$

Prediction: Risky

Summary

- Naive Bayes combines prior probabilities with feature likelihoods.
- Assumes conditional independence of features given the class.
- Computationally inexpensive and probabilistic
- It commonly performs very well and can handle cases with missing data
- It assumes that continuous predictor variables are normally distributed (typically).
- It assumes that predictor variables are independent of each other, which usually isn't true.