



MONASH University

ETC3555

Statistical Machine Learning

Linear models

7 August 2018

Outline

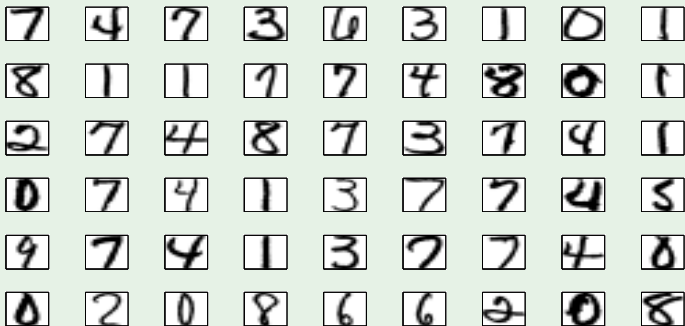
1 Linear classification

2 Linear regression

3 Logistic regression

A real data set

A real data set



Feature extraction

Input representation

'raw' input $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{256})$

linear model: $(w_0, w_1, w_2, \dots, w_{256})$

Features: Extract useful information, e.g.,

intensity and symmetry $\mathbf{x} = (x_0, x_1, x_2)$

linear model: (w_0, w_1, w_2)

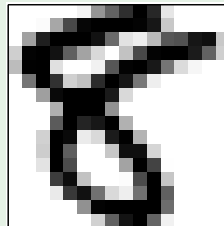


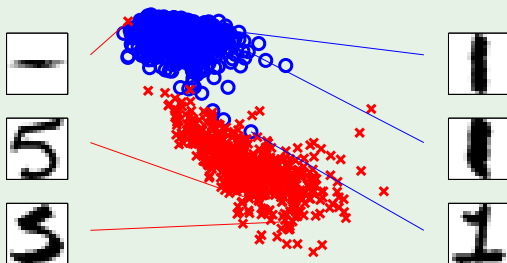
Illustration of features

Illustration of features

$$\mathbf{x} = (x_0, x_1, x_2)$$

x_1 : intensity

x_2 : symmetry



The perceptron learning algorithm

A simple learning algorithm - PLA

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$$

Given the training set:

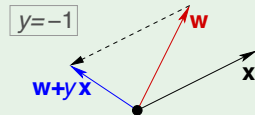
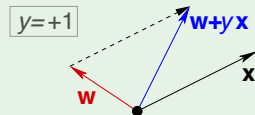
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$$

pick a **misclassified** point:

$$\text{sign}(\mathbf{w}^T \mathbf{x}_n) \neq y_n$$

and update the weight vector:

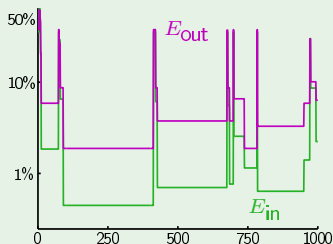
$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$$



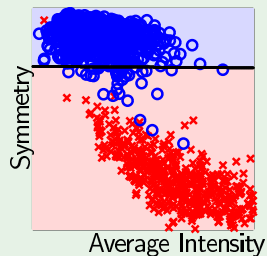
The perceptron learning algorithm

What PLA does

Evolution of E_{in} and E_{out}



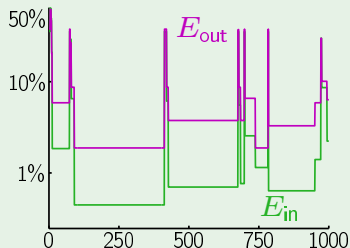
Final perceptron boundary



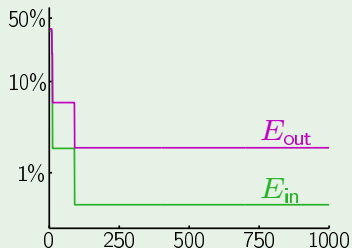
The pocket algorithm

The 'pocket' algorithm

PLA:



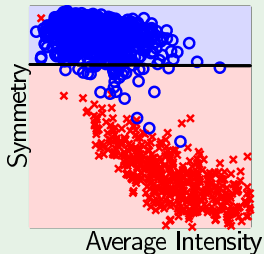
Pocket:



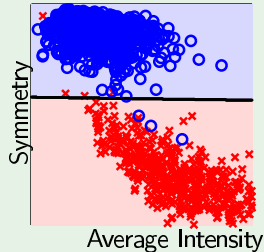
PLA versus Pocket

Classification boundary - PLA versus Pocket

PLA:



Pocket:



Outline

1 Linear classification

2 Linear regression

3 Logistic regression

Credit example

Credit again

Classification: Credit approval (yes/no)

Regression: Credit line (dollar amount)

Input: $\mathbf{x} =$

age	23 years
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
...	...

Linear regression output: $h(\mathbf{x}) = \sum_{i=0}^d w_i x_i = \mathbf{w}^T \mathbf{x}$

Error measure for regression

How to measure the error

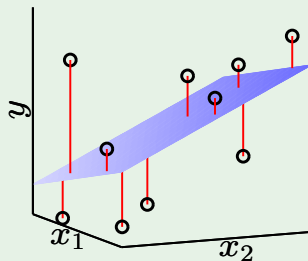
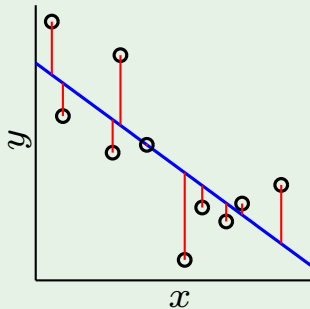
How well does $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ approximate $f(\mathbf{x})$?

In linear regression, we use squared error $(h(\mathbf{x}) - f(\mathbf{x}))^2$

in-sample error:
$$E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^N (h(\mathbf{x}_n) - y_n)^2$$

Geometry of linear regression

Illustration of linear regression



E_{in} in vector form

The expression for E_{in}

$$\begin{aligned} E_{\text{in}}(\mathbf{w}) &= \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n - y_n)^2 \\ &= \frac{1}{N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \end{aligned}$$

where

$$\mathbf{X} = \begin{bmatrix} -\mathbf{x}_1^T \\ -\mathbf{x}_2^T \\ \vdots \\ -\mathbf{x}_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Minimizing E_{in}

Minimizing E_{in}

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$$

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{2}{N} \mathbf{X}^T (\mathbf{X}\mathbf{w} - \mathbf{y}) = \mathbf{0}$$

$$\mathbf{X}^T \mathbf{X} \mathbf{w} = \mathbf{X}^T \mathbf{y}$$

$$\mathbf{w} = \mathbf{X}^\dagger \mathbf{y} \quad \text{where} \quad \mathbf{X}^\dagger = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$$

\mathbf{X}^\dagger is the 'pseudo-inverse' of \mathbf{X}

The linear regression algorithm

The linear regression algorithm

- 1: Construct the matrix X and the vector \mathbf{y} from the data set $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ as follows

$$X = \underbrace{\begin{bmatrix} -\mathbf{x}_1^\top - \\ -\mathbf{x}_2^\top - \\ \vdots \\ -\mathbf{x}_N^\top - \end{bmatrix}}_{\text{input data matrix}}, \quad \mathbf{y} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\text{target vector}}.$$

- 2: Compute the pseudo-inverse $X^\dagger = (X^\top X)^{-1} X^\top$.
- 3: Return $\mathbf{w} = X^\dagger \mathbf{y}$.

Linear regression for classification

Linear regression for classification

Linear regression learns a real-valued function $y = f(\mathbf{x}) \in \mathbb{R}$

Binary-valued functions are also real-valued! $\pm 1 \in \mathbb{R}$

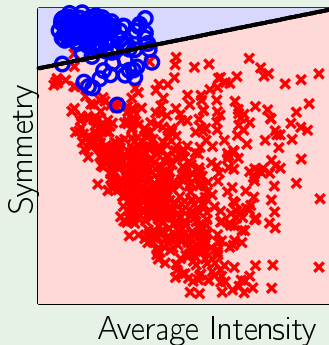
Use linear regression to get \mathbf{w} where $\mathbf{w}^T \mathbf{x}_n \approx y_n = \pm 1$

In this case, $\text{sign}(\mathbf{w}^T \mathbf{x}_n)$ is likely to agree with $y_n = \pm 1$

Good initial weights for classification

Linear regression boundary

Linear regression boundary



Outline

1 Linear classification

2 Linear regression

3 Logistic regression

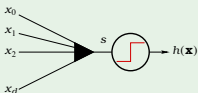
Logistic regression

A third linear model

$$s = \sum_{i=0}^d w_i x_i$$

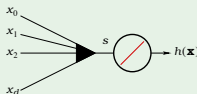
linear classification

$$h(\mathbf{x}) = \text{sign}(s)$$



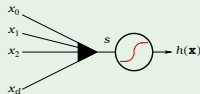
linear regression

$$h(\mathbf{x}) = s$$



logistic regression

$$h(\mathbf{x}) = \theta(s)$$



The logistic function

The logistic function θ

The formula:

$$\theta(s) = \frac{e^s}{1 + e^s}$$



soft threshold: uncertainty

sigmoid: flattened out 's'

Probability interpretation

Probability interpretation

$h(\mathbf{x}) = \theta(s)$ is interpreted as a probability

Example. Prediction of heart attacks

Input \mathbf{x} : cholesterol level, age, weight, etc.

$\theta(s)$: probability of a heart attack

The signal $s = \mathbf{w}^T \mathbf{x}$ "risk score"

Genuine probability

Genuine probability

Data (\mathbf{x}, y) with **binary** y , generated by a noisy target:

$$P(y | \mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{for } y = +1; \\ 1 - f(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

The target $f : \mathbb{R}^d \rightarrow [0, 1]$ is the probability

Learn $g(\mathbf{x}) = \theta(\mathbf{w}^\top \mathbf{x}) \approx f(\mathbf{x})$

The data does not give us the value of f explicitly. It gives us samples generated by this probability. How do we learn from such data?

Error measure

Error measure

For each (\mathbf{x}, y) , y is generated by probability $f(\mathbf{x})$

Plausible error measure based on **likelihood**:

If $h = f$, how likely to get y from \mathbf{x} ?

$$P(y | \mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1; \\ 1 - h(\mathbf{x}) & \text{for } y = -1. \end{cases}$$

Formula for likelihood

Since the data points are independently generated, the probability of observing all the y_n 's in the data set from the corresponding \mathbf{x}_n is

$$\prod_{n=1}^N P(y_n | \mathbf{x}_n).$$

The method of *maximum likelihood* selects the hypothesis h which maximizes this probability.

Maximizing the likelihood

$$\text{Maximize } \prod_{n=1}^N P(y_n | \mathbf{x}_n) \equiv \text{Minimize } -\frac{1}{N} \ln \left(\prod_{n=1}^N P(y_n | \mathbf{x}_n) \right)$$

$$\begin{aligned} & -\frac{1}{N} \ln \left(\prod_{n=1}^N P(y_n | \mathbf{x}_n) \right) \\ &= \frac{1}{N} \sum_{n=1}^N \ln \left(\frac{1}{P(y_n | \mathbf{x}_n)} \right) \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{y_n = +1\} \ln \left(\frac{1}{h(\mathbf{x}_n)} \right) + \mathbb{1}\{y_n = -1\} \ln \left(\frac{1}{1 - h(\mathbf{x}_n)} \right) \end{aligned}$$

Minimizing E_{in}

For the case $h(\mathbf{x}) = \theta(\mathbf{w}^T \mathbf{x})$, with $\theta(-\mathbf{s}) = 1 - \theta(\mathbf{s})$, we have

$$\begin{aligned} &= \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{y_n = +1\} \ln \left(\frac{1}{h(\mathbf{x}_n)} \right) + \mathbb{1}\{y_n = -1\} \ln \left(\frac{1}{1 - h(\mathbf{x}_n)} \right) \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{y_n = +1\} \ln \left(\frac{1}{\theta(\mathbf{w}^T \mathbf{x}_n)} \right) + \mathbb{1}\{y_n = -1\} \ln \left(\frac{1}{1 - \theta(\mathbf{w}^T \mathbf{x}_n)} \right) \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{y_n = +1\} \ln \left(\frac{1}{\theta(\mathbf{w}^T \mathbf{x}_n)} \right) + \mathbb{1}\{y_n = -1\} \ln \left(\frac{1}{\theta(-\mathbf{w}^T \mathbf{x}_n)} \right) \\ &= \frac{1}{N} \sum_{n=1}^N \ln \left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)} \right) = \underbrace{\frac{1}{N} \sum_{n=1}^N \ln \left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n} \right)}_{E_{\text{in}}(\mathbf{w})} \end{aligned}$$

→ “cross-entropy” error

Cross-entropy

For two probability distributions $\{p, 1 - p\}$ and $\{q, 1 - q\}$ with binary outcomes, the cross-entropy (from information theory) is

$$p \log \frac{1}{q} + (1 - p) \log \frac{1}{1 - q}.$$

The in-sample error above corresponds to a cross-entropy error measure on the data point (\mathbf{x}_n, y_n) , with $p = \mathbb{1}\{y_n = +1\}$ and $q = h(\mathbf{x}_n)$.

Formula for likelihood

Formula for likelihood

$$P(y \mid \mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{for } y = +1; \\ 1 - h(\mathbf{x}) & \text{for } y = -1. \end{cases}$$



Substitute $h(\mathbf{x}) = \theta(\mathbf{w}^\top \mathbf{x})$, noting $\theta(-s) = 1 - \theta(s)$

$$P(y \mid \mathbf{x}) = \theta(y \mathbf{w}^\top \mathbf{x})$$

Likelihood of $\mathcal{D} = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ is

$$\prod_{n=1}^N P(y_n \mid \mathbf{x}_n) = \prod_{n=1}^N \theta(y_n \mathbf{w}^\top \mathbf{x}_n)$$

Maximizing the likelihood

Maximizing the likelihood

Minimize

$$-\frac{1}{N} \ln \left(\prod_{n=1}^N \theta(y_n \mathbf{w}^\top \mathbf{x}_n) \right)$$
$$= \frac{1}{N} \sum_{n=1}^N \ln \left(\frac{1}{\theta(y_n \mathbf{w}^\top \mathbf{x}_n)} \right) \quad \left[\theta(s) = \frac{1}{1 + e^{-s}} \right]$$

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \underbrace{\ln \left(1 + e^{-y_n \mathbf{w}^\top \mathbf{x}_n} \right)}_{\epsilon(h(\mathbf{x}_n), y_n)} \quad \text{"cross-entropy" error}$$

How to minimize E_{in}

How to minimize E_{in}

For logistic regression,

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N \ln \left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n} \right) \quad \leftarrow \text{iterative solution}$$

Compare to linear regression:

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n - y_n)^2 \quad \leftarrow \text{closed-form solution}$$

Summary

Summary of Linear Models

