MONASH University

ETC3555

Statistical Machine Learning

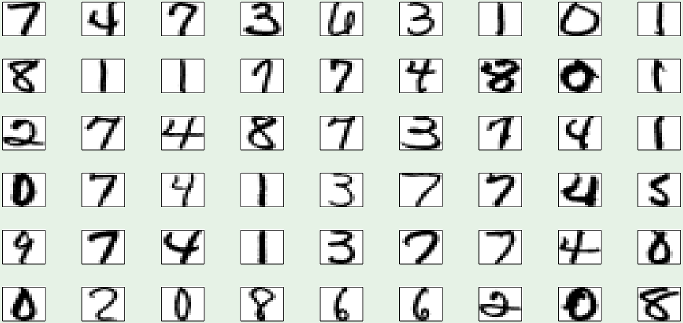
Linear models

7 August 2018

Outline

A real data set

A real data set

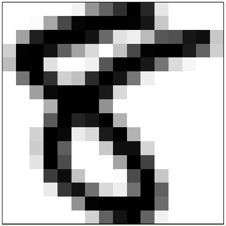


Feature extraction

Input representation

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

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



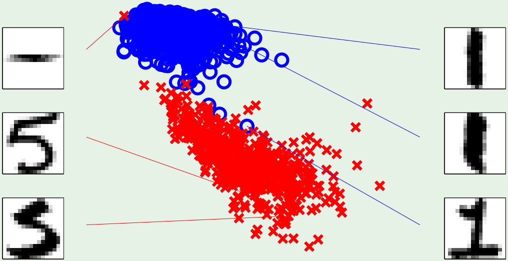
Features: Extract useful information, e.g.,

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

linear model: (w_0, w_1, w_2)

Illustration of features

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



5/23

The perceptron learning algorithm

A simple learning algorithm - PLA

The perceptron implements

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

Given the training set:

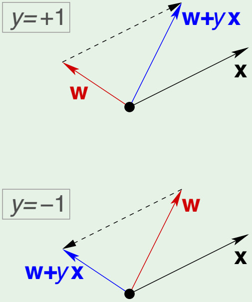
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

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

and update the weight vector:

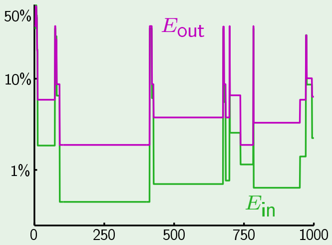
$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n\mathbf{x}_n}$$



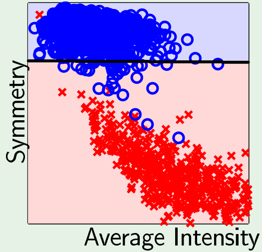
The perceptron learning algorithm

What PLA does

Evolution of E_{in} and E_{out}



Final perceptron boundary



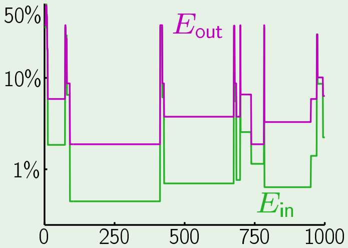
6/23

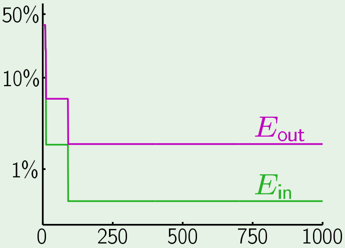
The pocket algorithm

The 'pocket' algorithm

PLA:

Pocket:





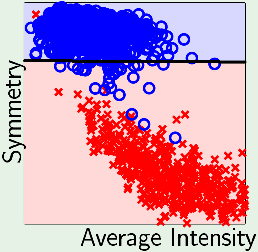
© Creator: Yaser Abu-Mostafa - LFD Lecture 3

7/23

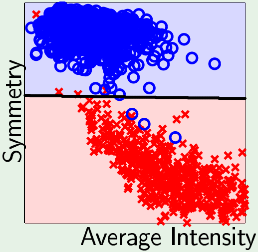
PLA versus Pocket

Classification boundary - PLA versus Pocket

PLA:



Pocket:



8/23

Outline

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
$\cdots$	$\cdots$

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

10/23

Error measure for regression

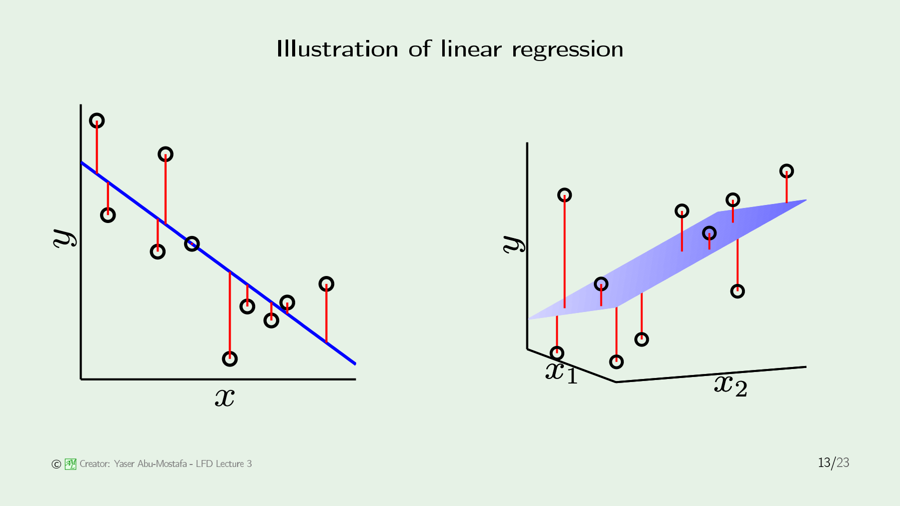
How to measure the error

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

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

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

Geometry of linear regression



13/23

E_{in} in vector form

The expression for E_{in}

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{w}^{\intercal}\mathbf{x}_n - y_n\right)^2$$

$$= \frac{1}{N}\|\mathrm{X}\mathbf{w} - \mathbf{y}\|^2$$

where $\mathrm{X} = \begin{bmatrix} \text{—}\mathbf{x}_1{}^{\intercal}\text{—} \\ \text{—}\mathbf{x}_2{}^{\intercal}\text{—} \\ \vdots \\ \text{—}\mathbf{x}_N{}^{\intercal}\text{—} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$

Minimizing E_{in}

Minimizing E_{in}

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

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

$$\mathrm{X}^{\intercal}\mathrm{X}\mathbf{w} = \mathrm{X}^{\intercal}\mathbf{y}$$

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

$\mathrm{X}^{\dagger}$ is the '**pseudo-inverse**' of X

The linear regression algorithm

The linear regression algorithm

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

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

2: Compute the pseudo-inverse $\mathrm{X}^\dagger = (\mathrm{X}^\intercal \mathrm{X})^{-1} \mathrm{X}^\intercal$.

3: Return $\mathbf{w} = \mathrm{X}^\dagger \mathbf{y}$.

 17/23

Linear regression for classification

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

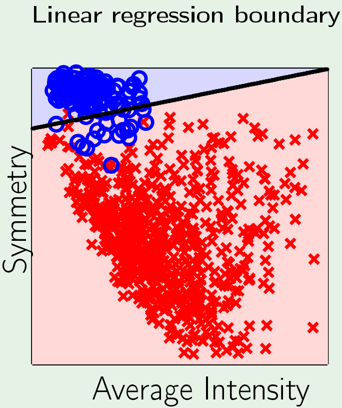
Binary-valued functions are also real-valued! $\pm 1 \in \mathbb{R}$

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

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

Good initial weights for classification

Linear regression boundary



18/23

Outline

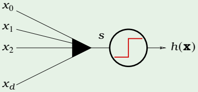
Logistic regression

A third linear model

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

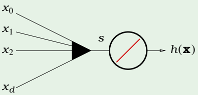
linear classification

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



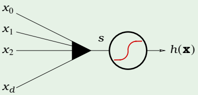
linear regression

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



logistic regression

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

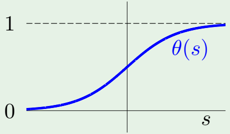


The logistic function

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

Genuine probability

Genuine probability

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

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

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

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

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 $\mathbf{x}$?

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

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 $\boldsymbol{x}_n$ is

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

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

Maximizing the likelihood

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

$$-\frac{1}{N} \ln\left(\Pi_{n=1}^{N} P(y_n|\boldsymbol{x}_n)\right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \ln\left(\frac{1}{P(y_n|\boldsymbol{x}_n)}\right)$$

$$= \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}\{y_n = +1\} \ln\left(\frac{1}{h(\boldsymbol{x}_n)}\right) + \mathbb{1}\{y_n = -1\} \ln\left(\frac{1}{1 - h(\boldsymbol{x}_n)}\right)$$

Minimizing E_{in}

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

$$= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\{y_n = +1\} \ln\left(\frac{1}{h(\boldsymbol{x}_n)}\right) + \mathbb{1}\{y_n = -1\} \ln\left(\frac{1}{1 - h(\boldsymbol{x}_n)}\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\{y_n = +1\} \ln\left(\frac{1}{\theta(w^T\boldsymbol{x}_n)}\right) + \mathbb{1}\{y_n = -1\} \ln\left(\frac{1}{1 - \theta(w^T\boldsymbol{x}_n)}\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\{y_n = +1\} \ln\left(\frac{1}{\theta(w^T\boldsymbol{x}_n)}\right) + \mathbb{1}\{y_n = -1\} \ln\left(\frac{1}{\theta(-w^T\boldsymbol{x}_n)}\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \ln\left(\frac{1}{\theta(y_n w^T\boldsymbol{x}_n)}\right) = \underbrace{\frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n w^T\boldsymbol{x}_n}\right)}_{E_{\text{in}}(\boldsymbol{w})}$$

$\rightarrow$ “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

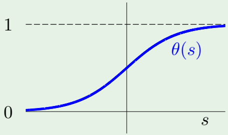
$$p \, log \, \frac{1}{q} + (1-p) \, log \, \frac{1}{1-q}.$$

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

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

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

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

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

15/24

Maximizing the likelihood

Maximizing the likelihood

Minimize
$$-\frac{1}{N}\ln\left(\prod_{n=1}^{N}\theta(y_n\,\mathbf{w}^{\mathsf{T}}\,\mathbf{x}_n)\right)$$

$$=\frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{\theta(y_n\,\mathbf{w}^{\mathsf{T}}\,\mathbf{x}_n)}\right)$$

$$\left[\theta(s)=\frac{1}{1\;+e^{-s}}\right]$$

$$E_{\text{in}}(\mathbf{w})=\frac{1}{N}\sum_{n=1}^{N}\underbrace{\ln\left(1+e^{-y_n\mathbf{w}^{\mathsf{T}}\mathbf{x}_n}\right)}_{\mathrm{e}\left(h(\mathbf{x}_n),y_n\right)}$$

"cross-entropy" error

How to minimize E_{in}

How to minimize E_{in}

For logistic regression,

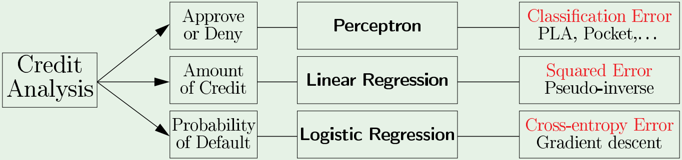
$$E_{\text{in}}(\mathbf{w}) \;=\; \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y_n \mathbf{w}^{\mathsf{T}} \mathbf{x}_n}\right) \qquad \longleftarrow \text{ iterative solution}$$

Compare to linear regression:

$$E_{\text{in}}(\mathbf{w}) \;=\; \frac{1}{N}\sum_{n=1}^{N} \left(\mathbf{w}^{\mathsf{T}} \mathbf{x}_n - y_n\right)^2 \qquad \longleftarrow \text{ closed-form solution}$$

Summary

Summary of Linear Models



24/24