### Machine Learning and Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

05.11.2014

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへで

## Overview and Structure

- 22.10.2014 Organisation
- 22.10.3014 Introduction (Def.: Machine learning, Supervised/Unsupervised, Examples)
- 29.10.2014 Machine Learning Basics (Toolchain, Features, Metrics, Rule-based)
- 05.11.2014 A simple Supervised learning algorithm
  - 12.11.2014 Excursion: Avoiding local optima with random search
  - 19.11.2014 -
- 26.11.2014 Bayesian learner
- 03.12.2014 -
- 10.12.2014 Decision tree learner
- 17.12.2014 k-nearest neighbour
- 07.01.2015 Support Vector Machines
- 14.01.2015 Artificial Neural networks and Self Organizing Maps
- 21.01.2015 Hidden Markov models and Conditional random fields
- 28.01.2015 High dimensional data, Unsupervised learning
- 04.02.2015 Anomaly detection, Online learning, Recom. systems

Machine Learning and Pervasive Computing

・ 同 ト ・ ヨ ト ・ ヨ ト … ヨ

Multivariate

ate Lo

Logistic regression

## Outline

Linear regression

Least squares estimation

Polynomial regression

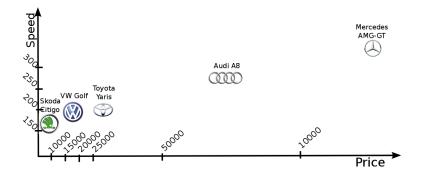
Multivariable linear regression

Multivariate linear regression

Logistic regression

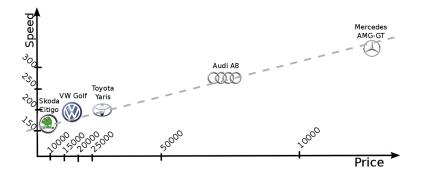
- ▲ ロ ト ▲ 団 ト ▲ 国 ト → 国 - りへで

### Linear regression



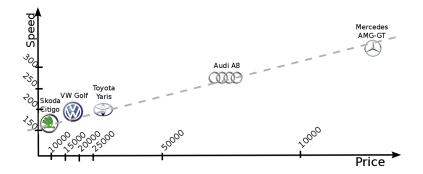
### ▲□▶▲□▶▲≡▶▲≡▶ ≡ の�?

### Linear regression



### ・ロト・日本・日本・日本・日本・今日・

### Linear regression



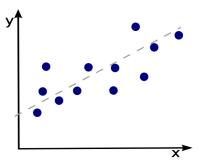
 $h(x) = w_0 + w_1 x$ 

### <ロ> <局> <目> <目> <日> <日> <の<</p>

Multivariate

Logistic regression

### Linear regression



$$h(x) = w_0 + w_1 x$$

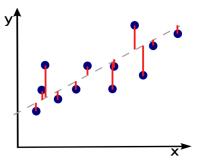
How to choose the parameter  $w_0$  and  $w_1$ ?

 ・ ◆ □ ▶ ◆ □ ▶ ◆ □ ▶ ↓ □ ▶ ◆ □ ▶ ◆ □ ♥ ○ ○
 Machine Learning and Pervasive Computing

Multivariate

Logistic regression

## Linear regression



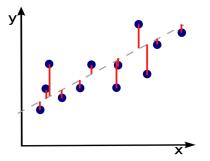
$$h(x) = w_0 + w_1 x$$

Cost function to estimate the quality of the current solution (Gradient descent).

Machine Learning and Pervasive Computing

3

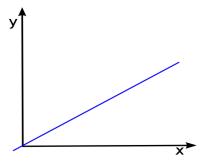
イロト イポト イヨト イヨト



$$h(x) = w_0 + w_1 x$$
  
minimize  $E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^n (h(x_i) - y_i)^2$ 

・ロ・・団・・ヨ・・ヨ・ ヨーのへで

### Gradient descent cost function - intuition



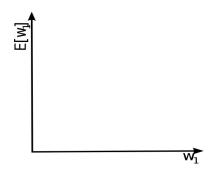
$$h(x) = w_0 + w_1 x$$
  
minimize  $E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^n (h(x_i) - y_i)^2$ 

For fixed  $w_1$  this is a function of x(additive constant  $w_0$  ignored in this figure)

Machine Learning and Pervasive Computing

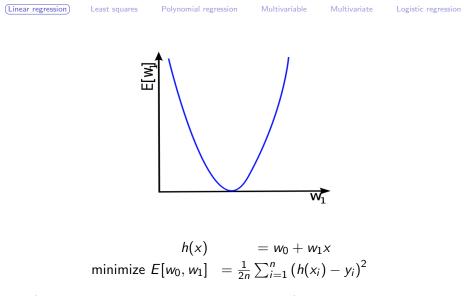
э

### Gradient descent cost function - intuition



$$\begin{array}{ll} h(x) &= w_0 + w_1 x \\ \text{minimize } E[w_0, w_1] &= \frac{1}{2n} \sum_{i=1}^n \left( h(x_i) - y_i \right)^2 \end{array}$$

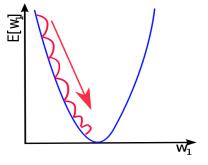
#### 



(additive constant  $w_0$  ignored in this figure)

◆□ → ◆□ → ◆ ■ → ◆ ■ → ● ■ 一 の Q ○ Machine Learning and Pervasive Computing

### Gradient descent cost function - Gradient descent



minimize 
$$E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^n (h(x_i) - y_i)^2$$
  
E.g.: $w_1 = w_1 - \delta \cdot \frac{\partial}{\partial w_1} E[w_0, w_1]$ 

Iterative approximation of  $w_1$ 

Machine Learning and Pervasive Computing

<ロ> <回> <回> <回> < 回> < 回> < 三</p>

Linear regression

Least squares

Polynomial regression

Multivariable

Multivariate

Logistic regression

### Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

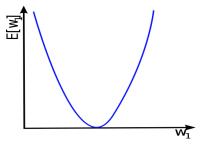
Logistic regression

・ ロ ト ・ 回 ト ・ 国 ト ・ 国 ・ ク ۹ ()

Linear regression

Logistic regression

### Least squares estimation



Given an error function

$$E[w_0, w_1] = \sum_{i=1}^n (y_i - (w_0 + w_1 x))^2$$

we can minimize the error by requiring

$$\frac{\partial E}{\partial w_0} = 0, \frac{\partial E}{\partial w_1} = 0$$

Machine Learning and Pervasive Computing

(ロ) (四) (E) (E) (E)

Linear regression

Multivariate

Logistic regression

### Least squares estimation

Given an error function

$$E[w_0, w_1] = \sum_{i=1}^n (y_i - (w_0 + w_1 x))^2$$

we can minimize the error by requiring

$$\frac{\partial E}{\partial w_0} = 0, \frac{\partial E}{\partial w_1} = 0$$

Differentiation yields

$$\frac{\partial E}{\partial w_0} = \sum_{i=1}^n 2(y_i - (w_1 x_i + w_0)) \cdot 1$$
$$\frac{\partial E}{\partial w_1} = \sum_{i=1}^n 2(y_i - (w_1 x_i + w_0)) \cdot (-x_i)$$

◆ロ → < 団 → < 豆 → < 豆 → < 豆 → < 豆 の Q (~ Machine Learning and Pervasive Computing)

### Least squares estimation

$$\frac{\partial E}{\partial w_0} = \sum_{i=1}^n 2(y_i - (w_1 x_i + w_0)) \cdot 1$$
$$\frac{\partial E}{\partial w_1} = \sum_{i=1}^n 2(y_i - (w_1 x_i + w_0)) \cdot (-x_i)$$

Setting

$$\frac{\partial E}{\partial w_0} = \frac{\partial E}{\partial w_1} = 0$$

will lead to

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) \cdot x_i = 0$$
$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) = 0$$

Logistic regression

Least squares estimation

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) \cdot x_i = 0$$
$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) = 0$$

rewrite as

$$\left(\sum_{i=1}^{n} x_i^2\right) w_1 + \left(\sum_{i=1}^{n} x_i\right) w_0 = \sum_{i=1}^{n} x_i y_i$$
$$\left(\sum_{i=1}^{n} x_i\right) w_1 + \left(\sum_{i=1}^{n} 1\right) w_0 = \sum_{i=1}^{n} y_i$$

 ・ ◆ □ ▶ ◆ □ ▶ ◆ □ ▶ ↓ □ ▶ ◆ □ ▶ ◆ □ ♥ ○ ○
 Machine Learning and Pervasive Computing

### Least squares estimation

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 \end{pmatrix} w_1 + \begin{pmatrix} \sum_{i=1}^{n} x_i \end{pmatrix} w_0 = \sum_{i=1}^{n} x_i y_i$$
$$\begin{pmatrix} \sum_{i=1}^{n} x_i \end{pmatrix} w_1 + \begin{pmatrix} \sum_{i=1}^{n} 1 \end{pmatrix} w_0 = \sum_{i=1}^{n} y_i$$

Consequently, values of  $w_0$  and  $w_1$  that minimize the error satisfy

$$\left(\begin{array}{cc}\sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i\\\sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1\end{array}\right) \left(\begin{array}{c} w_1\\w_0\end{array}\right) = \left(\begin{array}{c}\sum_{i=1}^{n} x_i y_i\\\sum_{i=1}^{n} y_i\end{array}\right)$$

Machine Learning and Pervasive Computing

(ロ) (四) (E) (E) (E)

### Least squares estimation

$$\left(\begin{array}{cc}\sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i\\\sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1\end{array}\right) \left(\begin{array}{c} w_1\\w_0\end{array}\right) = \left(\begin{array}{c}\sum_{i=1}^{n} x_i y_i\\\sum_{i=1}^{n} y_i\end{array}\right)$$

for an invertible matrix this implies

$$\begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n 1 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix}$$

By solving this linear equation system, optimal values of  $w_0$  and  $w_1$  can be determined.

#### ◆ロ → ◆ 部 → ◆ 書 → ▲ 書 → ● ● の へ ? Machine Learning and Pervasive Computing

### Least squares estimation

$$\left(\begin{array}{cc}\sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i\\\sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1\end{array}\right) \left(\begin{array}{c} w_1\\w_0\end{array}\right) = \left(\begin{array}{c}\sum_{i=1}^{n} x_i y_i\\\sum_{i=1}^{n} y_i\end{array}\right)$$

for an invertible matrix this implies

$$\left(\begin{array}{c}w_1\\w_0\end{array}\right) = \left(\begin{array}{cc}\sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i\\\sum_{i=1}^n x_i & \sum_{i=1}^n 1\end{array}\right)^{-1} \left(\begin{array}{c}\sum_{i=1}^n x_i y_i\\\sum_{i=1}^n y_i\end{array}\right)$$

By solving this linear equation system, optimal values of  $w_0$  and  $w_1$  can be determined.

However, for least squares to be applicable, it is necessary that the matrix is invertible.

Machine Learning and Pervasive Computing

3

イロン 不同 とくほう イヨン

Logistic regression

## Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression

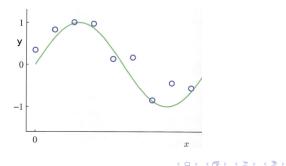


## Polynomial regression (Polynomial curve fitting)

### Example

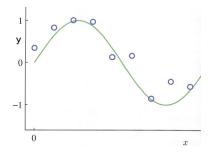
A curve shall be approximated by a machine learning approach

Sample points are created for the function  $sin(2\pi x) + N$  where N is a random noise value



We will try to fit the data points into a polynomial function:

$$h(x, \vec{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$



### - \* ロ \* \* 個 \* \* 注 \* \* 注 \* うへぐ

We will try to fit the data points into a polynomial function:

$$h(x, \overrightarrow{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$

This can be obtained by minimising an error function that measures the misfit between  $h(x, \vec{w})$  and the training data set:

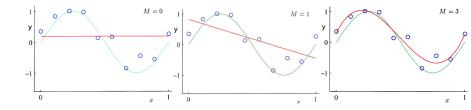
$$E(\overrightarrow{w}) = \frac{1}{2} \sum_{i=1}^{N} \left[ h(x_i, \overrightarrow{w}) - y_i \right]^2$$

 $E(\vec{w})$  is non-negative and zero if and only if all points are covered by the function

Machine Learning and Pervasive Computing

One problem is the right choice of the dimension  $\boldsymbol{M}$ 

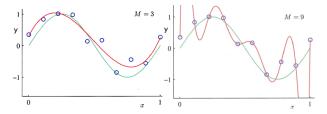
When M is too small, the approximation accuracy might be bad



### ・ロト・西ト・ヨト・ヨー 今々ぐ

However, when M becomes too big, the resulting polynomial will cross all points exactly

When M reaches the count of samples in the training data set, it is always possible to create a polynomial of order M that contains all values in the data set exactly.



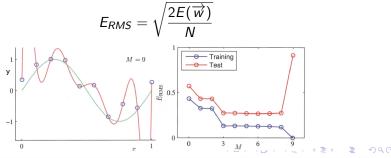
### ・

This event is called overfitting

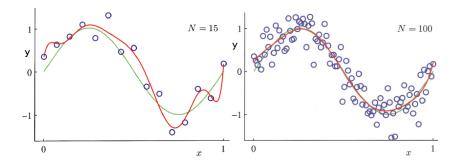
The polynomial is now trained too well to the training data

It will therefore perform badly on another sample of test data for the same phenomenon

We visualise it by the Root of the Mean Square (RMS) of  $E(\vec{w})$ 



With increasing number of data points, the problem of overfitting becomes less severe for a given value of M



### ▲□▶ ▲□▶ ▲目▶ ▲目▶ 目 のへで

One solution to cope with overfitting is regularisation

A penalty term is added to the error function

This term discourages the coefficients of  $\overrightarrow{w}$  from reaching large values

$$\overline{E}(\overrightarrow{w}) = \frac{1}{2} \sum_{i=1}^{N} \left[ h(x_i, \overrightarrow{w}) - y_i \right]^2 + \frac{\lambda}{2} ||\overrightarrow{w}||^2$$

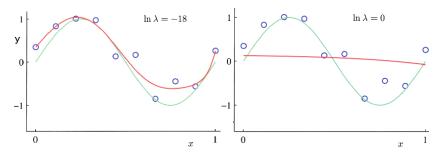
with

$$||\overrightarrow{w}||^2 = \overrightarrow{w}^T \overrightarrow{w} = w_0^2 + w_1^2 + \dots + w_M^2$$

Machine Learning and Pervasive Computing

イロト 不得 トイヨト イヨト 二日

Depending on the value of  $\lambda$ , overfitting is controlled



$$\overline{E}(\overrightarrow{w}) = \frac{1}{2} \sum_{i=1}^{N} \left[ h(x_i, \overrightarrow{w}) - y_i \right]^2 + \frac{\lambda}{2} ||\overrightarrow{w}||^2$$

・ロト・白 ・ ・ ヨ・ ・ ヨ・ うへぐ

Multivariate

Logistic regression

### Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression





Multivariate

Logistic regression

### Multivariable linear regression

In multivariable linear regression problems we assume that multiple regression variables (features) apply.



### Multivariable linear regression

In multivariable linear regression problems we assume that multiple regression variables (features) apply.

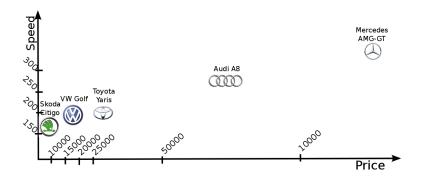
$$h(x_{j1}, \dots, x_{jm}) = \sum_{i=0}^{m} w_i x_{ji}$$
  
minimize  $E[W] = \frac{1}{2n} \sum_{j=1}^{n} (h(x_{j1}, \dots, x_{jm}) - y_j)^2$ 

#### 

Multivariate

Logistic regression

## Multivariable linear regression

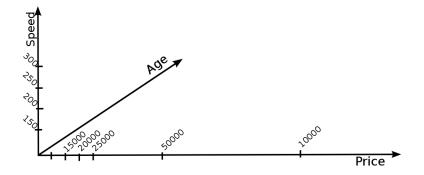


### - ▲ ロ ト ▲ 国 ト ▲ 国 ト シ 国 - つくぐ

Multivariate

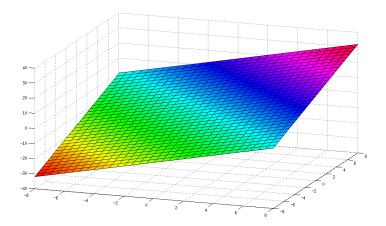
Logistic regression

## Multivariable linear regression



### ・ロト・日本・日本・日本・日本・クへの

## Multivariable linear regression



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 三臣 - のへで

$$h(x_{j1}, \dots, x_{jm}) = \sum_{i=0}^{m} w_i x_{ji}$$
  
minimize  $E[W] = \frac{1}{2n} \sum_{j=1}^{n} (h(x_{j1}, \dots, x_{jm}) - y_j)^2$ 

#### ▲□▶ ▲□▶ ▲目▶ ▲目▶ 目 のへで

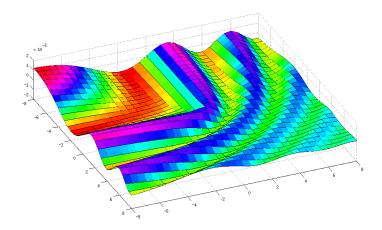
$$h(x_{j1},...,x_{jm}) = \sum_{i=0}^{m} w_i x_{ji}$$
  
minimize  $E[W] = \frac{1}{2n} \sum_{j=1}^{n} (h(x_{j1},...,x_{jm}) - y_j)^2$ 

$$\mathsf{E.g.}: w_i = w_i - \delta \cdot \frac{\partial}{\partial w_i} E[W]$$

 $w_i$  are optimised together over several iterations

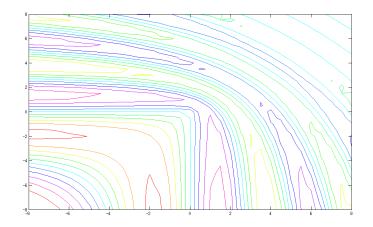
◆□ → < □ → < □ → < □ → < □ → < □ → □ → ○ Q ↔ Machine Learning and Pervasive Computing

## Local optima



#### 

## Local optima – contour plot



▲□▶ ▲圖▶ ▲蓋▶ ▲蓋▶ ― 蓋 … 釣んぐ

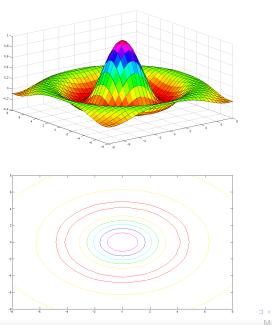
Linear regression

Polynomial regression

(Multivariable)

Multivariate

Logistic regression



コン・日本・エー・ モー うへの

Multivariate

Logistic regression

## Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression

- ▲ ロ > ▲ 国 > ▲ 国 > ▲ 国 > 今 Q @

(Multivariate)

## Multivariate linear regression

Multivariate linear regression describes a regression problem with multiple classes.

### Example e.g. from accelerometer data

Activities walking, standing, climbin/descending stairs, ...

Sentiment emotional states

Transportation mode office, riding tram, driving ...

Location Home, office, ...

### ふして 川 ふかく 山マ ふして

## Multivariate linear regression

Regression model is extended to multiple responses with respect to one class:  $Y_i = y_{i1}, \ldots, y_{il}$ 

$$y_{j1} = w_{01} + w_{11}x_{j1} + \dots + w_{n1}x_{jn}$$
  

$$y_{j2} = w_{02} + w_{12}x_{j1} + \dots + w_{n2}x_{jn}$$
  

$$\vdots \qquad \vdots$$
  

$$y_{jl} = w_{0l} + w_{1l}x_{j1} + \dots + w_{nl}x_{jn}$$

## Multivariate linear regression

Regression model is extended to multiple responses with respect to one class:  $Y_j = y_{j1}, \ldots, y_{jl}$ 

$$y_{j1} = w_{01} + w_{11}x_{j1} + \dots + w_{n1}x_{jn}$$
  

$$y_{j2} = w_{02} + w_{12}x_{j1} + \dots + w_{n2}x_{jn}$$
  

$$\vdots \qquad \vdots$$
  

$$y_{jl} = w_{0l} + w_{1l}x_{j1} + \dots + w_{nl}x_{jn}$$

It is then possible to estimate the regression coefficients associated with  $y_{ii}$  using only the *i*-th row of the matrix.

Machine Learning and Pervasive Computing

イロト 不得 とくほと くほとう ほ

## Multivariate linear regression

It is then possible to estimate the regression coefficients associated with  $y_{ii}$  using only the *i*-th row of the matrix.

$$W_i = \left(X^T X\right)^{-1} X^T Y_{(i)}$$

and collecting all univariate estimates into a matrix

$$W = \left(X^{\mathsf{T}}X\right)^{-1}X^{\mathsf{T}}Y$$

 $Y_{(i)}$  is the vector of n measurements of the i-th variable  $X^T$  denotes the transpose of X and  $X^{-1}$  its inverse

Machine Learning and Pervasive Computing

イロト 不得 トイヨト イヨト 二日



### Multivariate linear regression

The least squares estimator for W minimizes the sums of squares elements on the diagonal of the residual sum of squares and crossproducts matrix  $(Y - XW)^T (Y - XW)$ 

#### ▲ロト ▲ 団 ト ▲ 置 ト ▲ 置 か Q (\*) Machine Learning and Pervasive Computing

Multivariate

Logistic regression

## Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression



Multivariate

Logistic regression

## Logistic regression

Nominal classes

Classes might be nominal in real-world problems



## Logistic regression

Nominal classes

Classes might be nominal in real-world problems Weather Sunny, rainy Medical positive diagnosis, negative diagnosis Localisation indoor, outdoor

・ロト・日本・モート ヨー うえの

## Logistic regression

Nominal classes

Classes might be nominal in real-world problems Weather Sunny, rainy Medical positive diagnosis, negative diagnosis Localisation indoor, outdoor

In such case, classification is binary:  $y \in \{0, 1\}$ 

Machine Learning and Pervasive Computing

## Logistic regression

Nominal classes

Classes might be nominal in real-world problems Weather Sunny, rainy Medical positive diagnosis, negative diagnosis Localisation indoor, outdoor

In such case, classification is binary:  $y \in \{0, 1\}$ 

Linear regression: h(x) can be smaller than 0 or greater than 1

#### イロト イ団ト イヨト イヨト ヨークへで Machine Learning and Pervasive Computing

## Logistic regression

Nominal classes

Classes might be nominal in real-world problems Weather Sunny, rainy Medical positive diagnosis, negative diagnosis Localisation indoor, outdoor

In such case, classification is binary:  $y \in \{0, 1\}$ 

Linear regression: h(x) can be smaller than 0 or greater than 1 Logistic regression:  $0 \le h(x) \le 1$ 

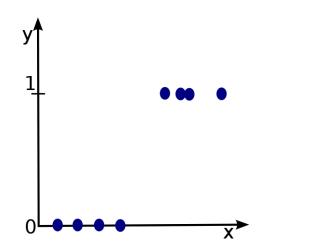
#### ◆ロ → ◆ 部 → ◆ 書 → ▲ 書 → ● ● の へ ? Machine Learning and Pervasive Computing

Multivariate

Logistic regression

## Logistic regression

Nominal classes



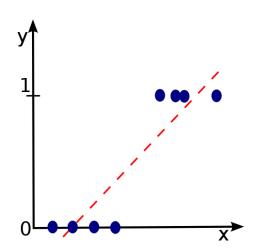
#### ▲□▶▲@▶▲≧▶▲≧▶ 差 のへで

Multivariate

Logistic regression

## Logistic regression

Nominal classes



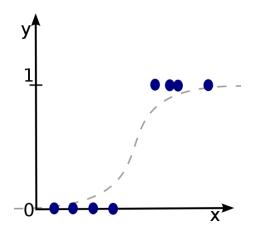
<ロ> < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Multivariate

Logistic regression

## Logistic regression

Cost function

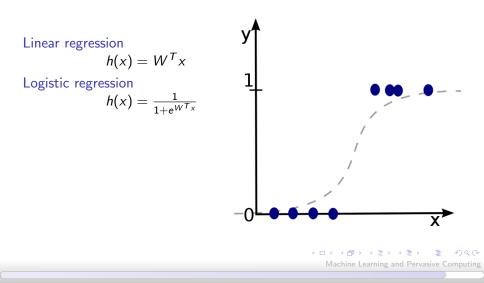


#### ・ロト・日本・日本・日本・日本・今日・

Multivariate

# Logistic regression

Cost function



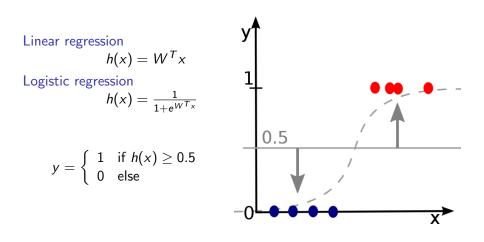
・ロト ・回ト ・ヨト ・ヨト

Machine Learning and Pervasive Computing

3

# Logistic regression

Cost function



# Logistic regression

Cost function

$$y = \begin{cases} 1 & \text{if } h(x) \ge 0.5 \\ 0 & \text{else} \end{cases}$$
$$E[h(x), y] = \begin{cases} -\log(h(x)) & \text{if } y = 1 \\ -\log(1 - h(x)) & \text{else} \end{cases}$$

◆□ → < □ → < □ → < □ → < □ → < □ → □ → ○ Q ↔ Machine Learning and Pervasive Computing

Multivariate

Logistic regression

## Outline

Linear regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression



Polynomial regression

Multivariable

Multivariate

Logistic regression

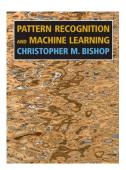
# **Questions?**

### Stephan Sigg stephan.sigg@cs.uni-goettingen.de

- \* ロ \* \* @ \* \* 国 \* \* 国 \* のへで

### Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.





Machine Learning and Pervasive Computing

3

イロト イポト イヨト イヨト