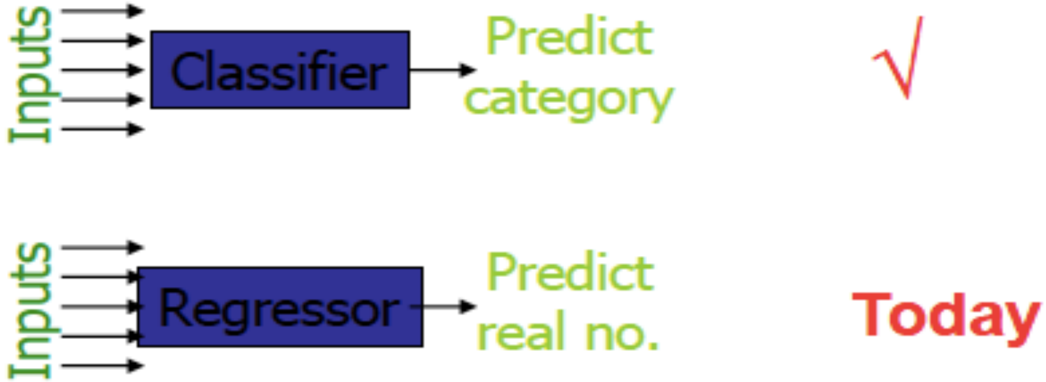
Machine Learning

Prof. Barbara Caputo

Dip. Ingegneria Informatica, Automatica e Gestionale, Roma

Sapienza Università di Roma

Where we are



Inputs
Classifier
Predict
category
√
Inputs
Regressor
Predict
real no.
Today

Choosing a restaurant

- In everyday life we need to make decisions by taking into account lots of factors
- The question is what weight we put on each of these factors (how important are they with respect to the others).

Choosing a restaurant

- In everyday life we need to make decisions by taking into account lots of factors
- The question is what weight we put on each of these factors (how important are they with respect to the others).
- Assume we would like to build a recommender system for ***ranking*** potential restaurants based on an individuals' preferences

Reviews (out of 5 stars)	$	Distance	Cuisine (out of 10)
4	30	21	7
2	15	12	8
5	27	53	9
3	20	5	6

Choosing a restaurant

- In everyday life we need to make decisions by taking into account lots of factors
- The question is what weight we put on each of these factors (how important are they with respect to the others).
- Assume we would like to build a recommender system for ***ranking*** potential restaurants based on an individuals' preferences
- If we have many observations we may be able to recover the weights

Reviews (out of 5 stars)	$	Distance	Cuisine (out of 10)
4	30	21	7
2	15	12	8
5	27	53	9
3	20	5	6

?

Linear regression

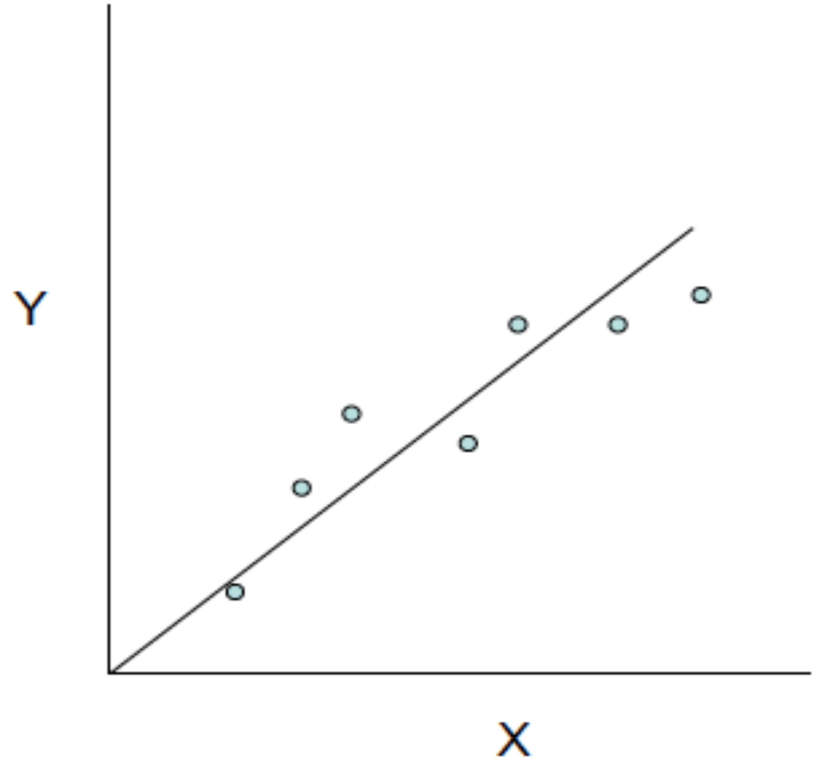
Linear regression

- Given an input x we would like to compute an output y
- For example:

  - Predict height from age

  - Predict Google's price from Yahoo's price

  - Predict distance from wall using sensor readings



Note that now Y can be **continuous**

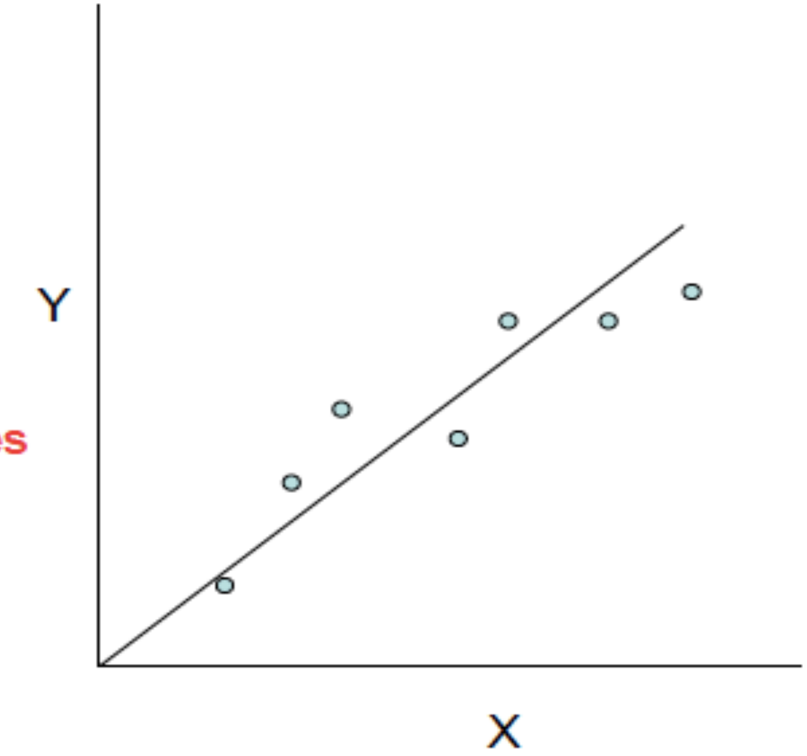
Linear regression

- Given an input x we would like to compute an output y
- In linear regression we assume that y and x are related with the following equation:

What we are trying to predict

Observed values

$$y = wx+\varepsilon$$



Linear regression

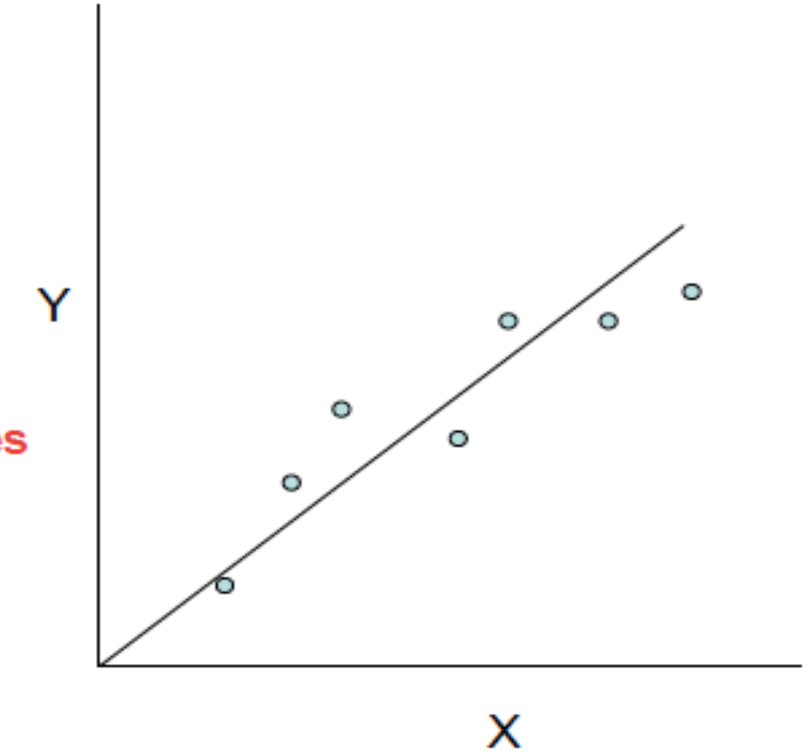
- Given an input x we would like to compute an output y
- In linear regression we assume that y and x are related with the following equation:

What we are trying to predict

Observed values

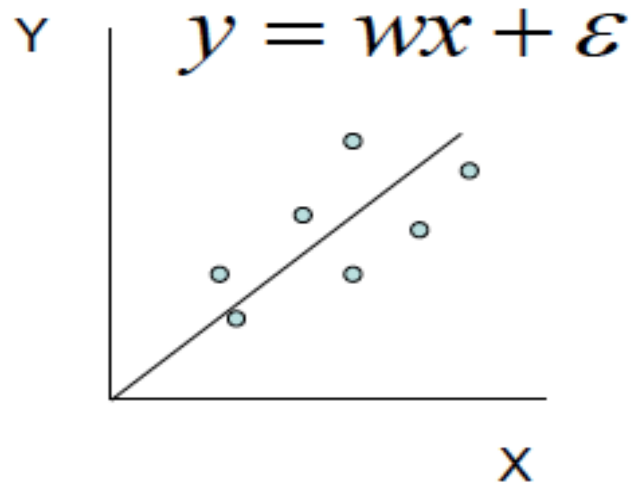
$$y = wx+\varepsilon$$

where w is a parameter and ε represents measurement or other noise



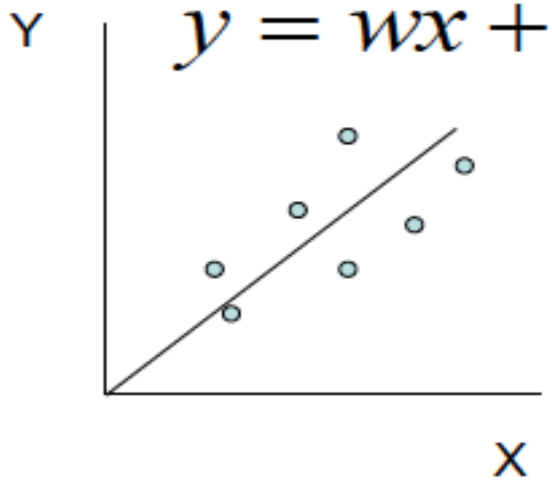
Linear regression

• Our goal is to estimate *w* from a training data of $<x_i,y_i>$ pairs



Linear regression

$$y = wx + \varepsilon$$



- Our goal is to estimate *w* from a training data of $<x_i,y_i>$ pairs
- One way to find such relationship is to minimize the a least squares error:

$$\arg\min_w \sum_i (y_i - wx_i)^2$$

- Several other approaches can be used as well
- So why least squares?
  - minimizes squared distance between measurements and predicted line
  - has a nice probabilistic interpretation
  - easy to compute

If the noise is Gaussian with mean 0 then least squares is also the maximum likelihood estimate of w

Solving linear regression using least squares minimization

- We just take the derivative w.r.t. to w and set to 0:

Solving linear regression using least squares minimization

- We just take the derivative w.r.t. to w and set to 0:

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

Solving linear regression using least squares minimization

- We just take the derivative w.r.t. to w and set to 0:

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

Solving linear regression using least squares minimization

- We just take the derivative w.r.t. to w and set to 0:

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

$$\sum_i x_i y_i = \sum_i w x_i^2 \Rightarrow$$

Solving linear regression using least squares minimization

- We just take the derivative w.r.t. to w and set to 0:

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

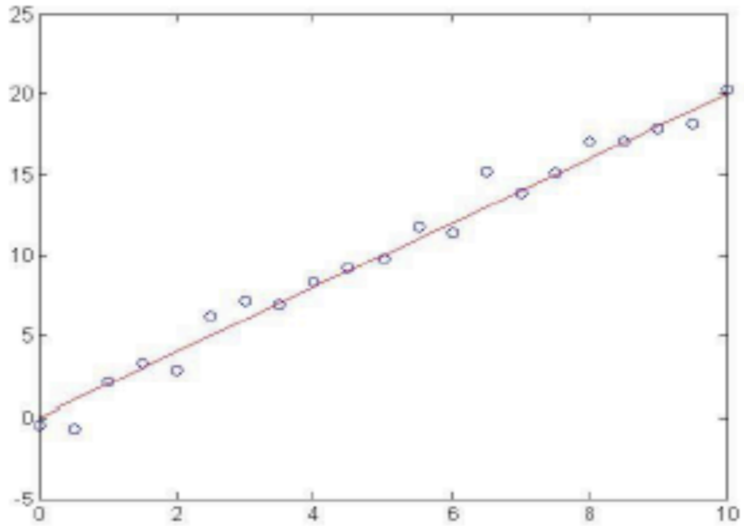
$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

$$\sum_i x_i y_i = \sum_i wx_i^2 \Rightarrow$$

$$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

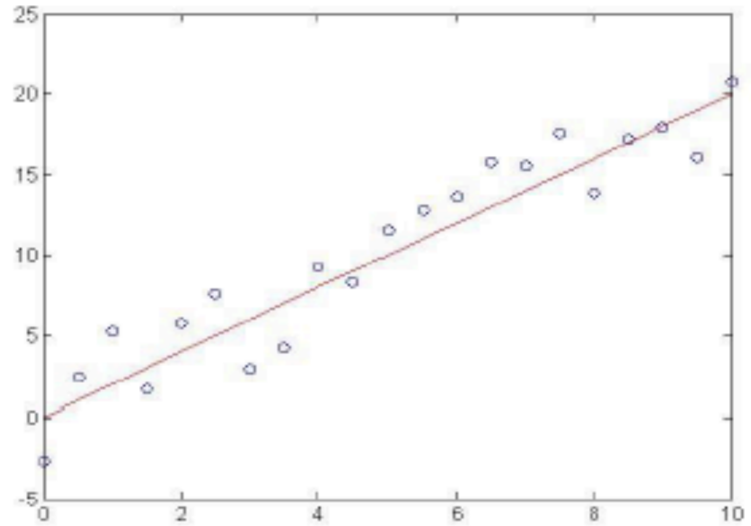
Regression example

- Generated: w=2
- Recovered: w=2.03
- Noise: std=1



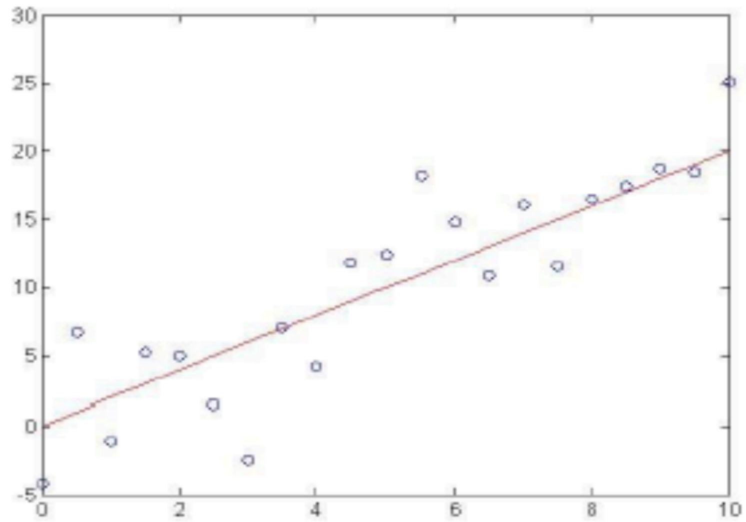
Regression example

- Generated: w=2
- Recovered: w=2.05
- Noise: std=2



Regression example

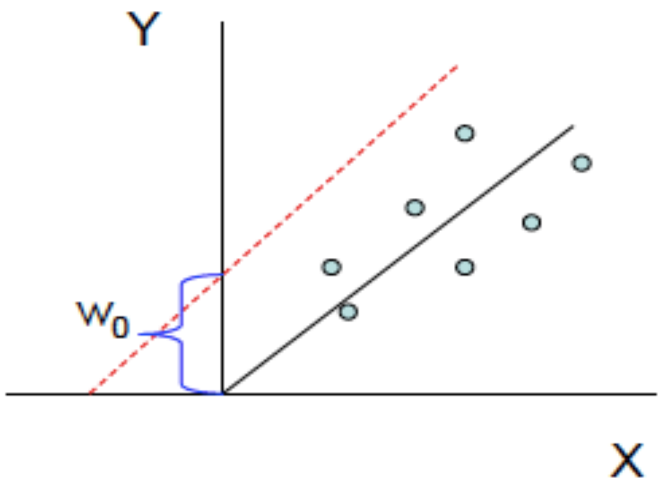
- Generated: w=2
- Recovered: w=2.08
- Noise: std=4



Bias term

- So far we assumed that the line passes through the origin
- What if the line does not?
- No problem, simply change the model to

$$y = w_0 + w_1 x + \varepsilon$$



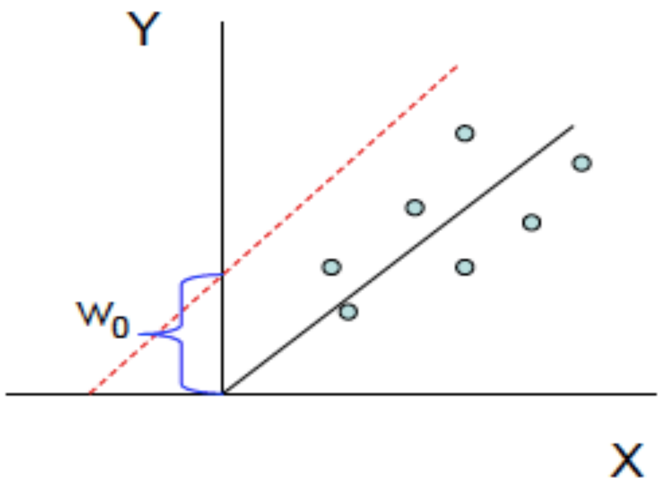
Bias term

- So far we assumed that the line passes through the origin
- What if the line does not?
- No problem, simply change the model to

$$y = w_0 + w_1 x + \varepsilon$$

- Can use least squares to determine w_0 , w_1

$$w_0 = \frac{\sum_i y_i - w_1 x_i}{n}$$

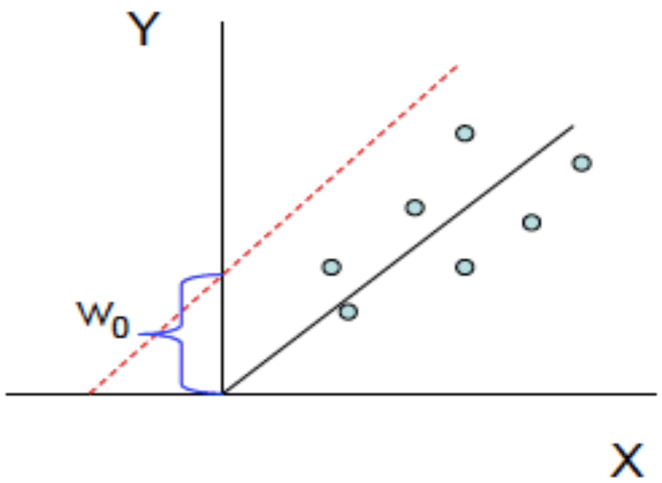


Bias term

- So far we assumed that the line passes through the origin
- What if the line does not?
- No problem, simply change the model to

$$y = w_0 + w_1 x + \varepsilon$$

- Can use least squares to determine w_0 , w_1



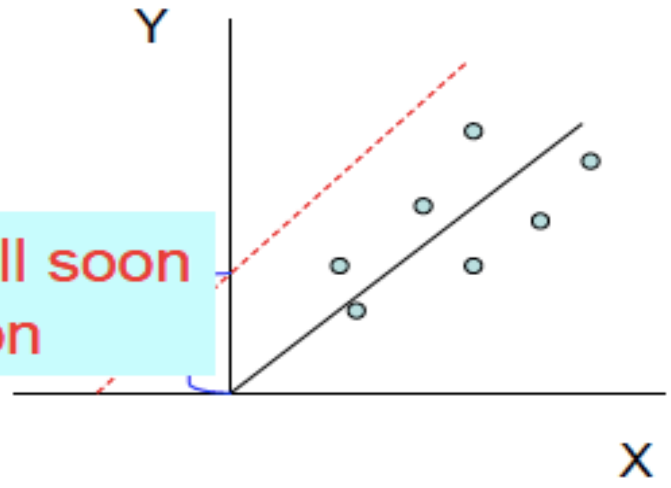
$$w_0 = \frac{\sum_i y_i - w_1 x_i}{n} \qquad w_1 = \frac{\sum_i x_i (y_i - w_0)}{\sum_i x_i^2}$$

Bias term

- So far we assumed that the line passes through the origin
- What if the line does not?
- No problem, simply change the model to

y = w

Just a second, we will soon give a simpler solution



- Can use least squares to determine w_0 , w_1

$$w_0 = \frac{\sum_i y_i - w_1 x_i}{n}$$

$$w_1 = \frac{\sum_i x_i (y_i - w_0)}{\sum_i x_i^2}$$

Multivariate regression

- What if we have several inputs?
  - Stock prices for Yahoo, Microsoft and Ebay for the Google prediction task

Multivariate regression

- What if we have several inputs?
  - Stock prices for Yahoo, Microsoft and Ebay for the Google prediction task
- This becomes a multivariate linear regression problem

Multivariate regression

- What if we have several inputs?
  - Stock prices for Yahoo, Microsoft and Ebay for the Google prediction task
- This becomes a multivariate linear regression problem
- Again, its easy to model:

$$y = w_0 + w_1 x_1 + \ldots + w_k x_k + \varepsilon$$

Google's stock price

Microsoft's stock price

Yahoo's stock price

Multivariate regression

- What if we have several inputs?

  - Stock prices for Yahoo, Microsoft and Ebay for the Goo

- This be   n problem
- Again, its easy to model:

$$y = w_0 + w_1x_1 + \ldots + w_kx_k + \varepsilon$$

Not all functions can be approximated using the input values directly

$$y=10+3x_1^2-2x_2^2+\varepsilon$$

In some cases we would like to use polynomial or other terms based on the input data, are these still linear regression problems?

$$y=10+3x_1^2-2x_2^2+\varepsilon$$

In some cases we would like to use polynomial or other terms based on the input data, are these still linear regression problems?

Yes. As long as the coefficients are linear the equation is still a linear regression problem!

Five mins break!

Non-Linear basis function

- So far we only used the observed values
- However, linear regression can be applied in the same way to functions of these values

Non-Linear basis function

- So far we only used the observed values
- However, linear regression can be applied in the same way to functions of these values
- As long as these functions can be directly computed from the observed values the parameters are still linear in the data and the problem remains a linear regression problem

Non-Linear basis function

- So far we only used the observed values
- However, linear regression can be applied in the same way to functions of these values
- As long as these functions can be directly computed from the observed values the parameters are still linear in the data and the problem remains a linear regression problem

$$y = w_0 + w_1 x_1^2 + \ldots + w_k x_k^2 + \varepsilon$$

Non-Linear basis function

- What type of functions can we use?
- A few common examples:

- Polynomial: $\phi_j(x) = x^j$ for $j=0 \ldots n$

Non-Linear basis function

- What type of functions can we use?
- A few common examples:

- Polynomial: $\phi_j(x) = x^j$ for j=0 … n

- Gaussian: $\phi_j(x) = \dfrac{(x-\mu_j)}{2\sigma_j^2}$

Non-Linear basis function

- What type of functions can we use?
- A few common examples:

- Polynomial: $\phi_j(x) = x^j$ for j=0 … n

- Gaussian: $$\phi_j(x) = \frac{(x-\mu_j)}{2\sigma_j^2}$$

- Sigmoid: $$\phi_j(x) = \frac{1}{1+\exp(-s_j x)}$$

Any function of the input values can be used. The solution for the parameters of the regression remains the same.

General linear regression problem

- Using our new notations for the basis function linear regression can be written as

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

- Where $\phi_j(x)$ can be either x_j for multivariate regression or one of the non linear basis we defined

General linear regression problem

- Using our new notations for the basis function linear regression can be written as

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

- Where $\phi_j(x)$ can be either x_j for multivariate regression or one of the non linear basis we defined
- Once again we can use ‘least squares’ to find the optimal solution.

LMS for the general linear regression problem

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$$

LMS for the general linear regression problem

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i \left(y^i - \sum_j w_j \phi_j(x^i)\right)^2$$

w – vector of dimension k+1

$\phi(x^i)$ – vector of dimension k+1

y^i – a scaler

Moving to vector notations we get:

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^T \phi(x^i))^2$$

LMS for the general linear regression problem

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$$

w – vector of dimension k+1

$\phi(x^i)$ – vector of dimension k+1

y^i – a scaler

Moving to vector notations we get:

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

LMS for the general linear regression problem

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i (y^i - \sum_j w_j \phi_j(x^i))^2$$

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

w – vector of dimension k+1

$\phi(x^i)$ – vector of dimension k+1

y^i – a scaler

Moving to vector notations we get:

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get $2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = 0 \Rightarrow$

LMS for the general linear regression problem

$$y = \sum_{j=0}^{n} w_j \phi_j(x)$$

Our goal is to minimize the following loss function:

$$J(\mathrm{w}) = \sum_i \Big(y^i - \sum_j w_j \phi_j(x^i)\Big)^2$$

w – vector of dimension k+1
$\phi(x^i)$ – vector of dimension k+1
y^i – a scaler

Moving to vector notations we get:

$$J(\mathrm{w}) = \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get $2 \sum_i (y^i - \mathrm{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = 0 \Rightarrow$

$$\sum_i y^i \phi(x^i)^{\mathrm{T}} = \mathrm{w}^{\mathrm{T}} \left[\sum_i \phi(x^i) \phi(x^i)^{\mathrm{T}} \right]$$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^T \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathbf{w}^T \phi(x^i))^2 = 2 \sum_i (y^i - \mathbf{w}^T \phi(x^i)) \phi(x^i)^T$$

Equating to 0 we get $\quad 2 \sum_i (y^i - \mathbf{w}^T \phi(x^i)) \phi(x^i)^T = 0 \Rightarrow$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^{\mathrm{T}}\phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w}\sum_i (y^i - \mathbf{w}^{\mathrm{T}}\phi(x^i))^2 = 2\sum_i (y^i - \mathbf{w}^{\mathrm{T}}\phi(x^i))\phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get

$$2\sum_i (y^i - \mathbf{w}^{\mathrm{T}}\phi(x^i))\phi(x^i)^{\mathrm{T}} = \mathbf{0} \Rightarrow$$

$$\sum_i y^i\phi(x^i)^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\left[\sum_i \phi(x^i)\phi(x^i)^{\mathrm{T}}\right]$$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get

$$2 \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = \mathbf{0} \Rightarrow$$

$$\sum_i y^i \phi(x^i)^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}} \left[\sum_i \phi(x^i) \phi(x^i)^{\mathrm{T}} \right]$$

Define:

$$\Phi = \begin{pmatrix} \phi_0(x^1) & \phi_1(x^1) & \cdots & \phi_m(x^1) \\ \phi_0(x^2) & \phi_1(x^2) & \cdots & \phi_m(x^2) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_0(x^n) & \phi_1(x^n) & \cdots & \phi_m(x^n) \end{pmatrix}$$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2$$

We take the derivative w.r.t **w**

$$\frac{\partial}{\partial w} \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i))^2 = 2 \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}}$$

Equating to 0 we get

$$2 \sum_i (y^i - \mathbf{w}^{\mathrm{T}} \phi(x^i)) \phi(x^i)^{\mathrm{T}} = 0 \Rightarrow$$

$$\sum_i y^i \phi(x^i)^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}} \left[\sum_i \phi(x^i) \phi(x^i)^{\mathrm{T}} \right]$$

Define:

$$\Phi = \begin{pmatrix} \phi_0(x^1) & \phi_1(x^1) & \cdots & \phi_m(x^1) \\ \phi_0(x^2) & \phi_1(x^2) & \cdots & \phi_m(x^2) \\ \vdots & \vdots & \cdots & \vdots \\ \phi_0(x^n) & \phi_1(x^n) & \cdots & \phi_m(x^n) \end{pmatrix}$$

Then deriving w we get:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

LMS for general linear regression problem

$$J(\mathbf{w}) = \sum_i (y^i - \mathbf{w}^T \phi(x^i))^2$$

Deriving w we get: $\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$

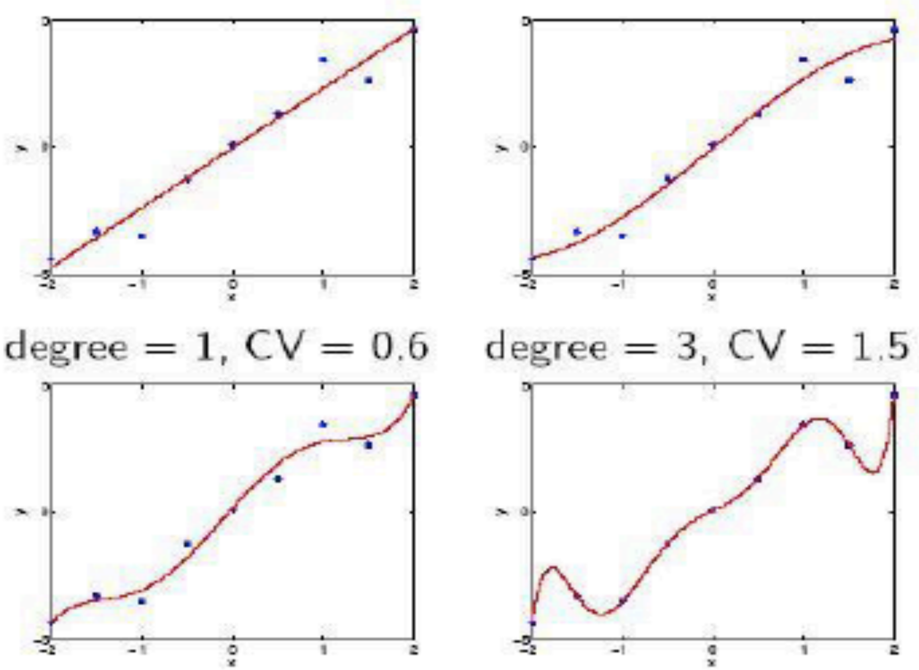
k+1 entries vector

n by k+1 matrix

n entries vector

This solution is
also known as
'psuedo inverse'

Example: Polynomial regression



degree = 1, CV = 0.6 degree = 3, CV = 1.5

degree = 5, CV = 6.0 degree = 7, CV = 15.6

A probabilistic interpretation

Our least squares minimization solution can also be motivated by a probabilistic in interpretation of the regression problem: $y = \mathbf{w}^{\mathrm{T}} \phi(x) + \varepsilon$

A probabilistic interpretation

Our least squares minimization solution can also be motivated by a probabilistic in interpretation of the regression problem: $y = \mathbf{w}^{\mathrm{T}} \phi(x) + \varepsilon$

The MLE for w in this model is the same as the solution we derived for least squares criteria:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

Other types of linear regression

- Linear regression is a useful model for many problems
- However, the parameters we learn for this model are **global**; they are the same regardless of the value of the input x
- Extension to linear regression adjust their parameters based on the region of the input we are dealing with

Five mins break!

Back to classification

1. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

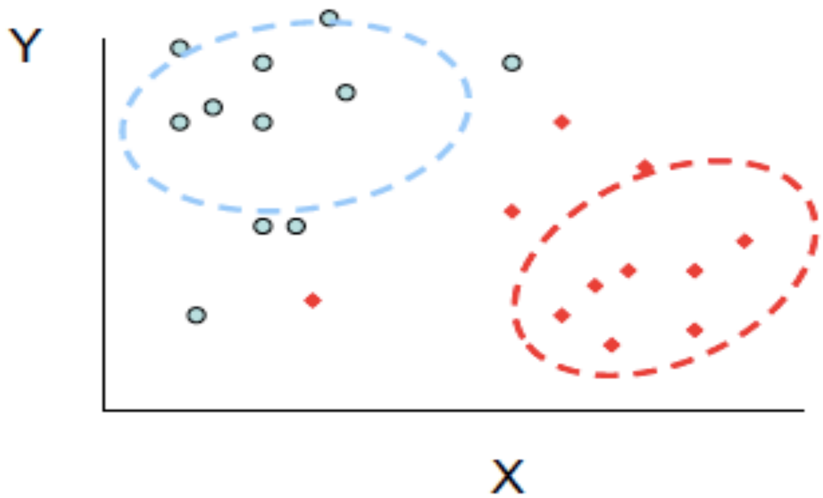
2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks

3. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., decision tree

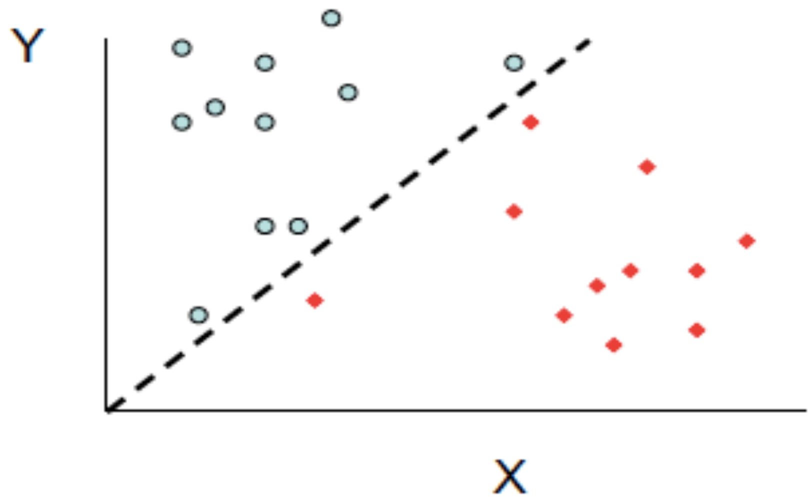
Generative vs. discriminative classifiers

- When using generative classifiers we relied on all points to learn the generative model
- When using discriminative classifiers we mainly care about the boundary

Generative model



Discriminative model



Regression for classification

- In some cases we can use linear regression for determining the appropriate boundary.
- However, since the output is usually binary or discrete there are more efficient regression methods

Regression for classification

- In some cases we can use linear regression for determining the appropriate boundary.
- However, since the output is usually binary or discrete there are more efficient regression methods
- Recall that for classification we are interested in the conditional probability $p(y \mid X ; \theta)$ where θ are the parameters of our model
- When using regression θ represents the values of our regression coefficients (w).

Regression for classification

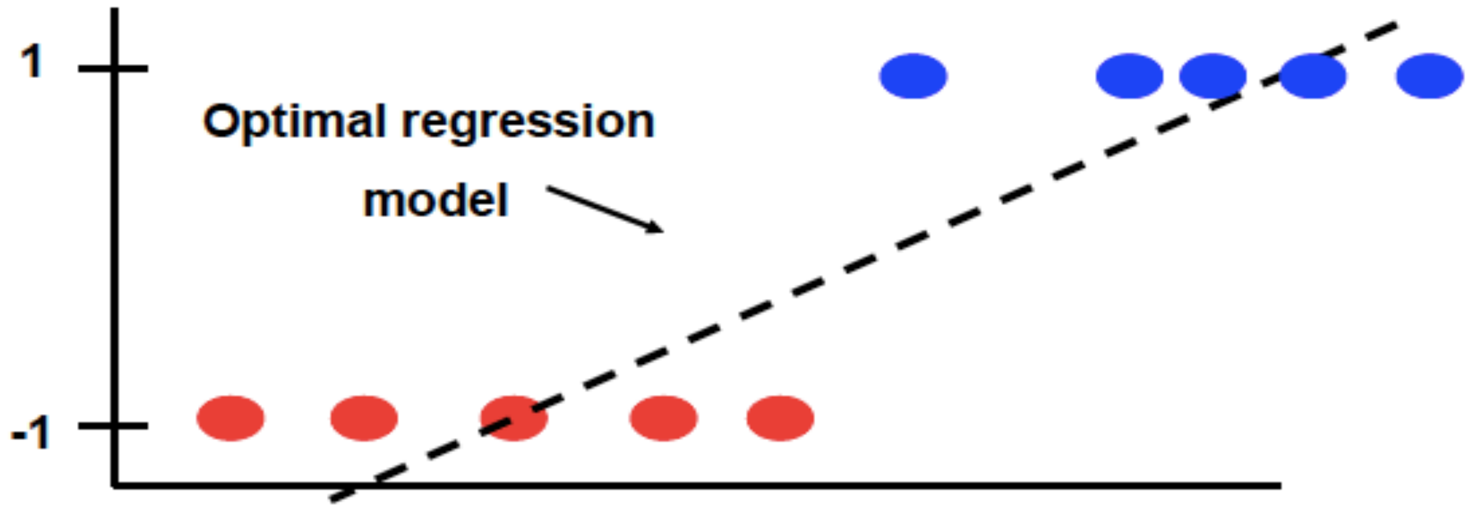
- Assume we would like to use linear regression to learn the parameters for $p(y \mid X ; \theta)$
- Problems?

Regression for classification

- Assume we would like to use linear regression to learn the parameters for $p(y \mid X ; \theta)$
- Problems?

$w^T X \geq 0 \Rightarrow$ classify as 1

$w^T X < 0 \Rightarrow$ classify as -1



The sigmoid function

$$p(y \mid X; \theta)$$

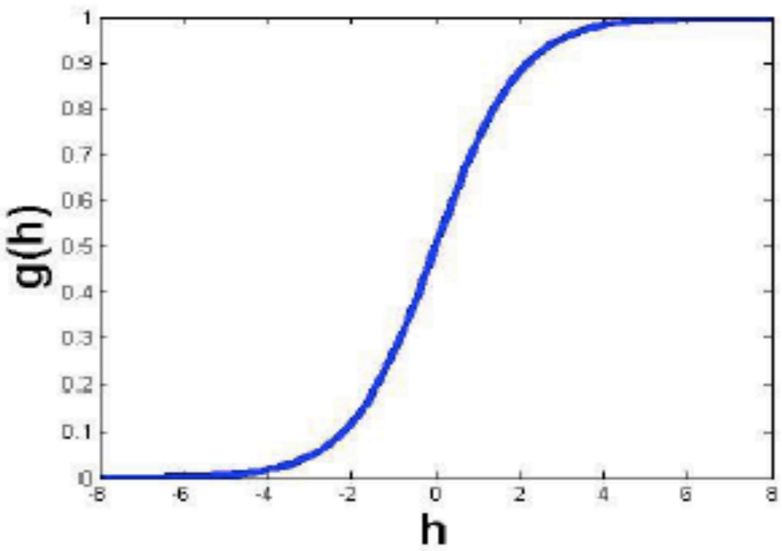
- To classify using regression models we replace the linear function with the sigmoid function:

The sigmoid function

$p(y \mid X; \theta)$

- To classify using regression models we replace the linear function with the sigmoid function:

Always between 0 and 1 $\longrightarrow$ $$g(h) = \frac{1}{1+e^{-h}}$$

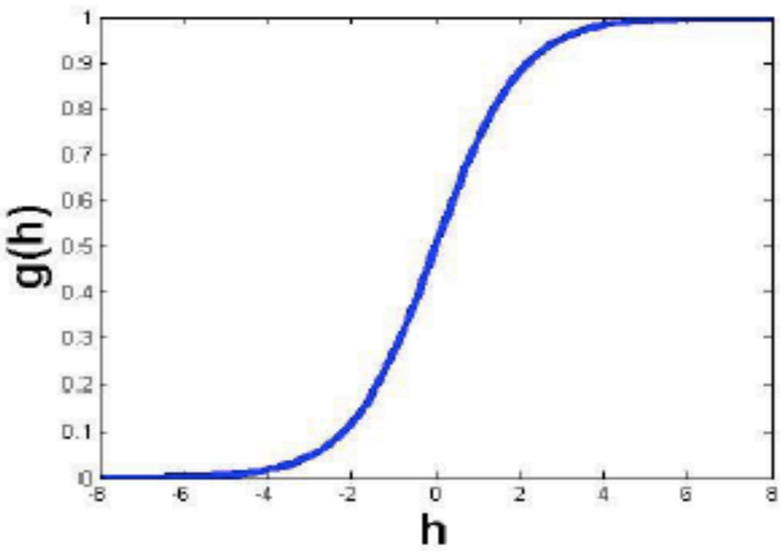


The sigmoid function

$$p(y \mid X;\theta)$$

- To classify using regression models we replace the linear function with the sigmoid function:

Always between 0 and 1 $\longrightarrow$ $$g(h) = \frac{1}{1+e^{-h}}$$



- Using the sigmoid we set (for binary classification problems)

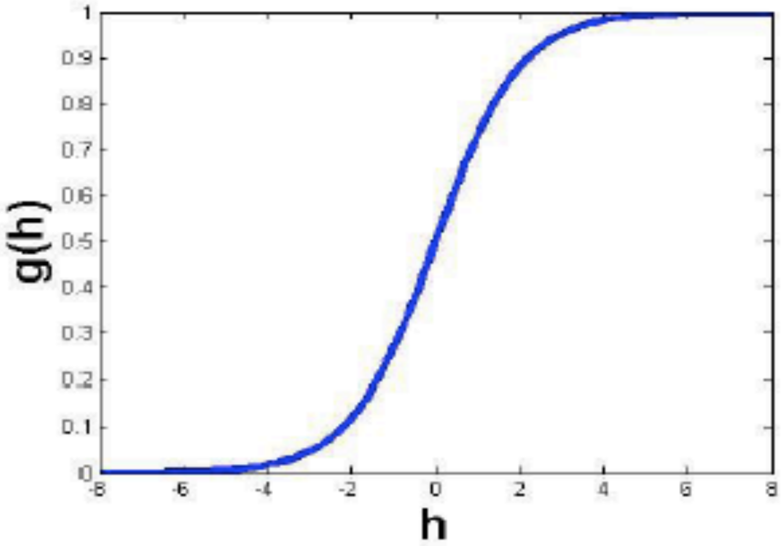
$$p(y=0 \mid X;\theta) = g(\mathrm{w}^{\mathrm{T}} X) = \frac{1}{1+e^{\mathrm{w}^{\mathrm{T}} X}}$$

The sigmoid function

$$p(y \mid X; \theta)$$

- To classify using regression models we replace the linear function with the sigmoid function:

Always between 0 and 1 $\longrightarrow$ $$g(h) = \frac{1}{1+e^{-h}}$$



- Using the sigmoid we set (for binary classification problems)

$$p(y=0 \mid X; \theta) = g(\mathrm{w}^{\mathrm{T}} X) = \frac{1}{1+e^{\mathrm{w}^{\mathrm{T}} X}}$$

$$p(y=1 \mid X; \theta) = 1 - g(\mathrm{w}^{\mathrm{T}} X) = \frac{e^{\mathrm{w}^{\mathrm{T}} X}}{1+e^{\mathrm{w}^{\mathrm{T}} X}}$$

The sigmoid function

$$p(y \mid X; \theta)$$

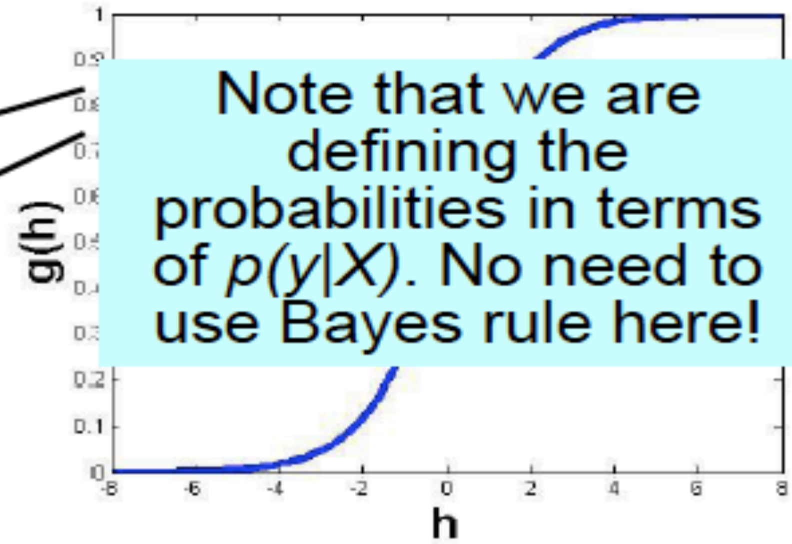
- To classify using regression models we replace the linear function with the sigmoid function:

$$g(h) = \frac{1}{1 + e^{-h}}$$

- Using the sigmoid we set (for binary classification problems)

$$p(y = 0 \mid X; \theta) = g(\mathrm{w}^T X) = \frac{1}{1 + e^{\mathrm{w}^T X}}$$

$$p(y = 1 \mid X; \theta) = 1 - g(\mathrm{w}^T X) = \frac{e^{\mathrm{w}^T X}}{1 + e^{\mathrm{w}^T X}}$$

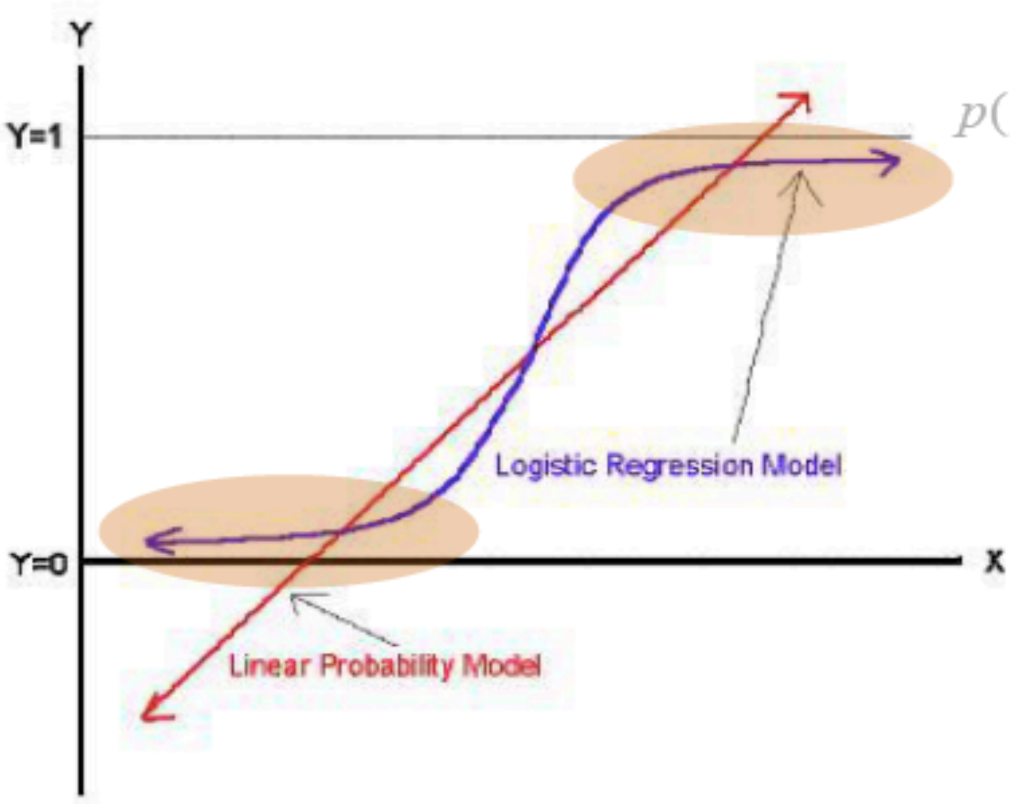


Note that we are defining the probabilities in terms of *p(y|X)*. No need to use Bayes rule here!

Logistic regression vs. Linear regression

$$p(y=0 \mid X;\theta) = g(\mathrm{w}^{\mathrm{T}}X) = \frac{1}{1+e^{\mathrm{w}^{\mathrm{T}}X}}$$

$$p(y=1 \mid X;\theta) = 1 - g(\mathrm{w}^{\mathrm{T}}X) = \frac{e^{\mathrm{w}^{\mathrm{T}}X}}{1+e^{\mathrm{w}^{\mathrm{T}}X}}$$



Determining parameters for logistic regression problems

- So how do we learn the parameters?

Determining parameters for logistic regression problems

- So how do we learn the parameters?
- Similar to other regression problems we look for the MLE for w
- The likelihood of the data given the model is:

Determining parameters for logistic regression problems

- So how do we learn the parameters?
- Similar to other regression problems we look for the MLE for w
- The likelihood of the data given the model is:

$$L(y \mid X; w) = \prod_i (1 - g(X_i; w))^{y_i} g(X_i; w)^{(1-y_i)}$$

Determining parameters for logistic regression problems

- So how do we learn the parameters?
- Similar to other regression problems we look for the MLE for w
- The likelihood of the data given the model is:

Defining a new function, *g*

$$p(y=0 \mid X;\theta) = g(X;w) = \frac{1}{1+e^{w^T X}}$$

$$p(y=1 \mid X;\theta) = 1 - g(X;w) = \frac{e^{w^T X}}{1+e^{w^T X}}$$

$$L(y \mid X;w) = \prod_i (1-g(X_i;w))^{y_i}\, g(X_i;w)^{(1-y_i)}$$

Solving logistic regression problems

$$g(X;w) = \frac{1}{1+e^{w^T X}}$$

$$1-g(X;w) = \frac{e^{w^T X}}{1+e^{w^T X}}$$

- The likelihood of the data is: $L(y \mid X;w) = \prod_i (1-g(X_i;w))^{y_i} g(X_i;w)^{(1-y_i)}$

Solving logistic regression problems

$$g(X;w) = \frac{1}{1+e^{w^T X}}$$

$$1 - g(X;w) = \frac{e^{w^T X}}{1+e^{w^T X}}$$

- The likelihood of the data is: $L(y \mid X;w) = \prod_i (1-g(X_i;w))^{y_i} g(X_i;w)^{(1-y_i)}$

- Taking the log we get:

$$LL(y \mid X;w) = \sum_{i=1}^{N} y_i \ln(1-g(X_i;w)) + (1-y_i) \ln g(X_i;w)$$

Solving logistic regression problems

$$g(X;w) = \frac{1}{1+e^{w^T X}}$$

$$1-g(X;w) = \frac{e^{w^T X}}{1+e^{w^T X}}$$

- The likelihood of the data is: $L(y \mid X;w) = \prod_i (1-g(X_i;w))^{y_i} g(X_i;w)^{(1-y_i)}$

- Taking the log we get:

$$LL(y \mid X;w) = \sum_{i=1}^{N} y_i \ln(1-g(X_i;w)) + (1-y_i) \ln g(X_i;w)$$

$$= \sum_{i=1}^{N} y_i \ln \frac{1-g(X_i;w)}{g(X_i;w)} + \ln g(X_i;w)$$

Solving logistic regression problems

$$g(X;w) = \frac{1}{1+e^{\mathrm{w}^{\mathrm{T}}X}}$$

$$1-g(X;w) = \frac{e^{\mathrm{w}^{\mathrm{T}}X}}{1+e^{\mathrm{w}^{\mathrm{T}}X}}$$

- The likelihood of the data is: $L(y \mid X;w) = \prod_i (1-g(X_i;w))^{y_i} g(X_i;w)^{(1-y_i)}$

- Taking the log we get:

$$\begin{aligned}
LL(y \mid X;w) &= \sum_{i=1}^{N} y_i \ln(1-g(X_i;w)) + (1-y_i)\ln g(X_i;w) \\
&= \sum_{i=1}^{N} y_i \ln \frac{1-g(X_i;w)}{g(X_i;w)} + \ln g(X_i;w) \\
&= \sum_{i=1}^{N} y_i \mathrm{w}^{\mathrm{T}} X_i - \ln(1+e^{w^T X_i})
\end{aligned}$$

Maximum likelihood estimation

$$\frac{\partial}{\partial w^j} l(w) = \frac{\partial}{\partial w^j} \sum_{i=1}^{N} \{y_i \mathrm{w}^{\mathrm{T}} X_i - \ln(1 + e^{\mathrm{w}^{\mathrm{T}} X_i})\}$$

$$g(X; w) = \frac{1}{1 + e^{\mathrm{w}^{\mathrm{T}} X}}$$

$$1 - g(X; w) = \frac{e^{\mathrm{w}^{\mathrm{T}} X}}{1 + e^{\mathrm{w}^{\mathrm{T}} X}}$$

Maximum likelihood estimation

$$\frac{\partial}{\partial w^j} l(w) = \frac{\partial}{\partial w^j} \sum_{i=1}^{N} \{ y_i \mathrm{w}^{\mathrm{T}} X_i - \ln(1 + e^{\mathrm{w}^{\mathrm{T}} X_i}) \}$$

$$= \sum_{i=1}^{N} X_i^j \{ y_i - (1 - g(X_i; w)) \}$$

$$g(X; w) = \frac{1}{1 + e^{\mathrm{w}^{\mathrm{T}} X}}$$

$$1 - g(X; w) = \frac{e^{\mathrm{w}^{\mathrm{T}} X}}{1 + e^{\mathrm{w}^{\mathrm{T}} X}}$$

Maximum likelihood estimation

$$\frac{\partial}{\partial w^j} l(w) = \frac{\partial}{\partial w^j} \sum_{i=1}^{N} \{y_i \mathrm{w}^T X_i - \ln(1 + e^{\mathrm{w}^T X_i})\}$$
$$= \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i; w))\}$$
$$= \sum_{i=1}^{N} X_i^j \{y_i - p(y^i = 1 | X_i; w)\}$$

$$g(X; w) = \frac{1}{1 + e^{\mathrm{w}^T X}}$$

$$1 - g(X; w) = \frac{e^{\mathrm{w}^T X}}{1 + e^{\mathrm{w}^T X}}$$

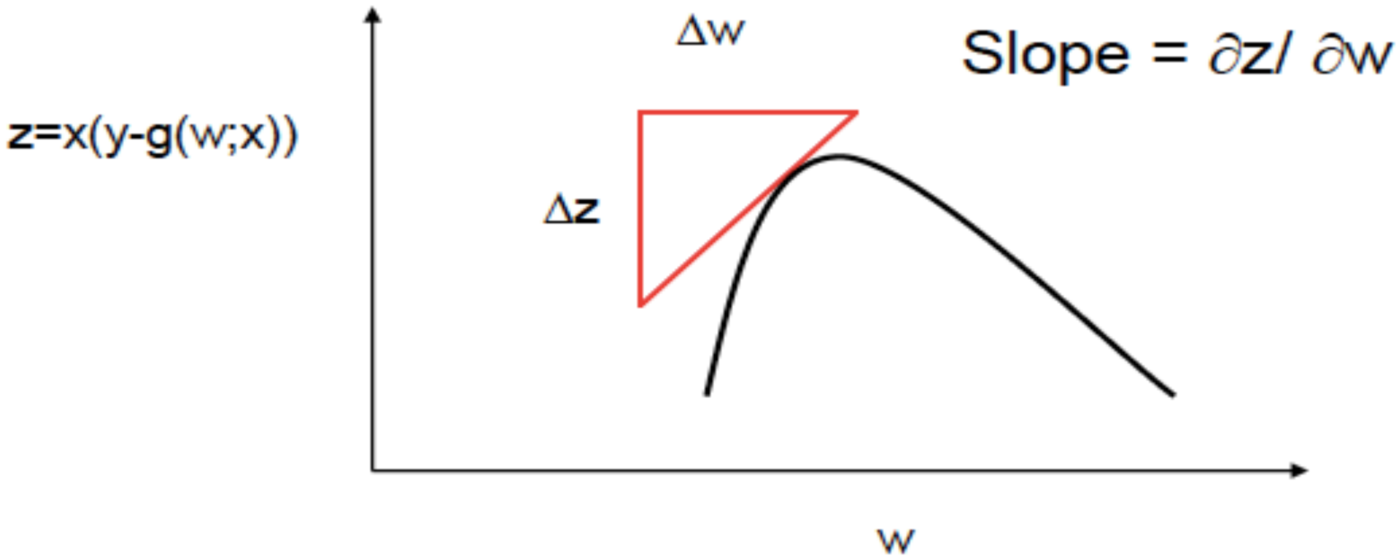
Taking the partial derivative w.r.t. each component of the **w** vector

Bad news: No close form solution!

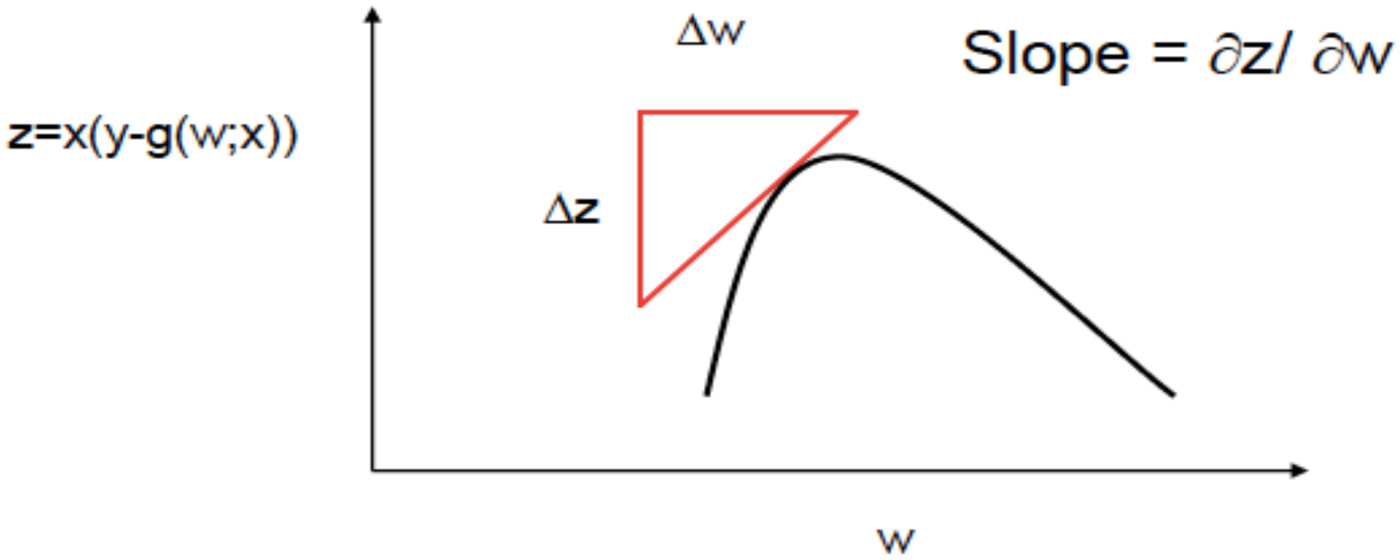
Good news: Concave function

Five mins break!

Gradient ascent

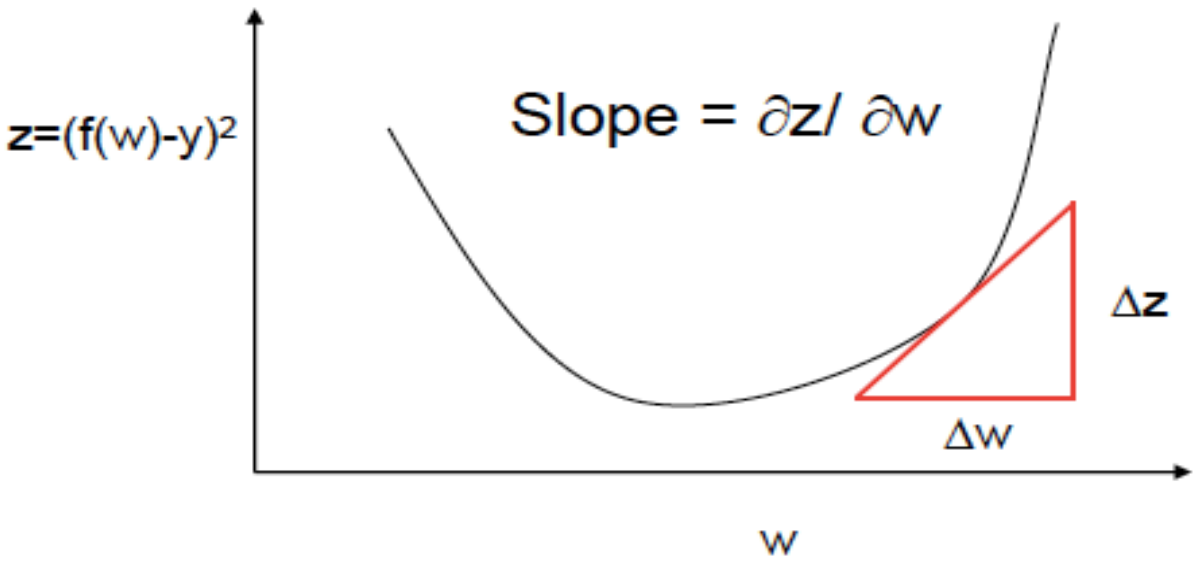


Gradient ascent

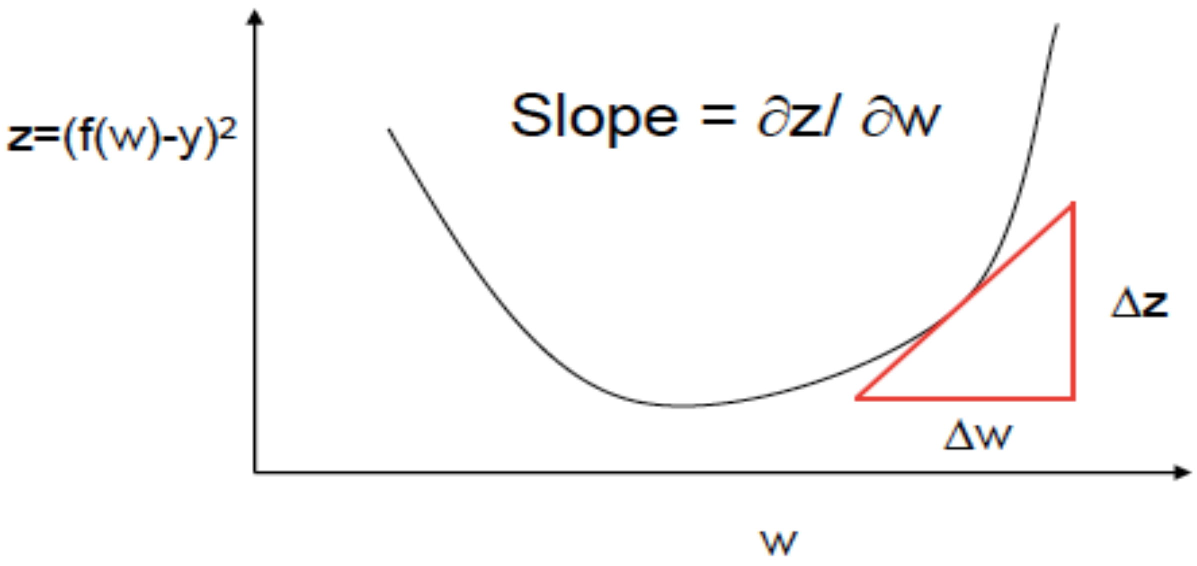


- Going in the direction to the slope will lead to a larger z
- But not too much, otherwise we would go beyond the optimal w

Gradient descent



Gradient descent



- Going in the *opposite* direction to the slope will lead to a smaller z
- But not too much, otherwise we would go beyond the optimal w

Gradient ascent for logistic regression

Gradient ascent for logistic regression

$$\frac{\partial}{\partial w^j} l(w) = \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i; w))\}$$

Gradient ascent for logistic regression

$$\frac{\partial}{\partial w^j} l(w) = \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i; w))\}$$

We use the gradient to adjust the value of w:

$$w^j \leftarrow w^j + \varepsilon \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i; w))\}$$

Where ε is a (small) constant

Algorithm for logistic regression

1. Chose λ
2. Start with a guess for **w**
3. For all j set
$$w^j \leftarrow w^j + \varepsilon \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i; w))\}$$
4. If no improvement for
$$LL(y \mid X; w) = \sum_{i=1}^{N} y_i \ln(1 - g(X_i; w)) + (1 - y_i) \ln g(X_i; w)$$

   stop. Otherwise go to step 3

Example

Regularization

- Similar to other data estimation problems, we may not have enough samples to learn good models for logistic regression classification
- One way to overcome this is to ‘regularize’ the model, impose additional constraints on the parameters we are fitting.

d

Regularization

- Similar to other data estimation problems, we may not have enough samples to learn good models for logistic regression classification
- One way to overcome this is to ‘regularize’ the model, impose additional constraints on the parameters we are fitting.
- For example, lets assume that w^j comes from a Gaussian distribution with mean 0 and variance σ^2 (where σ^2 is a user defined parameter): $w^j \sim N(0, \sigma^2)$
- In that case we have **a prior** on the parameters and so:

$$p(y=1,\theta \mid X) \propto p(y=1 \mid X;\theta)p(\theta)$$

Regularization

- If we regularize the parameters we need to take the prior into account when computing the posterior for our parameters

$$p(y=1,\theta \mid X) \propto p(y=1 \mid X;\theta)\,p(\theta)$$

Regularization

- If we regularize the parameters we need to take the prior into account when computing the posterior for our parameters

$$p(y=1,\theta \mid X) \propto p(y=1 \mid X;\theta)\, p(\theta)$$

- Here we use a Gaussian model for the prior.
- Thus, the log likelihood changes to :

$$LL(y;w \mid X) = \sum_{i=1}^{N} y_i \mathrm{w}^{\mathrm{T}} X_i - \ln(1+e^{w^T X_i}) - \sum_j \frac{(w^j)^2}{2\sigma^2}$$

Assuming mean of 0 and removing terms that are not dependent on w

Regularization

- If we regularize the parameters we need to take the prior into account when computing the posterior for our parameters

$$p(y=1,\theta \mid X) \propto p(y=1 \mid X;\theta)\,p(\theta)$$

- Here we use a Gaussian model for the prior.
- Thus, the log likelihood changes to :

$$LL(y;w \mid X) = \sum_{i=1}^{N} y_i \mathrm{w}^{\mathrm{T}} X_i - \ln(1+e^{w^T X_i}) - \sum_j \frac{(w^j)^2}{2\sigma^2}$$

Assuming mean of 0 and removing terms that are not dependent on w

- And the new update rule (after taking the derivative w.r.t. w^j) is:

$$w^j \leftarrow w^j + \varepsilon \sum_{i=1}^{N} X_i^j \{y_i - (1 - g(X_i;w))\} - \varepsilon \frac{w^j}{\sigma^2}$$

Also known as the MAP estimate

The variance of our prior model

Regularization

- There are many other ways to regularize logistic regression
- The Gaussian model leads to an L2 regularization (we are trying to minimize the square value of w)
- Another popular regularization is an L1 which tries to minimize $|w|$

Important points

- Advantage of logistic regression over linear regression for classification
- Sigmoid function
- Gradient ascent / descent
- Regularization
- Logistic regression for multiple classes

Logistic regression

- The name comes from the **logit** transformation:

$$\log \frac{p(y=i \mid X;\theta)}{p(y=k \mid X;\theta)} = \log \frac{g(z_i)}{g(z_k)} = w_i^0 + w_i^1 x^1 + \ldots + w_i^d x^d$$

That’s all!