

10-701

Machine Learning

Logistic regression

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

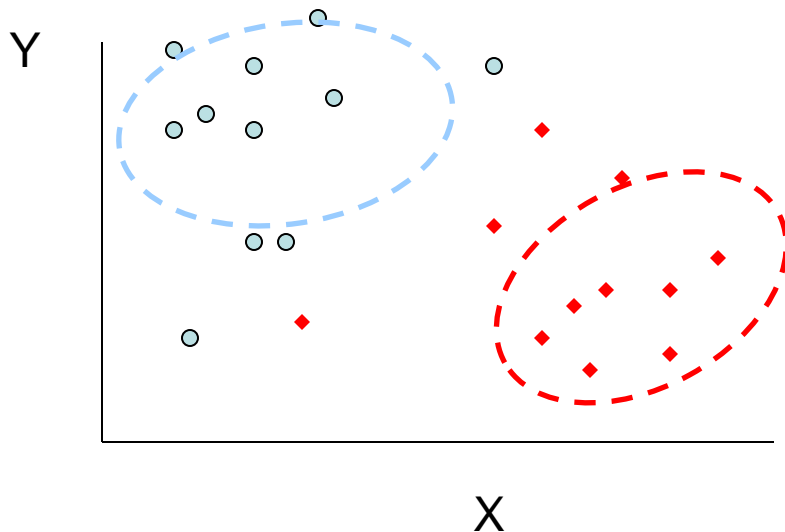
3. Discriminative

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

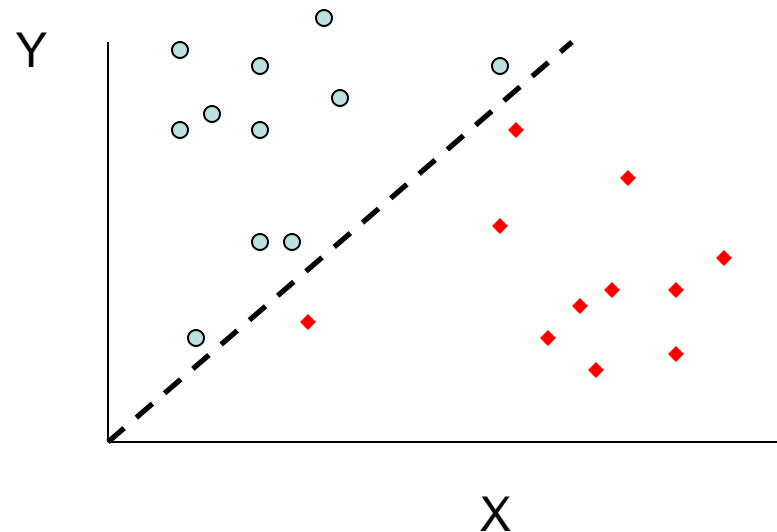
Generative vs. discriminative classifiers

- 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

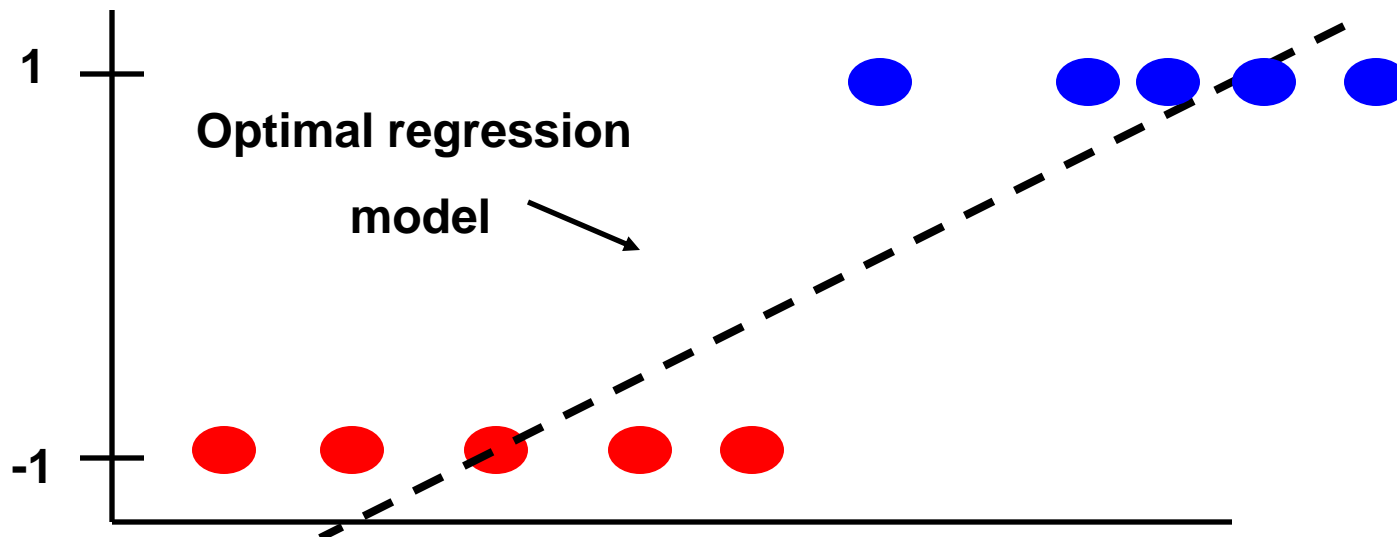
- In some cases we can use linear regression for determining the appropriate boundary.
- However, since the output is usually binary or discrete there are more efficient regression methods
- Recall that for classification we are interested in the conditional probability $p(y | X ; \theta)$ where θ are 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 $p(y | X ; \theta)$
- Problems?

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

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



The sigmoid function

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

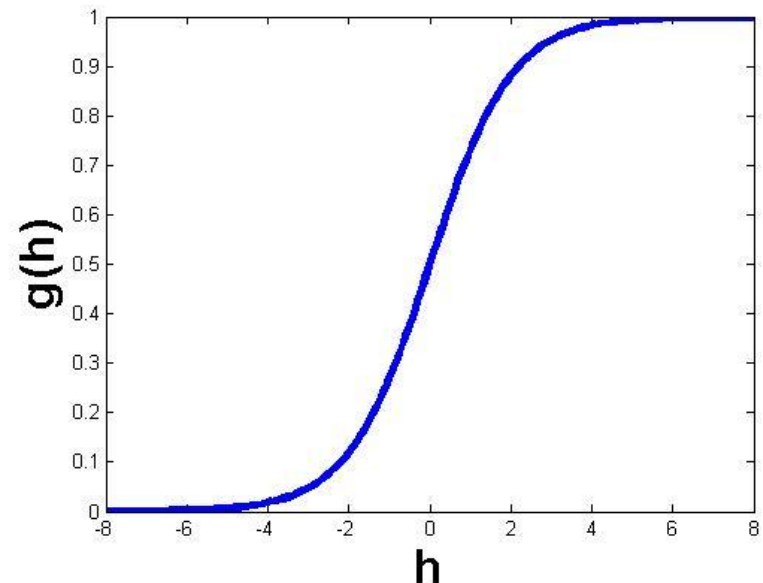
- To classify using regression models we replace the linear function with the sigmoid function:

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

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

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

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



The sigmoid function

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

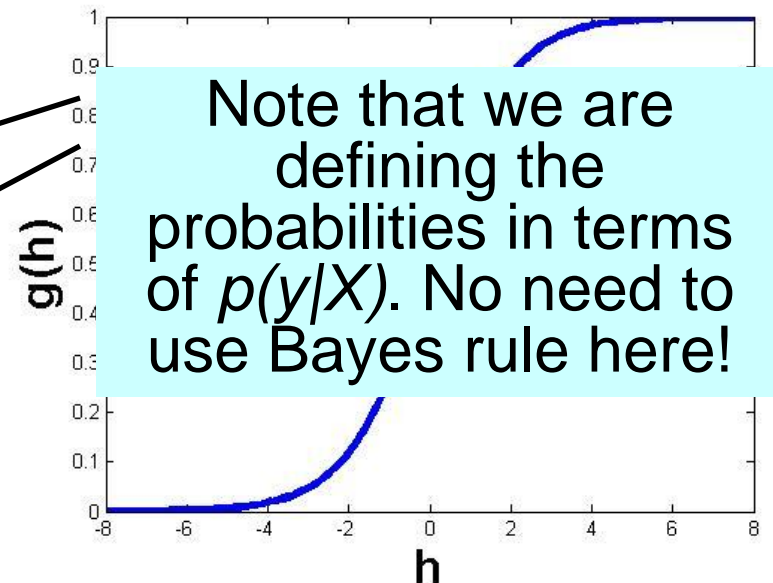
- To classify using regression models we replace the linear function with the sigmoid function:

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

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

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

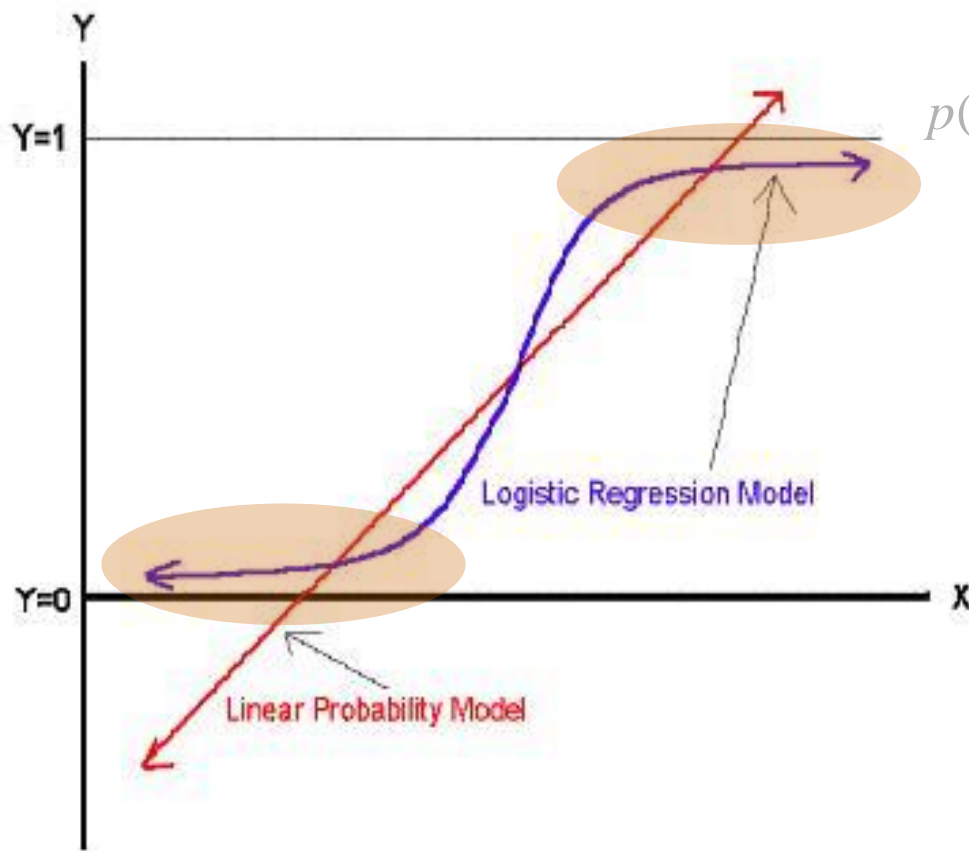
$$p(y = 1 | X; \theta) = 1 - g(w^T X) = \frac{e^{w^T X}}{1 + e^{w^T X}}$$



Logistic regression vs. Linear regression

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


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



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


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

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

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

- Taking the log we get:

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

Maximum likelihood estimation

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

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

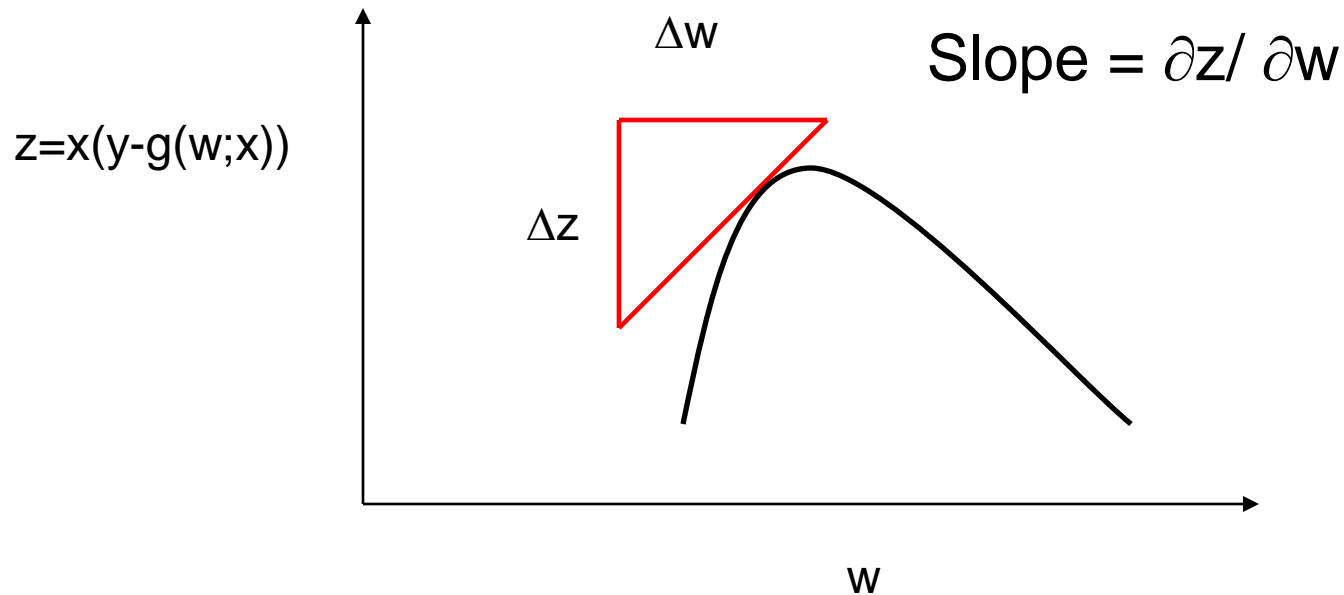
$$1 - g(X; w) = \frac{e^{w^T X}}{1 + e^{w^T X}}$$

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

Bad news: No close form solution!

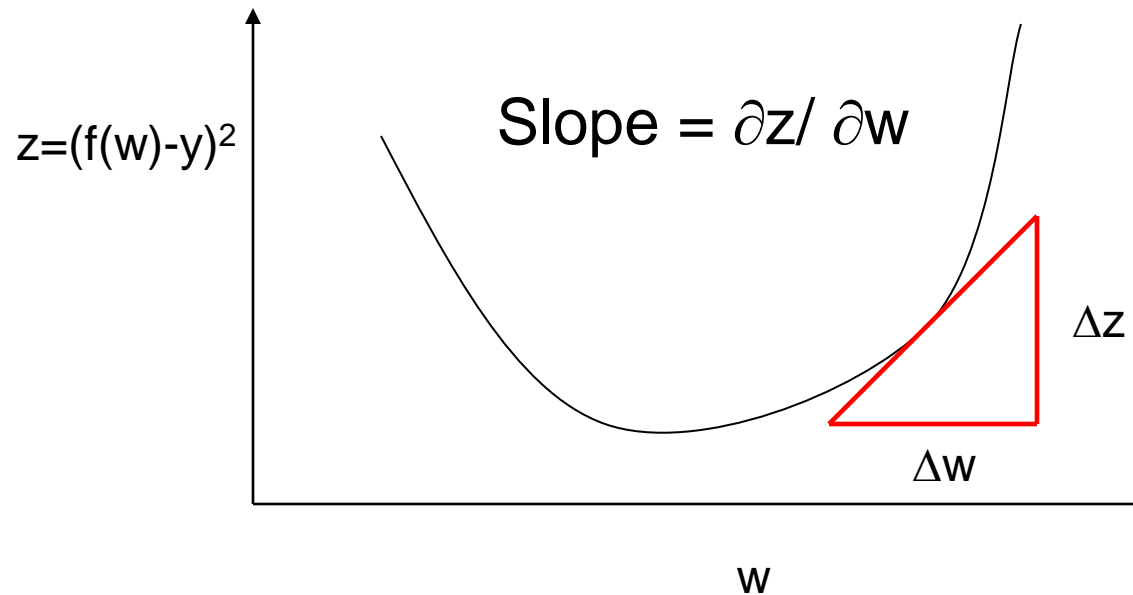
Good news: Concave function

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



- 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

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

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

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

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.
- For example, lets assume that w^i comes from a Gaussian distribution with mean 0 and variance σ^2 (where σ^2 is a user defined parameter): $w^i \sim N(0, \sigma^2)$
- In that case we have **a prior** on the parameters and so:

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

Regularization

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

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

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

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

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 $|w|$

Logistic regression for more than 2 classes

- Logistic regression can be used to classify data from more than 2 classes. Assume we have k classes then:
- for $i < k$ we set

$$p(y = i | X; \theta) = g(w_i^0 + w_i^1 x^1 + \dots + w_i^d x^d) = g(w_i^T X)$$

where

$$g(z_i) = \frac{e^{z_i}}{1 + \sum_{j=1}^{k-1} e^{z_j}} \quad z_i = w_i^0 + w_i^1 x^1 + \dots + w_i^d x^d$$

And for k we have

$$p(y = k | X; \theta) = 1 - \sum_{i=1}^{k-1} p(y = i | X; \theta) \Rightarrow$$

$$p(y = k | X; \theta) = \frac{1}{1 + \sum_{j=1}^{k-1} e^{z_j}}$$

Logistic regression for more than 2 classes

- Logistic regression can be used to classify data from more than 2 classes. Assume we have k classes then:
- for $i < k$ we set

$$p(y = i | X; \theta) = g(w_i^0 + w_i^1 x^1 + \dots + w_i^d x^d) = g(w_i^T X)$$

where $g(z_i) = \frac{e^{z_i}}{1 + \sum_{j=1}^{k-1} e^{z_j}}$ $z_i = w_i^0$ Binary logistic regression is a special case of this rule

And for k we have $p(y = k | X; \theta) = 1 - \sum_{i=1}^{k-1} p(y = i | X; \theta) \Rightarrow$

$$p(y = k | X; \theta) = \frac{1}{1 + \sum_{j=1}^{k-1} e^{z_j}}$$

Update rule for logistic regression with multiple classes

$$\frac{\partial}{\partial w_m^j} l(w) = \sum_{i=1}^N X_i^j \{ \delta_m(y_i) - p(y_i = m | X_i; w) \}$$

Where $\delta(y_i)=1$ if $y_i=m$
and $\delta(y_i)=0$ otherwise

The update rule becomes:

$$w_m^j \leftarrow w_m^j + \varepsilon \sum_{i=1}^N X_i^j \{ \delta_m(y_i) - p(y_i = m | X_i; w) \}$$

Data transformation

- Similar to what we did with linear regression we can extend logistic regression to other transformations of the data

$$p(y = 1 | X; w) = g(w_0^0 + w_1^1 \phi^1(X) + \dots + w_1^d \phi^d(X))$$

- As before, we are free to choose the basis functions

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:

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