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

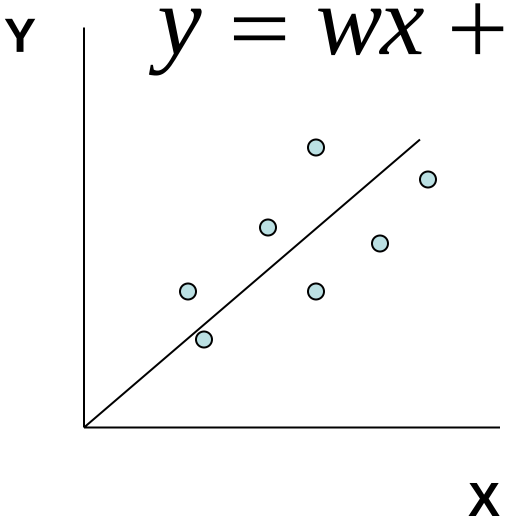
- When using generative classifiers we relied on all points to learn the generative model
- When using discriminative classifiers we mainly care about the boundary
- Unlike generative classifiers, discriminative classifiers *do not* compute P(X|Y)

Linear regression

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

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

$$y = wx + \varepsilon$$



Regression for classification

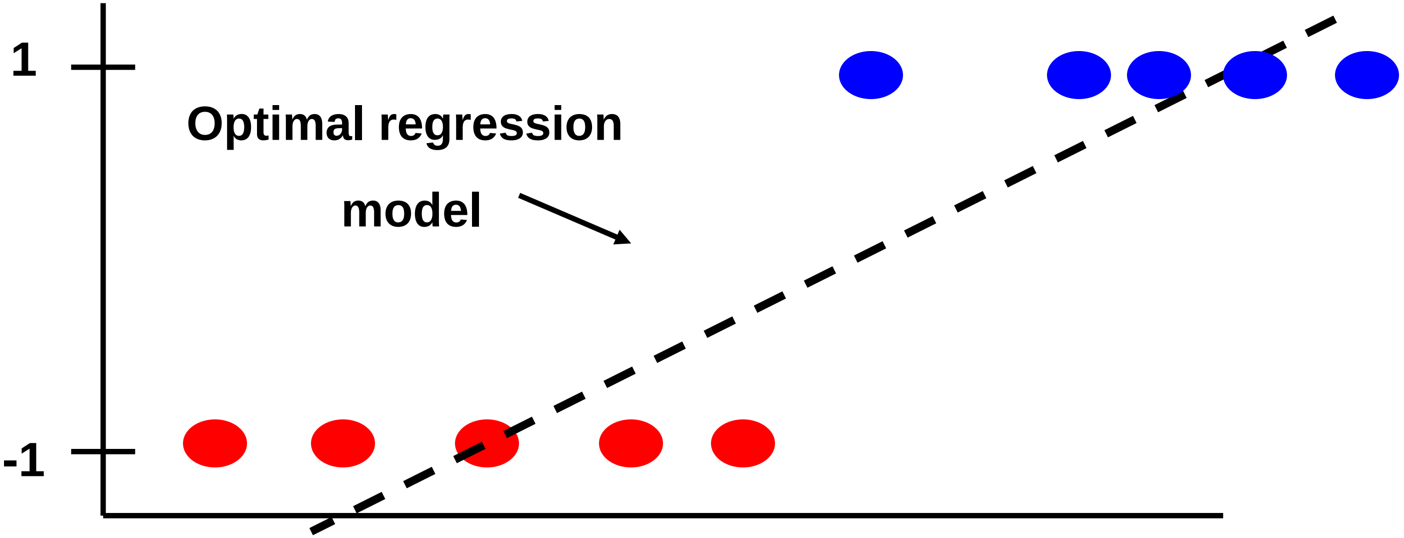
- In some cases we can use linear regression for determining the appropriate boundary.
- However, since the output is usually binary or discrete there are more efficient regression methods
- Recall that for classification we are interested in the conditional probability $p(y \mid X ; \theta)$ where θ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

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

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

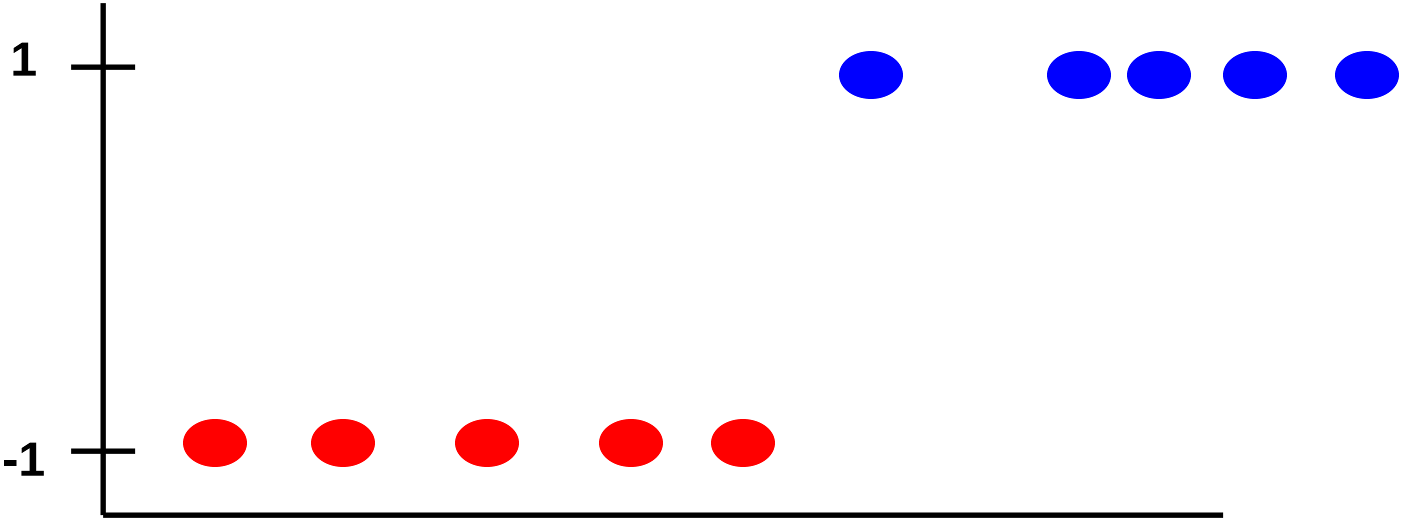


Regression for classification

- Assume we would like to use linear regression to learn the parameters for p(y | X ; θ)
- Problems?

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

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



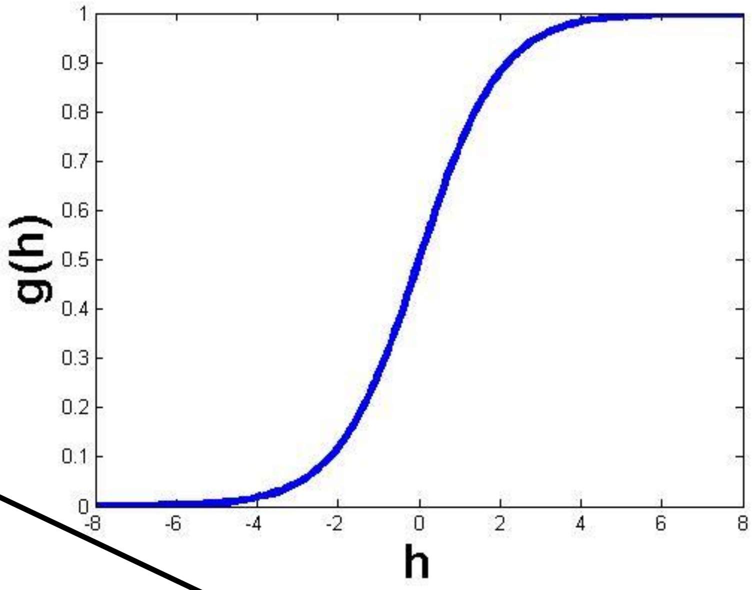
The sigmoid function

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

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

Always between 0 and 1

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



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

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

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

Parameters in the exponent, not a linear regression!

The sigmoid function

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

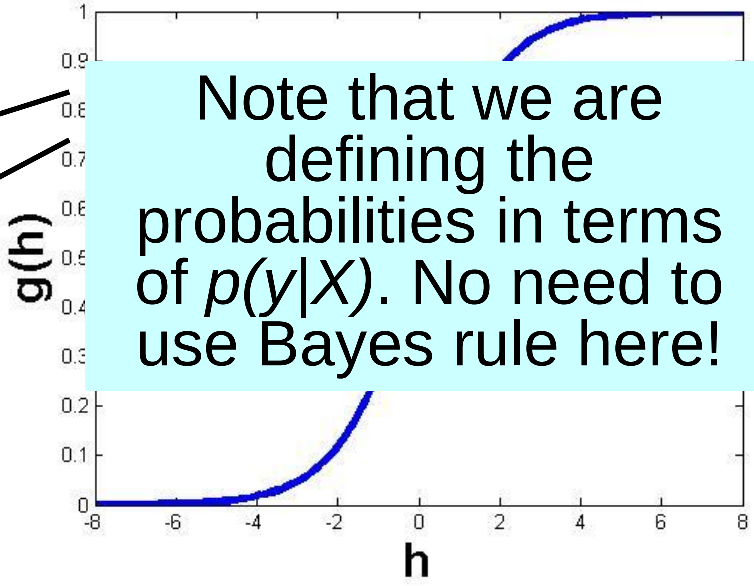
- To classify using regression models we replace the linear function with the sigmoid function:

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

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

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

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

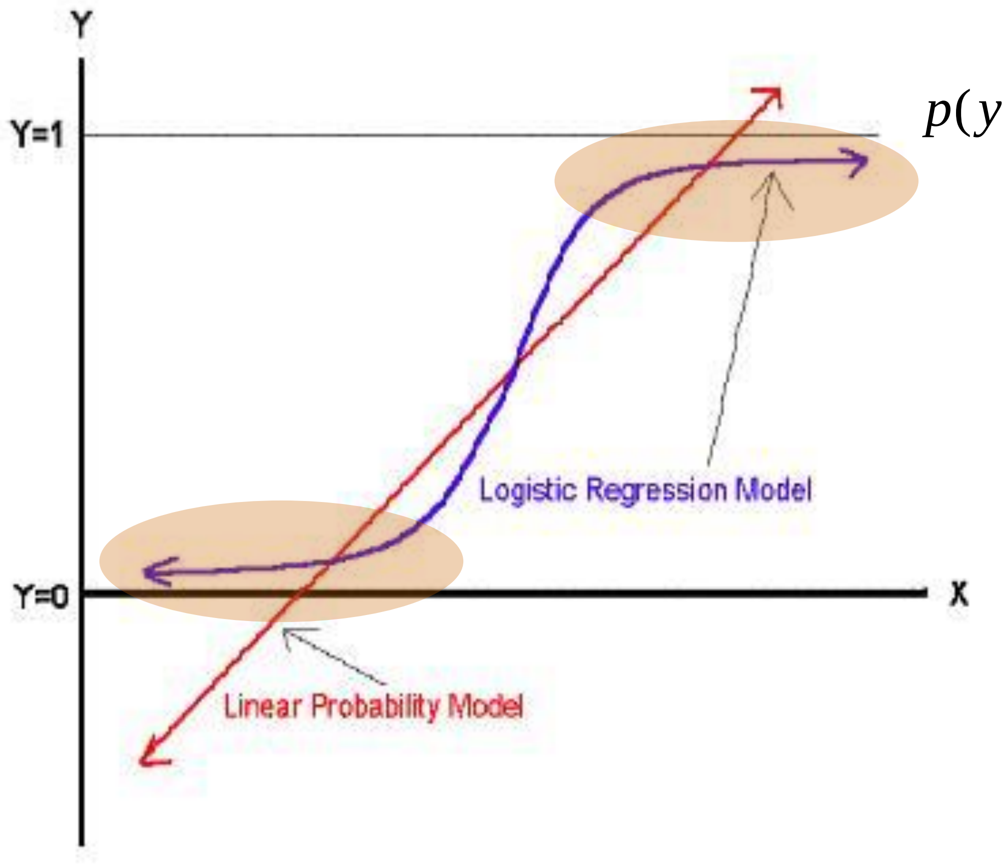


Note that we are defining the probabilities in terms of *p(y|X)*. No need to use Bayes rule here!

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?
- 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*

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

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

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

Solving logistic regression problems

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

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

- 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:

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

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

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

Maximum likelihood estimation

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

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

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

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

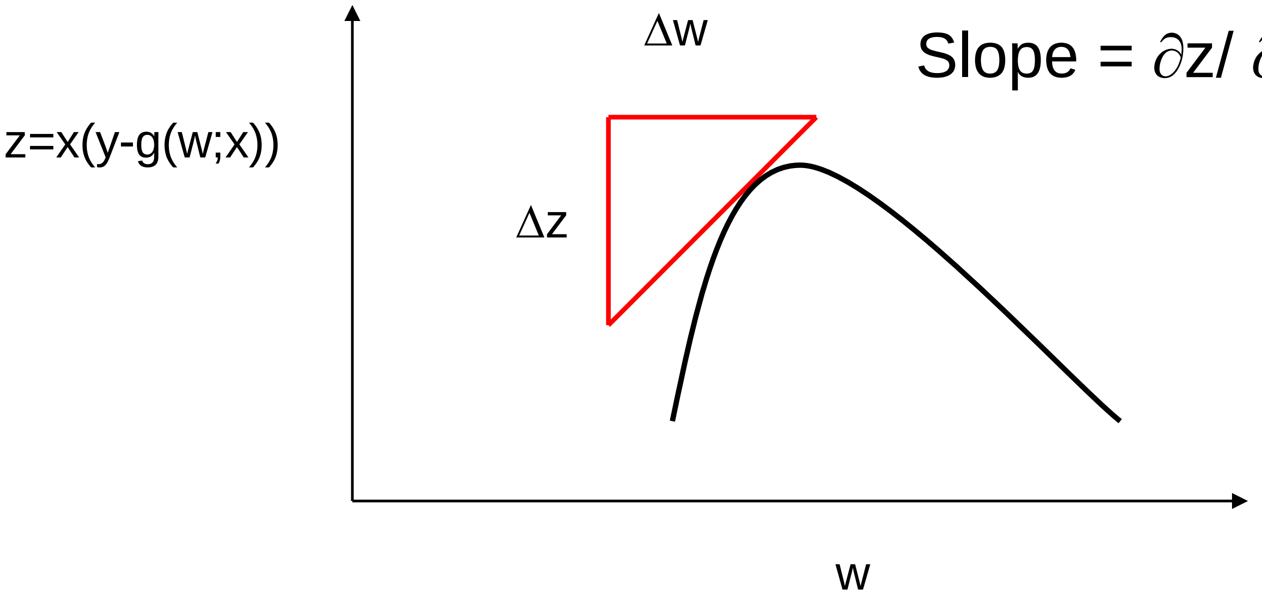
$$1 - g(X; w) = \frac{e^{\mathrm{w}^{\mathrm{T}} X}}{1 + e^{\mathrm{w}^{\mathrm{T}} X}}$$

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

Bad news: No close form solution!

Good news: Concave function

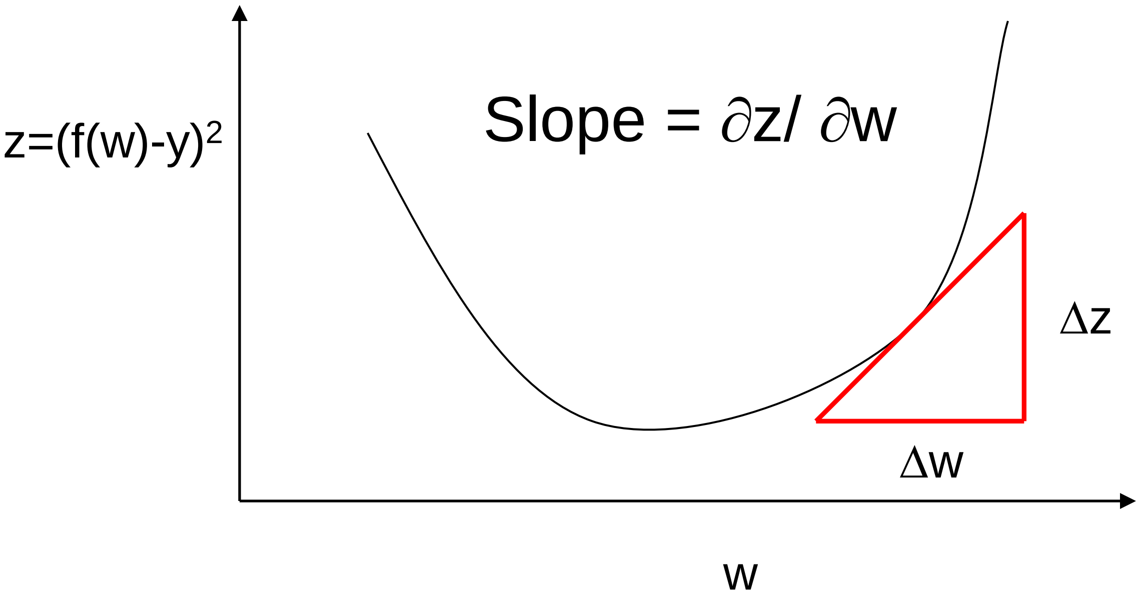
Gradient ascent



Slope = $\partial z / \partial w$

- 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 which is the
learning rate for this algorithm

Algorithm for logistic regression

1. Chose ε
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.
- 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

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

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

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

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^i) is:

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

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

$$p(y=i \mid X;\theta) = g(w_i^0 + w_i^1 x^1 + \ldots + w_i^d x^d) = g(\mathrm{w}_\mathrm{i}^\mathrm{T} X)$$

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

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

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

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

$$p(y=i \mid X;\theta) = g(w_i^0 + w_i^1 x^1 + \ldots + w_i^d x^d) = g(\mathrm{w}_\mathrm{i}^\mathrm{T} X)$$

where $$g(z_i) = \frac{e^{z_i}}{1+\sum_{j=1}^{k-1} e^{z_j}} \qquad z_i = w_i^0$$

Binary logistic regression is a special case of this rule

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

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

Update rule for logistic regression with multiple classes

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

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

The update rule becomes:

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

Data transformation

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

$$p(y=1 \mid X;w) = g(w_1^0 + w_1^1 \phi^1(X) + \ldots + w_1^d \phi^d(X))$$

- 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
- Optional reading: Murphy 8.1-3, 8.6

Logistic regression

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

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