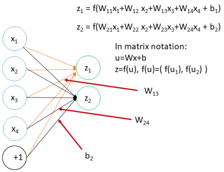
Chapter 3: Introduction to Machine Learning – Linear and Logistic Regression

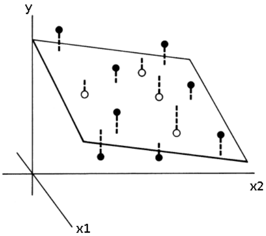


Christian Heumann
(largely based on slides from Nina Poerner, Benjamin Roth, Marina Speranskaya)

CIS LMU München, Department of Statistics LMU München

November 2020

Linear Regression



- In linear regression, the prediction $\hat{y}$ of a label y is a linear function:

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x} = \sum_{j=1}^{n} w_j x_j$$

- Note, that in this notation, the parameter vector $\boldsymbol{w} \in \mathbb{R}^n$ stands already for the *estimated* parameter vector.
- Weight w_j decides if value of feature x_j increases or decreases prediction $\hat{y}$.

Matrix notation I

- m samples
- Prediction for one sample:

$$\hat{y}_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

- Error for one sample (residual):

$$e_i = (y_i - \hat{y}_i)$$

- Squared error for one sample:

$$e_i^2 = (y_i - \hat{y}_i)^2 = (\hat{y}_i - y_i)^2$$

Matrix notation II

- Matrix notation:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_m^T \end{bmatrix} \hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w}$$

- $\boldsymbol{X}$ in detail: typically, first column contains all $\mathbf{1}$ for the intercept (bias, shift) term.

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{12} & x_{13} & \dots & x_{1n} \\ 1 & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

- Estimate parameters using $\boldsymbol{X}^{(train)}$ and $\boldsymbol{y}^{(train)}$.
- Make high-level decisions (which features...) using $\boldsymbol{X}^{(dev)}$ and $\boldsymbol{y}^{(dev)}$.
- Evaluate resulting model using $\boldsymbol{X}^{(test)}$ and $\boldsymbol{y}^{(test)}$.

Simple Example: Housing Prices

- Predict property prices (in 1K Euros) from just one feature: Square feet of property.

$$\boldsymbol{X} = \begin{bmatrix} 1 & 450 \\ 1 & 900 \\ 1 & 1350 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 730 \\ 1300 \\ 1700 \end{bmatrix}$$

- Prediction is:

$$\hat{\boldsymbol{y}} = \begin{bmatrix} w_1 + 450w_2 \\ w_1 + 900w_2 \\ w_1 + 1350w_2 \end{bmatrix} = \begin{bmatrix} 1 & 450 \\ 1 & 900 \\ 1 & 1350 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \boldsymbol{X}\boldsymbol{w}$$

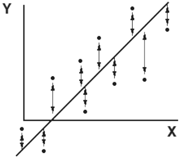
- $\boldsymbol{w}_1$ will contain costs incurred in any property acquisition
- $\boldsymbol{w}_2$ will contain remaining average price per square feet.
- Optimal parameters are for the above case:

$$\boldsymbol{w} = \begin{bmatrix} 273.3 \\ 1.08 \end{bmatrix} \quad \hat{\boldsymbol{y}} = \begin{bmatrix} 759.1 \\ 1245.1 \\ 1731.1 \end{bmatrix}$$

Linear Regression: Mean Squared Error

- Mean squared error of training (or test) data set is the sum of squared differences between the predictions and labels of all m instances.

$$MSE := \frac{1}{m}\sum_{i=1}^{m} e_i^2 = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$



- In matrix notation:

$$MSE := \frac{1}{m}||\hat{\boldsymbol{y}} - \boldsymbol{y}||_2^2$$

$$= \frac{1}{m}||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2$$

Learning: Improving on MSE

- Gradient of a function f: Vector whose components are the n partial derivatives of f wrt to the parameters, here $\boldsymbol{w}$.

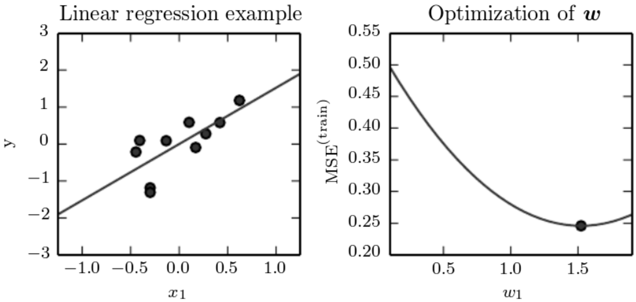
$$\nabla_{\boldsymbol{w}} f(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{w})}{\partial w_1} \\ \frac{\partial f(\boldsymbol{w})}{\partial w_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{w})}{\partial w_n} \end{bmatrix}$$

- View MSE as a function $f(\boldsymbol{w})$ of $\boldsymbol{w}$
- Minimum is where gradient is $\mathbf{0}$:

$$\nabla_{\boldsymbol{w}} MSE \stackrel{!}{=} \mathbf{0}$$

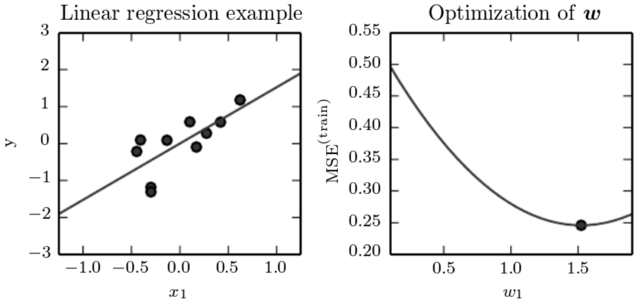
Learning: Improving on MSE

- View MSE as a function of $\boldsymbol{w}$



- Minimum is where gradient $\nabla_{\boldsymbol{w}} MSE = \mathbf{0}$.
- Why minimum and and not maximum or saddle point?
  - Because it is a quadratic function...
  - Check convexity for 1 dimensional function: Second derivative > 0.
  - Check for vector valued function: Hessian is positive-semidefinite.

Second Derivative Test



Second derivative of Mean Squared Error for Linear model with only one feature:

$$\frac{d^2}{dw^2}\sum_{i=1}^{m}(x^{(i)}w - y^{(i)})^2 = \frac{d^2}{dw^2}\sum_{i=1}^{m}(x^{(i)2}w^2 - 2x^{(i)}w + y^{(i)2}) = 2\sum_{i=1}^{m}x^{(i)2} > 0$$

Solving for $\boldsymbol{w}$

- We now know that minimum is where gradient is $\mathbf{0}$.

$$\nabla_{\boldsymbol{w}} MSE = \mathbf{0}$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \frac{1}{m} ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2 = \mathbf{0}$$

- Solve for $\boldsymbol{w}$:

$$\boldsymbol{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

  (Normal Equation)
- The inverse $(\boldsymbol{X}^T \boldsymbol{X})^{-1}$ exists, if X has full column rank (i.e. rank n).

Deriving the Normal Equation

- Function to minimize:

$$
\begin{aligned}
&||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||_2^2 \\
&= (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) \\
&= \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y} \\
&= \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}
\end{aligned}
$$

- Take the gradient[1] w.r.t. $\boldsymbol{w}$ and set equal to $\mathbf{0}$:

$$
\begin{aligned}
&2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{X}^T\boldsymbol{y} = \mathbf{0} \\
&\Rightarrow \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y} \\
&\Rightarrow (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}
\end{aligned}
$$

[1] [Matrix Cookbook. Petersen and Pedersen, 2012]:

$\nabla_{\boldsymbol{w}}\boldsymbol{w}^T\boldsymbol{a} = \boldsymbol{a}$

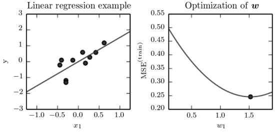
$\nabla_{\boldsymbol{w}}\boldsymbol{w}^T\boldsymbol{B}\boldsymbol{w} = 2\boldsymbol{B}\boldsymbol{w}$ for symmetric $\boldsymbol{B}$

Practical Remarks

What if $\boldsymbol{X}^T\boldsymbol{X}$ does not have an inverse?

- This can happen if there are infinitely many solutions:
  - one feature is the exact multiple of another
  - there are more features than training examples
  - $\Rightarrow$ a Moore-Penrose pseudoinverse picks solution with smallest Euclidean norm.
  - $\Rightarrow$ adding a regularization term makes the system of equations non-singular (uniquely solvable).
    - Normal Equation becomes: $\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
    - Also called *Ridge Regression*.
- In general: **Avoid computing the matrix inverse when implementing least squares (slow/unstable)!**
- There are usually special routines for solving linear least squares and rigde regression.

Second Derivative Test for Multi-Dimensional Case



- Second derivative test for multi-dimensional $\boldsymbol{w} \in \mathbb{R}^n$
  Is Hessian positive-semidefinite? ($\boldsymbol{z}^{\boldsymbol{T}} \mathbb{H} \boldsymbol{z} > 0$ for $\boldsymbol{z} \neq \boldsymbol{0}$)
  $\Leftrightarrow$ Function is convex.
- Hessian: Matrix of second-order partial derivatives.

$$\mathbb{H}_{j,k} = \frac{\partial^2 f(\boldsymbol{w})}{\partial w_j \partial w_k}$$

- Matrix algebra[2] gives:

$$\mathbb{H} = 2\boldsymbol{X}^T\boldsymbol{X} = 2\sum_{i=1}^{m} \boldsymbol{x}^{(i)} \boldsymbol{x}^{(i)T}; \quad \boldsymbol{z}^T \mathbb{H} \boldsymbol{z} = 2\sum_{i=1}^{m} (\boldsymbol{z}^T \boldsymbol{x}^{(i)})^2 > 0$$

[2] [Matrix Cookbook. Petersen and Pedersen, 2012]

Linear Regression: Summary

- Linear regression models simple linear relationships between $\boldsymbol{X}$ and $\boldsymbol{y}$
- The Mean squared error is a quadratic function of the parameter vector $\boldsymbol{w}$, and has a unique minimum.
- Normal equations: Find the minimum by setting the gradient to zero and solving for $\boldsymbol{w}$.
- Linear algebra packages have special routines for solving least squares linear regression.

Maximum Likelihood Estimation

- Machine learning models are often more interpretable if they are stated in a probabilistic way.
- Performance measure: What is the probability of the training data given the model parameters? Works only well for discrete data! More general: use densities instead of probabilities
- Likelihood: Probability density of data as a function of model parameters. Generally, likelihood is a function proportional to the density.
- $\Rightarrow$ Maximum Likelihood Estimation
- Many models can be formulated in a probabilistic way!

Probability density of a data set

- Data:
  - Set of m examples $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \dots \boldsymbol{x}^{(m)}\}$
  - Sometimes written as design matrix:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(1)T} \\ \vdots \\ \boldsymbol{x}^{(m)T} \end{bmatrix}$$

- Probability density of a data set $\boldsymbol{X}$, parametrized by $\boldsymbol{\theta}$:

$$p_{model}(\boldsymbol{X}; \boldsymbol{\theta})$$

Probability density of a data set

- Data points are assumed to be (stochastically) independent (and sometimes identically distributed) random variables (i.d. or i.i.d.)
  - Assumption made by many ML models.
  - Identically distributed: Examples come from same distribution.
  - Independent: Value of one example doesn't influence other example.
  - $\Rightarrow$ Probability density of the data set is the product of example probability densities.

$$p_{model}(\boldsymbol{X};\boldsymbol{\theta}) = \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)};\boldsymbol{\theta})$$

Maximum Likelihood Estimation

- Likelihood: Probability density of data viewed as function of parameters $\boldsymbol{\theta}$
- (Negative) Log-Likelihood (NLL):
  - Logarithm is monotonically increasing
    - Maximum of function stays the same
    - Easier to do arithmetic with (sums vs. products)
  - Optimization is often formulated as minimization ⇒ take negative of function.
- Maximum likelihood estimator for $\boldsymbol{\theta}$:

$$
\begin{aligned}
\boldsymbol{\theta}_{ML} &= \operatorname{argmax}_\theta \, p_{model}(\boldsymbol{X}; \boldsymbol{\theta}) \\
&= \operatorname{argmax}_\theta \prod_{i=1}^{m} p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= \operatorname{argmax}_\theta \sum_{i=1}^{m} \log p_{model}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})
\end{aligned}
$$

Conditional Log-Likelihood

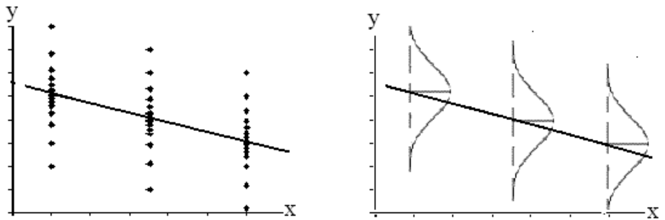
- Log-likelihood can be stated for supervised and unsupervised tasks.
- Unsupervised learning (e.g. density estimation).
  - Task: model $p_{model}(\boldsymbol{X};\boldsymbol{\theta})$ (as before)
  - $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \dots \boldsymbol{x}^{(m)}\}$
- Supervised learning (Predictive modeling):
  - Task: model $p_{model}(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta})$
  - $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \dots \boldsymbol{x}^{(m)}\}, \quad \boldsymbol{y} = \{y^{(1)}, \dots y^{(m)}\}$
- Maximum likelihood estimation for the supervised case (independent examples):

$$\boldsymbol{\theta}_{ML} = \operatorname{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta})$$

$$= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log P(y^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta})$$

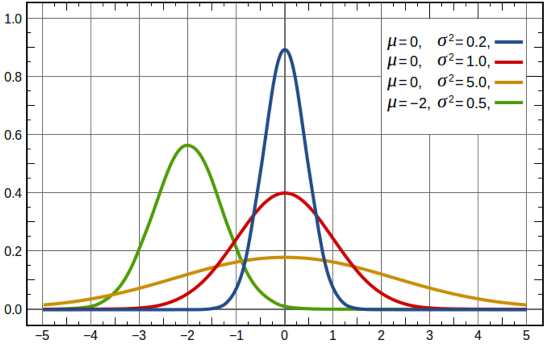
Linear Regression as Maximum Likelihood

- Instead of predicting one value $\hat{y}$ for an input $\boldsymbol{x}$, model probability distribution $p(y|\boldsymbol{x})$.
- For the same value of $\boldsymbol{x}$, different values of y may occur (with different probability).



Gaussian Distribution

- Gaussian distribution: $N(y|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right]$
  - Quadratic function as negative exponent, scaled by variance
  - Normalization factor $\frac{1}{\sigma\sqrt{2\pi}}$



Linear Regression as Maximum Likelihood

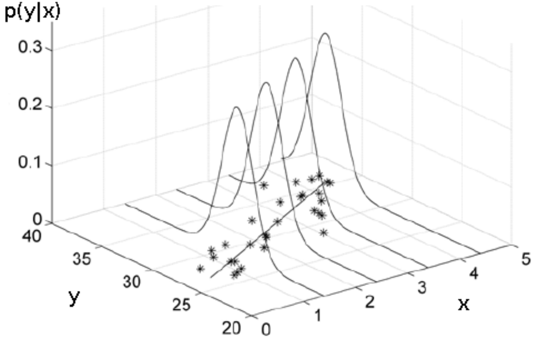
- Assume label y is distributed by a Gaussian, depending on features $\boldsymbol{x}$

$$p(y|\boldsymbol{x}) = N(y|\mu, \sigma^2)$$

where the mean is determined by the linear transformation

$$\mu = \boldsymbol{\theta}^T \boldsymbol{x}$$

and σ is a constant.



Linear Regression as Maximum Likelihood

- Gaussian distribution: $N(y|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right]$
  - Taking the log makes it a quadratic function!
- Conditional negative log-likelihood:

$$
\begin{aligned}
&- \log P(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}) \\
&= - \sum_{i=1}^{m} \log p(y^{(i)}|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= m \log \sigma + \frac{m}{2} \log(2\pi) + \sum_{i=1}^{m} \frac{(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2}{2\sigma^2} \\
&= \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2
\end{aligned}
$$

- What is the optimal value for $\boldsymbol{\theta}$?

Linear Regression as Maximum Likelihood

- Conditional negative log-likelihood:

$$NLL(\boldsymbol{\theta}) = \text{const} + \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y^{(i)} - \boldsymbol{\theta}^T\boldsymbol{x}^{(i)})^2$$

- Compare to previous result:

$$MSE(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{n}(y^{(i)} - \boldsymbol{w}^T\boldsymbol{x}^{(i)})^2$$

- Minimizing NLL under these assumptions is equivalent to minimizing MSE!
- The result of the minimization of the NLL is the MLE $\hat{\boldsymbol{\theta}}$.

A Caveat: Maximum Likelihood is not *Bayesian*

- MLE is simple: Identify distribution and parameters, then maximize!
- MLE accounts for (some) uncertainty: Instead of predicting a single value $\hat{y}$, treat y as a random variable.
- However, what about the uncertainty about our parameters $\boldsymbol{\theta}$?
- Shouldn't $\boldsymbol{\theta}$ be treated as a random variable, too?
  (Instead of a point estimate $\hat{\boldsymbol{\theta}}$?)

A Caveat: Maximum Likelihood is not *Bayesian*

- Uncertainty about $\boldsymbol{\theta}$ influenced by:
  - Our assumptions what reasonable values for $\boldsymbol{\theta}$ look like.
  - The amount of training data.
  - The properties (variance ...) of the training data.
- *Bayesian inference* is all about modelling randomness of $\boldsymbol{\theta}$ in a sound way.
- Why is it called *Bayesian*?

$$p(\boldsymbol{\theta}|\boldsymbol{X}) = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{X})} = \frac{p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

- The Bayesian approach is theoretically more appealing than working with point estimates.
- In practical terms: Sometimes going the Bayesian way pays off, sometimes it doesn't.

Maximum Likelihood: Summary

- Many machine learning problems can be stated in a probabilistic way.
- Mean squared error linear regression can be stated as a probabilistic model that allows for Gaussian random noise around the predicted value $\hat{y}$.
- A straightforward optimization is to maximize the likelihood of the training data.
- Maximum likelihood is not Bayesian, and may give undesirable results (e.g. if there is only little training data).
- In practice, MLE and point estimates are often used to solve machine learning problems.

Linear regression and mean squared error of prediction

- The linear model is given by (Y is a random variable, y its realization)

$$Y = \boldsymbol{\theta}^T \boldsymbol{x} + \varepsilon$$

with $\varepsilon \sim N(0, \sigma^2)$. Thus the *expectation* of Y is $E(Y) = \boldsymbol{\theta}^T \boldsymbol{x}$.

- Using the estimated parameter (Least Squares or MLE) $\hat{\boldsymbol{\theta}}$, the predicted value $\hat{y}$ is given by

$$\hat{y} = \hat{\boldsymbol{\theta}}^T \boldsymbol{x}$$

which is also the *estimated* expectation of y, $\widehat{E(Y)}$.

- The theoretical quantity, which is estimated by the introduced MSE, is the so-called MSEP (*mean squared error of prediction*), which is also an expectation, based on a *new* observation Y_{m+1} (we ignore the index in the following) and its prediction $\hat{y} = \hat{\boldsymbol{\theta}}^T \boldsymbol{x}_{m+1}$:

$$MSEP = E\left\{(Y - \hat{y})^2\right\} = E\{(Y - E(Y) + E(Y) - \hat{y})^2\} = Var(Y) + MSE(\hat{y}) \ .$$

Note, that some terms are zero, because of stochastic independence of Y_{m+1} and $\hat{y}$. Note, that $MSEP > Var(Y) = \sigma^2$.

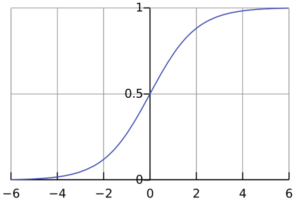
From Regression to Classification

- So far, linear regression:
  - A simple linear model.
  - Probabilistic interpretation.
  - Find optimal parameters using Maximum Likelihood Estimation.
- Can we do something similar for classification?
- $\Rightarrow$ Logistic Regression (*... it's not actually used for regression ...*)

Logistic Regression

- Binary logistic model:
  Estimate the probability of a binary response $y \in \{0, 1\}$ based on features $\boldsymbol{x}$.
- Logistic Regression is a *Generalized Linear Model*:
  Linear model is related to the response variable via a **link function**.

$$p(Y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = f(\boldsymbol{\theta}^T \boldsymbol{x})$$

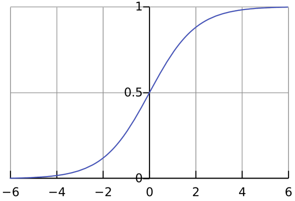


(Note: Y denotes a random variable, whereas y, $y^{(i)}, 0, 1$ denote values that the random variable can take on. If the random variable is obvious from the context, it may be omitted.)

Logistic Regression

- Recall linear regression: $p(y|\boldsymbol{x};\boldsymbol{\theta}) = N(y;\boldsymbol{\theta}^T\boldsymbol{x},\sigma^2\boldsymbol{I})$
  - Predicts $y \in \mathbb{R}$

- Classification: Outcome (per example) 0 or 1
  - Logistic sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$



  - Logistic Regression: Linear function + logistic sigmoid

$$p(Y=1|\boldsymbol{x};\boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^T\boldsymbol{x})$$

Binary Logistic Regression

Probability of different outcomes (for one example):

- Probability of positive outcome:

$$p(Y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

- Probability of negative outcome:

$$p(Y = 0|\boldsymbol{x}; \boldsymbol{\theta}) = 1 - p(Y = 1|\boldsymbol{x}; \boldsymbol{\theta})$$

Probability of a Training Example

- Probability for actual label $y^{(i)}$ given features $\boldsymbol{x}^{(i)}$
- Can be written for both labels (0 and 1) without case distinction
- Label exponentiation trick: use $x^0 = 1$

$$
\begin{aligned}
& p(Y = y^{(i)} | \boldsymbol{x}^{(\boldsymbol{i})}; \boldsymbol{\theta}) \\
&= \begin{cases} p(Y = 1 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) & \text{if } y^{(i)} = 1 \\ p(Y = 0 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) & \text{if } y^{(i)} = 0 \end{cases} \\
&= \begin{cases} p(Y = 1 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^1 p(Y = 0 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^0 & \text{if } y^{(i)} = 1 \\ p(Y = 1 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^0 p(Y = 0 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^1 & \text{if } y^{(i)} = 0 \end{cases} \\
&= p(Y = 1 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^{y^{(i)}} p(Y = 0 | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})^{1 - y^{(i)}}
\end{aligned}
$$

Binary Logistic Regression

- Conditional Negative Log-Likelihood (NLL):

$$
\begin{aligned}
NLL(\boldsymbol{\theta}) &= -\log p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\theta}) \\
&= -\log \prod_{i=1}^{m} p(Y = y^{(i)}|\boldsymbol{x}^{(\boldsymbol{i})};\boldsymbol{\theta}) \\
&= -\log \prod_{i=1}^{m} p(Y = 1|\boldsymbol{x}^{(\boldsymbol{i})};\boldsymbol{\theta})^{y^{(i)}}(1 - p(Y = 1|\boldsymbol{x}^{(\boldsymbol{i})};\boldsymbol{\theta}))^{1-y^{(i)}} \\
&= -\sum_{i=1}^{m} y^{(i)} \log \sigma(\boldsymbol{\theta}^T \boldsymbol{x}^{(\boldsymbol{i})}) + (1 - y^{(i)}) \log(1 - \sigma(\boldsymbol{\theta}^T \boldsymbol{x}^{(\boldsymbol{i})}))
\end{aligned}
$$

- No closed form solution for minimum
- Use numerical / iterative methods.
- LBFGS, Gradient descent ...

Logistic Regression

- Logistic regression: Logistic sigmoid function applied to a weighted linear combination of feature values.
- To be interpreted as the probability that the label for a specific example equals 1.
- Applying the model on test data: Predict $y^{(i)} = 1$ if

$$p(Y = 1|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) > 0.5$$

- No closed form solution for maximizing NLL, iterative methods necessary.