These are the slides of the lecture

Pattern Recognition

Winter term 2020/21

Friedrich-Alexander University of Erlangen-Nuremberg.

Erlangen, January 8, 2021
Prof. Dr.-Ing. Andreas Maier

Pattern Recognition (PR)

Prof. Dr.-Ing. Andreas Maier
Pattern Recognition Lab (CS 5), Friedrich-Alexander-Universität Erlangen-Nürnberg
Winter Term 2020/21

Logistic Regression II

Parameterization

- Until now, $F(\boldsymbol{x})$ was some arbitrary function in $\boldsymbol{x}$
  Example: $F(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \alpha^T \boldsymbol{x} + \alpha_0$ with components defined by Gaussian distributions

Parameterization

- Until now, $F(\boldsymbol{x})$ was some arbitrary function in $\boldsymbol{x}$
  Example: $F(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0$ with components defined by Gaussian distributions

- We can express a nonlinear $F(\boldsymbol{x})$ as a scalar product by lifting $\boldsymbol{x}$ into a higher dimensional space:
  Given
  $$\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2, \; \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \; \boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T, \; \alpha_0,$$
  then
  $$F(\boldsymbol{x}) = a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_0 .$$

Parameterization

- Until now, $F(\boldsymbol{x})$ was some arbitrary function in $\boldsymbol{x}$
  Example: $F(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0$ with components defined by Gaussian distributions

- We can express a nonlinear $F(\boldsymbol{x})$ as a scalar product by lifting $\boldsymbol{x}$ into a higher dimensional space:
  Given
  $$\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2, \ \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \ \boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T, \ \alpha_0,$$
  then
  $$F(\boldsymbol{x}) = a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_0 .$$

- Rewrite $F(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}'$ with $\boldsymbol{\theta}, \boldsymbol{x}' \in \mathbb{R}^6$:

Parameterization

- Until now, $F(\boldsymbol{x})$ was some arbitrary function in $\boldsymbol{x}$
  Example: $F(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\alpha}^T \boldsymbol{x} + \alpha_0$ with components defined by Gaussian distributions

- We can express a nonlinear $F(\boldsymbol{x})$ as a scalar product by lifting $\boldsymbol{x}$ into a higher dimensional space:
  Given
  $$\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2, \; \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \; \boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T, \; \alpha_0,$$
  then
  $$F(\boldsymbol{x}) = a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_0 .$$

- Rewrite $F(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}'$ with $\boldsymbol{\theta}, \boldsymbol{x}' \in \mathbb{R}^6$:
  $$\boldsymbol{\theta} = (a_{11}, a_{12} + a_{21}, a_{22}, \alpha_1, \alpha_2, \alpha_0)^T$$
  $$\boldsymbol{x}' = (x_1^2, x_1x_2, x_2^2, x_1, x_2, 1)^T$$

Parameterization (cont.)

Definition

We write the parameterized logistic function in the following:

$$g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

where $\boldsymbol{\theta}, \boldsymbol{x}$ are the lifted parameters of the original decision function F (if it was not already linear).

Log-Likelihood Function

- Let us assume the posteriors are given by

$$\begin{aligned} p(y=0|\boldsymbol{x}) &= 1-g(\boldsymbol{\theta}^T\boldsymbol{x}) \\ p(y=1|\boldsymbol{x}) &= g(\boldsymbol{\theta}^T\boldsymbol{x}) \end{aligned}$$

  where $g(\boldsymbol{\theta}^T\boldsymbol{x})$ is the sigmoid function parameterized in $\boldsymbol{\theta}$.

- The parameter vector $\boldsymbol{\theta}$ has to be estimated from a set S of m training samples:

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\} \quad .$$

Log-Likelihood Function

- Let us assume the posteriors are given by

$$\begin{aligned} p(y=0|\boldsymbol{x}) &= 1-g(\boldsymbol{\theta}^T\boldsymbol{x}) \\ p(y=1|\boldsymbol{x}) &= g(\boldsymbol{\theta}^T\boldsymbol{x}) \end{aligned}$$

  where $g(\boldsymbol{\theta}^T\boldsymbol{x})$ is the sigmoid function parameterized in $\boldsymbol{\theta}$.

- The parameter vector $\boldsymbol{\theta}$ has to be estimated from a set S of m training samples:

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \ldots, (\boldsymbol{x}_m, y_m)\} \quad .$$

- Method of choice: Maximum Likelihood Estimation

Log-Likelihood Function (cont.)

Before we work on the formulas of the ML-estimator, we rewrite the posteriors using Bernoulli probability:

$$p(y|\boldsymbol{x}) \quad =$$

Log-Likelihood Function (cont.)

Before we work on the formulas of the ML-estimator, we rewrite the posteriors using Bernoulli probability:

$$p(y|\boldsymbol{x}) \quad = \quad g(\boldsymbol{\theta}^T\boldsymbol{x})^y(1-g(\boldsymbol{\theta}^T\boldsymbol{x}))^{1-y}$$

which shows the great benefit of the chosen notation for class numbers.

Log-Likelihood Function (cont.)

Now we can compute the log-likelihood function
(assuming that the training samples are mutually independent):

$$\mathscr{L}(\boldsymbol{\theta}) = \log\left(\prod_{i=1}^{m} p(y_i|\boldsymbol{x}_i)\right)$$

Log-Likelihood Function (cont.)

Now we can compute the log-likelihood function
(assuming that the training samples are mutually independent):

$$
\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}) &= \log\left(\prod_{i=1}^{m} p(y_i|\boldsymbol{x}_i)\right) \\
&= \sum_{i=1}^{m} \log\left(g(\boldsymbol{\theta}^T\boldsymbol{x}_i)^{y_i}\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)^{1-y_i}\right)
\end{aligned}
$$

Log-Likelihood Function (cont.)

Now we can compute the log-likelihood function
(assuming that the training samples are mutually independent):

$$
\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}) &= \log\left(\prod_{i=1}^{m} p(y_i|\boldsymbol{x}_i)\right)\\
&= \sum_{i=1}^{m} \log\left(g(\boldsymbol{\theta}^T\boldsymbol{x}_i)^{y_i}\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)^{1-y_i}\right)\\
&= \sum_{i=1}^{m} \left(y_i \log g(\boldsymbol{\theta}^T\boldsymbol{x}_i) + (1-y_i)\log\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)\right)
\end{aligned}
$$

Log-Likelihood Function (cont.)

Simplification of the log-likelihood function:

$$\mathscr{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left(y_i \log g(\boldsymbol{\theta}^T \boldsymbol{x}_i) + (1 - y_i) \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right)\right)$$

Log-Likelihood Function (cont.)

Simplification of the log-likelihood function:

$$\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(y_i \log g(\boldsymbol{\theta}^T \boldsymbol{x}_i) + (1 - y_i) \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right)\right) \\
&= \sum_{i=1}^{m} \left(y_i \log \frac{e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} + (1 - y_i) \log \frac{1}{1 + e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} \right)
\end{aligned}$$

Log-Likelihood Function (cont.)

Simplification of the log-likelihood function:

$$
\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(y_i \log g(\boldsymbol{\theta}^T \boldsymbol{x}_i) + (1 - y_i) \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right)\right) \\
&= \sum_{i=1}^{m} \left(y_i \log \frac{e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} + (1 - y_i) \log \frac{1}{1 + e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} \right) \\
&= \sum_{i=1}^{m} \left(y_i \boldsymbol{\theta}^T \boldsymbol{x}_i + \log \frac{1}{1 + e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} \right)
\end{aligned}
$$

Log-Likelihood Function (cont.)

Simplification of the log-likelihood function:

$$\begin{aligned}
\mathscr{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(y_i \log g(\boldsymbol{\theta}^T \boldsymbol{x}_i) + (1-y_i) \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right)\right) \\
&= \sum_{i=1}^{m} \left(y_i \log \frac{e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}}{1+e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} + (1-y_i) \log \frac{1}{1+e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} \right) \\
&= \sum_{i=1}^{m} \left(y_i \boldsymbol{\theta}^T \boldsymbol{x}_i + \log \frac{1}{1+e^{\boldsymbol{\theta}^T \boldsymbol{x}_i}} \right) \\
&= \sum_{i=1}^{m} \left(y_i \boldsymbol{\theta}^T \boldsymbol{x}_i + \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right)\right)
\end{aligned}$$

Log-Likelihood Function (cont.)

Notes for the expert:

- The negative of the log-likelihood function is the cross entropy of y and $g(\boldsymbol{\theta}^T \boldsymbol{x})$.
- The negative of the log-likelihood function is a convex function.

Pattern Recognition Lab

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

Maximization of the log-likelihood function

- The log-likelihood function is concave.
- We use the ☞ Newton-Raphson algorithm to solve the unconstrained optimization problem:

For the $(k+1)$-st iteration step, we get:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \left(\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \mathscr{L}\left(\boldsymbol{\theta}^{(k)}\right) \right)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} \mathscr{L}\left(\boldsymbol{\theta}^{(k)}\right)$$

Note: If you write the Newton-Raphson iteration in matrix form, you will end up with a weighted least squares iteration scheme.

Newton-Raphson Iteration

Taylor's Theorem:

Approximation of a k-times differentiable function $f(x)$ around a given point x_0:

$$f(x_0+h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2!}h^2 + \ldots + \frac{f^{(k)}(x_0)}{k!}h^k + r_k(x_0+h)h^k, \quad \lim_{h\to 0} r_k(x_0+h) = 0$$

Newton-Raphson Iteration

Taylor's Theorem:

Approximation of a k-times differentiable function $f(x)$ around a given point x_0:

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(x_0)}{2!}h^2 + \ldots + \frac{f^{(k)}(x_0)}{k!}h^k + r_k(x_0 + h)h^k, \quad \lim_{h \to 0} r_k(x_0 + h) = 0$$

Second order Taylor polynomial:

$$f(x_0 + h) \approx f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2$$

Newton-Raphson Iteration (cont.)

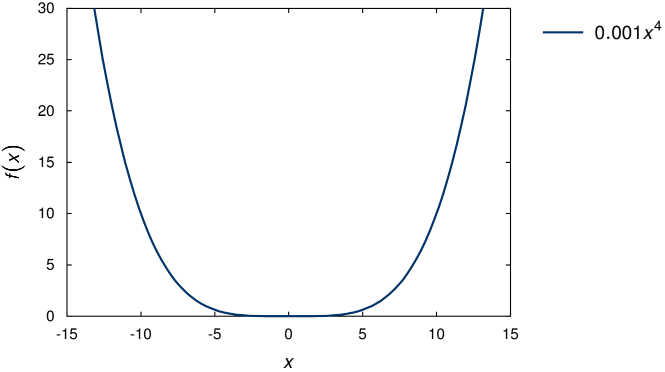
Extremum:

$$f'(x_0+h) = f'(x_0) + f''(x_0)h \overset{!}{=} 0$$

$$\hat{h} = -\frac{f'(x_0)}{f''(x_0)}$$

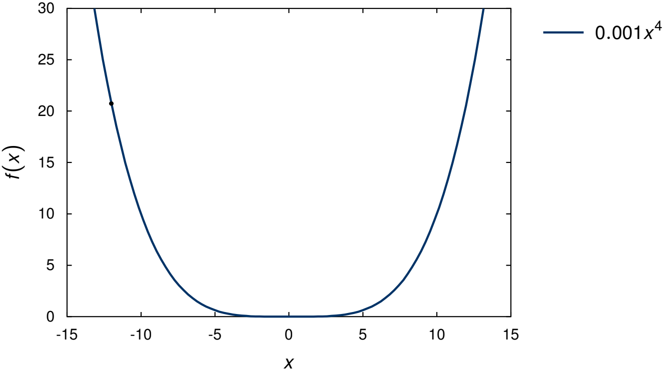
$$x_1 = x_0 + \hat{h} = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

Newton-Raphson Iteration (cont.)

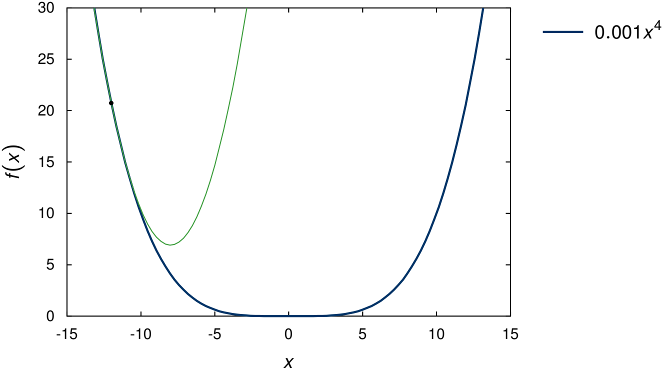


Newton-Raphson Iteration (cont.)

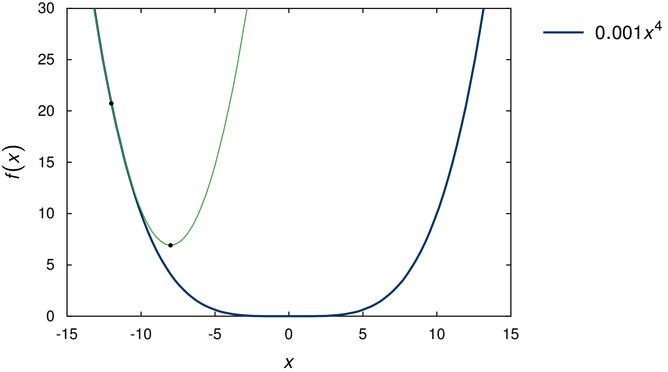
Newton-Raphson Iteration (cont.)



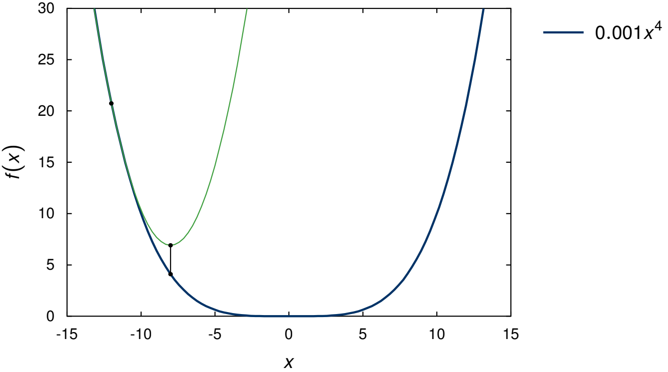
Newton-Raphson Iteration (cont.)



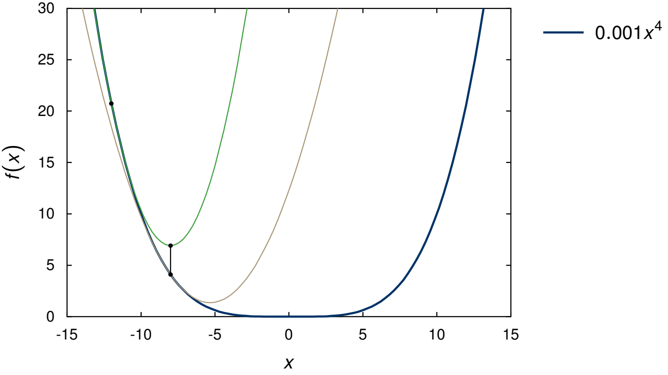
Newton-Raphson Iteration (cont.)



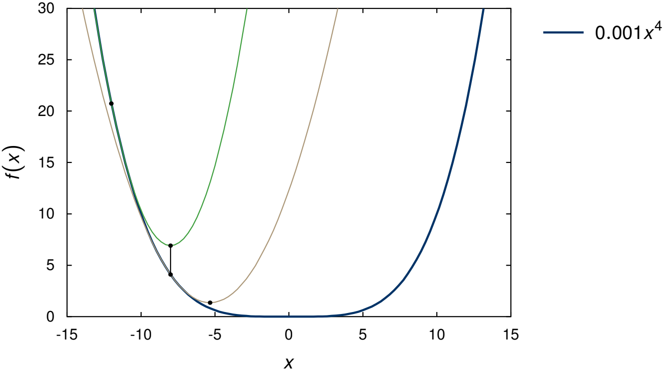
Newton-Raphson Iteration (cont.)



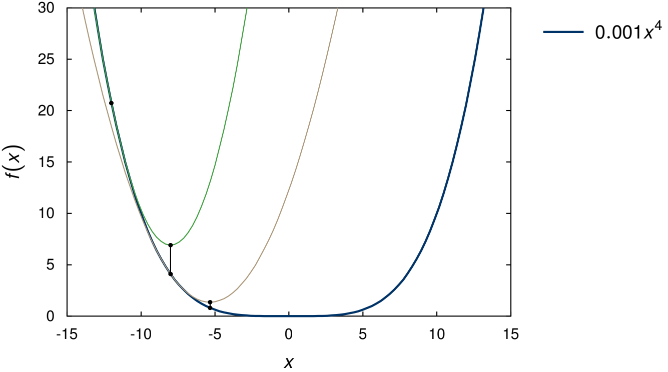
Newton-Raphson Iteration (cont.)



Newton-Raphson Iteration (cont.)



Newton-Raphson Iteration (cont.)



Gradient of the Log-Likelihood Function

The gradient:

$$\frac{\partial}{\partial \theta_j} \mathscr{L}(\boldsymbol{\theta}) \quad = \quad \frac{\partial}{\partial \theta_j} \left(\sum_{i=1}^{m} \left(y_i \boldsymbol{\theta}^T \boldsymbol{x}_i + \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i) \right) \right) \right)$$

Gradient of the Log-Likelihood Function

The gradient:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} \mathscr{L}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j} \left(\sum_{i=1}^{m} \left(y_i \boldsymbol{\theta}^T \boldsymbol{x}_i + \log \left(1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i) \right) \right) \right) \\
&= \sum_{i=1}^{m} \left(y_i x_{i,j} - \frac{1}{1 - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)} \frac{\partial}{\partial \theta_j} g(\boldsymbol{\theta}^T \boldsymbol{x}_i) \right)
\end{aligned}
$$

Gradient of the Log-Likelihood Function

The gradient:

$$\begin{aligned}\frac{\partial}{\partial \theta_j}\mathscr{L}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j}\left(\sum_{i=1}^{m}\left(y_i\boldsymbol{\theta}^T\boldsymbol{x}_i + \log\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)\right)\right)\\ &= \sum_{i=1}^{m}\left(y_i x_{i,j} - \frac{1}{1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)}\frac{\partial}{\partial \theta_j}g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)\end{aligned}$$

Now we use the derivative of the sigmoid function and get

$$\frac{\partial}{\partial \theta_j}\mathscr{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m}\left(y_i x_{i,j} - \frac{1}{1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)}g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)x_{i,j}\right)$$

Gradient of the Log-Likelihood Function

The gradient:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j}\mathscr{L}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_j}\left(\sum_{i=1}^{m}\left(y_i\boldsymbol{\theta}^T\boldsymbol{x}_i + \log\left(1 - g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)\right)\right) \\
&= \sum_{i=1}^{m}\left(y_i x_{i,j} - \frac{1}{1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)}\frac{\partial}{\partial \theta_j}g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)
\end{aligned}$$

Now we use the derivative of the sigmoid function and get

$$\begin{aligned}
\frac{\partial}{\partial \theta_j}\mathscr{L}(\boldsymbol{\theta}) &= \sum_{i=1}^{m}\left(y_i x_{i,j} - \frac{1}{1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)}g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)x_{i,j}\right) \\
&= \sum_{i=1}^{m}\left(y_i - g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)x_{i,j}
\end{aligned}$$

where $x_{i,j}$ is the j-th component of the i-th training feature vector.

Gradient of the Log-Likelihood Function (cont.)

Finally, we have a quite simple gradient:

$$\frac{\partial}{\partial \theta_j}\mathscr{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left(y_i - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right) x_{i,j}$$

where $x_{i,j}$ is the j–th component of the i–th training feature vector.

Or in vector notation:

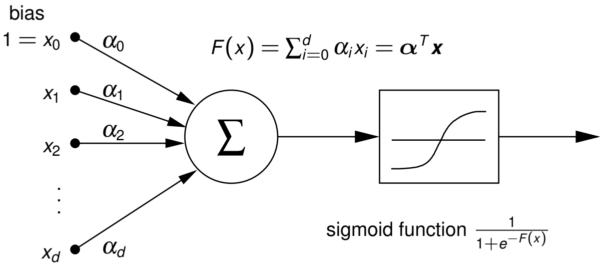
$$\frac{\partial}{\partial \boldsymbol{\theta}}\mathscr{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left(y_i - g(\boldsymbol{\theta}^T \boldsymbol{x}_i)\right) \boldsymbol{x}_i$$

Hessian of the Log-Likelihood Function

- The Newton-Raphson algorithm requires the Hessian matrix.
- Remember the derivative of the sigmoid function!

$$\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\mathscr{L}(\boldsymbol{\theta}) \quad = \quad -\sum_{i=1}^{m} g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\left(1-g(\boldsymbol{\theta}^T\boldsymbol{x}_i)\right)\boldsymbol{x}_i\boldsymbol{x}_i^T$$

Perceptron and Logistic Regression



Lessons Learned

- Posteriors can be rewritten in terms of a logistic function.
- Given the decision boundary $F(\boldsymbol{x}) = 0$, we can write down the posterior $p(y|\boldsymbol{x})$ right away.
- Decision boundary for normally distributed feature vectors for each class is a quadratic function.
- If Gaussians share the same covariances, the decision boundary is a linear function.

Pattern Recognition Lab

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Next Time in

Pattern Recognition

Further Readings

- T. Hastie, R. Tibshirani, and J. Friedman:
  The Elements of Statistical Learning –
  Data Mining, Inference, and Prediction,
  2nd edition, Springer, New York, 2009.

- David W. Hosmer, Stanley Lemeshow:
  Applied Logistic Regression, 2nd Edition,
  John Wiley & Sons, Hoboken, 2000.

Comprehensive Questions (cont.)

- How can a nonlinear function be written as a scalar product?

Comprehensive Questions (cont.)

- How can a nonlinear function be written as a scalar product?
- What is the objective function for the ML-estimation of the logistic regression parameters?

Comprehensive Questions (cont.)

- How can a nonlinear function be written as a scalar product?
- What is the objective function for the ML-estimation of the logistic regression parameters?
- What is the difference between a gradient descent and Newton-Raphson numerical optimization scheme?

Comprehensive Questions (cont.)

- How can a nonlinear function be written as a scalar product?
- What is the objective function for the ML-estimation of the logistic regression parameters?
- What is the difference between a gradient descent and Newton-Raphson numerical optimization scheme?
- What is the parameter update rule for the logistic regression parameters using the Newton-Raphson scheme?