Linear Classification

Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Review: Supervised Learning

- Review: in **supervised learning**, the goal is to make predictions about objects

Input: an **instance** (object) $\mathbf{x}$ that lives in an instance space $\mathcal{X}$

$$\underset{\in \mathcal{X}}{\mathbf{x}} \mapsto \underset{\in \mathcal{Y}}{y}$$

Output: a **label** or **target** y that lives in a target space $\mathcal{Y}$

- To obtain predictions, we are looking for a **model** f that produces a prediction $f(\mathbf{x}) \in \mathcal{Y}$ for an input instance $\mathbf{x}$

$$f : \mathcal{X} \to \mathcal{Y}$$

Input: instance $\mathbf{x}$

$$\mathbf{x} \mapsto f(\mathbf{x})$$

Output: prediction $f(\mathbf{x})$

- Model will be inferred from **training data**: a set of instances with observed targets

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

Training instances $\mathbf{x}_n \in \mathcal{X}$: observed objects in training data, for example flowers, images of digits, or emails

Observed labels or targets $y \in \mathcal{Y}$ in training data, for example classes of flowers, digits 0...9, or spam/legitimate classifications

Review: Classification Problems

- Review: in **classification**, the target space is a set of discrete classes:

$$\mathcal{Y} = \{c_1, \ldots, c_T\}$$

Classes can be for example
- {„spam", "legitimate"} for the spam filtering problem
- {„Iris Versicolor", "Iris Setosa", „Iris Virginica"} for the Iris classification problem
- {0,1,2,3,4,5,6,7,8,9} for the digit classification problem

- For mathematical convenience and without loss of generality, we can assume that classes are simply numbers: $\mathcal{Y} = \{1, \ldots, T\}$ (simply map original classes to numbers)

- Classification problems with two classes are called **binary classification** problems. Here, classes are usually encoded as $\mathcal{Y} = \{0,1\}$ for mathematical convenience (sometimes also as $\mathcal{Y} = \{-1,1\}$)

- Classification problems with more than two classes are called **multiclass classification**

Classification Example: Iris Data Set

- An example for a (toy) classification problem is the „Iris“ data set
- Three different species of Iris plants:

Iris versicolor

Iris setosa

Iris virginica

- An instance represents a particular flower by its petal and sepal width and heights, resulting in four features:

instance representation

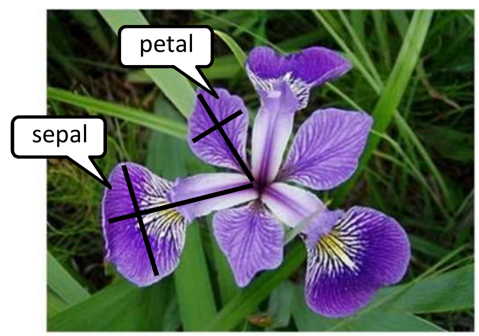
$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \in \mathbb{R}^4$$

sepal length

sepal width

petal length

petal width



„Sepal“ and „petal“ refer to different parts of the Iris flower

Example: Iris Data Set

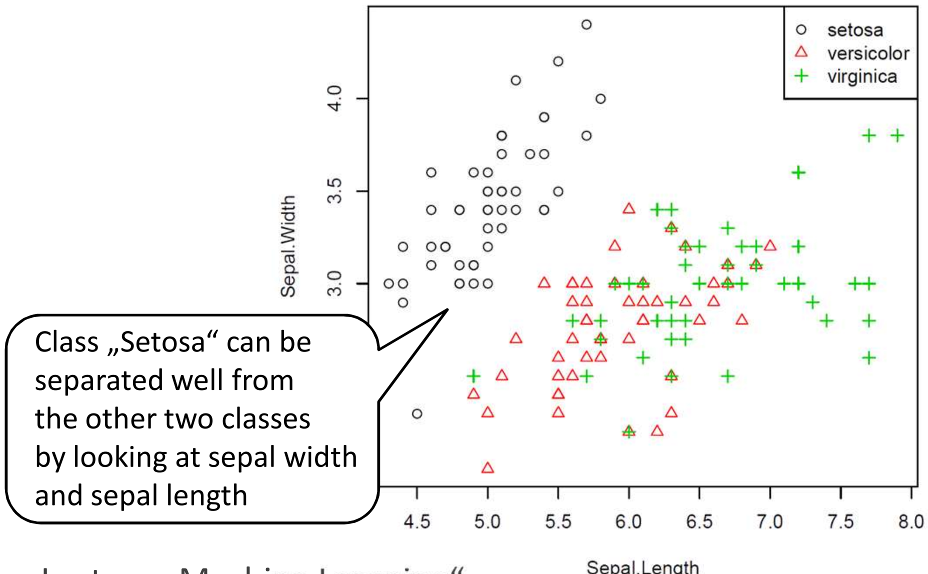
- The Iris data set contains 150 instances (50 per class)

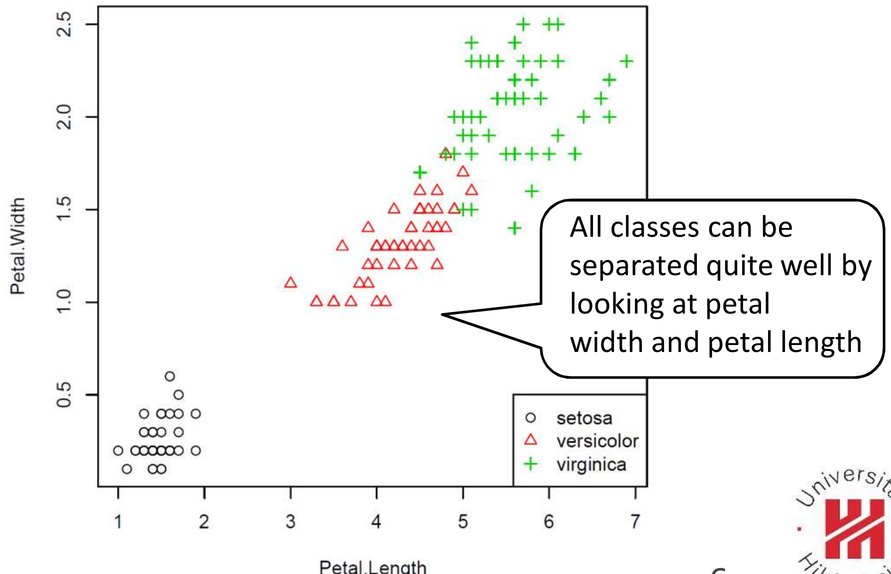
		$\mathbf{x}$				y
		Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
$(\mathbf{x}_1, y_1)$	1	5.10	3.50	1.40	0.20	setosa
$(\mathbf{x}_2, y_2)$	2	4.90	3.00	1.40	0.20	setosa
	3	4.70	3.20	1.30	0.20	setosa
	⋮	⋮	⋮	⋮	⋮	
	51	7.00	3.20	4.70	1.40	versicolor
	52	6.40	3.20	4.50	1.50	versicolor
	53	6.90	3.10	4.90	1.50	versicolor
	⋮	⋮	⋮	⋮	⋮	
	101	6.30	3.30	6.00	2.50	virginica
	⋮	⋮	⋮	⋮	⋮	
$(\mathbf{x}_N, y_N)$	150	5.90	3.00	5.10	1.80	virginica

Class information, could also be represented by number {1,2,3}

Visualizing Data: Scatter Plots

- For a given classification data set such as the Iris data set, we can visualize the distribution of instances together with their class labels in so-called scatter plots
  - To construct a scatter plot, we first need to choose two features x_i, x_j from the set of features $\{x_1,\ldots,x_M\}$ that describe an instance in the data set
  - A scatter plot is a 2D-plot, with axes x_i and x_j, in which each instance $\mathbf{x}_n$ is plotted as a colored marker (at position given by its feature values $x_{n,i}$, $x_{n,j}$) with the color corresponding to the class
  - From scatter plot, can infer which features separate classes well

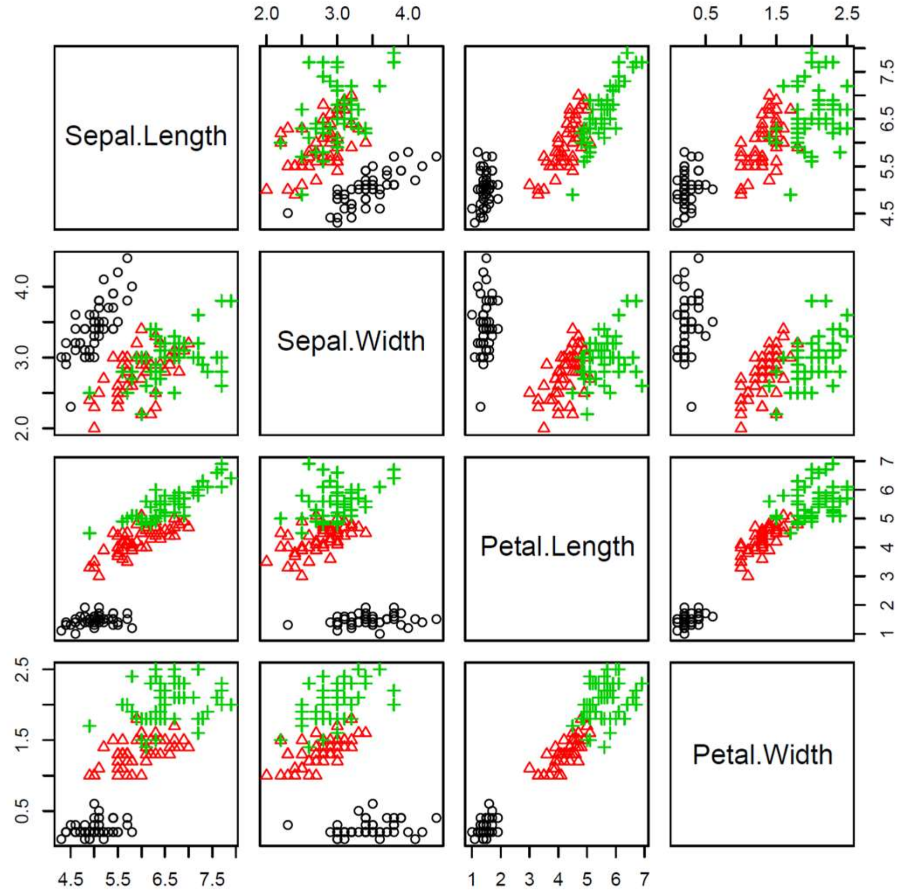




Visualizing Data: Scatter Plots

- For the Iris data set, there are overall 12 pairs of features x_i, x_j to look at
- Some feature pairs are more informative for separating classes than others

All feature pairs can be visualized in matrix (quadratic in number of features)



Binary Classification

- For the moment, we will look at binary classification problems:
  - Given a training data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ with $\mathbf{x}_n \in \mathbb{R}^M$ and $y_n \in \{0,1\}$
  - Find a model $f_{\boldsymbol{\theta}} : \mathbb{R}^M \to \{0,1\}$
- The $(\mathbf{x}_n, y_n)$ with $y_n = 0$ are also called negative examples
- The $(\mathbf{x}_n, y_n)$ with $y_n = 1$ are also called positive examples
- For example, could classify Iris Setosa versus other species as a binary problem derived from the Iris data set

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species setosa
1	5.10	3.50	1.40	0.20	1
2	4.90	3.00	1.40	0.20	1
3	4.70	3.20	1.30	0.20	1
⋮	⋮	⋮	⋮	⋮	
51	7.00	3.20	4.70	1.40	0
52	6.40	3.20	4.50	1.50	0
53	6.90	3.10	4.90	1.50	0
⋮	⋮	⋮	⋮	⋮	
101	6.30	3.30	6.00	2.50	0
⋮	⋮	⋮	⋮	⋮	
150	5.90	3.00	5.10	1.80	0

Binary label: Iris Setosa or other species

Binary Classification via Linear Regression?

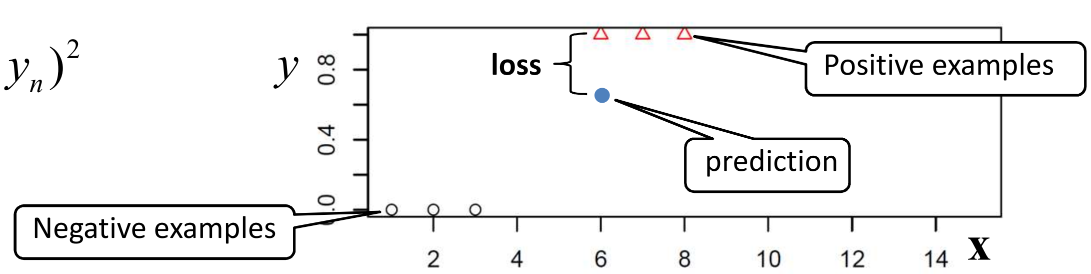
- **First idea**: can we use a linear regression model to perform classification?
  - Predict targets with a linear regression:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \theta_1 x_1 + \ldots + \theta_M x_M = \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}$$

  - Train the model using squared loss:

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}(f_{\boldsymbol{\theta}}(\mathbf{x}_n) - y_n)^2$$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$



- This is not a good model
  - The function $f_{\boldsymbol{\theta}}(\mathbf{x})$ returns continuous values, including values larger than one or smaller than zero. What do those mean?
  - Squared error as a loss function does not capture the goal in classification well

Logistic Regression Model

- **Better idea**: use a linear model, but adapt it for classification
- Instead of predicting the targets $y=0$ or $y=1$ directly, predict a conditional probability:

Depends on input $\mathbf{x}$ and model parameters $\boldsymbol{\theta}$

Probability according to the model $f_{\boldsymbol{\theta}}$ that the target for input $\mathbf{x}$ is $y=1$

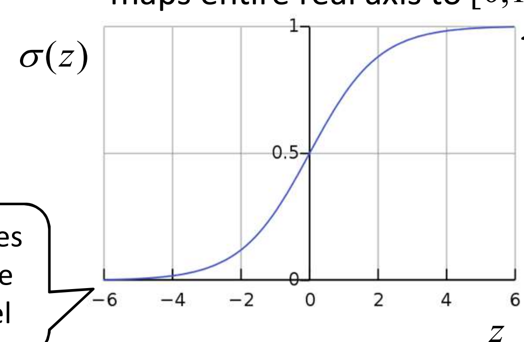
$$p(y=1 \mid \mathbf{x}, \boldsymbol{\theta}) \in [0,1]$$

- To model this probability, use a linear model but transform the output of the linear model (which is any real number) to the interval $[0,1]$ using the so-called sigmoid function σ :

Define $p(y=1 \mid \mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^{\mathrm{T}} \boldsymbol{\theta})$

where $\sigma(z) = \dfrac{e^z}{1+e^z}$

„Squashing function" σ maps entire real axis to $[0,1]$



Probability approaches one for large positive values of linear model

Probability approaches zero for large negative values of linear model

Target Space for Logistic Regression Model

- The logistic regression model produces continuous class probabilities, not classification labels:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\mathbf{x}^T \boldsymbol{\theta}) = p(y = 1 \mid \mathbf{x}, \boldsymbol{\theta})$$

- For the function $f_{\boldsymbol{\theta}}$ we therefore have $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow [0,1]$ rather than $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \{0,1\}$
- To still have $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$, we redefine the target space $\mathcal{Y}$ for classification as follows:
  – the training labels $y_1, \ldots, y_n$ indicate the probability that for training instance the true label is $y = 1$
  – The target space is therefore $\mathcal{Y} = [0,1]$
  – Because we know the true labels for the trainingt instances, a training label is still $y_n = 1$ for a positive example and $y_n = 0$ for a negative examples, but we interpret these numbers as the probability for the positive class

- From the class probabilities produced by the model, we can make a classification decision by predicting the positive class if $p(y = 1 \mid \mathbf{x}, \boldsymbol{\theta}) \geq 0.5$ and the negative class if $p(y = 1 \mid \mathbf{x}, \boldsymbol{\theta}) < 0.5$

Loss Function for Logistic Regression?

- To train a logistic regression model from data, we could look for a loss function as discussed for the linear regression model:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_n), y_n)$$

- Instead, we will take a slightly different approach, which is based on the probabilistic nature of the logistic regression model

- **Review (introductory lecture)**: we assume that the training data $\mathcal{D} = \{(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_N, y_N)\}$ is drawn by independently drawing instances $\mathbf{x}_1,\ldots,\mathbf{x}_N$ from a distribution over inputs,

$$\mathbf{x}_n \sim p(\mathbf{x}),$$

and by independently drawing outputs from a conditional distribution $p(y \mid \mathbf{x}_n)$

$$y_n \sim p(y \mid \mathbf{x}_n)$$

Likelihood for Logistic Regression

- Logistic regression is a probabilistic model: as a probabilistic criterion for how well the model fits the data, we can use the **likelihood**, that is, the probability of the data according to the model

- Logistic regression computes a probability $p(y=1 \mid \mathbf{x}, \boldsymbol{\theta})$, and this of course also gives us a probability $p(y=0 \mid \mathbf{x}, \boldsymbol{\theta})$ by $p(y=0 \mid \mathbf{x}, \boldsymbol{\theta}) = 1 - p(y=1 \mid \mathbf{x}, \boldsymbol{\theta})$
- Logistic regression thereby gives us a model $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ for the assumed true conditional distribution $p(y \mid \mathbf{x})$ from which the training labels have been drawn
- Logistic regression does not give us any model for $p(\mathbf{x})$

- We can measure how well a logistic regression model fits the training data by the **conditional likelihood**

$$p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}) = \prod_{n=1}^{N} \underbrace{p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})}_{\text{Obtained from logistic regression model}}$$

Independence assumption: targets for instances are independently drawn from a (modeled) conditional distribution $p(y \mid \mathbf{x}_n, \boldsymbol{\theta})$ given the instance $\mathbf{x}_n$

Likelihood for Logistic Regression

- According to the definition of the logistic regression model, the likelihood can be derived in more details as

$$p(y_1,\dots,y_N \mid \mathbf{x}_1,\dots,\mathbf{x}_N,\boldsymbol{\theta}) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n,\boldsymbol{\theta})$$

$$= \prod_{n=1}^{N} p(y=1 \mid \mathbf{x}_n,\boldsymbol{\theta})^{y_n}\, p(y=0 \mid \mathbf{x}_n,\boldsymbol{\theta})^{(1-y_n)}$$

$$= \prod_{n=1}^{N} \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}\,(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}$$

If label is positive, the probability for the label according to the model is $p(y=1 \mid \mathbf{x}_n,\boldsymbol{\theta})$, otherwise it is $p(y=0 \mid \mathbf{x}_n,\boldsymbol{\theta})$.
The exponents y_n and $(1-y_n)$ „select“ the right term because if e.g. $y_n = 0$, then $p(y=1 \mid \mathbf{x}_n,\boldsymbol{\theta})^{y_n} = 1$ and disappears from the product.

Plugging in definition of the model

- The likelihood characterizes how well the model fits the data: if the likelihood is high, the predictions of the model are in agreement with the observed training labels
- Model parameters are thus chosen according to **maximum (conditional) likelihood**:

Optimization Problem:

$$\boxed{\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(y_1,\dots,y_N \mid \mathbf{x}_1,\dots,\mathbf{x}_N,\boldsymbol{\theta})}$$

Log-Likelihood For Logistic Regression

- Instead of maximizing the likelihood, we can equivalently maximize the logarithmic likelihood (usually called **log-likelihood**):

The logarithm is strictly monotone, therefore the arg max is the same

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(y_1,\dots,y_N \mid \mathbf{x}_1,\dots,\mathbf{x}_N,\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \log p(y_1,\dots,y_N \mid \mathbf{x}_1,\dots,\mathbf{x}_N,\boldsymbol{\theta})\end{aligned}$$

- Unless noted otherwise, log refers to the natural logarithm
- For the logistic regression model, the log-likelihood can be derived as follows

Plugged in likelihood expression from last slide

$$\begin{aligned}\log p(y_1,\dots,y_N \mid \mathbf{x}_1,\dots,\mathbf{x}_N,\boldsymbol{\theta}) &= \log \prod_{n=1}^{N} \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)} \\ &= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\ &= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}\right) + \log\left((1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\ &= \sum_{n=1}^{N} y_n \log\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))\end{aligned}$$

Log-Likelihood For Logistic Regression

- Instead of maximizing the likelihood, we can equivalently maximize the logarithmic likelihood (usually called **log-likelihood**):

The logarithm is strictly monotone, therefore the arg max is the same

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta})\end{aligned}$$

- Unless noted otherwise, log refers to the natural logarithm
- For the logistic regression model, the log-likelihood can be derived as follows

$$\begin{aligned}\log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) &= \log \prod_{n=1}^{N} \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)} \\ &= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\ &= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}\right) + \log\left((1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\ &= \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))\end{aligned}$$

logarithm of product is sum of logarithms

Log-Likelihood For Logistic Regression

- Instead of maximizing the likelihood, we can equivalently maximize the logarithmic likelihood (usually called **log-likelihood**):

The logarithm is strictly monotone, therefore the arg max is the same

$$
\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta})
\end{aligned}
$$

- Unless noted otherwise, log refers to the natural logarithm
- For the logistic regression model, the log-likelihood can be derived as follows

$$
\begin{aligned}
\log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) &= \log \prod_{n=1}^{N} \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)} \\
&= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\
&= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}\right) + \log\left((1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\
&= \sum_{n=1}^{N} y_n \log\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))
\end{aligned}
$$

logarithm of product is sum of logarithms

Universität Hildesheim

Log-Likelihood For Logistic Regression

- Instead of maximizing the likelihood, we can equivalently maximize the logarithmic likelihood (usually called **log-likelihood**):

The logarithm is strictly monotone, therefore the arg max is the same

$$
\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta})
\end{aligned}
$$

- Unless noted otherwise, log refers to the natural logarithm
- For the logistic regression model, the log-likelihood can be derived as follows

$$
\begin{aligned}
\log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) &= \log \prod_{n=1}^{N} \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)} \\
&= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\
&= \sum_{n=1}^{N} \log\left(\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta})^{y_n}\right) + \log\left((1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))^{(1-y_n)}\right) \\
&= \sum_{n=1}^{N} y_n \log\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))
\end{aligned}
$$

$\boxed{\log a^b = b\log a}$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))$$

Plugging in definition $\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) = \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} + (1-y_n)\log(1-\frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\log(e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)(-\log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}))$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) + y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))$$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} + (1-y_n)\log(1-\frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\log(e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$\log \frac{a}{b} = \log a - \log b$

Transform to common denominator

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)(-\log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}))$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) + y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))$$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} + (1-y_n)\log\left(1-\frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}\right)$$

$$= \sum_{n=1}^{N} y_n \left(\log(e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right) + (1-y_n)\log\left(\frac{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}\right)$$

$$= \sum_{n=1}^{N} y_n \left(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right) + (1-y_n)\log\left(\frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}\right)$$

$$\boxed{\log(e^{z}) = z}$$

$$= \sum_{n=1}^{N} y_n \left(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right) + (1-y_n)\left(-\log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) + y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1-y_n)\log(1-\sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))$$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} + (1-y_n)\log(1-\frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\log(e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)\log(\frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1-y_n)(-\log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}))$$

$$\boxed{\log(\frac{1}{z}) = -\log z}$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) + y_n \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}) + (1 - y_n) \log(1 - \sigma(\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}))$$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} + (1 - y_n) \log(1 - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\log(e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1 - y_n) \log(\frac{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1 - y_n) \log(\frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) + (1 - y_n)(-\log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}))$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - y_n \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})) - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}) + y_n \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

multiplying out

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})$$

Log-Likelihood For Logistic Regression

- Derivation continued from last slide (first line simply copied)

$$\ldots = \sum_{n=1}^{N} y_n \log \sigma(\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}) + (1 - y_n) \log(1 - \sigma(\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}))$$

$$= \sum_{n=1}^{N} y_n \log \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} + (1 - y_n) \log(1 - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\log(e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})) + (1 - y_n) \log(\frac{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})) + (1 - y_n) \log(\frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}})$$

$$= \sum_{n=1}^{N} y_n (\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})) + (1 - y_n)(-\log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}))$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - y_n \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})) - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) + y_n \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})$$

terms cancel

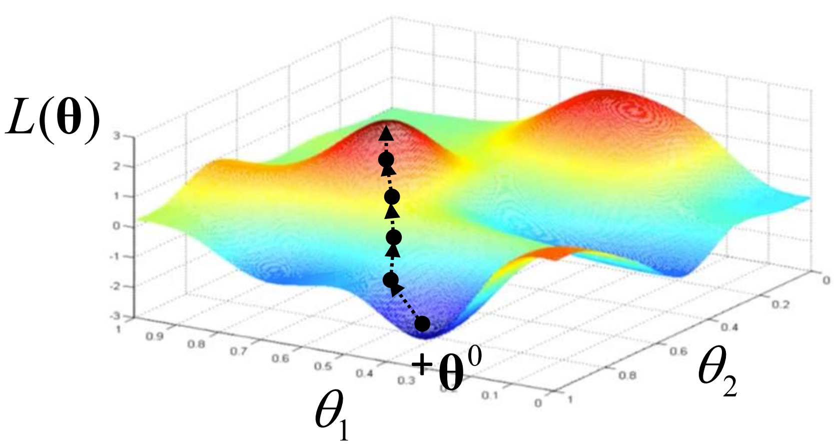
$$= \sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}})$$

Gradient Ascent

- For learning a logistic regression model, we need to solve the optimization problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L_{cll}(\boldsymbol{\theta}) \qquad\qquad L_{cll}(\boldsymbol{\theta}) := \log p(y_1,...,y_N \mid \mathbf{x}_1,...,\mathbf{x}_N, \boldsymbol{\theta})$$

- This can be done with a gradient ascent algorithm, similarly as we used gradient descent for finding a model with minimal loss
  - To solve the maximization problem, need to take small steps into the direction of the positive (rather than negative) gradient
  - Otherwise, exactly the same as gradient descent



Gradient ascent algorithm

1. $\boldsymbol{\theta}_0 = \text{randomInitialization}()$
2. for $i = 0,...,i_{max}$:
3. $\quad \boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \eta \nabla L_{cll}(\boldsymbol{\theta}_i)$
4. $\quad$ if $L_{cll}(\boldsymbol{\theta}_{i+1}) - L_{cll}(\boldsymbol{\theta}_i) < \epsilon$:
5. $\qquad$ return $\boldsymbol{\theta}_{i+1}$
6. raise Exception("Not converged in i_{max} iterations")

Gradient For Logistic Regression

- We can explicitly derive the gradient for the logistic regression model as follows

$$\nabla L_{cll}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

Plugged in log likelihood expression from above

$$= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \left(y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \frac{\partial}{\partial \boldsymbol{\theta}} \left(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \mathbf{x}_n$$

$$= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \right)$$

$$= \sum_{n=1}^{N} \mathbf{x}_n (y_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n))$$

Gradient For Logistic Regression

- We can explicitly derive the gradient for the logistic regression model as follows

$$
\begin{aligned}
\nabla L_{cll}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \boldsymbol{\theta}}\left(\sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right) \\
&= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}}\left(y_n \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta} - \log(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}})\right) \\
&= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} \frac{\partial}{\partial \boldsymbol{\theta}}\left(1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}\right) \\
&= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}} e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}} \mathbf{x}_n \\
&= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - \frac{e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}{1+e^{\mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}}}\right) \\
&= \sum_{n=1}^{N} \mathbf{x}_n (y_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n))
\end{aligned}
$$

Gradient of sum
is sum of gradients

Gradient For Logistic Regression

- We can explicitly derive the gradient for the logistic regression model as follows

$$\nabla L_{cll}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

$$= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \left(y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \frac{\partial}{\partial \boldsymbol{\theta}} \left(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \right)$$

Derivative of vector product: $\frac{\partial}{\partial \mathbf{v}} \mathbf{u}^{\mathrm{T}} \mathbf{v} = \mathbf{u}$

Chain rule: outer derivate $\frac{\partial}{\partial x} \log x = \frac{1}{x}$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \mathbf{x}_n$$

$$= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \right)$$

$$= \sum_{n=1}^{N} \mathbf{x}_n (y_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n))$$

Gradient For Logistic Regression

- We can explicitly derive the gradient for the logistic regression model as follows

$$\nabla L_{cll}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

$$= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \left(y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \frac{\partial}{\partial \boldsymbol{\theta}} \left(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \right)$$

$$= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \mathbf{x}_n$$

$$= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \right)$$

$$= \sum_{n=1}^{N} \mathbf{x}_n (y_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n))$$

Chain rule: outer derivative $\frac{\partial}{\partial z} e^z = e^z$,

inner derivate $\frac{\partial}{\partial \boldsymbol{\theta}} \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} = \mathbf{x}_n$

Gradient For Logistic Regression

- We can explicitly derive the gradient for the logistic regression model as follows

$$
\begin{aligned}
\nabla L_{cll}(\boldsymbol{\theta}) &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{n=1}^{N} y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right) \\
&= \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \left(y_n \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta} - \log(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}) \right) \\
&= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \frac{\partial}{\partial \boldsymbol{\theta}} \left(1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \right) \\
&= \sum_{n=1}^{N} y_n \mathbf{x}_n - \frac{1}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}} \mathbf{x}_n \\
&= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}} \right) \\
&= \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - f_{\boldsymbol{\theta}}(\mathbf{x}_n) \right)
\end{aligned}
$$

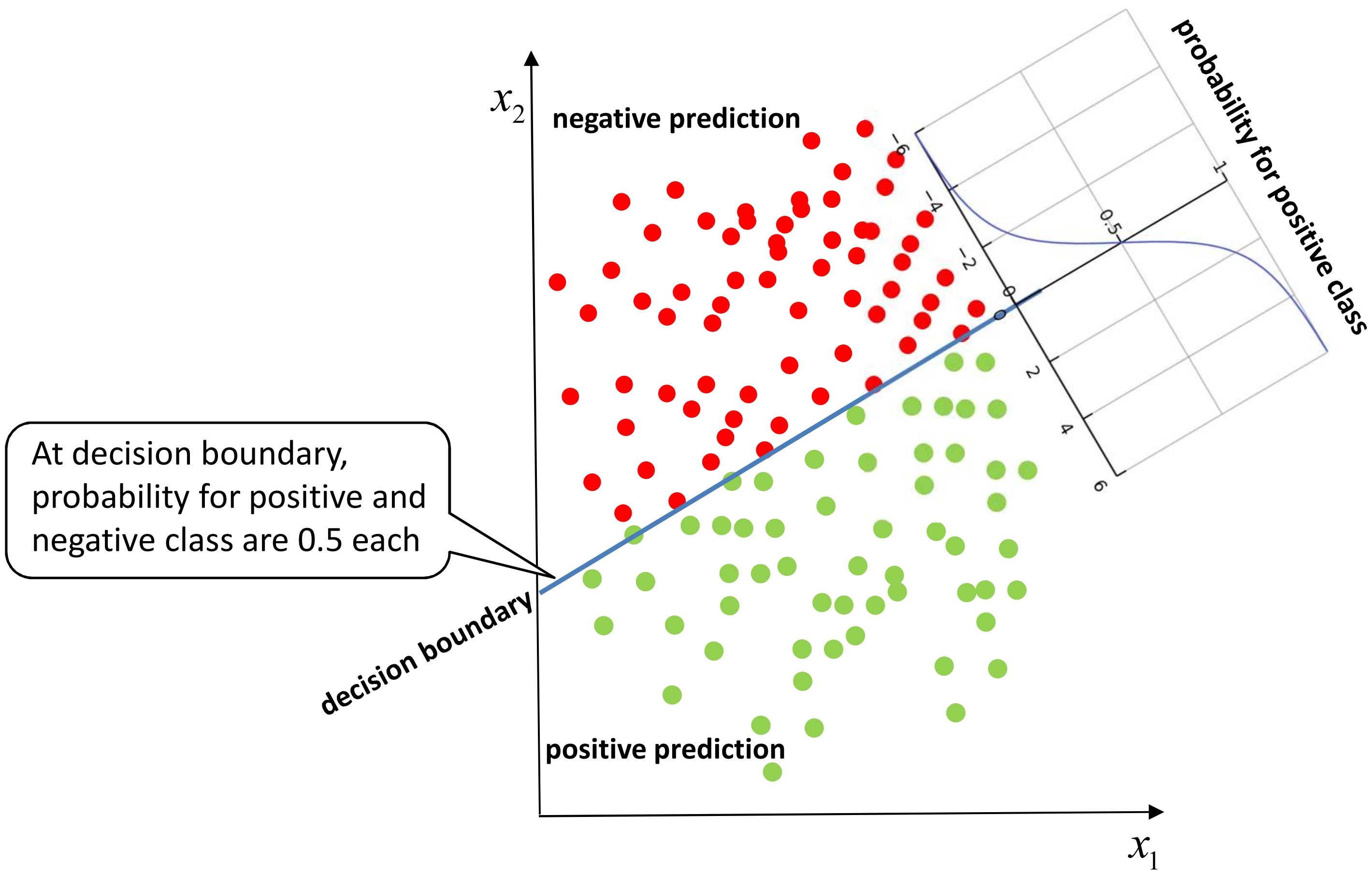
Factoring out $\mathbf{x}_n$

Plugging in definition of $f_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}{1 + e^{\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}}}$

- Gradient is easily computed from data and current predictions of model

Logistic Regression: Visualization

- The result of learning a logistic regression model can be visualized by plotting the **decision boundary** of the classifier, that is, those $\mathbf{x} \in \mathbb{R}^M$ with $f_{\boldsymbol{\theta}}(\mathbf{x}) = 0.5$:



Multiclass Classification

- So far, we talked about binary classification problems
  - Labels are $y \in \{0,1\}$, for example „spam“ and „legitimate“ in email spam filtering
  - The model $f_{\boldsymbol{\theta}}$ predicts a probability for the positive class, that is, $f_{\boldsymbol{\theta}} : \mathbb{R}^M \rightarrow [0,1]$

- Many application domains require **multiclass classification**, that is, $y \in \{1,\ldots,T\}$
- As a start, consider the problem of predicting T continuous targets simultaneously using linear regression models (with explicit bias terms b_t):

$$f_{\boldsymbol{\theta}_1}^{(1)} = \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}_1 + b_1 \qquad \boldsymbol{\theta}_1 \in \mathbb{R}^M, b_1 \in \mathbb{R} \qquad \text{(model for Target 1)}$$

$$\ldots \qquad\qquad \ldots$$

$$f_{\boldsymbol{\theta}_T}^{(T)} = \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}_T + b_T \qquad \boldsymbol{\theta}_T \in \mathbb{R}^M, b_T \in \mathbb{R} \qquad \text{(model for Target } T\text{)}$$

- These linear regression models can be written down as a single model by

$$f_{\boldsymbol{\theta}}^{(lin)}(\mathbf{x}) = \mathbf{B}\mathbf{x} + \mathbf{b} \qquad \mathbf{B} = \begin{pmatrix} \boldsymbol{\theta}_1^{\mathrm{T}} \\ \ldots \\ \boldsymbol{\theta}_T^{\mathrm{T}} \end{pmatrix} \in \mathbb{R}^{T \times M}, \mathbf{b} = \begin{pmatrix} b_1 \\ \ldots \\ b_T \end{pmatrix} \in \mathbb{R}^T, \quad \boldsymbol{\theta} = (\mathbf{B}, \mathbf{b})$$

For optimization etc., still viewing $\boldsymbol{\theta}$ as a vector (concatenate all parameters into long vector)

Multiclass Classification

- In a multiclass classification setting, we could view the T outputs of the model $f_{\boldsymbol{\theta}}^{(lin)}(\mathbf{x}) = \mathbf{B}\mathbf{x} + \mathbf{b}$ as scores for the different classes $y \in \{1,\dots,T\}$
- Similar to the sigmoid function for the binary case, we can use the so-called softmax function to transform the class scores to probabilities
- The softmax function takes a vector $\mathbf{v} = (v_1,\dots,v_T)^{\mathrm{T}} \in \mathbb{R}^T$ as input and returns another vector $\mathrm{softmax}(\mathbf{v}) \in \mathbb{R}^T$ whose entries are all positive and sum to one:

$$\mathrm{softmax}(\mathbf{v}) = \begin{pmatrix} \dfrac{e^{v_1}}{\sum_{t=1}^{T} e^{v_t}} \\ \dots \\ \dfrac{e^{v_T}}{\sum_{t=1}^{T} e^{v_t}} \end{pmatrix}$$

Exponentiate each class score (to ensure that outputs are positive)

Divide by the sum of all exponentiated class scores (to ensure that outputs sum to one)

- The final multiclass logistic regression model is then

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathrm{softmax}(\mathbf{B}\mathbf{x} + \mathbf{b}) \qquad \mathbf{B} \in \mathbb{R}^{T \times M}, \mathbf{b} \in \mathbb{R}^T, \quad \boldsymbol{\theta} = (\mathbf{B}, \mathbf{b})$$

Example: Multiclass Logistic Regression

- The multiclass logistic regression model returns a vector of probabilities for the classes $\{1,\ldots,T\}$:

$$p(y=t \mid \mathbf{x},\boldsymbol{\theta}) = f_{\boldsymbol{\theta}}(\mathbf{x})_t$$

t-th element in output $f_{\boldsymbol{\theta}}(\mathbf{x})$

- Example for converting class scores to probabilities by softmax function (*T=3*):

$$\underbrace{\begin{pmatrix} -1.2 \\ 1.5 \\ 2.\mathbf{7} \end{pmatrix}}_{\mathbf{Bx}+\mathbf{b}} \xrightarrow{\text{exp}} \begin{pmatrix} 0.30 \\ 4.48 \\ \mathbf{14.88} \end{pmatrix} \xrightarrow{\text{divide by sum}} \underbrace{\begin{pmatrix} 0.015 \\ 0.228 \\ \mathbf{0.757} \end{pmatrix}}_{f_{\boldsymbol{\theta}}(\mathbf{x})} \begin{matrix} p(y=1 \mid \mathbf{x}_i,\boldsymbol{\theta}) \\ p(y=2 \mid \mathbf{x}_i,\boldsymbol{\theta}) \\ p(y=3 \mid \mathbf{x}_i,\boldsymbol{\theta}) \end{matrix}$$

Learning Multiclass Logistic Regression

- The multiclass logistic regression model can be learned from data in the same way as the binary logistic regression model, using gradient ascent in the likelihood:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log p(y_1,\ldots,y_N \mid \mathbf{x}_1,\ldots,\mathbf{x}_N,\boldsymbol{\theta}) && \text{can maximize log-likelihood because logarithm is monotone} \\
&= \arg\max_{\boldsymbol{\theta}} \log \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n,\boldsymbol{\theta}) && \text{independence assumption about training instances} \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log \underbrace{p(y_n \mid \mathbf{x}_n,\boldsymbol{\theta})}_{\text{plug in model } p(y=t\mid \mathbf{x},\boldsymbol{\theta}) = f_{\boldsymbol{\theta}}(\mathbf{x})_t} && \text{log of product is sum of logs}
\end{aligned}$$

Discriminant Analysis

- Logistic regression models the conditional probability $p(y \mid \mathbf{x}, \boldsymbol{\theta})$
- **Discriminant analysis** is an alternative modeling approach where we model the full joint probability of inputs and targets:

$$p(y, \mathbf{x} \mid \boldsymbol{\theta}) = p(y \mid \boldsymbol{\theta})\, p(\mathbf{x} \mid y, \boldsymbol{\theta})$$

product rule of probability

- Thus, in discriminate analysis, there is
  - a probability of seeing a data point from a particular class $y \in \{1, \ldots, T\}$, which is given by

    $$p(y = t \mid \boldsymbol{\theta}) = \pi_t$$

    where $(\pi_1, \ldots, \pi_T)^{\mathrm{T}} = \boldsymbol{\pi} \in \mathbb{R}^T$ are model parameters
  - given a class t , a probability over instances $\mathbf{x} \in \mathbb{R}^M$, which is given by a multivariate normal distribution

    $$p(\mathbf{x} \mid y = t, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$

    where $\boldsymbol{\mu}_t \in \mathbb{R}^M$ and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{M \times M}$ are class-specific means and variances of the normal distribution
- Model parameters $\boldsymbol{\theta}$ include $\boldsymbol{\pi} \in \mathbb{R}^T$ and all $\boldsymbol{\mu}_t \in \mathbb{R}^M$ and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{M \times M}$

Multivariate Normal Distribution

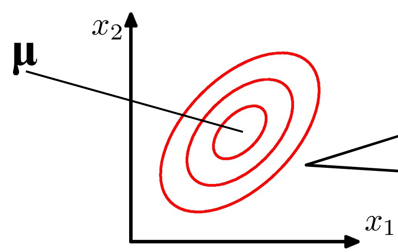
- Excursion: the **multivariate normal distribution** defines a probability distribution over vectors $\mathbf{x} \in \mathbb{R}^M$

- Given by density function with parameters $\boldsymbol{\mu} \in \mathbb{R}^M$ (mean vector) and $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$ (covariance matrix):

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \underbrace{\frac{1}{Z}}_{\text{normalizer}} \exp\left(-\frac{1}{2} (\mathbf{x} - \overset{\text{mean vector}}{\boldsymbol{\mu}})^T \overset{\text{covariance matrix}}{\boldsymbol{\Sigma}^{-1}} (\mathbf{x} - \boldsymbol{\mu}) \right) \qquad \text{Normalizer } Z = 2\pi^{M/2} \underbrace{|\Sigma|}_{\text{determinant of matrix}}^{1/2}$$

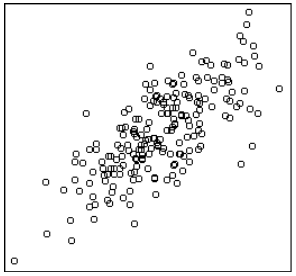
- The parameters $\boldsymbol{\mu} \in \mathbb{R}^M$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$ determine the location and shape of the distribution

- Visualization for $M=2$: can plot isobares of the density, or samples from distribution



(a)

density is highest at mean $\boldsymbol{\mu}$, shapes of isobares determined by covariance matrix $\boldsymbol{\Sigma}$



Discriminant Analysis: Prediction

- Given a discriminant analysis model, a prediction for an instance $\mathbf{x}$ is obtained by finding the argmax over the classes of the joint probability:

$$\arg\max_t p(y=t,\mathbf{x}\,|\,\boldsymbol{\theta}) = \arg\max_t p(y=t\,|\,\boldsymbol{\theta})\,p(\mathbf{x}\,|\,y=t,\boldsymbol{\theta})$$

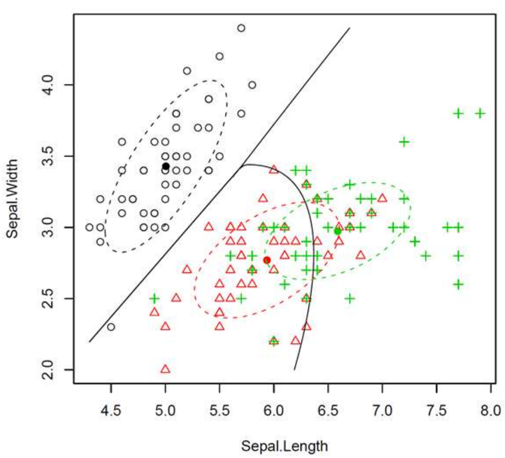
product rule of probability

$$= \arg\max_t \pi_t \mathcal{N}(\mathbf{x}\,|\,\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$$

plugging in model definition

- In general, this leads to non-linear decision boundaries, for example:

Discriminant analysis model for Iris data set, using attributes sepal width and sepal length



- The nonlinearity stems from squared terms in the density of the normal distribution
- The model is therefore also known as quadratic discriminant analysis or QDA

Discriminant Analysis: Learning

- Learning a discriminant analysis model from data can be carried out by maximizing the full likelihood,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(y_1,\ldots,y_N,\mathbf{x}_1,\ldots,\mathbf{x}_N \mid \boldsymbol{\theta})$$

- The maximum likelihood parameters can be computed as follows (no proof):

number of examples with *y=t*

$$n_t = \sum_{n=1}^{N} I(y_n = t) \qquad \text{where } I(condition) = \begin{cases} 1 : condition \text{ is true} \\ 0 : condition \text{ is false} \end{cases}$$

$$\pi_t = \frac{n_t}{N}$$

intuitively: probability for class *t* is fraction of examples with *y=t*

$$\boldsymbol{\mu}_t = \frac{1}{n_t} \sum_{n=1}^{N} I(y_n = t)\mathbf{x}_n$$

intuitively: mean vector for class is average over instances in that class (the *I(...)* expression picks out the instances of class *t*)

Discriminant Analysis: Learning

- Learning a discriminant analysis model from data can be carried out by maximizing the full likelihood,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(y_1,\dots,y_N,\mathbf{x}_1,\dots,\mathbf{x}_N \mid \boldsymbol{\theta})$$

- The maximum likelihood parameters can be computed as follows (no proof):

$$\boldsymbol{\Sigma}_t = \frac{1}{n_t}\sum_{n=1}^{N} I(y_n = t)\underbrace{(\mathbf{x}_n - \boldsymbol{\mu}_t)(\mathbf{x}_n - \boldsymbol{\mu}_t)^{\mathrm{T}}}_{\in \mathbb{R}^{M\times M}}$$

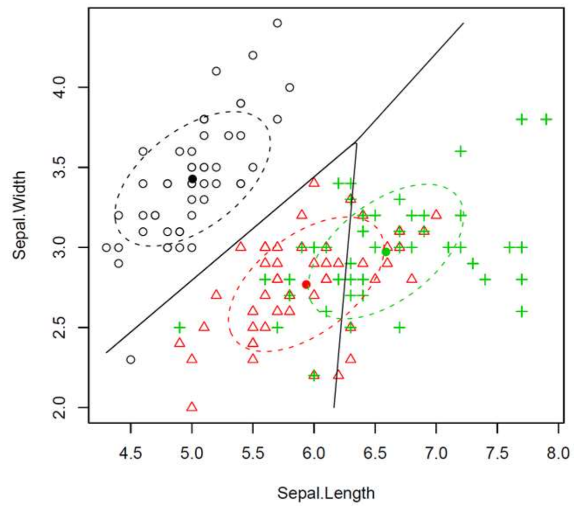
covariance matrix for class *t* is computed from instances of that class

Linear Discriminant Analysis

- An important special case of discriminant analysis is **linear discriminant analysis**
  - In linear discriminant analysis (or **LDA**), we require that the covariance matrices of the class-specific distributions are all identical:

$$\mathbf{\Sigma}_t = \mathbf{\Sigma}_{t'} \qquad \text{for all } t, t' \in \{1, \ldots, T\}$$

  - This leads to linear decision boundaries:



  - The joint covariance matrix can be estimated as $\mathbf{\Sigma} = \sum_{t=1}^{T} \frac{n_t}{N} \mathbf{\Sigma}_t$

Summary

- Similar to the linear regression models discussed in the last lecture, linear models are also an important class of models for classification

- Most widely used is the **logistic regression** model
  - Probabilistic model that defines a conditional distribution $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ by a linear model and the sigmoid function (or the softmax function in the multiclass setting)
  - This leads to linear decision boundaries
  - Parameters of the model can be estimated from data using gradient ascent in the likelihood

- **Discriminant analysis** is an alternative approach where we model the joint distribution of inputs and outputs, $p(\mathbf{x}, y \mid \boldsymbol{\theta})$, rather than the conditional $p(y \mid \mathbf{x}, \boldsymbol{\theta})$
  - Joint distribution is modeled using class-specific multivariate normal distributions
  - Leads to non-linear decision boundaries unless covariance matrix is shared across classes