Universität Stuttgart
KI – Institute for Artificial Intelligence

Analytic Computing

Machine Learning
7 Logistic Regression

Prof. Dr. Steffen Staab
Nadeen Fatallah
Daniel Frank
Akram Sadat Hosseini
Jiaxin Pan
Osama Mohamed
Arvindh Arunbabu
Tim Schneider
Yi Wang

https://www.ki.uni-stuttgart.de/

Learning objectives

- Expected prediction error
- Loss functions: squared error, zero-one loss, cross entropy
- Bayes error
- Decision boundaries
- Logistic regression
  - logistic function, logits

Bayes error

Expected prediction error for numeric values: squared error loss

$$EPE(\hat{f}) = E\left(Y - \hat{f}(X)\right)^2 = \int \left[y - \hat{f}(x)\right]^2 P(dx, dy)$$

- leads to conditional **mean** as best solution:

$$\hat{f}(x) = E(Y|X = x)$$

Expected prediction error for numeric values: absolute error loss

$$EPE(\hat{f}) = E(|Y - \hat{f}(X)|) \quad = \int |y - \hat{f}(x)| \quad P(dx, dy)$$

- leads to conditional **median** as best solution:

$$\hat{f}(x) = \mathrm{median}(Y|X = x)$$

Not much used in practice of machine learning, because |...| is not differentiable everywhere – though it has its strengths

Expected prediction error for classification

- Let $K \times K$ matrix $\boldsymbol{L}$ represents the loss $L_{G,\hat{G}}$,
  given set of categories $\mathcal{g}$,
  the cost for classifying some object from category G into $\hat{G}$,
  and K is the cardinality of $\mathcal{g}$
- Correct classification has loss 0

$$\boldsymbol{L} = \begin{pmatrix} 0 & L_{1,2} & \cdots & & & L_{1,k} \\ L_{2,1} & 0 & \ddots & & & \\ L_{3,1} & \ddots & \ddots & & & \vdots \\ \vdots & & & 0 & & \\ & & & & 0 & L_{k-1,k} \\ L_{k,1} & \cdots & & & L_{k,k-1} & 0 \end{pmatrix}$$

Zero-one loss

- Let $K \times K$ matrix $\boldsymbol{L}$ represents the loss $L_{G,\hat{G}}$, the cost for classifying some object from category G into $\hat{G}$,
- Correct classification has loss 0

$$\boldsymbol{L} = \begin{pmatrix} 0 & 1 & \dots & & \cdots & 1 \\ 1 & 0 & 1 & & & \vdots \\ 1 & 1 & \ddots & & & \vdots \\ \vdots & & \ddots & 0 & \ddots & 1 \\ & & & \ddots & 0 & 1 \\ 1 & \cdots & & 1 & 1 & 0 \end{pmatrix}$$

Expected prediction error for classification (zero-one loss)

$$EPE(\hat{f}) = E\left[L_{G,\hat{G}(X)}\right] = E_X\left(\sum_{G\in\mathcal{G}} L_{G,\hat{G}(X)} \cdot P(G|X)\right)$$

leading to

$$\hat{G}(X) = \underset{G\in\mathcal{G}}{\operatorname{argmin}}[1 - P(G|X = x)]$$

or

$$\hat{G}(X) = \underset{G\in\mathcal{G}}{\operatorname{argmax}}\, P(G|X = x)$$

Bayes classifier

Given a classification problem with categories $\mathcal{g}$

- Assume we know **the true distribution** $P(Y = y|X = x)$
- Without further knowledge the optimal classifier (also known as *Bayes classifier*) is:

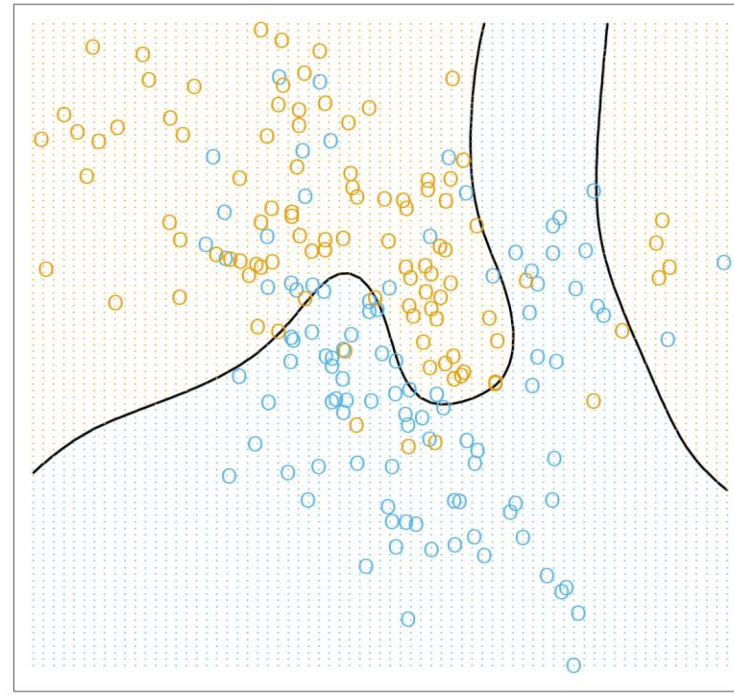
$$\underset{y}{\operatorname{argmax}}\, P(y|X = x)$$

Demonstrating the Optimal Bayes classifier

Generating function f:

- Generation of 10 means m_k from a bivariate Gaussian distribution $N((1,0)^T, \mathbf{I})$ labeled this class BLUE.
- 10 more were drawn from $N((0,1)^T, \mathbf{I})$ and labeled class ORANGE.
- For both classes, 100 samples were generated:
  - for each observation, m_k was picked randomly with probability $\frac{1}{10}$, and then generated from $N(m_k, \mathbf{I}/5)$

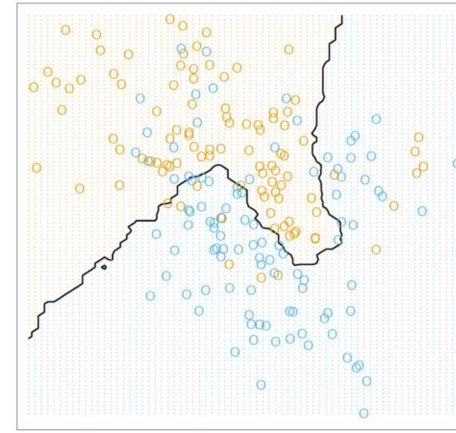
Optimal Bayes classifier $\hat{f}$



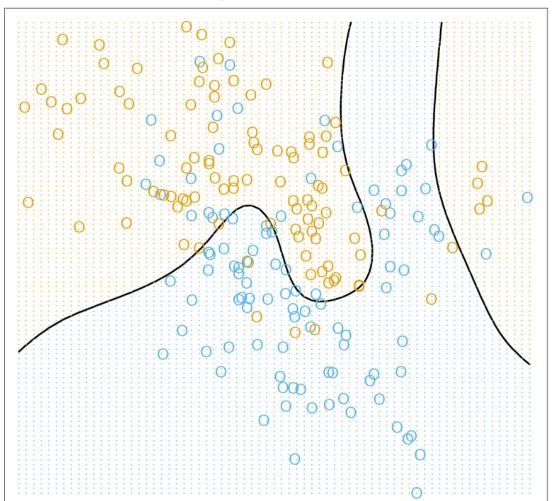
Demonstrating the Optimal Bayes classifier

1. We do not know a priori what is the best strategy
2. Even the best strategy is not error free

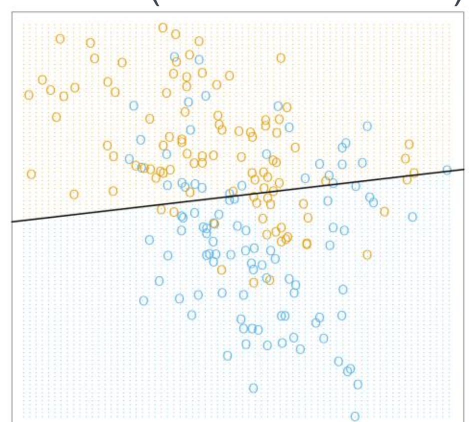
15-NN



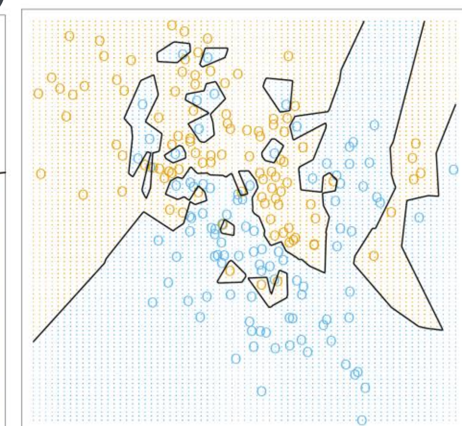
Optimal Bayes classifier



Linear separation
(more on that later)

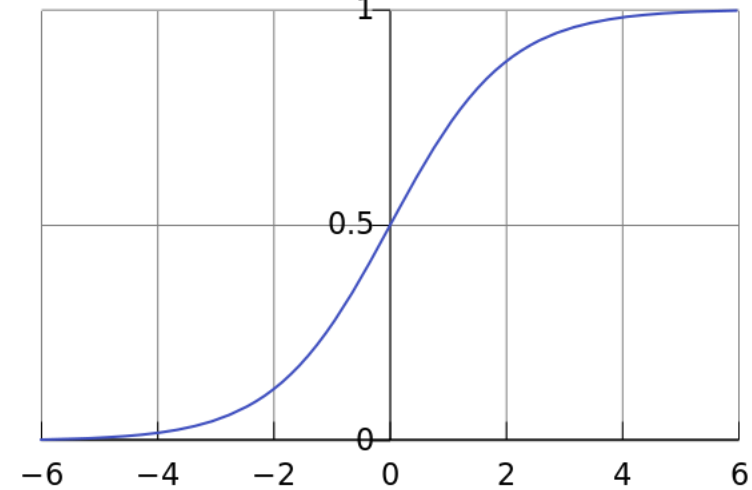


1-NN



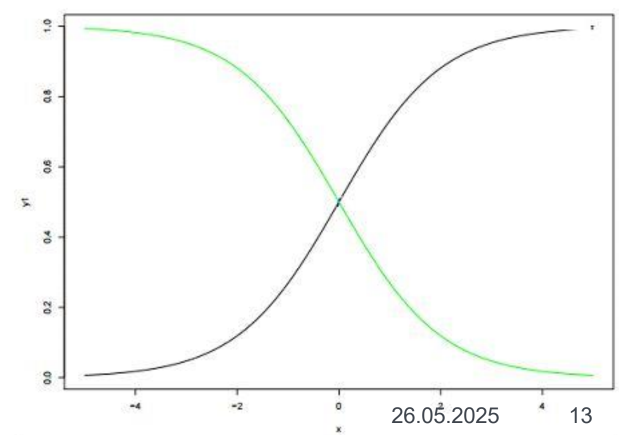
Classification by Logistic regression

Core ideas of logistic regression



- Learn weights similarly as in linear regression
  - Use the logistic function $\frac{e^x}{1+e^x}$
  - Map $x^T\beta$ using the logistic function onto $\frac{e^{x^T\beta}}{1+e^{x^T\beta}}$
- Predict conditional probabilities $\in [0,1]$
  - 2-class example
    - $P(G=1|X=x) = \frac{e^{x^T\beta}}{1+e^{x^T\beta}}$
    - $P(G=2|X=x) = \frac{1}{1+e^{x^T\beta}}$

    Sum is 1



Decision boundary

Logit transformation: $\log \frac{p}{1-p}$

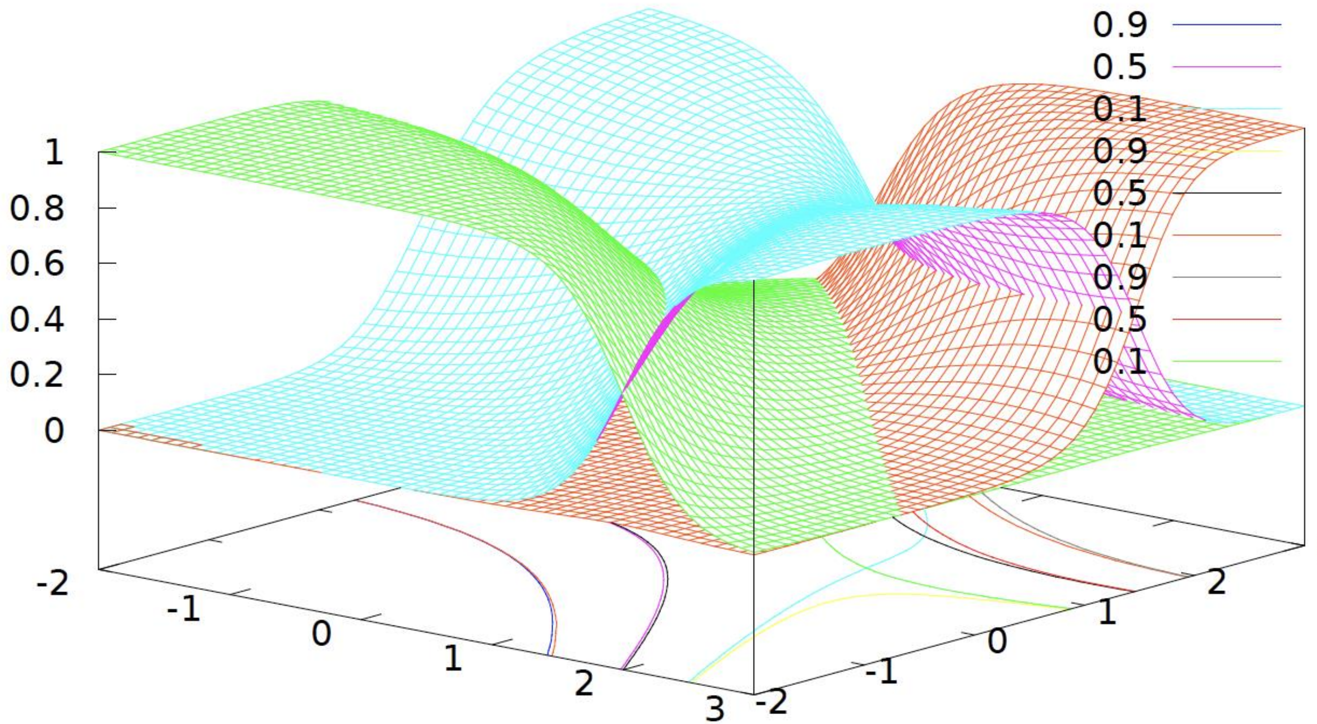
$$\log \frac{P(G=1|X=x)}{P(G=2|X=x)} = \log \frac{\frac{e^{x^T\beta}}{1+e^{x^T\beta}}}{\frac{1}{1+e^{x^T\beta}}} = \log e^{x^T\beta} = x^T\beta$$

Decision boundary where *log-odds* (also called *logits*) are 0:

$$\{x \mid x^T\beta = 0\}$$

*we typically assume $\log x = \ln x$,
but this is largely irrelevant

Multi-class case



Graphics by Toussaint 2019

Logistic regression: Multi-class case

- Data $D = \{(x_i, y_i)\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \dots, K\}$
- For each y, i.e. each column of the indicator matrix $\mathbf{Y}$:
  - choose $f(x, y) = \phi(x)^T \beta_y$ with separate vector β_y for each y
  - define conditional class probabilities

$$P(Y = y|X = x) = \frac{e^{f(x,y)}}{\sum_{y'} e^{f(x,y')}} \Leftrightarrow f(x, y) - f(x, z) = \log \frac{P(y|x)}{P(z|x)}$$

Maximum log-likelihood
(minimize neg-log-likelihood)

Given Data $D = \{(x_i, y_i)\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \dots, K\}$

The likelihood of a model for N observations is

$$P(Y, X; \beta) = \prod_{i=1}^{N} P(y_i | x_i; \beta)$$

which can be rewritten into log-likelihood

$$\log P(Y, X; \beta) = \sum_{i=1}^{N} \log P(y_i | x_i; \beta)$$

Logistic regression: Multi-class case

- Data $D = \{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, K\}$

- For each y, i.e. each column of the indicator matrix $\mathbf{Y}$:
  - choose $f(x, y) = \phi(x)^T \beta_y$ with separate vector β_y for each y
  - define conditional class probabilities

$$P(Y = y|X = x) = \frac{e^{f(x,y)}}{\sum_{y'} e^{f(x,y')}} \Leftrightarrow f(x, y) - f(x, z) = \log \frac{P(y|x)}{P(z|x)}$$

- Minimize the regularized neg-log-likelihood

$$\mathrm{loss}^{\mathrm{logistic}}(\beta) = -\sum_{i=1}^{N} \log P(y_i|x_i) + \lambda \, \|\beta\|^2$$

Cross entropy

- minor, but widely-used generalization of neg-log-likelihood
- Assume category is encoded in **one-hot-vector**

$$\bar{y}_i = \boldsymbol{e}_{y_i} = (0, \dots, 0,1,0, \dots, 0)^T, \bar{y}_{i,z} = [y_i = z]$$

- Write neg-log-likelihood as

$$\text{loss}^{\text{nll}}(\beta) = -\sum_{i=1}^{N} \sum_{z=1}^{K} \bar{y}_{i,z} \log P(Y = z | X = x_i) \;= \sum_{i=1}^{N} H(\bar{y}_i, P(\cdot\, | X = x_i))$$

with $H(p, q) = -\sum_z p(z) \log q(z)$ being the **cross entropy** between two multinomial probability distributions p and q

- Loss based on cross entropy generalizes from the special case of one-hot-vectors to arbitrary probabilistic vectors $\bar{y}_i$

Logistic regression: Loss in the multi-class case using cross entropy

- Minimize the regularized neg-log-likelihood

$$\mathrm{loss}^{\mathrm{logistic}}(\beta) = -\sum_{i=1}^{N} \log P(y_i | x_i) + \lambda \, ||\beta||^2$$

- Minimize Loss based on cross entropy and regularization

$$\mathrm{loss}^{\mathrm{logistic}}(\beta) = -\sum_{i=1}^{N} \sum_{z=1}^{K} \bar{y}_{i,z} \log P(Y = z | X = x_i) + \lambda \, ||\beta||^2 =$$

$$\sum_{i=1}^{N} H(\bar{y}_i, P(\cdot \, | X = x_i)) + \lambda \, ||\beta||^2$$

2 classes, log-likelihood optimization

$$\log P(Y, X; \beta) = \sum_{i=1}^{N} \left(y_i \log \hat{f}(x_i; \beta) + (1 - y_i) \log(1 - \hat{f}(x_i; \beta)) \right)$$

$$= \sum_{i=1}^{N} \left(y_i x_i^T \beta - \log \left(1 + e^{x_i^T \beta} \right) \right)$$

where $Y = 1$ implies $y = 1$ and $Y = 2$ implies $y = 0$

Set the derivation to 0:

$$\frac{\partial \log P(Y, X; \beta)}{\partial \beta} = \sum_{i=1}^{N} \left(y_i x_i^T - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \cdot x_i^T \right) = \sum_{i=1}^{N} \left(y_i - \hat{f}(x_i; \beta) \right) x_i^T = 0$$

Solving

Zero value of $\dfrac{\partial \log P(Y,X;\beta)}{\partial \beta}$ cannot be determined analytically.

Numeric solution can e.g. by found by iterative Newton method

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T}\right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

with

$$\ell(\beta) = \log P(Y,X;\beta)$$

More about this topic later

Universität Stuttgart
KI

Thank you!

Steffen Staab

E-Mail Steffen.staab@ki.uni-stuttgart.de

Telefon +49 (0) 711 685-88100

www.ki.uni-stuttgart.de/

Universität Stuttgart
Analytic Computing, KI
Universitätsstraße 32, 50569 Stuttgart