Linear Classifiers

Volker Tresp
Winter 2023-2024
Classification

- Classification is the central task of pattern recognition
- Sensors supply information about an object: to which class does the object belong (dog, cat, ...)?
The beauty of Machine Learning is that a few model classes (neural networks, kernel approaches, ...) can be applied to almost any supervised learning task.

This hides a bit that the data settings can be quite different.

There are problems where class boundaries are well defined but maybe quite complex; an example is OCR; here Deep Neural Networks, manifold learning and kernel systems are quite effective; this concerns often our Cases I and II.

In other applications there is little structure in the data and classes overlap; this the situation encountered in many healthcare applications (biomedicine); this concerns often our Cases III and IV.

Often, the problem is not as much to separate classes, but to show that there is a signal at all; the question might be if there is a detectable positive effect of the new medication!
Linear Classifiers

- Linear classifiers separate classes by a linear hyperplane
- In high dimensions a linear classifier often can separate the classes
- Linear classifiers cannot solve the exclusive-or problem
- In combination with basis functions, kernels or a neural network, linear classifiers can form nonlinear class boundaries
Hard and Soft (sigmoid) Transfer Functions

• First, the activation function of the neurons in the hidden layer are calculated as the weighted sum of the inputs $x_i$ as

$$h(x) = \sum_{j=0}^{M} w_j x_j$$

(note: $x_0 = 1$ is a constant input, so that $w_0$ corresponds to the bias)

• The sigmoid neuron has a soft (sigmoid) transfer function

$$\hat{y} = \text{sig}(h(x))$$

Perceptron :  $\hat{y} = \text{sign}(h(x))$

Sigmoid function:  $\hat{y} = \text{sig}(h(x))$
Binary Classification Problems

- We will focus first on binary classification where the task is to assign binary class labels $y_i = 1$ and $y_i = 0$ (or $y_i = 1$ and $y_i = -1$)

- We already know the *Perceptron*. Now we learn about additional approaches

  - I. Generative models for classification
  - II. Logistic regression
  - III. Classification via regression
Two Linearly Separable Classes
Two Classes that Cannot be Separated Linearly
The Classical Example not two Classes that cannot be Separated Linearly: XOR
Separability is not a Goal in Itself. With Overlapping Classes the Goal is the Best Possible Hyperplane
I. Generative Model for Classification

• In a generative model one assumes a probabilistic data generating process (likelihood model). Often generative models are complex and contain unobserved (latent, hidden) variables

• Here we consider a simple example: data is generated from class-specific Gaussian distributions

• First we have a model how classes are generated $P(y)$. $y = 1$ could stand for a good customer and $y = 0$ could stand for a bad customer.
Generative Model for Classification (cont’d)

• Then we have a model how attributes are generated, given the classes \( P(\tilde{x}|y) \). This could stand for

  – Income, age, occupation (\( \tilde{x} \)) given a customer is a good customer (\( y = 1 \))
  – Income, age, occupation (\( \tilde{x} \)) given a customer is not a good customer (\( y = 0 \))

• Using Bayes formula, we then derive \( P(y|\tilde{x}) \): the probability that a given customer is a good customer \( y = 1 \) or bad customer \( y = 0 \), given that we know the customer’s income, age and occupation
How is Data Generated?

• We assume that the observed classes $y_i$ are generated with probability

\[ P(y_i = 1) = \kappa_1 \quad P(y_i = 0) = \kappa_0 = 1 - \kappa_1 \]

with $0 \leq \kappa_1 \leq 1$.

• In a next step, a data point $\tilde{x}_i$ has been generated from $P(\tilde{x}_i | y_i)$

• (Note, that $\tilde{x}_i = (x_{i,1}, \ldots, x_{i,M})^T$, which means that $\tilde{x}_i$ does not contain the bias $x_{i,0}$)

• We now have a complete model: $P(y_i)P(\tilde{x}_i | y_i)$
Bayes’ Theorem

• To classify a data point $\tilde{x}_i$, i.e. to determine the $y_i$, we apply Bayes theorem and get

$$P(y_i|\tilde{x}_i) = \frac{P(\tilde{x}_i|y_i)P(y_i)}{P(\tilde{x}_i)}$$

$$P(\tilde{x}_i) = P(\tilde{x}_i|y_i = 1)P(y_i = 1) + P(\tilde{x}_i|y_i = 0)P(y_i = 0)$$
Distinguishing between ashes ("Eschen") and birches ("Birken")

(The magic of Bayes formula):

\[
P(y = 1|x) = \frac{P(x|y = 1)P(y = 1)}{P(x)}
\]

\[
P(x) = P(x|y = 1)P(y = 1) + P(x|y = 0)P(y = 0)
\]
ML Estimate for Class Probabilities

- Maximum-likelihood estimator for the prior class probabilities are

\[ \hat{P}(y_i = 1) = \hat{\kappa}_1 = \frac{N_1}{N} \]

and

\[ \hat{P}(y_i = 0) = \hat{\kappa}_0 = \frac{N_0}{N} = 1 - \hat{\kappa}_1 \]

where \( N_1 \) and \( N_0 \) is the number of training data points for class 1, respectively class 0.

- Thus the class probabilities simply reflect the class mix.
Class-specific Distributions

• To model $P(\tilde{x}_i|y_i)$ one can choose an application specific distribution

• A popular choice is a Gaussian distribution (normal discriminant analysis)

$$P(\tilde{x}_i|y_i = l) = \mathcal{N}(\tilde{x}_i; \mu^{(l)}, \Sigma)$$

with

$$\mathcal{N}(\tilde{x}_i; \mu^{(l)}, \Sigma) = \frac{1}{(2\pi)^{M/2} \sqrt{|\Sigma|}} \exp \left( -\frac{1}{2} \left( \tilde{x}_i - \mu^{(l)} \right)^T \Sigma^{-1} \left( \tilde{x}_i - \mu^{(l)} \right) \right)$$

• Note, that both Gaussian distributions have different modes (centers) but the same covariance matrices. This has been shown to often work well
Maximum-likelihood Estimators for Modes and Covariances

• One obtains a maximum likelihood estimators for the modes

\[
\hat{\mu}(l) = \frac{1}{N_l} \sum_{i: y_i = l} \tilde{x}_i
\]

• One obtains as unbiased estimators for the covariance matrix

\[
\hat{\Sigma} = \frac{1}{N - M} \sum_{l=0}^{1} \sum_{i: y_i = l} (\tilde{x}_i - \hat{\mu}(l))(\tilde{x}_i - \hat{\mu}(l))^T
\]
Expanding the Quadratic Terms in the Exponent

- Note that

\[-\frac{1}{2} \left( \tilde{x}_i - \mu^{(l)} \right)^T \Sigma^{-1} \left( \tilde{x}_i - \mu^{(l)} \right) \]

\[= -\frac{1}{2} \tilde{x}_i^T \Sigma^{-1} \tilde{x}_i - \frac{1}{2} \mu^{(l)T} \Sigma^{-1} \mu^{(l)} + \mu^{(l)T} \Sigma^{-1} \tilde{x}_i \]
The Difference of the Quadratic

• Now we calculate the difference of the quadratic terms of the two Gaussians

\[-\frac{1}{2} (\tilde{x}_i - \tilde{\mu}^{(0)})^T \Sigma^{-1} (\tilde{x}_i - \tilde{\mu}^{(0)}) + \frac{1}{2} (\tilde{x}_i - \tilde{\mu}^{(1)})^T \Sigma^{-1} (\tilde{x}_i - \tilde{\mu}^{(1)})\]

\[= -\frac{1}{2} \tilde{x}_i^T \Sigma^{-1} \tilde{x}_i - \frac{1}{2} \tilde{\mu}^{(0)}^T \Sigma^{-1} \tilde{\mu}^{(0)} + \tilde{\mu}^{(0)}^T \Sigma^{-1} \tilde{x}_i\]

\[+ \frac{1}{2} \tilde{x}_i^T \Sigma^{-1} \tilde{x}_i + \frac{1}{2} \tilde{\mu}^{(1)}^T \Sigma^{-1} \tilde{\mu}^{(1)} - \tilde{\mu}^{(1)}^T \Sigma^{-1} \tilde{x}_i\]

• .... since two terms cancel,

\[= (\tilde{\mu}^{(0)} - \tilde{\mu}^{(1)})^T \Sigma^{-1} \tilde{x}_i - \frac{1}{2} \tilde{\mu}^{(0)}^T \Sigma^{-1} \tilde{\mu}^{(0)} + \frac{1}{2} \tilde{\mu}^{(1)}^T \Sigma^{-1} \tilde{\mu}^{(1)}\]
A Posteriori Distribution

- It follows that

\[
P(y_i = 1 | \tilde{x}_i) = \frac{P(\tilde{x}_i | y_i = 1)P(y_i = 1)}{P(\tilde{x}_i | y_i = 1)P(y_i = 1) + P(\tilde{x}_i | y_i = 0)P(y_i = 0)}
\]

\[
= \frac{1}{1 + \frac{P(\tilde{x}_i | y_i = 0)P(y_i = 0)}{P(\tilde{x}_i | y_i = 1)P(y_i = 1)}}
\]

\[
= \frac{1}{1 + \frac{\kappa_0}{\kappa_1} \exp \left( (\mu^0 - \mu^1)^T \Sigma^{-1} \tilde{x}_i - \frac{1}{2} \mu^0^T \Sigma^{-1} \mu^0 + \frac{1}{2} \mu^1^T \Sigma^{-1} \mu^1 \right)}
\]

\[
= \text{sig} \left( w_0 + \tilde{x}_i^T \tilde{w} \right) = \text{sig} \left( w_0 + \sum_{j=1}^{M} x_{i,j} w_j \right)
\]
Weights

• We get ($\tilde{w}$ is without $w_0$)

$$\tilde{w} = \Sigma^{-1} \left( \hat{\mu}^{(1)} - \hat{\mu}^{(0)} \right)$$

• Note that $\tilde{w}$ is independent of $\kappa_1$ and $\kappa_0$ and is thus independent of the class proportions in the training data! This is important, e.g., for case-control studies

• Recall: $\text{sig}(\text{arg}) = 1/(1 + \exp(-\text{arg}))$
Bias Term

- We get,

\[ w_0 = \log \frac{\kappa_1}{\kappa_0} + \frac{1}{2} \bar{\mu}^{(0)}^T \Sigma^{-1} \bar{\mu}^{(0)} - \frac{1}{2} \bar{\mu}^{(1)}^T \Sigma^{-1} \bar{\mu}^{(1)} \]

- \( w_0 \) clearly reflects the class proportions
Comments

• This specific generative model leads to linear class boundaries
• But we do not only get class boundaries, we get probabilities
• Although we have used Bayes formula, the analysis was frequentist. A Bayesian analysis with a prior distribution on the parameters is also possible
• If the two class-specific Gaussians have different covariance matrices \((\Sigma^{(0)}, \Sigma^{(1)})\) the approach is still feasible but one would need to estimate two covariance matrices and the decision boundaries are not linear anymore; still, one can simply apply Bayes rule to obtain posterior probabilities

• The generalization to multiple classes is straightforward: simply estimate a different Gaussian for each class (with shared covariances or not) and apply Bayes rule

• *Generative-Discriminative pair*: (1) Gaussian Analysis (as a generative model) and (2) logistic regression as a discriminant model

• Generalization to basis functions is straight forward: \(x\) is replaced by \(\phi(x)\)

• With an explicit \(P(\tilde{x}_i|y_i = l) = \mathcal{N}(\tilde{x}_i; \mu^{(l)}, \Sigma)\), we can apply Bayes formula for a posteriori class estimation

• This is not easy, or even impossible, e.g., for GANs, which are able to generate samples but where the likelihood is not easily evaluated (likelihood free methods)
Special Case: Naive Bayes

- With diagonal covariances matrices, one obtains a Naive-Bayes classifier

\[ P(\tilde{x}_i | y_i = l) = \prod_{j=1}^{M} \mathcal{N}(x_{i,j}; \mu_j^{(l)}, \sigma_j^2) \]

- The naive Bayes classifier has considerable fewer parameters but completely ignores class-specific correlations between features; this is sometimes considered to be naive

- Even more naive (all Gaussian have identical variance):

\[ P(\tilde{x}_i | y_i = l) = \prod_{j=1}^{M} \mathcal{N}(x_{i,j}; \mu_j^{(l)}, \sigma^2) \]
Logistic Regression from Naive Bayes

• We have parameters, for the latter case,

\[ w_j = \frac{1}{\sigma^2} \left( \mu_j^{(1)} - \mu_j^{(0)} \right) \]

\[ w_0 = \log \kappa_1 / \kappa_0 + \frac{1}{2\sigma^2} \sum_j \left( \mu_j^{(0)} \right)^2 - \left( \mu_j^{(1)} \right)^2 \]

• Note that \( w_j \) is completely independent of other inputs; adding or removing other inputs does not change \( w_j \);

• In contrast \( w_0 \) depends on all dimensions

• The smaller \( \sigma^2 \), the sharper the transition
Special Case: Bernoulli Naive Bayes

• Naive Bayes classifiers are popular in text analysis with often more than 10000 features (key words). For example, the classes might be SPAM and no-SPAM and the features are keywords in the texts.

• Instead of a Gaussian distribution, a Bernoulli distribution is employed.

• \( P(\text{word}_j = 1 | \text{SPAM}) = \gamma_{j,s} \) is the probability of observing word \( \text{word}_j \) in the document for SPAM documents (Bernoulli distribution).
Special Case: Bernoulli Naive Bayes

- We also consider the other cases

- \( P(\text{word}_j = 0|\text{SPAM}) = 1 - \gamma_{j,s} \) is the probability of not observing word \( \text{word}_j \) in the document for SPAM documents

- \( P(\text{word}_j = 1|\text{no-SPAM}) = \gamma_{j,n} \) is the probability of observing word \( \text{word}_j \) in the document for non-SPAM documents

- \( P(\text{word}_j = 0|\text{no-SPAM}) = 1 - \gamma_{j,n} \) is the probability of not observing word \( \text{word}_j \) in the document for non-SPAM documents

- Note that there are two parameters per dimension: \( \gamma_{j,s} \) and \( \gamma_{j,n} \)
Special Case: Bernoulli Naive Bayes (cont’d)

• Then

\[
P(\text{SPAM}|\text{doc}) = \frac{\kappa_s \prod_j \gamma_{j,s}^{\text{word}_j} (1 - \gamma_{j,s})^{1-\text{word}_j}}{\kappa_s \prod_j \gamma_{j,s}^{\text{word}_j} (1 - \gamma_{j,s})^{1-\text{word}_j} + \kappa_n \prod_j \gamma_{j,n}^{\text{word}_j} (1 - \gamma_{j,n})^{1-\text{word}_j}}
\]

• Simple ML estimates are \(\gamma_{j,s} = \frac{N_{j,s}}{N_s}\) and \(\gamma_{j,n} = \frac{N_{j,n}}{N_n}\)

\((N_s\) is the number of SPAM documents in the training set, \(N_{j,s}\) is the number of SPAM documents in the training set where \(\text{word}_j\) is present\)

\((N_n\) is the number of no-SPAM documents in the training set, \(N_{j,n}\) is the number of no-SPAM documents in the training set where \(\text{word}_j\) is present)
Note, that we can also write

\[ P(\text{SPAM}|\text{doc}) = \text{sig}(w_0 + \sum_j w_j \text{word}_j) \]

with

\[ w_j = [\log \gamma_{j,s} - \log \gamma_{j,n}] - [\log(1 - \gamma_{j,s}) - \log(1 - \gamma_{j,n})] \]

\[ w_0 = \log \kappa_s/\kappa_n + \sum_j \log(1 - \gamma_{j,s}) - \log(1 - \gamma_{j,n}) \]

**Generative-Discriminative pair:** (1) Bernoulli naive Bayes classifier and (2) logistic regression
II. Logistic Regression

- In I. (Generative models for classification) we first defined a generative model for $P(x, y)$; from this model we then derived $P(y|x) = P(y)P(x|y)$ which models $x$ given $y$ (generative modelling).

- Here, we model the reverse $P(y|x)$ (standard supervised learning).

- With logistic regression as the discriminant version, we model discriminatively

$$\hat{y}_i = P(y = 1|x_i) = \text{sig} \left( x_i^T w \right)$$

(now we include the bias $x_i^T = (x_{i,0} = 1, x_{i,1}, \ldots, x_{i,M-1})^T$). $\text{sig}()$ as defined before (logistic function).

- One now optimizes the likelihood of the conditional model

$$L(w) = \prod_{i=1}^{N} \text{sig} \left( x_i^T w \right)^{y_i} \left( 1 - \text{sig} \left( x_i^T w \right) \right)^{1-y_i}$$
Log-Likelihood Function is the Negative Cross Entropy

- Log-likelihood function

$$l = \sum_{i=1}^{N} y_i \log \left( \text{sig} \left( x_i^T w \right) \right) + (1 - y_i) \log \left( 1 - \text{sig} \left( x_i^T w \right) \right)$$

- Cross-entropy cost function (negative log-likelihood)

$$l = - \left[ \sum_{i=1}^{N} y_i \log \left( \text{sig} \left( x_i^T w \right) \right) + (1 - y_i) \log \left( 1 - \text{sig} \left( x_i^T w \right) \right) \right]$$
Log-Likelihood

- Log-likelihood function

\[
l = \sum_{i=1}^{N} y_i \log \left( \sigma \left( x_i^T w \right) \right) + (1 - y_i) \log \left( 1 - \sigma \left( x_i^T w \right) \right)
\]

\[
l = \sum_{i=1}^{N} y_i \log \left( \frac{1}{1 + \exp(-x_i^T w)} \right) + (1 - y_i) \log \left( \frac{1}{1 + \exp(x_i^T w)} \right)
\]

\[
= - \sum_{i=1}^{N} y_i \log(1 + \exp(-x_i^T w)) + (1 - y_i) \log(1 + \exp(x_i^T w))
\]
Adaption

• The derivatives of the log-likelihood with respect to the parameters

\[
\frac{\partial l}{\partial w} = \sum_{i=1}^{N} y_i \frac{x_i \exp(-x_i^T w)}{1 + \exp(-x_i^T w)} - (1 - y_i) \frac{x_i \exp(x_i^T w)}{1 + \exp(x_i^T w)}
\]

\[
= \sum_{i=1}^{N} y_i x_i (1 - \text{sig}(x_i^T w)) - (1 - y_i) x_i \text{sig}(x_i^T w)
\]

\[
= \sum_{i=1}^{N} (y_i - \text{sig}(x_i^T w)) x_i = \sum_{i=1}^{N} (y_i - \hat{y}_i) x_i
\]

• A gradient-based optimization of the parameters to maximize the log-likelihood

\[
w \leftarrow w + \eta \frac{\partial l}{\partial w}
\]

• Typically one uses a Newton-Raphson optimization procedure
Logistic Regression as a Generalized Linear Models (GLM)

- Consider a Bernoulli distribution with $P(y = 1) = \theta$ and $P(y = 0) = 1 - \theta$, with $0 \leq \theta \leq 1$

- In the theory of the exponential family of distributions, one sets
  
  $$\theta = \text{sig}(\eta)$$

  Now we get valid probabilities for any $\eta \in \mathbb{R}$!

- $\eta$ is called the natural parameter and $\text{sig}(\cdot)$ the inverse parameter mapping for the Bernoulli distribution
Logistic Regression as a Generalized Linear Models (GLM) (cont’d)

• This is convenient if we make $\eta$ a linear function of the inputs and one obtains a Generalized Linear Model (GLM)

$$P(y_i = 1|x_i, w) = \text{sig}(x_i^T w)$$

• Thus logistic regression is the GLM for the Bernoulli likelihood model
Application to Neural Networks and other Systems

- Logistic regression essentially defines a new cost function
- It can be applied as well to neural networks, as we have done before,

\[ P(y_i = 1|x_i, w) = \text{sig}(\text{NN}(x_i)) \]

or systems of basis functions or kernel systems
Multiple Classes and Softmax

• Consider a multinomial distribution with $P(y = c) = \theta_c$, with $\theta_c \geq 0$ and $\sum_{c=1}^{C} \theta_c = 1$. $c$ is the class index and $C$ is the number of classes.

• We reparameterize (exponential family of distributions)

$$\theta_c = \frac{\exp(\eta_c)}{\sum_{c' = 1}^{C} \exp(\eta_{c'})}$$

• The $\eta_c$ are unconstrained; softmax notation: $\theta_c = \text{softmax}_c(\vec{\eta}_c)$
Multiple Classes and Softmax: GLM

- In GLM, we set $\eta_c = x^T w_c$ and

$$\hat{y}_c = P(y = c|x) = \frac{\exp(x^T w_c)}{\sum_{c'=1}^{C} \exp(x^T w_{c'})}$$

- In passing: With $\hat{y} = Wx$ we would have a model that is linear in $x$ with normalization constraints on $W$ (see Markov chain); with softmax $\hat{y} = \text{softmax}(Wx)$ we have a model that is nonlinear in $x$ with no normalization constraints on $W$
Multiple Classes and Softmax (cont’d)

- The negative log-likelihood (softmax cross entropy) becomes

\[
-l = - \sum_{i=1}^{N} \left( \sum_{c=1}^{C} y_{i,c} x_i^T w_c - \log \sum_{c=1}^{C} \exp(x_i^T w_c) \right)
\]
Multiple Classes and Softmax (cont’d)

• The gradient becomes

\[- \frac{\partial l}{\partial w_{j,c}} = - \sum_i \left( y_{i,c} x_{i,j} - \frac{x_{i,j} \exp(x_i^T w_c)}{\sum_{c=1}^C \exp(x_i^T w_c)} \right)\]

and SGD becomes

\[w_{j,c} \leftarrow w_{j,c} + \eta x_{i,j} (y_{i,c} - \hat{y}_{i,c})\]
III. Classification via Regression

- Linear Regression:

\[ f(x_i, w) = w_0 + \sum_{j=1}^{M-1} w_j x_{i,j} \]

\[ = x_i^T w \]

- We define as target \( y_i = 1 \) if the pattern \( x_i \) belongs to class 1 and \( y_i = 0 \) (or \( y_i = -1 \)) if pattern \( x_i \) belongs to class 0

- We calculate weights \( w_{LS} = (X^T X)^{-1} X^T y \) as LS solution, exactly as in linear regression

- For a new pattern \( x \) we calculate \( f(x) = x^T w_{LS} \) and assign the pattern to class 1 if \( f(x) > 1/2 \) (or \( f(x) > 0 \)) ; otherwise we assign the pattern to class 0
Bias

- Asymptotically, a LS-solution converges to the posterior class probabilities, although a linear function is typically not able to represent $P(c = 1|x)$. The resulting class boundary can still be sensible.

- One can expect good class boundaries in high dimensions and/or in combination with basis functions, kernels, and neural networks; in high dimensions sometimes consistency can be achieved. In essence it is necessary that the linear model can model the expected probability $P(c = 1|x)$. 
Classification via Regression with Linear Functions
Classification via Regression with Radial Basis Functions
Causal Effect

- Assume that all relevant inputs are considered in the model (no other confounders) and that we use “Classification via Regression”

- The causal effect is independent of the individual, and can be estimated as

\[ P(Y = 1|x : x_1 = 1) - P(Y = 1|x : x_1 = 0) = w_1 \]

- \( x_1 = 1 \) means that the individual has received the treatment, and \( x_1 = 0 \) means that the individual has not received the treatment,

- \( Y = 1 \) means that the patient is healthy after the treatment
Performance

- Although the approach might seem simplistic, the performance can be excellent (in particular in high dimensions and/or in combination with basis functions, kernels and neural networks). The calculation of the optimal parameters can be very fast!

- Regression is commonly used in treatment effect prediction in medicine if the influence of the treatment is small, on average
Markov Chain and Quantum Hamiltonian

- Consider the marginalization \( P(Y = k) = \sum_j P(Y = k | X = j) P(X = j) \)
- With \( x_j = P(X = j) \) and \( y_k = P(Y = k) \) and \( W_{k,j} = P(Y = k | X = j) \), we can write
  \[
  y = Wx
  \]

- \( y \) is a linear superposition of the columns of \( W \); \( W \) is a stochastic matrix

- This marginalization is important in Markov chains and hidden Markov models: \( P(X) \) describes the state at time \( t \) and \( P(Y) \) describes the state at time \( t + 1 \)

- In quantum mechanics, all entries are complex and, e.g., \( W \) is a (unary) Hamiltonian matrix, \( x \) is the initial quantum state and \( y \) is the quantum state after applying the Hamiltonian; this is the linearity of quantum mechanics
Logistic Regression in Medical Statistics
Logistic Regression in Medical Statistics

- Logistic regression has become one of the most important tools in medical statistics to analyse the outcome of treatments, e.g., a new medication, and to evaluate the effect of preconditions (gender, age, smoking, environmental effects).

- An important task is to distinguish between correlation and causation.
Epidemiology

- In epidemiology, the output $y = 1$ means that the patient has the disease.

- $x_1 = 1$ might represent the fact that the patient was exposed (e.g., by a genetic variant, smoking, or an environmental factor) and $x_1 = 0$ might mean that the patient was not exposed; the other inputs are often typical confounders (age, sex, ...)

$$P(y_i = 1|x_i, w) = \text{sig} \left( \sum_{j=0}^{M} w_j x_j \right)$$

- Thus, $w_1$ is the quantity of interest! If $w_1$ is significantly larger than zero, then the exposure was harmful!

- For model fitting we need data from individuals, which were randomly chosen out of the population; for rare diseases, this can be a problem (see later discussion on the log-odds ratio).
Treatment Evaluation

- All individuals in the population have the disease

- In treatment evaluation, \( x_1 = 1 \) means that the patient received the treatment, and \( x_1 = 0 \) means that the patient did not receive the treatment

- The output represents the outcome after treatment; e.g., \( y = 1 \) can mean that the patient is cured by the treatment

\[
P(y_i = 1 | x_i, w) = \text{sig} \left( \sum_{j=0}^{M} w_j x_j \right)
\]

- Of course, of great interest is if \( w_1 \) is significantly nonzero
Causal Effect Depends on the Individual

- In the model, the causal effect depends on the individual,

\[ P(y = 1|x : x_1 = 1) - P(y = 1|x : x_1 = 0) \]

\[ = \text{sig}(x : x_1 = 1) - \text{sig}(x : x_1 = 0) \]

- We can calculate the average causal effect, e.g., on subgroups (stratification))

- Maybe we can also find an interpretation of \( w_1 \), which we analyse next
Log-Odds

- The **odds** for a patient with properties $x_i$ is defined as

\[
Odds(x_i) = \frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)}
\]
Log-Odds for Logistic Regression

• In medical statistics, one is interested in the interpretation of the terms in logistic regression

• For logistic regression, the log odds is

\[
\text{LogOdds} = \log \frac{P(y_i = 1|x_i)}{P(y_i = 0|x_i)} = \log \frac{1}{1 + \exp(-x_i^Tw)}
\]

\[
= \log \frac{1}{\exp(-x_i^Tw)} = x_i^Tw
\]

• Thus the log odds of the outcome is \( h = x_i^Tw \), which is the net input, also called the logit (with \( y = \text{sig}(x) \), \( x = \text{sig}^{-1}(y) = \log(y) - \log(1 - y) \))

• Thus logistic regression is linear in predicting the log odds
The odds ratio is defined as

\[ OR = \frac{Odds(x_{x_1=1})}{Odds(x_{x_1=0})} \]

The log odds ratio evaluates the effect of the treatment

\[ \log(OR) = \log Odds(x_{x_1=1}) - \log Odds(x_{x_1=0}) \]
The Log Odds Ratio for Logistic Regression

- In logistic regression, the log odds ratio is identical to $w_1$, since

\[(w_0 + w_1 + \sum_{j=2}^{N} x_{i,j}) - (w_0 + 0 + \sum_{j=2}^{N} x_{i,j}) = w_1\]

- If $w_1$ is significantly nonzero, then the exposure/treatment has an effect.

- Thus, in logistic regression, the causal effect might be different for each individual, the log odds ratio is the same; $w_1 \geq 0$ is the increase in the Log Odds of a patient that obtains the treatment.
Case Control Studies and Imbalanced Classes

- Consider a rare disease that only affects one in a million; then if I would collect data from 1 million random individuals I might only have one individual with the disease.

- It is easier to collect let’s say 1000 individuals who have the disease (e.g., breast cancer) and 1000 who do not have the disease; in general, selecting data based on the output can be dangerous.

- Fortunately, if the individuals in both groups are very similar (e.g., both are women of a certain age, ...), then $w_1$ obtained in logistic regression, i.e. the log odds ratio, is insensitive to the class proportions (see our discussion on generative models for classification) and is also meaningful!
Causality

• Confounders are variables that influence the output $y$ and $x_1$

• For example R. A. Fisher argued that there might be a genetic variant which makes you want to smoke and which gives you lung cancer; thus you would get lung cancer independently if you smoked

• This turned out to be (mostly) untrue

• If possible confounders should be inputs to the model (age, gender, ...), or one does a separate model for each subgroup (stratification), thus a separate model for each age/gender class

• By far the most apparent paradoxes result from unmodelled confounders (Simpson’s paradox)
Personalized Medicine

- A linear model assumes that the effect of an input on the output is independent of the other inputs.
- A log-linear model assumes that the effect of an input on the log-odds of the output is independent of the other inputs.
- The idea behind personalized medicine is that a given medication only works for a subclass of the population.
- Thus one either tries to identify as good as possible the group (strata) for which the medication works.
- If many factors might contribute to the effectiveness of a drug, one might try multivariate nonlinear models, e.g., neural networks.