Multivariate Verfahren

6. Supervised Learning

Hannah Schulz-Kümpel

Institut für Statistik, LMU München

Sommersemester 2024

Contents

1. Introduction
   - The ultimate probabilistic question: Bayesian or Frequentist?
2. 6.1 Multivariate Regression
   - Generalized Linear models
3. 6.2 Classification Methods
   - Introduction to Classification
     - Foundation of generative models
     - How can one classify based on $P(Y \mid X)$?
   - Logistic regression
   - Naive Bayes (NB)
   - Discriminant Analysis
     - Linear discriminant analysis (LDA)
     - Quadratic discriminant analysis (QDA)
   - Connection between log.reg., NB, LDA, and QDA
   - Outlook: Decision Trees and SVMs

What is supervised learning? I

The term *supervised learning* describes the following setting for statistical inference:

- The given data contains clearly labelled input (feature/independent variable) and output (target/dependent variable) elements.

When writing data as $\mathcal{D} = (x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, the y_i are usually the targets and the x_is the feature vectors. Additionally, in machine learning (ML), the outputs are sometimes called the *labels* of the inputs.

- The goal of a supervised learning model is to characterize the relationship between the input and output that allows for drawing conclusions about a new output observation given the corresponding input observations.

What is supervised learning? II

- All supervised methods share a common framework: the goal is to "learn" a function from a set $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$.
- Here, f_θ is a function of the inputs that is intended to return (a transformation of) the output, and all that is unknown about it is the value of θ.
- Usually, $\mathcal{F}$ should also be defined in a way so that the preimage or domain of all f_θs contains $\mathcal{X}$ and the image of all f_θs contains $\mathcal{Y}$. Common examples would be
  - $f_\theta : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, for $d_1, d_2 \in \mathbb{N}_{>0}$
  - $f_\theta : \mathbb{R}^d \to \{0, 1\}$ (or $[0, 1]$)
  - $f_\theta : \mathbb{R}^{d_1} \times \Omega \to \mathbb{R}^{d_2}$, for $d_1, d_2 \in \mathbb{N}_{>0}$ and some (product-) space Ω of categorical inputs.

What is supervised learning? III

- The goal of any supervised learning algorithm is to find the "optimal" $f_\theta \in \mathcal{F}$ by choosing θ accordingly.
- Here, again, one can choose to *utilize probabilistic modelling or approach the problem from a simply geometric/algebraic perspective*.
- Either way, we need a loss function - i.e. a distance or similarity measure between the value of f_θ and the output; with the kind of measure depending on the perspective that one is taking.

What is supervised learning? IV

- A classical approach to choosing the "optimal" f_θ *using a purely geometric/algebraic perspective* is choosing θ as

$$\theta^* = \text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} L\left(f_\theta\left(x_i\right), y_i\right) .$$

- Meanwhile, when choosing the "optimal" f_θ *using probabilistic approach*, we **require a Likelihood function** $\mathcal{L}(\theta; x)$.

- Note that, last lecture, we showed that OLS (wich is equivalent to the ML approach), which utilizes squared loss, is equivalent to minimizing the KL divergence for i.i.d. data and finite dimensional vector θ.

A philosophical debate in probabilistic thinking

- A fundamental philosophical question in probability and statistics is

"What does probability quantify?"

A philosophical debate in probabilistic thinking

- A fundamental philosophical question in probability and statistics is

  "What does probability quantify?"

- Here are the two most popular answers:
  1. The frequency at which an event will (at least asymptotically) happen # **Frequentistic**
  2. One's own uncertainty about an event happening # **Bayesian**

A philosophical debate in probabilistic thinking

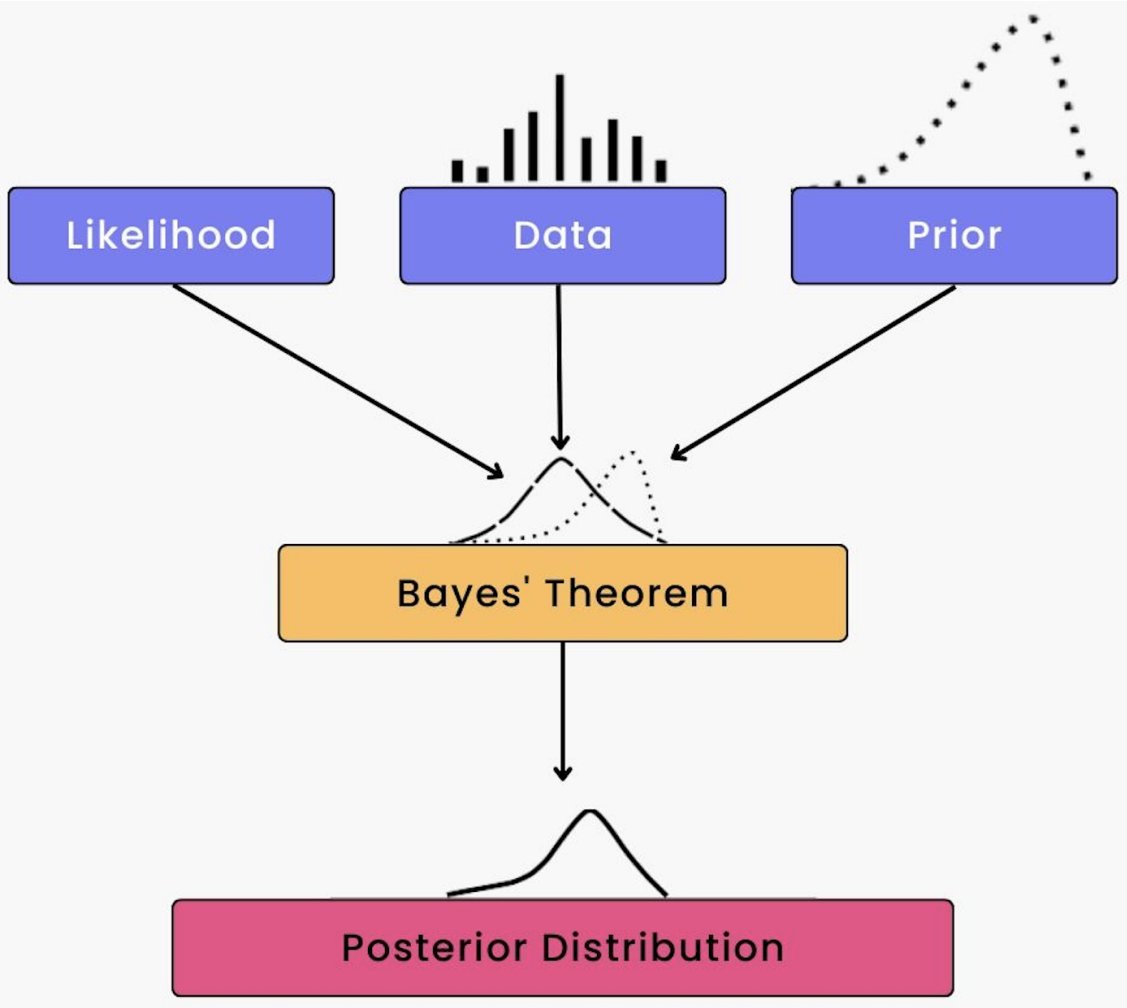
- A fundamental philosophical question in probability and statistics is

  "What does probability quantify?"

- Here are the two most popular answers:
  1. The frequency at which an event will (at least asymptotically) happen # **Frequentistic**
  2. One's own uncertainty about an event happening # **Bayesian**

- From these philosophical approaches, it follows that
  1. In frequentist inference, a fixed, **true**, but unknown parameter θ_0, the value of which we want to find, is assumed.
  2. In Bayesian inference, the parameter θ is modelled as a random variable.

Before we continue, a quick summary of Bayesian inference!



Source: https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners

A summary of Bayesian inference I

Generally, Bayesians are more concerned with the **degree of uncertainty** of their parameter estimation than a true value. This is modelled by a so called **prior distribution**. In turn, Bayesian inference consists of considering all possible parameter values for a fixed set of data.

$\Rightarrow$ Inference is then based entirely on the **posterior distribution** of the parameter given the data.

A summary of Bayesian inference I

Generally, Bayesians are more concerned with the **degree of uncertainty** of their parameter estimation than a true value. This is modelled by a so called **prior distribution**. In turn, Bayesian inference consists of considering all possible parameter values for a fixed set of data.

$\Rightarrow$ Inference is then based entirely on the **posterior distribution** of the parameter given the data.

Frequentist Inference

We start with data $\mathbf{X}$ and

- A distributional assumption given by the likelihood $L(\theta, x)$

We can immediately make inference about the "true" parameter θ_0 based on $L(\theta, x)$

- Point estimates are easily calculated, e.g. using MLE
- **Uncertainty modelling is rather complicated methodologically (the borders of CIs are random)**

Bayesian Inference

We start with data $\mathbf{X}$ and

- A sampling distribution $f(x \mid \theta)$ (= $L(\theta, x)$) and
- A prior distribution $\pi(\theta)$

  $\Rightarrow \theta$ is considered a RV

We need the posteriori for inference

Posteriori $\Pi(\theta \mid x)$

- Inference is entirely based on the posterior
- One common point estimate is the posterior mean $\hat{=}$ expected value of θ under $\Pi(\theta \mid x)$
- **Uncertainty is very straigtforward to model (e.g. using quantiles of $\Pi(\theta \mid x)$)**

A summary of Bayesian inference II

- How do we get the posterior distribution? **Bayes' theorem**:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- Specifically, given a *prior distribution* $\pi(\theta)$ and a *sampling distribution* $f(x|\theta)$ **(which, for our purposes, is always equal to the frequentist Likelihood $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x})$)**, the posterior distribution is given by

$$\Pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int_\Theta f(x|\theta) d\mu_\pi(\theta)} \Bigg\} \begin{array}{l}\text{this is the } \textit{compound probability-} \\ \text{or } \textit{mixture- distribution } f(x)\end{array}$$

- Note that

$$\int_\Theta f(x|\theta) d\mu_\pi(\theta) := \begin{cases} \int_\Theta f(x|\theta)\pi(\theta)d\theta, & \text{if } \pi(\theta) \text{ is a density,} \\ \sum_{\theta\in\Theta} f(x|\theta)\pi(\theta), & \text{if } \pi(\theta) \text{ is a probability function.} \end{cases}$$

How is a Baysian posterior distribution obtained? I

- Given that $\int_\Theta f(x|\theta) d\mu_\pi(\theta)$ is a constant, it immediately follows that

$$\Pi(\theta|x) \propto f(x|\theta)\pi(\theta)\,.$$

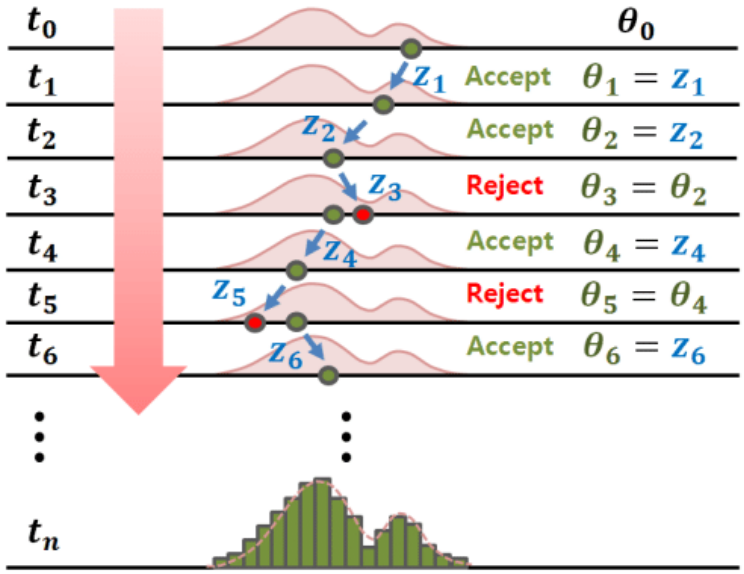
  Sometimes, this is sufficient to find the parameter θ for the optimal f_θ, e.g. by maximizing the posterior probability.

- Additionally, for some combination of prior and likelihoods, there exist theoretical results telling us which distribution family the posterior will belong to. Such priors are called *conjugate* to the distribution family of the corresponding likelihood.

How is a Baysian posterior distribution obtained? II

- Mostly, however, **Markov Chain Monte Carlo (MCMC)** are used. These algorithms are a class of methods used for sampling from probability distributions whose probability density is proportional to a known function.

MCMC algorithms exceed the scope of this class, but here is a quick visualization:



Source: Adaptive Markov chain Monte Carlo algorithms for Bayesian inference: recent advances and comparative study - Scientific Figure on ResearchGate. Available from here

Multivariate Regression models

In a typical multivariate regression setting we choose a probabilistic modelling approach and consider have output Y and input vector X of $p \in \mathbb{N}$ regressors with the following relation:

$$\mathbb{E}[Y|X] = f_\theta(X)\,.$$

One of the most common type of multivariate regressions is the following:

Definition (Generalized Linear Model (GLM))

A *generalized linear model (GLM)* is a multivariate regression model consisting of the following elements:

1. A **linear predictor** $\eta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i \cdot X_i$
2. A **link function** g, which, together with the linear predictor defines the regression function via $f_\theta := g^{-1} \circ \eta$.
3. A **distributional assumption** $Y \sim \mathcal{D}$.

Generalized Linear Models

- In a GLM, the pdf or pmf of the assumed distribution $\mathcal{D}$ should have the expected value $\mathbb{E}[Y|X]$ as a parameter.

- Let's write $\mu = \mathbb{E}[Y|X]$ and denote by
  - ν the vector of remaining distribution parameters (which may well be an empty set) and
  - $p_Y(\mu, \nu)$ the pdf or pmf of $\mathcal{D}$.

- Then, the likelihood is always given by

$$\mathcal{L}(\theta; x_1, \ldots, x_n) = \prod_{i=1}^{n} p_Y\left(\mu = f_\theta(x_i), \nu\right).$$

- The following slides now present three popular GLMs.

Linear regression

- Linear regression is a generalized linear model with the following specifications:
  1. The **link function** g,is chosen as the identity function, so
  $$f_\theta := \eta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i \cdot X_i \,.$$
  2. The **distributional assumption** $Y \sim \mathcal{N}(\mathbb{E}[Y|X], \sigma^2)$

- Note that when applying OLS, we are only estimating θ to optimize $\mathbb{E}[Y|X]$, so σ^2 has to be estimated separately.

- A nice asset of linear regression is how easily the β coefficients may be estimated: "*With all other features/regressors staying equal, the conditional expectation of Y is expected to be β_i greater if X_i is increased by 1*".

Poisson regression

- The poisson GLM is used in situations where the output/target/label is a count variable, which means it represents the number of times an event occurs within a fixed interval of time, space, or other contexts.

- Poisson regression is a generalized linear model with the following specifications:
  - The **link function** g,is chosen as the $\log$ function, so
  $$f_\theta := e^{\eta(X)}\,.$$
  - The **distributional assumption** $Y \sim \text{Pois}(\mathbb{E}[Y|X])$

- Additionally, for T denoting the number of time etc. intervals, the linear predictor is expanded to
$$\eta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i \cdot X_i + \log(T)\,.$$

Logistic regression I

- Logistic regression usually refers to a GLM with Binomial distributional assumption an logit link-function.
- It models the probability **one** of two mutually exclusive events taking place.
- Specifically, the target variable Y is assumed to follow a Binomial distribution $\mathrm{Bin}\left(n, p(X)\right)$, where the parameter p is modelled as a function of the vector of regressors, with each of the the rows $\boldsymbol{x}_i$, $i = 1, \ldots, n$, of the data matrix $\boldsymbol{X}$ being an observation of an independent copy of X.

Logistic regression II

- n *could* be any natural number. However, in many contexts, logistic regression simply refers to the case $\boldsymbol{n = 1}$, i.e. $Y \sim \text{Ber}\left(p(X)\right)$.

- For those who are interested: logistic regression with $n \in \mathbb{N}_{>1}$ may easily be fit
  - using the "Wilkinson-Rogers" format with the `glm`-function of the R-package `stats` (frequentist inference) or
  - most tools for Bayesian inference.

- In regression settings with a dichotomous target variable (i.e. a categorical variables with two categories or levels) where the goal is to make inference about the probability of one of these categories occurring, logistic regression is defined as follows:

Logistic regression III

Logistic regression with dichotomous target variable

We assume that the target variable (recoded as $Y \in \{0, 1\}$, if necessary) is a random variable with $Y \sim \text{Ber}\left(p(X)\right)$, where X is the vector of $p \in \mathbb{N}$ regressors that takes values in $\mathbb{R}^p$. Additionally,

- The model has a logit link function, i.e. we assume

$$\text{logit}\left(p(X)\right) = \eta(X) := \beta_0 + \sum_{i=1}^{p} \beta_i \cdot X_i$$

$$\Leftrightarrow \mathbb{E}[Y|X] = p(X) = \text{inv. logit}\left(\eta(X)\right) = \frac{e^{\eta(X)}}{\left(1 + e^{\eta(X)}\right)}.$$

- Note that, since p is the only parameter of the Bernulli distribution, no additional assumptions are necessary.

- The likelihood of this model is given by

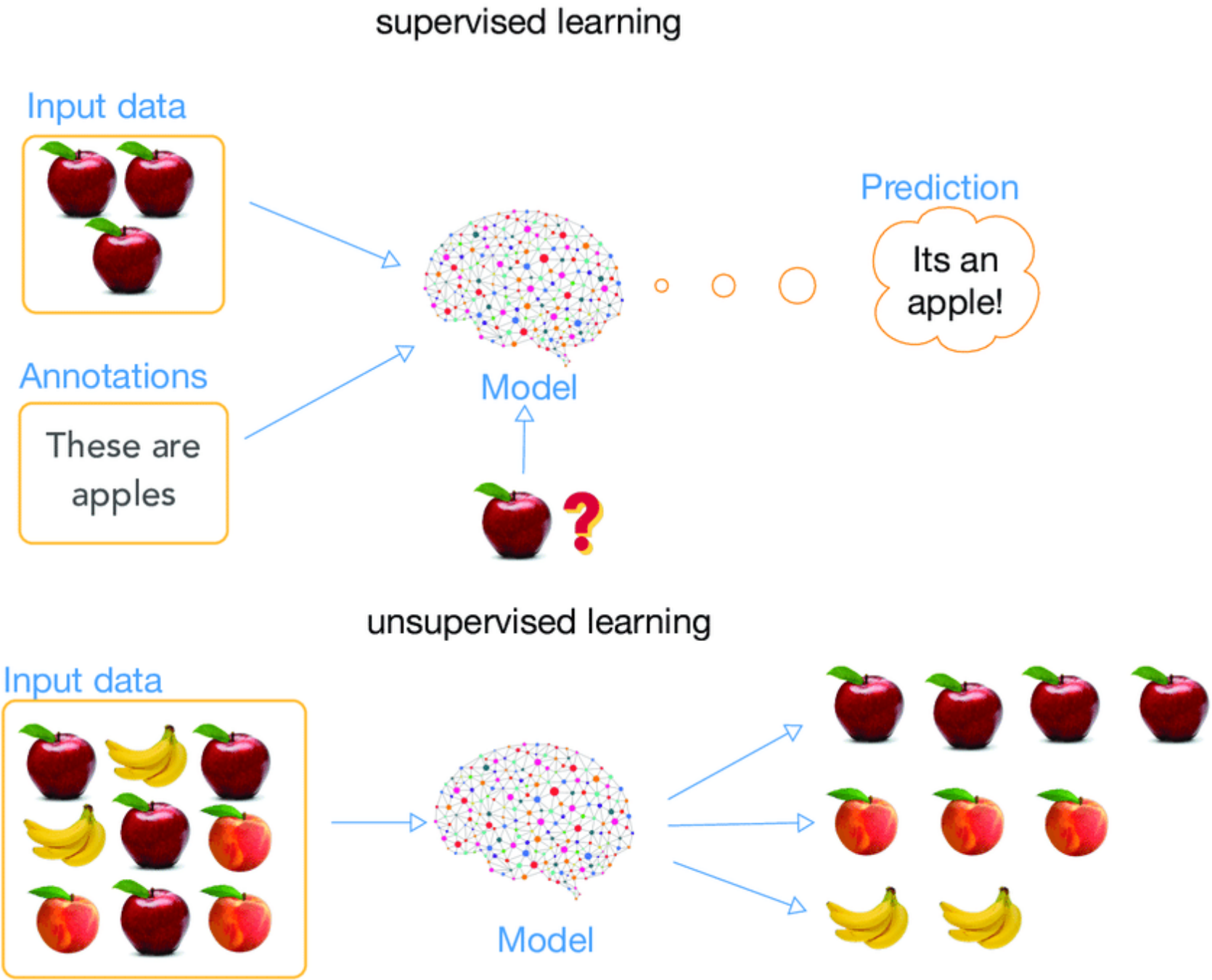
$$\mathcal{L}(\theta, x) = \prod_{i=1}^{n} p(\boldsymbol{x}_i)^{y_i} \left(1 - p(\boldsymbol{x}_i)\right)^{1-y_i}.$$

Introduction I

- In the context of Machine Learning (ML), **classification** refers to models that assign observations to **exactly one of** $K \in \mathbb{N}$ different classes/categories based on their features/variables/column entries.

- Of course, it is important to remember that some models may be used for a variety of purposes, so whether a specific method, s.a. *logistic regression* is referred to as classification method or simply regression/dimension reduction technique/etc. really depends on the scientific context and what it is being used for.

- A distinction that is often clearly possible, however, is the one between **supervised learning** (which is used as synonym for classification) and **unsupervised learning**. We recall:


supervised learning
Input data
Annotations
These are apples
Model
Prediction
Its an apple!
unsupervised learning
Input data
Model

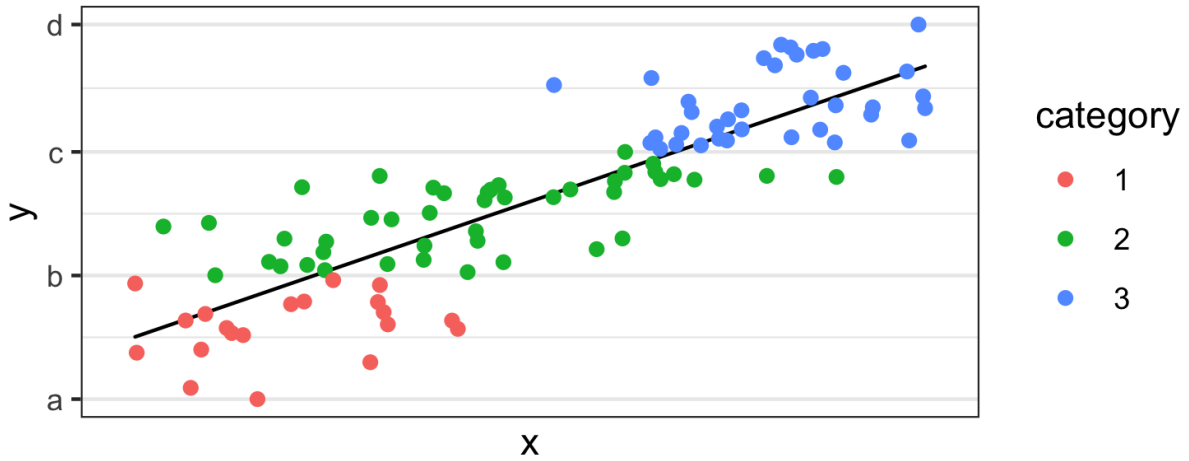
Introduction II

- You may think of classification as *regression with a categorical target variable*.
- Indeed, we *hypothetically could* turn any regression with a metric target variable into a classifier by mapping the expected value of the target variable (often called *model prediction*) onto a discrete space.
- For example: Consider any regression with metric target variable defined by the model function $\mathbb{E}[Y|X] = h_\theta(X)$, with $h_\theta : \mathbb{R}^p \longrightarrow \mathbb{R}$, with $p \in \mathbb{N}$ denoting the number of regressors, and the mapping, for some $a, b, c, d \in \mathbb{R}$ with $a < b < c < d$

$$m : \mathbb{R} \longrightarrow \{1, 2, 3\}, \quad x \longmapsto \begin{cases} 1, & \text{if } x \in [a, b], \\ 2, & \text{if } x \in (b, c), \\ 3, & \text{if } x \in [c, d]. \end{cases}$$

Introduction III

- In this setting, the function $m \circ h_\theta : \mathbb{R}^p \longrightarrow \{1, 2, 3\}$ clearly assigns each vector of regressors/features to one of three categories.
- Example for linear regression:



- **Of course, this example is backwards** - in supervised learning classification, our data is already classified (i.e. we have no metric target variable, but a categorical one) and we need to fit a model that assigns new observations into the categories -
for which linear regression is rarely suitable!

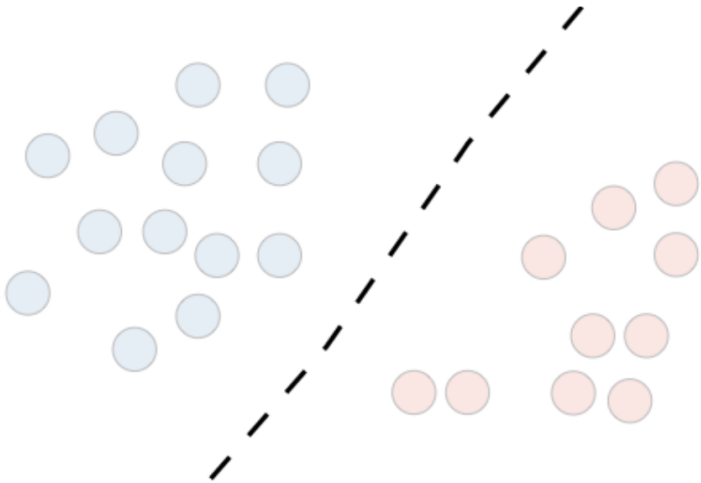
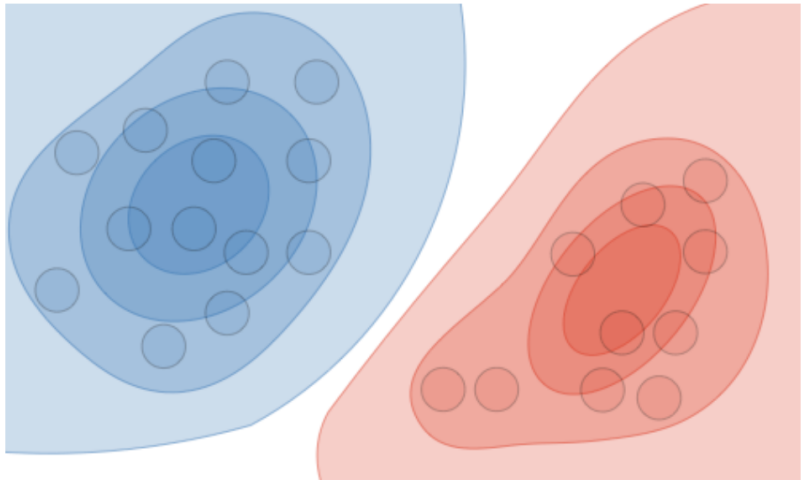
Introduction IV

Some common classification methods are

- Logistic regression
- Naive Bayes
- **Discriminant Analysis (DA)**
- Decision Trees
- Support Vector Machine

For the rest of this lecture, we will focus mostly at all highlighted methods with special focus on **Discriminant analysis and how linar discriminant analysis (LDA) can be used for dimension reduction**.

Another distinction: Discriminative vs. Generative

	Discriminative model	**Generative model**
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Source: https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning#generative-learning

Foundation of generative models I

- In generative models, the exact same principles apply, but instead of necessarily considering the parameter vector θ as a random variable, we consider all categories into which observations can be classified as *realizations of a categorical random variable* Y.

- As we are familiar with from regression, $X = (X_1, \ldots, X_p)^\top$, $p \in \mathbb{N}$, denotes the (random) vector of regressors/features/inputs, with each of the rows $\boldsymbol{x}_i$, $i = 1, \ldots, n$, of the data matrix $\boldsymbol{X}$ containing only the regressors/features/inputs as columns being an observation of an independent copy of X.

- From this, we get the following setting, where $K \in \mathbb{N}$ denotes the number of categories, which are coded as numbers from 1 to K, i.e. Y takes values in $\{1, \ldots, K\}$:

Foundation of generative models II

Generative models: Notation

$f_{X\|Y}(\boldsymbol{x}\|y)\hat{=}f(\boldsymbol{x}\|y)$	sampling distribution	(known, at least as an estimate)
$P(Y=y)\hat{=}p(y)$	a priori-probability	(known, at least as an estimate)
$f(\boldsymbol{x})=\sum_{y=1}^{K} f(\boldsymbol{x}\|y)p(y)$	mixture distribution	(can be calculated from the two previous items)
$P(Y=y\|X=\boldsymbol{x})\hat{=}p(y\|\boldsymbol{x})$	a posteriori-probability	(unknown)

Note that, for simplicity's sake, we will sometimes denote the

- the a priori-probability of the kth, $k \in \{1,\ldots,K\}$, category as π_k and
- the sampling distribution conditional on the the kth, $k \in \{1,\ldots,K\}$, category as $f_k(\boldsymbol{x})$.

How can one classify based on $P(Y \mid X)$?

- In both discriminative and generative classification models, we consider the possible categories as possible realizations of a categorical random variable Y.
- We also get an estimate of the **conditional distribution of Y given the random vector or regressors/features/inputs X.** Either
  - directly (in discriminative models)
  - from the sampling distribution $f_{X|Y}(\boldsymbol{x}|y)$ combined with the a-priori probability $P(Y)$ (in generative models)
- But how do we get a classification rule from conditional distribution of Y given X?

How can one classify based on $P(Y \mid X)$?

- In both discriminative and generative classification models, we consider the possible categories as possible realizations of a categorical random variable Y.
- We also get an estimate of the **conditional distribution of Y given the random vector or regressors/features/inputs X.** Either
  - directly (in discriminative models)
  - from the sampling distribution $f_{X|Y}(\boldsymbol{x}|y)$ combined with the a-priori probability $P(Y)$ (in generative models)
- But how do we get a classification rule from conditional distribution of Y given X?
- **Intuitively, the goal is to minimize the error rate, i.e. the rate at which observations are wrongly classified.**

Disclaimer: The issue of prediction or generalization error is not uncomplicated and not what this lecture will be concerned with. Instead, we will look at **different concrete options to derive classification rules in different settings (models) with no attempt at completeness**. Frankly, the amount of implemented options is too vast and the research regarding their attributes not entirely conclusive yet.

- In Bayes classification, for example, this is achieved by assigning a new observation to the class the posterior probability is maximal - more later.
- For now, let us focus on classification based on misclassification probabilities.
- **Throughout, we denote by $\delta : \mathbb{R}^p \longrightarrow \{1, \ldots, K\}$, $K \in \mathbb{N}$, a specific, fixed classifier, i.e. a rule that assigns an observation in $\mathbb{R}^p$, $p \in \mathbb{N}$, of X to exactly one of K categories/classes of Y.**

Misclassification probabilities I

For $g \in \mathbb{N}$ we can write all individual probabilities that interest us in a matrix/table as follows:

Classification	**Actual category**			
	$Y = 1$	$Y = 2$	...	$Y = g$
$Y = 1$	✓	$P(\delta(\mathbf{x}) = 1 \mid Y = 2)$	...	$P(\delta(\mathbf{x}) = 1 \mid Y = g)$
$Y = 2$	$P(\delta(\mathbf{x}) = 2 \mid Y = 1)$	✓	...	$P(\delta(\mathbf{x}) = 2 \mid Y = g)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Y = g$	$P(\delta(\mathbf{x}) = g \mid Y = 1)$	$P(\delta(\mathbf{x}) = g \mid Y = 2)$	...	✓

- Note that, in the context of supervised learning, the matrix that contains the frequencies instead of conditional probabilities in above table as entries (i.e. the entries the corresponding square contingency table) is called **confusion matrix**.

Misclassification probabilities II

In this context, we furthermore define

1. Confusion probability as, for $r, s \in \{1, \ldots, K\}$,

$$\varepsilon_{rs} = P\left(\delta(X) = s | Y = r\right) = \begin{cases} \int_{\{\mathbf{x}:\delta(X)=s\}} f(\mathbf{x}|r) d\boldsymbol{x}, & \text{if } f(\mathbf{x}|r) \text{ is a pdf,} \\ \sum_{\{\mathbf{x}:\delta(X)=s\}} f(\mathbf{x}|r), & \text{if } f(\mathbf{x}|r) \text{ is a pmf.} \end{cases}$$

2. Misclassification probability, given category/class $r \in \{1, \ldots, K\}$

$$\varepsilon_r = P\left(\delta(X) \neq r | Y = r\right) = \sum_{\substack{r,s \in \{1,\ldots,K\}, \\ r \neq s}} \varepsilon_{rs} \,.$$

Misclassification probabilities III

3. (Total) Error rate as

$$\varepsilon = P\left(\delta(X) \neq Y\right) .$$

4. Missclassification, given $\mathbf{x} \in \mathbb{R}^p$

$$\begin{aligned} \varepsilon(\mathbf{x}) &= P\left(\delta(X) \neq Y | X = \mathbf{x}\right) \\ &= 1 - P\left(\delta(X) = Y | X = \mathbf{x}\right) . \end{aligned}$$

Misclassification probabilities IV

The following holds, for $r \in \{1, \ldots, K\}$

$$\varepsilon = P\left(\delta(X) \neq Y\right) = \sum_{r=1}^{K} P\left(\delta(X) \neq r | Y = r\right) p(r)$$
$$= \sum_{r=1}^{K} \varepsilon_r p(r) = \sum_{r=1}^{K} \sum_{s \neq r} \varepsilon_{rs} p(r)\,.$$

and

$$\varepsilon = P\left(\delta(X) \neq Y\right) = \int P\left(\delta(X) \neq Y | X = \mathbf{x}\right) f(\mathbf{x}) d\mathbf{x}$$
$$= \int \varepsilon(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}\,.$$

Bayes and ML classification I

- **Bayes classification:** Assign the object with feature vector $\mathbf{x}$ to the class/category for which the a posteriori probability is maximal, i.e.:

$$\delta(\boldsymbol{x}) = r \;\Leftrightarrow\; P(r|\mathbf{x}) = \max_{j\in\{1,\ldots,K\}} P(j|\mathbf{x}) \quad r \in \{1,\ldots,K\},\ \boldsymbol{x} \in \mathbb{R}^p\,.$$

- **ML classification:** Assign the object with feature vector $\mathbf{x}$ to the class/category for which density is maximal, i.e.:

$$\delta_{\mathsf{ML}}(\mathbf{x}) = r \;\Leftrightarrow\; f(\mathbf{x}|r) = \max_{j\in\{1,\ldots,K\}} f(\mathbf{x};j) \quad r \in \{1,\ldots,K\},\ \boldsymbol{x} \in \mathbb{R}^p\,.$$

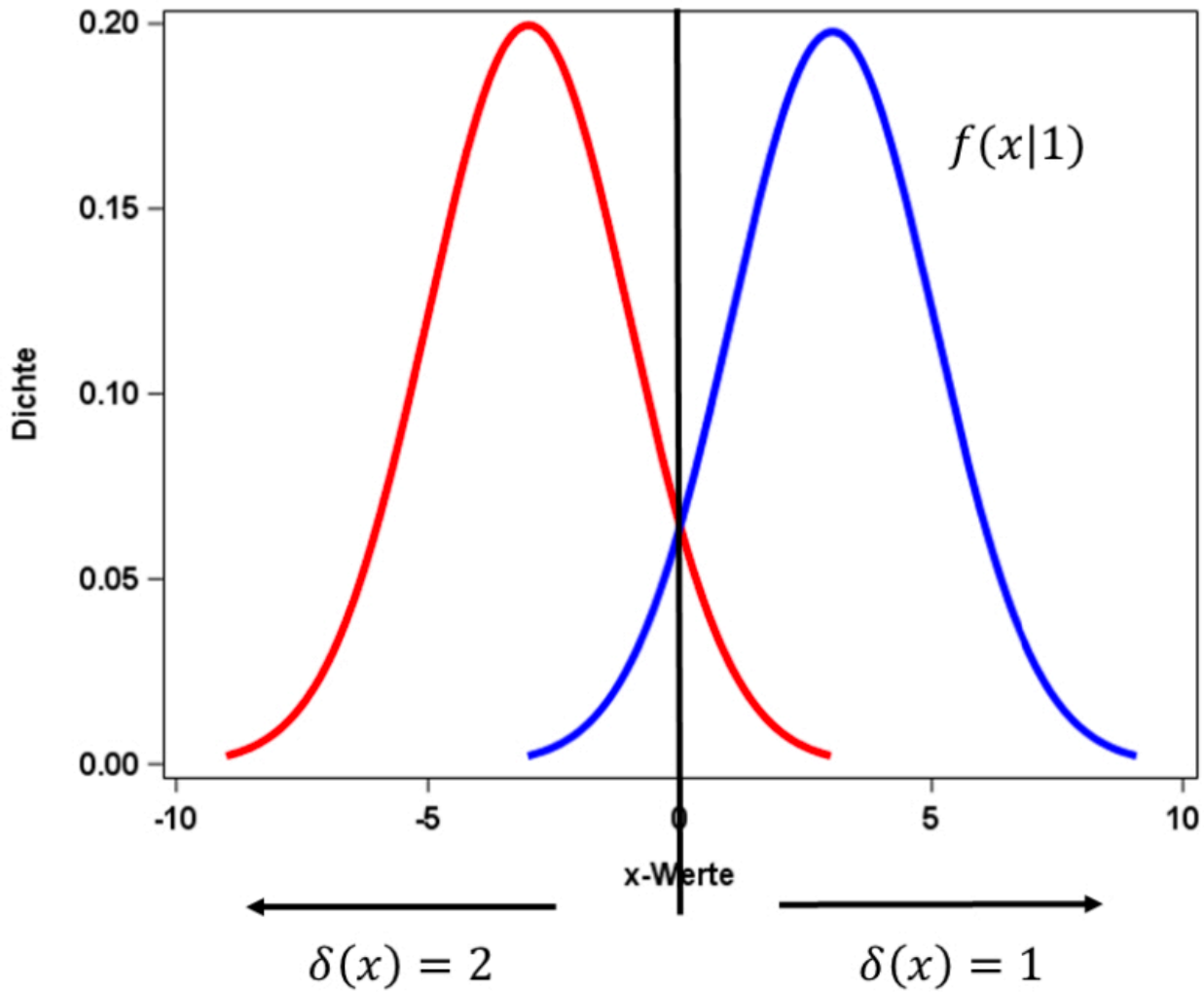
Note: ML classification is equivalent to Bayes classification without considering the a-priori probabilities or assuming equal a prior probabilities, i.e. $p(1) = \ldots = p(K) = \frac{1}{K}$.

Bayes and ML classification II

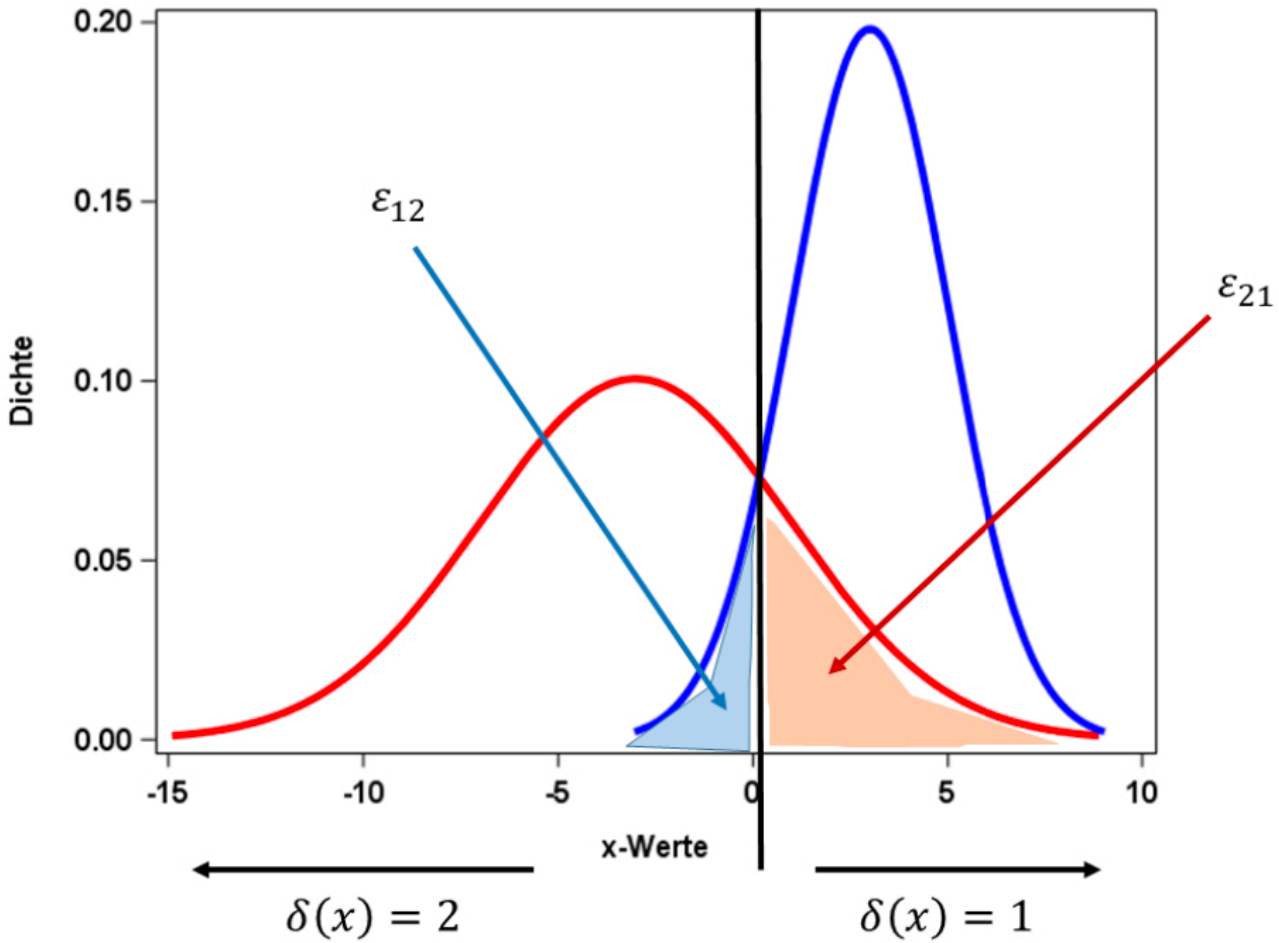
- The total error rate ε becomes minimal if $\varepsilon(\mathbf{x})$ is minimal for all $\mathbf{x}$.
- Thus, the best rule in terms of the smallest possible total error rate is obtained by minimizing $\varepsilon(\mathbf{x}) = 1 - P\left(\delta(X)|X = \mathbf{x}\right)$.

$\Rightarrow$ Bayes classification minimizes the total error rate ε.

Bayes and ML classification III



Bayes and ML classification IV



Bayes and ML classification V

- In certain settings, it may be suitable to assign specific confusions more or less weight, which is achieved by defining a cost function

$$c : \{1, \ldots, K\} \times \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}, \quad (r, \hat{r}) \mapsto c_{r\hat{r}} \,,$$

  which gives the cost of assigning an object of class r to class $\hat{r}$ ($\hat{=}$ risk or damage).

- Clearly, it holds that $c_{r\hat{r}} \geq 0$ and $c_{rr} = 0$ for any cost function c.

Bayes and ML classification VI

- To determine the classifier δ, first consider the conditional risk given X, defined as $r(\mathbf{x}) = \sum_{r=1}^{K} c_{r,\delta(X)} P(r|\mathbf{x})$

- The total risk is then calculated as follows:

$$R = \mathbb{E}_X \left(\sum_{r=1}^{K} c_{r,\delta(X)} P(r|\mathbf{x}) \right)$$
$$= \int \sum_{r=1}^{K} c_{r,\delta(X)} P(r|\mathbf{x}) f(x) dx = \int r(x) f(x) dx$$

$\longrightarrow$ Minimizing $r(\mathbf{x})$ for each $\mathbf{x}$ results in a minimization of the total risk R.

Bayes and ML classification VII

- I.e. a new observation $\boldsymbol{x}$ is classified into a category s.t. the cost is minimal:

$$\delta_C(\mathbf{x}) = r \;\Leftrightarrow\; \sum_{k=1}^{K} c_{kr} P(k|\mathbf{x}) = \min_{j \in \{1,\ldots,K\}} \sum_{k=1}^{K} c_{kj} P(k|\mathbf{x}), \quad r = 1, \ldots, K\,.$$

- Special cases:
  - $c_{r\hat{r}} = c,\ r \neq \hat{r}$, i.e. each confusion has the same cost.
    $\Rightarrow$ Bayes classification
  - $c_{r\hat{r}} = \frac{c}{p(r)}$, i.e. the cost is proportional to the proportion of the classes in the training data.
    $\Rightarrow$ ML classification

Logistic regression for classification

How can logistic regression be used for classification?

Logistic regression for classification

How can logistic regression be used for classification?

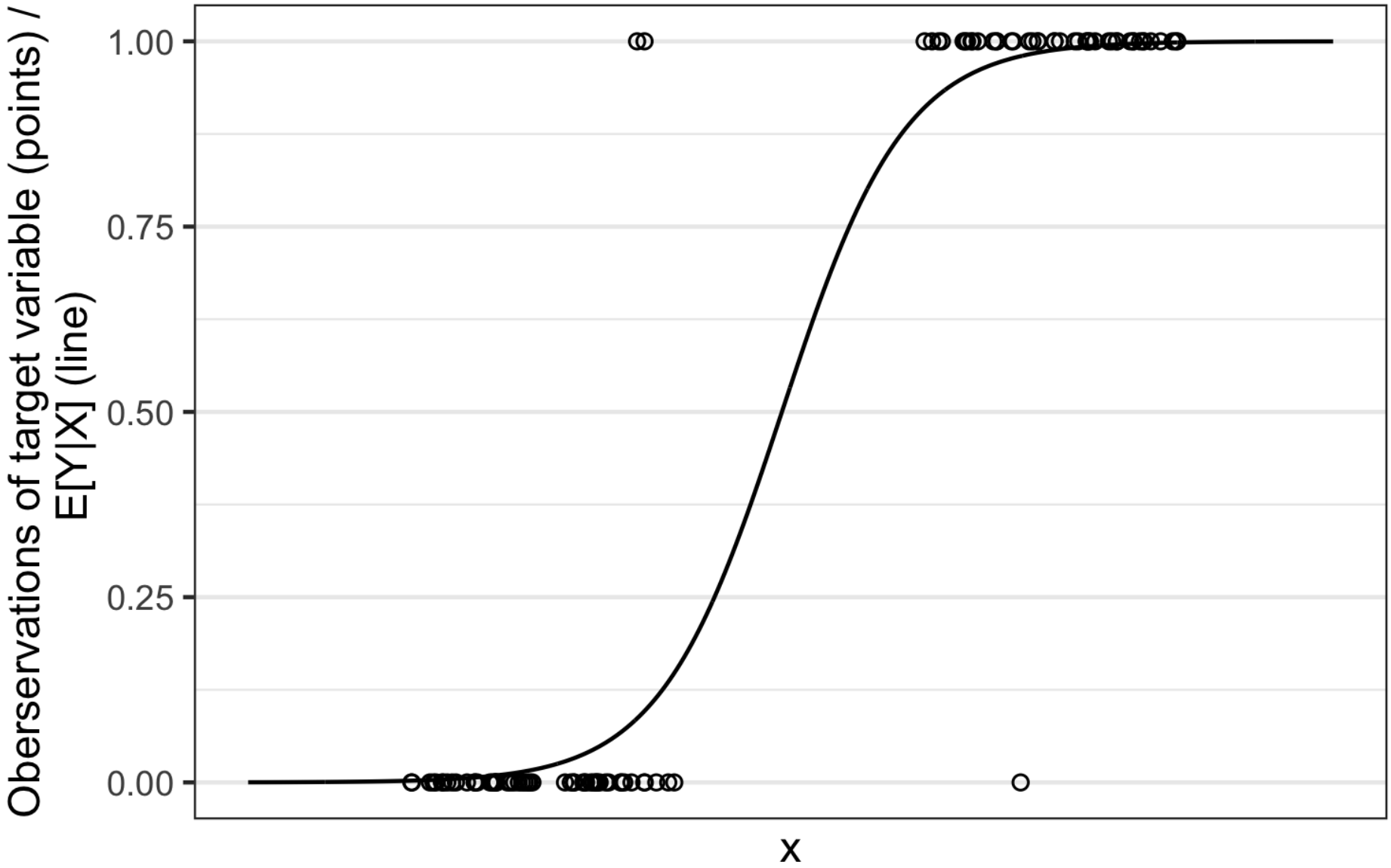
If we "**select*** a threshhold value" in $[0, 1]$ for $p(X)$, we automatically have a binary classification model, specifically a **discriminative** one.

Example:

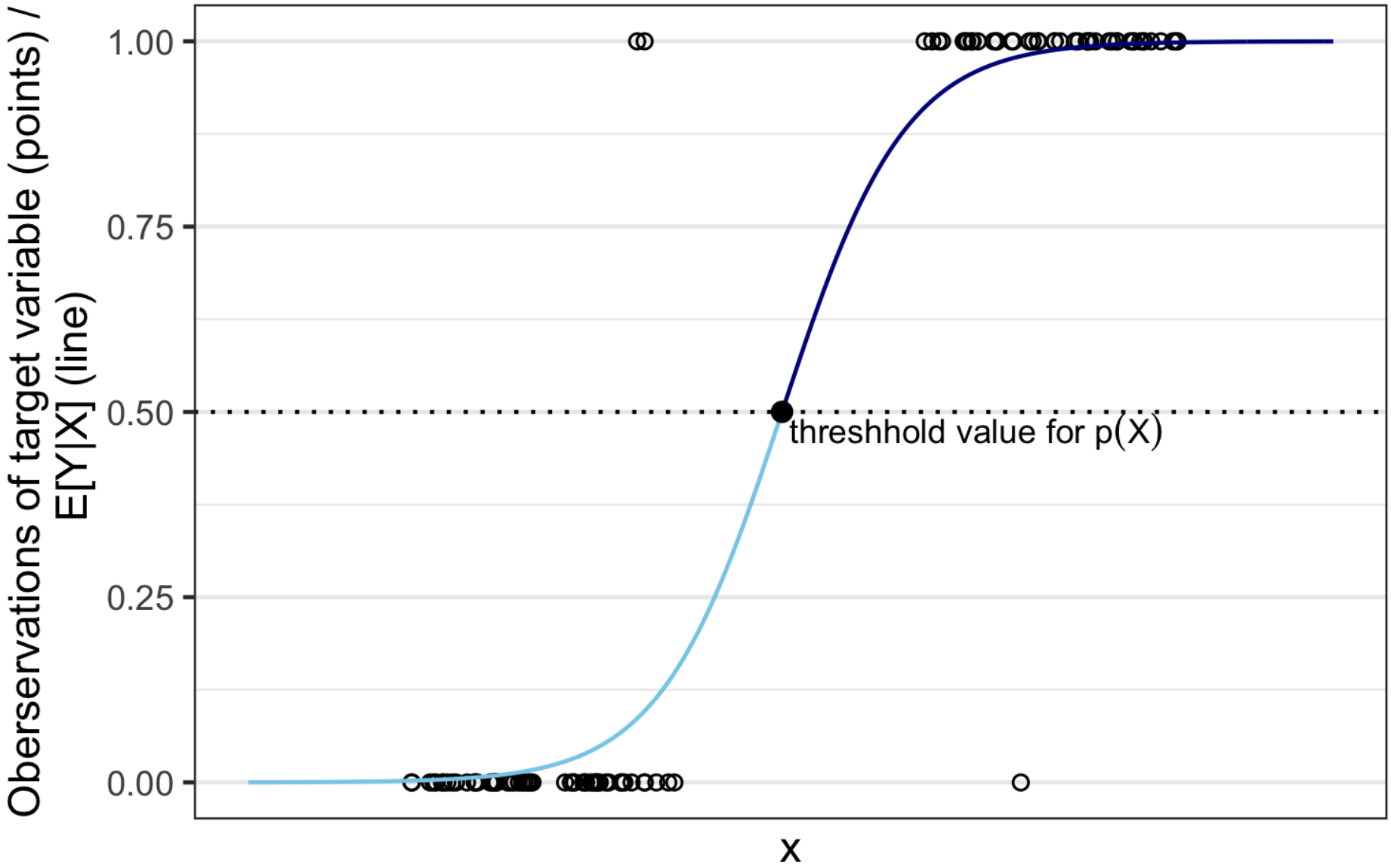
```
library(ggplot2)
library(boot)
library(stats)

set.seed(2023)
x1<-c(runif(55,0,0.35),runif(45,0.65,1))
data<-data.frame(x=x1,y=sapply(x1,function(x)rbinom(n=1,size=1,p=inv.logit(-6+12*x))),
                 category=as.factor(ifelse(inv.logit(logistic$coefficients[1]+logistic$coefficients[2]*x1)
                 <0.5,1,2))
                                  )
logistic<-glm(y~x,binomial(),data)
```

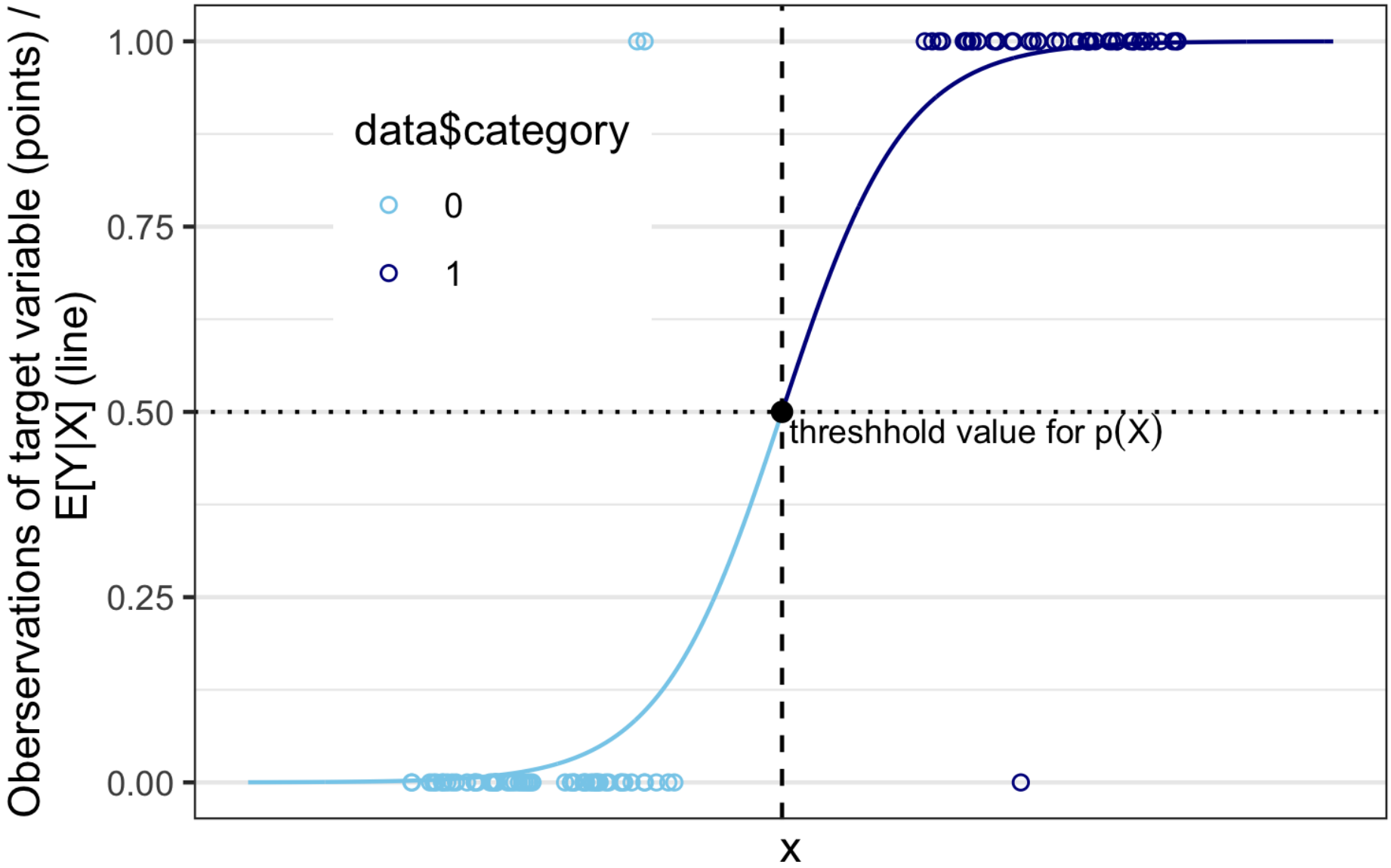
Logistic regression: Example with one regressor



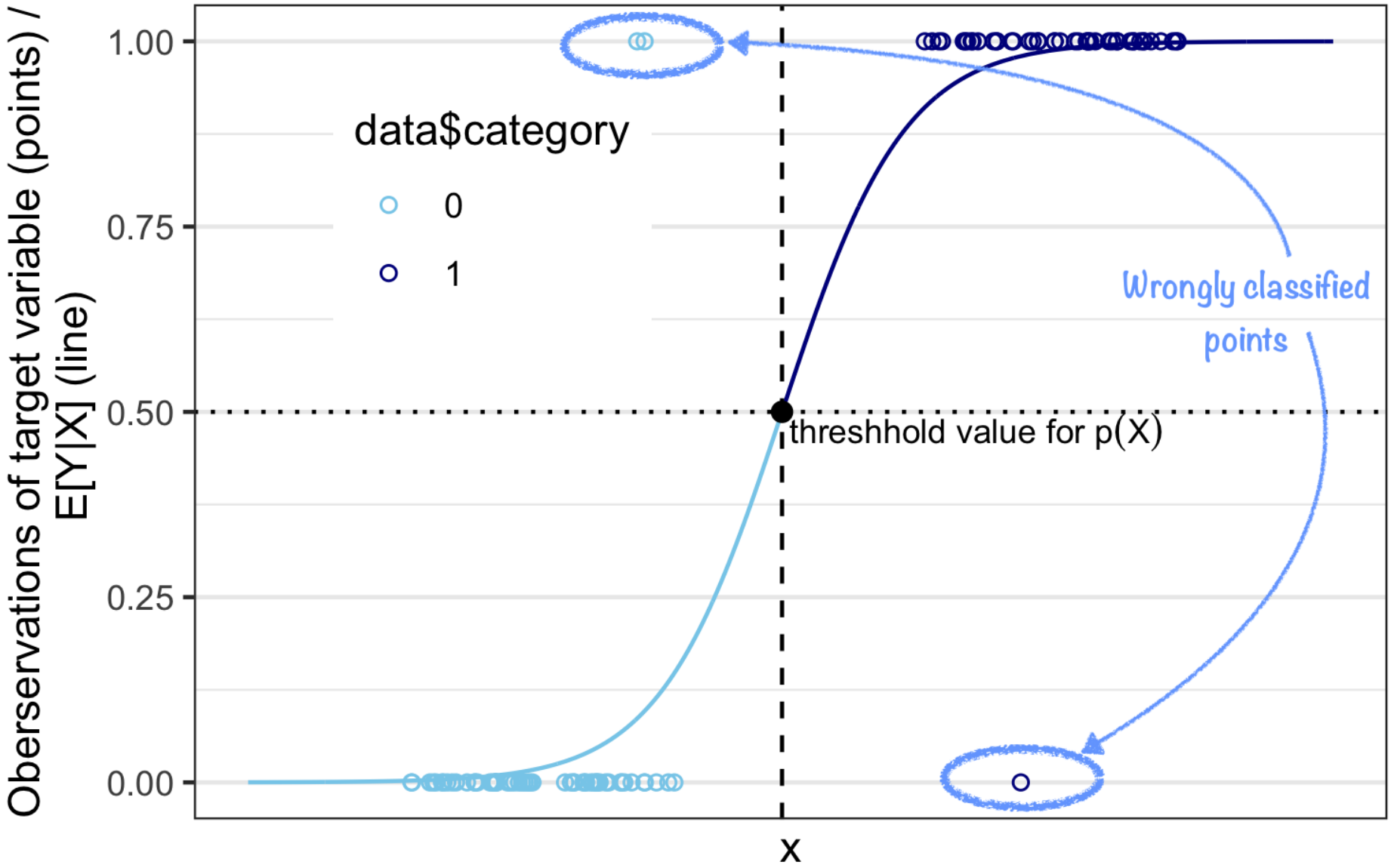
Logistic regression: Example with one regressor



Logistic regression: Example with one regressor



Logistic regression: Example with one regressor



How do we "select" the threshhold? I

There are different ways of selecting the threshhold.
A popular one is using the *Receiver Operating characteristic (ROC)*, which plots the *true positive rate* against the *false positive rate*.

Definition (True- and False positive rate)

Given a classification setting in which there are only two possible categories (coded as 0 and 1), we define the

- **true positive rate** as

$$\text{TPF}(T) = \begin{cases} \int_{\{\mathbf{x}\colon f(\mathbf{x}|Y=1)>T\}} f(\mathbf{x}|Y=1)d\boldsymbol{x}, & \text{if } f(\mathbf{x}|r) \text{ is a pdf,} \\ \sum_{\{\mathbf{x}\colon f(\mathbf{x}|Y=1)>T\}} f(\mathbf{x}|Y=1), & \text{if } f(\mathbf{x}|r) \text{ is a pmf.} \end{cases}$$

- **false positive rate** as

$$\text{FPR}(T) = \begin{cases} \int_{\{\mathbf{x}\colon f(\mathbf{x}|Y=0)>T\}} f(\mathbf{x}|Y=0)d\boldsymbol{x}, & \text{if } f(\mathbf{x}|r) \text{ is a pdf,} \\ \sum_{\{\mathbf{x}\colon f(\mathbf{x}|Y=0)>T\}} f(\mathbf{x}|Y=0), & \text{if } f(\mathbf{x}|r) \text{ is a pmf.} \end{cases}$$

How do we "select" the threshhold? II

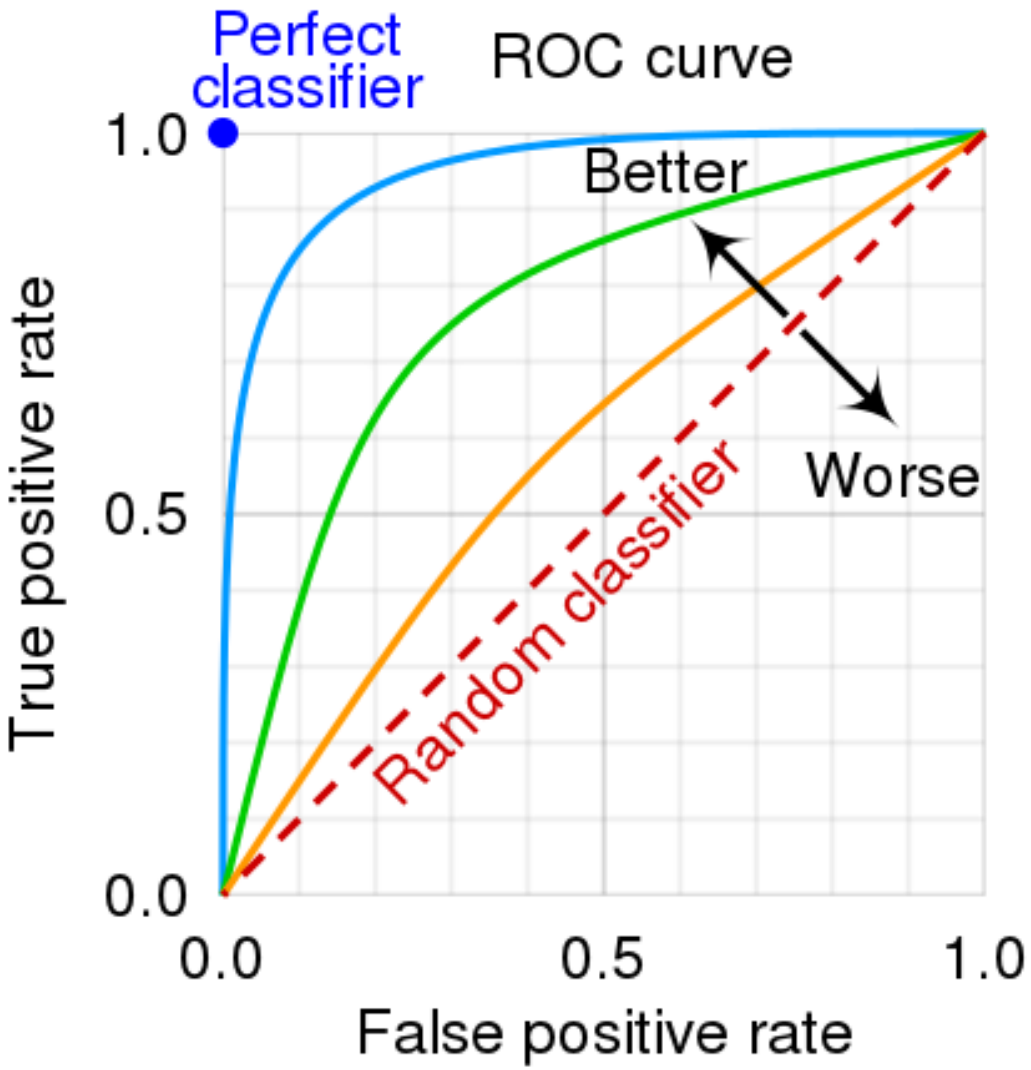


Figure: Source: cmglee, MartinThoma, Roc curve, CC BY-SA 4.0

How do we "select" the threshhold? III

- While each line in the plot represents a possible model, each point on said line represents a specific chosen thresshold value.

$\Rightarrow$ Choosing a threshhold value using the ROC means balancing between a desirable TPF and FPF.

- The ROC also gives rise to a popular measure of model performance (far beyond the setting of logistic regression) **area under the curve (AUC)**, which calculates the integral beneath the *Receiver Operating characteristic (ROC)*.

- Given that it calculates integrals under functions defined on $[0, 1] \times [0, 1]$, the AUC always lies between 0 and 1, with an AUC of 1 indicating an optimal model.

Multinomial logistic regression I

- While the term *logistic regression* is used to only refer to GLMs with Binomial distributional assumption and dichotomous target variable in many settings, in the context of Classification, *multinomial logistic regression*, i.e. a GLM with logit link-function and multinomial distributional assumption is often relevant!

- Recall that, while the Binomial distribution models $n \in \mathbb{N}$ independent trials of an experiment with two possible outcomes, the multinomial distribution is a generalization to n independent trials with $K \in \mathbb{N}$ mutually exclusive outcomes.

- Equivalently, multinomial logistic regression extends the logistic regression with dichotomous target variable to a setting where the target variable has K classes/categories.

Multinomial logistic regression II

- To do this, we first select a single class to serve as the baseline (or reference category). W.l.o.g., we can select the K th class for this role. Then we assume, for Y_k, $k \in \{1, \ldots, K\}$, denoting the kth category/class and η_k the corresponding linear predictior (i.e. we have K different sets of β-coefficients)

$$\mathbb{E}(Y_k \mid X) = p_k(X) = \frac{e^{\eta_k(X)}}{1 + \sum_{l=1}^{K-1} e^{\eta_l(X)}}$$

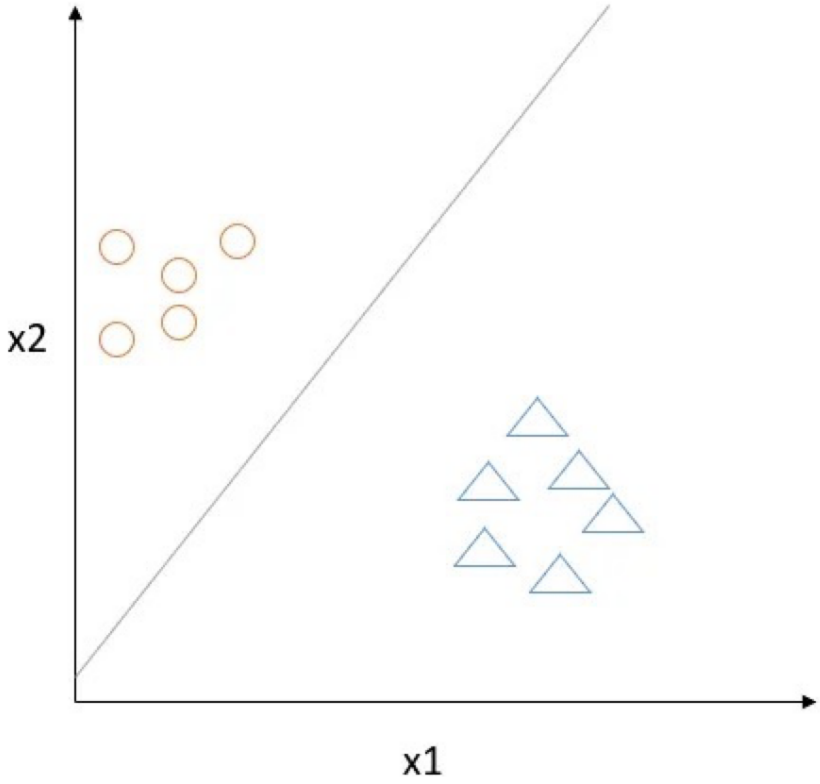
for $k = 1, \ldots, K-1$, and

$$\mathbb{E}(Y_K \mid X) = p_K(X) = \frac{1}{1 + \sum_{l=1}^{K-1} e^{\eta_l(X)}}$$

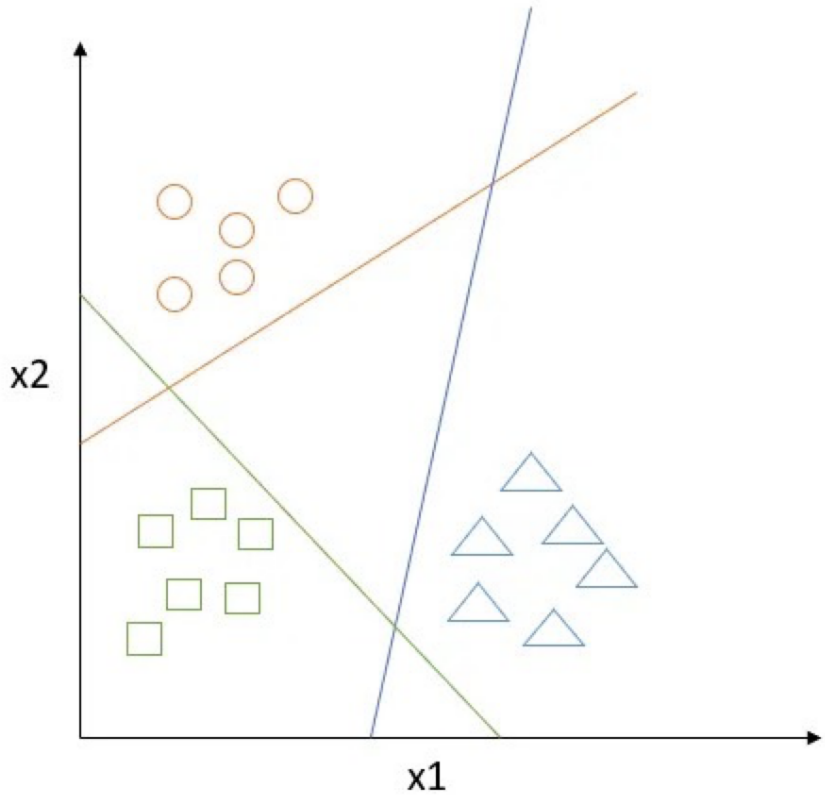
It is not hard to show that, for $j = 1, \ldots, K-1$,

$$\log\left(\frac{p_k(X)}{p_K(X)}\right) = \eta_k(X).$$

Visualization of Logistic regressions with two regressors



Binomial logistic regression



Multinomial logistic regression

Source: https://medium.com/data-sensitive/multiple-or-multinomial-classification-using-logistic-regression-explained-using-mnist-dataset-19d2dfc10c94

Naive Bayes (NB)

- Naive Bayes refers to a class of generative classification models.

- The "*naive*" element of this approach is the underlying assumption that each of the regressors/features/inputs **are conditionally independent of each other given $\boldsymbol{Y}$**.

Definition (Conditional independence)

A set of random variables $X_1, \ldots, X_m$, $m \in \mathbb{N}$, with possible values in $\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_m$ is called *conditionally independent given the random variable* Y with possible values in $\boldsymbol{\mathcal{Y}}$, if

$$F_{X_1,\ldots,X_n|Y=y}(x_1, \ldots, x_n) = \prod_{i=1}^{m} F_{X_i|Y=y}(x_i) \quad \forall\, x_i \in \mathcal{X}_i,\ y \in \mathcal{Y}\,.$$

Sampling and mixture distribution under Naive Bayes

Denoting, for $k \in \{1, \ldots, K\}$, the a priori-probability of the kth, category as π_k and the sampling distribution conditional on the the kth category as $f_k(\boldsymbol{x})$, we get

- The probability density/mass function of the sampling distribution

$$f_k(\boldsymbol{x}) = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p),$$

where f_{kj} is the probability density function of the j th predictor among observations in the kth category/class.

and

- The mixture distribution

$$\mathrm{P}(Y = k \mid X = \boldsymbol{x}) = \frac{\pi_k \times f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)}{\sum_{l=1}^{K} \pi_l \times f_{l1}(x_1) \times f_{l2}(x_2) \times \cdots \times f_{lp}(x_p)}.$$

How is f_{kj} estimated?

There are several different options to estimate the one-dimensional probability density/mass functions f_{kj}, $k = 1, \ldots, K$; $j = 1, \ldots, p$. Some of the most common options are:

- For metric regressors/inputs: Using kernel density estimation (KDE) or making an assumption about the distributional family.
- For categorical regressors/inputs: Using the relative frequency observed in the training data.

Discriminant Analysis

- Unfortunately, *Discriminant Analysis* is yet another term that is widely used without having a universally agreed upon definition.
- The good news are that there are many "special cases" of discriminant analysis which are well defined and established.
- In this course, we will focus on the following:
  1. **Linear discriminant analysis (LDA)** - sometimes used synonymously to Discriminant analysis
  2. **Quadratic discriminant analysis (QDA)**

Idea behind LDA and QDA

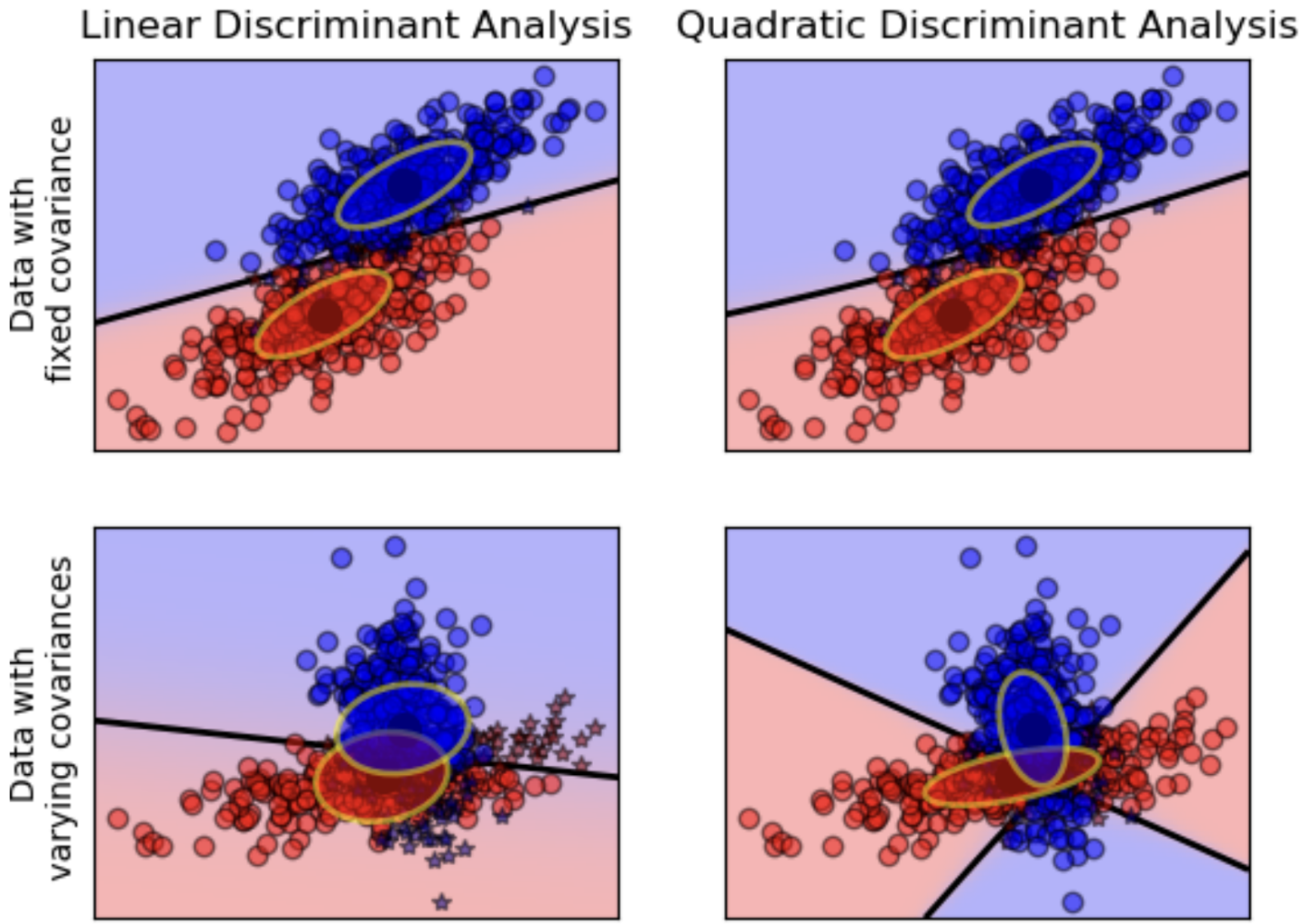
- Both Linear and Quadratic discriminant analysis are generative classification models.

- Instead of assuming conditional independence (as in NB), the underlying assumption in both approaches is that of a Gaussian sampling distribution, specifically, for $k = 1, \ldots, K$, $\mu_k \in \mathbb{R}^p$ and $\Sigma_k \in \mathbb{R}^{p \times p}$:

$$f(\boldsymbol{x}|y = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \mu_k)^t \Sigma_k^{-1}(\boldsymbol{x} - \mu_k)\right)$$

- While both LDA and QDA allow for varying means μ_k, the difference is that in LDA, the covariance is assumed to be constant, i.e. $\Sigma_1 = \Sigma_2 = \ldots = \Sigma_K$.

Visual comparison of LDA and QDA



Source:

https://deeplearning.buzz/2018/10/02/linear-discriminant-analysis-and-quadratic-discriminant-analysis/

Specific LDA setting I

- Linear discriminant analysis (LDA) is the special case when we assume that the classes have a common covariance matrix $\mathbf{\Sigma}_k = \mathbf{\Sigma} \in \mathbb{R}^{p \times p}$ $\forall k \in \{k = \ldots, K\}$. The log-ratio, is sufficient to compare any two classes k and l, $l, k \in \{1, \ldots, K\}$, $l \neq k$:

$$\log \frac{\Pr(G = k \mid X = x)}{\Pr(G = \ell \mid X = x)} = \log \frac{f_k(x)}{f_\ell(x)} + \log \frac{\pi_k}{\pi_\ell}$$

$$= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2} \left(\mu_k + \mu_\ell\right)^T \mathbf{\Sigma}^{-1} \left(\mu_k - \mu_\ell\right) + x^T \mathbf{\Sigma}^{-1} \left(\mu_k - \mu_\ell\right),$$

  which is linear in x.

- The linear log-odds function implies that the decision boundary between any pair of classes is linear so all the decision boundaries are linear.

Specific LDA setting II

- Furthermore, maximizing the a posteriori probability $P(Y = k|X = \boldsymbol{x})$, $k \in \{1, \ldots, K\}$, is equivalent to maximizing the function

$$d_k(\boldsymbol{x}) = \log(f(\boldsymbol{x}|k)) + \log(p(k)) \,,$$

  which is often referred to as **discriminant function**.

- In the case of LDA, the discriminant functions are given by

$$d_k(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k \,,$$

  - which is again linear in $\boldsymbol{x}$.

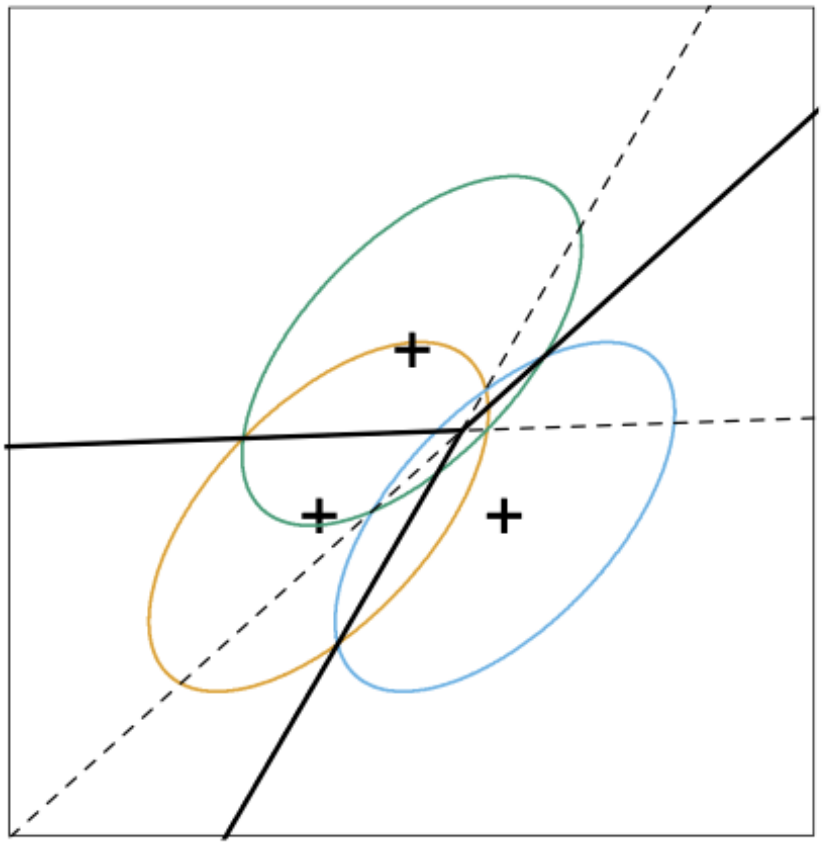
- The Bayes decision rule is then given by $\delta(\boldsymbol{x}) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \, d_k(\boldsymbol{x})$.

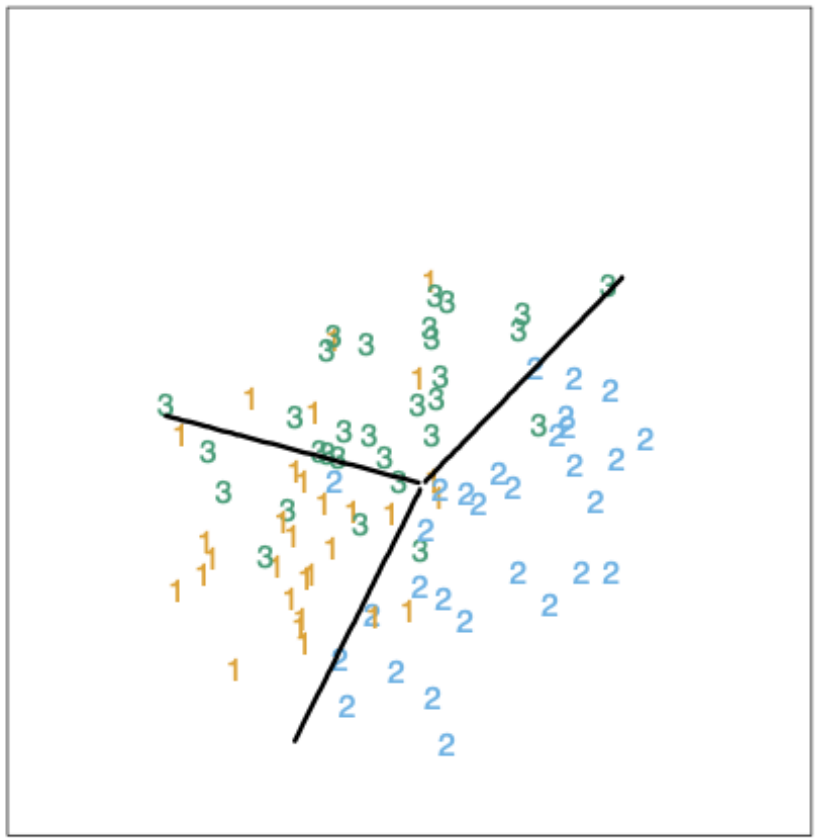
Specific LDA setting III

- Since the parameters of the Gaussian distributions are not known in practice, they are estimated as follows using the training data of size n:
  - $\hat{\pi}_k = N_k/n$, where N_k is the number of class-k observations
  - $\hat{\mu}_k = \sum_{\{x_i : y_i = k\}} x_i/N_k$ - i.e. the in-class arithmetic mean
  - $\hat{\mathbf{\Sigma}} = \sum_{k=1}^{K} \sum_{\{x_i : y_i = k\}} \left(x_i - \hat{\mu}_k\right)\left(x_i - \hat{\mu}_k\right)^T /(n-K)\,.$

Specific LDA setting IV

Visualization of LDA for three classes:





Source: Figure 4.5 of the book https://hastie.su.domains/Papers/ESLII.pdf.

Specific QDA setting

- The setting remains the same in quadratic discriminant analysis, but contrary to LDA, the covariance is assumed to be varying just as the mean.

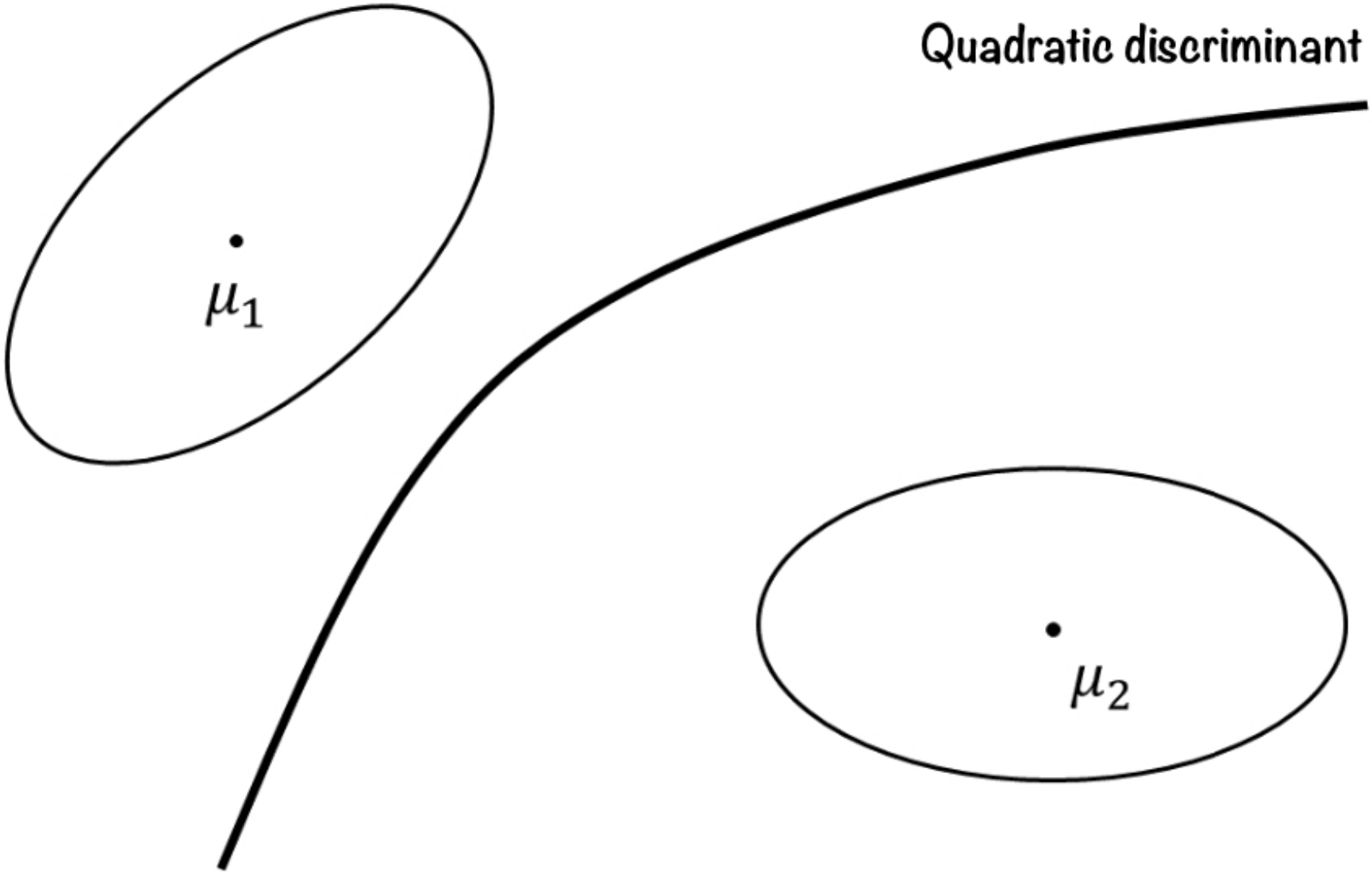
- While the estimation of $\hat{\pi}_k$ and $\hat{\mu}_k$ remains the same, the covariance matrices $\boldsymbol{\Sigma}_k$ are now estimated as the *in-class sample covariance matrices*.

- The resulting **quadratic** discriminant functions are given by

$$\begin{aligned} d_k(\boldsymbol{x}) &= \log(f(\boldsymbol{x}|k)) + \log(p(k)) \\ &= -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}-\mu_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\mu_k) + \log \pi_k \,. \end{aligned}$$

- The Bayes decision rule is again given by $\delta(\boldsymbol{x}) = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\, d_k(\boldsymbol{x})$.

Conceptual sketch of QDA for two classes



Connection between log.reg., NB, LDA, and QDA I

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

Setting

As before, Y has K classes and an observation is assigned to the class that maximizes $\mathrm{P}(Y = k \mid X = x)$. Equivalently, we can set K as the baseline class and assign an observation to the class that maximizes

$$\log\left(\frac{\mathrm{P}(Y = k \mid X = x)}{\mathrm{P}(Y = K \mid X = x)}\right)$$

for $k = 1, \ldots, K$.

Connection between log.reg., NB, LDA, and QDA II

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

1. Logistic regression: Recall that in (multinomial) logistic regression, the following holds:

$$\log\left(\frac{\mathrm{P}(Y=k \mid X=x)}{\mathrm{P}(Y=K \mid X=x)}\right) = \log\left(\frac{p_k(X)}{p_K(x)}\right) = \eta(x) = \beta_{0k} + \sum_{i=1}^{p} \beta_{ik} x_i \,.$$

---

2. LDA: Using Bayes' Theorem, the assumption that the predictors within each class are drawn from a multivariate normal density with class-specific mean and shared covariance matrix gives

Connection between log.reg., NB, LDA, and QDA III

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

$$
\begin{aligned}
\log\left(\frac{\mathrm{P}(Y=k \mid X=x)}{\mathrm{P}(Y=K \mid X=x)}\right) &= \log\left(\frac{\pi_k f_k(x)}{\pi_K f_K(x)}\right) \\
&= \log\left(\frac{\pi_k \exp\left(-\frac{1}{2}\left(x-\mu_k\right)^T \mathbf{\Sigma}^{-1}\left(x-\mu_k\right)\right)}{\pi_K \exp\left(-\frac{1}{2}\left(x-\mu_K\right)^T \mathbf{\Sigma}^{-1}\left(x-\mu_K\right)\right)}\right) \\
&= \log\left(\frac{\pi_k}{\pi_K}\right) - \frac{1}{2}\left(x-\mu_k\right)^T \mathbf{\Sigma}^{-1}\left(x-\mu_k\right) + \frac{1}{2}\left(x-\mu_K\right)^T \mathbf{\Sigma}^{-1}\left(x-\mu_K\right) \\
&= \log\left(\frac{\pi_k}{\pi_K}\right) - \frac{1}{2}\left(\mu_k+\mu_K\right)^T \mathbf{\Sigma}^{-1}\left(\mu_k-\mu_K\right) + x^T \mathbf{\Sigma}^{-1}\left(\mu_k-\mu_K\right) \\
&= a_k + \sum_{j=1}^{p} b_{kj} x_j,
\end{aligned}
$$

where $a_k = \log\left(\frac{\pi_k}{\pi_K}\right) - \frac{1}{2}\left(\mu_k+\mu_K\right)^T \mathbf{\Sigma}^{-1}\left(\mu_k-\mu_K\right)$ and b_{kj} is the j th component of $\mathbf{\Sigma}^{-1}\left(\mu_k-\mu_K\right)$.

Connection between log.reg., NB, LDA, and QDA IV

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

Hence LDA, like logistic regression, assumes that the log odds of the posterior probabilities is linear in x.

---

3. QDA: Using similar calculations, we get

$$\log\left(\frac{\mathrm{P}(Y=k \mid X=x)}{\mathrm{P}(Y=K \mid X=x)}\right) = a_k + \sum_{j=1}^{p} b_{kj}x_j + \sum_{j=1}^{p}\sum_{l=1}^{p} c_{kjl}x_j x_l,$$

where a_k, b_{kj}, and c_{kjl} are functions of $\pi_k, \pi_K, \mu_k, \mu_K, \mathbf{\Sigma}_k$ and $\mathbf{\Sigma}_K$. Again, as the name suggests, QDA assumes that the log odds of the posterior probabilities is quadratic in x.

Connection between log.reg., NB, LDA, and QDA V

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

4. Naive Bayes: Recall that in this setting, $f_k(x)$ is modeled as a product of p one-dimensional functions $f_{kj}(x_j)$ for $j = 1, \ldots, p$. Hence,

$$\begin{aligned}
\log\left(\frac{\mathrm{P}(Y=k \mid X=x)}{\mathrm{P}(Y=K \mid X=x)}\right) &= \log\left(\frac{\pi_k f_k(x)}{\pi_K f_K(x)}\right) = \log\left(\frac{\pi_k \prod_{j=1}^{p} f_{kj}(x_j)}{\pi_K \prod_{j=1}^{p} f_{Kj}(x_j)}\right) \\
&= \log\left(\frac{\pi_k}{\pi_K}\right) + \sum_{j=1}^{p} \log\left(\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}\right) \\
&= a_k + \sum_{j=1}^{p} g_{kj}(x_j),
\end{aligned}$$

where $a_k = \log\left(\frac{\pi_k}{\pi_K}\right)$ and $g_{kj}(x_j) = \log\left(\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}\right)$. Hence, in this setting we actually get the form of a generalized additive model!

Connection between log.reg., NB, LDA, and QDA VI

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

From these results, we get the following connections:

- The linear form of LDA is identical to the model equation of multinomial logistic regression; in both cases, $\log\left(\frac{\mathrm{P}(Y=k|X=x)}{\mathrm{P}(Y=K|X=x)}\right)$ is a linear function of the predictors. In LDA, the coefficients in this linear function are functions of estimates for π_k, π_K, μ_k, μ_K, and $\mathbf{\Sigma}$ obtained by assuming that $X_1, \ldots, X_p$ follow a normal distribution within each class. By contrast, in logistic regression, the coefficients are chosen to maximize the likelihood function.

  $\Rightarrow$ We expect LDA to outperform logistic regression when the normality assumption (approximately) holds, and we expect logistic regression to perform better when it does not.

Connection between log.reg., NB, LDA, and QDA VII

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

- LDA is a special case of QDA with $c_{kjl} = 0$ for all $j = 1, \ldots, p$, $l = 1, \ldots, p$, and $k = 1, \ldots, K$. (Of course, this is not surprising, since LDA is simply a restricted version of QDA with $\mathbf{\Sigma}_1 = \cdots = \mathbf{\Sigma}_K = \mathbf{\Sigma}$.)

- Any classifier with a linear decision boundary is a special case of naive Bayes with $g_{kj}(x_j) = b_{kj}x_j$. In particular, this means that LDA is a special case of naive Bayes!

- If we model $f_{kj}(x_j)$ in the naive Bayes classifier using a one-dimensional Gaussian distribution $N(\mu_{kj}, \sigma_j^2)$, then we end up with $g_{kj}(x_j) = b_{kj}x_j$ where $b_{kj} = (\mu_{kj} - \mu_{Kj})/\sigma_j^2$. In this case, naive Bayes is actually a special case of LDA with $\mathbf{\Sigma}$ restricted to be a diagonal matrix with j th diagonal element equal to σ_j^2.

Connection between log.reg., NB, LDA, and QDA VIII

This section is based on pages 158-160 from James, Gareth, et al. *An Introduction to Statistical Learning: With Applications in R*. Germany, Springer New York, 2013.

- Neither QDA nor naive Bayes is a special case of the other. Naive Bayes can produce a more flexible fit, since any choice can be made for $g_{kj}(x_j)$. However, it is restricted to a purely additive fit. By contrast, QDA includes multiplicative terms of the form $c_{kjl}x_jx_l$. Therefore, QDA has the potential to be more accurate in settings where interactions among the predictors are important in discriminating between classes.

  $\Rightarrow$ None of these methods uniformly dominates the others: in any setting, the choice of method will depend on the true distribution of the predictors in each of the K classes, as well as other considerations, such as the values of n and p. The latter ties into the bias-variance trade-off.

Outlook: Decision Trees and SVMs

TBD!