Supervised Learning

20_KIN2 – Artificial Intelligence and Machine Learning

Overview

1. Modeling Theory
2. Definitions
3. Formalization
4. Regression
5. Classification
6. Model Selection

Data-based Modeling

Prerequisites

1. A relation between the input and output features exists and it can be modelled.
2. The relation is unknown or cannot be easily described with mathematical equations.
3. Data is present.

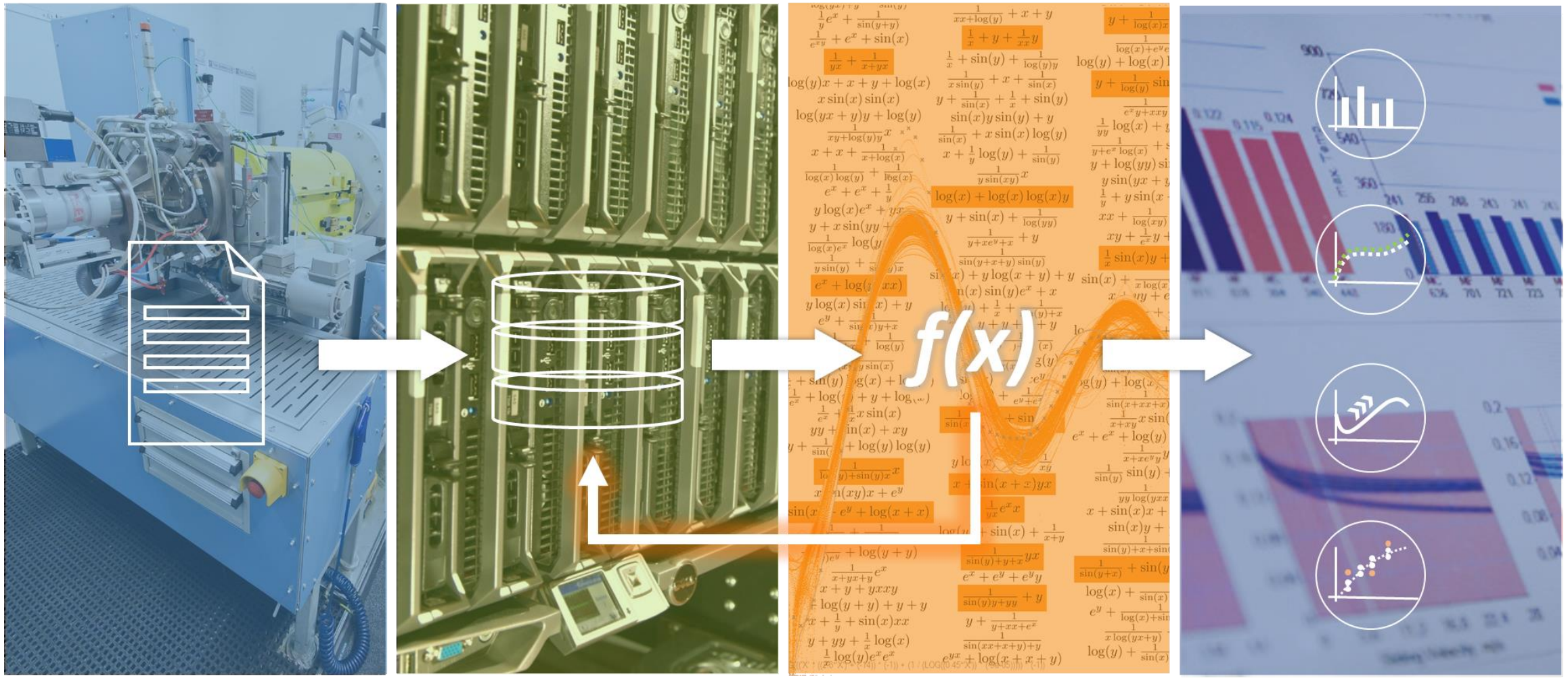
Consequences

No relation exists → Modeling will fail or the results are insufficient.

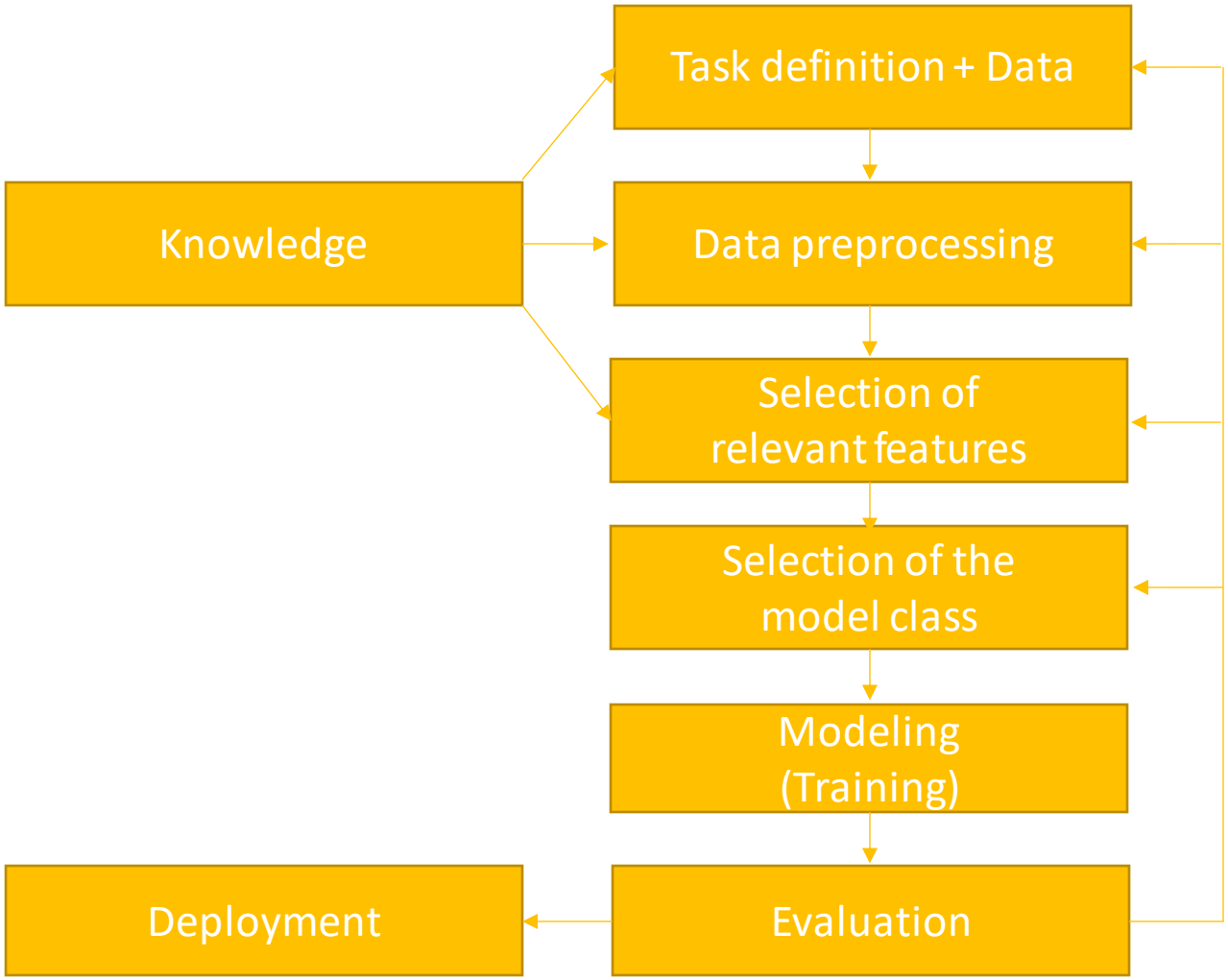
Relation is known → Data-based modeling is not the suitable approach.

No data is present → Data-based modeling is not possible.

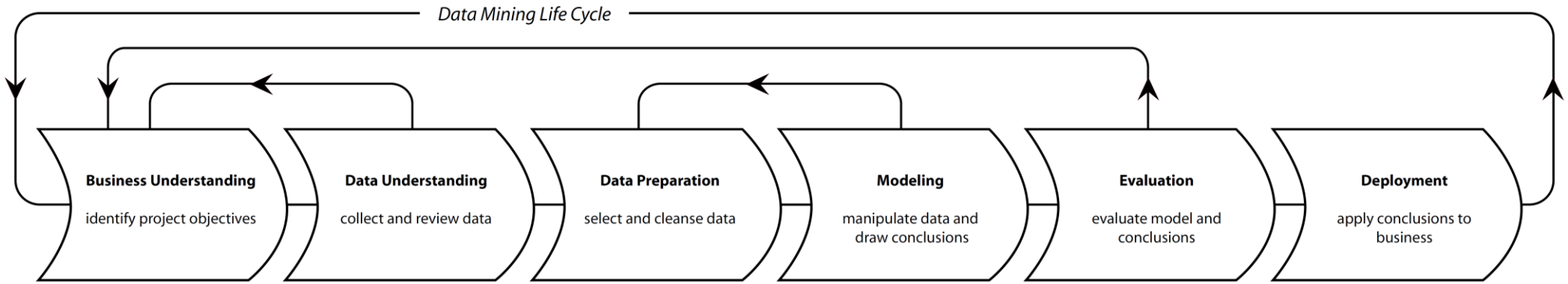
Sources | Storage | Modeling / Analysis | Benefits



Workflow


Task definition + Data
Knowledge
Data preprocessing
Selection of relevant features
Selection of the model class
Modeling (Training)
Deployment
Evaluation

CRISP-DM / Cross-Industry Standard Process for Data Mining



Determine Business Objectives
Background
Business Objectives
Business Success Criteria
(Log and Report Process)

Assess Situation
Inventory of Resources, Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
(Log and Report Process)

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria
(Log and Report Process)

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques
(Log and Report Process)

Collect Initial Data
Initial Data Collection Report
(Log and Report Process)

Describe Data
Data Description Report
(Log and Report Process)

Explore Data
Data Exploration Report
(Log and Report Process)

Verify Data Quality
Data Quality Report
(Log and Report Process)

Data Set
Data Set Description
(Log and Report Process)

Select Data
Rationale for Inclusion/ Exclusion
(Log and Report Process)

Clean Data
Data Cleaning Report
(Log and Report Process)

Construct Data
Derived Attributes
Generated Records
(Log and Report Process)

Integrate Data
Merged Data
(Log and Report Process)

Format Data
Reformatted Data
(Log and Report Process)

Select Modeling Technique
Modeling Technique
Modeling Assumptions
(Log and Report Process)

Generate Test Design
Test Design
(Log and Report Process)

Build Model Parameter Settings
Models
Model Description
(Log and Report Process)

Assess Model
Model Assessment
Revised Parameter
(Log and Report Process)

Evaluate Results
Align Assessment of Data Mining Results with Business Success Criteria
(Log and Report Process)

Approved Models
Review Process
Review of Process
(Log and Report Process)

Determine Next Steps
List of Possible Actions
Decision
(Log and Report Process)

Plan Deployment
Deployment Plan
(Log and Report Process)

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan
(Log and Report Process)

Produce Final Report
Final Report
Final Presentation
(Log and Report Process)

Review Project
Experience Documentation
(Log and Report Process)

Additional Topics

Data Preparation & Data Cleaning

- Remove redundant features
- Impute missing values
- Remove outliers
- Scale / Normalize features
- Feature Engineering

Feature Selection

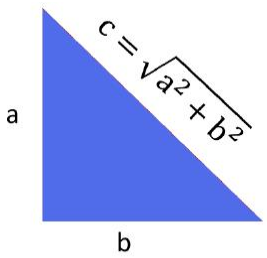
- Correlation-based
- Forward Selection
- Backward Selection

Feature Importance

- Permutation Feature Importance

Definition

"The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience."
(Tom Mitchell, 1997)



$$c = \sqrt{a^2 + b^2}$$

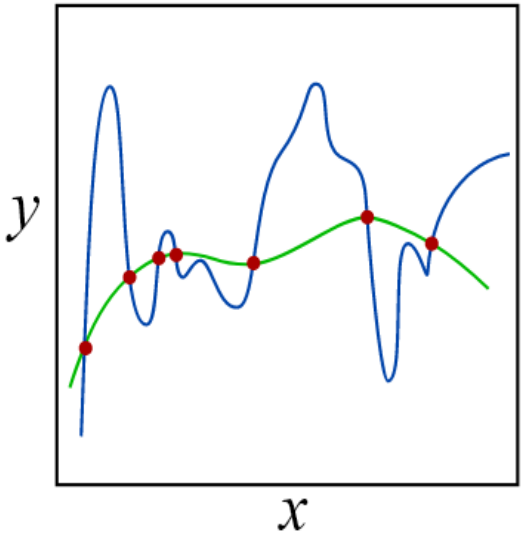
Definition

Given: A set of examples (x_i, y_i)

$S = \{(x_1, y_1), (x_2, y_2), \dots, (xn, yn)\} \subseteq x \times y$

Transduction: Output y for input x_0 ??

Induction: Complete functionally relation $f : X \rightarrow Y$??



Definition

Learning is used when a pattern exists, we can't capture it mathematically, but we have data on it. (using a set of observations to uncover an underlying process)

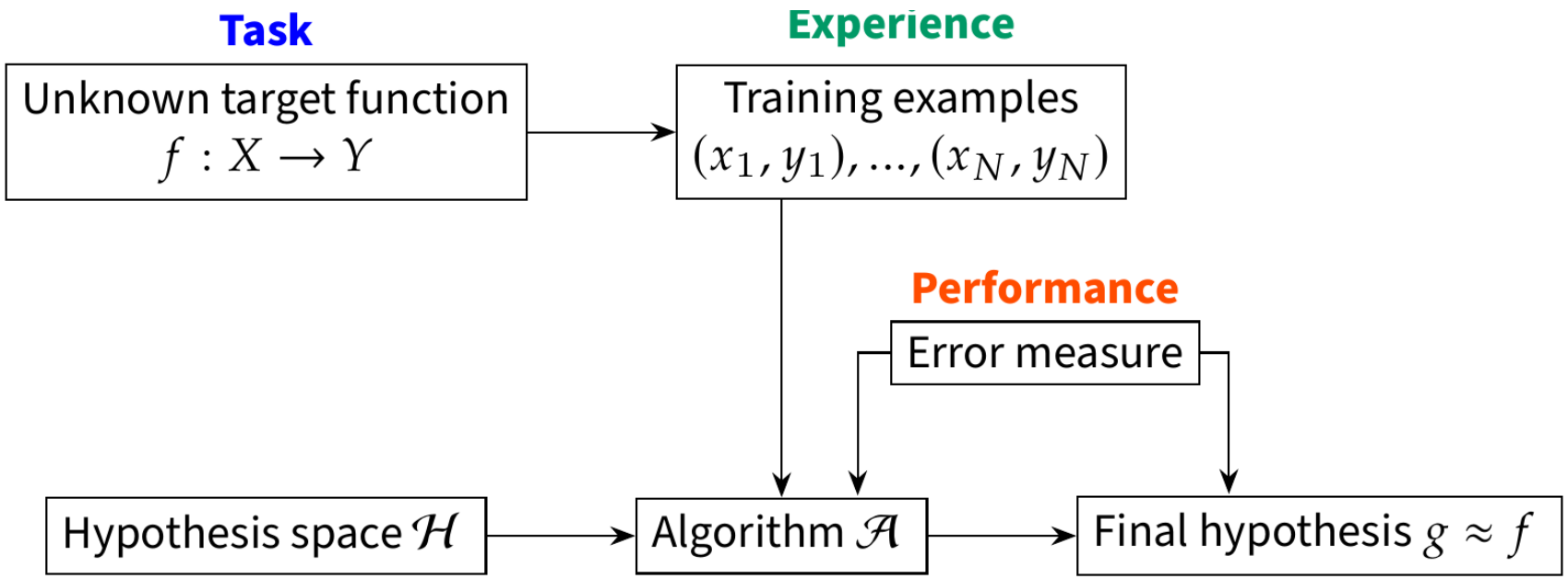
Machine Learning

Study of algorithms that improve their **performance P** at some **task T** with **experience E**. (Tom M. Mitchell, Machine Learning)

Example Regression

- **Task T**: Find mapping between target and predictors
- **Experience E**: Finite set of example mappings x → y
- **Performance P**: Minimize loss $\mathcal{L}(y, \hat{y})$

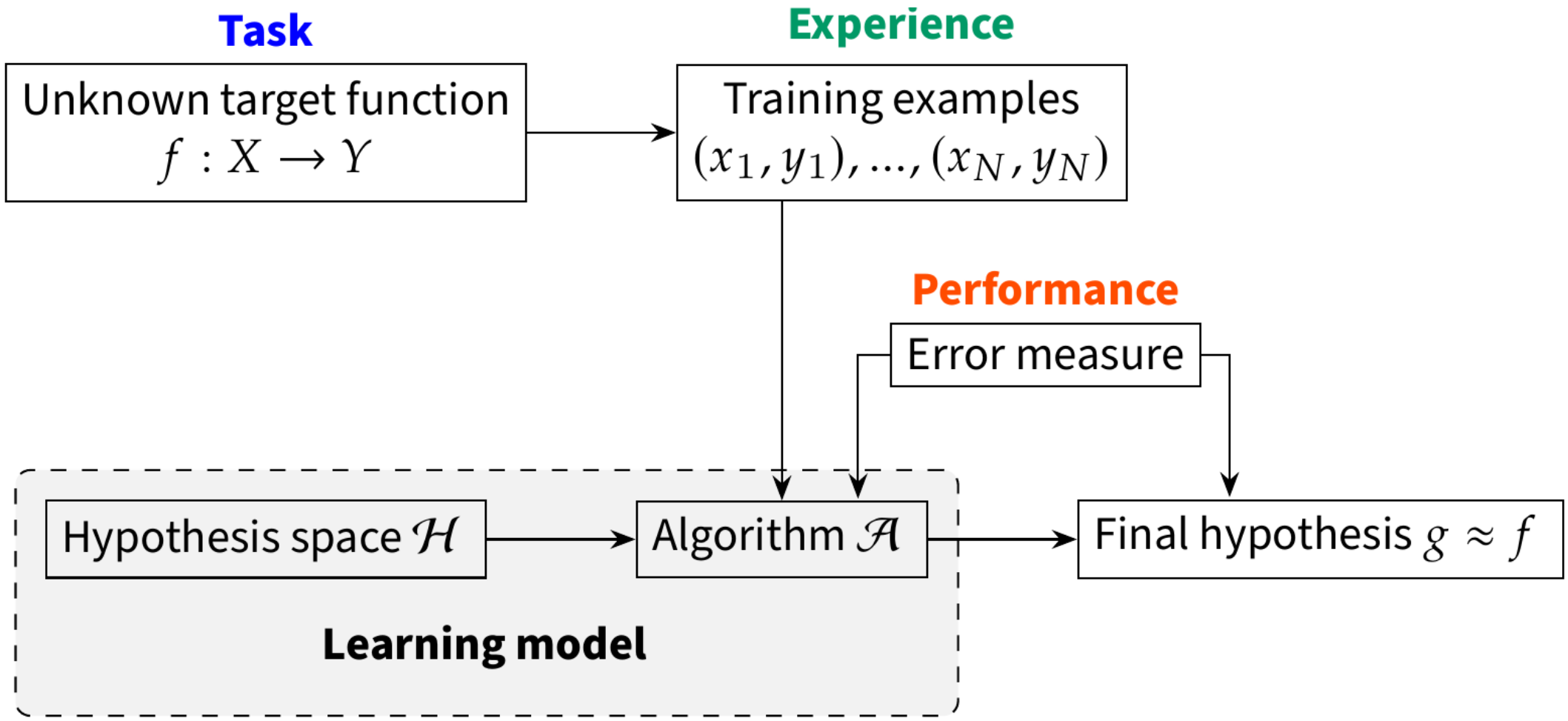
Definition



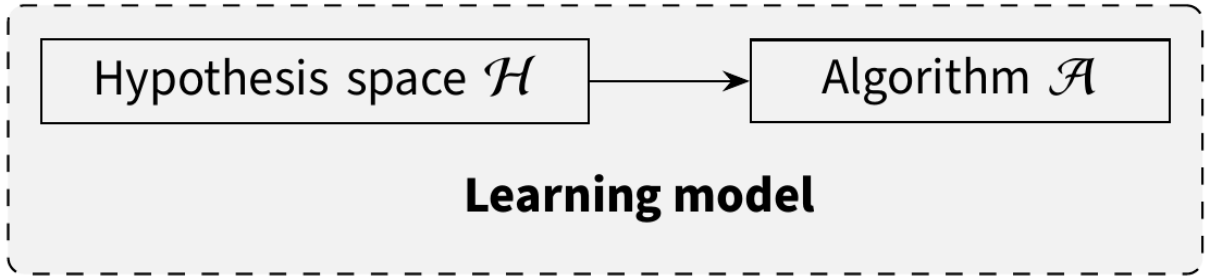
Main challenge

Ensure that the hypothesis remains valid when new data becomes available.

Learning Model



Learning Model



Example

Learning Model	Hypothesis Space	Algorithm
Polynomial regression	All polynomials up to a degree	Least squares fitting
Neural network classification	All possible neural networks	Gradient descent

Bias-variance tradeoff

- Less complex hypothesis space -> better chance of generalization
- More complex hypothesis space -> better chance of approximating *f*

Learning Model

Generative Learning Model

What is the likelihood that this class generated this data point?

- Typically employed to estimate probabilities and likelihood
- Learn a probability distribution for the dataset
- Often rely on Bayes Theorem

Discriminative Learning Model

What side of the decision boundary is this data point found in?

- The goal is to identify the decision boundary between classes
- Classes are separated based on conditional probability
- No assumption about individual data points

Taxonomy

	Classif/regr	Gen/Discr	Param/Non
Discriminant analysis	Classif	Gen	Param
Naive Bayes classifier	Classif	Gen	Param
Tree-augmented Naive Bayes classifier	Classif	Gen	Param
Linear regression	Regr	Discrim	Param
Logistic regression	Classif	Discrim	Param
Sparse linear/ logistic regression	Both	Discrim	Param
Mixture of experts	Both	Discrim	Param
Multilayer perceptron (MLP)/ Neural network	Both	Discrim	Param
Conditional random field (CRF)	Classif	Discrim	Param
K nearest neighbor classifier	Classif	Gen	Non
(Infinite) Mixture Discriminant analysis	Classif	Gen	Non
Classification and regression trees (CART)	Both	Discrim	Non
Boosted model	Both	Discrim	Non
Sparse kernelized lin/logreg (SKLR)	Both	Discrim	Non
Relevance vector machine (RVM)	Both	Discrim	Non
Support vector machine (SVM)	Both	Discrim	Non
Gaussian processes (GP)	Both	Discrim	Non
Smoothing splines	Regr	Discrim	Non

Formalization

Optimization

$$x^* = \arg\min_{x \in \Theta} L(x)$$

Θ	Hypothesis / Solution space
x	Solution candidate
x^*	Optimal solution
$L(x)$	Loss function (minimization)
	Objective / Fitness function (maximization)

Formalization

Example Regression

$$f^*(x) = \arg\min_{f \in \mathrm{F}} E_{x,y}[L(y, f(x))|D]$$

F	Model class
$f(x)$	Candidate model
$f^*(x)$	Optimal model
$L(x)$	Loss function (minimization)
y	Target
x	Input features
D	Data

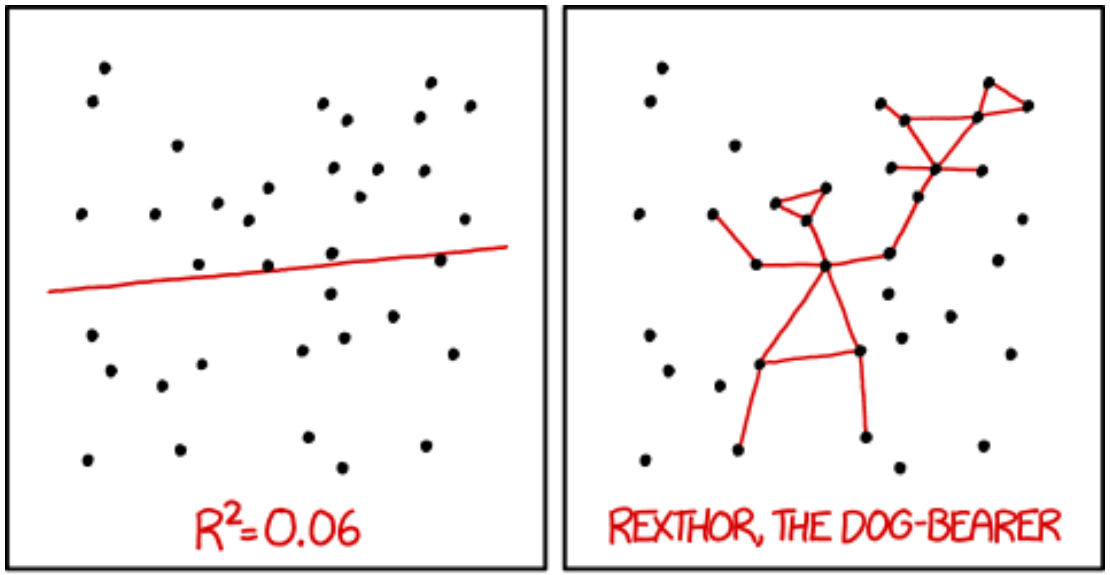
General Taxonomy

Main tasks

- Regression
- Classification

Model structure

- Linear
- Non-linear



https://xkcd.com/1725/

Regression

Regression

The goal is to build a model for the unknown relationship

$$f : X \rightarrow Y, X \in \mathbb{R}^{n \times m}, Y \in \mathbb{R}^{n}$$

An appropriate model class must be chosen by the user

(assuming a certain relationship between target and predictors)

Goal

Find optimal values for the free parameters of the model, such that error is minimized.

Linear Regression

Assumes linear dependency between target and predictors

$$\hat{y} = \sum_{i=1}^{|F|} \beta_i x_i + \beta_0$$

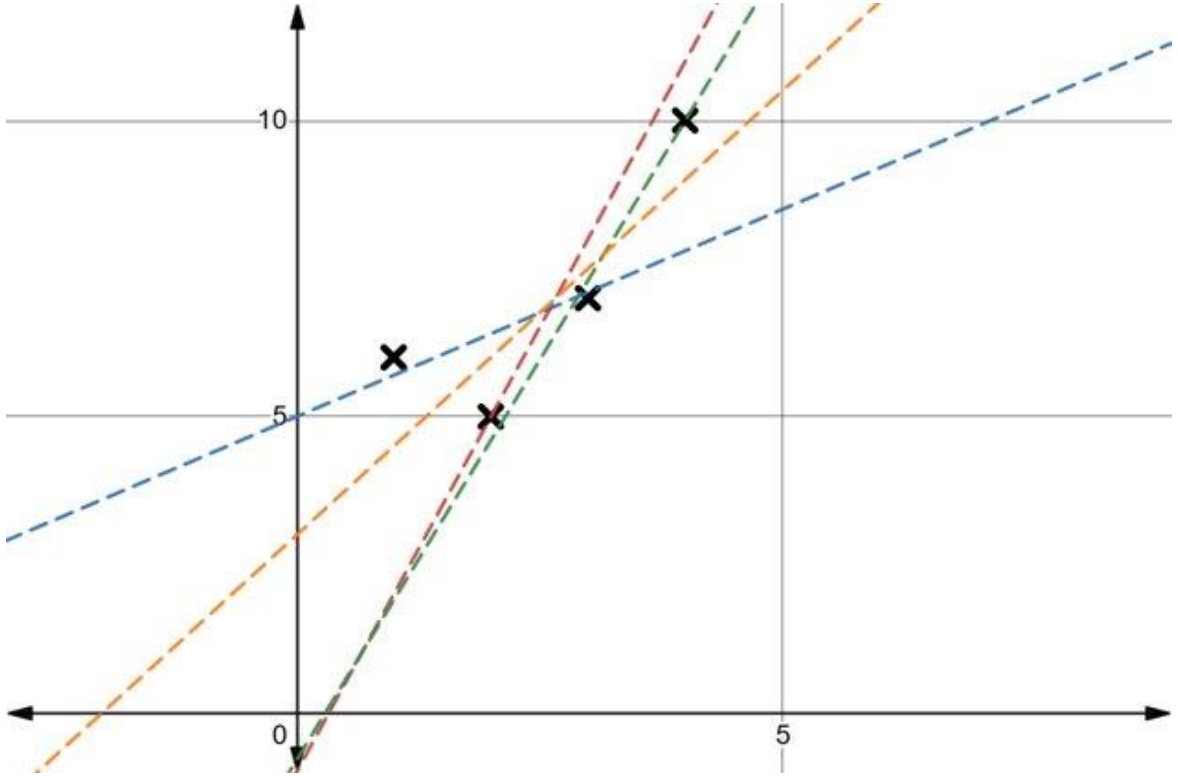
Example

$$\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_0$$

Length of parameter vector β will be |F| + 1. How do we determine β?

Linear Regression

Which linear model fits best?

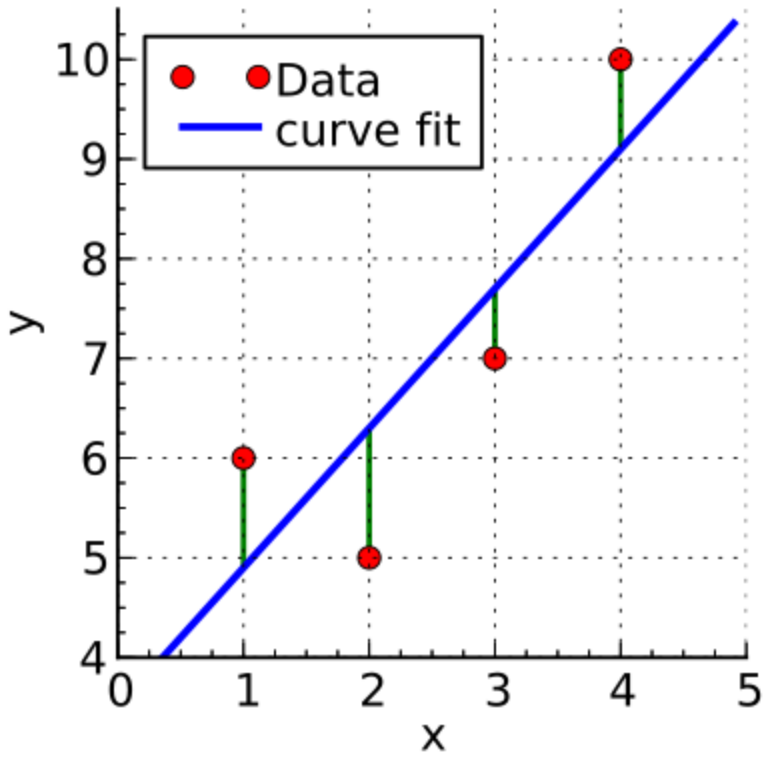


Linear Least Squares

Model class: linear models

Loss function: squared error loss

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} (y - \beta X)^2$$



Analytical Solution

One-dimensional example:

– Data $x, y \in \mathbb{R}^N$ Model class $f(x) = \beta_1 x + \beta_0$

Minimization of squared errors

$$L(\beta) = \sum_{i=1}^{N} (f(x_i) - y_i)^2 = \sum_{i=1}^{N} (\beta_1 x_i + \beta_0 - y_i)^2$$

$$\frac{\delta L(\beta)}{\delta \beta_1} = \sum_{i=1}^{N} 2x_i(\beta_1 x_i + \beta_0 - y_i) = 0$$

$$\frac{\delta L(\beta)}{\delta \beta_0} = \sum_{i=1}^{N} 2(\beta_1 x_i + \beta_0 - y_i) = 0$$

Analytical Solution - Continuation

$$\left(\sum_{i=1}^{N} x_i^2\right)\beta_1 + \left(\sum_{i=1}^{N} x_i\right)\beta_0 = \sum_{i=1}^{N} x_i y_i$$

$$\left(\sum_{i=1}^{N} x_i\right)\beta_1 + N\beta_0 = \sum_{i=1}^{N} y_i$$

Two equations for two variables

Distinct solution if N >= 2

Result: regression line

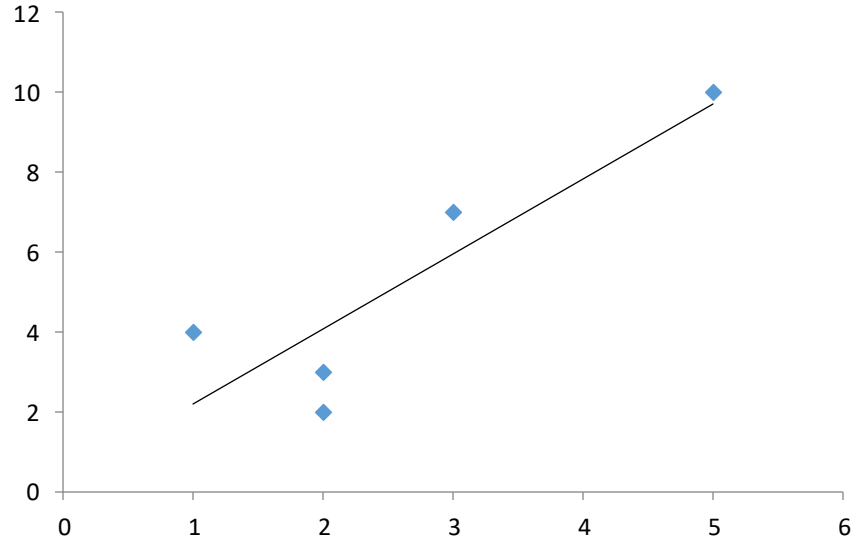
Example

$X =$

1
3
2
2
5

$y =$

4
7
2
3
10



$$N = 5, \qquad \sum x_i = 13, \qquad \sum x_i^2 = 43, \qquad \sum y_i = 26, \qquad \sum x_i y_i = 85$$

$$5\,\beta_0 + 13\beta_1 = 26, \qquad \beta_0 = \frac{26 - 13\beta_1}{5}$$

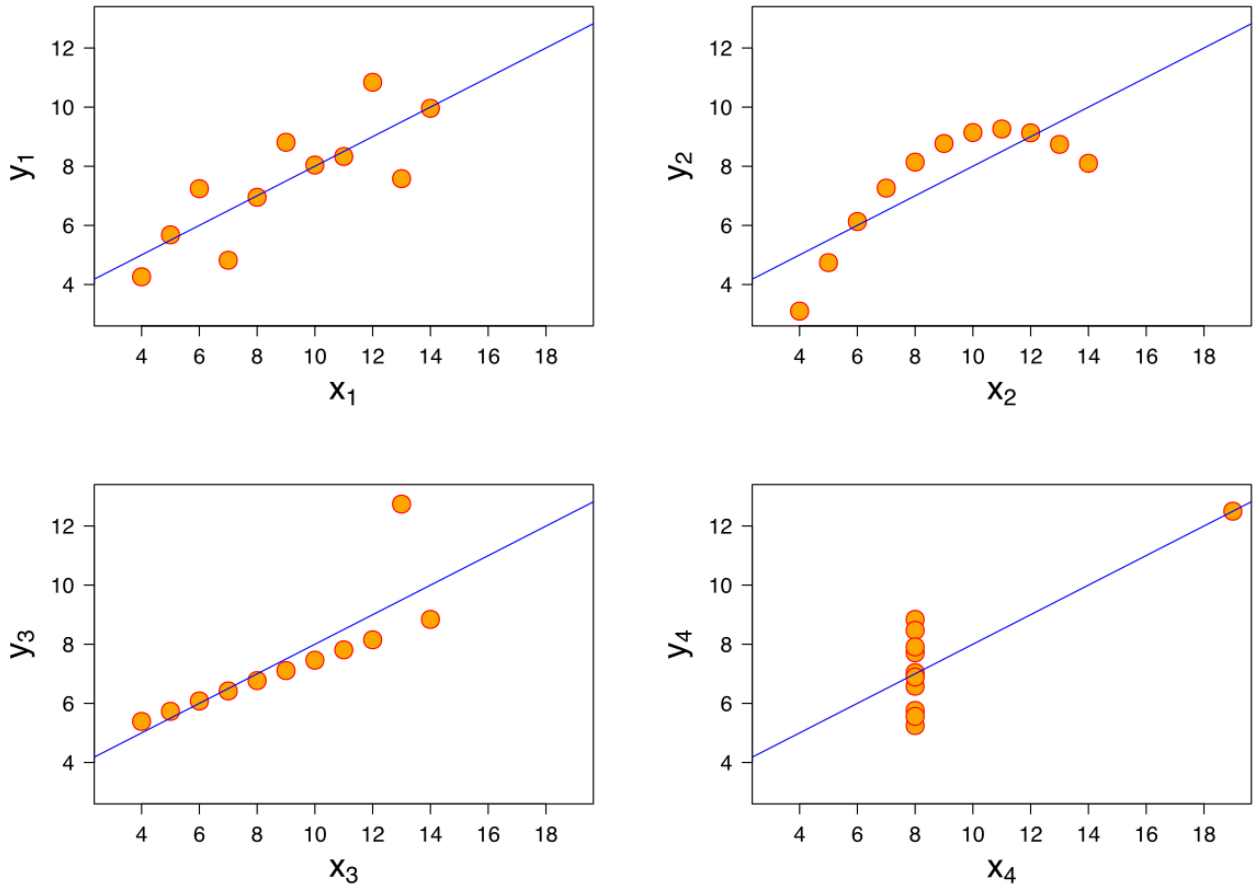
$$43\beta_1 + \frac{13}{5}(26 - 13\beta_1) = 85, \qquad \beta_1 = 1.89, \beta_0 = 0.28$$

https://en.wikipedia.org/wiki/Linear_regression#Interpretation

Pitfalls

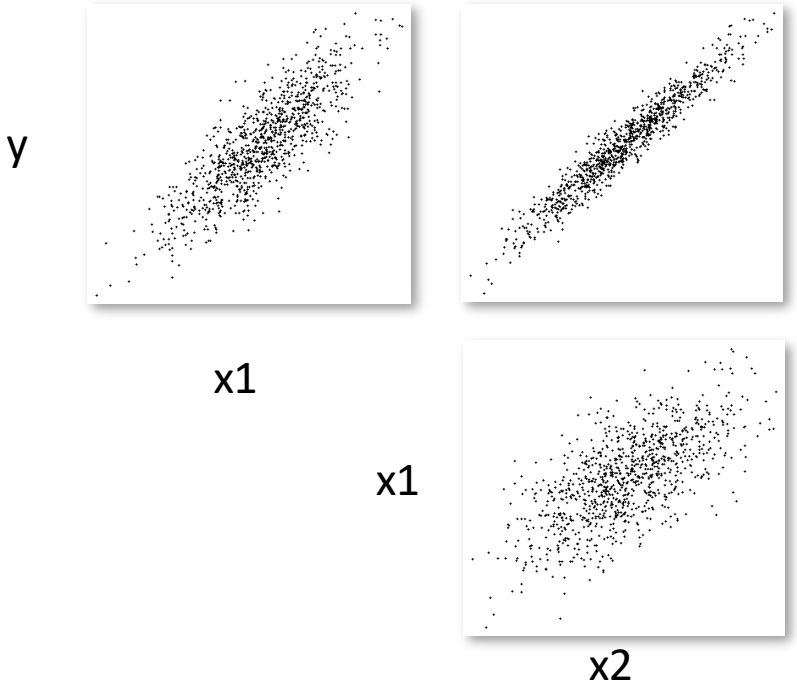
Four two-dimensional benchmark datasets yielding the same linear regression model

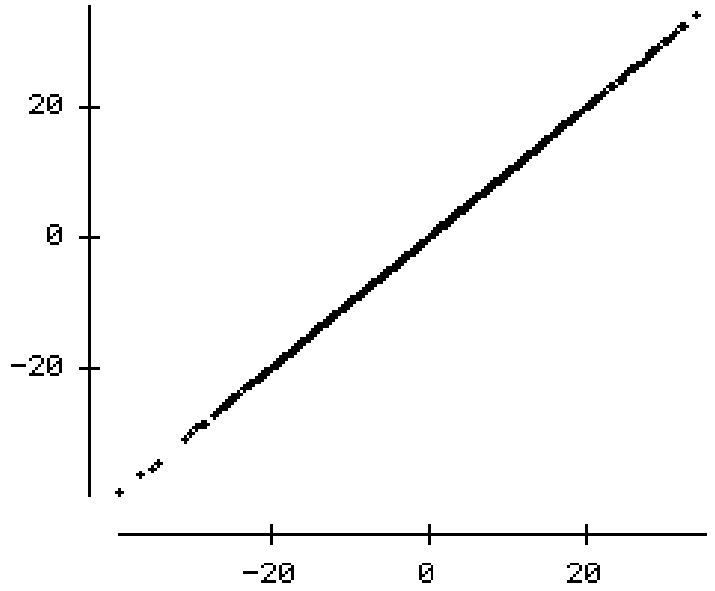
Which model could be trusted?

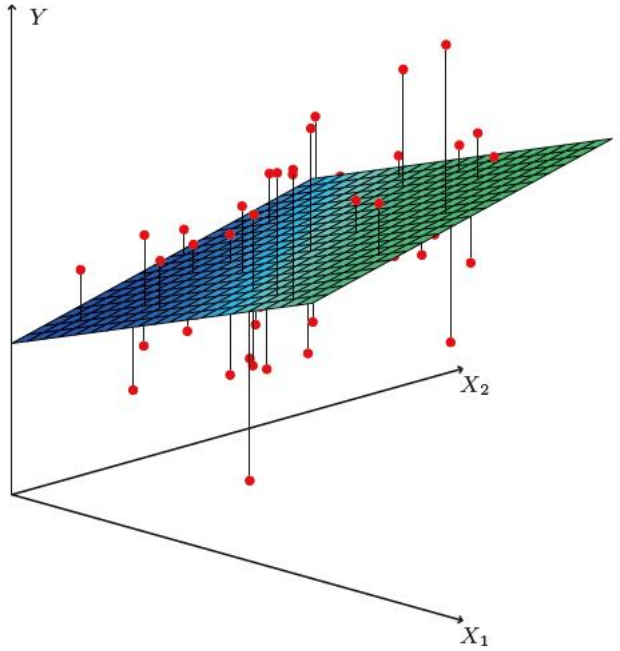


Multi-dimensional Least Squares

$$\beta^* = (X^T X)^{-1} X^T y$$







Interpretation of Linear Models

$$Result = (c_0 \cdot \mathrm{x1} + c_1 \cdot \mathrm{x2} + c_2 \cdot \mathrm{x3} + c_3 \cdot \mathrm{x4} + c_4 \cdot \mathrm{x5} + c_5 \cdot \mathrm{x6} + c_6 \cdot \mathrm{x7} + c_7 \cdot \mathrm{x8} + c_8 \cdot \mathrm{x9} + c_9 \cdot \mathrm{x10} + c_{10})$$

$$\begin{aligned}
c_0 &= 0.081337 \\
c_1 &= 0.19906 \\
c_2 &= -0.029881 \\
c_3 &= 0.078892 \\
c_4 &= -0.010307 \\
c_5 &= 0.031685 \\
c_6 &= -0.047071 \\
c_7 &= -0.029194 \\
c_8 &= 0.0015768 \\
c_9 &= 0.10525 \\
c_{10} &= 0.020099
\end{aligned}$$

Increasing x1 by 1 increases the model response by 0.081.

Increasing x3 by 1 decreases the model response by 0.029.

Linear Models

Properties

- Simple and understandable calculation
- Parameterless method
- Easy interpretation
- Efficient calculation even for large data
- Robust and stable models
- Similar results even for slightly different data
- Easy application (methodological errors are rare)

Not suitable for every application!
Modeling of nonlinear systems!

Importance of Least Squares

How can linear models be used to model nonlinear relations?

Transformation of input features

- $x_1' = \frac{1}{x_1}$
- $x_2' = \log(x_2)$
- $x_3' = x_1^2 x_3$
- $x_4' = x_4$
- $x_5{}' = x_2 x_4$

Learn model with x'

Resulting model

$$f(x) = \beta_1 \frac{1}{x_1} + \beta_2 \log(x_2) + \beta_3 x_1^2 x_3 + \beta_4 x_4 + \beta_5 x_2 x_4 + \beta_0$$

Model still must be linear in its parameters.

Transformations must be determined in advance.

Transformation of Nonlinear Models

$f(x) = \beta_1 x_1 e^{\beta_2 x_2}$ transform by log

$\log\big(f(x)\big) = \log\big(\beta_1 x_1 e^{\beta_2 x_2}\big)$

$\log\big(f(x)\big) = \log(\beta_1) + \log(x_1) + \beta_2 x_2$ substitute $\log(\beta_1)$

$\log\big(f(x)\big) - \log(x_1) = \beta_3 + \beta_2 x_2$ β_2, β_3 are linear independent

Polynomial Regression

Find a model from the following model class: $\sum_{f=1}^{|F|} \sum_{e=1}^{p} \beta_f x_f^e + \beta_0$

p: maximum degree of polynomial

Example model : $\beta_1 x_1^3 + \beta_2 x_1^2 + \beta_3 x_1^1 + \beta_4 x_2^3 + \beta_5 x_2^2 + \beta_6 x_2^1 + \beta_0$

Number of free parameters β = Number of features $|F| *$ Maximum degree of polynomial $p + 1$

How to determine the parameter vector β?

Polynomial Regression - Interactions

Terms with references to more than one variable, (i.e. $x_1x_2, x_1x_2^2, x_1x_2x_3$)

Two variables $\sum_{e=0}^{p}\sum_{f=0}^{p-e}\beta_{e,f}x_1^e x_2^f$ Three Variables $\sum_{e=0}^{p}\sum_{f=0}^{p-e}\sum_{g=0}^{p-e-f}\beta_{e,f,g}\,x_1^e x_2^f x_3^g$

Quadratic example model with three variables:

$$\beta_{0,0,0}\,x_1^0x_2^0x_3^0 +$$
$$\beta_{0,0,1}\,x_1^0x_2^0x_3^1 + \beta_{0,0,2}\,x_1^0x_2^0x_3^2 +$$
$$\beta_{0,1,0}\,x_1^0x_2^1x_3^0 + \beta_{0,1,1}\,x_1^0x_2^1x_3^1 + \beta_{0,2,0}\,x_1^0x_2^2x_3^0 +$$
$$\beta_{1,0,0}\,x_1^1x_2^0x_3^0 + \beta_{1,0,1}\,x_1^1x_2^0x_3^1 + \beta_{1,1,0}\,x_1^1x_2^1x_3^0 + \beta_{2,0,0}\,x_1^2x_2^0x_3^0$$

$$\beta_0 +$$
$$\beta_1x_3 + \beta_2x_3^2 +$$
$$\beta_3x_2 + \beta_4x_2x_3 + \beta_5x_2^2 +$$
$$\beta_6x_1 + \beta_7x_1x_3 + \beta_8x_1x_2 + \beta_9x_1^2$$

Regularization

Feature Selection

Least squares uses all available input features.

Can we reformulate the optimization so that only relevant features occur?

$$f^*(x) = \arg\min_{f \in \mathbf{F}} E\ [L(y, f(x))|D]$$

$$b^* = \arg\min_{b \in \{0,1\}^m} L(f^*(x)|D)$$

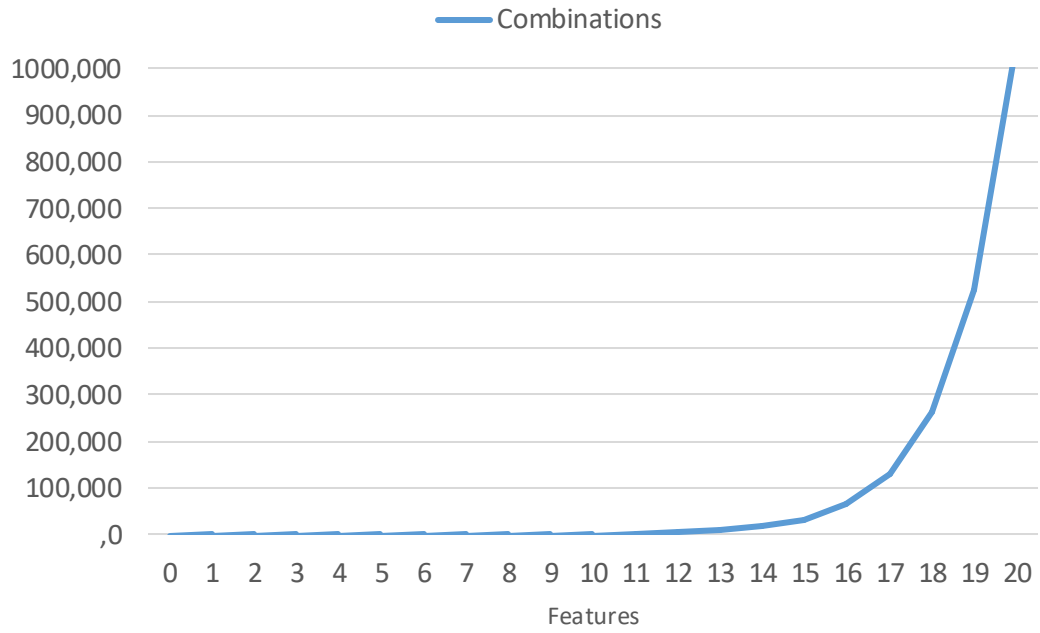
Feature Selection - Enumeration

Dataset with 20 features

- $2^{20} = 1{,}048{,}576$ combinations
- Exponential growth

Execution time

- Calculation 10 ms / model
- All combinations 2.91 hours



LASSO Regression

Least Squares

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} (y - \beta X)^2 = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2$$

Least Absolute Shrinkage and Selection Operator (LASSO)

– Reduce weights to zero

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2 \quad \text{subject to} \quad ||\beta||_1 \leq t$$

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2 + \lambda ||\beta||_1$$

Ridge Regression

Tikhonov regularization / weight decay

– Reduce all weights to small values

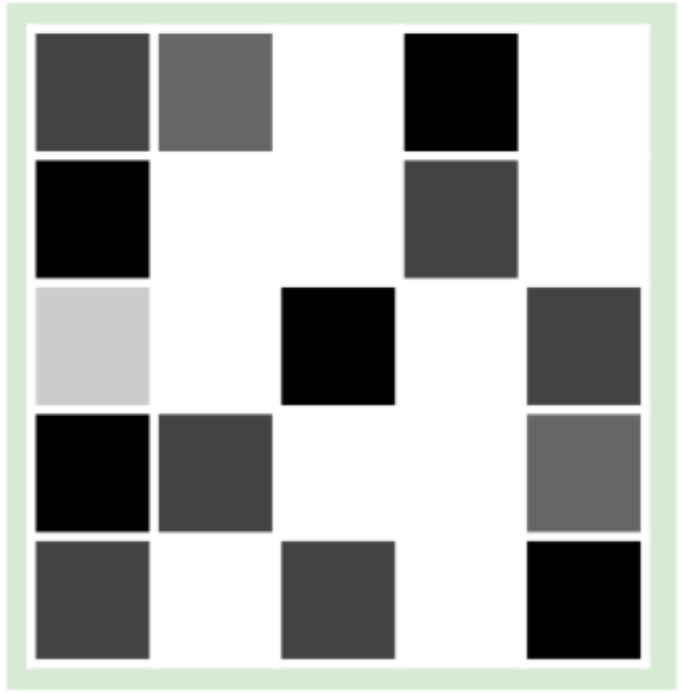
$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2 + \lambda ||\beta||_2^2$$

Features X should be scaled to zero mean and 1 variance.

Parameter λ must be tuned to the problem at hand.

Regularization

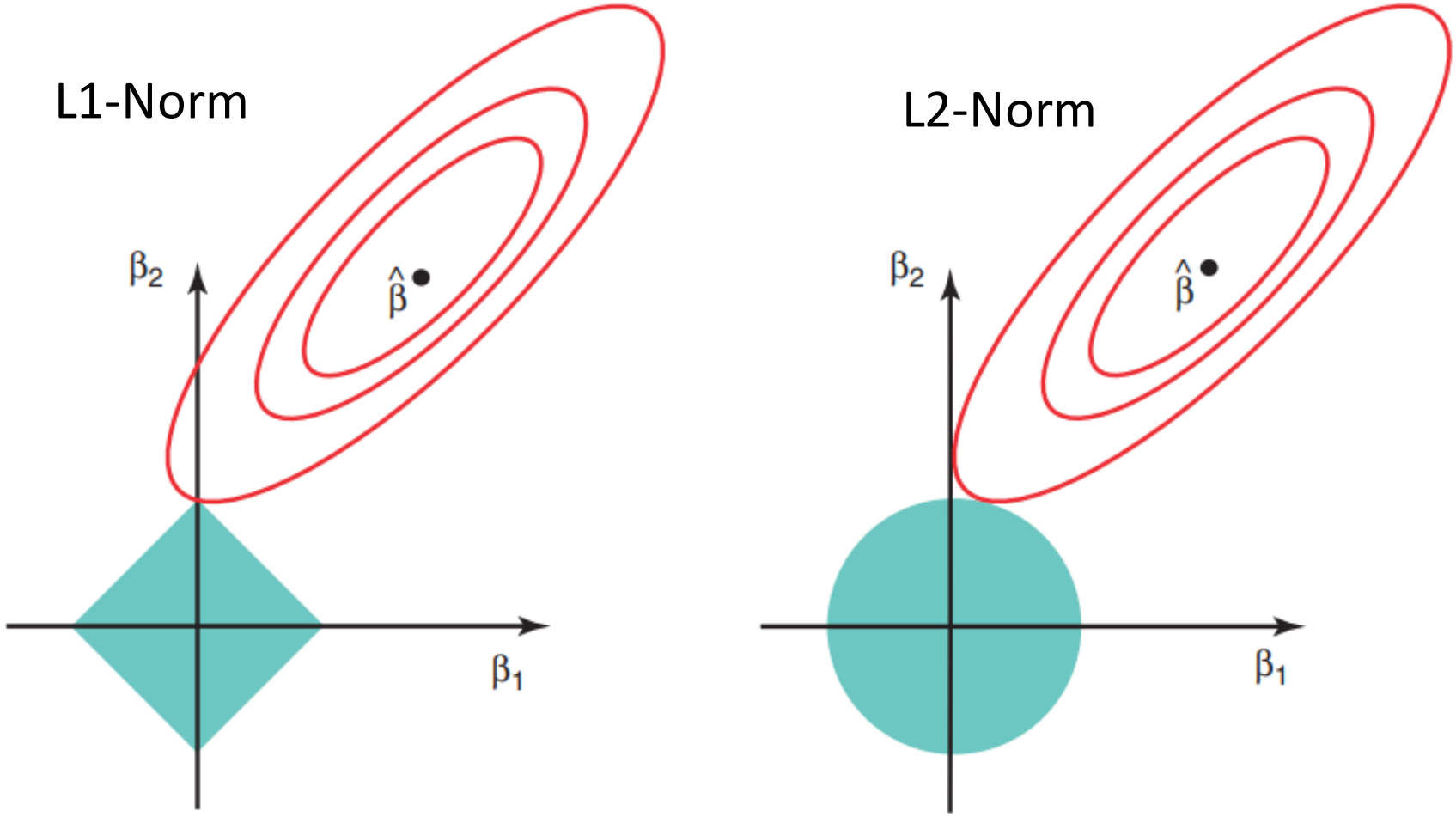
Baseline



L1 Regularization

L2 Regularization

Geometric Interpretation



Elastic Net Regularization

Combines Lasso and Ridge Regression

Tries to reduce the number of used features and minimizes the weights

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2 + \lambda_2 ||\beta||_2^2 + \lambda_1 ||\beta||_1$$

If $\lambda_2 = 0$ and $\lambda_1 = \lambda$ → Lasso Regression

If $\lambda_1 = 0$ and $\lambda_2 = \lambda$ → Ridge Regression

Elastic Net Regularization

Two free parameters λ_1, λ_2

$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} L(\lambda_1, \lambda_2, \beta)$$

Combine tradeoff between Lasso and Ridge regression into a single parameter α.

λ steers the strength of the regularization.

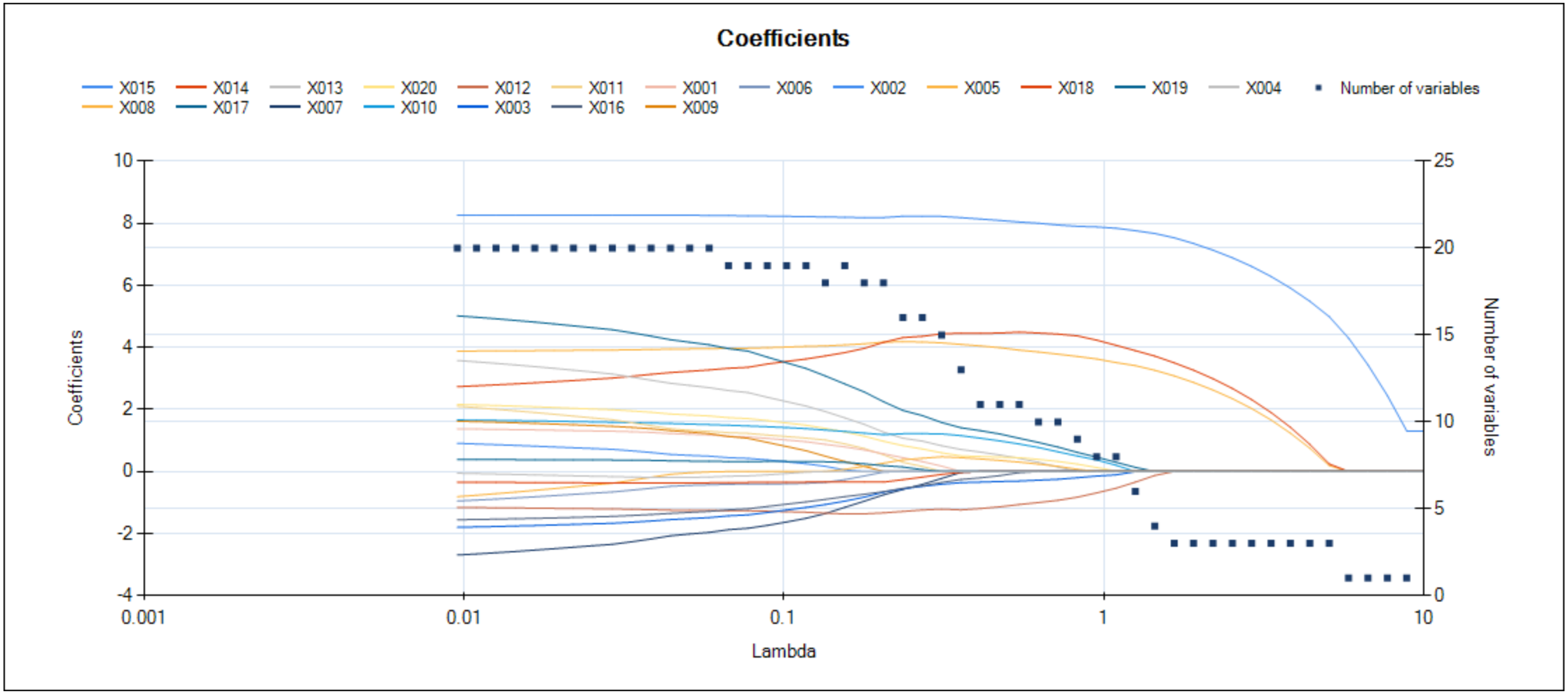
$$\beta^* = \arg\min_{\beta \in \mathbb{R}^m} ||y - \beta X||_2^2 + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1]$$

If $\alpha = 1$ → Lasso Regression

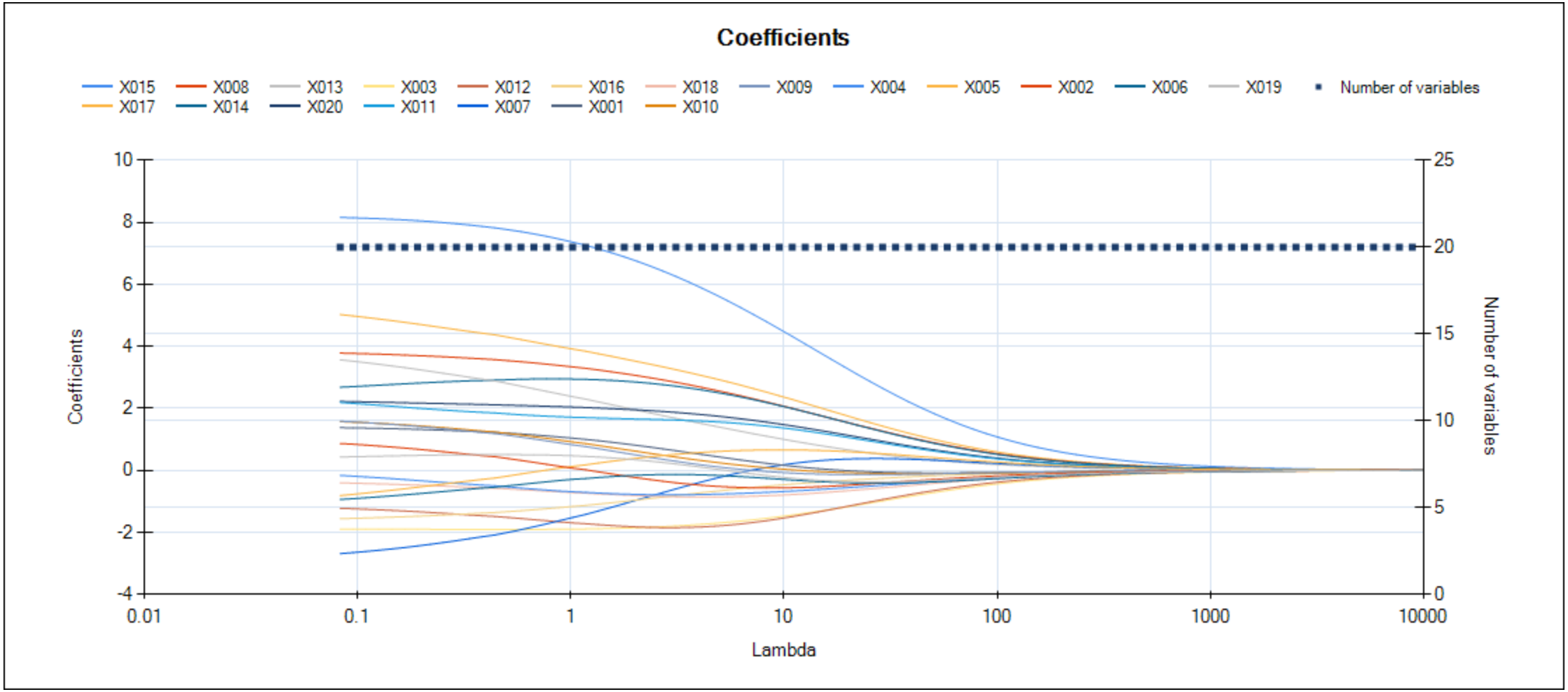
If $\alpha = 0$ → Ridge Regression

Path for different λ values can be efficiently calculated for a fixed α

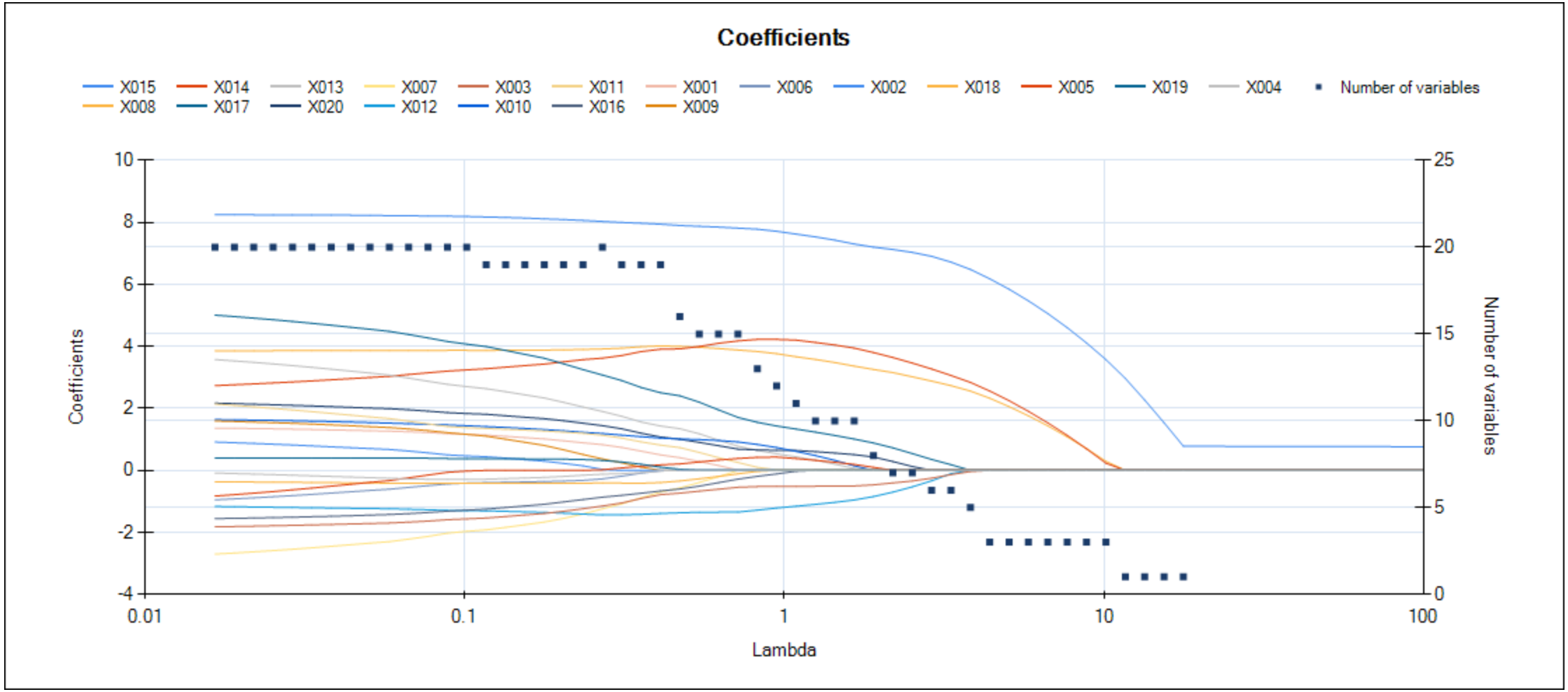
Elastic Net ($\alpha = 1$, LASSO) Regression



Elastic Net ($\alpha = 0$, Ridge) Regression

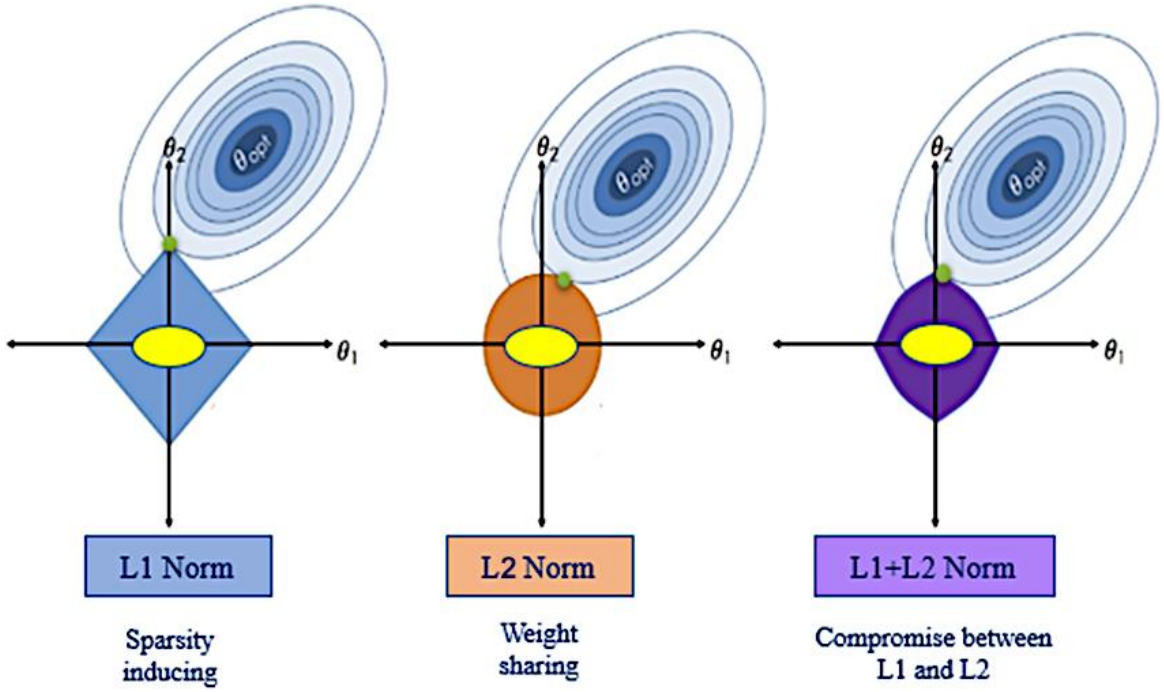


Elastic Net ($\alpha = 0.5$) Regression



ElasticNet

- L1 regularization produces sparse models, but may fail on some datasets (↓ robustness)
- L2 regularization is more robust, but it does not produce sparse models (↓ interpretability)
- ElasticNet is a linear combination of L1 and L2 regularization (balance controlled by α)



Linear Models

When do linear modeling techniques fail?

If the model parameters β are not linearly independent anymore.

Examples:

- Parameters of fractions of polynomials $\frac{\beta_1 x_1}{\beta_2 x_2 + \beta_3 x_3} + \beta_0$
- Parameters within exponentials or logarithms $\log(\beta_1 x_1 + \beta_2 x_2) + \beta_0$
- Parameters within trigonometric functions $\sin(\beta_1 x_1 + \beta_2) + \beta_0$
- ….

Nonlinear Regression – Function Fitting

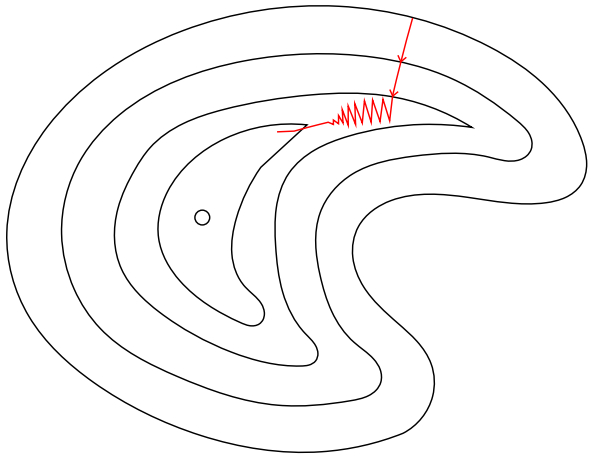
Find a model from the following model class: $\frac{\theta_1 x^2+\theta_2+\theta_3 x^3}{\theta_4 x+\theta_5 x^4}, \theta \in \mathbb{R}^5$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^5} \left(y - f(X, \theta)\right)^2$$

How to determine the parameter vector θ^*, when Linear Least Squares fails due to nonlinear parameters?

Fall back to optimization techniques:

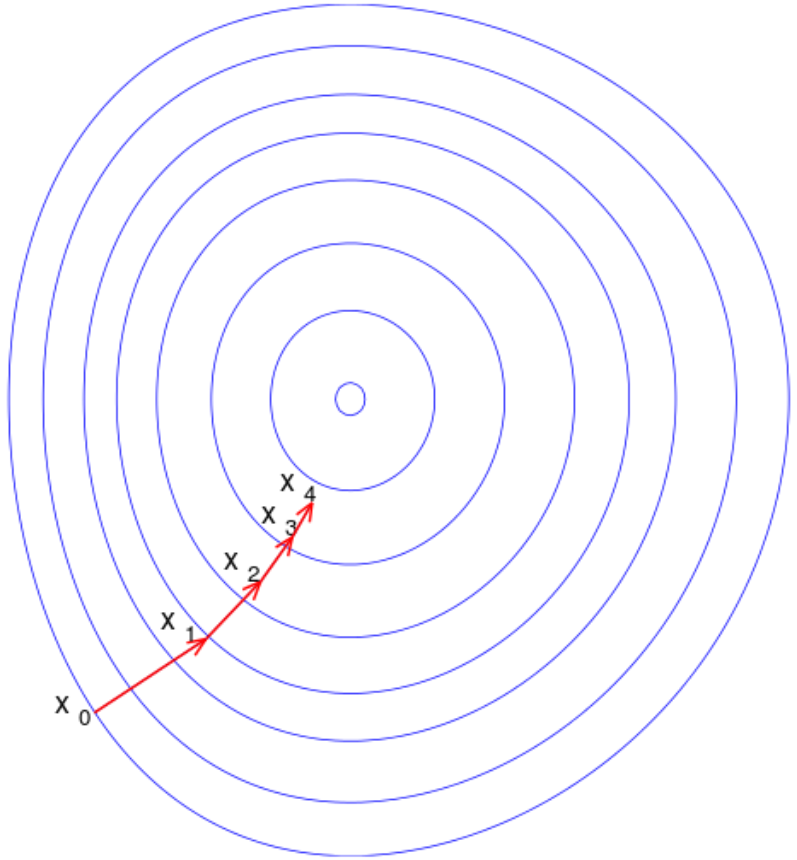
- Initialize with random starting values
- Optimize parameters with gradient descent



https://en.wikipedia.org/wiki/Gradient_descent

Gradient Descent

- Function Minimization
- Follow the gradient of the function
- $x_{n+1} = x_n + \gamma \nabla F(x_n)$
- Step size γ is adapted



Nonlinear Regression

Advantages

- Can be used for arbitrary model structures, if gradient information is available.
- Otherwise fall back to gradient free optimization methods.
- Convex (unimodal) optimization problem if parameters are linear.

Disadvantages

- Random initialization
- Converges into local optima
- Multiple repetition necessary
- Results depend heavily on the concrete model structure

Classification

Classification

Given:

- $\mathrm{X} = \left(x_{i,j}\right)_{i=1..N, j=1..m}$ matrix with N observations of m input features, $x_{i,j} \in \mathbb{R}$
- $y = (y_i)_{i=1..N}$ vector of N observations of the target
  $y_i \in \{\mathrm{class}_1, \mathrm{class}_2, \mathrm{class}_3, \ldots, \mathrm{class}_\mathrm{p}\}$
- Structure of the assumed relation (model class) between the input features X and target y

Wanted:

- Values for the free parameters of the predefined model class to estimate the target values with minimal deviations (errors).

$$x^* = \arg\min_{x \in \Theta} L(x)$$

Features / Variables / Attributes

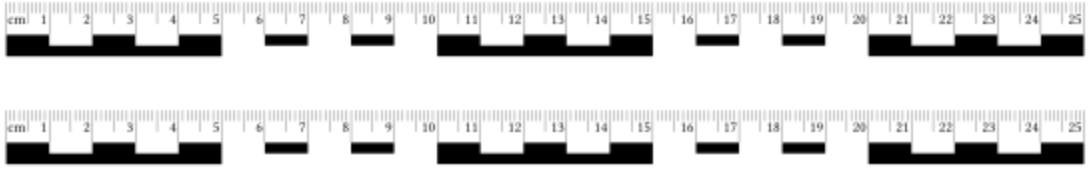
Nominal / Categorical

- Finite range, no order, equality
- Example: color ∈ {red, blue, green}

Ordinal

- Finite range, linear order
- Example: hotel category ∈ {economy, mid scale, luxury}

Numerical



Confusion Matrix

- Positive Class (P)
  Samples with positive class label
- Negative Class (N)
  Samples with negative class label
- Positive Prediction
  Samples with positive label prediction
- Negative Prediction
  Samples with negative label prediction

- True Positive (TP)
  Positive sample with positive prediction
- False Positive (FP)
  Negative sample with positive prediction
- False Negative (FN)
  Positive sample with negative prediction
- True Negative (TN)
  Negative sample with negative prediction

	Positive Class	**Negative Class**
Positive Prediction	True Positive	False Positive
Negative Prediction	False Negative	True Negative

Classification Measures

- Sensitivity / Recall (True Positive Rate, TPR)
  TPR = TP / P
- Specificity (True Negative Rate, TNR)
  TNR = TN / N
- Positive Predictive Value (Precision, PPR)
  PPV = TP / (TP + FP)
- Negative Predictive Value (NPR)
  NPR = TN / (TN + FN)
- False Positive Rate (FPR)
  FPR = FP / N
- False Negative Rate (FNR)
  FNR = FN / P

	Positive Class	**Negative Class**
Positive Prediction	True Positive	False Positive
Negative Prediction	False Negative	True Negative

- Accuracy = (TP + TN) / (P + N)
- F1-Score
  F1 = 2x (PPV x TPR) / (PPV +TPR)
  F1 = 2x TP / (2x TP + FP + FN)
- Matthew's correlation coefficient (MCC) [-1,1]

$$= \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Classification Measures

#	Feature 1	Feature 2	…	Feature m	Class label	Prediction	State
1					True	True	True positive
2					False	True	False positive
…					…	…	
S-1					True	False	False negative
S					False	False	True negative

	Positive class	Negative class
Positive prediction	**True positive**	**False positive**
Negative prediction	**False negative**	**True negative**

We have S total samples, out of which

- N samples have a negative class label
- P samples have a positive class label

True positive rate (TPR): TP / P
True negative rate (TNR): TN / N

Classification Measures

	Positive class	Negative class
Positive prediction	**True positive**	**False positive**
Negative prediction	**False negative**	**True negative**

True positive rate (TPR): TP / P (sensitivity)

True negative rate (TNR): TN / N (specificity)

	Covid (disease)	No covid (no disease)	
Positive test	**True positive**	**False positive**	**Sensitivity** (% of disease cases identified)
Negative test	**False negative**	**True negative**	**Specificity** (% of non-disease cases identified)

Classification Measures

	Covid (disease)	No covid (no disease)	Prevalence ($=\frac{\text{disease}}{\text{total}}$)
Positive test	**True positive**	**False positive**	**FPR = FP / P**
Negative test	**False negative**	**True negative**	**FNR = FN / N**
	TPR = TP / P **Sensitivity** (% of disease cases identified)	**TNR = TN / N** **Specificity** (% of non-disease cases identified)	**Accuracy** **(TP+TN)/total**

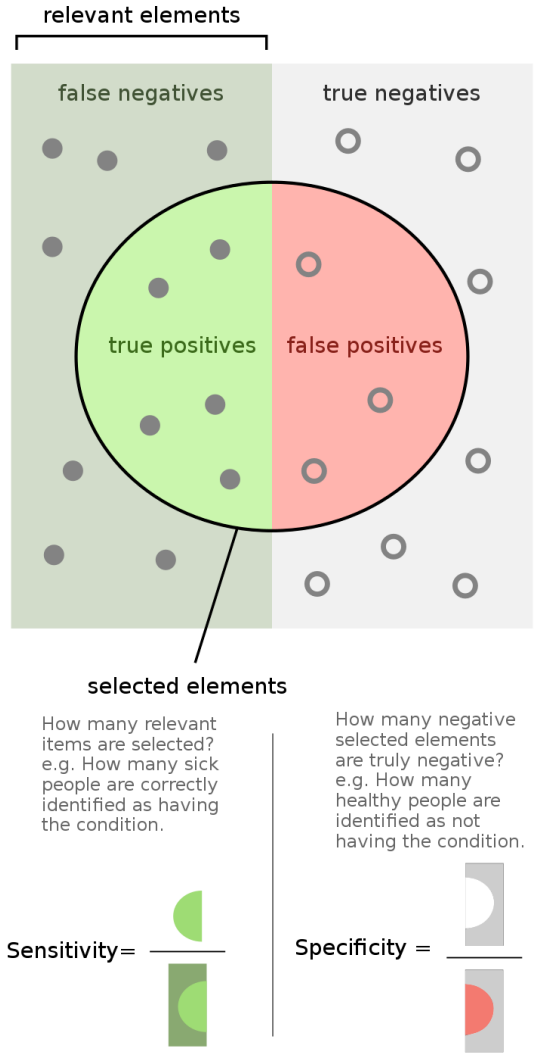
Specificity
The ability of a test to label the no-disease cases

Covid test approval: minimum 95% specificity (FDA)

Classification Measures

A test which reliably detects the presence of a condition, resulting in a high number of true positives and low number of false negatives, will have a high sensitivity.

Important when the consequence of failing to treat the condition is serious and/or the treatment is very effective and has minimal side effects



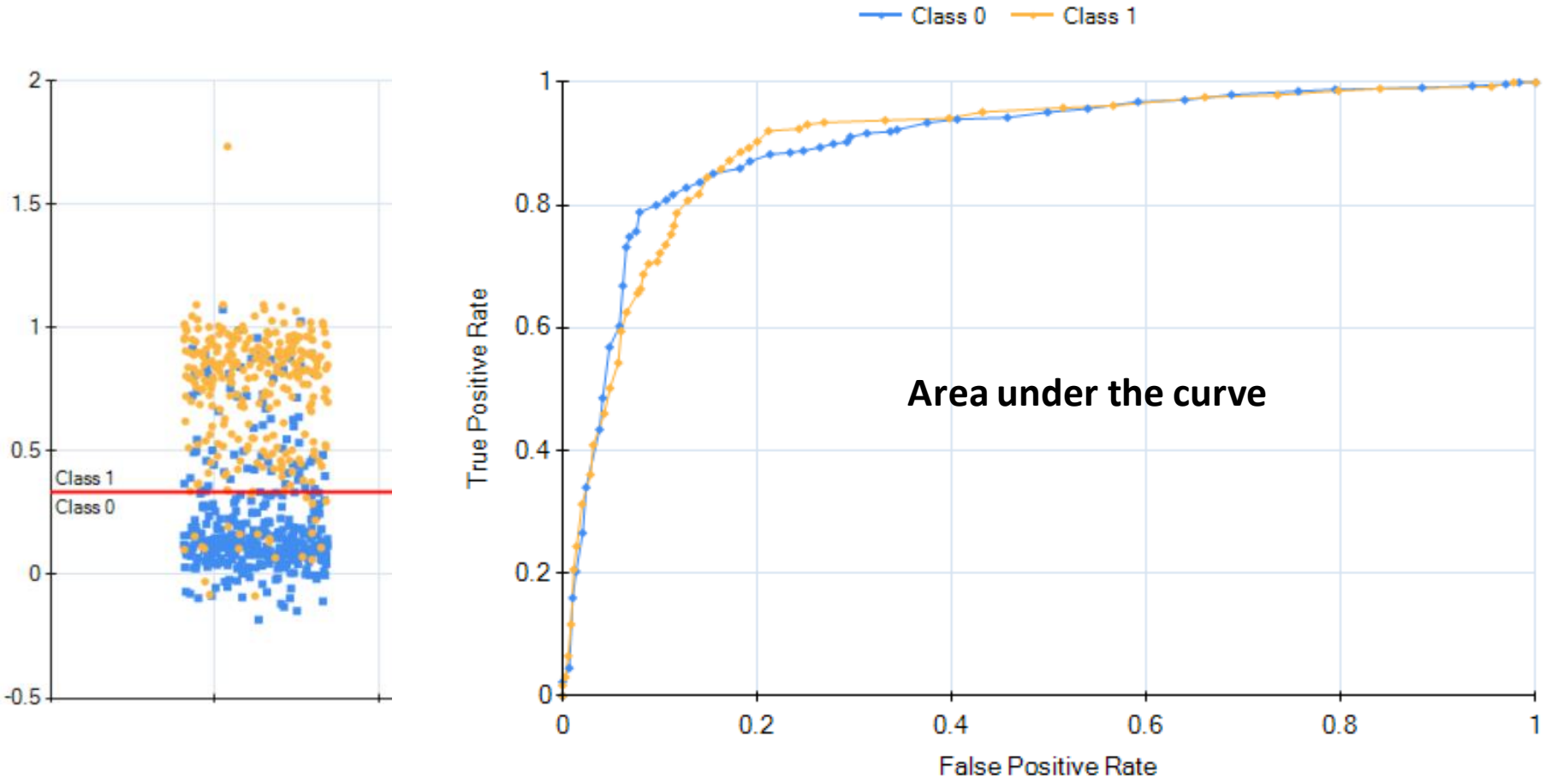
A test which reliably excludes individuals who do not have the condition, resulting in a high number of true negatives and low number of false positives, will have a high specificity.

Important when people who are identified as having a condition may be subjected to more testing, expense, stigma, anxiety, etc.

https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Receiver Operating Characteristics

– Plot of true positive rate vs. false positive rate



ZeroR Model

Classify every sample with the prevalent class label.

$f(X) = y_i$ where y_i is the dominant class

$$\text{Accuracy} = \frac{|\text{samples with } y_i|}{|\text{samples}|}$$

True positive rate = 1.0

False positive rate = 1.0

Useful for judging accuracy values of classification task with imbalanced classes

OneR Model

Properties

- Builds a model using one feature
- Selects the most appropriate feature automatically
- Acts as a baseline for comparison
- Gain knowledge about the data before evaluation more complex models
- Discretization necessary for handling numeric features

Pseudo-Code

```
Foreach input feature
    Foreach value of the feature
        Count class labels of that feature value
        Determine most frequent class label
        Create prediction rule for the feature value and class label combination
    Calculate the error of the created rules for the selected feature
Choose create rules for the features with the smallest error
```

Classification Example

No.	Outlook	Temp (num.)	Temp (nom.)	Humidity (num.)	Humidity (nom.)	Windy	Play Golf
1	sunny	85	hot	85	high	FALSE	no
2	sunny	80	hot	90	high	TRUE	no
3	overcast	83	hot	86	high	FALSE	yes
4	rainy	70	mild	96	high	FALSE	yes
5	rainy	68	cool	80	normal	FALSE	yes
6	rainy	65	cool	70	normal	TRUE	no
7	overcast	64	cool	65	normal	TRUE	yes
8	sunny	72	mild	95	high	FALSE	no
9	sunny	69	cool	70	normal	FALSE	yes
10	rainy	75	mild	80	normal	FALSE	yes
11	sunny	75	mild	70	normal	TRUE	yes
12	overcast	72	mild	90	high	TRUE	yes
13	overcast	81	hot	75	normal	FALSE	yes
14	rainy	71	mild	91	high	TRUE	no

ZeroR and OneR Models

ZeroR

14 total instances / samples

- 9 instances of play golf
- 5 instances of don't play golf

$f(x) = \text{play golf}$

Accuracy = 9 / 14 = 0.643

OneR

Accuracy = 0.714		Play Golf	
		Yes	No
Outlook	sunny	2	3
	overcast	4	0
	rainy	3	2

Accuracy = 0.643		Play Golf	
		Yes	No
Temp.	hot	2	2
	mild	4	2
	cool	3	1

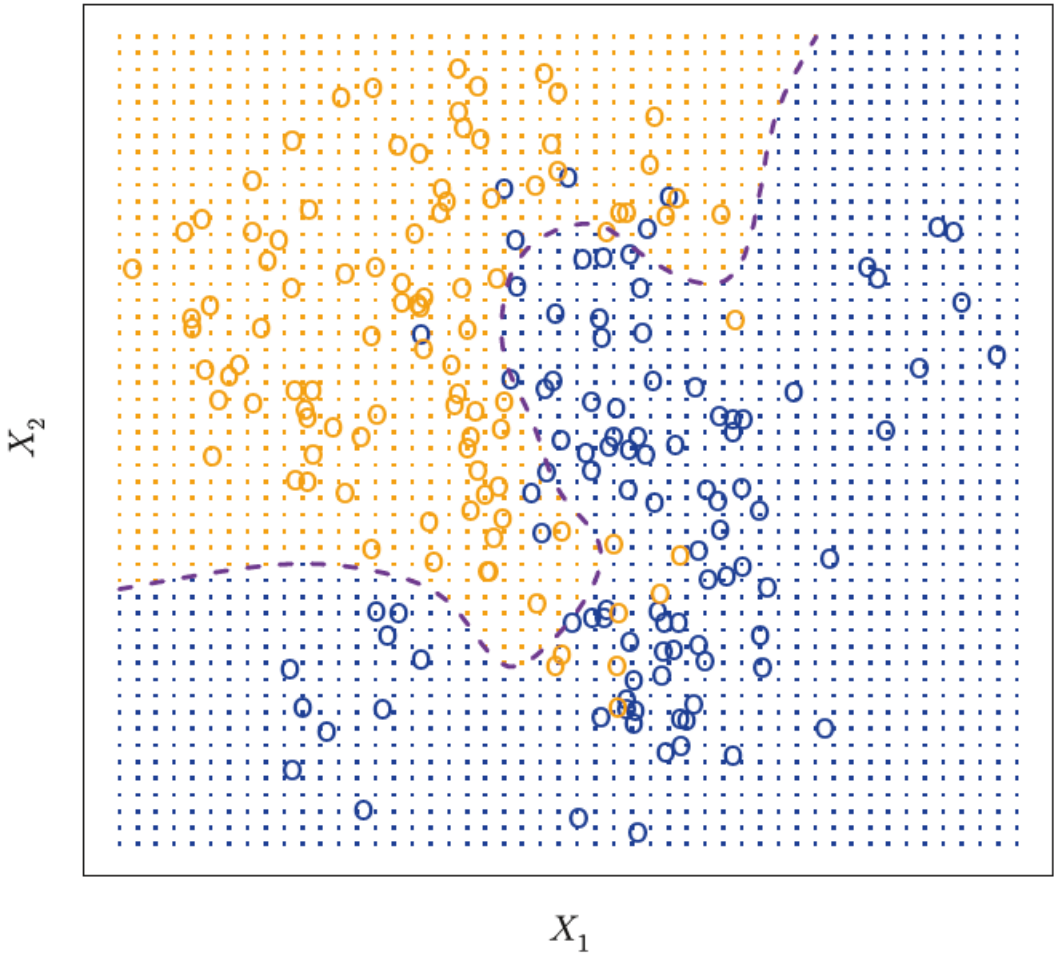
Accuracy = 0.714		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

Accuracy = 0.643		Play Golf	
		Yes	No
Windy	TRUE	3	3
	FALSE	6	2

K-Nearest Neighbor Algorithm

Classification

- Two-dimensional example binary classification example
- Purple line shows optimal Bayes decision boundary
- Bayes classifier $\Pr(Y = j|X = x_0)$
  - Threshold of 0.5 for binary problems
- Theoretical approach because conditional probabilities are in general unknown



K-Nearest Neighbor Classification (KNN)

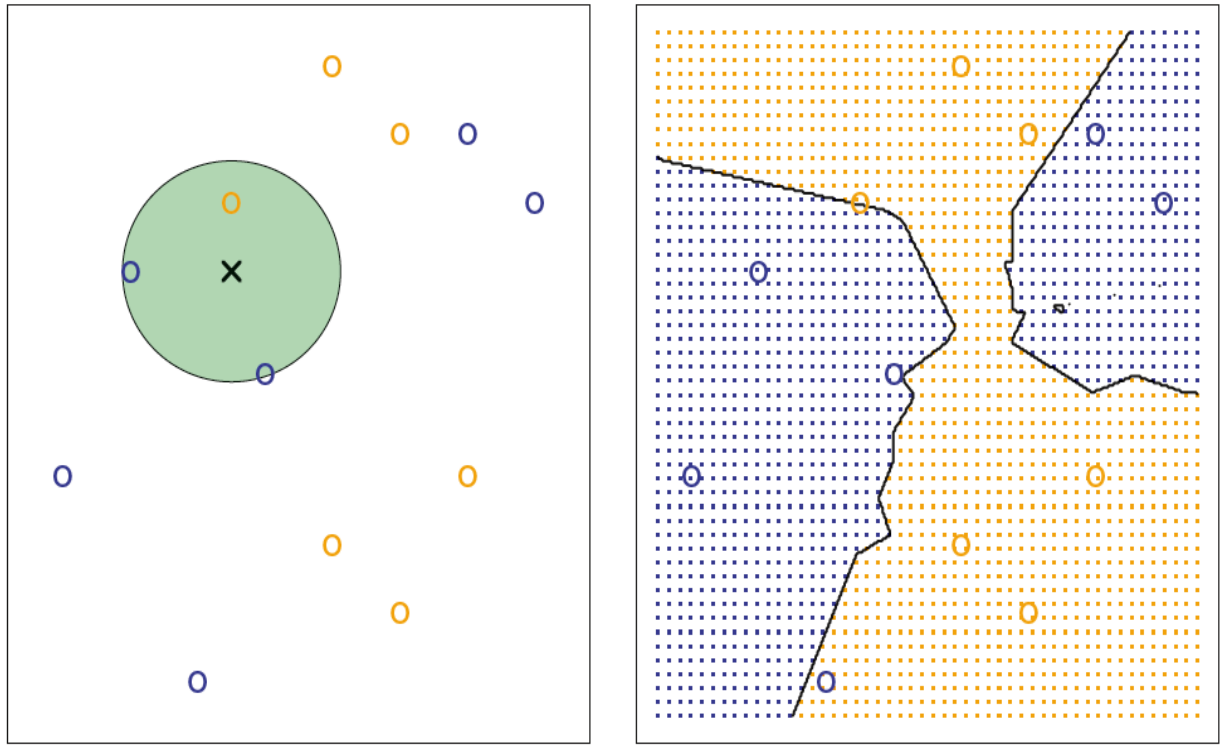
$$\Pr(Y = j|X = x_0) = \frac{1}{K}\sum_{i\in\mathcal{N}_0} I(y_i = j)$$

$\mathcal{N}_0$ = set of K nearest data points

Dependent on distance functions between data points

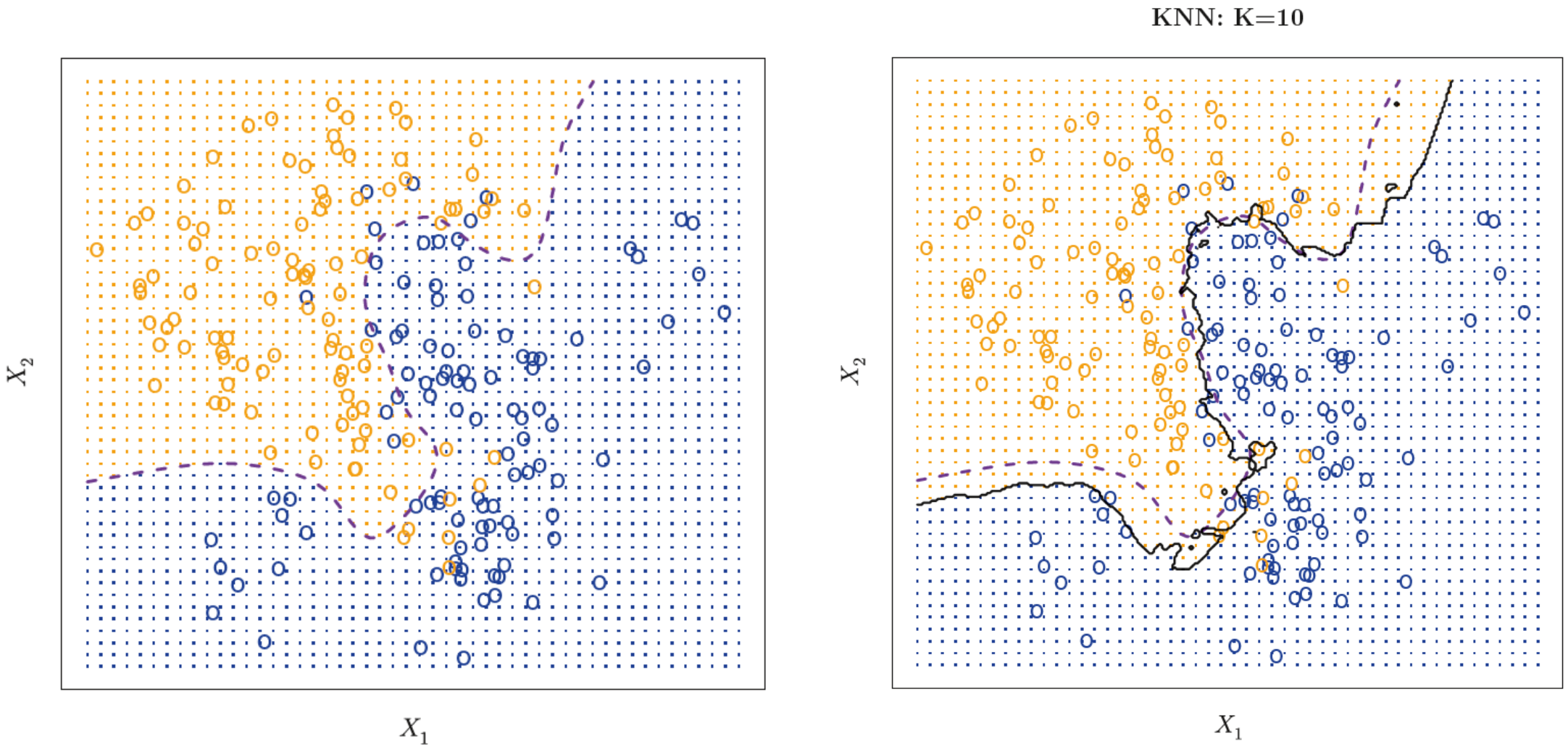
Normalization of features is highly recommended

Lazy learning algorithm



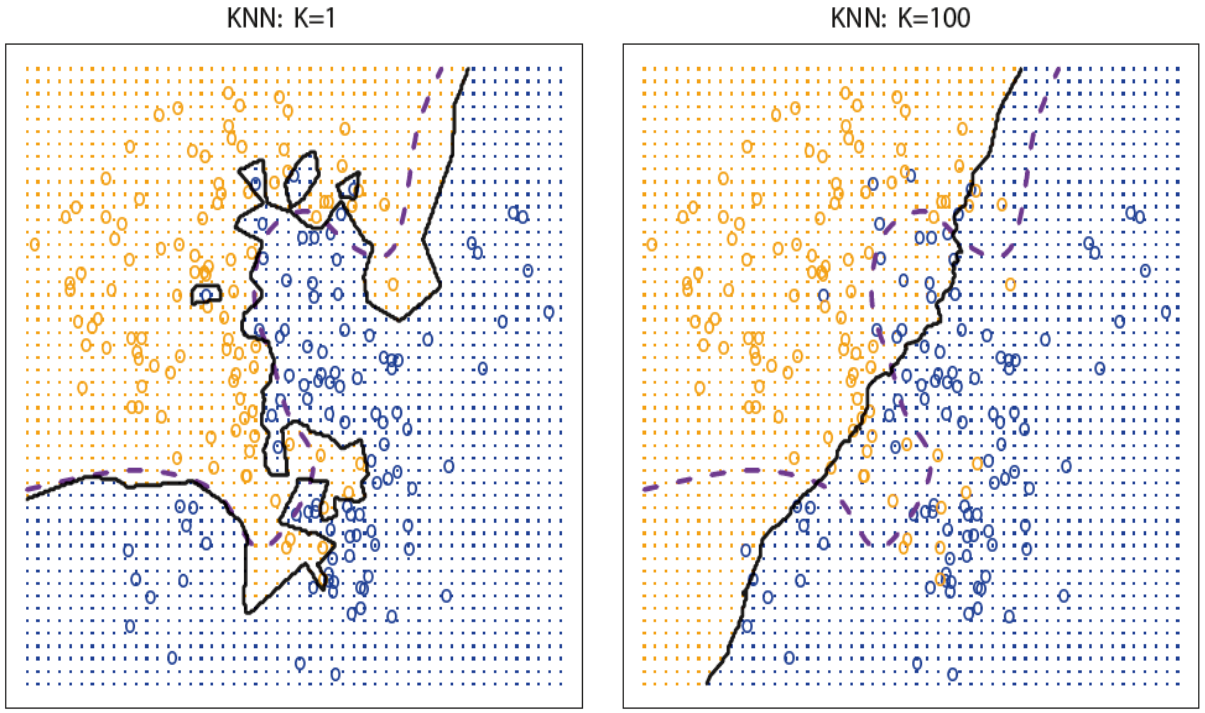
K=3

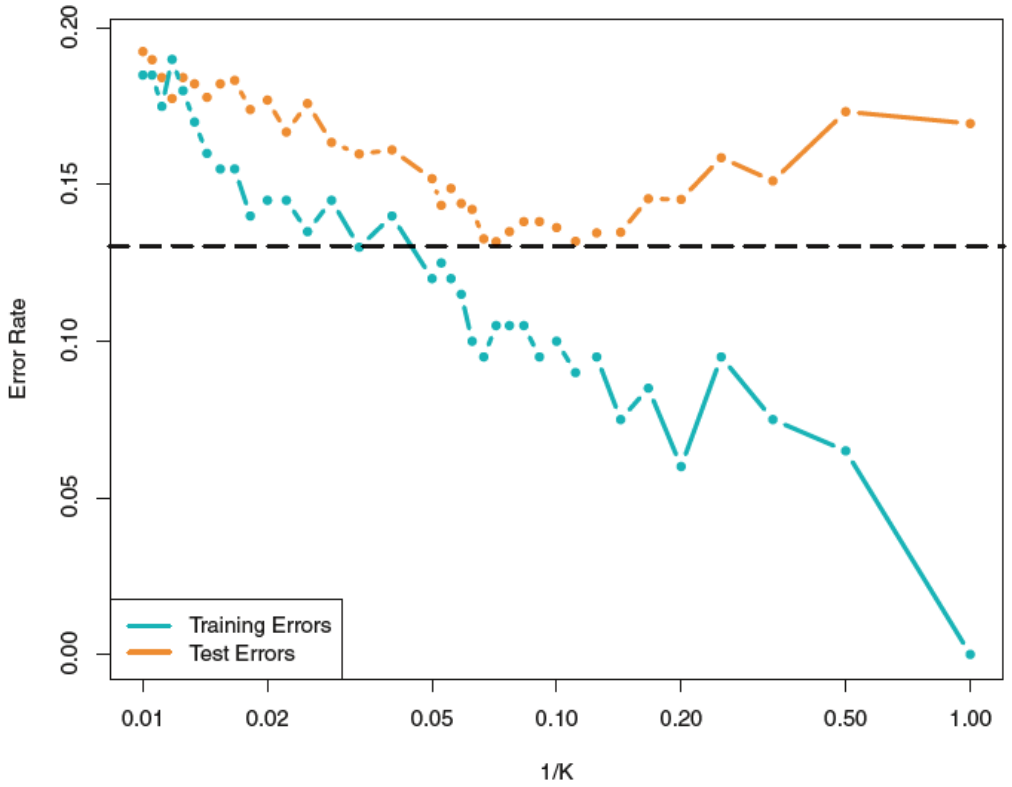
K-Nearest Neighbor Classification



KNN Comparison

Parameter tuning of K (and optionally the distance function) is necessary





Logistic Regression

Why are classification problems not solved with regression algorithms?

- Assign numerical values for class labels
- Learn regression model
- Introduce thresholds to assign classes

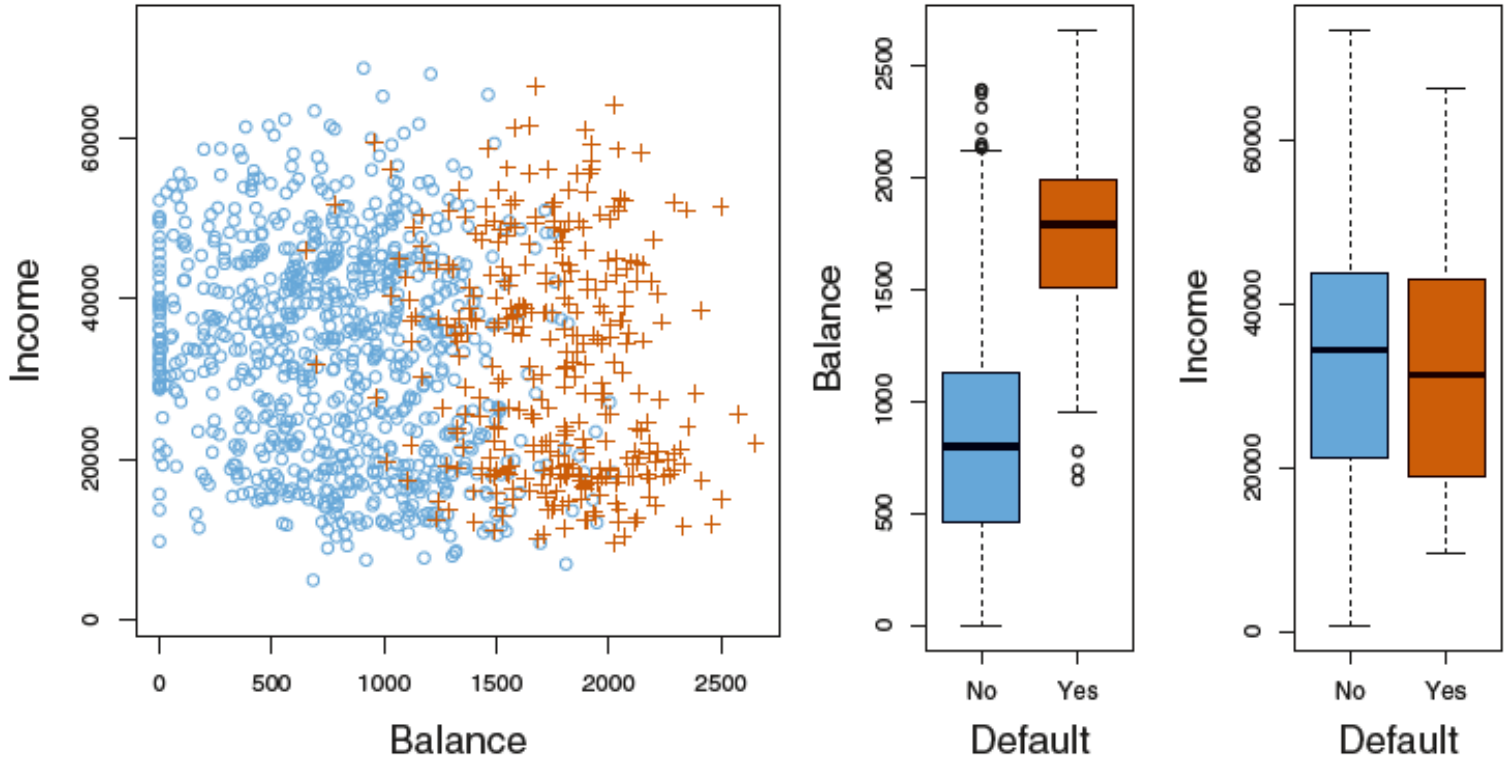
- What happens in the case of multi-class classification?
  - Only reasonably possible for ordinal class labels
  - Not applicable for nominal class labels

- Possible for most binary classification tasks

Default Classification

Binary classification of whether customers can pay their credit card debt.

- Simulated data
- 10,000 instances
  - 333 Default: yes
  - 9667 Default: no



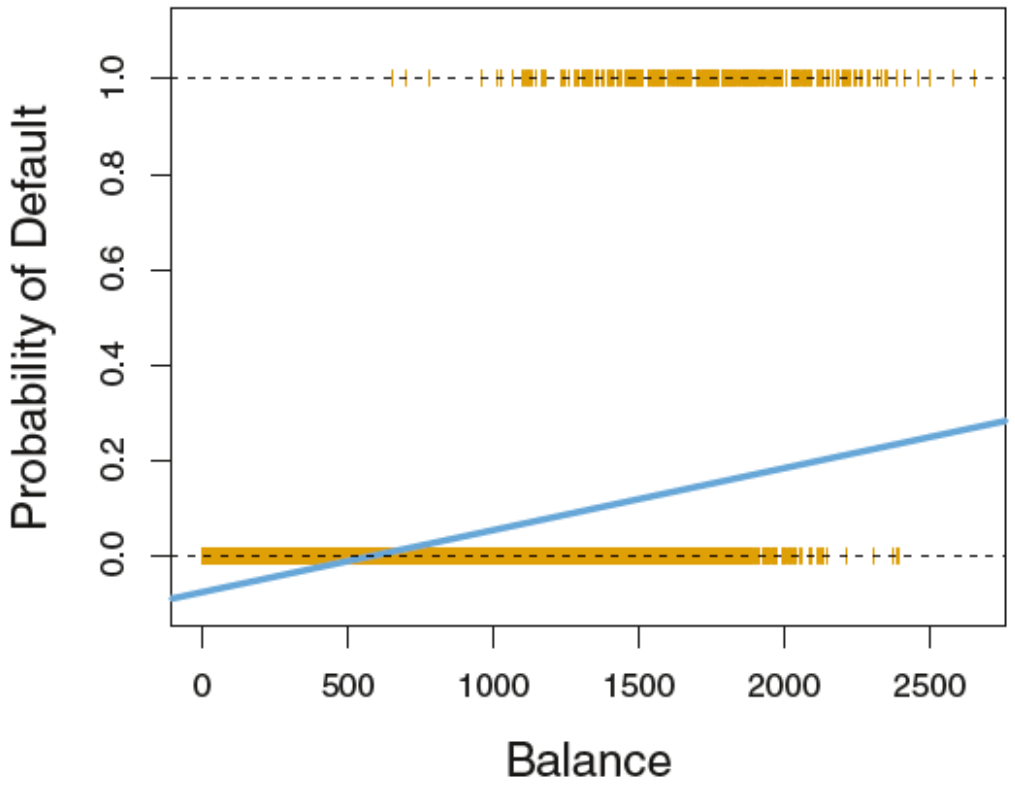
Linear Regression

Learn probability of default as function of the remaining balance.

Encode default = yes with 1.0 and default = no with 0.0

$$f(X) = 0.0001299 \text{ balance } - 0.0751920$$

Probability is not bounded in [0,1]



https://en.wikipedia.org/wiki/Logistic_function

Logistic function

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

x_0 value of the midpoint (0)

L maximum of the curve (1)

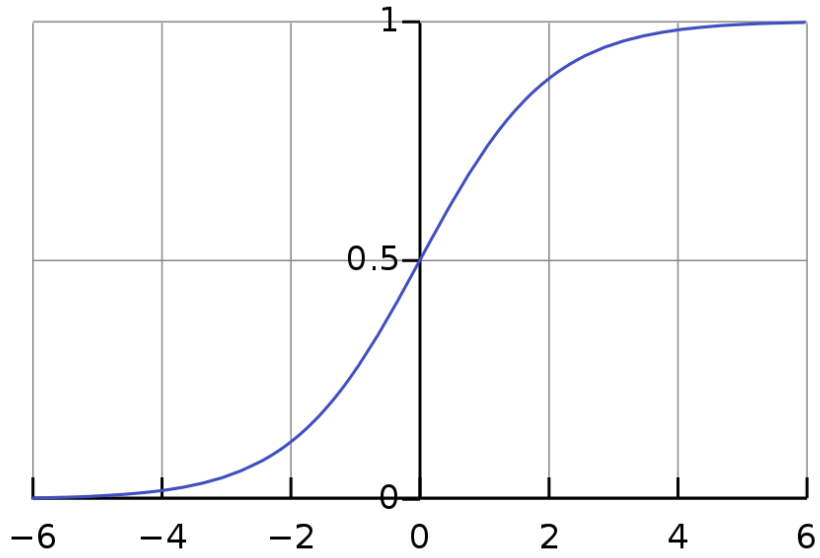
k steepness (1)

Equivalent formulation

$$f(x) = \frac{e^x}{1 + e^x}$$

Use a logistic function instead of a linear one for classification.

$$f(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$



Logistic Regression

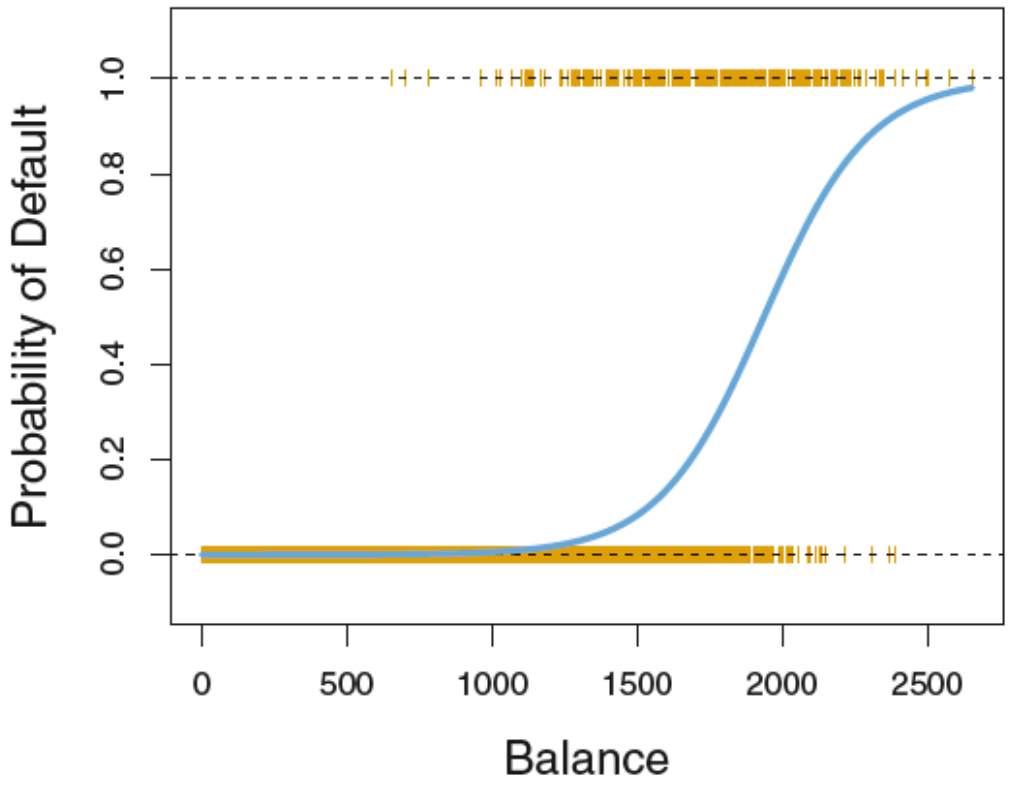
$$\frac{f(X)}{1-f(X)} = e^{\beta_0+\beta X}$$

$$\text{logit}(f(X)) = \beta_0 + \beta X$$

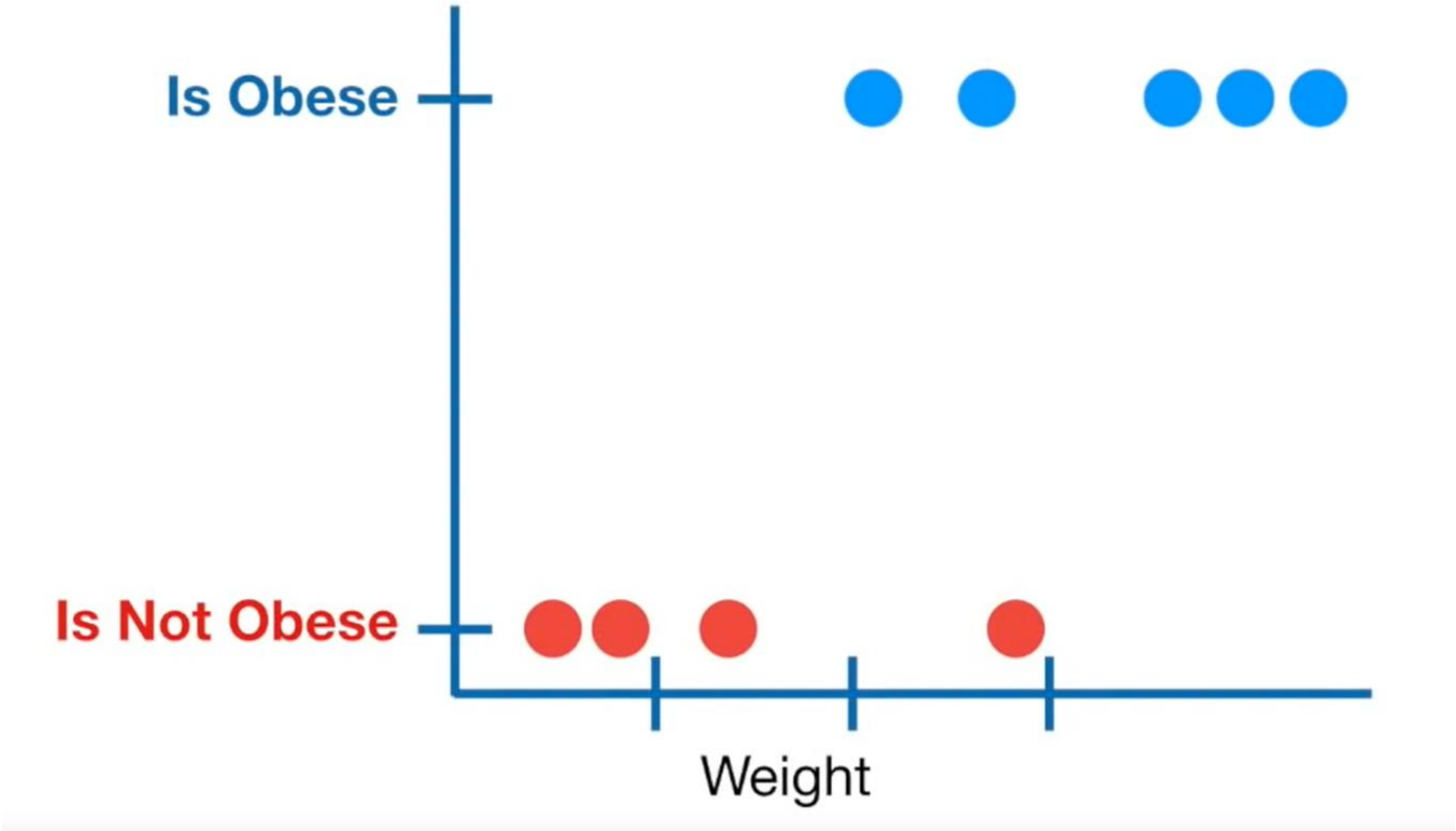
$$f(\text{X}) = \frac{e^{-10.6513+0.0055\,\text{balance}}}{1+e^{-10.6513+0.0055\,\text{balance}}}$$

β is estimated by maximum likelihood method

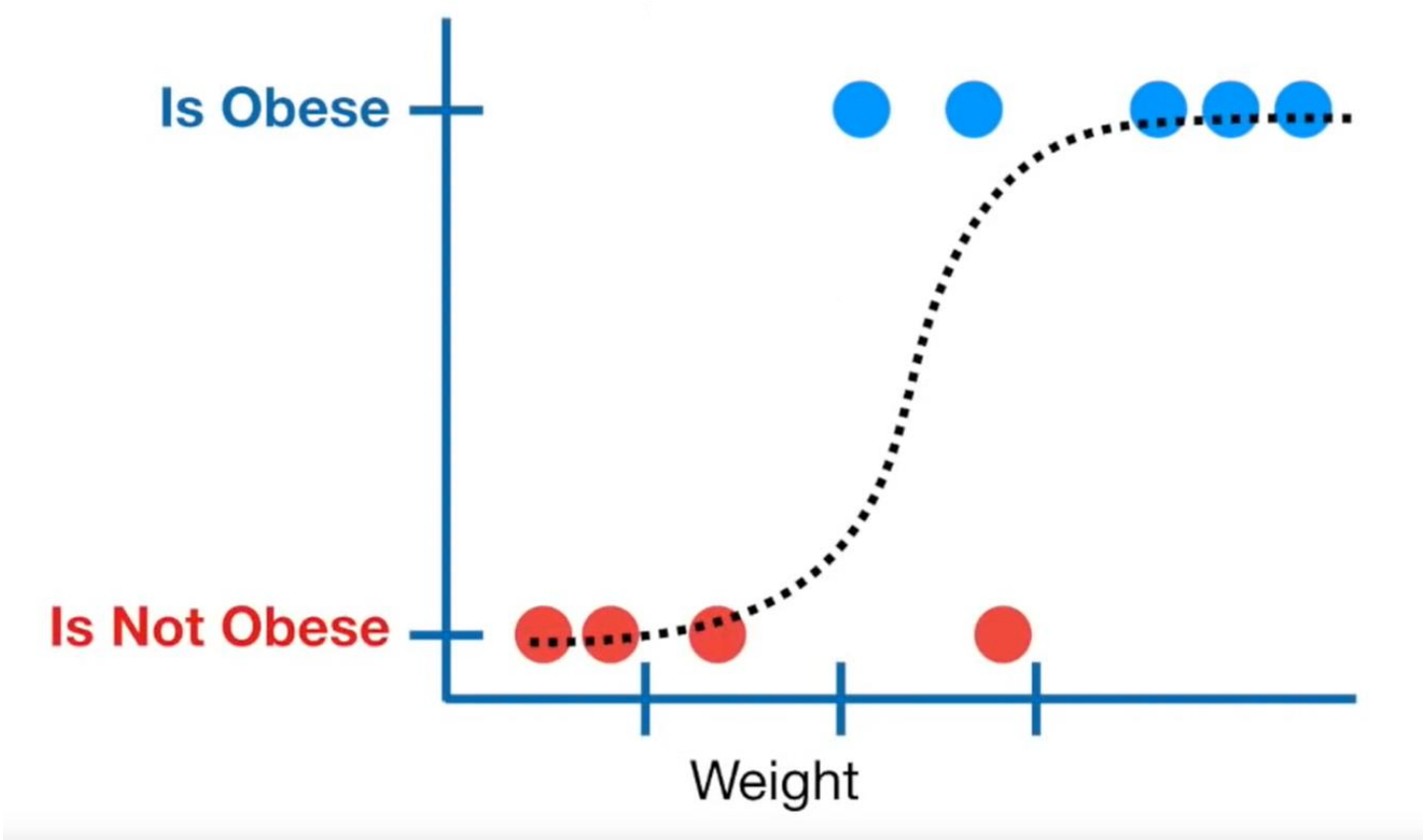
$$L(\beta) = \prod_{i:y_i=1} f(X_i) \prod_{i:y_i=0} (1-f(X_i))$$



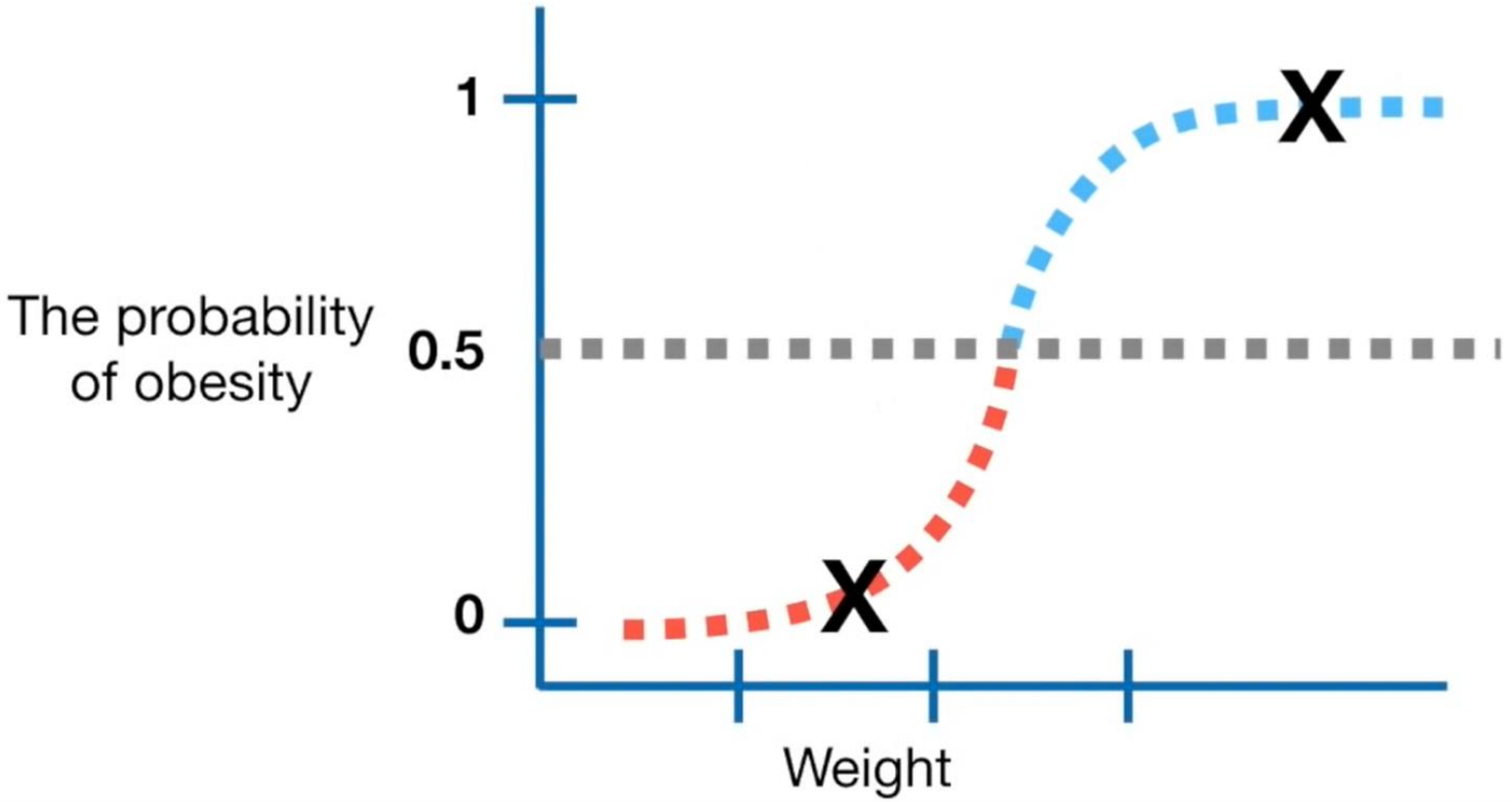
Logistic Regression - Example



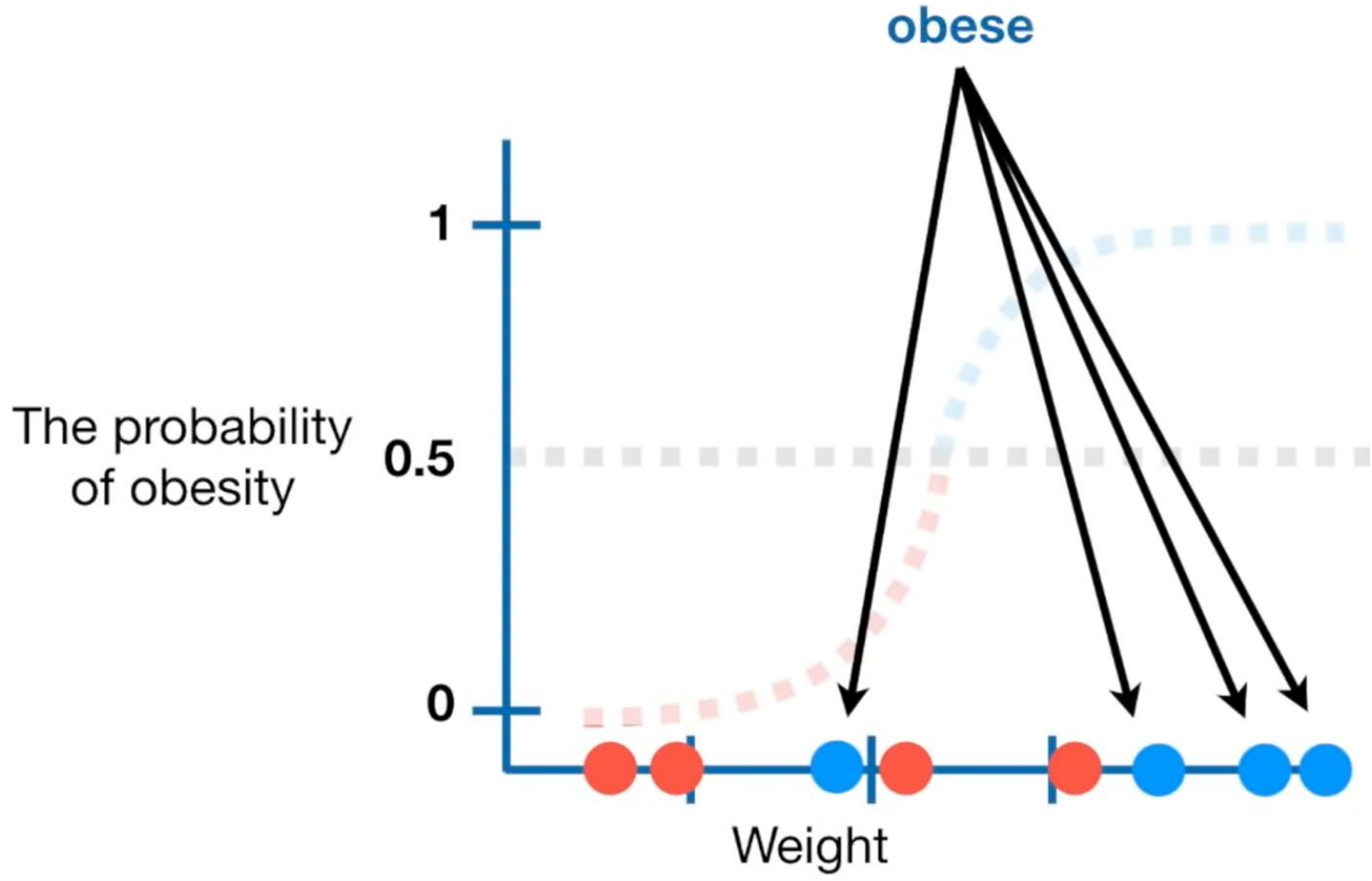
Logistic Regression - Example



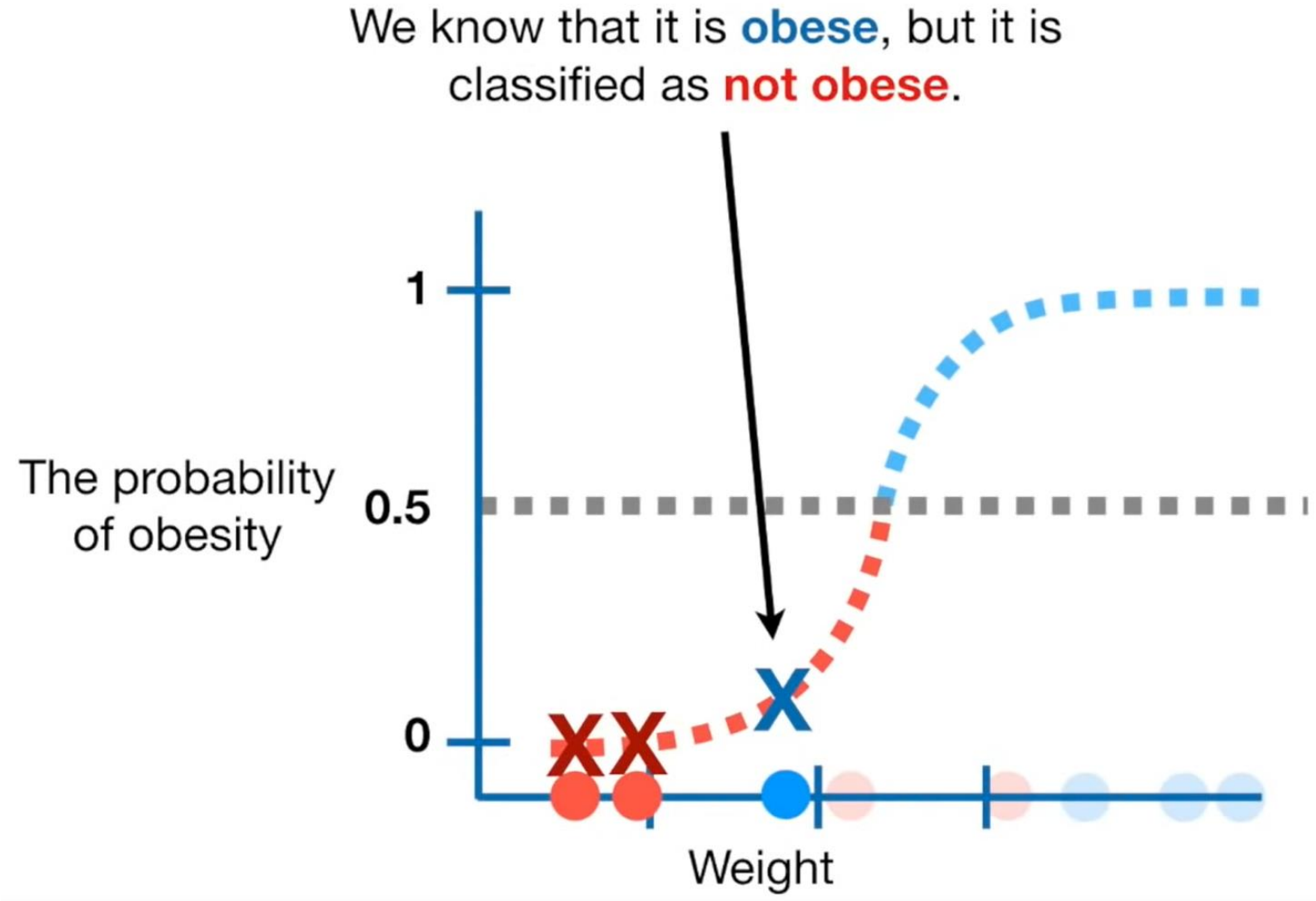
Logistic Regression - Example



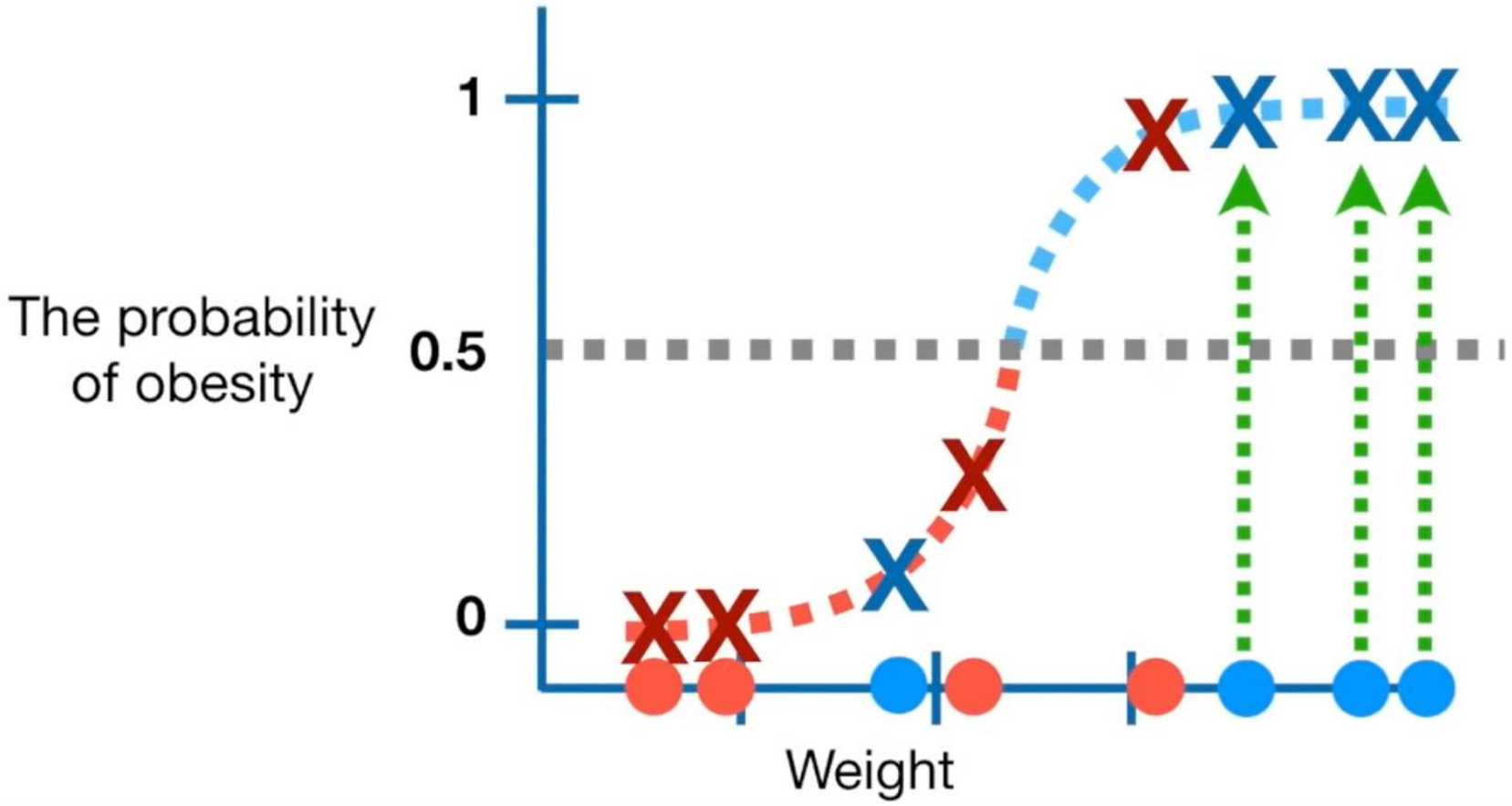
Logistic Regression - Example



Logistic Regression - Example



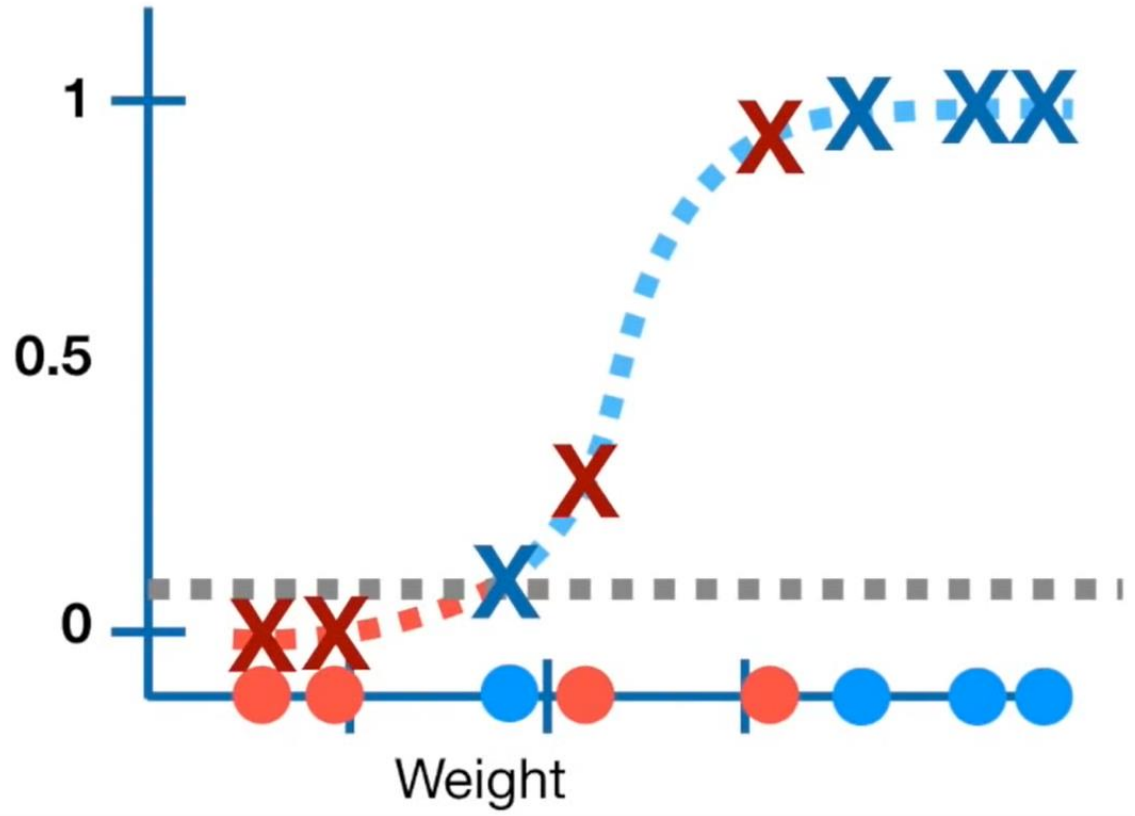
Logistic Regression - Example



Logistic Regression - Example

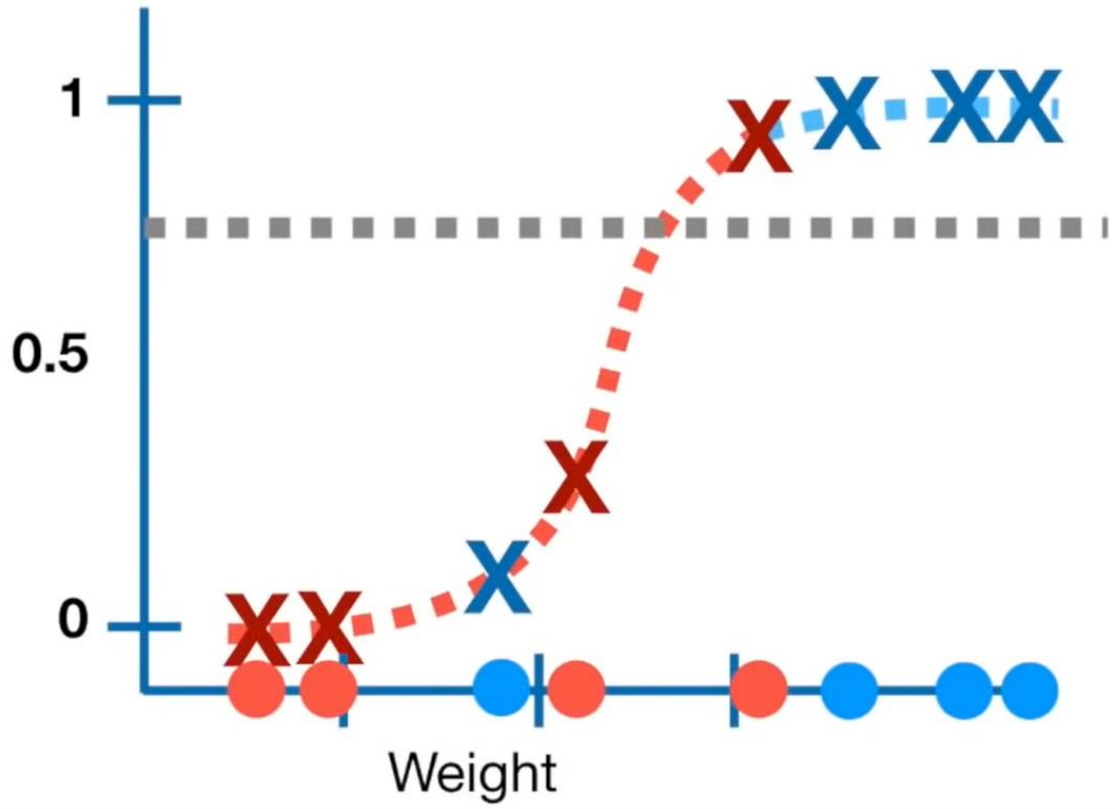
		Actual: Is Obese	Actual: Is Not Obese
Predicted	Is Obese	3	1
	Is Not Obese	1	3

Logistic Regression - Example



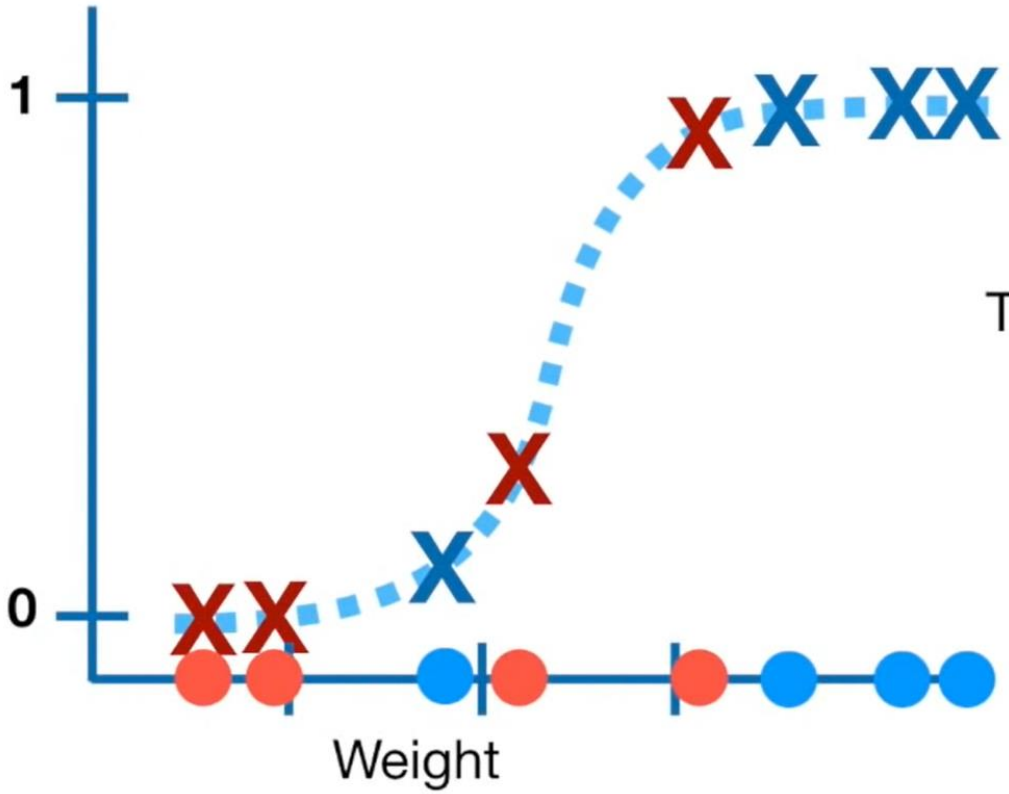
		Actual: Infected	Actual: Not Infected
Predicted	Infected	4	2
Predicted	Not Infected	0	2

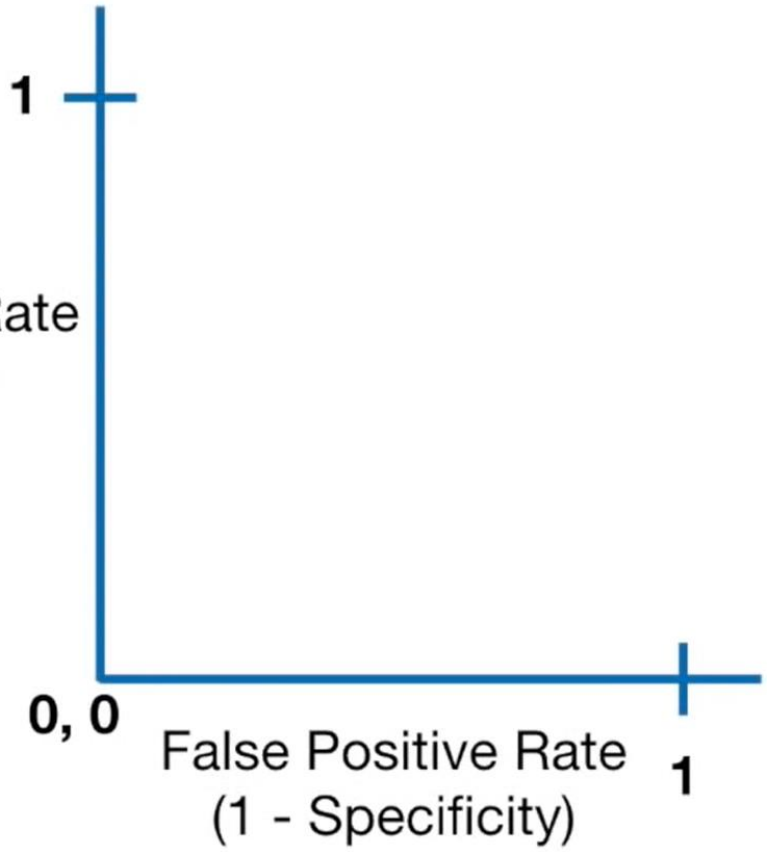
Logistic Regression - Example



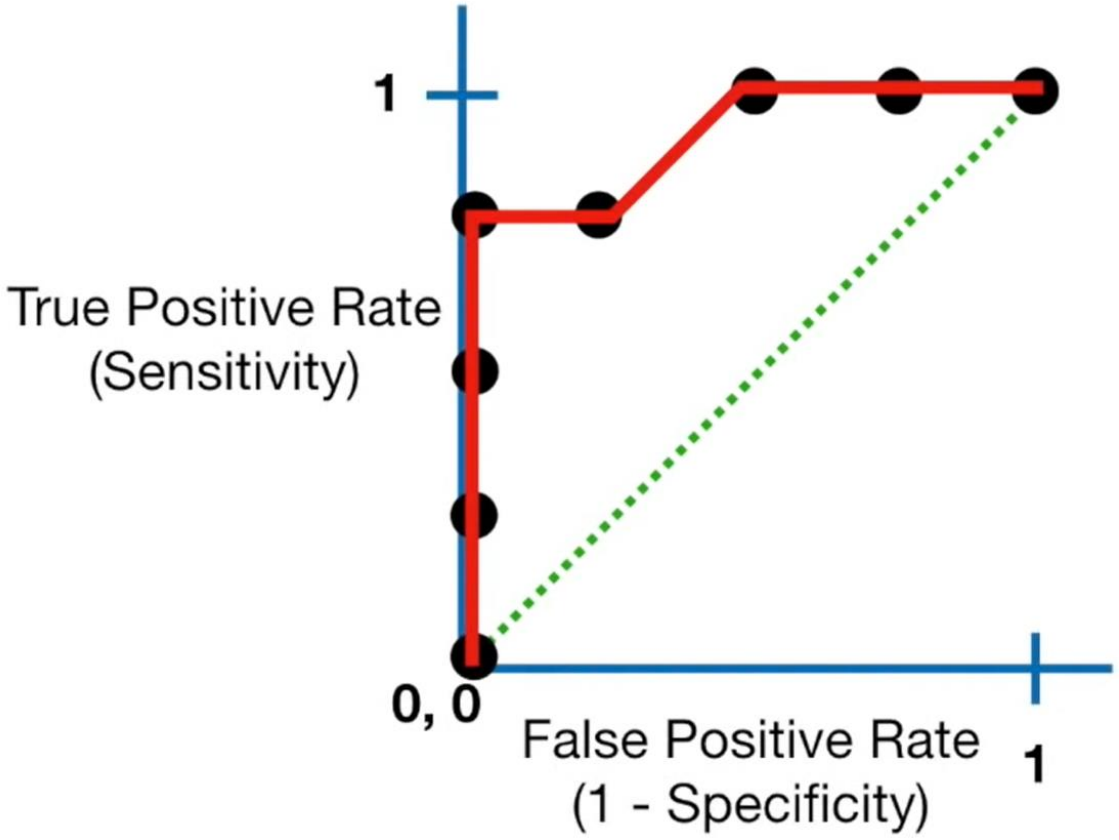
		Actual: Is Obese	Actual: Is Not Obese
Predicted	Is Obese	3	1
	Is Not Obese	1	3

Logistic Regression - Example

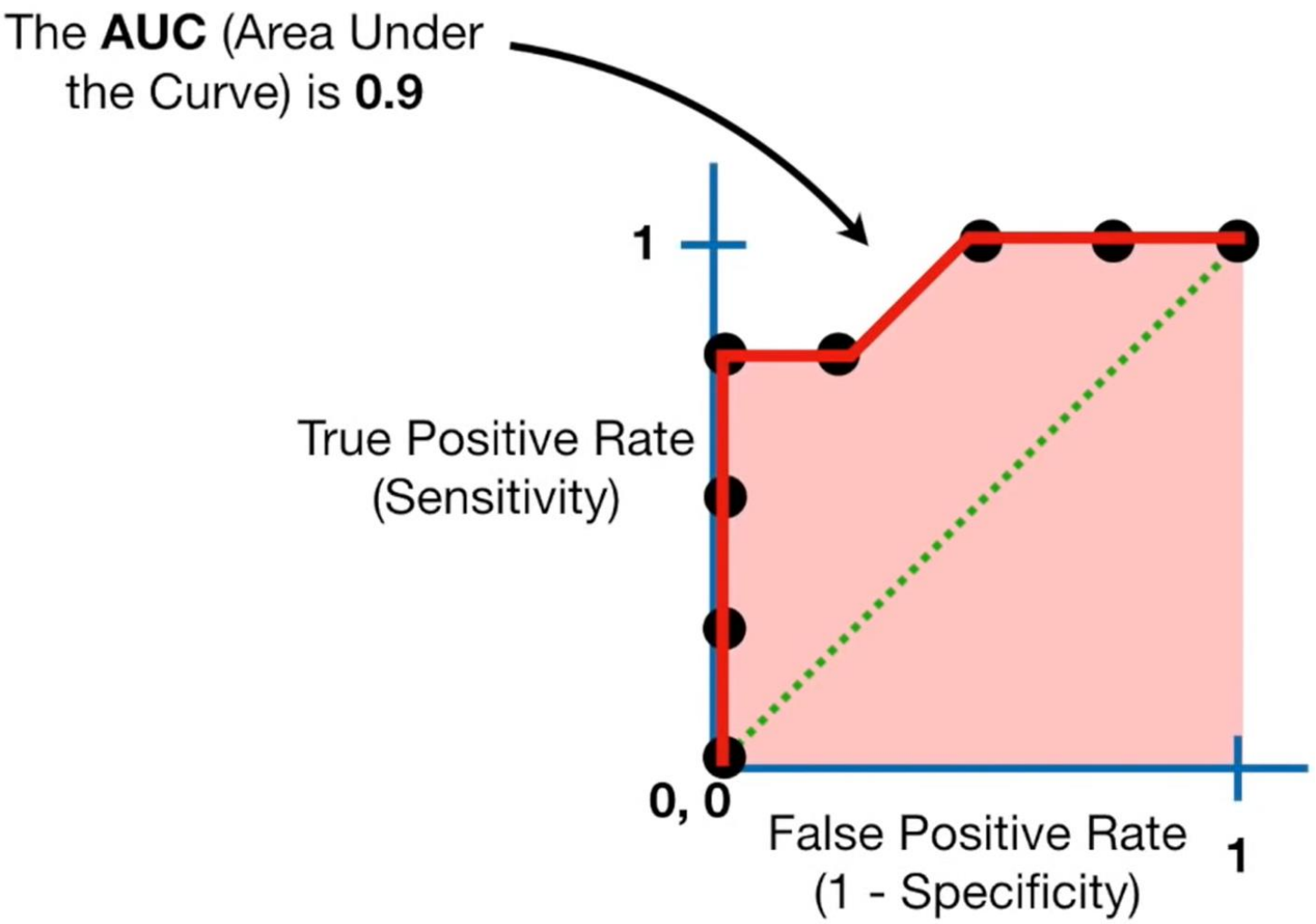




Logistic Regression - Example



Logistic Regression - Example

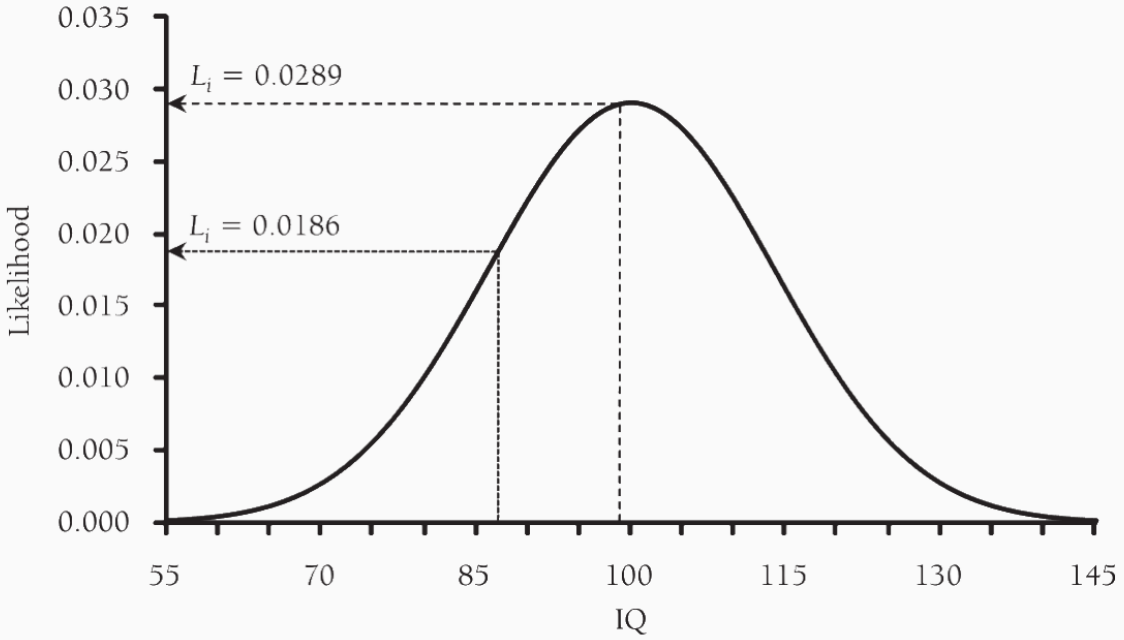


Maximum Likelihood Estimation

Maximum likelihood estimation

Estimate the parameters of the distribution underlying the data.

$$L_i = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-0.5(y_i - \mu)^2}{\sigma^2}$$



IQ	Job performance
78	9
84	13
84	10
85	8
87	7
91	7
92	9
94	9
94	11
96	7
99	7
105	10
105	11
106	15
108	10
112	10
113	12
115	14
118	16
134	12

Maximum Likelihood Estimation

Maximum likelihood estimation

$$L = \prod L_i$$

Since L_i values are very small, the product is prone to numerical issues.

For this reason, we resort (again) to a $\log$ transformation:

$$\log L = \sum \log L_i$$

This makes the sample log-likelihood equal to the sum of the individual log-likelihood values. [4]

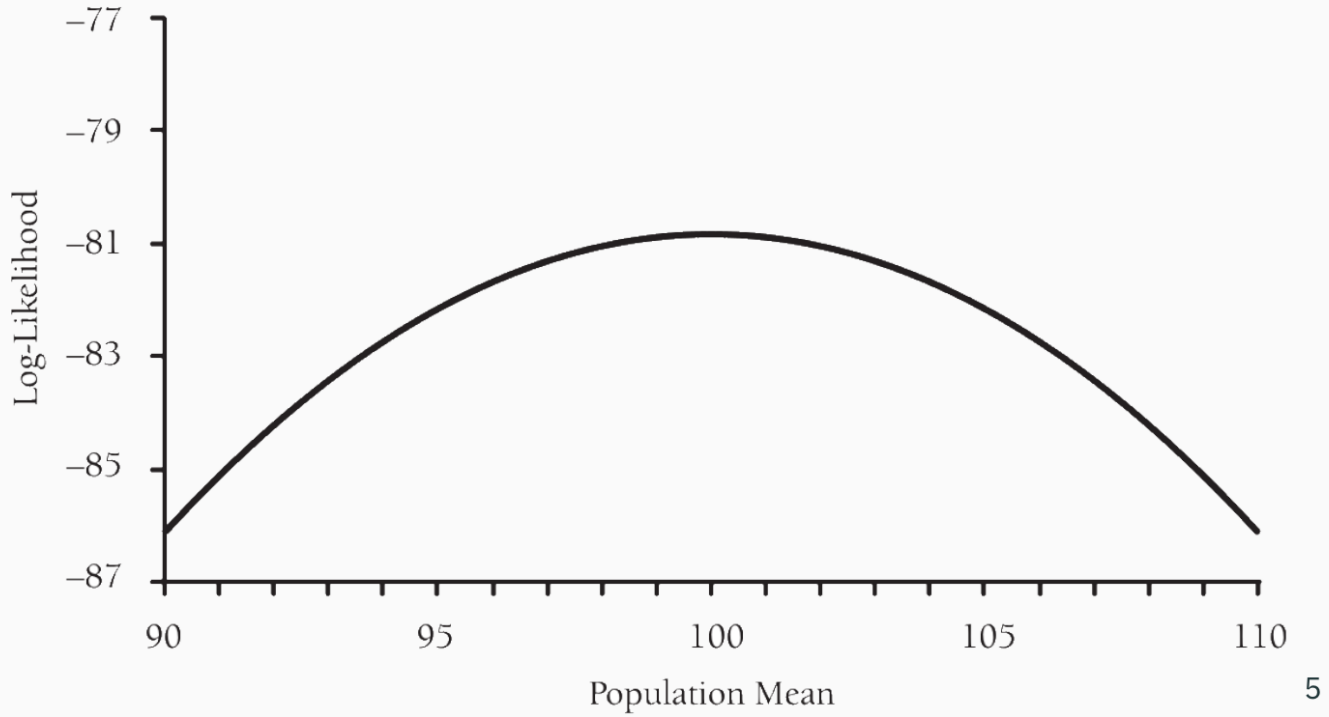
[4] http://www.appliedmissingdata.com/

Maximum Likelihood Estimation

Maximum likelihood estimation

Population parameters typically need to be estimated from the data.

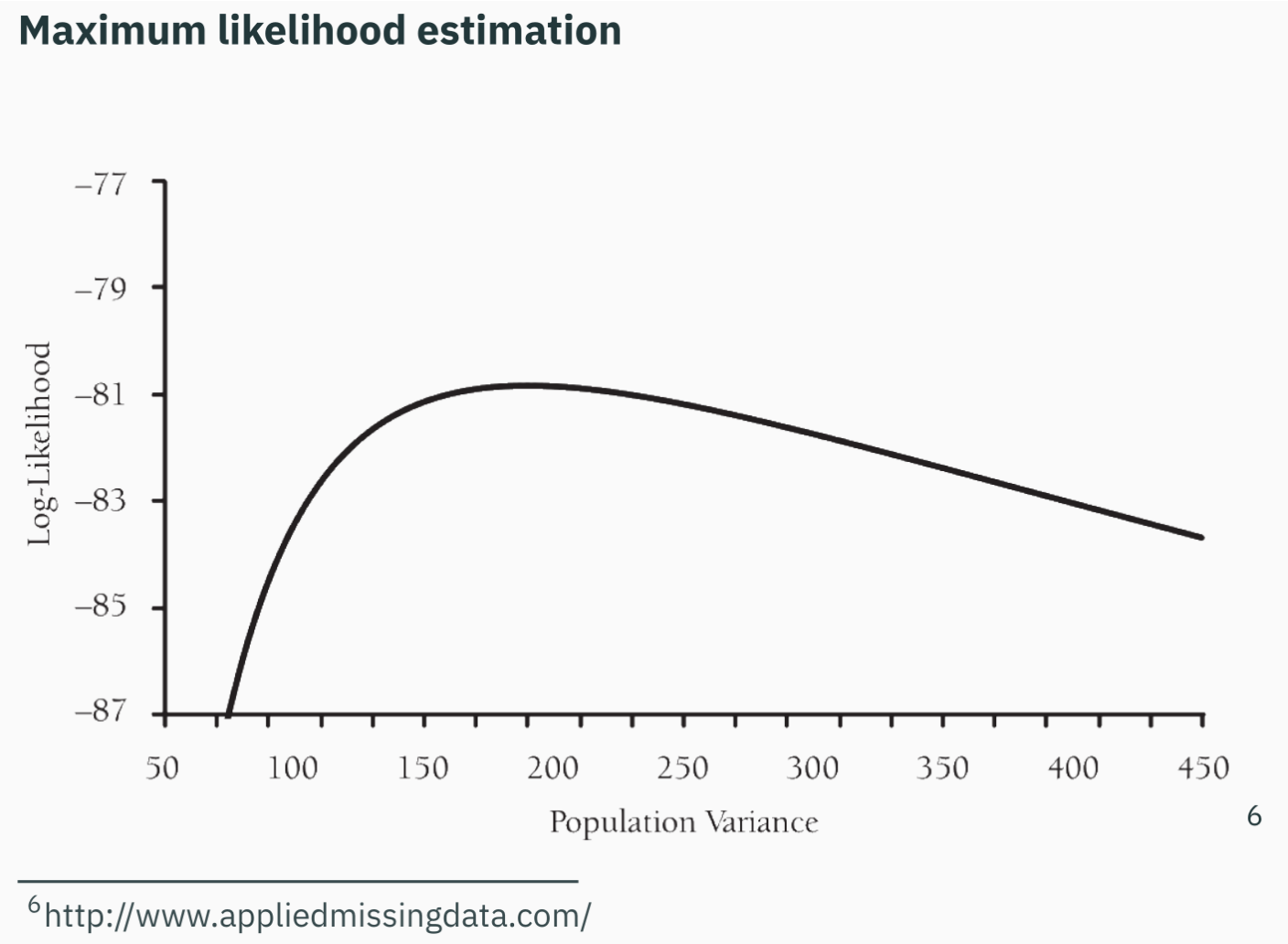
We iteratively test a set of plausible values and calculate the log-likelihood.



5

[5] http://www.appliedmissingdata.com/

Maximum Likelihood Estimation



[6] http://www.appliedmissingdata.com/