Advanced Text Analysis

Day 3 / Session 1

Petro Tolochko

Machine Learning

- Supervised
  - An outcome variable is defined
  - Focus is on prediction
- Unsupervised
  - No outcome variable has been defined
  - Focus is on patterns

Machine Learning

- **Supervised**
  - **An outcome variable is defined**
  - **Focus is on prediction**
- Unsupervised
  - No outcome variable has been defined
  - Focus is on patterns

Supervised

- Objective:
    - Classification of documents into pre existing categories

Supervised

- Create a labeled data set
- Classify documents with supervised learning algorithm
- Check performance

Labeled Dataset

- How:
    - Human coders annotate parts of the corpus (see slides in session 1 of today)
    - Found data (e.g., self-reported profession in users' profile)

- Considerations:
    - Sampling should be representative for the corpus (e.g., Random, Stratified sample e.g., across time and source)
    - Quality of human coding matters (Assess the intercoder reliability)
    - Number of documents

Labeled Dataset

- Number of documents
    - the higher the number of categories and the lower the reliability of the coders, the higher the number of documents (Barberá et al., 2021)
- increase the sizes of manually coded validation dataset as large as possible, preferably to more than N = 1,300 (i.e., more than 1% of all data to be examined), assuming acceptable reliability (equal to or higher than .7) (Song et al., 2021)

Splitting the Data

- Split labeled data in training data and test data (validation data)
- Training data
    - The subset that is used to learn the model parameters
- Test data
    - Another subset used to evaluate the model's predictive quality
    - Not used for learning!
- Validation data

Document Classification

- Classifier learns the mapping between features and the labels in the training set
- define a model f(Y)=g(X)
- And apply a learning algorithm to establish which features in X (features extracted from the training documents) matter to recover Y (i.e, the labels of the training documents)
- We fit the model

Classify documents with supervised learning

- Considerations:
  - Feature representation (Bag of words representation or embeddings)
  - Feature selection (remove irrelevant features)
  - Classifier selection
    - E.g., Naive Bayes, SVM, KNN, or ensemble methods

Checking Performance

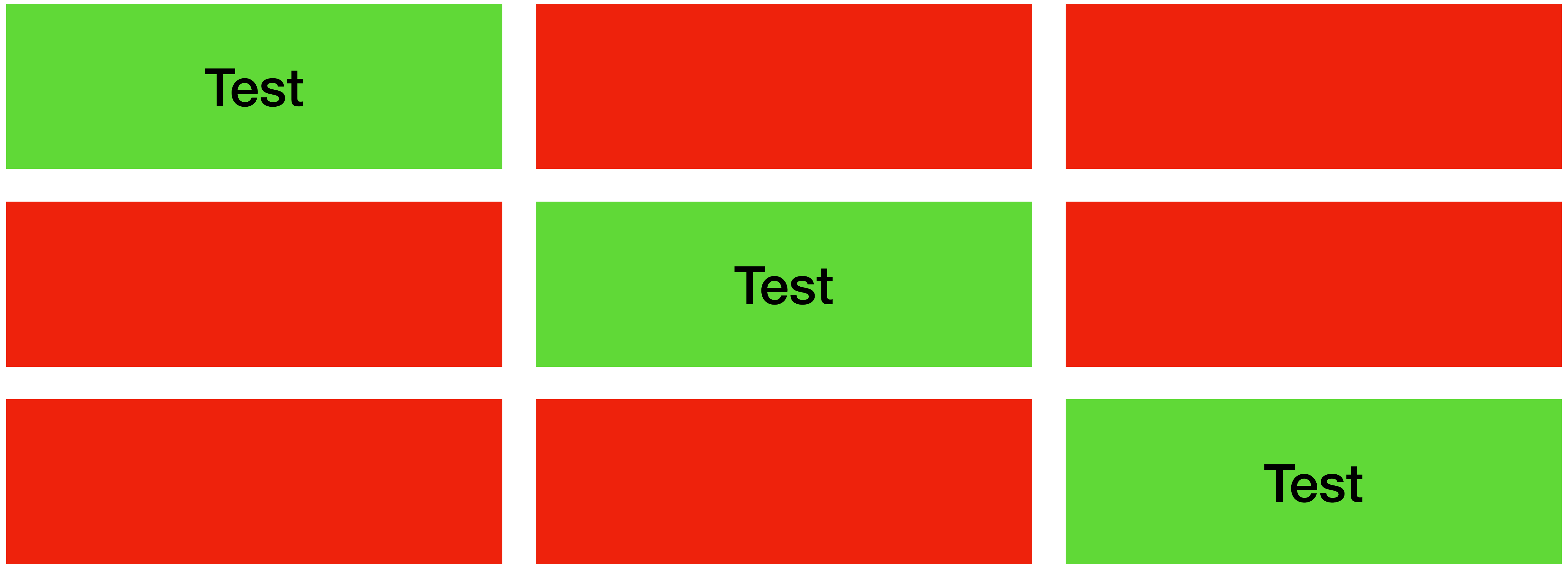
- The fitted model (the trained classifier) is applied to a held-out test set (which is a part of the labeled set but was not used for training the model).

- Considerations:
    - Danger of overfitting (focus on features that work well with training set but do not generalize)
    - Solutions: cross-validation
    - Performance metric (i.e., recall, precision)

Checking performance

- k-fold cross-validation
  - We randomly split the data into k sets (“folds”) of roughly equal size
  - Each set is hold out once as test set, while training on the remaining sets
  - The problem of a lucky split is reduced

K-Fold Cross-Validation



Confusion Matrix

	Actual label	
	Negative	Positive
Negative	True negative	False positive
Positive	False negative	True positive

Precision/Recall

$$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Negative + True\ Positive + False\ Negative + True\ Positive}$$

$$Precision_{positive} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall_{positive} = \frac{True\ Positive}{True\ Positive + False\ Negatives}$$

Dictionary vs. supervised machine learning

- Dictionaries can be applied directly to a new corpus (but validate!)

- Supervised machine learning requires (potentially larger amounts) labeled data

- If the training sample is large enough supervised learning will outperform dictionaries

Additional considerations

- Hyperparameter selection
    - Via systematic comparison of different hyperparameters per algorithm
- Random undersampling (Galar et al., 2011)
- Method to deal with unbalanced classes: use the max. number of positive instances per class and randomly sample the same number of instances of the negative class

Naïve Bayes Classifier

- Probabilistic classifier
- Simple
- Fast
- Good Accuracy

Bayes Theorem

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

Bayes Theorem

Likelihood of observing the evidence B given that event A is true

Prior probability of event A, representing our initial belief or knowledge about A before considering the evidence

Posterior probability of event A given the evidence B

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

Probability of observing the evidence B

Bayes Theorem

$$P(A|B) \propto P(B|A) \times P(A)$$

Naïve Bayes Classifier

$$P(C_k | x_1, x_2, ..., x_n)$$

Naïve Bayes Classifier

Features

$$P(C_k|x_1, x_2, ..., x_n)$$

Class

Naïve Bayes Classifier

Features

$$P(C_k|x_1, x_2, \dots, x_n)$$

Class

Features are assumed to be independent. Hence, "***Naïve***"

Naïve Bayes Classifier

$$P(C_k|\mathbf{x}) = \frac{P(C_k) \times P(\mathbf{x}|C_k)}{P(\mathbf{x})}$$

Naïve Bayes Classifier

$$P(C_k|\mathbf{x}) \propto P(C_k) \times P(\mathbf{x}|C_k)$$

Naïve Bayes Classifier

$$\begin{aligned}
p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\
&\propto p(C_k)\ p(x_1 \mid C_k)\ p(x_2 \mid C_k)\ p(x_3 \mid C_k)\ \cdots \\
&\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,,
\end{aligned}$$

Decision Rule

$$\hat{y} = argmax\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Naïve Bayes Classifier

- Implemented in many stats/ML packages

Support Vector Machine

- Comes from computer science
- Very good
- Rather difficult math

- Considered one of the best of-the-shelf classification algorithms

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:

$$\beta_0 + \beta_1 X_1 = 0$$

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:
- line equation

$$\beta_0 + \beta_1 X_1 = 0$$

Hyperplane

- n-1 dimensional plane that separates the n-dimensional space
- 2-dimensional hyperplane:
- line equation

$$\beta_0 + \beta_1 X_1 = 0$$

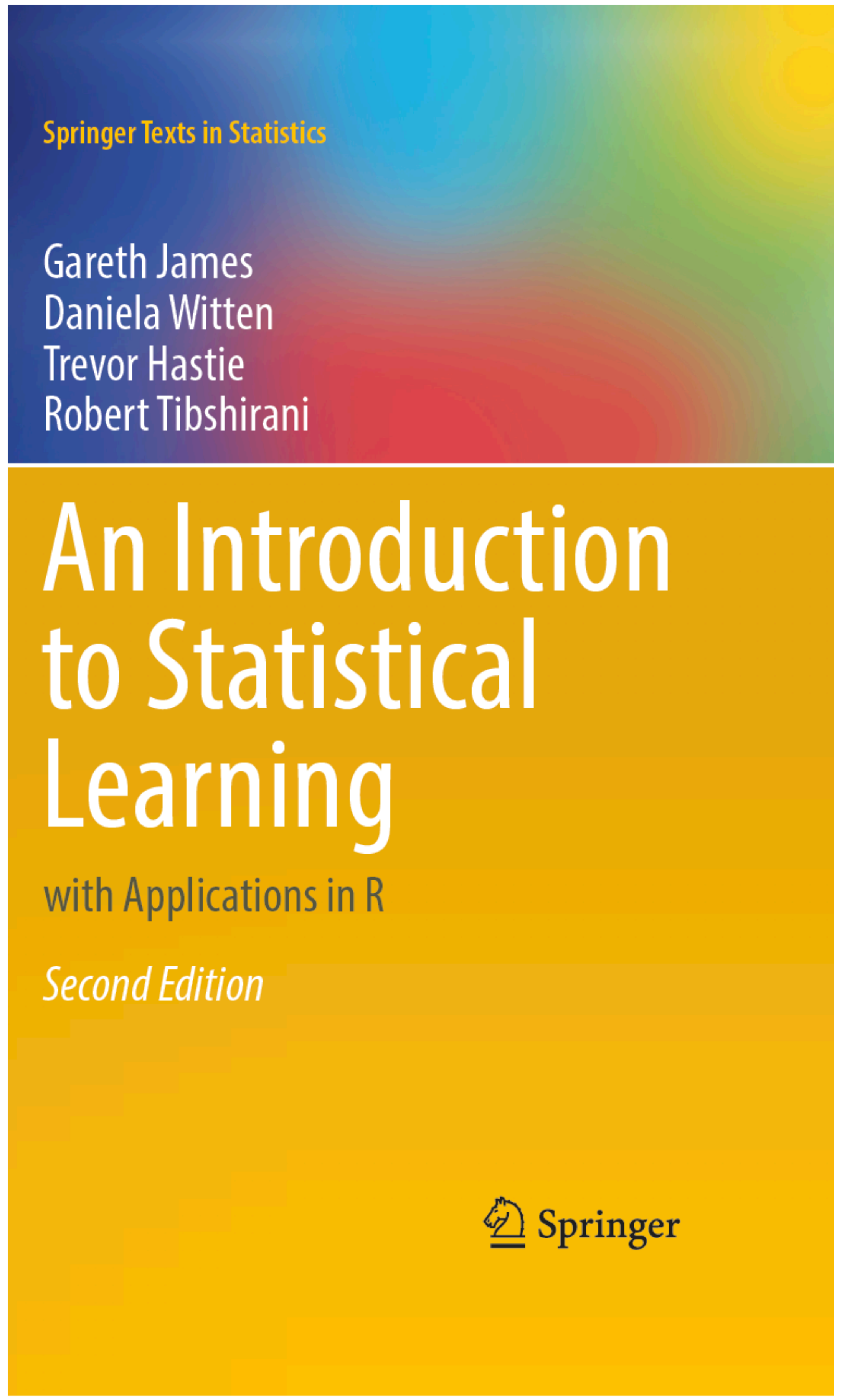
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

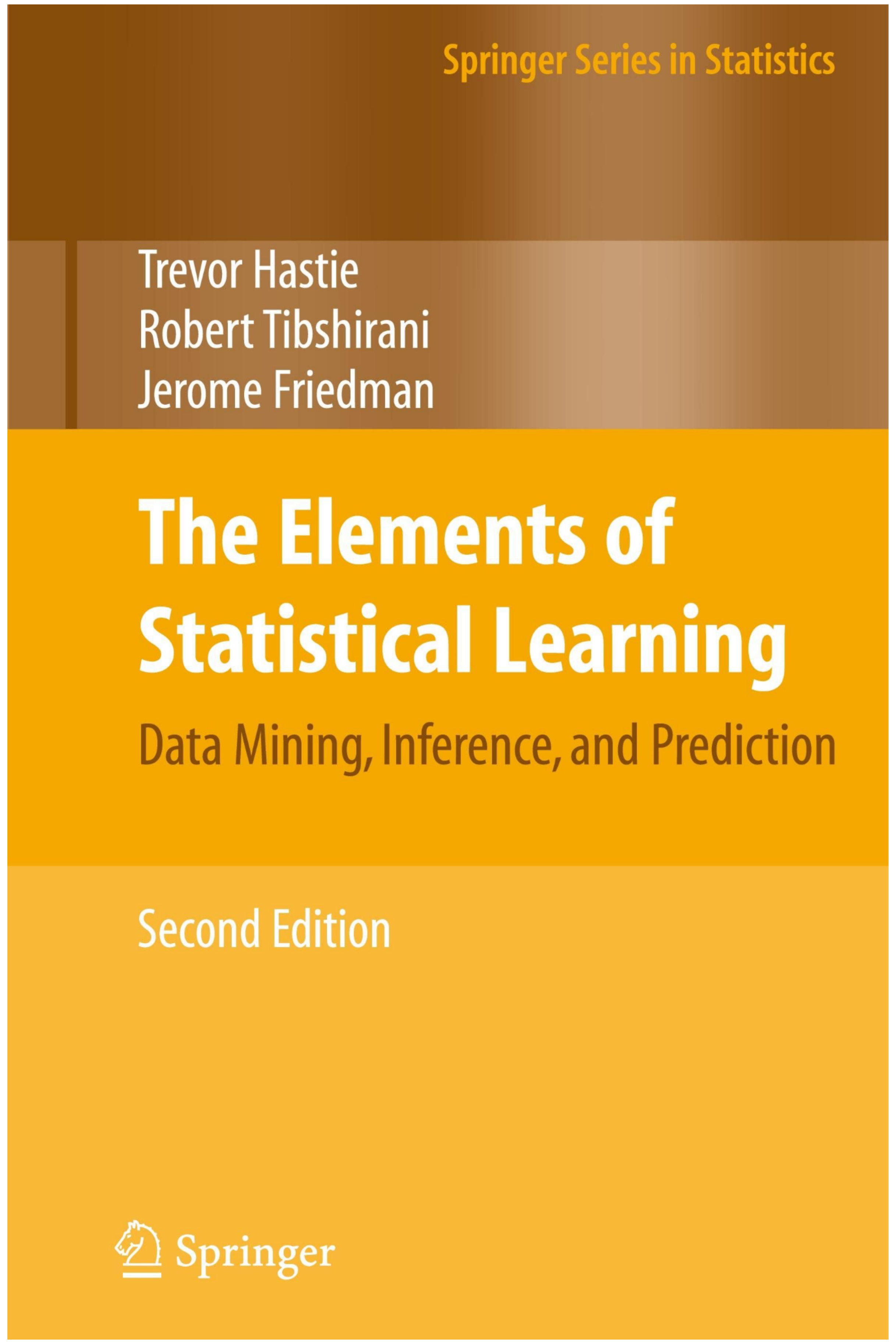
Classification

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0.$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0,$$

Following images from:





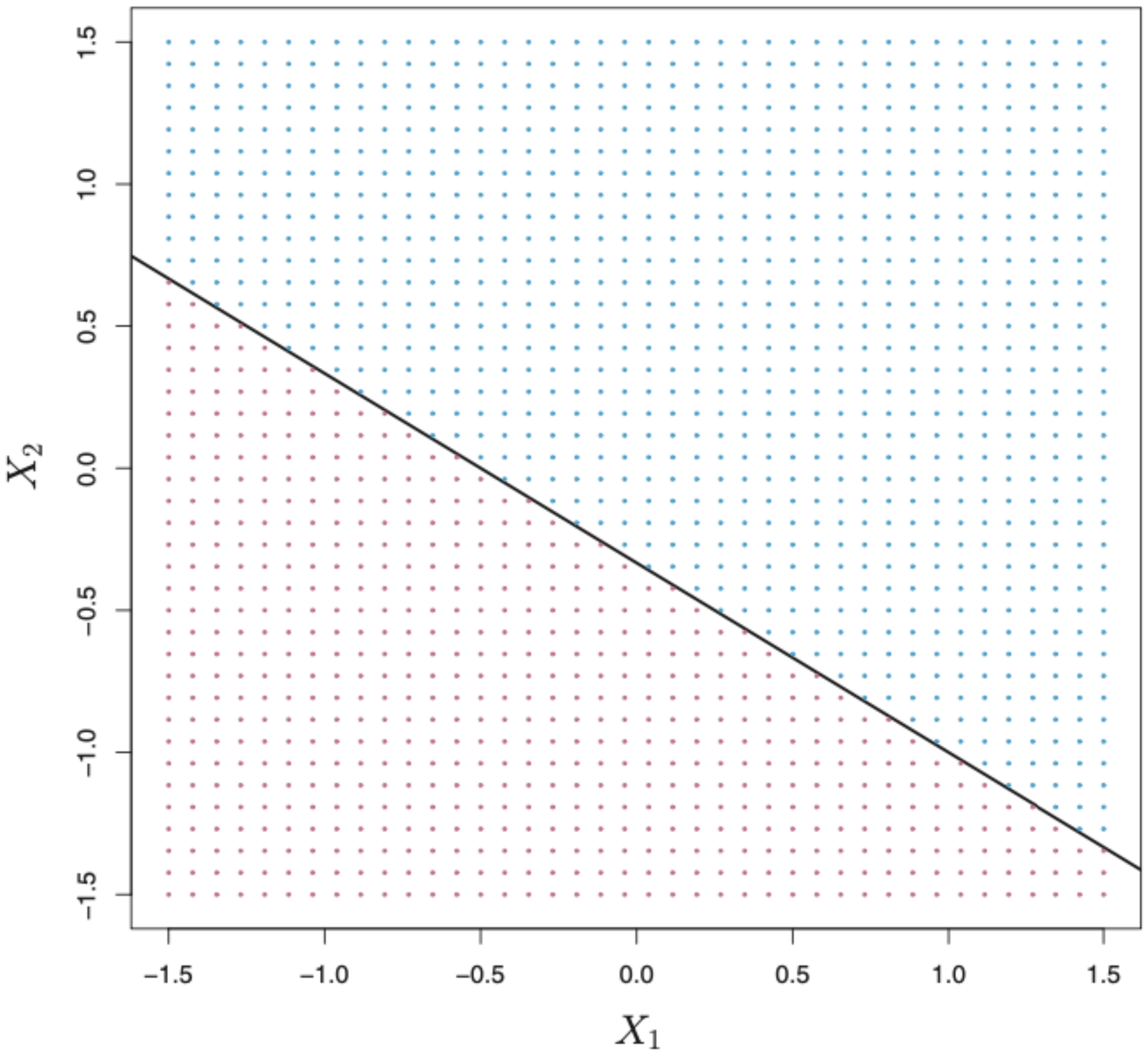
Springer Texts in Statistics

Gareth James · Daniela Witten · Trevor Hastie ·
Robert Tibshirani · Jonathan Taylor

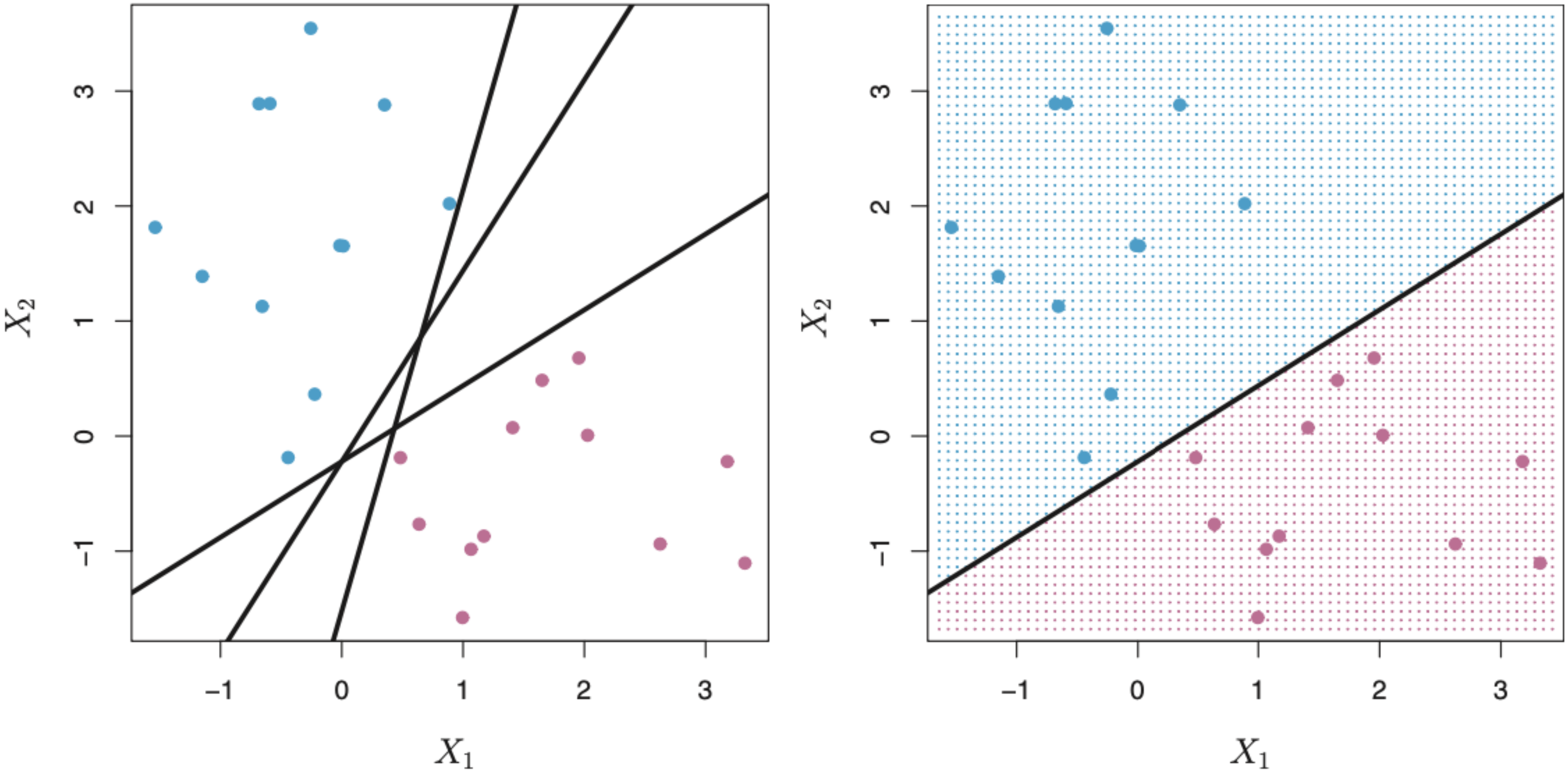
An Introduction to Statistical Learning

with Applications in Python

Springer

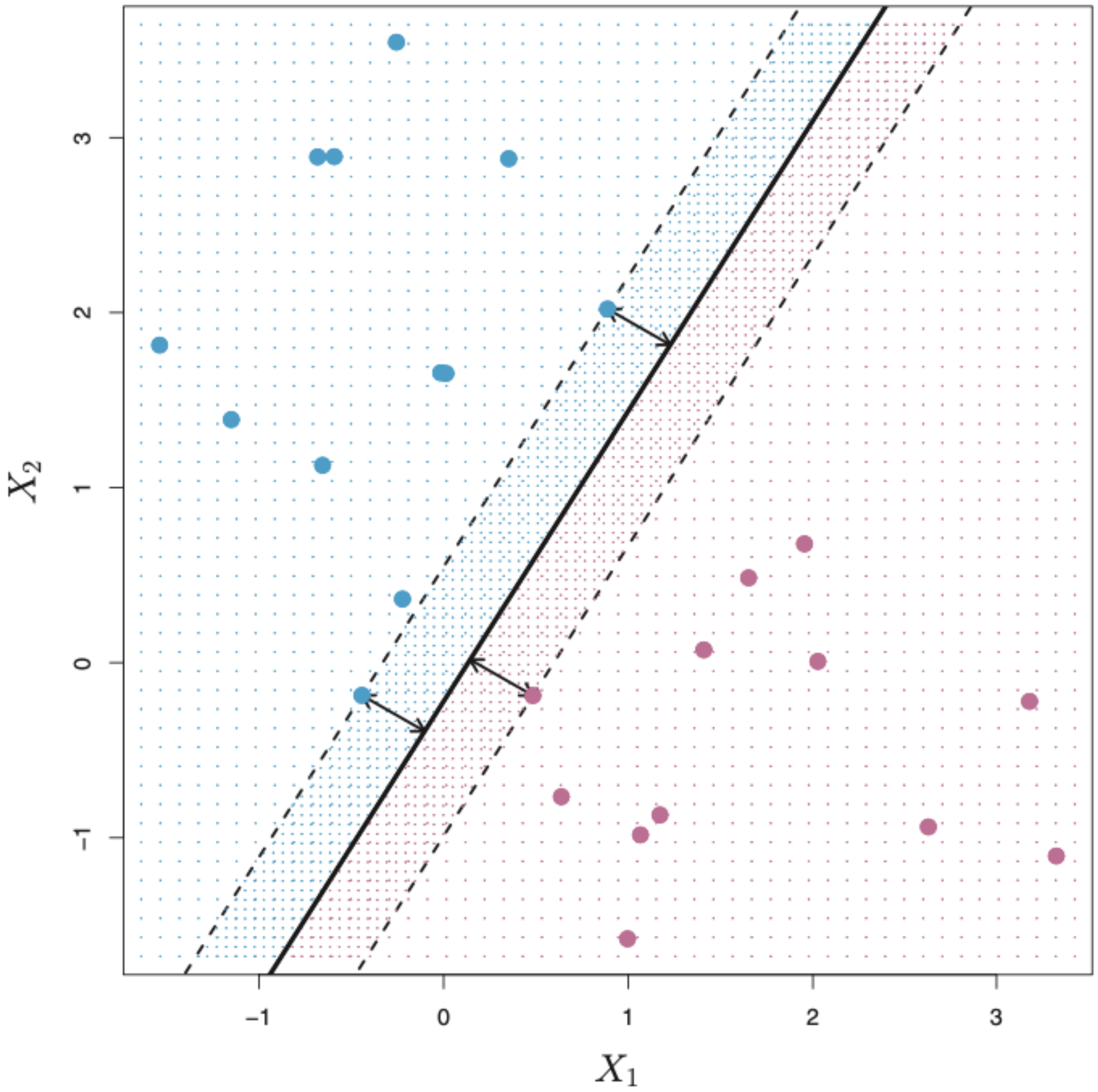


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
X_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X_1



X_2
X_1
−1
0
1
2
3
X_2
X_1

SV Classifier



Support Vector Machine

- Non-linear version of the Support Vector Classifier
- Extension using Kernels

Support Vector Machines

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

Support Vector Machines

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

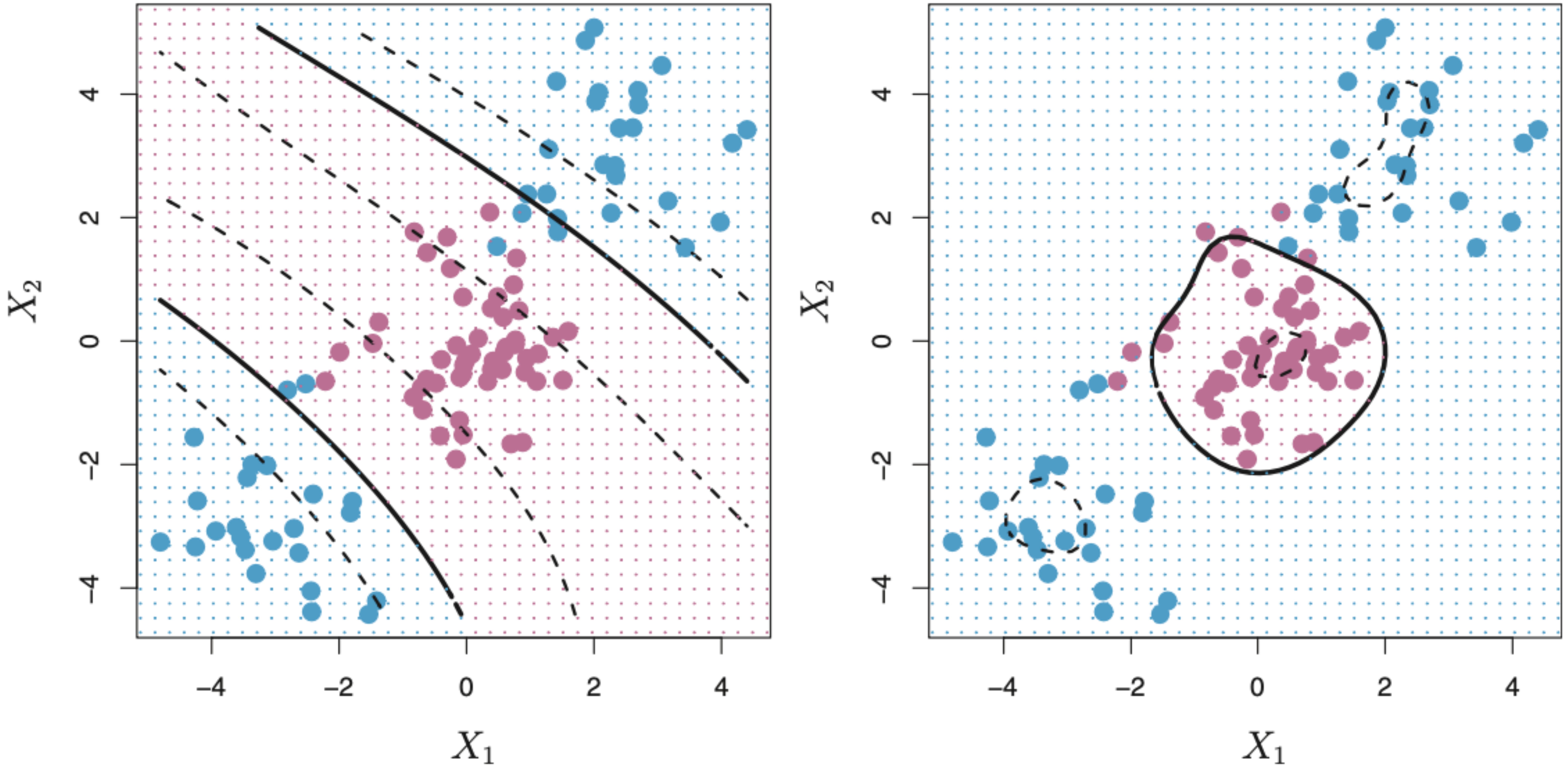
$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

Kernel function

Support Vector Machines

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d.$$

Polynomial Kernel

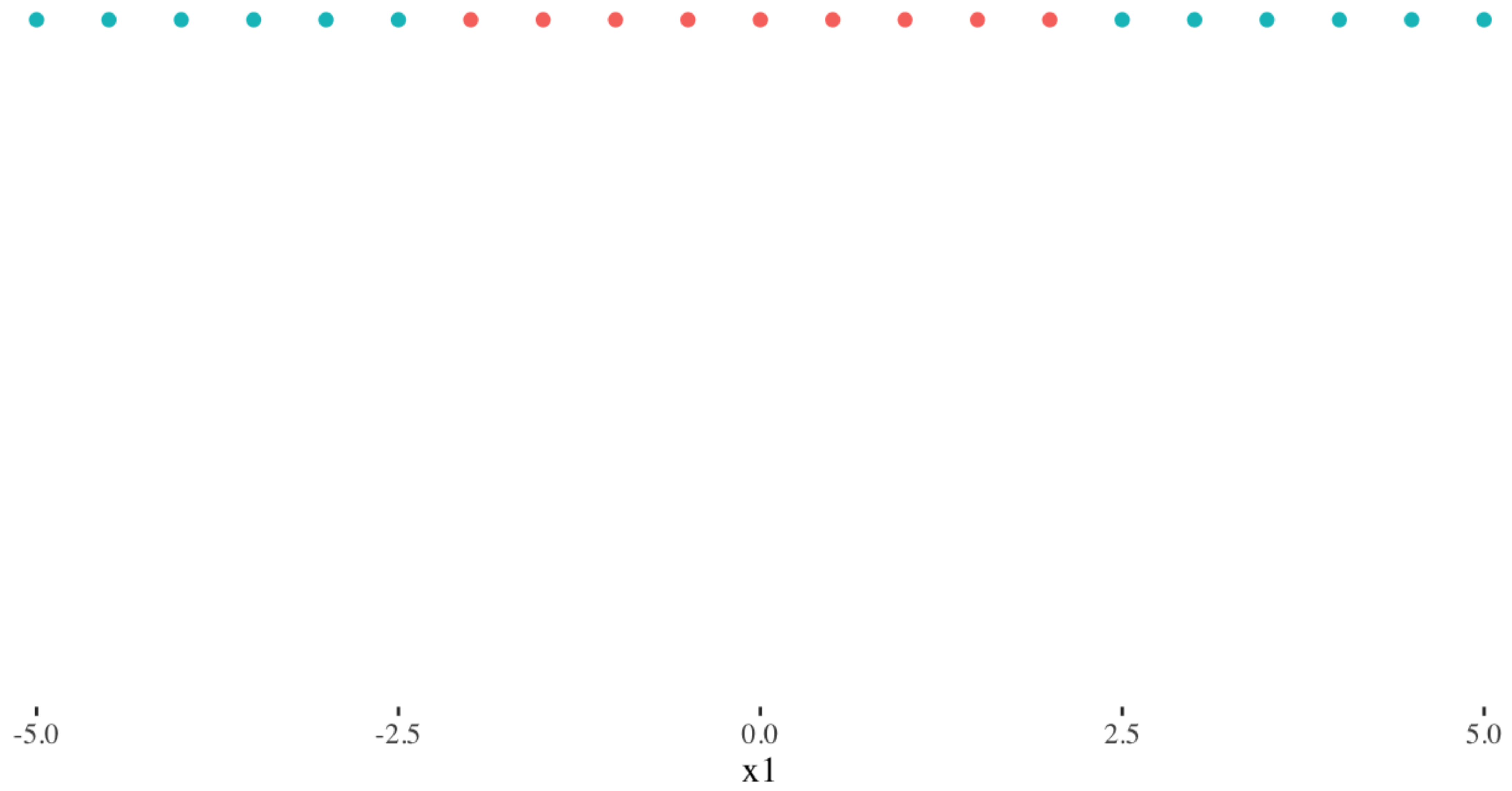


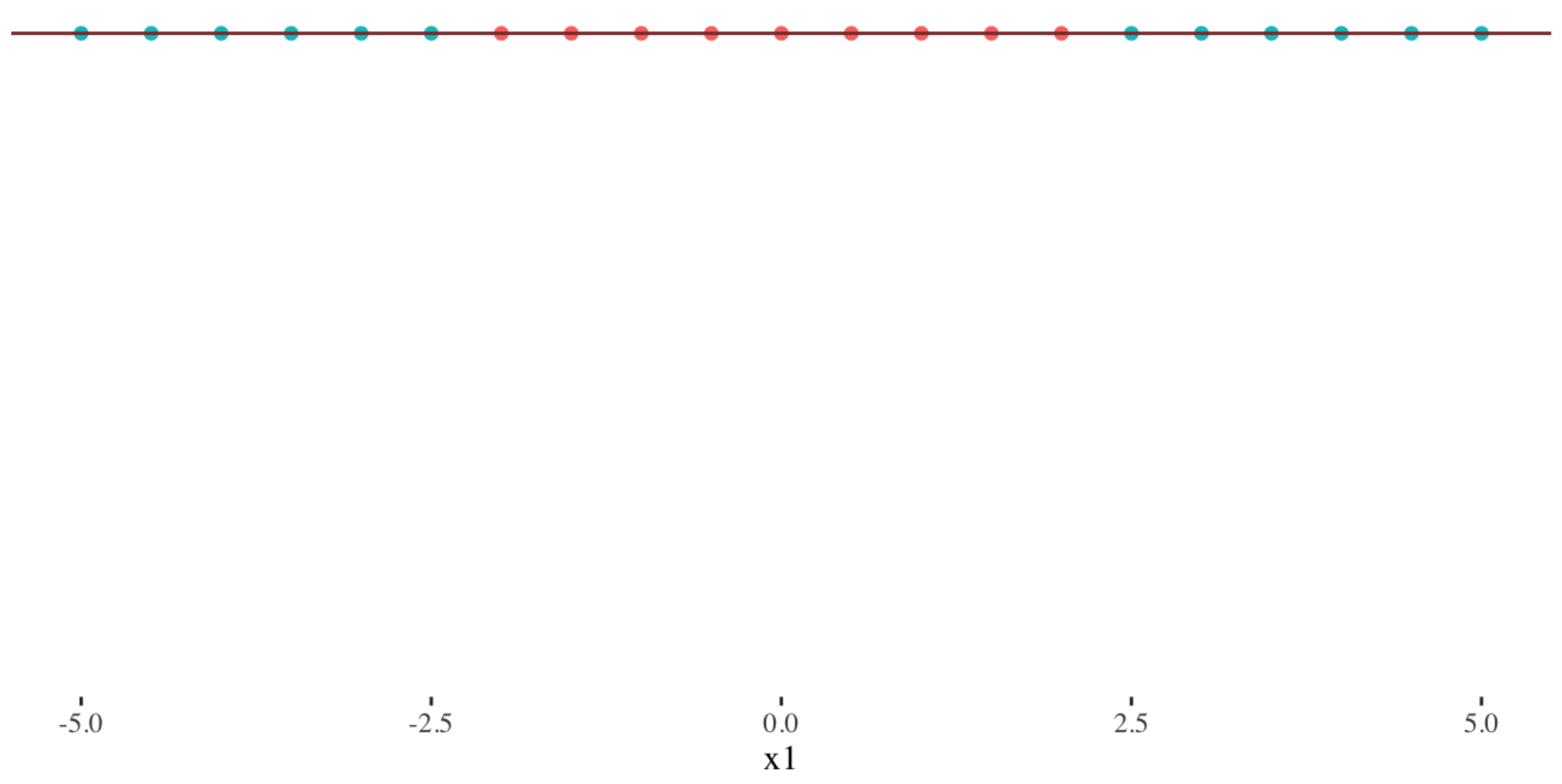
X_2
4
2
0
−2
−4
−4
−2
0
2
4
X_1
X_2
4
2
0
−2
−4
−4
−2
0
2
4
X_1

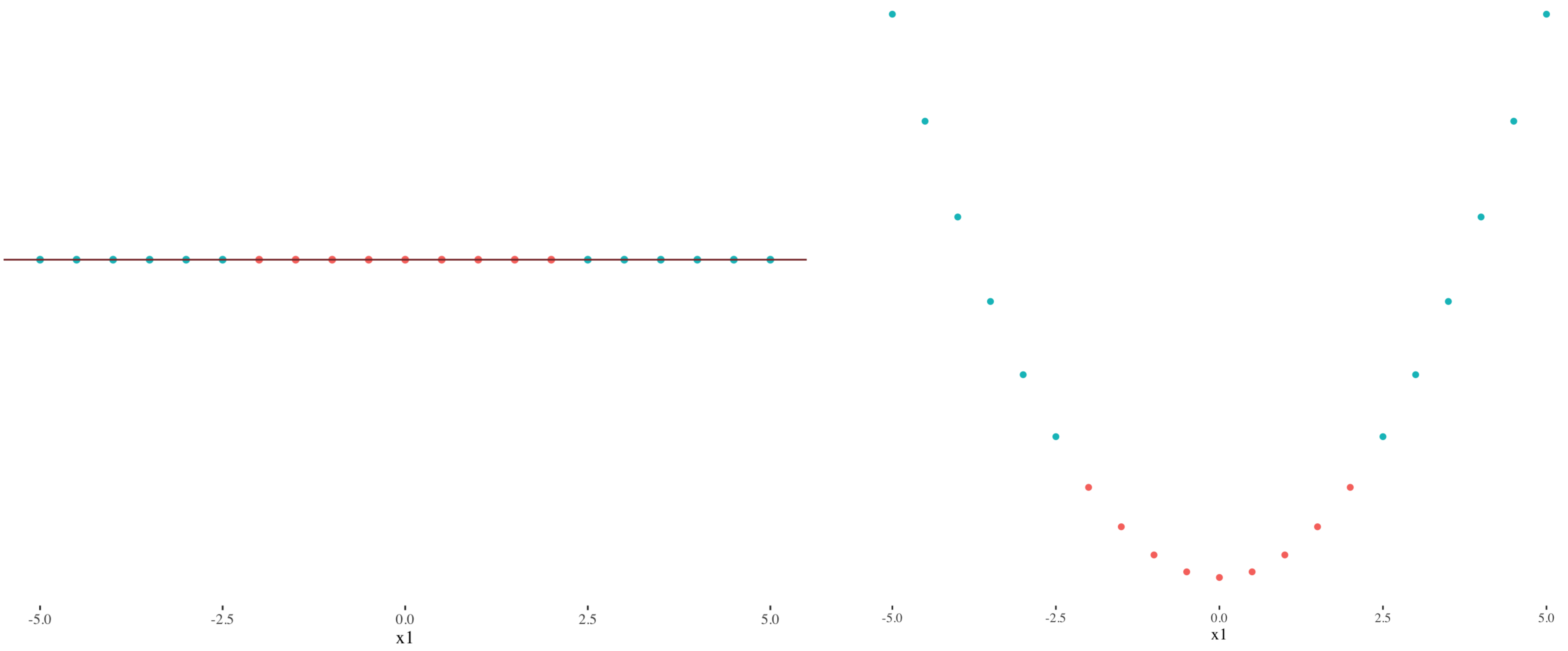
Kernel Trick

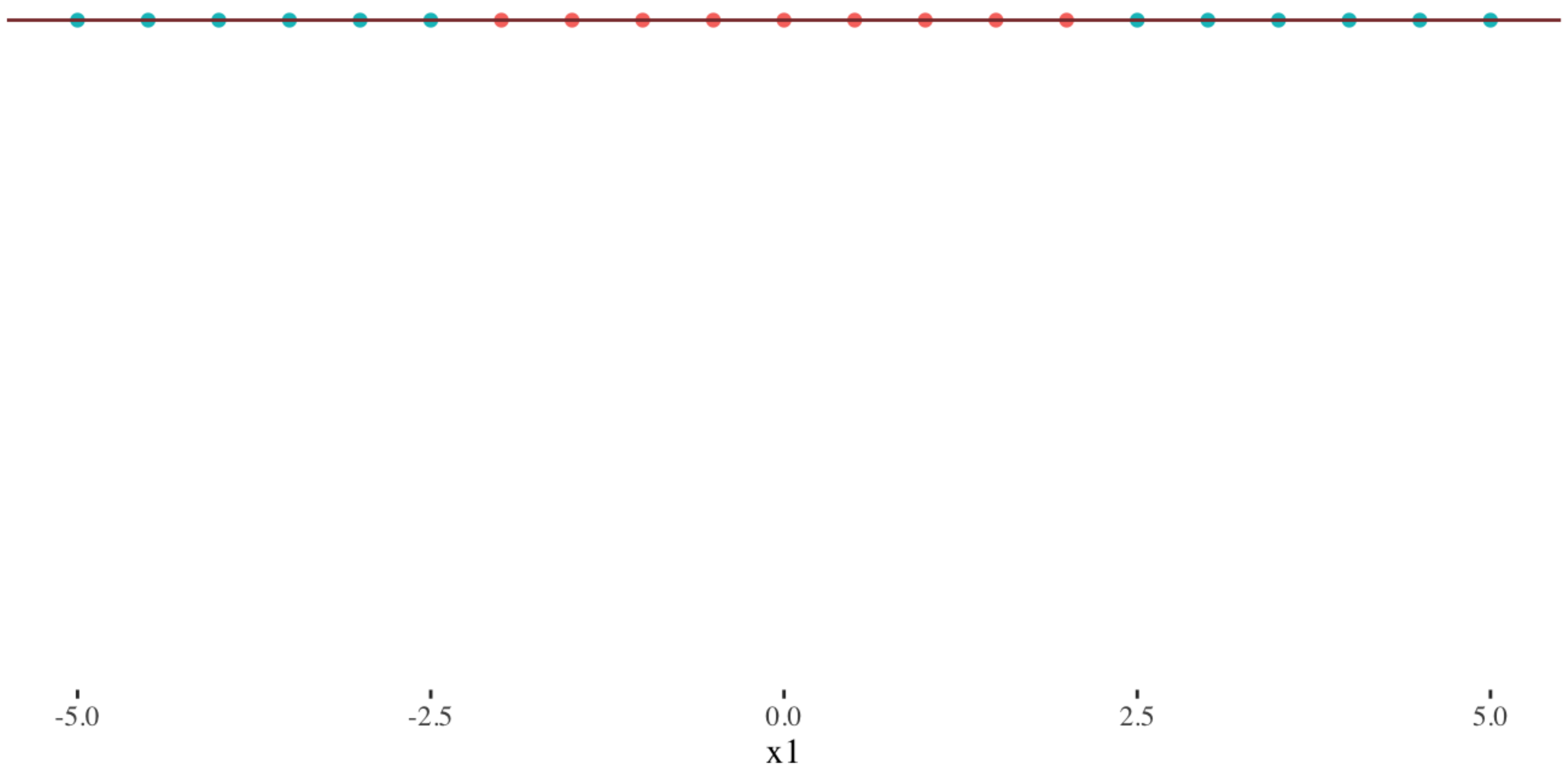
Kernel Trick

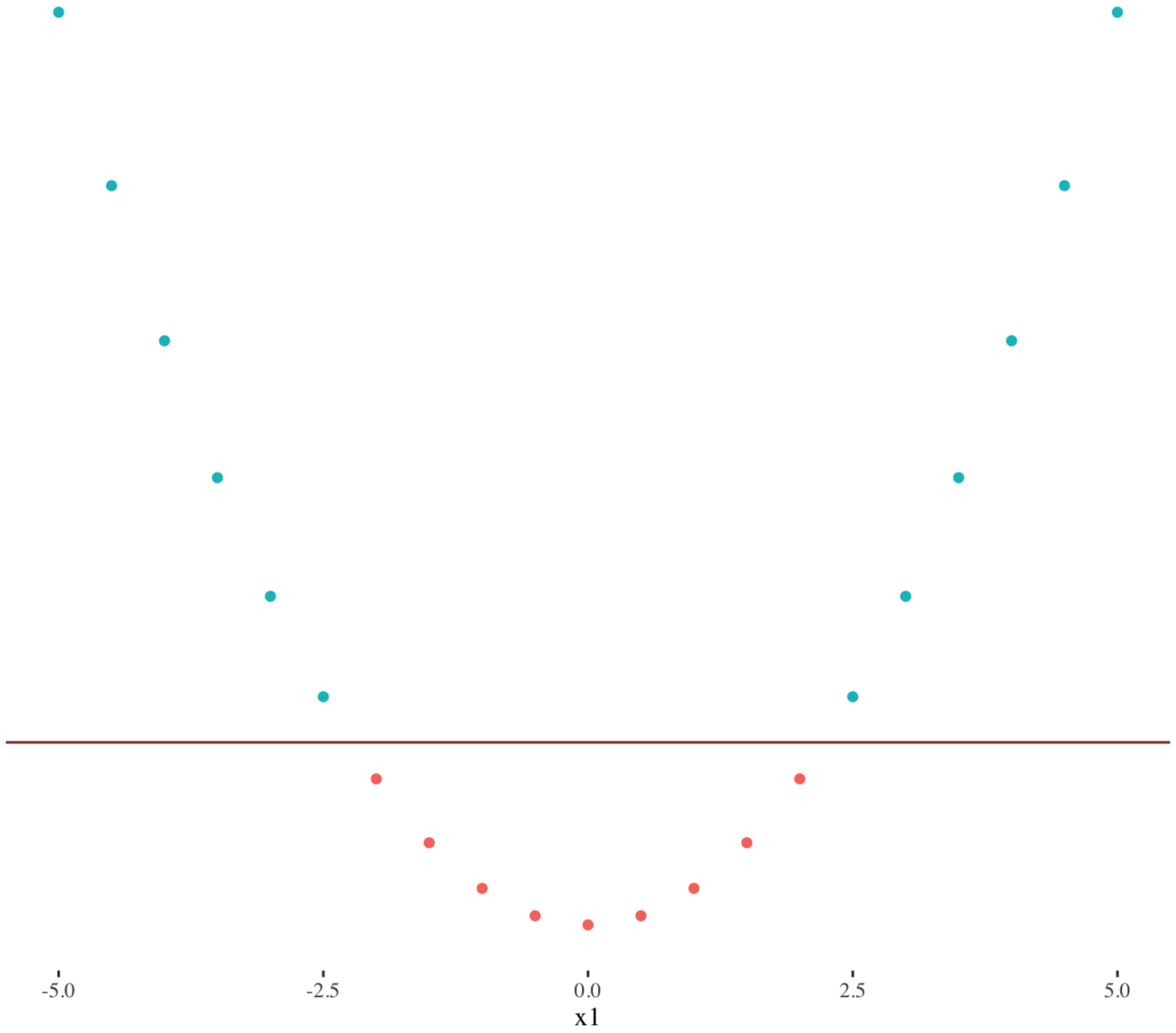
- Actual name
- Attempt to place n-dimensional data into n+1 dimensional space


-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1
-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1


-5.0
-2.5
0.0
2.5
5.0
x1

Unsupervised Learning

Machine Learning

- Supervised
    - An outcome variable is defined
    - Focus is on prediction
- **Unsupervised**
    - **No outcome variable has been defined**
    - **Focus is on patterns**

How to use the *supervised* methods?

- Easy
- At least conceptually
- ***Clear objective function***

How to use the supervised methods?

$$Y = (y_1, y_2, ..., y_n)$$
$$X = (x_1, x_2, ..., x_n)$$
$$(y_1, x_1), (y_2, x_2), ..., (y_n, x_n)$$

Task to predict $\hat{y}$ as close to y

How to use the supervised methods?

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$\hat{y} = \underset{\theta}{\operatorname{argmin}} \; E\left[L(model(\mathbf{x}, \theta), y)\right]$$

How to use *unsupervised* learning

- Objective function?

How to use *unsupervised* learning

- Objective function?
- Quantity of interest?

How to use *unsupervised* learning

- Objective function?
- Quantity of interest?
- Objective function = your quantity of interest


LIGHTYEAR

How to use *unsupervised* learning

- Objective function?
- Quantity of interest?
- Objective function = your quantity of interest
- ***This is difficult***

Measurement

- A collection of quantitative or numerical data that describes a property of an object or event

Measurement

- A collection of quantitative or numerical data that describes a property of an object or event
- What is the ***object***?

Measurement

In Social Science

- Operationalisation ➔ Data Collection ➔ Analyses ➔ Measurement
- The quantity of interest is ***extrinsic*** to the model

Measurement

In Computer Science

- Model Building ➔ Measurement of Performance
- The quantity of interest is ***intrinsic*** to the model

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Main quantity of
Interest for
computer science

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Means to an end
for social science

Main quantity of
Interest for
computer science

$$L(\theta, \hat{\theta}) = (\hat{\theta} - \theta)^2$$

Means to an end
for social science

Main quantity of
Interest for
computer science

$$\hat{\theta} \approx \theta$$

What social
science wants

Prediction vs. Inference

- Computer scientists often emphasise *prediction*
- Social scientists are often more interested in *inference*
- Vast, multidimensional parameter space = not suitable for inference
    - Good for prediction
- E.g., Turing test
    - Machine passes
    - *Why* does it pass/not pass

Problems

- Translation of Social Science concepts
- Connecting Methods to Theory
- Difficult to understand what is being *measured*

Unsupervised Learning Example

- Clustering algorithms are great tools
- Not well suitable for the “standard” social science paradigm
- Needs external validation, but there is no “best” method
- “Validation” based on “theory” or “expectation” leads to biases

Sticking to the paradigm

- The “normal” paradigm works only if we assume that there is one “correct” classification
- Need to adapt to different methods

Focus is on *Discovery*

Objectives

- Descriptive analysis/Discriminating words:
    - What are the characteristics of a corpus? How do some documents compare to each other
    - Collocation analysis, readability scores, Cosine/Jaccard similarity
- Clustering and scaling:
    - What groups of documents are in the corpus? Can the documents be placed on a dimension?
    - Cluster analysis, principal component analysis, wordfish..
- Topic modeling:
    - What are the main themes in a corpus?
    - LDA, STM

K-Means Clustering

- Simple(ish) algorithmic method
- Partitions the data into K non-overlapping clusters

Setup

$$C_1, C_2, ..., C_k$$

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

$$C_k \cap C_{k'} = \emptyset \text{ for all } k \neq k'$$

Assumption and task

- Optimal clustering solution is the one where ***within-cluster variation is as small as possible***

$$W(C_k)$$

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Within cluster variation

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Algorithm

- Randomly assign cluster numbers (1 through K) for each observation
    - Iterate until no further changes to the cluster assignment:
    - For each cluster determine the centroid (average of all observations in the cluster)
    - Re-assign observations to a cluster with the closest centroid (calculated with a distance metric).

Guarantees convergence at a local optimum

- Cannot guarantee the best solution
- But are rather good one
- Sensitive to random assignment at the start

Cluster Algorithms Validation

- Data assumptions (think data generation)
- Internal validity (best results for the data)
- External validity (matches with pre-existing understanding of data)
- Cross-validity (similar results across similar datasets)
- You are the validation method

Topic Modelling

Topic Modelling

- Covered it yesterday

Text scaling methods

- Attempts to fit documents into a unidimensional space
- Documents are “scaled” based on the frequency of used terms
- Assume “discriminating” words have a Poisson distribution

- “Ideological” successor to log odds ratio we’ve seen

Wordfish (Slapin and Proksch 2008)

$$w_{ik} \sim \mathrm{Poisson}(\lambda_{ik})$$

$$\lambda_{ik} = exp(\alpha_i + \psi_k + \beta_k \times \theta_i)$$

α_i Text size from type i

ψ_k Frequency of word k

β_k Discrimination power of word k

θ_i Ideological position of type i

Wordfish (Slapin and Proksch 2008)

$$w_{ik} \sim \text{Poisson}(\lambda_{ik})$$

$$\lambda_{ik} = exp(\alpha_i + \psi_k + \beta_k \times \theta_i)$$

α_i Text size from type i

ψ_k Frequency of word k

β_k Discrimination power of word k

θ_i Ideological position of type i

Wordfish (Slapin and Proksch 2008)

$$w_{ik} \sim \text{Poisson}(\lambda_{ik})$$

$$\lambda_{ik} = exp(\alpha_i + \psi_k + \beta_k \times \theta_i)$$

α_i Text size from type i

ψ_k Frequency of word k

β_k Discrimination power of word k

θ_i Ideological position of type i