Advanced quantitative text analysis (2023W)

Petro Tolochko, Fabienne Lind

Day 3

Contents (smaller changes possible)

Day	Session 1	Session 2	Session 3
1	Intro	Text representation	Feature Engineering
2	Concepts	Data	Dictionaries
3	Supervised machine learning	Unsupervised machine learning	Multilingual text analysis
4	Neural network models, transformers 1	Neural network models, transformers 2	Project talks

Supervised classification

Day 3 Session 1

Types of machine learning

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

Types of machine learning

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

Supervised machine learning

Objectives:

- Classification (for categorical variables)
  - E.g.: classify documents into pre defined classes
- Regression (for continuous variables)

Steps

1. Create a labeled data set
2. Classify documents with supervised learning algorithm
3. Check performance
4. Use the measures

Grimmer et al., 2022

1. Create a labeled data set

How:

- Human coders annotate parts of the corpus (see also slides in session on dictionaries)
- 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

Grimmer et al., 2022

1. Create a labeled data set

Number of documents

- the higher the number of classes per category 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)

Table 2. Simulation input parameters.

Factors	Input Parameters
N of human coders	2 (minimum), 5 (intermediate), & 10 (large manual coding)
Intercoder reliability	0.5 (low), 0.7 (acceptable), & 0.9 (high levels of reliability)
N of validation data	600 (0.5%), 1300 (1%), 6500 (5%), & 13000 (10%) of total data
Sampling variability	Random sample vs. nonrandom (biased) subset for validation
Coding per entry	Sole coding vs. duplicated coding for each entry

Song et al., 2021, p. 555

Song, H., Tolochko, P., Eberl, J. M., Eisele, O., Greussing, E., Heidenreich, T., ... & Boomgaarden, H. G. (2020). In validations we trust? The impact of imperfect human annotations as a gold standard on the quality of validation of automated content analysis. *Political Communication*, *37*(4), 550-572.

Barberá, P., Boydstun, A. E., Linn, S., McMahon, R., & Nagler, J. (2021). Automated text classification of news articles: A practical guide. *Political Analysis*, *29*(1), 19-42.

1. Create a labeled data set

Split labeled data in training data, test data, validation set

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

- Only used to evaluate in the end, no further optimization allowed

2. Classify documents with supervised learning

Classifier learns the mapping between features and the class labels in the training set

- We 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 related to the classes of the training documents)
- We fit the model

Grimmer et al., 2022

2. 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

Grimmer et al., 2022

3. Check 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:

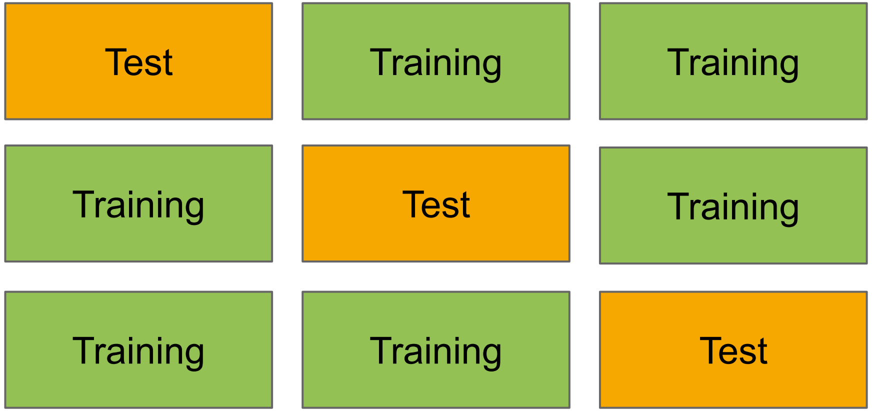
- Danger of overfitting (focus on features that work well with training set but do not generalize)
  - Solutions: cross-validation
- Performance metric (i.e., recall, precision, F1, Accuracy)

Grimmer et al., 2022

3. Check 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



3. Check performance

Performance metrics

Confusion matrix

	Actual label	
Classification (algorithm)	Negative	Positive
Negative	True negative	False negative
Positive	False negative	True positive

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

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

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

Grimmer et al., 2022

3. Check performance

Check convergent validity (Adcock & Collier, 2001)

- Report recall, precision, F1 for the held-out validation data
- Compare the measures with other established measures, e.g.,
    - Use trained model to classify texts (open-ended answers relationship uncertainty as described by participants of an online survey) and compare it with the self-assessment in response to a closed question, see Pilny et al. 2019

Adcock, R., & Collier, D. (2001). Measurement validity: A shared standard for qualitative and quantitative research. *American political science review*, *95*(3), 529-546.

Pilny, A., McAninch, K., Slone, A., & Moore, K. (2019). Using supervised machine learning in automated content analysis: An example using relational uncertainty. *Communication Methods and Measures*, *13*(4), 287-304.

4. Using the measures

Remember that we labeled only parts of our original corpus.

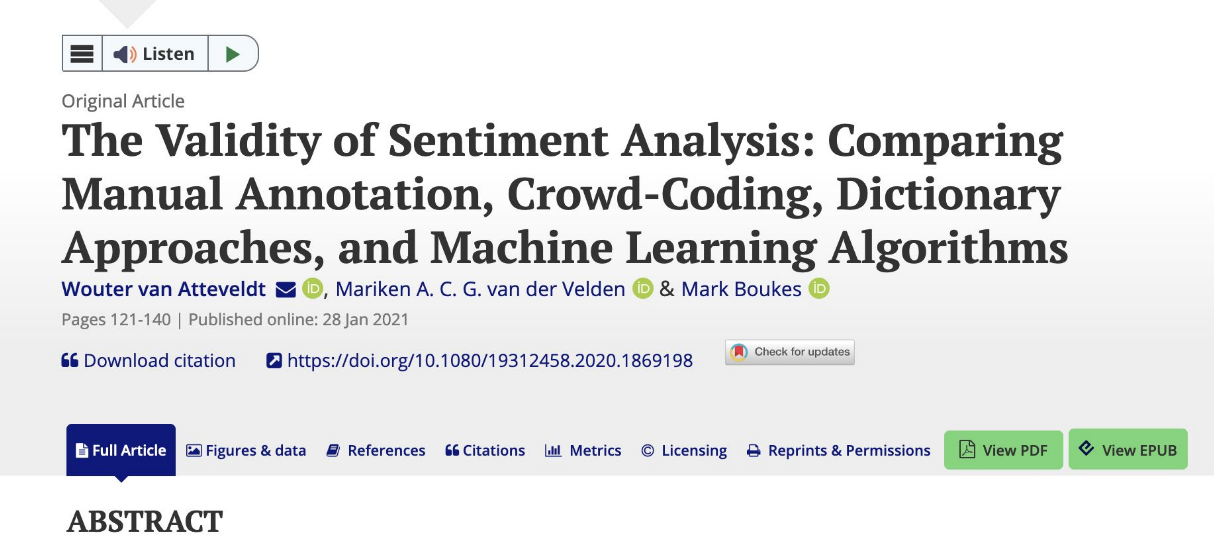
When you are satisfied with the performance of the classifier, you can use it to classify all documents in the corpus

Grimmer et al., 2022

Dictionary vs. supervised machine learning

Category: sentiment

Result: machine learning significantly outperformed dictionary coding



Listen

Original Article

The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms

Wouter van Atteveldt, Mariken A. C. G. van der Velden & Mark Boukes

Pages 121-140 | Published online: 28 Jan 2021

Download citation https://doi.org/10.1080/19312458.2020.1869198 Check for updates

Full Article | Figures & data | References | Citations | Metrics | Licensing | Reprints & Permissions | View PDF | View EPUB

ABSTRACT

Van Atteveldt el al. (2021)

Dictionary vs. supervised machine learning

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

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

González-Bailón, S., & Paltoglou, G. (2015). Signals of public opinion in online communication: A comparison of methods and data sources. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 95-107.

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

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484.

Naïve Bayes Classifier

- Probabilistic classifier
- Simple
- Fast
- Good Accuracy

Bayes Theorem

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

Bayes Theorem

Conditional probability

Independent probability

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

Conditional probability

Independent probability

Bayes Theorem

Likelihood

Prior probability

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

Posterior probability

Probability of Data
(Marginal probability)

Bayes Theorem

Likelihood

Prior probability

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

Posterior probability

Probability of Data
(Marginal probability)
(Evidence)

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

Class

Features

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

universität wien

Naïve Bayes Classifier

Class

Features

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

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

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

universität wien

Decision Rule

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

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:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

universität wien

Hyperplane

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

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

Hyperplane

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

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

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

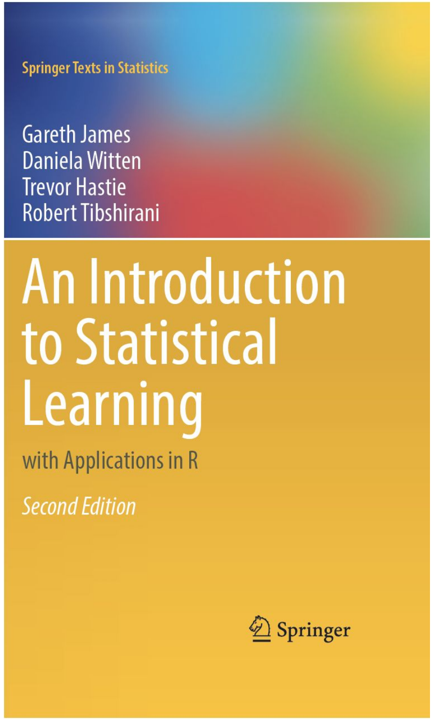
universität wien

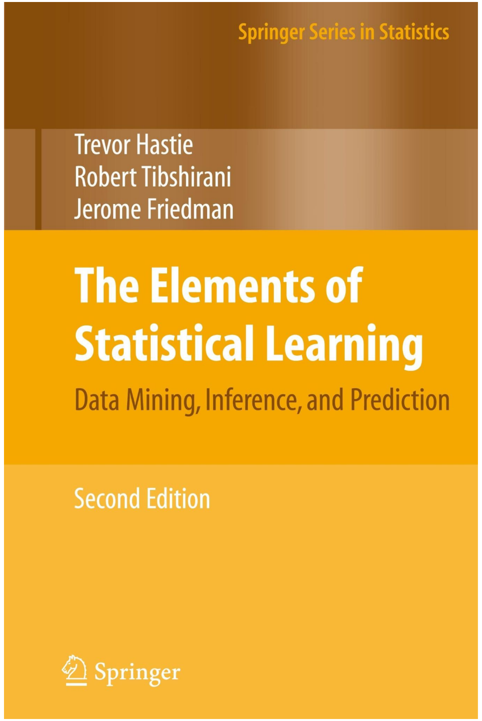
Classification

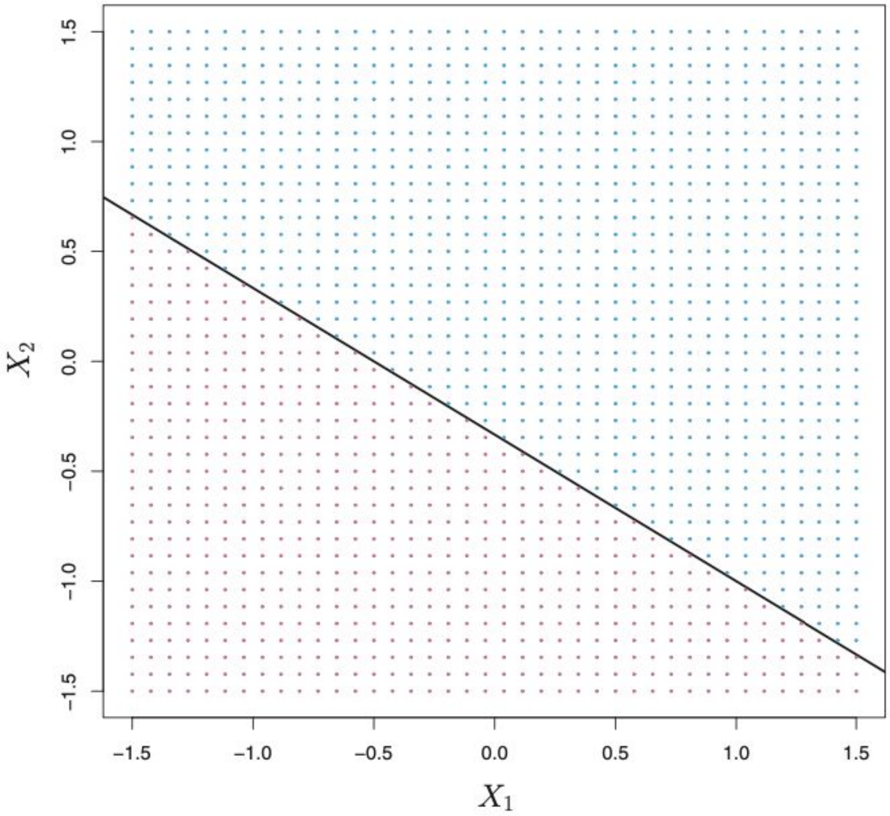
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$$

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

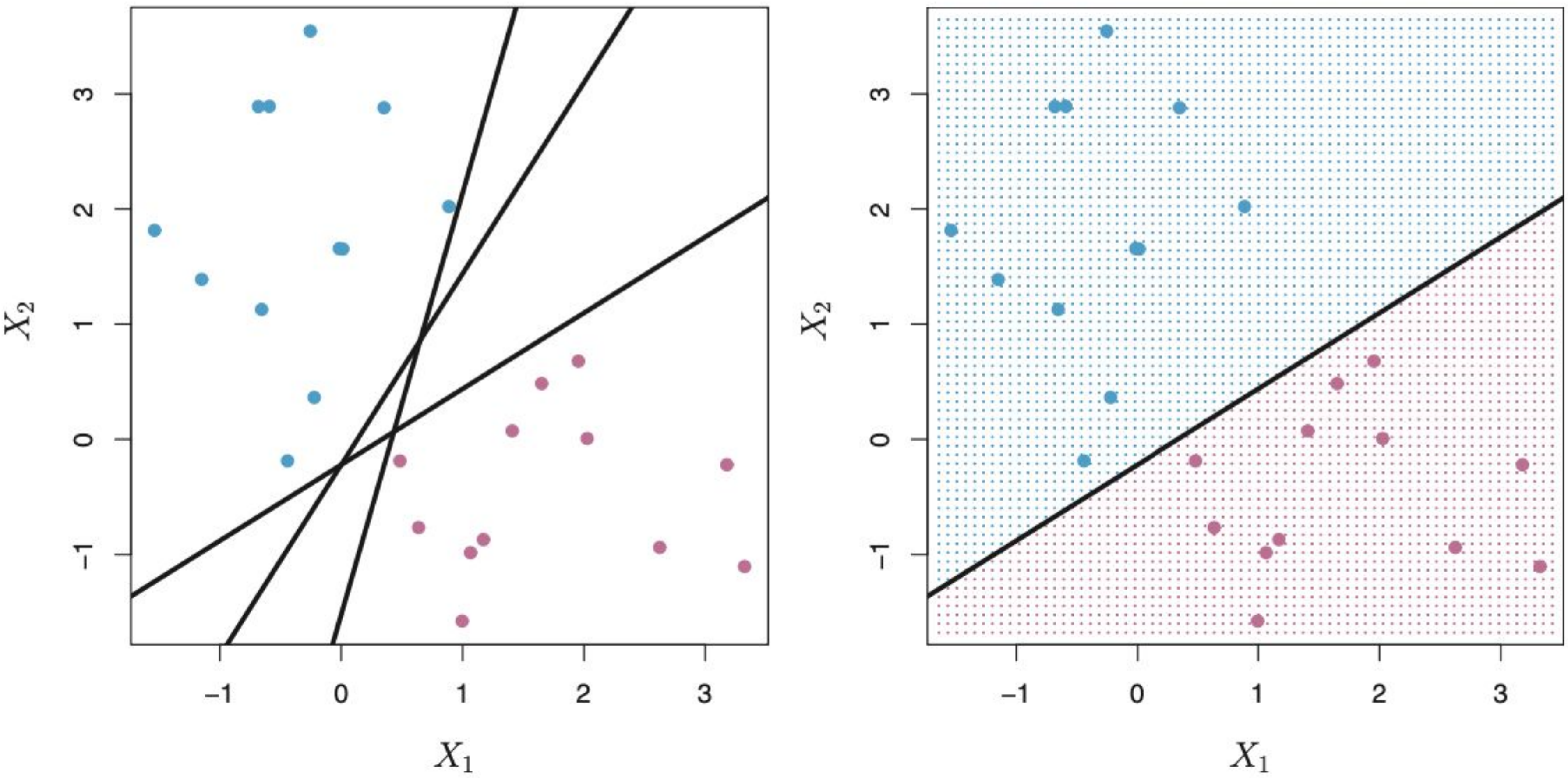
Following images from:





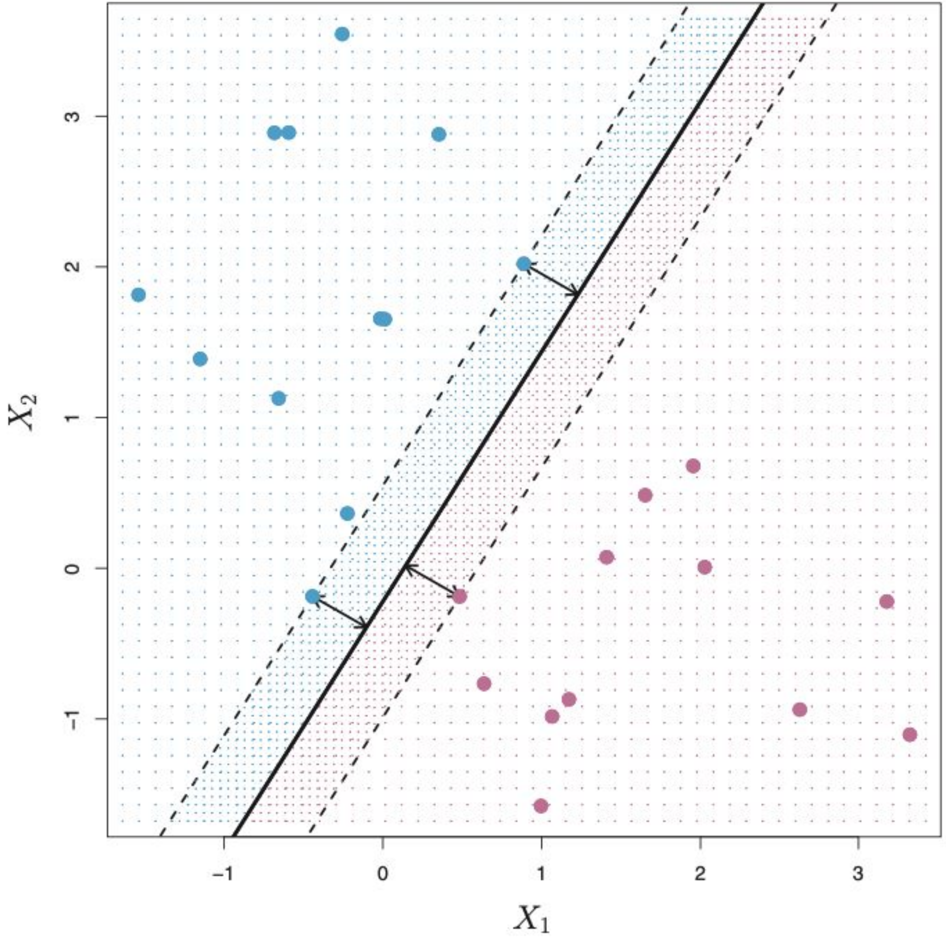


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
X_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X_1



3
2
1
0
−1
X_2
−1
0
1
2
3
X_1
3
2
1
0
−1
X_2
−1
0
1
2
3
X_1

SV Classifier



Support Vector Machine

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

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

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

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

Kernel function

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

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

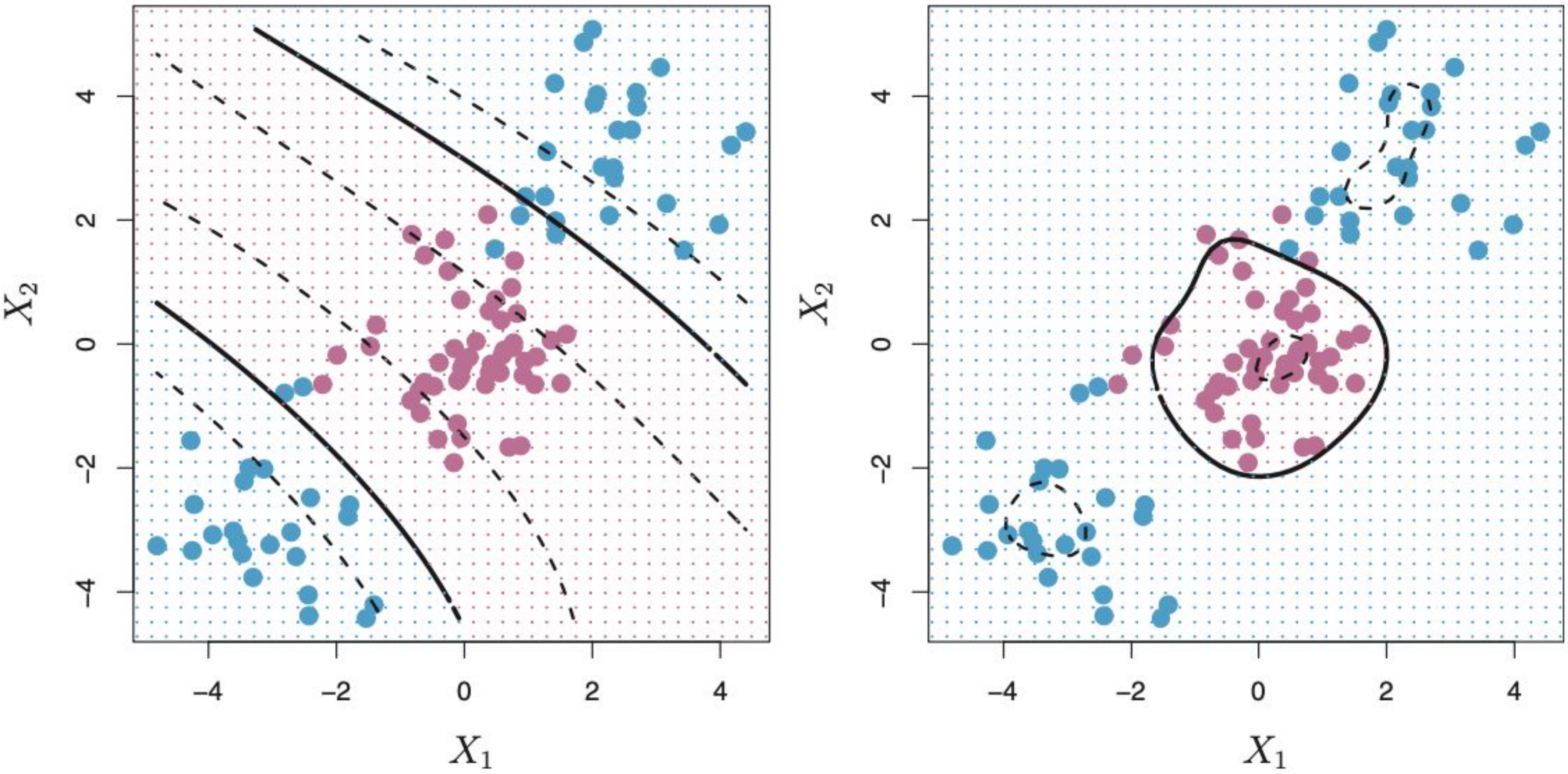
Kernel function

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

Polynomial Kernel
Non-linear

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

universität wien



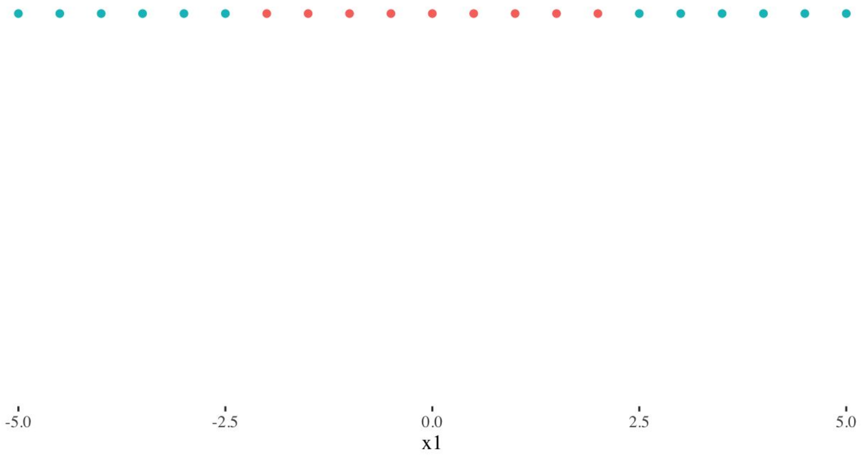
universität wien

Kernel Trick

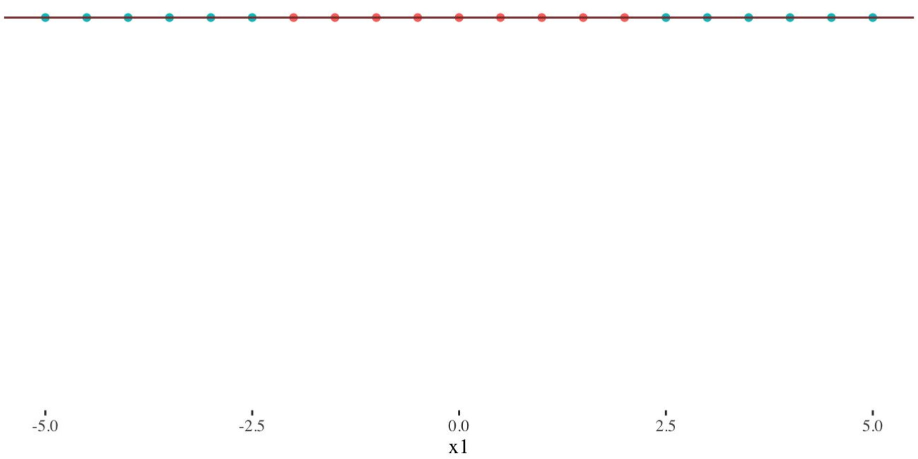
- ***Actual name***

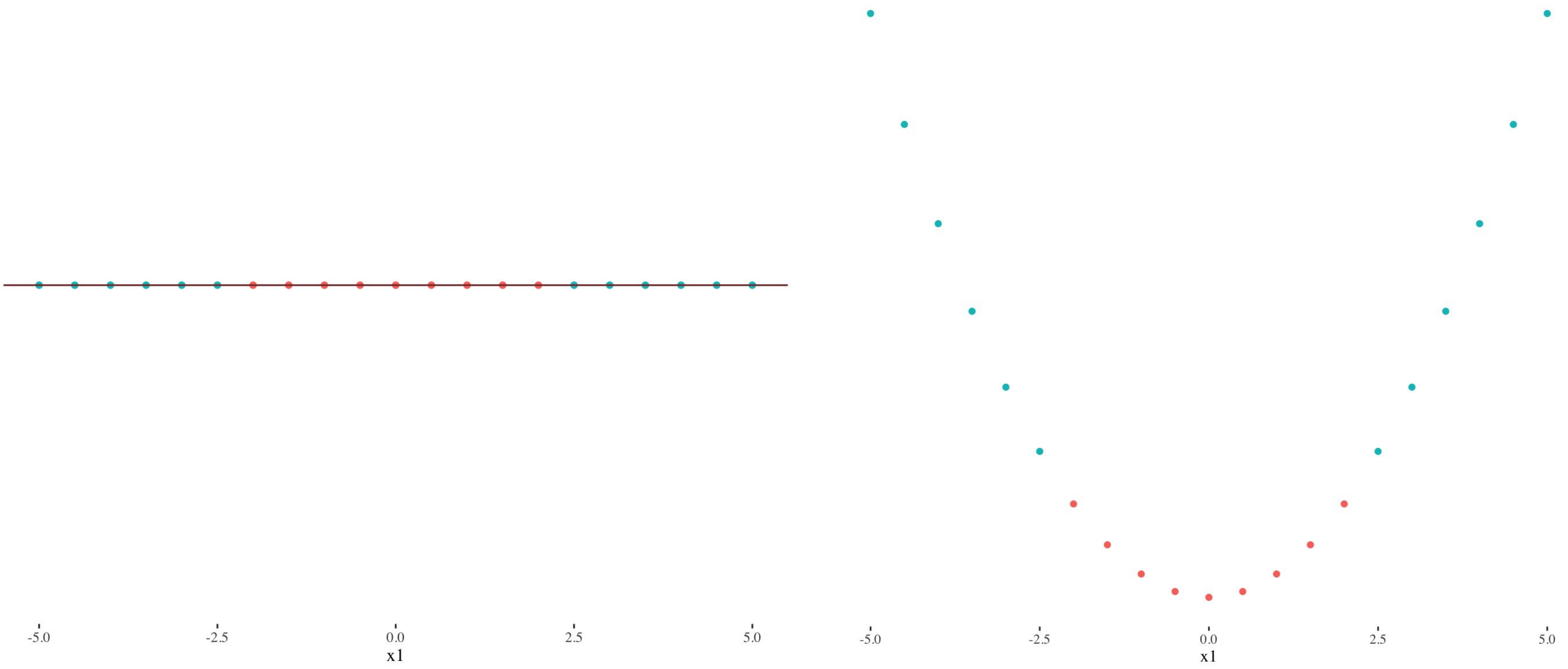
Kernel Trick

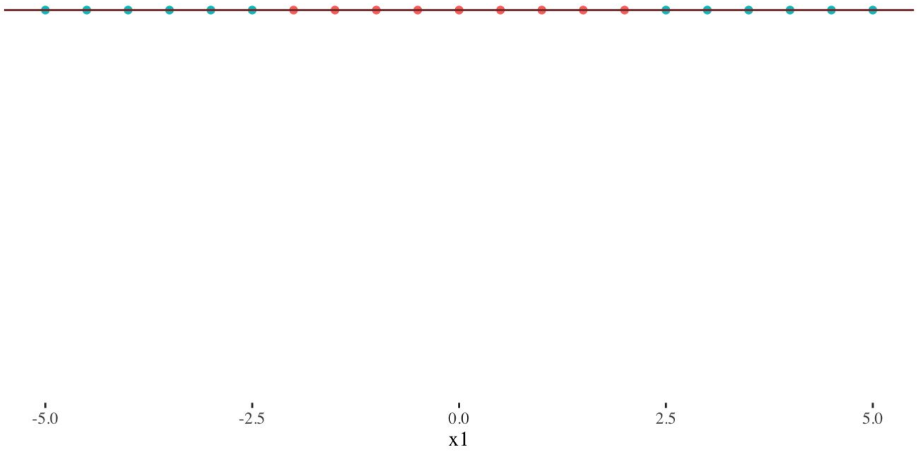
- ***Actual name***
- Attempt to place n-dimensional data into n+1 dimensional space



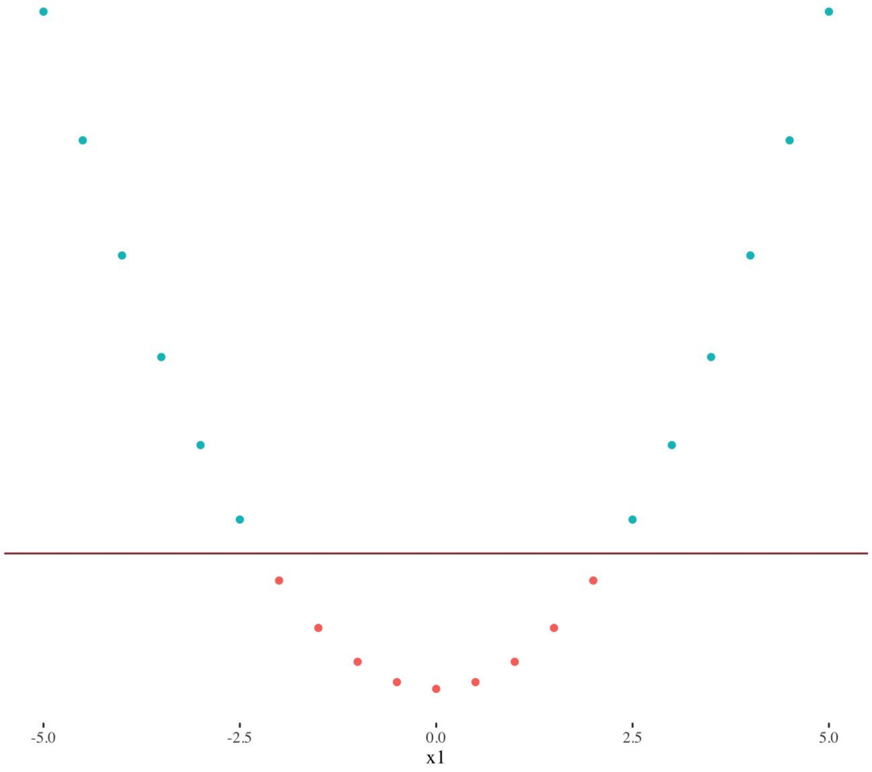
-5.0
-2.5
0.0
2.5
5.0
x1


-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

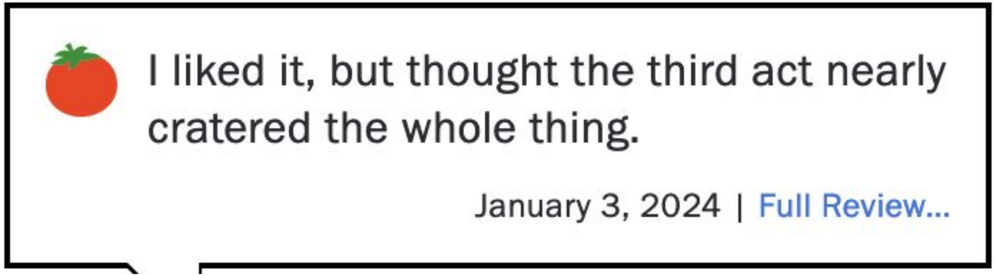
Performing supervised machine learning in R and Python

R: quanteda, caret, e1071, klaR, C5.0, OneR

Python: scikit-learn

Coding session

- Sentiment in movie reviews



universität wien

universität wien

Coding session

- Sentiment in movie reviews

I liked it, but thought the third act nearly cratered the whole thing.

January 3, 2024 | Full Review...

An intense, inventively filmed, and well-acted biopicture about the kinds of events that are hard on the heart.

Full Review | Original Score: 5/5 | Nov 20, 2023

In a movie about impending global catastrophe, he gives a close-up of a face, and a twitch of a lip the power of an atom bomb.

Full Review | Original Score: A | Nov 17, 2023

I liked it, but thought the third act nearly cratered the whole thing.

January 3, 2024 | Full Review...

An intense, inventively filmed, and well-acted biopicture about the kinds of events that are hard on the heart.

Full Review | Original Score: 5/5 | Nov 20, 2023

In a movie about impending global catastrophe, he gives a close-up of a face, and a twitch of a lip the power of an atom bomb.

Full Review | Original Score: A | Nov 17, 2023

Unsupervised classification

Day 3 Session 2

How would you describe what you see here?

- coffee
- cafe
- espresso
- latte
- barista
- beans
- brew
- cappuccino
- aroma
- roast

- programming
- code
- software
- development
- algorithm
- python
- function
- variable
- debugging
- Java

- fitness
- exercise
- health
- workout
- gym
- nutrition
- weight
- strength
- cardio
- yoga

Topic 1: Label?

- coffee
- cafe
- espresso
- latte
- barista
- beans
- brew
- cappuccino
- aroma
- roast

Topic 2: Label?

- programming
- code
- software
- development
- algorithm
- python
- function
- variable
- debugging
- Java

Topic 3: Label?

- fitness
- exercise
- health
- workout
- gym
- nutrition
- weight
- strength
- cardio
- yoga

Types of machine learning

1. Supervised
   - An outcome variable is defined
   - Focus is on prediction
2. 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?

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

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 = ***your*** quantity of interest


LIGHTYEAR

How to use unsupervised learning

- 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 the social sciences)

- 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

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

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

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

The paradigm

- Approximating a data-generating process

The paradigm

- Approximating a data-generating process
- Assumption: there is one (and only one) “true” data-generating process
- It ***is*** the reality

Sticking to the paradigm

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

Sticking to the paradigm

- The “normal” paradigm works only if we assume that there is one “correct” classification
- Need to adapt to different methods
- Unsupervised methods are ***meaningless*** in conjunction with the “true” data-generating process assumption

Focus is on Discovery

Grimmer et al. 2022

What can you do with unsupervised techniques?

Descriptive analysis/Discriminating words:

- What are the **characteristics** of a corpus? How do some documents compare to each other
- Keyness, 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?
- Latent semantic scaling, Cluster analysis, principal component analysis, wordfish..

Topic modeling:

- What are the main **themes** in a corpus?
- LDA, STM, BERTopic

Grimmer et al. 2022, Barberá, 2018

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

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

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Algorithm

1. Randomly assign cluster numbers (1 through K) for each observation
2. Iterate until no further changes to the cluster assignment:
   a. For each cluster determine the ***centroid*** (average of all observations in the cluster)
   b. 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

- A model to discover latent topics
- ***Not*** synonymous with LDA
- LDA is one of topic models

- Latent semantic analysis
- Singular value decomposition
- Even clustering methods (like the one we just discussed)

Latent Dirichlet Allocation

- Bayesian generative hierarchical model
- First introduced as a way to simultaneously model traits and genes (Pritchard, 2000)
- Adjusted for text analysis ML applications (Blei et al., 2003)

Latent Dirichlet Allocation

- Estimates a distribution of ***words*** across ***documents*** across ***latent topics***

Latent Dirichlet Allocation

- By modeling distributions of topics over words and words over documents, topic models identify the most discriminatory groups of documents automatically.

Assumption: if a document is about a certain topic, one would expect words, that are related to that topic, to appear in the document more often than in documents that deal with other topics.

Latent Dirichlet Allocation

- **Initialization:** LDA starts by randomly assigning each word in each document to one of the topics. These initial assignments serve as a starting point for the algorithm.
- **Iteration:** LDA iteratively updates the topic assignments of words and estimates the topic distributions. In each iteration, it goes through each word in each document and reassigns it to a topic based on the current topic distributions and the observed word frequencies.
- **Inference:** As the iterations progress, LDA gradually infers the most likely topic assignments for each word and the overall topic distributions across the document collection. This inference process continues until the algorithm converges or reaches a predefined stopping criterion.

van Atteveld et al., 2022

How to: Steps

1. Data selection
2. Data preprocessing (what features are good topic representations?)
3. Applying LDA
4. Once the iterations are complete, LDA provides 2 outputs:
    a. the estimated topic distributions for each document
    b. the word distributions for each topic

   These distributions give insights into the main topics present in the document collection
5. Validation: Inspecting top words per topic and the top topics per document; and many more methods

Mixture Models

$$P(x) = \Sigma_{k=1}^{K} \pi_k \times P(x \mid \theta_k)$$

- P(x) represents the probability density function (PDF) of the observed data point x
- K is the number of components in the mixture model
- π_k is the mixing proportion or weight assigned to the k-th component, representing the probability of a data point belonging to that component. These weights satisfy the condition sum(π_k) = 1 and 0 <= π_k <= 1
- P(x|θ_k) represents the conditional probability of observing data point x given the parameters θ_k of the k-th component distribution

Hierarchical Models

Hierarchical Models

$$y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \epsilon$$

Hierarchical Models

$$y \sim Normal(\mu, \sigma)$$

$$\mu = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

$$\beta \sim Normal(0,5)$$

$$\sigma \sim Exponential(1)$$

Hierarchical Models

Likelihood Function

$$y \sim Normal(\mu, \sigma)$$

$$\mu = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

$$\beta \sim Normal(0,5)$$

$$\sigma \sim Exponential(1)$$

Prior distribution of the parameters

Hierarchical Models

$$y \sim Normal(\mu, \sigma)$$

$$\mu = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$$

$$\beta \sim Normal(0,5)$$

$$\sigma \sim Exponential(1)$$

Hyperparameters

LDA

$$\theta_k \sim Dirichlet(\alpha)$$ for each topic k

$$\eta_d \sim Dirichlet(\beta)$$ for each document d

$$z_{d,w} \sim Multinomial(\eta_d)$$ for each word w in document d

$$x_{d,w} \sim Multinomial(\theta_{z_{d,w}})$$ for each word w in document d

LDA

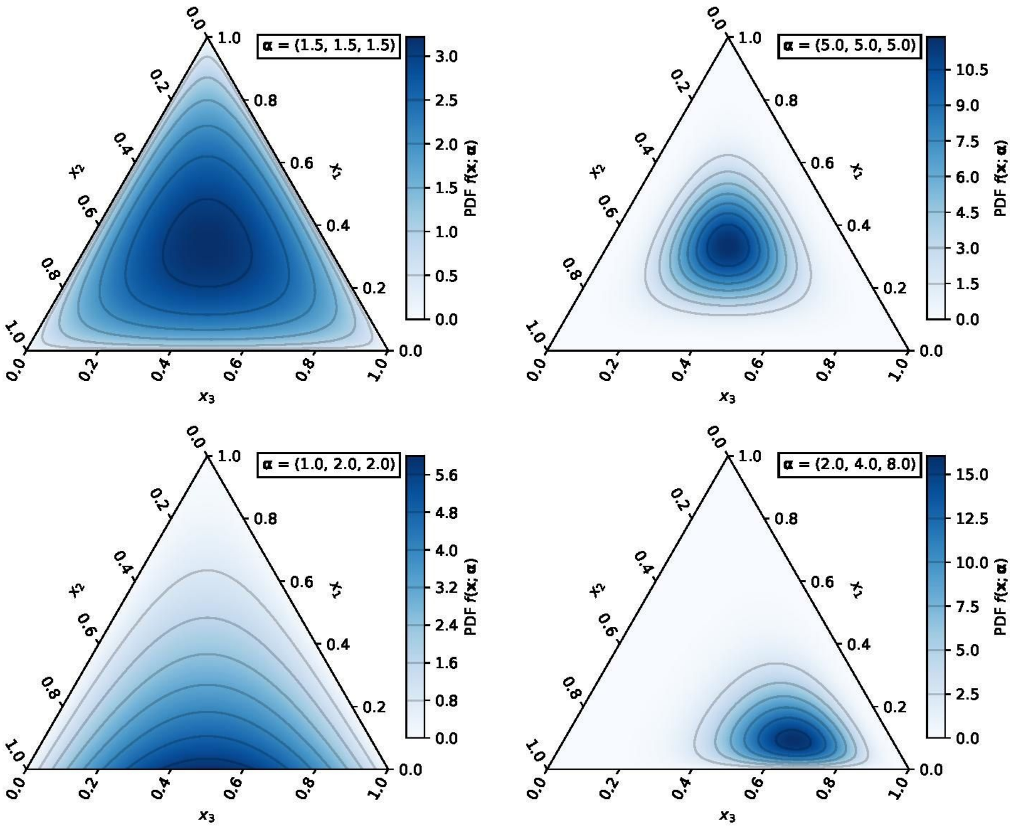
- θ_k is the topic-word distribution for topic k, representing the probability of each word given the topic.
- η_d is the document-topic distribution for document d, representing the probability of each topic in the document.
- z_d,w is the topic assignment for word w in document d, indicating which topic generated the word.
- x_d,w is the observed word in document d.

LDA

- For each topic $k \in \{1, ..., K\}$:
  - Draw a distribution over words $\theta\_k \sim \text{Dirichlet}(\alpha)$, where α is a hyperparameter representing the topic-word prior.
  - For each document $d \in \{1, ..., D\}$:
  - Draw a distribution over topics $\eta\_d \sim \text{Dirichlet}(\beta)$, where β is a hyperparameter representing the document-topic prior.
- For each word w in document d:
  - Draw a topic assignment $z\_d,w \sim \text{Multinomial}(\eta\_d)$, indicating which topic generated the word.
  - Draw a word $x\_d,w \sim \text{Multinomial}(\theta\_\{z\_d,w\})$, indicating the specific word generated by the chosen topic.

LDA

- The goal of LDA is to infer the posterior distributions of the latent variables θ and η given the observed documents.
- Once the posterior distributions are estimated, LDA can be used to assign topics to new documents or extract the most probable words for each topic.



α = (1.5, 1.5, 1.5)
α = (5.0, 5.0, 5.0)
α = (1.0, 2.0, 2.0)
α = (2.0, 4.0, 8.0)
x_1
x_2
x_3
PDF $f(\mathbf{x}; \boldsymbol{\alpha})$

Tricky to estimate

$$\int_{\theta_j} P(\theta_j;\alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j)\, d\theta_j = \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^{i}+\alpha_i-1}\, d\theta_j$$

$$= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^{i} + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^{i} + \alpha_i\right)} \int_{\theta_j} \frac{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^{i} + \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^{i} + \alpha_i)} \prod_{i=1}^{K} \theta_{j,i}^{n_{j,(\cdot)}^{i}+\alpha_i-1}\, d\theta_j$$

$$= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^{i} + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^{i} + \alpha_i\right)}.$$

$$\int_{\boldsymbol{\varphi}} \prod_{i=1}^{K} P(\varphi_i;\beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}})\, d\boldsymbol{\varphi}$$

$$= \prod_{i=1}^{K} \int_{\varphi_i} P(\varphi_i;\beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}})\, d\varphi_i$$

$$= \prod_{i=1}^{K} \int_{\varphi_i} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \prod_{r=1}^{V} \varphi_{i,r}^{\beta_r - 1} \prod_{r=1}^{V} \varphi_{i,r}^{n^i_{(\cdot),r}}\, d\varphi_i$$

$$= \prod_{i=1}^{K} \int_{\varphi_i} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \prod_{r=1}^{V} \varphi_{i,r}^{n^i_{(\cdot),r} + \beta_r - 1}\, d\varphi_i$$

$$= \prod_{i=1}^{K} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \frac{\prod_{r=1}^{V} \Gamma(n^i_{(\cdot),r} + \beta_r)}{\Gamma\left(\sum_{r=1}^{V} n^i_{(\cdot),r} + \beta_r\right)}.$$

Need Markov Chain Monte Carlo Simulation

MCMC

https://chi-feng.github.io/mcmc-demo/app.html

Extension of LDA: Structural Topic Models (STM)

We often have information about the document

- Article metadata (year, source)
- Speaker data (year, party, gender)
- Respondent data for open questions

STM allows to use that data

- e.g. topic proportions change over time
- e.g. words di er between speakers, parties

.Roberts et al. (2014). Structural Topic Models for Open Ended Survey Responses. AJPS.

Other extensions of LDA

Dynamic Topic Models (Blei & Lafferty, 2006)

- Analyzing the evolution of topics over time

Polylingual Topic Model (Mimno et al., 2009)

- Deriving at shared topics across multiple languages

Mimno, D., Wallach, H., Naradowsky, J., Smith, D. A., & McCallum, A. (2009, August). Polylingual topic models. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (pp. 880-889).

Blei, D. M., & Lafferty, J. D. (2006). *Dynamic topic models.* In Cohen W. & Moore, A. (Program Chairs), *Proceedings of the 23rd International Conference on Machine Learning* (pp. 113–120). Association for Computing Machinery, Pittsburgh, Pennsylvania, USA.

Topic modeling with language embeddings

BERTopic (Grootendorst, 2022)

- generates document embedding with pre-trained transformer-based language models, clusters these embeddings
- generates topic representations with the class-based TF-IDF procedure
- transformer-based topic model
- *not* a statistical model (like LDA), but a pipeline of data science techniques

Implementation with Python and Tutorials:

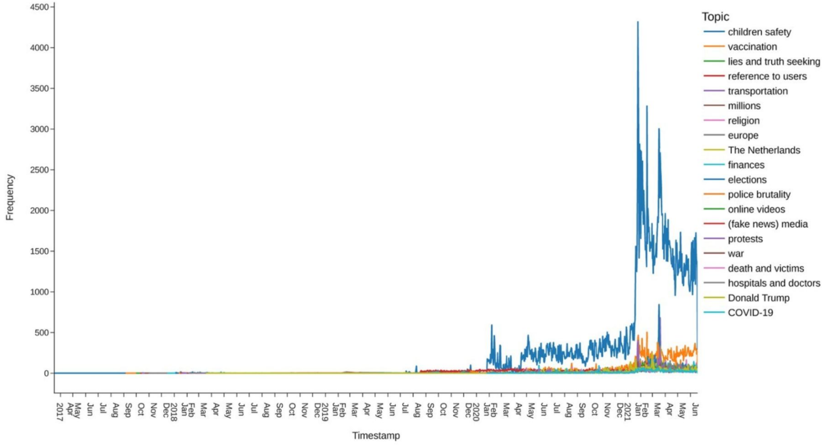
Github: https://github.com/MaartenGr/BERTopic

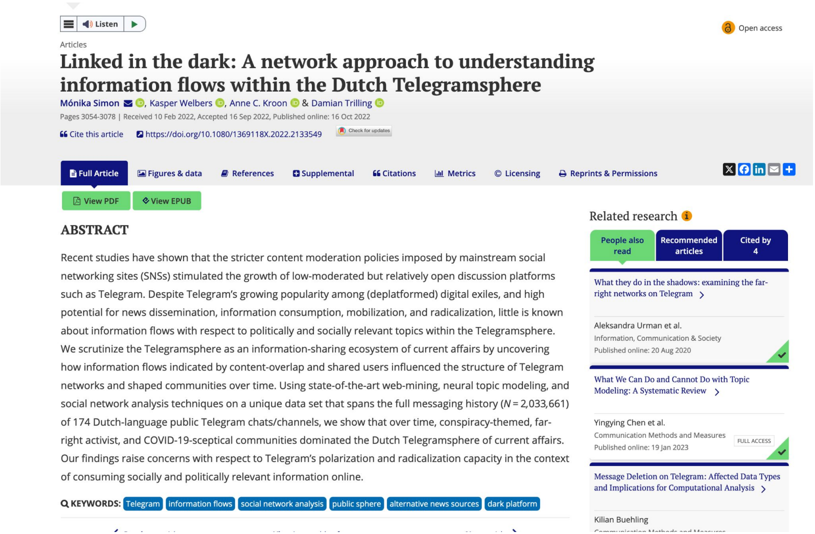
Documentation: https://maartengr.github.io/BERTopic/api/bertopic.html

BERTopic Applications in Social Science

Figure 5 of 6

Figure 5. The top 20 topics overtime (18-03-2017–18-06-2021).





Articles

Open access

Linked in the dark: A network approach to understanding information flows within the Dutch Telegramsphere

Mónika Simon, Kasper Welbers, Anne C. Kroon & Damian Trilling

Pages 3054-3078 | Received 10 Feb 2022, Accepted 16 Sep 2022, Published online: 16 Oct 2022

Cite this article https://doi.org/10.1080/1369118X.2022.2133549

Full Article | Figures & data | References | Supplemental | Citations | Metrics | Licensing | Reprints & Permissions

View PDF | View EPUB

ABSTRACT

Recent studies have shown that the stricter content moderation policies imposed by mainstream social networking sites (SNSs) stimulated the growth of low-moderated but relatively open discussion platforms such as Telegram. Despite Telegram's growing popularity among (deplatformed) digital exiles, and high potential for news dissemination, information consumption, mobilization, and radicalization, little is known about information flows with respect to politically and socially relevant topics within the Telegramsphere. We scrutinize the Telegramsphere as an information-sharing ecosystem of current affairs by uncovering how information flows indicated by content-overlap and shared users influenced the structure of Telegram networks and shaped communities over time. Using state-of-the-art web-mining, neural topic modeling, and social network analysis techniques on a unique data set that spans the full messaging history ($N = 2{,}033{,}661$) of 174 Dutch-language public Telegram chats/channels, we show that over time, conspiracy-themed, far-right activist, and COVID-19-sceptical communities dominated the Dutch Telegramsphere of current affairs. Our findings raise concerns with respect to Telegram's polarization and radicalization capacity in the context of consuming socially and politically relevant information online.

KEYWORDS: Telegram | information flows | social network analysis | public sphere | alternative news sources | dark platform

Related research

People also read | Recommended articles | Cited by 4

What they do in the shadows: examining the far-right networks on Telegram
Aleksandra Urman et al.
Information, Communication & Society
Published online: 20 Aug 2020

What We Can Do and Cannot Do with Topic Modeling: A Systematic Review
Yingying Chen et al.
Communication Methods and Measures
Published online: 19 Jan 2023

Message Deletion on Telegram: Affected Data Types and Implications for Computational Analysis
Kilian Buehling

Simon, M., Welbers, K. C., Kroon, A., & Trilling, D. (2022). Linked in the dark: A network approach to understanding information flows within the Dutch Telegramsphere. *Information, Communication & Society*, 1–25. https://doi.org/10.1080/1369118X.2022.2133549

Validating topic models

Tests of:

- topic semantic validity: assess the extent to which the keywords within each topic have a coherent underlying meaning, and how these meanings behave across topics
- convergent validity: topic probabilities per document are compared with an external trusted variable like manual coding for the same documents

A possible validation workflow proposed by Maier et al. (2018): combi of quantitative topic summary metrics and human expert evaluations.

See also Bernhard et al., (2023) for an overview of validation methods.

Bernhard, J., Teuffenbach, M., & Boomgaarden, H. G. (2023). Topic Model validation methods and their impact on Model selection and evaluation. *Computational Communication Research*, *5*(1), 1.

Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., ... & Adam, S. (2021). Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. In *Computational methods for communication science* (pp. 13-38). Routledge.

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)

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

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

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

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