Winter semester 2022/23

344.175 VL: Natural Language Processing
Document Classification – BoW and Semantic Methods

Navid Rekab-saz

Email: navid.rekabsaz@jku.at
Office hours: https://navid-officehours.youcanbook.me

JOHANNES KEPLER
UNIVERSITY LINZ

Institute of
Computational
Perception

Agenda

- Sentiment Analysis
- Document representation with Bag of Words
- Semantic document representation with SVD
- Document classification overview

Notation

- $a \longrightarrow$ a value or a scalar

- $\boldsymbol{b} \longrightarrow$ an array or a vector
  - i^{th} element of $\boldsymbol{b}$ is the scalar b_i

- $\boldsymbol{C} \longrightarrow$ a set of arrays or a matrix
  - i^{th} vector of $\boldsymbol{C}$ is $\boldsymbol{c}_i$
  - j^{th} element of the i^{th} vector of $\boldsymbol{C}$ is the scalar $c_{i,j}$

Linear Algebra – Transpose

- $\boldsymbol{a}$ is in $1 \times d$ dimensions $\rightarrow$ $\boldsymbol{a}^{\mathrm{T}}$ is in $d \times 1$ dimensions

- $\boldsymbol{A}$ is in $e \times d$ dimensions $\rightarrow$ $\boldsymbol{A}^{\mathrm{T}}$ is in $d \times e$ dimensions

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Linear Algebra – Dot product

- $\boldsymbol{a} \cdot \boldsymbol{b}^T = c$
  - dimensions: $1 \times d \cdot d \times 1 = 1$

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = 5$$

- $\boldsymbol{a} \cdot \boldsymbol{B} = \boldsymbol{c}$
  - dimensions: $1 \times d \cdot d \times e = 1 \times e$

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 5 & 2 \end{bmatrix}$$

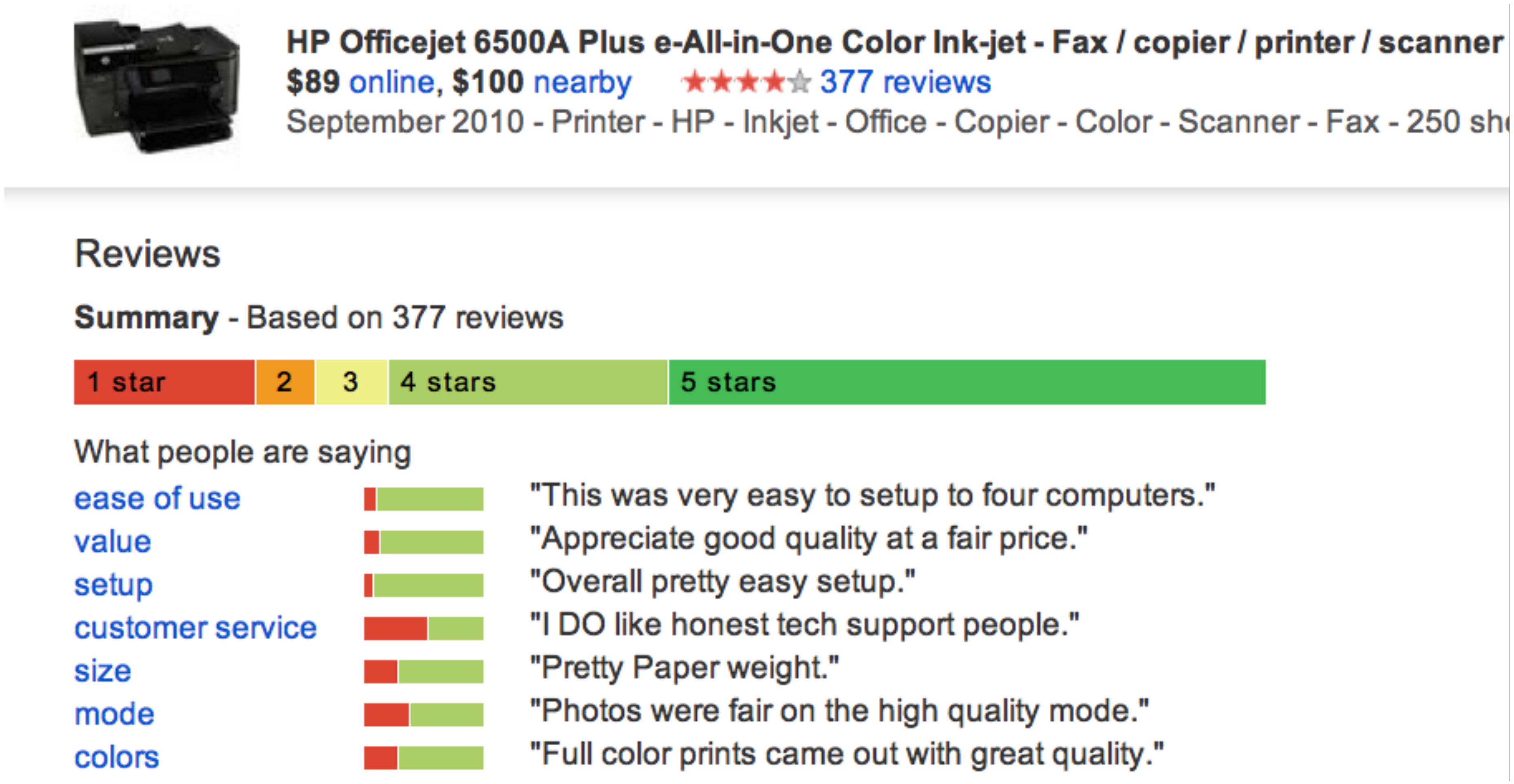
- $\boldsymbol{A} \cdot \boldsymbol{B} = \boldsymbol{C}$
  - dimensions: $l \times m \cdot m \times n = l \times n$

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ 0 & 0 & 5 \\ 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 3 & 2 \\ 5 & -5 \\ 8 & 13 \end{bmatrix}$$

Agenda

- **Sentiment Analysis**
- Document representation with Bag of Words
- Semantic document representation with SVD
- Document classification overview

Sentiment Analysis – Market Intelligence



HP Officejet 6500A Plus e-All-in-One Color Ink-jet - Fax / copier / printer / scanner
$89 online, **$100** nearby ★★★★☆ 377 reviews
September 2010 - Printer - HP - Inkjet - Office - Copier - Color - Scanner - Fax - 250 sh

Reviews

Summary - Based on 377 reviews

What people are saying

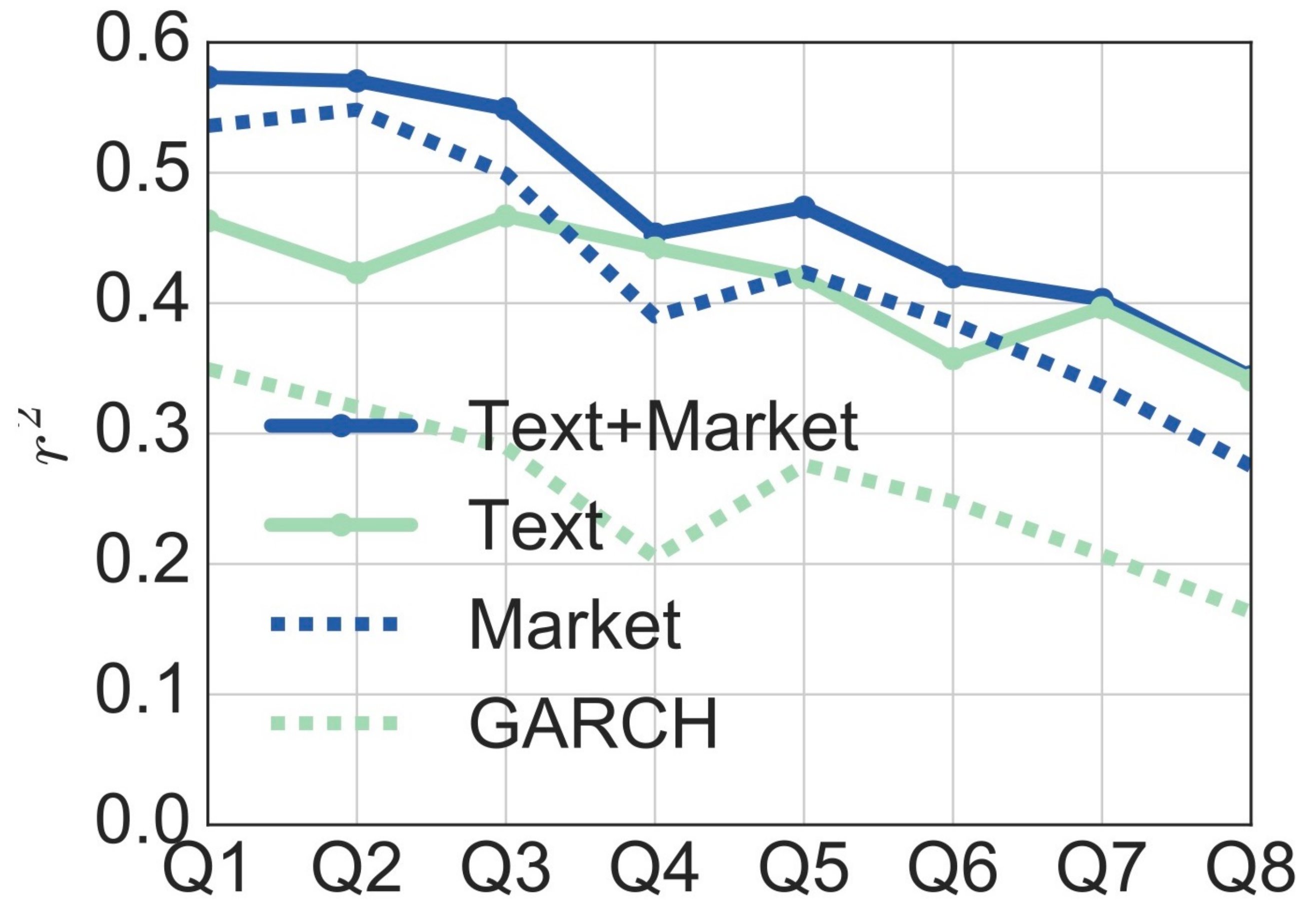
ease of use	"This was very easy to setup to four computers."
value	"Appreciate good quality at a fair price."
setup	"Overall pretty easy setup."
customer service	"I DO like honest tech support people."
size	"Pretty Paper weight."
mode	"Photos were fair on the high quality mode."
colors	"Full color prints came out with great quality."

Sentiment Analysis – Stance Detection

Tweet	Target	Sentiment	Stance Label
@rimmedlarry Actually, the tag was made by feminists so they can narcissistically post selfies to prove they're not ugly.	Feminist Movement	Negative	Against
SO EXCITING! Meaningful climate change action is on the way!	Climate Change is a Real Concern	Positive	Favor
When the last tree is cut down, the last fish eaten & the last stream poisoned, you will realize that you cannot eat money.	Climate Change is a Real Concern	Negative	Favor
dear lord thank u for all ofur blessings forgive my sins lord give me strength and energy for this busy day ahead.	Atheism Sentiment	Positive	Against

https://www.saifmohammad.com/WebPages/StanceDataset.htm
Semeval-2016 Task 6: Detecting Stance in Tweets. Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. In Proceedings of the International Workshop on Semantic Evaluation (SemEval '16). June 2016. San Diego, California.

Sentiment Analysis – Market/Volatility prediction



Volatility Prediction using Financial Disclosures Sentiments with Word Embedding-based IR Models. *Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Allan Hanbury, Alexander Duer, Linda Anderson.* In proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2017
https://aclanthology.org/P17-1157.pdf

[5]

A *tough* Example!

"This past Saturday, I bought a Nokia phone and my girlfriend bought a Motorola phone with Bluetooth. We called each other when we got home. The voice on my phone was clear, better than my previous Samsung phone. The battery life was however short. My girlfriend was quite happy with her phone. I wanted a phone with good sound quality. So my purchase was a real disappointment. I returned the phone yesterday."

Text-level Sentiment Analysis

- Text-level sentiment analysis assumes that whole the text expresses one sentiment about one opinion target
  - Not like the previous example! That example is more suited for aspect-based sentiment analysis

How to create a sentiment prediction model?

A model can leverage specific words/phrases/etc. as signals to predict sentiment

- Lexicon-based methods: use a lexicon of sentiment words to output a sentiment score
- Supervised methods: learn a machine learning model to predict sentiment scores based on some representation/featurization of the text

Lexicon-based Sentiment Analysis

- Some sentiment lexicons:
  - SentiWordNet
    - Each term has a positive and a negative score
  - Financial sentiment dictionary
    - Groups of negative/positive/uncertain words
  - …

Terms	Neg_score	Pos_score
able	0	0.125
unable	0.75	0
emerging	0	0

Group	Sample terms
Negative	discontinued, penalties, misconduct
Positive	achieve, efficient, profitable
Uncertainty	approximate, fluctuate, uncertain, variability

<u>E.g., one way using SentiWordNet would be:</u>

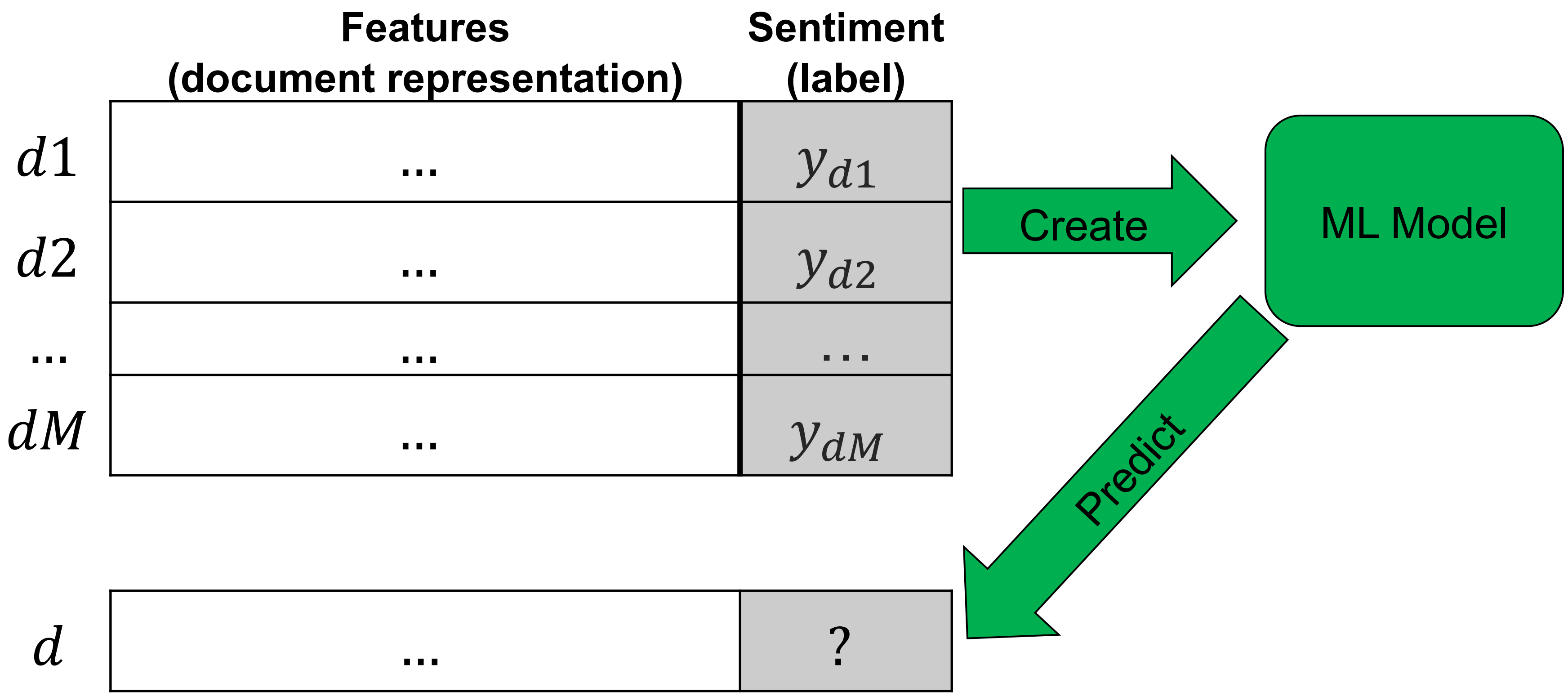
- The positive sentiment of the document d is the sum of the positive scores of the terms in the lexicon (V^{lex}), which appear in d:

$$pos\_sentiment(d) = \sum_{v \in \{V^{lex} \cap d\}} pos\_score(v)$$

Some resources
SentiWordNet: https://github.com/aesuli/SentiWordNet
Bing Liu's opinion lexicon: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
Loughran-McDonold financial sentiment dictionary: https://sraf.nd.edu/textual-analysis/resources/

Supervised Sentiment Analysis

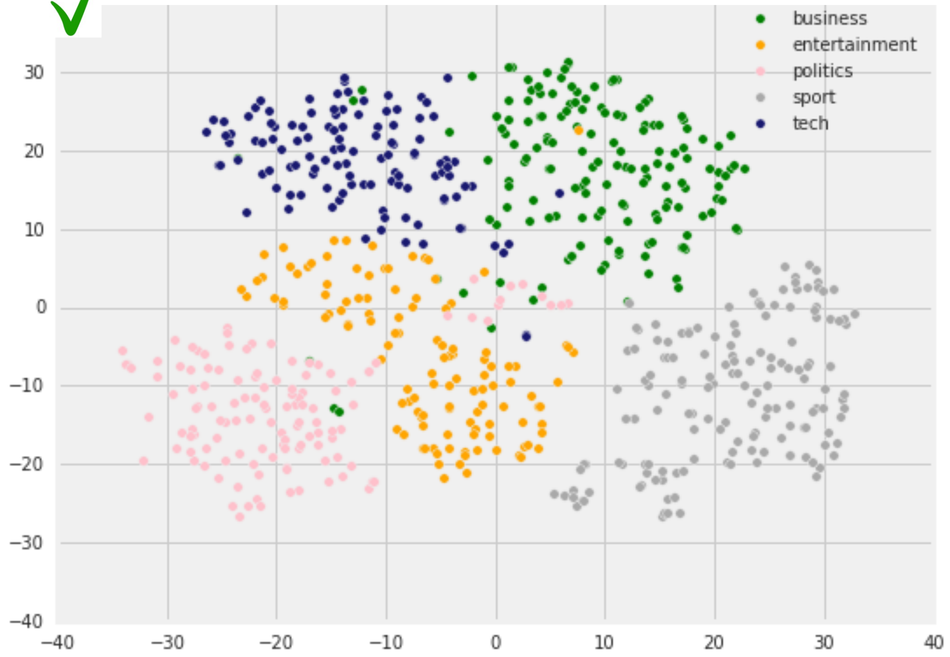
- A dataset consist of M documents and their assigned sentiments
- Possible sentiment values:
  - [-1, 0, 1] → [negative, neutral, positive] (**classification** problem)
  - Real-valued numbers e.g. stock price (regression problem)



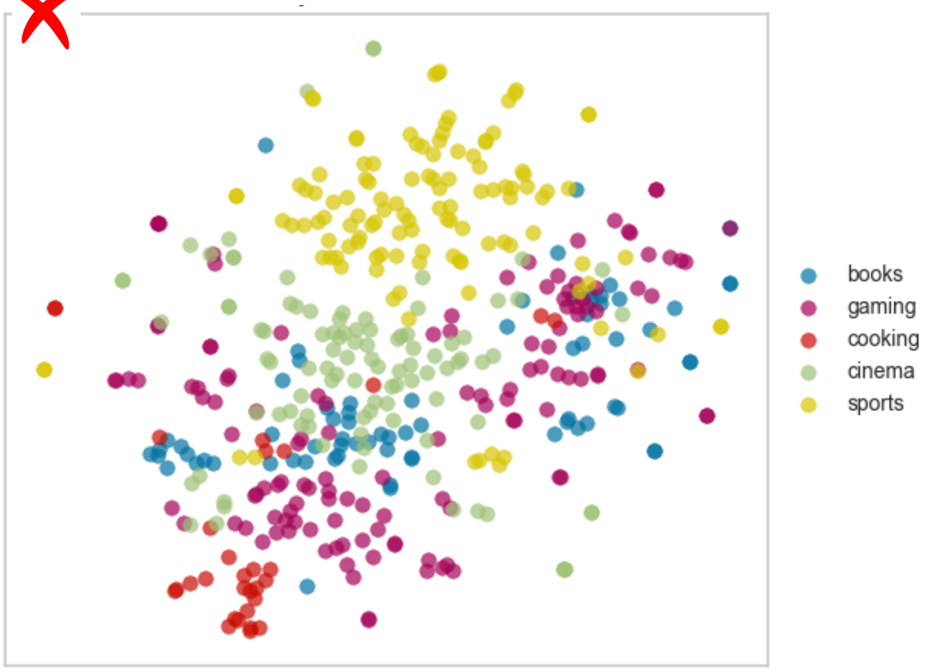
Representation is everything!

- The key to *good* classification is in creating *good* text representations!
  - Feature extraction in classical ML
  - Representation learning in deep learning
- When text representations are *well-separated* and *well-generalized*, ML models can easily find proper hyperplanes to classify/separate data points

✓



✗



Two sample document representation sets projected to two-dimensional spaces

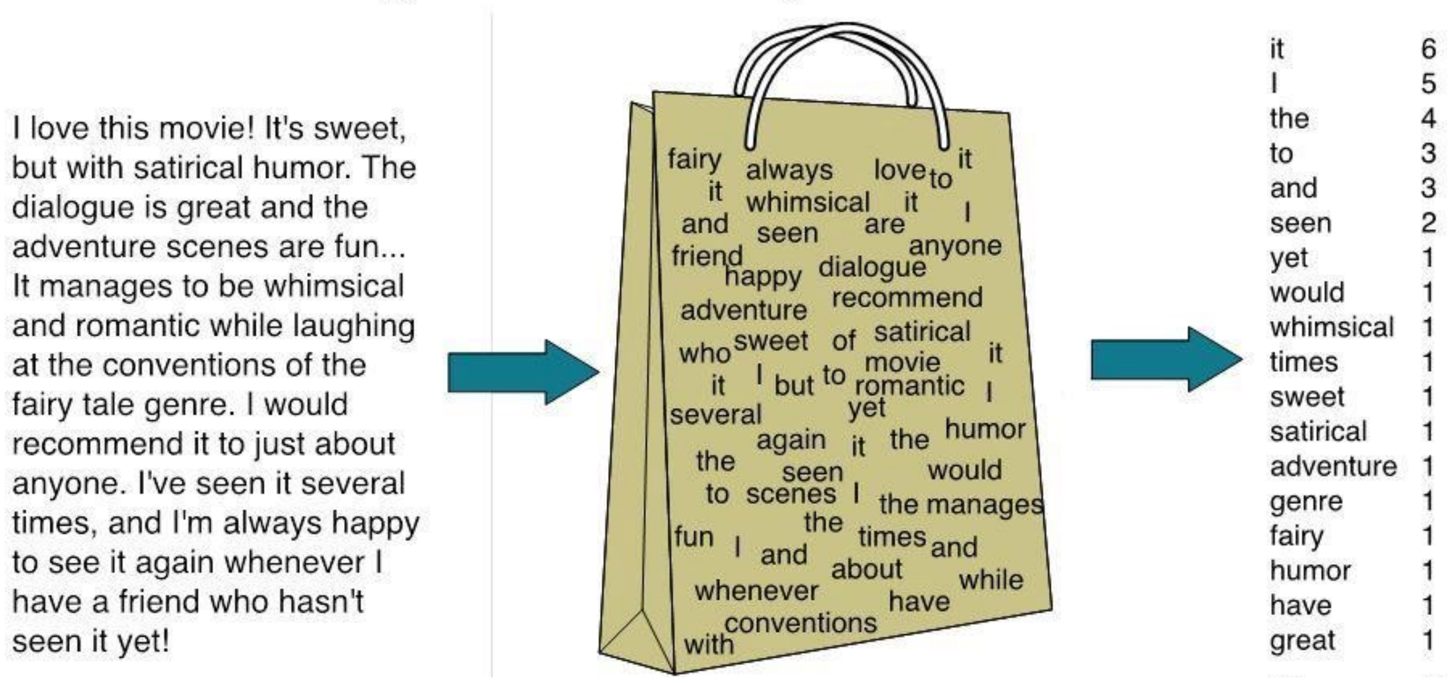
source source

Agenda

- Sentiment Analysis
- **Document representation with Bag of Words**
- Semantic document representation with SVD
- Document classification overview

Towards creating a document representation
Bag-of-words (BoW) Approach

- In BoW, the order and position of words is ignored and only the number occurrences of words are considered



I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

- What are the possible limitations of BoW approaches?

Source: https://www.programmersought.com/article/4304366575/

Towards BoW document representation
Creating Dictionary

- We first create a dictionary $\mathbb{V}$, consisting of N words (terms):

$$\mathbb{V} = \{v1, v2, \ldots, vN\}$$

- Dictionary should be created only using <u>training data</u>
- Dictionary need to be preprocessed, e.g., by:
  - keeping only top-*N* most frequent words
  - removing any word with a lower frequency than a threshold
- The tokens that do not appear in the dictionary are called Out-Of-Vocabulary (OOV)
- OOVs also need to be separately handled, e.g., by:
  - replacing them with a special token <oov>, and adding <oov> to the dictionary
  - Ignoring them completely in the processing (removing them from the text)

Document-Word matrix

- Featurization is done in a document-word matrix
- Document are data points (rows)
- Words in the dictionary are features (dimensions, columns)
- $x_{v,d}$ is the feature value of word v in document d
- Each value $x_{v,d}$ is calculated using a word (term) weighting model

	$v1$	$v2$	…	vN	**sentiment (label)**
$d1$	$x_{v1,d1}$	$x_{v2,d1}$	…	$x_{vN,d1}$	y_{d1}
$d2$	$x_{v1,d2}$	$x_{v2,d2}$	…	$x_{vN,d2}$	y_{d2}
…	…	…	…	…	…
dM	$x_{v1,dM}$	$x_{v2,dM}$	…	$x_{vN,dM}$	y_{dM}

Term weighting models
(1) Term count

Term (word) count:

- One common word weighting approach is to simply count the number of occurrences of a word in a document,
  - Example: number of times ***JKU*** appears in each news document.

$$x_{v,d} = \mathrm{tc}_{v,d} = \#\,\text{of occurrences of word } v \text{ in } d$$

Term weighting models
(2) Term frequency

Term frequency (version 1)

- Normalize term count over the length of document $|d|$

$$x_{v,d} = \mathrm{tf}_{v,d} = \frac{\mathrm{tc}_{v,d}}{|d|}$$

$|d|$ number of tokens in document d

Term frequency (version 2)

- The Importance of a word (probably) does not one-to-one increase with the number of occurrences
- Based on experimental results, logarithm is commonly used to dampen raw counts, resulting in …

$$x_{v,d} = \mathrm{tf}_{v,d} = \log(1 + \mathrm{tc}_{v,d})$$

On informativeness of less frequent words

- Words that do not appear often, usually carry more information in comparison with highly frequent ones
  - E.g., ***JKU*** in a large news should be more informative than the word ***university*** that appears more often.

- Inverse document frequency (idf) measures the importance of the words according to whole the collection of documents:

$$\mathrm{idf}_v = \log(\frac{M}{\mathrm{df}_v + 1})$$

  - M is the number of documents in the collection
  - df_v (document frequency of v) is the number of documents that contain word v

- Higher idf_v means that the word appears less often in the collection, and is therefore more informative (more important)
  - ***JKU*** has a higher idf than ***university***, and ***the*** gets a very low idf.

Term weighting models
(3) Term frequency-Inverse document frequency

- $\mathbf{tf{-}idf}$ weighting model is the product of $\mathrm{tf}_{v,d}$ to idf_v

$$x_{v,d} = \mathrm{tf{-}idf}_{v,d} = \mathrm{tf}_{v,d} \times \mathrm{idf}_v = \log\left(1 + \mathrm{tc}_{v,d}\right) \times \log\left(\frac{M}{\mathrm{df}_v + 1}\right)$$

tf increases with the number of occurrences <u>within the document</u>

idf increases with the rarity of the word in <u>whole the collection of documents</u>

- A well-known word weighting method!

Term weighting models
(4) Pivoted Length Normalization

- Pivoted Length Normalization (PL) model

Term matching score

$$x_{v,d} = \mathrm{PL}_{v,d} = \frac{\boxed{\log\left(1 + \mathrm{tc}_{v,d}\right)}}{\boxed{1 - b + b\,\frac{|d|}{avgdl}}} \times \boxed{\mathrm{idf}\,(v)}$$

Term Salience

Document length normalization

$avgdl$ average length of the documents in the collection

b a hyper parameter that controls document length normalization

Term weighting models
(5) BM25

- BM25 model *(slightly simplified)*:

Term matching score & normalization

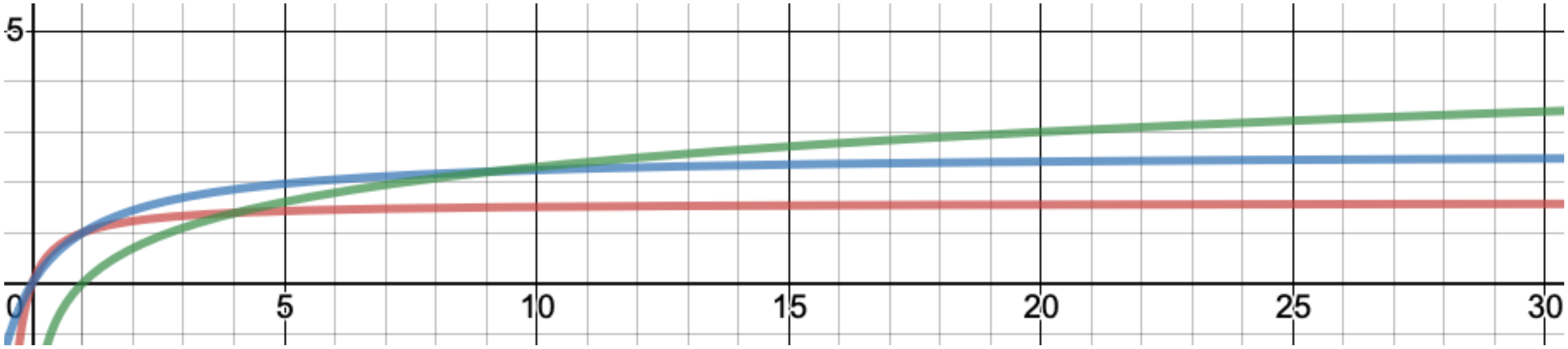
$$x_{v,d} = \mathrm{BM25}_{v,d} = \frac{(k_1 + 1)\mathrm{tc}_{v,d}}{k_1 \underbrace{\left(1 - b + b\frac{|d|}{avgdl}\right)}_{\text{Length normalization}} + \mathrm{tc}_{v,d}} \times \underbrace{\mathrm{idf}\,(v)}_{\text{Term Salience}}$$

$avgdl$ average length of the documents in the collection

b a hyper parameter that controls length normalization

k_1 a hyper parameter that controls term frequency *saturation*

BM25 – effect of term frequency saturation



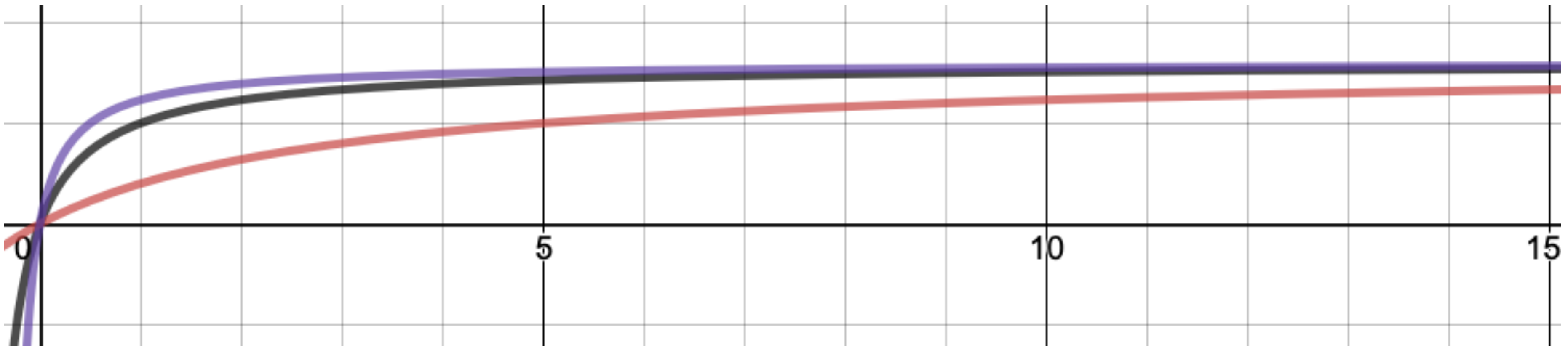
BM25 models with $b = 0$ (no length normalization)

Red: $\frac{(0.6+1)\mathrm{tc}_{v,d}}{0.6+\mathrm{tc}_{v,d}} \rightarrow$ BM25 with $k_1 = 0.6$

Blue: $\frac{(1.6+1)\mathrm{tc}_{v,d}}{1.6+\mathrm{tc}_{v,d}} \rightarrow$ BM25 with $k_1 = 1.6$

Green: $\log\ \mathrm{tc}_{v,d} \rightarrow$ TF

BM25 – effect of document length



BM25 models with $k_1 = 0.6$ and $b = 1$

Purple: $\frac{(0.6+1)\mathrm{tc}_{q_i,d}}{0.6(1-1+1(\frac{1}{2}))+\mathrm{tc}_{q_i,d}} \rightarrow |d|$ is half of $avgdl$

Black: $\frac{(0.6+1)\mathrm{tc}_{q_i,d}}{0.6(1-1+1(\frac{2}{2}))+\mathrm{tc}_{q_i,d}} \rightarrow |d|$ is the same as $avgdl$

Red: $\frac{(0.6+1)\mathrm{tc}_{q_i,d}}{0.6(1-1+1(\frac{10}{2}))+\mathrm{tc}_{q_i,d}} \rightarrow |d|$ is 5 times higher than $avgdl$

Supervised BoW text classification – summary

- Create the dictionary
- Use one of the word weightings to create document-word matrix

$$\begin{array}{c|cccc|c} & \multicolumn{4}{c|}{\overbrace{\hspace{12em}}^{\textbf{Features } (X)}} & \\ & v1 & v2 & \dots & vN & \textbf{sentiment (label)} \\ \hline d1 & x_{v1,d1} & x_{v2,d1} & \dots & x_{vN,d1} & y_{d1} \\ d2 & x_{v1,d2} & x_{v2,d2} & \dots & x_{vN,d2} & y_{d2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ dM & x_{v1,dM} & x_{v2,dM} & \dots & x_{vN,dM} & y_{dM} \end{array}$$

- The rest is applying standard machine learning pipeline (last topic of the lecture)

☞ Every row of features can be seen as a **document representation** and can be used e.g., for calculating its similarities to other documents (using a metric like cosine), clustering, visualization, etc.

Agenda

- Sentiment Analysis
- Document representation with Bag of Words
- **Semantic document representation with SVD**
- Document classification overview

Text classification with BoW

	$v1$	$v2$	...	vN	**sentiment (label)**
$d1$	$x_{v1,d1}$	$x_{v2,d1}$	...	$x_{vN,d1}$	y_{d1}
$d2$	$x_{v1,d2}$	$x_{v2,d2}$	...	$x_{vN,d2}$	y_{d2}
...	...	...	...	...	…
dM	$x_{v1,dM}$	$x_{v2,dM}$	...	$x_{vN,dM}$	y_{dM}

# of data points:

$$M \sim [10K - 100K]$$

# of features (dimensions):

$$N \sim [20K - 500K]$$

- BoW document representations are
  - sparse (a lot zeros) and …
  - typically in a very high dimension

Why low-dimensional vectors?

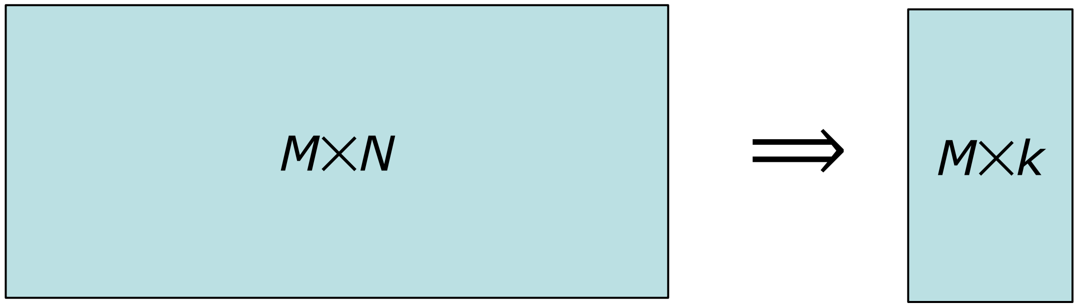
- Easier to store and load
- More efficient when used as features in ML models
- Better generalization due to the reduction of noise in data
- Able to capture higher-order relations:
  - Synonyms like *car* and *automobile* might be merged into the same dimensions

  - Polysemies like *bank (financial institution)* and *bank (bank of river)* might be separated into different dimensions

Polysemy is the capacity for a word or phrase to have multiple related meanings.

How to reduce features (dimensions)?

- Feature selection
  - Keep some important features and get rid of the rest!
- Dimensionality reduction
  - Project data from high to a low dimensional space



Feature selection

- During preprocessing
  - Remove stop words or very common words
    - tf–idf do it in a “soft” way, *why?*
  - Remove very rare words
    - Usually done as dictionary is created
  - Stemming & lemmatization

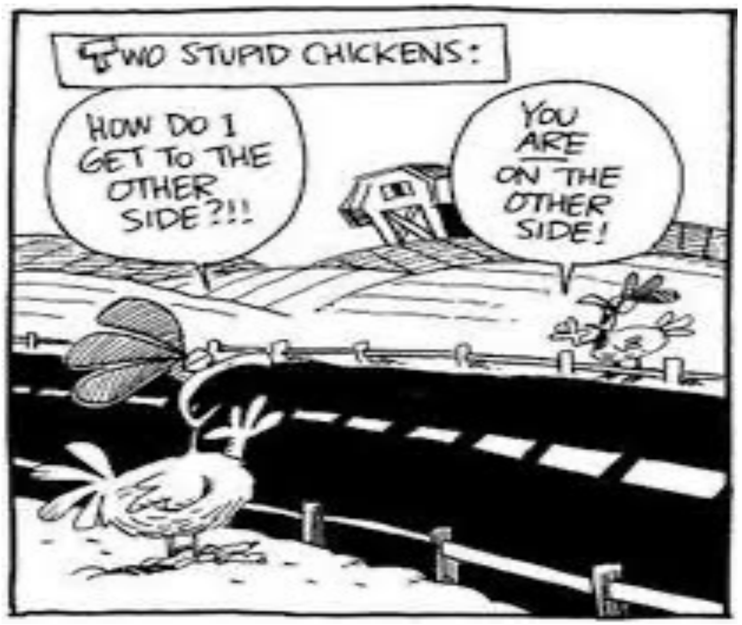
- Explicitly specifying features
  - E.g., by limiting the dictionary (and therefore features) to only the words of a domain-specific lexicon

- Post-processing
  - Keep important features using some informativeness measures
  - Subset selection

Dimensionality reduction with LSA

- Latent Semantic Analysis (LSA)
  - A common method to create semantic vectors
  - Based on Singular Value Decomposition (SVD)

Semantics matters!



Singular Value Decomposition

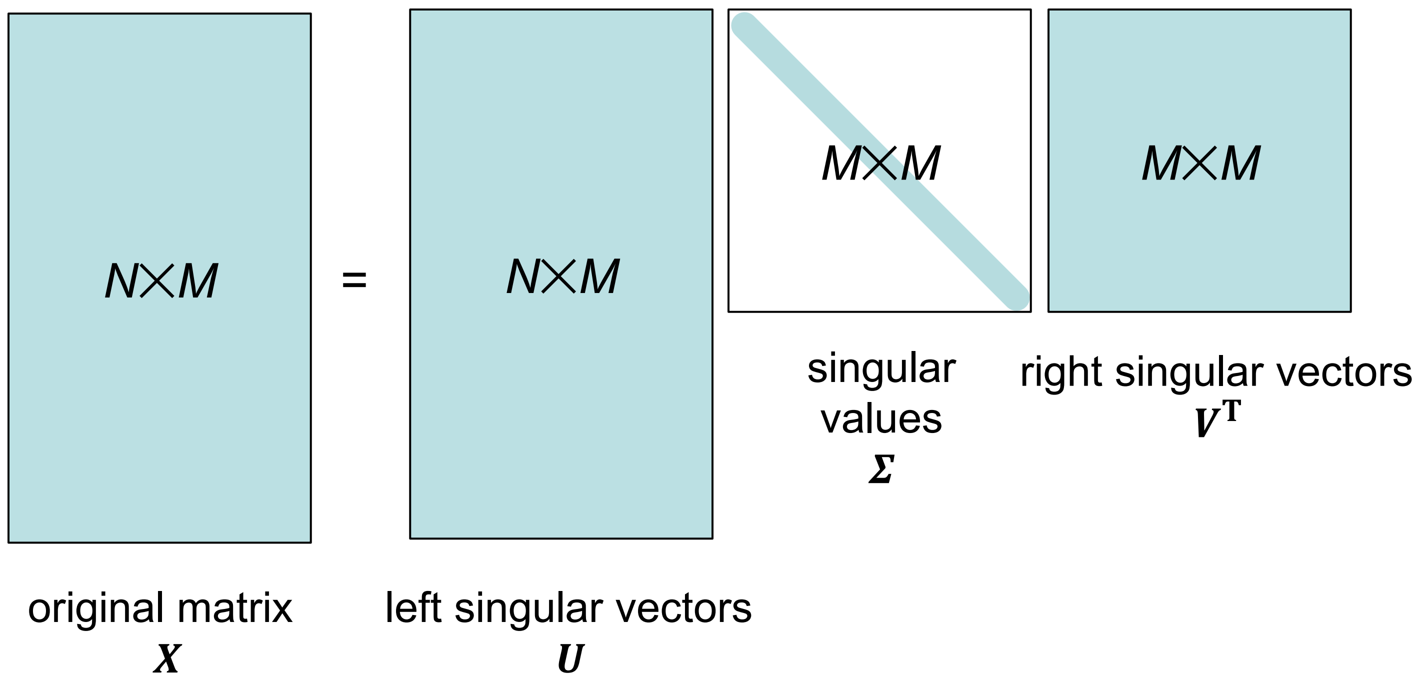
- An $N \times M$ matrix $\boldsymbol{X}$ can be factorized to three matrices:

$$\boldsymbol{X} = \boldsymbol{U\Sigma V}^{\mathrm{T}}$$

- $\boldsymbol{U}$ left singular vectors is an $N \times M$ unitary matrix
- $\boldsymbol{\Sigma}$ is an $M \times M$ diagonal matrix, diagonal entries
  - are singular values,
  - show the importance of corresponding M dimensions in $\boldsymbol{X}$
  - are all positive and sorted from large to small values
- $\boldsymbol{V}^{\mathrm{T}}$ right singular vectors is an $M \times M$ unitary matrix

* The definition of SVD is simplified. Refer to https://en.wikipedia.org/wiki/Singular_value_decomposition for the exact definition

Singular Value Decomposition



Latent Semantic Analysis – training

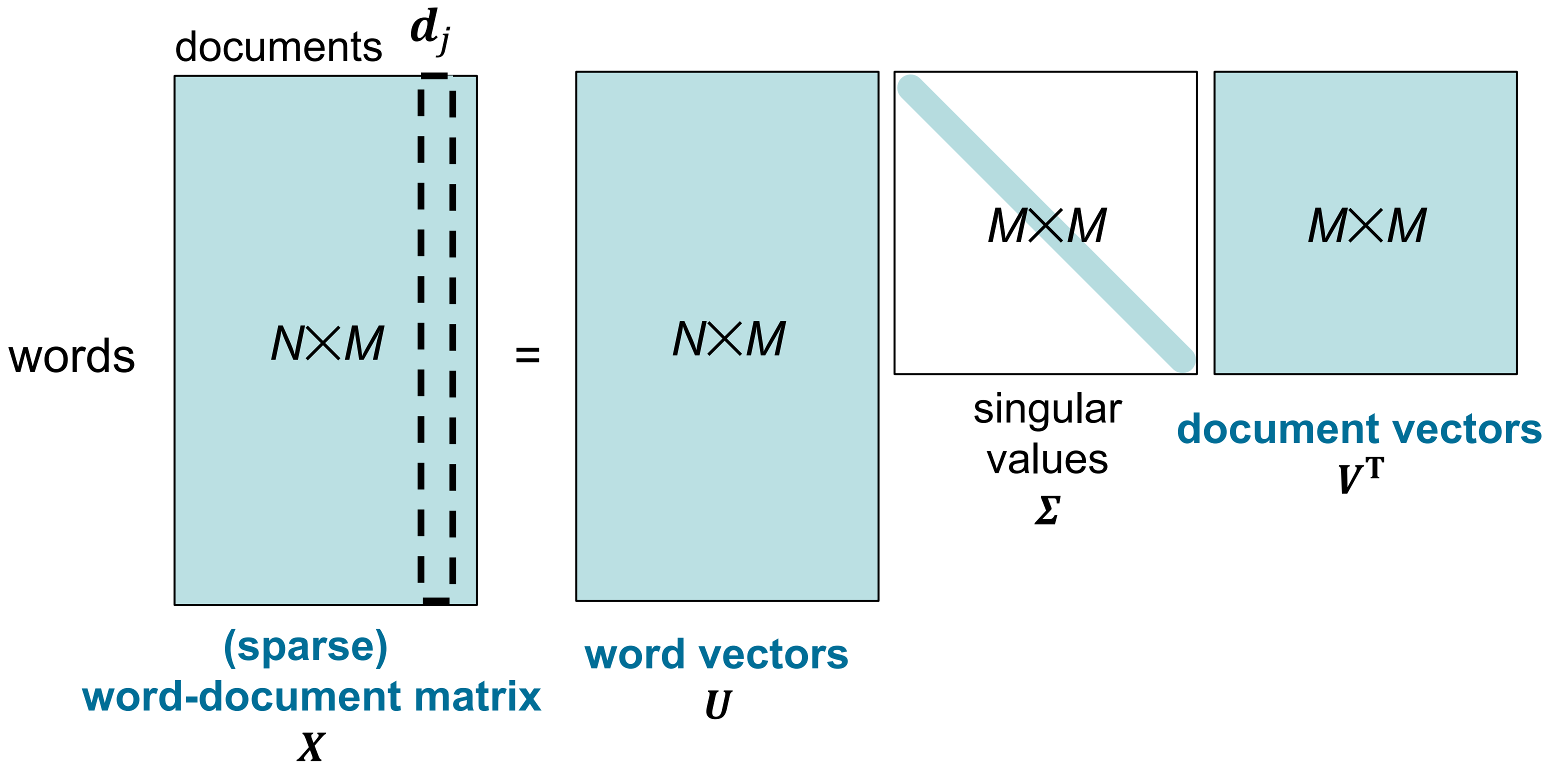
Training Time

Step 1

- Prepare the **word-document** matrix using training data

☞ Note that here we are using word-document matrix and **not** the document-word matrix (as we talked before)! While it is also technically possible to start with the document-word matrix, we follow here the common definition of LSA.

- Apply SVD to the matrix



Latent Semantic Analysis – training

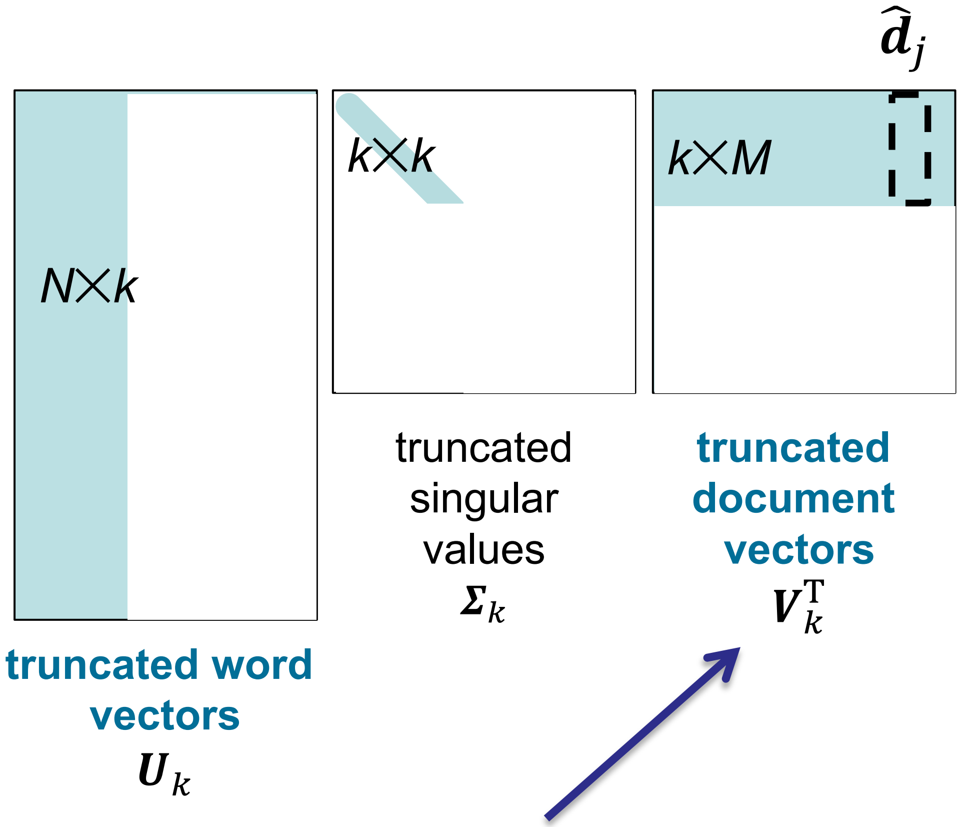
Training Time

Step 2

- Keep only top k singular values in $\boldsymbol{\Sigma}$ and set the rest to zero, called $\boldsymbol{\Sigma}_k$
- Truncate the $\boldsymbol{U}$ and $\boldsymbol{V}^{\mathrm{T}}$ matrices, resulting in $\boldsymbol{U}_k$ and $\boldsymbol{V}_k^{\mathrm{T}}$

- Columns in $\boldsymbol{V}_k^{\mathrm{T}}$ are the new low-dimensional document representations
  - Vectors of $\boldsymbol{V}_k^{\mathrm{T}}$ are used to train ML models

Latent Semantic Analysis – training


Training Time
$\hat{\boldsymbol{d}}_j$
$N \times k$
$k \times k$
$k \times M$
truncated word vectors $\boldsymbol{U}_k$
truncated singular values $\boldsymbol{\Sigma}_k$
truncated document vectors $\boldsymbol{V}_k^{\mathrm{T}}$

- $\boldsymbol{V}_k^{\mathrm{T}}$ is the matrix of dense low-dimensional document vectors
  - Used for training the ML models

Latent Semantic Analysis – example

Training Time

- A word-document matrix with 3 documents and 7 words
- Apply SVD to the matrix

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 5 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 0.04 & 0.35 & -0.91 \\ 0.21 & 0.16 & 0.20 \\ 0.94 & -0.17 & -0.04 \\ 0.12 & 0.88 & 0.27 \\ 0.18 & -0.03 & 0.00 \\ 0.02 & 0.20 & 0.20 \end{bmatrix}}_{\boldsymbol{U}} \cdot \underbrace{\begin{bmatrix} 5.21 & 0 & 0 \\ 0 & 4.59 & 0 \\ 0 & 0 & 1.66 \end{bmatrix}}_{\boldsymbol{\Sigma}} \cdot \underbrace{\begin{bmatrix} 0.15 & 0.04 & 0.98 \\ -0.92 & -0.34 & 0.15 \\ -0.34 & 0.93 & 0.01 \end{bmatrix}}_{\boldsymbol{V}^{\mathrm{T}}}$$

Latent Semantic Analysis – example

Training Time

$$\underbrace{\begin{bmatrix} 0.04 & 0.35 & -0.91 \\ 0.21 & 0.16 & 0.20 \\ 0.94 & -0.17 & -0.04 \\ 0.12 & 0.88 & 0.27 \\ 0.18 & -0.03 & 0.00 \\ 0.02 & 0.20 & 0.20 \end{bmatrix}}_{\boldsymbol{U}} \cdot \underbrace{\begin{bmatrix} 5.21 & 0 & 0 \\ 0 & 4.59 & 0 \\ 0 & 0 & 1.66 \end{bmatrix}}_{\boldsymbol{\Sigma}} \cdot \underbrace{\begin{bmatrix} 0.15 & 0.04 & 0.98 \\ -0.92 & -0.34 & 0.15 \\ -0.34 & 0.93 & 0.01 \end{bmatrix}}_{\boldsymbol{V}^{\mathrm{T}}}$$

- Keep only the top $k = 2$ singular values:

New k -dimensional document representations

$$\underbrace{\begin{bmatrix} 0.04 & 0.35 \\ 0.21 & 0.16 \\ 0.94 & -0.17 \\ 0.12 & 0.88 \\ 0.18 & -0.03 \\ 0.02 & 0.20 \end{bmatrix}}_{\boldsymbol{U}_k} \cdot \underbrace{\begin{bmatrix} 5.21 & 0 \\ 0 & 4.59 \end{bmatrix}}_{\boldsymbol{\Sigma}_k} \cdot \underbrace{\begin{array}{ccc} \hat{\boldsymbol{d}}_1 & \hat{\boldsymbol{d}}_2 & \hat{\boldsymbol{d}}_3 \\ \downarrow & \downarrow & \downarrow \\ \left[\begin{matrix} 0.15 \\ -0.92 \end{matrix}\right. & \begin{matrix} 0.04 \\ -0.34 \end{matrix} & \left.\begin{matrix} 0.98 \\ 0.15 \end{matrix}\right] \end{array}}_{\boldsymbol{V}_k^{\mathrm{T}}}$$

Latent Semantic Analysis – inference

<u>**Inference Time (Validation/Test)**</u>

- Given a high-dimensional document vector $\boldsymbol{d}^*$ in $N\times 1$ dimensions, we want to project it to the low-dimensional space, resulting in a new vector $\widehat{\boldsymbol{d}^*}$ with $k\times 1$ dimensions

- This is achieved through this calculation:

$$\widehat{\boldsymbol{d}^*} = \boldsymbol{\Sigma}_k^{-1}\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{d}^*$$

Exercise: examine this formula by calculating if the chain of dot products starting from $\boldsymbol{d}^*$ ends up to the correct dimension of $\widehat{\boldsymbol{d}^*}$

Latent Semantic Analysis – inference example

Inference Time (Validation/Test)

- Example: high-dimensional document $\boldsymbol{d}^*$

$$\boldsymbol{d}^* = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

- And the matrices calculated at train time:

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} 5.21 & 0 \\ 0 & 4.59 \end{bmatrix} \qquad \boldsymbol{\Sigma}_k^{-1} = \begin{bmatrix} 0.19 & 0 \\ 0 & 0.21 \end{bmatrix} \qquad \boldsymbol{U}_k = \begin{bmatrix} 0.04 & 0.35 \\ 0.21 & 0.16 \\ 0.94 & -0.17 \\ 0.12 & 0.88 \\ 0.18 & -0.03 \\ 0.02 & 0.20 \end{bmatrix}$$

$$\widehat{\boldsymbol{d}^*} = \boldsymbol{\Sigma}_k^{-1} \boldsymbol{U}_k^{\mathrm{T}} \boldsymbol{d}^* = \begin{bmatrix} 0.11 \\ 0.62 \end{bmatrix}$$

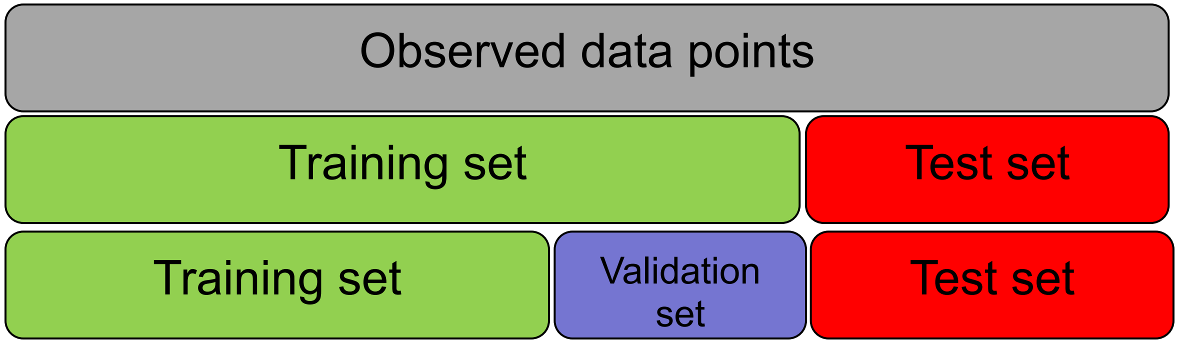
Agenda

- Sentiment Analysis
- Document representation with Bag of Words
- Semantic document representation with SVD
- **Document classification overview**

Splitting dataset

- Data is split into:
  - **Training set**: for training the model
  - **Validation set**: for tuning model's hyper-parameters
  - **Test set**: for evaluating model's performance

- Some common train – validation – test splitting sizes
  - 60%, 20%, 20%
  - 70%, 15%, 15%
  - 80%, 10%, 10%



Prepare document representations

- Create the dictionary
- Use one of the explained method to create document-word matrix

	Features (X)			label
			…	
$d1$	$x_{1,d1}$	$x_{2,d1}$	…	y_{d1}
$d2$	$x_{1,d2}$	$x_{2,d2}$	…	y_{d2}
…	…	…	…	…
dM	$x_{1,dM}$	$x_{2,dM}$	…	y_{dM}

☞ Only training set should be used for calculating dictionary and any collection-level statistic

- E.g., IDF should be calculated only using training set
- The same goes for calculating $\boldsymbol{U}_k$ and $\boldsymbol{\Sigma}_k$ in LSA

Machine learning models

- Some common models:
  - Linear Regression / Logistic Regression
  - Support Vector Machines (SVM)
  - Decision Tree / Random Forest
  - Neural Networks (Multi-layer Perceptron)
- Model parameters
  - Each model has a set of variables (parameters) which should be learned during training
- Loss function
  - A function that measures the discrepancies between the predicted outputs $\hat{\boldsymbol{y}}$ and the actual labels $\boldsymbol{y}$
  - The model uses loss function to find an *optimum* set of parameters that reduce the loss
- Regularization
  - A regularization method introduces additional information (assumptions) to **avoid overfitting** by **decreasing variance**

Which ML model?

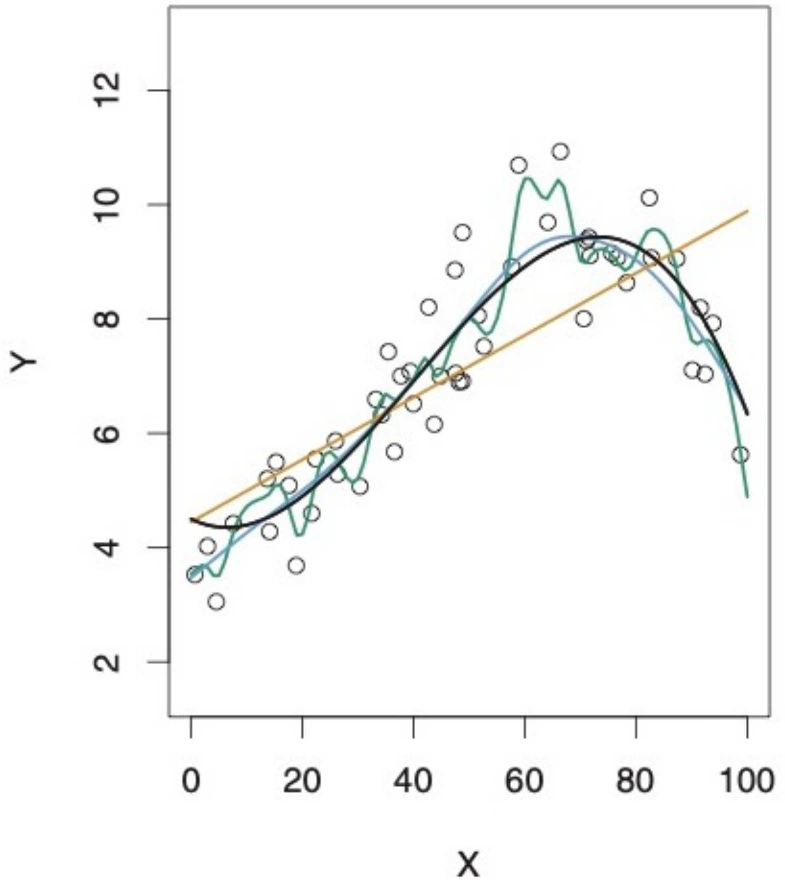
Model Capacity

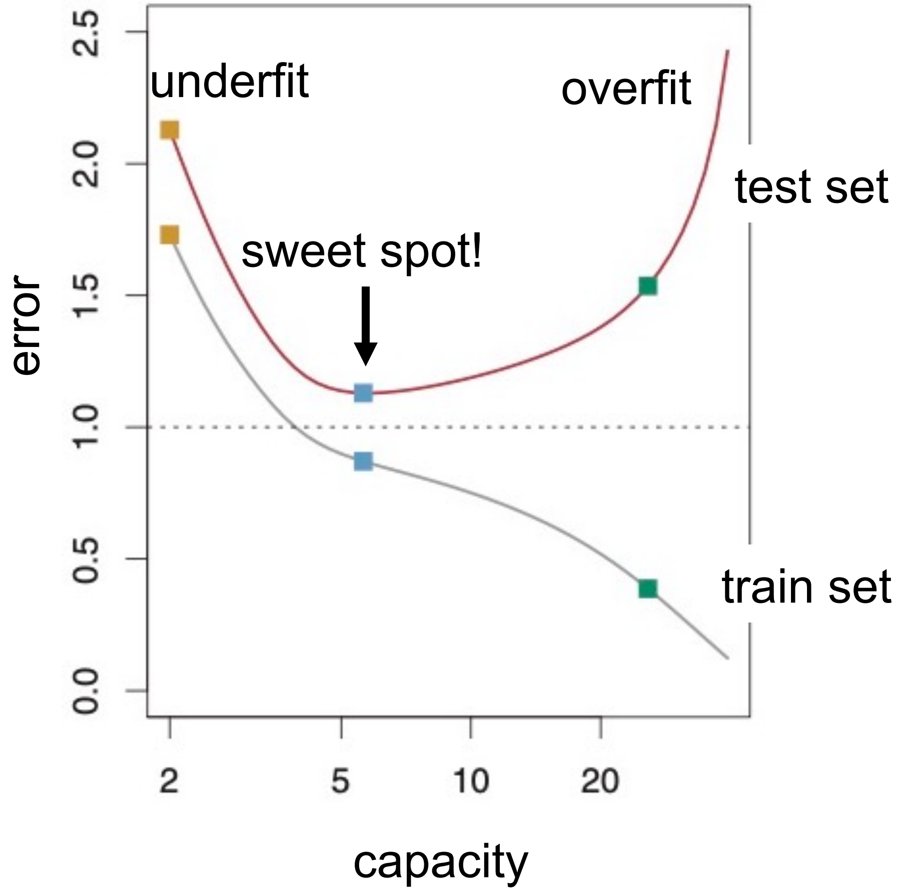
low ⟷ high

low	high
less flexible	more flexible
less parameters	more parameters
lower variance	higher variance
higher bias	lower bias
prone to underfitting	prone to overfitting

Terms of the day!
(Statistical) Bias indicates the amount of assumptions, taken to define a model. Higher bias means more assumptions and less flexibility, as in linear regression.
Variance: in what extent the estimated parameters of a model vary when the values of data points change (are resampled).
Overfitting: When the model exactly fits to training data, namely when it also captures the noise in data.

Which ML model?





Models:
black → f^{TRUE}
orange → linear regression (higher bias, less variance)
blue and **green**→ two smoothing spline models (less bias, higher variance)

Evaluation

- Measuring model's performance by comparing predictions with actual labels

Classification

- **Accuracy** $\frac{\#\ of\ correct\ predictions}{\#\ of\ samples}$
- Precision $\frac{TP}{TP+FP}$
- Recall $\frac{TP}{TP+FN}$
- F-measure $\frac{2 * precision * recall}{precision + recall}$

Model selection

- Hyperparameters
  - A set of specifications of each model, set before training
    - E.g., the kernel shape of SVM, or regularization weight in logistic regression
- Model selection
  - Train different variations of a model such as different hyperparameters' values, different weightings
  - Evaluate each model on validation set according to an evaluation metric
  - Select the best performing model on the validation set and report its evaluation results on test set