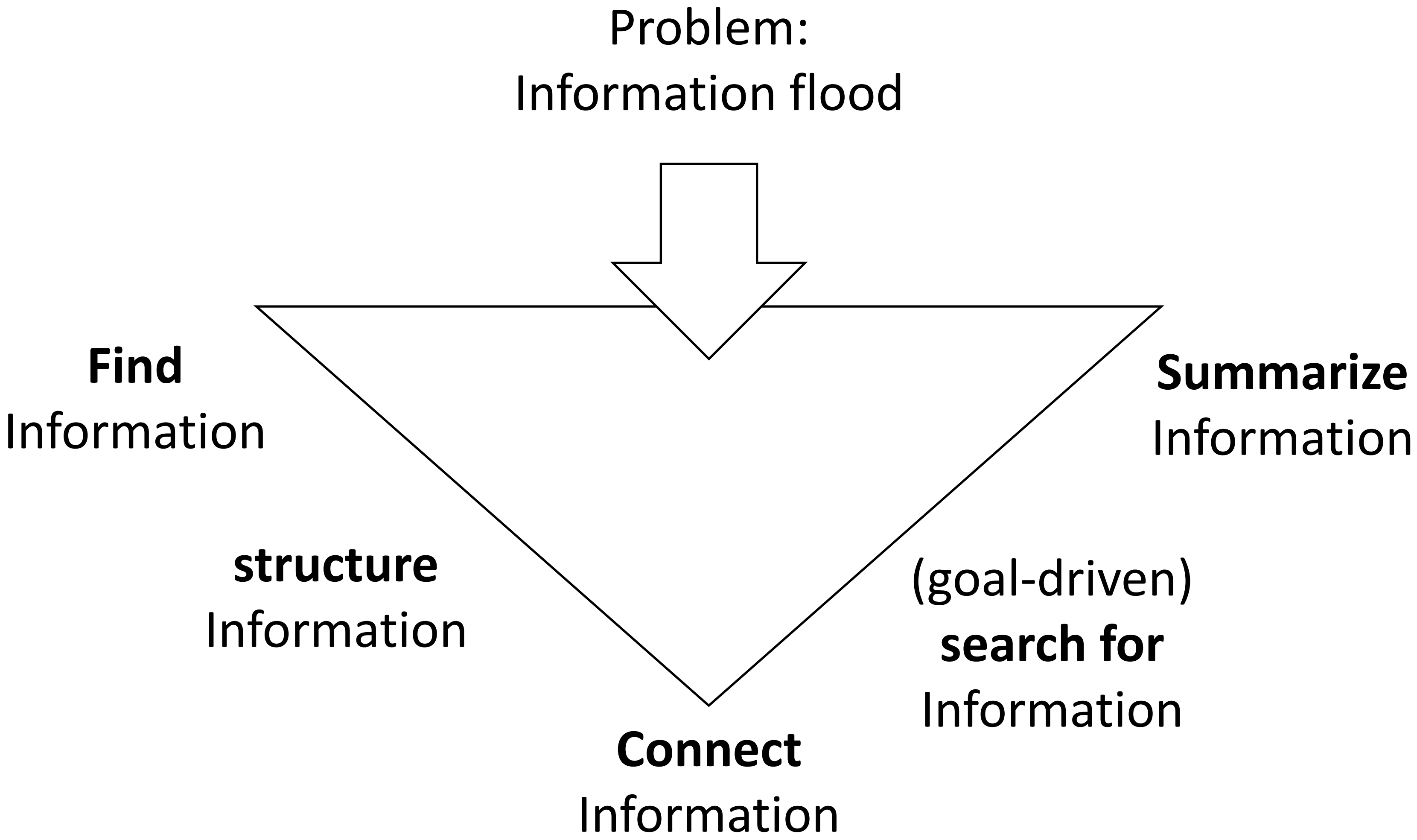
Technische Hochschule Ingolstadt

Information Retrieval

„Information retrieval is the activity of obtaining information resources relevant for a user’s information need from a collection of information resources“

https://www.uni-mannheim.de/media/Einrichtungen/dws/Files_People/Profs/goran/1-Introduction-to-IR.pdf

Introduction to Information Retrieval



Technische Hochschule Ingolstadt

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

Proposed by [Karen Spärck Jones] in a 1972

Technische Hochschule Ingolstadt

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$

N=2 since we have two documents, A and B

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$
$$n_t = |\{d \in D : t \in d\}|$$

documents that contain the term t

$n_{engine} = 1$

Technische Hochschule Ingolstadt

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$
$$n_t = |\{d \in D : t \in d\}|$$
$$IDF(t, D) = \log \frac{N}{n_t}$$

Problem?

$n_{house} = 0$

$$\begin{aligned} IDF &= -\log P(t|D) \\ &= \log \frac{1}{P(t|D)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t, D) = \log \frac{N}{n_t} = -\log \frac{n_t}{N}$$

Problem?

$n_{house} = 0$

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t, D) = \log \frac{N}{n_t} = -\log \frac{n_t}{N} \neq -\log \frac{1 + n_t}{N}$$

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$
$$n_t = |\{d \in D : t \in d\}|$$

Little bit of smoothing

$$IDF(t, D) = \log\left(\frac{N}{1 + n_t}\right) + 1$$

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$
$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t, D) = \log\left(\frac{\max_{t' \in d} n_{t'}}{1 + n_t}\right)$$

Prevents bias from long documents

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

$$N = |D|$$
$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t, D) = \log \frac{N - (1 + n_t)}{1 + n_t}$$

„attempt“ to define a log-probability

IDF := Inverse Document Frequency

„The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.“

Dokument A: „The car has an engine. The engine requires an energy source. …“

Dokument B: „A boat swims on the water. Thus it displaces water. …“

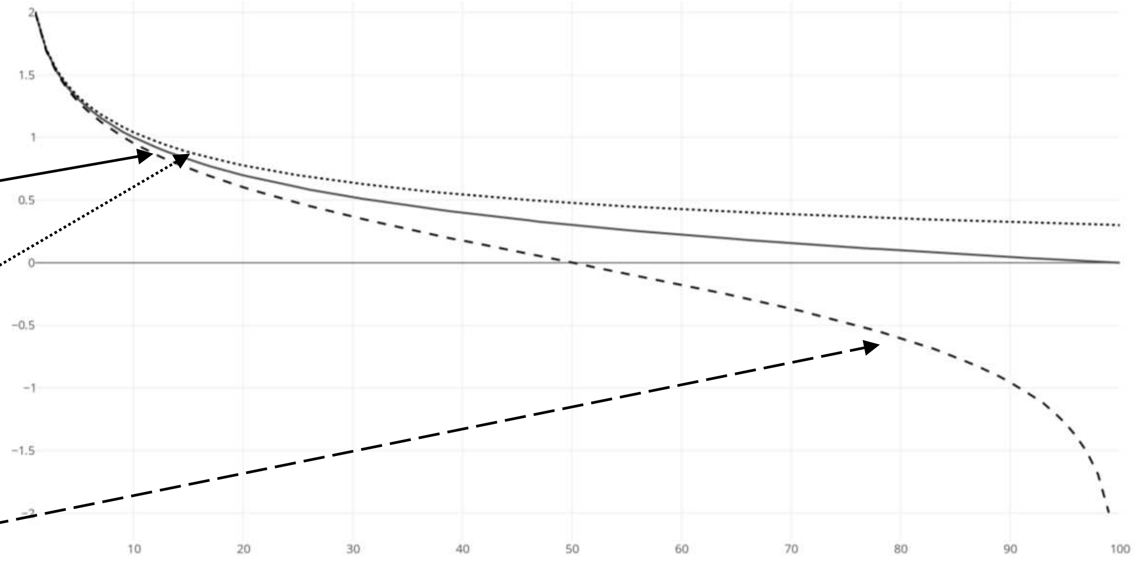
$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t, D) = \log \frac{N}{n_t} = -\log \frac{n_t}{N}$$

$$IDF(t, D) = \log \left(\frac{N}{1 + n_t} \right) + 1$$

$$IDF(t, D) = \log \frac{N - (1 + n_t)}{1 + n_t}$$



TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. …“

Document B: „A boat swims on the water. Thus it displaces water. …“

An n-gram would also take into account

„requires an“

„on the“

„has an“

But do these words carry relevant information?

TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. …“

Document B: „A boat swims on the water. Thus it displaces water. …“

$$TF(t,d) = \begin{cases} 1 & t \text{ in } d \\ 0 & \text{else} \end{cases}$$

$$\left.\begin{array}{l} t = \text{„engine“} \\ d = \text{„Document A“} \end{array}\right\} \quad TF(\text{engine, Document A}) = 1$$

$$\left.\begin{array}{l} t = \text{„engine“} \\ d = \text{„Document B“} \end{array}\right\} \quad TF(\text{engine, Document B}) = 0$$

Technische Hochschule Ingolstadt

TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. ...“

Document B: „A boat swims on the water. Thus it displaces water. ...“

$$TF(t, d) = C_{t,d}$$

t = „engine“
d = „Document A“

TF(engine, Document A) = 2

t = „engine“
d = „Document B“

TF(engine, Document B) = 0

Technische Hochschule Ingolstadt

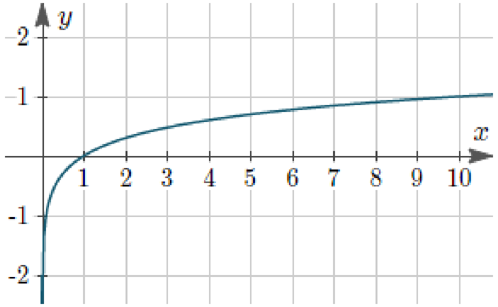
TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. …“

Document B: „A boat swims on the water. Thus it displaces water. …“

Log-normalized



$$TF(t, d) = \log(1 + C_{t,d})$$

t = „engine“
d = „Document A“

TF(engine, Document A) = log(1+2)

Technische Hochschule Ingolstadt

TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. ...“

Document B : „A boat swims on the water. Thus it displaces water. ...“

normalized over document length

$$TF(t, d) = \frac{C_{t,d}}{\sum_{t' \in d} C_{t',d}}$$

t = „engine“
d = „Document A“

TF(engine, Document A) = 2/10 = 0.2

Technische Hochschule Ingolstadt

TF := Term Frequency

„The weight of a term that occurs in a document is simply proportional to the term frequency.“

Document A: „The car has an engine. The engine requires an energy source. …“

Document B: „A boat swims on the water. Thus it displaces water. …“

t = „engine“
d = „Document A“

TF(engine, Document A) = 2/2 = 1

Normalized over document length. Mitigates bias from long documents.

$$TF(t,d) = \lambda + (1-\lambda))\frac{C_{t,d}}{\max_{t' \in d} C_{t',d}}$$

Technische Hochschule Ingolstadt

TFIDF

“Frequency of occurence
vs.
Document frequency”

Technische Hochschule Ingolstadt

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

"Its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it."

https://github.com/kavgan/nlp-in-practice/tree/master/tfidftransformer

https://github.com/gearmonkey/tfidf-python

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

It‘s all about Entropy

$$H(\mathcal{D}|\mathcal{T}=t) = -\sum_d p_{d|t} \log p_{d|t} = -\log \frac{1}{|\{d \in D : t \in d\}|} = \log \frac{|\{d \in D : t \in d\}|}{|D|} + \log |D| = -\mathrm{idf}(t) + \log |D|$$

“Its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.”

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$\begin{aligned} IDF &= -\log P(t|D) \\ &= \log \frac{1}{P(t|D)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

$$H(\mathcal{D}|\mathcal{T}=t) = -\sum_d p_{d|t} \log p_{d|t} = -\log \frac{1}{|\{d \in D : t \in d\}|} = \log \frac{|\{d \in D : t \in d\}|}{|D|} + \log |D| = -\mathrm{idf}(t) + \log |D|$$

"Its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it."

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$\begin{aligned} IDF &= -\log P(t|D) \\ &= \log \frac{1}{P(t|D)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

$$\underbrace{H(\mathcal{D}|\mathcal{T}=t) = -\sum_d p_{d|t} \log p_{d|t} = -\log \frac{1}{|\{d \in D : t \in d\}|} = \log \frac{|\{d \in D : t \in d\}|}{|D|} + \log |D| = -\mathrm{idf}(t) + \log |D|}$$

$$M(\mathcal{T};\mathcal{D}) = H(\mathcal{D}) - H(\mathcal{D}|\mathcal{T}) = \sum_t p_t \cdot (H(\mathcal{D}) - H(\mathcal{D}|W=t)) = \sum_t p_t \cdot \mathrm{idf}(t)$$

“Its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.”

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$\begin{aligned} IDF &= -\log P(t|D) \\ &= \log \frac{1}{P(t|D)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

$$\underbrace{H(\mathcal{D}|\mathcal{T}=t) = -\sum_d p_{d|t} \log p_{d|t} = -\log \frac{1}{|\{d \in D : t \in d\}|} = \log \frac{|\{d \in D : t \in d\}|}{|D|} + \log |D| = -\mathrm{idf}(t) + \log |D|}$$

$$M(\mathcal{T};\mathcal{D}) = H(\mathcal{D}) - H(\mathcal{D}|\mathcal{T}) = \sum_t p_t \cdot (H(\mathcal{D}) - H(\mathcal{D}|W=t)) = \sum_t p_t \cdot \mathrm{idf}(t)$$

$$M(\mathcal{T};\mathcal{D}) = \sum_{t,d} p_{t|d} \cdot p_d \cdot \mathrm{idf}(t) = \sum_{t,d} \mathrm{tf}(t,d) \cdot \frac{1}{|D|} \cdot \mathrm{idf}(t) = \frac{1}{|D|} \sum_{t,d} \boxed{\mathrm{tf}(t,d) \cdot \mathrm{idf}(t)}.$$

“Its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.”

Technische Hochschule Ingolstadt

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$TF(t,d) = \begin{cases} 1 & t \text{ in } d \\ 0 & \text{else} \end{cases}$$

$$TF(t,d) = C_{t,d}$$

$$TF(t,d) = \log(1 + C_{t,d})$$

$$TF(t,d) = \frac{C_{t,d}}{\sum_{t' \in d} C_{t',d}}$$

$$TF(t,d) = \lambda + (1-\lambda)) \frac{C_{t,d}}{\max_{t' \in d} C_{t',d}}$$

$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t,D) = \log \frac{N}{n_t} = -\log \frac{n_t}{N} \neq -\log \frac{1+n_t}{N}$$

$$IDF(t,D) = \log\left(\frac{N}{1+n_t}\right) + 1$$

$$IDF(t,D) = \log\left(\frac{\max_{t' \in d} n_{t'}}{1+n_t}\right)$$

$$IDF(t,D) = \log \frac{N-(1+n_t)}{1+n_t}$$

Technische Hochschule Ingolstadt

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$TF(t,d) = \begin{cases} 1 & t \text{ in } d \\ 0 & \text{else} \end{cases}$$

$$TF(t,d) = C_{t,d}$$

$$TF(t,d) = \log(1 + C_{t,d})$$

$$TF(t,d) = \frac{C_{t,d}}{\sum_{t' \in d} C_{t',d}}$$

$$TF(t,d) = \lambda + (1-\lambda)) \frac{C_{t,d}}{\max_{t' \in d} C_{t',d}}$$

$$IDF(t,D) = \log \frac{N}{n_t} = -\log \frac{n_t}{N} \neq -\log \frac{1+n_t}{N}$$

$$IDF(t,D) = \log \left(\frac{N}{1+n_t} \right) + 1$$

$$IDF(t,D) = \log \left(\frac{\max_{t' \in d} n_{t'}}{1+n_t} \right)$$

$$IDF(t,D) = \log \frac{N - (1+n_t)}{1+n_t}$$

TFIDF

$$TFIDF(D,t,d) = TF(t,d) \cdot IDF(t,D)$$

$$TF(t,d) = \begin{cases} 1 & t \text{ in } d \\ 0 & \text{else} \end{cases}$$

$$TF(t,d) = C_{t,d}$$

$$TF(t,d) = \log(1 + C_{t,d})$$

$$TF(t,d) = \frac{C_{t,d}}{\sum_{t' \in d} C_{t',d}}$$

$$TF(t,d) = \lambda + (1-\lambda))\frac{C_{t,d}}{\max_{t' \in d} C_{t',d}}$$

$$N = |D|$$

$$n_t = |\{d \in D : t \in d\}|$$

$$IDF(t,D) = \log\frac{N}{n_t} = -\log\frac{n_t}{N} \neq -\log\frac{1+n_t}{N}$$

$$IDF(t,D) = \log\left(\frac{N}{1+n_t}\right) + 1$$

$$IDF(t,D) = \log\left(\frac{\max_{t' \in d} n_{t'}}{1+n_t}\right)$$

$$IDF(t,D) = \log\frac{N-(1+n_t)}{1+n_t}$$

TFIDF
in Application

“Document classification
using
Naive Bayes”

Bayes-Document-TFIDF-classifier

$$D(\underline{t}) \quad = \operatorname{argmax}_{d=\{\dots\}} P(d|\underline{t})$$

https://towardsdatascience.com/training-a-naive-bayes-model-to-identify-the-author-of-an-email-or-document-17dc85fa630a

Bayes-Document-TFIDF-classifier

$$D(\underline{t}) = \text{argmax}_{d=\{\ldots\}}\ P(d \mid \text{TFIDF}(D,\underline{t},d))$$

Merkmale:

- Wörter
  (Language Identification)
- TFIDF
  (Document Classification)
- …

https://towardsdatascience.com/training-a-naive-bayes-model-to-identify-the-author-of-an-email-or-document-17dc85fa630a

Bayes-Document-TFIDF-classifier

$$D(\underline{t}) = \text{argmax}_{d=\{\ldots\}}\, P(d|TFIDF(D,\underline{t},d))$$

$$= \text{argmax}_{d=\{\ldots\}}\, P(TFIDF(D,\underline{t},d)|d)P(d)$$

https://towardsdatascience.com/training-a-naive-bayes-model-to-identify-the-author-of-an-email-or-document-17dc85fa630a

Bayes-Document-TFIDF-classifier

$$D(\underline{t}) = \operatorname{argmax}_{d=\{\dots\}} P(d \mid TFIDF(D,\underline{t},d))$$

$$= \operatorname{argmax}_{d=\{\dots\}} P(TFIDF(D,\underline{t},d) \mid d)P(d)$$

$$P(TFIDF(D,\underline{t},d) \mid d) = P_d(TFIDF(D,\underline{t},d))$$

Technische Hochschule Ingolstadt

Bayes-Document-TFIDF-classifier

$$D(\underline{t}) = \text{argmax}_{d=\{...\}} P(d|TFIDF(D,\underline{t},d))$$

$$= \text{argmax}_{d=\{...\}} P(TFIDF(D,\underline{t},d)|d)P(d)$$

$$P(TFIDF(D,\underline{t},d)|d) = P_d(TFIDF(D,\underline{t},d))$$

$$P(TFIDF(D,\underline{t},d)|d) = P_d(TFIDF(D,F(\underline{t}),d))$$

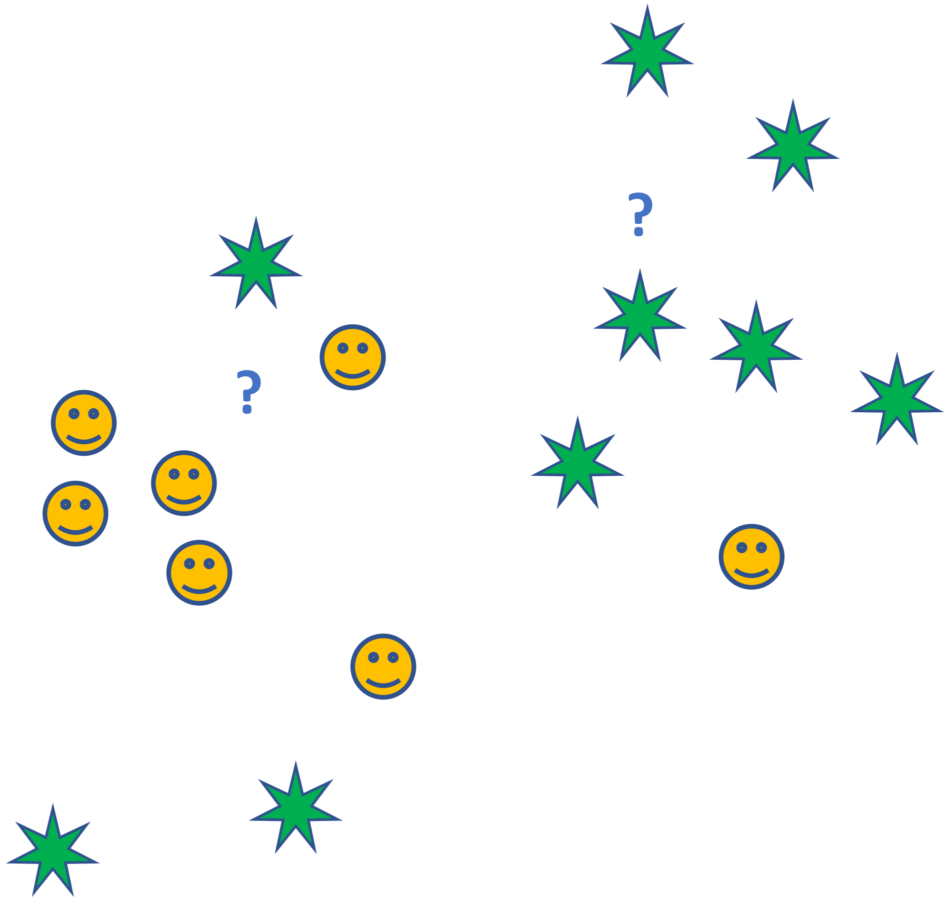
$F(\underline{w})$:= Text preprocessing

- Tokenization
- Character Set
- Punctuation
- ....

https://towardsdatascience.com/training-a-naive-bayes-model-to-identify-the-author-of-an-email-or-document-17dc85fa630a

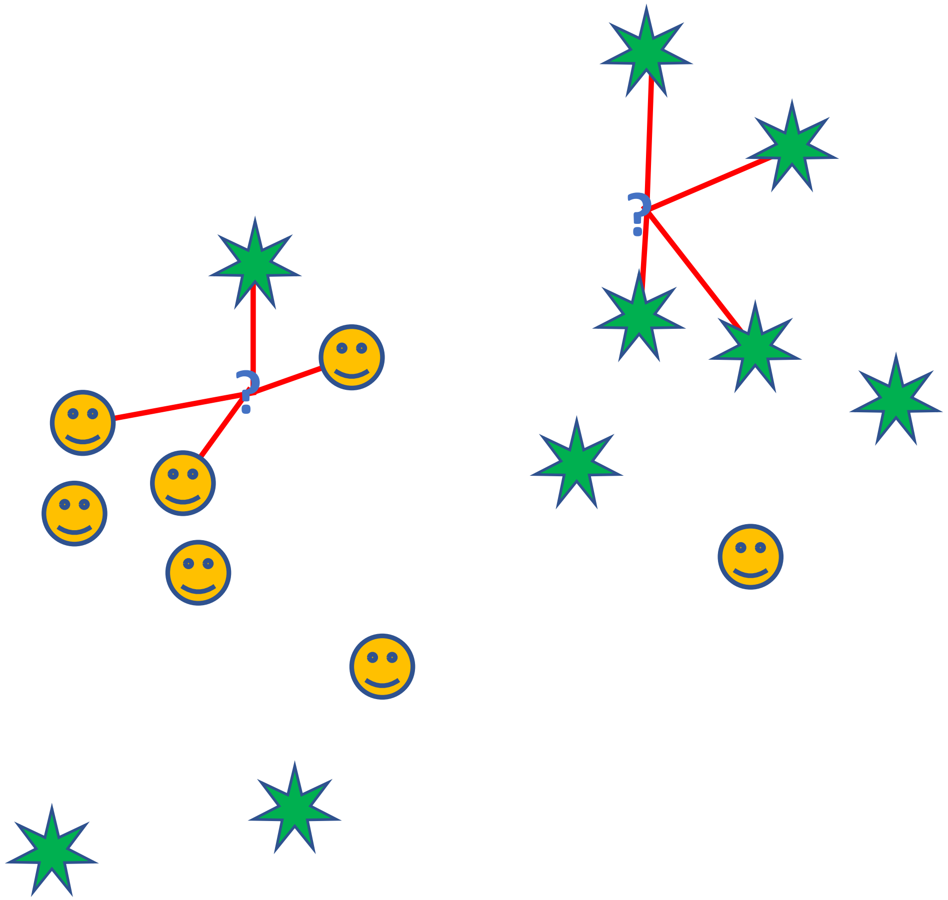
Technische Hochschule Ingolstadt

KNN refresh



KNN refresh

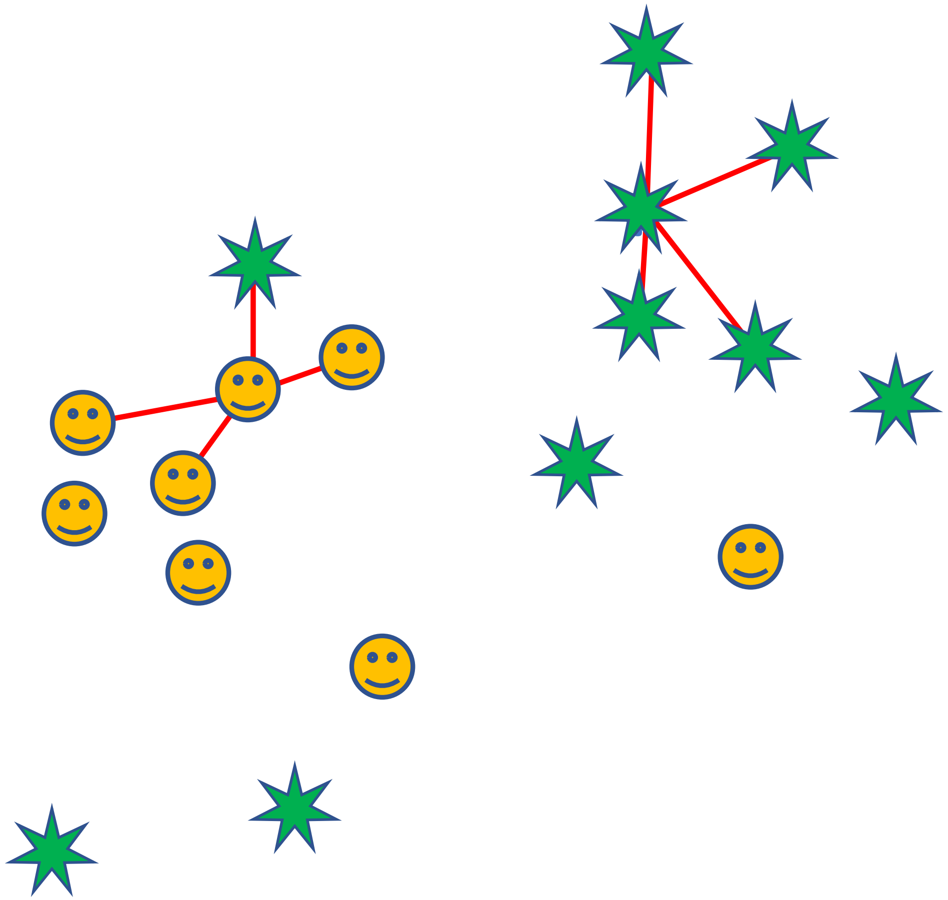
(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?



Technische Hochschule Ingolstadt

KNN refresh

(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?



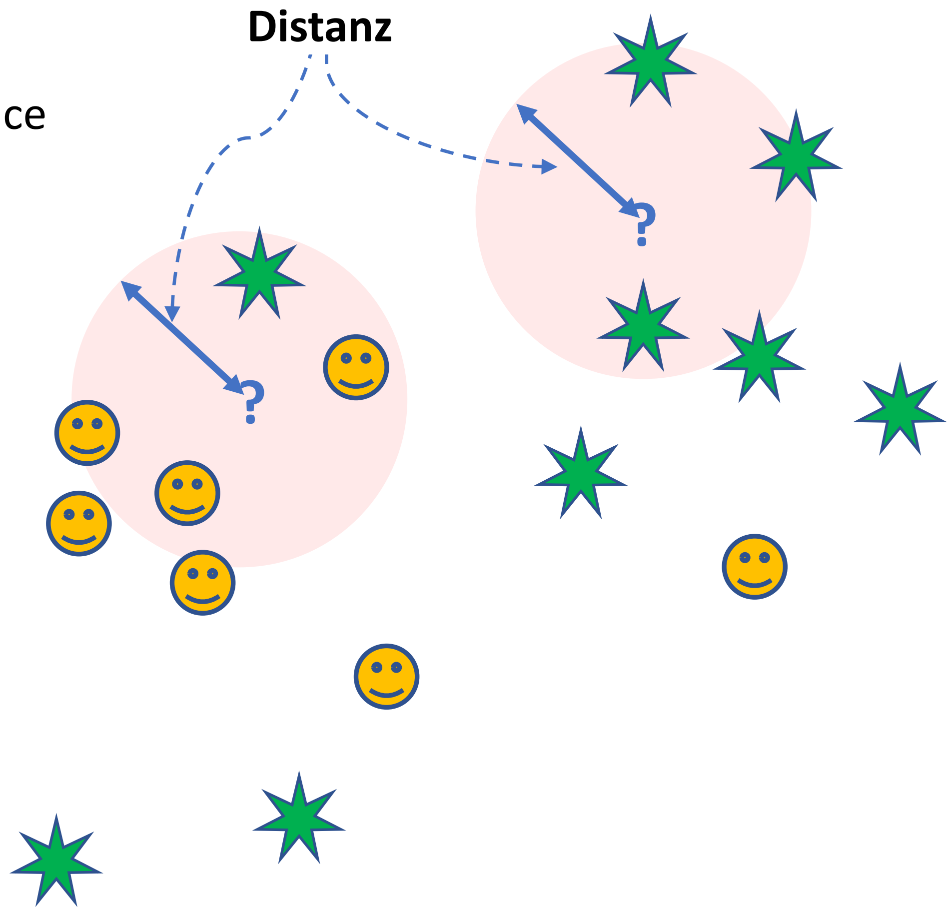
Technische Hochschule Ingolstadt

KNN refresh

(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?

Distanz:

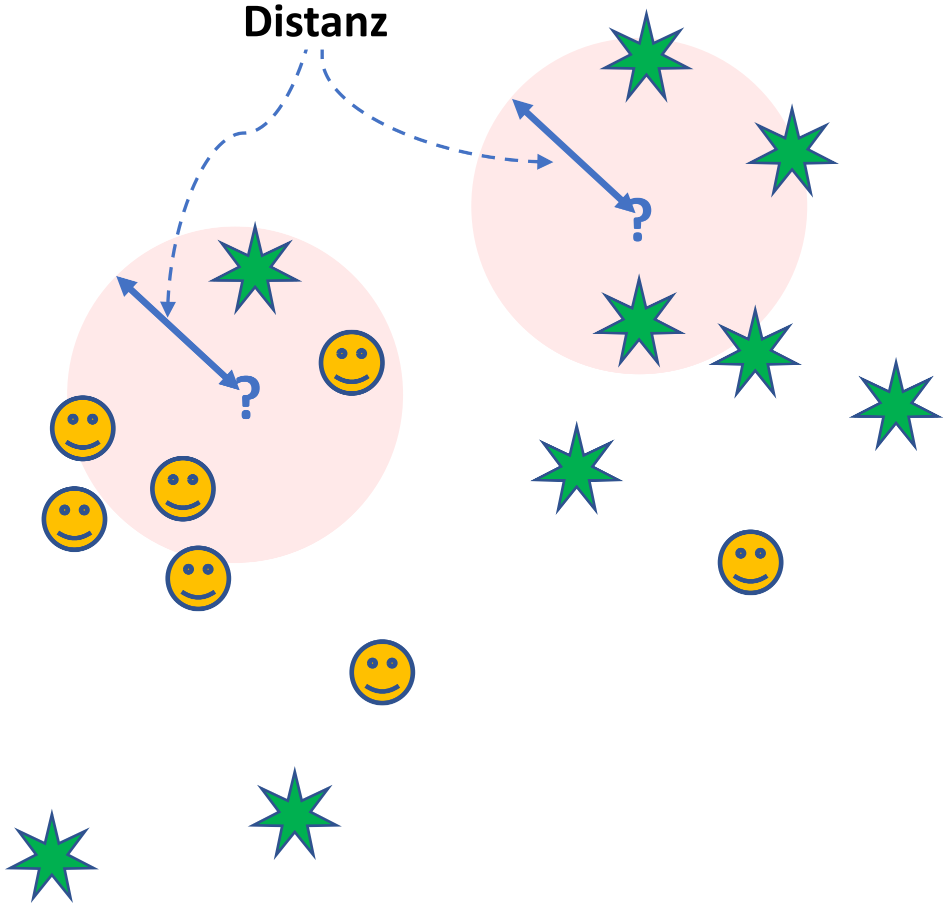
- Levenshtein/Edit distance
- Euklidischer Abstand
- …



KNN refresh

(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?

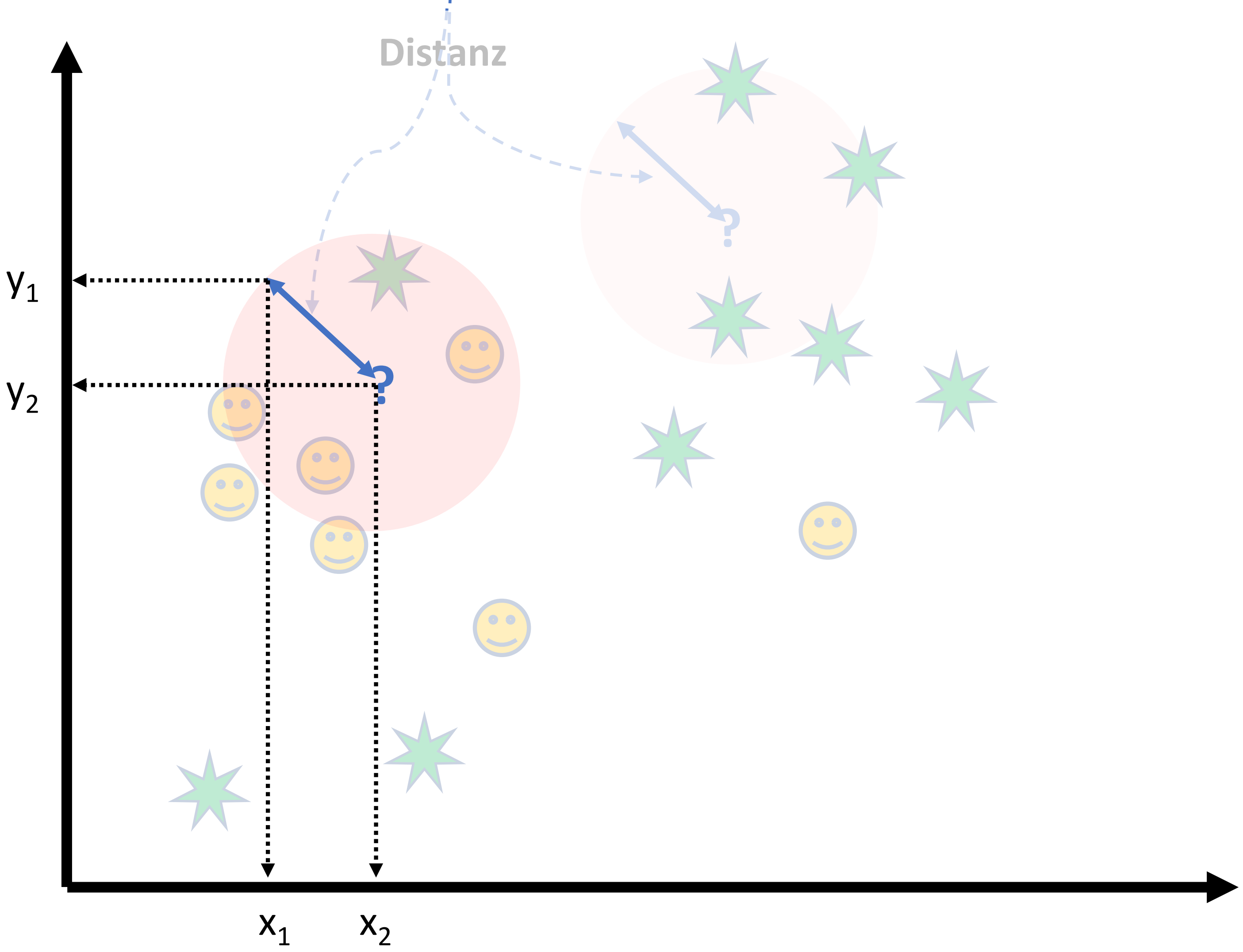
Distanz:
- ?



Technische Hochschule Ingolstadt

KNN refresh (Euklidischer Abstand)

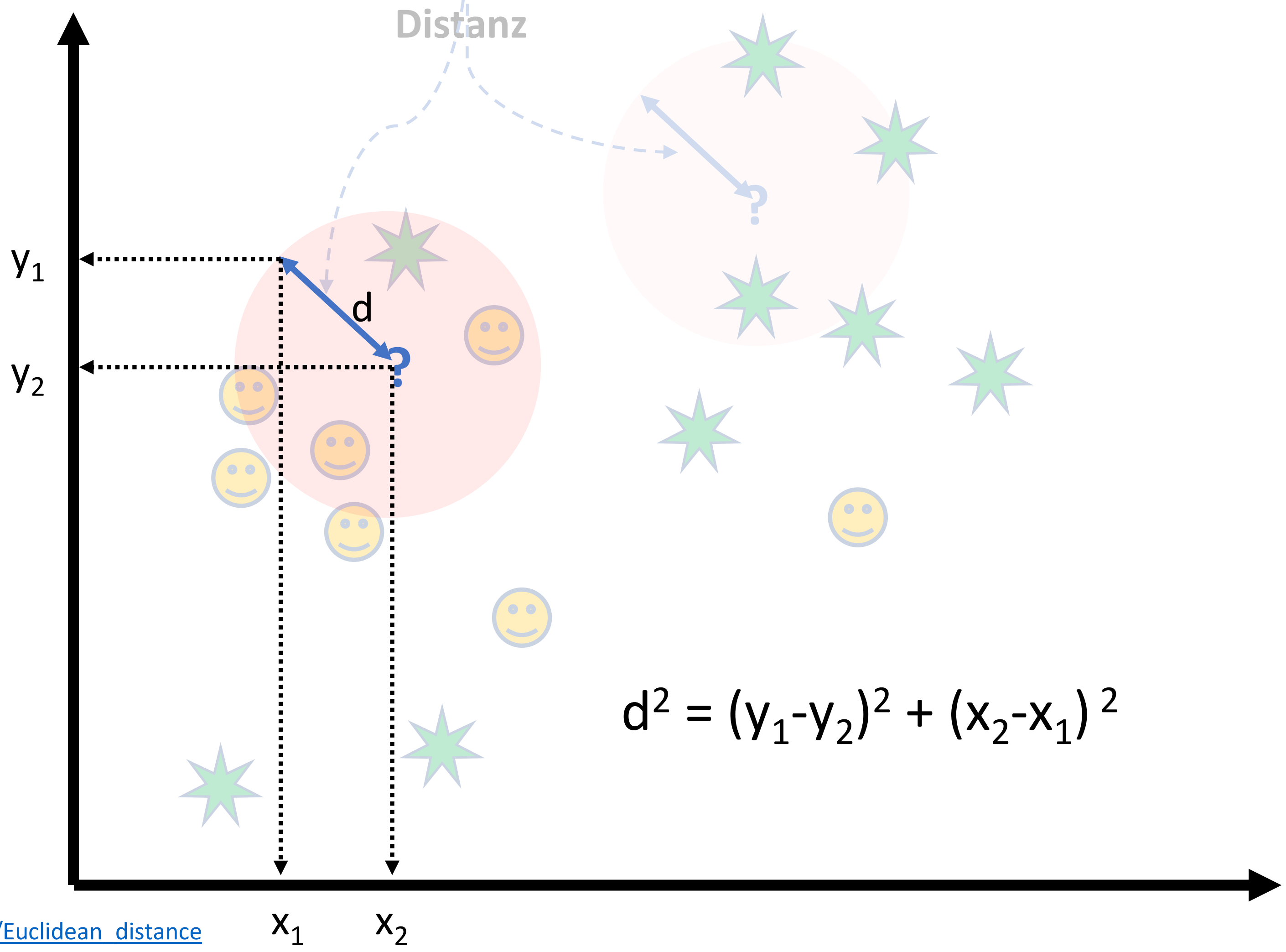
(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?



Technische Hochschule Ingolstadt

KNN refresh

(KNN mit K=4) := was sind die „4-Nächste Nachbarn“ ?



$$d^2 = (y_1 - y_2)^2 + (x_2 - x_1)^2$$

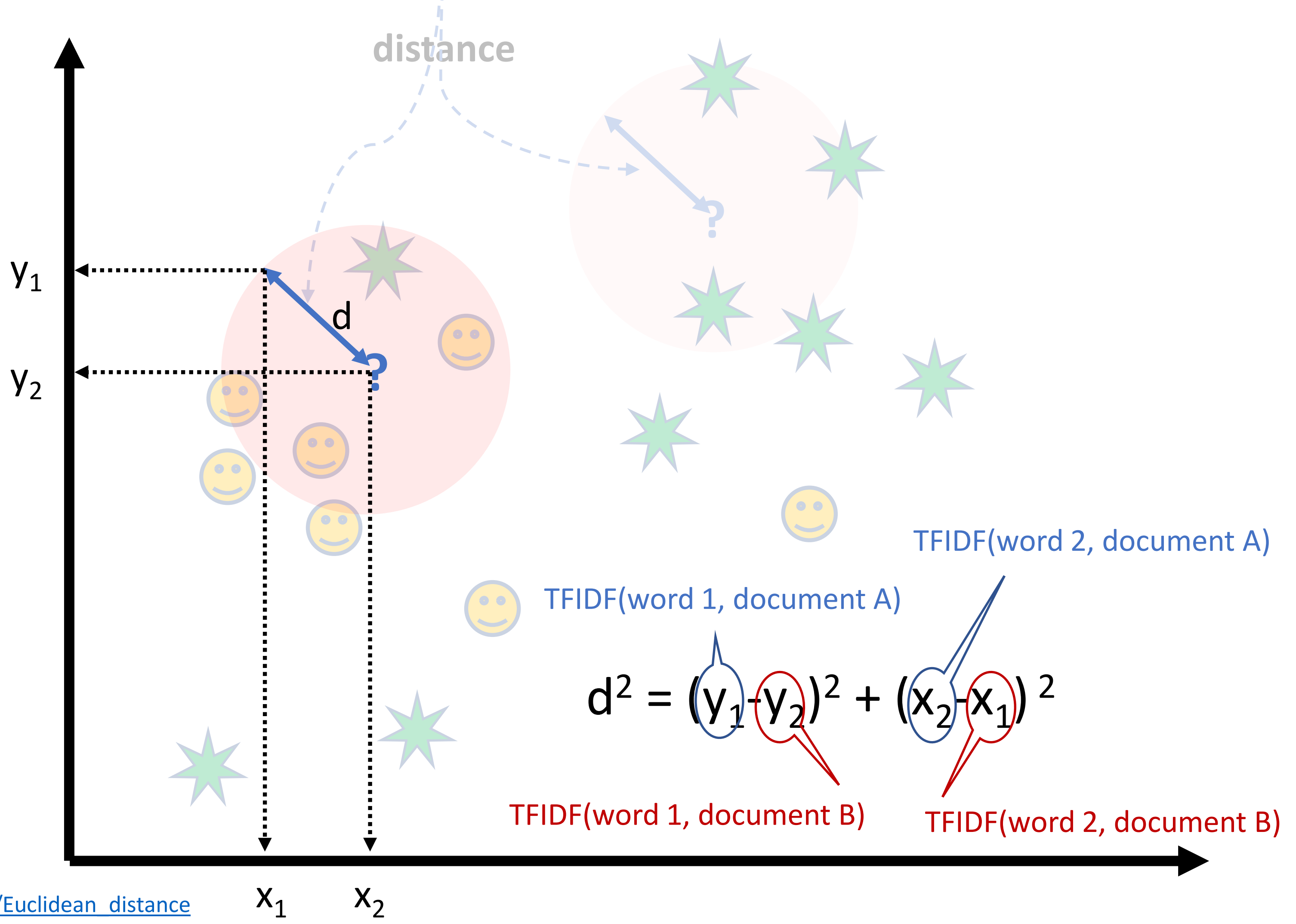
https://en.wikipedia.org/wiki/Euclidean_distance

TFIDF in Application

“Document classification With KNN”

KNN für document classification

(KNN with K=4) := what are the „4-Nearest neighbours“ ?

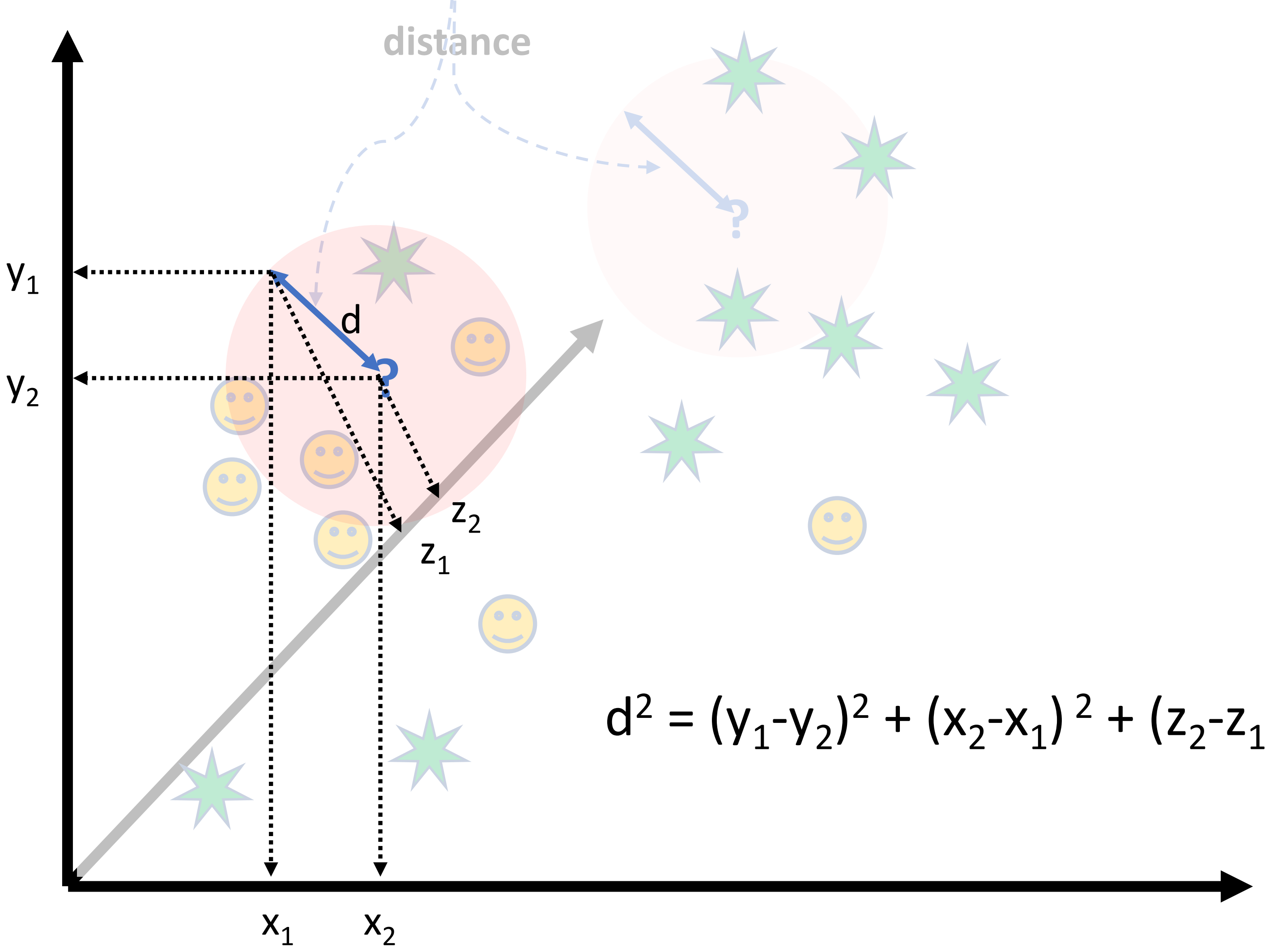


$$d^2 = (y_1 - y_2)^2 + (x_2 - x_1)^2$$

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

KNN for document classification

(KNN with K=4) := what are the „4-Nearest neighbours“ ?



$$d^2 = (y_1-y_2)^2 + (x_2-x_1)^2 + (z_2-z_1)^2$$

Mean Squared Error

$$d^2 = (y_1 - y_2)^2 + (x_2 - x_1)^2 + (z_2 - z_1)^2$$

generalization

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Mean Squared Distance between two points Y and $\hat{Y}$

$\Leftrightarrow$

Mean error of point Y w.r.t. point $\hat{Y}$

Mean Squared Error

$$d^2 = (y_1 - y_2)^2 + (x_2 - x_1)^2 + (z_2 - z_1)^2$$

generalization

TFIDF(word i, document A)

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

TFIDF(word i, document B)

Mean Squared Distance between two points Y and $\hat{Y}$

$\Leftrightarrow$

Mean error of point Y w.r.t. point $\hat{Y}$

TFIDF
in application

"Determining Stop Words"

Technische Hochschule Ingolstadt

Stop Words

SW = {d, t: TFIDF(D,t,d) < threshold }

Oft: threshold < 0.5

„SW = All t over all d for which: TFIDF(D,t,d) < threshold“

Text preprocessing:

1. sentence segmentation
2. Word segmentation (Tokenization)
3. normalization, noise reduction, etc.
4. Stemming/Lemmatization
5. Stop word Removal (using co-occurrence or TFIDF)

Technische Hochschule Ingolstadt

Text preprocessing:

1. sentence segmentation
2. Word segmentation (Tokenization)
3. normalization, noise reduction, etc.
4. Stemming/Lemmatization
5. Stop word Removal (using co-occurrence or TFIDF)

Often context-dependent, i.e. don‘t delete stop words in the early beginning…

Bemerkung: Stop Words

a
about
above
after
again
against
all
am
an
and
any
are
aren't
as
at
be
...

Bad Practice:

Use of ready-to-go lists ….

Best Practice:

Create an own list of stop words for the task => TFIDF

https://dev.mysql.com/doc/refman/8.0/en/fulltext-stopwords.html

Dirichlet distribution

“A way to model random probability mass functions for finite sets. A book of length k words can be modeled by a Dirichlet distribution with a probability mass functions of length k”

Auch bekannt als „multivariate beta distribution“.

Dirichlet-distribution

"A distribution of distributions"

$$f\left(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K\right) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

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

Dirichlet-distribution

"A distribution of distributions"

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

$$\sum_{i=1}^{K} x_i = 1$$

probability distributions

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

Dirichlet-distribution

“A distribution of distributions”

„Concentration parameter“

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

$$\sum_{i=1}^{K} x_i = 1$$

probability distributions

https://en.wikipedia.org/wiki/Concentration_parameter
https://en.wikipedia.org/wiki/Dirichlet_distribution

Dirichlet-distribution

“A distribution of distributions”

„Concentration parameter“

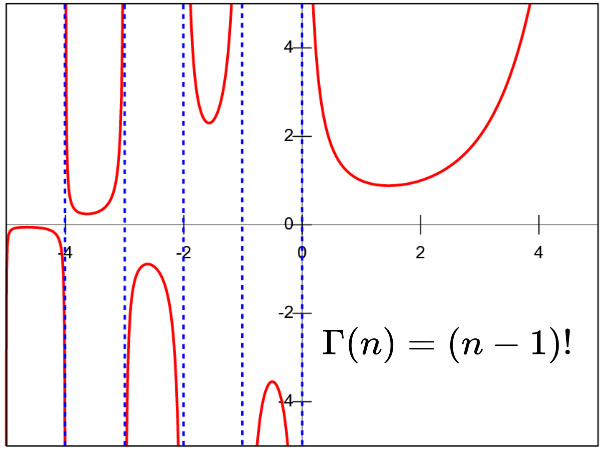
$$f(x_1,\ldots,x_K;\alpha_1,\ldots,\alpha_K) = \frac{1}{\mathrm{B}(\boldsymbol\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}$$

$$\sum_{i=1}^K x_i = 1$$

Probability distributions

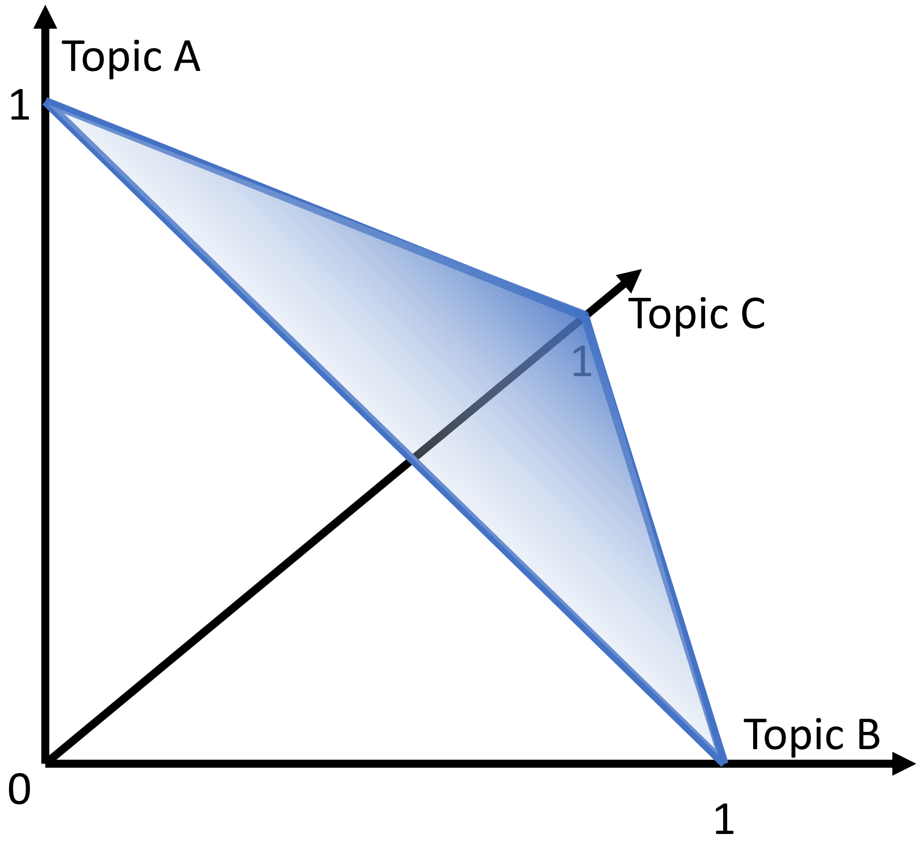
$$\mathrm{B}(\boldsymbol\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}, \qquad \boldsymbol\alpha = (\alpha_1,\ldots,\alpha_K).$$

:= Multivariate Beta Funktion



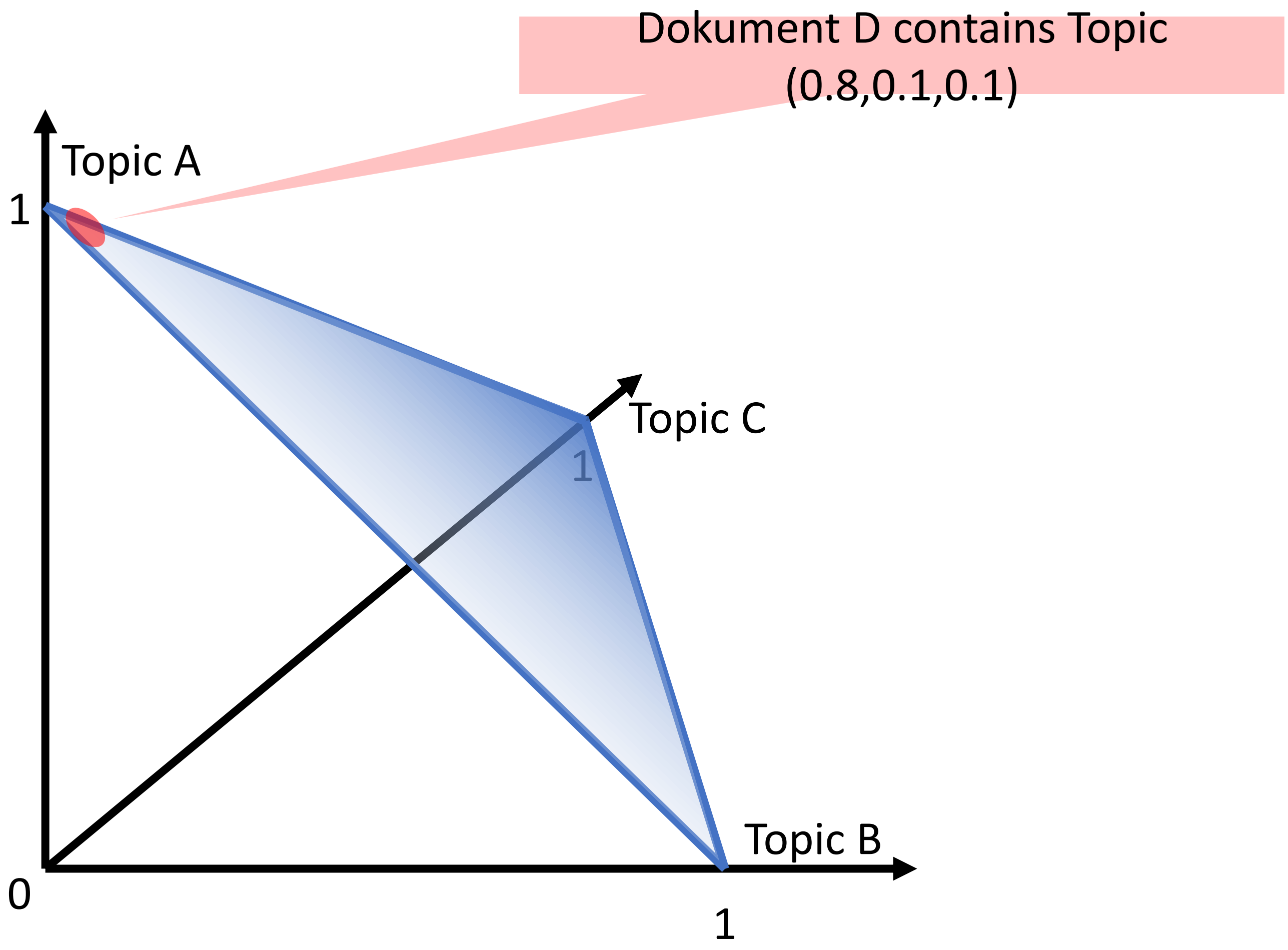
https://en.wikipedia.org/wiki/Concentration_parameter
https://en.wikipedia.org/wiki/Gamma_function
https://en.wikipedia.org/wiki/Dirichlet_distribution

Dirichlet-distribution

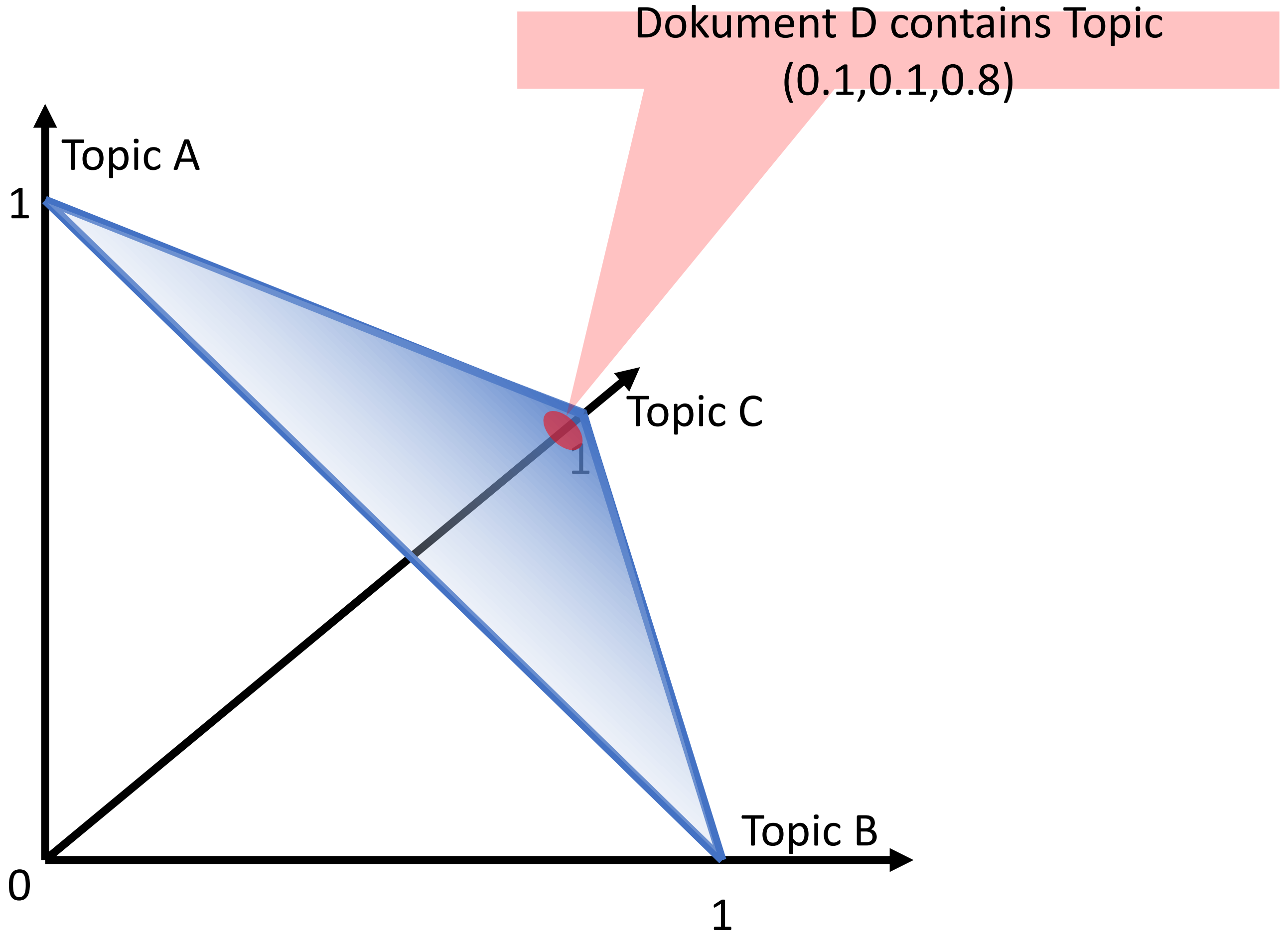


https://mc-stan.org/docs/2_24/functions-reference/dirichlet-distribution.html

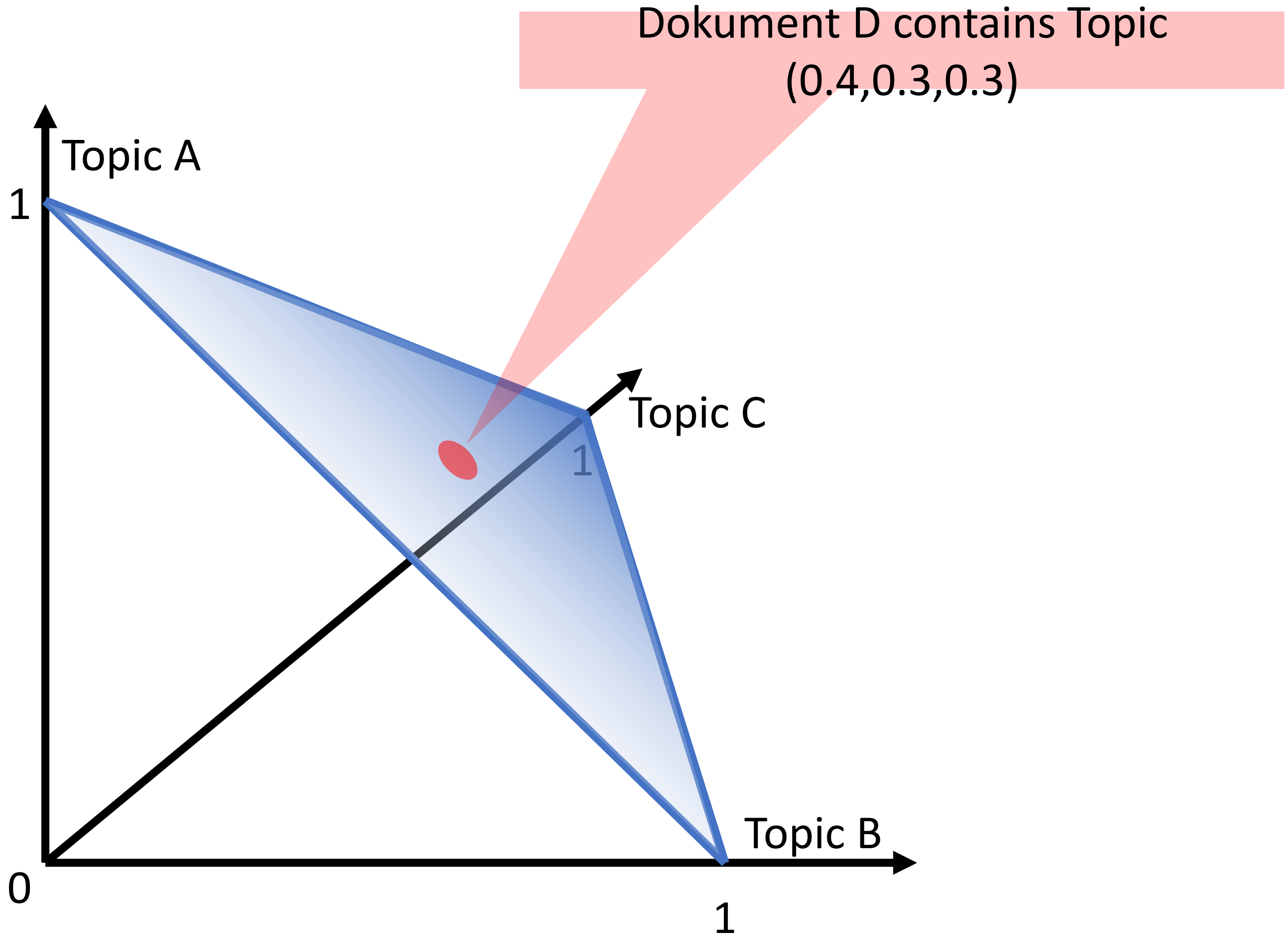
Dirichlet-distribution



Dirichlet-distribution



Dirichlet-distribution

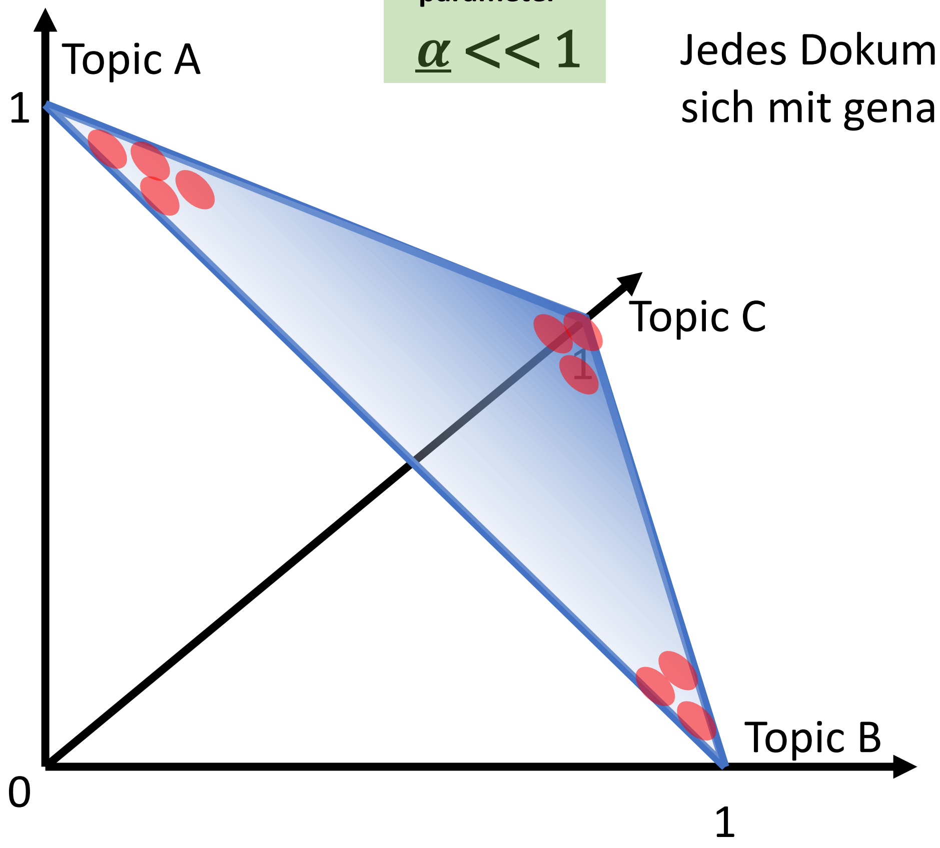


Technische Hochschule Ingolstadt

Dirichlet-distribution

„Concentration parameter“

$\underline{\alpha} << 1$

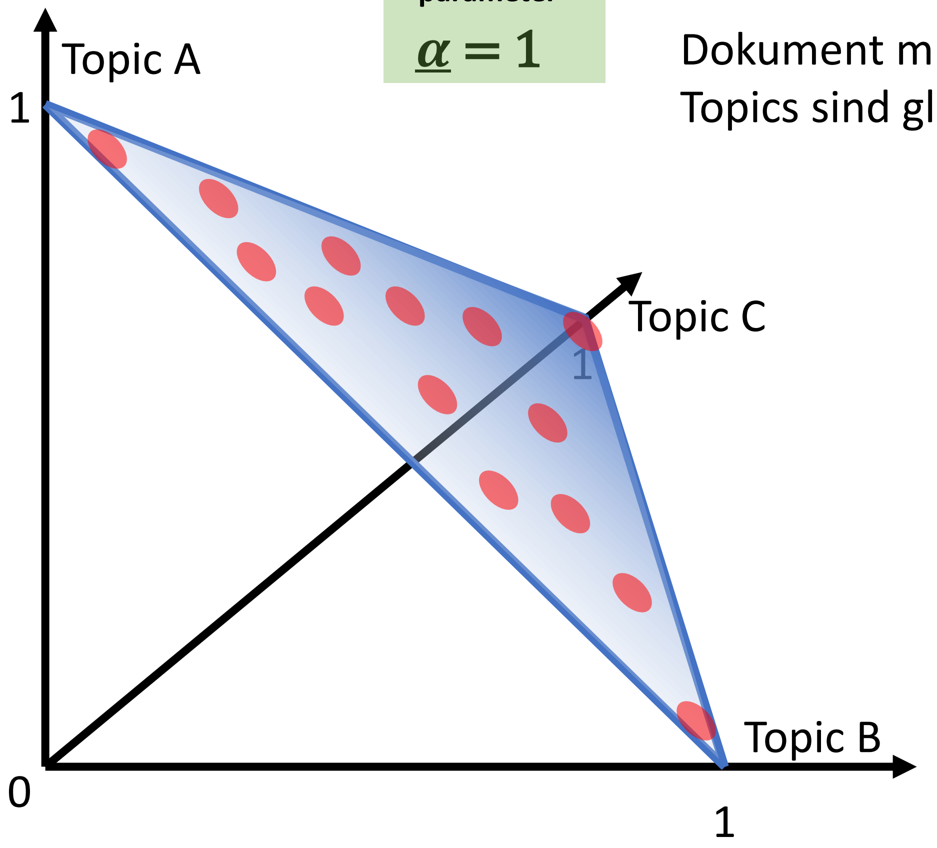


Jedes Dokument beschäftigt sich mit genau einem Topic

Dirichlet-distribution

„Concentration parameter“

$\underline{\alpha} = 1$



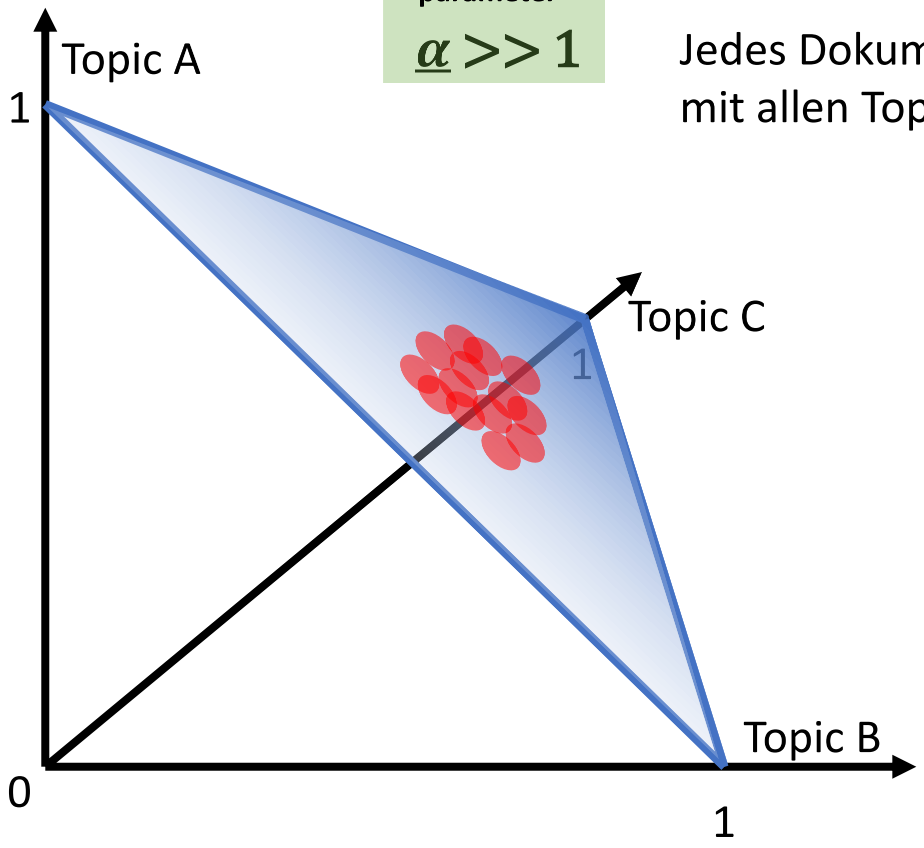
Dokument mit einem oder mehreren Topics sind gleichverteilt möglich.

Technische Hochschule Ingolstadt

Dirichlet-distribution

„Concentration parameter“

$\underline{\alpha} >> 1$

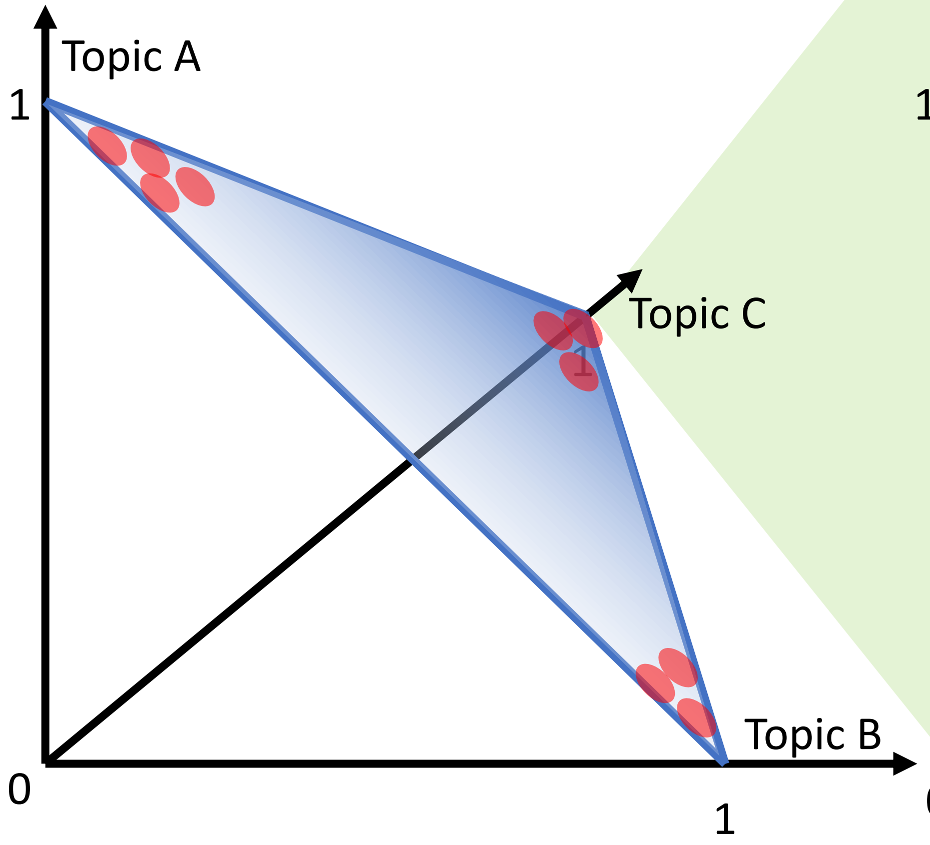


Jedes Dokument beschäftigt sich mit allen Topics gleichermaßen

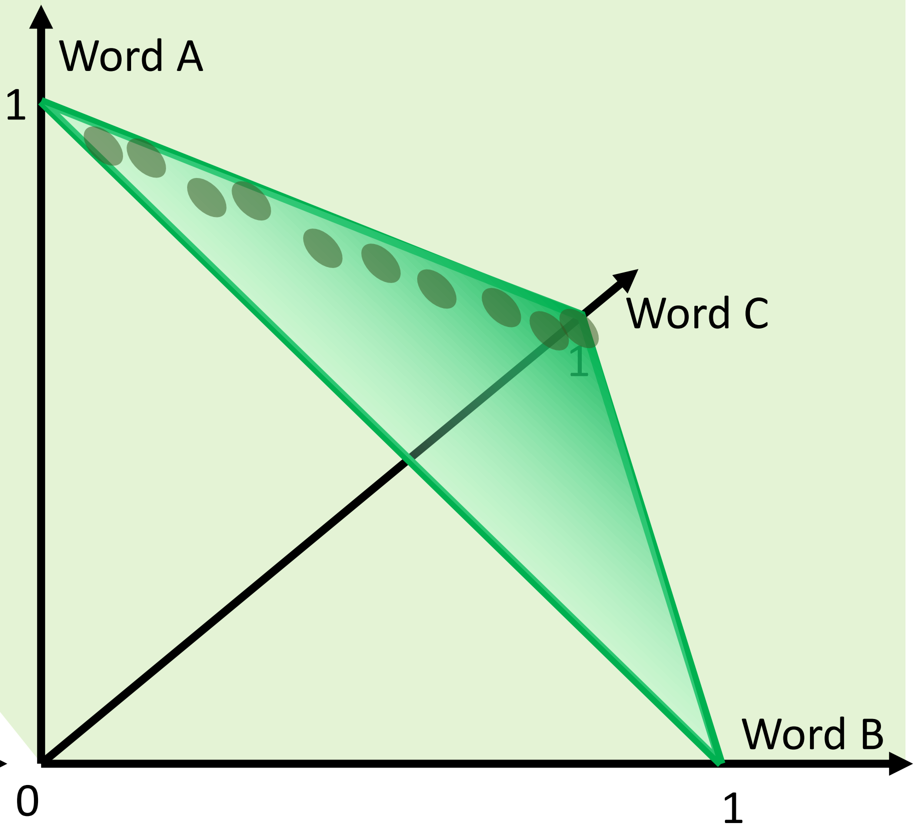
Technische Hochschule Ingolstadt

Dirichlet-distribution

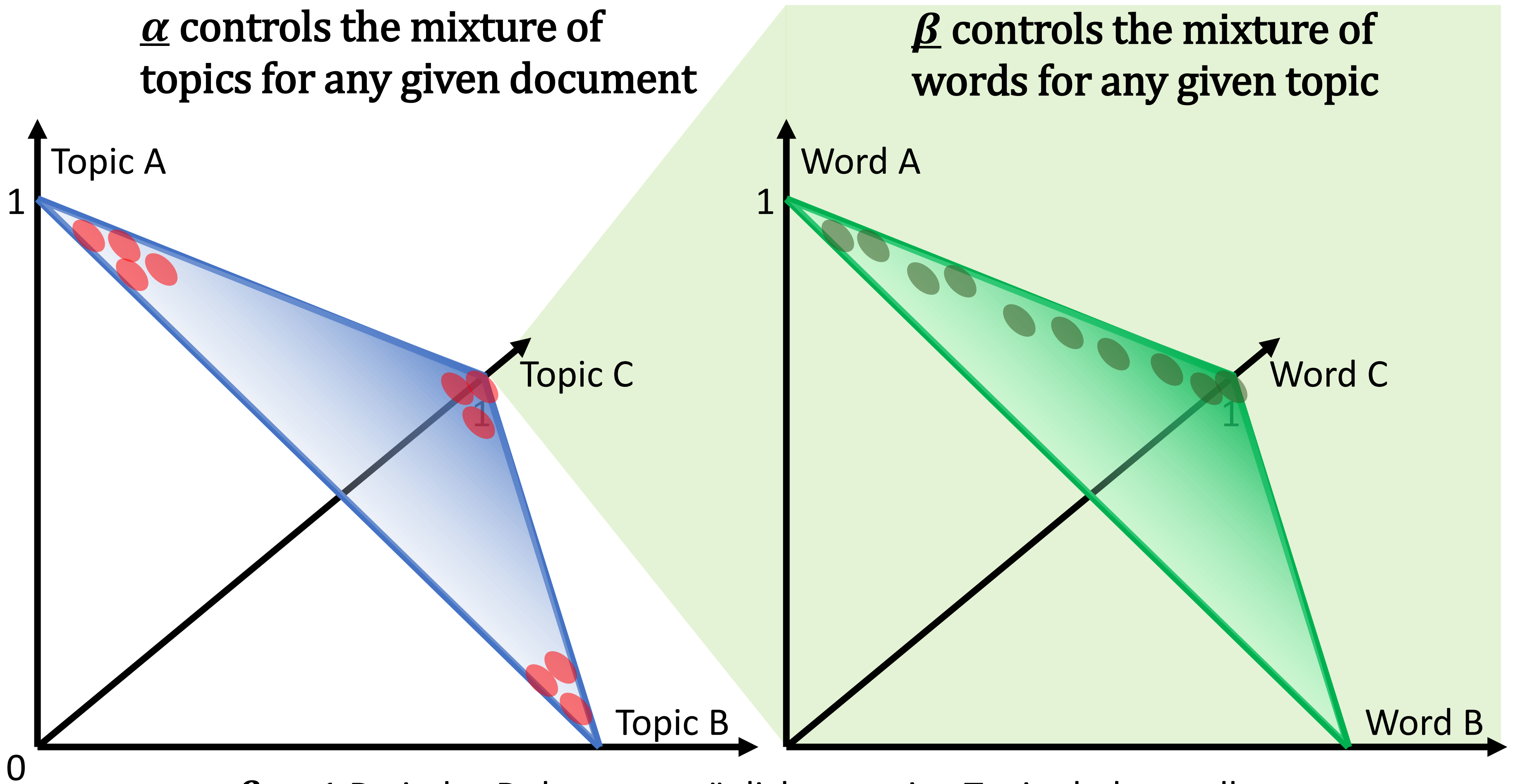
$\underline{\alpha}$ controls the mixture of topics for any given document



$\underline{\beta}$ controls the mixture of words for any given topic



Dirichlet-distribution



$\underline{\alpha, \beta}$ << 1 Da jedes Dokument möglichst wenige Topics haben sollte und jedes Wort möglichst eindeutig zugewiesen werden sollte.

BOW Term-document-Matrix

	document 1	document 2	document 3
Auto	?	?	?
Boot	?	?	?
Kuh	?	?	?
Esel	?	?	?
Tomate	?	[illegible]	?
Zwiebel	?	[illegible]	?
Traktor	?	?	?
Giraffe	?	?	?
Schwein	?	?	?
Motor	?	?	?

„Auto“ in document 1?

Technische Hochschule Ingolstadt

BOW Term-document-Matrix

	document 1	document 2	document 3
Auto	**1**	?	?
Boot	?	?	?
Kuh	?	?	?
Esel	?	?	?
Tomate	?		?
Zwiebel	?		?
Traktor	?	?	?
Giraffe	?	?	?
Schwein	?	?	?
Motor	?	?	?

„Auto“ in document 2?

BOW Term-document-Matrix

	document 1	document 2	document 3
Auto	**1**	**0**	?
Boot	?	?	?
Kuh	?	?	?
Esel	?	?	?
Tomate	?	[illegible]	?
Zwiebel	?	[illegible]	?
Traktor	?	[illegible]	?
Giraffe	?	?	?
Schwein	?	?	?
Motor	?	?	?

„Auto“ in document 2?

BOW Term-document-Matrix

	document 1	document 2	document 3
Auto	**1**	**0**	**0**
Boot	**1**	**0**	**0**
Kuh	**0**	**1**	**1**
Esel	**0**	**1**	**0**
Tomate	**0**	**0**	**1**
Zwiebel	**0**	**0**	**1**
Traktor	**1**	**0**	**1**
Giraffe	**0**	**1**	**0**
Schwein	**0**	**1**	**1**
Motor	**1**	**0**	**0**

BOW Term-document-Matrix

	document 1	document 2	document 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

Bag-of-Words (weighted Bag-Of-Words ⇔ 1-gram)

Technische Hochschule Ingolstadt

BOW Term-document-Matrix

		document 1	document 2	document 3
	Auto	1	0	0
	Boot	1	0	0
„Polysemie“	Kuh	0	1	1
	Esel	0	1	0
	Tomate	0	0	1
	Zwiebel	0	0	1
„Polysemie“	Traktor	1	0	1
	Giraffe	0	1	0
„Polysemie“	Schwein	0	1	1
	Motor	1	0	0
	Topics	tech	animals	food

Gibbs Sampling

Input: Number of topics, Corpus of documents
Output: Words assigned to topics

//Initiale Topicvergabe
for all <Documents in Corpus> **do**
  for all <Words in actual Document> **do**
    $topicForWordInDocument \leftarrow randomTopic$
  end for
end for
//Update der Themen aufgrund des Vorwissens
for all <Documents in Corpus> **do**
  for all <Words in Document> **do**
    for all <Topics> **do**
      //Verhältnis von Wörtern des Dokumentes die Topic t zugewiesen sind
      $ProportionOfWordsAssignedToTopic \leftarrow p(\text{topic t} \mid \text{document d})$ ϑ
      //Verhältnis von Zuweisungen des aktuellen Wortes an Topic t (über alle Dokumente)
      //Bedenke: w kann in verschiedenen Dokumenten anderen Topics zugewiesen sein
      $ProportionOfAssignmentToTopicOverallFromWord \leftarrow p(\text{word w} \mid \text{topic t})$ φ
    end for
    //Neue Topic t mit Wahrscheinlichkeit p zuweisen (p $\widehat{=}$ Wahrscheinlichkeit von t generierte w)
    $newTopicForWord \leftarrow wordsAssignedToTopic * topicsAssignedToWord$
  end for
end for

„An algorithm, that helps generating a sequence of samples of a joint probability distribution of two or more random variables.

Goal: approximate the unknown joint distribution.“

https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/schoeneberg/bericht.pdf
https://en.wikipedia.org/wiki/Gibbs_sampling

Latent Dirichlet Allocation

"A generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar."

Proposed to find population genetics by J. K. Pritchard, M. Stephens and P. Donnelly in 2000. Applied to document classification by David Blei, Andrew Ng und Michael I. Jordan in 2003.

Latent Semantic Analysis

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

=

Bag-of-Words
N-grams
TFIDF Scores
…
und viele mehr
…

https://en.wikipedia.org/wiki/Latent_semantic_analysis#Term-document_matrix

Topic Model

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

A =

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

φ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ

Bag-of-Words
N-grams
TFIDF Scores
…
und viele mehr
…

φ := phi
ϑ := theta

Topic Model

Topics

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

$$A = \varphi \, \vartheta$$

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

φ („**words/n-grams** vs. **topics**")

ϑ („**documents** vs. **topics**" matrix)

φ := phi
ϑ := theta

Topic Model

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

A =

Topics

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

φ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ

φ („**words/n-grams** vs. **topics**")

ϑ („**documents** vs. **topics**" matrix)

φ := phi
ϑ := theta

Topic Model

Topics

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

A =

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

φ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ

ϑ („**documents** vs. **topics**“ matrix)

φ („**words/n-grams** vs. **topics**“)

- **Latent Dirichlet Allocation**
- “a” Matrix Factorization
- Singular Value Decomposition
- Tensor Factorization with Stochastic Gradient Descent

φ := phi
ϑ := theta

Topic Model

Topics

	Dokument 1	Dokument 2	Dokument 3
Auto	1	0	0
Boot	1	0	0
Kuh	0	1	1
Esel	0	1	0
Tomate	0	0	1
Zwiebel	0	0	1
Traktor	1	0	1
Giraffe	0	1	0
Schwein	0	1	1
Motor	1	0	0

A =

	Topic 1	Topic 2
Auto	1	0
Boot	1	0
Kuh	0	1
Esel	0	1
Tomate	0	0
Zwiebel	0	0
Traktor	1	0
Giraffe	0	1
Schwein	0	1
Motor	1	0

φ

	Dokument 1	Dokument 2	Dokument 3
Topic 1	1	0	0
Topic 2	1	0	0

ϑ

ϑ („**documents** vs. **topics**“ matrix)

φ („**words/n-grams** vs. **topics**“)

Bag-of-Words
N-grams
TFIDF Scores
…
And many more
…

- **Latent Dirichlet Allocation**
- Tensor Factorization with Stochastic Gradient Descent

φ := phi
ϑ := theta

LDA „parameters”

- φ := „words/n-grams vs. topics“ matrix
- ϑ := „docı

	Topic 1	Topic 2	Topic 3
Word 1	213	67	4
Word 2	2	145	89
Word 3	2	23	75
Word 4	12	23123	2
Word 5	567	7	5
Word 6	55	234	5

Problem: we don‘t know which and how many topics exist

φ := phi
ϑ := theta

http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/schoeneberg/bericht.pdf

LDA „parameters"

- φ := **„words/n-grams vs. topics" matrix**
- ϑ := **„doc**

Dirichlet distribution

	Topic 1	Topic 2	Topic 3
Word 1	213	67	4
Word 2	[illegible]	[illegible]	[illegible]
Word 3	2	[illegible]	[illegible]
Word 4	12	[illegible]	2
Word 5	567	[illegible]	[illegible]
Word 6	55	234	5

Sample from data using $\boldsymbol{\beta}$ (e.g. using Gibbs) and then compute probabilities

φ := phi
ϑ := theta

http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/schoeneberg/bericht.pdf

LDA „parameters”

- φ := „**words/n-grams** vs. **topics**“ **matrix**
- ϑ := „**documents** vs. **topics**“ **matrix**

	Topic 1	Topic 2	Topic 3
Dokument 1	2	5	8
Dokument 2	31	564	6
Dokument 3	4	6	48
Dokument 4	2	2	2
Dokument 5	6	7	3
Dokument 6	4	234	63

Problem: we don‘t know which and how many topics exist

φ := phi
ϑ := theta

http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-

LDA „parameters”

- φ := „**words/n-grams** vs. **topics**“ **matrix**
- ϑ := „**documents** vs. **topics**“ **matrix**

Dirichlet distributio n

	Topic 1	Topic 2	Topic 3
Dokument 1	2	5	8
Dokument 2	[illegible]	[illegible]	[illegible]
Dokument 3	4	[illegible]	43
Dokument 4	2	[illegible]	2
Dokument 5	6	7	[illegible]
Dokument 6	4	234	63

Sample from data using α (e.g. using Gibbs) and then compute probabilities

φ := phi
ϑ := theta

http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf
https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2014-aw2/schoeneberg/bericht.pdf

The LDA algorithm

1. **Define vocabulary (words) or n-grams**
2. **Set number of documents**
3. **Set number of words per document**
4. **Set number of topics**

http://173.236.226.255/tom/papers/SteyversGriffiths.pdf
http://webdoc.sub.gwdg.de/pub/mon/dariah-de/dwp-2016-18.pdf
https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d

The LDA algorithm

1. **Define vocabulary (words) or n-grams**
2. **Set number of documents**
3. **Set number of words per document**
4. **Set number of topics**
5. **Set β= 0.01**
6. **Set α = 0.5**

$\left.\begin{array}{l}\beta = 0.01\\ \alpha = 0.5\end{array}\right\}$ Dirichlet-distribution

The LDA algorithm

1. **Define vocabulary (words) or n-grams**
2. **Set number of documents**
3. **Set number of words per document**
4. **Set number of topics**
5. **Set $\underline{\beta}$ = 0.01**
6. **Set $\underline{\alpha}$ = 0.5**

Dirichlet distribution

Set Hyperparameters

We suppose, that Topics-documents resp. Words-Topics follow a **Dirichlet distribution**

The LDA algorithm

1. **Define vocabulary (words) or n-grams**
2. **Set number of documents**
3. **Set number of words per document**
4. **Set number of topics**
5. **Set $\underline{\boldsymbol{\beta}}$= 0.01**
6. **Set $\underline{\boldsymbol{\alpha}}$ = 0.5**
7. **Estimate φ using $\underline{\boldsymbol{\beta}}$ („words/n-grams vs. topics“ matrix)**
8. **Estimate ϑ using $\underline{\boldsymbol{\alpha}}$ („documents vs. topics“ matrix)**
9. **For each Document**

Dirichlet distribution

Set Hyperparameters

Initialize the probabilistic LDA model

The LDA algorithm

1. **Define vocabulary (words) or n-grams**
2. **Set number of documents**
3. **Set number of words per document**
4. **Set number of topics**
5. **Set $\underline{\beta}$= 0.01**
6. **Set $\underline{\alpha}$ = 0.5**
7. **Estimate φ using $\underline{\beta}$ („words/n-grams vs. topics“ matrix)**
8. **Estimate ϑ using $\underline{\alpha}$ („documents vs. topics“ matrix)**
9. **For each Document**
   1. look up its row in the „documents vs. topics“ matrix
   2. sample a topic based on the probabilities in the row
   3. go to the „words/n-grams vs. topics“ matrix
   4. look up the topic sampled
   5. sample a word/n-gram based on the probabilities in the column
   6. repeat from step 2 until you’ve reached how many words/n-grams this document was set to have

Set Hyperparameters

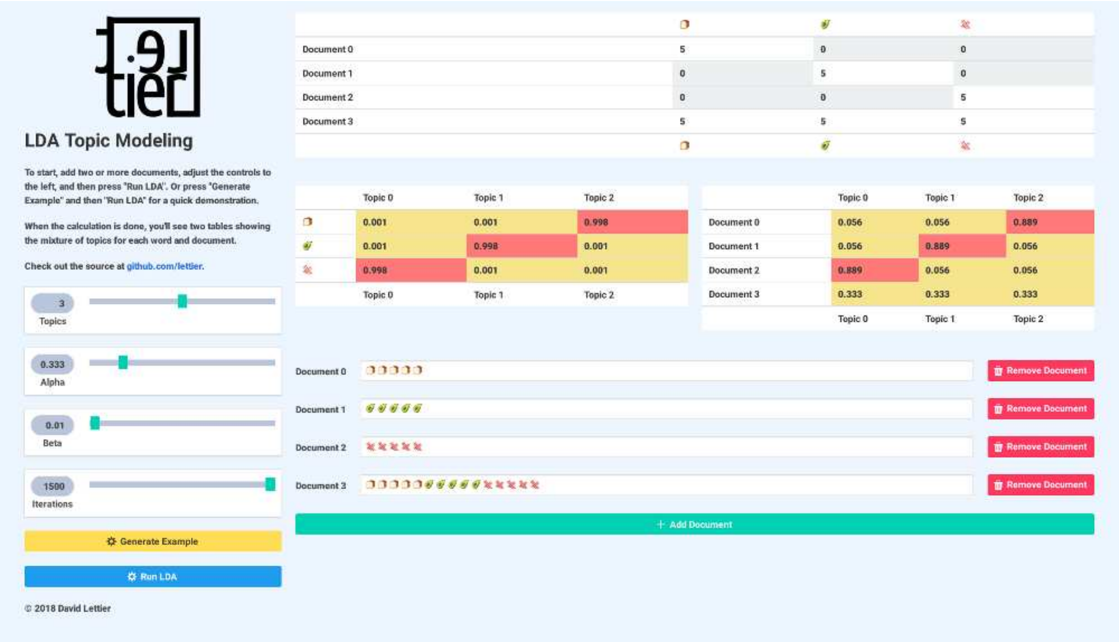
Dirichlet-Verteilung

Train the probabilistic LDA model

Let‘s try it out

https://lettier.com/projects/lda-topic-modeling/

https://github.com/lettier/lda-topic-modeling/



LDA not to be confused with LDA!

Latent Dirichlet Allocation

$$\neq$$

Linear Discriminant Analysis

Bayes-Dokument-LDA-Klassifikator

Topics aka „latent vector“

$$D(\underline{t}) = \mathrm{argmax}_{d=\{\ldots\}} P(d|LDA(\underline{t}))$$

$$= \mathrm{argmax}_{d=\{\ldots\}} P(LDA(\underline{t})|d)P(d)$$

$$P(LDA(\underline{t})|d) = P_d(LDA(\underline{t}))$$

$$P(LDA(\underline{t})|d) = P_d(LDA(F(\underline{t})))$$

$F(\underline{w})$:= Text preprocessing

- Tokenization
- Character Set
- Punctuation
- ….

LDA
in der Anwendung

“Topic Modeling”

Topic Modeling mit LDA

$$\mathrm{LDA}(\underline{t}) := P(\mathrm{topic} \mid \underline{t})$$

?

Topic Modeling mit LDA

$$\mathrm{LDA}(\underline{t}) := P(\mathrm{topic}|\underline{t})$$

⇧

Wahrscheinlichkeitsverteilung
der „Topics“ im Dokument $\underline{t}$

#Topics ist ein Hyperparameter

Topic Modeling mit LDA

$$\mathrm{LDA}(\underline{t}) := P(\mathrm{topic} \mid \underline{t})$$

⇧

Wahrscheinlichkeitsverteilung
der „Topics“ im Dokument $\underline{t}$

#Topics ist ein Hyperparameter

Topic Modeling mit LDA

$$\mathrm{LDA}(\underline{t}) := P(\mathrm{topic}|\underline{t})$$

Wahrscheinlichkeitsverteilung der „Topics“ im Dokument $\underline{t}$

#Topics ist ein Hyperparameter

aka LDA does unsupervised clustering