

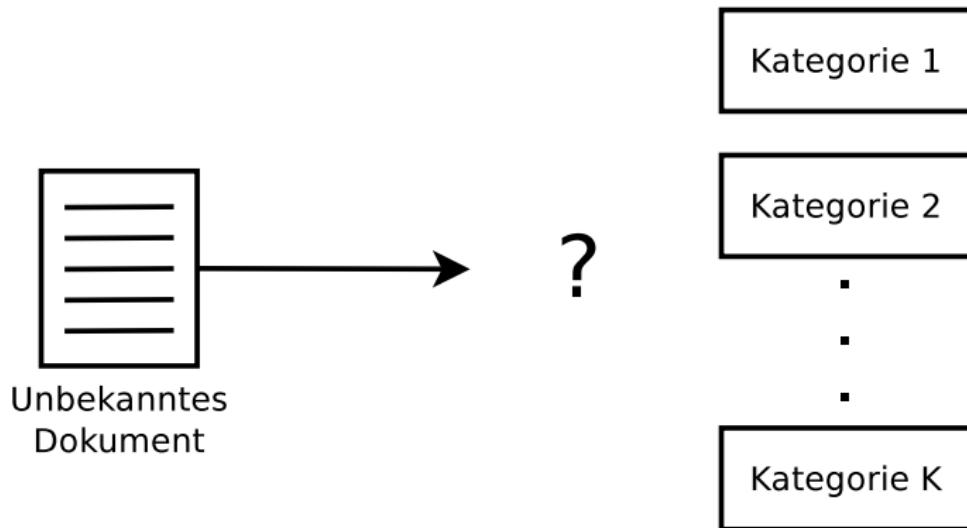
Fachprojekt Dokumentenanalyse

Fabian Wolf Gernot A. Fink

Arbeitsgruppe Mustererkennung,
Lehrstuhl 12, Technische Universität Dortmund

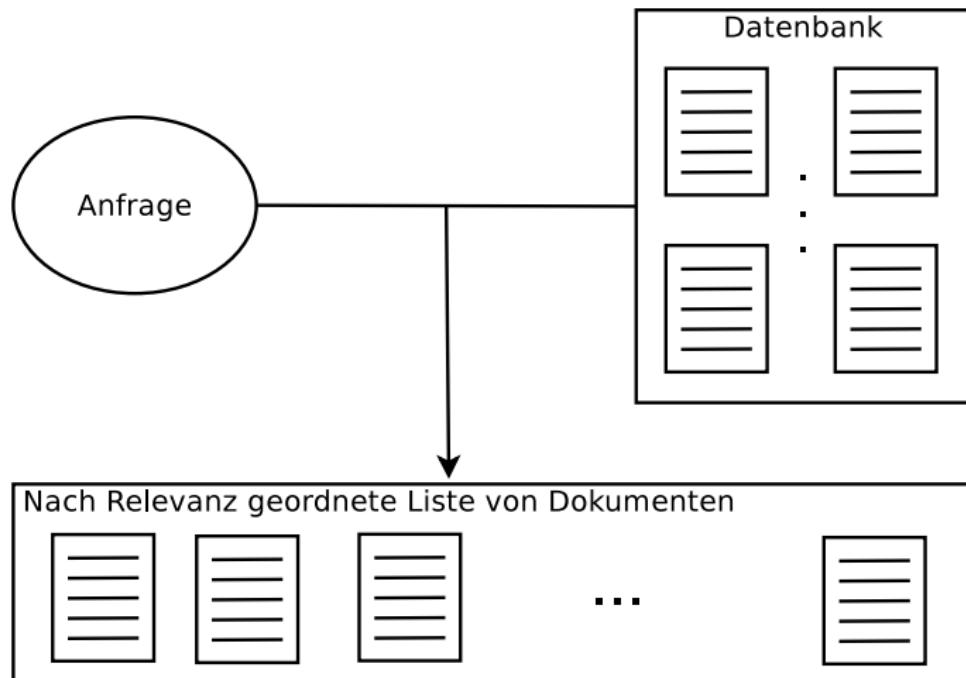
Wintersemester 2018/2019

Kategorisierung von Dokumenten



Kategorisierung eines unbekannten Dokuments zu *einer* Kategorie.

Retrieval von Dokumenten



Suche nach relevanten Dokumenten in einer Datenbank.

Maschinenlesbare Texte

Automatische semantische Analyse von Texten

“Genetics”	“Evolution”	“Disease”	“Computers”
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers

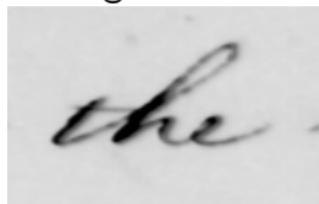
Abbildung: Beispiel für *typische* Wörter verschiedener Kategorien.

Anwendungen im Internet:

- ▶ Kategorisierung: Facebook, Twitter
- ▶ Retrieval: Google

Visuelle Analyse von Dokumentenabbildern

Anfragebild



Word Spotting Ergebnis

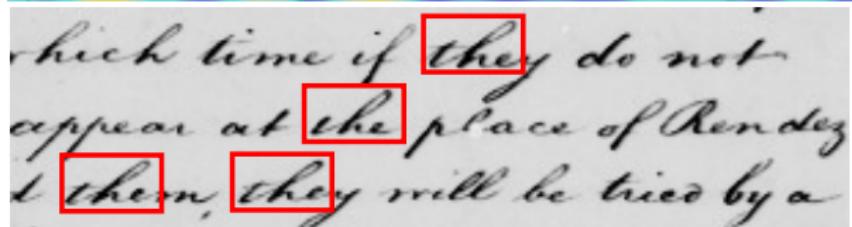


Abbildung: Beispiel für Retrieval von Bildausschnitten.

Anwendungen:

- ▶ Analyse historischer Dokumente (siehe oben)
- ▶ Klassifikation gescannter Geschäftsbriefe

Inhaltlicher Ablauf des Fachprojekts

Aufgaben zur Analyse maschinenlesbarer Texte

- ▶ Konkret formulierte aufeinander aufbauende Aufgaben
- ▶ 2 bis 3 Personen
- ▶ Woche 1 - Woche 10 / 11

Projekte aus dem Bereich der bildbasierten Dokumentenanalyse

- ▶ Projektbeschreibung gibt nur den thematischen Rahmen vor
- ▶ bis zu 5 Personen
- ▶ ab Woche 11 / 12
- ▶ Abschlusspräsentation in Woche 15

Umsetzung der Aufgaben erfolgt in Python.

Geplanter Ablauf

Aufgabe	Zeitraum	Inhalt
intro	Woche 1 / 2	Einführung
aufgabe1	Woche 2 / 3	Brown Corpus
aufgabe2	Woche 4 / 5	Bag-of-Words, Klassifikation
aufgabe3	Woche 5 / 6	Kreuzvalidierung, Term Weighting
aufgabe4	Woche 6 / 7	Unterräume
aufgabe5	Woche 8 / 9	statistische Klassifikation (optional)
aufgabe6	Woche 10 / 11	Bag-of-Features (Bilddomäne)
Projektphase	Wochen 12 - 15	bildbasierte Dokumentenanalyse

Organisatorisches

- ▶ Regelmäßiges Treffen (Anwesenheitspflicht) Dienstags von 14 Uhr bis 18 Uhr in der Otto-Hahn-Str. 16 Raum U08 (CI-Lab)
- ▶ Modulprüfung: Erfolgreiche Bearbeitung der Aufgaben, aktive Mitarbeit und Präsentation der Ergebnisse der Projektarbeit
- ▶ Abbruch des Fachprojekts zählt als Fehlversuch

Arbeiten im Rechner-Pool OH16 U08

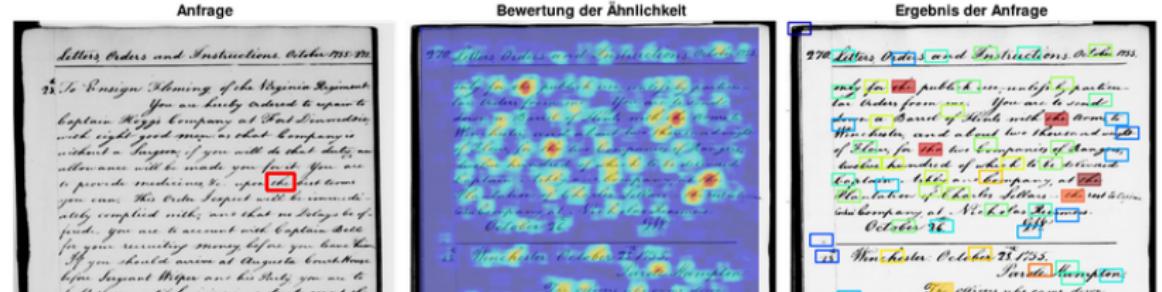
- ▶ Starten der Rechner.
- ▶ Laden der VM *PatRec-Tutorials*.
- ▶ Login mit persönlichem Account **GNOME Classic**.
- ▶ Home-Verzeichnisse sind auf allen Rechnern verfügbar (NFS).
- ▶ Zugang zum Internet funktioniert über Proxy
System Settings – Network – Proxy – Manual
proxy.cs.tu-dortmund.de:3128
- ▶ Am Ende der Sitzung bitte VM und Rechner herunterfahren.

Materialien

http://patrec.cs.tu-dortmund.de/cms/en/home/Teaching/WS18/FP_Dokumentenanalyse/index.html

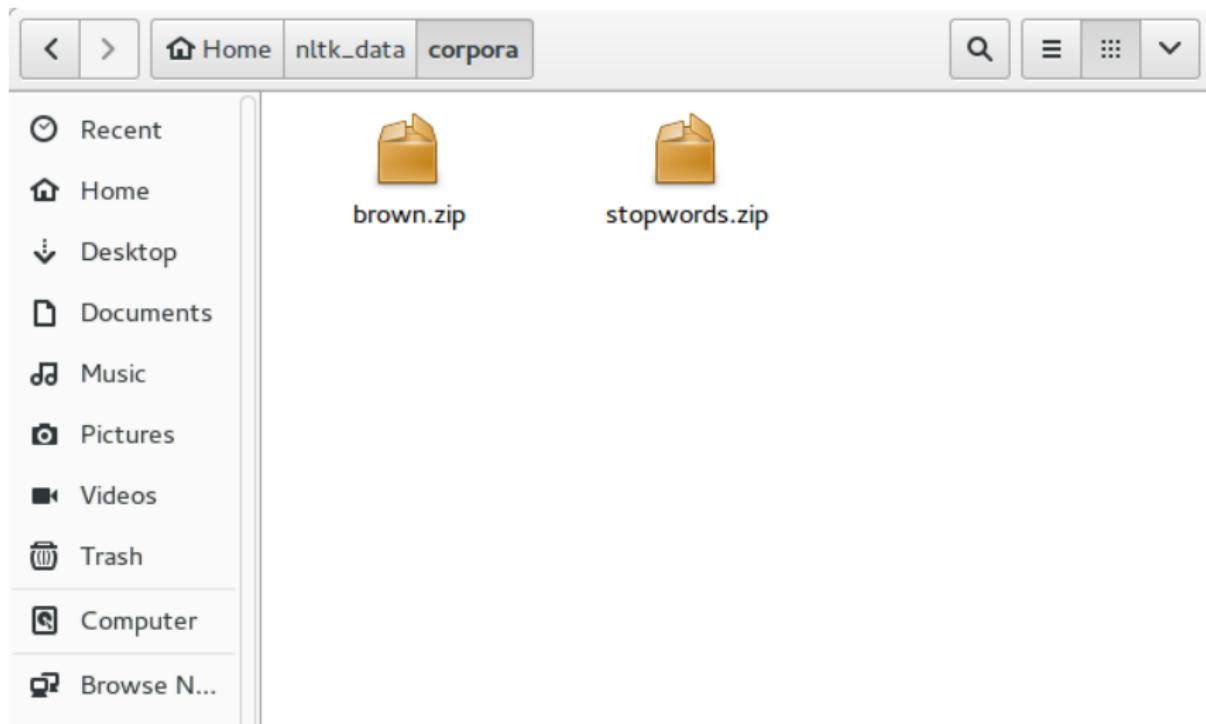
- ▶ document-analysis.pdf (Folien),
document-analysis_data.zip (Text Corpora),
aufgaben.zip (Eclipse Projekt)
- ▶ Nutzer: *da2018*, Passwort: *washington*

Durch die Verallgemeinerung dieses Konzepts zu Bag-of-Features lässt sich die Vorgehensweise auch auf Bilder anwenden. Ein Wort entspricht dabei einem lokalen Bilddeskriptor. In dem in Abbildung 2 gezeigten Beispiel sollen alle Vorkommen von "the" gefunden werden. Ein Nutzer selektiert dazu exemplarisch ein Vorkommen (links). Über den Bag-of-Features Ansatz lassen sich nun Bildbereiche identifizieren, die der Anfragebildregion visuell ähnlich sind (mitte). Schließlich kann dem Nutzer das Ergebnis als sortierte Liste relevanter Bildbereiche angezeigt werden (rechts). Dies ist konzeptionell äquivalent zu der Verwendung einer Suchmaschine im Internet.



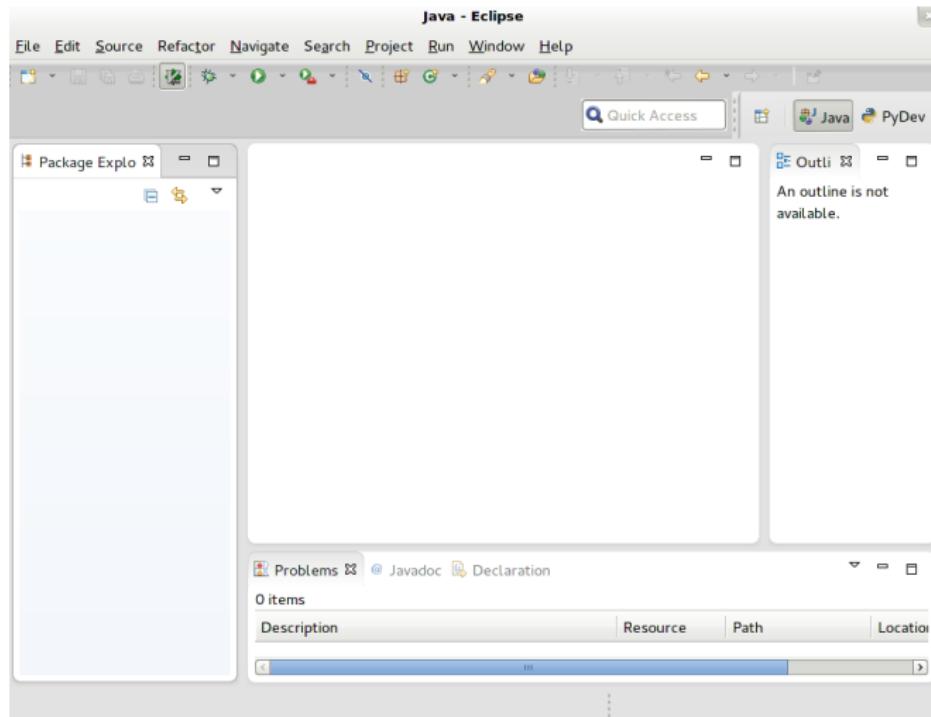
Vorbereitungen: Daten

Entpacken Sie *document-analysis-data.zip* in Ihr *Home* Verzeichnis:



Vorbereitungen: Aufgaben

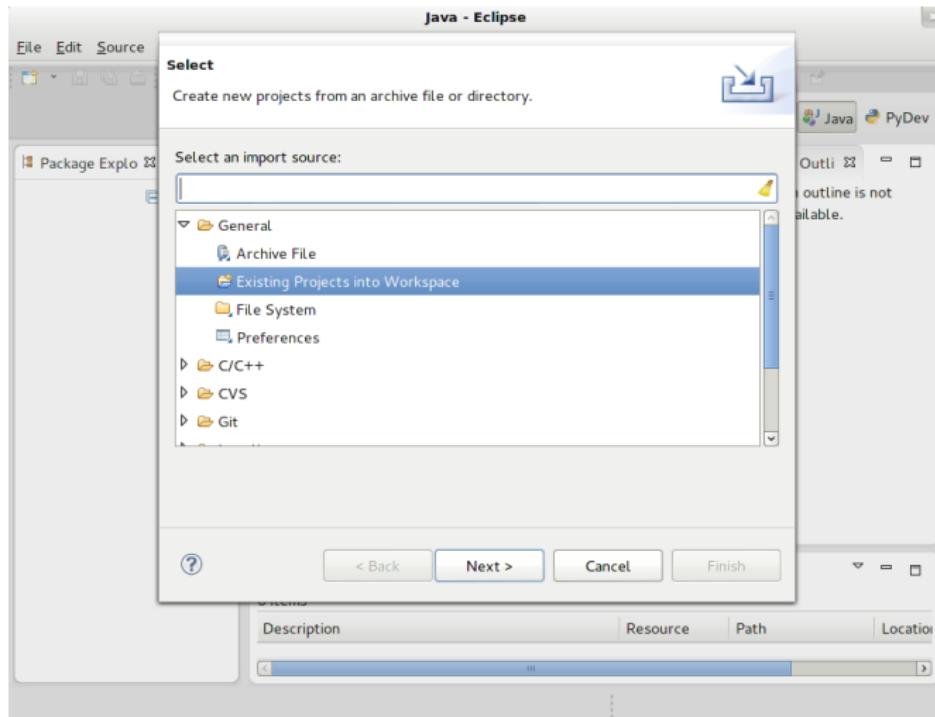
Starten Sie Eclipse und öffnen Sie einen *workspace*.
über *Terminal (ALT-F2)* – `/local/install/eclipse/eclipse`



Vorbereitungen: Aufgaben

Importieren Sie das zip Archiv

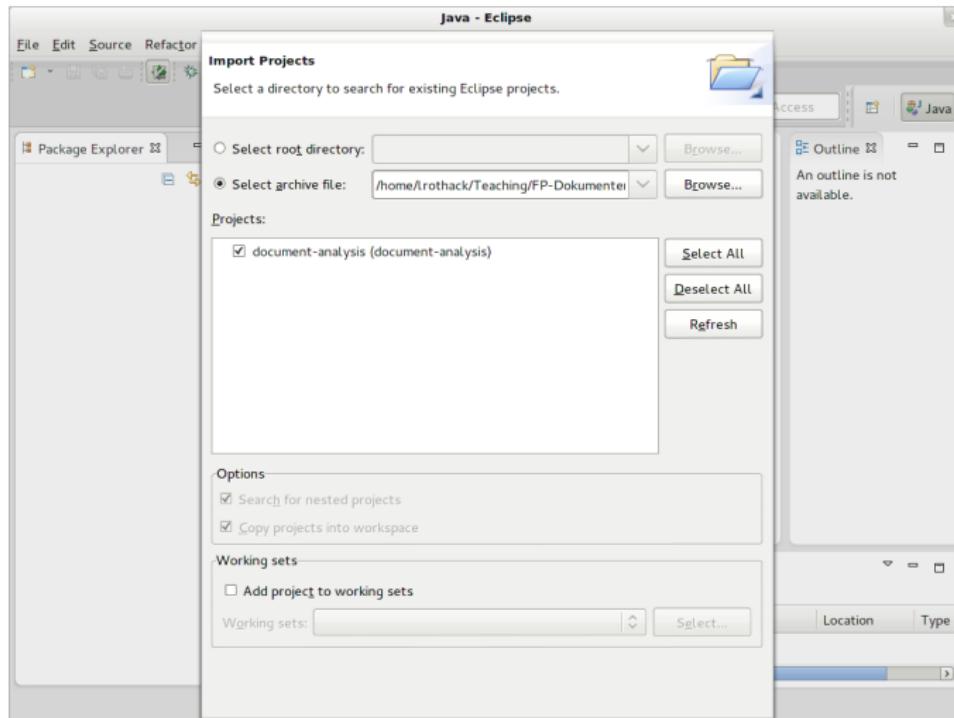
Import – General / Existing Project into workspace



Vorbereitungen: Aufgaben

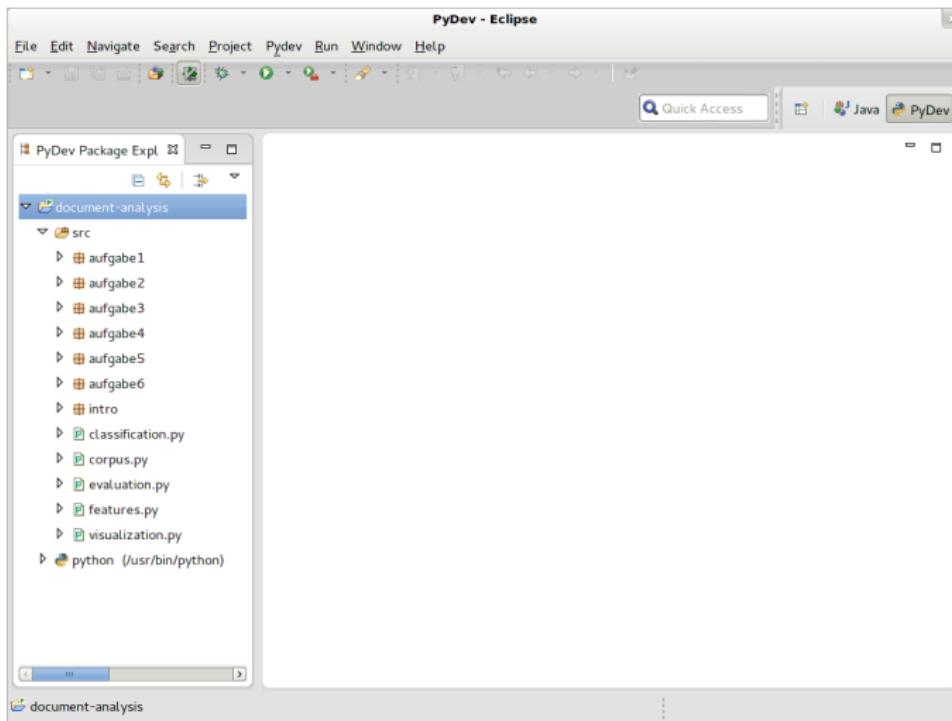
Wählen Sie das Archiv *aufgaben.zip*.

Select archive file – Browse



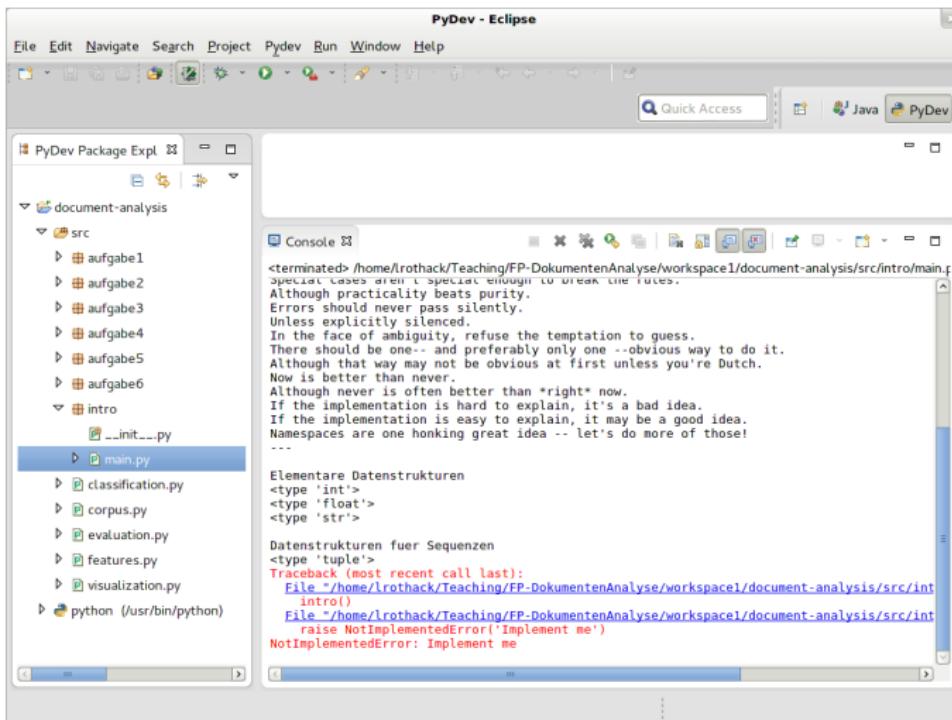
Vorbereitungen: Aufgaben

Konfigurieren Sie die Python Umgebung mit *Quick Auto Config* und wechseln die *PyDev Perspective*.



Vorbereitungen: Aufgaben

Führen Sie das Python Modul *intro/main.py* aus.
Run As – Python Run



The screenshot shows the PyDev - Eclipse interface. On the left, the PyDev Package Explorator displays a project structure under 'document-analysis'. The 'src' folder contains several subfolders ('aufgabe1' through 'aufgabe6') and an 'intro' folder containing files like '__init__.py', 'main.py', 'classification.py', 'corpus.py', 'evaluation.py', 'features.py', and 'visualization.py'. A file 'python (/usr/bin/python)' is also listed. The 'main.py' file is currently selected. On the right, the 'Console' tab shows the output of running the script. The output includes a quote from PEP 8 about practicality beats purity, followed by code examples and a traceback:

```
<terminated> /home/lrothack/Teaching/FP-DokumentenAnalyse/workspace1/document-analysis/src/intro/main.py
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
...
Elementare Datenstrukturen
<type 'int'>
<type 'float'>
<type 'str'>
Datenstrukturen fuer Sequenzen
<type 'tuple'>
Traceback (most recent call last):
  File "/home/lrothack/Teaching/FP-DokumentenAnalyse/workspace1/document-analysis/src/intro/intro.py", line 1, in <module>
    raise NotImplemented("Implement me")
NotImplementedError: Implement me
```

Vorbereitungen: Aufgaben

Führen Sie Unit Tests aus (*test* Paket), um Ihre Implementierung zu überprüfen. *Run As – Python unit-test*

The screenshot shows the Eclipse PyDev interface with the following details:

- Project Explorer (left):** Shows a Python project named "document-analysis" with several source files under "src" and a "test" directory containing "intro_test.py".
- Code Editor (center):** Displays the content of "intro.py". It includes a docstring and a note about reading comments carefully.
- PyUnit View (bottom center):** Shows the test results:
 - Runs: 11 / 11
 - 0 errors
 - 0 failures
 - Finished in: 0.01 secs.
- Captured Output (bottom right):** Shows the captured output of the tests, including NumPy array access results.

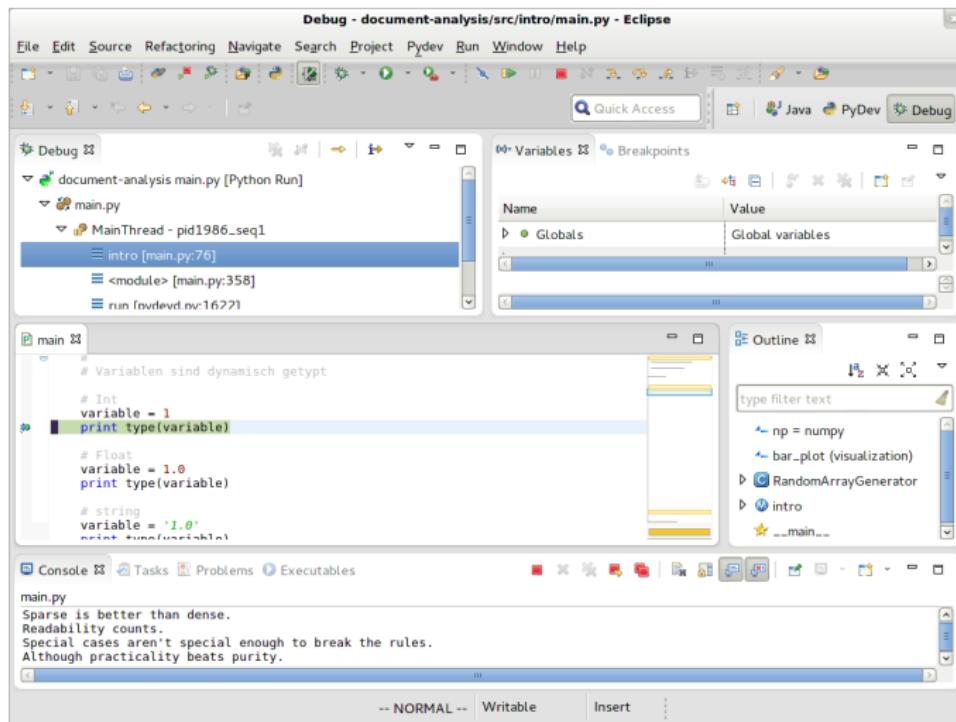
```
# Willkommen beim Fachprojekt Dokumentenanalyse!
#
# In diesem "intro" Modul sollen Sie sich mit den grundlegenden
# Eigenschaften und Funktionen von Python, NumPy/SciPy und matplotlib
# vertraut machen. Die Aufgaben sind dazu gedacht Ihnen den Einstieg
# zu erleichtern. Weiterführende Informationen sind auf der Webseite
# des Fachprojekts verlinkt.

# WICHTIG: Lesen Sie die Kommentare sorgfältig! Oft finden sich Hinweise
# zu Python / NumPy / SciPy Funktionen, die Ihnen die Arbeit erheblich

[NumPyIntro::array_access]
(4, 7)
[[ 199.   11.  192.  179.   93.    9.  170.]
 [ 192.  165.   41.  113.   24.   73.  131.]
 [ 94.  154.   53.  133.  136.  198.  150.]
 [ 4.  177.   25.   85.   68.   72.  105.]]
```

Vorbereitungen: Aufgaben

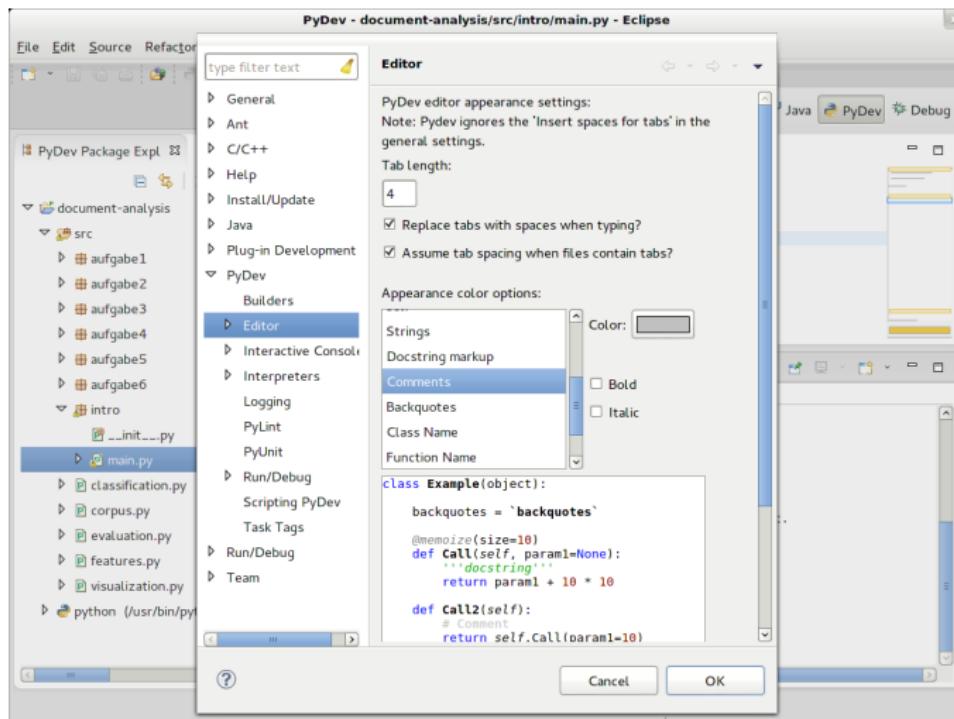
Verwenden Sie den Debugger!



Vorbereitungen: Aufgaben

Verbessern Sie die Lesbarkeit der Kommentare!

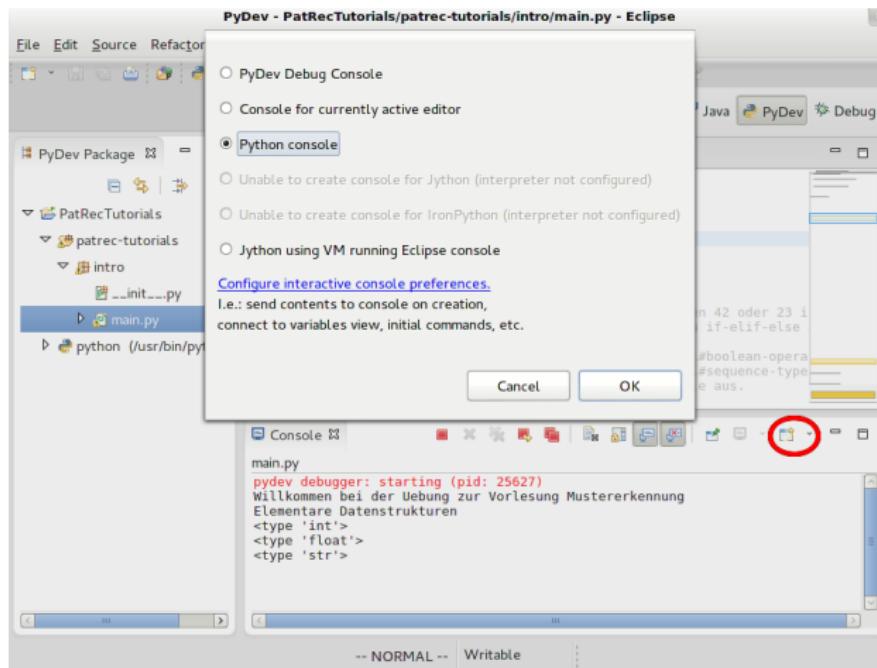
Preferences – PyDev / Editor



Vorbereitungen: Aufgaben

Verwenden Sie die interaktive PyDev Konsole! Eine zusätzliche Proxy Konfiguration in Eclipse ist dazu erforderlich:

Preferences – General – Network Connections – Manual(!)



Vorbereitungen: Aufgaben

Alternativ verwenden Sie die interaktive IPython Konsole!

```
haneke:/home/lrothack
File Edit View Search Terminal Help
lrothack@haneke:~$ ipython
Python 2.7.9 (default, Mar 1 2015, 12:57:24)
Type "copyright", "credits" or "license" for more information.

IPython 2.3.0 -- An enhanced Interactive Python.
?           -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help        -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: import numpy as np

In [2]: A = np.arange(9).reshape(3,3)

In [3]: A
Out[3]:
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])

In [4]:
```

Aufgabe 1: Brown Corpus

Datensatz zum Experimentieren mit semantischer Analyse maschinenlesbarer Texte. Besteht aus:

- ▶ 500 Dokumenten
- ▶ in 15 Kategorien
- ▶ Zu jedem Dokument ist die Kategorie bekannt.

Aufgabe 1: Eigenschaften des Brown Corpus

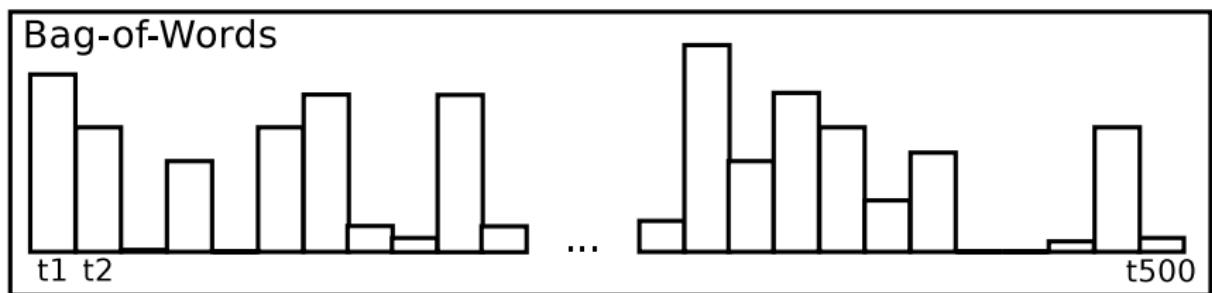
Untersuchen Sie statistische Eigenschaften des Brown Corpus:

- ▶ Anzahl Kategorien, Dokumente, Wörter.
- ▶ Anzahl Dokumente / Wörter per Kategorie
- ▶ Häufigste Wörter im Corpus / per Kategorie.
- ▶ Stopwords: Nicht semantisch relevant (the, a, on, ...)
- ▶ Stemming: Entfernen grammatischer Varianten
walking, walked,..⇒ walk

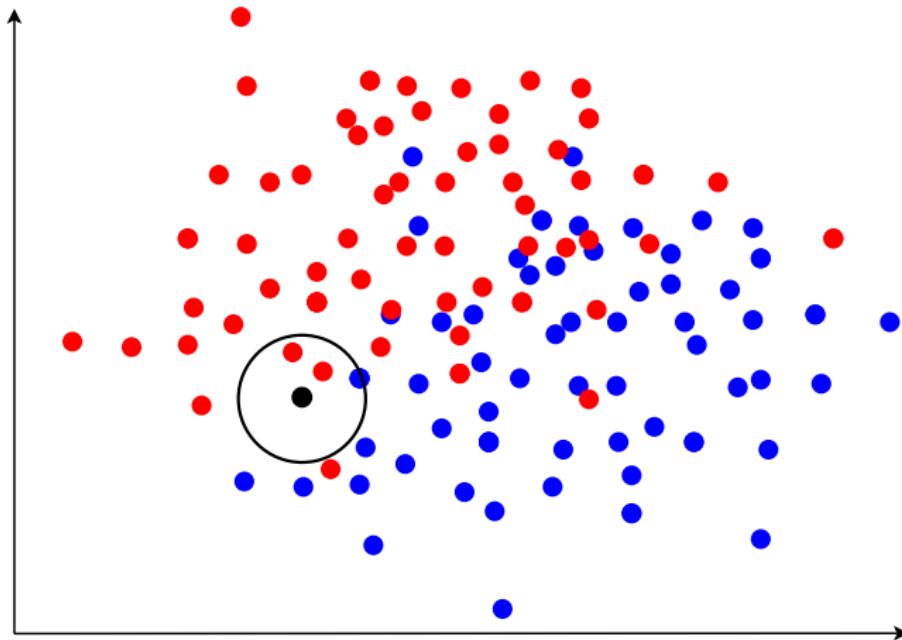
Algorithmen basieren auf der Analyse (Frequenzen) typischer Wörter.

Aufgabe 2: Bag-of-Words

1. Vorverarbeitung: Stopwords entfernen und Stemming
2. Vokabular typischer Wörter erstellen (*terms*)
⇒ N häufigste Wörter im Corpus
3. Jedes Dokument wird durch Histogramm über *terms* repräsentiert
⇒ *term vector*



Aufgabe 2: K-Nächste Nachbarn Klassifikator



- ▶ K-Nächste Nachbarn bestimmen
- ▶ Mehrheitsentscheid (*tie breaking*)

Aufgabe 2: Bewertungsmaße

Für zwei Punkte \mathbf{a}, \mathbf{b} im \mathbb{R}^n

- ▶ Euklidischer Abstand

$$d_{\text{eucl}}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{\nu=1}^n (a_\nu - b_\nu)^2}$$

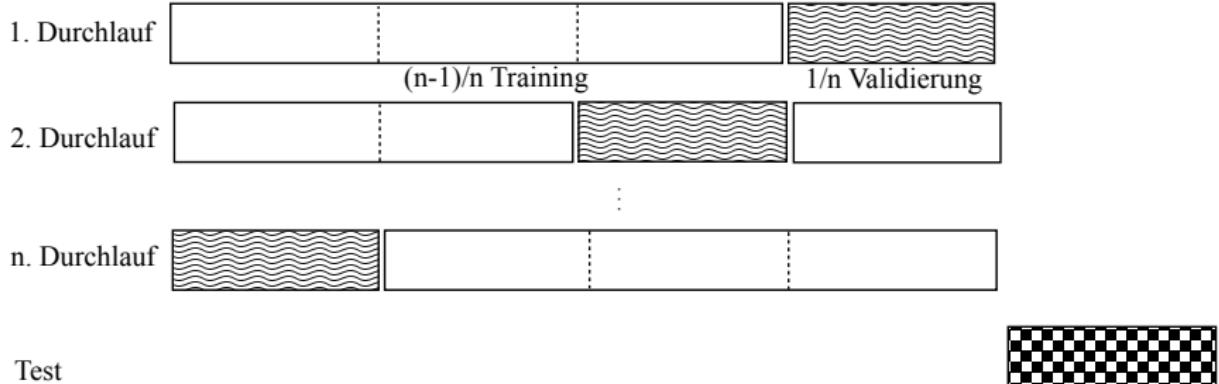
- ▶ Cityblock Abstand

$$d_{\text{cityb}}(\mathbf{a}, \mathbf{b}) = \sum_{\nu=1}^n \|a_\nu - b_\nu\|_1$$

- ▶ Cosinus Abstand

$$d_{\text{cos}}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$$

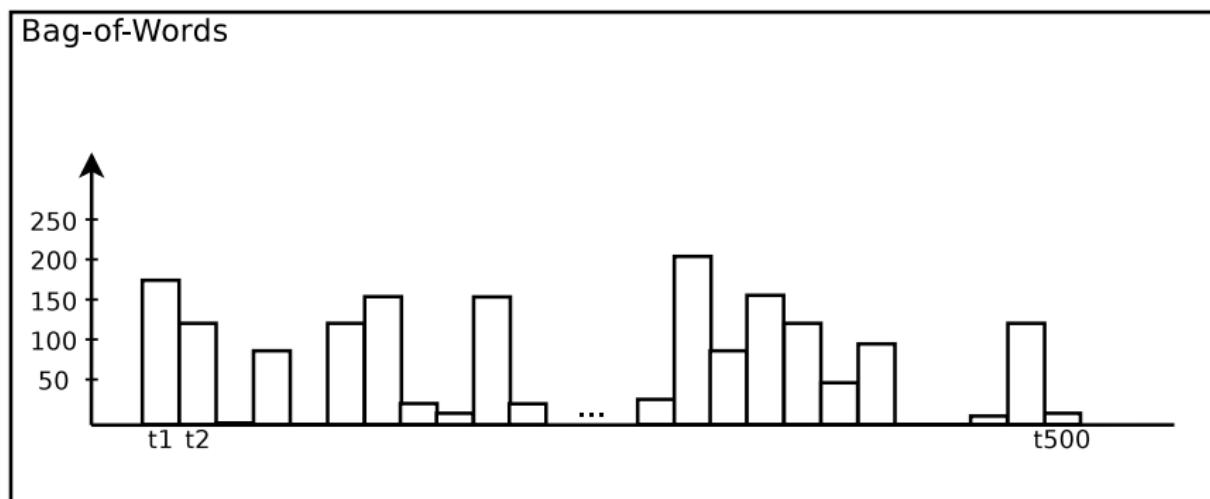
Aufgabe 3: Kreuzvalidierung



- ▶ Fehlerraten über die Validierung werden gemittelt
- ▶ Wir betrachten hier keinen separaten Test Datensatz
 - ⇒ Fehlerraten aus der Validierung lassen sich in der Regel nicht auf komplett unbekannte Daten verallgemeinern!

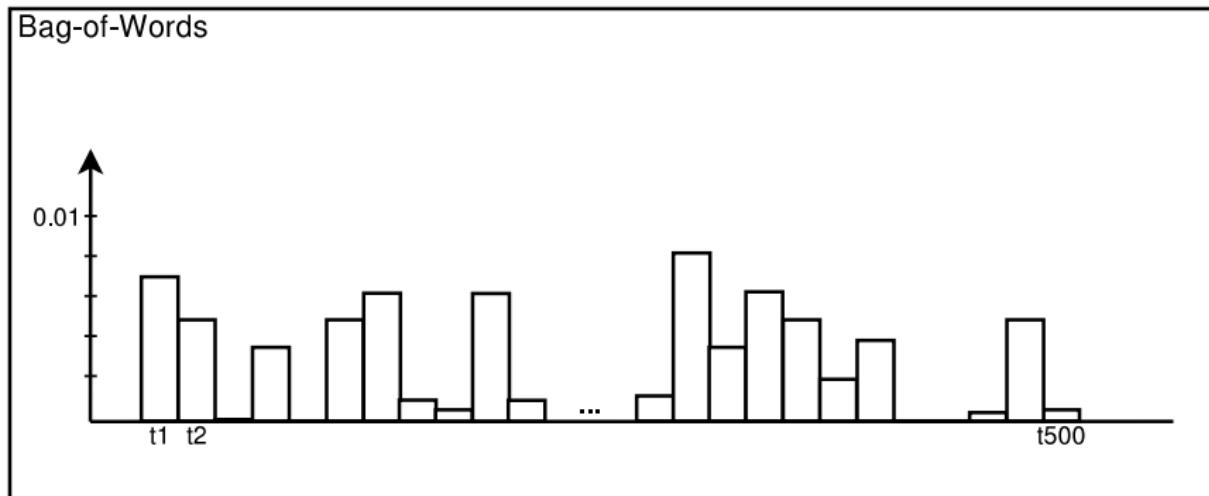
Aufgabe 3: Term Weighting

Absolute Frequenzen: Zählen der *terms* im Dokument (nach Vorverarbeitung).



Aufgabe 3: Term Weighting

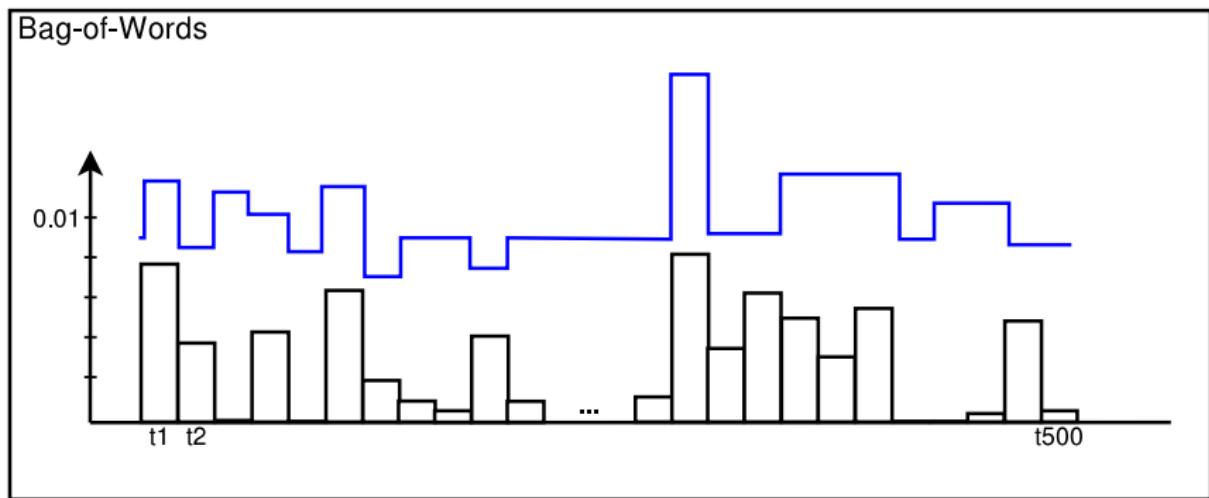
Relative Frequenzen: Normalisieren des Histogramms auf Summe Eins.



Unabhängigkeit von der Länge (Anzahl Wörter) eines Dokuments.

Aufgabe 3: Term Weighting

Term Frequency - Inverse Document Frequency: Zusätzliches Gewichten der *terms*.



Gewichtung nach Diskriminativität: *terms*, die in vielen Dokumenten vorkommen haben keine große semantische Bedeutung (s. Stopwords).

Aufgabe 4: Topic Repraesentationen

“Genetics”	“Evolution”	“Disease”	“Computers”
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers

Datengetriebene Analyse von Wortzusammenhängen

- ▶ Beispieldatensatz (keine Ground Truth nötig)
- ▶ Modellierung in einem mathematischen Unterraum

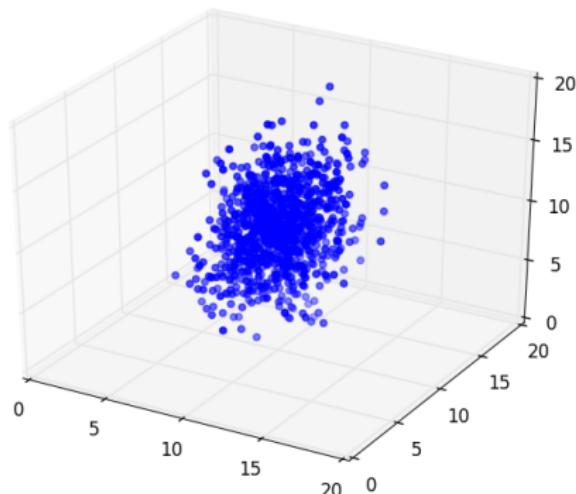
Aufgabe 4: Beispiel Hauptkomponentenanalyse

Statistisches Schätzen des Unterraums aus Beispieldaten

- ▶ Aktuelles Koordinatensystem

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- ▶ Neues Koordinatensystem
wird **datenabhängig**
⇒ Unterraum



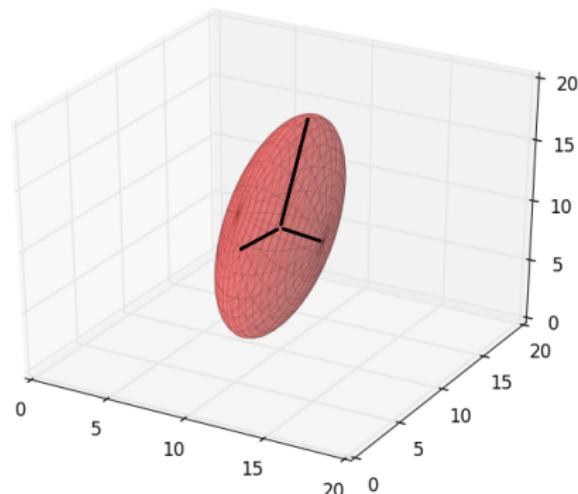
Aufgabe 4: Beispiel Hauptkomponentenanalyse

Statistisches Schätzen des Unterraums aus Beispieldaten

- ▶ Aktuelles Koordinatensystem

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- ▶ Neues Koordinatensystem
wird **datenabhängig**
⇒ Unterraum



Modellierung durch Kovarianzmatrix der Daten (hier im \mathbb{R}^3)

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix}$$

$$cov(x, y) = E\{(x - E\{x\})(y - E\{y\})\}$$

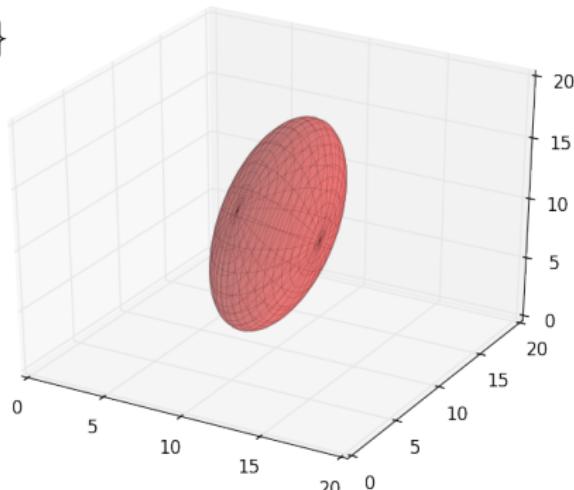
$$cov(x, y) = cov(y, x)$$

$$cov(x, x) = var(x)$$

Schätzen der Kovarianzmatrix:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_i, \quad \mathbf{s}_i \in \mathbb{R}^3$$

$$\hat{C} = \frac{1}{N} \sum_{i=1}^N (\mathbf{s}_i - \hat{\mu})(\mathbf{s}_i - \hat{\mu})^T$$



Bestimmen des neuen Koordinatensystems.

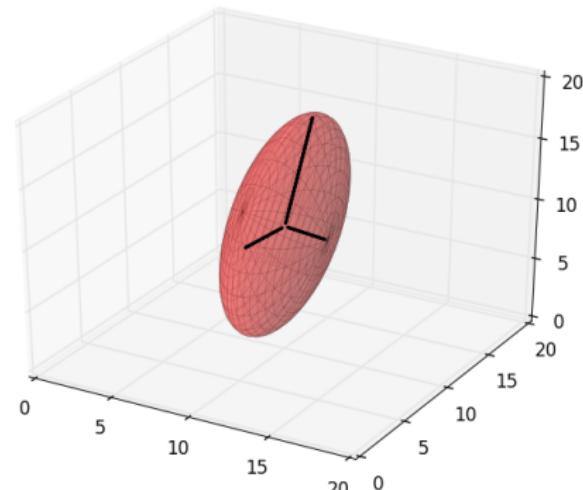
- ▶ Ursprung: Mittelwert der Beispieldaten
- ▶ Achsen: Eigenwert/-vektor Analyse der Kovarianzmatrix.

$$\hat{C}\nu_j = \lambda_j \nu_j \quad \text{mit} \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$$

$$\text{und} \quad \nu_i^T \nu_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- ▶ Anzahl von Eigenvektoren mit von Null verschiedenen Eigenwerten: Rang von \hat{C} .
- ▶ Eigenwerte $\hat{\equiv}$ Varianzen
- ▶ **Eigenvektoren bilden neues Koordinatensystem.**
- ▶ Transformation von Vektor s :

$$s' = [\nu_1, \nu_2, \dots, \nu_m]^T (s - \hat{\mu})$$

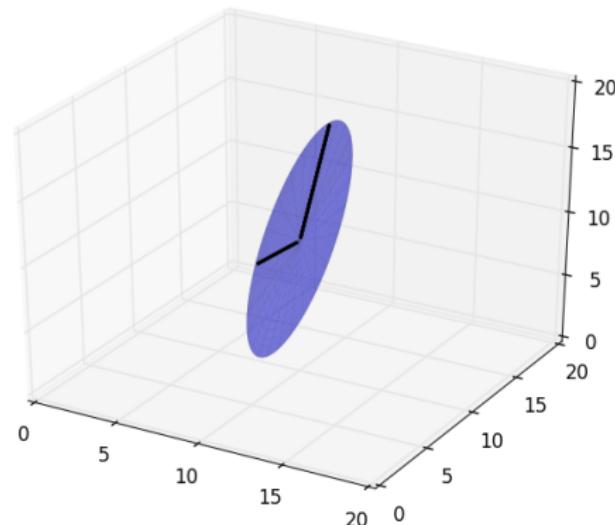


Dimensionsreduktion

Reduktion auf die Eigenvektoren zu den k höchsten Eigenwerten.

- ▶ Maximieren der Varianz
- ▶ Kompakte Repräsentation ohne großen Informationsverlust.
⇒ **relevante** Information?
- ▶ Anzahl der Dimensionen
 $k < m$ wird häufig empirisch bestimmt.
- ▶ Transformation von Vektor s :

$$s'_k = [\nu_1, \nu_2, \dots, \nu_k]^T (s - \hat{\mu})$$



Anwendung auf Bag-of-Words: Topic Modell

Durch Singulärwertzerlegung der Term-Dokument Matrix:

$$\begin{pmatrix} \left| \begin{array}{c} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_d \end{array} \right| \\ X \end{pmatrix} = \begin{pmatrix} \left| \begin{array}{c} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \end{array} \right| \\ T \end{pmatrix} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_m \end{pmatrix} \begin{pmatrix} \left| \begin{array}{c} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_d \end{array} \right| \\ D^T \end{pmatrix}$$

$$\text{mit } \mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad \text{und} \quad \mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

- ▶ Vektoren \mathbf{f}_i enthalten Wortfrequenzen pro Dokument.
- ▶ Singulärvektoren \mathbf{u}_j definieren Unterraum (vgl. Eigenvektoren).
- ▶ Singulärwerte s_j skalieren Axen \mathbf{u}_j (vgl. Eigenwerte)
- ▶ \mathbf{v}_i sind Koeffizienten der Vektoren \mathbf{f}_i im Unterraum (Topic Raum).

Zusammenhang mit Haupkomponentenanalyse

Singulärwertzerlegung der Term-Dokument Matrix entspricht Eigenvektor / -wert Analyse der Matrix:

$$\sum_{i=1}^N \mathbf{f}_i \mathbf{f}_i^T$$

Vergleiche mit Kovarianzmatrix:

$$\frac{1}{N} \sum_{i=1}^N (\mathbf{s}_i - \hat{\mu})(\mathbf{s}_i - \hat{\mu})^T \quad \text{mit} \quad \hat{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_i$$

Aufgabe 5: Naive Bayes Klassifikation

Term	Genetics	Evolution	Disease	Computers
one	0.05%	0.001%	0.03%	0.01%
would	0.07%	0.03%	0.006%	0.04%
:	:	:	:	:
information	0.04%	0.02%	0.09%	0.5%
dna	0.7%	0.3%	0.1%	0.0008%
:	:	:	:	:
rocket	0.0001%	0.000006%	0.00004%	0.05%

Probabilistische Modellierung

- ▶ Zuordnung zu Kategorien nach maximaler Wahrscheinlichkeit
- ▶ Statistisches Schätzen der Term - Kategorie Zugehörigkeit

Naive Bayes Klassifikator

Satz von Bayes für bedingte Wahrscheinlichkeiten:

$$P(K_i|\mathbf{f}) = \frac{P(K_i)P(\mathbf{f}|K_i)}{P(\mathbf{f})}$$

- ▶ $P(K_i)$ Klassen a-priori Wahrscheinlichkeit
- ▶ $P(\mathbf{f}|K_i)$ Klassenbedingte Wahrscheinlichkeit
- ▶ $P(\mathbf{f})$ Wahrscheinlichkeit für Dokument $\mathbf{f} = (f_1, \dots, f_{|\mathcal{V}|})^T$
- ▶ \mathbf{f} entspricht Bag-of-Words, $|\mathcal{V}|$ der Vokabulargröße

Klassifikation nach maximaler a-posteriori Wahrscheinlichkeit

$$k = \operatorname{argmax}_i P(K_i)P(\mathbf{f}|K_i)$$

Naive Bayes Klassifikator

Modellannahmen

- ▶ $P(K_i)$: Empirische Verteilung
- ▶ $P(\mathbf{f}|K_i)$: Multinomialverteilung

Klassifikation nach maximaler a-posteriori Wahrscheinlichkeit

$$P(K_i)P(\mathbf{f}|K_i) = P(K_i) \prod_{t=1}^{|\mathcal{V}|} P(w_t|K_i)^{f_t}$$

mit $P(w_t|K_i)$: W'keit für Term w_t (aus Vokabular) ggb. Klasse K_i

- ▶ Komponenten $\mathbf{f} = (f_1, \dots, f_{|\mathcal{V}|})^T$ unabhängig
- ▶ Berechnung in der *log*-Domäne

Schätzen von $P(K_i)$ und $P(w_t|K_i)$ aus repräsentativer Stichprobe

Naive Bayes Klassifikator

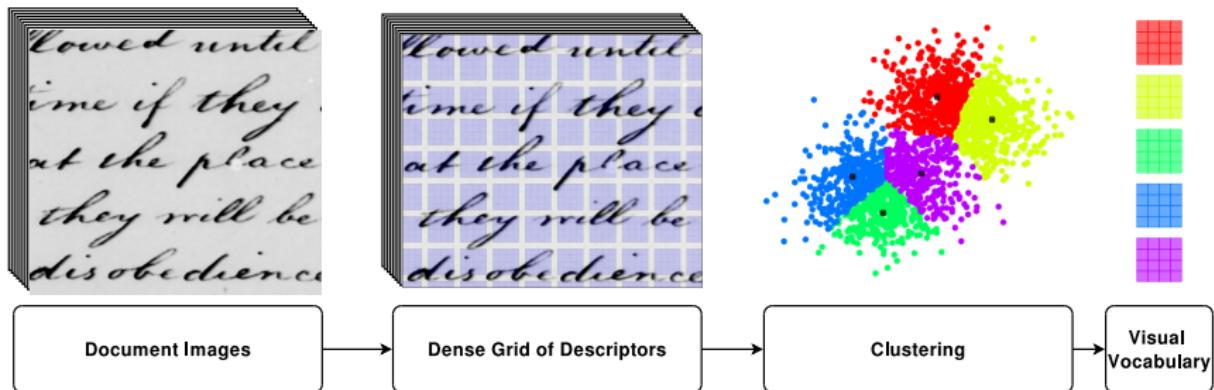
Schätzen des Modells auf Grundlage der Trainingsstichprobe

- ▶ $P(K_i)$: Relativer Anteil von Klasse K_i
- ▶ $P(w_t|K_i)$: Relativer Anteil des Vorkommens von Term w_t in Klasse K_i zu allen Vorkommen von Termen in Klasse K_i

$$P(w_t|K_i) = \frac{1 + \sum_{\mathbf{f} \in K_i} f_t}{|\mathcal{V}| + \sum_{\mathbf{f} \in K_i} \sum_{s=1}^{|\mathcal{V}|} f_s}, |\mathcal{V}| : \text{Vokabulargröße}$$

Laplace'sches Glätten zur Vermeidung von Null Wahrscheinlichkeiten
(für absolute Frequenzen)

Aufgabe 6: Bag-of-Features



Aufgabe 6: Bag-of-Features

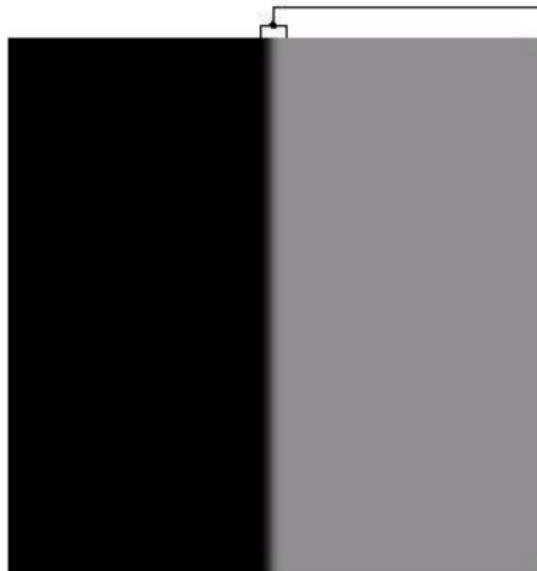
lowered until
time if they
at the place
they will be
disobedience

lowered until
time if they
at the place
they will be
disobedience

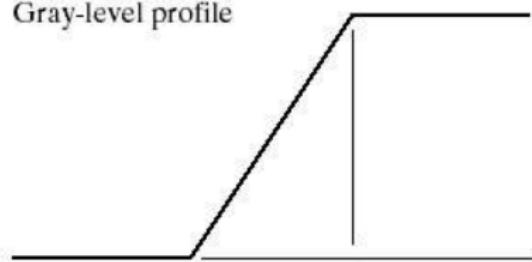
lowered until
time if they
at the place
they will be
disobedience



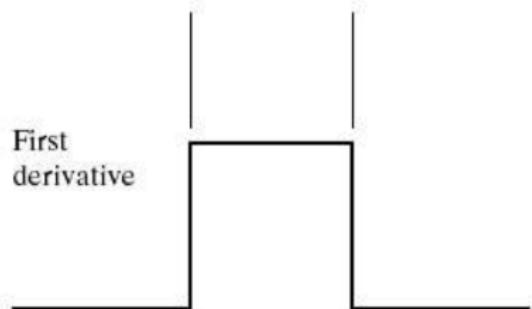
Bild Gradienten Einführung



Gray-level profile



First derivative

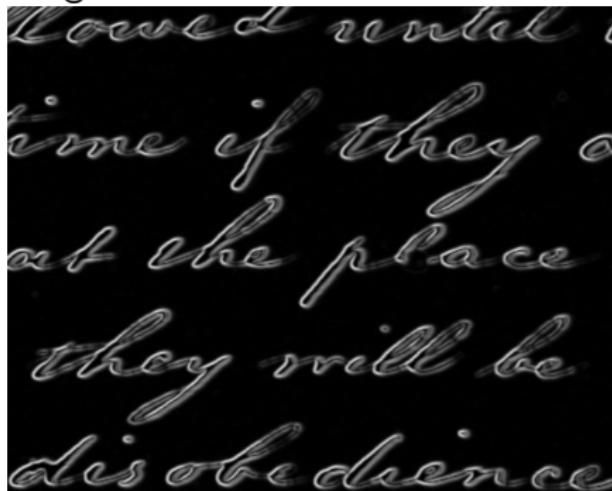


Beispiel Dokumentenabbild

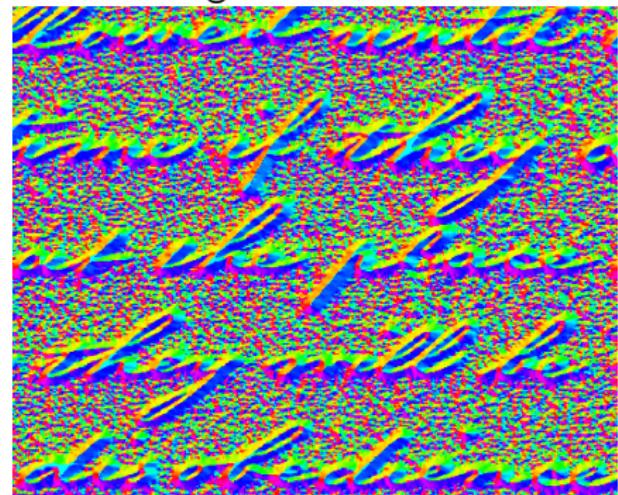
llowed until
ime if they
at the place
they will be
disobedience

Bild Gradienten

Magnituden



Orientierungen



Sobel Operator

Kreuzkorrelation

$$I(x, y) * K = \sum_{s=-a}^a \sum_{t=-b}^b K(s, t) I(x + s, y + t),$$

$K(0, 0)$ $\hat{=}$ Zentrum von K

Sobelmasken zur Detektion von horizontalen und vertikalen Kanten

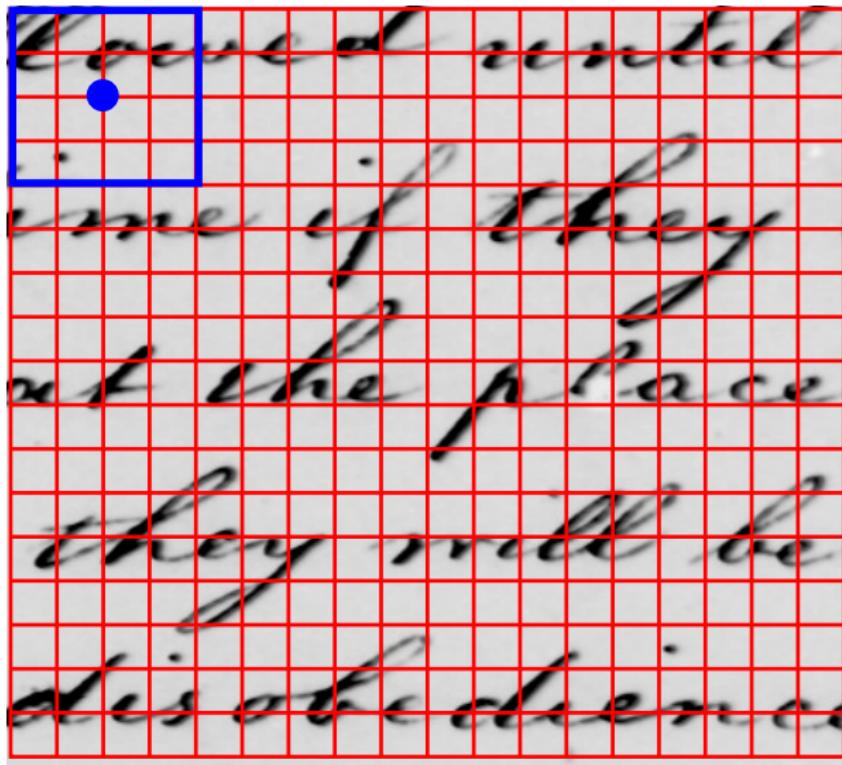
$$S_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad S_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Gradienten, Magnituden und Orientierungen

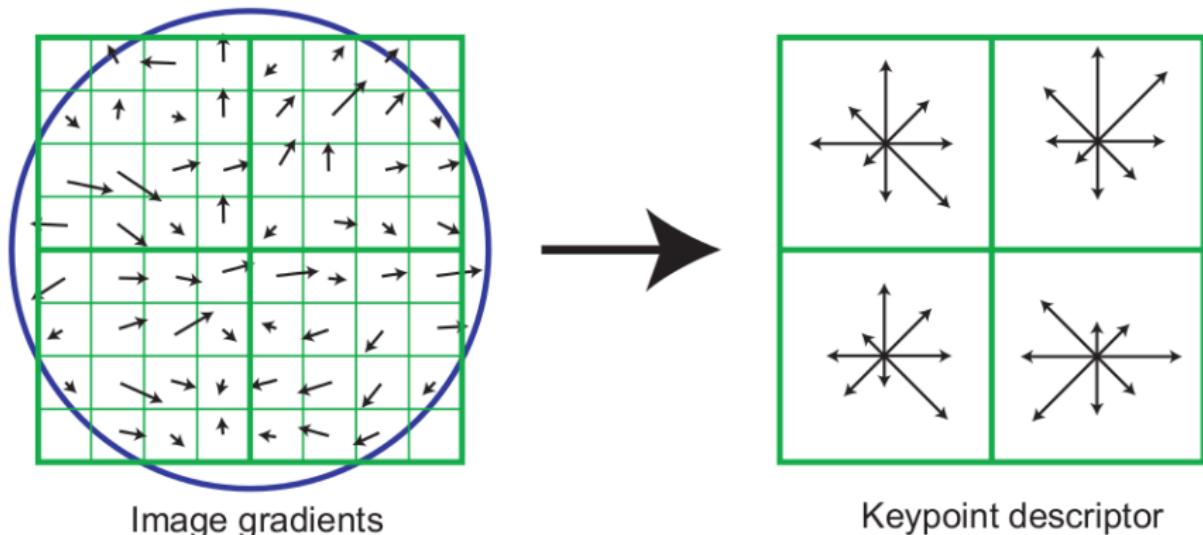
$$\mathbf{G} = \begin{pmatrix} G_x \\ G_y \end{pmatrix} \text{ mit } G_x = I * S_v, \quad G_y = I * S_h$$

$$G_{mag} = \sqrt{G_x^2 + G_y^2}, \quad G_{dir} = \arctan \frac{G_y}{G_x}$$

Regelmäßiges Grid von SIFT Deskriptoren

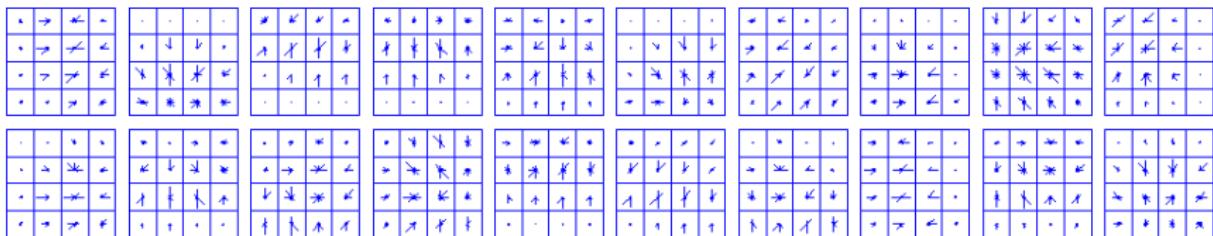


SIFT Deskriptoren



In diesem Beispiel werden 2×2 Zellen mit Subhistogrammen betrachtet.

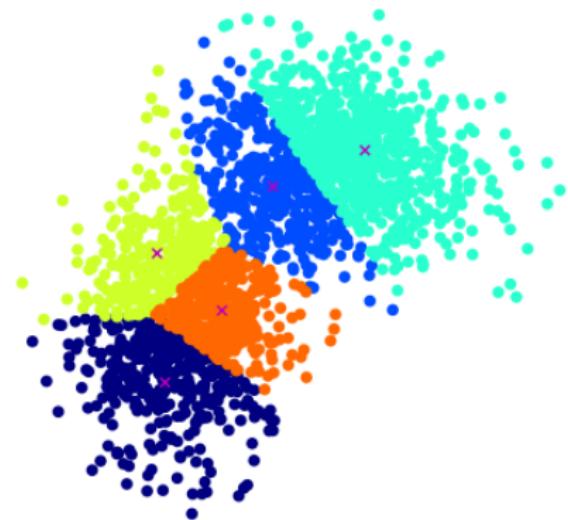
Visuelles Vokabular



- ▶ Typische SIFT Deskriptoren
- ▶ Clusteranalyse von Deskriptoren aus einem Beispieldatensatz

Lloyds Algorithmus

- ▶ Quantisierungsfehler minimieren
- ▶ Iteratives Verfahren
(lokales Optimum!)
- ▶ Nächster Nachbar Bedingung
- ▶ Centroid Bedingung



Lloyds Algorithmus

Geg.: Trainingsmenge $\omega = \{\underline{f}^{\varrho} | \varrho = 1, \dots, N\}$ von Vektoren;

Initiales Kodebuch $B^{(0)} = \{\underline{b}_{\nu} | \nu = 1, \dots, L\}$; (z.B. zufällig gewählt)

Schwellwert Θ für relative Abnahme des Quantisierungsfehlers

Setze Iterationsschritt $M = 0$;

Berechne Quantisierungsfehler $\epsilon^{(0)}$ mit $B^{(0)}$ und ω

REPEAT

Setze Iterationsschritt $m = m + 1$

Berechne Zuordnung der Werte $\underline{f}^{\varrho} \in \omega$ mit aktuellem Kodebuch $B^{(m-1)}$

Berechne neue Kodewörter $\underline{b}_{\nu}, \nu = 1, \dots, L$

Berechne neuen Fehler $\epsilon^{(m)}$

UNTIL relativer Fehler $\frac{\epsilon^{(m-1)} - \epsilon^{(m)}}{\epsilon^{(m)}} \leq \Theta$

Deskriptor Quantisierung und Bag-of-Features

