technische universität
dortmund

Bachelorarbeit

Visualisierung von Embeddings zur Analyse großer Dokumenten-Kollektionen

Phillip Kilian
Juni 2017

Gutachter:
Prof. Dr. Katharina Morik
M.Sc. Lukas Pfahler

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl für Künstliche Intelligenz (LS-8)
http://www-ai.cs.tu-dortmund.de

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Analyse großer Dokumenten-Kollektionen durch Visualisierung. Ziel ist es eine geeignete Verarbeitung hochdimensionaler und großer Datensätze aufzuzeigen und durch die praktische Anwendung an einer realen Problemstellung die Möglichkeiten und Grenzen des Vorgehens festzustellen. Es geht also darum, einen Ansatz für die Visualisierung hochdimensionaler großer Dokumenten-Kollektionen zu finden. Im Abschnitt 1.1 sind die Motivation, wieso ein solcher Ansatz überhaupt relevant ist und einige Hintergrundinformationen zu dem konkreten Projekt, welches von einer solchen Lösung profitieren könnte, beschrieben.

1.1 Motivation und Hintergrund

Im relNet Projekt möchten Religionswissenschaftler die Struktur des Forums von `www.jesus.de` untersuchen. Dabei soll festgestellt werden können, welche Diskussionen inhaltliche Gemeinsamkeiten aufweisen und worin sie sich unterscheiden. Besonderes Augenmerk liegt auf der Zusammensetzung einzelner Themen. So soll man untersuchen können, welche Themenbereiche sich zu einem größeren abstrahieren lassen. Dabei ist das Ziel die Struktur, also den Aufbau bzw. die Zusammensetzung des Forums zu entschlüsseln und grafisch analysierbar zu machen. Die grafische Darstellung ist deshalb wünschenswert, da sie dem Benutzer, in diesem Fall den Religionswissenschaftlern, eine interpretierbare Grafik liefert und Interaktionen ermöglicht. Diese Grafik soll ein zweidimensionales Streudiagramm sein. Damit soll die visuelle Analyse mögliche Zusammenhänge aufdecken, die das menschliche Gehirn durch die Fülle an Daten und der hohen Dimensionalität nicht erkennen würde. Bestimmte Interessensbereiche und Gruppen auszumachen wäre dabei optimal.

Da jede Diskussionsrunde ein Objekt darstellt, muss die Rückverfolgbarkeit vom grafischen Punkt in zwei Dimensionen zur jeweiligen Diskussion gegeben sein. Darüber hinaus sind Funktionen zum genaueren Untersuchen bestimmter Bereiche für die Interaktion zwischen Nutzer und Grafik sinnvoll, um z.B. in die Ansicht hereinzoomen zu können. Nutzer

sind vor allem an den Informationen, die zu den entsprechenden Punkten gehören interessiert, dazu kann es sinnvoll sein, auch die nächsten Nachbarn direkt abrufen zu können. Dadurch kann festgestellt werden, wieso sich Diskussionsrunden im Forum ähnlich sind.

Die in der Einleitung angesprochene Hochdimensionalität wird deutlich, wenn man die Repräsentation einer Diskussionsrunde erläutert. So wird jede Diskussion, die ein Dokument darstellt, als Vektor repräsentiert. Dadurch wird jede im Wortschatz der Diskussionen vorkommende Vokabel als eigenes binäres Merkmal betrachtet. Enthält eine Diskussionsrunde beispielsweise das Wort „Gott“, wird an der entsprechenden Stelle der Wert dieses Merkmals auf 1 gesetzt und bleibt andernfalls 0. Was das konkret für den Anwendungsfall im relNet Projekt bedeutet, wird in Abschnitt 4.1 genauer erläutert. Die Ähnlichkeit von Diskussionsrunden kann anhand unterschiedlicher Faktoren bewertet werden. In diesem Fall basiert sie auf dem verwendeten Vokabular. Diese Eigenschaft ist die, an welche man bei der Verarbeitung von Texten intuitiv denkt. So sind z.B. Plagiate für den Menschen ähnliche bzw. gleiche Dokumente, sobald der Wortlaut über mehrere Abschnitte deckungsgleich ist, also die gleichen Wörter vorkommen. Demnach ist ein ähnliches Vokabular Indiz für ein gleiches Thema, dieser Zusammenhang wird in Abschnitt 4.1 ebenfalls vertieft.

Dieses Projekt ist beispielhaft für die Problemstellung mit großen Datenmengen, sowie vielen Merkmalen umgehen zu können. Es stellt Herausforderungen an die Analysierbarkeit, da man riesige Dokumenten-Kollektionen verarbeiten muss, welche eine Vielzahl an Merkmalen besitzen. Diese Anforderungen finden sich in vielen vergleichbaren Projekten wieder, wodurch Lösungsansätze universell einsetzbar sind und nicht auf die zuvor beschriebene Anwendung beschränkt. Mit zunehmender Vernetzung in der Industrie, den Haushalten und praktisch allen Bereichen des Lebens, werden auch immer mehr Daten gesammelt. Die dabei erhobenen Informationen umfassen zunehmend mehr Merkmale und sind dadurch schwieriger aufzufassen. Demnach besteht die Forderung nach einer Darstellung der gesammelten Informationen, die für den Menschen interpretierbar ist.

Letzendlich ist es die Aufgabe der Religionswissenschaftler, von den dargestellten Zusammenhängen und Informationen zu abstrahieren und das Wissen abzuleiten. Die Informatik muss dabei das methodische Vorgehen liefern, mit den Anforderungen umgehen zu können und eine interpretierbare Visualisierung der Daten bereitstellen. Die besonderen Anforderungen der interaktiven Kommunikation stellen einige Herausforderungen an die technische Umsetzung, welche im Kapitel 7 aufgegriffen werden.

1.2 Problemstellung

Das Problem auf informationstechnischer Ebene besteht darin, sowohl mit der Anzahl, als auch mit der Dimensionalität der Datensätze umgehen zu können. Das Hauptproblem im Bezug auf die zweidimensionale Visualisierung liegt in der Dimensionalität der Daten. So ist es nicht trivial ein Objekt mit einer hohen Anzahl an Merkmalen in zwei Dimensionen

angemessen zu repräsentieren. Angemessen bedeutet dabei so viel Information im Zusammenhang des Datensatzes zu erhalten. Möchte man beispielsweise eine Menge von Vektoren mit sechs Koordinaten bzw. Merkmalen visuell darstellen, geschieht dies üblicherweise in zwei oder drei Dimensionen. Gibt es aber z.B. gerade sieben zueinander äquidistante Vektoren, so kann diese Eigenschaft in weniger als sechs Dimensionen nicht beibehalten werden. Aus diesem Grund sind Methoden zur Reduktion der Dimensionen notwendig, um für ein hochdimensionales Objekt, sprich die Dimension ist größer als drei $d > 3$, der Eingabemenge einen zweidimensionalen Repräsentanten zu bestimmen. Das Gebiet der *Dimensionsreduktion* beschäftigt sich mit der gerade angesprochenen angemessenen Repräsentation, sodass möglichst wenig Information verloren geht und der zweidimensionale Punkt im Bezug auf eine Qualitätsfunktion möglichst gut platziert ist. Aus der Beschreibung lässt sich schon ableiten, dass für unterschiedliche Einsatzgebiete und Anforderungen an die Reduktion unterschiedliche Ansätze und Verfahren sinnvoll sein können. Aus diesem Grund widmet sich Kapitel 2 mit der Differenzierung innerhalb des Gebietes und geht weiter auf die Ansätze ein.

Die zweite Herausforderung steckt in der Bewältigung großer Dokumenten-Kollektionen. Dabei stellen diese bestimmte Anforderungen an die effiziente Berechnung einer Visualisierung wegen der hohen Anzahl an Objekten. Darüber hinaus müssen Algorithmen auf diesen Mengen skalieren und können Probleme bei der Laufzeit im Bezug auf die Nutzerfreundlichkeit hervorrufen.

Im Bereich der künstlichen Intelligenz in der Informatik beschäftigt man sich unter anderem mit dem sogenannten *maschinellen Lernen*, dieser Aufgabenbereich wird im Verlauf der Arbeit eine wichtige Rolle spielen und wird daher im folgenden Abschnitt 1.2.1 näher erläutert. Es dient dabei als Einstiegspunkt für die formale und mathematische Herangehensweise und Umsetzung des vorgestellten Weges.

1.2.1 Maschinelles Lernen

Mit dem Ziel, Wissen aus einer Menge von Daten abzuleiten, versuchen maschinelle Lernverfahren Muster bzw. Gesetzmäßigkeiten in den Daten zu entdecken respektive zu lernen. Methodisch geht es darum, eine Funktion zu finden, die jedem Objekt einer Eingabemenge, $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ einen Ausgabewert $y \in Y$ durch $f : X \to Y$ zuweist. Dabei nennt man die Aufgabe eine solche Funktion zu finden *Lernaufgabe*. Je nach Lernaufgabe und gegebener Qualitäts- bzw. Kostenfunktion ist es das Ziel des Lernverfahrens die Qualitätsfunktion zu optimieren und dadurch eine optimale Funktion zur Abbildung in Y zu finden. Je nach Anwendungsgebiet kann die Ausgabe qualitativer oder auch quantitativer Natur sein. Mögliche Lernaufgaben sind zum Beispiel Clustering, Klassifikation und Regression. Beim Clustering kann z.B. eine Anzahl an Klassen Y vorgegeben werden, sodass das

Lernverfahren anhand der Informationen aus der Eingabemenge eine Zuordnung in diese Klassen lernt.

Um das Feld des maschinellen Lernens weiter zu kategorisieren, kann man es in *überwachtes* und *unüberwachtes Lernen* unterteilen. Das überwachte Lernen hat dabei neben der Eingabemenge eine sogenannte Trainingsmenge T, diese enthält eine Zuordnung einiger Objekte aus X zu einer Ausgabe aus Y. Mit Hilfe dieser bekannten Beispiele ermitteln die Algorithmen aus dem überwachten Lernen dann die Zuordnung bzw. Klassifikation neuer Objekte zu den bestehenden Ausgaben. Es wird also eine Vorhersage über die Ausgabe für Objekte der Eingabemenge bestimmt. Die Trainingsmenge enthält r Beispiele $T = (x_1, y_1), (x_2, y_2), \ldots, (x_r, y_p)$ die einer beliebigen Anzahl $r < n$ an Objekten eine Ausgabe $y \in \{y_1, y_2, \ldots, y_p\}$ zuweisen. Das Y ist dabei entweder durch die Beispiele der Trainingsmenge komplett gegeben oder wird im Laufe des Verfahrens durch weitere Klassen ergänzt. Im Vergleich dazu muss beim unüberwachten Lernen erst eine Einteilung Y bestimmt werden und erst dann können den Eingabeobjekten eine Ausgabe anhand der gelernten Abbildung zugewiesen werden.

Ein großer Unterschied zwischen den beiden Gebieten ist die Messung des Erfolgs einer Vorhersage. So kann beim überwachten Lernen genau bestimmt werden, ob die gelernte Abbildung gut ist, indem man die Ausgaben der Trainingsmenge mit der gelernten Funktion vorhersagt und anschließend die vorhergesagten Labels mit den wirklichen vergleicht. Beim unüberwachten Lernen ist diese nicht möglich, da man nicht weiß, ob die getroffene Zuteilung exakt ist, da man keinen Vergleichswert hat.

Nach dieser kurzen Einführung wird im folgenden Kapitel 2 das grundlegende Problem der *Dimensionsreduktion* mit den verschiedenen Ansatzmöglichkeiten erläutert.

Kapitel 2

Dimensionsreduktion

Die Dimensionsreduktion beschäftigt sich mit der Aufgabe einen gegebenen Datensatz in der Anzahl seiner Attribute zu verringern. Man geht davon aus, dass die Attribute der vorhandenen Beobachtungen bzw. des Datensatzes Funktionen weniger latenter Attribute sind. Latent bedeutet in diesem Zusammenhang versteckt oder verborgen und meint, dass die Attribute nicht auf den ersten Blick erkenntlich sind. Gerade bei vielen Beobachtungsmerkmalen sind daher Methoden erforderlich, um eine Beschreibung mit weniger Eigenschaften zu finden.

Dieser Zusammenhang lässt sich visuell mit Hilfe der *Mannigfaltigkeit* beschreiben. Mannigfaltigkeit bedeutet, dass die Daten auch in einem Raum mit niedriger Dimensionalität beschrieben werden können, der dem originalen Bild sehr ähnlich ist. Das bedeutet, dass ein Objekt in $\mathbb{R}^n$ lokal auch im euklidischen Raum $\mathbb{R}^m$ mit $m < n$ modelliert werden kann (vgl. Lee 2011 [9]). Ein anschauliches Beispiel ist die Erde z.B. in Form einer Landkarte in zwei Dimensionen anstatt in drei abzubilden. Dabei wird deutlich, dass man durch die Reduktion der Attribute in diesem Fall der dritten optischen Dimension, eine bessere Intuition im Bezug auf die Struktur z.B. für die Navigation erhält. Daraus ergibt sich das Ziel der Dimensionsreduktion, bei dem man versucht eine geeignete Funktion zu finden, die in den Raum mit niedriger Dimensionalität abbildet. Dabei gilt es zu beachten, möglichst viel der ursprünglichen Information zu erhalten, sprich nur minimal an Informationsgehalt einzubüßen.

Der folgende Abschnitt 2.1 erläutert die Einordnung der Dimensionsreduktiom im Kontext des maschinellen Lernens und gibt Aufschluss über die groben Unterteilungsmöglichkeiten in Hinsicht auf Einsatzgebiete und die Methodik der Disziplin. In 2.2 und 2.3 werden die beiden methodischen Unterschiede innerhalb der unüberwachten Dimensionsreduktion zur Eigenschaftsextraktion beschrieben. Abschnitt 2.4 fasst das Kapitel Dimensionsreduktion in Hinblick auf die Notwendigkeit von Verfahren zur Visualisierung zusammen und gibt einen Ausblick auf das in 3 vorgestellte Verfahren.

2.1 Einsatzgebiete und Schwierigkeiten

Die Einsatzgebiete der Dimensionsreduktion lassen sich ganz grob in zwei Bereiche unterteilen, wobei eine klare Einordnung der Verfahren zur Dimensionsreduktion in eines der beiden Gebiete nicht scharf definiert ist. Das erste Einsatzgebiet dient der Vorverarbeitung der Daten und zielt im Wesentlichen darauf ab, den Datensatz so zu vereinfachen, dass die Performanz weiterer Analyseverfahren gesteigert bzw. erst möglich wird (vgl. Abschnitt 2.1.1). Ein weiterer Zweck von Dimensionsreduktion ist speziell zur Visualisierung, hier ist eine Transformation in den zwei- oder dreidimensionalen Raum angedacht.

Dabei kann man innerhalb der Dimensionsreduktion in zwei Vorgehensweisen unterscheiden, wobei sich diese Arbeit auf die Zweite beschränkt. Geht man davon aus, dass einige Eigenschaften unrelevant oder redundant sind, so wird häufig eine Teilmenge der bestehenden Attribute ausgewählt und man spricht von Selektion, um die Anzahl der Dimensionen zu reduzieren. In diesem Fall wird die sogenannte *Eigenschaftsselektion* genutzt, um eine Beschreibung des vorhandenen Datensatzes mit weniger Attributen zu finden. Da es keine annotierten Trainingsbeispiele gibt, ist es damit dem unüberwachten Lernen zuzuordnen. Um die Relevanz und Redundanz von Attributen zu ermitteln, kann man die Eigenschaftsselektion in drei Ansätze teilen: Embedded-, Filter- und Wrapper-Ansatz. Diese Arbeit beschränkt sich, wie oben schon angedeutet, auf das Konstruieren neuer Attribute und geht daher nicht weiter auf die Ansätze der Selektion ein (vgl. Steinbach et al. [15][S. 50ff]).

Der zweite Ansatz zum Reduzieren der Dimensionen ist das Generieren neuer Attribute aus den bereits vorhanden. Die sogenannte *Extraktion* konstruiert dabei neue Attribute und wird anhand einiger Beispiele im weiteren Verlauf des Kapitels näher erläutert. Im Folgenden beschränkt sich diese Arbeit für das weitere Verständnis auf die unüberwachten Algorithmen zur Extraktion.

Ein häufiges Problem, das überhaupt erst zur Notwendigkeit der Dimensionsreduktion führt ist der nachfolgend beschriebene *Fluch der Dimensionalität* und rechtfertigt in erster Linie das Gebiet der Vorverarbeitung. Darüber hinaus ist es oft wünschenswert einen hochdimensionalen Datensatz graphisch zu visualisieren, um ein Gefühl für die Struktur zu bekommen. Da Menschen einen hochdimensionalen Raum nicht angemessen interpretieren können, bietet sich eine Visualisierung zur Analyse an. In diesem Fall gibt es eventuell andere Anforderungen für die Reduktion, welche besonders in der Zusammenfassung 2.4 nochmal aufgegriffen werden und in das nachfolgende Kapitel überleiten.

2.1.1 Fluch der Dimensionalität

Der Fluch der Dimensionalität, wie er schon von Bellman 1961 [2] erwähnt wurde, beschreibt ein Problem mit hoher Dimensionalität in Datensätzen. Im Bezug auf maschinelle Lernverfahren geht es darum, dass Analysemethoden wie zum Beispiel die Klassifikation

bei hochdimensionalen Datensätzen schlechte Ergebnisse liefern können. Mit zunehmender Dimensionalität steigt das Volumen des betrachteten Raumes und die vorhandenen Daten können immer spärlicher werden. Spärliche Datensätze sind für Algorithmen, die auf statistischer Signifikanz arbeiten schwierig zu nutzen. So verlieren beispielsweise die für die Klassifikation zentralen Begriffe wie Dichte und Abstandsmaß an Aussagekraft. Der Informationsgewinn aus dem Abstand zweier Punkte nimmt ab und erschwert so die Datenanalyse. Statistisch gesehen, benötigt man mehr Beispiele, um Lernaufgaben mit hochdimensionalen Daten zu bewältigen, als wenn diese mit weniger Attributen beschrieben wären.

Aus diesem Grund kann es sinnvoll sein, auf dem zu analysierenden Datensatz ein Verfahren zur Dimensionsreduktion anzuwenden. So können Attribute mit geringem Informationsgehalt, sprich niedriger Varianz, eliminiert und das gerade beschriebene Problem vermieden oder eingedämmt werden. Zusätzlich kann die Reduktion in vielen Fällen positive Auswirkungen auf die Laufzeit und den Speicheraufwand anschließender Algorithmen mit sich bringen. Um die gerade beschriebenen Probleme zu vermeiden werden häufig, aber nicht ausschließlich, sogenannte *lineare Verfahren* eingesetzt. Diese können z.B. angewandt werden, falls ein hochdimensionaler Datensatz zu Laufzeitproblemen (Algorithmus skaliert nicht) bei der weiteren Verarbeitung führt oder Clusteranalysen schlechte Ergebnisse liefern, da die Daten zu spärlich sind, um differenzierte Aussagen über Cluster treffen zu können.

2.2 Lineare Verfahren

Lineare Verfahren zur Reduktion der Dimensionen gehen davon aus, dass es einen linearen Zusammenhang zwischen den bestehenden und neu zu berechnenden Attributen gibt. Das bedeutet, dass die gesuchte Funktion eine lineare Abbildung der vorhandenen Attribute ist. Sie werden häufig im Zuge der Vorverarbeitung für andere Verfahren eingesetzt, womit sie die Unterscheidungskategorie Vorverarbeitung begründen. Zu den populärsten linearen Verfahren zählen unter anderem die Hauptkomponentenanalyse (Principal Component Analysis, kurz PCA), die Unabhängigkeitsanalyse (Independent Component Analysis, kurz ICA) und die Singulärwertzerlegung (Single Value Decomposition, kurz SVD).

Da die Hauptkomponentenanalyse im weiteren Verlauf genutzt wird, ist sie in 2.2.1 erläutert und soll als Beispiel für die oben genannten Verfahren dienen.

2.2.1 Principal Component Analysis

PCA, wie es nach Pearson 1901 [10] und Hotelling 1933 [5] beschrieben wurde, ist ein Verfahren aus der linearen Algebra für kontinuierliche Attribute. Es werden die n *Hauptkomponenten* eines Datensatzes durch lineare Kombination der originalen Attribute gebildet. Dabei sind die konstruierten Komponenten orthogonal zueinander und versuchen jeweils

die Varianz, sprich ihren Informationsgewinn, zu maximieren. Wie in der Einleitung schon angesprochen, ist das Verfahren dem unüberwachten Lernen zuzuordnen. Zu den vielfältigen Einsatzgebieten gehören unter anderem die Datenkompression, Bildanalyse, Regression und die Vorhersage von Zeitreihen (vgl. Tipping und Bishop 1999 [16]).

Nach Jolliffe 2002 [6] lässt sich die Hauptkomponentenanalyse wie folgt definieren: Ausgehend von einem Datensatz $\mathbf{X}$ mit d unterschiedlichen Merkmalen und n Beobachtungen, wird mathematisch gesehen eine Hauptachsentransformation durchgeführt. Das Ziel ist es, wenige abgeleitete Attribute p mit $p \ll d$ zu finden, welche die meiste Information in $\mathbf{X}$ erhalten. Jede dieser neuen Attribute p wird eine Hauptkomponente, im Englischen Principal Component, kurz PC, genannt.

In den folgenden Erläuterungen beschreibt x_i mit $i \in \mathbb{N} \wedge i \leq n$ die i-te Beobachtung und x^j mit $j \in \mathbb{N} \wedge j \leq d$ das j-te Merkmal des Datensatzes. Mit x_{ij} wird der j-te Merkmalswert des i-ten Eintrags des Datensatzes referenziert.

Die einzelnen Werte der Beobachtungen müssen für die Analyse zentriert werden, indem man den Durchschnitt jedes Merkmals von den jeweiligen Beobachtungswerten abzieht. Man fordert also, dass $\sum_{i=1}^{n} x_i = 0$ gelten muss. Mit dem entsprechenden Durchschnittswert $\bar{x}^j = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$, der für jedes der d Merkmale berechnet wird, ergibt sich der zentrierte Datensatz $\tilde{\mathbf{X}}$ durch die Werte in Formel (2.1).

$$\tilde{x}_{ij} = x_{ij} - \bar{x}^j \tag{2.1}$$

Im Anschluss stellt man für die Transformation die symmetrische $(d \times d)$ Kovarianzmatrix $\mathbf{C}$ auf, welche die paarweisen Kovarianzen aller Merkmale enthält und Aufschluss über die Korrelation und die Varianz der d unterschiedlichen Merkmale liefert.

$$c_{ij} = \begin{cases} Cov(x^i, x^j) & i \neq j \\ Var(x^i) & i = j \end{cases} \tag{2.2}$$

Demnach enthält die Matrix auf der Diagonalen, also für $i = j$ die Varianz der einzelnen Merkmale, die restlichen Einträge geben Einsicht über die Kovarianzen zwischen den jeweiligen Attributen. Die Kovarianz ist dabei das Zusammenhangsmaß für die Merkmale und in diesem Fall eine Schätzung einer linearen Korrelation. Sie gibt nur eine Richtung an, über die Stärke des Zusammenhangs gibt sie keinen Aufschluss, sprich sie sind nicht standardisiert miteinander vergleichbar. Ist die Kovarianz zweier Merkmale gleich Null, also $Cov(x^i, x^j) = 0$, so besteht kein monotoner Zusammenhang zwischen den beiden Dimensionen.

Im dritten Schritt bestimmt man nun die Eigenwerte und Eigenvektoren aus der quadratischen Kovarianzmatrix $\mathbf{C}$ mit $\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$. Die Eigenwerte $\lambda_1, \lambda_2, \ldots, \lambda_d$ werden in der Diagonalmatrix $\mathbf{\Lambda}$ der Größe nach absteigend sortiert, sodass $\forall n \in \mathbb{N}, n+1 \leq d$: $\lambda_n > \lambda_{n+1}$. Die entsprechenden normierten Eigenvektoren $\mathbf{v}_i$ bilden die orthogonale Eigen-

vektormatrix $\mathbf{V}$, sodass $\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ gilt. Fasst man die Spalten von $\mathbf{V}$ zu $\mathbf{\Gamma}$ zusammen, gilt $\mathbf{\Lambda} = \mathbf{\Gamma}^T \mathbf{C} \mathbf{\Gamma}$ mit $\mathbf{\Gamma}^T$ transponierter Vektor. Jede der d Eigenvektoren ist orthogonal zueinander, anhand der entsprechenden Eigenwerte λ_d, lässt sich feststellen, wie viel Varianz das jeweilige Attribut besitzt. Da im vorherigen Schritt alle Attribute anhand ihrer Eigenwerte absteigend sortiert wurden, besitzt das erste Attribut den höchsten Informationsgehalt. Das zweite Attribut die zweit höchste Varianz und so weiter. Jedes dieser Attribute ist eine Hauptkomponente, oft auch latente Variable genannt, wobei der erste Eigenvektor den größten Eigenwert besitzt. Diese Eigenschaft hilft im nächsten Schritt den originalen Datensatz so zu transformieren, dass die Anzahl der Dimensionen reduziert werden kann und man jeweils den größten Informationsgewinn als nächste Achse wählt. Sie beschreibt somit den signifikantesten Zusammenhang zwischen den originalen Merkmalen.

Sobald die Eigenvektoren aus der Kovarianzmatrix gefunden und anhand ihrer Eigenwerte absteigend sortiert wurden, kann man mit Formel (2.3) den transformierten Datensatz bestimmen. Dazu werden zunächst die Eigenvektoren und -werte für alle d Merkmale berechnet und danach nur die p neuen Attribute aus $\mathbf{V}$ genutzt, wodurch sich der neue Datensatz mit $(n \times p)$ Werten ergibt.

$$PCA(\mathbf{X}, p) = \tilde{\mathbf{X}} \mathbf{V}_p \tag{2.3}$$

Durch die Anwendung der Hauptkomponentenanalyse verlieren die Achsen oft ihre natürliche Interpretation, so kann man später nur selber vermuten, welche Eigenschaften wohl zusammengefasst wurden. Oft kann man aber bestimmte Muster erkennen, sodass zum Beispiel Merkmale wie Länge, Breite und Höhe zu einem neuen Merkmal Größe zusammengefasst wurden.

Angenommen der zu analysierende Datensatz ist normalverteilt, so wären die Daten nach der Hauptkomponentenanalyse sowohl unkorreliert als auch statistisch unabhängig. Damit bietet PCA eine optimale Dekomposition für normalverteile Datensätze.

Zur Ermittlung der Anzahl an Hauptkomponenten kann zum Beispiel ein Streudiagramm konstruiert werden, welcher die Anzahl der Komponenten gegen ihre kumulierte Varianz aufträgt. So kann man nach der Hauptkomponentenanalyse bestimmen, wie viele der berechneten Hauptkomponenten im weiteren Verlauf genutzt werden sollen. Alternativ kann man die einzelnen Eigenwerte der Hauptkomponenten aufsummieren, um so ein Maß für die kumulierte Varianz bis zur k-ten Hauptkomponente ermitteln. So lässt sich schnell erkennen, mit wie vielen Hauptkomponenten man welchen Teil der Gesamtvarianz abbilden kann. So liefert der Graph bzw. die Summe einen Anhaltspunkt für die Wahl der Anzahl an den zu nutzenden Hauptkomponenten. Nach Jollife 2002 [6][S. 112f 6.1.1] sollte die totale Variation, also die kumulierte Varianz des gewählten n, mindestens 80 Prozent ausmachen.

2.3 Nicht-Lineare Verfahren

Im vorherigen Abschnitt wird ein linearer Zusammenhang angenommen. Diese Annahme kann sich besonders im Bezug auf die Visualisierung als Problem darstellen und ist eine potentielle Einschränkung. Je nach vorliegendem Datensatz kann das gerade beschriebene Vorgehen also problematisch werden, wie das folgende Beispiel zeigen soll (vgl. Hastie et al. [17][S. 546]).

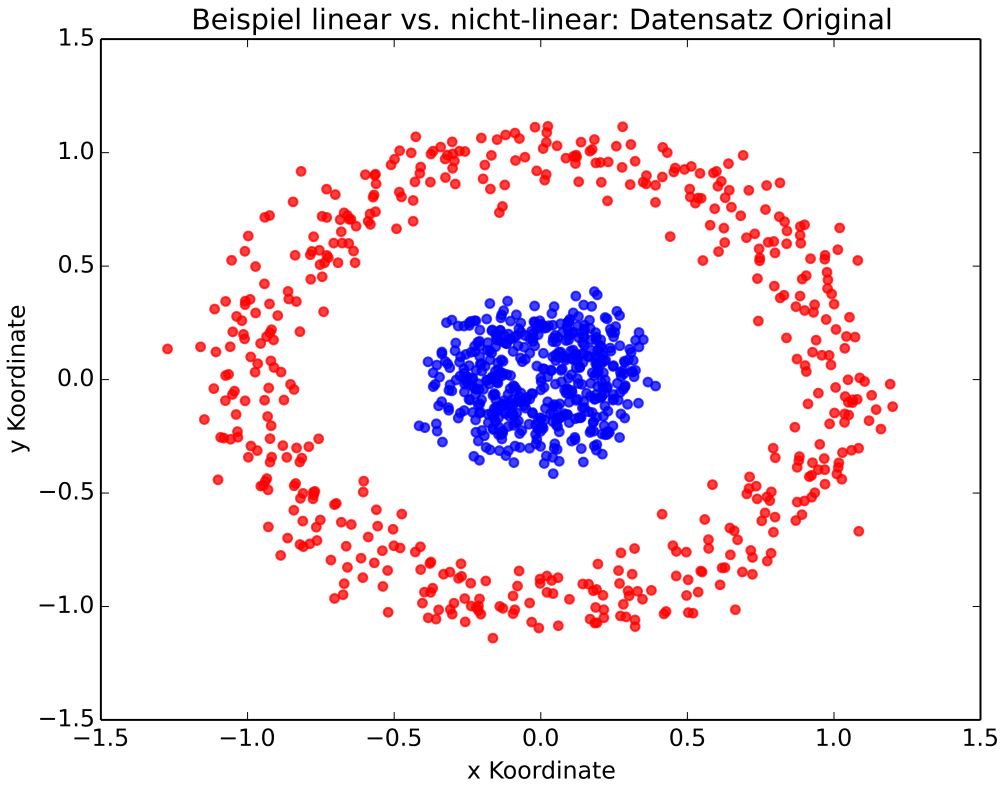
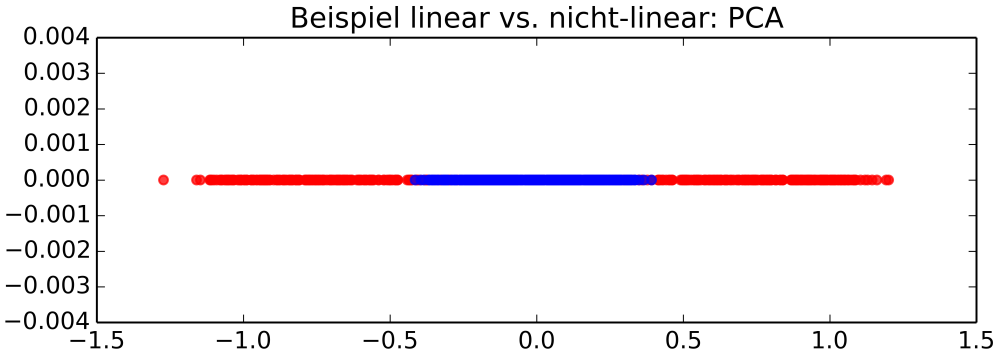


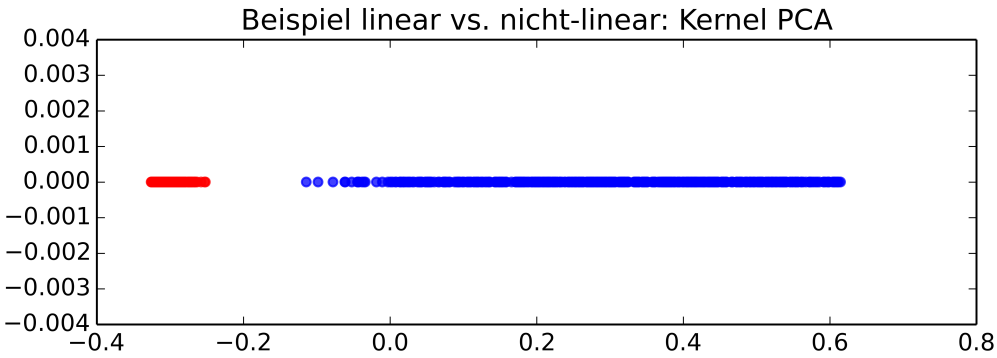
Abbildung 2.1: Beispiel linear vs. nicht-linear: Datensatz Original

Angenommen man möchte den in Abbildung 2.1 gezeigten Datensatz auf eine Dimension reduzieren, so zeigt Abbildung 2.2a das entstehende Abbild durch die Anwendung der Hauptkomponentenanalyse. Es wird schnell deutlich, dass das Verfahren die Punkte anhand eines linearen Zusammenhangs abbildet. In dem hier vorliegenden Fall von zwei separaten Kreisen führt dies zu einer suboptimalen Lösung. Suboptimal in dem Sinne, dass das Verfahren die beiden Kreise nicht separiert voneinander platziert. Die Ursache liegt darin, dass es von einem linearen Zusammenhang zwischen den Attributen ausgeht. Denkt man sich die Farbe in Abbildung 2.2a weg, so wird schnell deutlich, dass das Verfahren bei der Aufgabe die beiden Kreise voneinander zu trennen versagt.

Für das oben gezeigte Beispiel mit den zwei Kreisen, wäre eine klare Trennung in den blauen und roten Kreis in der Reduktion auf eine Dimension wünschenswert. Aus dem



(a) Principal Component Analysis



(b) Kernel Principal Component Analysis

Abbildung 2.2: Beispiel linear vs. nicht-linear: Vergleich

reduzierten Datensatz per PCA geht nicht hervor, dass der originale Datensatz aus zwei klar voneinander getrennten Kreisen besteht und hilft demnach nicht die Struktur zu verstehen. Aus diesem Grund ist der zweite methodische Ansatz innerhalb der Dimensionsreduktion das nicht-linearen Vorgehen, welches oft auch als Manifold Learning bezeichnet wird, da sein Einsatzgebiet oft das Lernen niedriger Mannigfaltigkeiten ist.

Abbildung 2.2b zeigt die Reduktion der beiden Kreise per *Kernel PCA*, der nicht-linearen Variante der Hauptkomponentenanalyse, welche im folgenden weiter erläutert wird (Die genutze Kernelfunktion ist die radiale Basisfunktion mit $\gamma = 15$, vgl. Abschnitt 2.3.1). Hier erkennt man auch ohne die Farben eine klare Trennung der beiden Kreise in einer Dimension.

Nicht-lineare Verfahren beruhen häufig auf dem Prinzip, den Datensatz in eine höhere Dimension abzubilden in der eine lineare Trennbarkeit gegeben ist und anschließend nach dem Prinzip der Hauptkomponentenanalyse vorzugehen.

2.3.1 Kernel Principal Component Analysis

Die kernel-basierte Hauptkomponentenanalyse nach Scholkopf et al. 1999 [13], kurz Kernel PCA, bildet die Beobachtungen für die Analyse zunächst in einen neuen Merkmalsraum mit

beliebiger Dimensionalität ab und ermittelt anschließend die Hauptkomponenten. Grundlegende Idee ist, dass die Daten durch die Abbildung in eine höhere Dimension linear trennbar werden. Die Transformation wird dabei mit Hilfe von Kernelfunktionen ähnlich einer Support Vector Machine durchgeführt.

Im ersten Schritt bildet man den Datensatz nicht-linear in einen neuen Merkmalsraum ab, dazu definiert man eine Abbildung Φ mit $\Phi : \mathbf{R}^d \to F, x \mapsto f$ und $x_i \in \mathbf{R}^d, i = 1, 2, \ldots, n$. Angenommen die Daten in F sind zentriert, sprich es gilt $\sum_{i=1}^{n} \Phi(x_i) = 0$ für alle n Beobachtungen. So möchte man anschließend eine Hauptkomponentenanalyse auf der Kovarianzmatrix, vgl. Formel (2.4), durchführen.

$$C = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T \tag{2.4}$$

Man sucht also die Eigenwerte $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ und entsprechenden Eigenvektoren $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ mit Eigenvektormatrix $\mathbf{V} \in F \setminus \{0\}$, sodass $\lambda \mathbf{v}_i = C\mathbf{v}_i$ gilt.

Dadurch, dass alle Eigenvektoren im durch $\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)$ aufgespannten Vektorraum liegen, gilt (2.5) für alle $i = 1, 2, \ldots, n$.

$$\lambda(\Phi(x_i) \cdot \mathbf{V}) = (\Phi(x_i) \cdot C\mathbf{V}) \tag{2.5}$$

Mit den Koeffizienten $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in}$ gilt für den i-ten Eigenvektor

$$\mathbf{v}_i = \sum_{j=1}^{n} \alpha_{ij} \Phi(x_j). \tag{2.6}$$

Definiert man eine $n \times n$ Matrix K mit

$$K_{ij} := k(x_i, x_j) = (\Phi(x_i) \cdot \Phi(x_j)^T) \tag{2.7}$$

und ersetzt in (2.5) das C durch die Kovarianzmatrix (2.4) und das $\mathbf{V}$ durch den jeweiligen Eigenvektor (2.6), so kommt man auf (2.8) (vgl. Schölkopf et al. 1999 [13][S. 2]).

$$n\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha} \Leftrightarrow n\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \tag{2.8}$$

Anschließend löst man das Eigenwertproblem in (2.8), wobei $\boldsymbol{\alpha}$ die Eigenvektoren von K mit den Einträgen $\alpha_1, \alpha_2, \ldots, \alpha_n$ sind. Dann normalisiert man die Ergebnisse $\boldsymbol{\alpha}^k$, also die k-te Komponente, welche zu nicht negativen Eigenwerten gehören, indem man fordert, dass die entsprechenden Vektoren in F durch $(\mathbf{V}^k \cdot (\mathbf{V}^k)^T) = 1$ normalisiert werden.

Um am Ende auf die Hauptkomponenten zu kommen, berechnet man die Projektion einer Beobachtung auf die in F liegenden Eigenvektoren $\mathbf{V}^k$ durch Formel (2.9).

$$(\mathbf{V}^k \cdot \Phi(x)^T) = \sum_{i=1}^{n} \alpha_i^k (\Phi(x_i) \cdot \Phi(x)^T) \tag{2.9}$$

Da Φ linear unabhängige Vektoren berechnet, gibt es keine Kovarianzen auf denen man die normale Hauptkomponentenanalyse durchführen kann. Bei jedem Auftreten von $\Phi(x)\Phi(y)$ wird auf die Kernelfunktion zurückgegriffen und die Berechnung über K ermittelt. Die Projektion durch Φ wird also nie explizit berechnet, sondern immer nur auf der Punktmatrix, die durch K gegeben ist, bestimmt. Das heißt, dass die Kovarianzmatrix in F nie konkret berechnet wird (vgl. Schölkopf & Smola 2001 [14][S. 10]). Dadurch werden im eigentlichen Sinne auch keine Hauptkomponenten direkt ermittelt, sondern nur die Abbildung auf diese.

Die Wahl der Kernelfunktion zur Bestimmung von K, kann mit den für die Support Vector Machine bewährten Funktionen (vgl. Schölkopf & Smola 2001 [14][S. 25-60]) wie z.B. dem polynomial Kernel $k(x,y) = (x \cdot y)^d$ mit $d \in \mathbb{N}$, der radialen Basisfunktion $k(x,y) = \exp(-\frac{||x-y||^2}{\gamma})$ mit $\gamma > 0$ oder der Sigmoidfunktion $k(x,y) = tanh(\kappa(x \cdot y) + \Theta)$ getroffen werden.

Um also eine nicht-lineare Hauptkomponentenanalyse durchzuführen, stellt man zunächst K auf und löst anschließend das Eigenwertproblem in (2.8), indem man K diagonalisiert. Im dritten Schritt normalisiert man die Eigenvektoren, mit der Forderung nach $(\mathbf{V}^k \cdot (\mathbf{V}^k)^T) = 1$. Anschließend kann man die Abbildung der Beobachtungen auf die Hauptkomponenten durch (2.9) bestimmen. Im Zuge der Dimensionsreduktion, wählt man dann nur die ersten p Dimensionen dieser Abbildung und erhält so den reduzierten Datensatz.

2.4 Zusammenfassung

Ist der Zweck der Dimensionsreduktion die Komprimierung oder die Robustheit zu stärken, so ist der Einsatz von linearen Verfahren geeignet. Da man mit diesen aber nur lineare Mannigfaltigkeiten entdecken kann, ist es sinnvoll, auch nicht-lineare Verfahren im Bereich der Visualisierung einzusetzen. Aus diesem Grund ist es notwendig, das Einsatzgebiet bei der Wahl eines Verfahrens zu beachten. In Abbildung 2.3 wird die Einteilung, welche nicht scharf definiert ist, nochmal aufgezeigt. Dabei gilt es zu bedenken, dass die hier vorgestellten Verfahren alle auf der sogenannten Feature Extraction basieren, also neue Merkmale erzeugen, und dem unüberwachten Lernen zuzuordnen sind. Die in der Abbildung getroffene Unterteilung ist somit nicht als komplette Trennung zu verstehen, sondern soll vielmehr eine Intuition für die Kategorisierung der hier beschriebenen Verfahren dienen. Im Bereich der Feature Selection bzw. Feature Subset Selection sind meistens die Verfahren des überwachten Lernens maßgebend.

Möchte man einen hochdimensionalen Datensatz analysieren ist ein bloßes Betrachten der Rohaten vor allem bei großen Datensätzen kaum möglich. Eine visuelle Darstellung der Daten kann dabei in vielen Fällen neue Einsichten über die Daten liefern und ist für das menschliche Gehirn intuitiver zu beurteilen, da sich die Realität auch in drei visuellen Dimensionen abspielt. Nutzt man für die Darstellung eines Datensatzes Verfahren zur Di-

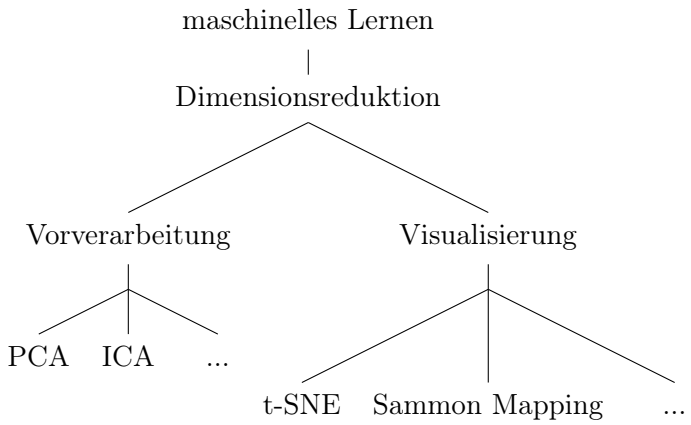


Abbildung 2.3: Einordnung Verfahren zur Dimensionsreduktion im maschinellen Lernen

mensionsreduktion aus dem linearen Bereich, bedeutet dies automatisch die Annahme eines linearen Zusammenhangs und schränkt die Analyse von vorneherein ein. Die nicht-linearen Verfahren sind frei von dieser Annahme und sind zu bevorzugen, falls diese zwischen den Attributen nicht angenommen werden kann. Beim hier vorgestellten Kernel PCA wird der Datensatz in eine höhere Dimension abgebildet, in der man die Daten dann linear trennen kann. Dieses Prinzip ist ähnlich wie bei Support Vector Machines als Kernel Trick bekannt und vereinfacht die Problemstellung, sodass wieder mit einem linearen Ansatz wie der Hauptkomponentenanalyse gearbeitet werden kann.

Im Bezug auf die Visualisierung wird dabei aber auch schnell deutlich, dass diese Verfahren die maximale Varianz des Datensatzes abzubilden versuchen. Das bedeutet grob gesagt, dass die Achsen gefunden werden, welche den größten Unterschied zwischen den Beobachtungen abbilden. Dadurch werden im Bereich des Manifold Learning, also den nicht-linearen Verfahren, häufig globale Kostenfunktionen optimiert, die sich demnach auf die globale Struktur des Datensatzes konzentrieren. Mit der Hauptkomponentenanalyse wurde bereits ein Verfahren vorgestellt, welches versucht die lineare Struktur global zu erhalten. In der Klasse der multidimensionalen Skalierung nach Kruskal et al. 1978 [7], kurz MDS, versuchen die Verfahren die Distanzen der Objekte im hochdimensionalen Raum auch im Niedrigdimensionalen zu erhalten. Es werden große Abstände durch jeweils hohe Distanzen im Graph modelliert. Dadurch konzentriert man sich bei der Optimierung auf den Erhalt der globalen Geometrie. Ist das Ziel die Struktur visuell zu analysieren, kann es intuitiv interessanter sein, Nachbarschaftsbeziehungen kenntlich zu machen. Ähnlich wie beim Clustering wäre es sinnvoll, sehr ähnliche Objekte des Datensatzes nah bei einander darzustellen und unähnliche Objekte mit größeren Abständen zu modellieren. Dazu wäre es optimal, wenn man die natürlichen Cluster auch in der Visualisierung finden könnte. Anstatt der global optimierten Kostenfunktion könnte es daher sinnvoll sein, die lokalen Nachbarschaftsstrukturen in den Vordergrund zu stellen. Dies könnte gelingen, indem man

die Ähnlichkeit der Objekte in der Visualisierung optimiert und dabei besonderen Wert auf lokale Strukturen legt.

Aus diesem Grund wird im folgenden Kapitel 3 ein Verfahren speziell zur Visualisierung von Datensätzen vorgestellt. Dabei werden Nachbarschaftsstrukturen durch die Ähnlichkeit von Beobachtungen anhand aller ihrer Merkmale optimiert und so dem Problem der Optimierung einer globalen Kostenfunktion entgegengewirkt. (Da sich diese nur auf einige wenige Merkmale beschränken.)

Nach diesem Kapitel sollte klar werden, weshalb die Unterteilung in Vorverarbeitung und Visualisierung das Gebiet nicht perfekt abdeckt. So kann man natürlich auch die ersten zwei oder drei Hauptkomponenten eines Datensatzes visualisieren und bekommt eine Darstellung in der man die maximalen Unterschiede zwischen den Objekten gut erkennen kann. Im Bezug auf den Einsatz zur Visualisierung mit dem Ziel die Struktur der Daten analysieren zu können, liefert uns diese Abbildung allerdings wenig Erkenntnis, sollte die Mannigfaltigkeit mehr als zwei Dimensionen betragen.

Kapitel 3

t-Distributed Stochastic Neighbor Embedding

Dieses Kapitel erläutert die Funktionsweise des t-Distributed Stochastic Neighbor Embedding Algorithmus, kurz t-SNE, nach van der Maaten und Hinton [20]. Das Verfahren ermöglicht die Reduktion hochdimensionaler Daten in den zwei- oder dreidimensionalen Raum. In 3.1 wird das Verfahren zunächst detailiert beschrieben. Im Abschnitt 3.2 wird eine Verbesserung der Laufzeit mit Hilfe von Approximationen dargestellt, welche den Einsatz des Verfahrens auch bei großen Datenmengen praktikabel macht. Ausschnitt 3.3 fasst die wichtigsten Eigenschaften und Erkenntnisse nochmal zusammen.

Innerhalb der Dimensionsreduktion ist das Verfahren den nicht-linearen Methoden, sprich dem Manifold Learning, zuzuordnen. Zweckmäßig kann t-SNE dem Bereich der Visualisierung zugeordnet werden, da es mit Bedacht für die visuelle Darstellung von Nachbarschaftsbeziehungen in zwei oder drei Dimensionen entwickelt wurde.

Um den folgenden Abschnitt 3.1 besser einordnen zu können und die Zusammenhänge im Voraus deutlich werden, ist hier eine kurze Vorschau, wie das Verfahren arbeitet. Nach der Definition zweier Ähnlichkeitsmaße, welche die Daten im hoch- und niedrigdimensionalen Raum beschreiben, ist das übergeordnete Ziel die Ähnlichkeit der Repräsentationen zu maximieren.

Das Verfahren lässt sich also wie folgt skizzieren:

Gegeben hochdimensionale Daten (z.B. als Koordinaten im euklidischen Raum)

Finde Repräsentation der Daten im zwei- oder dreidimensionalen Raum

Sodass die paarweisen Ähnlichkeiten zwischen den einzelnen Punkten der niedrigdimensionalen Repräsentation zu den Originaldaten maximiert wird

Einfach beschrieben versucht das Verfahren die Ähnlichkeit der Punkte im hochdimensionalen Raum zu den Ähnlichkeiten der Punkte im niedrigdimensionalen Raum zu

maximieren. Wenn zwei Punkte in den Eingabedaten nah beieinander liegen, so sollen sie auch in zwei- oder drei Dimensionen nah beieinander sein. So wird sichergestellt, dass lokale Nachbarschaftsbeziehungen in der Visualisierung deutlich werden und man im Optimalfall klar abgegrenzte Cluster erkennt.

3.1 Vorgehensweise

In den nachfolgenden Erläuterungen ist der Begriff *Datenpunkt* synonym für einen Punkt x_i des hochdimensionalen Raums aus den Originaldaten, welche in $\mathbb{R}^D$ mit $D > 3$ definiert sind. Der Begriff *Graphpunkt* verweist auf einen Punkt y_i des niedrigdimensionalen Raumes, welche grafisch dargestellt werden sollen. Ziel ist es, eine geeignete Bijektion der Datenpunkte auf die Graphenpunkte in $\mathbb{R}^2$ oder $\mathbb{R}^3$ zu bestimmen.

Um die niedrigdimensionale Repräsentation zu lernen, minimiert t-SNE die Divergenz zweier Ähnlichkeitsmatrizen. Die Konstruktion der Matrizen, welche die paarweisen Ähnlichkeiten der Datenpunkte und die der Graphpunkte bestimmen, werden nachfolgend erläutert. Im Anschluss wird das Optimierungsproblem, welches sich aus der Minimierung der Divergenz der beiden Matrizen ergibt, genauer beschrieben.

3.1.1 Paarweise Ähnlichkeiten in $\mathbb{R}^n$

Zu Beginn werden aus den hochdimensionalen Daten paarweise Ähnlichkeiten berechnet. Dieser Schritt geschieht durch das Ermitteln von bedingten Wahrscheinlichkeiten $p_{j|i}$, dafür, dass x_i den Datenpunkt x_j als nächsten Nachbarn wählt. Somit kann die bedingte Wahrscheinlichkeit als Ähnlichkeitsmaß interpretiert werden. Für die Wahl des nächstgelegenen Datenpunktes wird eine Gauß'sche Verteilung um den Punkt x_i angenommen, sodass sich die bedingte Wahrscheinlichkeit wie folgt berechnen lässt:

$$p_{j|i} = \frac{\exp\left(-d(x_i, x_j)^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-d(x_i, x_k)^2 / 2\sigma_i^2\right)} \tag{3.1}$$

Da nur die paarweisen Ähnlichkeiten unterschiedlicher Punkte für die weitere Berechnung von Interesse sind, setzt man $p_{i|i} = 0$. $d(x_i, x_j)$ beschreibt die genutzte Metrik zur Berechnung der Abstände. Oft wird diese als euklidische Norm mit $d(x_i, x_j) = ||x_i - x_j||$ oder als Kosinus-Distanz gewählt.

Da es wahrscheinlich Regionen mit unterschiedlicher Dichte gibt, macht es Sinn diese in die Berechnung der Verwandtschaft von Punkten einzubeziehen. Je dichter die Region, desto kleiner die Varianz. Dabei induziert σ_i eine Wahrscheinlichkeitsverteilung P_i über alle anderen Punkte, wobei die Entropie der Verteilung dabei proportional zur Varianz ist. Durch das Festlegen einer Perplexität durch den Nutzer wird mittels binärer Suche ein σ_i ermittelt, welches die Verteilung P_i produziert (vgl. [20][S. 2582]).

$$Perp(P_i) = 2^{H(P_i)} \tag{3.2}$$

Die Perplexität wird hier mit Hilfe der Shannon Entropie (3.3) definiert, gesucht wird also ein σ_i, das die vorher festgelegte Perplexität (3.2) besitzt.

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{i|j} \tag{3.3}$$

Die Perplexität lässt sich als Anzahl der effektiven Nachbarn interpretieren, sie ist ein vom Nutzer festzulegender Parameter und sollte nach van der Maaten zwischen 5 und 50 liegen (vgl. van der Maaten & Hinton [20][S. 2582]). Sie gibt ein Maß an, wieviele Punkte als nah behandelt werden sollen, sodass die Bandbreite von P_i so gewählt wird, dass sie dieses Maß erreicht.

In der ursprünglichen SNE Variante von Hinton und Roweis 2003 [4] besitzen die paarweisen Ähnlichkeiten mit Ausreißern nur sehr geringe Werte. Dadurch ist ihr Einfluss auf die später zu optimierende Kostenfunktion nur sehr gering und es wird schwierig, eine repräsentative Position für die jeweiligen Graphpunkte der Ausreißer zu bestimmen. Durch die Nutzung einer symmetrischen Verteilung zur Berechnung der Ähnlichkeiten im hochdimensionalen Raum nach Cook et al. 2007 [3], tragen alle Datenpunkte x_i mindestens $\sum_j p_{ij} > \frac{1}{2n}$ zur Kostenfunktion bei und verbessern somit die Darstellung lokaler Nachbarschaftsstrukturen. Dabei ist n die Anzahl der Datenpunkte. Das Verfahren wird als symmetrisches SNE bezeichnet, da $\forall i, j : p_{ij} = p_{ji}$ gilt.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{3.4}$$

Die Verteilung p_{ij} wird dann nach (3.4) berechnet und führt darüber hinaus dazu, dass der Gradient zur Optimierung der Kostenfunktion einfacher berechnet werden kann. Insgesamt wird also die symmetrische Ähnlichkeitsmatrix P für den hochdimensionalen Raum mit (3.4) aufgestellt. Demnach beschreibt jedes p_{ij} die Wahrscheinlichkeit, dass x_i den Datenpunkt x_j als nächsten Nachbarn wählt und es gelten $p_{ii} = 0$, sowie $p_{ij} = p_{ji}$.

3.1.2 Repräsentation im niedrigdimensionalen Raum

Für die Repräsentation im niedrigdimensionalen Raum definiert man eine ähnliche Verteilung in Form einer Ähnlichkeitsmatrix. Auch hier werden die Ähnlichkeiten wieder als Wahrscheinlichkeiten interpretiert. Nach Hinton und Roweis 2003 [4] wurden die Nachbarschaften wieder mit Hilfe einer Gauß'schen Verteilung bestimmt. Durch das Festlegen der Varianz auf $\frac{1}{2}$ ergibt sich die Formel (3.5), wie sie im SNE [4][S. 2] verwendet wird.

$$q_{ij} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq i} \exp(-||y_i - y_k||^2)} \tag{3.5}$$

Um das aus der Wahl der Gauß'schen Verteilung entstehende Problem besser zu verstehen, wird in Cook et al. 2007 [3][S. 2] eine physikalische Interpretation des hiernach zu berechnenden Gradientenabstiegs präsentiert. Nach dieser Analogie ist jeder Graphpunkt y_i mit jedem anderen Punkt y_j durch eine Feder verbunden. Die Richtung wird dabei durch $y_i - y_j$ bestimmt und die Stärke der durch die Feder resultierenden Kraft ist proportional zu $||y_i - y_j||$. Durch die einwirkenden Kräfte der anderen Punkte wird so die Position für jedes y_i bestimmt.

Ausgehend von einem d-dimensionalen Raum, kann es genau $d + 1$ zueinander äquidistante Punkte geben. Es gibt dabei aber keine Möglichkeit diese wirklichkeitsgetreu in weniger als d Dimensionen darzustellen. Ein weiteres Problem ergibt sich aus der Modellierung ähnlicher paarweiser Distanzen in zwei Dimensionen. Angenommen ein Datenpunkt x_i besitzt viele Nachbarn mit ähnlichem Abstand. Möchte man diese Distanzen in zwei Dimensionen erhalten, so benötigt man viel Platz. Man müsste die ähnlichen Punkte kreisförmig um den entsprechenden Graphpunkt verteilen und einen großen Radius wählen, damit sich die Punkte nicht überlappen. Wenn man nun etwas weiter entfernte Datenpunkte von x_i darstellen möchte, müssen diese in zwei Dimensionen sehr weit weg platziert werden, damit das Verhältnis beibehalten wird.

Da jeder dieser weit entfernten Punkte eine kleine Kraft auf den entsprechenden Graphpunkt y_i auswirkt, wird die Darstellung auf Grund der hohen Anzahl dieser Kräfte in den Ursprung gedrückt. Es entsteht das *Crowding Problem*, durch welches die optischen Abtrennungen einzelner Bereiche verschwimmen und keine Cluster mehr erkennbar sind (vgl. [20][S. 2584f]).

Um dem Crowding Problem entgegenzuwirken, wurde in Cook et al. 2007 das sogenannte UNI-SNE vorgestellt (vgl. [3][S. 5]). Hier werden den q_{ij} im wesentlichen kleine Werte durch eine gleichmäßige Hintergrundabbildung zugerechnet, sodass $q_{ij} > p_{ij}$ für moderat weit entfernte Punkte gilt. Dadurch resultiert in der späteren Gradientenberechnung eine leichte Abstoßung dieser Paare.

Mit diesen in UNI-SNE beschriebenen Änderungen lässt sich die Kostenfunktion allerdings nicht mehr direkt optimieren. Falls mehrere Teile eines Clusters zu Beginn der Optimierung getrennt werden, gibt es später keine Kräfte mehr, die diese wieder zusammenführen. Dieser Effekt liegt an der Tatsache, dass zwei weit entfernte Punkte ihr q_{ij} durch die Hintergrundabbildung bekommen und selbst wenn ihr p_{ij} groß ist, keine anziehende Kraft mehr resultieren kann (vgl. [20][S. 2585]).

Um dem Crowding Problem nun entgegenzuwirken, ohne die Einsatzfähigkeit des Verfahrens zu mindern, nutzt man beim t-SNE Verfahren eine studentsche t-Verteilung für die niedrigdimensionale Repräsentation der Datenpunkte. Diese endlastige Verteilung wird eingesetzt, um Abstände in Wahrscheinlichkeiten zu berechnen. Dadurch werden moderat distanzierte Datenpunkte durch etwas größere Abstände in Graphpunkten modelliert, wodurch die ungewünscht einwirkenden Kräfte eigentlich unähnlicher Punkte eliminiert

werden [20][S. 2585]. Somit kann die eigentliche Ursache für das Auftreten des Crowding Problem behoben werden.

Mit der studentschen t-Verteilung mit einem Freiheitsgrad, äquivalent zur Cauchy-Verteilung, berechnet sich die gemeinsame Verteilung q_{ij} durch Formel (3.6). Der Vorteil gegenüber der Gauß'schen Verteilung liegt in der endlastigen Verteilung einer studentschen t-Verteilung, da diese weit entfernten Punkten nur noch sehr geringe anziehende bzw. leicht abstoßende Wirkung haben und somit die 'unnötigen' Kräfte, welche die Punkte im Zentrum der Darstellung halten, vermieden werden.

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||y_k - y_l||^2)^{-1}} \tag{3.6}$$

Die studentsche t-Verteilung kann als unendliche Zusammensetzung von Gauß'schen Verteilungen mit unterschiedlicher Varianz geschrieben werden und macht die Wahl der endlastigen Verteilung somit plausibel. Dazu ist die Berechnung schneller, da keine Exponentialfunktion berechnet werden muss (vgl. [20][S. 2586]).

Insgesamt legt t-SNE also Wert auf darauf, sich stark unterscheidende Datenpunkte mit großen paarweisen Abständen und ähnliche Datenpunkte durch kleine paarweise Abstände zu modellieren.

3.1.3 Optimierungsproblem

Angenommen die Datenpunkte x_i und x_j modellieren die Graphenpunkte y_i und y_j optimal, so wären ihre Wahrscheinlichkeiten p_{ij} und q_{ij} identisch. Ausgehend von dieser Gleichheit ist nun das Ziel, den Abstand der Ähnlichkeitsmatrizen P und Q zu minimieren, um die hochdimensionalen Daten möglichst effektiv im niedrigdimensionalen Raum abzubilden. Die Minimierung des Abstandes kann mit Hilfe der *Kullback-Leibler Divergenz*, kurz KL-Divergenz, über P und Q beschrieben werden. Die Wahrscheinlichkeitsverteilung P mit $n \times n$ Einträgen ergibt sich aus p_{ij}, welche die einzelnen Einträge definiert. Q wird analog über q_{ij} bestimmt. Diese Herausforderung stellt ein Optimierungsproblem dar, in welchem die Kostenfunktion (3.7) optimiert werden soll. Die KL-Divergenz erhält dabei lokale Strukturen besonders gut, da der Kostenfunktion hohe Werte zugerechnet werden, falls eng beisammenliegende Datenpunkte durch die Bijektion als Graphenpunkte weit auseinander liegen. (D.h. wenn p_{ij} groß und q_{ij} klein, ist das Summenglied groß.)

$$C_{sym} = KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.7}$$

Zur Lösung wird die Kostenfunktion mit dem Gradientenverfahren nach y_i optimiert, um eine entsprechende Repräsentation für die niedrigdimensionalen Graphpunkte zu finden. Diese Berechnung erfolgt numerisch und löst die Gleichung (3.8). Da die Kostenfunktion (3.7) nicht konvex ist, können unterschiedliche lokale Minima gefunden werden. Daher

ist es sinnvoll mehrere Durchläufe des Gradientenverfahrens zu errechnen und die Lösung mit der niedrigsten KL-Divergenz auszuwählen.

Sollte ein großes p_{ij} durch ein kleines q_{ij} modelliert werden, so ist das Summenglied in der KL-Divergenz bzw. der Fehler groß. Sollte allerdings ein kleines p_{ij} durch ein großes q_{ij} modelliert werden, so ist der addierte Wert relativ klein. Durch dieses Verhalten der KL-Divergenz wird hauptsächlich die lokale Struktur der hochdimensionalen Daten in den zwei- oder drei Dimensionen erhalten. (Vgl. anders als bei der Hauptkomponentenanalyse, wo die Kostenfunktion auf die globale Struktur abzielt, indem der maximale Unterschied gesucht wird.)

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3.8}$$

Dabei kann der Gradientenabstieg wie folgt interpretiert werden: Man versucht y_i so zu bewegen, dass q_{ij} maximale Ähnlichkeit zu p_{ij} besitzt, indem C dabei möglichst klein ist. $(y_i - y_j)$ ist dabei symbolisch für die physikalische Interpretation einer Feder zwischen y_i und y_j. Angenommen $(p_{ij} - q_{ij})$ wäre gerade Null, so wäre die Ähnlichkeit der Objekte x_i und x_j perfekt modelliert und y_i müsste nicht bewegt werden. Ist der Term $(p_{ij} - q_{ij})$ positiv, also $p_{ij} > q_{ij}$, so gäbe es eine Anziehung in der Stärke von $(1 + ||y_i - y_j||^2)^{-1}$. Falls $p_{ij} < q_{ij}$ wäre die Kraft abstoßend. Die Summe symbolisiert dabei die resultierende Kraft auf y_i durch alle anderen y_j.

3.1.4 Algorithmus

Der Algorithmus 3.1 beschreibt das t-Distributed Stochastic Neighbor Embedding Vorgehen als Pseudocode. Durch kleine Schritte nähert sich t-SNE einer optimalen Lösung an. Die Eingabe ist dabei ein Datensatz $\mathcal{X}$ mit hochdimensionalen Daten. Parameter mit Auswirkung auf die Optimierung sind die Anzahl der Iterationen T, die Lernrate η für die Gewichtung der einzelnen Schritte und der Impuls $\alpha(t)$ der eine Art Trägheit definiert. Für die Berechnung des initialen Ergebnisses $\mathcal{Y}^{(0)}$ eignen sich die ersten zwei bzw. drei Hauptkomponenten (per PCA/Hauptkomponentenanalye bestimmt) oder eine zufällig berechnete Position für jeden Datenpunkt. Die Wahl der Perplexität wirkt sich dabei auf die Kostenfunktion aus und kann als Maß für die Anzahl der effektiven Nachbarn interpretiert werden.

3.2 t-SNE auf großen Datensätzen

Mit der Berechnung des Gradienten (3.8) wird schnell deutlich, dass dieser der Flaschenhals des Algorithmus ist. Weil die Ähnlichkeiten für alle $n \times n$ Paare von Punkten berechnet werden müssen, skaliert die Formel quadratisch in der Anzahl der Datenpunkte. Mit quadratischer Laufzeit ist der Einsatz des t-Distributed Stochastic Neighbor Embeddings nicht

Algorithmus 3.1 t-Distributed Stochastic Neighbor Embedding Pseudocode

Eingabe: Datensatz $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, Perplexität Perp, Anzahl der Iterationen T, Lernrate η, Impuls $\alpha(t)$

Ausgabe: niedrigdimensionale Repräsentation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.

berechne paarweise Verwandschaften $p_{j|i}$ mit Perplexität Perp (Formel (3.1))
nutze symmetrische Verteilung (Formel (3.4))
erzeuge initialies Ergebnis $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$
for $t = 1$ **to** T **do**
  berechne niedrigdimensionale Verwandschaften q_{ij} (Formel (3.6))
  berechne Gradienten $\frac{\partial C}{\partial \mathcal{Y}}$ (Formel (3.8))
  setze $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$
end for

für große Datensätze geeignet. In einem weiteren Paper von van der Maaten 2014 [19] wird daher ein Verfahren zur Beschleunigung der Berechnung des Gradienten vorgestellt, um die Laufzeit auf $\mathcal{O}(n \log n)$ zu reduzieren.

Dazu werden die beiden Verteilungen P und Q, welche die Ähnlichkeiten der Eingabedaten und der niedrigdimensionalen Repräsentation berechnen, approximiert und beschleunigen so die Gradientenberechnung. Im ersten Schritt wird dazu die Ähnlichkeitsmatrix für die Datenpunkte mit Hilfe eines *Metric Trees* approximiert, anschließend erfolgt eine Abschätzung der Ähnlichkeitsmatrix für die Graphpunkte durch die sogenannte *Barnes-Hut Approximation*.

3.2.1 Metric Tree Approximation

Dadurch, dass die Wahrscheinlichkeiten p_{ij} mit Hilfe einer Gauß'schen Verteilung berechnet werden, besitzen sich stark unterscheidende Datenpunkte nur sehr kleine Werte (vgl. Gradientenberechnung (3.10)). Also haben weit auseinanderliegende Datenpunkte nur geringen Einfluss auf die Position des entsprechenden Graphpunktes. Aus dieser Beobachtung heraus wird eine spärliche Matrix ermittelt, welche weit genug entfernte Punkte nicht mehr in die Berechnung einbezieht und deren $p_{ij} = 0$ setzt. Daher wird die Ermittlung der paarweisen Ähnlichkeiten so angepasst, dass nur noch die $\lfloor 3u \rfloor$ nächsten Nachbarn berücksichtigt werden (vgl. [19][S. 6 4.1]). Dabei ist u die vorher vom Nutzer festgelegte Perplexität. Nachdem also für jedes Objekt i aus den Eingabedaten die jeweils nächsten Nachbarn $\mathcal{N}_i$ ermittelt wurden, werden die bedingten Wahrscheinlichkeiten mit Hilfe von Formel (3.9) aufgestellt.

$$p_{j|i} = \begin{cases} \frac{\exp\left(-d(x_i,x_j)^2/2\sigma_i^2\right)}{\sum_{k \in N_i} \exp\left(-d(x_i,x_k)^2/2\sigma_i^2\right)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{sonst} \end{cases} \tag{3.9}$$

Das bedeutet, dass die Ähnlichkeit von x_j zu x_i nur berechnet wird, falls x_j in der Menge der $\lfloor 3u \rfloor$ nächsten Nachbarn liegt. Danach wird die symmetrische Ähnlichkeitsverteilung wie gewohnt durch (3.4) bestimmt. Um jeweils die nächsten Nachbarn zu finden, wird ein *Vantage Point Tree* auf den Eingabedaten konstruiert und darauf anschließend eine exakte Nachbarschaftssuche durchgeführt. Der nach Yianilos 1993 [21] beschriebene Vantage Point Tree enthält in jedem Knoten ein Objekt und einen Radius. Im mehrdimensionalen Raum wird der Radius für eine Kugel, zentriert um das im Knoten enthaltene Objekt, interpretiert. In den jeweils linken Kindknoten aller nicht-Blätter werden Objekte eingeordnet, die innerhalb des Radius liegen. Dem rechten Kindknoten werden die Objekte außerhalb des Radius untergeordnet. Der Vorteil dieses Verfahrens, welches oft auch als Metric Tree nach Uhlmann 1991 [18] zitiert wird, liegt in der Ermittlung der nächsten Nachbarn $\mathcal{N}_i$ in $\mathcal{O}(n \log n)$ Zeit (vgl. [19][S. 6 4.1]). Dabei werden die Bedingungen an die einzuordnenden Objekte auf die Eigenschaften eines beliebigen metrischen Raumes beschränkt. Das bedeutet, dass neben einer Distanzfunktion d, welche den Abstand zweier Objekte zueinander beschreibt, noch die Eigenschaften der positiven Definitheit, der Symmetrie und der Geltung der Dreiecksungleichung gefordert werden. So wird die Berechnung der Verteilung P, welche die paarweisen Ähnlichkeiten der Eingabedaten repräsentiert, approximiert und vereinfacht so die Berechnung des Gradienten, wie im nächsten Abschnitt genauer erläutert.

3.2.2 Barnes-Hut Approximation

Durch die Approximation von P mit Hilfe des oben beschriebenen Verfahrens, kann die Berechnung des Gradienten bereits deutlich beschleunigt werden. Teilt man den Gradienten (3.8) in anziehende und abstoßende Kräfte F_{attr} und F_{rep} wie in (3.10), lässt sich die Vereinfachung gut beschreiben. Da nur noch über alle nicht-null Werte von P summiert werden muss und $q_{ij}Z = (1 + d(y_i, y_j)^2)^{-1}$ dabei in $\mathcal{O}(1)$ ermittelt werden kann, besitzt F_{attr} nur noch lineare Laufzeit (vgl. [19][S. 7 4.2]).

$$\frac{\delta C}{\delta y_i} = 4(F_{attr} + F_{rep}) = 4\left(\sum_{j \neq i} p_{ij} q_{ij} Z(y_i - y_j) - \sum_{j \neq i} q_{ij}^2 Z(y_i - y_j)\right) \tag{3.10}$$

Der zweite Teil aus (3.10), sprich die abstoßenden Kräfte F_{rep}, benötigen bisher ebenfalls quadratische Laufzeit (wegen q_{ij}^2). Mit Hilfe des von Barnes und Hut 1986 [1] vorgestellten Verfahren werden auch die abstoßenden Kräfte abgeschätzt und so eine Laufzeit von $\mathcal{O}(n \log n)$ erzielt. Diese hatten die Methodik ursprünglich zur Berechnung von Gravitationskräften in einem N-Körpersystem für die Astrophysik entwickelt.

Angenommen es gibt drei beliebige Graphpunkte y_i, y_j und y_k für die gilt $||y_i - y_j|| \approx ||y_i - y_k|| \gg ||y_j - y_k||$, sprich die Punkte mit dem Index i und j haben in etwa den gleichen Abstand zueinander wie i und k, sowie der Abstand zwischen j und k ist viel kleiner als zwischen i, j und i, k. So wäre der Beitrag von y_j und y_k zu F_{rep} in Bezug auf y_i circa

gleich. Diese Beobachtung macht sich das Verfahren von Barnes und Hut zunutze und geht dabei wie folgt vor.

Ausgehend von einer Abbildung in zwei Dimensionen, wird ein *Quadtree* nach Samet 1982 [12] auf der zu berechnenden niedrigdimensionalen Repräsentation konstruiert. Mit einer Tiefensuche über den entstandenen Baum wird an jedem Knoten entschieden, ob dieser als Repräsentation für alle seine Kindknoten dienen kann. So müssen nicht mehr die paarweisen Abstände aller Graphpunkte berechnet werden, sondern falls Punkte weit genug entfernt sind, kann der Einfluss auf F_{rep} für alle umliegenden Punkte geschätzt werden.

Ein Quadtree unterteilt die Graphpunkte in sogenannte Zellen, welche jeweils einen Quadranten repräsentieren, der den jeweiligen Elternknoten in vier Teile zerlegt. Das heißt, dass jeder Knoten maximal vier Kinder besitzt. Der Wurzelknoten repräsentiert dabei das komplette Embedding. Die Blätter stehen für Zellen, die maximal einen Punkt enthalten. In den Knoten werden der Schwer- bzw. Mittelpunkt y_{cell} und die Anzahl N_{cell} der Punkte in der Zelle gespeichert. Der Mittelpunkt errechnet sich dabei aus allen in der Zelle enthaltenen Punkten. Für die Konstruktion des Quadtrees für N Punkte aus dem Embedding benötigt man $\mathcal{O}(N)$ Zeit. Die Punkte werden nacheinander in den Baum eingefügt, Blattknoten werden aufgeteilt, falls ein zweiter Punkt innerhalb der Zelle eingefügt wird. An den entsprechenden Knoten werden dann y_{cell} und N_{cell} rekursive bis zum Wurzelknoten aktualisiert (vgl. [19][S. 7 f]).

Damit nun nicht mehr alle paarweisen Ähnlichkeiten berechnet werden müssen, kann mit Hilfe einer Tiefensuche über den Quadtree diejenige Zelle gesucht werden, die ausreichend klein und weit genug weg von y_i ist, sodass alle anderen Punkte in der Zelle circa den gleichen Einfluss auf y_i besitzen. Mathematisch formuliert können die Kräfte (für alle y_j der entsprechenden Zelle) dann durch $N_{cell}q_{i,cell}^2 Z(y_i - y_{cell})$ abgeschätzt werden. Dabei entspricht N_{cell} der Anzahl der Punkte in der Zelle, y_{cell} dem Schwerpunkt und es gilt $q_{i,cell}Z = (1 + ||y_i - y_{cell}||^2)^{-1}$.

Mit Hilfe des Quadtrees kann F_{rep} nun mit Formel (3.11) bestimmt werden. Dabei wird $F_{rep}Z = -q_{ij}^2 Z^2(y_i - y_j)$ durch Tiefensuche über den Baum geschätzt, indem an jedem Knoten entschieden wird, ob dieser als Repräsentation aller Punkte in der entsprechenden Zelle dienen kann. $Z = \sum_{i \neq j}(1 + ||y_i - y_j||^2)^{-1}$ wird analog abgeschätzt.

$$F_{rep} = \frac{F_{rep}Z}{Z} \tag{3.11}$$

Die Entscheidung, ob ein Knoten als Zusammenfassung aller Punkte unter ihm herangezogen werden kann, lässt sich mit Formel (3.12) bestimmen (vgl. [19][S. 9]). Diese vergleicht den Abstand zwischen Punkt und Zelle mit der Größe der Zelle. Die Größe r_{cell} wird dabei durch die Diagonale ermittelt.

$$\frac{r_{cell}}{||y_i - y_{cell}||^2} < \theta \tag{3.12}$$

Das bedeutet, dass man durch den Parameter θ einen Schwellwert für die Berechnung besitzt, mit dem man zwischen Schnelligkeit und Genauigkeit wählen kann. Das Theta ist also ein weiterer Parameter, den der Nutzer zum Anpassen des Ergebnisses beeinflussen kann. Je größer θ, desto schneller und spärlicher wird die Berechnung der abstoßenden Kräfte und damit auch des Gradienten.

3.3 Zusammenfassung

Das t-Distributed Stochastic Neighbor Embedding nähert sich in kleinen Schritten einer optimalen Darstellung hochdimensionaler Daten in zwei oder drei Dimensionen an. Dazu optimiert es per Gradientenabstieg die Kullback-Leibler Divergenz zwischen den Verteilungen im hoch- und niedrigdimensionalen Raum, welche die paarweisen Ähnlichkeiten der Punkte zueinander bestimmen. Dadurch wird sichergestellt, dass sich ähnliche Datenpunkte auch im resultierenden Graphen ähnlich, sprich nah beieinander, sind. So sollen die Strukturen des Datensatzes optisch greifbar werden und zur Interpretation der möglicherweise sichtbaren Cluster beitragen.

Mit Hilfe der Approximationen der beiden Verteilungen in den Abschnitten 3.2.1 und 3.2.2 wird eine praktische Anwendung auch auf großen Datensätzen ermöglicht. Mit einer Laufzeit von $\mathcal{O}(n \log n)$ beträgt die Visualisierung des MNIST Datensatzes[1], 70.000 Datenpunkte, mit vorheriger Vorverarbeitung per PCA auf 30 Dimensionen, circa 13 Minuten (vgl. [19])[S. 14f].

[1] Bilder handgeschriebener Ziffern, verfügbar unter http://yann.lecun.com/exdb/mnist/index.html

Kapitel 4

Zwischenstand

Das in Kapitel 3 beschriebene t-SNE ist für sich schon ein sehr umfangreiches Verfahren, um hochdimensionale Datensätze zu visualisieren. Durch die Approximation der Ähnlichkeitsmatrizen ist der Algorithmus auch auf großen Datenmengen effizient einsetzbar. Demnach stellt sich die Frage, ob das Verfahren alleine ausreichend ist, um die in der Einleitung beschriebenen Anforderungen zu erfüllen?

Um die Frage zu beantworten geht der folgende Abschnitt kurz auf den relNet Datensatz ein. Anschließend wird dieser per t-SNE visualisiert und eine kurze Evaluation im Hinblick auf die Erkenntnisse aus dieser Darstellung durchgeführt.

4.1 relNet Datensatz

Im Titel der Arbeit wird auf große Dokumenten-Kollektionen verwiesen, welche im Fall des relNet Projektes eine Menge an Diskussionsrunden sind. Damit diese sogenannten Threads, also Diskussionsrunden, maschinell verarbeitet werden können, repräsentiert man sie durch Vektoren. In dieser Repräsentation enthalten die Vektoren für jedes Wort, welches mindestens zehn mal unter allen Threads vorkommt, einen binären Eintrag, ob dieses Wort im jeweiligen Thread enthalten ist oder nicht. Da für jedes zehn mal vorkommende Wort im Wortschatz aller Diskussionen ein einzelnes Attribut vorgesehen wird, erhält man beim relNet Projekt eine Dimensionalität von 57.621 Merkmalen. Diese hochdimensionalen Daten werden anschließend durch ein Document-Embedding nach Le & Mikolov 2014 [8] auf 300 Dimensionen reduziert. Mit Document-Embedding ist dabei das in einem neuen Merkmalsraum repräsentierte Dokument, also die Diskussionsrunde gemeint. Das bedeutet, dass Dokumente als ähnlich interpretiert werden, wenn sie auf einen ähnlichen Wortschatz zurückgreifen. Dieses sogenannte Embedding beschreibt die Kollektion von Dokumenten aus dem spärlich besetzten Raum mit 57.621 Merkmalen nun in einem niedrigdimensionaleren Raum mit 300 Merkmalen. Wenn im folgenden der relNet Datensatz referenziert wird, ist das auf allen 23.718 Diskussionsrunden berechnete Embedding mit 300 Dimensionen ge-

meint. Für die Details zum Verfahren, sowie Feinheiten im Umgang mit Rechtschreib- oder Tippfehlern, siehe ebenfalls Le & Mikolov 2014 [8].

4.2 t-SNE Visualisierung

Nach der Übersicht des Datensatzes zeigt Abbildung 4.1 die Berechnung per t-SNE. Es lassen sich schon einige Details über die Struktur erkennen. So gibt es im oberen Bereich eine klar abgetrennte Gruppe, am linken und rechten Bildrand sind ebenfalls klare Abspaltungen zu erkennen. Am unteren Rand sind darüber hinaus zwei weitere Abspaltungen zu erahnen, welche allerdings nicht klar abgetrennt sind. Diese geben bereits Aufschluss über die Struktur und deuten auf bestimmte Interessensgruppen, Themen oder ähnlichem hin. Zusätzlich lassen sich einige dichtere Regionen erkennen, welche womöglich ebenfalls wichtige Informationen beinhalten, sich aber nicht so klar von den anderen Gruppierungen differenzieren lassen. Im Anhang A.1 sind die Visualisierungen der relNet Dokumenten-Kollektion mit anderen Verfahren aufgeführt, um einen visuellen Vergleich zu t-SNE zu ermöglichen.

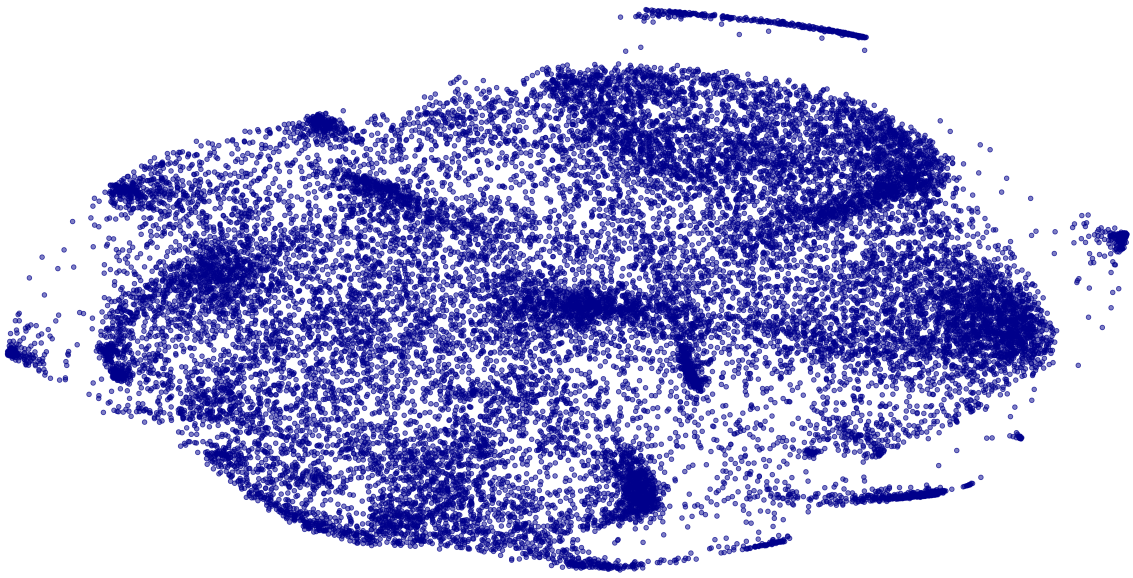


Abbildung 4.1: t-SNE Visualisierung relNet Datensatz

Die t-SNE Visualisierung zeigt bereits einige klare Abgrenzungen, darüber hinaus aber auch, dass sie alleine nicht ausreichend ist, um die Struktur tiefer gehend zu analysieren. So wäre es wahrscheinlich aufschlussreich, eine identifizierte Gruppe weiter aufzuspalten und die Struktur innerhalb eines bestimmten Bereiches zu untersuchen. Insgesamt wird klar, dass eine interaktive Form der Darstellung wünschenswert ist, damit man z.B. die gerade angesprochenen Gruppen weiter untersuchen kann. Interaktiv meint in diesem Fall Funktionen für den Betrachter der Abbildung, welche Punkte zu ihren jeweiligen Objekten

der Eingabemenge zuordnen. Weitere denkbare Funktionen, die zum Verständnis bzw. der Analyse dienen können, sind eine Art Zoom-Funktion, um bestimmte Bereiche genauer zu erkunden und eventuell eine Suchfunktion, sodass auch eine Verbindung von Eingabeobjekt zum Punk im Graph besteht. Demnach motiviert diese erste Visualisierung die weitere Entwicklung eines geeigneten Ansatzes, mit dem sich die Abbildung weiter untersuchen lässt.

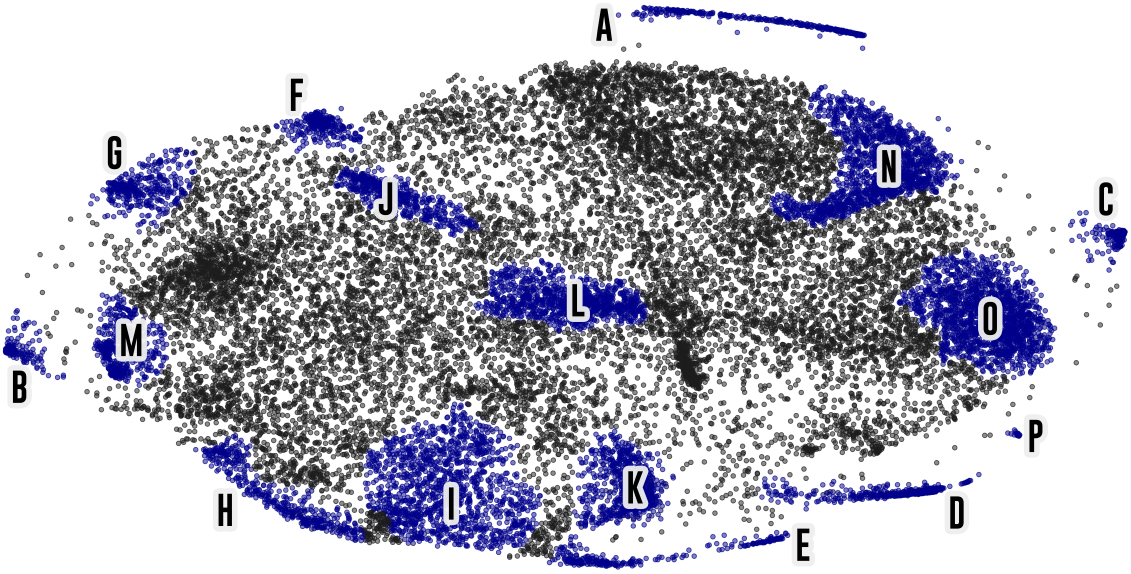


Abbildung 4.2: t-SNE Visualisierung relNet Datensatz mit Markierungen

In einer ersten Umsetzung in interaktiver Form, lassen sich die Punkte zu ihren jeweiligen Diskussionsrunden zuordnen, der Inhalt eines Beitrages in der Diskussion anzeigen und ihre nächsten Nachbarn berechnen. Dadurch lassen sich die oben angesprochenen interessanten Bereiche bereits oberflächlich untersuchen. Abbildung 4.2 zeigt die Visualisierung des Datensatzes. Die blauen Bereiche markieren die interessanten Bereiche, die Tabelle 4.1 beschreibt die markierten Bereiche und gibt so Aufschluss über die groben Themengebiete.

Innerhalb der Markierungen kann man bereits interessante Zusammenhänge feststellen. So sind Threads in denen es um Anti-Virus Computerprogramme geht nah am Bereich Gesundheit in dem es um Viruserkrankungen geht. Im Bereich H (Technik) kann man bestimmte Unterthemen feststellen, so geht es im linken Bereich eher um Handys und rechts um Computer und Software.

Aus dem gerade beschriebenen Beispiel und den markierten Themengebieten lässt sich bereits schließen, dass es einen sinnvollen Zusammenhang zwischen den Objekten gibt. Die Anordnung ergibt bereits ganz grobe Themenbereiche und gruppiert Objekte dementsprechend. Für den Anspruch die Struktur zu analysieren, reicht diese einfache Form allerdings nicht aus. Es muss sichergestellt werden, dass man einzelne Themengebiete bzw. Bereiche genauer untersuchen kann. So wäre es z.B. interessant zu sehen, in welche Themen sich

Bereich	Themengebiet/Beschreibung
A	verschobene Beiträge
B	Bücher (Autor- und Titelempfehlungen, ISBN Nummern)
C	Glückwünsche, Wünsche und Segen
D	Rezepte, Kochen
E	Neugeborene, Schwangerschaft, Glückwünsche zum Kind
F	Musik, Songtexte, Youtube-Verlinkungen
G	Umzug, Wohnungen (Mietpreise, Gemeinden)
H	Technik (Handys, Computer)
I	Gesundheit, Krankheiten, Operationen (vor allem technische Hintergründe)
J	Rechtsfragen (Gericht, Anwalt, Recht, Widerspruch)
K	Technikfragen und -probleme im Forum
L	Beziehungen
M	Diskussionsrunden/Smalltalk (kein abgrenzbares Themengebiet erkennbar)
N	Politik (EU, UN, Wahlen, Lobbyismus)
O	Glaube, Bibelzitate
P	Hochsensibilität (hochsensible Person, kurz HSP)

Tabelle 4.1: Bereiche und Themengebiete

der Bereich F von Musik, Songtexten und Youtube-Verlinkungen unterteilen lässt und ob man innerhalb dieses Bereiches eventuell neue ungeahnte Gruppierungen finden kann. Wie lassen sich die einzelnen Bereiche also weiter gezielt untersuchen? Auch vom relNet Projekt abstrahiert stellt sich die Frage nach Struktur und wie diese auf unterschiedlichen Ebenen zusammenhängt.

Ein Ansatz für diese Problemstellung ist die einzelnen Bereiche erneut per t-SNE zu visualisieren. Dabei würde man die Eingabemenge für das Verfahren auf die Daten des jeweils interessanten Bereiches beschränken. Dieses Vorgehen könnte dann rekursiv fortgesetzt werden, bis man eine klar abgetrennte Struktur in einzelne distinkte Gruppen erkennen kann. Dabei entsteht eine Art Verschachtelung von Themengebieten in Form einer Hierarchie. Durch eine vom Programm bereitgestellte hierarchische Unterteilung in ähnliche Objekte könnte der Nutzer in der Analyse unterstützt werden. Dadurch entsteht dann eine strukturgestützte Zoom-Funktion.

Die dafür benötigte Hierarchie lässt sich dabei z.B. per *hierarchischem Clustering* bestimmen. Dieses konstruiert einen binären Baum, welcher dann die gewünschte Verschachtelung der Eingabeobjekte bereitstellt. Anhand der vom Clustering berechneten Hierarchie kann dann eine Navigation durch die Visualisierung erfolgen. Diese wird in jedem Schritt spezifischer, da sie sich auf immer weniger Objekte bezieht. Die aktuelle Ansicht kann jeweils in die zwei nächst größten Bereiche unterteilt werden und dient somit als eine Art

Nutzerführung in immer differenziertere Gebiete. Das folgende Kapitel 5 gibt einen kurzen Einblick in das Clustering und erklärt danach eine Variante des hierarchischen Clusterings, welches im weiteren Verlauf zum Einsatz kommt.

In wie weit diese Kombination aus Clustering und Visualisierung in der Aufgabe der Strukturanalyse weiterhilft, wird in Kapitel 6 aufgegriffen und anhand des relNet Datensatzes exemplarisch untersucht.

Kapitel 5

Clustering

Die Lernaufgabe Clustering löst das Problem, einen Datensatz in k Cluster bzw. Gruppen zu unterteilen. Es ist dem unüberwachten maschinellen Lernen unterzuordnen, da die sogenannten *Clusterlabel* ausschließlich aus der Eingabemenge bestimmt werden. Die gesuchte Abbildung nach Y soll in k unterschiedliche Klassen abbilden, also $|Y| = k$ mit $y_1, y_2, \ldots, y_k$. Ein Clusterlabel ist dabei ein Wert aus Y. Die eigentliche Lernaufgabe besteht demnach im Finden einer Zuordnung bzw. Abbildung $f : X \to Y$, sodass jedem Eingabeobjekt ein Clusterlabel zugeordnet, also $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_k)$ gefunden wird. Die Klassen bzw. Clusterlabel in Y sind dabei unbekannt und müssen durch das Verfahren erst bestimmt werden. Die entsprechende Funktion f welche nach Y abbildet wird durch ein entsprechendes maschinelles Verfahren gelernt.

Grundlegend unterscheiden Steinbach et al. 2013 [15][S. 490ff] die Ansätze in *partitionierend* und *hierarchisch*. Das übergeordnete Ziel ist dabei immer, dass die Ähnlichkeit der Objekte innerhalb eines Cluster maximal ist. Zwischen den Clustern sollte die Ähnlichkeit minimal sein, interpretiert man die Ähnlichkeit als Distanz, so ist ein maximaler Abstand gesucht.

5.1 Partitionierend vs. hierarchisch

Beim partitionierenden Clustering wird die Eingabemenge demnach genau so in k unterschiedliche Cluster unterteilt, dass die gerade genannte Eigenschaft optimiert wird. Der hierarchische Ansatz hingegen baut eine Hierarchie von Clustern z.B. in Form eines binären Baumes auf. Jedes Cluster besteht dann wieder aus zwei Unterclustern, bis die jeweiligen Cluster nur noch ein Objekt enthalten bzw. ein Blatt im Baum sind.

Je nach Einsatzgebiet gibt es weitere Unterteilungsmöglichkeiten innerhalb des Clusterings. So kann man zum Beispiel unterscheiden, ob Objekte exklusiv nur einem oder mehreren Clustern zugeordnet werden können. Alternativ kann man auch eine Teilnahmegewichtung von 0 bis 1, wie in der Fuzzy-Logik nutzen. Eine weitere Frage die entsprechend

dem Einsatz des Clusterings beantwortet werden sollte ist, ob die gesamte Menge an Objekten zugeordnet werden muss oder eine partielle Zuteilung ausreichend ist, wenn die Zuordnung zu einem einzigen Cluster z.B. nicht genau definiert werden kann. Diese weiteren Unterteilungen decken noch nicht den kompletten Bereich des Clusterings ab und sollen als kleine Einordnung in das Thema ausreichend sein, um ein grundlegendes Verständnis für die nachfolgend erklärten Methoden zu erlangen.

Das wahrscheinlich bekannteste Verfahren aus dem partitionierendem Bereich ist das k-means Clustering (vgl. Hastie et al. [17][S. 460 13.2.1]), welches auf den Schwerpunkten der jeweiligen Cluster arbeitet. Da es häufig auf Abstandsmaßen wie der Manhattendistanz, dem Jaccard-Maß oder der euklidischen Distanz arbeitet und die Summe der quadrierten Fehler optimiert wird, kann es besonders lokale Minima gut bestimmen. So ist der Einsatz bei gut trennbaren Daten mit gleicher Dichte und ähnlicher Clustergröße optimal (vgl. Steinbach et al. 2013 [15][S. 510 8.2.5]).

Ein weiteres Verfahren ist das partitionierende DBSCAN, welches auf dem zentralen Begriff der Dichte arbeitet. Da die paarweisen Abstände im hochdimensionalen Raum teuer zu berechnen sind und die Aussagekraft der Dichte in hohen Dimensionen schwierig zu interpretieren ist (vgl. Abschnitt 2.1.1), ist der Einsatz in dem hier vorgestellten Umfeld allerdings ungeeignet.

Der folgende Abschnitt beschreibt das sogenannte *agglomerative hierarchische Clustering*, welches später für die einfache Navigation durch die Eingabedaten dienen soll. Da die Vorteile der entstehenden Hierarchie für die Navigation durch den Datensatz genutzt werden sollen, wird auf eine weitere Erläuterung der anderen Verfahren und weiteren Differenzierungsmöglichkeiten verzichtet.

5.2 Agglomeratives hierarchisches Clustering

Innerhalb des hierarchischen Clusterings gibt es zwei Ansatzmöglichkeiten, so kann man den entstehenden Baum top-down oder bottom-up konstruieren. Beim agglomerativem Vorgehen startet man bottom-up bei den Blättern, welche zu Beginn ein Cluster mit genau einem Objekt repräsentieren und fasst diese schrittweise zusammen. So werden in jedem Iterationsschritt jeweils zwei Cluster miteinander verschmolzen, bis nur noch zwei Cluster übrig bleiben und diese im Wurzelknoten zusammengefasst werden. Dieser repräsentiert dann die komplette Eingabemenge. Durch dieses Vorgehen entsteht ein binärer Baum, in dem jeder Knoten ein Cluster definiert, das wiederum aus zwei Unterclustern zusammengesetzt ist. Die Blätter sind Cluster die nur aus genau einem Objekt der Eingabemenge bestehen. Jedes Eingabeobjekt ist demnach exklusiv einem Cluster zugeordnet, welche anschließend verschachtelt werden.

Die Entscheidung, welche Cluster verschmelzt werden, ist grundlegend von zwei Parametern abhängig. Diese sind zum einen das Proximitätsmaß, sowie der gewählte Fusio-

Verfahren	**Berechnung**
Single Link	$D_{Single}(A,B) := \min_{a\in A,b\in B}\{d(a,b)\}$
Complete Link	$D_{Complete}(A,B) := \max_{a\in A,b\in B}\{d(a,b)\}$
Average Link	$D_{Average}(A,B) := \frac{1}{\lvert A\rvert\lvert B\rvert}\sum_{a\in A,b\in B} d(a,b)$

Tabelle 5.1: Verfahren zum Fusionieren von Clustern

nieralgorithmus. Für die Bestimmung, wie Cluster miteinander verbunden sind bzw. welche zusammen verschmolzen werden sollen, kann man beispielsweise auf die Verfahren *Single-*, *Complete-* oder *Average Link* zurückgreifen (vgl. Steinbach et al. 2013 [15][S. 519-524]). Das Single Link Verfahren nutzt dabei den kürzesten Abstand zwischen den Punkten der beiden Cluster, Complete Link den größten Abstand und Average Link den durchschnittlichen Abstand aller Punkte der beiden Cluster. Für zwei Cluster A und B lassen sich die Abstände dieser beiden Cluster durch die in Tabelle 5.1 gegebenen Formeln bestimmen. Jedes Verfahren hat nach Steinbach et al. 2013 [15][S. 519-522] dabei bestimmte Eigenschaften, so ist das Single Link Verfahren anfällig für Ausreißer und Rauschen, aber gut im Erkennen nicht-elliptischer Formen. Das Complete Link Verfahren hingegen bevorzugt globale Formen und kann größere Cluster ungewollter Weise zerteilen, ist dafür aber weniger anfällig für Rauschen und Ausreißer.

Die Wahl des Proximitätsmaßes $d(a,b)$ fällt dabei häufig auf die euklidische Distanz, also die L_2-Norm mit $d(a,b)_{L_2} = ||a-b||_2 = \sqrt{\sum_i (a_i - b_i)^2}$ oder die Manhattan Distanz, sprich die L_1-Norm mit $d(a,b)_{L_1} = ||a-b||_1 = \sum_i |a_i - b_i|$. Diese sind Beispiele für Distanzmaße, also nur anwendbar, falls den Daten eine Metrik zugrunde liegt. Bei nominalen und ordinalen Skalen muss dazu auf Ähnlichkeitsmaße zurückgegriffen werden.

Für die Bestimmung der Ähnlichkeiten bzw. Abstände der Cluster zueinander, verwendet der Algorithmus einen Nachbarschaftsgraphen bzw. Abstandsmatrix, welche in jedem Schritt aktualisiert wird (vgl. Pseudocode 5.1). Falls die Berechnung des Nachbarschaftsgraphen/Abstandsmatrix initial bestimmt werden muss, ist dies der rechenintensivste Schritt. Alternativ können diese auch als zusätzliche Eingabe im Algorithmus aufgeführt werden. Angenommen die Abstandsmatrix ist symmetrisch, so skaliert der Algorithmus quadratisch in der Anzahl der Objekte aus der Eingabemenge, also $\mathcal{O}(n^2)$ für n Eingabeobjekte (vgl. Steinbach et al. 2013 [15][S. 518]). Neben der Menge an Objekten muss man zusätzlich die Anzahl der zu ermittelnden Cluster k angeben. Um im Anschluss k distinkte Clusterlabel zu erhalten, kann man den Baum an der entsprechenden Stelle so schneiden, dass genau k Cluster entstehen. Die Clusterlabel sind dabei oft natürliche Zahlen und ordnen jedem Objekt durch Traversieren des Baumes an den entsprechenden Knoten ein Label zu.

Abbildung 5.1 zeigt ein beispielhaftes Dendogram, also die Hierarchie der Cluster. Die roten Linien sind die entsprechenden Stellen, an denen man den Baum bzw. das Dendogram

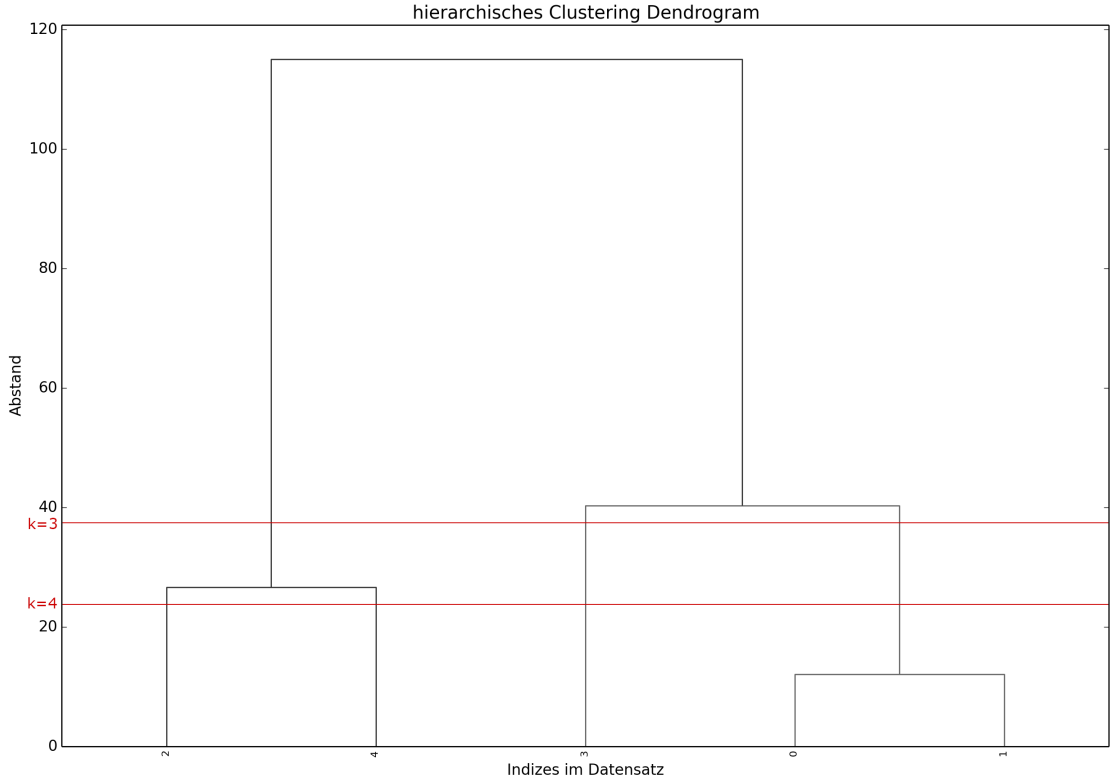


Abbildung 5.1: hierarchisches Clustering Dendogram

schneidet, um $k = 3$ und $k = 4$ Cluster zu erhalten. Ausgehend von den Schnittpunkten zwischen der Linie und dem Dendogram, wird den Objekten des entsprechenden Pfades eines der k Clusterlabel zugewiesen.

Algorithmus 5.1 agglomeratives hierarchisches Clustering Pseudocode

Eingabe: Datensatz $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$, Anzahl an Clustern k, Proximitätsmaß d, Fusionierverfahren D

Ausgabe: Zuordnung der Eingabeobjekte zu Clustern $Y = \{y_1, y_2, \ldots, y_k\}$.

initialisiere jedes Objekt zu eigenem Cluster
berechne Nachbarschafts- bzw. Abstandsmatrix zwischen Clustern
repeat
    verschmelze die zwei nächsten/ähnlichsten Cluster
    aktualisiere Abstand zwischen den Clustern
until nur noch ein Cluster übrig bleibt

Der alternative top-down Ansatz geht zu Beginn von einem großen Cluster aus, welches alle Objekte der Eingabemenge enthält. In jedem Schritt teilt er die Cluster in zwei Untercluster und bekommt daher den Namen divisives hierarchisches Clustering. Dieser

Iterationsschritt wird so lange wiederholt, bis jedes Cluster nur noch aus einem einzelnen Objekt besteht.

Kapitel 6

Fallstudie relNet Projekt

Der in Kapitel 4 beschriebene Ansatz aus der Kombination von Clustering und Visualisierung wird hier aufgegriffen und am Beispiel des relNet Projektes evaluiert. Dazu wurde der zur Verfügung stehende Datensatz (vgl. Abschnitt 4.1) zunächst per hierarchischem Clustering gruppiert und damit eine Hierarchie gebildet. Anschließend werden die durch das von Le & Mikolov 2014 [8] berechneten Document-Embeddings per Hauptkomponentenanalyse auf 30 Dimensionen reduziert. Dieses Vorgehen entspricht den t-SNE Experimenten nach van der Maaten (vgl. [20][S. 2589 4.2]). Auf dem so reduzierten Datensatz wird eine t-SNE Berechnung für zwei Dimensionen bestimmt, welche dann in einer Webapplikation visualisiert wird. Anhand der binären Hierarchie ist es möglich, die jeweils beiden Untercluster der aktuellen Ansicht weiter zu untersuchen. So hat man die Möglichkeit, anhand der gefundenen Cluster durch den Datensatz zu zoomen, indem man bei jeder Ansicht zwischen zwei Bereichen wählen kann, welche dann erneut per t-SNE visualisiert werden. Die beiden Cluster aus denen sich die aktuelle Ansicht zusammensetzt sind jeweils unterschiedlich gefärbt. Dadurch erhofft man sich bestimmte Bereiche immer weiter aufschlüsseln zu können, sodass man ein tiefergehendes Verständnis für die Struktur bzw. die Zusammensetzung der Daten erlangt. Im Falle des relNet Projektes möchte man dabei Themengebiete innerhalb der Diskussionsrunden finden und die gefundenen Themen soweit aufspalten, bis erkenntlich wird, aus welchen Bereichen sich das Thema zusammensetzt. Ob der Ansatz diese Anforderungen erfüllt, gilt es in diesem Kapitel zu untersuchen. Dafür wird anhand konkreter Beispiele gezeigt, ob man die jeweiligen Unterthemen durch erneute Visualisierung per t-SNE eines bestimmten Bereiches erkennen kann.

In Abbildung 6.1 ist der komplette Datensatz, also die anfängliche Ansicht, visualisiert und die beiden Untercluster aus denen er zusammengesetzt ist, farblich markiert. Ausgehend von dieser Darstellung ist es nun möglich „entlang“ der Clusterhierarchie durch den Datensatz zu navigieren und so die in Tabelle 4.1 beschriebenen Bereiche genauer zu untersuchen. Darüber hinaus sollen auch die anderen Gebiete, in welchen man oberflächlich keinen genauen Zusammenhang finden konnte, besser eingeordnet bzw. einem Themenge-

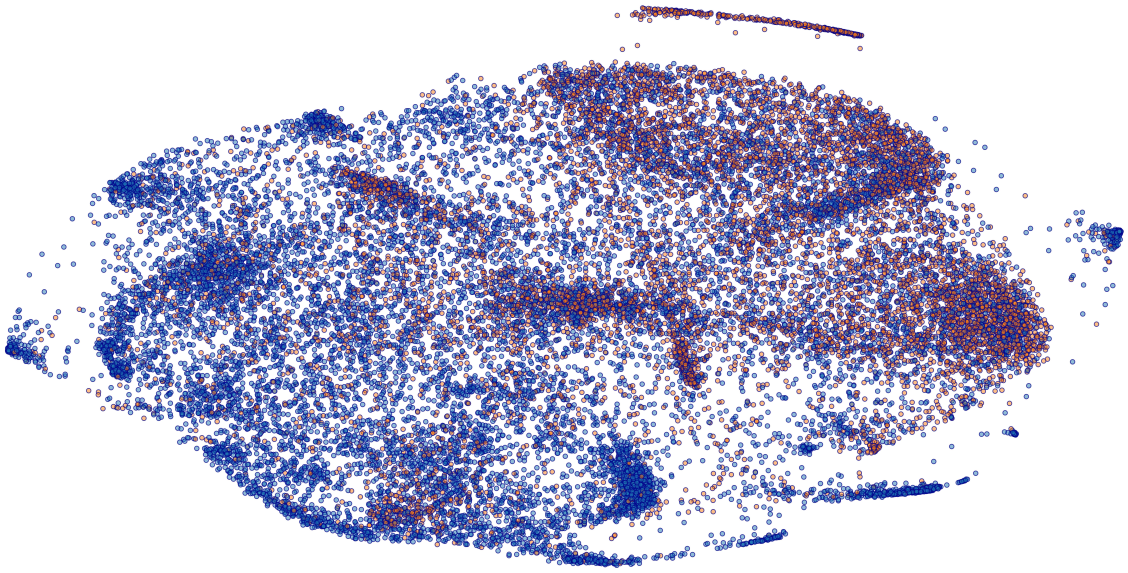


Abbildung 6.1: t-SNE Visualisierung relNet Datensatz mit Färbung der Untercluster

biet zugeschrieben werden können. Um eine Intuition über die Aufspaltung des Datensatzes zu bekommen, zeigt Abbildung 6.2 die beiden ersten Untercluster der Ansicht auf den gesamten Datensatz. Die entsprechenden Themengebiete für Abbildung 6.2a sind in Tabelle 6.1 aufgelistet. Tabelle 6.2 enthält die Übersicht zu Abbildung 6.2b.

Der Bereich Politik ist ein gutes Beispiel für die Aufspaltung der Themen. Zu Beginn konnte man nur erkennen, dass es einen groben Bereich gibt, in dem es um politische Themen geht. Im ersten Zoom-Schritt können bereits bessere Aufteilungen nach innen- und außenpolitischen Bereichen getroffen werden. Je tiefer man in der Hierarchieebene kommt, desto mehr definite Themen gibt es. Diskussionen über Zustände, Maßnahmen und Meinungen der Situation in Israel und Schuldenprobleme im Zusammenhang mit Griechenland konnten dabei gefunden werden, um nur zwei Beispiele zu nennen. Eine weitere interessante Erkenntnis ist die Aufspaltung der Glückwünsche, hier kann im ersten Zoom-Schritt in Geburtstagswünsche und Glückwünsche zur Geburt differenziert werrden.

In Abbildung 6.2b erkennt man die Abspaltung des Bereiches A (vgl. Tabelle 6.2), die dort angesiedelten Diskussionsrunden beinhalten alle eine Kombination aus einem Psalm bzw. Zitaten o.ä., sowie einem passenden Rezept. Das hier verwendete Vokabular bzw. der Wortschatz stimmt mit keinem der anderen Diskussionsrunden überein und ist somit als abgetrenntes Cluster dargestellt. Die zusätzliche Besonderheit bei diesem Cluster ist der Autor. Alle Beiträge stammen vom Nutzer Murphyline und wurden im Gruppenforum „Katholischer Kreis“ veröffentlicht. Ein Beispiel ist der Beitrag mit dem Titel „6. Dezember Nikolaus, Stutenkerle und Spekulatius“ in dem ein Rezept für Spekulatius und der historische Hintergrund des Nikolausbrauches in der Kirche erläutert wird. Diese Abtrennung konnte in der vorherigen Ansicht auf den gesamten Datensatz nicht ausfindig gemacht

Bereich	Themengebiet/Beschreibung
A	verschobene Beiträge
B	Ernährung und Abnehmen (Vitamin- und Mineralienmangel, Rezepte)
C	Vom Forum-Team angeleitete Diskussionsthemen zu aktuellen Artikeln auf jesus.de
D	Psalme und Diskussionen über Bedeutung von einzelnen Versen („Jesu Ruhe + Kraft - Insel/Oase“)
E	Krankheiten/Symptome und Diskussion zu Behandlungsmöglichkeiten (Vitamine kommen besonders häufig vor)
F	Schwangerschaft/Verhütung (Frage ob schwanger und wie man damit umgehen soll)
G	Austausch von Erfahrungen (z.B. welcher Kinderwagen am besten ist)
H	Beziehungen, Sex, Beziehungsprobleme, Ehe, Trennung, Partnerschaft / direkt daneben SVV = Selbstverletzendes Verhalten, Trigger, Depression (Probleme werden überwiegend anonym veröffentlicht)
I	Kinder, Wut, Streit, schlechtes Verhalten von Kindern, Umgang mit Kindern, Schulleben
J	Fragen zur Arbeitswelt/Arbeitsmarkt, Arbeitsrecht, Lohn, Urlaubsanspruch, Rentenversicherung, Verträge, Gehaltsverhandlungen, Ausbildung
K	Erbe, Mietrecht, Steuerfragen (Grundsteuer, Einkommenssteuer), Häuserkauf
L	christlicher Podcast, Radio, GEMA, lizenzfreie Musik
M	Kino- und Filmtrailer, Filmmusik
N	politische Diskussionen über Frieden in Israel (Palästina/Jerusalem/Gazastreifen), häufig Psalme in der Argumentation genutzt
O	politische Diskussion zu Krieg und Kriesen (Syrien, Ukraine, IS-Terror)
P	Atomkraftwerke, Kernenergie, Fukushima, physikalische Fragestellungen (Gravitationskräfte, Raumfahrt)
Q	Religion vs. Wissenschaft (Evolutionstheorie, Glaube und Evolution)
R	innenpolitische Diskussion, Pegida, Asyl/Flüchtlinge, Islamismus
S	wirtschaftliche und finanzpolitische Diskussionen, Schulden, Mindestlohn
T	besonders katholische Themen (Ökumene, Homosexualität)
U	Impfung, Pille danach, Legalisierung von Cannabis, Ebola
V	Diskussionen zu Feiertagen (Buß- und Bettag, Islam-Feiertage)
W	Zitate und Bibelstellen (häufig Interpretationsfragen, noch kein genaues Themengebiet ausmachbar)

Tabelle 6.1: Bereiche und Themengebiete rote Markierungen der Gesamtansicht

werden und stellt somit einen Informationsgewinn für den Analysten des Datensatzes dar. Darüber hinaus liefert dieses Beispiel mögliche Erkenntnisse über relevante Nutzer in einem bestimmten Gebiet. Da es in diesem Fall möglich war eine wichtige Person in einem bestimmten Themenbereich zu entdecken, ist dies eventuell auch in weiteren Bereichen der Fall. So konnte im Bezug auf Buchempfehlungen der Nutzer heaven4u als interessant bzw. einflussreich entdeckt werden, da er viele Diskussionen selber begonnen hat und zudem sehr aktiv im Bereich Bücher ist. Einflussreiche Personen sind dabei solche, die sehr aktiv in einem Themengebiet engagiert sind, also viele Diskussionen selber starten und häufig andere Nutzer kommentieren.

Die oben genannten Beispiele und Auflistungen der groben Themengebiete der ersten Ansicht auf die Untercluster des gesamten Datensatzes, zeigen bereits, wie vielfältig und aufschlussreich diese sind. Sehr spezifische Bereiche, wie zum Beispiel die Diskussionen zu Kinotrailern und Filmmusik, welche sich aus weniger Diskussionsrunden zusammensetzen, können bereits klar identifiziert werden. Als Kontrast dazu gibt es im Bereich Bibelstellen und Politik eine Fülle an Diskussionen, welche erst weiter aufgespalten werden müssen, um ein Verständnis für die Struktur zu erlangen.

Darüber hinaus soll das folgende Beispiel die Aufschlüsselung eines bestimmten Themengebiets erläutern, welches durch das Hereinzoomen entdeckt wurde. In der Ansicht auf den kompletten Datensatz gibt es einen Bereich in dem es oberflächlich um Beziehungen geht (vgl. Markierung L in Abbildung 4.2). Durch Auswahl von blau -> blau -> blau -> blau -> rot kann man das Themengebiet soweit aufschlüsseln, dass man den Bereich Beziehungen genauer versteht. So gibt es einige Diskussionen zum Thema Partnerschaft zwischen evangelischen und katholischen Anhängern, sowie zwischen Christen und anderen Religionen. Dabei werden besonders Befürchtungen und Ängste besprochen, die Betroffene haben, wenn sie sich auf Partnerschaften mit Personen anderer Konfession, Religion oder Atheisten einlassen. Häufig kommt die Frage nach Erfahrungen und Einschätzungen anderer Nutzer im Forum auf. Zusammen mit den gerade beschriebenen Diskussionsrunden gibt es auch einige zum Thema Partnersuche im Internet.

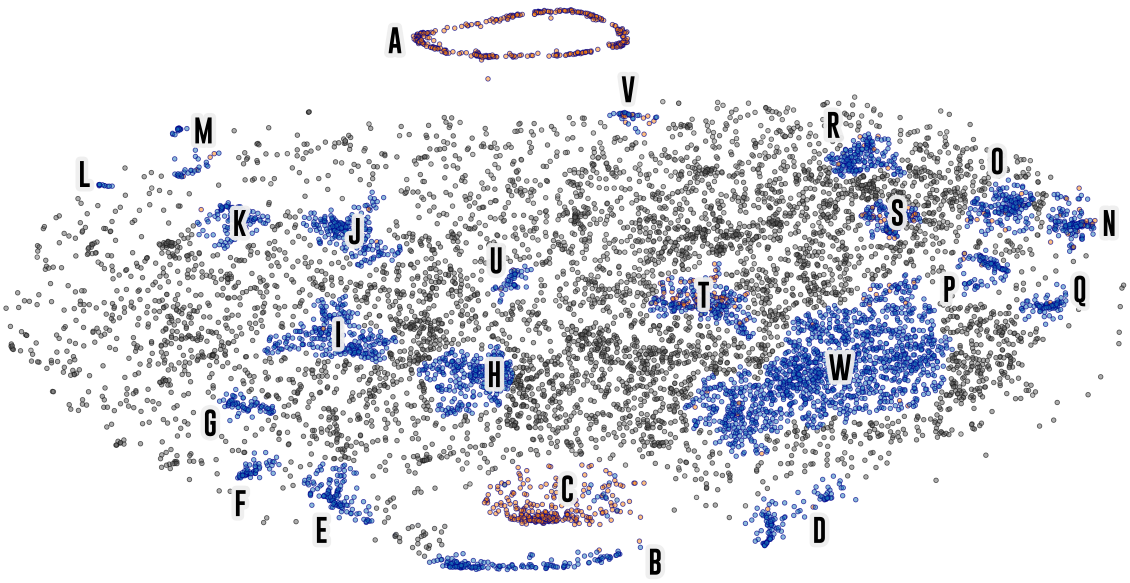
In der gleichen Ansicht finden sich weitere klar abgetrennte Themengebiete, die eine Spezialisierung vorher gefundener Themen sind. So gibt es einen Bereich in dem sich vorwiegend über Bücher und Buchempfehlungen unterhalten wird. Hier sind Titel wie „Was lest ihr im Januar/Februar/März/..“ vorzufinden. Weitere kleine Abspaltungen enthalten Diskussionen über Kinderbücher. In diesem Bereich wird besonders die pädagogische Seite und Vermittlung christlicher Werte des Buchinhaltes für Kinder diskutiert. Eine weitere Gruppierung in dieser Ansicht sind Beiträge zum Thema Kindernamen. So findet man Auflistungen für Namensvorschläge, Diskussionen zur historischen Bedeutung von Namen im Zusammenhang mit der Bibel und Fragen nach den unterschiedlichen Schreibweisen und damit zusammenhängenden Bedeutungen. Folgt man der Unterteilung weiter, findet man

Bereich	Themengebiet/Beschreibung
A	Psalm Beiträge kombiniert mit Rezepten
B	Kochen und Rezepte
C	Glückwünsche zur Geburt
D	Schwangerschaft, Elektrokrampftherapie, „KugelBauchPlauderThread“
E	erstes Lebensjahr, Probleme und Fragen zu Kindern in den ersten Jahren (Erziehung)
F	Kinderwunsch, Fragen zur Schwangerschaft (ab wann ist man schwanger?)
G	Glückwünsche zum Geburtstag
H	Technische Fragen zu e-Books, Fragen zu Software (Browser, Mailprogramme, Updates, Betriebssysteme), Fotografie (Equipment und Kameraeinstellungen)
I	Diskussionsrunden/Smalltalk (sogenannte Cafés und „Single Plauder Faden“, kurz SPF, „Mamathread“, häufig geht es um Wichteln)
J	Empfehlungen und Rezensionen von Büchern
K	Austausch über das Fernsehprogramm (TV-Serien, Tatort)
L	Urlaub, Reiseziele, Preisvergleiche und Empfehlungen für Urlaubsziele
M	Smalltalk zum Wetter (Sonne, Regen, Gewitter, Temperaturangaben, etc. häufig zu finden)
N	Suche nach Liedern (oft im christlichen Zusammenhang, z.B. für Jugendgruppen in der Kirche), YouTube Verlinkungen, Meinungen zu bestimmten Musikstücken
O	Haustiere, Alltagsfragen zu Katzen und Hunden
P	Gartenbau, Erfahrungen und Fragen zu Pflanzen (häufig Rosen und Avocados)
Q	Nähen, Stricken (Anleitungen und Fragen), Basteln, Do-It-Yourself Ideen (oft im Zusammenhang mit Kindergeburtstagen)
R	Kosmetik (Sonnenschutz, Deodorant, Lippenstift), Haushalt (Waschmittel, Pflege von Kleidung), Transport von Babys (Kinderwagen, Tragetuch/Tragehilfe)
S	Technische Fragen zum Forum (Gruppenbildung und -auflösung), Forum-Interne Diskussionen (Nutzer kündigen Abwesenheit an, Grundsatzdebatten über bestimmte Beiträge im Forum)
T	Technische Fragen zum Forum (Anmeldeverfahren, Probleme bei Funktionen wie Zitieren, Ansicht anpassen, Versand von Nachrichten, etc.)
U	politischre Diskussionen zu Parteien
V	Politik im Zusammenhang mit Flüchtlingen, „Flüchtlingskrise“
W	Interpretation/Auslegung bestimmter Bibelstellen, Fragen zur Bedeutung und Kontextfragen
X	Trauer um Verstorbene
Y	seelische Probleme, Missbrauch, Therapie, Bewältigung von Trauer
Z	Diskussionen zu Sportarten (vor allem Fußball), Sportereignisse und Sportnachrichten

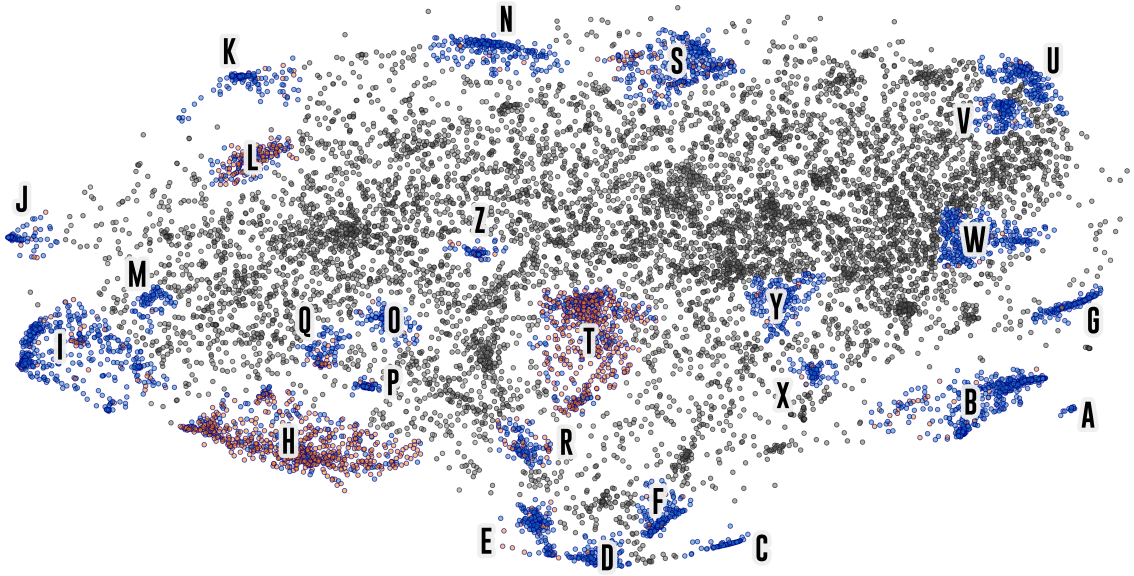
Tabelle 6.2: Bereiche und Themengebiete blaue Markierungen der Gesamtansicht

heraus, dass eine Gruppe der Namenssuchenden besonders an biblischen Namen interessiert ist.

Diese Zusammenhänge sind in der fünften Hierarchieebene des Clusterings gefunden worden (vgl. Abbildung 6.3) und zeigen bereits, wie detailiert man die Informationen aufspalten kann. Aufgrund der Konstruktion kann man dieses Vorgehen bis zu dem Punkt weiterführen, an dem ein Cluster nur noch aus einem Objekt besteht.



(a) rotes Untercluster



(b) blauses Untercluster

Abbildung 6.2: Erste Unterteilung des gesamten Datensatzes aus Abbildung 6.1

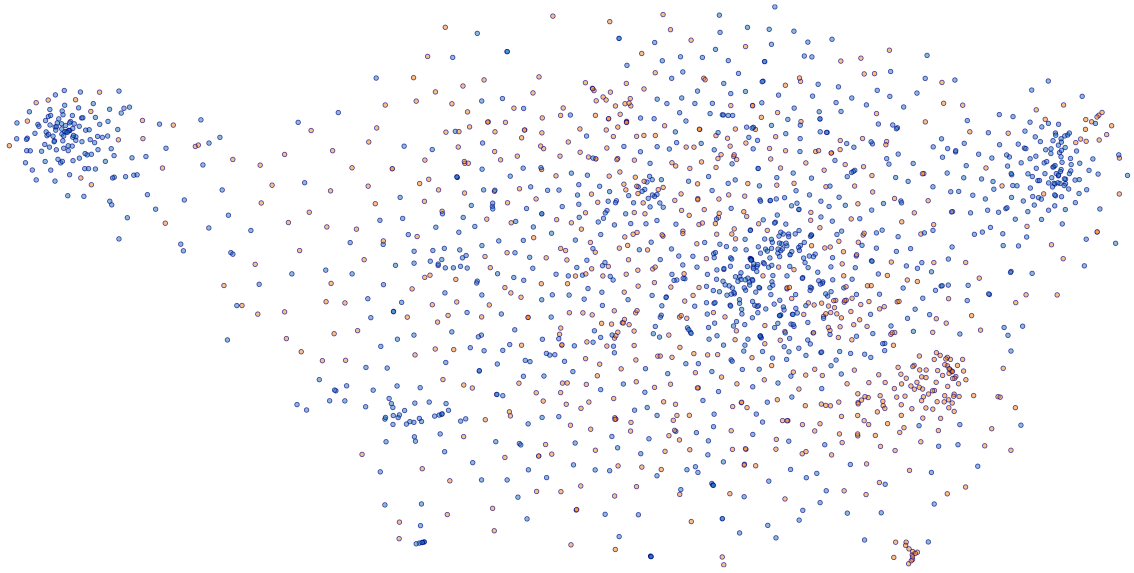


Abbildung 6.3: Ansicht nach Auswahl von: blau, blau, blau, blau, rot

Kapitel 7

Implementierung

Wie in Kapitel 4 schon angedeutet, wurde das vorgeschlagene Vorgehen im Rahmen dieser Bachelorarbeit beispielhaft in Python implementiert. Die letzte Version des entstandenen Tools ist unter `https://gitlab.com/phillip1221/interactive-visualizer/` verfügbar. Das Werkzeug zur Visualisierung der Document-Embeddings wurde auf die Bedürfnisse und Gegebenheiten im relNet Projekt zugeschnitten, ist aber universell einsetzbar und kann durch einige kleine Änderungen beim Auslesen des Datensatzes sofort genutzt werden. Die beiden folgenden Abschnitte erläutern die Details des Back- und Frontends. Im letzten Abschnitt gibt es einen Ausblick auf mögliche Anpassungen und Weiterentwicklungen des Tools.

7.1 Backend

Für die maschinellen Lernverfahren wurde die scikit-learn[1] Bibliothek (vgl. Pedregosa et al. 2011 [11]) in Python genutzt und durch den leichtgewichtigen Webserver Flask[2] können die darüber berechneten Darstellungen über eine REST-API per HTTP-Anfragen abgerufen werden. Für die Hierarchie zum Navigieren durch den Datensatz wurde das in scikit-learn implementierte agglomerative Clustering genutzt. Das Proximitätsmaß ist dabei die Kosinus-Distanz und die Fusionierung von Clustern beruht auf dem Complete Link Verfahren. Die Kosinus-Distanz ergibt sich dabei aus eins minus der Kosinus-Ähnlichkeit

$$d(x, y) = 1 - \frac{x^T \cdot y}{||x|| ||y||} = 1 - \frac{\sum_{i=1}^{d} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \cdot \sqrt{\sum_{i=1}^{d} y_i^2}}.$$

Um zu Beginn bereits eine intuitiv gute Visualisierung, sprich eine bei der man viele Abspaltungen erkennen kann, zu erhalten, wurde das t-SNE Verfahren zehn mal auf dem kompletten Datensatz durchgeführt und die Lösung mit der niedrigsten Kullback-Leibler Divergenz ausgewählt. Die vorherige Reduktion per Hauptkomponentenanalyse auf

[1] `http://scikit-learn.org/stable/`
[2] `http://flask.pocoo.org`

30 Merkmale wurde genau wie die t-SNE Berechnung auf zwei Dimensionen mit Hilfe der Implementierungen in der scikit-learn Bibliothek ermittelt. Dabei wird vor der Berechnung sichergestellt, dass der Datensatz skaliert ist, also bei allen Merkmalen das gleiche Maß genutzt wird. Die berechneten Document-Embeddings wurden auf der Kosinus-Ähnlichkeit berechnet, daher wird die Kosinus-Distanz auch als Metrik für die t-SNE Berechnung genutzt. Die weiteren Parameter für die t-SNE Berechnung mit dem Barnes-Hut Verfahren und der Metric Tree Approximation wurden wie folgt gewählt. Die Perplexität wurde mit $Perp = 30$ festgelegt, die Anzahl der Iterationen liegt bei $T = 1000$, Lernrate $\eta = 1000$, Schwellwert für die Barnes-Hut Berechnung $\theta = 0.5$. Die Optimierung bricht nach $i = 30$ Iterationen ohne Forschritt ab, sollte sich die Position des aktuellen Graphpunktes nicht mehr ändern.

Für alle Ansichten unterhalb des Wurzelknotens in der Hierarchie werden die Graphpunkte mit ihrer vorherigen Position initialisiert. Dadurch soll sichergestellt werden, dass die Themengebiete in etwa im ähnlichen Radius ihrer vorherigen Darstellung wiederzufinden sind.

7.2 Frontend

Um die ermittelten Graphpunkte zu visualisieren, wurde mit Hilfe der D3[3] JavaScript Bibliothek ein Streudiagramm aus den Daten konstruiert. Damit die Informationen zu den Koordinaten und Clusterzugehörigkeiten, sowie den Inhalten der Diskussionsrunden und ihren nächsten Nachbarn über die REST-API abgerufen werden können, nutzt die Webapplikation AJAX Anfragen zum Abrufen dieser Daten.

Beim Abrufen der Koordinaten liefert das Backend das entsprechende Array, bestehend aus Arrays mit den beiden Kooridnaten sowie dem Clusterlabel. Sollte die abzurufende Ansicht bereits berechnet worden sein, wird auf die abgespeicherten Informationen zurückgegriffen und andernfalls die t-SNE Berechnung und das Auslesen der Clusterlabels für den entsprechenden Knoten in der Hierarchie angestoßen und die Ergebnisse gesichert. Dieser Schritt ist auf Grund der nicht vorhandenen Konvexität der t-SNE Kostenfunktion notwendig, da ihre Berechnung bei jedem Durchlauf andere lokale Minima optimiert. Somit liefern t-SNE Berechnungen auf dem selben Datensatz unterschiedliche Lösungen. Damit ist es möglich die Ansicht, auch beim zurücknavigieren in eine verallgemeinernde Darstellung, konsistent zu halten.

Abbildung 7.1 zeigt die Übersicht der Webapplikation zum Navigieren und dem Abrufen der Informationen. Durch das Auswählen eines Punktes im Streudiagramm werden die berechneten Koordinaten, sowie das Clusterlabel ausgelesen und im Bereich Information angezeigt. Sobald ein Punkt ausgewählt ist, kann man über show post und compute nearest neighbors den Inhalt und die drei nächsten Nachbarn auslesen lassen. Der Inhalt ist dabei

[3] `https://d3js.org`

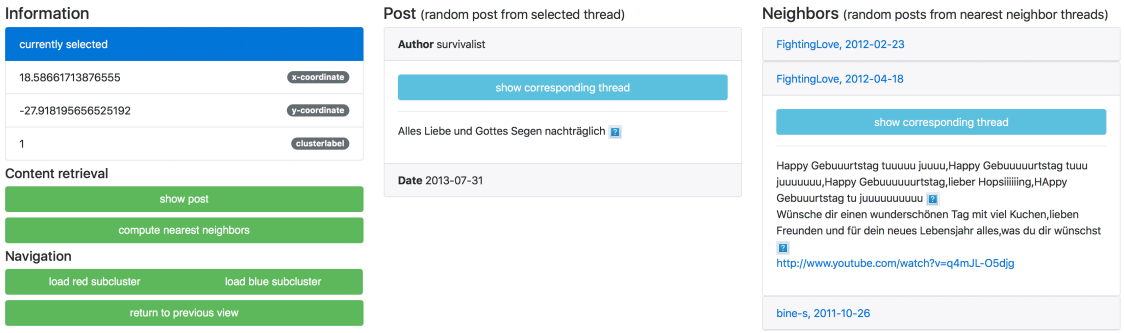


Abbildung 7.1: Interface der Webapplikation

ein beliebiger Beitrag aus der Diskussionsrunde, welcher zuerst gecrawlt wurde. Aus diesem Grund lässt sich die eigentliche Diskussion über den jeweiligen Button show corresponding thread aufrufen, dieser lädt die Diskussionsrunde in einem neuen Fenster. Den Inhalt der nächsten Nachbarn kann man durch das Klicken auf den Namen des Autors und des Datums der Veröffentlichung ausklappen lassen, auch hier lässt sich die komplette Diskussionsrunde per Klick auf den Button abrufen.

Damit die Oberfläche auf allen Geräten unterschiedlicher Größe benutzerfreundlich zu bedienen ist, skaliert das Streudiagramm anhand der Fensterbreite. Durch das HTML, CSS und JavaScript Framework Bootstrap[4] passen sich auch die anderen Elemente responsiv dem Gerät bzw. der Fenstergröße an.

7.3 Ausblick

In den folgenden Absätzen finden sich einige Verbesserungsvorschläge, die in einer weiteren Version umgesetzt werden könnten. Da der bestehende Quellcode frei verfügbar ist und auf Grund der modularen Umsetzung, können weitere Funktionen einfach ergänzt werden.

Dabei wäre es interessant die t-SNE Berechnungen für eine Ansicht jeweils zehn mal zu bestimmen und daraus wieder die Beste, sprich mit der niedrigsten KL-Divergenz, zu wählen. Da diese Berechnung unter Umständen sehr lange dauert, wäre eine parallele Ausführung wünschenswert. Zusätzlich könnte man eine Vorberechnung der Ansichten bis zu einem bestimmten Schwellwert an Eingabeobjekten einbauen. Dadurch würde sichergestellt, dass nutzerunfreundliche Wartezeiten vermeiden werden.

Darüber hinaus könnten weitere Funktionen, wie eine Suche zur Rückverfolgbarkeit von Diskussionsrunden zu ihren entsprechenden Graphpunkten, implementiert werden. Durch diese Funktion wäre es möglich, den Zusammenhang einzelner gezielter Beobachtungen, in diesem Fall Diskussionsrunden, zu bestimmten Bereichen zu ermitteln. So wäre es im relNet Projekt besipielsweise interessant, ob eine kritische Meinung zu einem bestimmten Thema Anklang in anderen Themenkreisen finden.

[4] `https://v4-alpha.getbootstrap.com`

Sollte der Datensatz zur Analyse mit dem hier vorgestellten Ansatz die Größe des relNet Datensatzes weit überschreiten, wäre es denkbar einen Schwellwert festzulegen, bis zu welchem man nur eine Stichprobe der Eingabeobjekte berechnet. So könnten t-SNE Visualisierungen für Eingabemengen mit z.B. mehr als 20.000 Objekten durch eine Stichprobe dieser Objekte repräsentiert werden. Erst wenn die Anzahl der zu berechnenden Punkte unterhalb dieses Schwellwertes liegt, wird eine Berechnung für alle Objekte ermittelt. In einem solchen Ansatz müsste vorher definiert werden, wie man Stichproben effizient ziehen kann, sodass die berechnete Visualisierung eine möglichst gute Repräsentation des gesamten Datensatzes ist.

Kapitel 8

Zusammenfassung

Ziel dieser Arbeit war es ein geeignetes Vorgehen zu finden, welches dabei hilft, große Dokumenten-Kollektionen im hochdimensionalen Raum grafisch analysierbar zu machen. Dazu wurde nach Definition der Problemstellung und Motivation anhand des relNet Projektes das Themengebiet der Dimensionsreduktion vorgestellt. Mit dem t-Distributed Stochastic Neighbor Embedding wurde ein nicht-lineares Verfahren beschrieben, das sich auf den Erhalt lokaler Nachbarschaftsstrukturen konzentriert und speziell für die Reduktion in den zwei- bzw. dreidimensionalen Raum entwickelt wurde. Mit Hilfe von auf Baumstrukturen basierenden Approximationen ist das Verfahren auch auf großen Datenmengen effizient anwendbar. In Kapitel 4 konnte die reine Visualisierung der Diskussionsrunden aus dem relNet Projekt nicht als eigenständige Lösung überzeugen. Daraufhin wurde das Vorgehen aus Kombination von hierarchischem Clustering zur Navigation durch die anfängliche Visualisierung und erneuter t-SNE Berechnung des ausgewählten Knotens in der Hierarchie beschrieben. Anhand der Dokumenten-Kollektion des relNet Projektes zeigt Kapitel 6 die Anwendung und Erkenntnisse aus dem vorher beschriebenen Ansatz. Die Details zur Implementierung wurden in Kapitel 7 aufgegriffen und kurz erläutert.

Der folgende Abschnitt zieht Bilanz aus den Ergebnissen und der Anwendung. In Abschnitt 8.2 wird ein Ausblick auf relevante Forschungen und Verbesserungen des Ansatzes gegeben und schließt diese Arbeit somit ab.

8.1 Fazit

Kapitel 2 verdeutlicht die Relevanz von nicht-linearen Verfahren zur Dimensionsreduktion, dabei bringt die Wahl von t-SNE einige Vorteile für die Analyse der Struktur mit sich. Im Vergleich zu anderen Visualisierungsverfahren entdeckt t-SNE Strukturen auf unterschiedlichen Skalen bzw. Dimensionen. So bildet t-SNE mehrere niedrigdimensionalere Mannigfaltigkeiten in einer einzigen Grafik ab und ist somit von Vorteil, wenn der Datensatz auf unterschiedlichen niedrigdimensionaleren Mannigfaltigkeiten beruht. Dar-

über hinaus hat es nicht die Tendenz, Punkte im Zentrum der Darstellung zu platzieren und trägt somit zur übersichtlichen visuellen Analyse bei. Auf der anderen Seite ist das stochastische Verfahren mit nicht konvexer Kostenfunktion in dem Sinne unstabil, da es bei jedem Durchlauf andere Darstellungen liefert. Mit angemessenem Umgang, sprich der Verwendung der Visualisierung mit niedrigster Kullback-Leibler Divergenz aus mehreren Durchläufen und dem Abspeichern der Ergebnisse, ist dies aber kein Problem für den produktiven Einsatz. Zusätzlich sollte man im Hinterkopf behalten, dass das Verfahren lokale Nachbarschaftsstrukturen optimiert und die Anordnung keine Aussage über die globale Struktur macht. Globale Zusammenhänge aus der ursprünglichen Dokumenten-Kollektion werden nicht erhalten und damit sind optische Cluster die weit auseinander liegen nicht unbedingt grundsätzlich verschieden. Diese Information ist besonders bei der Interpretation der Darstellung wichtig, um keine falschen Schlüsse zu ziehen.

Mit Hilfe des hierarchischen Clusterings wird dem Nutzer eine Navigationsmöglichkeit durch den Datensatz geboten und konnte dem Problem der großen Mengen an Daten entgegengewirkt werden. Da in jedem Navigationsschritt, also der Auswahl eines der beiden Untercluster der aktuellen Ansicht, ein Großteil der Objekte ausgeschlossen wird, resultiert eine immer spezifischere Menge an zu untersuchenden Objekten. Mit dem Complete Link Verfahren wurde eine Methode zum Verschmelzen der Cluster gewählt, die gegenüber Rauschen und Ausreißern robust ist und so für eine ausgeglichene Navigation sorgt.

Das t-SNE ist ein geeignetes Mittel für den Umgang mit der Hochdimensionalität von Daten, dabei ist das Verfahren zur Visualisierung immer in Abhängigkeit des Zieles der Darstellung zu wählen. Durch die Navigation entlang der Hierarchie kann man darüber hinaus der Menge an Daten Herr werden. Aus der beschriebenen Kombination beider Techniken, lässt sich das in der Einleitung genannte Problem lösen und Daten effektiv visuell analysierbar machen. Mit den im Kapitel 6 vorgestellten Ergebnissen konnte der Ansatz als nützlich verifiziert werden. Die dort genannten Beispiele belegen die Erfüllung der anfangs aufgezählten Anforderungen an die interaktive Visualisierung mit dem Ziel der Strukturanalyse. Es konnten Zusammensetzungen komplexer Themengebiete wie die der Politik und der Technik aufgeschlüsselt und in einzelne aussagekräftige Unterthemen geteilt werden. Besonders die Zusammenhänge innerhalb großer Themengebiete wurden dabei deutlich. Im Zusammenspiel der beiden Verfahren erhält man so die Möglichkeit global immer spezifischer zu werden und in Hinsicht auf die gerade dargestellte Ansicht die lokalen Nachbarschaften bzw. Ähnlichkeiten der Dokumente zu erkunden. Dieses Verhalten ist im Bezug auf die Analyse der Struktur interessant, da man sich wie bei alleinigem Einsatz von Methoden zur Dimensionsreduktion nur auf die Optimierung einer Eigenschaft festlegen kann.

Da man bei der Visualisierung keine qualitativen Maßstäbe besitzt, nach denen man eine erfolgreiche Darstellung beurteilen könnte, muss auf Alternativen zurückgegriffen werden. Die in der Einleitung beschriebenen Anforderungen wurden durch den Ansatz erfüllt,

da man sowohl eine erste Intuition über die Verteilung, in diesem Falle Ähnlichkeiten der Eingabeobjekte, sowie eine schrittweise Aufspaltung der einzelnen Themengebiete bekommt. Aufgrund der großen Mengen an Daten ist ein besonderes Augenmerk auf die Skalierbarkeit der Verfahren zu werfen, mögliche Laufzeitprobleme werden im Ausblick 8.2 aufgegriffen. In den Experimenten der Fallstudie in Kapitel 6 konnten dabei aber keine Probleme festgestellt werden (vgl. dazu das MDS Experiment in Anhang A.1).

8.2 Ausblick

Im Verlauf der Bearbeitung des Themas sind einige wünschenswerte Verbesserungen und interessante Fragestellungen aufgekommen, welche in zukünftigen Weiterentwicklungen und Forschungsarbeiten untersucht werden könnten.

Auf Seiten der Algorithmen wäre es gerade für das t-Distributed Stochastic Neighbor Embedding interessant, neben der Barnes-Hut- und Metric Tree Approximation, auch den Gradienten direkt zu approximieren. Dadurch könnten sich weitere Vorteile für die Laufzeit ergeben. Es gilt dabei zu prüfen, ob und inwieweit eine solche Anpassung weiterhin zuverlässige Ergebnisse im Bezug auf die Darstellung von Nachbarschaftsbeziehungen im hochdimensionalen Raum liefert.

Da zu Beginn das agglomerative hierarchische Clustering durchgeführt wird und dieses einen Nachbarschaftsgraphen zur Bestimmung der Proximität generiert, wäre es sinvoll, diesen in einer optimierten Version zu sichern. Im Zuge der Dimensionsreduktion könnte dieser vom t-Distributed Stochastic Neighbor Embedding wiederverwendet werden. In Anbetracht großer Datenmengen macht es daher also Sinn, die quadratisch skalierende Berechnung des Nachbarschaftsgraphen wiederzuverwenden (vgl. Abschnitt 5.2).

Der Einfluss von Datensätzen mit starkem Rauschen auf das Entdecken von Mannigfaltigkeiten ist ein weiterer Punkt, der im Bezug auf die Ergebnisse untersucht werden könnte. Dabei können Ausreißer und Rauschen dazu führen, dass keine klar trennbaren Cluster mehr gefunden werden können, da diese die natürlichen Gruppen verbinden. In einigen Ansichten gab es starke Unterscheidungen der vom hierarchischen Clustering gefundenen Abgrenzungen und der visuallen Anordnung durch t-SNE. Diese sind vor allem der binären Struktur zuzuschreiben, die in einigen Fällen zu grob ist, um die Abgrenzungen der Objekte optimal zu beschreiben. So waren Punkte im Streudiagramm durch t-SNE zusammen angeordnet, welche zu unterschiedlichen Unterclustern der Hierarchie gehören. Diese Beobachtungen konnten vor allem bei großer Anzahl an Punkten festgestellt werden (vgl. Abbildung 6.1). Je weniger Punkte visualisiert werden, desto seltener gibt es diese Art an Diskrepanz zwischen den Clusterzugehörigkeiten. Gerade für diesen Fall wäre es interessant, die Unterschiede eines bestimmten Bereiches, welche vom Nutzer selber gewählt werden können, erneut per t-SNE zu visualisieren. Sprich der Nutzer wird mit einem Auswahlwerkzeug dazu befähigt, selbst einen Bereich zur erneuten Visualisierung

auszuwählen. So könnte man den Nutzern weiterhin eine Analysemöglichkeit dieser Gruppierungen bieten, auch wenn die optisch zusammen angeordneten Punkte unterschiedlichen Unterclustern angehören.

Je weiter man in die Ansicht hereinzoomt und je weniger Punkte dargestellt werden, desto mehr gleicht die Anordnung einer Gleichverteilung. Eine solche Verteilung der Punkte widerspricht der Anforderung klar abgetrennte Cluster zu finden und erschwert die grafische Analyse der Ansicht, da der Eindruck entsteht es gebe keine starken Zusammenhänge zwischen den Objekten. Dabei liegt die Vermutung nahe, dass durch die Übernahme der Positionen aus der vorherigen Ansicht die Punkte soweit auseinander liegen, dass die geringe Anzahl an Punkten in den betroffenen Ansichten nicht ausreichend ist, um wohlgeformte Cluster zu bilden. (Die vorhandenen Punkte haben zu wenig Einfluss, die Punkte so zu bewegen, dass klare CLuster entstehen.) Um dieses Problem zu vermeiden, könnte man anstatt der letzten Position zur Initialisierung der zweidimensionalen Darstellung per t-SNE die ersten beiden Hauptkomponenten einer PCA nutzen, damit die Punkte besser voneinander getrennt sind. Somit könnten auch bei wenig Punkten optisch unterscheidbare Cluster entstehen. Diese wären mit der normalen Variante nicht sichtbar, weil zu wenig Einfluss anderer Punkte zum Bestimmen geeigneter Positionen vorhanden sind. Möglicher Nachteil ist, dass die Cluster wahrscheinlich in anderen Positionen im Diagramm zu finden sind, als sie vorher waren. Dies könnte eventuell die Orientierung erschweren.

Anhang A

Weitere Informationen

In diesem Anhang finden sich im Abschnitt A.1 zusätzliche Visualisierungen der Kollektion an Dokumenten des relNet Projektes und ergänzende Anmerkungen zur Zusammenfassung in Abschnitt A.2.

A.1 Zusätzliche Visualisierungen

Die in diesem Abschnitt abgebildeten Darstellungen zeigen die zweidimensionale Berechnung der relNet Dokumenten-Kollektion per PCA (Abbildung A.1) und Kernel PCA (Abbildung A.2). Sie sollen die Vorteile des t-SNE im Bezug auf die zu Beginn formulierten Anforderungen verdeutlichen. Der Verusch eine Visualisierung per MDS zu bestimmen wurde abgebrochen, nachdem die Berechnung innerhalb von vier Stunden nicht zu einem Ergebnis kam. Dabei wurde wieder auf die Implementierung in der scikit-learn Bibliothek zurückgegriffen. Mit den Standardeinstellungen, also der metrischen Variante zur Berechnung von zwei Komponenten mit Hilfe des euklidischen Abstandes zur Bestimmung der Unähnlichkeit, wurde die Berechnung auf einem Intel Core i5-6360U ausgeführt.

Die PCA Visualisierung zeigt das Diagramm, welches aus den ersten beiden Hauptkomponenten resultiert. Bei der Kernel PCA ist die radiale Basisfunktion als Kernel mit $\gamma = 15$ gewählt und ebenfalls die ersten beiden Komponenten abgebildet. Die Eingabemenge für das MDS Experiment war der per PCA auf 30 Dimensionen reduzierte relNet Datensatz, wie er auch als EIngabe für die t-SNE Berechnung genutzt wurde. Die Laufzeit des MDS verdeutlicht die nicht triviale Skalierung bei großen Mengen an Daten und zeigt, dass t-SNE Laufzeitvorteile gegenüber dieser Variante der Visualisierung aufweist.

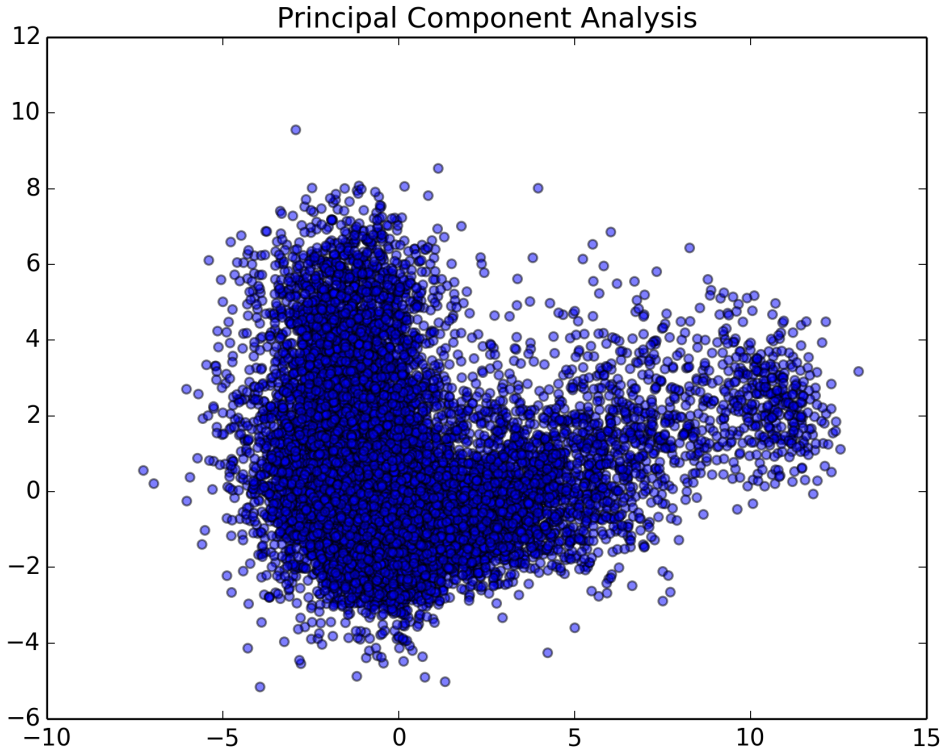


Abbildung A.1: Visualisierung relNet Dokumenten-Kollektion per PCA

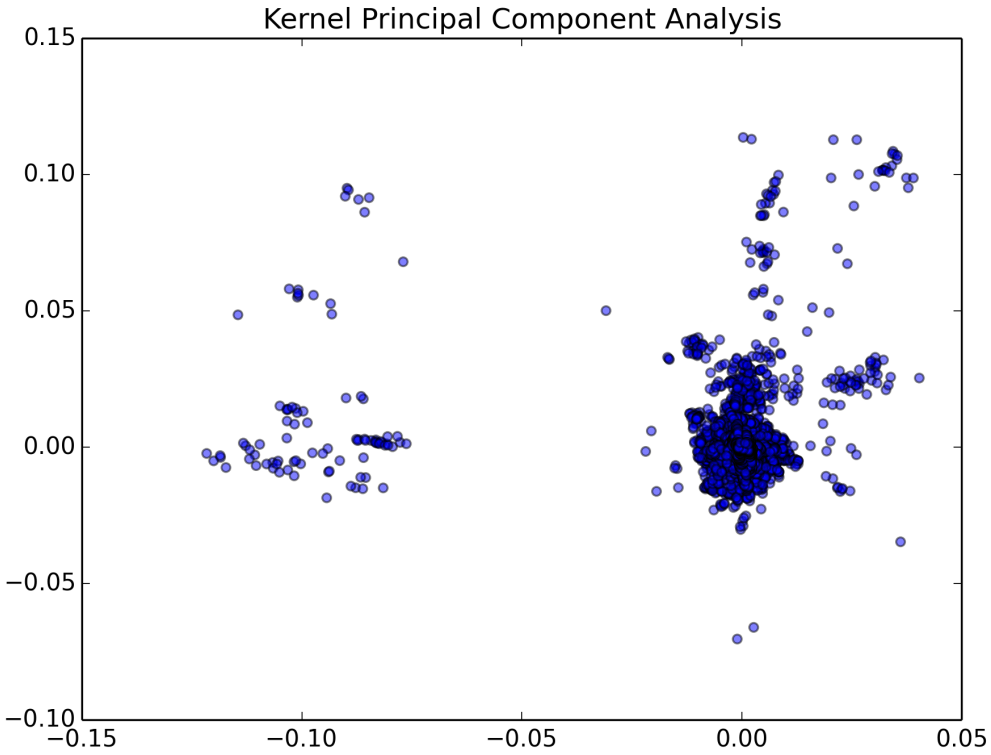


Abbildung A.2: Visualisierung relNet Dokumenten-Kollektion per Kernel PCA

A.2 Weitere Anmerkungen

Um das Fazit 8.1 nicht zu überladen, sind hier einige weitere Gedankengänge und Anmerkungen zum Ansatz, seinen Besonderheiten, sowie Vorschläge für Anpassungen zusammengefasst.

Ein weiterer wichtiger Bestandteil für den Erfolg der Visualisierung sind die Eingabedaten, in diesem Fall die bereits berechneten Document-Embeddings. Sie sind Ausgangspunkt der Ähnlichkeitsberechnung bzw. was die Algorithmen zum Clustering und der Visualisierung als ähnlich bewerten und somit essentiell bedeutend für die Gruppierung der Daten. Da dieses Themengebiet aktueller Forschungsgegenstand ist, wäre es sinnvoll, die Auswirkungen weiterer Verfahren zur Textrepräsentation auf die Darstellung nach dem hier beschriebenen Ansatz zu untersuchen.

Mit der Anpassung der Parameter wie dem Fusionierverfahren beim Clustering und der Methode zur Dimensionsreduktion, die sich auf den Erhalt anderer Eigenschaften konzentrieren, ist es möglich das Vorgehen anderen Gegebenheiten anzupassen. Bei der Wahl des Verfahrens zur Dimensionsreduktion kommt es besonders auf das Einsatzgebiet und das gewünschte Ergebnis an. Es gibt kein Verfahren, das hochdimensionale Datensätze so in der Anzahl ihrer Merkmale reduziert, sodass die komplette Struktur erhalten bleibt. Demnach gilt es besonders das Ziel der Reduktion zu bedenken. Für die Visualisierung gibt es viele Verfahren, die unterschiedliche Schwerpunkte setzen. So konzentriet sich die Hauptkomponentenanalyse auf den Erhalt der linearen Struktur, MDS auf die globale Geometrie und t-SNE auf die Nachbarschaftsstrukturen. Alle Verfahren können dabei helfen eine Intuition für die zu analysierenden Daten zu erlangen. Dabei muss aber immer im Anwendungsfall entschieden werden, wie die Prioritäten festzulegen sind.

Darüber hinaus ist es denkbar, z.B. mit dem hierarchischen Clustering und dem Single Link Verfahren, Ausreißer zu finden und den Datensatz so zu filtern. Im Falle des relNet Projektes hätten so beispielsweise die verschobenen Beiträge ausgeblended werden können, welche zur Analyse der Struktur inhaltlich nicht zwangsläufig notwendig sind. Da diese in der Visualisierung des kompletten Datensatzes klar abgetrennt sind, kann man diese Schlussfolgerung auch aus der Visualisierung ziehen. Diese und andere Methoden zur Filterung bzw. Vorverarbeitung des Datensatzes müssen dabei wieder im Kontext gesehen werden und hängen von den entsprechenden Informationen der Dokumente und dem jeweiligen Ziel der Analyse ab. So kann man in einer Vorverarbeitung die Ergebnisse stabiler machen, indem unbedeutende Randfälle, wie verschobene Beiträge die keinerlei Informationsgehalt für die Analyse der Themenstruktur bietet, im Voraus ausschließen.

Wie im Fallbeispiel schon angedeutet, konnte man teilweise die einflussreichen Personen eines bestimmten Themengebietes ausfindig machen. Diese wurden bisher eher zufällig entdeckt, daher könnte weiter erforscht werden, wie man mit eine grafischen Analyse

einflussreiche Autoren bzw. Personen ermitteln kann oder diese in das hier vorgestellte Vorgehen einbindet.

Abbildungsverzeichnis

Algorithmenverzeichnis

Literaturverzeichnis

[1] Barnes, Josh und Piet Hut: *A hierarchical O(N log N) force-calculation algorithm.* Nature, 324(4):446–449, December 1986.

[2] Bellman, Richard Ernest: *Adaptive Control Processes: A Guided Tour.* Princeton Legacy Library. Princeton University Press, 1961.

[3] Cook, James, Ilya Sutskever, Andriy Mnih und Geoffrey E. Hinton: *Visualizing similarity data with a mixture of maps.* In: *In AI and Statistics, 2007. Society for Artificial Intelligence and Statistics*, 2007.

[4] Hinton, Geoffrey E. und Sam Roweis: *Stochastic Neighbor Embedding.* Advances in neural information processing systems, 15:833–840, 2003.

[5] Hotelling, Harold: *Analysis of a Complex of Statistical Variables Into Principal Components.* Journal of Educational Psychology, 24:417–441 and 498–520, 1933.

[6] Jolliffe, Ian T.: *Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, New York, 2002.

[7] Kruskal, Joseph B und Myron Wish: *Multidimensional scaling*, Band 11. Sage, 1978.

[8] Le, Quoc und Tomas Mikolov: *Distributed Representations of Sentences and Documents.* In: Xing, Eric P. und Tony Jebara (Herausgeber): *Proceedings of the 31st International Conference on Machine Learning*, Band 32 der Reihe *Proceedings of Machine Learning Research*, Seiten 1188–1196, Bejing, China, 22–24 Juni 2014. PMLR.

[9] Lee, John M.: *Introduction to Topological Manifolds.* Springer New York, 2011.

[10] Pearson, Karl: *LIII. On lines and planes of closest fit to systems of points in space.* Philosophical Magazine Series 6, 2(11):559–572, 1901.

[11] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas,

A. Passos, D. Cournapeau, M. Brucher, M. Perrot und E. Duchesnay: *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[12] Samet, Hanan: *Neighbor finding techniques for images represented by quadtrees.* Computer Graphics and Image Processing, 18(1):37 – 57, 1982.

[13] Schölkopf, Bernhard, Alexander Smola und Klaus-Robert Müller: *Kernel principal component analysis.* In: *Advances in Kernel Methods - Support Vector Learning*, Seiten 327–352. MIT Press, 1999.

[14] Schölkopf, Bernhard und Alexander J. Smola: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[15] Steinbach, Michael, Vipin Kumar und Pang-Ning Tan: *Introduction to Data Mining: Pearson New International Edition.* Pearson, 2013.

[16] Tipping, Michael E. und Christopher M. Bishop: *Probabilistic Principal Component Analysis.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3):611–622, 1999.

[17] Trevor Hastie, Robert Tibshirani, Jerome Friedman: *The Elements of Statistical Learning.* Springer-Verlag New York, 2 Auflage, 2009.

[18] Uhlmann, Jeffrey K.: *Metric trees.* Applied Mathematics Letters, 4(5):61 – 62, 1991.

[19] van der Maaten, Laurens: *Accelerating t-SNE using tree-based algorithms.* Journal of Machine Learning Research, 15(1):3221–3245, October 2014.

[20] van der Maaten, Laurens und Geoffrey E. Hinton: *Visualizing Data Using t-SNE.* Journal of Machine Learning Research, 9:2579–2605, 2008.

[21] Yianilos, Peter N.: *Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces.* In: *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, Seiten 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

Eidesstattliche Versicherung

Kilian, Phillip | 165606

Name, Vorname | Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit/~~Masterarbeit~~* mit dem Titel

Visualisierung von Embeddings zur Analyse großer Dokumenten-Kollektionen

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, 03.06.2017 | _____

Ort, Datum | Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Dortmund, 03.06.2017 | _____

Ort, Datum | Unterschrift