

Analyse des UMAP Verfahrens

Christopher Reiners

Geboren am 9. April 1998 in Detmold

6. August 2019

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Jochen Garcke

Zweitgutachter: Dr. Bastian Bohn

MATHEMATISCHES INSTITUT FÜR NUMERISCHE SIMULATION

MATHEMATISCHE-NATURWISSENSCHAFTLICHE FAKULTÄT DER
RHEINISCHEN FRIEDRICH-WILHELMUS-UNIVERSITÄT BONN

Danksagung

An dieser Stelle möchte ich gerne Prof. Dr. Jochen Garcke für die Vergabe dieses sehr interessanten und spannenden Themas, danken. Besonderer Dank gilt Leland McInnes für das persönliche Gespräch. Dadurch konnte ich die Motivation, welche er für das UMAP Verfahren hatte, aus erster Hand erfahren.

Gerne möchte ich auch meinen Freunden danken, welche mich im Laufe der Studienzeit, besonders während der Bachelorarbeit, begleitet haben. Annalena für die vielen Eispausen und die lustigen Unterhaltungen auch spät in der Nacht, Tobi, Hendrik und Lenard für die Motivation und den mathematischen Austausch und Kim und Lukas für die hilfreichen und aufmunternden Gespräche. Ohne euch wäre diese Arbeit nicht entstanden.

Zum Schluss danke ich meinen Eltern und Caro für Ihre bedingungslose Unterstützung, trotz Einsilbigkeit meiner Antworten in den letzten Wochen meiner Arbeit.

Inhaltsverzeichnis

1 Einleitung	1
2 Grundlagen	5
2.1 Topologische Räume	5
2.2 Kategorientheorie	8
2.3 Simpliziale Mengen	12
2.4 Unscharfe Mengen	15
3 UMAP	19
3.1 Approximation der Mannigfaltigkeit	19
3.2 Topologische Repräsentation	21
3.3 Einbettung der Repräsentation	23
4 Implementierung	25
4.1 Numerische Formulierung des UMAP Verfahrens	25
4.2 Profiling	28
4.3 Gradientenverfahren	28
4.4 Nächste-Nachbarn-Klassifikation	29
4.5 Hyperparameter	31
5 Experimente	33
5.1 Alternative Verfahren	33
5.1.1 t-SNE	33
5.1.2 TriMap	34
5.2 Bewertung der Ergebnisse	35
5.3 Cartoon Set	35
5.4 MNIST	38
5.5 Laufzeitanalyse	38
6 Zusammenfassung und Ausblick	43
6.1 Zusammenfassung	43
6.2 Ausblick	43
Literatur	45

Kapitel 1

Einleitung

Wer heutzutage eine Zeitung aufschlägt wird mit hoher Wahrscheinlichkeit kontroverse Artikel über künstliche Intelligenz, Machine Learning, Deep Learning oder Big Data finden. Diese Themen haben nicht nur Relevanz und Auswirkungen für große Konzerne wie Google, Facebook oder Amazon, sondern auch für unser alltägliches privates Leben. Jeden Tag werden neue Daten erzeugt und damit die Suche nach effizienten Algorithmen, die mit großen Datenbeständen umgehen können wichtiger. Algorithmen helfen Muster in vorhandenen Datenbeständen zu erkennen, Vorhersagen zu treffen oder Daten zu klassifizieren. Mit mathematischen Modellen können neue Erkenntnisse auf Grundlage dieser Muster gewonnen werden. Ziel ist es also große Datenmengen in Informationen und diese Informationen in Erkenntnisse zu überführen. Die Aufgabe, effiziente Verfahren zur Datenanalyse zu entwerfen, ist somit von hoher Relevanz.

Um einen Überblick der Aufgabenbereiche der Datenanalyse zu erhalten kann man diese in drei Felder unterteilen¹. Die *deskriptive* Datenanalyse beschreibt Daten, beispielsweise durch graphische Visualisierung. Damit ist sie verbunden mit der *explorativen* Datenanalyse. Diese dient zum Entdecken neuer Zusammenhänge, oft werden dazu graphische Methoden genutzt. Die *inferentielle* Datenanalyse nutzt gegebene Stichproben, um auf nicht erhobene Stichproben zu schließen. Dieser Bereich hat in den vergangenen Jahren stark an Bedeutung gewonnen, insbesondere aufgrund der sehr guten Resultate, die durch neuronale Netzwerke erzielt wurden.

In dieser Arbeit möchten wir uns mit *Verfahren zur Dimensionsreduktion* beschäftigen. Diese dienen der deskriptiven Datenanalyse und der explorativen Datenanalyse. Zusätzlich lassen sie sich dazu nutzen, Verfahren zur inferentiellen Datenanalyse zu verbessern. Wir werden etwas später auf die Anwendungen zurückkommen. Doch zu erst möchten wir die Problemstellung formulieren.

Dimensionsreduktion

Sei X eine Menge an Daten. Dabei betrachten wir von nun an Daten, welche uns in Form von D -dimensionalen reellen Vektoren geben sind, somit ergibt sich für N Datenpunkte, $X = \{\mathbf{x}_i\}_{1 \leq i \leq N}$ mit $\mathbf{x}_i \in \mathbb{R}^D$. Wenn $D > 10$ sprechen wir meist von *hochdimensionalen Daten*. Dabei ist die Beschränkung auf Daten, welche sich als Vektoren beschreiben lassen, sinnvoll, da uns die Darstellung einerseits die mathematische Formulierung des Problems erleichtert. Andererseits können sehr verschiedene Daten in dieser Form angegeben werden. So kann man ein digitales Bild als Vektor auffassen, indem die Dimension der Anzahl an Pixeln entspricht und die Farbwerte der Pixel die Einträge der Vektoren geben. Wörter können ebenfalls als Vektoren gegeben werden indem beispielsweise die Dimension der Gesamtzahl der Wörter entspricht und die Wörter durch Einheitsvektoren darstellt werden. Wir sehen, dass die Anzahl an Dimensionen sehr groß werden kann. Eine moderne Kamera macht Bilder mit meist

¹<https://wirtschaftslexikon.gabler.de/definition/datenanalyse-30331/version-164492>



ABBILDUNG 1.1: Einbettung von Bildern mit $D = 784, N = 70\,000$ (MNIST Datensatz). Die Farben markieren unterschiedliche Ziffern von 0 bis 9. Eine genauere Beschreibung findet sich in Kapitel 5.

mehr als 1920×1080 Pixeln, dies entspricht $D = 2073\,600$. Die deutsche Sprache hat circa 23 Millionen unterschiedliche Wörter, somit wäre $D = 23\,000\,000$. Wir sprechen von einer *Dimensionsreduktion* von X , wenn wir eine *Einbettung* Y angeben, mit $Y = \{\mathbf{y}_i\}_{1 \leq i \leq N}$ und $\mathbf{y}_i \in \mathbb{R}^d$, wobei $d \ll D$, so dass sich X und Y ähnlich sind.

Wir möchten nun zurück auf die Ziele der Datenanalyse kommen. Die Einbettung Y kann dazu dienen, die Daten X zu visualisieren, wenn $d \in \{1, 2, 3\}$. Da X und Y ähnlich sind, können durch die Visualisierung von Y Rückschlüsse auf X gemacht werden. Oft lassen sich in der Visualisierung sogenannte Cluster erkennen, siehe dazu Abbildung 1.1. Neben der Visualisierung kann die Einbettung Y zur Vorverarbeitung genutzt werden, also als Eingabe für einen anderen Algorithmus der Datenanalyse. So werden Dimensionsreduktionen beispielsweise für die Gesichtserkennung genutzt. Die Einbettung von X benötigt zudem aufgrund der deutlich kleineren Dimension weniger physischen Speicherplatz, dieser Aspekt ist zwar nicht entscheidend für die Datenanalyse, dennoch erwähnenswert, insbesondere dann, wenn nur begrenzt viel Speicher zur Verfügung steht.

Eine häufig getroffene Annahme ist, dass die Daten X auf einer niedrigdimensionalen Struktur liegen. In der Literatur ist diese Annahme als Mannigfaltigkeit-Hypothese (engl.: *manifold hypothesis*) bekannt [30, 36]. Dabei kann sich eine Mannigfaltigkeit als Raum vorgestellt werden, welcher lokal einem niedrigdimensionalen Raum gleicht. So ist beispielsweise der Erdball eine 2-dimensionale Mannigfaltigkeit, da sie lokal gesehen flach ist. Die Intuition, welche diese Annahme rechtfertigt ist somit, dass wir Daten messen, welche nicht relevant sind. Beispielsweise ist der Hintergrund in einem Bild von einem Gesicht bei der Gesichtserkennung nicht relevant.

Wir müssen nun den Begriff der *Ähnlichkeit* von X und Y präzisieren und angeben wie wir Y systematisch finden können. Dabei lässt sich die Frage nach der *Ähnlichkeit* mathematisch unterschiedlich formulieren. Wir verweisen den Leser auf [18, 34].

Verfahren zur Dimensionsreduktion

Um für hochdimensionale Daten X eine niedrigdimensionale Einbettung zu finden, wendet man *Dimensionsreduktionsverfahren* (kurz: *DR Verfahren*) an. Dabei gibt es

DR Verfahren, welche bei der Einbettung globale Distanzen bevorzugen. Bekannte Verfahren dabei sind PCA, MDS und Sammon Mapping. Zu den Verfahren, welche die lokalen Distanzen bevorzugen, gehören Isomap, Laplacian Eigenmaps und t-SNE. Die Begriffe *lokal* und *global* sind dabei nicht fest definiert. Wir werden später genauer auf einige Verfahren eingehen.

UMAP

Der Fokus dieser Arbeit liegt auf der Darstellung und Analyse des UMAP Verfahrens [27]. UMAP ist ein Verfahren zur Dimensionsreduktion und kommt aus dem Bereich der *topologischen Datenanalyse*, diese nutzt mathematische Instrumente der Topologie um Daten zu beschreiben. Das Verfahren nimmt die Mannigfaltigkeit-Hypothese an. Im ersten Schritt des Verfahrens betrachtet man von jedem Datenpunkt die Distanz zu seinen benachbarten Punkten und nutzt diese und zusammen mit einer Aussage der Riemannschen Geometrie um lokale Distanzen, auf der Mannigfaltigkeit, zwischen den Datenpunkten zu bestimmen. Diese Konstruktion liefert uns für jeden Datenpunkt einen *metrischen Raum*, wobei die *Metriken* zwischen den Räumen a priori nicht miteinander kompatibel sind. Um dieses Problem zu lösen werden wir für jeden metrischen Raum eine *unscharfe simpliziale Menge* konstruieren, welche die wichtigen Informationen des metrischen Raumes enthält. Die simplizialen Mengen können wir miteinander vereinigen. Ähnlich konstruieren wir für eine initiale Einbettung Y eine zugehörige simpliziale Menge. Wir werden dann einen Abstandsbegriff zwischen den simplizialen Mengen nutzen um die Darstellung von Y zu optimieren.

Eigene Beiträge

Wir möchten an dieser Stelle darauf hinweisen, dass sich die Darstellung der Theorie des UMAP Verfahrens nach der ursprünglichen Veröffentlichung von McInnes et. al. [27] richtet. Allerdings wurde diese an vielen Stellen durch anschauliche Erklärungen und Intuitionen ergänzt. Die von uns gemachten Beiträge sind dabei,

- eine übersichtliche Zusammenfassung der für das UMAP Verfahren wichtigen Definitionen und Sätze mit intuitiven Erläuterungen,
- eine um Intuitionen ergänzte Darstellung des UMAP Verfahrens,
- eine Betrachtung der rechenintensiven Schritte der Implementierung und alternative Methoden für diese,
- eine Anwendung des UMAP Verfahrens auf einen neuen komplexen Datensatz,
- Ansätze für weitere Betrachtungen zur Verbesserung des UMAP Verfahrens.

Aufbau der Arbeit

In Kapitel 2 werden wir die für das UMAP Verfahren benötigten Definitionen und Sätze angeben. Dabei werden wir zuerst auf den Begriff der riemannschen Mannigfaltigkeit hinarbeiten, welcher aufgrund der Mannigfaltigkeit-Hypothese die Grundlage für weitere Überlegungen legt. Dann werden *Kategorien* und wichtige Aussagen der Kategorientheorie vorgestellt. Mithilfe von Kategorien können wir dann *simpliziale Mengen* einführen, welche die Grundlage für die Repräsentation der Daten legen. Diesen Begriff werden wir auf unscharfe simpliziale Mengen erweitern.

Die Aussagen werden uns in Kapitel 3 helfen, die Theorie des UMAP Verfahrens darzustellen. Diese soll in drei Schritten beschrieben werden, zuerst werden wir die

Mannigfaltigkeit approximieren und eine Familie metrischer Räume konstruieren. Danach werden wir diese metrischen Räume in unscharfe simpliziale Mengen umwandeln. Dies wird uns eine Repräsentation \mathcal{X} der Daten X liefern. Zuletzt werden wir eine ähnliche Repräsentation \mathcal{Y} für Y konstruieren und beschreiben, wie die Repräsentation \mathcal{Y} optimiert werden kann, damit sie \mathcal{X} möglichst ähnlich ist. Das bezüglich \mathcal{X} optimierte \mathcal{Y} kann dann mittels der eingeführten Theorie als metrischer Raum dargestellt werden, dieser liefert uns die Einbettung von X .

Kapitel 4 wird einen Fokus auf die Implementierung des UMAP Verfahren legen. Zunächst soll die UMAP Theorie angepasst werden, um eine effiziente Implementierung zu ermöglichen. Wir werden dann eine Implementierung [28] betrachten und die rechenintensiven Subroutinen angeben. Weitere Überlegungen sollen Alternativen beschreiben. Wir werden diese Kapitel mit einer Übersicht der Hyperparameter des UMAP Verfahrens abschließen.

Das UMAP Verfahren soll in Kapitel 5 mit zwei verwandten Verfahren auf unterschiedlichen Datensätzen analysiert werden. Dazu werden wir eine kurze Beschreibung des t-SNE und TriMap Verfahren geben. Mit diesen drei Verfahren werden wir einen Datensatz mit Cartoon Gesichtern analysieren. Diese Analyse soll testen, wie gut die Einbettungen der Verfahren sind. Zusätzlich sollen die Verfahren auf dem bekannten MNIST Datensatz getestet werden.

Die Resultate dieser Arbeit sollen in Kapitel 6 zusammengefasst werden. Außerdem geben wir in diesem Kapitel Ansätze nennen um das UMAP Verfahren zu verbessern.

Kapitel 2

Grundlagen

Das UMAP Verfahren entstammt dem Gebiet der topologischen Datenanalyse. Die Theorie für das Verfahren nutzt Grundlagen aus den Bereichen der (algebraischen) Topologie, Kategorientheorie und Mengentheorie. Wir wollen diese nun einführen. Dazu geben wir die wichtigsten Definitionen und Sätze und werden diese anschaulich erklären und in den Rahmen des UMAP Verfahren fassen.

Die grundlegenden Definitionen *topologischer und metrischer Räume* in Abschnitt 2.1 werden uns helfen (*riemannsche*) *Mannigfaltigkeiten* einzuführen. Der Begriff der Mannigfaltigkeit formalisiert den niedrigdimensionalen Raum auf, welchem der zu untersuchende Datensatz X liegt.

Die geometrische und topologische Struktur dieser Räume sollen durch *simpliziale Mengen* dargestellt werden. Um diese in Abschnitt 2.3 einzuführen, benötigen wir grundlegende Definitionen aus der Kategorientheorie, siehe dazu Abschnitt 2.2.

In Abschnitt 2.4 werden *unscharfe Mengen* eingeführt, diese werden in Kapitel 3 benötigt um der topologischen Repräsentation der Daten X eine metrische Struktur zu verleihen.

2.1 Topologische Räume

Der Grundlegende Begriff der Topologie ist der des topologischen Raumes.

Definition 2.1. Sei X eine nichtleere Menge. Ein Mengensystem $\tau \subset \mathcal{P}(X)$ heißt *Topologie auf X* , falls die folgenden drei Bedingungen erfüllt sind:

1. $\emptyset, X \in \tau$,
2. die Vereinigung beliebig vieler Mengen aus τ liegt wieder in τ ,
3. sind $U, V \in \tau$, so liegt auch der Durchschnitt $U \cap V$ in τ .

Das Paar (X, τ) heißt *topologischer Raum*. Die Mengen $U \in \tau$ nennt man *offene Mengen* des topologischen Raumes.

Bemerkung. Wenn die Topologie τ eines topologischen Raumes (X, τ) aus dem Kontext klar ist, wird diese nicht explizit erwähnt.

Man kann den Begriff erweitern indem man einen Abstandsbegriff auf der Menge X definiert. Dies führt uns zum metrischen Raum.

Definition 2.2. Sei X eine nichtleere Menge. Eine Abbildung $d : X \times X \rightarrow \mathbb{R}_{\geq 0}$ heißt *Metrik* auf X , falls für beliebige Elemente $x, y, z \in X$ die folgenden drei Bedingungen erfüllt sind:

1. $d(x, y) = 0 \iff x = y$,

2. $d(x, y) = d(y, x)$,
3. $d(x, y) \leq d(x, z) + d(z, y)$.

Das Paar (X, d) heißt *metrischer Raum*. Die Metrik d heißt *Pseudometrik*, wenn die erste Bedingung durch $d(x, y) = 0 \Leftrightarrow x = y$ abgeschwächt wird.

Falls die Metrik den Wert ∞ annehmen kann, sprechen wir von einer *erweiterten Metrik*.

Bemerkung. In Fakt ist die Erweiterung einer Metrik um den Wert ∞ keine Einschränkung. Für eine gegebene erweiterte Metrik d kann nämlich stets äquivalente eine (echte) Metrik d' konstruiert werden, zum Beispiel ist $d' = \frac{d}{1+d}$ äquivalent zu einer erweiterten Metrik d . Eine formale Definition der Äquivalenz zweier metrischer Räume soll hier nicht gegeben werden. Es genügt zu wissen, dass wichtige topologische Eigenschaften erhalten zwischen den beiden Räumen übertragen werden können. Hier gilt $x + \infty = \infty + x = \infty, x \in [0, \infty]$.

Bemerkung. Für einen metrischen Raum (X, d) , ist ein offener *Ball mit Radius $r > 0$* und Mittelpunkt p mit p aus X gegeben durch:

$$B_r(p) := \{x \in X \mid d(x, p) < r\}. \quad (2.1)$$

Für den Fall, das X der n -dimensionale euklidische Raum ist, ist das n -dimensionale Volumen bezüglich der euklidischen Metrik:

$$V_n(B_r(p)) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)} r^n \quad (2.2)$$

In der Einleitung bereits erwähnt, werden wir Annnehmen, dass unsere Daten $X = \{\mathbf{x}_i\}_{i=1}^N, (\mathbf{x}_i \in \mathbb{R}^D)$ mit D gemessenen Eigenschaften mittels $d, (d \ll D)$ Eigenschaften dargestellt werden können. Um dies in die Sprache der Topologie zu fassen, werden wir den Begriff der *Mannigfaltigkeit* benötigen. Anschaulich ist eine d -dimensionale Mannigfaltigkeit ein topologischer Raum welcher lokal dem euklidischen Raum \mathbb{R}^d gleicht. Bevor wir Mannigfaltigkeiten einführen benötigen wir *homöomorphe Abbildungen*.

Definition 2.3. Seien X und Y topologische Räume. Eine Abbildung $f : X \rightarrow Y$ ist ein *Homöomorphismus*, wenn gilt:

1. f ist bijektiv,
2. f ist stetig also, wenn die Urbilder offener Mengen wieder offen sind,
3. die Umkehrfunktion f^{-1} ist ebenfalls stetig.

Wenn ein Homöomorphismus $f : X \rightarrow Y$ gibt, so nennen wir X und Y *homöomorph*.

Definition 2.4. Sei \mathcal{M} ein topologischer Raum, $d \in \mathbb{N}$, er heißt *d-dimensionale Mannigfaltigkeit*, wenn folgende drei Bedingungen erfüllt sind:

1. für alle paarweise verschiedenen Punkte $p, q \in \mathcal{M}$ existieren disjunkte offene Mengen $U, V \subseteq \mathcal{M}$ mit $p \in U$ und $q \in V$,
2. die Topologie von \mathcal{M} besitzt eine abzählbare Basis

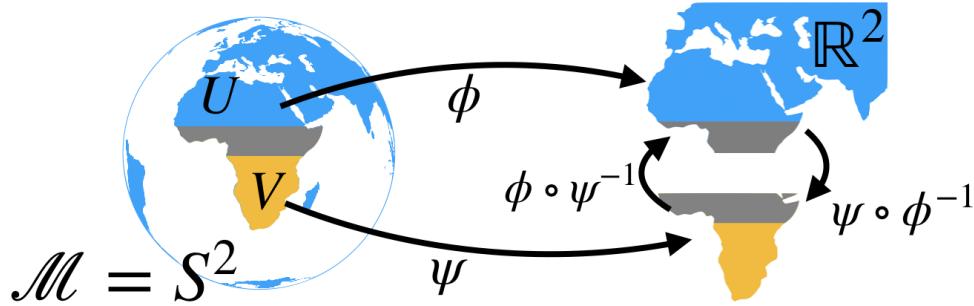


ABBILDUNG 2.1: Zwei Karten einer Mannigfaltigkeit.

3. jeder Punkt in \mathcal{M} besitzt eine Umgebung, die homöomorph zu einer offenen Teilmenge des \mathbb{R}^d ist.

Um eine *riemannsche Mannigfaltigkeit* definieren zu können, benötigen wir noch einige Definitionen.

Definition 2.5. Eine Abbildung $f : U \rightarrow V$ zwischen offenen Mengen $U, V \subset \mathbb{R}^n$ heißt *C^k-Diffeomorphismus*, falls

1. f ist bijektiv
2. f ist überall k -mal stetig differenzierbar
3. die Umkehrabbildung f^{-1} ist überall k -mal stetig differenzierbar.

Definition 2.6. Es sei \mathcal{M} eine Mannigfaltigkeit der Dimension d . Eine *Karte* auf \mathcal{M} ist ein Paar (U, ϕ) , wobei $U \subseteq \mathcal{M}$ eine offene Menge und $\phi : U \rightarrow \phi(U) \subseteq \mathbb{R}^d$ ein Homöomorphismus mit $\phi(U) \subseteq \mathbb{R}^d$ ist.

Sind (U, ϕ) und (V, ψ) zwei Karten von \mathcal{M} mit $U \cap V \neq \emptyset$, so nennt man die Abbildung

$$\psi \circ \phi^{-1} : \phi(U \cap V) \rightarrow \psi(U \cap V) \quad (2.3)$$

einen *Kartenwechsel*.

Ein *Atlas* für \mathcal{M} ist dann eine Familie $(U_i, \phi_i)_{i \in I}$ von Karten, so dass $\mathcal{M} = \cup_{i \in I} U_i$ gilt. Man nennt einen Atlas *C^k-differenzierbar* mit $k \geq 1$, wenn alle seine Kartenwechsel *C^k-Diffeomorphismen* sind.

Karten werden genutzt um zwischen der Mannigfaltigkeit und dem \mathbb{R}^d zu übersetzen, siehe Abbildung 2.1.

Definition 2.7. Eine *differenzierbare* Mannigfaltigkeit ist ein Paar $(\mathcal{M}, \mathcal{A} = (U_i, \phi_i)_{i \in I})$, wobei \mathcal{M} eine n -dimensionale Mannigfaltigkeit und \mathcal{A} ein C^1 -Atlas auf \mathcal{M} ist. Man nennt den Atlas oft *C¹-Struktur* auf \mathcal{M} .

Nun können wir im Fall der Dimensionsreduktion die Vermutung aufstellen, dass unsere Daten X auf einer d -dimensionale Mannigfaltigkeit \mathcal{M} im \mathbb{R}^D liegen. Zusätzlich gehen wir davon aus, dass unsere Daten einen Abstandsbegriff erlauben. Somit lässt sich die Definition der Mannigfaltigkeit auf die Riemannschen Mannigfaltigkeiten erweitern.

Definition 2.8. Ein differenzierbares Vektorfeld X auf $\Sigma \subset \mathbb{R}^n$ ist eine Abbildung, welche jedem Punkt $p \in \Sigma$ einen Vektor $v(p) \in \mathbb{R}^n$ zuordnet. Ist X differenzierbar, so ist X ein differenzierbares Vektorfeld.

Sei \mathcal{M} eine n -dimensionale differenzierbare Mannigfaltigkeit. Eine *Riemannsche Metrik* g auf \mathcal{M} ist eine Familie positiv-definierter innerer Produkte $g_p : T_p \mathcal{M} \times T_p \mathcal{M} \rightarrow \mathbb{R}$, für $p \in \mathcal{M}$, so dass für jedes Paar differenzierbare Vektorfelder X, Y auf \mathcal{M} ,

$$\mathcal{M} \rightarrow \mathbb{R}, \quad p \mapsto g_p(X|_p, Y|_p) \quad (2.4)$$

eine glatte Funktion ist.

Wir nennen das Paar (\mathcal{M}, g) eine *Riemannsche Mannigfaltigkeit*.

Bemerkung. Eine bekannte Aussage der Riemannschen Geometrie ist, dass jede differenzierbare Mannigfaltigkeit eine riemannsche Metrik besitzt. Somit ist die Definition der riemannschen Mannigfaltigkeit keine starke Einschränkung.

Nun möchten wir den Begriff der Abstandsfunktion einer Riemannschen Mannigfaltigkeit geben. Dieser wird uns erlauben die kürzesten Wege auf einer riemannschen Mannigfaltigkeit zu betrachten.

Definition 2.9. Sei (\mathcal{M}, g) eine Riemannsche Mannigfaltigkeit. Die *Abstandsfunktion* auf (\mathcal{M}, g) ist gegeben durch

$$d(x, y) := \inf L(\gamma) | \gamma : [0, 1] \rightarrow \mathcal{M}, \gamma(0) = x, \gamma(1) = y, \gamma \text{ ist stückweise differenzierbar}, \quad (2.5)$$

wobei

$$L(\gamma) := \int_0^1 \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \quad (2.6)$$

Wenn γ ein Weg mit konstanter durchlaufender Geschwindigkeit ist welcher lokal die kürzeste Verbindung realisiert, heißt *Geodäte*. Eine formale Definition dieser Aussage benötigt einige Vorarbeit und es muss die *Levi-Civita-Ableitung* definiert werden. Dies würde den Rahmen der Arbeit überschreiten und wir verweisen den interessierten Leser auf ein Lehrbuch der Riemannschen Geometrie. Wichtig für weitere Überlegungen wird sein, dass eine Geodäte die kürzeste Distanz zwischen zwei Punkten einer Riemannschen Mannigfaltigkeit definiert.

2.2 Kategorientheorie

Die für die mathematischen Grundlagen des UMAP Verfahren benötigten Definitionen werde ich mithilfe der Kategorientheorie einführen. Diese sehr abstrakte Form mathematische Objekte und Zusammenhänge zu formalisieren wurde erstmals in den vierziger Jahren von Eilenberg und Mac Lane eingeführt [10].

Eine Kategorie ist eine Menge an Objekten und Abbildungen zwischen diesen, die Abbildungen nennt man Morphismen. Dabei können sehr unterschiedliche mathematische Konstrukte in der Sprache der Kategorientheorie formuliert werden. Zuerst möchten wir eine Kategorie definieren und einige Beispiele geben. Für das UMAP Verfahren werden wir die Kategorie der metrischen Räume und der *simplizialen Mengen* betrachten. Die Formulierung eines Konstruktions, zum Beispiel der metrischen Räume, als Kategorie ist meist trivial. Eine Bedeutung erhält eine Kategorie erst, wenn sie in Verbindung mit anderen Kategorien gebracht wird. Dafür werden wir in diesem

Abschnitt die *Adjunktion* einführen. Die Kategorientheorie wird oft dazu genutzt Belege zu vereinfachen, insbesondere wird dafür die relativ unscheinbare Kategorie **Set** verwendet, siehe dazu das *Yoneda Lemma*.

Definition 2.10. Eine Kategorie \mathcal{C} besteht aus folgenden Daten:

1. Eine Klasse $Ob(\mathcal{C})$, deren Elemente wir *Objekte* nennen
2. zu je zwei Objekten $A, B \in Ob(\mathcal{C})$ einer Menge $\text{Hom}_{\mathcal{C}}(A, B)$, deren Elemente wir mit $f : A \rightarrow B$ notieren und *Morphismen* von A nach B nennen,
3. zu je drei Objekten $A, B, C \in Ob(\mathcal{C})$ einer Abbildung

$$\text{Hom}_{\mathcal{C}}(A, B) \times \text{Hom}_{\mathcal{C}}(B, C) \rightarrow \text{Hom}_{\mathcal{C}}(A, C)$$

die wir mit $(f, g) \mapsto g \circ f$ notieren und *Komposition von Morphismen* nennen,

4. zu jedem Objekt $A \in Ob(\mathcal{C})$ einen ausgezeichneten Morphismus

$$id_A \in \text{Hom}_{\mathcal{C}}(A, A),$$

welchen wir die *Identität* von A nennen.

Diese Daten müssen den folgenden Regeln genügen:

1. Die Komposition von Morphismen ist *assoziativ*: Für drei Morphismen der Form $f : A \rightarrow B, g : B \rightarrow C, h : C \rightarrow D$ in \mathcal{C} gilt

$$h \circ (g \circ f) = (h \circ g) \circ f$$

als Morphismen $A \rightarrow D$.

2. Die Identität sind *beidseitig neutral* bezüglich der Komposition: Für jeden Morphismus $f : A \rightarrow B$ in \mathcal{C} gilt

$$f \circ id_A = f = id_B \circ f$$

Eine *Unterkategorie* einer Kategorie \mathcal{C} besitzt als Objekte und Morphismen je eine Teilmenge der Objekte bzw. Morphismen von \mathcal{C} .

Die *duale Kategorie* \mathcal{C}^{op} einer Kategorie \mathcal{C} ist gegeben durch $Ob(\mathcal{C}^{op}) = Ob(\mathcal{C})$ und $\text{Hom}_{\mathcal{C}^{op}}(A, B) = \text{Hom}_{\mathcal{C}}(B, A)$.

Bemerkung. Anstelle von $A \in Ob(\mathcal{C})$ schreibt man meistens $A \in \mathcal{C}$. Falls die Kategorie \mathcal{C} aus dem Kontext bekannt ist, werden wir $\text{Hom}_{\mathcal{C}}(A, B)$ mit $\text{Hom}(A, B)$ abkürzen.

Bemerkung. Bei Klassen kann es sich um Ansammlungen von Objekten handeln, die „größer“ als Mengen sind. Für eine Definition einer Klasse verweisen wir auf [19]. In unseren Beispielen genügt die Vorstellung einer Menge. Meist ist $Ob(\mathcal{C})$ sogar eine Menge. Dann spricht man formal von einer strikten kleinen Kategorie.

Wir werden nun einige Beispiele von Kategorien geben, dabei soll später auf diese Beispiele zurückgegriffen werden, um die Theorie des UMAP Verfahrens weiter aufzubauen.

Beispiel 2.1. Die Kategorie der Mengen **Set** hat als Objekte Mengen und als Morphismen Funktionen. Diese sehr einfache Definition scheint auf den ersten Blick etwas nutzlos. Es wird sich im Laufe des Abschnitts zeigen, dass **Set** in der Kategorientheorie eine zentrale Rolle spielt.

Beispiel 2.2. **Top** bezeichnet die Kategorie der topologischen Räume, wobei die Objekte topologische Räume sind und Morphismen stetige Funktionen.

Beispiel 2.3. Die Kategorie **EPMet** hat als Objekte erweiterte pseudo-metrische Räume und für Morphismen $\phi \in \text{Hom}_{\text{EPMet}}((X, d), (X', d'))$ gilt, $d'(\phi(x), \phi(y)) \leq d(x, y)$, die Morphismen nennen wir *nicht erweiternde Abbildungen*. Die Unterkategorie der endlichen erweiterten pseudo-metrischen Räume bezeichnen wir mit **Fin-EPMet**. Diese Kategorie wird in Kapitel 3 verwendet um unsere Daten X zu beschreiben.

Beispiel 2.4. Die *Kategorie der offenen Teilmengen* **Op**(X) eines topologischen Raumes (X, τ) hat als Objekte die offenen Mengen, also $U \subset \tau$ und als Morphismen die Inklusionsabbildungen, also $f : U \hookrightarrow V$ für $U \subset U \subset \tau$.

Wir möchten nun den Begriff des Funktors zwischen zwei Kategorien einführen. Ein Funktor ordnet Objekte einer Kategorie \mathcal{C} Objekten einer Kategorie \mathcal{D} zu, und entsprechend für Morphismen. Insbesondere bleibt die Eigenschaft der Isomorphie zwischen zwei Objekten erhalten.

Definition 2.11. Es seien \mathcal{C} und \mathcal{D} zwei Kategorien. Ein *Funktör*

$$F : \mathcal{C} \rightarrow \mathcal{D}$$

von \mathcal{C} nach \mathcal{D} besteht aus folgenden Daten:

1. für jedes Objekt $A \in \mathcal{C}$ ein Objekt $F(A) \in \mathcal{D}$,
2. für jeden Morphismus $f : A \rightarrow B$ in \mathcal{C} einen Morphismus

$$F(f) : F(A) \rightarrow F(B)$$

in \mathcal{D} .

Dabei soll gelten:

1. Für jedes Objekt $A \in \mathcal{C}$ ist $F(id_A) = id_{F(A)}$.
2. Für je zwei Morphismen $f : A \rightarrow B$, $g : B \rightarrow C$ in \mathcal{C} gilt in \mathcal{D} :

$$F(g \circ f) = F(g) \circ_{\mathcal{D}} F(f)$$

In manchen Fällen spricht man von der *Wirkung* des Funktors auf Morphismen und bezeichnet damit das Bild von f unter F .

Bemerkung. Beziiglich der Kategorie \mathcal{C} ist ein Funktor $F : \mathcal{C} \rightarrow \mathcal{D}$ kovariant, während $F : \mathcal{C}^{op} \rightarrow \mathcal{D}$ kontravariant (bzgl. \mathcal{C}) ist.

Bemerkung. Insbesondere kann man für eine strikte kleine Kategorie \mathcal{C} und Objekte $A, B, C \in \mathcal{C}$ den kontravarianten *Hom-Funktör* definieren, indem man

$$\text{Hom}(-, B) : \mathcal{C} \rightarrow \mathbf{Set} \tag{2.7}$$

betrachtet. Der Hom-Funktör bildet ein Objekt $A \in \mathcal{C}$ auf die Menge der Morphismen $\text{Hom}(A, B)$ ab, und einen Morphismus $h : A \rightarrow C$ auf die Funktion

$$\text{Hom}(h, B) : \text{Hom}(C, B) \rightarrow \text{Hom}(A, B), \text{ wobei } g \mapsto g \circ h \text{ für } g \in \text{Hom}(C, B) \tag{2.8}$$

Analog lässt sich der kovariante Hom-Funktör $\text{Hom}(A, -)$ definieren.

Eine nützliche Definition um auf sinnvolle Weise Funktoren ineinander zu überführen und dabei die Komposition der Morphismen zu berücksichtigen sind *natürliche Transformationen*.

Definition 2.12. Seien $F, G : \mathcal{C} \rightarrow \mathcal{D}$. Eine *natürliche Transformationen* t von F nach G ist eine Familie von Morphismen t , mit

- für jedes $A \in \mathcal{C}$ gibt es einen Morphismus $t_A : F(A) \rightarrow G(A)$ zwischen Objekten in \mathcal{D} ,
- für beliebige Objekte A, B und beliebige Morphismus $f \in \text{Hom}_{\mathcal{C}}(A, B)$ gilt: $t_A \circ F(f) = G(f) \circ t_B$.

Wir nennen F und G *natürlich äquivalent*, wenn es natürliche Transformationen $t : F \rightarrow G, u : G \rightarrow F$ gibt, so dass $t \circ u = id_G$ und $u \circ t = id_F$. Falls t_A für ein beliebiges $A \in \mathcal{C}$ ein Isomorphismus in \mathcal{D} ist, dann bezeichnen wir \mathcal{D} als *natürlichen Isomorphismus*.

Die folgende Definition wird uns eine abgeschwächte Form einer Äquivalenz von Kategorien liefern. Sie ermöglicht uns verschiedene Kategorien miteinander zu verbinden.

Definition 2.13. Seien \mathcal{C}, \mathcal{D} Kategorien und $F : \mathcal{C} \rightarrow \mathcal{D}$ und $G : \mathcal{D} \rightarrow \mathcal{C}$ Funktoren. Dann bilden \mathcal{C} und \mathcal{D} ein *adjungiertes Funktorpaar*, wenn die Funktoren

$$(A, B) \mapsto \text{Hom}_{\mathcal{D}}(A, F(B)) \quad (A, B) \mapsto \text{Hom}_{\mathcal{D}}(G(A), B), \quad (2.9)$$

natürlich äquivalent sind. Dabei bilden die Funktoren von $\mathcal{D}^{op} \times \mathcal{C}$ in die Kategorie der Mengen **Set** ab. Man sagt zusammen mit den beiden Kategorien und den beiden Funktoren bildet die natürliche Äquivalenz eine *Adjunktion*. Wir bezeichnen F als *rechtsadjungiert* zu G und G *linksadjungiert* zu F .

Bemerkung. Eine spezielle Form einer Adjunktion ist der *Limes* Begriff, diesen werden wir später verwenden. Eine formale Definition dessen würde den Rahmen dieser Arbeit überschreiten. Wir verweisen auf [4]. Wichtig für weitere Überlegungen ist eine anschauliche Betrachtung des Limes zum verknüpfen kleinerer Strukturen zu einer Großen.

Die Anschauung einer Adjunktion als leicht abgeschwächte Form der Äquivalenz zweier Kategorien wird für weitere Überlegungen des UMAP Verfahren eine wichtige Rolle spielen.

Wir werden nun das *Yoneda Lemma* nennen. Dieses wird im Allgemeinen dazu genutzt die Struktur der Kategorie **Set** und einfache Begriffe welche auf dieser gegeben sind auf beliebige Kategorien zu übertragen. Das Lemma wird angewendet um die Yoneda Einbettung zu geben. In dieser Arbeit werden wir es benötigen um Aussagen über simpliziale Mengen zu zeigen. Außerdem wird es verwendet um Satz 3.2 zu beweisen.

Yoneda Lemma. Sei \mathcal{C} eine Kategorie, $F : \mathcal{C} \rightarrow \mathbf{Set}$ ein Funktor, dann ist

$$\begin{aligned} \Phi : \text{Nat}(\text{Hom}(-, B), F) &\rightarrow F(A) \\ t &\mapsto t_A(id_A) \end{aligned} \quad (2.10)$$

ein Isomorphismus. Wobei $\text{Nat}(G, F)$ die Menge der natürlichen Transformationen von G nach F angibt. Somit ist $t_A : \text{Hom}_{\mathcal{C}}(A, B) \rightarrow F(A)$.

Bemerkung. Die Wohldefiniertheit von Φ ist sofort ersichtlich, die Injektivität und Surjektivität können mithilfe der Eigenschaften der natürlichen Transformationen nachgerechnet werden. Der Beweis lässt sich auch mit dem kovarianten Hom-Funktör durchführen.

Wir werden diese benutzen um Kategorien zu definieren, mit Objekten als Funktoren und den Morphismen die natürlichen Transformationen. Eine häufig verwendete Form eines kontravarianten Funktors ist die Prägarbe (*engl.: presheaf*). Wir werden später eine Prägarbe verwenden um *simpliziale Mengen* einzuführen.

Definition 2.14. Eine Prägarbe auf einer kleinen Kategorie \mathcal{C} ist ein Funktor

$$\mathcal{P} : \mathcal{C}^{op} \rightarrow \mathbf{Set}$$

von der dualen Kategorie \mathcal{C}^{op} von \mathcal{C} in die Kategorie **Set** von Mengen.

Definition 2.15. Sei $\widehat{\mathcal{C}}$ die Prägarbenkategorie einer Kategorie \mathcal{C} : Objekte sind Prägarben, also Funktoren $\mathcal{P} : \mathcal{C}^{op} \rightarrow \mathbf{Set}$, und Morphismen sind natürliche Transformationen der Funktoren.

Wir Kategorien eingeführt und einige Beispiele für diese gegeben. Mit der Definition eines Funktors können wir Zusammenhänge zwischen Kategorien betrachten und mit dem Begriff einer Adjunktion können wir *gute* Zusammenhänge beschreiben. Das Yoneda Lemma wird sich für Beweise als nützlich erweisen. Die Definition einer Prägarbe wird im nächsten Abschnitt bei der Definition simplizialer Mengen wichtig sein.

2.3 Simpliziale Mengen

Für die Konstruktion der topologischen Repräsentation der Daten werden wir *simpliziale Mengen* benötigen. Diese stellen eine Verallgemeinerung der in der TDA häufig verwendeten *Simplizialkomplexe* dar. Wir möchten den interessierten Leser an dieser Stelle auf die sehr verständlich und ausführlich gestalteten Notizen von Friedman [11] verweisen, dort wird der Unterschied zwischen diesen beiden Konstrukten sehr illustrativ erläutert.

Um simpliziale Mengen einzuführen werden wir die kategorientheoretische Definition verwenden. Dafür wird die *Simplexkategorie* benötigt. Zuerst werden wir der Vollständigkeit halber die Definition eines *geometrischen Simplex* geben.

Definition 2.16. Ein (*geometrischer*) n -*Simplex* ist eine von $n + 1$ geometrisch unabhängigen Punkten $\{v_0, \dots, v_n\}$ aufgespannte konvexe Hülle im euklidischen Raum. Die geometrische Unabhängigkeit der Vektoren bedeutet dabei, das $v_1 - v_0, \dots, v_n - v_0$ linear unabhängig sind. Die Vektoren v_i werden *Knoten* genannt und die von den Teilmengen von $\{v_0, \dots, v_n\}$ aufgespannten konvexen Hüllen, *Facetten* des n -Simplex.

Definition 2.17 (*Simplexkategorie*). Die Objekte der *Simplexkategorie* Δ sind die Mengen $[n] := \{0, 1, \dots, n\}$ für $n \in \mathbb{N}$, und Morphismen sind monoton wachsende Abbildungen. Dabei ist $[n] := \{1, \dots, n\}$.

Um eine Verbindung zwischen der Simplexkategorie und geometrischen n -Simplizes (siehe Abbildung 2.2) herzustellen, betrachtet man die n -elementigen Mengen in Δ und den Funktor $|\cdot| : \Delta \rightarrow \mathbf{Top}$, gegeben durch

$$|\cdot| : [n] \mapsto |\Delta^n| := \left\{ (t_0, \dots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^n t_i = 1, t_i \geq 0 \right\}$$

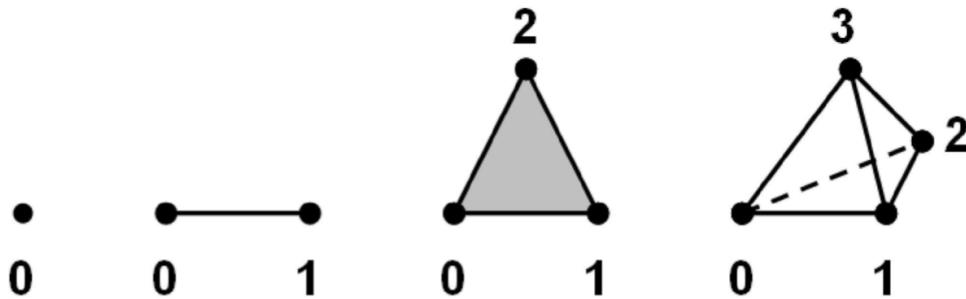


ABBILDUNG 2.2: Die geometrischen 0-, 1-, 2- und 3-Simplizes, in manchen Fällen werden diese auch als (geordnete) Standardsimplizes bezeichnet.

Das Bild von $[n]$ unter $|\cdot|$ ist somit ein geometrischer n -Simplex.

Definition 2.18. Eine *simpliziale Menge* ist ein Funktor $X : \Delta^{op} \rightarrow \mathbf{Set}$. Üblicherweise wird $X([n])$ als X_n geschrieben, und wir bezeichnen die Elemente $x \in X_n$ als n -Simplizes. Der n -dimensionale *Standardsimplex* ist

$$\Delta^n := \mathbf{Hom}(-, [n]).$$

Bemerkung. Simpliziale Mengen sind also Hom-Funktoren (siehe Gleichung (2.7)). Eine etwas verständlichere Definition einer simplizialen Menge werden wir in Definition 2.22 geben.

Definition 2.19. Die Objekte der Kategorie der simplizialen Mengen \mathbf{sSet} sind simpliziale Mengen und ihre Morphismen sind natürliche Transformationen.

Um ein besseres Verständnis für eine simpliziale Menge zu bekommen ist es oft sinnvoll an einen *Simplizialkomplex* zu denken.

Definition 2.20. Ein (*geometrischer*) *Simplizialkomplex* \mathcal{K} in \mathbb{R}^n ist eine Menge von (geometrischen) Simplizes, möglicherweise unterschiedlicher Dimension, in \mathbb{R}^n , so dass

1. jede Facette eines Simplex aus \mathcal{K} in \mathcal{K} ist, und
2. der Schnitt zweier Simplizes aus \mathcal{K} ist eine Facette beider Simplizes.

Bemerkung. Ein Simplizialkomplex ist anschaulich betrachtet ein geometrisches Objekt, welches aus mehreren Simplizes besteht. Die Simplizes dürfen dabei nur entlang ihrer Facetten *zusammengefügt* werden.

Simplizialkomplexe können dabei helfen topologische Räume zu beschreiben, die kombinatorische Struktur des Simplizialkomplexes kann dann dazu genutzt werden, Aussagen über den zugrundeliegenden topologischen Raum zu treffen. Dabei sind die genauen räumlichen Lagebeziehungen der Simplizes oft zu vernachlässigen und es kann folgende Verallgemeinerung gemacht werden:

Definition 2.21. Ein *abstraktes Simplex* σ ist eine nichtleere endliche Menge. Ein Element eines abstrakten Simplex σ nennen wir *Knoten*, eine nichtleere Teilmenge $\sigma' \subset \sigma$ ist wieder ein abstrakter Simplex und wird *Facette* von σ genannt. Die *Dimension* eines abstrakten Simplex σ welcher aus einer $k+1$ -Elementigen Menge besteht ist k , oft nennen wir σ dann einen (abstrakten) k -Simplex.

Ein *abstrakter Simplicialkomplex* \mathcal{K} ist eine Menge von Simplizes mit der Eigenschaft, dass jede Facette $\sigma' \subset \sigma$ eines Simplizes $\sigma \in \mathcal{K}$ wieder zu \mathcal{K} gehört, also $\sigma' \in \mathcal{K}$. Das k -Skelett eines Simplicialkomplexes \mathcal{K} ist die Menge aller seiner Simplizes der Dimension $\leq k$.

Abstrakte Simplicialkomplexe besitzen im Allgemeinen also keine Informationen über die relative räumliche Lage der Knoten, insbesondere können die Knoten beliebige Objekte sein. Ähnlich sind simpliziale Mengen zu verstehen, allerdings enthalten diese noch *mehr* Informationen über die Simplizes, siehe dazu [11].

Ähnlich zum abstrakten Simplicialkomplex können wir für eine simpliziale Menge $X \in \mathbf{sSet}$ die n -Simplizes als geometrische n -Simplizes auffassen. Wobei die niedrigdimensionalen Simplizes die Facetten sind und die 0-Simplizes geordnet werden können. Allerdings müssen die 0-Simplizes nicht nötigerweise unterschiedlich sein, diese Anschauung ist [35] entnommen und wird dort genauer ausgeführt. Eine äquivalente zu der Definition simpliziale Mengen als Prägarbe auf Δ zu betrachten ist wie folgt gegeben.

Definition 2.22. Eine *simpliziale Menge* X ist eine Familie von Mengen $X_n, n \geq 0$ zusammen mit Abbildungen $d_i : X_n \rightarrow X_{n-1}$ und $s_i : X_n \rightarrow X_{n+1}$ für alle $0 \leq i \leq n$, so dass folgende Bedingungen erfüllt sind,

$$\begin{aligned} d_i d_j &= d_{j-1} d_i, & i < j \\ s_i s_j &= s_{j+1} s_i, & i \leq j \\ d_i s_j &= \begin{cases} 1, & i = j, j + 1 \\ s_{j-1} d_i, & i < j \\ s_j d_{i-1}, & i > j + 1. \end{cases} \end{aligned} \tag{2.11}$$

Diese Definition verdeutlicht die Verbindung von simplizialen Mengen und Simplicialkomplexen.

Eine hilfreiche Eigenschaft (abstrakter) Simplicialkomplexe ist, dass sich diese aus den einfach zu beschreibenden geometrischen n -Simplizes zusammensetzen. Diese Eigenschaft lässt sich auf simpliziale Mengen mittels Yoneda Lemma übertragen. Sei X eine simpliziale Menge, dann gibt es für alle $x \in X_n$ einen Morphismus $x : \Delta^n \rightarrow X$. Eine Anwendung von [26] (Kap.7, Thm.1) liefert uns:

$$X \simeq \varinjlim \Delta^n, \tag{2.12}$$

wobei der Limes über eine von X bestimmte Indexkategorie genommen wird. Anschaulich werden die Objekte (hier die Δ^n) in einer passenden Weise *zusammengefügt*, dies bedeutet also, dass sich eine simpliziale Menge aus den Standardsimplizes zusammensetzt.

Bemerkung. Somit können wir analog zu Simplicialkomplexen das k -Skelett einer simplizialen Menge als Menge aller k -Simplizes und die Elemente des 0-Skeletts als Knoten bezeichnen.

Wie bereits erwähnt lassen sich für topologische Räume geeignete (abstrakte) Simplicialkomplexe konstruieren, ähnliches gilt auch für simpliziale Mengen. In der Tat gibt es aus kategorientheoretischer Sichtweise eine *gute* Beziehung zwischen simplizialen Mengen und topologischen Räumen, diese lässt sich durch zwei adjungierte Funktoren wie folgt beschreiben:

Satz 2.1. Die geometrische Realisierung gegeben durch:

$$|\cdot| : \mathbf{sSet} \rightarrow \mathbf{Top}, \quad |X| \mapsto \varinjlim |\Delta^n|, \quad (2.13)$$

und der singuläre Mengen Funktor

$$S : \mathbf{Top} \rightarrow \mathbf{sSet}, \text{ mit } S(Y) : [n] \rightarrow \mathbf{Hom}_{\mathbf{Top}}(|\Delta^n|, Y), \quad (2.14)$$

bilden eine Adjunktion.

Diese Adjunktion werden wir in Kapitel 3 für metrische Räume erweitern. Dabei erinnern wir daran, dass wir Adjunktionen als abgeschwächte Formen der Äquivalenz illustriert haben. Somit gibt es einen *guten* Zusammenhang simplizialer Mengen und topologischer Räume. Hier kann argumentiert werden, dass simpliziale Mengen aufgrund ihrer sehr flexiblen Struktur viele Räume konstruieren können. Umgekehrt ist die Definition eines topologischen Raumes nicht zu restriktiv, so dass selbst *ausgefältere* simpliziale Mengen durch topologische Räume modelliert werden können.

2.4 Unscharfe Mengen

Ein geometrischer Simplizialkomplex enthält Informationen über die Lage der Knoten, abstrakte Simplizialkomplexe und simpliziale Mengen fehlt diese Eigenschaft. In Kapitel 3 werden wir die hochdimensionalen Daten \mathbf{x}_i betrachten und für diese eine simpliziale Menge konstruieren. Dazu möchten wir auch die Eigenschaft nutzen, dass die betrachteten Daten X eine Metrik besitzen. Um simplizialen Mengen, genauer gesagt den Knoten, einen *Abstandsbegriff* zuzuordnen werden wir den Begriff der *unscharfen Menge* nutzen.

In der klassischen Mengentheorie ist die Zugehörigkeit eines Elementes x zu einer Menge X eine binäre Funktion. Entweder gilt $x \in X$ oder $x \notin X$. Eine *unscharfe Menge* verallgemeinert die Zugehörigkeit.

Definition 2.23. Sei A eine Menge und $\mu : A \rightarrow [0, 1]$. Wir nennen des Paar (A, μ) *unscharfe Menge* und μ die *Zugehörigkeitsfunktion*. Für $a \in A$ nennen wir $\mu(a)$ den *Zugehörigkeitsgrad von a* .

Bemerkung. Dabei wird das Bild von $a \in A$ unter μ als Wahrscheinlichkeit interpretiert, dass $a \in A$. Eine *klassische* Menge A ist ein Spezialfall einer unscharfen Menge, mit $\mu(a) = 1$ für alle $a \in A$.

Definition 2.24. Seien $(A, \mu), (B, \nu)$ unscharfe Mengen. Man sagt (A, μ) ist leer, falls μ die konstante Nullfunktion ist. Die *Gleichheit* von $(A, \mu), (B, \nu)$ ist gegeben, wenn $A = B$ und $\mu(a) = \nu(a)$ für alle $a \in A$.

Bemerkung. Die Elemente mit Zugehörigkeitsgrad 0 spielen also eine entschiedene Rolle bei der Angabe einer unscharfen Menge. Beispielsweise sind unscharfen Mengen $(\{1, 2\}, (\mu(1) = 1, \mu(2) = 0)), (\{1, 3\}, (\mu(1) = 1, \mu(3) = 0))$ nicht gleich.

Eine Verallgemeinerung der klassischen Mengenoperationen für unscharfe Mengen zu definieren benötigt die *t-Normen*.

Definition 2.25. Sei $\top : [0, 1] \times [0, 1] \rightarrow [0, 1]$. Dann ist \top eine *t-Norm*, wenn gilt,

- $\top(a, b) = \top(b, a)$,
- $a \leq b, c \leq d \Rightarrow \top(a, b) \leq \top(c, d)$,

- $\top(a, \top(b, c)) = \top(\top(a, b), c)$,
- $\top(1, a) = \top(a, 1) = 1$.

Eine monoton fallende Abbildung $\neg : [0, 1] \rightarrow [0, 1]$ mit $\neg(0) = 1, \neg(1) = 0$ wird *Negator* genannt. Die Abbildung \perp , mit $\perp(a, b) = \neg\top(\neg a, \neg b)$ wird als *zu \top über \neg zugehörige t-Conorm* bezeichnet.

Bemerkung. Analog zur anderen Binäroperationen verwendet man oft $\top(a, b) = a \top b$ und $\perp(a, b) = a \perp b$.

Definition 2.26. Das Tripel (\top, \perp, \neg) heißt *De-Morgan-Triplett*, wenn \top eine t-Norm, \neg ein Negator und \perp die zu \top über \neg zugehörige t-Conorm ist.

Beispiel 2.5. Die Abbildungen $\top(a, b) := ab, \perp(a, b) := a + b - ab, \neg(a) := 1 - a$ bilden ein De-Morgan-Triplett.

Nun können wir die Definition der Mengenoperationen aus der Fuzzy-Logik geben.

Definition 2.27. Sei (\top, \perp, \neg) ein De-Morgan-Triplett und $(A, \mu), (A, \nu)$ unscharfe Mengen. Dann ist

- $(A, \mu \cap \nu)$, mit $(\mu \cap \nu)(a) := \top(\mu(a), \nu(a)), a \in A$, der *Schnitt von (A, μ) und (A, ν)* ,
- $(A, \mu \cup \nu)$, mit $(\mu \cup \nu)(a) := \perp(\mu(a), \nu(a)), a \in A$, die *Vereinigung von (A, μ) und (A, ν)* ,
- $(A, \neg \circ \mu)$, das *Komplement von (A, μ)* .

Bemerkung. Es gilt $\neg((A, \mu \cap \nu)) = (A, (\neg \circ \mu) \cup (\neg \circ \nu))$ und $\neg((A, \mu \cup \nu)) = (A, (\neg \circ \mu) \cap (\neg \circ \nu))$. Die definierten Mengenoperationen auf unscharfen Mengen erfüllen also die De-Morgan Regeln der klassischen Mengenlehre.

Um die Theorie des UMAP Verfahrens in Kapitel 3 zu formal beschreiben zu können, werden wir die Kategorie unscharfer simplizialer Mengen benötigen. Seit der Einführung unscharfer Mengen in [42] gab es verschiedene Ansätze diese Kategorie zu definieren, unter anderem von Goguen [13]. Diese Definition hat laut [2] den Nachteil, dass zwei Morphismen unscharfer Mengen nicht gleich sind, wenn sie sich nur auf Elementen mit Zugehörigkeitsgrad 0 unterscheiden. Da wir Elemente mit Zugehörigkeitsgrad 0 als Elemente interpretieren möchten, welche nicht zur Menge gehören, ist Goguens Definition problematisch.

Die für das UMAP Verfahren verwendete Definition richtet sich nach [2].

Definition 2.28. Sei $I = ((0, 1] \subset \mathbb{R}, \tau = \{(0, a), a \in (0, 1]\})$ ein topologischer Raum und $\mathcal{O}_{\sqrt{I}}(I)$ die Kategorie der offenen Teilmengen, siehe Beispiel 2.4. Eine *unscharfe Menge* ist eine Prägarbe $\mathcal{P} : \mathcal{O}_{\sqrt{I}}(I)^{op} \rightarrow \mathbf{Set}$, so dass alle Morphismen $\mathcal{P}(a \leq b)$ Injektionen sind.

Dabei können wir die Menge $\mathcal{P}((0, a))$ als Menge aller Elemente mit Zugehörigkeitsgrad größer als a betrachten. Die liefert uns die Kategorie der unscharfen simplizialen Mengen,

Definition 2.29. Die Kategorie **Fuzz** der unscharfen Mengen hat als Objekte die Prägarben auf $\mathbf{Op}(I)$ welche unscharfe Mengen sind und als Morphismen die Morphismen von $\mathbf{Op}(I)$. Eine unscharfe simpliziale Menge X ist gegeben durch $X : (\Delta \times \mathbf{Op}(I)) \rightarrow \mathbf{Set}$. Dabei schreiben wir $X([n], (0, 1)) = \Delta_{<a}^n$. Die Kategorie der unscharfen simplizialen Mengen **sFuzz** hat als Objekte unscharfe simpliziale Mengen und als Morphismen natürliche Transformationen.

Wir können die beiden Definitionen unscharfer Mengen miteinander in Verbindung bringen, indem wir eine Prägarbe \mathcal{P} auf $\mathbf{Op}(I)$ betrachten. Dann erhalten wir durch $A = \bigcup_{a \in (0, 1]} (\mathcal{P}((0, a)))$ und $\mu(x) = \sup\{a \in (0, 1] | x \in \mathcal{P}((0, a))\}$.

Um dieses Kapitel abzuschließen werden wir die Kreuzentropie zwischen zwei unscharfen Mengen einführen. Diese soll zum vergleichen der unscharfen simplizialen Mengen Darstellungen dienen. Seien $(A, \mu), (A, \nu)$ unscharfe Mengen mit gleicher Grundmenge A . Dann ist die Kreuzentropie gegeben durch,

$$C_{\text{cross}}((A, \mu), (A, \nu)) := \sum_{a \in A} \left(\mu(a) \log \left(\frac{\mu(a)}{\nu(a)} \right) + (1 - \mu(a)) \log \left(\frac{1 - \mu(a)}{1 - \nu(a)} \right) \right). \quad (2.15)$$

Kapitel 3

UMAP

In diesem Kapitel soll das UMAP Verfahren eingeführt werden. Dabei wird angenommen, dass die Daten $X = \{\mathbf{x}_i\}_{i=1}^N$, ($\mathbf{x}_i \in \mathbb{R}^D$) auf einer d -dimensionalen riemannschen Mannigfaltigkeit liegen.

Das UMAP Verfahren approximiert lokal die geodätische Distanz der \mathbf{x}_i . Dies führt dazu, dass wir für jeden Datenpunkt \mathbf{x}_i einen metrischen Raum X_i erhalten. Diese Konstruktion wird in Abschnitt 3.1 beschrieben.

Da die Metriken der X_i a priori nicht miteinander kompatibel sind, wird in Abschnitt 3.2 die Adjunktion aus Satz 2.1 auf metrische Räume und unscharfe simpliziale Mengen erweitert. Diese wird dazu genutzt die X_i als unscharfe simpliziale Mengen darzustellen. Vereinigen wir die Mengen, erhalten wir eine topologische Darstellung der hochdimensionalen Daten. Aufgrund der konstruierten Metriken enthält diese lokale und aufgrund der unscharfen simplizialen Mengen globale Eigenschaften der Daten.

Um die Daten in den \mathbb{R}^d einzubetten und somit zu einer niedrigdimensionalen Darstellung Y zu gelangen, wird in Abschnitt 3.3 ebenfalls eine topologische Repräsentation vom \mathbb{R}^d konstruiert. Die Angabe einer Funktion, welche den Unterschied der beiden Repräsentationen darstellt, ermöglicht uns dann die Repräsentation vom \mathbb{R}^d so zu optimieren, dass sie der Repräsentation von X möglichst ähnlich ist, somit erhalten wir eine d -dimensionale Darstellung Y der Daten, welche mittels eines geeigneten Funktors in einen metrischen Raum überführt werden kann.

Wir werden uns in diesem Kapitel nach der in McInnes et. al. [27] gewählten Beschreibung des UMAP Verfahrens richten und diese insbesondere durch intuitive Erklärungen ergänzen.

3.1 Approximation der Mannigfaltigkeit

Wir nehmen nun an, dass (\mathcal{M}, g) die d -dimensionale riemannsche Mannigfaltigkeit ist, auf welcher unsere Daten X liegen, also $X \subseteq \mathcal{M}$. Für den Fall, dass die Mannigfaltigkeit nicht bekannt ist, möchten wir nun die Geodäten auf \mathcal{M} , und damit zwischen je zwei Datenpunkten auf X , approximieren. Dazu nutzen wir folgendes Lemma:

Lemma 3.1. *Sei $p \in \mathcal{M}$ ein Punkt. Wenn*

1. *g lokal konstant auf einer offenen Umgebung U von p ist, so dass g eine Diagonalmatrix bezüglich der Umgebungskoordinaten ist,*
2. *$B_r(p) \subseteq U$ ein Ball mit Volumen $\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$ bezüglich g ist,*

dann ist die Geodäte von p zu jedem q aus $B_r(p)$ durch $\frac{1}{r}d_{\mathbb{R}^n}(p, q)$ gegeben. Dabei ist $d_{\mathbb{R}^n}$ die Metrik des Umgebungsraumes von \mathcal{M} und r der Radius von B bezüglich des Umgebungsraumes.

Bemerkung. Ein Beweis des Lemmas findet sich in [27]. Die Idee lässt sich wie folgt skizzieren. Die Determinante von g kann explizit angegeben werden, da das Volumen des Balls gegeben ist. Da g zusätzlich eine Diagonalmatrix ist lässt sich g in diesem Fall eindeutig aus der Determinante bestimmen. Die explizite Form von g ermöglicht es uns die Geodäte zwischen p und q berechnen.

Wir möchten nun argumentieren, dass die beiden Bedingungen aus Lemma 3.1 für unsere Daten erfüllt sind. Die erste Bedingung ist erfüllt, falls wir annehmen, dass die Datenpunkte \mathbf{x}_i gleichverteilt bezüglich g auf \mathcal{M} liegen. Betrachten wir einen Ball B_r auf (\mathcal{M}, g) , wobei r so gewählt ist, dass B_r genau k , ($k \in \mathbb{N}$) Elemente aus X enthält. Da die \mathbf{x}_i gleichverteilt bezüglich g sind liegen in jedem B'_r ebenfalls k Elemente aus X . Ein Ball $B(\mathbf{x}_i)$ welcher die k -nächsten-Nachbarn von \mathbf{x}_i enthält hat somit ein festes Volumen. Wir skalieren g mit der inversen Distanz zum k -ten Nachbarn, dann gilt für das Volumen von B , $V(B(\mathbf{x}_i)) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$, somit ist auch die zweite Bedingung aus Lemma 3.1 für unsere Daten X erfüllt.

Bemerkung. Dabei ist der j -te Nachbar von \mathbf{x}_i bzgl. d gegeben durch \mathbf{x}_{i_j} , so dass $d(\mathbf{x}_i, \mathbf{x}_{i_1}) \leq \dots \leq d(\mathbf{x}_i, \mathbf{x}_{i_j}) \leq \dots \leq d(\mathbf{x}_i, \mathbf{x}_{i_N})$. Die k -nächsten-Nachbarn eines Punktes sind somit die $1, \dots, k$ -ten-Nachbarn.

Wir können nun für jedes \mathbf{x}_i einen metrischen Raum X_i definieren, so dass die Distanz zu den k -nächsten-Nachbarn die Geodäte auf der riemannschen Mannigfaltigkeit ist. Sei d die zu unseren Daten gehörende Metrik. Dann definieren wir für $\mathbf{x}_i \in X$ den metrischen Raum (X, \tilde{d}_i) mit

$$\tilde{d}_i(x, y) := \frac{d(x, y)}{d(\mathbf{x}_i, \mathbf{x}_{i_k})}, \quad (3.1)$$

Diese Definition der d_i ist für den Kontext nicht sinnvoll, da für \mathbf{x}_i mit h -ten Nachbarn \mathbf{x}_h und j -ten Nachbarn \mathbf{x}_j , nach Lemma 3.1 \tilde{d}_i nur für die Paare $(\mathbf{x}_i, \mathbf{x}_h), (\mathbf{x}_i, \mathbf{x}_j)$, mit $h, j \leq k$, die Geodäte angibt. Wir setzen,

$$\bar{d}_i(x, y) := \begin{cases} \tilde{d}_i(x, y), & \text{falls } x = \mathbf{x}_i \vee y = \mathbf{x}_i, \\ \infty, & \text{sonst.} \end{cases} \quad (3.2)$$

Somit sind die \bar{d}_i erweiterte Metriken.

Eine bekannte Problematik, wenn man hochdimensionale Daten betrachtet ist der *Fluch der Dimensionen*. Dieses Phänomen beschreibt die Effekte der Volumenvergrößerung in hochdimensionalen Räumen. Um zwei Auswirkungen auf paarweise Distanzen zu beschreiben, betrachten wir die paarweisen Distanzen randomisierter gleichverteilter Punkte in n -dimensionalen euklidischen Räumen, siehe Abbildung 3.1. Die liefert uns (1) mit zunehmender Größe der Dimension erhöhen sich die paarweisen Distanzen, (2) dass die paarweisen Distanzen sind ungefähr normalverteilt, wobei die Varianz der Normalverteilung für höhere Dimensionen abnimmt. Dadurch sind die Distanzen eines Punktes zu seinen ersten, zweiten, \dots , k -ten Nachbarn im hochdimensionalen Raum annähernd gleich. Für eine genauere Analyse der Auswirkungen hochdimensionaler Räume auf die nächsten Nachbarn siehe [3].

Unter anderem kann man dem *Fluch entfliehen*, in dem die Distanzen mit der Distanz zum ersten Nachbarn subtrahiert werden, somit werden die relativen Distanzen zwischen den Nachbarn eines Punktes \mathbf{x}_i vergrößert. Dies wenden wir auf unsere erweiterten Metriken \bar{d}_i an und erhalten erweiterte Pseudometriken,

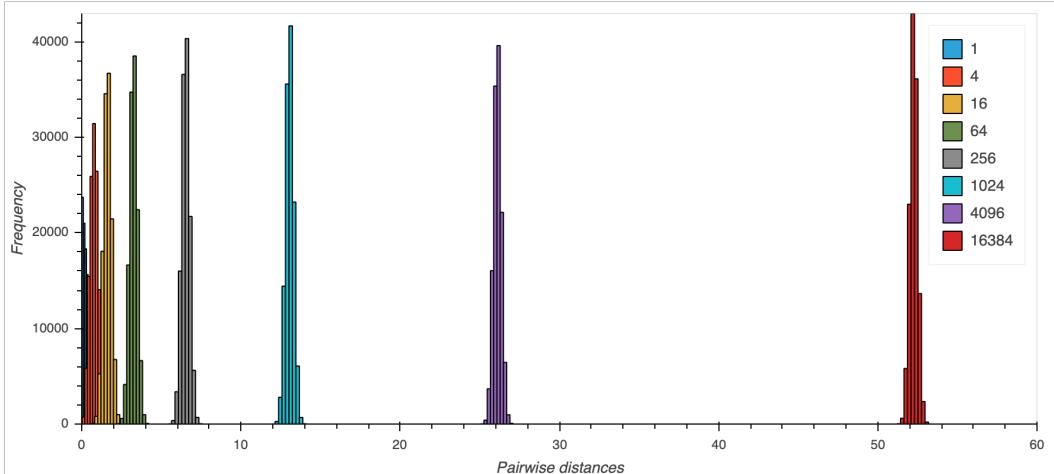


ABBILDUNG 3.1: Paarweise Distanzen von $N = 500$ zufällig gleichverteilten Punkten im R^D .

$$d_i(\mathbf{x}_i, \mathbf{x}_j) := \max(0, \bar{d}_i(\mathbf{x}_i, \mathbf{x}_j) - \bar{d}_i(\mathbf{x}_i, \mathbf{x}_{i_1})). \quad (3.3)$$

Bemerkung. Wir nehmen an, dass unsere Daten X keine Duplikate enthalten. Diese Annahme ist gerechtfertigt, da wir primär Aussagen über die Beziehung zwischen den Datenpunkten treffen möchten. Der erste Nachbar ist also ein *echter Nachbar*, mit $\bar{d}_i(\mathbf{x}_i, \mathbf{x}_{i_1}) > 0$.

Bemerkung. Für den Fall, das die Metrik der zugrundeliegenden Mannigfaltigkeit d_M bekannt ist, setzen wir in Gleichung (3.1) $\tilde{d}_i := d_M$ und wenden die Modifikationen aus Gleichungen (3.2) und (3.3) an um d_i zu erhalten.

Die erweiterten Pseudometriken d_i liefern uns lokal die Geodäte, welche hilfreich ist die zugrundeliegende Mannigfaltigkeit zu beschreiben. Allerdings sind die Metriken nicht zwingend miteinander kompatibel. Eine Lösung für die Inkompatibilität der Metriken werden wir im folgenden Abschnitt beschreiben.

3.2 Topologische Repräsentation

In Satz 2.1 haben wir gesehen, dass es eine Adjunktion zwischen topologischen Räumen und simplizialen Mengen gibt. Wir könnten die in Gleichung (3.3) definierten Metriken als topologische Räume mit $\{(X, \tau_i)\}_{1 \leq i \leq N}$ und der von d_i induzierten Topologie τ_i auffassen, diese mittels singuläre Mengen Funktors in simpliziale Mengen überführen und die Mengen Vereinigen. Durch diese Konstruktion würden uns wichtige Informationen verloren gehen. Um dies zu vermeiden, werden wir eine Adjunktion zwischen der Kategorie der erweiterten pseudo-metrischen Räume **EPMet** und der Kategorie der unscharfen simplizialen Mengen **sFuzz** konstruieren.

Bemerkung. Da wir nur endliche Datensätze betrachten, werden wir uns auf die *Unterkategorien* der endlichen erweiterten pseudo-metrischen Räume **Fin-EPMet** und endlichen unscharfen simpliziale Mengen **Fin-sFuzz** beschränken. Eine Unterkategorie besteht aus Teilmengen der Objekte und Morphismen der zugehörigen Kategorie.

Definition 3.1. Der Funktor $\text{FinReal} : \text{Fin-sFuzz} \rightarrow \text{Fin-EPMet}$ ist gegeben durch

$$\text{FinReal}(X) := \varinjlim \text{FinReal}(\Delta_{\leq a}^n), \quad (3.4)$$

wobei,

$$\text{FinReal}(\Delta_{\leq a}^n) := (\{x_1, \dots, x_n\}, d_a), \quad (3.5)$$

$$d_a(x_i, x_j) := \begin{cases} 0 & , \text{ falls } i = j \\ -\log(a) & , \text{ sonst.} \end{cases} \quad (3.6)$$

Die Wirkung des Funktors **FinReal** auf einem Morphismus $\Delta_{\leq a}^n \rightarrow \Delta_{\leq b}^m$, mit $a \leq b$ und $\sigma : \Delta^n \rightarrow \Delta^m$, ist gegeben durch $(\{x_1, \dots, x_n\}, d_a) \mapsto (\{x_{\sigma(1)}, \dots, x_{\sigma(n)}\}, d_b)$.

Bemerkung. Der Funktor ist wohldefiniert, da aus $a \leq b, d_a \geq d_b$ folgt. Somit ist die Wirkung von **FinReal** auf einem Morphismus von **Fin-sFuzz** nichtexpansiv und somit ein Morphismus von **Fin-EPMet**.

Satz 3.2. *Die Funktoren **FinReal** und **FinSing** : **Fin-sFuzz** \rightarrow **Fin-EPMet**, wobei für $Y \in \mathbf{Fin-EPMet}$ gilt,*

$$\text{FinSing}(Y) : ([n], [0, a)) \rightarrow \text{Hom}_{\mathbf{Fin-EPMet}}(\text{FinReal}(\Delta_{\leq a}^n), Y), \quad (3.7)$$

sind zueinander adjungiert.

Bemerkung. Ein Beweis findet sich in [27]. Die wesentliche Idee ist dabei, dass Funktoren welche Limiten erhalten einen rechts adjungierten Funktor besitzen, nach Konstruktion erhält **FinReal** Limiten. Zusätzlich wird für den Beweis das Yoneda Lemma und Gleichung (2.12) verwendet.

Die konstruierte Adjunktion ermöglicht es uns nun die erweiterten pseudo-metrischen Räume $\{(X, d_i)\}_{1 \leq i \leq N}$, mit d_i aus Gleichung (3.3), mittels des **FinSing** Funktors als unscharfe simpliziale Mengen darzustellen. Diese verknüpfen wir mittels t-Conorm und erhalten die *unscharfe topologische Repräsentation* des Datensatzes X

$$\mathcal{X} := \perp_{i=1}^N \text{FinSing}((X, d_i)). \quad (3.8)$$

Intuition der Repräsentation

Der folgende Absatz soll die Konstruktion der unscharfen topologischen Repräsentation intuitiv erklären. Dabei werden wir an einigen Stellen mathematische Strenge gegen eine illustrative Herangehensweise eintauschen.

Lokale Eigenschaften der Daten werden dadurch erhalten, dass wir mit Lemma 3.1 die Geodäten auf der X zugrundeliegenden Mannigfaltigkeit bestimmt haben. Für die erweiterten pseudo-metrischen Räume, wird dann eine geeignete unscharfe simpliziale Menge konstruiert. In Abschnitt 2.3 haben wir argumentiert, dass es intuitiv oft genügt anstelle einer simplizialen Menge einen Simplizialkomplex zu betrachten. Deshalb können wir uns den Funktor **FinSing** als *Abbildung* des erweiterten pseudo-metrischen Raumes X_i auf einen unscharfen Simplizialkomplex \mathcal{K}_i vorstellen, wobei jeder Simplex einen Zugehörigkeitsgrad hat.

Betrachten wir nun das 1-Skelett von \mathcal{K}_i , so sind die 0-Simplizes die $\mathbf{x}_j \in X$ mit Zugehörigkeitsgrad 1. Die 1-Simplizes beschreiben Abstände zwischen den \mathbf{x}_j ,

wobei der Zugehörigkeitsgrad eines 1-Simplizes aus \mathcal{K}_i , mit Facetten $\mathbf{x}_j, \mathbf{x}_l$, gerade $\exp(-d_i(\mathbf{x}_j, \mathbf{x}_l))$ entspricht. Die Wahl der Transformation der d_i mit $\exp(-d_i)$ um den Zugehörigkeitsgrad des 1-Simplex zu erhalten ist aus Gleichung (3.6) ersichtlich. Der Zugehörigkeitsgrad des ersten Nachbarn von \mathbf{x}_i in \mathcal{K}_i ist also stets 1 und nimmt für weiter entfernte Nachbarn exponentiell ab. Die Repräsentation erhält also metrische Informationen der \mathbf{x}_i , bevorzugt dabei allerdings stark die lokalen Abstände. 2-Simplizes würden Eigenschaften über die Fläche zwischen drei Punkten erhalten, 3-Simplizes über das Volumen welches von 4 Punkten eingeschlossen ist. Simplizes höherer Ordnung die entsprechenden Analogon für höher dimensionale Räume.

Wir vereinigen die \mathcal{K}_i , siehe Gleichung (3.8), indem wir unterschiedliche Zugehörigkeitsgrade der unscharfen simplizialen Mengen mittels t-Conorm vereinheitlichen.

Wir werden später, in Kapitel 4, auf diese Interpretation der 1-Simplizes zurückkommen. Doch zuerst möchten wir noch beschreiben, wie die unscharfe topologische Repräsentation \mathcal{X} im UMAP Verfahren genutzt wird, um eine niedrigdimensionale Darstellung der Daten zu finden.

3.3 Einbettung der Repräsentation

Die Theorie des UMAP Verfahren ist nun fast vollständig. Zu jedem $\mathbf{x}_i \in X$ konstruieren wir einen erweiterten pseudo-metrischen Raum mittels der in Gleichung (3.3) definierten Pseudometrik, welchen wir als unscharfe simpliziale Menge auffassen und mit den Repräsentationen der anderen $\mathbf{x}_j \in X$ vereinigen, diese Repräsentation bezeichnen wir mit \mathcal{X} .

Die Mannigfaltigkeit, in welcher wir unsere Daten einbetten möchten ist typischerweise der euklidische \mathbb{R}^d . Deswegen beschränken wir uns nun auf diesen Fall, alternativ könnten wir hier allerdings auch andere erweiterte pseudo-metrische Räume wählen. Um eine Repräsentation des \mathbb{R}^d zu konstruieren, wählen wir N gleichverteilte Punkte im \mathbb{R}^d und bezeichnen diese mit $Y = \{\mathbf{y}_i\}_{1 \leq i \leq N}$. In Kapitel 4 werden wir eine Alternative zur randomisierten Wahl von Y betrachten.

Da unsere Mannigfaltigkeit der \mathbb{R}^d ist können wir direkt die unsichere simpliziale Menge betrachten,

$$\mathcal{Y} := \text{FinSing}((Y, \|\cdot\|_2)). \quad (3.9)$$

Es ist zu beachten, dass für die j -Simplizes $\mathcal{Y}_j = (A_{\mathcal{Y}_j}, \nu_j)$ von \mathcal{Y} gilt, dass $A_{\mathcal{Y}_j} = A_{\mathcal{X}_j}$, mit $\mathcal{X}_i = (A_{\mathcal{X}_i}, \mu_i)$. Somit sind die Zugehörigkeitsfunktionen μ_j und ν_j für weitere Überlegungen entscheidend. Um die beiden topologischen Repräsentationen \mathcal{X}, \mathcal{Y} zu vergleichen verwenden wir die Kreuzentropie C_{cross} unscharfer Mengen. Dabei nutzen wir die Definition simplizialer Mengen als Folge von Mengen mit Kompositionen der Abbildungen d_i und s_i . Somit erhalten wir

$$C_N(\mathcal{X}, \mathcal{Y}) := \sum_{i=1}^N \lambda_i C_{\text{cross}}(\mathcal{X}_i, \mathcal{Y}_i), \quad (3.10)$$

mit $\lambda_i \in \mathbb{R}$.

Bemerkung. \mathcal{X}_i und \mathcal{Y}_i bezeichnen die unscharfen i -Simplizes der unscharfen topologischen Repräsentationen \mathcal{X} bzw. \mathcal{Y} . Wir betrachten nur Simplizes der Dimension $\leq N$, da die uns gegebenen Daten X aus N Datenpunkten bestehen.

Die Wahl der Kreuzentropie zwischen den n -Simplizes von \mathcal{X} und \mathcal{Y} , kann dabei wie folgt argumentiert werden. Man betrachtet $\mathcal{X}_i = (A_i, \mu_i)$, $\mathcal{Y}_i = (A_i, \nu_i)$, wobei A_i die zugrundeliegenden Simplizes von \mathcal{X}_i beschreibt und μ_i bzw. ν_i die Zugehörigkeitsfunktion der i -Simplizes beschreiben. Dann können wir, nach Normalisierung, μ_i, ν_i als Wahrscheinlichkeitsverteilungen auf A betrachten. Die Kreuzentropie gibt dann ein Maß für die *Unterschiedlichkeit* von μ_i und ν_i an.

Somit erhalten wir eine bezüglich der \mathbf{x}_i optimierte niedrigdimensionale Darstellung der \mathbf{y}_i indem wir Gleichung (3.10) nach \mathcal{Y} , genauer gesagt nach den ν_i minimieren. Die optimierte Repräsentation von \mathcal{Y} können wir mittels modifiziertem Realisierungsfunktor **FinReal** als metrischen Raum auffassen.

Das nächste Kapitel wird eine leicht angepasste numerische Formulierung des Problems liefern.

Kapitel 4

Implementierung

In diesem Kapitel möchten wir die Implementierung des UMAP Verfahren beschreiben. Ideal-typischerweise würden wir dafür die metrischen Räume konstruieren und diese mittels des modifizierten Singulären Mengen Funktors mit der in Gleichung (3.8) beschriebenen unscharfen topologischen Repräsentation darstellen. Praktikabel ist dies nicht für große Datensätze umsetzbar, da wir für die Berechnung der unscharfen simplizialen Mengen alle 2^N Teilmengen unseres N -elementigen Datensatzes betrachten müssten.

In aktuellen Implementierungen werden hingegen nur alle zweielementigen Teilmengen betrachtet. Wir werden in Kapitel 6 Ansätze erwähnen um Simplizes höherer Ordnungen effizient zu finden.

Zunächst werden wir in Abschnitt 4.1 die Notation anpassen und die Formulierung der Optimierung konkretisieren. Wir werden dann mittels Profiling, in Abschnitt 4.2, die rechenintensiven Subroutinen der von uns verwendeten Implementierung [28] identifizieren. In den darauffolgenden Abschnitten werden wir die aufwendigen Routinen genauer betrachten, in dem wir in Abschnitt 4.3 das verwendete Verfahren zur Optimierung der Einbettung beschreiben und in Abschnitt 4.4 zwei Verfahren zur Berechnung der 1-Simplizes diskutieren. Abschnitt 4.5 gibt eine Beschreibung der Hyperparameter an.

4.1 Numerische Formulierung des UMAP Verfahrens

Um weitere Überlegungen zu vereinfachen und die praktische Implementierung zu beschreiben, werden wir nun die Notation anpassen.

Dabei ist zu bemerken, dass wir uns von nun an auf die Konstruktion des 1-Skeletts der unscharfen simplizialen Mengen beschränken. Die Vereinfachung, dass wir keine Simplizes höherer Ordnungen betrachten werden wir in Abschnitt 6.2 betrachten und Ansätze erwähnen, um Simplizes höherer Ordnungen zu nutzen.

Wir werden von nun an die 0-Simplizes einer (unscharfen) Menge als Knoten, die 1-Simplizes als Kanten, das 1-Skelett als Graph und den Zugehörigkeitsgrad einer Kante als Gewicht der Kante bezeichnen.

Sei \mathcal{X} die in Gleichung (3.8) definierte topologische Repräsentation der Daten aus X und sei $k \in \mathbb{N}$ gegeben.

Das 1-Skelett von \mathcal{X} lässt sich durch einen Graph \mathcal{X}_G beschreiben. Dafür benötigen wir einen Zwischenschritt, in welchem wir N Graphen \mathcal{X}_{G_i} konstruieren, diese beschreiben das Bild $\text{FinSing}((X, d_i))$. Dabei sind die Gewichte von \mathcal{X}_{G_i} durch,

$$w_{\mathcal{X}_i}(x, y) := \begin{cases} \exp\left(\frac{-\max(0, d(x, y) - \rho_i)}{\sigma_i}\right) & , \text{falls } x = \mathbf{x}_i, y = \mathbf{x}_{i_j} \\ 0 & , \text{sonst} \end{cases} \quad (4.1)$$

mit $1 \leq j \leq k$, gegeben. \mathbf{x}_{i_j} ist wieder der j -te Nachbar von \mathbf{x}_i , ρ_i ist die Distanz von \mathbf{x}_i zum ersten Nachbarn. In Kapitel 3 wurde σ_i als Distanz zum k -ten Nachbarn gewählt. In der Praxis werden wir diese Wahl leicht modifizieren, indem wir σ_i so wählen, dass folgende Gleichung erfüllt ist:

$$\sum_{j=1}^k \exp\left(\frac{-\max(0, d(\mathbf{x}_i, \mathbf{x}_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k) \quad (4.2)$$

Der gewichtete Graph \mathcal{X}_{G_i} besitzt also die Knotenmenge $V(\mathcal{X}_{G_i}) = X$ und die Kantenmenge $E(\mathcal{X}_{G_i}) = (X \times X)$ mit Kantengewichten $w_{\mathcal{X}_i}$.

Die Vereinigung der \mathcal{X}_{G_i} zu \mathcal{X}_G lässt sich als Anwendung der t-Conorm auf die Gewichte beschreiben. Somit erhalten wir, für $x, y \in X$ eine neue Gewichtsfunktion $w_{\mathcal{X}}$, mit

$$w_{\mathcal{X}}(x, y) := w_{\mathcal{X}_1}(x, y) \perp w_{\mathcal{X}_2}(x, y) \perp \cdots \perp w_{\mathcal{X}_N}(x, y). \quad (4.3)$$

Um in Abschnitt 3.3 die Wahl der Kreuzentropie zu begründen, haben wir argumentiert, dass man die Zugehörigkeitsfunktionen der Simplizes als Wahrscheinlichkeitsverteilung auf den Simplizes betrachten kann. Diese Argumentation kann man auch anwenden um die Wahl der t-Conorm, aus Beispiel 2.5, gegeben durch $\perp(a, b) := a + b - ab$, zu begründen. Diese entspricht nämlich genau der Wahrscheinlichkeit der Vereinigung unabhängiger Ereignisse.

Wir werden nun \mathcal{Y}_G aus Gleichung (3.9) konstruieren. Dafür setzen wir

$$\Psi(x, y) := \begin{cases} 1 & , \text{ falls } \|x - y\|_2 \leq \text{min-dist} \\ \exp(-(\|x - y\|_2 - \text{min-dist})) & , \text{ sonst.} \end{cases} \quad (4.4)$$

Die Wahl des Hyperparameter min-dist wird später diskutiert. Dieser dient dazu, eine ähnliche Transformation wie im Gleichung (3.3) durchzuführen. Um später die Kreuzentropie zwischen \mathcal{X}_G und \mathcal{Y}_G zu minimieren muss die Gewichtsfunktion von \mathcal{Y}_G differenzierbar sein. Offensichtlich ist Ψ bei min-dist nicht differenzierbar. Somit betrachten wir eine stetige Approximation von Ψ ,

$$w_{\mathcal{Y}}(x, y) := (1 + a\|x - y\|_2^{2b})^{-1}, \quad (4.5)$$

wobei a und b Hyperparameter sind. Die Wahl von a und b kann beispielsweise mittels der Methode der kleinsten Quadrate bezüglich Ψ optimiert werden.

Die Kreuzentropie lässt sich auf gewichtete Graphen übertragen, wenn wir die Gewichte als Zugehörigkeitsgrade interpretieren,

$$C_{\text{cross}}(\mathcal{X}_G, \mathcal{Y}_G) := \sum_{e \in X \times X} w_{\mathcal{X}}(e) \log \left(\frac{w_{\mathcal{X}}(e)}{w_{\mathcal{Y}}(e)} \right) + (1 - w_{\mathcal{X}}(e)) \log \left(\frac{1 - w_{\mathcal{X}}(e)}{1 - w_{\mathcal{Y}}(e)} \right) \quad (4.6)$$

$$= \sum_{e \in X \times X} w_{\mathcal{X}}(e) \log(w_{\mathcal{X}}(e)) + (1 - w_{\mathcal{X}}(e)) \log(1 - w_{\mathcal{X}}(e)) \quad (4.7)$$

$$- \sum_{e \in X \times X} w_{\mathcal{X}}(e) \log(w_{\mathcal{Y}}(e)) + (1 - w_{\mathcal{X}}(e)) \log(1 - w_{\mathcal{Y}}(e))$$

$$= C_{\mathcal{X}} - \sum_{e \in X \times X} w_{\mathcal{X}}(e) \log(w_{\mathcal{Y}}(e)) + (1 - w_{\mathcal{X}}(e)) \log(1 - w_{\mathcal{Y}}(e)) \quad (4.8)$$

Dabei ist $C_{\mathcal{X}}$ von \mathcal{Y} unabhängig und somit bei der Minimierung der Kreuzentropie nicht von Relevanz. Eine Modifizierung des stochastischen Gradienten Verfahrens zur Minimierung von Gleichung (4.8) werden wir in Abschnitt 4.3 betrachten.

Die Überlegungen können wir nun im Pseudo-Code zusammenfassen.

Algorithm 1 UMAP Algorithmus

```

1: function UMAP( $X, N, D, d, \text{min-dist}, \text{n-epochs}$ )
2:   for  $\mathbf{x}_i \in X$  do
3:      $knn(\mathbf{x}_i) \leftarrow k\text{-nächste-Nachbarn}(\mathbf{x}_i)$ 
4:      $graph(\mathbf{x}_i) \leftarrow (\mathcal{X}_{G_i}, w_{\mathcal{X}_i})$ 
5:      $\mathcal{X}_G \leftarrow \text{gewichtete Adjazenzmatrix}(\bigcup_{i=1}^N graph(\mathbf{x}_i))$ 
6:      $D \leftarrow \text{Grad-Matrix des Graphen } \mathcal{X}_G$ 
7:      $L \leftarrow D^{1/2}(D - \mathcal{X}_G)D^{1/2}$  → Symmetrische normalisierte Laplace-Matrix
8:      $evec \leftarrow \text{sortierte Eigenvektoren von } L$ 
9:      $Y \leftarrow evec[1, \dots, d+1]$ 
10:     $Y \leftarrow \text{OPTIMIERE EINBETTUNG}(Y, \text{min-dist}, \text{n-epochs})$  → siehe Alg. 2
11:   return  $Y$ 

```

In Abschnitt 4.4 werden zwei effiziente Verfahren für die k -nächste-Nachbarn Suche (Zeile 3) angeben.

Die in Zeile 4 beschriebene Form die Graphen der \mathbf{x}_i zu konstruieren scheint auf den ersten Blick sehr viel Speicher zu benötigen, nämlich $\mathcal{O}(N^3)$. Dabei wird anstatt N nur eine Adjazenzmatrix A benötigt, in A mit $a_{ij} = w_{\mathcal{X}_i}(\mathbf{x}_i, \mathbf{x}_j)$. Wir haben zuvor die Notation mit N Adjazenzmatrizen gewählt, um die Verbindung mit der Konstruktion des UMAP Verfahrens aus Kapitel 3 zu verdeutlichen. Das betrachten von A ermöglicht uns eine effiziente Implementierung der t-Conorm in Zeile 5, dabei gilt $\mathcal{X}_G = A + A^\top - A \circ A^\top$, wobei \circ das elementweise Produkt ist.

Die Zeilen 6 - 9 beschreiben eine Alternative Initialisierung von Y mit der spektralen Einbettung der symmetrischen Laplace-Matrix. Die Grad-Matrix D in Zeile 6 ist eine Diagonalmatrix, wobei $d_{ii} := \sum_{1 \leq j \leq N} w_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)$, ($1 \leq i \leq N$). Für den Spezialfall, einer ungewichteten Adjazenzmatrix, beschreibt d_{ii} also den Grad des Knoten i . Die in Zeile 7 beschriebene Matrix L nennt man symmetrische normalisierte Laplace-Matrix. Die Eigenwerte einer Adjazenzmatrix nennt man das *Spektrum des Graphen*. Dabei wählt man den Spektrale Einbettung der Laplace-Matrix, da diese eine diskrete Form des Laplace-Beltrami Operators ist [7]. Der Laplace-Beltrami Operator ist eine Erweiterung des Laplace Operators auf Riemannsche Mannigfaltigkeiten. Dabei

D	Laufzeit NN-Descent	Laufzeit der Optimierung
100	9%	75,3%
500	12%	73,8%
1000	14%	72,9%
5000	30,4%	58%
10000	44%	45,1%
50000	78,8%	14,8%

TABELLE 4.1: D beschreibt die Größe der Umgebungsdimension. Abhängig von D haben wir die Laufzeit des UMAP Verfahrens profiliert. Die zweite und dritte Spalte beziehen sich auf die relativen Laufzeiten des kNN Verfahrens und der Optimierung der Einbettung.

ist zu beachten, dass eine Lösung des Eigenwertproblems nur von der Größe der Einbettungsdimension abhängig ist. Da wird stets $d \ll D$ betrachtet, ist eine effiziente Implementierung mittels sukzessiver Eigenwertsuche möglich.

Eine Beschreibung der Optimierung in Zeile 10 liefern wir in Abschnitt 4.3.

4.2 Profiling

In Kapitel 5 werden wir genauer auf die tatsächliche Laufzeit des UMAP Algorithmus eingehen. An dieser Stelle möchten wir die rechenintensiven Subroutinen des Verfahrens ausmachen. Dafür haben wir den Python cProfiler verwendet, dieser misst die Laufzeit der aufgerufenen Funktionen. Um für verschiedene Umgebungsdimensionen vergleichbare Ergebnisse zu erhalten, haben wir $N = 10\,000$ Datenpunkte in 10 unterschiedlichen $D = [100, 500, 1000, 5000, 10000, 50000]$ -dimensionalen Gauß-verteilten Datenwolken gewählt. Diese Daten wurden dann in den zweidimensionalen Raum eingebettet.

Dabei ist uns aufgefallen, dass besonders der k-nächste-Nachbarn-Algorithmus und die Optimierung mittels stochastischem Gradienten Verfahren einen großen Teil der Laufzeit des Verfahrens beanspruchen. In Tabelle 4.1 sind die Ergebnisse der Profilierung zusammengefasst. Insbesondere scheint die k-nächste-Nachbarn Suche die Laufzeit des UMAP Verfahrens für hochdimensionale Daten stark zu beeinflussen. Wir werden beide rechenintensiven Subroutinen im folgenden betrachten und geeignete Verbesserungen diskutieren.

4.3 Gradientenverfahren

Um die Zielfunktion aus Gleichung (4.8) zu minimieren bietet sich die Wahl eines Gradientenverfahrens an, da eine differenzierbare Approximation des Zugehörigkeitsgrades (siehe Gleichung 4.5) gegeben ist.

In den vergangenen Jahren gab es viele Weiterentwicklungen, insbesondere bezüglich der Konvergenzgeschwindigkeit, von Gradientenverfahren. Diese werden unter anderem für das trainieren neuronaler Netzwerke bei der Backpropagation genutzt. In [22] werden verschiedene Implementierungen verglichen.

Für das UMAP Verfahren wird eine modifizierte Version des stochastischen Gradientenverfahrens angewendet. Diese möchten wir kurz beschreiben. Um den Rechenaufwand im Gradientenverfahren zu verringern, wird der Gradient in zwei Summanden aufgeteilt. Dabei wird folgende Beobachtung genutzt: Für Kanten $\{i, j\}$ mit einem hohen Zugehörigkeitsgrad ($w_{ij} \approx 1$) ist der Term $(1 - w_{ij}) \log(1 - w_{ij})$ aus Gleichung

(4.8) nahe Null, deshalb ist es sinnvoll nur den Gradienten des Terms $w_{\mathcal{X}} \log(w_{\mathcal{Y}})$ zu betrachten. Für Kanten mit $w_{\mathcal{X}} \approx 0$ sollte hingegen der Gradient des Terms $(1 - w_{\mathcal{X}}) \log(1 - w_{\mathcal{Y}})$ betrachtet werden. Die Idee dieser Optimierung stammt aus [29] und dort zum finden eines Kontext für ein gegebenes Wort verwendet. Es wird eine Beschleunigung der Optimierung um den Faktor $2x - 10x$ erzielt. Für das UMAP Verfahren wird deswegen folgende Implementierung vorgeschlagen:

Algorithm 2 Optimiere die Einbettung mittels modifiziertem SGD

```

1: function OPTIMIEREINBETTUNG( $Y, V, W, n\text{-epochs}$ )
2:    $\alpha \leftarrow 1.0$ 
3:   for  $e \leftarrow 1, \dots, n\text{-epochs}$  do
4:     for all  $\{\mathbf{y}_i, \mathbf{y}_j\}$  do
5:       if Random()  $\leq w_{\mathcal{X}}(\mathbf{y}_i, \mathbf{y}_j)$  then
6:          $\mathbf{y}_i \leftarrow \mathbf{y}_i + \alpha \cdot \nabla(\log(w_{\mathcal{Y}}))(\mathbf{y}_i, \mathbf{y}_j)$ 
7:         for  $l \leftarrow 1, \dots, n\text{-neg-samples}$  do
8:            $m \leftarrow \mathcal{U}\text{unif}((0, N))$ 
9:            $\mathbf{y}_i \leftarrow \mathbf{y}_i + \alpha \cdot \gamma \cdot \nabla(\log(1 - w_{\mathcal{Y}}))(\mathbf{y}_i, \mathbf{y}_m)$ 
10:       $\alpha \leftarrow 1.0 - e/n\text{-epochs}$ 
11:   return  $Y$ 

```

Die in Zeile 5 beschriebene Ziehung der Stichprobe dient dazu in Zeile 6 keine zusätzliche Multiplikation mit $w_{\mathcal{X}}$ zu machen. Diese Idee entstammt [38]. Dadurch wird der Gradient nicht durch den Wert von $w_{\mathcal{X}}$ beeinflusst.

Das in jedem Durchlauf des modifizierten SGD mehrere *negative samples* betrachtet werden geht auch [29] zurück. Im wesentlichen verhindert dies, dass sich die Vektoren der niedrigdimensionalen Darstellung häufen. Die dabei $(0, N)$ -gleichverteilt gezogene Stichprobe ist eine Modifizierung von [39]. Die Wahl des Hyperparameter $n\text{-neg-samples}$ soll laut [29] zwischen zwei und 20 liegen.

Der Gradient in Zeile 6 ist gegeben durch:

$$\nabla(\log(w_{\mathcal{Y}}))(\mathbf{y}_i, \mathbf{y}_j) = \frac{-2ab\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2(b-1)}}{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2)}(\mathbf{y}_i - \mathbf{y}_j) \quad (4.9)$$

und der Gradient in Zeile 9 durch:

$$\nabla(1 - \log(w_{\mathcal{Y}}))(\mathbf{y}_i, \mathbf{y}_j) = \frac{2b}{(\epsilon + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2)(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2b})}(\mathbf{y}_i - \mathbf{y}_j), \quad (4.10)$$

wobei der ϵ -Parameter eine Division mit Null vermeidet.

Dies vervollständigt die Angabe des UMAP Verfahrens in der Praxis. Die genannten Hyperparameter werden wir in Abschnitt 4.5 betrachten.

4.4 Nächste-Nachbarn-Klassifikation

Zum effizienten finden der 1-Simplizes der topologischen Repräsentation unserer Daten, benötigen wir einen k-nächste-Nachbarn-Algorithmus (kurz: *kNN-Algorithmus*).

Das Ergebnis eines kNN-Algorithmus wird meist in einem ungerichteten Graph – dem kNN-Graph – dargestellt, wobei die Knoten den Datenpunkten entsprechen und die Kanten den Nachbarschaftsbeziehungen, somit besitzt jeder Knoten Grad k.

Bei einer naiven Implementierung beträgt die Laufzeit $\mathcal{O}(N^2D)$ (wobei N die Anzahl der Datenpunkte und D die Dimension der Datenpunkte ist). Mit einer effizienten

Implementierung ist in der Praxis eine annähernd linear in N Laufzeit möglich. Die Herangehensweisen lassen sich nach [39] in drei Kategorien einteilen. (1) Baum basierte Verfahren auf Partitionen des Raumes, (2) Hashfunktionen auf lokalen Teilgebieten des Raumes (3) Nachbarschafts-Erkundungen.

Wir möchten nun zwei Verfahren vorstellen.

NN-Descent

Der NN-Descent Algorithmus [9] beruht auf dem Prinzip der Nachbarschafts-Erkundungen. Dabei wird ein initialer kNN-Graph iterativ verbessert, unter der Annahme, dass die Nachbarschaftsbeziehung transitiv ist, für zwei vorhandene Nachbarschaftspaares $(x, y), (y, z)$ also mit hoher Wahrscheinlichkeit auch ein Nachbarschaftspaar (x, z) im kNN-Graph existiert. Der initiale Graph im NN-Descent Verfahren wird dabei zufällig gewählt. Dies kann dazu führen, dass nur lokal optimale k-NN-Graphen gefunden werden. Dies könnte laut [12] dadurch verbessert werden, indem für die Initialisierung „random projection trees“, wie in [39], verwendet werden.

Ein Vorteil des NN-Descent Verfahren ist, dass kein globaler Index der verwaltet werden muss. Somit ist eine Anwendung auf großen Datensätzen möglich welche nicht komplett in den Arbeitsspeicher (RAM) des verwendeten Rechners geladen werden können.

Nachteil des NN-Descent Algorithmus ist die Speicherplatzkomplexität, diese ist durch $\mathcal{O}(N^2)$ beschränkt. Im wesentlichen ist dies dadurch begründet, dass paarweise die Ähnlichkeit, welche im Falle von UMAP durch die Metrik des Umgebungsraums gegeben ist, gespeichert wird. Aufgrund dessen, dass nur lokale Optima garantiert sind, ist das Ergebnis des NN-Descent Verfahren approximativ. In [27] wird jedoch argumentiert, dass dies wegen des Informationsverlust bei Dimensionsreduktionen kaum Auswirkungen auf die resultierende Einbettung hat.

FAISS

Die FAISS Bibliothek [17] nutzt die Architektur einer GPU aus. Dabei baut FAISS eine effiziente Datenstruktur, welche für die Vektoren die nächsten Nachbarn speichert. Somit ist eine sehr schnelle Implementierung für das aufstellen des k-NN-Graphen möglich.

Der RAM der meisten GPUs ist stark begrenzt. Um dennoch mit großen Datensätzen zu arbeiten werden komprimierte Darstellungen der Vektoren genutzt. Für FAISS werden *product quantization codes* genutzt. Die Idee dieser Repräsentation ist, dass der hochdimensionale Raum in das Kartesische Produkt niedrigdimensionaler Teilräume zerlegt wird. Ein Vektor im Suchraum wird dann durch kurze Sequenzen aus Indizes der Teilräume beschrieben. Die euklidische Distanz zweier Datenpunkte kann anhand der Sequenzen errechnet werden, für eine genauere Beschreibung siehe [15]. Somit kann eine Datenstruktur verwaltet werden, welche eine schnelle Berechnung der euklidischen Distanzen ermöglicht.

Vorteile der FAISS Datenstruktur sind die effiziente Implementierung auf GPUs und das sowohl exakte Ergebnisse sowie Approximationen für die nächsten Nachbarn angegeben werden können. Die Rückgabe approximativer Ergebnisse erhält Laufzeit sowie Speicherplatz Vorteile.

Nachteil des FAISS Verfahren ist, dass zurzeit nur die euklidische Distanz unterstützt wird.

4.5 Hyperparameter

- **n_neighbors**: Die Größe der lokalen Nachbarschaft. Größere Werte erzielen eine globalere Ansicht der Daten. Eine kleine Wahl des Parameters stellt hingegen lokale Distanzen besser dar. Zusätzlich ist die Wahl des Parameters abhängig von der Anzahl der Daten. Dabei kann man festhalten, je größer N , desto größer sollte auch **n_neighbors** gewählt sein.
- **n_epochs** Die Anzahl der Iterationen im SGD. Je größer die Wahl des Parameters, desto besser sollte die Minimierung sein. Allerdings wird hier kein globales Minimum garantiert. Jedoch liefert die Initiale Einbettung beruhend auf der spektralen Einbettung einen guten Startwert für die Optimierung. Die Wahl der Schrittgröße im Gradientenverfahren ist antiproportional zu **n_epochs** gewählt und wird in jeder Iteration verringert. Dies hat sich in der Praxis als sinnvolle Wahl der Schrittgröße erwiesen um nicht zu schnell in lokalen Minima festzustecken.
- **set_op_mix_ratio** Dieser Parameter gibt an, in welchem Verhältnis eine Vereinigung mittels t-Conorm, oder ein Schnitt mittels Produkt t-Norm durchgeführt werden soll. In den theoretischen Überlegungen haben wir uns stets auf die Vereinigung beschränkt. Anschaulich gesehen setzen sich bei der Vereinigung, welche sich für den Fall der 1-Skelette als Symmetrisierung beschreiben lässt die größeren Gewichte durch. Somit werden Punkte in eine Zusammenhangkomponente *gezwungen*. Wenn man im Vergleich dazu den Schnitt, in unserem Fall die Produkt t-Norm, betrachtet werden die kleineren Gewichte bevorzugt. Somit erhält man mehr Zusammenhangkomponenten. Dabei beschreibt **set_op_mix_ratio = 1** die vorherigen Überlegungen nur die Vereinigung zu betrachten, **set_op_mix_ratio = 0** betrachtet nur den Schnitt, **set_op_mix_ratio = a** mit $a \in (0, 1)$ beschreiben das Verhältnis zwischen Vereinigung und Schnitt. Eine kleine Wahl kann beispielsweise dann sinnvoll sein, wenn man Ausreißer in den Daten vermutet.
- **min-dist** Kleine Werte des Parameters erzeugen dichtere Strukturen. Größere Distanzen sorgen dafür, dass die Distanz zwischen den Punkten in der Einbettung vergrößert wird. Der Startwert 0,1 liefert meist gute Ergebnisse. Wenn viele Datenpunkte eingebettet werden sollte **min-dist** entsprechend größer gewählt werden um ein *overplotting* zu vermeiden.

Kapitel 5

Experimente

Nach ausführlicher Darstellung der Theorie des UMAP Verfahrens, möchten wir nun UMAP auf drei Datensätzen mit alternativen Verfahren empirisch testen. Wir werden eine möglichst vollständige Darstellung der Ergebnisse in dieser Arbeit präsentieren. Allerdings ist es zu empfehlen die Ergebnisse in einem interaktiven Jupyter notebook zu betrachten. Dieses befindet sich auf der beigelegten CD oder auf GitHub¹.

In diesem Kapitel werden wir in Abschnitt 5.1 zwei alternative Verfahren beschreiben. Wir werden in Abschnitt 5.2 beschreiben, wie wir die Einbettungen bewertet haben. Abschnitt 5.3 wird einen neuen Datensatz, das Cartoon Set vorstellen. Dieses besitzt eine komplexere Struktur und bietet uns somit mehr Raum für eine Analyse der Einbettungen. In Abschnitt 5.4 werden wir die Einbettungen verschiedener Verfahren auf dem bekannten MNIST Datensatz betrachten. Dieses Kapitel werden wir mit der Betrachtung der Laufzeiten abschließen.

5.1 Alternative Verfahren

Wir haben uns dazu entschieden UMAP mit t-SNE, TriMap und PCA zu vergleichen. Die ersten beiden Verfahren sollen kurz eingeführt werden, wobei wir insbesondere die wichtigen Hyperparameter der Verfahren angeben. Eine Kenntnis des PCA Verfahrens setzen wir voraus. An dieser Stelle ist zu bemerken, dass in den letzten Jahren zahlreiche neue DR Verfahren entwickelt wurden. Aufgrund dessen können wir in dieser Arbeit nur eine kleine Teilmenge der DR Verfahren betrachten. Ein sehr ausführlicher Vergleich findet sich in [23].

Da die Implementierungen des Laplacian Eigenmaps Verfahrens und Isomap Verfahrens schlecht für große Datensätze skalieren, haben wir uns bewusst dazu entschieden, diese nicht mit in die Analyse aufzunehmen.

5.1.1 t-SNE

Das t-SNE Verfahren [25] ist zur Zeit eines der bekanntesten und meistgenutzten nicht-linearen Dimensionsreduktionsverfahren. Dabei wird t-SNE fast ausschließlich zur Visualisierung genutzt, da die Laufzeit für höhere Einbettungsdimensionen schlecht ist.

t-SNE konstruiert zuerst eine Wahrscheinlichkeitsverteilung P auf Paaren (i, j) der hochdimensionalen Datenpunkten. Diese ist so gewählt, dass Paare ähnlicher Objekte eine höhere Wahrscheinlichkeit zugeordnet bekommen, wohingegen sehr unterschiedliche Datenpunkte eine Wahrscheinlichkeit nahe 0 haben. Die Ähnlichkeit der Punkte wird dabei meist mittels der euklidischen Distanz gemessen, kann aber ähnlich wie im UMAP Algorithmus durch andere Metriken ersetzt werden. Um P zu konstruieren wird eine Gaußverteilung genutzt, wobei die Varianz abhängig vom perplexity

¹https://github.com/reinerschristopher/review_DR_algorithms

Parameter ist. Die so erhaltenen Wahrscheinlichkeiten $p_{i|j}$ sind im Allgemeinen nicht symmetrisch. Die Symmetrie wird durch mitteln der Daten erhalten.

Ähnlich wird eine Wahrscheinlichkeitsverteilung Q im niedrigdimensionalen Raum mithilfe der studentschen t-Verteilung konstruiert. Ursprünglich wurde Q ebenfalls durch eine Gausverteilung konstruiert, das so erhaltene Verfahren (SNE [16]) ist allerdings aufgrund einer schwierig zu optimierenden Zielfunktion und dem „crowding problem“ wenig praktikabel.

Um die d-dimensionale Repräsentation der Daten zu optimieren wir die Kullback-Leiber Divergenz von zwischen P und Q bezüglich der y_i minimiert.

Seit der Veröffentlichung des Verfahrens wurden zahlreiche Verbesserungen, insbesondere für die Laufzeit, vorgeschlagen [37, 21]. Dabei ist besonders Barnes-Hut-SNE [24] zu erwähnen, allerdings sollte hier beachtet werden, dass aufgrund der Konstruktion einer speziellen Datenstruktur die Laufzeit für $d > 3$ sehr schlecht ist.

Die von t-SNE produzierte Repräsentation der Daten ist vom **perplexity** Parameter abhängig. Dabei kann man festhalten, je größer die **perplexity** ist, desto größer ist die Varianz des Gaußverteilung. Somit werden für große **perplexity** Werte globalere Strukturen erfasst, da der Gaußkern sehr breit ist. Wenn der **perplexity** Parameter in der Größenordnung der Anzahl an Datenpunkten N liegt gleicht t-SNE dem MDS Verfahren.

Der zweite wichtige Hyperparameter, welchen wir beschreiben möchten ist die **exaggeration**. meistens wird hier zwischen **early-** und **late-exaggeration** unterschieden. Im wesentlichen verbessert der Parameter die Optimierung des Gradienten und sorgt dafür, dass Punkte desselben Clusters möglichst schnell in der niedrigdimensionalen Repräsentation gruppiert werden [20]. Der **late-exaggeration** wie in [21] beschrieben kontrahiert gefundene Cluster, so lassen sich in einer 2- oder 3-dimensionalen Darstellung leichter Cluster bestimmen - entweder visuell oder mittels Clustering-Verfahren.

Wir werden reale Datensätze analysieren und das Verhalten der Hyperparameter beschreiben, um ein zusätzliches Verständnis für die von t-SNE genutzten Hyperparameter zu bekommen empfiehlt sich [41], dort werden interaktiv auf künstlich erzeugten Datensätzen die Auswirkung der Parameter gezeigt.

Für unsere Experimente haben wir die scikit Implementierung des t-SNE Verfahren genutzt [32]. Zusätzlich haben wir die openTSNE [33] Implementierung genutzt. Diese beschleunigt die Laufzeit des t-SNE Algorithmus durch eine zusätzliche Fouriertransformation [21]. Die openTSNE Implementierung besitzt im Vergleich zur scikit Implementierung die Möglichkeit den **late-exaggeration** Parameter zu spezifizieren.

Um einen Laufzeitvergleich mit der GPU Implementierung des UMAP Verfahrens zu ermöglichen, werden wir eine GPU Implementierung von t-SNE betrachten [5]. Eine Beschreibung findet sich in [6].

5.1.2 TriMap

Das TriMap Verfahren [1] soll eine globalere Repräsentation der Daten finden als beispielsweise t-SNE, da nicht nur paarweise die Ähnlichkeit zweier Objekte i, j betrachtet wird, sondern stets Triple i, j, k . Die so erhaltene globale Struktur der Daten soll die Cluster-Abstände der Daten repräsentieren. Die von uns gewählte Implementierung des Verfahrens findet sich in [40].



ABBILDUNG 5.1: Sechs zufällig gewählte Gesichter des Cartoon Set.

Wir haben diesem Algorithmus gewählt, da die Tripletts Ähnlichkeiten mit 2-Simplizes des UMAP Verfahren haben. Die Tripletts sind vergleichbar mit den 2-Simplizes des UMAP Verfahrens. Insbesondere können die Ansätze eine lineare Teilmenge ($O(N)$) an Tripletts zu finden weitere Entwicklungen des UMAP Verfahren motivieren.

5.2 Bewertung der Ergebnisse

Um die lokale Qualität der Algorithmen zu analysieren haben wir uns die *Cluster* in der 2-dimensionalen Repräsentation angeschaut, wobei wir als Cluster eine Teilmenge der Daten bezeichnen, welche deutlich von den anderen Datenpunkten getrennt ist. Insbesondere bei der Analyse des Cartoon Set 5.3 konnten wir gut lokale Strukturen erkennen, da jeder Datenpunkt mehrere Eigenschaften gegeben hat – im Vergleich zum MNIST und FMNIST Datensatz, wo uns nur ein *Label* pro Datenpunkt gegeben ist.

Die globale Struktur der Repräsentation qualitativ zu bewerten ist subjektiv. Dabei ist insbesondere die Frage – „Wie stark unterscheiden sich die Cluster?“ – zu beantworten.

Für die qualitative Analyse wird die Fähigkeit des Gehirns genutzt Strukturen zu erkennen. Nachteile der qualitativen Analyse sind, (1) die Subjektivität und somit Abhängigkeit vom Betrachter, (2) dass sie nur im d -dimensionalen ($d \leq 3$) möglich ist.

Wir verweisen den Leser auf [14, 18, 34] für verschiedene Ansätze DR Verfahren auf eine quantitative Weise zu bewerten.

In den folgenden Experimenten haben wir uns auf eine qualitative Analyse der Daten beschränkt.

5.3 Cartoon Set

In diesem Abschnitt werden wir den *Cartoon Set* Datensatz analysieren [8]. Dabei werden wir,

- die Einbettungen von UMAP mit denen von t-SNE und TriMap visuell bewerten,
- und die drei Verfahren auf globale und lokale Strukturen vergleichen,
- an einigen Stellen verdeutlichen wie schwierig die Bewertung der Qualität der Einbettung ist.

Beschreibung des Datensatzes

Der Cartoon Datensatz enthält 100 000 unterschiedliche Bilder von gezeichneten Gesichtern (siehe Abbildung 5.1).

Die Bilder wurden aus 16 Labels zusammengesetzt (u.a. Gesichtsform, Gesichtsfarbe, Frisur, Haarfarbe), dabei variiert die Anzahl der Möglichkeiten pro Label zwischen zwei (Augenlid, Wimpern,...) und 111 (Anzahl mögliche Frisuren). Die Farben der Komponenten wurden aus einem diskreten RGB Raum gewählt. Insgesamt ergibt sich eine mögliche Anzahl von 10^{13} Gesichtern. Für die Analyse haben wir verschiedene Eigenschaften zusammengefasst um einen besseren Überblick zu haben. Beispielsweise haben wir die 111 Frisuren, nach qualitativer Analyse, zu 19 Frisurformen zusammengefasst.

Die ursprüngliche Größe eines Bildes betrug 500×500 Pixel mit vier Farbkanälen (CYMK-Darstellung der Farben). Aufgrund des großen Randes haben wir uns dazu entschieden die Größe der Bilder auf 300×300 ohne nennenswerten Informationsverlust zu verringern. Somit beträgt die Dimension des Cartoon Set $D = 360\,000$ und die Anzahl an Beispielen N variiert zwischen 10 000 und 100 000. An dieser Stelle ist die sehr hohe Umgebungsdimension hervorzuheben. Typischerweise werden Daten mit einer Umgebungsdimension $D > 10\,000$ mit PCA auf eine Dimension in den Bereich $D' \in [100, 1000]$ vorverarbeitet. Später werden wir sehen, dass diese Vorverarbeitung bezogen auf unsere Bilddaten keine großen Nachteile birgt.

Wir haben uns für diesen Datensatz entschieden um UMAP auf Daten mit einer komplexeren Struktur zu testen als dies in [27] gemacht wird. Dabei ist auch zu beachten, dass aufgrund der 16 Labels aus welchen die Gesichter bestehen, die Qualität einer Einbettung schwieriger zu beurteilen ist als beispielsweise im MNIST Datensatz (siehe Abschnitt 5.4). Wir haben die Bewertung der Einbettung unter der Annahme gemacht, dass *ähnliche* Gesichter *ähnliche* Hautfarben, Frisuren, Haarfarben, Brillen und Bärte besitzen. Diese fünf Eigenschaften möchten wir besonders hervorheben, da sie die dominantesten Merkmale des Gesichts beschreiben. Somit werden wir eine Einbettung des Cartoon Set als *gut* bewerten, wenn sie zwischen diesen fünf Merkmalen unterscheidet.

Qualitative Analyse der Ergebnisse

Wir werden nun die Einbettungen bewerten. Für die Einbettung des Cartoon Set haben wir die voreingestellten Hyperparameter gewählt. Für das UMAP Verfahren haben wir nur den Parameter `n_neighbors` = 50 und für t-SNE den Parameter `perplexity` = 40 gewählt, kleinere Parameter der Wert haben Einbettungen mit zu vielen Clustern ergeben. Abbildung 5.2 zeigt eine Einbettung des Cartoon Set mittels PCA. Dies dient als Startwert unserer Analyse. Im linken Plot sind dabei zwei Linien eingezeichnet. Diese verdeutlichen, dass mittels PCA die Gesichter in 3 Klassen eingeteilt werden können. Die Farben des Punktes in diesem Plot stellen die Frisurtypen dar. Die oberen blauen und roten Punkte stellen Gesichter mit wenigen Haaren dar, die Punkte eingeschlossen von den beiden Linien stellen Gesichter mit mittellangen und die unteren Punkte Gesichter mit langen Haaren dar. Ebenfalls in Abbildung 5.2 sind exemplarisch einige Gesichter mit ihrer zugehörigen Lage in der Einbettung zu sehen. Der mittlere Plot visualisiert die Gleiche Einbettung, es wurden nur andere labels für die Daten genutzt. Die Farben repräsentieren die Gesichtsfarben, dabei entspricht gelb hellen Hauttypen, lila sehr dunklen und türkis/grün den restlichen Hauttypen. Ähnlich werden im rechten Plot die Farben der Punkte entsprechend ihrer Haarfarben gewählt, wobei der Verlauf von gelb über grün nach lila den Verlauf schwarzen über braunen nach blonden/grauen Haarfarben entspricht. Wir werden diese Farbwahl in den anderen Plots beibehalten. Auch hier ist eine Struktur zu erkennen, auffällig ist dabei, dass die linearen Strukturen vom oberen Drittel schlechter sind als die im

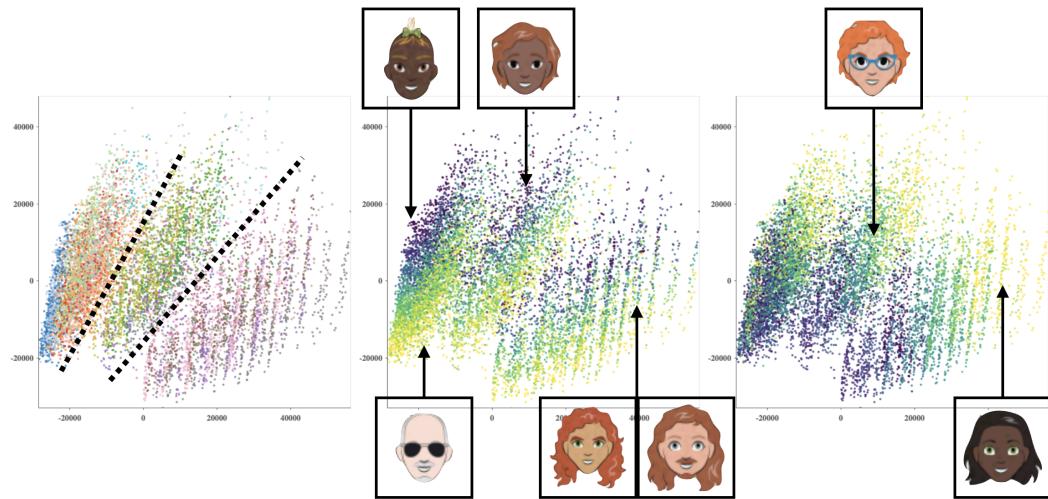


ABBILDUNG 5.2: PCA auf dem Cartoon Set, mit $N = 50\,000, D = 360\,000$. Dabei sind die Punkte bezüglich der folgenden Attribute gekennzeichnet (v.l.n.r): Frisurtyp, Gesichtsfarbe, Haarfarbe.

unteren Drittel. Dies können wir dadurch erklären, dass sich längere Haartypen leichter unterscheiden lassen als beispielsweise die Unterscheidung zwischen keinen und sehr wenig kurzen Haaren. Wir können an dieser Stelle unsere Annahme bestätigen, dass das Cartoon Set eine zugrundeliegende Struktur besitzt, welche unsere Intuition widerspiegelt.

In Abbildung 5.3 betrachten wir Einbettungen von UMAP, t-SNE und TriMap. Durch das einzeichnen der Cluster möchten wir untersuchen, wie gut die Verfahren eine *globale* Struktur erhalten. Die Ergebnisse der Einbettungen unterscheiden sich stark von der PCA Einbettung. Alle drei Verfahren erkennen die globale Einteilung der Daten in verschiedene Frisurtypen. Dabei unterscheiden sie zusätzlich zu PCA innerhalb der drei Klassen, kurze, mittellange und lange, in feinere Klassen. Diese entsprechen den verschiedenen Labels der Frisurtypen, wie man an den Farben der Cluster erkennen kann. Dabei scheint UMAP die globale Struktur auf den ersten Blick am besten zu erkennen, da die langen und kurzen Haartypen global große Abstände aufweisen. Allerdings kann auch argumentiert werden, dass TriMap die globale Struktur am besten darstellt, da das gelbe und blaue Cluster zwar durch Haartypen mittellanger Haare getrennt sind aber die Distanzen zwischen den Clustern eher fließend sind, analog zu PCA. Dies weist auf die Schwierigkeit hin, ein Maß der globalen Qualität einer Einbettung zu definieren.

Wir möchten uns nun den lokalen Eigenschaften der Einbettungen zu wenden. Dafür betrachten wir die gleiche Einbettung allerdings haben wir die Farben der Punkte diesmal bezüglich der Haarfarbe, beziehungsweise der Hautfarbe gewählt. Den Verlauf der Farben bezüglich dieser Attribute haben wir bereits oben beschrieben. Abbildung 5.4 zeigt die von UMAP, t-SNE und TriMap erzeugte Einbettung eines Teils der Daten. Wir sehen, dass alle drei Verfahren lokal die Cluster, welche die Frisurtypen darstellen, als zweidimensionale Mannigfaltigkeiten auffassen. Dabei ist eine Dimension die Haarfarbe und die zweite Dimension die Hautfarbe. Alle drei Plots der oberen Reihe weisen ein Cluster auf, welches anscheinend keine Struktur bezüglich der Haarfarbe aufweist. Diese Punkte beschreiben allerdings Gesichter ohne Haare. Die Einbettung des t-SNE Verfahrens zeigt lokal keine deutlichen Unterschiede in der Dichte. Die Übergänge zwischen den lokalen Eigenschaften in der TriMap Einbettung sind fließender als in

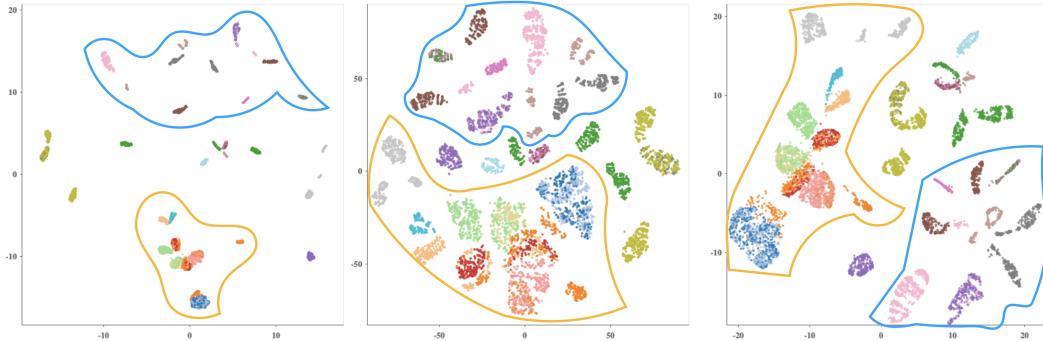


ABBILDUNG 5.3: (v.l.n.r) UMAP, t-SNE, TriMap auf dem Cartoon Set, mit $N = 50\,000, D = 360\,000$. Dabei entsprechen die Punkte des gelben Gebiets dem oberen Drittel der PCA Einbettung, das blaue Gebiet dem unteren Drittel der PCA Einbettung und die Punkte außerhalb eines Gebietes dem mittleren Teil der PCA Einbettung

der t-SNE und UMAP Einbettung.

Abbildung 5.5 beruht auf der für Abbildung 5.4 beschriebenen Einbettung. Die Farben sind bezüglich der Haarfarbe. Es ist deutlich zu erkennen, dass alle drei Verfahren die Gleiche lokale Struktur für dieses CLuster finden.

Wir können festhalten, das die drei Verfahren lokal sehr ähnlich sind, die Wahl eines Verfahrens ist somit von der konkreten Anwendung abhängig, oder im Falle der Visualisierung vom subjektiven Urteil des Betrachters. Global enthält die Darstellung von t-SNE sehr wenige Informationen. Das UMAP und TriMap Verfahren erhalten globale Eigenschaften, dabei ist der Begriff *global* allerdings nicht formal gegeben und sollte genauer untersucht werden.

5.4 MNIST

Wir möchten kurz die Einbettungen der Verfahren auf dem MNIST Datensatz zeigen. Dieser besteht aus Bildern der Größe 28×28 , welche die Ziffern von 0 bis 9 darstellen. Dabei ist zu erkennen, dass alle nicht linearen Verfahren 10 Cluster erkennen. Das UMAP Verfahren spiegelt die mittels PCA gefundene globale Struktur am besten wieder. Dabei ist auch zu erwähnen, dass TriMap eine gewisse globale Struktur darstellt, da es das blaue Cluster (Ziffer 1), das pinke, türkise und orange Cluster (Ziffern 7, 4, 9) und die restlichen Cluster von einander separiert. Zusätzlich ist zu erkennen, das die Cluster von openTSNE separierter von einander sind, als die vom tSNE Verfahren, dies ist auf die Implementierung des `late-exaggeration` Parameters im openTSNE Code zurückzuführen.

5.5 Laufzeitanalyse

Die praktischen Tests der Verfahren wurden auf Rechnern mit einer Linux-Architektur ausgeführt. Die CPU Tests haben wir auf Intel Xeon 6136 CPUs mit 48 Kernen und 384 GB RAM ausgeführt. Für die Verfahren welche mittels Berechnungen auf einer Graphikkarte verbessert wurden, haben wir Intel Xeon Gold 6136 CPUs mit 188 GB RAM und Nvidia V100 GPUs genutzt. In Tabelle 5.1 stellen wir die Laufzeit von UMAP abhängig vom `n_neighbors` Parameter da. Die Laufzeit des Verfahrens erhöht sich um den Faktor $8x$, wenn der Parameter von 5 auf 100 erhöht wird. Dies spiegelt die in Abschnitt 4.2 gemachte Aussage wieder, dass das NN-Descent Verfahren

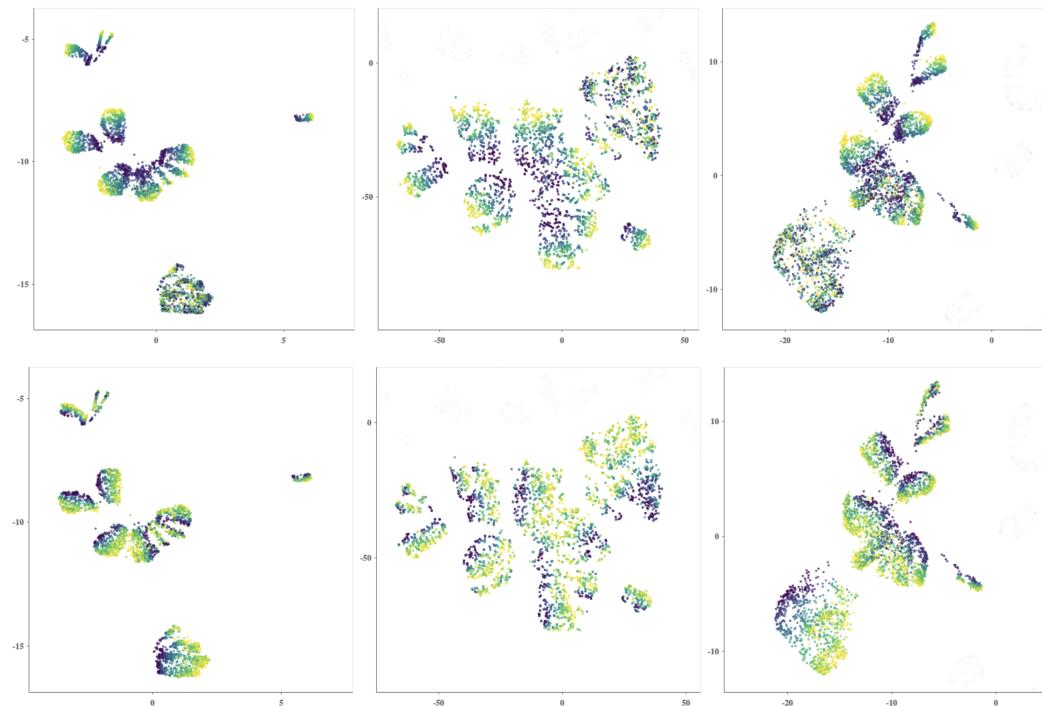


ABBILDUNG 5.4: (v.l.n.r) UMAP, t-SNE, TriMap auf dem Cartoon Set, mit $N = 10\,000, D = 360\,000$. Die Datenpunkte der oberen Reihe sind nach der Haarfarbe und die der unteren Reihe nach der Hautfarbe gefärbt. Es wurde nur ein Teil des Datensatzes betrachtet.

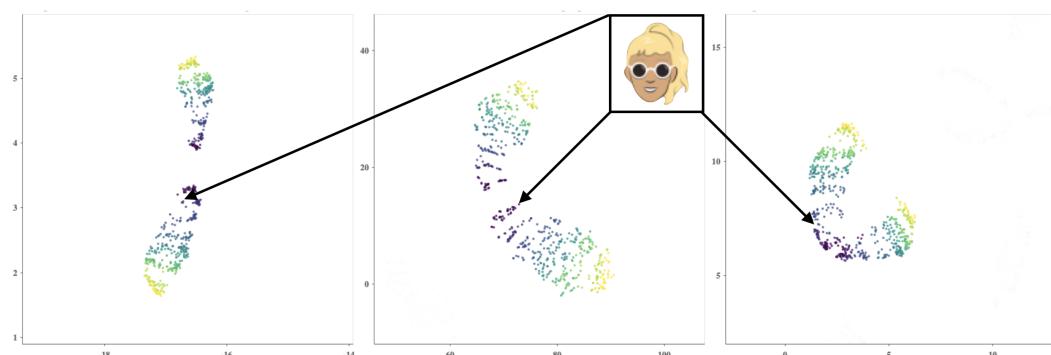


ABBILDUNG 5.5: (v.l.n.r) UMAP, t-SNE, TriMap auf dem Cartoon Set, mit $N = 50\,000, D = 360\,000$. Die Position eines Datenpunktes bezüglich der verschiedenen Verfahren.

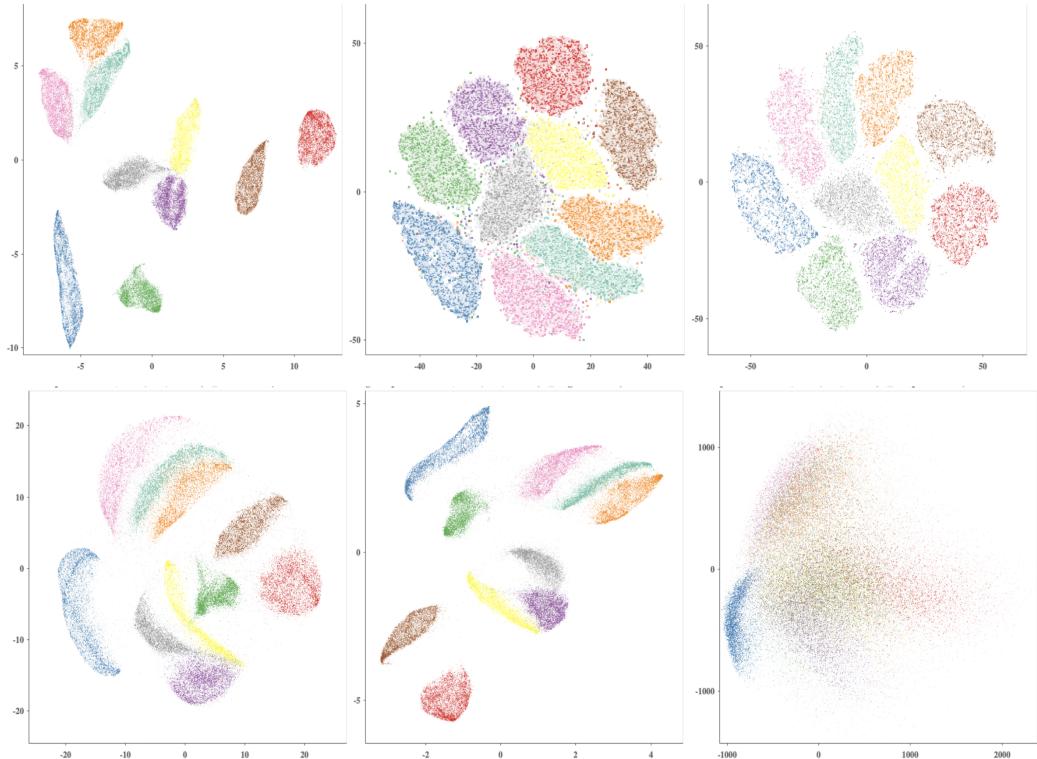


ABBILDUNG 5.6: (oben v.l.n.r) UMAP, t-SNE, OpenTSNE auf FM-NIST, mit $N = 70\,000, D = 784$. (unten v.l.n.r) TriMap, GPU UMAP, PCA

n_neighbors	MNIST	FMNIST	Cartoon 10k	Cartoon 50k
5	3,7 min	4 min	7,4 min	62 min
10	5,3 min	6 min	11,4 min	-
20	10,6 min	12 min	23 min	-
40	25 min	24 min	77 min	7 h
100	34 min	32 min	86 min	8 h

TABELLE 5.1: Laufzeit des UMAP Verfahren mit unterschiedlichen Werten des `n_neighbors` Parameters.

	MNIST	FMNIST	Cartoon 10k
UMAP	25 min	23,8 min	77 min
t-SNE	2 h	130 min	17 h
openTSNE	11,6 min	12 min	-
TriMap	4 min	4 min	5,4 min
Cuda UMAP	8 sec	7,7 sec	-
Cuda t-SNE	5 sec	5,4 sec	-
PCA	1,3 sec	1,2 sec	9,2 min

TABELLE 5.2: Laufzeit verschiedener Verfahren der Dimensionsreduktion

zur Nachbarschaftssuche eine Schwachstelle im UMAP Verfahren ist. Betrachtet man die Laufzeit der GPU Implementierung [31] auf dem FMNIST Datensatz, so erhöht sich die Laufzeit von 2,63 Sekunden auf 3 Sekunden. Leider konnten wir die GPU Implementierung nicht auf dem vollen Cartoon Set testen, da die aktuelle Version den Datensatz in den GPU Arbeitsspeicher lädt. Wir hatten nur 16 GB GPU RAM zur Verfügung, dies hat für die hohe Dimension des Cartoon Set nicht ausgereicht hat. Um dennoch das GPU Verfahren mit höher dimensional Daten zu testen, haben wir das Cartoon Set zu erst mit PCA auf 784 Dimensionen vorverarbeitet, die Laufzeit auf dem vorverarbeiteten Datensatz variierte zwischen 4 Sekunden und 7,4 Sekunden.

Natürlich möchten wir die Laufzeit des UMAP Verfahrens auch mit der von anderen Verfahren vergleichen, dazu haben wir drei Implementierungen des t-SNE Verfahrens betrachtet, das oben beschriebene TriMap Verfahren und PCA.

Wir können somit festhalten, dass die GPU Implementierungen aufgrund ihrer Laufzeit für Zwecke genutzt werden können, wo bisher nur einfache lineare DR Verfahren wie PCA genutzt werden konnten. Die TriMap Implementierung ist das schnellste nicht-lineare CPU DR Verfahren, welches wir betrachtet haben. Insbesondere skaliert es sehr gut in der Umgebungsdimension, wobei wir hervor heben möchten, dass es auf dem Cartoon Set mit $N = 10\,000$, $D = 360\,000$ schneller als PCA war. Somit scheint eine ausführlichere Betrachtung des TriMap Verfahrens aufgrund seiner Laufzeit und Qualität der Ergebnisse sinnvoll zu sein.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit haben wir eine kompakte Einführung in die Datenanalyse gegeben und dabei betont, dass Daten, insbesondere Daten hoher Dimensionen eine immer wichtigere Rolle spielen da sie in großen Mengen gesammelt werden können. Wir haben eine Klasse an Verfahren vorgestellt, welche zu gegebenen hochdimensionalen Daten eine Einbettung in einem niedrigdimensionalen Raum finden.

Um das UMAP Verfahren vorzustellen haben wir Aussagen aus unterschiedlichen mathematischen Teilgebieten kennengelernt. Dabei haben wir einerseits einen Fokus auf die mathematische Korrektheit gelegt, andererseits mittels Erklärungen und Beispielen die Aunschauung hinter den eingeführten Konzepten erläutert.

Die Theorie der unscharfen topologischen Repräsentation wurde eingeführt und mit Erklärungen ergänzt. Es wurde eine Beschreibung der Implementierung gegeben und diese genutzt um Datensätze zu analysieren. Zudem haben wir die Problemstellung ein Qualitätsmaß für Einbettungen zu finden betrachtet und einen neuen Datensatz ausführlich analysiert.

Dabei hoffen wir, dass wir den Leser für das UMAP Verfahren und die (topologische) Datenanalyse begeistern konnten und ihn zu weiteren Überlegungen motivieren konnten.

6.2 Ausblick

In den vergangenen Monaten haben wir uns intensiv mit dem UMAP Verfahren auseinandergesetzt und sind dabei zu einigen Überlegungen gekommen.

An dieser Stelle möchten wir eine Problematik aufgreifen, welche bei der Entwicklung des UMAP Verfahren auftritt. In Kapitel 3 wurde die Theorie des Verfahrens entwickelt. Die dort verwendeten Aussagen um die Theorie des UMAP Verfahrens mathematisch zu begründen baut auf unsicheren simplizialen Mengen auf. Die aktuellen praktischen Implementierungen greifen dabei nicht auf diese Mengen zurück, sondern auf eine (starke) Vereinfachung. Es sollte genauer untersucht werden, wie stark diese Vereinfachung in der Praxis ist. Im Rahmen dieser Arbeit haben wir zwar versucht Ansätze dafür zu entwickeln, sind aber leider aufgrund unserer fehlenden Kenntnisse im Bereich der Topologie zu keinen wesentlichen Resultaten gekommen. Für unsere Ansätze haben wir Čech-Komplexe und VR-Komplexe betrachtet. Diese werden in der topologischen Datenanalyse an verschiedenen Stellen verwendet. Kurz gesagt, kann man sich unter dem VR-Komplex den konstruierten Graphen und unter einem Čech-Komplex die unscharfe topologische Repräsentation \mathcal{X} vorstellen, dies ist nicht ganz richtig, da beide Konstrukte nur Simplizialkomplexe sind und nicht die allgemeinere Form der simplizialen Mengen darstellen. Diese Herrangehensweise würde

die Theorie der Praxis angleichen. Umgekehrt könnte man auch die praktische Implementierung auf n -Simplizes erweitern. Ein interessanter Ansatz ist dabei die Wahl der Tripel in [1].

Eine weitere Baustelle ist die quantitative Bewertung von Einbettungen. Dies stellt, wie wir gesehen haben, eine schwierige Aufgabe dar.

Im Rahmen dieser Arbeit haben wir uns auf Bilddaten beschränkt, wir möchten den Leser dazu motivieren das UMAP Verfahren auf anderen Datensätzen zu analysieren

Wir haben uns nicht mit der Frage nach der intrinsischen Motivation beschäftigt, diese Frage ist keineswegs trivial. Insbesondere spielt die Frage eine Rolle, wenn UMAP zur Vorverarbeitung von Daten genutzt wird. Dabei können wir uns vorstellen, dass sich die Theorie der simplizialen Mengen als nützlich erweisen kann, da eine simpliziale Menge im wesentlichen aus unterschiedlich-dimensionalen Simplizes besteht, somit könnten verschiedene Bereiche des der Daten eine unterschiedliche Dimension annehmen.

Wir möchten die Arbeit mit folgendem Zitat abschließen,

„The goal is to turn data into information, and information into insight.“

Carly Fiorina

Literatur

- [1] Ehsan Amid und Manfred K. Warmuth. „A more globally accurate dimensionality reduction method using triplets“. In: *CoRR* abs/1803.00854 (2018). arXiv: [1803.00854](https://arxiv.org/abs/1803.00854). URL: <http://arxiv.org/abs/1803.00854>.
- [2] Michael Barr. „Fuzzy Set Theory and Topos Theory“. In: *Canadian Mathematical Bulletin* 29.04 (Dez. 1986), S. 501–508. ISSN: 1496-4287. DOI: [10.4153/cmb-1986-079-9](https://doi.org/10.4153/cmb-1986-079-9).
- [3] Kevin Beyer u. a. „When Is “Nearest Neighbor” Meaningful?“ In: *Database Theory — ICDT’99*. Hrsg. von Catriel Beeri und Peter Buneman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, S. 217–235. ISBN: 978-3-540-49257-3.
- [4] Martin Brandenburg. *Einführung in die Kategorientheorie*. Springer Berlin Heidelberg, 2016. ISBN: 9783662470688. DOI: [10.1007/978-3-662-47068-8](https://doi.org/10.1007/978-3-662-47068-8).
- [5] David M Chan. *t-SNE GPU Implementation*. URL: <https://github.com/CannyLab/tsne-cuda> (besucht am 05.08.2019).
- [6] David M Chan u. a. „GPU accelerated t-distributed stochastic neighbor embedding“. In: *Journal of Parallel and Distributed Computing* 131 (2019), S. 1–13.
- [7] F.R.K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series Nr. 92. Conference Board of the Mathematical Sciences. ISBN: 9780821889367.
- [8] Forrester Cole, Shiraz Fuman und Aaron Sarna. *Cartoon Set*. URL: <https://google.github.io/cartoonset/download.html> (besucht am 19.07.2019).
- [9] Wei Dong, Charikar Moses und Kai Li. „Efficient K-nearest Neighbor Graph Construction for Generic Similarity Measures“. In: *Proceedings of the 20th International Conference on World Wide Web*. WWW ’11. Hyderabad, India: ACM, 2011, S. 577–586. ISBN: 978-1-4503-0632-4.
- [10] Samuel Eilenberg und Saunders MacLane. „General theory of natural equivalences“. In: *Transactions of the American Mathematical Society* 58 (1945), S. 231–231. ISSN: 0002-9947. DOI: [10.1090/s0002-9947-1945-0013131-6](https://doi.org/10.1090/s0002-9947-1945-0013131-6).
- [11] Greg Friedman. „An elementary illustrated introduction to simplicial sets“. In: *arXiv:0809.4221v5* (2016).
- [12] Cong Fu und Deng Cai. „EFANNA : An Extremely Fast Approximate Nearest Neighbor Search Algorithm Based on kNN Graph“. In: *CoRR* abs/1609.07228 (2016). arXiv: [1609.07228](https://arxiv.org/abs/1609.07228). URL: <http://arxiv.org/abs/1609.07228>.
- [13] Joseph A. Goguen. „Concept representation in natural and artificial languages: Axioms, extensions and applications for fuzzy sets“. In: *International Journal of Man-Machine Studies* 6.5 (1974), S. 513–561. ISSN: 0020-7373. DOI: [https://doi.org/10.1016/S0020-7373\(74\)80017-9](https://doi.org/10.1016/S0020-7373(74)80017-9).
- [14] Stefan Harmeling. „Exploring model selection techniques for nonlinear dimensionality reduction“. (EDI-INF-RR-0960). Edinburgh, UK: School of Informatics, University of Edinburgh. 2007.

- [15] Cordelia Schmid Hervé Jégou Matthijs Douze. „Product Quantization for Nearest Neighbor Search“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Bd. 33. Institute of Electrical und Electronics Engineers. 2011, S. 117–128.
- [16] Geoffrey E Hinton und Sam T. Roweis. „Stochastic Neighbor Embedding“. In: *Advances in Neural Information Processing Systems 15*. Hrsg. von S. Becker, S. Thrun und K. Obermayer. MIT Press, 2003, S. 857–864. URL: <http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf>.
- [17] Jeff Johnson, Matthijs Douze und Hervé Jégou. „Billion-scale similarity search with GPUs“. In: *arXiv preprint arXiv:1702.08734* (2017).
- [18] John A. Lee und Michel Verleysen. „Quality assessment of dimensionality reduction: Rank-based criteria“. In: *Neurocomputing* 72.7-9 (März 2009), S. 1431–1443. ISSN: 0925-2312. DOI: [10.1016/j.neucom.2008.12.017](https://doi.org/10.1016/j.neucom.2008.12.017).
- [19] Paul Blain Levy. „Formulating Categorical Concepts using Classes“. In: (2018).
- [20] George C. Linderman und Stefan Steinerberger. „Clustering with t-SNE, Provably“. In: *SIAM Journal on Mathematics of Data Science* 1.2 (Jan. 2019), S. 313–332. ISSN: 2577-0187. DOI: [10.1137/18m1216134](https://doi.org/10.1137/18m1216134). URL: <http://dx.doi.org/10.1137/18m1216134>.
- [21] George C. Linderman u. a. „Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data“. In: *Nature Methods* 16.3 (Feb. 2019), S. 243–245. ISSN: 1548-7105. DOI: [10.1038/s41592-018-0308-4](https://doi.org/10.1038/s41592-018-0308-4). URL: <http://dx.doi.org/10.1038/s41592-018-0308-4>.
- [22] Yujing Ma, Florian Rusu und Martin Torres. „Stochastic Gradient Descent on Modern Hardware: Multi-core CPU or GPU? Synchronous or Asynchronous?“ In: *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Institute of Electrical und Electronics Engineers (IEEE), 2019, S. 1063–1072.
- [23] L.J.P. van der Maaten u. a. „Dimensionality Reduction: A Comparative Review“. In: (2008). DOI: [10.1.1.112.5472](https://doi.org/10.1.1.112.5472). URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.5472>.
- [24] Laurens van der Maaten. „Accelerating t-SNE using Tree-Based Algorithms“. In: *Journal of Machine Learning Research* 15 (2014), S. 3221–3245. URL: <http://jmlr.org/papers/v15/vandermaaten14a.html>.
- [25] Laurens van der Maaten und Geoffrey Hinton. „Visualizing Data using t-SNE“. In: *Journal of Machine Learning Research*. Hrsg. von Yoshua Bengio. Bd. 9. 2008, S. 2579–2605.
- [26] Saunders Mac Lane. „Categories for the working mathematician“. In: 2. Aufl. Graduate texts in mathematics. Springer New York, 1998. Kap. 3.
- [27] Leland McInnes, John Healy und James Melville. „UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction“. In: (). URL: <https://arxiv.org/abs/1802.03426>.
- [28] Leland McInnes u. a. *UMAP Implementation*. 2018. URL: <https://github.com/lmcinnes/umap> (besucht am 22.07.2018).
- [29] Tomas Mikolov u. a. „Distributed Representations of Words and Phrases and their Compositionality“. In: *CoRR* abs/1310.4546 (2013). arXiv: [1310 . 4546](https://arxiv.org/abs/1310.4546). URL: <http://arxiv.org/abs/1310.4546>.

- [30] Hariharan Narayanan und Sanjoy Mitter. „Sample Complexity of Testing the Manifold Hypothesis“. In: *Advances in Neural Information Processing Systems 23*. Hrsg. von J. D. Lafferty u. a. Curran Associates, Inc., 2010, S. 1786–1794. URL: <http://papers.nips.cc/paper/3958-sample-complexity-of-testing-the-manifold-hypothesis.pdf>.
- [31] Corey J Nolet, Dante Dessavre und Thejaswi Rao. <https://github.com/rapidsai/cuml>. (Besucht am 19.07.2019).
- [32] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830.
- [33] Pavlin Policar. *openTSNE Implementation*. URL: <https://github.com/pavlin-policar/openTSNE> (besucht am 22.07.2019).
- [34] B. Rieck und H. Leitte. „Persistent Homology for the Evaluation of Dimensionality Reduction Schemes“. In: *Computer Graphics Forum* 34.3 (Juni 2015), S. 431–440. ISSN: 0167-7055. DOI: [10.1111/cgf.12655](https://doi.org/10.1111/cgf.12655). URL: <http://dx.doi.org/10.1111/cgf.12655>.
- [35] Emily Riehl. *A leisurely introduction to simplicial sets*. Aug. 2008. URL: <http://www.math.jhu.edu/~eriehl/ssets.pdf>.
- [36] Salah Rifai u. a. „The Manifold Tangent Classifier“. In: *Advances in Neural Information Processing Systems 24*. Hrsg. von J. Shawe-Taylor u. a. Curran Associates, Inc., 2011, S. 2294–2302. URL: <http://papers.nips.cc/paper/4409-the-manifold-tangent-classifier.pdf>.
- [37] Erich Schubert und Michael Gertz. „Intrinsic t-Stochastic Neighbor Embedding for Visualization and Outlier Detection“. In: *Lecture Notes in Computer Science* (2017), S. 188–203. ISSN: 1611-3349. DOI: [10.1007/978-3-319-68474-1_13](https://doi.org/10.1007/978-3-319-68474-1_13).
- [38] Jian Tang u. a. „LINE“. In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15* (2015). DOI: [10.1145/2736277.2741093](https://doi.org/10.1145/2736277.2741093). URL: <http://dx.doi.org/10.1145/2736277.2741093>.
- [39] Jian Tang u. a. „Visualizing Large-scale and High-dimensional Data“. In: *Proceedings of the 25th International Conference on World Wide Web. WWW '16*. International World Wide Web Conferences Steering Committee, 2016, S. 287–297. ISBN: 978-1-4503-4143-1. DOI: [10.1145/2872427.2883041](https://doi.org/10.1145/2872427.2883041).
- [40] *TriMap Implementation*. URL: <https://github.com/eamid/trimap> (besucht am 05.08.2019).
- [41] Martin Wattenberg, Fernanda Viégas und Ian Johnson. „How to Use t-SNE Effectively“. In: *Distill* (2016). DOI: [10.23915/distill.00002](https://doi.org/10.23915/distill.00002). URL: <http://distill.pub/2016/misread-tsne>.
- [42] L.A. Zadeh. „Fuzzy sets“. In: *Information and Control* 8.3 (1965), S. 338–353. ISSN: 0019-9958. DOI: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).