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

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-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 in Ottawa, Kanada. 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 Lennard 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

Kapitel 1

Einleitung

Wer eine Tageszeitung aufschlägt, wird nicht vermeiden können, mindestens einen Artikel über die „Algorithmen der Zukunft“, „Daten hungrigen Konzerne“, ... zu lesen. Fakt ist heutzutage werden mehr Daten generiert und gespeichert als je zuvor in der Geschichte der Menschheit und wir benötigen aussagekräftige, schnelle und zuverlässige Verfahren um die gesammelten Daten auszuwerten, zu strukturieren und einen Mehrwert für die Gesellschaft zu generieren.

Wir werden uns in dieser Arbeit mit einer Klasse an Verfahren befassen, welche hochdimensionale Daten visualisiert. Insbesondere werden wir das UMAP (Uniform Manifold Approximation and Projection) Verfahren

Aufgabenstellung

Datenanalyse

Eine neuere Form der Datenanalyse - die topologische Datenanalyse (*kurz: TDA*) - nutzt mathematische Instrumente des Teilgebiets der Topologie (*griechisch: Lehre vom Ort/Platz*) um Daten zu beschreiben und zu strukturieren.

Wir beschreiben die Situation einen Datensatz X zu analysieren wie folgt. Wir betrachten einen D-dimensionalen Raum und ein Objekt K. In unserem Fall werden wir uns größtenteils mit dem euklidischen Raum $\mathbb{R}^D$ versehen mit der euklidischen Norm $\|\cdot\|$ beschäftigen.

K kann beispielsweise eine abgeschlossene Menge sein. Die genaue Struktur von K bleibt uns allerdings unbekannt. Später werden wir argumentieren, dass K gewisse Regularitätseigenschaften erfüllt und in vielen Fällen lokal einem niedrigdimensionalen Raum $\mathbb{R}^d, (d \ll D)$ gleicht.

Statt K ist uns eine endliche Menge an N Punkten $X = \{\mathbf{x}_i\}_{i=1}^N,\ (\mathbf{x}_i \in \mathbb{R}^D)$, gegeben. X wird als *Punktwolke* bezeichnet und beschreibt in einem Experiment gemessene Daten, beispielsweise Sensormessdaten, biologische Informationen oder Bilddatensätze.

In Kapitel 5 werden wir uns mit einem Bilddatensatz mit 100 000 Farbbildern mit einer Auflösung von 300×300 Pixeln beschäftigen. Wir können diesen Datensatz als $N = 100\,000$-elementige Punktwolke im $\mathbb{R}^{300\times300\times4}$ auffassen, wobei wir die vier Farbkanäle der Bilder berücksichtigen.

Wir gehen dabei davon aus, dass K und X in einem gewissen Sinne „ähnlich“ sind. Nun ist es Ziel der TDA mittels Methoden der Topologie Aussagen über die Struktur von X zu treffen, welche, aufgrund der Ähnlichkeit, auch für K gelten sollen. Die K zugrundeliegende Struktur kann uns dann dabei helfen Aussagen über weitere gemessene Daten zu treffen, da diese auch in der uns unbekannten Menge K liegen sollten. Zusätzlich hilft

Dimensionsreduktion / Problemstellung

Sei $X = \{\mathbf{x}_i\}_{i=1}^N$, $(\mathbf{x}_i \in \mathbb{R}^D)$. Wir bezeichnen D als die Dimension unserer Daten, beziehungsweise als die Anzahl der gemessenen Eigenschaften. N ist die Anzahl der verfügbaren Datenpunkte. In der Praxis können D und N sehr groß sein. So gilt für den Bilddatensatz welchen wir in Kapitel 5 betrachten werden, $N = 100\,000, D = 360\,000$.

Eine Herangehensweise die d-dimensionale Repräsentation zu finden ist es eine $N \times N$ Matrix M aufzustellen, die ersten d Eigenvektoren von M, wobei die Eigenvektoren absteigend nach der Größe der zugehörigen Eigenvektoren geordnet sind, geben

Hier sollen die zwei Arten an DR Algorithmen vorgestellt werden (Matrix-Faktorisierung und Graph Layout). Zusätzlich sollen PCA, Isomap, Laplacian Eigenmaps und t-SNE vorgestellt werden.

Die meisten Verfahren zur Dimensionsreduktion beruhen auf der Annahme, das reale hochdimensionale Daten sich in der Umgebung einer niedrigdimensionalen Mannigfaltigkeit konzentrieren. In der Literatur ist diese Annahme als Mannigfaltigkeit-Hypothese (*engl.: manifold hypothesis*) bekannt [25, 30]. Ansätze um einen gegebenen Datensatz auf diese Annahme zu testen finden sich in [7].

Ziele der Arbeit

Eigene Beiträge

Wir möchten nun die Ziele dieser Arbeit formulieren.

- Einführung in die grundlegenden Werkzeuge der topologischen Datenanalyse
- Mögliche Schritte wie die Lücke zwischen Theorie und Praxis des UMAP Verfahrens geschlossen werden kann
- UMAP anhand sinnvoller Datensätze mit anderen Dimensionsreduktionsverfahren vergleichen
-

Insbesondere werden wir die in Kapitel 2 von [21] beschriebene Theorie ausführlicher beschreiben und in einen allgemeineren Kontext setzen.

Aufbau der Arbeit

In Kapitel 2 werden wir die theoretische Grundlage des UMAP Verfahrens erklären. Dies wird einige Definitionen und Grundlagen aus der Kategorientheorie und der (algebraischen) Topologie erfordern. Wir haben uns bemüht diese möglichst vollständig darzustellen. Für zusätzliche Informationen empfiehlt sich [4, 2, 24].

Kapitel 3 wird die Theorie des UMAP Verfahrens darstellen. Dabei werden wir auf die in Kapitel 2 gelegten Grundlagen zurückgreifen. Zusätzlich soll argumentiert werden, dass eine praktische Implementierung wie sie in der Theorie beschrieben ist zu rechenaufwendig ist und somit wenig praktischen nutzen hat. Deshalb

Vergleich mit Original Paper

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 $d : X \times X \to \mathbb{R}_{\geq 0}$ 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 $d(x, y) = 0 \Leftarrow x = y$ 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:

$$B_r(p) :- \{x \in X | d(x, p) < r\}. \tag{2.1}$$

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

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

In der Einleitung bereits erwähnt, werden wir Annehmen, 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 $f : X \to Y$ 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 $f : X \to Y$ 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 $q \in V$,

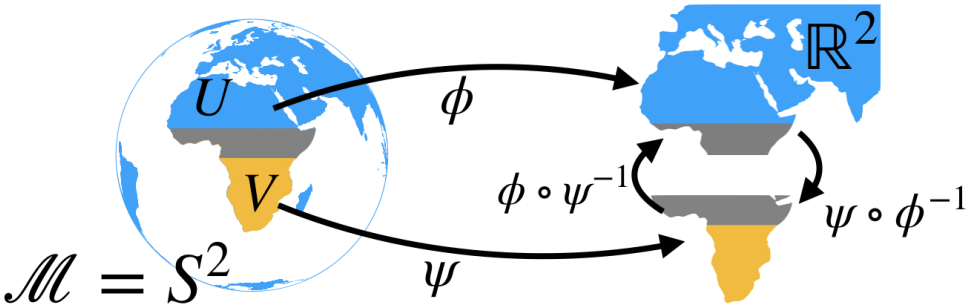

ABBILDUNG 2.1: Atlas einer Mannigfaltigkeit.

2. die Topologie von $\mathcal{M}$ besitzt eine abzählbare Basis
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 $f : U \to V$ 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 sein $\mathcal{M}$ eine Mannigfaltigkeit der Dimension d. Eine *Karte* auf $\mathcal{M}$ ist ein Paar (U, ϕ), wobei $U \subseteq \mathcal{M}$ eine offene Menge und $\phi : U \to \phi(U)$ 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

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

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.

Definition 2.7. Eine *differenzierbare* Mannigfaltigkeit ist eine...

Beispiel 2.1. S^n ist eine n-dimensionale Mannigfaltigkeit im $\mathbb{R}^{n+1}$.

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 erweitern.

Definition 2.8 (Riemannsche Mannigfaltigkeit)**.** Sei g,...

Ein Tupel $(\mathcal{M}, g)$, wobei $\mathcal{M}$ eine Mannigfaltigkeit und g eine riemannsche Metrik ist heißt *riemannsche Mannigfaltigkeit*.

Bemerkung. Eine riemannsche Mannigfaltigkeit ist stets metrisierbar im Sinne, das ...

Die Riemannmetrik beschreibt eine Distanz auf der Mannigfaltigkeit. Die Länge eines kürzesten Weges auf $\mathcal{M}$ zwischen zwei Punkten $p, q \in \mathcal{M}$ wird als Geodäte bezeichnet und ist definiert als

Definition 2.9 (Geodäte)**.** Seien $p, q \in \mathcal{M}$ ...

Der Begriff der riemannschen Mannigfaltigkeit ermöglicht es uns unsere zentrale Annahme, das $X \subseteq \mathbb{R}^D$ einer niedrigdimensionalen Struktur entnommen ist, zu formalisieren. Wir möchten nun diese niedrigdimensionale Struktur genauer beschreiben.

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 Samuel Eilenberg und Saunders Mac Lane eingeführt.
Die Definitionen sind dem Buch von Brandenberg [4] entnommen.

Definition 2.10 (Kategorie)**.** 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 $\mathsf{Hom}_{\mathcal{C}}(A, B)$, deren Elemente wir mit $f : A \to B$ notieren und *Morphismen* von A nach B nennen,
3. zu je drei Objekten $A, B, C \in Ob(\mathcal{C})$ einer Abbildung

   $$\mathsf{Hom}_{\mathcal{C}}(A, B) \times \mathsf{Hom}_{\mathcal{C}}(B, C) \to \mathsf{Hom}_{\mathcal{C}}(A, C)$$

   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

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

   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 $f : A \to B,\ g : B \to C,\ h : C \to D$ in $\mathcal{C}$ gilt

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

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

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

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 $\mathsf{Hom}_{\mathcal{C}}(A, B)$ mit $\mathsf{Hom}(A, B)$ abkürzen.

Bemerkung. Eine Klasse ist eine Menge, welche zu groß ist um eine Menge zu sein. Für eine Definition einer Klasse verweisen wir auf [14]. 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.

Beispiel 2.2. *Passende Beispiele von Kategorien, welche später wieder genutzt werden.* **Top**, **Set**, $(Fin)EPMet$

Ein weiterer für die folgenden Definitionen wichtiger Begriff ist der der dualen Kategorie.

Definition 2.11 (Duale Kategorie)**.** Es sei $\mathcal{C}$ eine Kategorie. Dann können wir eine neue Kategorie $\mathcal{C}^{op}$ konstruieren: Sie besitzt dieselben Objekte wie $\mathcal{C}$, allerdings werden die Morphismen „umgedreht“: Für $A, B \in \mathcal{C}$ sei

$$\mathsf{Hom}_{\mathcal{C}^{op}}(A, B) \coloneqq \mathsf{Hom}_{\mathcal{C}}(B, A).$$

Die Identitäten verändern sich nicht. Die Komposition

$$\circ^{op} : \mathsf{Hom}_{\mathcal{C}^{op}}(A, B) \times \mathsf{Hom}_{\mathcal{C}^{op}}(B, C) \to \mathsf{Hom}_{\mathcal{C}^{op}}(A, C)$$

ist durch

$$\mathsf{Hom}_{\mathcal{C}}(B, A) \times \mathsf{Hom}_{\mathcal{C}}(C, B) \cong \mathsf{Hom}_{\mathcal{C}}(C, B) \times \mathsf{Hom}_{\mathcal{C}}(C, B) \times \mathsf{Hom}_{\mathcal{C}}(B, A) \xrightarrow{\circ} \mathsf{Hom}_{\mathcal{C}}(C, A)$$

definiert, d.h. $f \circ^{op} g \coloneqq g \circ f$. Auf diese Weise ist $\mathcal{C}^{op}$ tatsächlich eine Kategorie und heißt die *zu* $\mathcal{C}$ *duale Kategorie.*

Bemerkung. Eine Eigenschaft der dualen Kategorie ist, dass Aussagen, welche für alle Kategorien bewiesen wurden, auch für alle dualen Kategorien gelten.

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 2.12 (Funktor)**.** Es seien $\mathcal{C}$ und $\mathcal{D}$ zwei Kategorien. Ein *Funktor*

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

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 $f : A \to B$ in $\mathcal{C}$ einen Morphismus

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

   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 $f : A \to B,\ g : B \to C$ in $\mathcal{C}$ gilt in $\mathcal{D}$:

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

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

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

$$\mathsf{Hom}(-, B) : \mathcal{C} \to \mathbf{Set} \tag{2.4}$$

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

$$\mathsf{Hom}(h, B) : \mathsf{Hom}(C, B) \to \mathsf{Hom}(A, B), \text{wobei } g \mapsto g \circ h \text{ für } g \in \mathsf{Hom}(C, B) \tag{2.5}$$

Eine häufig verwendete Form eines kontravarianten Funktors ist die Prägarbe (*engl.: presheaf*). Wir werden diesen Funktor später verwenden um *simpliziale Mengen* einzuführen.

Definition 2.13 (Prägarbe)**.** Eine Prägarbe auf einer kleinen Kategorie $\mathcal{C}$ ist ein Funktor

$$F : \mathcal{C}^{op} \to \mathbf{Set}$$

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

Definition 2.14 (Prägarbenkategorie)**.** Sei $\widehat{\mathcal{C}}$ die Prägarbenkategorie einer Kategorie $\mathcal{C}$: Objekte sind die Funktoren $F : \mathcal{C}^{op} \to \mathbf{Set}$, und Morphismen sind natürliche Transformationen der Funktoren.

Bemerkung. Allgemeiner kann man auch die Kategorie $[\mathcal{C}, \mathcal{D}]$ einführen, deren Objekte Funktoren $F : \mathcal{C} \to \mathcal{D}$ sind und deren Morphismen ebenfalls natürliche Transformationen der Funktoren sind.

Yoneda Lemma. *Random Text*

Definition 2.15 (Adjunktion)**.**

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 [8] 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 $\{v_0, \ldots, v_n\}$ aufgespannte konvexe Hülle im euklidischen Raum. Die geometrische Unabhängigkeit der Vektoren bedeutet dabei, das $v_1 - v_0, \ldots v_n - v_0$ linear unabhängig sind. Die Vektoren v_i werden *Knoten* genannt und die Teilmengen von $\{v_0, \ldots, v_n\}$, *Facetten* des n-Simplex.

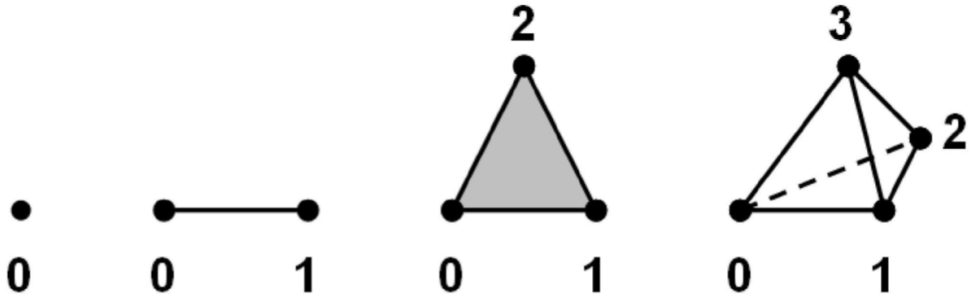

ABBILDUNG 2.2: Die geometrischen 0-, 1-, 2- und 3-Simplizes, in manchen Fällen werden diese auch als (geordnete) Standardsimplizes bezeichnet.

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

Dabei ist $[n] := \{1, \ldots, n\}$.

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 $|\cdot| : \Delta \to \mathbf{Top}$, gegeben durch

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

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

Definition 2.18. Eine *simpliziale Menge* ist ein Funktor $X : \Delta^{op} \to \mathbf{Set}$. 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

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

Bemerkung. Simpliziale Mengen sind also Hom-Funktoren (siehe Gleichung (2.4)).

Definition 2.19. Die Objekte der Kategorie der simplizialen Mengen **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 *zusammengefügt* wird. Die Simplizes dürfen dabei nur entlang ihrer Facetten *zusammengefügt* werden.

Ein Simplizialkomplex kann dabei helfen einen topologischen Raum 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 *abstrakter Simplizialkomplex* besteht aus einer Menge S^0 an *Knoten* und, für jedes k, einer Menge S^k, bestehend aus Teilmengen von S^0 der Kardinalität $k+1$, so dass jede $(i+1)$-elementige Teilmenge einer Menge aus S^k auch in S^i ist. Die Menge S^k wird *k-Skelett* genannt.

Bemerkung. Analog zum k-Skelett eines Simplizialkomplexes werden wir im folgenden die k-Simplizes einer simplizialen Menge als k-Skelett bezeichnen. Für simpliziale Mengen wird diese Bezeichnung nicht konsistent in der Literatur genutzt.

Abstrakte Simplizialkomplexe 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 [8].

Eine hilfreiche Eigenschaft (abstrakter) Simplizialkomplexe 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 $x : \Delta^n \to X$. Eine Anwendung von [20] (§7, Thm. 1) liefert uns:

$$X \simeq \varinjlim \Delta^n, \tag{2.6}$$

wobei der Kolimit über eine von X bestimmte Indexkategorie genommen wird.

Bemerkung. Eine mathematische Definition des Kolimes wird in dieser Arbeit nicht gegeben, da diese einige Vorbereitungen benötigen würde. Wir verweisen den Leser auf geeignete Literatur, beispielsweise [4].

Dennoch soll kurz erläutert werden wie der Kolimes zu verstehen ist. Er ist das duale Konstrukt zum Limes, deshalb operiert er auf der dualen Kategorie. Anschaulich werden die Objekte (hier die Δ^n) in einer passenden Weise *zusammengefügt*. In Gleichung (2.6) bedeutet dies also, dass sich eine simpliziale Menge aus den Standardsimplizes zusammensetzt.

Wie bereits erwähnt lassen sich für topologische Räume geeignete (abstrakte) Simplizialkomplexe 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:*

$$|\cdot| : \mathbf{sSet} \to \mathbf{Top}, \quad |X| \mapsto \varinjlim |\Delta^n|, \tag{2.7}$$

und der singuläre Mengen Funktor

$$S : \mathbf{Top} \to \mathbf{sSet}, \textit{ mit } \quad S(Y) : [n] \to \mathsf{Hom}_{\mathbf{Top}}(|\Delta^n|, Y), \tag{2.8}$$

bilden eine Adjunktion.

Auf diese Adjunktion werden wir in Kapitel 3 zurückkommen und diese für metrische Räume anpassen.

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 diese indirekt eine simpliziale Menge konstruieren. Dazu möchten wir auch die Eigenschaft nutzen, dass für die Daten eine Metrik gegeben ist. 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 $x \in X \text{oder} x \notin X$. Eine *unscharfe Menge* verallgemeinert die Zugehörigkeit.

Definition 2.22.

Kapitel 3

UMAP

In diesem Kapitel soll das UMAP Verfahren eingeführt werden. Dabei wird angenommen, das 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. [21] 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 [21]. 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 $d(\mathbf{x}_i, \mathbf{x}_{i_1}) \leq \cdots \leq d(\mathbf{x}_i, \mathbf{x}_{i_j}) \leq \cdots \leq d(\mathbf{x}_i, \mathbf{x}_{i_N})$. Die k-nächsten-Nachbarn eines Punktes sind somit die 1-, $\ldots$, 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

$$\tilde{d}_i(x, y) \coloneqq \frac{d(x, y)}{k_{x_i}}, \tag{3.1}$$

dabei bezeichnet k_{x_i} den k-ten Nachbarn von $\mathbf{x}_i$. 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 $\mathbf{x}_j$, mit $h, j \leq k$, nach Lemma 3.1 $\tilde{d}_i$ nur für die Paare $(\mathbf{x}_i, \mathbf{x}_h), (\mathbf{x}_i, \mathbf{x}_j)$ die Geodäte angibt. Wir setzen,

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

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, $\ldots$, 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 normalisiert werden. Dies wenden wir auf unsere erweiterten Metriken $\bar{d}_i$ an und erhalten erweiterte Pseudometriken,

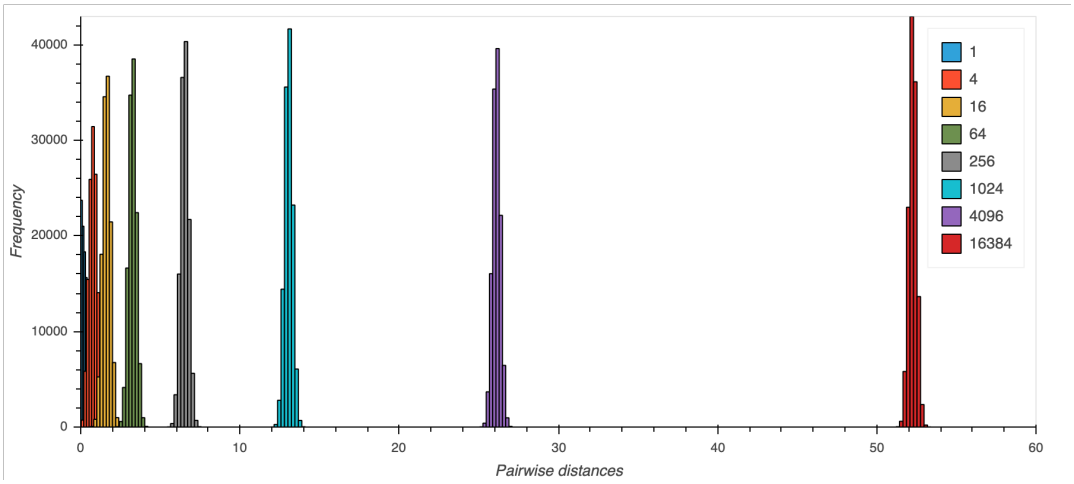

ABBILDUNG 3.1: Paarweise Distanzen von $N = 500$ zufällig gleichverteilten Punkten im R^D.

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

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 $d_{\mathcal{M}}$ bekannt ist, setzen wir in Gleichung (3.1) $\tilde{d}_i \coloneqq d_{\mathcal{M}}$ 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 $\mathsf{FinReal} : \mathbf{Fin\text{-}sFuzz} \to \mathbf{Fin\text{-}EPMet}$ ist gegeben durch

$$\mathsf{FinReal}(X) \coloneqq \varinjlim \mathsf{FinReal}(\Delta^n_{<a}), \tag{3.4}$$

wobei,

$$\mathsf{FinReal}(\Delta^n_{<a}) \coloneqq (\{x_1, \dots, x_n\}, d_a), \tag{3.5}$$

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

Die Wirkung des Funktors FinReal auf einem Morphismus $\Delta^n_{<a} \to \Delta^m_{<b}$, mit $a \leq b$ und $\sigma : \Delta^n \to \Delta^m$, 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** $\to$ **Fin-EPMet**, *wobei für* $Y \in$ **Fin-EPMet** *gilt,*

$$\mathsf{FinSing}(Y) : ([n], [0, a)) \to \mathsf{Hom}_{\mathbf{Fin\text{-}EPMet}}(\mathsf{FinReal}(\Delta^n_{<a}), Y), \tag{3.7}$$

sind zueinander adjungiert.

Bemerkung. Ein Beweis findet sich in [21]. 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 (2.6) 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

$$\perp_{i=1}^{N} \mathsf{FinSing}((X, d_i)). \tag{3.8}$$

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. 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. Dies spiegelt Aussage des Lemma 3.1 wieder, dass wir nur lokal zu $\mathbf{x}_i$ die Geodäte bestimmen können.

Die Vereinigung der unscharfen simplizialen Mengen in Gleichung (3.8) lässt sich wie folgt für den stark vereinfachten Fall des 1-Skelett der Simplizialkomplexe $\mathcal{K}_i$ beschreiben. Die 0-Simplizes bleiben unverändert.

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

3.3 Einbettung

In diesem Abschnitt soll eine

Um die niedrigdimensionale Repräsentation der hochdimensionalen anzupassen wird für das UMAP Verfahren die Kreuzentropie zwischen zwei unscharfen (simplizialen) Mengen vorgeschlagen.

$$C((A,\mu),(A,\nu)) \coloneqq \sum_{a\in A}\left(\mu(a)\log\left(\frac{\mu(a)}{\nu(a)}\right)+(1-\mu(a))\log\left(\frac{1-\mu(a)}{1-\nu(a)}\right)\right) \tag{3.9}$$

In Abschnitt 4.1 werden wir die Kreuzentropie genauer betrachten.

Kapitel 4

Implementierung

In diesem Kapitel möchten wir die Implementierung des UMAP Verfahren beschreiben. Die vollständige Berechnung aller Simplizes hat eine exponentielle Laufzeit, da hierfür alle 2^N Teilmengen unseres N-elementigen Datensatzes betrachtet werden müssten. In aktuellen Implementierungen werden hingegen nur alle zweielementigen Teilmengen betrachtet. Wie wir in Kapitel 5 sehen werden, liefert uns diese Approximation des Čech-Komplexes sehr gute visuelle Ergebnisse. In Kapitel 6 werden wir Ansätze erwähnen um Simplizes höherer Ordnungen effizient zu finden.

Zunächst werden wir die Notation aus Kapitel 3 anpassen, da wir nur das 1-Skelett der topologischen Repräsentationen betrachten. Danach werden wir die Zielfunktion explizit angeben und den Gradienten herleiten. In Abschnitt 4.2 werden wir den UMAP Algorithmus im Pseudo-Code angeben. Die darauf folgenden Abschnitte identifizieren rechenintensive Schritte der von uns verwendeten Implementierung [22] und beschreiben Alternativen. Abschnitt 4.7 gibt eine Beschreibung der Hyperparameter an.

4.1 Numerische Formulierung der Optimierung

Um eine Unterscheidung zwischen der theoretischen Sichtweise auf das UMAP Verfahren, welche alle k-Simplizes ($1 \leq k \leq N$) berücksichtigt, und der praktischen Implementierung zu verdeutlichen, werden wir die verwendete Notation anpassen.

Die unsicheren Mengen $(A, \mu), (A, \nu)$ aus Gleichung (3.9) lassen sich als gewichtete Graphen, in Form einer Adjazenzmatrix darstellen. Das 1-Skelett der hochdimensionalen Repräsentation notieren wir als Adjazenzmatrix V mit $v_{ij} = \mu(a)$, für ein 1-Simplex a, mit Facetten i und j.

Die Wahl des Zugehörigkeitsgrades für 1-Simplizes der niedrigdimensionalen Repräsentation der Daten ist wie folgt gegeben:

$$w_{ij} = (1 + a(\|\mathbf{y}_i - \mathbf{y}_j\|_2^2)^b)^{-1}, \tag{4.1}$$

wobei $\mathbf{y}_i, 1 \leq i \leq N$ die Vektoren der Einbettung sind. Die Wahl der Werte a, b werden wir in Abschnitt 4.7 beschreiben.

Somit erhalten wir folgende Kreuzentropie zwischen den Graphen der hoch- und niedrigdimensionalen Darstellung der Daten:

$$C(V,W) = \sum_{i,j=1}^{N} v_{ij} \log\left(\frac{v_{ij}}{w_{ij}}\right) + (1-v_{ij}) \log\left(\frac{1-v_{ij}}{1-w_{ij}}\right) \tag{4.2}$$

$$= \sum_{i,j=1}^{N} (v_{ij} \log(v_{ij}) + (1-v_{ij}) \log(1-v_{ij})) \tag{4.3}$$

$$- \sum_{i,j=1}^{N} (v_{ij} \log(w_{ij})) - \sum_{i,j=1}^{N} ((1-v_{ij}) \log(1-w_{ij}))$$

$$= C_v - \sum_{i,j=1}^{N} (v_{ij} \log(w_{ij}) + (1-v_{ij}) \log(1-w_{ij})) \tag{4.4}$$

Der Term C_v bleibt während der Minimierung der Funktion bezüglich der $\mathbf{y}_i$ konstant.

Aufgrund der Wahl einer differenzierbaren Funktion für den Zugehörigkeitsgrad der 1-Simplizes, bzw. der Gewichte des niedrigdimensionalen Graphen, lässt sich nun der Gradient der Zielfunktion (4.4) herleiten.

Mittels der Kettenregel ergibt sich mit $d_{ij} \coloneqq \|\mathbf{y}_i - \mathbf{y}_j\|_2$:

$$\frac{\partial C}{\partial \mathbf{y}_i} = \sum_{k,l=1}^{N} \frac{\partial C}{\partial w_{kl}} \sum_{m,n=1}^{N} \frac{\partial w_{kl}}{\partial d_{mn}^2} \sum_{p,q=1}^{N} \frac{\partial d_{mn}^2}{\partial d_{pq}} \frac{\partial d_{pq}}{\partial \mathbf{y}_i} \tag{4.5}$$

$$= \sum_{k,l=1}^{N} \frac{\partial C}{\partial w_{kl}} \sum_{m,n=1}^{N} \frac{\partial w_{kl}}{\partial d_{mn}^2} \frac{\partial d_{mn}^2}{\partial d_{mn}} \frac{\partial d_{mn}}{\partial \mathbf{y}_i} \tag{4.6}$$

$$= 2 \sum_{k,l=1}^{N} \frac{\partial C}{\partial w_{kl}} \sum_{m=1}^{N} \frac{\partial w_{kl}}{\partial d_{mi}^2} \frac{\partial d_{mi}^2}{\partial d_{mi}} \frac{\partial d_{mi}}{\partial \mathbf{y}_i} \tag{4.7}$$

$$= 2 \sum_{m=1}^{N} \left(\sum_{k,l=1}^{N} \frac{\partial C}{\partial w_{kl}} \frac{\partial w_{kl}}{\partial d_{mi}^2} \right) \frac{\partial d_{mi}^2}{\partial d_{mi}} \frac{\partial d_{mi}}{\partial \mathbf{y}_i} \tag{4.8}$$

$$= 4 \sum_{m=1}^{N} \left(\sum_{k,l=1}^{N} \frac{\partial C}{\partial w_{kl}} \frac{\partial w_{kl}}{\partial d_{mi}^2} \right) (\mathbf{y}_i - \mathbf{y}_m) \tag{4.9}$$

Bei der Umformung von (4.8) nach (4.9) haben wir verwendet, dass d_{mi} die euklidische Norm ist. Für Einbettungen in andere metrische Räume müsste der Gradient an dieser Stelle entsprechend angepasst werden. Da wir das UMAP Verfahren im wesentlichen zur Visualisierung im $\mathbb{R}^2$ benutzen ist die Wahl der euklidischen Norm gerechtfertigt. Die übrigen Umformungen ergeben sich aus umordnen und wegfallen der Terme.

(4.9) lässt sich weiter umformen, dazu nutzen wir:

$$\frac{\partial C}{\partial w_{ij}} = -\frac{v_{ij}}{w_{ij}} + \frac{1-v_{ij}}{1-w_{ij}} = \frac{w_{ij} - v_{ij}}{w_{ij}(1-w_{ij})} \tag{4.10}$$

$$\frac{\partial w_{ij}}{\partial d_{ij}^2} = -bad_{ij}^{2(b-1)} w_{ij}^2 = -\frac{b}{d_{ij}^2} w_{ij}(1-w_{ij}) \tag{4.11}$$

Der UMAP Gradient ist also gegeben durch:

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_{j=1}^{N} (w_{ij} - v_{ij}) \frac{b}{d_{ij}^2} (\mathbf{y}_i - \mathbf{y}_j) \tag{4.12}$$

4.2 Pseudo-Code

Nun können wir das UMAP Verfahren für die 1-Skelette der topologischen Repräsentation angeben.

Algorithm 1 UMAP Algorithmus

1: **function** UMAP($X, N, D, d, min\_dist, n\_epochs$)
2: **for** $x \in X$ **do**
3: $knn(x) \leftarrow k$-nächste-Nachbarn(x)
4: $graph(x) \leftarrow \perp_{y \in knn(x)}(\{x, y\}, exp(-d_{x,y})) \perp \perp_{y \in X \setminus \{x\}}(\{x, y\}, 0)$
5: $V \leftarrow$ gewichtete Adjazenzmatrix($\bigcup_{x \in X} graph(x)$)
6: $D \leftarrow$ Grad-Matrix des Graphen V
7: $L \leftarrow D^{1/2}(D - V)D^{1/2}$ → Symmetrische normalisierte Laplace-Matrix
8: $evec \leftarrow$ sortierte Eigenvektoren von L
9: $Y \leftarrow evec[1, \ldots, d{+}1]$
10: $Y \leftarrow$ OptimiereEinbettung(Y, min_dist, n_epochs) → siehe Algorithmus 2
11: **return** Y

In Abschnitt 4.6 werden zwei effiziente Verfahren für die k-nächste-Nachbarn Suche (Zeile 3) angeben.

Der in Zeile 4 verwendete $\perp$ Operator verdeutlicht, dass wir die Kantengewichte mittels wahrscheinlichkeitstheoretischer t-Conorm vereinigen. In den von uns verwendeten Implementierung des UMAP Verfahrens [22, 26] wird ein zusätzlicher Hyperparameter verwendet, welcher eine Verallgemeinerung der Vereinigung darstellt (siehe: `set_op_mix_ratio` in Abschnitt 4.7). Zusätzlich ist zu beachten, dass die Vereinigung in Zeile 4 ungerichtete Kanten betrachtet. Der so erhaltene ungerichtete Graph besitzt aufgrund der Symmetrie der t-Conorm wohldefinierte Kantengewichte.

Die Grad-Matrix D (Zeile 6) ist eine Diagonalmatrix, wobei $d_{ii} \coloneqq \sum_{1 \leq k \leq N} v_{ik}, (1 \leq i \leq N)$. Für den Spezialfall, der ungewichteten Adjazenzmatrix, beschreibt d_{ii} also den Grad des Knoten i.

Die Initialisierung der niedrigdimensionalen Repräsentation Y in Zeile 9 ist die spektrale Einbettung des Graphen. Eine genauere Beschreibung warum die so gewählte Initialisierung sinnvoll ist findet sich in Abschnitt 4.3. Dabei ist zu beachten, dass eine Lösung des Eigenwertproblems nur von der Größe der Einbettungsdimension abhängig ist. Da wird stets $d \ll D$ betrachten ist eine effiziente Implementierung mittels sukzessiver Eigenwertsuche möglich.

4.3 Spektrale Einbettung

Der Laplace Operator ist eine diskrete Approximation des Laplace-Beltrami Operators. Spektrale Einbettung ...

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.

4.4 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 Verfahren für hochdimensionale Daten stark zu beeinflussen. Wir werden beide rechenintensiven Subroutinen im folgenden betrachten und geeignete Verbesserungen diskutieren.

4.5 Gradientenverfahren

Um die Zielfunktion (4.4) zu optimieren bietet sich die Wahl eines Gradientenverfahrens an, da eine differenzierbare Approximation des Zugehörigkeitsgrades (siehe Gleichung 4.1) 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 Backpropgation genutzt. In [17] werden verschiedene Implementierungen verglichen.

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 ($v_{ij} \approx 1$) ist der Term $(1 - v_{ij}) \log(1 - w_{ij})$ aus Gleichung (4.4) nahe Null, deshalb ist es sinnvoll nur den Gradienten des Terms $v_{ij} \log(w_{ij})$ zu betrachten. Für Kanten mit $v_{ij} \approx 0$) sollte hingegen der Gradient des Terms $(1 - v_{ij}) \log(1 - w_{ij})$ betrachtet werden. Für das UMAP Verfahren wird deswegen folgende Implementierung vorgeschlagen:

Die in Zeile 5 beschriebene Ziehung der Stichprobe dient dazu in Zeile 6 keine zusätzliche Multiplikation mit v_{ij} zu machen. Diese Idee entstammt [32]. Dadurch wird der Gradient nicht durch den Wert von v_{ij} beeinflusst.

Algorithm 2 Optimiere die Einbettung mittels modifiziertem SGD

```
1:  function OptimiereEinbettung(Y, V, W, n_epochs)
2:      α ← 1.0
3:      for e ← 1, ..., n_epochs do
4:          for all v_ij do
5:              if Random() ≤ v_ij then
6:                  y_i ← y_i + α · ∇(log(w_ij))
7:                  for l ← 1, ..., n_neg_samples do
8:                      m ← Unif((0, N))
9:                      y_i ← y_i + αγ · ∇(log(1 − w_im))
10:     α ← 1.0 − e/n_epochs
11:     return Y
```

Das in jedem Durchlauf des modifizierten SGD mehrere *negative samples* betrachtet werden geht auch [23] zurück. Im wesentlichen verhindert dies, das sich die Vektoren der niedrigdimensionalen Darstellung häufen. Die dabei $(0, N)$-gleichverteilt gezogene Stichprobe ist eine Modifizierung von [33].

Der Gradient in Zeile 6 ist gegeben durch:

$$\nabla(\log(w_{ij})) = \frac{-2ab\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2(b-1)}}{(1 + a(\|\mathbf{y}_i - \mathbf{y}_j\|_2^2)^b)^{-1}}(\mathbf{y}_i - \mathbf{y}_j) \tag{4.13}$$

und der Gradient in Zeile 9 durch:

$$\nabla(1 - \log(w_{ij})) = \frac{2b}{(\epsilon + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2)(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2b})}(\mathbf{y}_i - \mathbf{y}_j), \tag{4.14}$$

wobei der ϵ-Parameter eine Division mit Null vermeidet.

4.6 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 in N lineare Laufzeit möglich. Die Herangehensweisen lassen sich nach [33] 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 [6] 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 Nachbarschaftspaare $(x, y), (y, z)$ 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 [9] dadurch verbessert werden, indem für die Initialisierung „random projection trees“, wie in [33], 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 [21] wird jedoch argumentiert, dass dies wegen des Informationsverlust bei Dimensionsreduktionen kaum Auswirkungen auf die resultierende Einbettung hat.

FAISS

Das FAISS Bibliothek [13] 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.

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.

4.7 Hyperparameter

- `n_neighbors`: beschreibender Text
- `metric` Text
- `n_epochs` Text
- `set_op_mix_ratio`
- `local_connectivity`
- `repulsion_strength`
- `a, b`: Erwähne wie a und b standardmäßig gewählt werden, zeige Plot von a, b Erwähne `min_dist` und `spread`
- `negative_sample_weight`

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 GitHub [1].

5.1 Alternative Verfahren

Wir haben uns dazu entschieden UMAP mit folgenden Verfahren zu vergleichen:

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 [19] ist zur Zeit eines der bekanntesten und meistgenutzten nichtlinearen Dimensionsreduktionsverfahren. Dabei wird UMAP 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` 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 [12]) 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 [31, 16]. Dabei ist besonders Barnes-Hut-SNE [18] zu erwähnen, allerdings sollte hier beachtet werden, dass aufgrund der Konstruktion einer speziellen Datenstruktur die Laufzeit für $d > 3$ sehr schlecht ist.

[1] https://github.com/reinerschristopher/bachelorthesis

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 [15]. Der `late-exaggeration` wie in [16] 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 [34] - zeigt interaktiv auf künstlich erzeugten Datensätzen die Auswirkung der Parameter.

Für unsere Experimente haben wir die scikit Implementierung des t-SNE Verfahren genutzt [28]. Zusätzlich haben wir die openTSNE [29] Implementierung genutzt. Diese beschleunigt die Laufzeit des t-SNE Algorithmus durch eine zusätzliche Fouriertransformation [16]. Die openTSNE Implementierung besitzt im Vergleich zur scikit Implementierung die Möglichkeit den `late-exaggeration` Parameter zu spezifizieren.

5.1.2 TriMap

Das TriMap Verfahren [1] soll eine globalere Repräsentation der Daten finden als beispielsweise t-SNE, das 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.

Wir haben diesem Algorithmus gewühlt, 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 d-dimensionalen Repräsentationen der Daten zu bewerten gibt es verschiedene Ansätze. Diese sollen nun vorgestellt und verglichen werden.

Ein DR-Verfahren, welches die statistischen Eigenschaften der zugrundeliegenden Daten erfasst, sollte invariant bezüglich Rauschen der Daten sein. Hingegen sollte eine schlechte Einbettung wenig stabil sein, wenn zusätzliches Rauschen in den Daten auftritt.

Eine weitere Methodik um die Qualität der Einbettung zu erfassen, erhält man dadurch, dass

In [11] werden zwei Methoden zur Auswahl eines geeigneten DR-Verfahrens verglichen.

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.

Abbildung 5.1: Sechs zufällig gewählte Gesichter des Cartoon Set.

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. Beispielsweise sollten die Cluster welche die Ziffern 1 und 7 darstellen näher zueinander liegen als die Cluster der Ziffern 1 und 6.

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.

In den folgenden Experimenten haben wir uns auf eine qualitative Analyse der Daten beschränkt. Einerseits ist dies unserer Erfahrung nach (und Einträgen in Internetforen, ...) die meistgenutzte Art die Repräsentation in der Praxis zu bewerten. Zusätzlich bietet die Wahl der Datensätze eine gute Gelegenheit die Repräsentationen visuell zu bewerten.

5.3 Cartoon Set

In diesem Abschnitt werden wir den *Cartoon Set* Datensatz analysieren [5]. Dabei werden wir:

- sehen, dass UMAP eine vergleichbare Laufzeit mit der von FIT-SNE hat
- das Verhalten der niedrigdimensionalen Darstellung unter verschiedenen Hyperparametern betrachten
- eine exemplarische Beschreibung der Hyperparameter geben
- UMAP mit anderen Dimensionsreduktionsverfahren vergleichen
- sehen, dass UMAP zugrundeliegende Mannigfaltigkeiten erkennt und darstellt

Beschreibung des Datensatzes

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

Die Bilder wurden aus 16 Komponenten zusammengesetzt (u.a. Gesichtsform, Gesichtsfarbe, Frisur, Haarfarbe), dabei variiert die Anzahl der Möglichkeiten pro Komponente 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.

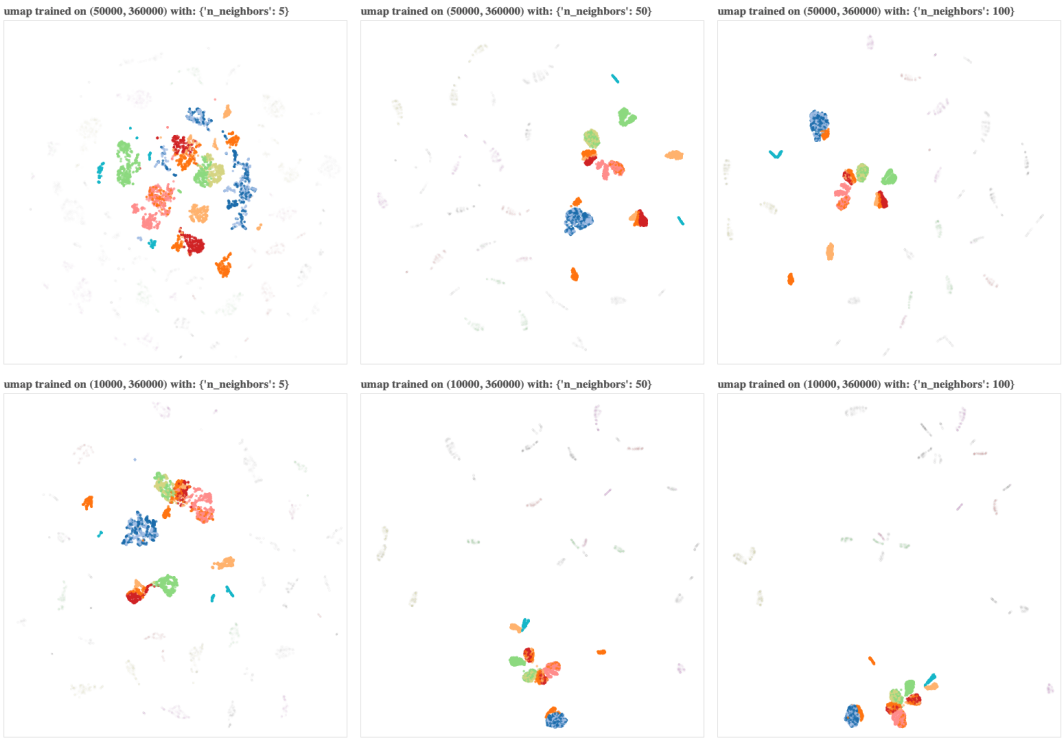

Abbildung 5.2: TODO: Beschreibung des Bildes

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.

Wir haben uns für diesen Datensatz entschieden um UMAP auf Daten mit einer komplexeren Struktur zu testen als dies in [21] gemacht wird. Dabei ist auch zu beachten, dass es aufgrund der 16 Komponenten aus welchen die Gesichter bestehen kein richtige oder falsche Einbettung der Daten gibt.

Wir haben die Bewertung der Einbettung unter der Annahme gemacht, dass *ähnliche* Gesichter *ähnliche* Hautfarben, Frisuren, Haarfarben, Brillen und Bärte haben. Diese fünf Eigenschaften möchten wir besonders hervorheben, da sie die dominantesten Merkmale des Gesichts beschreiben.

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. Weiterhin haben wir uns für die Bewertung der Einbettung auf 10 000 Bilder beschränkt. Somit beträgt die Dimension des Cartoon Set $D = 360\,000$ und die Anzahl an Beispielen $N = 100\,000$.

Qualitative Analyse der Ergebnisse

5.4 MNIST

FMNIST

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. Insgesamt haben wir über 100 Experimente gemacht um genauere Aussagen über die Laufzeit der Verfahren zu treffen und diese in Abhängigkeit der wichtigen Parameter zu setzen.

5.6 Stabilität unter sub-sampling

5.7 Zusammenfassung der Ergebnisse

Kapitel 6

Zusammenfassung und Ausblick

Anhang A

Appendix A

A.1 Homologie

Eine bekannte Aussage, welche der Topologie entstammt, ist, dass eine Kaffeetasse das Gleiche *topologische Objekt* wie ein Donut ist. Dies ist dadurch begründet, dass man eine Kaffeetasse durch strecken und stauchen (ohne reißen) in einen Donut transformieren kann. Diese *Gleichheit* lässt sich wie folgt mathematisch darstellen:

Definition A.1 (Stetige Abbildung)**.** Sei ...

Definition A.2 (Homöomorphismus)**.** Sei ...

Nun werden wir den Begriff der Homologie einführen. Dafür werden wir den Begriff der simplizialen Komplexe benötigen um zuerst die simpliziale Homologie einzuführen.

Definition A.3 (Simplizialer Komplex)**.** Sei ...

Definition A.4 (Abstrakter simplizialer Komplex)**.** Sei ...

Beispiel A.1. Čech-Komplex

Beispiel A.2. VR-Komplex

Definition A.5 (Homologie)**.** Sei ...

Definition A.6 (Homologiegruppe)**.** Sei ...

Beispiel A.3. Die Homologiegruppen eines Balls, Donuts, Graphen, ...

Definition A.7 (Filtration)**.** Sei $T \subseteq \mathbb{R}$, eine *Filtration* $\mathcal{X}$ über T ist eine Familie von topologischen Räumen $\{X_i\}_{i \in T}$, so dass $X_i \subseteq X_j$ für $i \leq j \in T$.

Bemerkung. Eine Filtration $\mathcal{X}$ wird meist nicht als Familie verschiedener topologischer Räume X_i angesehen, sondern als ein einziger Raum, welcher sich im Laufe der Zeit „transformiert“. Siehe dazu [27]. Dies stimmt mit unserem Bild überein, den ϵ-Parameter eines Čech-Komplexes immer größer werden zu lassen. Wir werden *persistente Homologie* nutzen um die homologischen Eigenschaften zu beschreiben, welche für alle $i \in T$ erhalten (bzw. persistent) bleiben.

Beispiel A.4. Insbesondere ist somit der Čech-Komplex eine Filtration, da ...

Beispiel A.5. Eine weitere Filtration ist der Vietoris-Rips-Komplex

Bemerkung. Der VR-Komplex ist ein Beispiel eines Clique-Komplexes und wird als solcher eindeutig durch sein 1-Skelet identifiziert. Sei V ein VR-Komplex und V^1 das zugehörige 1-Skelet. Dann erhält man V aus V^1, indem man alle Cliquen im V^1 zugrundeliegenden Graphen bestimmt. Eine k-Clique ist ein vollständiger Subgraph mit k Knoten.

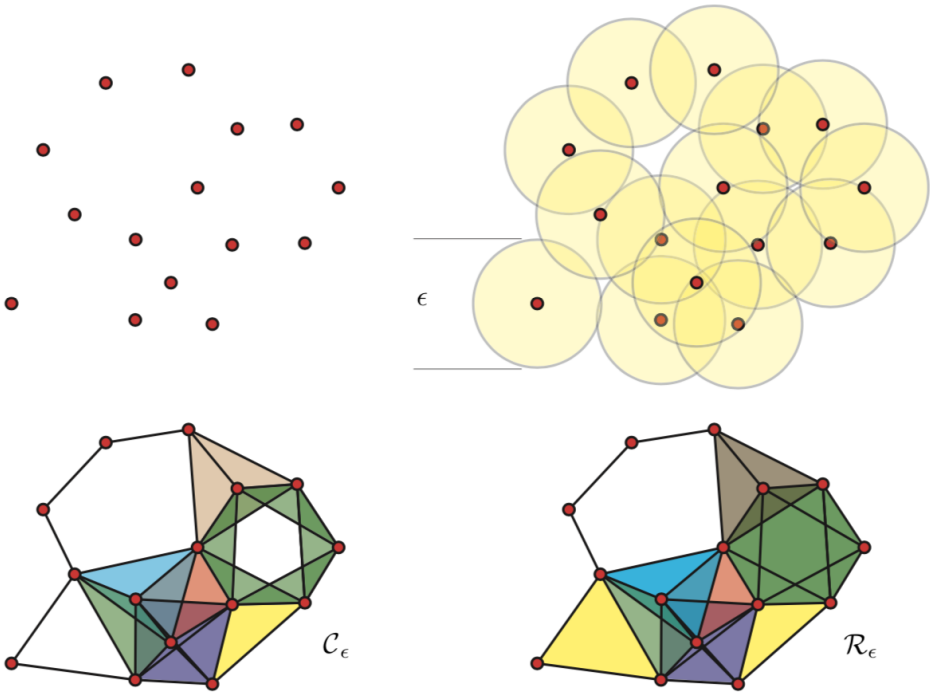

Abbildung A.1: Eine Punktwolke [oben links] kann zu einem Čech-Komplex [unten links] oder einem VR-Komplex [unten rechts] basierend auf dem Parameter ϵ konstruiert werden. Die Homotopietypen der $\epsilon/2$ Überdeckung vom Čech-Komplex $(S^1 \vee S^1 \vee S^1)$, während das VR-Komplex die Homotopietypen $(S^1 \vee S^2)$ besitzt. (*Quelle:* [10])

Es gibt weitere Formen von Komplexen, die bekanntesten sind die CW-Komplexe und Zellkomplexe. Ziel ist es stets mit Hilfe einfacher Konstrukte die Topologie eines Raums zu beschreiben.

Die vom Čech-Komplex und vom Vietoris-Rips-Komplex beschriebene Homologie kann sich unterscheiden (siehe Abbildung A.1).

Wir benötigen nun noch den Begriff einer *guten Überdeckung* um dann eine geeignete Aussage über einen Čech-Komplex einer Menge zu machen.

Definition A.8 (Überdeckung)**.** Eine Überdeckung eines topologischen Raumes ist ... Eine gute Überdeckung ist ...

Nerv Theorem. *Sei ...*

Das Nerv Theorem A.1 liefert uns eine Aussage darüber, dass ein topologischer Raum X zu einer endlichen guten Überdeckung von X homotopieäquivalent ist. Wie wir sehen ist die Konstruktion stark abhängig vom ϵ-Parameter. Um dem entgegenzuwirken nutzt man die Idee der *persistenten Homologie*. Dabei wird betrachtet wie sich die Homologie eines Raumes für eine monoton wachsende Folge $(\epsilon_i)_{i\in\mathbb{N}}$ verhält.

Für den von uns betrachteten Fall, die Struktur einer Riemannschen Mannigfaltigkeit $(\mathcal{M}, g)$ zu beschreiben, werden wir Annehmen, dass die Daten $\mathbf{x}_i \in X$ gleichverteilt bezüglich der Riemannmetrik g sind, bzw. g so wählen, das die Annahme erfüllt ist. Somit erhalten wir eine gute Überdeckung ohne eine Abhängigkeit von ϵ zu haben.

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. 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. URL: http://dx.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. URL: http://dx.doi.org/10.1007/978-3-662-47068-8.

[5] Forrester Cole, Shiraz Fuman und Aaron Sarna. *Cartoon Set*. URL: https://google.github.io/cartoonset/download.html (besucht am 19. 07. 2019).

[6] 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.

[7] Charles Fefferman, Sanjoy Mitter und Hariharan Narayanan. „Testing the manifold hypothesis“. In: *Journal of the American Mathematical Society* 29.4 (Feb. 2016), S. 983–1049. ISSN: 1088-6834. DOI: 10.1090/jams/852. URL: http://dx.doi.org/10.1090/jams/852.

[8] Greg Friedman. „An elementary illustrated introduction to simplicial sets“. In: *arXiv:0809.4221v5* (2016).

[9] 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. URL: http://arxiv.org/abs/1609.07228.

[10] Robert Ghrist. „Barcodes: The persistent topology of data“. In: *Bulletin of the American Mathematical Society* 45.01 (Okt. 2007), S. 61–76. ISSN: 0273-0979. DOI: 10.1090/s0273-0979-07-01191-3. URL: http://dx.doi.org/10.1090/S0273-0979-07-01191-3.

[11] Stefan Harmeling. „Exploring model selection techniques for nonlinear dimensionality reduction“. (EDI-INF-RR-0960). Edinburgh, UK: School of Informatics, University of Edinburgh. 2007.

[12] 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.

[13] Jeff Johnson, Matthijs Douze und Hervé Jégou. „Billion-scale similarity search with GPUs“. In: *arXiv preprint arXiv:1702.08734* (2017).

[14] Paul Blain Levy. „Formulating Categorical Concepts using Classes“. In: (2018).

[15] 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. URL: http://dx.doi.org/10.1137/18m1216134.

[16] 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. URL: http://dx.doi.org/10.1038/s41592-018-0308-4.

[17] 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.

[18] 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.

[19] 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.

[20] Saunders Mac Lane. „Categories for the working mathematician“. In: 2. Aufl. Graduate texts in mathematics. Springer New York, 1998. Kap. 3.

[21] Leland McInnes, John Healy und James Melville. „UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction“. In: ().

[22] Leland McInnes u. a. *UMAP*. 2018. URL: https://github.com/lmcinnes/umap (besucht am 22. 07. 2018).

[23] Tomas Mikolov u. a. „Distributed Representations of Words and Phrases and their Compositionality“. In: *CoRR* abs/1310.4546 (2013). arXiv: 1310.4546. URL: http://arxiv.org/abs/1310.4546.

[24] James R Munkres. *Elements of algebraic topology*. Menlo Park u.a.: Addison-Wesley, 1984. ISBN: 0201045869, 9780201045864.

[25] 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.

[26] Corey J Nolet, Dante Dessavre und Thejaswi Rao. *https://github.com/rapidsai/cuml*. (Besucht am 19. 07. 2019).

[27] Steve Oudot. „Persistence Theory - From Quiver Representations to Data Analysis“. In: *Mathematical surveys and monographs*. Bd. 207. Inria Saclay, Palaiseau, France. American Mathematical Society (AMS), 2015.

[28] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830.

[29] Pavlin Policar. *openTSNE*. URL: https://github.com/pavlin-policar/openTSNE (besucht am 22. 07. 2019).

[30] 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.

[31] 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. URL: http://dx.doi.org/10.1007/978-3-319-68474-1_13.

[32] Jian Tang u. a. „LINE“. In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15* (2015). DOI: 10.1145/2736277.2741093. URL: http://dx.doi.org/10.1145/2736277.2741093.

[33] Jian Tang u. a. „Visualizing Large-scale and High-dimensional Data“. In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, S. 287–297. ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883041. URL: https://doi.org/10.1145/2872427.2883041.

[34] Martin Wattenberg, Fernanda Viégas und Ian Johnson. „How to Use t-SNE Effectively“. In: *Distill* (2016). DOI: 10.23915/distill.00002. URL: http://distill.pub/2016/misread-tsne.