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

Ich bin den Personen sehr dankbar und verbunden welche mich auch dem Weg zu dieser Arbeit begleitet haben.

Zuerst möchte ich Annalena Lange, Lennard Schiefelbein und Nico Gerber für die vielen anregenden Gespräche danken und Hendrik Baers und Tobias Bork, welche die letzten drei Jahre zu einer spannenden und herrausfordernden Studienzeit gemacht haben.

Besonderer Dank gilt Prof. Dr. Jochen Garcke für das sehr interessante Thema und die wegweisende Betreueung. Gerne möchte ich auch Leland McInnes für das persönliche Gespräch in Ottawa, Kanada danken. So konnte ich die Motivation, welche er für das UMAP Verfahren hatte, aus erster Hand erfahren.

Zum Schluss meiner Familie und Freunden für Ihre bedingungslose Unterstützung.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 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 Stituation 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 geschlossene 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 $X = \{x_i\}_{i=1}^N$, $(x_i \in \mathbb{R}^D)$, 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

1.2 Dimensionsreduktion

Sei $X = \{x_i\}_{i=1}^N$, $(x_i \in \mathbb{R}^D)$ 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, $N = 100\,000$, $D = 360\,000$.

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 dimensionsreduktions Verfahren beruhen auf der Annahme, das reale hochdimensionale Daten sich in der Umgebung einer niedrigdimensionalen Mannigfaltigkeit konzentrieren. In der Literatur ist diese Annahme als Mannigfaltigkeits Hypothese (*engl.: manifold hypothesis*) bekannt [15, 19]. Ansätze um einen gegebenen Datensatz auf diese Annahme zu testen finden sich in [4].

1.3 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 [12] beschriebene Theorie ausführlicher beschreiben und in einen allgemeineren Kontext setzen.

1.4 Gliederung

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 mich bemüht diese möglichst vollständig darzustellen. Für zusätliche Informationen empfielt sich [2, 1, 14].

Kapitel 3 wird die Theorie des UMAP Verfahrens darstellen. Dabei werden wir auf die in 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

Kapitel 2

Grundlagen

Um die Geometrie „mathematischer Räume“ zu beschreiben, ist es aus topologischer Sicht ausreichend, die „Löcher“ der Räume zu charakterisieren. Um diese Aussage zu formalisieren, und damit die Grundlage für das UMAP-Verfahren zu legen, werden wir einige Begriffe aus der (algebraischen) Topologie benötigen.

Die grundlegenden Definitionen *topologischer und metrischer Räume* werden uns helfen *(riemannsche) Mannigfaltigkeiten* einzuführen. Der Begriff der Mannigfaltigkeit formalisiert den niedrigdimensionalen Raum auf welchem unsere Daten liegen.

Die geometrische Struktur dieser Räume werden wir mit Hilfe der *Homologie* einführen und sehen wie diese visuelle Eigenschaften beschreibt. Um diese strukturelle Darstellung auf eine endliche Punktwolke anwenden zu können, werden wir den Begriff der *persistenten Homologie* einführen müssen.

Die Begriffe und benötigten Aussagen werden vollständig und anschaulich eingeführt. An einigen Stellen werden wir den Leser auf geeignete ergänzende Literatur verweisen.

2.1 Topologische Räume

Der Grundlegende Begriff der Topologie ist der des topologischen Raumes.

Definition 2.1 (Topologischer Raum)**.** Eine Topologie ist ein Mengensystem T ...

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

Definition 2.2 (Metrischer Raum)**.** Eine Metrik ist eine Abbildung,...

In der Einleitung bereits erwähnt, werden wir Annehmen, dass unsere Daten $X = \{x_i\}_{i=1}^N, (x_i \in \mathbb{R}^D)$ mit D gemessenen Eigenschaften mittels $d, (d \ll D)$ Eigenschaften dargestellt werden können. Dies werden wir formalisieren indem wir Mannigfaltigkeiten einführen. Anschaulich ist eine d-dimensionale Mannigfaltigkeit ein topologischer Raum welcher lokal dem euklidischen Raum $\mathbb{R}^d$ gleicht.

Definition 2.3 (Mannigfaltigkeit)**.** Sei $\mathcal{M}$ ein topologischer Raum,...

Bemerkung. Eine *differenzierbare* Mannigfaltigkeit ist eine...

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

Unter der Annahme, dass wir nur d Eigenschaften benötigen um unsere Daten X darzustellen, können wir sagen, das X eine d-dimensionale Mannigfaltigkeit im $\mathbb{R}^D$ ist. Wir können eine Mannigfaltigkeit $\mathcal{M}$ mit einem Abstandbegriff erweitern. Diese zusätzliche Regularitätseigenschaft ermöglicht es uns von Distanzen, Winkeln und Kurven auf $\mathcal{M}$ zu sprechen. Diesen Abstandsbegriff nennen wir *riemannsche Metrik* und er ist wie folgt definiert:

Definition 2.4 (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 ...

Bemerkung. Es ist sinnvoll in unserem Fall X als riemannsche Mannigfaltigkeit zu betrachten, da wir erwarten, wenn x_i Nahe bei x_j und x_k liegt, dass x_j auch Nahe bei x_j liegt. Wenn X einen Bilddatensatz beschreibt, nehmen wir an, dass eine kleine Veränderung im Bild den Inhalt nicht wesentlich ändert.

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 2.5 (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 niedirgdimensionalen Struktur entnommen ist, zu formalisieren. Wir möchten nun diese niedrigdimensionale Struktur genauer beschreiben.

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

Definition 2.6 (Stetige Abbildung)**.** Sei ...

Definition 2.7 (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 2.8 (Simplizialer Komplex)**.** Sei ...

Definition 2.9 (Abstrakter simplizialer Komplex)**.** Sei ...

Definition 2.10 (Homologie)**.** Sei ...

Definition 2.11 (Homologiegruppe)**.** Sei ...

Beispiel 2.2. Die Homologiegruppen eines Balls, Dounuts, Graphen, ...

Definition 2.12 (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 [16]. 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 2.3. Insbesondere ist somit der Čech-Komplex eine Filtration, da ...

Beispiel 2.4. Eine weitere Filtration ist der Vietoris-Rips-Komplex

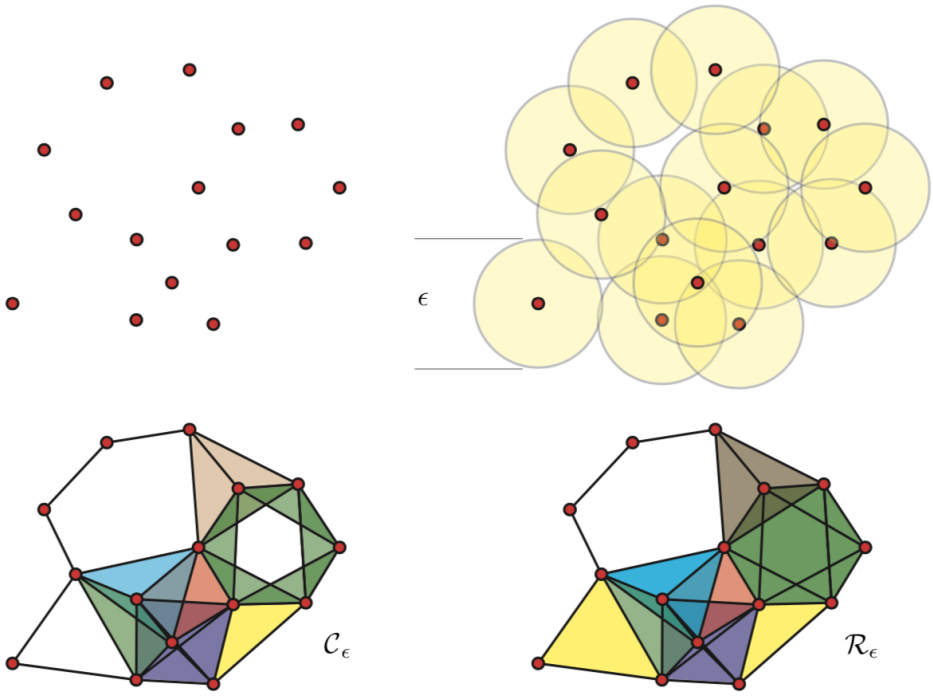

ABBILDUNG 2.1: Eine Punktwolke [oben links] kann zu einem Čech-Komplex [unten links] oder einem VR-Komplex [unten rechts] basierend auf dem Parameter ϵ konstruiert werden. Die Homotopietypen der $\epsilon/2$ Überdeckung vom Čech-Komplex ($S^1 \vee S^1 \vee S^1$), während das VR-Komplex die Homotopietypen ($S^1 \vee S^2$) besitzt. (*Quelle:* [5])

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 erhällt 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.

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.

Nerv Theorem. *Sei ...*

Wie wir sehen ist die Konstruktion stark abhängig vom ϵ-Parameter. Um dem entgegenzuwirken führt man die Idee der *persistenten Homologie* ein. Dabei betrachtet man wie sich die Homologie eines Raumes für eine monoton wachsende Folge $(\epsilon_i)_{i\in\mathbb{N}}$ verhält.

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 1940ern von Samuel Eilenberg und Saunders Mac Lane eingeführt.
Die Definitionen sind dem Buch von Brandenberg [2] entnommen.

Definition 2.13 (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 [7]. 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.

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

$$\mathsf{Hom}(-, B) : \mathcal{C} \to \mathbf{Set} \tag{2.1}$$

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

$$\mathsf{Hom}(h, B) : \mathsf{Hom}(C, B) \to \mathsf{Hom}(A, B), \text{wobei } g \mapsto g \circ h \text{ für } g \in \mathsf{Hom}(C, B) \tag{2.2}$$

Beispiel 2.5. *Passende Beispiele von Kategorien, welche später wieder genutzt werden.* **Top**, **Set**

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

Definition 2.14 (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

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

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. $f \circ^{op} g := g \circ f$. 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 2.15 (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.

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

Definition 2.16 (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 2.17 (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.

2.3 Simpliziale Mengen

Simpliziale Mengen,...

Definition 2.18 (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.

Wir können die n-elementigen Mengen in Δ als geometrische n-Simplizes auffassen. Dazu betrachten wir den Funktor $|\cdot| : \Delta \to \mathbf{Top}$, gegeben durch

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

Definition 2.19. 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

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

Der Standardsimplex ist somit ein Hom-Funktor (siehe 2.1).

Kapitel 3

UMAP

In diesem Kapitel werden wir die aus 2 erlangten Grundlagen verwenden um eine geeignete topologische Repräsentation unserer Daten $X = \{x_i\}_{i=1}^{N}$, $(x_i \in \mathbb{R}^D)$ zu erlangen. Eine ähnliche Repräsentation werden wir von einem d-dimensionalen Raum $(d \ll D)$ betrachten, um dann mit einem geeigneten Begriff des Abstandes der beiden Repräsentationen die niedrigdimensionale Repräsentation der hochdimensionalen anzupassen. Dies wird uns zum UMAP Verfahren führen.

3.1 Topologische Repräsentation

Blabla bla

Fluch der Dimensionen

Ein kritischer Punkt beim analysieren hochdimensionaler Daten ist der sogenannte Fluch der Dimensionen. Was das bedeutet und wie das UMAP Verfahren macht um diesen zu vermeiden werde ich im Folgenden kurz erklären. Unter dem Fluch der Dimensionen versteht man.

Siehe Abbildung 3.1

3.2 Ergänzungen

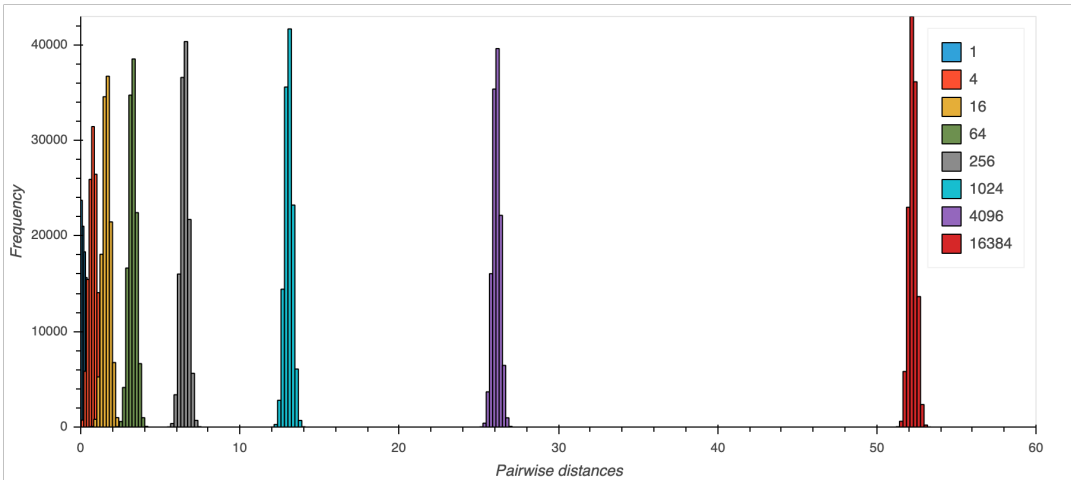

ABBILDUNG 3.1: Paarweise Distanzen von $N = 500$ zufällig gleichverteilten Punkten im R^D.

Kapitel 4

Implementierung

In diesem Kapitel möchten wir die Implementierung des UMAP Verfahren beschreiben. Die vollständige Berechnung aller Simplizes hat eine exponnentielle Laufzeit, da hierfür alle Teilmengen unseres N-elementigen Datensatzes betrachtet werden müssten. In der Implementierung von Leland McInnes et. al. [13] werden hingegen nur alle zwei-elementigen Teilmengen betrachtet. Wie wir in Kapitel 5 sehen werden, liefert uns diese Approximation des Čech-Komplexes sehr gute visuelle Ergebbnisse. Zuerst werden wir den Algorithmus mit seinen Subroutinen in Pseudo-Code angeben. Danach werden wir Ansätze nennen um die Lücke zwischen Theorie und Praxis zu schließen. Zusätzlich werden wir die rechenintensiven Schritte des Verfahrens betrachten und eine effizientere Implementierung auf der GPU betrachten.

4.1 Pseudo-Code

Das berechnen des

Algorithm 1 UMAP algorithm

function UMAP($X, N, D, d, min\_dist, n\_epochs$)
  for $x \in X$ **do**
    $knn(x) \leftarrow k\text{-}NearestNeighbour(x)$
    $graph(x) \leftarrow \bigcup_{y \in knn(x)}(\{x, y\}, exp(-d_{x,y})) \cup \bigcup_{y \notin knn(x)}(\{x, y\}, 0)$
  end for
  $A \leftarrow$ weighted adjacency matrix$(\bigcup_{x \in X} graph(x))$
  $D \leftarrow$ degree matrix for the graph A
  $L \leftarrow D^{1/2}(D - A)D^{1/2}$ $\rightarrow$ Symmetric normalized Laplacian
  $evec \leftarrow$ sorted Eigenvectors of L
  $Y \leftarrow evec[1, ..., d{+}1]$
  $Y \leftarrow OptimizeEmbedding(Y, min\_dist, n\_epochs)$
  return Y
end function

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.

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.

t-SNE

Das t-SNE Verfahren [11] ist zur Zeit eines der bekanntesten und meistgenutzten nicht-linearen 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 hochedimensionalen 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 erhalenen Wahrscheinlichkeiten $p_{i|j}$ sind im Allgemeinen nicht symmetrisch. Die Symmetrie wird durch mitteln der Daten erhalten.

Ähnlich wird eine Wahrscheinlichkeitsverteilung Q im niedirgdimensionalen Raum mithilfe der studentschen t-Verteilung konstruiert. Ursprünglich wurde Q ebenfalls durch eine Gausverteilung konstruiert, das so erhaltene Verfahren (SNE [6]) 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 [20, 9]. Dabei ist besonders Barnes-Hut-SNE [10] zu erwähnen, allerdings sollte hier beachtet werden, dass aufgrund der Konstruktion einer speziellen Datenstruktur die Laufzeit für $d > 3$ sehr schlecht ist.

ABBILDUNG 5.1: Sechs zufällig gewählte Gesichter des Cartoon Set.

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 [8]. Der `late-exaggeration` wie in [9] beschrieben kontrahiert gefundene Cluster, so lassen sich in einer 2- oder 3-dimensionalen Darstellung leichter Cluster bestimmen - entweder visuell oder mittels Clustering-Verfahren.

Für unsere Experimente haben wir die scikit Implementierung des t-SNE Verfahren genutzt [17]. Zusätzlich haben wir die openTSNE [18] Implementierung genutzt. Diese bescheleunigt die Laufzeit des t-SNE Algorithmus durch eine zusätzliche Fouriertransformation [9]. Die openTSNE Implementierung besitzt im Vergleich zur scikit Implementierung die Möglichkeit den `late-exaggeration` Parameter zu spezifizieren.

5.2 Cartoon Set

In diesem Abschnitt werden wir den *Cartoon Set* Datensatz analysieren [3]. 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

5.2.1 Beschreibung des Datensatzes

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

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

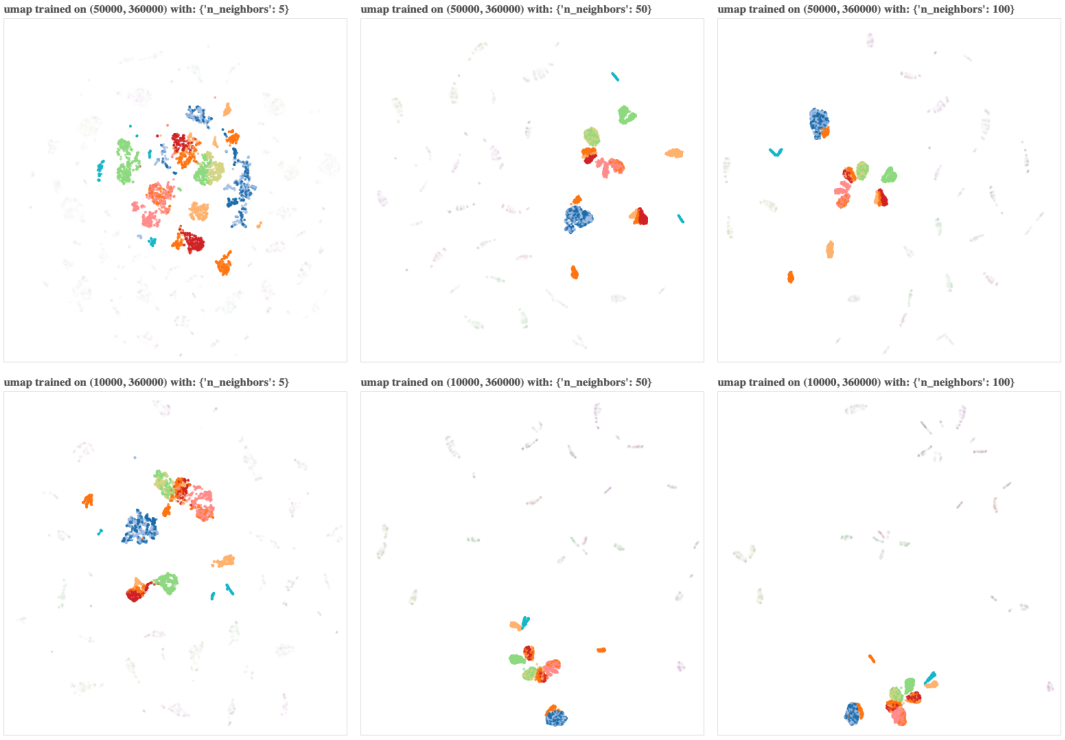

ABBILDUNG 5.2: TODO: Beschreibung des Bildes

Komponente zwischen zwei (Augenlied, 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 Frisurtypen zusammengefasst.

Wir haben uns für diesen Datensatz entschieden um UMAP auf Daten mit einer komplexeren Struktur zu testen als dies in [12] 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 $N = 100\,000$.

5.2.2 Qualitative Analyse der Ergebnisse

5.3 MNIST

FMNIST

5.4 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.

5.5 Stabilität unter sub-sampling

5.6 Zusammenfassung der Ergebnisse

Kapitel 6

Zusammenfassung

Literatur

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.