
Eduard Wiebe
Diplomarbeit

Schwieberdingen 2004

Diplomarbeit

Dichtebasierter Clustering-Verfahren in der industriellen Bildverarbeitung

Eduard Wiebe

13. September 2004

Betreuer:
Dr. habil. Benno Stein
Dipl.-Inform. Sören Hader



Universität
Paderborn

Universität Paderborn
Fakultät für Elektrotechnik, Mathematik und Informatik
Fürstenallee 11
33102 D-Paderborn

Diese Arbeit ist an der Robert Bosch GmbH in Schwieberdingen entstanden und wurde mit Textsatzsystem L^AT_EX erstellt.

Software- und Hardwarebezeichnungen, die in dieser Arbeit erwähnt werden, sind in den meisten Fällen eingetragene Marken und unterliegen als solche den gesetzlichen Bestimmungen.

Für meine Eltern

Inhaltsverzeichnis

Vorwort

i

1. Einleitung	1
1.1. Motivation	1
1.2. Ziel der Arbeit	1
1.3. Notation	3
2. Clustering	4
2.1. Problemdefinition	4
2.2. Komplexität	5
2.3. Standardverfahren	6
2.3.1. Exakte Algorithmen	6
2.3.2. Approximationsalgorithmen	7
2.3.3. Heuristiken	9
2.3.4. Stochastisches Clustering	13
3. Dichteschätzung	19
3.1. Einleitung	19
3.2. Grundlagen der Schätztheorie	19
3.3. Parametrisierte und nichtparametrisierte Dichteschätzung	20
3.4. Nichtparametrisierte Dichteschätzungstechniken	21
3.4.1. Univariater Fall	21
3.4.2. Multivariater Fall	26
3.5. Approximationseigenschaften	28
4. BST-Algorithmus	31
4.1. Analyse	31
4.2. Basin Spanning Trees	31
4.3. Nachbarschaft	32
4.4. Topographie	33
5. Wahl der Parameter	36
5.1. Stochastische Modelle	36
5.1.1. Kleinste-Quadrat-Kreuzvalidierung	36
5.1.2. Likelihood-Kreuzvalidierung	39

Inhaltsverzeichnis

5.2.	Heuristische Wahl	40
5.2.1.	MAXDEG	40
5.2.2.	MSTMAX	41
5.2.3.	MAXRADIUS	42
5.3.	Bewertung der Zerlegung	43
5.3.1.	Stabilität	43
5.3.2.	Inter- vs. Intra-Kriterien	44
5.3.3.	Isolation und Kompaktheit	45
5.3.4.	Isolation und Ausprägung	46
6.	Clustermerging	50
6.1.	Motivation	50
6.2.	Ausprägungskriterium	50
6.3.	Sattelpunkt-Kriterium	51
7.	Implementierung	54
7.1.	Matlab	54
7.2.	Toolbox zur Dichteschätzung	54
7.3.	Laufzeitmessungen	55
7.4.	Optimierungstechniken	56
8.	Experimentelle Resultate	59
8.1.	Testplattform	59
8.2.	Synthetische Daten	59
8.3.	Reale Datensätze	59
8.4.	Bilder	62
8.5.	Spritzer Klassifizierung	62
8.6.	Ergebnisse	62
8.6.1.	Evaluierung der nicht überwachten Lernmethoden	62
8.6.2.	Heuristische Wahl	65
8.6.3.	Stochastische Modelle und Bewertungsstrategien	66
8.6.4.	Clustermerging	66
8.6.5.	Spritzer	67
8.7.	Zusammenfassung	67
9.	Fazit	68
9.1.	Folgerungen	68
9.2.	Ausblick	69
A.	Ergebnisprotokolle	70
A.1.	Heuristische Wahl	71
A.2.	Bewertungsfunktionen und stochastische Modelle	83
A.3.	Clustermerging	96
A.4.	Spritzererkennung	110

Abbildungsverzeichnis

1.1. Daten, die es zu gruppieren gilt.	2
1.2. Gewünschte Gruppierung der Daten	2
2.1. Dendrogramm für den Smile Datensatz am Beispiel der SINGLE-LINK Heuristik	10
2.2. Verkettungseigenschaft (chaining)	15
2.3. Unimodale Verteilung in \mathbb{R} mit Modus Z_1	16
2.4. Bimodale Verteilung in \mathbb{R} mit Modalwerten Z_1 und Z_2	16
2.5. Multimodale Verteilung im \mathbb{R}^2	16
2.6. Zum Verfahren von Schnell	17
3.1. Plots der typischen Kernfunktionen	23
3.2. Histogramm (blau) und der naive Schätzer (rot)	24
3.3. Konstruktion der Dichtefunktion (rot) vermöge Kernschätzers. Hier am Beispiel der Gaußfunktion (blau).	24
3.4. Dichteschätzung mit einer größeren Bandweite.	24
4.1. Gegeben sind Punkte aus einem \mathbb{R}^d Raum.	34
4.2. Schritt 1: Die Dichteschätzung vermöge Kernschätzers. Hier mittels Höhenlinien angedeutet.	34
4.3. Schritt 2: Das Berechnen der Nachbarschaft, hier am Beispiel der Delaunay-Triangulierung.	35
4.4. Schritt 3: Konstruktion der Basin Spanning Trees. Die roten Punkte stellen die Wurzeln der Bäumen dar.	35
5.1. Bei einer kleinen Wahl der Bandweite besitzt die Dichteschätzung viele lokale Maxima. Hier am Beispiel des Smile Datensatzes deutlich erkennbar.	37
5.2. Vergrößerung der Bandweite minimiert die Anzahl der Modi.	37
5.3. Bei einer großen Bandweite nimmt die Kernschätzung lokal die Form der Kernfunktion an. Hier die Form des Epanechnikov Kerns.	37
5.4. Motivation der heuristischen Regel MAXDEG	40
5.5. Gegenbeispiel der MAXDEG Regel	40
5.6. Annahmen, die zur MSTMAX Heuristik führen.	41
5.7. Falsche Resultate der MSTMAX Heuristik	41
5.8. Motivation der MAXRADIUS Heuristik.	42

5.9. Gegenbeispiel zur MAXRADIUS Heuristik.	42
5.10. Stabilität als Auswahlkriterium: Der Stabilitätsverlauf der Zerlegungen des Smile Datensatzes für 50 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 1.0]. In diesem Fall ist die Bandweite Nummer 23, die zur Zerlegung von 4 Klassen führt, die gesuchte Bandweite.	44
5.11. Silhouetten-Koeffizienten für die Zerlegung des Smile Datensatzes mit der Bandweitengröße 2.0. Die vielen negativen Silhouetten-Koeffizienten sind ein Indiz für das fehlende Optimum der Zerlegung.	45
5.12. Dichteschätzung für zwei gut ausgeprägte Cluster.	47
5.13. Schlecht ausgeprägte Cluster deuten auf ein großes gemeinsames Cluster hin. Hier die Cluster um Modi Z_2 und Z_3	47
5.14. Passübergang zwischen zwei Clustern berechnet nach Algorithmus 5.4	49
 6.1. Approximation des Wahrscheinlichkeitsmaßes. Die Wahrscheinlichkeit in der Umgebung von Z_m und X_s ist proportional zu $\hat{f}(Z_m)$ und $\hat{f}(X_s)$	53
 7.1. Vergleich der Implementierungen für die Auswertung der Kernfunktion am Beispiel des Epanechnikov Kerns und 10 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 10].	58
8.1. Turing-Test Datensatz	61
8.2. Histogramme der Merkmalsverteilungen der Spritzerobjekte und der Aufhellungen (Merkmale v.l.n.r. und v.o.n.u.)	64
8.3. Merkmalsverteilungen des Datensatzes, der nur Spritzerobjekte enthält (Merkmale v.l.n.r. und v.o.n.u.)	64
 A.1. Turing-Test (Heuristiken v.l.n.r.: MAXDEG, MSTMAX, MAXRADIUS)	71
A.2. Turing-Test (Bewertungsfunktionen v.l.n.r.): Stabilitätskriterium, Likelihood-Kreuzvalidierung, Kleinste-Quadrat-Kreuzvalidierung . . .	83
A.3. Turing-Test (Fortsetzung der Bewertungsfunktionen): Silhouetten-Koeffizienten, Isolation-und-Ausprägung und schließlich Isolation-und-Kompaktheit	84
A.4. Turing-Test (v.l.n.r.): Initiale Lösung, Merging nach Ausprägungskriterium, Merging nach Sattelpunkt-Kriterium	96
A.5. Links: Originalbild. Rechts: Differenzbild + gekennzeichnetes Objekt (Bounding Box). Das Clustering wurde mittels Silhouetten-Koeffizienten-Bewertungsfunktion bestimmt.	110

Tabellenverzeichnis

3.1. Die gängigsten Kernfunktionen und deren Effizienz. $I(\cdot)$ ist die Indikatorfunktion.	23
7.1. Laufzeitverhalten der einzelnen Routinen am Beispiel der synthetischen Daten. Dabei wurden zweidimensionale Instanzen mit jeweils unterschiedlicher Anzahl Samples generiert. Anschließend wurde deren Dichte mit Bandweite $h = 3.0$ geschätzt.	56
8.1. Charakterisierung synthetischer Daten	60
8.2. Charakteristika realer Datensätze	60
8.3. Merkmalbeschreibung der Spritzerobjekte	63
A.1. Turing-Test: Resultate die mittels topologischer Heuristiken (MAXDEG, MSTMAX, MAXRADIUS) erzielt wurden.	71
A.2. Synthetische Daten: Clustering Resultate vermöge heuristischer Bandweitenwahl (MAXDEG, MSTMAX, MAXRADIUS).	72
A.3. Reale Datensätze. Obere Tabelle: Daten die mittels PCA auf 3 Dimensionen reduziert sind. Untere Tabelle: Reduktion auf 4 Dimensionen.	73
A.4. Turing-Test: Ermittelte Ergebnisse für 50 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 2.0]	85
A.5. Synthetische Daten: Ergebnisse der Bewertungsstrategien für 50 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 2.0]	86
A.6. Reale Datensätze: Resultate der Bewertungsstrategien	87
A.7. Turing-Test: Ergebnisse des Clustermergings mit dem Ausprägungskriterium. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.	96
A.8. Turing-Test: Ergebnisse, die mittels Clustermerging Methode nach dem Sattelpunkt-Kriterium erzielt wurden. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.	97
A.9. Synthetische Daten: Clustermerging nach dem Ausprägungskriterium. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.	97
A.10. Synthetische Daten: Clustermerging mit dem Sattelpunkt-Kriterium. Die initiale Lösung entstand mit der Bandweite $h = 0.25$	98
A.11. Realer Datensatz: Ergebnisse, die vermöge des Clustermergings mit dem Ausprägungskriterium erzielt wurden.	99

Tabellenverzeichnis

A.12. Realer Datensatz: Diese Ergebnisse sind mittels Clustermerging mit Sattelpunkt-Kriterium berechnet.	100
--	-----

Algorithmenverzeichnis

2.1.	Greedy Clustering	7
2.2.	Bottleneck Algorithmus	9
2.3.	Agglomeratives Clustering	10
2.4.	Clustering mit Seeds	12
2.5.	Graphenbasiertes Clustering	13
2.6.	Nächster-Nachbar Clustering	13
2.7.	Das Verfahren von Wishart	14
2.8.	Gradientenverfahren von Schnell	17
4.1.	Basin Spanning Trees Algorithmus	33
5.1.	MAXDEG Heuristik	40
5.2.	MSTMAX Heuristik	41
5.3.	MAXRADIUS Heuristik	43
5.4.	Verfahren zur Bestimmung der Passhöhen	48
6.1.	Clustermerging nach dem Ausprägungskriterium	51
6.2.	Clustermerging nach dem Sattelpunkt-Kriterium	52

Vorwort

Clustering-Probleme tauchen in mehreren Disziplinen auf und finden eine breite Anwendung, sei es nun in der Datenkomprimierung oder Informationsgewinnung, Mustererkennung oder Data Mining. So unterschiedlich die Anwendungen auch sind, so groß ist die Anzahl an verschiedenen Cluster-Verfahren, die währenddessen untersucht wurden. Doch eins haben sie alle gemeinsam: Das Finden einer Zerlegung der gegebenen Daten unter Maximierung einer Zielfunktion.

Ziel dieser Arbeit ist die Untersuchung der Clustering-Verfahren, die auf *nichtparametrisierter Dichteschätzung* basieren. Einer der Schwerpunkte ist die Untersuchung der Methoden für die automatische Wahl der *Bandweite des Dichteschätzers*.

Aufbau

Die vorliegende Arbeit ist wie folgt aufgebaut:

- Nach den ersten einleitenden Worten werden wir in Kapitel 1 unsere Ziele formulieren und die Notation festlegen.
- In Kapitel 2 werden wir das allgemeine Clustering-Problem kennenlernen. Dabei gehen wir auf die Komplexität ein und stellen die gängisten Verfahren und Methoden vor.
- Die Grundlagen der Dichteschätzung werden wir in Kapitel 3 näher erläutern.
- Darauf aufbauend stellen wir in Kapitel 4 einen Clustering Algorithmus vor und diskutieren die Vor- und Nachteile dieses Ansatzes.
- In Kapitel 5 gehen wir auf die Wahl der Parameter ein. Darin werden wir insbesondere die Bandweitenwahl ausführlich behandeln.
- Kapitel 6 stellt zwei hybride Heuristiken vor.
- Mit Implementierungsdetails befassen wir uns in Kapitel 7.
- Auf die experimentelle Ergebnisse, die wir mit entwickelten und untersuchten Techniken erzielten, werden wir in Kapitel 8 eingehen.
- Im letzten Kapitel ziehen wir ein Fazit und geben einen Ausblick auf weitere Arbeiten.

Danksagung

An dieser Stelle möchte ich die Gelegenheit ergreifen und einen Dank an alle aussprechen die diese Arbeit ermöglichten.

In erster Linie gilt mein Dank den Mitarbeitern der FV/PLF2 Abteilung der Robert Bosch GmbH in Schwieberdingen, insbesondere Sören Hader und Fred Hamprecht für anregende Diskussionen und die Betreuung der Arbeit. Benno Stein, meinem Betreuer an der Universität bin ich ebenso zu Dank verpflichtet als auch der Familie Rosenko und Stefan Lücking für das gewissenhafte Lesen des Manuskripts. Für die Lebensfreude, die mir täglich von meinen klasse Mädels, Anna und Marie, bereitet wird, bin ich im besonderen Maße dankbar.

1. Einleitung

In diesem Kapitel wollen wir die Beweggründe dieser Arbeit benennen. Ferner werden wir die Ziele präzisieren und anschließend die Formalismen und Bezeichnungen für das weitere Vorgehen festlegen.

1.1. Motivation

Die Datenanalyse ist ein wichtiger Aspekt vieler Rechner-unterstützter Anwendungen, sei es in der Designphase oder als Laufzeitoperation. Wir verstehen unter der *Clusteranalyse* die unüberwachte Organisation der Daten (gewöhnlich als Ähnlichkeitsvektoren oder Punkte in einem mehrdimensionalen Raum repräsentiert) in Klassen oder *Cluster* basierend auf einem *Ähnlichkeitsmaß*.

Hierbei unterscheiden wir zwischen nicht überwachter, dem Clustering, und der überwachten Klassifikation, der Diskriminanzanalyse. Während der überwachten Klassifikation steht uns ein sogenannter *gelabelter* Datensatz zur Verfügung. Das Problem besteht in der Zuordnung der neuen Daten zu den bestehenden Kategorien. Typischerweise werden die bekannten Daten, auch Testdaten genannt, für das Erlernen der Klassenkategorien eingesetzt. Im Falle des Clusterings jedoch, suchen wir nach einer möglichen Gruppierung der nicht gelabelten Daten in sinnvolle Klassen. D.h. die einzelnen Kategorien werden alleine aus Daten abgeleitet. Daher lassen sich die Muster erst nach der Klassifizierung der Daten erkennen bzw. beurteilen. Die Bilder 1.1 und 1.2 verdeutlichen die Problemstellung.

Clustering ist ein wichtiger Bestandteil unterschiedlicher Musteranalyseverfahren. Sei es bei der Gruppierung der Daten, unterstützter Entscheidungsfindung oder maschinellem Lernen. Wobei das maschinelle Lernen in verschiedenen Bereichen wie Data Mining, Informationsgewinnung, Bildverarbeitung oder visuelle Mustererkennung stattfindet.

Die Clustering-Algorithmen lassen sich in vielfältiger Weise einsetzen, z.B. in der Bioinformatik, in medizinischen Diagnose Verfahren ([Wei u. a. \(2001\)](#), [Golub u. a. \(1999\)](#)), in statistischen Analysen, Computerlinguistik und Text Mining, Informati-onssuche in und Analysen von Datenbanken, Neuronalen Netzwerken und in vielen anderen Gebieten.

1.2. Ziel der Arbeit

In der Abteilung FV/PLF2 der Robert Bosch GmbH in Schwieberdingen ist eine MATLAB-Toolbox zur Untersuchung der dichtebasierteren Clustering-Verfahren ent-

1. Einleitung

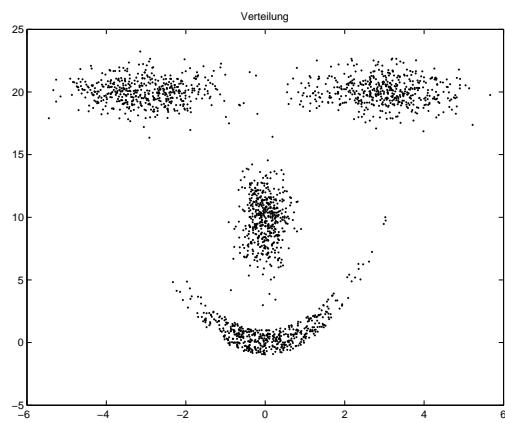


Abbildung 1.1.: Daten, die es zu gruppieren gilt.

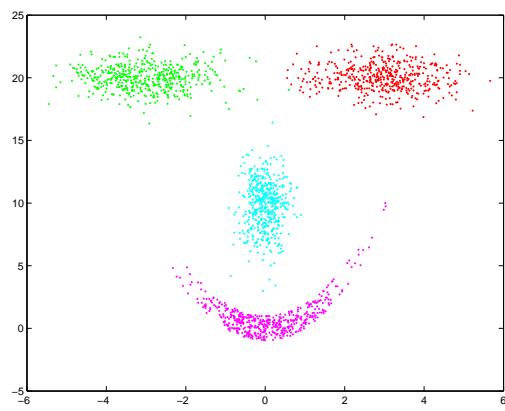


Abbildung 1.2.: Gewünschte Gruppierung der Daten

standen. Die Erkenntnisse, die vermöge der Toolbox gewonnen werden, sollen in die Entwicklung und Produktion innovativer Produkte einfließen. Das Ziel dieser Arbeit besteht aus folgenden Aufgaben:

- Einarbeitung in die Toolbox
- Optimierung der bestehenden Routinen
- Entwicklung eines oder mehrerer Verfahren zur automatischen Bestimmung der *Bandweite* für ein dichtebasierter Clustering Verfahren
- vergleichen der entwickelten Verfahren mit herkömmlichen Methoden
- Integration und Dokumentation der Verfahren innerhalb der Toolbox

1.3. Notation

In weiteren Kapiteln dieser Arbeit werden wir immer von Punkten, Beobachtungen, Samples und Daten, die es zu clustern bzw. zu klassifizieren gilt, sprechen. Diese Begriffe wollen wir unabhängig vom Kontext synonym behandeln. Ähnlich wollen wir die Begriffe des Clusterings, der Datenzerlegung, der Gruppierung und der Klassifizierung handhaben.

Nichtsdestotrotz werden wir unterschiedliche Notation für denselben Sachverhalt benutzen. Einerseits wollen wir die historisch bedingte Notation der unterschiedlichen Bereiche beibehalten, um ein schnelles Zurechtfinden des Lesers in der entsprechenden Fachliteratur zu ermöglichen. Andererseits verdeutlichen gerade diese Unterschiede die aktuelle Sichtweise auf das Problem.

Weiterhin wollen wir nicht die Notation zwischen den Punkten eines eindimensionalen und mehrdimensionalen Raumes formal unterscheiden. Eine Ausnahme bildet Kapitel 3. Darin erarbeiten wir Gemeinsamkeiten bzw. Unterschiede in den verschiedenen Ansätzen.

Die Notation, die in dieser Arbeit verwendet wird, ist in Form einer Tabelle im Anhang B auf Seite 111 zu finden.

2. Clustering

In diesem Kapitel wollen wir die Grundlage dieser Arbeit schaffen. Zuerst werden wir Clustering aus der geometrischen Sicht formal definieren, um die Komplexität des Problems zu verdeutlichen. Mit dem Überblick über die gängigsten Verfahren schließen wir das Kapitel ab.

2.1. Problemdefinition

Sei \mathcal{S} eine Menge von Punkten eines d -dimensionalen reellen Raumes. Eine Partitionierung von \mathcal{S} in $\mathcal{C} = \{C_1, \dots, C_k\}$ disjunkte, nicht leere Teilmengen heißt k -*Clustering* und die Teilmengen C_i heißen *Cluster* oder *Klassen*. Im Allgemeinen verstehen wir unter einem Clustering-Problem das Bestimmen eines k -Clusterings, so dass eine gegebene Kostenfunktion c optimiert wird.

Definition 2.1. Für eine Distanzfunktion Δ bezeichnen wir mit $z_i \in \mathbb{R}^d$ ein *Clusterzentrum*, falls dieser den Ursprungspunkt der kleinsten umschreibenden Sphäre des Clusters $C_i \in \mathcal{C}$ bezüglich Δ darstellt. Weiter wollen wir mit $Z = \{z_1, \dots, z_k\}$ die Menge aller Clusterzentren für ein k -Clustering \mathcal{C} bezeichnen.

Die Distanz zwischen zwei Punkten ist eine beliebige Funktion $\Delta: \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^{\geq 0}$. Diese hängt in erster Linie von der Anwendung ab. In vielen Fällen stellt die Distanzfunktion eine Metrik da, die nicht notwendigerweise die Dreiecksungleichung erfüllt.

Die kleinste Größe ρ für ein k -Clustering im geometrischen Raum für die gilt, entweder:

1. ρ ist die maximale Distanz zwischen zwei Punkten eines Clusters oder
2. $\rho/2$ ist die maximale Distanz zwischen allen Punkten des Clusters und dem Clusterzentrum

wollen wir *Clustergröße* nennen.

In Abhängigkeit von der Clustergrößendefinition sprechen wir von dem *paarweisen Clustering* oder *zentralen Clustering*. Weiterhin unterscheiden wir zwischen *allgemeinem zentralen Clustering*, falls das Clusterzentrum ein beliebiger Punkt in einem d -dimensionalem Raum ist und dem *diskreten zentralen Clustering*, wenn das Zentrum der Menge \mathcal{S} angehört.

Die geläufigsten Clusteringprobleme haben folgende Gestalt:

- (F1) Gegeben sind \mathcal{S} und k , finde ein k -Clustering mit der minimalen Clustergröße.

- (F2) Gegeben sind \mathcal{S} und eine positive reelle Zahl w . Zerlege \mathcal{S} in k Cluster so, dass k minimal ist und die Clustergröße w nicht überschreitet.

Die Probleme der Form (F1) können wir als eine Optimierungsaufgabe mit der Clustergröße als Zielfunktion c formulieren:

$$c(\mathcal{S}) = \max_{1 \leq i \leq k} \rho_i \rightarrow \min.$$

Deshalb werden die Probleme dieses Typs auch *min-max* Probleme genannt. In der Graphentheorie werden Probleme dieser Art, bei denen es um Maximierung eines Minimums bzw. Minimierung eines Maximums geht, auch *Bottleneck*-Probleme genannt.

Eine weit verbreitete Variation der Clustering-Probleme minimiert die Summe der Clustergrößen. Die Zielfunktion kann dabei wie folgt geschrieben werden:

$$c(\mathcal{S}) = \sum_{i=1}^k \rho_i \rightarrow \min.$$

Diese Art von Problemen ist auch als *k-median* Problem oder auch *min-sum* Problem bekannt.

Eine weitere Klassifizierung der Clusteringprobleme kann unter der Hinzunahme der Metrik geschehen. Mit der häufig verwendeten L_2 -Metrik (euklidischer Abstand) wird das zentrale Clustering Problem zum Abdeckungsproblem der Punkte mittels eines Balles mit dem Radius $r = \rho/2$. Dieses Problem ist in der Literatur unter dem Namen *euklidisches k-Zentren Problem* bekannt. Für die L_∞ -Metrik besteht die Aufgabe in der Abdeckung der Punkte mit k d -dimensionalen Parallelogrammen der Seitenlänge ρ .

Im Allgemeinen betrachten wir einen vollständigen Graphen. Die Punkte der Menge \mathcal{S} sind die Knoten und die Distanzen sind die Gewichte der Kanten. In diesem Fall sprechen wir vom *graphenbasierten Clustering*. Ist die Distanz zwischen zwei Punkten durch eine Metrik L_q für $q \geq 1$ gegeben, so sprechen wir über ein *geometrisches Clustering*.

2.2. Komplexität

In einer Dimension kann sowohl paarweises als auch zentrales Clustering mittels dynamischer Programmierung in polynomieller Zeit gelöst werden (siehe Brucker (1977)). In zwei Dimensionen ist zentrales Clustering dagegen \mathcal{NP} -vollständig (Fowler u. a. (1981), Megiddo und Supowit (1983)). Das Problem bleibt sogar dann \mathcal{NP} -vollständig, wenn nur eine approximative Lösung gesucht wird (Feder und Greene (1988), Gonzalez (1985), Ko u. a. (1990)). Für die Metriken, welche die Dreiecksungleichung erfüllen, kann paarweises Clustering auf zentrales Clustering reduziert werden und ist somit auch \mathcal{NP} -vollständig. Dabei erfolgt die Reduktion auf das folgende Entscheidungsproblem:

2. Clustering

(F3) Gegeben sind \mathcal{S} , k und w . Existiert ein k -Clustering der Größe w ?

Das bemerkenswerte an dieser Technik ist, dass sie einen Einblick in die Problematik der Approximation der optimalen Clustergröße für die Probleme der Form (F1) gewährt. Es sind viele Reduktionen unterschiedlicher \mathcal{NP} -vollständiger Probleme auf die (F3) Form veröffentlicht. Sie alle haben das Ziel, die Approximationsschranke der optimalen Clustergröße zu verbessern. Megiddo und Supowit (1983) zeigten, dass für den Faktor¹ $\alpha := \frac{\rho_{\text{APPROX}}}{\rho_{\text{optimal}}}$ der optimalen Clustergröße das zentrale L_2 -Clustering Problem für $\alpha < 1.154$ und L_∞ -Clustering für $\alpha < 1.5$ \mathcal{NP} -schwer bleiben. Ebenso zeigten sie, dass das k -median Problem \mathcal{NP} -schwer ist. Gonzalez (1985) zeigte, dass paarweises Clustering in 2 Dimensionen für den Faktor $\alpha < 1.732$ und für $\alpha < 2$ in 3 Dimensionen \mathcal{NP} -schwer ist. Die besten bekannten Schranken für den Approximationsfaktor zeigten Feder und Greene (1988): 1.82 für zentrales L_2 -Clustering, 1.92 für paarweises L_2 -Clustering und 2 für paarweises und zentrales L_1 - und L_∞ -Clustering (alle Resultate in $d \geq 2$ Dimensionen). Weiterhin zeigten sie, dass das approximative Clustering in $d \geq 2$ Dimensionen \mathcal{NP} -schwer für den Faktor $\alpha < 1.732$ für das diskrete zentrale Clustering ist. Eine analoge Aussage gilt für das L_1 und L_∞ diskrete zentrale Clustering mit dem Faktor echt kleiner 2. Hsu und Nemhauser (1979) zeigten die Schranke $\alpha < 2$ für alle Metriken, welche die Dreiecksungleichung erfüllen.

2.3. Standardverfahren

2.3.1. Exakte Algorithmen

Ein exakter Algorithmus für das euklidische k -Zentren Problem kann aus einer einfachen Beobachtung hergeleitet werden. Eine Kugel in einem d dimensionalen Raum kann eindeutig durch $d+1$ Punkte festgelegt werden. Somit ist die optimale Clustergröße $r = \rho/2$, dies entspricht dem Radius einer Kugel, die durch $d+1$ Punkte der Menge \mathcal{S} beschrieben wird. Es gibt $\mathcal{O}(n^{d+1})$ Möglichkeiten $d+1$ Punkte aus \mathcal{S} zu wählen, mit $n := |\mathcal{S}|$. Analog dazu können die restlichen $k-1$ Kugeln auf $\mathcal{O}(n^{d(k-1)})$ unterschiedliche Weisen gewählt werden. Insgesamt sind es $\mathcal{O}(n^{dk+1})$ mögliche Kombinationen für k Bälle. Die Verifizierung, dass k Bälle die Menge \mathcal{S} abdecken, kann in $\mathcal{O}(n)$ geschehen. Daher ergibt sich für diesen naiven Algorithmus eine Laufzeit von $\mathcal{O}(n^{dk+2})$.

Für den planaren Fall gab Drezner (1984) einen $\mathcal{O}(n^{2k+1} \log n)$ Algorithmus an, der zu einem $\mathcal{O}(n^{2k-1})$ verbessert werden kann, wenn wir den Algorithmus für das euklidische 1-Zentrum Problem mit Laufzeit $\mathcal{O}(n)$ von Megiddo (1983) oder Dyer (1986) verwenden. Später verbesserten Hwang u. a. (1993) die Laufzeit auf $n^{\mathcal{O}(\sqrt{k})}$.

Sharir und Welzl (1996) konnten zeigen, dass rechteckig zentrales k -Clustering in der Ebene in $\mathcal{O}(n^{k-4} \log^5 n)$, für $k \geq 5$ gelöst werden kann. Das Resultat verbesserte schließlich Nussbaum (1997) auf $\mathcal{O}(n^{k-4} \log n)$.

¹Diesen Approximationfaktor wollen wir auch *Güte* nennen.

2.3.2. Approximationsalgorithmen

Greedy Algorithmus

Der GREEDY Algorithmus (Alg. 2.1), der von [Gonzalez \(1985\)](#) stammt, ist für beide Problemtypen des Clusterings geeignet. Er funktioniert mit jeder Metrik L_q und erreicht, wie schon erwähnt, einen Approximationsfaktor von 2. Die Eigenschaften für jede beliebige Metrik, welche die Dreiecksungleichung erfüllen, bleiben erhalten.

Algorithmus 2.1 Greedy Clustering

```

1: Input:  $\mathcal{S} = \{p_1, \dots, p_n\}$  samples,  $k$  number of clusters
2: Output:  $\mathcal{C} = \{C_1, \dots, C_k\}$  clustering
3:  $H \leftarrow \{p_1\}$  //  $p_1$  ist zufällig gewählt aus  $\mathcal{S}$ 
4:  $C_1 \leftarrow \mathcal{S}$ 
5: for  $i \leftarrow 1$  to  $n$  do
6:    $dist(p_i) \leftarrow \Delta(p_i, p_1)$ 
7: end for
8: for  $i \leftarrow 2$  to  $k$  do
9:    $C_i \leftarrow \emptyset$ 
10:   $p \leftarrow \text{argmax}\{dist(p_\ell) \mid p_\ell \in \mathcal{S} \setminus H\}$ 
11:   $H \leftarrow H \cup \{p\}$ 
12:  for  $j \leftarrow 1$  to  $n$  do
13:    if  $\Delta(p_j, p) < dist(p_j)$  then
14:       $dist(p_j) \leftarrow \Delta(p_j, p)$ 
15:      reassign( $p_j$ ,  $C_i$ )
16:    end if
17:  end for
18: end for

```

Der Algorithmus arbeitet nach der bewährten Greedy-Manier. Es wird heuristisch eine Teilmenge $H \subseteq S$ mit k Punkten, die am weitesten von allen anderen Punkten entfernt sind, während der Iteration aufgebaut. Für jeden Punkt p der komplementären Menge $S \setminus H$ wird die Größe $dist(p) = \min_{q \in H} \Delta(p, q)$ verwaltet. Jeder Punkt $h_i \in H$ bestimmt einen Cluster C_i . Ein Punkt aus der Menge $S \setminus H$ wird dem Cluster C_i zugewiesen, wenn er dem Punkt h_i näher ist, als den anderen Punkten aus der Menge $H \setminus \{h_i\}$.

Die Laufzeit des Algorithmus ist offensichtlich $\mathcal{O}(nk)$ mit $n := |\mathcal{S}|$. Diese verbesserten später [Feder und Greene \(1988\)](#) auf $\mathcal{O}(n \log k)$. Dabei wendeten sie die sogenannte *Box Decomposition Scheme* Technik an. Zuerst teilen wir die Punktmenge in rechteckige Boxen auf, wobei eine Approximation mit Faktor 6 erreicht wird. Daraufhin verfeinern wir die Lösung bis der gewünschte Faktor 2 erreicht ist. [Feder und Greene](#) zeigten, dass deren Algorithmus unter dem algebraischen Entscheidungsbaum Modell optimal ist.

2. Clustering

Bottleneck Ansatz

Hochbaum und Shmoys (1986) stellten ein allgemeines Approximationsschema für das Bottleneck-Problem dar. Wir wollen einen Aprroximationsalgoritmus (Hochbaum und Shmoys (1985)) für das k -Zentren Problem vorstellen, der ebenfalls einen Näherungsfaktor 2 hat.

Betrachten wir \mathcal{S} als einen vollständigen Graphen $\mathcal{G} = (V, E)$ mit Kantengewichten, welche die Dreiecksungleichung erfüllen. Wir sortieren die Kanten aufsteigend nach den Gewichten und nennen diese $w_1, \dots, w_{|E|}$. Sei nun \mathcal{G}_i ein Teilgraph, der durch die Kanten mit Gewicht höchstens w_i induziert wird.

Wir lassen schrittweise immer längere Kanten — und damit Wege — zu, bis k Zentren ausreichend sind, um alle Punkte abzudecken, bzw. aus einem der Zentren erreichbar sind. Dabei erlauben wir nur direkte Wege, die nur aus einer Kante bestehen, zwischen den Punkten und den Clusterzentren. Ein *Domination Set* ist eine Teilmenge von Knoten, die von allen Knoten eines Graphen über genau eine Kante erreicht werden kann.

Definition 2.2. Sei $\mathcal{G} = (V, E)$ und $D \subseteq V$. D ist *Dominating Set*, wenn jeder Knoten aus $V \setminus D$ einen Nachbarn in D hat.

Die optimale Clustergröße, das Minimum des Maximalen Weges zum nächsten Zentrum, ist offensichtlich gleich einer Kante mit einem Gewicht w_i . Das Entscheidungsproblem, ob ein Graph \mathcal{G} ein Dominating Set der Größe k besitzt, ist \mathcal{NP} -vollständig. Wir benutzen daher den folgenden Zusammenhang zwischen dem *Independent Set* und dem Dominating Set.

Definition 2.3. Sei $\mathcal{G} = (V, E)$. $I \subseteq V$ heißt *Independent Set*, wenn kein Knoten aus I einen Nachbarknoten in I hat. Independent Set I heißt maximal, falls für alle $v \in V \setminus I$ gilt $I \cup v$ ist kein Independent Set.

Lemma 2.1. Ein maximales Independent Set ist auch ein Dominating Set.

Beweis. Sei I ein maximales Independent Set. Also hat jeder Knoten aus $V \setminus I$ einen Nachbarn in I , sonst wäre I nicht maximal. Was wiederum bedeutet, dass I auch ein Dominating Set ist. \square

Definition 2.4. Sei $\mathcal{G} = (V, E)$, dann ist $\mathcal{G}^k = (V, E^k)$ die k -te Potenz von \mathcal{G} , wobei E^k die Kantenmenge von \mathcal{G} ist, erweitert um Kanten zwischen Knoten, die einen Weg der Länge kleiner gleich k im \mathcal{G} bilden.

Lemma 2.2. Sei I ein Independent Set in \mathcal{G}^2 , $|I| > k$, dann existiert kein Dominating Set D für \mathcal{G} mit $|D| \leq k$.

Beweis. Sei I ein Independent Set in \mathcal{G} . Zwei beliebige Knoten $v, w \in I$ und $v \neq w$, sind mindestens drei Kanten voneinander entfernt. Somit hat ein Dominating Set D für jeden Knoten von I einen individuellen dominierenden Knoten und daher $|D| \geq |I| > k$. \square

Korollar 2.1. Sei I ein maximales Independent Set in \mathcal{G}^2 mit $|I| \leq k$. Es existiert ein Dominating Set für \mathcal{G}^2 mit $|D| \leq k$, nämlich $D = I$.

Somit können wir ein Algorithmus wie folgt formulieren (s. Alg. 2.2). Die Korrektheit des Algorithmus folgt sofort aus:

1. Annahme: Abbruch bei i
2. Es gibt ein Independet Set I in \mathcal{G}_{i-1}^2 mit $|I| > k$. Nach Lemma 2.2 folgt, dass es kein Dominating Set der Größe maximal k gibt. Somit ist die optimale Clustergröße mindestens w_i .
3. Da I ein maximales Independent Set für \mathcal{G}_i^2 , ist nach Korollar 2.1 I auch Dominating Set für \mathcal{G}_i^2 . $|I|$ ist maximal k . Die längste Kante in \mathcal{G}_i^2 ist kleiner gleich $2 \cdot w_i$.

Algorithmus 2.2 Bottleneck Algorithmus

```

1: Input:  $\mathcal{G} = (V, E)$  complete graph,  $k$  number of clusters
2: Output:  $\mathcal{C} = \{C_1, \dots, C_k\}$  clustering
3: sort( $E$ ) s.t.  $w(e_1) \leq \dots \leq w(e_{|E|})$ 
4: for all  $w_i \in w(E)$  do
5:    $\mathcal{G}_i \leftarrow \text{subgraph}(V, \{e \in E \mid w(e) \leq w_i\})$ 
6:    $\mathcal{G}_i^2 \leftarrow \text{power-graph}(\mathcal{G}_i, 2)$ 
7:    $I \leftarrow \text{maximal-independent-set}(\mathcal{G}_i^2)$ 
8:   if  $|I| \leq k$  then
9:      $\mathcal{C} \leftarrow \text{assign}(\mathcal{S}, I)$ ;
10:  end if
11: end for

```

2.3.3. Heuristiken

In diesem Abschnitt werden Heuristiken vorgestellt, die sich nicht „unbedingt“ auf die Distanzfunktion verlassen bzw. keinen Gebrauch von der Geometrie machen. Diese Art der heuristischen Algorithmen wurde sehr populär in den Bereichen, in denen Clustering notwendig ist, aber die Ähnlichkeitsskriterien keine intuitive Interpretation im geometrischen Raum zulassen. Diese Art von Heuristiken werden in zwei Gruppen eingeteilt: in *hierarchisches* und *partitionierendes* Clustering.

Hierarchische Heuristiken

Definition 2.5. Ein Clustering \mathcal{C} heißt in \mathcal{C}' *eingebettet*, falls jeder der Cluster $C_i \in \mathcal{C}$ eine Teilmenge eines Clusters $C'_j \in \mathcal{C}'$ darstellt.

Das Ziel der hierarchischen Heuristiken ist, eine Folge der eingebetteten Zerlegungen zu produzieren. Wobei die niedrigste Ebene aus n 1-Punkt-Clustern besteht und die höchste Ebene einen einzigen Cluster der gesamten Menge bildet. Die Methode

2. Clustering

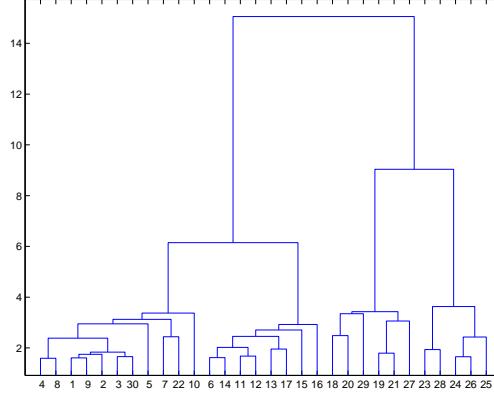


Abbildung 2.1.: Dendrogramm für den Smile Datensatz am Beispiel der SINGLE-LINK Heuristik

kann als ein hierarchischer Baum bildhaft dargestellt werden, das sogenannte *Dendrogramm* (siehe Abb. 2.1). Wir ziehen eine Kante im Dendrogramm zwischen \mathcal{C}' und \mathcal{C} , falls \mathcal{C} in \mathcal{C}' eingebettet ist. Die waagerechten Ebenen des Dendrogramms repräsentieren eine Zerlegung der Daten pro Algorithmusschritt.

Starten wir mit n einzelnen Clustern und fügen diese zur Gesamtmenge (starten mit der ganzen Menge und zerlegen diese in n einzelne Cluster), dann heißen diese *agglomerativ (divisiv)*. Offensichtlich kann eine Methode als Umkehrung der anderen angesehen werden. Daher geben wir an dieser Stelle nur die agglomerative Variante an (Alg. 2.3). Im Schritt 7 der Methode werden zwei ähnliche Cluster bestimmt

Algorithmus 2.3 Agglomeratives Clustering

- 1: **Input:** $\mathcal{S} = \{p_1, \dots, p_n\}$ samples, $D := (d_{ij})$ distance matrix
 - 2: **Output:** \mathcal{C} dendrogram
 - 3: **for all** p_i **do**
 - 4: $C_i \leftarrow \{p_i\}$
 - 5: **end for**
 - 6: **for** $k \leftarrow 1$ to n **do**
 - 7: $(C_r, C_s) \leftarrow \text{most-similar-clusters}(\mathcal{C}, D)$
 - 8: $\text{merge}(\mathcal{C}, C_r, C_s)$
 - 9: **end for**
-

(z.B. sei die Distanz zwischen den Clustern r und s minimal). Diese werden dann im Schritt 8 zusammengefasst und die Distanzmatrix D aktualisiert.

Es sind verschiedene Variationen der Methode in der Literatur bekannt. Diese leiten sich aus den unterschiedlichen Implementierungen der `merge`-Prozedur ab. Genauer: Aus der Berechnung der Unähnlichkeitsmaße $d_{tk} := \Delta(C_t, C_k)$ für den neu entstandenen Cluster C_t zu den restlichen Clustern C_k . Hier wollen wir einige verbreitete Varianten vorstellen. Im Folgenden wollen wir mit C_t den neuen Cluster, der durch das Zusammenführen der Cluster C_r und C_s entsteht, bezeichnen sowie

einen weiteren Cluster aus der aktuellen Zerlegung mit C_k . Ferner sei $n_i := |C_i|$.

Single-link clustering: $d_{tk} = \min(d_{rk}, d_{sk})$. Betrachten wir die Punkte als Knoten und die Distanzen aus D als Kantengewichte eines Graphen. Sei nun \mathcal{G}' ein Teilgraph, der nur die Kanten mit einer Gewichtung kleiner als d_{rs} besitzt. Dann stellen die größten Zusammenhangskomponenten im \mathcal{G}' die Cluster, vor dem Schritt in dem die Cluster C_r und C_s verbunden werden, in den Daten dar. Ferner entsprechen die Cluster C_r und C_s den Zusammenhangskomponenten, die durch eine Kante² zusammengefügt werden können. Daher röhrt auch der Name der Methode.

Complete-link clustering: $d_{tk} = \max(d_{rk}, d_{sk})$.

Definition 2.6. Sei $\tau \geq 0$ eine Distanzschanke. Eine Teilmenge \mathcal{C} von $\mathcal{S} = \{p_1, \dots, p_n\}$ heißt eine Clique (genauer: Clique der Stufe τ) genau dann, wenn die Gleichung $\Delta(p_i, p_j) \leq \tau$ für alle $p_i, p_j \in \mathcal{C}$ gilt. Eine Clique \mathcal{C} heißt *maximal*, wenn für alle $p_k \in \mathcal{S} \setminus \mathcal{C}$ gilt $\mathcal{C} \cup \{p_k\}$ ist keine Clique.

Betrachten wir erneut den Teilgraphen \mathcal{G}' , wie oben beschrieben. Die maximalen Cliques korrespondieren in diesem Teilgraphen \mathcal{G}' mit Clustern bevor C_r und C_s im Schritt 7 zusammengeführt werden. Schließlich können die Cluster C_r und C_s in eine gemeinsame maximale Clique durch die Hinzunahme einer Kante mit dem größten Gewicht transformiert werden.

Average-link clustering: $d_{tk} = n_r/(n_s + n_r)d_{rk} + n_s/(n_s + n_r)d_{sk}$ (ungewichteter Fall) oder $d_{tk} = 1/2(d_{rk} + d_{sk})$ (gewichteter Fall). Während der Distanzberechnung zwischen zwei Punkten wird die mittlere Unähnlichkeit beider Cluster benutzt. Im ungewichteten Fall ziehen wir die Anzahl der Elemente der einzelnen Cluster in die Betrachtung mit ein. In der gewichteten Variante ist dies nicht der Fall. Damit sind die Elemente aus kleineren Clustern mehr gewichtet, als solche aus größeren.

Centroid-link clustering: $d_{tk} = n_r/(n_s + n_r)d_{rk} + n_s/(n_s + n_r)d_{sk} - n_s n_r / (n_s + n_r)^2 d_{rs}$ (ungewichteter Fall) oder $d_{tk} = 1/2(d_{rk} + d_{sk}) - 1/4d_{rs}$ (gewichteter Fall, auch als *median clustering* bekannt). Diese Methode ist ähnlich dem AVERAGE-LINK Clustering, mit der Ausnahme, dass die Distanz zu den Schwerpunkten der einzelnen Cluster einfließt.

Ward's clustering: $d_{tk} = (n_r + n_s)/(n_r + n_s + n_k)d_{rk} + (n_s + n_k)/(n_r + n_s + n_k)d_{sk} - n_k/(n_r + n_s + n_k)d_{rs}$. Diese Methode minimiert die Varianzänderung während der Zusammenführung zweier Cluster. Die Varianz stellt die quadratische Abweichung eines Punktes vom Schwerpunkt dar.

In der Praxis tendiert die SINGLE-LINK Methode dazu, die Cluster-Hierarchie als eine Kette darzustellen. Die COMPLETE-LINK Methode hingegen führt oft zur einer

²engl.: link

2. Clustering

Zerlegung mit schlecht isolierten Clustern. Die zuletzt genannten Methoden versuchen diese Probleme zu vermeiden, wobei die WARD's Methode in den Vergleichsstudien am besten abschneidet. Der interessierte Leser kann bei [Anderberg \(1973\)](#) mehr darüber erfahren.

Partitionierende Heuristiken

Das Ziel der Heuristiken, die im vorangegangenen Abschnitt vorgestellt wurden, besteht im Aufbau einer Clustering-Hierarchie, welche die Einbettung einer Zerlegung in eine andere optimiert. Weitere Verfahren, die in diesem Abschnitt vorgestellt werden, generieren dagegen ein k -Clustering der Eingabe, die sogenannte Startlösung und verfeinern diese sukzessive, bis ein Abbruchkriterium erfüllt ist. Ziel ist es eine Lösung, die sehr nah an der optimalen Lösung liegt, zu bestimmen. Die Suche läuft heuristisch ab. Wir erhoffen uns in wenigen Iterationen, die gewünschte Lösung zu erhalten. Es kann jedoch gezeigt werden, dass eine schlechte Wahl der Initiallösung zu einer resultierenden Zerlegung führen kann, welche die optimale Lösung um einen beliebig großen Faktor approximiert. Im Folgenden wollen wir drei der häufigsten Arten partitionierender Heuristiken vorstellen: *Clustering mit Seeds*, *graphenbasiertes Clustering* und *Nächster-Nachbar Clustering*.

Algorithmus 2.4 Clustering mit Seeds

```

1: Input:  $S = \{p_1, \dots, p_n\}$  samples,  $k$  number of clusters
2: Output:  $\mathcal{C} = \{C_1, \dots, C_k\}$  clustering
3:  $seeds \leftarrow choose-k-seeds()$  // nicht notwendigerweise Punkte aus  $S$ 
4: for all  $seeds_i$  do
5:    $C_i \leftarrow \emptyset$ 
6: end for
7: while  $\neg abort()$  do
8:   for all  $p_i \in S$  do
9:      $C_j \leftarrow closest(seeds, \mathcal{C})$ 
10:     $reassign(p_i, C_j)$ 
11:   end for
12:    $seeds \leftarrow recompute(seeds, \mathcal{C})$ 
13: end while

```

Das allgemeine Ablaufschema des Clusterings mit Seeds kann dem Algorithmus 2.4 entnommen werden. Um die initialen Punkte auszuwählen, wird eine Vielzahl an Heuristiken benutzt:

- Zufällige Wahl (MacQueen's Algorithmus, besser als k -means bekannt)
- Forgy's Algorithmus wählt eine zufällige k -Partitionierung der Daten und bestimmt die Startpunkte als Clusterschwerpunkte. Hierfür wird ein hierarchisches Verfahren zur Generierung der initialen Lösung eingesetzt.

Die Wahl der neuen Startpunkte hängt in erster Linie von der Wahl der Distanzfunktion ab. Sehr beliebt ist die quadratische Fehlerfunktion. In diesem Fall werden

neue Startpunkte durch die Schwerpunkte vorgegeben.

Der Algorithmus hält an, falls die maximale Anzahl der Iterationen erreicht ist oder die Lösungsqualität konvergiert. Das heißt diese befindet sich innerhalb einer ε -Umgebung mit vorherigen Lösungen.

Andere Kategorien der zerlegenden Heuristiken benutzen Graphen, um Punkte in Cluster zu zerlegen. Zunächst gehen wir von einem vollständigen Graphen zwischen den Punkten aus, deren Kantengewichte den Distanzen zweier Punkte entsprechen. Der Ablauf ist im Algorithmus 2.5 dargestellt.

Algorithmus 2.5 Graphenbasiertes Clustering

```

1: Input:  $\mathcal{G}$  complete graph
2: Output:  $\mathcal{G}'$  subgraph, represented clustering
3:  $\mathcal{G}' \leftarrow \text{subgraph}(\mathcal{G})$ 
4: for all  $e \in E'$  do
5:   if is-inconsistent( $e$ ) then
6:      $E' \leftarrow E' \setminus \{e\}$ 
7:   end if
8: end for
```

Es können unterschiedliche Teilgraphen Strukturen verwendet werden: Minimaler Spannbaum, relativer Nachbarschaftsgraph (Punkte p_i und p_j sind durch eine Kante verbunden, genau dann, wenn $\Delta(p_i, p_j) < \max(\Delta(p_i, p_k), \Delta(p_k, p_j))$ für alle $p_k \neq p_i, p_j$), Gabriel Graph (Punkte p_i, p_j sind verbunden, genau dann, wenn $\Delta(p_i, p_j)^2 < \Delta(p_i, p_k)^2 + \Delta(p_k, p_j)^2$) oder auch die Delaunay-Triangulierung der Punkte.

Der Konsistenztest der Kante wird meistens durch einen Vergleich der Kantenlänge mit der mittleren Kantenlänge durchgeführt. Dies kann jedoch von Anwendung zu Anwendung variieren.

Die letzte partitionierende Heuristik, die wir vorstellen wollen, benutzt die *Nächster-Nachbar Regel*, um Punkte zu clustern. Hierfür wird ein Schwellwert t verwendet, der die Anzahl der entstehenden Cluster regelt (Alg. 2.6).

Algorithmus 2.6 Nächster-Nachbar Clustering

```

1: Input:  $\mathcal{S} = \{p_1, \dots, p_n\}$  samples,  $t$  maximum distance between neighbors
2: Output:  $\mathcal{C} = \{C_1, \dots, C_k\}$  clustering
3:  $C_{p_1} \leftarrow \{p_1\}$ 
4: for  $i \leftarrow 2$  to  $n$  do
5:    $p_j \leftarrow \text{nearest-neighbor}(p_1, \dots, p_{i-1}, t)$ 
6:   if  $\Delta(p_i, p_j) \leq t$  then
7:      $C_{p_j} \leftarrow C_{p_j} \cup \{p_i\}$ 
8:   else
9:      $C_{p_i} \leftarrow \{p_i\}$ 
10:  end if
11: end for
```

2. Clustering

2.3.4. Stochastisches Clustering

Die bislang behandelten Clusteringmethoden gingen davon aus, dass für Punkte $\mathcal{S} = \{p_1, \dots, p_n\}$ eine unbekannte Zerlegung $\mathcal{C} = \{C_1, \dots, C_k\}$ existiert und dass deren Clusteranzahl k bekannt ist. Letzteres ist in der Praxis aber nur selten der Fall: Im Allgemeinen muss die Anzahl der vorhandenen Cluster ebenso wie die unbekannte Partitionierung \mathcal{C} anhand der Beobachtungen x_1, \dots, x_n geschätzt werden.

Beim stochastischen Clustering ziehen wir die Analyse der Verteilungsdichte zur Konstruktion eines unbekannten Clusterings heran. Nehmen wir grundsätzlich an, dass für jedes der zu gruppierenden Objekte p_1, \dots, p_n d quantitative Merkmale in Form einer $n \times d$ Matrix vorliegen haben. So werden die n Objekte durch zugehörige n Beobachtungen $\mathcal{X} = \{X_1, \dots, X_n\} \in \mathbb{R}^d$, die in einer bestimmten Dichte zueinander stehen, repräsentiert. Dann sind einzelne Cluster, die mittels *unimodaler* Verteilung (s. Abb. 2.3) gekennzeichnet sind, durch Regionen geringerer Dichte von einander getrennt. Diese Feststellung ermöglicht das Erkennen von Clustern, die räumlich stark unterschiedliche Strukturen besitzen.

Wir wollen an dieser Stelle exemplarisch zwei Methoden vorstellen, die mit Hilfe der Dichteschätzung eine Zerlegung der Daten generieren.

Das Verfahren von Wishart

Im Verfahren von Wishart wird die Dichte an der Stelle X_k durch die Anzahl der Punkte $X_i \in \mathcal{X}$ mit der Entfernung $\Delta(X_k, X_i) \leq \tau$ geschätzt, dabei ist $\tau > 0$ eine vorgegebene Schranke. Es wird eine Gruppierung \mathcal{C}' konstruiert, die weitgehend dem unbekannten Clustering \mathcal{C} , das ein bestimmtes Dichteniveau besitzt, gleicht. Der Zusammenhangsbegriff³ in \mathbb{R}^d wird durch den Begriff des Zusammenhangs in einem Graphen \mathcal{G} mit Kantengewichten kleiner gleich τ ersetzt. Das entspricht der Forderung, dass zwei Punkte X_k und X_i mit $\Delta(X_i, X_k) \leq \tau$ in der gleichen Klasse von \mathcal{C} liegen sollen. Das entsprechende Verfahren umfasst folgende Schritte (s. Alg. 2.7).

Algorithmus 2.7 Das Verfahren von Wishart

- 1: **Input:** $\mathcal{X} = \{X_1, \dots, X_n\}$ samples, distance $\tau > 0$, density threshold $\zeta \geq 0$
 - 2: **Output:** $\mathcal{C} = \{C_1, \dots, C_k\}$ clustering
 - 3: **for all** $X_i \in \mathcal{X}$ **do**
 - 4: $\hat{f}(X_i) \leftarrow |\{X_j \in \mathcal{X} \mid \Delta(X_j, X_i) \leq \tau\}|$
 - 5: **end for**
 - 6: $U \leftarrow \{X_j \in \mathcal{X} \mid \hat{f}(X_j) \leq \zeta\}$
 - 7: $\bar{\mathcal{X}} \leftarrow \mathcal{X} \setminus U$
 - 8: $\mathcal{G} \leftarrow \text{graph}(\bar{\mathcal{X}}, \{(X_i, X_j) \mid X_i, X_j \in \bar{\mathcal{X}} \wedge \Delta(X_i, X_j) \leq \tau\})$
 - 9: $\mathcal{C} \leftarrow \text{find-connected-components}(\mathcal{G})$
-

Eine Verkleinerung des Parameters τ (bei festem ζ) bewirkt, dass die Anzahl der entfernten Punkte zunimmt und sich bereits gebildete Gruppen in Untergruppen

³ Eine offene Menge $G \subset \mathbb{R}^d$ heißt *zusammenhängend* (oder genauer *wegzusammenhängend*), wenn es zu je zwei Punkten $x, y \in G$ einen in G verlaufenden Weg gibt, der x und y verbindet.

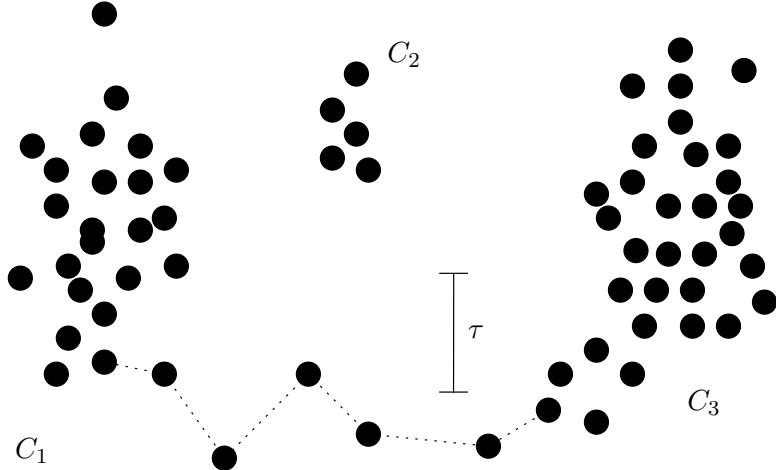


Abbildung 2.2.: Verkettungseigenschaft (chaining)

aufspalten. Wir können diesen Effekt zur Konstruktion von Hierarchien verwenden.

Für $\zeta = 0$ ergibt sich das wohlbekannte SINGLE-LINK Verfahren. Damit ist es möglich, dass mehrere getrennt liegende, in sich dicht gescharte Punktmengen, die jede eine eigene Klasse bilden sollten, durch das Vorhandensein vereinzelter Zwischenpunkte zu einer einzigen Gruppe vereinigt werden (vgl. Abb. 2.2). Das Verfahren von Wishart ($\zeta \geq 0$) vermeidet diesen Nachteil durch Entfernung der störenden Zwischenpunkte. Hierfür wird die Information über die Dichte der einzelnen Punkte ausgenutzt.

Das Verfahren von Wishart eignet sich zur Behandlung großer Datenmengen. Durch die Variation der Parameter ζ und τ können die Anzahl und die Homogenität der Gruppen weitgehend beeinflusst werden. Das Verfahren ist auch bei qualitativen oder gemischten Merkmalen anwendbar, weil die oben genannten Schritte nur eine Distanzmatrix verwenden.

Gradientenverfahren von Schnell

Das Verfahren von Schnell dient zur Bestimmung der *Modalwerte* (Abb. 2.3, 2.4, 2.5) der Dichte.

Definition 2.7. Die Dichte $f(x)$ heißt *unimodal*, wenn die Funktion $f(x)$ in \mathbb{R}^d nur ein einziges, lokales Maximum $f(Z)$ besitzt; die zugehörige Stelle $X_i = Z$ heißt der *Modalwert* oder *Modus* von f . Demgegenüber heißt eine Dichte $f(x)$ *multimodal*, wenn sie mehrere lokale Maxima mit zugehörigen Modalwerten Z_1, Z_2, \dots besitzt.

Das bedeutet, das Verfahren versucht jene Stellen im \mathbb{R}^d zu bestimmen, welche eine hohe Punktkonzentration aufweisen. Gleichzeitig verteilt es die Punkte X_i auf die Modalwerte, um so eine vollständige und disjunkte Gruppierung $\mathcal{C} = \{C_1, \dots, C_k\}$ zu erreichen. Dabei besteht die Klasse C_i aus den Punkten X_j , die demselben Modalwert Z_i angehören.

2. Clustering

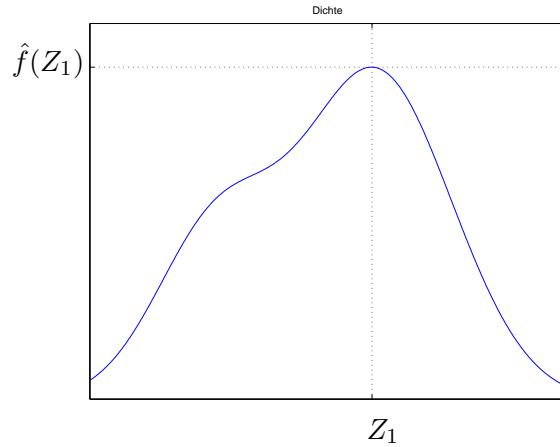


Abbildung 2.3.: Unimodale Verteilung in \mathbb{R} mit Modus Z_1

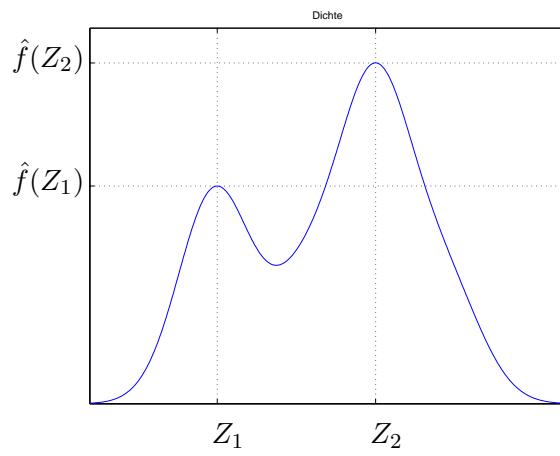


Abbildung 2.4.: Bimodale Verteilung in \mathbb{R} mit Modalwerten Z_1 und Z_2

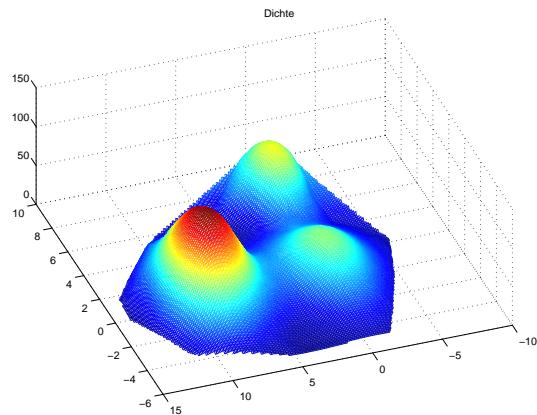


Abbildung 2.5.: Multimodale Verteilung im \mathbb{R}^2

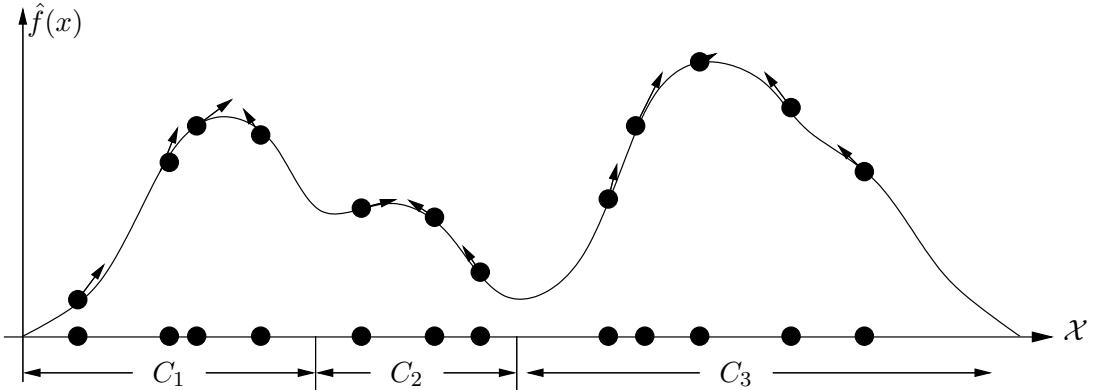


Abbildung 2.6.: Zum Verfahren von Schnell

Gehen wir davon aus, dass uns eine Schätzung \hat{f} der Dichtefunktion f vorliegt. Das Verfahren von Schnell findet die lokalen Maxima der Dichteschätzung \hat{f} , indem es alle Punkte X_i sukzessive in Gebiete höherer Punktedichte verschiebt. Hierbei rücken diese in die Nähe der gesuchten Modalwerte Z_1, Z_2, \dots (vgl. Abb. 2.6).

Algorithmus 2.8 Gradientenverfahren von Schnell

```

1: Input:  $\mathcal{X} = \{X_1, \dots, X_n\}$  samples,  $\hat{f}$  density estimation,  $\delta$  step width
2: Output:  $\mathcal{C} = \{C_1, \dots, C_k\}$  clustering
3: for all  $X_i \in \mathcal{X}$  do
4:    $C_i \leftarrow \{X_i\}$ 
5:    $t \leftarrow 0$ 
6:    $X_i^t \leftarrow X_i$ 
7:   repeat
8:      $t \leftarrow t + 1$ 
9:      $X_i^t \leftarrow X_i^{t-1} + \delta \frac{\nabla \hat{f}(X_i^{t-1})}{\|\nabla \hat{f}(X_i^{t-1})\|}$ 
10:    until  $\hat{f}(X_i^t) < \hat{f}(X_i^{t-1})$ 
11:     $X_i^* \leftarrow X_i^t$ 
12:   end for
13:   for all  $X_i, X_j$  with  $i \neq j$  do
14:     if  $\Delta(X_i^*, X_j^*) \leq 2\delta$  then
15:       merge( $\mathcal{C}, C_{X_i}, C_{X_j}$ )
16:     end if
17:   end for

```

Sei nun die Größe der Verschiebung durch ein $\delta > 0$ charakterisiert. Um nun die Verschiebung in die Richtung der Maxima zu realisieren, liegt es nahe, jeden der n Punkte in die Richtung des größten (oder auch des steilsten) Aufstiegs zu verschieben. Diese Richtung wird gerade durch den Gradientenvektor $\nabla \hat{f}$ vorgegeben. Dies

2. Clustering

führt zum folgenden Ablaufschema (s. Alg. 2.8).

Die Iteration wird solange durchgeführt, bis für das betreffende t keine weitere Verbesserung erreicht werden kann. Dies bedeutet, dass der Punkt $X_i^* := X_i^t$ i.a. in einer ε -Umgebung eines Modalwertes Z_ℓ von \hat{f} liegt.

Nach dem Abschluss der Iteration sind wir in der Lage festzustellen (vorausgesetzt die Schrittweite δ ist hinreichend klein), ob zwei Punkte X_i und X_j zum selben Modalwert Z_ℓ gehören. Dies ist genau dann der Fall, wenn für die entsprechend verschobenen Punkte X_i^* und X_j^* gilt:

$$\Delta(X_i^*, X_j^*) \leq 2\delta.$$

Das Gradientenverfahren von Schnell ist analytischer (im Gegensatz zu: kombinatorischer) Art und kann, da es relativ wenig Speicherplatz benötigt, bequem auch für größere Objektmengen durchgeführt werden. Pro Iteration muss jedoch die Dichteschätzung und die Berechnung des Gradienten durchgeführt werden.

Im Gradientenverfahren von Schnell stand uns die Dichteschätzung \hat{f} der Dichtefunktion f zur Verfügung. Im nächsten Kapitel werden wir Techniken erläutern mit deren Hilfe wir solch eine Schätzung gewinnen können.

3. Dichteschätzung

Hier werden die mathematischen Grundlagen der nichtparametrisierten Dichteschätzung vorgestellt.

3.1. Einleitung

Die Wahrscheinlichkeitsdichtefunktion ist ein grundlegendes Konzept der Statistik. Betrachten wir eine stetige Zufallsvariable X und deren Wahrscheinlichkeitsdichte f . Die Spezifizierung der Funktion f ermöglicht eine Aussage über die Wahrscheinlichkeitsverteilung der Zufallsgröße X . Dabei genügt die Wahrscheinlichkeit des Ereignisses „ $a < X < b$ “ innerhalb eines beliebigen Intervalls $[a, b]$ der folgenden Integralgleichung:

$$P(a < X < b) = \int_a^b f(x)dx \quad \text{für alle } a < b.$$

Nehmen wir an, uns steht eine Menge von Beobachtungen $\mathcal{X} = \{X_1, \dots, X_n\}$ zur Verfügung, welche einer unbekannten Verteilungsfunktion unterliegen. Die *Dichteschätzung* ist die Konstruktion einer Schätzung der unbekannten Wahrscheinlichkeitsdichte f aus den beobachteten Daten \mathcal{X} .

Zu diesem Thema erschien eine Vielzahl an Publikationen. Eine gute Einführung in die Thematik bietet [Silverman \(1986\)](#). Die theoretischen Aspekte der Dichteschätzung werden intensiv von [Rao \(1983\)](#) behandelt. Zuerst aber einige Begriffe der Schätztheorie.

3.2. Grundlagen der Schätztheorie

Definition 3.1. Um einen *statistischen Raum* zu beschreiben, benötigen wir:

1. eine nichtleere höchstens abzählbare Menge \mathcal{X} , den sogenannte Stichprobenraum
2. eine Familie $\{P_\theta \mid \theta \in \Theta\}$ von Wahrscheinlichkeitsmaßen auf \mathcal{X} ; hierbei nehmen wir an, dass $\Theta \subset \mathbb{R}^d$ für ein d und dass Θ ein verallgemeinertes Intervall ist.

Definition 3.2. Sei $g: \Theta \rightarrow \mathcal{Y}$ eine Funktion. Ein *Schätzer* für $g(\theta)$ ist eine Funktion $\hat{g}: \mathcal{X} \rightarrow \mathcal{Y}$ (im Falle von $\hat{g}(\theta) = \theta$ schreiben wir statt \hat{g} auch $\hat{\theta}$).

3. Dichteschätzung

Definition 3.3 (Likelihood). Ist \mathcal{X} eine diskrete Menge, so heißt die Funktion $\theta \rightarrow L_x(\theta) = P_\theta(x)$ Likelihood-Funktion. Sind $\mathcal{X} = \{X_1, \dots, X_n\}$ unabhängige Zufallsvariablen mit Werten in \mathbb{R}^d und $\{P_\theta | \theta \in \Theta\}$ eine Familie von Verteilungen von X_1, \dots, X_n . Ist P_θ verteilt mit einer d -dimensionalen Dichte $f(\cdot|\theta)$, so heißt die Funktion $\theta \rightarrow L_x = f(x|\theta)$ die *Likelihood-Funktion*. Nimmt $L_x(\cdot)$ einen Maximalwert in $\hat{\theta}(x)$ an, so dass

$$L_x(\hat{\theta}) = \sup\{L_x(\theta) | \theta \in \Theta\}$$

gilt, so nennen wir $\hat{\theta}(x)$ eine Maximum-Likelihood-Schätzung von θ und $g(\hat{\theta}(x))$ eine Maximum-Likelihood-Schätzung von $g(\theta)$. Betrachten wir die Funktion $\mathcal{L}_x := \log L_x$, wobei \log wie üblich den natürlichen Logarithmus bezeichnet, dann nennen wir diese *log-Likelihood-Funktion*.

Die log-Likelihood-Funktion wird häufig aus dem Grund der Monotonie der log-Funktion betrachtet und somit diese das Maximum an der gleichen Stelle wie L_x annimmt.

Definition 3.4 (Erwartungstreue, Bias). Ein Schätzer \hat{g} heißt *erwartungstreu*, wenn für alle $\theta \in \Theta$ die Gleichung

$$E_\theta [\hat{g}(X)] = g(\theta)$$

gilt. $b(\theta, \hat{g}) := E_\theta [\hat{g}(X)] - g(\theta)$ heißt *Bias* der Schätzung \hat{g} . Ein Schätzer ist also genau dann erwartungstreu¹, wenn sein Bias = 0 ist.

3.3. Parametrisierte und nichtparametrisierte Dichteschätzung

Wir unterscheiden bei der Dichteschätzung zwischen zwei Ansätzen:

Parametrisierter Ansatz: Angenommen, uns sind Beobachtungen einer bekannten parametrisierten Familie der Verteilungsfunktionen gegeben, z. B. der Normalverteilung mit dem Mittelwert μ und der Streuung σ^2 . Die Dichtefunktion f , die den Daten unterliegt, kann durch eine Schätzung der Größen μ und σ^2 aus den Daten und dem Einsetzen der Schätzungen in die Formel für die Normalverteilung rekonstruiert werden.

Die Annahme einer bestimmten Verteilungsfunktion entspricht in einigen Fällen nicht der Realität. Dies führt dann zu Schätzfehlern.

Nichtparametrisierter Ansatz: Es werden keine Annahmen über die Parameter der Dichtefunktion f gemacht. Vielmehr versuchen wir die Schätzung der Dichtefunktion aus den vorliegenden Daten \mathcal{X} herzuleiten. Die Idee wurde zum ersten Mal von [Fix und Hodges \(1951\)](#) vorgestellt.

¹engl.: unbiased

3.4. Nichtparametrisierte Dichteschätzungstechniken

Die Techniken und Methoden, die im Folgenden erläutert werden, werden zuerst für den univariaten Fall erklärt. Darauffolgend verallgemeinern wir die Ansätze und gehen dabei auf die Probleme der Dichteschätzung in mehrdimensionalen Räumen ein.

3.4.1. Univariater Fall

Histogramm

Der älteste und weitverbreitete Dichteschätzer ist das Histogramm. Gegeben sind Ursprung x_0 und die Klassengröße $2h$. Wir definieren ein Intervall $[x_0 + 2hm, x_0 + 2h(m + 1))$ als eine *Histogrammkasse* für jede ganze Zahl m . Die Dichteschätzung \hat{f} ist also wie folgt definiert:

$$\hat{f}(x) = \frac{1}{2hn} (\text{Anzahl } X_i \text{ in der selben Klasse wie } x).$$

Das Histogramm können wir als Klassifikation der relativen Häufigkeiten auffassen (vgl. Abb. 3.2). Um das Histogramm zu konstruieren, müssen wir sowohl den Ursprung als auch die Klassengröße festlegen, wobei die Klassengröße die Glattheit der Schätzung festlegt. Formal, unter einer beliebigen Aufteilung der Daten in einzelne Klassen, gilt:

$$\hat{f}(x) = \frac{1}{n} \times \frac{(\text{Anzahl der } X_i \text{ in derselben Klasse wie } x)}{\text{Größe der Klasse, die } x \text{ beinhaltet}}.$$

Das Histogramm ist eine exzellente Methode für die Dichteschätzung, jedoch ist die Methode mit einigen Nachteilen verbunden. Die Genauigkeit der Schätzung kann zwar durch eine Verfeinerung der Klassengröße verbessert werden, dennoch existieren Methoden, die eine weit bessere Genauigkeit der Dichteschätzung ermöglichen. Hinzu kommt, dass die Methode, bedingt durch die Sprungstellen, nicht in Anwendungen, welche die Ableitung der Dichte benötigen, verwendet werden kann.

Naiver Schätzer

Aus der Definition der Wahrscheinlichkeitsverteilung einer Zufallsvariable X folgt für die Dichte f :

$$f(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(x - h < X < x + h).$$

Für jedes h sind wir in der Lage die Wahrscheinlichkeit $P(x - h < X < x + h)$ durch den Anteil der Beobachtungen aus dem Intervall $(x - h, x + h)$ abzuschätzen. Wählen wir nun h genügend klein, so ergibt sich eine Schätzung \hat{f} wie folgt:

$$\hat{f}(x) = \frac{1}{2nh} (\text{Anzahl der } X_i \in \mathcal{X} \text{ in der Klasse } (x - h, x + h)).$$

3. Dichteschätzung

Um den Schätzer besser beschreiben zu können, definieren wir eine Gewichtsfunktion ψ :

$$\psi(u) = \begin{cases} \frac{1}{2}, & \text{falls } u \in [-1, 1), \\ 0, & \text{sonst.} \end{cases}$$

Damit kann der *naive* Schätzer neu formuliert werden:

$$\hat{f}(u) = \frac{1}{hn} \sum_{i=1}^n \psi\left(\frac{u - X_i}{h}\right). \quad (3.1)$$

Es folgt aus der Definition von ψ , dass die Dichteschätzung sich aus der Platzierung einer „Box“ mit der Breite $2h$ und einer Höhe von $(2nh)^{-1}$ über jedem Datenpunkt und der Aufsummierung der Höhenwerte ergibt (s. Abb. 3.2). Lägen alle Punkte in der Mitte der Klassen, so würde der naive Schätzer das Histogramm widerspiegeln.

Der naive Schätzer hat weiterhin die Nachteile des Histogramms. Die Schätzung \hat{f} ist keine stetige Funktion, da diese Sprungstellen an den Stellen $X_i + h$ aufweist. Hinzu kommt, dass an allen anderen Stellen die Ableitung verschwindet. Eine natürliche Verallgemeinerung des naiven Schätzers wird im nächsten Abschnitt vorgestellt.

Kerndichteschätzung

Ersetzen wir die Gewichtsfunktion ψ in (3.1) durch eine symmetrische Funktion, welche den Bedingungen:

$$\int_{\mathbb{R}} K(u)du = 1, \quad (3.2)$$

$$\int_{\mathbb{R}} u^j K(u)du = 0, \text{ für } j = 1, \dots, k-1, \quad (3.3)$$

$$\int_{\mathbb{R}} u^k K(u)du = C_k \neq 0 \quad (3.4)$$

genügt, so erhalten wir eine Schar von Dichteschätzern. Derartige Schätzer nennen wir *Kerntyp-Schätzer* oder einfach *Kern* der Ordnung k .

Zum Beispiel stellt die Dichtefunktion $\Phi(x)$ der Standard-Normalverteilung $N(0, 1)$ einen Kernschätzer der 2-ten Ordnung dar. Die Bedingungen (3.2), (3.3) charakterisieren Φ als eine symmetrische Dichtefunktion mit einer Konstante C_k , welche der Varianz der Normalverteilung entspricht.

Die gängigsten Kernfunktionen sind in der Tabelle 3.1 aufgelistet und einige typische Plots in der Abbildung 3.1 dargestellt. Formal:

$$\hat{f}(u) = \frac{1}{hn} \sum_{i=1}^n K\left(\frac{u - X_i}{h}\right). \quad (3.5)$$

Analog zum naiven Schätzer kann der Kernschätzer als eine Aufsummierung von „Hütchen“ oder „Beulen“ über den Beobachtungspunkten angesehen werden. Die Kernfunktion K bestimmt dabei die Form der „Hütchen“ und h deren Breite. Die

3.4. Nichtparametrisierte Dichteschätzungsstechniken

Kernfunktion	$K(u)$	$\text{eff}(K)$
Epanechnikov	$\frac{3}{4}(1 - u ^2)$	$I(u \leq 1)$ 1
Quadrat	$\frac{15}{16}(1 - u ^2)^2$	$I(u \leq 1)$ $\sqrt{3087/3125} \approx 0.9939$
Dreieck	$(1 - u)$	$I(u \leq 1)$ $\sqrt{243/250} \approx 0.9859$
Rechteck	$\frac{1}{2}$	$I(u \leq 1)$ $\sqrt{198/125} \approx 0.9295$
Gauß	$\frac{1}{\sqrt{2\pi}}e^{-0.5u^2}$	$\sqrt{36\pi/125} \approx 0.9512$
Kubik	$\frac{35}{32}(1 - u ^2)^3$	$I(u \leq 1)$
Kosinus	$\frac{\pi}{4} \cos(\frac{\pi}{2}u)$	$I(u \leq 1)$
Direchlet	$\frac{\sin(2n+1)\pi u}{\sin \pi u}$	$I(u \leq 1)$
Fejér	$\frac{1}{n+1}(\frac{\sin(2n+1)\pi u}{\sin \pi u})$	$I(u \leq 1)$

Tabelle 3.1.: Die gängigsten Kernfunktionen und deren Effizienz. $I(\cdot)$ ist die Indikatorkfunktion.

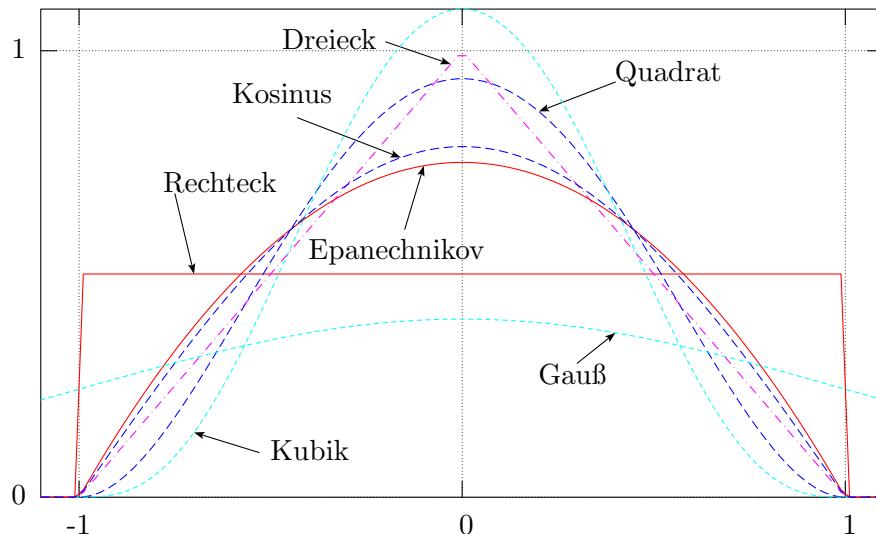


Abbildung 3.1.: Plots der typischen Kernfunktionen

3. Dichteschätzung

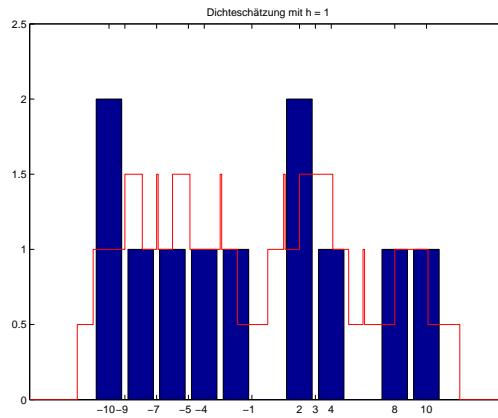


Abbildung 3.2.: Histogramm (blau) und der naive Schätzer (rot)

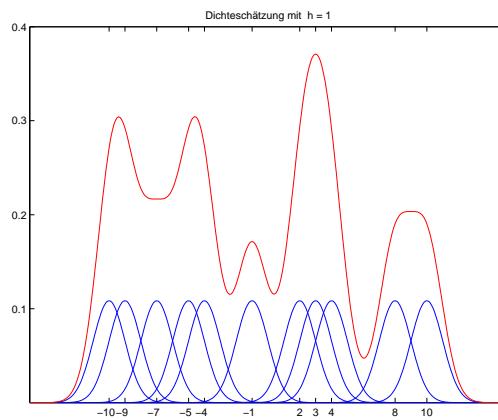


Abbildung 3.3.: Konstruktion der Dichtefunktion (rot) vermöge Kernschätzers. Hier am Beispiel der Gaußfunktion (blau).

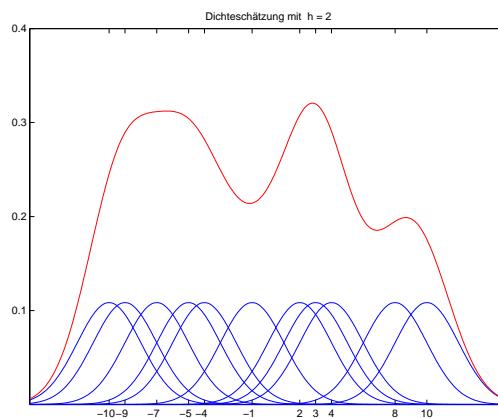


Abbildung 3.4.: Dichteschätzung mit einer größeren Bandweite.

3.4. Nichtparametrisierte Dichteschätztechniken

Bilder 3.3 und 3.4 veranschaulichen die Methode. Die Größe h wollen *Bandweite*, gelegentlich auch *Suchfenster*, nennen.

Alle Eigenschaften des Kernschätzers folgen aus seiner Definition. D.h. alle Stetigkeits- und Differenzierbarkeitseigenschaften der Dichtefunktion erfüllt auch der Kernschätzer. Ein Beispiel: Legen wir die Normalverteilung als Kernfunktion zu Grunde, so ist auch \hat{f} unendlich oft differenzierbar.

Abgesehen von dem Histogramm ist der Kernschätzer wahrscheinlich der meist eingesetzte Schätzer. Obwohl auch er, bedingt durch die fixe Bandweite, einige Nachteile mit sich bringt. Diese äußern sich vor allem bei der Schätzung von Dichtefunktionen mit langen Tälern. In diesen Fällen neigt der Kernschätzer zum „Rauschen“. Die Kernschätzer, welche diese Schwierigkeiten meistern, werden nun in den nächsten beiden Abschnitten vorgestellt.

k -Nächster-Nachbar Methode

Die k -Nächster-Nachbar Methode kann als ein Versuch angesehen werden, die Dichtefunktion *lokal* zu bestimmen. Wobei der Parameter k die Glattheitseigenschaft der Schätzung beeinflusst. Dieser sollte kleiner als die Anzahl der Beobachtungen gewählt werden (typischerweise $k \approx n^{1/2}$). Sei nun wie vereinbart $\Delta(x, y)$ die Distanz zwischen zwei Punkten. Für jeden Punkt t definieren wir

$$d_1(t) \leq d_2(t) \leq \dots \leq d_n(t)$$

die aufsteigend angeordnete Distanzwerte zwischen dem Punkt t und allen anderen Punkten.

Die k -Nächster-Nachbar Dichteschätzung ist definiert durch:

$$\hat{f}(t) = \frac{k-1}{2nd_k(t)}. \quad (3.6)$$

Zum Verständnis: Die Dichte des Punktes t ist $f(t)$. Für n Beobachtungen wird erwartet, dass $2hnf(t)$ der n Werte ins Intervall $[t-h, t+h]$ für jedes $h > 0$ fallen (siehe Abschnitt **Naiver Schätzer**). Nach der Definition fallen $k-1$ Beobachtungen ins Intervall $[t-d_k(t), t+d_k(t)]$. Somit ergibt sich also:

$$k-1 = 2d_k(t)n\hat{f}(t).$$

Durch die Umstellung der Gleichung erfolgt die Definition (3.6). Während beim naiven Schätzer die Bandweite fest gehalten und die Anzahl der Beobachtungen variiert, fixieren wir bei der k -Nächster-Nachbar Methode die Anzahl der Beobachtungen und variieren die Bandweite.

Es ist möglich die k -Nächster-Nachbar Methode, ähnlich der Kernschätzmethoden, zu verallgemeinern. Sei wiederum K eine Kernfunktion wie beschrieben, dann lässt sich die *allgemeine* k -Nächster-Nachbar Methode formal wie folgt formulieren:

$$\hat{f}(t) = \frac{1}{nd_k(t)} \sum_{i=1}^n K\left(\frac{t-X_i}{d_k(t)}\right). \quad (3.7)$$

3. Dichteschätzung

Offensichtlich ist \hat{f} eine Auswertung des Kernschätzers an der Stelle t mit der Bandweite $d_k(t)$.

Variabler Kernschätzer

Die variable Methode ist ähnlich der k -Nächster-Nachbar Methode und ein weiterer Versuch der lokalen Näherung der Dichtefunktion. Die Schätzung basiert auf der gewöhnlichen Kernschätzung, jedoch erlauben wir uns die Bandweite von Punkt zu Punkt zu variieren.

Sei K eine Kernfunktion und k eine positive ganze Zahl. Definiere d_{jk} als Distanz zwischen X_j und dem k nächsten Nachbarn. Darauffolgend definieren wir die *variable Dichteschätzung* mittels:

$$\hat{f}(t) = \frac{1}{n} \sum_{j=1}^n \frac{1}{hd_{jk}} K\left(\frac{t - X_j}{hd_{jk}}\right). \quad (3.8)$$

Die Bandweite h des Kerns für den Punkt X_j ist proportional zu d_{jk} , somit werden die Punkte in schwachbesetzten Regionen mit einem flachen Kern geschätzt. Die Glattheitseigenschaft der Schätzung bestimmt weiterhin die Bandweite h . Der Parameter k kontrolliert, in wie weit Lokalität als Eigenschaft in die Schätzung einfließt.

3.4.2. Multivariater Fall

Kernschätzung

Die Definition der Kernschätzung kann sehr leicht auf den mehrdimensionalen Fall verallgemeinert werden:

$$\hat{f}(\mathbf{u}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{u} - \mathbf{X}_i}{h}\right). \quad (3.9)$$

Wobei die Kernfunktion $K(x)$ nun für den d -dimensionalen Raum definiert ist:

$$\int_{\mathbb{R}^d} K(\mathbf{x}) d\mathbf{x} = 1. \quad (3.10)$$

Ferner wird bei der Benutzung der skalaren Größe h die Normierung des Kerns, welcher über den Beobachtungspunkten plaziert wird, in alle Richtungen vorausgesetzt. In einigen Fällen ist es jedoch sinnvoll einen Bandweitenvektor bzw. eine Matrix zu verwenden (siehe auch Comaniciu und Meer (2002)). Das ist dann der Fall, wenn die Ausbreitung der Punkte in eine Richtung wesentlich größer ist als in eine andere. Die Normierung sollte fast immer vorgezogen werden, denn dies verhindert unterschiedliche Variationen der Daten und vereinfacht zudem die Form des Kernschätzers.

k -Nächster-Nachbar Schätzung

Wir definieren $h_k(t)$ als die Distanz von Punkt t zum k nächsten Nachbarn. Sei weiterhin $V_k(t)$ Volumen einer d -dimensionalen Kugel mit dem Radius $h_k(t)$, folglich

3.4. Nichtparametrisierte Dichteschätztechniken

$V_k(t) = c_d h_k(t)^d$, wobei c_d das Volumen der Einheitskugel der Dimension d ($c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$ usw.) darstellt. Somit kann die k -Nächster-Nachbar Dichteschätzung wie folgt definiert werden:

$$\hat{f}(\mathbf{t}) = \frac{k/n}{V_k(\mathbf{t})} = \frac{k/n}{c_d h_k(\mathbf{t})^d}. \quad (3.11)$$

Die Begründung aus (3.6) kann auch hier auf den mehrdimensionalen Fall erweitert werden. Von n Beobachtungen werden $n f(\mathbf{t}) V_k(\mathbf{t})$ Beobachtungen, die in die Kugel mit dem Radius $h_k(\mathbf{t})$ und dem Ursprungspunkt \mathbf{t} fallen, erwartet. Setzen wir die Anzahl gleich k , so erhalten wir die Schätzung (3.11).

Die allgemeine k -Nächster-Nachbar Methode kann analog zum eindimensionalen Fall formuliert werden. Betrachten wir die Kernschätzung mit folgender Kernfunktion:

$$K(\mathbf{x}) = \begin{cases} c_d^{-1}, & \text{für } |\mathbf{x}| \leq 1, \\ 0, & \text{sonst.} \end{cases} \quad (3.12)$$

Dann ist (3.11) die Auswertung der Kernfunktion an der Stelle \mathbf{t} mit der Bandweite $h_k(\mathbf{t})$. Damit ist die allgemeine k -Nächster-Nachbar Schätzung eine Verallgemeinerung für eine beliebige Kernfunktion mit der Gestalt:

$$\hat{f}(\mathbf{t}) = \frac{1}{n h_k(\mathbf{t})^d} \sum_{i=1}^n K\left(\frac{\mathbf{t} - \mathbf{X}_i}{h_k(\mathbf{t})}\right). \quad (3.13)$$

Schwierigkeiten

Bisher nahmen wir stillschweigend an, dass die Dichteschätzung im mehrdimensionalen Fall analog zum eindimensionalen Fall erfolgt. An dieser Stelle wollen wir zwei Phänomene vorstellen, welche die Schwierigkeit der multidimensionalen Dichteschätzung verdeutlichen.

Das erste Phänomen, das von Bellmann im Jahr 1961 unter dem Begriff *curse of dimensionality* geprägt wurde, besagt, dass für die Genauigkeit einer Schätzung einer Funktion mit mehreren Veränderlichen, unter Weglassen einfacher Annahmen, die Anzahl der notwendigen Beobachtungspunkte exponentiell anwächst. Beispielsweise basieren die meisten Dichteschätzer auf dem Mittel der Beobachtungen aus der direkten Nachbarschaft. Bei der Suche nach einer ausreichenden Anzahl der Nachbarn in mehrdimensionalen Räumen kann jedoch die Lokalität verloren gehen.

Ein ähnliches Phänomen, das den Fluch der Dimensionalität begründet, ist der *empty space* Effekt (Scott und Thompson (1983)): Hochdimensionale Räume sind schwach besetzt. So beträgt z.B. die Wahrscheinlichkeit für einen Punkt, der der Gleichverteilung unterliegt, dass er in der 10-dimensionalen Einheitskugel liegt und höchstens 0.9 vom Zentrum entfernt ist, nur 0.35. Das ist ein schwerwiegendes Problem in der multivariaten Dichteschätzung. Denn Regionen mit schwach ausgeprägter Dichte können einen beachtlichen Anteil der Beobachtungen beinhalten. Dagegen beherbergen dicht besetzte Regionen scheinbar nur einen Bruchteil dieser Punkte.

3. Dichteschätzung

Dies wird am Beispiel der Normalverteilung klar. Im eindimensionalen Fall befinden sich 70% der normalverteilten Punkte in der Einheitskugel. In dem 10-dimensionalen Raum sind es hingegen nur 0.02%. Um die 70% zu erhalten, müssen wir den Radius der Kugel verdreifachen. Somit bestätigen diese Beobachtungen, dass den Tälern in den mehrdimensionalen Räumen eine größere Bedeutung als den in eindimensionalen zukommt.

3.5. Approximationseigenschaften

Um die Genauigkeit der Schätzung beurteilen zu können, benötigen wir eine Distanzfunktion zwischen der wahren Dichtefunktion und der Dichteschätzung. Wir definieren an dieser Stelle den *integrierten quadratischen Fehler*:

$$ISE(h) = \int (\hat{f}(x) - f(x))^2 dx \quad (3.14)$$

und dessen Erwartungswert:

$$MISE(h) = E \left[\int (\hat{f}(x) - f(x))^2 dx \right]. \quad (3.15)$$

An dieser Stelle sei angemerkt, dass die Notation *nur* von der Bandweite h abhängt und nicht von der Kernfunktion K . Es gibt für diesen Sachverhalt in erster Linie zwei Gründe: zum einen vereinfacht es die Notation und zum anderen kommt der Wahl der Bandweite weitaus größere Bedeutung zu.

Im Weiteren wollen wir mit h_{ISE}^* bzw. h_{MISE}^* die Bandweiten bezeichnen, für die der jeweilige Fehler minimal ist.

Aus der Definition des *MISE* Fehlers folgt sofort:

$$MISE(h) = E \left[\int (\hat{f}(x) - f(x))^2 dx \right] \quad (3.16)$$

$$= \int E[(\hat{f}(x) - f(x))^2] dx \quad (3.17)$$

$$= \int E[(\hat{f}(x) - E[\hat{f}(x)] + E[\hat{f}(x)] - f(x))^2] dx \quad (3.18)$$

$$= \int E[(\hat{f}(x) - E[\hat{f}(x)])^2 + (E[\hat{f}(x)] - f(x))^2] dx \quad (3.19)$$

$$= \int \text{Var } \hat{f}(x) dx + \int \text{bias}^2 \hat{f}(x) dx. \quad (3.20)$$

Diese Zerlegung des mittleren integrierten quadratischen Fehlers demonstriert das zentrale Problem der Kernschätzung. Um das *Bias* zu eliminieren, sollten wir die Bandweite h möglichst klein wählen (Bestrafung der Überschätzung). Die Varianz dagegen wird mit wachsendem nh klein (Bestrafung der Unterschätzung). Mit anderen Worten, eine gute Schätzung erreichen wir mittels einer kleinen Bandweite und ausreichender Anzahl der Beobachtungen.

3.5. Approximationseigenschaften

Nehmen wir an, dass die Dichtefunktion f mindestens $k + 2$ beschränkte Ableitungen besitzt und die Kernfunktion der Ordnung k ist, dann definieren wir einen *asymptotisch* mittleren integrierten Fehler:

$$AMISE(h) = \frac{1}{nh} \int K^2(x) dx + h^{2k} \left(\frac{\int x^k K(x) dx}{k!} \right)^2 \int (f^{(k)})^2(x) dx. \quad (3.21)$$

Für ein $h \rightarrow 0$ und $nh \rightarrow \infty$ folgt, dass $MISE(h) = AMISE(h) + o(AMISE(h))$. Weiterhin wollen wir mit h_{AMISE}^* die Bandweite, für die der $AMISE$ Fehler minimal ist, bezeichnen.

Aus der Bias Definition folgt:

$$\text{bias}(x) = E[f(x)] - f(x) \quad (3.22)$$

$$= \int h^{-1} K((x-y)/h) f(y) dy - f(x). \quad (3.23)$$

Ändern wir die Variable $y = x - h$ und nehmen wir $\int K(t) dt = 1$ an, so können wir schreiben:

$$\text{bias}(x) = \int K(t) f(x - ht) dt - \underbrace{f(x)}_{f(x) \int K(t) dt} \quad (3.24)$$

$$= \int K(t) (f(x - ht) - f(x)) dt. \quad (3.25)$$

Aus der Taylor Reihenentwicklung

$$f(x - ht) = f(x) - h t f'(x) + \frac{1}{2} h^2 t^2 f''(x) + \dots$$

und den Annahmen über K folgt:

$$\text{bias}(x) = -h \underbrace{f'(x) \int t K(t) dt}_{=0} + \frac{1}{2} h^2 t^2 f''(x) \underbrace{\int t^2 K(t) dt}_{=:k_2} + \dots \quad (3.26)$$

$$= \frac{1}{2} h^2 f''(x) k_2 + \text{Terme der höheren Ordnung in } h. \quad (3.27)$$

Somit ergibt sich für die symmetrischen Kerntypschatzer der 2-ten Ordnung aus der Taylorentwicklung (3.27) und (3.20) für das Bias:

$$\int \text{bias}(x)^2 \approx \frac{h^4 k_2^2}{4} \int f''(x)^2 dx \quad (3.28)$$

und für die Varianz:

$$\int \text{Var } \hat{f}(x) dx \approx \frac{1}{nh} \int K(t)^2 dt. \quad (3.29)$$

3. Dichteschätzung

Die ideale Bandweite, welche den asymptotisch mittleren integrierten Fehler:

$$\frac{1}{4}h^4k_2^2 \int f''(x)^2 dx + \frac{1}{nh} \int K(t)^2 dt \quad (3.30)$$

minimiert, ist mit

$$h_{AMISE}^* = k_2^{-2/5} \left[\int K(t)^2 dt \right]^{1/5} \left[\int f''(x)^2 dx \right]^{-1/5} n^{-1/5} \quad (3.31)$$

gegeben (nach Parzen (1962)).

Das Rücksinsetzen des Wertes h_{AMISE}^* aus (3.31) in (3.30) liefert für $MISE$, wenn die Bandweite h optimal gewählt ist, einen approximativen Wert des mittleren integrierten Fehlers:

$$\frac{5}{4}C(K) \left[\int f''(x)^2 dx \right]^{1/5} n^{-4/5}, \quad (3.32)$$

wobei die Konstante $C(K)$ durch

$$C(K) = k_2^{2/5} \left(\int K(t)^2 dt \right)^{4/5} \quad (3.33)$$

gegeben ist.

Unter den nichtnegativen Kernfunktionen minimiert die Epanechnikov Kernfunktion (Abb. 3.1) $C(K)$. Somit können wir den Effizienzkoeffizienten für eine Kernfunktion wie folgt definieren:

$$\text{eff}(K) = \left(\frac{C(K_{\text{Epanechnikov}})}{C(K)} \right)^{\frac{5}{4}}. \quad (3.34)$$

Die Effizienz einiger Kernfunktionen ist in der Tabelle 3.1 aufgelistet. Wie wir sehen, ist die Effizienz der meisten Kernfunktionen nahe Eins. Aus asymptotischer Sicht ist die Wahl der Kernfunktion zweitrangig und sollte an anderen Kriterien fest gemacht werden, z. B. am Grad der Differenzierbarkeit.

4. BST-Algorithmus

In den Kapiteln 2 und 3 lernten wir sowohl die Grundlagen des stochastischen Clusterings als auch der Dichteschätzung kennen. Hier wollen wir einen Algorithmus vorstellen, der, vermöge der Kerndichteschätzung, das Clustern mehrdimensionaler Daten ermöglicht. Nachfolgend wollen wir die Vor- und Nachteile des Ansatzes diskutieren.

Der BST-Algorithmus (Alg. 4.1) bildet die Grundlage der Toolbox und ist ausführlich bei [Hader und Hamprecht \(2003\)](#) beschrieben. Wir folgen dieser Darstellung.

4.1. Analyse

In Kapitel 2.3.4 lernten wir das Gradientenverfahren von Schnell kennen. In diesem wird jeder Punkt in Richtung des steilsten Anstiegs der Dichte geschoben bis der Punkt eines der lokalen Maxima erreicht. Dabei ist die naive Implementierung der Iteration sehr aufwendig. Die Kosten der Gradientenauswertung eines Punktes sind $\mathcal{O}(n)$ und müssen mehrfach wiederholt werden bis einer der Maximapunkte erreicht ist. Nehmen wir an, dass jeder Punkt ein lokales Maximum in t Schritten erreicht, zu $\mathcal{O}(n)$ Kosten, dann beläuft sich der Gesamtaufwand für die Auswertung des Gradienten auf $\mathcal{O}(nt)$. Aus dieser Überlegung können wir direkt folgern, es die Anzahl der Schritte t bei einer gegebenen Schrittweite δ zu minimieren. Um die Konvergenzbedingung im Verfahren von Schnell zu gewährleisten, schlug [Kowalewski \(1995\)](#) folgende Strategie vor: Solange wie möglich, die Schrittweite so kurz wie notwendig zu halten.

Ein weiterer nahe liegender Ansatz für die Beschleunigung der Operation ist die Reduzierung der Daten und somit die Reduzierung der Genauigkeit der Dichteschätzung ([Domeniconi und Gunopoulos \(2001\)](#)). Dies ist jedoch, aus Gründen die wir im Kapitel 3.4.2 angaben, nicht immer möglich.

Ferner wird der Aufwand der Berechnungen minimiert, in dem die Kernfunktion mit einer beschränkten Trägermenge (Tabelle 3.1) eingesetzt wird. [Hinneburg und Keim \(1998\)](#) schlugen die Zerlegung des Raumes in Teilräume als eine weitere Optimierungstechnik vor.

Wir wollen im Folgenden eine „diskrete“ Variante des Schnell-Verfahrens vorstellen, die gänzlich auf die mehrmalige Auswertung der Kernfunktion als auch auf die des Gradienten verzichtet.

4.2. Basin Spanning Trees

Der Algorithmus, der hier vorgestellt wird, wertet die Kernfunktion (bzw. den Gradienten) nur einmal aus, dabei wird der Mehraufwand in das Vorverarbeiten der Daten investiert. Der graphentheoretische Ansatz, der die Grundlage des BST-Algorithmus bildet, wird von [Fukunaga \(1990\)](#) erläutert. Dieser kann grob in vier Schritte unterteilt werden (vgl. Alg. 4.1):

- Wie in Kapitel 3.1 beschrieben, wird im ersten Schritt für jeden Punkt die Dichteschätzung mittels eines Kernschätzers bestimmt.
- Im zweiten Schritt des Algorithmus wird die topologische Nachbarschaft der Punkte berechnet. An dieser Stelle können unterschiedliche Definitionen der Nachbarschaft, z.B. k -Nächster-Nachbar, relative Nachbarschaft oder die Delaunay-Triangulierung, herangezogen werden.
- Im Schritt drei bilden wir einen Graphen. Hierbei ziehen wir eine gerichtete Kante von einem Punkt zu einem seiner Nachbarn genau dann, wenn folgende Bedingungen von diesem erfüllt werden:
 1. er besitzt eine größere Dichte als der Startpunkt,
 2. die Dichtefunktion hat den steilsten Anstieg unter allen Nachbarn des Startpunktes, welche der Bedingung 1 genügen.

Die beiden Bedingungen garantieren das ein azyklischer Graph mit Ausgangsgrad höchstens 1 entsteht. Ferner nimmt die Dichtefunktion an den Punkten mit Ausgangsgrad 0 ihre Maxima an und somit bilden diese Punkte die Wurzeln der entstehenden Bäume.

- Nun werden die einzelnen Cluster durch die Wurzeln der Bäume identifiziert. Die Clusterzugehörigkeit der Punkte wird durch die Zugehörigkeit zum jeweiligen Baum definiert.

Da entstehende Bäume die *Becken* der Dichtelandschaft aufspannen, wollen wir sie *Basin Spanning Trees* nennen. Die einzelne Schritte sind in den Abbildungen 4.1, 4.2, 4.3 und 4.4 dargestellt.

4.3. Nachbarschaft

Aus dem Algorithmus wird klar, dass der Nachbarschaft eine zentrale Rolle kommt. Gehen wir von einem vollständigen Graphen aus, so werden Punkte, deren geschätzte Dichte nah beim Maximum liegt, direkt über die Täler der Dichtefunktion, und somit über die Clustergrenzen hinweg, mit den lokalen Maxima verbunden. Hinzu kommt, dass eine große Nachbarschaft die Laufzeit der Methode beeinträchtigt.

Algorithmus 4.1 Basin Spanning Trees Algorithmus

```

1: Input:  $\mathcal{X}$  samples
2: Output:  $\mathcal{C}$  clustering
3: for all  $X_i \in \mathcal{X}$  do
4:    $kde(X_i) \leftarrow \text{kernel-density-estimation}(X_i)$ 
5: end for
6:  $\mathcal{N} \leftarrow \text{delaunay-triangulation}(\mathcal{X})$ 
7: for all  $X_i \in \mathcal{X}$  do
8:    $X_k \leftarrow \text{find-steepest-gradient}(X_i, \mathcal{N}(X_i))$ 
9:    $bst(X_i) \leftarrow X_k$ 
10: end for
11: for all  $X_i \in \mathcal{X}$  do
12:   if  $bst(X_i) = \emptyset$  then
13:      $C_{X_i} \leftarrow \{X_i\}$ 
14:   else
15:      $C_{bst(X_i)} \leftarrow C_{bst(X_i)} \cup \{X_i\}$ 
16:   end if
17: end for

```

Die Delaunay-Triangulierung liefert eine Menge von Simplizes mit der Eigenschaft, dass Sphären, in welche die Simplizes einbeschrieben sind, keine weiteren Punkte beinhalten. In anderen Worten: die Delaunay-Triangulierung bestimmt die nächsten Nachbarn aus unterschiedlichen Richtungen. Da die Komplexität der Delaunay-Triangulierung $\mathcal{O}(n^{d/2})$ beträgt¹, eignet sich diese Vorgehensweise nur für Daten der Dimension kleiner gleich vier. In höheren Dimensionen ist die k -Nächster-Nachbar Nachbarschaft effizienter zu bestimmen. Wobei diese auch einige Nachteile mit sich bringt. Denn bei einer zu kleinen Wahl von k erhalten wir eine unvollständige Nachbarschaft. Dagegen führt eine zu große Wahl von k oft zum Verlust der Lokalität. Wir erhalten mit k einen weiteren Freiheitsgrad, den es zu optimieren gilt.

4.4. Topographie

Außer der Nachbarschaft entscheidet auch der Verlauf der Dichteschätzung über die Zugehörigkeit zweier Punkte zur selben Klasse, denn die Dichte auf der gerichteten BST-Kante darf nicht unter das Niveau am Startknoten fallen. Andernfalls ist dies ein Indiz für den Verlauf der Clustergrenze zwischen den Punkten der Kante. Formal ausgedrückt, jede zulässige BST-Kante (X_i, X_j) muß folgende Bedingung erfüllen:

$$\forall \lambda \in [0, 1]: \hat{f}(X_i + \lambda(X_j - X_i)) \geq \hat{f}(X_i).$$

Die Verifikation der Bedingung durch die Hinzunahme zusätzlicher Abtastpunkte entlang einer Kante und dem Vergleich deren Dichte mit dem des Startpunktes wür-

¹Hier für Qhull-Algorithmus von Barber u. a. (1996)

4. BST-Algorithmus

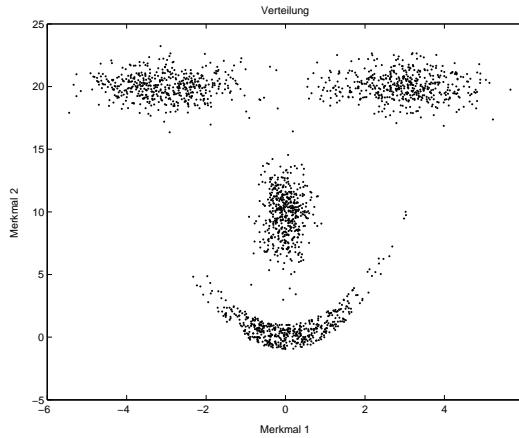


Abbildung 4.1.: Gegeben sind Punkte aus einem \mathbb{R}^d Raum.

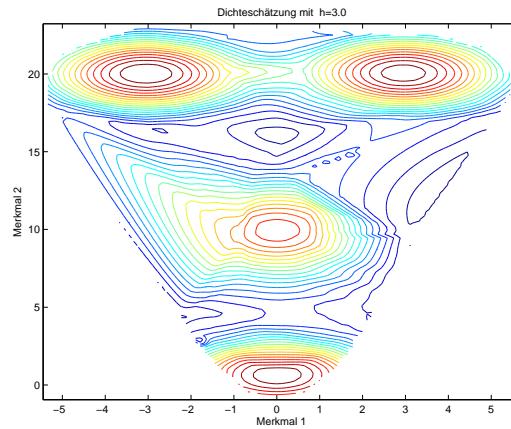


Abbildung 4.2.: Schritt 1: Die Dichteschätzung vermöge Kernschätzers. Hier mittels Höhenlinien angedeutet.

de in einem hohen rechnerischen Aufwand resultieren. Da alle Dichteschätzungen an den Abtastpunkten zusätzlich erfolgen müssen, bzw. nicht aus schon bekannten abgeleitet werden können. Folglich sind approximative und schnelle Verfahren gefragt.

Eine einfache Abschätzung erhalten wir bei der Betrachtung des lokalen Gradienten am Startpunkt. Weist die Ableitung der Dichte entlang der BST-Kante keine positive Steigung auf, dann fällt die Dichte entlang dieser Kante unterhalb des Ursprungsniveaus und die Kante ist somit unzulässig. Daher können wir mit dieser Heuristik alle Nachbarn, in deren Richtung die Dichteableitung negative Steigung aufweist, aus weiteren Betrachtungen ausschließen.

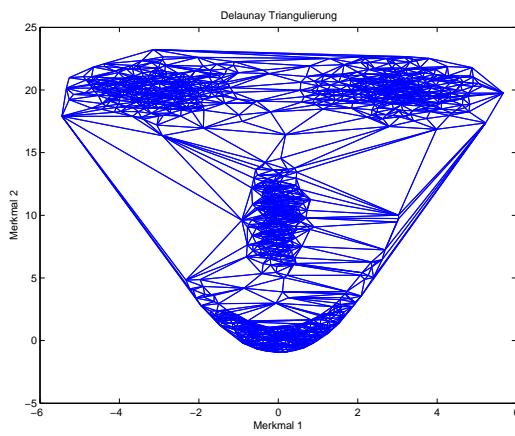


Abbildung 4.3.: Schritt 2: Das Berechnen der Nachbarschaft, hier am Beispiel der Delaunay-Triangulierung.

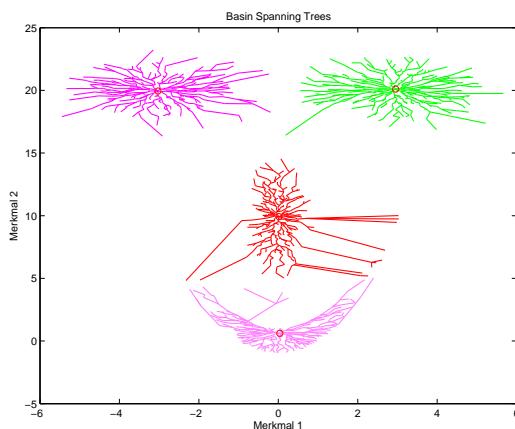


Abbildung 4.4.: Schritt 3: Konstruktion der Basin Spanning Trees. Die roten Punkte stellen die Wurzeln der Bäumen dar.

5. Wahl der Parameter

Im vorherigen Kapitel haben wir den BST-Algorithmus vorgestellt. Dabei gingen wir immer von den vorgegebenen Parametern einer Bandweite h und einer Kernfunktion K aus. In diesem Kapitel wollen wir den Schwerpunkt auf die Wahl dieser Parameter legen.

Wie aus dem Kapitel 3.5 bekannt ist, ist die Wahl der Kernfunktion für die Dichteschätzung von geringerer Bedeutung. Für das weitere Vorgehen wollen wir für die frei wählbare Kernfunktion immer den Epanechnikov Kern betrachten. Das hat im Wesentlichen folgende Gründe: Der Epanechnikov Kern ist asymptotisch optimal und die Trägermenge ist beschränkt (s. Abschnitt 3.5).

An dieser Stelle wollen wir hingegen unseren Fokus auf die Wahl der Bandweite lenken (Abb. 5.1, 5.2 und 5.3). In diesem Kapitel werden wir mehrere Verfahren zur automatischen Bestimmung der Bandweite vorstellen. Diese können wir grob in drei Bereiche einordnen: Stochastische Modelle, Heuristiken basierend auf der Topologie der Nachbarschaft und Bewertungsfunktionen.

5.1. Stochastische Modelle

Die Methoden, die in diesem Abschnitt vorgestellt werden, sind seit Jahren bekannt und erprobt (vgl. Silverman (1986)). Sie alle basieren auf einem stochastischen Modell.

5.1.1. Kleinst-Quadrat-Kreuzvalidierung

Die *Kleinst-Quadrat-Kreuzvalidierung*¹ stellt eine vollständige automatische Methode der Bandweitenwahl dar. Der Ansatz basiert auf einer einfachen Idee und war erstmals von Rudemo (1982) vorgestellt.

Sei \hat{f} die Schätzung der Dichte f . Der integrierte quadratische Fehler ergibt sich aus:

$$\int (\hat{f} - f)^2 = \int \hat{f}^2 - 2 \int \hat{f}f + \int f^2. \quad (5.1)$$

Der letzte Term der Gleichung (5.1) ist nicht von \hat{f} abhängig. Die Minimierung des Terms

$$R(\hat{f}) := \int \hat{f}^2 - 2 \int \hat{f}f \quad (5.2)$$

¹engl.: least squares crossvalidation

5.1. Stochastische Modelle

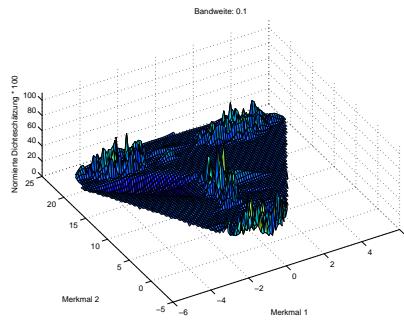


Abbildung 5.1.: Bei einer kleinen Wahl der Bandweite besitzt die Dichteschätzung viele lokale Maxima. Hier am Beispiel des Smile Datensatzes deutlich erkennbar.

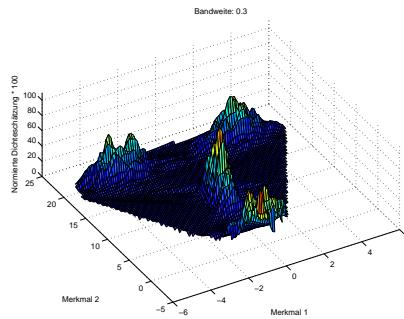


Abbildung 5.2.: Vergrößerung der Bandweite minimiert die Anzahl der Modi.

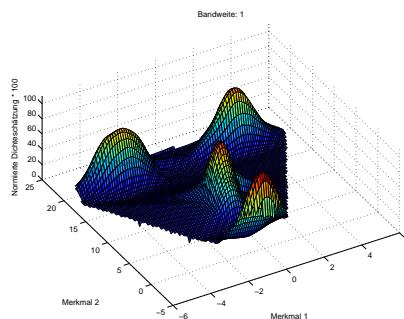


Abbildung 5.3.: Bei einer großen Bandweite nimmt die Kernschätzung lokal die Form der Kernfunktion an. Hier die Form des Epanechnikov Kerns.

5. Wahl der Parameter

führt zur Minimierung des integrierten quadratischen Fehlers und eignet sich somit ideal zur Bestimmung der optimalen Bandweite.

Die Aufgabe besteht in der Schätzung des Terms $R(\hat{f})$ aus vorhandenen Daten und dessen Minimierung für die gesuchte Bandweite h .

Der Term $\int \hat{f}^2$ kann aus der Dichteschätzung \hat{f} bestimmt werden. Definieren wir \hat{f}_{-i} als Dichteschätzung, die auf allen Punkten außer X_i basiert. Das ist die sogenannte *Leave-one-out* Methode, die ursprünglich auf [Lachenbruch \(1968\)](#) zurück geht. Formal ausgedrückt (vgl. (3.5)):

$$\hat{f}_{-i}(x) = \frac{1}{(n-1)h} \sum_{\substack{j=1 \\ j \neq i}}^n K\left(\frac{x-X_j}{h}\right). \quad (5.3)$$

Nun definieren wir die Bewertungsfunktion M_0 für die Bandweite h :

$$M_0(h) = \int \hat{f}^2 - \frac{2}{n} \sum_{i=1}^n \hat{f}_{-i}(X_i). \quad (5.4)$$

Diese basiert ausschließlich auf den Beobachtungen (Daten). Die Idee ist, die Bewertungsfunktion M_0 für die Bandweite h zu minimieren.

Um zu verstehen, warum die Minimierung der Funktion M_0 die gewünschte Bandweite liefert, betrachten wir den Erwartungswert der Funktion M_0 . Der Erwartungswert des Summationsterms aus (5.4) ist:

$$E\left[\frac{1}{n} \sum_{i=1}^n \hat{f}_{-i}(X_i)\right] = E\left[\hat{f}_{-i}(X_n)\right] \quad (5.5)$$

$$= E\left[\int \hat{f}_{-i}(x)f(x)dx\right] \quad (5.6)$$

$$= E\left[\int \hat{f}(x)f(x)dx\right], \quad (5.7)$$

da $E[\hat{f}]$ nur von der Kernfunktion und der Bandweite — nicht von der Anzahl der Punkte — abhängt. Aus dem Einsetzen von (5.7) zurück in die Definition von M_0 ergibt sich:

$$E[M_0(h)] = E[R(f)]. \quad (5.8)$$

Da $\int f^2$ für alle h gleich ist, korrespondiert der Erwartungswert der Funktion M_0 mit dem Erwartungswert des integrierten quadratischen Fehlers.

Nehmen wir nun an, die Bandweite, die M_0 minimiert, liegt in der Nähe der Bandweite, die den Erwartungswert vom M_0 minimiert. Folglich haben wir mit (5.8) berechtigte Hoffnung eine gute Dichteschätzung zu erhalten.

5.1.2. Likelihood-Kreuzvalidierung

Wir benutzen die Idee des Likelihood zur Beurteilung der Angemessenheit eines statistischen Models (vgl. Silverman (1986)), dabei verwenden wir Begriffe und Notation, die wir im Kapitel 3.2 eingeführt haben. Als ausführliche Einführung in die Thematik ist Krengel (2000) zu empfehlen.

Angenommen, wir haben zusätzlich zu den Beobachtungen \mathcal{X} die Beobachtungen \mathcal{Y} der Dichte f . Somit ist die log-Likelihood-Funktion der Dichte f , die den Beobachtungen \mathcal{Y} zu Grunde liegt, $\log f(\mathcal{Y})$. Betrachten wir \hat{f} als parametrisierte Familie der Dichtefunktionen, die von der Bandweite h mit festen X_1, \dots, X_n abhängt, dann können wir $\log \hat{f}(\mathcal{Y})$ als Funktion von h , also log-Likelihood-Funktion der Bandweite h betrachten.

Nun, da uns eine solche zusätzliche Menge von Beobachtungen nicht zur Verfügung steht, können wir eines der ursprünglichen Beobachtungen X_i als die Menge \mathcal{Y} benutzen. Da keine Besonderheiten bei der Wahl der ausgelassenen Beobachtung existieren, ergibt sich somit log-Likelihood als der Mittelwert für jede ausgelassene Beobachtung und stellt folgende Zielfunktion auf (vgl. (5.3)):

$$CV(h) = \frac{1}{n} \sum_{i=1}^n \log \hat{f}_{-i}(X_i).$$

Die gesuchte Bandweite h maximiert die Zielfunktion $CV(h)$ der gegebenen Daten.

Der Ansatz wurde zum ersten Mal von Duin (1976) vorgestellt, wobei die Maximierung der Zielfunktion $CV(h)$ die Minimierung der Kullback-Leibler Informati onsdistanz zu Dichtefunktion f darstellt. Diese ist wie folgt definiert:

$$I(f, \hat{f}) = \int f(x) \log \frac{f(x)}{\hat{f}(x)} dx.$$

Aus heuristischer Sicht erhalten wir, wobei die Schätzung, die auf den $(n - 1)$ Beobachtungen basiert, mit \hat{f}_{n-1} notiert wird:

$$E [CV(h)] = E \left[\log \hat{f}_{-n}(X_n) \right] \quad (5.9)$$

$$= E \left[\int f(x) \log \hat{f}_{n-1}(x) dx \right] \quad (5.10)$$

$$\approx E \left[\int f(x) \log \hat{f}(x) dx \right] \quad (5.11)$$

$$= -E \left[I(f, \hat{f}) \right] + \int f \log f. \quad (5.12)$$

Damit stellt $-CV(h)$, bis auf eine Konstante, einen erwartungstreuen Schätzer für den erwarteten Kullback-Leibler Fehler dar und gibt uns eine intuitive Interpretation der Maximierung von $CV(h)$ an.

5. Wahl der Parameter

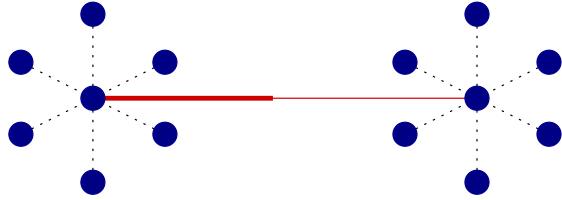


Abbildung 5.4.: Motivation der heuristischen Regel MAXDEG

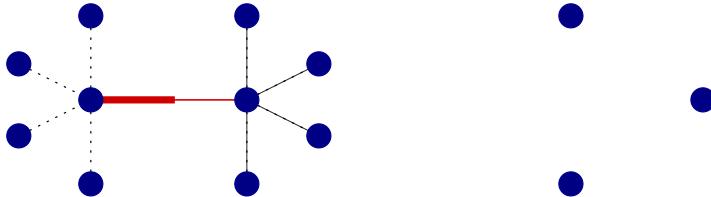


Abbildung 5.5.: Gegenbeispiel der MAXDEG Regel

5.2. Heuristische Wahl

In diesem Abschnitt werden wir die von uns entwickelten Heuristiken zur Bestimmung der optimalen Bandweite kennenlernen. Sie alle nutzen die Topologie des Nachbarschaftsgraphen, der in dem BST-Algorithmus auf Seite 33 für die Bestimmung der Cluster mittels Basin Spanning Trees erzeugt wird.

5.2.1. MaxDeg

Motivation

Wie im BST-Algorithmus (Alg. 4.1) beschrieben, bilden Punkte, die lokale Dichtemaxima besitzen, die Zentren der Cluster. Gehen wir von der Annahme aus, dass diese Punkte dementsprechend viele Nachbarn haben und bilden wir die Umkehrung der Aussage, dann können wir eine heuristische Regel für die automatische Wahl der Bandweite formulieren. Die Aussage „viele Nachbarn“ im graphentheoretischen Sinne

Algorithmus 5.1 MAXDEG Heuristik

- 1: **Input:** \mathcal{S} samples, \mathcal{N} neighborhood of samples
 - 2: **Output:** h heuristic bandwidth
 - 3: $candidates \leftarrow \text{choose-nodes-with-high-degree}(\mathcal{S}, \mathcal{N})$
 - 4: $(i, j) \leftarrow \text{choose-pair-with-max-distance}(candidates)$
 - 5: $h \leftarrow \Delta(i, j)/2$
-

bedeutet, der Grad des Knoten im Nachbarschaftsnetz ist hoch. Um eine eindeutige Trennung der Punkte mit lokalen Maxima zu erreichen, suchen wir nach der größten Distanz zwischen diesen Punkten. Die Halbstrecke dieser Distanz ist die gesuchte

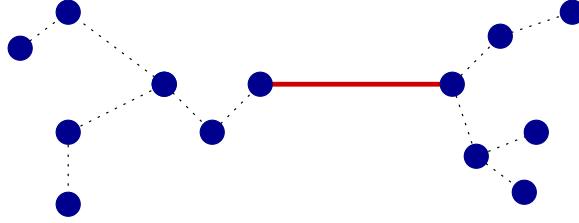


Abbildung 5.6.: Annahmen, die zur MSTMAX Heuristik führen.

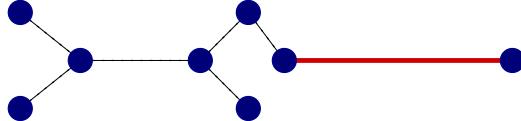


Abbildung 5.7.: Falsche Resultate der MSTMAX Heuristik

Bandweite. In Abbildung 5.4 wird die Regel erläutert. Diese Heuristik haben wir im Alg. 5.1 beschrieben.

Dass diese Regel nur eine Heuristik darstellt, können wir uns an einem Beispiel verdeutlichen (Abb. 5.5). Bei zwei gegebenen Clustern unterschiedlicher Dichte ist es sehr wahrscheinlich, dass die beiden Knoten mit den meisten Nachbarn unmittelbar nebeneinander liegen. Folglich führt das zur Wahl einer zu kurzen Bandweite.

Komplexität

Die Laufzeit der Heuristik können wir aus dem Algorithmus 5.1 ablesen. Das Bestimmen der Knoten mit dem höchsten Grad erfolgt linear in $\mathcal{O}(|V|)$. Das Bestimmen der längsten Distanz zwischen diesen Knoten kann höchstens in $\mathcal{O}(|V|^2)$ Schritten für den degenerierten Fall, in dem alle Knoten auf einem Pfad liegen, erfolgen. Nach dem \mathcal{O} -Kalkül ergibt sich also eine Gesamtkomplexität von $\mathcal{O}(|V|^2)$.

5.2.2. MstMax

Motivation

Im Gegensatz zur MAXDEG Heuristik stellen wir uns nun die Frage, wie groß die Bandweite höchstens gewählt werden darf, damit die Datenpunkte eines Clusters nicht in die Dichteschätzung eines benachbarten Clusters einfließen. Nehmen wir an, die Cluster sind wohldefiniert und eindeutig voneinander getrennt. So können wir

Algorithmus 5.2 MSTMAX Heuristik

- 1: **Input:** \mathcal{S} samples, \mathcal{N} neighborhood of samples
 - 2: **Output:** h heuristic bandwidth
 - 3: $\mathcal{G} = (V, E') \leftarrow \text{mst}(\mathcal{S}, \mathcal{N})$
 - 4: $h \leftarrow \max\{w(e) \mid \forall e \in E'\}$
-

5. Wahl der Parameter

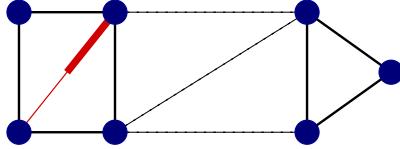


Abbildung 5.8.: Motivation der MAXRADIUS Heuristik.

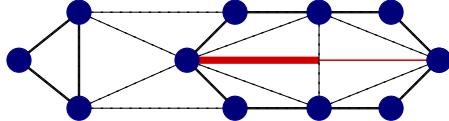


Abbildung 5.9.: Gegenbeispiel zur MAXRADIUS Heuristik.

folgern, dass die Kanten innerhalb eines Clusters stets kürzer sind als die Kanten zwischen den Clustern. Damit sind Cluster durch eine lange Kante in der Topologie der Nachbarschaft getrennt. Die längste Kante eines minimalen Spannbaumes liefert uns einen heuristischen Wert für die Länge dieser Kante (Abb. 5.6). Diese Heuristik können wir nun wie folgt formulieren (Alg. 5.2):

Der heuristische Ansatz wird deutlich, wenn wir uns den Algorithmus genauer ansehen. Die Bandweite wird durch die längste Kante des minimalen Spannbaumes festgelegt. Diese stellt jedoch eine kritische Größe in Bezug auf Ausreißer dar. Die Abbildung 5.7 verdeutlicht die Argumentation. Kommen in den Daten starke Ausreißer vor, so ist das Ergebnis der MSTMAX-Heuristik eine viel zu große Bandweite.

Komplexität

Die Komplexität der Heuristik wird durch die konkrete Implementierung für die Berechnung des minimalen Spannbaums bestimmt. Bei stark besetzten Graphen hat der Prim Algorithmus asymptotisch besseres Verhalten. Das Verfahren von Kruskal eignet sich dagegen bei schwach besetzten Graphen. Beim Greedy Algorithmus von Prim würde die Heuristik ein quadratisches Laufzeitverhalten in Abhängigkeit von der Eingabegröße aufweisen. Da die Suche nach der längsten Kanten in $\mathcal{O}(|E|)$ erfolgt, ergibt sich für einen vollständigen Graphen eine Gesamtlaufzeit von $\mathcal{O}(|V|^2)$. Im Spezialfall der planaren Delaunay-Triangulierung, wie sie im BST-Algorithmus verwendet wird, beträgt die Gesamtkomplexität nach Kruskal Algorithmus $\mathcal{O}(|V| \log |V|)$, da die Anzahl der Kanten in der Delaunay Nachbarschaft proportional zur Anzahl der Knoten wächst.

5.2.3. MaxRadius

Motivation

Der Grundgedanke hinter dieser Heuristik ist ähnlich dem der zuletzt vorgestellten Heuristiken. Wir versuchen dabei, die Clustergröße heuristisch zu bestimmen und verwenden diese Größe als Bandweite für das dichtebasierete Clustering.

Im ersten Schritt zerlegen wir den Nachbarschaftsgraphen in einzelne Zusammenhangskomponenten. Anschließend werden diese aufgezählt und die jeweiligen Radien bestimmt. Dabei machen wir eine pessimistische Annahme mit der Wahl des größten Radius, um den Zerfall der größten Komponente zu verhindern. Die Motivation und das Gegenbeispiel für diese Heuristik kann den Abbildungen 5.8 und 5.9 entnommen werden.

Algorithmus 5.3 MAXRADIUS Heuristik

- 1: **Input:** \mathcal{S} samples, \mathcal{N} neighborhood of samples
 - 2: **Output:** h heuristic bandwidth
 - 3: $\bar{w} \leftarrow \frac{1}{|E|} \sum_{e \in E} w(e)$
 - 4: $\mathcal{G} = (V, E') \leftarrow \text{subgraph}(\mathcal{N}, \bar{w})$ with $E' = \{e \in E \mid w(e) \leq \bar{w}\}$
 - 5: $\mathcal{C} \leftarrow \text{connected-components}(G)$
 - 6: $h \leftarrow \text{max-radius}(\mathcal{C})$
-

Komplexität

Um eine Komplexitätsanalyse der Heuristik durchführen zu können, müssen die einzelnen Schritte des Algorithmus festgelegt werden. Wie schon im Kapitel 2.3.3 vorgestellt, können wir den Graphen in Teilgraphen zerlegen, in dem wir die inkonsistenten Kanten aus dem Graphen entfernen. Wie aus dem Algorithmus 5.3 hervorgeht, erreichen wir dies durch das Entfernen aller Kanten, deren Kantenlänge über der durchschnittlichen Kantenlänge liegt. Dies geschieht in $\mathcal{O}(|E|)$. Das Aufzählen der Zusammenhangskomponenten wird mittels Tiefensuche erreicht, d.h. in $\mathcal{O}(|E| + |V|)$. Die Bestimmung der Radien einzelner Zusammenhangskomponenten ist der größte Aufwand dieser Methode. Hierzu wird zunächst eine konvexe Hülle der Zusammenhangskomponenten bestimmt und anschließend die größte Distanz zwischen allen äußeren Punkten. Somit beläuft sich der Gesamtaufwand auf $\mathcal{O}(\max\{|C_i|^{d/2}\})$.

5.3. Bewertung der Zerlegung

An dieser Stelle wollen eine weitere Verfahrenskategorie für die automatische Wahl der Bandweite vorstellen. Diese ist gekennzeichnet durch die Beurteilung der Bandweite nach ihrer Zerlegung der Daten. Im Folgenden wird eine Reihe von Zerlegungen generiert und anschließend beurteilt. Wir entscheiden uns für die Bandweite, die folglich die beste Zerlegung der Datenpunkte erzeugte. Ferner benötigen wir eine Bewertungsfunktion für die Zerlegungen. Wir wollen einige der untersuchten und entwickelten Methoden hier vorstellen.

5.3.1. Stabilität

Diese Standardtechnik, die wir im Folgenden präsentieren, wird von Fukunaga (1990) und Bock (1974) beschrieben und ist immer noch ein Bestandteil neuer Verfahren, wie Hinneburg und Keim (1998) zeigen.

5. Wahl der Parameter

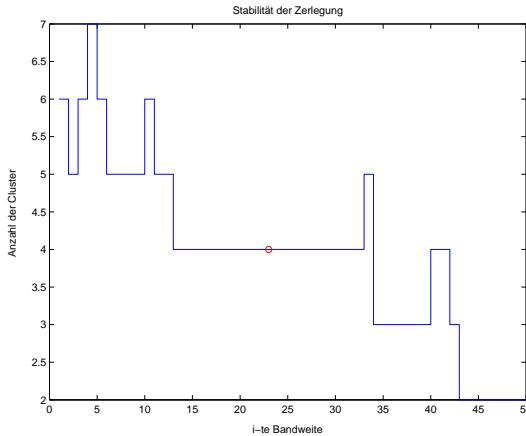


Abbildung 5.10.: Stabilität als Auswahlkriterium: Der Stabilitätsverlauf der Zerlegungen des Smile Datensatzes für 50 logarithmisch gewählte Bandweiten aus dem Intervall $[0.1, 1.0]$. In diesem Fall ist die Bandweite Nummer 23, die zur Zerlegung von 4 Klassen führt, die gesuchte Bandweite.

Wir betrachten unterschiedliche Bandweiten h_i und bestimmen das größte Intervall $[h_{min}, h_{max}]$, in dem die Anzahl der Cluster konstant bleibt. Die Bandweite, welche die Mitte des Intervalls darstellt, ist die gesuchte Bandweite. Die Abbildung 5.10 verdeutlicht diesen Ansatz an einem Beispiel.

5.3.2. Inter- vs. Intra-Kriterien

Eine weitere weitverbreitete Technik ist die Maximierung einer Zielfunktion, die typischerweise die Inter- gegen Intra-Kriterien vergleicht (s. [Kauffmann und Rousseeuw \(1990\)](#)), beziehungsweise die Isolation und die Kompaktheit eines Clusters gegeneinander abwägt ([Pauwels und Frederix \(1999\)](#)).

Den erfolgreichsten Ansatz wollen wir im Weiteren vorstellen. Dieser geht auf [Kauffmann und Rousseeuw \(1990\)](#) zurück und ist unter dem Namen *Silhouetten Koeffizient* geläufig.

Wir definieren den Silhouetten-Koeffizienten eines Punktes wie folgt:

Definition 5.1. $\text{sc}: \mathcal{S} \rightarrow [-1, 1]$ mit

$$\text{sc}(p_i) \mapsto \frac{\min\{ab(p_i, k) - av(p_i)\}}{\max\{av(p_i), \min\{ab(p_i, k)\}\}}$$

nennen wir *Silhouetten-Funktion* und den Wert $\text{sc}(p_i)$ den *Silhouetten-Koeffizienten*. Wobei $ab(p_i, k)$ den mittleren Abstand von p_i zu den Punkten aus dem Cluster k und $av(p_i)$ den mittleren Abstand zu den Punkten innerhalb des eigenen Clusters von p_i darstellen.

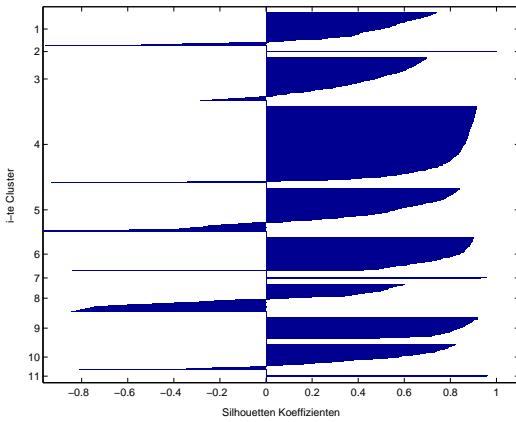


Abbildung 5.11.: Silhouetten-Koeffizienten für die Zerlegung des Smile Datensatzes mit der Bandweitengröße 2.0. Die vielen negativen Silhouetten-Koeffizienten sind ein Indiz für das fehlende Optimum der Zerlegung.

Dabei nehmen Punkte, die „gut“ klassifiziert sind, positive Werte an. Dagegen nehmen Punkte, die durch die Klassifizierung in andere Klassen eine Verbesserung der Gesamtklassifikation liefern würden, negative Werte an. Das Bild 5.11 visualisiert die Koeffizienten an einem Beispiel. Um die einzelnen Zerlegungen miteinander vergleichen zu können, bewerten wir diese anhand des Mittelwerts der Clusterschwerpunkte der Silhouetten-Koeffizienten.

5.3.3. Isolation und Kompaktheit

Die Bewertungsfunktionen (auch *Indizes* genannt) die Intra- mit Inter-Kriterien vergleichen, sei es der Silhouetten-Koeffizient, Hubert-Koeffizient oder der BD-Index ([Jain und Dubes \(1988\)](#)), ziehen kugelähnliche, wohldefinierte Cluster vor. Dagegen bereiten ihnen beliebig geformte Cluster Schwierigkeiten.

Hier möchten wir einen Ansatz vorstellen, der in erster Linie durch die Arbeit von [Pauwels und Frederix \(1999\)](#) motiviert war. Sie stellten einen Index zur Bewertung der einzelnen Zerlegung vor, wobei sie die Informationen der Dichteschätzung ausnutzten.

Im Folgenden wollen wir eine leicht abgewandelte Form dieses Indizes nachvollziehen. Wir benutzen nur die geometrische Information der Zerlegung, d.h. in erster Linie die Distanz zwischen den Punkten und den Grad eines Punktes, um die Isolation und die Kompaktheit eines Clusters messen zu können. Wir führen nun formal zwei Maße ein:

Definition 5.2. $\mathfrak{N} := \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} v(x)$ heißt *Nächster-Nachbar-Norm* (NN-Norm), wobei $v(x)$ den Anteil der direkten Nachbarn für den Punkt x , welche demselben Cluster angehören, darstellt.

5. Wahl der Parameter

Offensichtlich sind viele Knoten innerhalb eines Clusters nur von Knoten des gleichen Clusters umgeben. Somit erwarten wir für einen wohldefinierten Cluster $v(x) \approx 1$.

Der Nachteil der Norm wirkt sich beim Zusammenführen zweier gut isolierter Cluster zu einem gemeinsamen Cluster aus. Denn nach NN-Norm führt das zu keiner Abwertung der Zerlegung. Daher benötigen wir ein weiteres Maß das dem „entgegen“ wirkt.

Definition 5.3. $\mathfrak{C} := \frac{1}{k} \sum_{i=1}^k \frac{|C_i|}{mc(C_i) + \varepsilon}$ heißt Kompakt-Norm (C-Norm), wobei k die Anzahl der Cluster und mc die Kosten des minimalen Spannbaumes innerhalb eines Clusters darstellen.

Die Intuition, die hinter diesem Maß steht, ist die Bestrafung der Zusammenführung mehrerer Punkte, die weit auseinander liegen. Mit anderen Worten: die Cluster mit unveränderter Knotenzahl, aber zu „kleinerem“ Preis sollen bevorzugt werden.

Ziel ist es nun, eine Bewertungsfunktion aufzustellen, die beide Normen berücksichtigt. Um beide Normen direkt miteinander vergleichen zu können, normalisieren wir diese. Dies geschieht mittels der *MAD*-Normierung (5.13), da sie gegenüber den Ausreißern weniger anfällig ist:

$$Z(a_i) := \frac{a_i - \text{median}(a)}{\text{MAD}(a)}. \quad (5.13)$$

Dabei repräsentiert $a = \{a_1, \dots, a_n\}$ die Stichproben und *MAD* die *absolute Median-Abweichung*:

$$\text{MAD}(a) = \text{median}\{|a_i - \text{median}(a)|, i = 1, \dots, n\}.$$

Dann sieht die robuste Bewertungsfunktion für ein Clustering i wie folgt aus:

$$Z^i = Z(\mathfrak{N}^i) + Z(\mathfrak{C}^i).$$

Gesucht ist nun eine Bandweite h_i für deren Zerlegung die Zielfunktion Z maximal ist.

5.3.4. Isolation und Ausprägung

Zum Abschluss dieses Kapitels wollen wir eine Bewertungsfunktion vorstellen, die auf der Dichteschätzung der Daten basiert. Hierzu definieren wir einen Index, der die Ausprägung der jeweiligen Cluster berücksichtigt. Die Bilder 5.12 und 5.13 verdeutlichen den Begriff der Ausprägung.

Definition 5.4. Das Verhältnis des Kuppenvolumens zum Gesamtvolumen der Dichteschätzung eines Clusters wollen wir im weiteren Verlauf als einen Wert der Ausprägung interpretieren.

Offensichtlich können wir die Cluster mit der schwachen Ausprägung als instabil betrachten, da solch eine Zerlegung sehr empfindlich gegenüber den Schwankungen der Bandweite ist (s. Abb. 5.13). Wir präzisieren nun den Begriff der Kuppe:

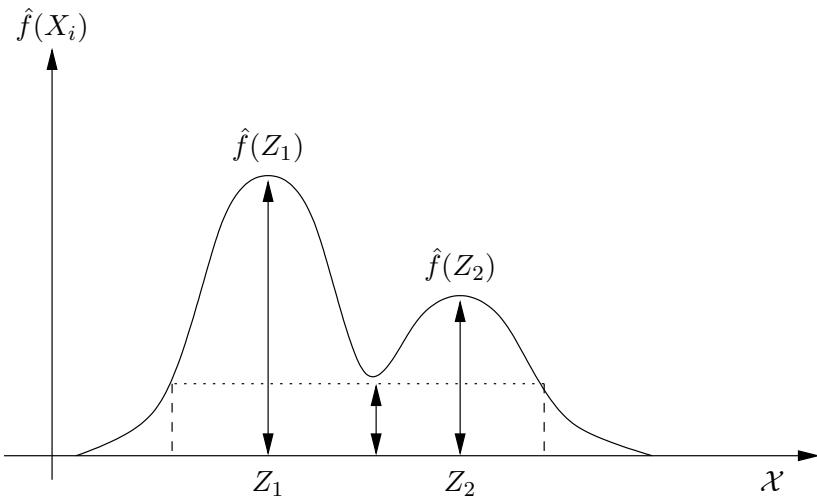


Abbildung 5.12.: Dichteschätzung für zwei gut ausgeprägte Cluster.

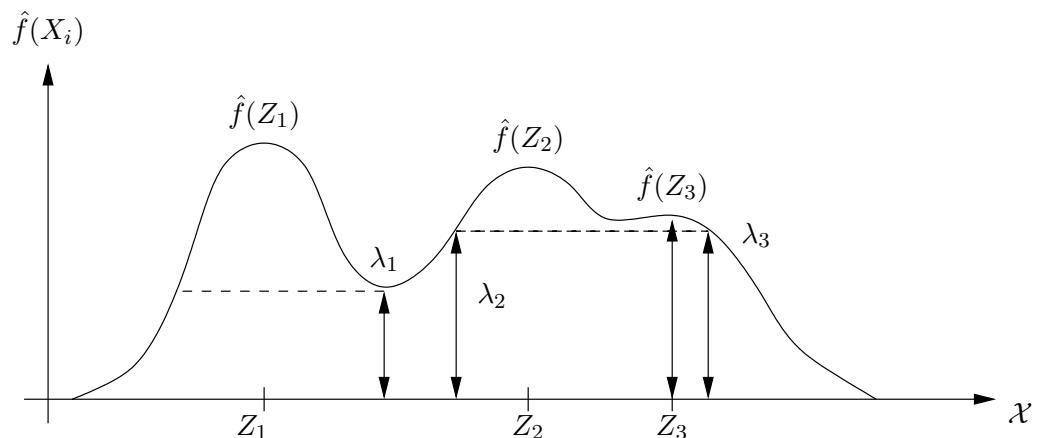


Abbildung 5.13.: Schlecht ausgeprägte Cluster deuten auf ein großes gemeinsames Cluster hin. Hier die Cluster um Modi Z_2 und Z_3 .

5. Wahl der Parameter

Definition 5.5. Sei \hat{f} die Schätzung der Dichte f und C_i ein Cluster der Zerlegung \mathcal{C} . Unter der Kuppe des Clusters C_i für eine Dichteschätzung λ verstehen wir die Menge $L(C_i, \lambda) = \{X_k \mid X_k \in C_i, \hat{f}(X_k) \geq \lambda\}$.

Formal lässt sich der Ausprägungswert eines Clusters für eine Dichteschätzung λ nun wie folgt schreiben:

$$\mathfrak{p}(C_i, \lambda) = \frac{|L(C_i, \lambda)|}{|C_i|}. \quad (5.14)$$

Die P-Norm

$$\mathfrak{P}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \mathfrak{p}(C_i, \lambda_i) \quad (5.15)$$

beschreibt die mittlere Ausprägung eines Cluster in der jeweiligen Zerlegung.

Algorithmus 5.4 Verfahren zur Bestimmung der Passhöhen

```

1: Input:  $\mathcal{C}$  clustering,  $\hat{f}$  density estimation
2: Output:  $\Lambda = \{\lambda_1, \dots, \lambda_{|\mathcal{C}|}\}$  level set density
3: for all  $C_i \in \mathcal{C}$  do
4:    $\lambda_i \leftarrow 0$ 
5:    $q \leftarrow \text{Queue}(\text{bst-root}(C_i))$ 
6:   while  $q \neq \emptyset$  do
7:      $v \leftarrow \text{dequeue}(q)$ 
8:     if  $\exists w \in \mathcal{N}(v), w \notin C_i$  and  $\lambda_i < \hat{f}(w)$  then
9:        $\lambda_i \leftarrow \hat{f}(w)$ 
10:    end if
11:     $\text{enqueue}(q, \text{bst-succ}(v))$ 
12:   end while
13: end for

```

Dabei ist λ_i aus (5.15) die Dichteschätzung des höchsten Tals (Passübergangs) zwischen dem Cluster C_i und einem anderen Cluster (vgl. Abb. 5.13). Diese kann vermöge Algorithmus 5.4 bestimmt werden. Dabei gehen wir wie folgt vor: Nach der vorgenommenen Dichteschätzung und der Erzeugung der Basin Spanning Trees, durchsuchen wir den Cluster von der Wurzel hin bis zu den Blättern nach einem Knoten. Dieser muss eine Nachbarschaft zu einem Knoten eines anderen Clusters aufweisen, während der Nachbarknoten die größtmögliche Dichteschätzung besitzt. Dieses Knotenpaar bildet mit den jeweiligen Pfaden zu den Wurzeln einen Passübergang zwischen den Clusterzentren (Abb. 5.14). Die Suche nach solch einem Knoten geschieht vermöge der Breitensuche.

Die P-Norm hat den Nachteil, dass eine Zerlegung mit vielen gut ausgeprägten Clustern nicht abgewertet wird. Um dem entgegen zu wirken, kombinieren wir die \mathfrak{P} -Norm mit der \mathfrak{N} -Norm (Def. 5.2) aus dem vorherigen Abschnitt und erhalten nach

5.3. Bewertung der Zerlegung

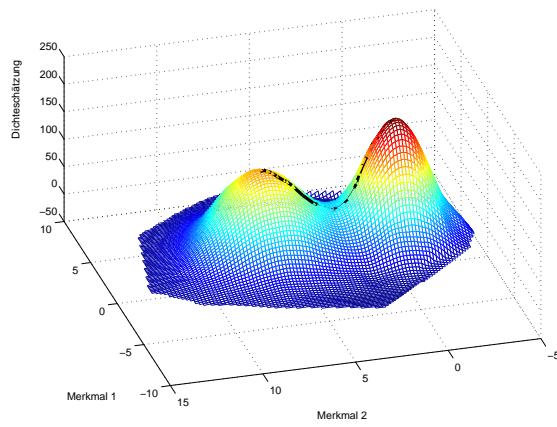


Abbildung 5.14.: Passübergang zwischen zwei Clustern berechnet nach Algorithmus 5.4

der Normierung wiederum eine robuste Bewertungsfunktion, die es zu maximieren gilt:

$$Z^k = Z(\mathfrak{N}^k) + Z(\mathfrak{P}^k). \quad (5.16)$$

6. Clustermerging

In diesem Kapitel werden wir zwei hybride hierarchische Verfahren vorstellen. Es sind agglomerative Heuristiken, welche die Informationen der Dichteschätzung nutzen, um eine Hierarchie der eingebetteten Cluster, wie sie in Abschnitt 2.3.3 vorgestellt, aufzubauen.

6.1. Motivation

Das Clustering mit einer kleinen Bandweite h kann offensichtlich sehr effizient durchgeführt werden. Das Resultat enthält jedoch sehr viele Cluster, dementsprechend ist die Qualität der Zerlegung sehr niedrig, da sie nicht die natürliche Gruppierung der Daten widergibt. Wir fragen uns, ob es möglich ist durch ein nachträgliches Verschmelzen der einzelnen Cluster die Qualität des Clusterings zu verbessern. Es bleibt zu klären nach welchen Kriterien das Zusammenführen (*merging*) zweier Cluster erfolgen, bzw. wann das Postprocessing terminieren soll.

Die sukzessive Verschmelzung der einzelner Cluster kann als eine hierarchische Clustering-Methode aufgefaßt werden. Die konkrete Ausprägung der Hierarchie hängt in erster Linie von der initialen Zerlegung \mathcal{C} ab.

Wir erhoffen uns durch nachfolgende Postprocessing-Prozedur eine Einschränkung des Suchraums der Bandweite. Die anschließende Zusammenführung einiger Cluster simuliert dann eine Dichteschätzung mit einer größeren Bandweite und sollte daher die Qualität der ursprünglichen Zerlegung erheblich steigern.

6.2. Ausprägungskriterium

Das erste Verfahren, das wir vorstellen wollen, greift die Idee der Ausprägung aus dem Abschnitt 5.3.4 auf. Wir führen die Idee der Ausprägung fort und leiten ein Kriterium für die Zusammenführung zweier Cluster her.

Wir gehen von der Beobachtung aus, dass schlecht ausgeprägte Cluster eine Tendenz zur Verschmelzung mit einem ihrer Nachbarcluster aufweisen. Wobei gilt, je schlechter die Ausprägung desto größer diese Tendenz (vgl. Abb. 5.13). Wir wollen diesen Trend der Verschmelzung als ein Kriterium für die Zusammenführung zweier Cluster in einem agglomerativen Algorithmus benutzen. Wir definieren hierzu die Verschmelzungstendenz für ein Cluster C_i mit einem Cluster C_j wie folgt:

$$trend(C_i, C_j) = \begin{cases} \frac{L(C_i, \lambda_{ij})}{|C_i|}, & \text{falls } \hat{f}(Z_i) < \hat{f}(Z_j) \text{ und } \lambda_{ij} \neq 0, \\ 0, & \text{sonst,} \end{cases} \quad (6.1)$$

wobei Z_i und Z_j Zentren der Cluster darstellen. Die Berechnung der Werte λ_{ij} erfolgt mittels einer Variante des Algorithmus 5.4 auf Seite 48. Somit können wir die agglomerative Heuristik wie folgt angeben (Alg. 6.1).

Algorithmus 6.1 Clustermerging nach dem Ausprägungskriterium

```

1: Input:  $\mathcal{C}$  clustering
2: Output:  $\mathcal{C}'$  new clustering
3:  $\mathcal{C}' \leftarrow \mathcal{C}$ 
4:  $trends \leftarrow \text{get-merge-trends}(\mathcal{C})$ 
5: for all  $C_i \in \mathcal{C}$  do
6:    $C_j \leftarrow \text{neighbour-with-largest-merge-trend}(\mathcal{C}, C_i, trends)$ 
7:   if  $C_j \neq \emptyset$  then
8:      $\text{merge}(\mathcal{C}', C_i, C_j)$ 
9:      $trends(C_i, C_j) \leftarrow 0$ 
10:     $trends(C_j, C_i) \leftarrow 0$ 
11:   end if
12: end for
    
```

Im Schritt 5 des Algorithmus erzwingen wir die Zusammenführung eines Clusters mit einem seiner Nachbarn, für den die höchste positive Verschmelzungstendenz vorliegt. Nach der erfolgreichen Verschmelzung werden die Tendenzen beteiligter Cluster (Schritte 8–9) auf Null gesetzt. Nach der Betrachtung aller Cluster terminiert der Algorithmus.

6.3. Sattelpunkt-Kriterium

In diesem Abschnitt möchten wir einen Algorithmus vorstellen, der die Sattelpunkte als Kriterium für die Stabilität der Cluster verwendet. Die Idee des Algorithmus geht auf Comaniciu u. a. (2002) zurück, die einen statistischen Test für die Signifikanz der Clustergültigkeit mittels Sattelpunkte entwickelten. Dabei ersetzen wir den iterativen *Mean-Shift*-Prozess gegen BST-Algorithmus und benutzen Algorithmus 5.4 zur Bestimmung der Sattelpunkte.

Wie schon bemerkt, stellt ein Sattelpunkt X_s mit einer hohen Dichte den schwächsten Punkt eines Clusters C_m mit dem Zentrum Z_m dar. Es genügt eine kleine Verschiebung der Wahrscheinlichkeitsdichte von Z_m in Richtung X_s und Cluster, die durch ein Tal getrennt waren, gehen in einen gemeinsamen Cluster über. Um diesen Prozess zu charakterisieren, nehmen wir an, die geschätzte Dichte am Clusterzentrum und Sattelpunkt verhält sich proportional zur wahren Dichte an diesen Punkten. Die Abbildung 6.1 verdeutlicht den Sachverhalt.

Im Folgenden wollen wir die Zugehörigkeit der Daten zu einem Cluster der Mächtigkeit $|C_m|$ als eine binomialverteilte Zufallsvariable modellieren. Dabei setzt sich die Wahrscheinlichkeit für einen Punkt, dass der in der Nachbarschaft eines Zen-

6. Clustermerging

Algorithmus 6.2 Clustermerging nach dem Sattelpunkt-Kriterium

```

1: Input:  $\mathcal{C}$  clustering
2: Output:  $\mathcal{C}'$  new clustering
3:  $\mathcal{C}' \leftarrow \mathcal{C}$ 
4: for all  $C_i \in \mathcal{C}$  do
5:   for all  $C_j \in \mathcal{C}$  do
6:      $sd \leftarrow \text{get-saddle-density}(C_i, C_j)$  // Algorithmus 5.4
7:      $confidence \leftarrow \text{get-confidence}(sd)$  // (6.7)
8:     if  $confidence \leq 0.9$  then
9:        $\text{merge}(\mathcal{C}', C_i, C_j)$ 
10:    end if
11:   end for
12: end for

```

trums liegt, zusammen aus:

$$\hat{p} = \frac{\hat{f}_K(Z_m)}{\hat{f}_K(Z_m) + \hat{f}_K(X_s)}. \quad (6.2)$$

Wenn ein Punkt in der Nähe eines Sattelpunkts liegt, beträgt die Wahrscheinlichkeit:

$$\hat{q} = 1 - \hat{p} = \frac{\hat{f}_K(X_s)}{\hat{f}_K(Z_m) + \hat{f}_K(X_s)}. \quad (6.3)$$

Die Verteilung von \hat{p} (hier als eine Zufallsvariable betrachtet) kann unter schwachen Bedingungen als normalverteilt vorausgesetzt werden. Wobei der Mittelwert und die Varianz durch:

$$\mu_p = \hat{p} \quad \sigma_p^2 = \frac{\hat{p}(1 - \hat{p})}{|C_i|} \quad (6.4)$$

gegeben sind. Die Nullhypothese mit der wir die Existenz eines Clusters testen, lautet also:

$$H_0: p \geq 0.5 \quad \text{gegen} \quad H_1: p < 0.5. \quad (6.5)$$

Daraus folgt die Statistik:

$$z = \frac{\hat{p} - 0.5}{\sigma_p}. \quad (6.6)$$

Setzen wir (6.2) und (6.3) ein, so erhalten wir:

$$z = \frac{\sqrt{|C_i|}(\hat{f}_K(Z_m) - \hat{f}_K(X_s))}{2\sqrt{\hat{f}_K(Z_m)\hat{f}_K(X_s)}}.$$

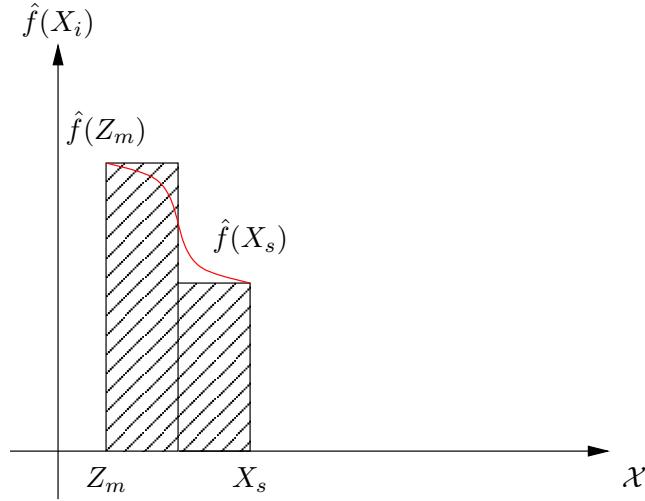


Abbildung 6.1.: Approximation des Wahrscheinlichkeitsmaßes. Die Wahrscheinlichkeit in der Umgebung von Z_m und X_s ist proportional zu $\hat{f}(Z_m)$ und $\hat{f}(X_s)$.

Der p -Wert des Tests ist die Wahrscheinlichkeit, dass die Zufallsgröße z — die mit $\mathcal{N}(0, 1)$ verteilt ist — positiv ist:

$$P(z \geq 0) = \frac{1}{\sqrt{2\pi}} \int_{-z}^{\infty} e^{-t^2/2} dt. \quad (6.7)$$

Den Konfidenzwert von 0.95 erhalten wir für ein $z = 1.65$.

Somit können wir erneut ein hybrides Verfahren angeben. Wir verwenden den Konfidenzwert als Schwellwert für die Existenz der Cluster und den Algorithmus im Alg. 6.2 noch einmal zusammengefaßt.

7. Implementierung

In diesem Kapitel werden wir auf die Implementierungsdetails eingehen. Nach einer kurzen Einführung in MATLAB und die Toolbox für die Dichteschätzung gehen wir auf das Laufzeitverhalten ein und stellen eine Optimierungstechnik vor.

7.1. Matlab

MATLAB ist eine Sprache für technische Berechnungen. Dabei vereinigt MATLAB eine Vielzahl von Komponenten für die Entwicklung, Berechnung und Visualisierung in einer intuitiv zu bedienenden Umgebung. Zu den typischen Aufgaben, die mit MATLAB gelöst werden können, zählen:

- Mathematische Berechnungen
- Algorithmenentwicklung
- Modellierung, Simulation und Prototyping
- Datenanalyse und Visualisierung
- Darstellung von Graphiken aus dem naturwissenschaftlichen Bereich und dem Ingenieurswesen
- Entwicklung von Anwendungen.

Der Name MATLAB steht für *matrix laboratory*. MATLAB sollte ursprünglich den Zugang für LINPACK und EISPACK Projekte erleichtern, die den Stand der Techniken bei den Berechnungen mit Matrizen darstellen.

MATLAB bietet eine Vielzahl von anwendungsspezifischen Lösungen, die sogenannten *Toolboxen*. Eine Toolbox ermöglicht das Erlernen und das Anwenden einer speziellen Technologie. Sie stellt eine umfangreiche Sammlung von MATLAB Funktionen, welche die MATLAB Umgebung um die gewünschte Funktionalität erweitern, zur Verfügung. Zu den Bereichen, für die eine Toolbox vorhanden ist, zählen Signalverarbeitung, Neuronale Netzwerke, Fuzzy Logik, Wavelets, Simulation und einige mehr.

7.2. Toolbox zur Dichteschätzung

Wie schon in Kapitel 1 erwähnt, wollen wir hier eine Toolbox zur Dichteschätzung und zum Clustering von Daten vorstellen. Diese entwickelte der Diplom-Informatiker

Sören Hader in der FV/PLF2 Abteilung der Robert Bosch GmbH in Schwieberdingen. Sie stellt eine umfangreiche Sammlung von MATLAB Funktionen zur Verfügung, vermöge denen das Clustering mehrdimensionaler Daten untersucht und angewandt werden kann.

Den Kern der Toolbox bildet die Funktion `kd_clust`. Diese stellt einen Stufenprozess dar, der durch eine Vielzahl von Parametern konfiguriert werden kann. Wir wollen die einzelnen Stufen hier kurz vorstellen:

Parse der Parameter bildet die erste Stufe, dabei wird die aktuelle Konfiguration des Clusteringprozesses festgelegt.

Preprocessing normiert die Daten in Abhängigkeit von der Konfiguration. Dieser Arbeitsschritt ist optional.

Dimensionsreduktion erfolgt mittels der *Principal-Component-Analysis*. Dabei muss die Anzahl der Dimensionen, die berücksichtigt werden, spezifiziert werden. Diese Prozessstufe ist optional.

Triangulierung: In diesem Schritt wird die Delaunay-Triangulierung der Daten mittels des Qhull-Pakets ([Barber u. a. \(1996\)](#)) berechnet. Anschließend wird die Triangulierung in einen Nachbarschaftsgraphen konvertiert (Graphkonstruktion).

Dichteschätzung erfolgt vermöge eines spezifizierten Kerns. Dabei stehen einige Optimierungstechniken zur Verfügung, welche die lokale Nachbarschaftssuche beschleunigen.

Bst-Algorithmus benutzt nun die Informationen der Dichteschätzung, um ein Clustering zu generieren. Die genaue Darstellung des Algorithmus erfolgte bereits in Kapitel 4.

Postprocessing: Diese Stufe stellt einen einfachen Mechanismus für die Reduzierung der Clusteranzahl bereit. Dabei werden Cluster innerhalb eines spezifizierten Radius zusammengefasst. Dieser Arbeitsschritt ist auch optional.

Visualisierung ermöglicht dem Benutzer die Basin Spanning Trees graphisch darzustellen. Hierbei werden mehrdimensionale Daten mittels der *Principal-Component-Analysis* auf einen 2D bzw. 3D Raum abgebildet.

7.3. Laufzeitmessungen

Wie aus Kapitel 4 hervorgeht, stellen die Stufen Triangulierung und Dichteschätzung die aufwändigsten Schritte dar. Der wachsende Zeitverbrauch der Triangulierung in höheren Dimensionen ist auf die Komplexität der Aufgabe zurückzuführen. Mittels BST-Algorithmus haben wir eine einmalige Auswertung der Kernfunktion pro Datenpunkt erreicht. Wie jedoch aus Tabelle 7.1 (vierte Spalte) exemplarisch hervorgeht, bleibt dieser Schritt der aufwändigste in der gesamten Berechnungskette.

7. Implementierung

Samples	Prozentualer Anteil der Laufzeit (%)					
	Delaunay	Tri.	Graphkonst.	Dichteschätzung	BST	Clustering
1000	12.87	6.74		40.50	24.57	15.32
2000	9.67	4.99		53.67	19.63	12.03
3000	7.59	3.60		63.41	15.93	9.47
4000	6.14	2.86		69.63	13.75	7.62
5000	6.36	2.52		73.91	10.79	6.42
6000	4.57	2.00		78.51	9.39	5.52
7000	3.96	1.76		81.24	8.20	4.84
8000	3.54	1.51		83.29	7.33	4.31
9000	3.18	1.37		85.07	6.55	3.83

Tabelle 7.1.: Laufzeitverhalten der einzelnen Routinen am Beispiel der synthetischen Daten. Dabei wurden zweidimensionale Instanzen mit jeweils unterschiedlicher Anzahl Samples generiert. Anschließend wurde deren Dichte mit Bandweite $h = 3.0$ geschätzt.

7.4. Optimierungstechniken

Wie wir im vorherigen Kapitel sahen, bildet die Kernfunktionsauswertung an den Datenpunkten den Schwerpunkt der Berechnungen. Hier möchten wir eine Technik vorstellen, die eine effiziente und gleichzeitige Auswertung der Kernfunktion für mehrere Bandweiten ermöglicht.

Naiver Ansatz

Der nahe liegende Ansatz besteht darin, die berechnete Distanz zwischen zwei Punkten für die Auswertung mehrerer Bandweiten zu nutzen. Der Vorteil des Ansatzes liegt in der Unabhängigkeit bezüglich der Kernfunktion. Der Nachteil der Methode ist der Platzverbrauch, da pro Bandweite eine Tabelle für die Dichteschätzung gepflegt wird.

Verbesserter Ansatz

Die Idee hinter dieser Technik führt die Idee der gleichzeitigen Auswertung der Kernfunktion für mehrere Bandweiten fort und erweitert diese, in dem wir den Wert der Kernfunktion der Bandweite h_i immer auf den Dichteschätzwert der ersten Bandweite zurückführen.

Nehmen wir an, dass die Bandweiten wie folgt zueinander in Beziehung stehen:

$$h_1 < h_2 < h_3 < \dots < h_n.$$

Dann existiert für jede Bandweite h_i ein Koeffizient $c_i > 1$ (mit $2 \leq i \leq n$), so dass jede Bandweite durch die Bandweite h_1 ausgedrückt werden kann:

$$h_1 < c_2 h_1 < c_3 h_1 < c_4 h_1 < \dots < c_n h_1. \quad (7.1)$$

Da wir die Dichteschätzung der Bandweite h_i auf die Dichteschätzung der Bandweite h_1 zurückführen wollen, müssen wir eine konkrete Ausprägung der Kernfunktion betrachten. Rufen wir die Definition des Epanechnikov Kerns in Erinnerung:

$$K(x) = \begin{cases} 1 - |x|^2, & \text{falls } |x| \leq 1, \\ 0, & \text{sonst.} \end{cases}$$

Betrachten wir nun die Dichteschätzung für den Punkt x mit der Bandweite h_i unter der Annahme, dass die Norm von x kleiner gleich 1 ist. Somit folgt aus der Definition des Epanechnikov Kerns:

$$K_i\left(\frac{x - X_j}{h_i}\right) = 1 - \left|\frac{x - X_j}{h_i}\right|^2 \quad (7.2)$$

$$\iff \left|\frac{x - X_j}{h_i}\right| \leq 1 \quad (7.3)$$

$$\iff \left|\frac{x - X_j}{h_1}\right| \leq c_i < c_{i+1} < \dots < c_n. \quad (7.4)$$

Wir sehen, dass die Bedingung der Norm für den Epanechnikov Kern für die Bandweiten h_{i+1}, \dots, h_n nach (7.1) erfüllt ist, falls diese für die Bandweite h_i gilt. Folglich kann die Schätzung für jede Bandweite h_i auf eine Schätzung der Bandweite h_1 zurückgeführt werden. Im Weiteren benutzen wir den Trick der „nahrhaften“ Null:

$$K_i\left(\frac{x - X_j}{h_i}\right) = 1 - \left|\frac{x - X_j}{h_i}\right|^2 \quad (7.5)$$

$$= 1 - \left|\frac{x - X_j}{h_i}\right|^2 + \left|\frac{x - X_j}{h_1}\right|^2 - \left|\frac{x - X_j}{h_1}\right|^2 \quad (7.6)$$

$$= K_1\left(\frac{x - X_j}{h_1}\right) - \left|\frac{x - X_j}{h_i}\right|^2 + \left|\frac{x - X_j}{h_1}\right|^2 \quad (7.7)$$

$$= K_1\left(\frac{x - X_j}{h_1}\right) + \left|\frac{x - X_j}{h_1}\right|^2 \left(1 - \frac{1}{c_i^2}\right). \quad (7.8)$$

Somit ergibt sich die Kernschätzung für jede Bandweite h_i als Kernschätzung mittels Bandweite h_1 und einem Korrekturterm, der abhängig von dem Koeffizienten c_i ist. Für zwei gleiche Bandweiten ergibt sich nach (7.8) die gleiche Kernschätzung. Analoge Rechnungen lassen sich für die meisten Kernfunktionen mit endlicher Trägermenge aufstellen.

Betrachten wir nun den Fall $\left|\frac{(x-X_i)}{h_1}\right|^2 > 1$, dann gilt nach der Definition des Epanechnikov Kerns $K\left(\frac{x-X_i}{h_1}\right) = 0$. Fahren wir mit der Betrachtung der Bandweiten

$$h_2 < h_3 < \dots < h_n.$$

analog (7.2)ff. fort.

7. Implementierung

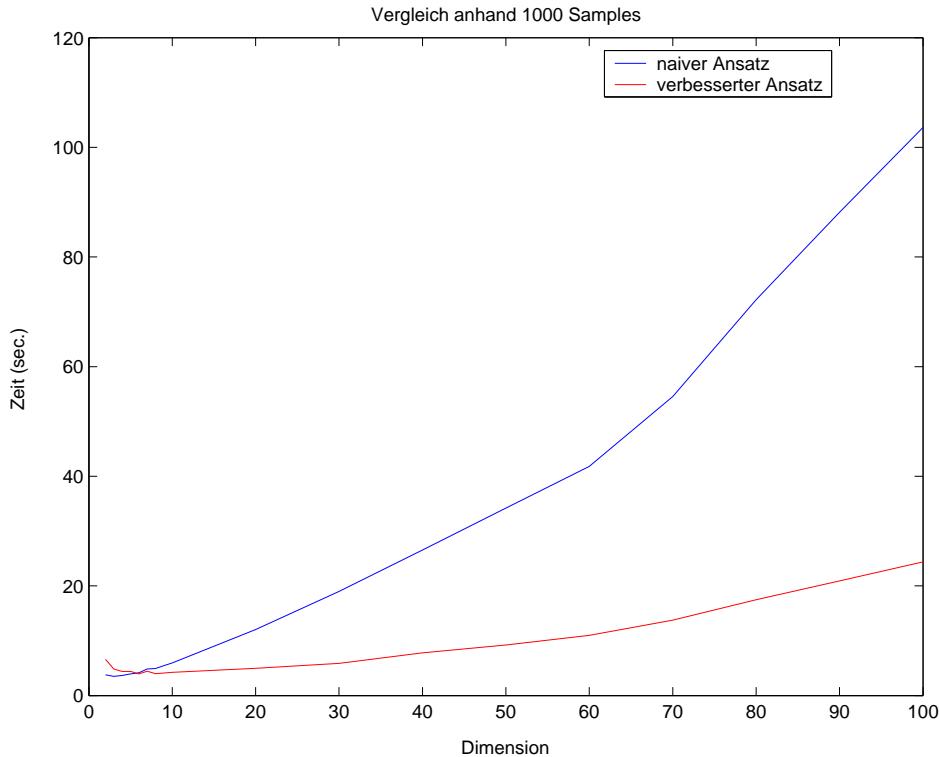


Abbildung 7.1.: Vergleich der Implementierungen für die Auswertung der Kernfunktion am Beispiel des Epanechnikov Kerns und 10 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 10].

Der Nachteil der naiven Methode, sprich der Platzbedarf, bleibt jedoch weiterhin bestehen. Hinzu kommt die Abhängigkeit von der Kernfunktion. Wir erreichen aber, wie im nächsten Abschnitt gezeigt wird, ein besseres Laufzeitverhalten, da wir eine Vektoroperation durch eine skalare Operation ersetzen. Einen weiteren Faktor für die Beschleunigung der Laufzeit stellt die Unabhängigkeit des Ansatzes von der Anzahl der Samples dar.

Laufzeitvergleich

Wir haben beide Methoden miteinander verglichen. Wir ließen pro abgetragene Dimension (s. Abb. 7.1) einen Datensatz generieren und wendeten beide Verfahren mit denselben Bandweiten an. Die einzelnen Dichteschätzungen ließen wir dabei fünf mal wiederholen und bildeten anschließend den Mittelwert, um die Einflüsse des Systems zu minimieren. Wie deutlich zu erkennen ist (s. Abb. 7.1), kann die verbesserte Variante ihren Vorteil deutlich in höheren Dimensionen ausspielen. Die verbesserte Methode hat jedoch bei relevanten Dimensionen leichte Nachteile. Allem Anschein nach ist der Mehraufwand in Dimensionen dieser Ordnung noch zu groß.

8. Experimentelle Resultate

Nun wollen wir erzielte Resultate dieser Arbeit vorstellen. Nach der Beschreibung der Testumgebung und der Charakteristika der Testdaten folgt eine Zusammenfassung und Bewertung der Ergebnisse.

8.1. Testplattform

Die Datensätze wurden auf einem Personal Computer mit einem Intel Pentium III 500 MHz Prozessor und 512 MB RAM gerechnet. Zu Grunde lag das Microsoft Windows 2000 Professional Betriebssystem mit Service Pack 2.

Die Implementierung entstand unter MATLAB der Firma The MathWorks, Inc. Uns stand die Version 6.5.0 Release 13 zur Verfügung.

8.2. Synthetische Daten

Turing-Test

Pauwels und Frederix (1999) stellten in Anlehnung an den Turing-Test vier nichtlineare Datenmengen vor, um die Qualität der generierten Zerlegungen beurteilen zu können und um sich einen schnellen Überblick über die Leistungsfähigkeit der Clustering Algorithmen zu verschaffen. Wir haben einen ähnlichen Datensatz generiert und unsere Verfahren darauf getestet (s. Abb. 8.1 auf Seite 61).

Gauß-Mixturen

Zusätzlich zum Turing-Test generierten wir eine Reihe von Datensätzen, die jeweils aus unterschiedlichen Gaußverteilungen entstanden sind. Dabei ließen wir die Mächtigkeit, Kompaktheit und die Streuung der einzelnen Cluster variieren. Um sicher zu gehen, ob ein sinnvolles Clustering möglich ist, wurden die Daten visuell begutachtet, indem wir die Daten mittels *Principal-Component-Analysis* (kurz PCA) auf den 3D-Raum abbildeten. Die genauen Parameter der so entstandenen Mixturen haben wir noch einmal in der Tabelle 8.1 aufgelistet.

8.3. Reale Datensätze

Mit folgenden realen Datensätzen, die in der Tabelle 8.2 beschrieben sind, testeten wir unsere Verfahren. Diese und weitere Datensätze können über eine der folgenden Internet Adressen bezogen werden:

8. Experimentelle Resultate

Name/Cluster/Samples/Dimension/	μ	σ^2
synth/2/1000/2	2.1	2.9
synth/3/1000/2	2.8	3.8
synth/4/1000/2	3.9	5.4
synth/5/1000/2	2.6	2.8
synth/2/1000/3	1.3	3.2
synth/3/1000/3	3	3
synth/4/1000/3	2.3	3.2
synth/5/1000/3	3.2	3.2
synth/2/1000/4	1.3	3.2
synth/3/1000/4	2.3	4
synth/4/1000/4	2.8	4.2
synth/5/1000/4	4.5	5
synth/2/1000/5	1.5	2.4
synth/3/1000/5	2.5	4
synth/4/1000/5	4.1	3.9
synth/5/1000/5	4	3.6
synth/2/1000/6	1.2	2
synth/3/1000/6	2.6	3.3
synth/4/1000/6	1.2	2.8
synth/5/1000/6	2.9	3.8

Tabelle 8.1.: Charakterisierung synthetischer Daten

Name	Samples	Dimension	Klassen
hayes-roth	132	5	3
iris	150	4	3
wine	178	13	3
glass	214	10	7
liver-disorders	345	6	2
tic-tac-toe	958	9	2
yeast	1484	17	7
abalone	4117	8	8

Tabelle 8.2.: Charakteristika realer Datensätze

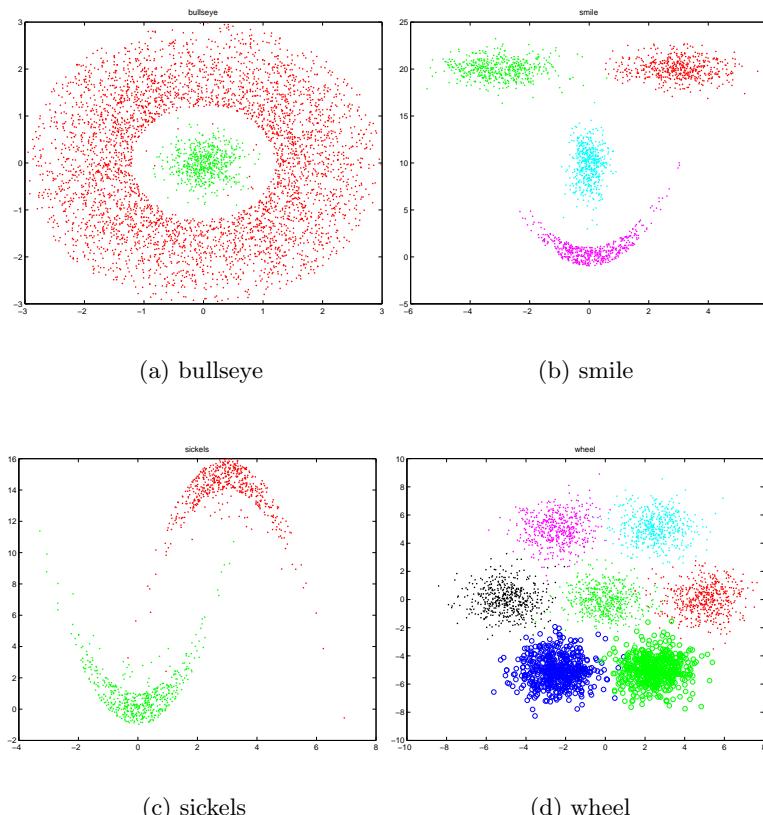


Abbildung 8.1.: Turing-Test Datensatz

8. Experimentelle Resultate

- <ftp://ics.uci.edu/pub/machine-learning-databases/>
- lib.stat.cmu.edu/datasets

Unter diesen Adressen ist auch eine detaillierte Beschreibung über die Herkunft, bisherige Benutzung und Analyse der Daten sowie eine inhaltliche Erläuterung der Messungen erhältlich.

Die Instanzen bildeten wir mittels PCA auf Räume der dritten und vierten Dimension ab. Daraufhin folgte das Clustering.

8.4. Bilder

Die erläuterten Verfahren wandten wir auch an digitalen Bildern an. Farbinformation ohne die Ortsposition im Bild diente als Grundlage. Nach dem Einlesen skalierten wir die Bilder. Darauf folgte eine Transformation aus dem RGB in den HSV Farbraum. (Informationen über die Eigenschaften und Spezifika der Farbräume können aus [Colantoni und Trémeau \(2003\)](#) entnommen werden.) Nach diesen Vorverarbeitungsschritten bildeten wir Cluster. Anschließend färbten wir Pixel aus gleichen Klassen mit dem Wert des zugehörigen Clusterzentrums und konvertierten die Bilder zurück in den RGB Farbraum.

8.5. Spritzer Klassifizierung

In der FV/PLF2 Abteilung der Robert Bosch GmbH existiert eine Datenbank der Spritzerobjekte — wegfliegende Materialpartikel, die heller (heißer) als der Hintergrund sind — die während einer Laserschweißung entstehen. Diese sind mittels 20 Kriterien, die in der Tabelle 8.3 beschrieben sind, charakterisiert.

Die Aufnahme $g(x, y, t)$ beschreibt die Temperaturwerte in der x, y Ebene zum Zeitpunkt t . $f(x, y, t)$ ist das Änderungsbild. Es stellt die Abweichung von dem mittleren, normierten Bild der kompletten Aufnahmesequenz dar.

Die Verfahren testeten wir auf zwei Datensätzen. Der erste beinhaltet 1641 typische Spritzerobjekte. Der zweite Datensatz besteht aus 15736 Objekten und bildet eine Obermenge des ersten. Den größten Anteil dieser Menge bilden Aufnahmen mit einer Aufhellung. Diese sind typisch für Kameraeffekte, die während der Überwachung der Laserschweißprozesse auftreten. Die normierte Darstellung der Merkmalsverteilungen beider Sätze kann den Histogrammen in den Abbildungen 8.2 und 8.3 entnommen werden.

8.6. Ergebnisse

8.6.1. Evaluierung der nicht überwachten Lernmethoden

Um die Ergebnisse der nicht überwachten Lernmethoden beurteilen zu können, wenden wir diese an bereits klassifizierten Daten an. Somit können wir die entstande-

Merkmal	Beschreibung
1	$vol := area \cdot val$
2	$area$ — Flächeninhalt des Spritzers
3	val — Mittelwert von $f(x, y, t)$
4	$\text{Var}(y)$ — Abweichung in y Richtung
5	$\text{Var}(x)$ — Abweichung in x Richtung
6	$\text{Cov}(x, y)$ — Kovarianz von x, y
7	φ — Orientierungswinkel im Bogenmaß
8	ε — Exzentrizität, Formbeschreibung
9	val_2 — Mittelwert von $f(x, y, t)^2$
10	$vol_2 := area * val_2$
11	val_3 — Mittelwert von $f(x, y, t)^3$
12	val_4 — Mittelwert von $f(x, y, t)^4$
13	val_5 — Mittelwert von $f(x, y, t)^5$
14	$highval$ — Mittelwert der 5 größten Werten von $f(x, y, t)$
15	Normierte Differenz von $f(x, y, t_i) - f(x, y, t_{i-1})$
16	Normierte Differenz von $f(x, y, t_i) - f(x, y, t_{i+1})$
17	Normierte Differenz von $f(x, y, t_{i-1} + 2f(x, y, t_i) - f(x, y, t_{i+1})$
18	$higharea$ — Flächeninhalt bei einer doppelt so hohen Binarisierungsschwelle
19	d — Abstand zum Schmelzbad
20	$1/d$ — reziproker Abstand zum Schmelzbad

Tabelle 8.3.: Merkmalbeschreibung der Spritzerobjekte

8. Experimentelle Resultate

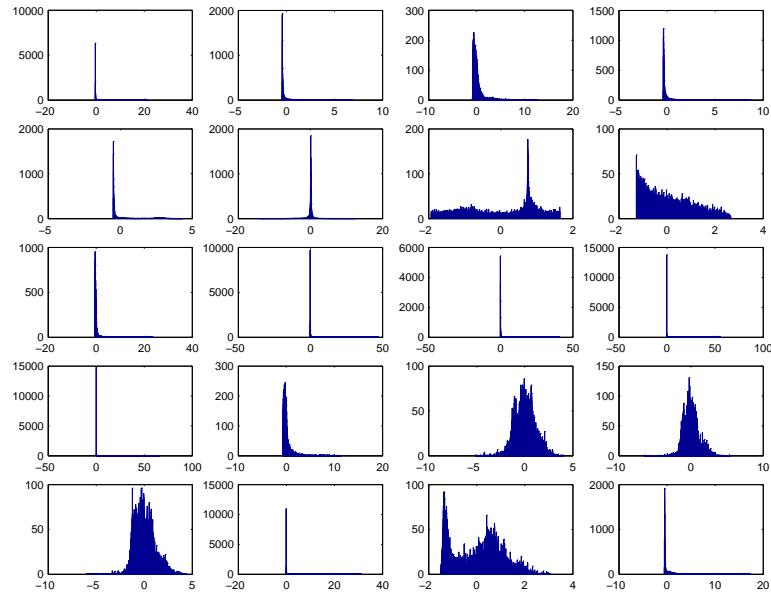


Abbildung 8.2.: Histogramme der Merkmalsverteilungen der Spritzerobjekte und der Aufhellungen (Merkmale v.l.n.r. und v.o.n.u.)

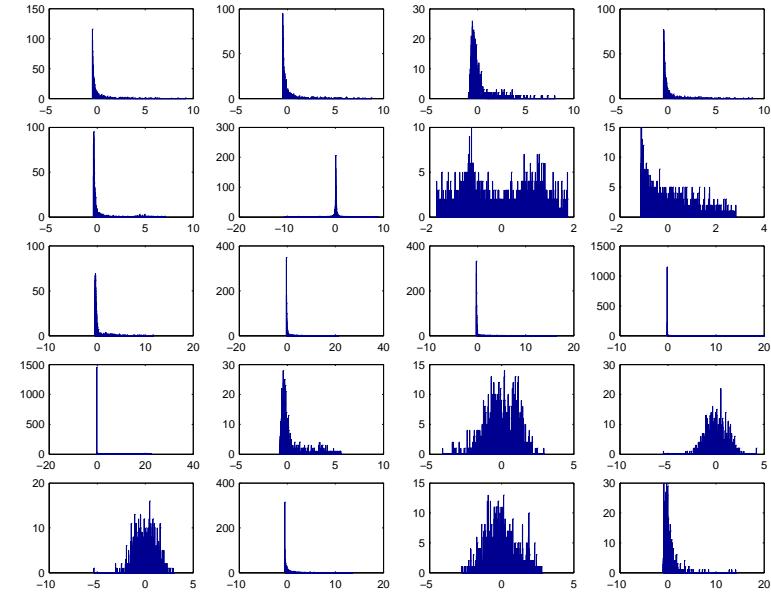


Abbildung 8.3.: Merkmalsverteilungen des Datensatzes, der nur Spritzerobjekte enthält (Merkmale v.l.n.r. und v.o.n.u.)

nen Zerlegungen mit der wirklichen Partitionierung der Daten vergleichen. Für das weitere Vorgehen müssen wir in der Lage sein, den Grad der Übereinstimmung zu beurteilen. Wir haben uns für den von Fowlkes und Mallows (1983) definierten Index entschieden.

Seien \mathcal{C}^1 und \mathcal{C}^2 zwei Zerlegungen derselben Punktmenge \mathcal{S} . Sei nun p_{12} die Wahrscheinlichkeit dafür, dass ein zufällig gewähltes Punktpaar aus einem Cluster der Zerlegung \mathcal{C}^1 auch einem Cluster der Zerlegung \mathcal{C}^2 angehört. Analog ist p_{21} definiert. Der *Fowlkes-Mallows-Index*, kurz $I_{FM}(\mathcal{C}^1, \mathcal{C}^2)$, ist als geometrischer Mittelwert der Wahrscheinlichkeiten p_{12} und p_{21} definiert:

$$I_{FM}(\mathcal{C}^1, \mathcal{C}^2) = \sqrt{p_{12}p_{21}}.$$

8.6.2. Heuristische Wahl

Hier stellen wir Ergebnisse, die mittels topologischer Heuristiken (vgl. Kapitel 5.2 auf Seite 40) erzielt wurden. Die detaillierte Ergebnissprotokolle finden sich auf den Seiten 65 – 66.

Synthetische Daten

Die Heuristiken lieferten auf den synthetischen Daten gute Ergebnisse. Eine Ausnahme bildete der *bullseye* Datensatz aus dem Turing-Test (vgl. Seite 71). Hier ist die Struktur des Datensatzes die Ursache. Ferner jedoch wurden Zerlegungen von teilweise sehr hoher Qualität generiert, wie aus den Tabellen A.1 und A.2 hervorgeht.

Reale Datensätze

Auf realen Datensätzen erzielten die topologischen Heuristiken schlechte bis mittelmäßige Resultate. Die Ausnahme bildete der *iris* Datensatz. MAXDEG und MST-MAX Heuristiken bildeten 2 Cluster, die deutlich durch PCA Projektion der Samples auf den 3D-Raum zu erkennen sind. Das schlechte Abschneiden auf dem Datensatz *abalone* ist auf die große Überlappung der Daten zurückzuführen. Die 3 Cluster, die vermöge PCA deutlich sichtbar werden, wurden nicht als solche erkannt.

Die restlichen niedrigen Werte sind auf zu kleine Anzahl der Beobachtungen zurück zu führen. Hinzu kommt, dass die Klassen der Daten nicht notwendigerweise durch eine natürliche Häufung der Punkte repräsentiert werden.

Ferner sind diese Datensätze für das Testen der Klassifikatoren bestimmt. Die Klassifikation bildet im Vergleich zum Clustering eine einfachere Aufgabe, denn Klassifikatoren passen sich durch flexible Trennebenen jeder Klasseneinteilung an.

Bilder

Aus den Bildern wird noch einmal der heuristische Aspekt der Regeln deutlich. Vor allem die kontrastarmen Bilder bereiten Schwierigkeiten. Die Ursache für dieses Phänomen ist auf schlechte Cluster trennung im HSV Raum zurückzuführen.

Die besten Resultate liefern MAXDEG und MSTMAX Heuristiken. Was vor allem für die MAXDEG Heuristik bemerkenswert ist, da diese die wenigsten Informationen über die Topologie des Nachbarschaftsgraphen benutzt. Die schlechte Qualität der

8. Experimentelle Resultate

MAXRADIUS Heuristik ist im stärkeren Zusammenhang des Delaunay-Graphen in mehrdimensionalen Räumen begründet. Da die Anzahl der Delaunay-Nachbarn exponentiell in der Dimension wächst, fällt der Graph nicht wie gewünscht auseinander, was in eine zu große Bandweite resultiert.

8.6.3. Stochastische Modelle und Bewertungsstrategien

In diesem Abschnitt stellen wir Resultate, die wir mittels stochastischer Modellen und der Bewertungsfunktionen erzielen. Die Ergebnisprotokolle sind auf den Seiten 83 bis 96 zu finden.

Synthetische Daten

Aus den Bewertungsstrategien für die Wahl der Bandweite schnitten Silhouetten- und Stabilitäts-Bewertungsfunktionen besonders gut gegenüber den anderen Verfahren ab. Die restlichen Strategien bevorzugten Zerlegungen mit zu großer Bandweite. Dieses war besonders im letzten Datensatz des Turing-Tests (*wheel*) sichtbar (Abb. A.2 und A.3).

Das Stabilitätskriterium erwies sich auch auf den restlichen synthetischen Daten als sehr gut. Der Silhouetten-Index schwankte in seiner Qualität und wies die meisten Aussetzer auf. Stochastische Modelle lieferten mittlere bis gute Ergebnisse, wobei Kleinst-Quadrate-Kreuzvalidierung leicht bessere Ergebnisse erzielte als Maximum Likelihood. Ähnlich gute Ergebnisse lieferte Isolation-und-Kompaktheit-Index, dicht gefolgt von Isolation-und-Ausprägung-Index. Die Werte sind in der Tabelle A.5 zusammengefasst.

Reale Datensätze

Auf den realen Datensätzen wurden ähnlich schlechte Resultate, wie mit heuristischer Wahl der Bandweite, erreicht. Fast alle Methoden bevorzugten Zerlegungen mit nur einem Cluster. Nur der Index nach dem Stabilitätskriterium konnte den *iris* Datensatz in 2 Cluster zerlegen.

8.6.4. Clustermerging

Nun folgen Ergebnisse, die mittels Clustermerging Strategien (s. Kapitel 6) entstanden sind. Die Ergebnissen sind auf den Seiten A.3 bis A.4 protokolliert.

Synthetische Daten

Aus dem direktem Vergleich (vgl. Abb. A.4) wird deutlich, dass die statistische Methode benutzend die Sattelpunkte als Kriterium deutlich besser abschnitt. Der Zwang der ersten Methode, Cluster zusammen zu führen, „schießt oft über das Ziel hinaus“ und resultierte meist in einem oder mehreren großen Clustern, die keine natürliche Gruppierung der Daten widerspiegeln. Die initiale Lösung generierten wir mit dem BST-Algorithmus, wobei die Bandweite auf den festen Wert von 0.25 für alle Instanzen gesetzt wurde.

Auf den mehrdimensionalen synthetischen Datensätzen erzielte dagegen die Methode basierend auf dem Ausprägungskriterium leicht bessere Ergebnisse (vgl. Tabellen A.9 und A.10).

Reale Datensätze

Auch an realen Datensätzen konnten wir zeigen, dass die Qualität der initialen Lösung deutlich gesteigert werden konnte. Jedoch blieben diese erneut unter unseren Erwartungen (vgl. A.11 und A.12).

Bilder

Der Trend, der in den synthetischen Daten sich abzeichnete, setzte sich in den Bilder fort. Das Erzwingen der Clusterzusammenführung nach dem Ausprägungskriterium stellte eine zu starke Konsequenz dieser Eigenschaft der Cluster dar. Eine genauere Kontrolle der Zusammenführung zweier Cluster ist notwendig. Das statistische Testen der Clustersignifikanz dagegen, eliminierte, wie in einigen Bildern deutlich wird (Seite 96 ff.), nur die wirklich schwach ausgeprägten Cluster. Der Konfidenzwert für die Signifikanz der Sattelpunkte stellt einen sinnvollen Kontrollparameter, denn wir beim Merging nach Ausprägung vermissten, dar.

8.6.5. Spritzer

Die Klassifizierung der Spritzerobjekte gestaltete sich sehr schwierig. Alle Versuche der Dimensionsreduzierung mit anschließendem Clustering scheiterten. Erst unter manueller Auswahl bestimmter Merkmale gelang es die Spritzerobjekte in mehrere Subtypen einzuteilen. Dies geschah unter der Hinzunahme der Merkmalsverteilung (s. Abb. 8.2 und 8.3). Merkmale die Subklassifizierung ermöglichen sind 2, 15 und 19. Auf der Seite 110 sind die Prototypen, die Spritzobjekte, die Zentren der Cluster darstellen, abgebildet. Dieses Clustering wurde mittels Silhouetten-Koeffizienten-Bewertungsfunktion bestimmt.

8.7. Zusammenfassung

Aus den topologischen Heuristiken haben sich besonders MAXDEG und MSTMAX Heuristiken bewährt. Das Stabilitätskriterium stellt für die Beurteilung mehrerer Zerlegung eine effiziente und aussagekräftige Methode dar. Die Silhouetten-Koeffizienten-Bewertungsfunktion und Isolation-und-Kompaktheit-Bewertungsfunktionen schneiden bei den Bildern am besten ab. Aus Clustermerging Strategien geht die Strategie nach dem Sattelpunkt-Kriterium eindeutig als Sieger hervor.

9. Fazit

Zum Schluss wollen wir ein Fazit aus den ermittelten Ergebnissen ziehen und einen Ausblick auf weitere Untersuchungen geben.

9.1. Folgerungen

Bst-Algorithmus

In Kapitel 4 stellten wir den BST-Algorithmus vor. Wir beschrieben, wie die iterative Version des Schnell-Algorithmus (Alg. 2.8) durch einen graphenbasierten Algorithmus ersetzt werden kann. Die Vorteile dieser Variante liegen auf der Hand, nämlich einmalige Auswertung der Kernfunktion und des Gradienten. Andererseits entsteht der Nachteil der Triangulierung, die den Algorithmus für die Dimensionen größer gleich vier unattraktiv macht, bzw. der k -Nachbarschaft, die einen neuen Freiheitsgrad einführt. Die Anwendung des Algorithmus auf sehr hochdimensionierten Daten wirft die Fragen nach den Dimensionreduktionstechniken bzw. Merkmalsselektionsalgorithmen auf. Diese stellen jedoch abgeschlossene Problemklassen in sich dar. Die Problematik der mehrdimensionalen Kerndichteschätzung wurde während der Spritzerklassifizierung sehr deutlich.

Bandweitenwahl

Zudem stellten wir in dieser Arbeit mehrere Methoden zur automatischen Bestimmung der Bandweite vor und testeten diese auf unterschiedlichen Instanzen. Hierbei erwiesen sich topologische Heuristiken auf den Daten, denen eine Struktur unterliegt, als teilweise sehr gut. Für hochdimensionale Verteilungen gilt das Problem der optimalen Bandweitenwahl weiterhin als offen.

Clustermerging

Zwei hybride hierarchische Verfahren wurden vorgestellt. Wir zeigten, dass beide Verfahren die Qualität der initialen Lösungen erheblich steigern können. Somit reduzierten wir das Problem der Bandweitenwahl auf die Wahl einer sehr kleinen Bandweite. Hierbei adaptierten wir erfolgreich den statischen Test für die Clustersignifikanz von Comaniciu u. a. (2002). Wir ersetzten den iterativen Mean-Shift-Prozess für die Bestimmung der Sattelpunkte durch einen graphenbasierten Ansatz. Hierzu nutzten wir die Eigenschaft der Basin Spanning Trees und der Nachbarschaft der Punkte aus.

Optimierungstechnik

Wir haben eine Technik vorgestellt vermöge der eine Auswertung der Epanechnikov-Kernfunktion beschleunigt werden konnte. Diese lässt sich auf eine Reihe von weiteren Kernfunktionen mit einem endlichen Trägersupport anwenden.

9.2. Ausblick

Es bleibt zu klären, in wie weit ein graphenbasierter Ansatz des Dichteclusterings für die Analyse sehr hochdimensionaler Daten erweiterbar ist. Dies würde Untersuchungen und Definition neuer Nachbarschaften, z.B. eine Kombination zwischen Delaunay-Triangulierung und k -Nachbarschaft, mit effizienter Berechnung und „beserer“ Lokalität als Schwerpunkte erfordern.

Laufzeit

Da die Implementation der untersuchten Methoden unter MATLAB, die eine interpretierte Programmiersprache darstellt, entstand, erwarten wir eine erhebliche Laufzeitsteigerung für systemnahe Programmiersprachen (z.B. C/C++).

A. Ergebnisprotokolle

Im Weiteren sind die Tabellen wie folgt zu lesen. Die erste Spalte trägt die Spezifikation der Testinstanz. Diese setzt sich aus dem Namen, Anzahl der Klassen (falls spezifiziert), der Anzahl der Datenpunkte und deren Dimension zusammen. Darauf folgt eine Reihe von Zahlenpaaren. Die erste Zahl gibt die Anzahl der Klassen und die zweite die Qualität (nach dem FM-Index) für die (bevorzugte) generierte Zerlegung einer Methode an. Die Qualität einer Zerlegung teilen wir in vier Kategorien auf. Die Qualität der Zerlegung gilt als **sehr gut**, wenn diese Werte im Intervall (0.9, 1.0] annimmt, **gut** für die Werte aus dem Intervall (0.75, 0.9], **mittelmäßig** für Werte aus (0.5, 0.75] und **schlecht**, wenn die Werte im Bereich [0.0, 0.5] liegen.

Die Bilder sind nach folgendem Schema gruppiert: Wir betrachten die Bilder von rechts nach links und von oben nach unten. Zuerst kommt das Original, gefolgt von der skalierten Darstellung, welche die Eingabe für den Clustering Algorithmus bildet.

Im Falle der heuristischen Bandweite folgen Zerlegungen, die mittels MAXDEG, MSTMAX und MAXRADIUS Heuristiken entstanden sind.

Im Falle der Clustermerging-Methoden stellt das mittlere Bild die initiale Zerlegung dar. Es folgen resultierende Zerlegungen, die von Clustermerging nach Ausprägungs- und Sattelpunkt-Kriterium erzeugt wurden.

A.1. Heuristische Wahl

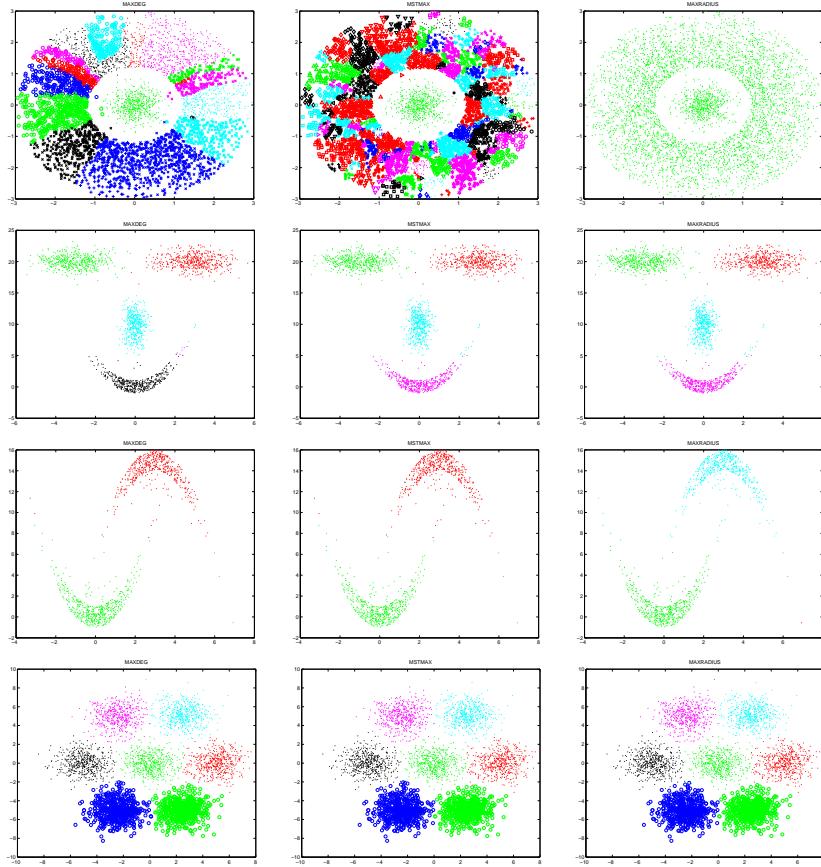


Abbildung A.1.: Turing-Test (Heuristiken v.l.n.r.: MAXDEG, MSTMAX, MAXRADIUS)

Datensatz	MAXDEG		MstMAX		MAXRADIUS	
	Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $
bullseye/004560/02	18	0.32	132	0.18	1	0.89
sickels/001050/02	2	0.97	2	0.97	3	0.97
smile/002000/02	5	0.99	4	0.98	4	0.98
wheel/003500/02	7	0.97	7	0.97	7	0.97

Tabelle A.1.: Turing-Test: Resultate die mittels topologischer Heuristiken (MAXDEG, MSTMAX, MAXRADIUS) erzielt wurden.

A. Ergebnisprotokolle

Datensatz	MAXDEG		MSTMAX		MAXRADIUS	
	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
synth/2/001000/02	22	0.64	3	0.88	2	0.91
synth/2/001000/03	2	0.99	3	0.99	2	0.97
synth/2/001000/04	4	1	2	1	1	0.73
synth/2/001000/05	4	1	2	1	2	1
synth/2/001000/06	2	0.72	2	0.72	1	0.72
synth/3/001000/02	3	0.96	3	0.97	3	0.95
synth/3/001000/03	54	0.88	2	0.87	3	0.95
synth/3/001000/04	3	0.8	2	0.8	1	0.6
synth/3/001000/05	3	1	3	1	3	1
synth/3/001000/06	105	0.78	2	0.73	1	0.6
synth/4/001000/02	5	0.94	4	0.93	2	0.75
synth/4/001000/03	29	0.96	3	0.96	2	0.87
synth/4/001000/04	2	0.71	5	0.93	1	0.54
synth/4/001000/05	4	0.99	4	0.94	4	0.97
synth/4/001000/06	5	0.62	2	0.6	1	0.56
synth/5/001000/02	4	0.6	3	0.59	1	0.51
synth/5/001000/03	10	0.97	5	0.97	1	0.58
synth/5/001000/04	14	0.97	6	0.99	1	0.52
synth/5/001000/05	14	0.97	3	0.9	1	0.61
synth/5/001000/06	46	0.94	3	0.75	3	0.76

Tabelle A.2.: Synthetische Daten: Clustering Resultate vermöge heuristischer Bandweitenwahl (MAXDEG, MSTMAX, MAXRADIUS).

A.1. Heuristische Wahl

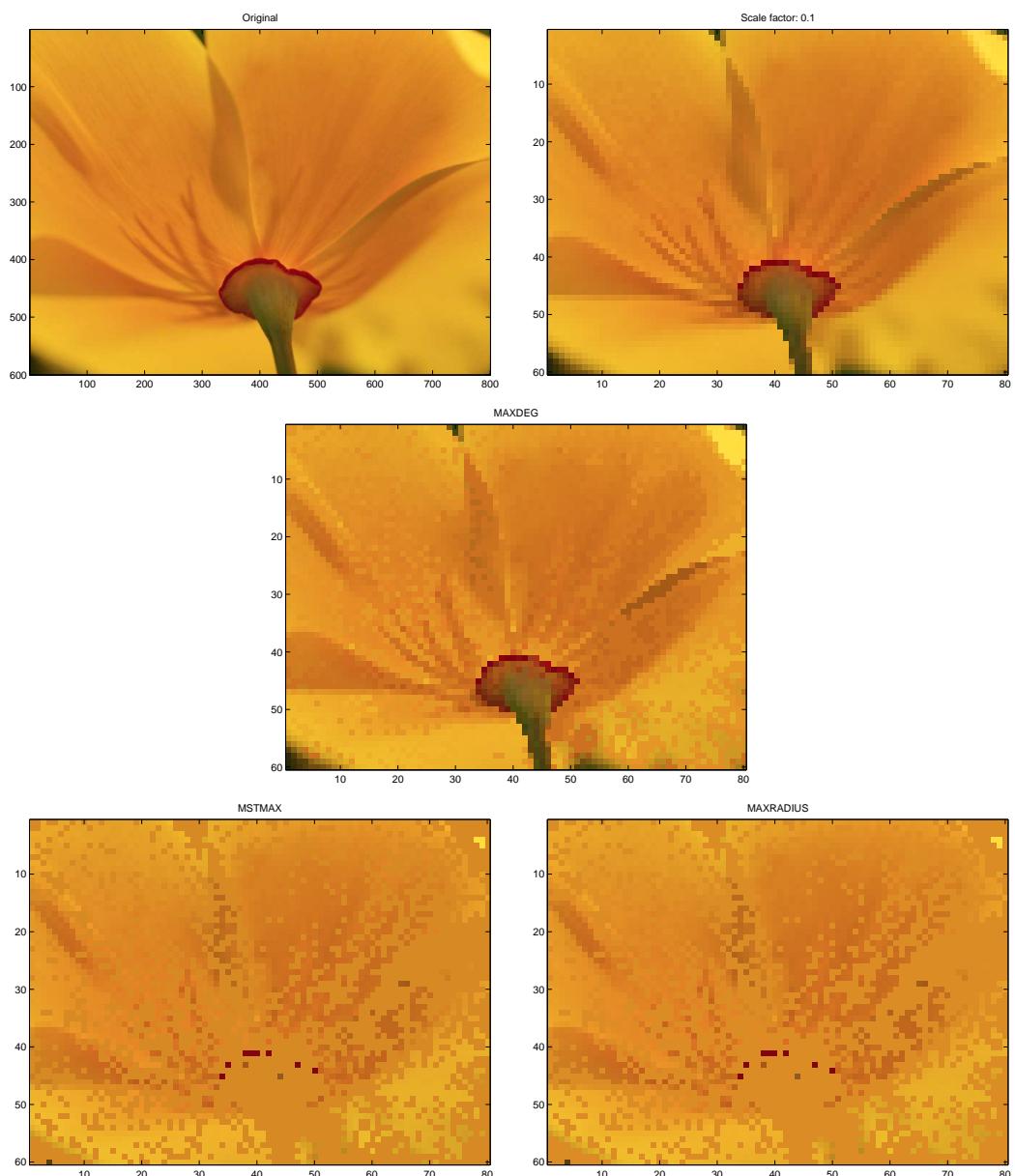
Datensatz	MAXDEG		MSTMAX		MAXRADIUS	
Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
abalone/004177/03	1	0.32	1	0.32	1	0.32
glass/000214/03	11	0.58	3	0.54	1	0.51
hayes-roth/000132/03	1	0.59	6	0.32	1	0.59
iris/000150/03	2	0.76	2	0.76	1	0.57
liver-disorders/000345/03	8	0.7	1	0.72	1	0.72
tic-tac-toe/000958/03	1	0.74	1	0.74	1	0.74
wine/000178/03	3	0.57	1	0.58	1	0.58
yeast/001484/03	1	0.47	1	0.47	1	0.47
abalone/004177/04	4	0.32	1	0.32	1	0.32
glass/000214/04	7	0.51	1	0.51	1	0.51
hayes-roth/000132/04	1	0.59	1	0.59	1	0.59
iris/000150/04	1	0.57	2	0.77	1	0.57
liver-disorders/000345/04	12	0.69	1	0.72	2	0.71
tic-tac-toe/000958/04	1	0.74	9	0.64	1	0.74
wine/000178/04	3	0.57	1	0.58	1	0.58
yeast/001484/04	2	0.47	1	0.47	1	0.47

Tabelle A.3.: Reale Datensätze. Obere Tabelle: Daten die mittels PCA auf 3 Dimensionen reduziert sind. Untere Tabelle: Reduktion auf 4 Dimensionen.

A. Ergebnisprotokolle



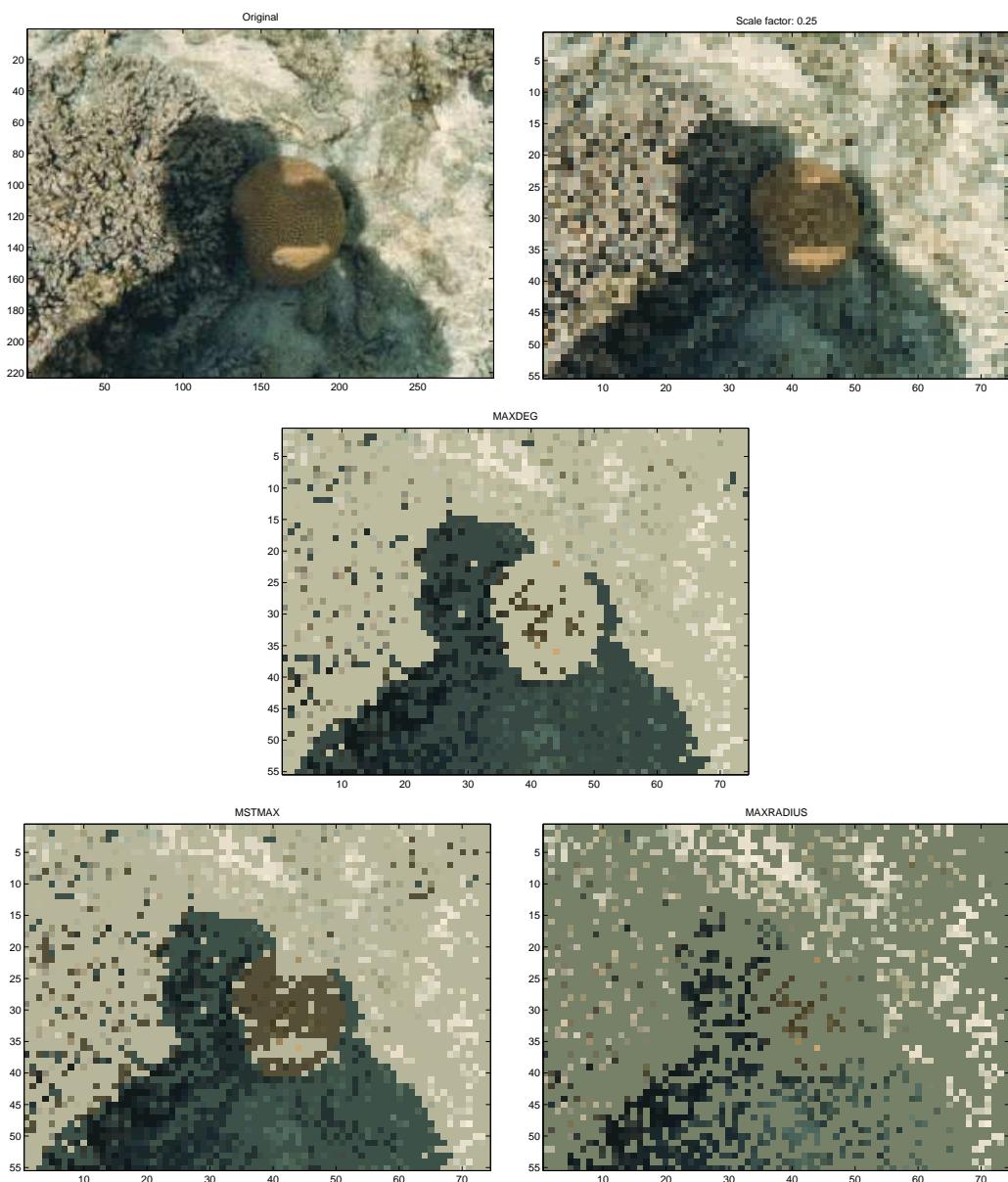
A.1. Heuristische Wahl



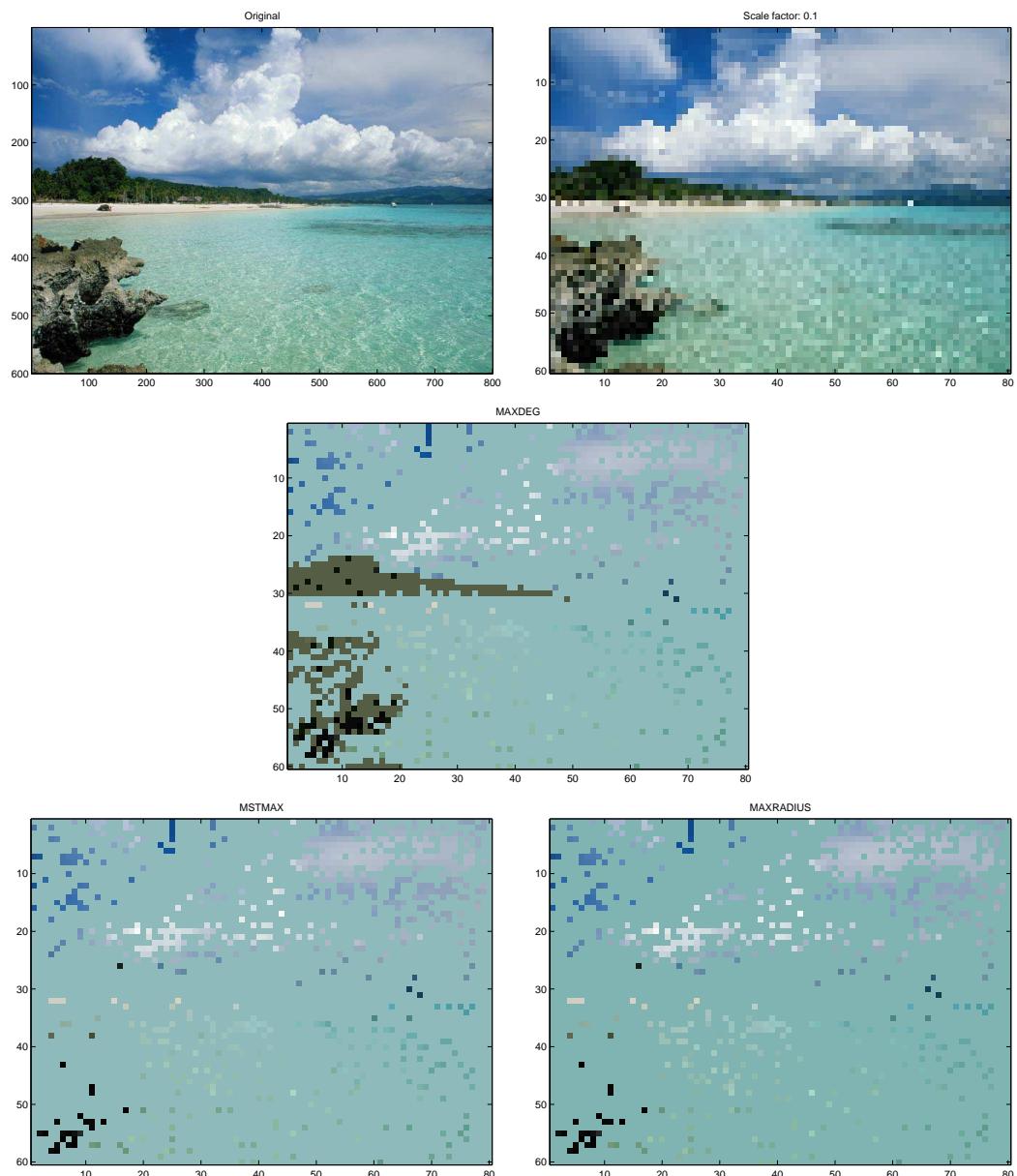
A. Ergebnisprotokolle



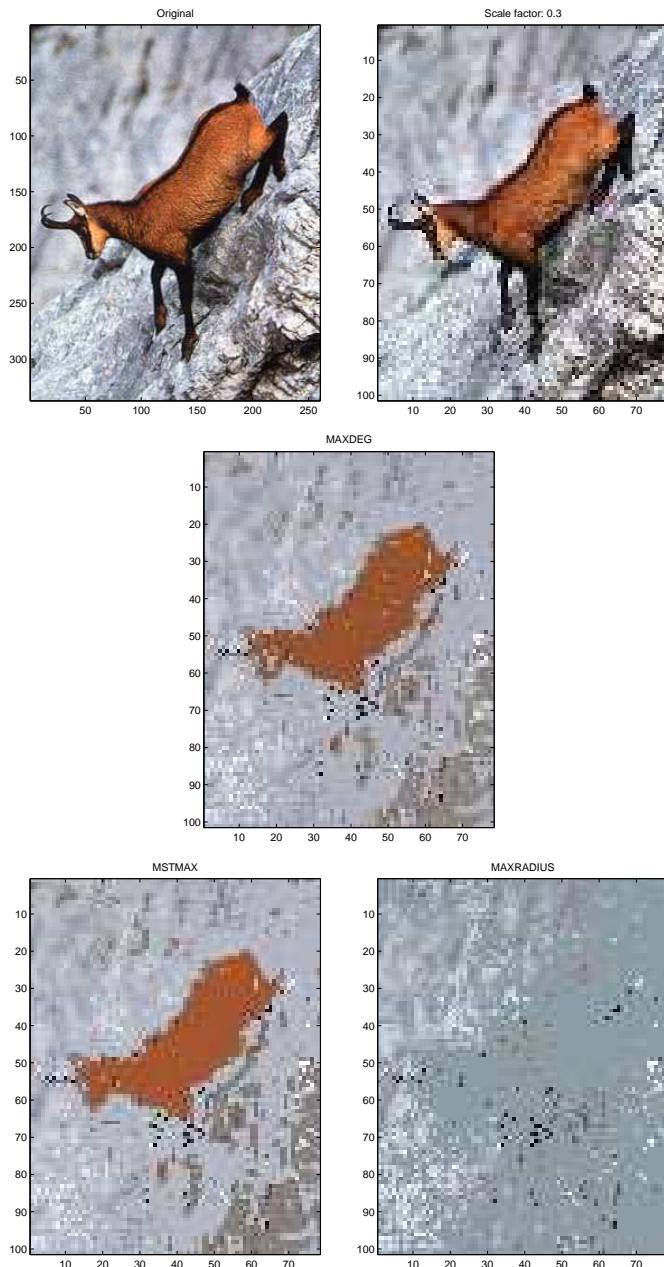
A.1. Heuristische Wahl



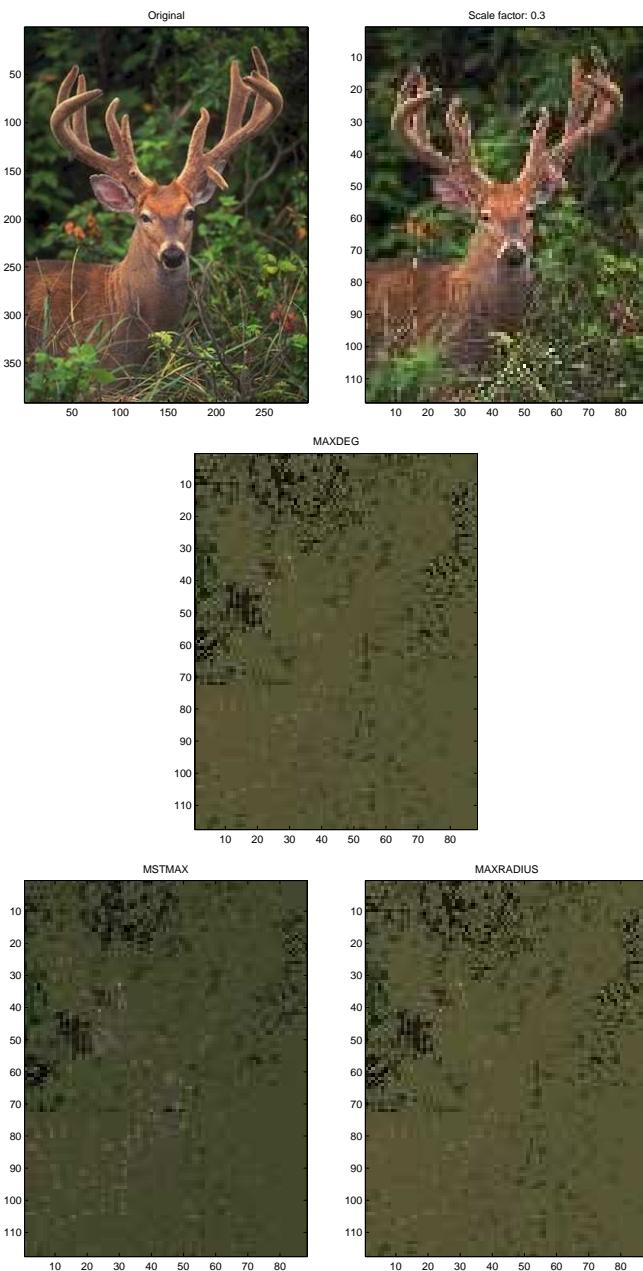
A. Ergebnisprotokolle



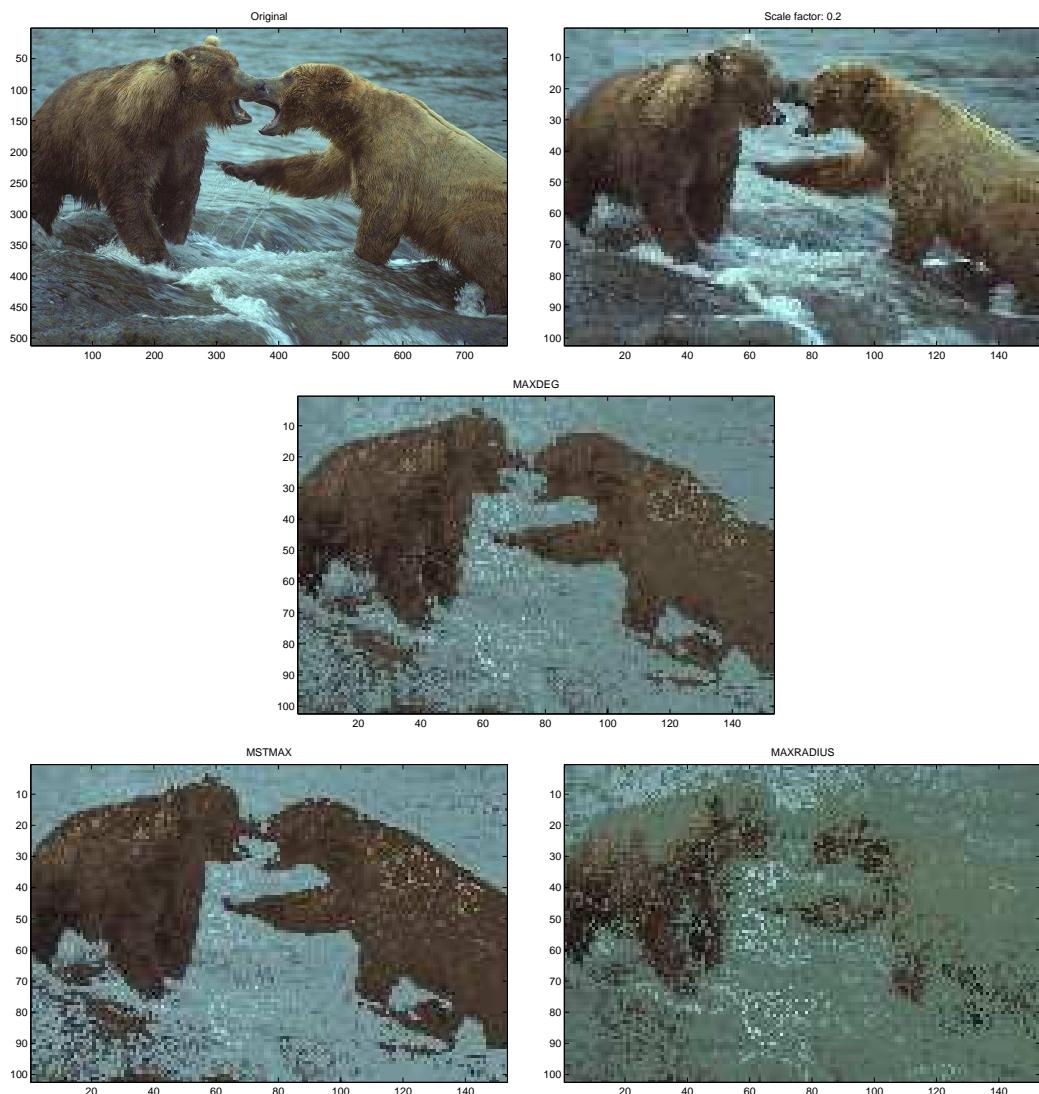
A.1. Heuristische Wahl



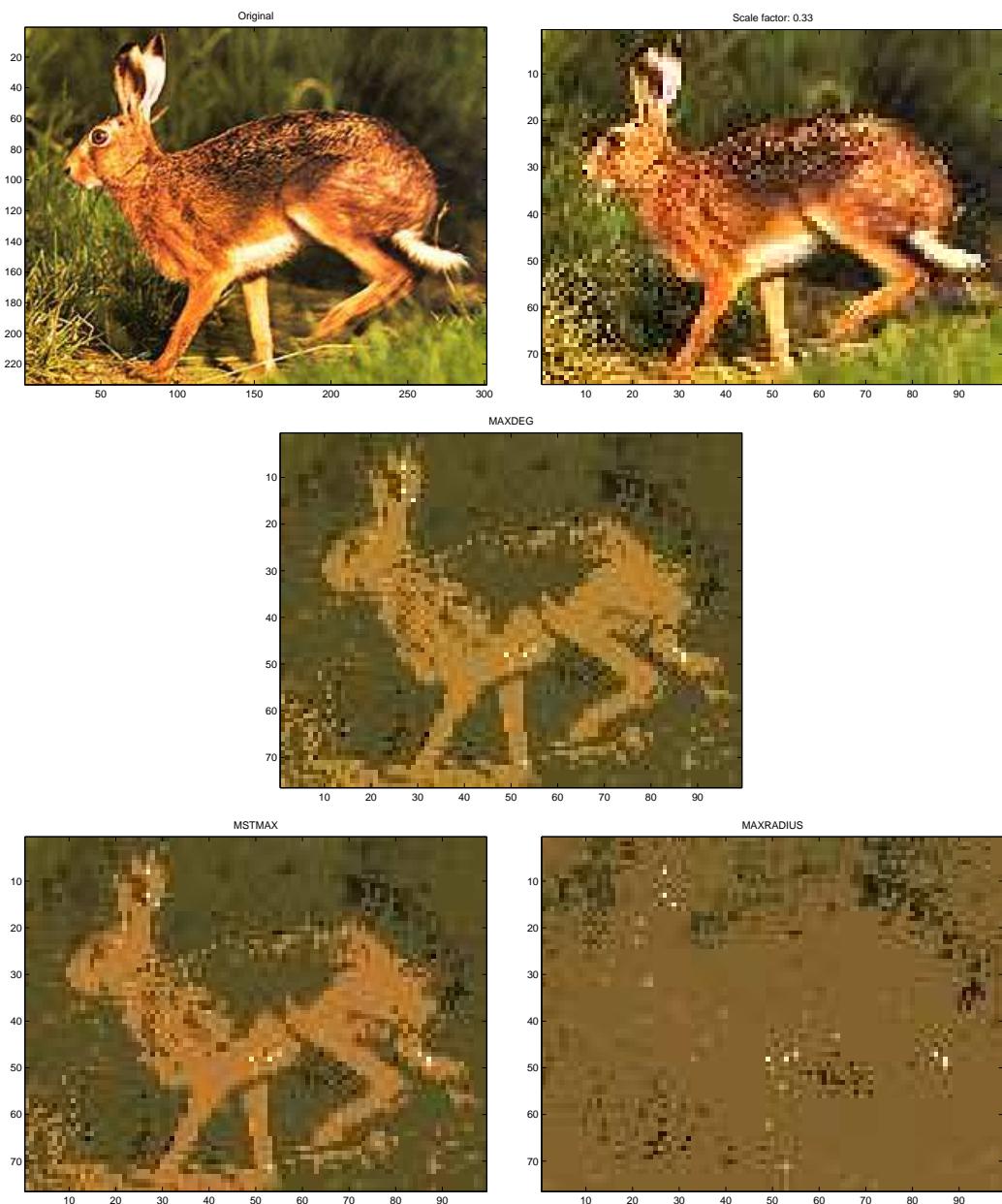
A. Ergebnisprotokolle



A.1. Heuristische Wahl



A. Ergebnisprotokolle



A.2. Bewertungsfunktionen und stochastische Modelle

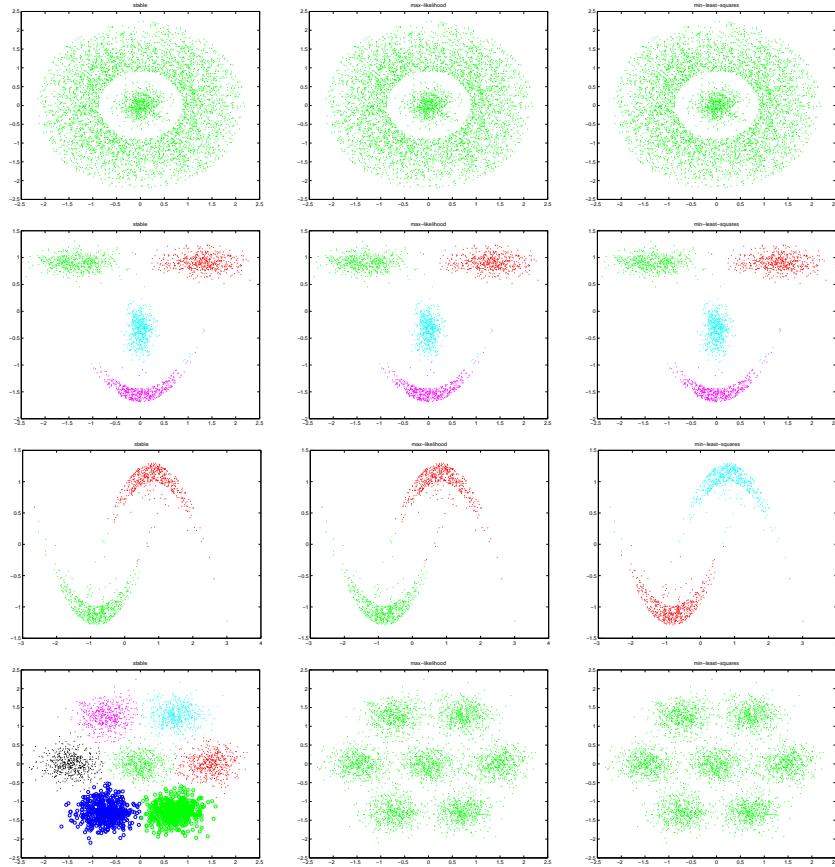


Abbildung A.2.: Turing-Test (Bewertungsfunktionen v.l.n.r.): Stabilitätskriterium, Likelihood-Kreuzvalidierung, Kleinstes-Quadrat-Kreuzvalidierung

A. Ergebnisprotokolle

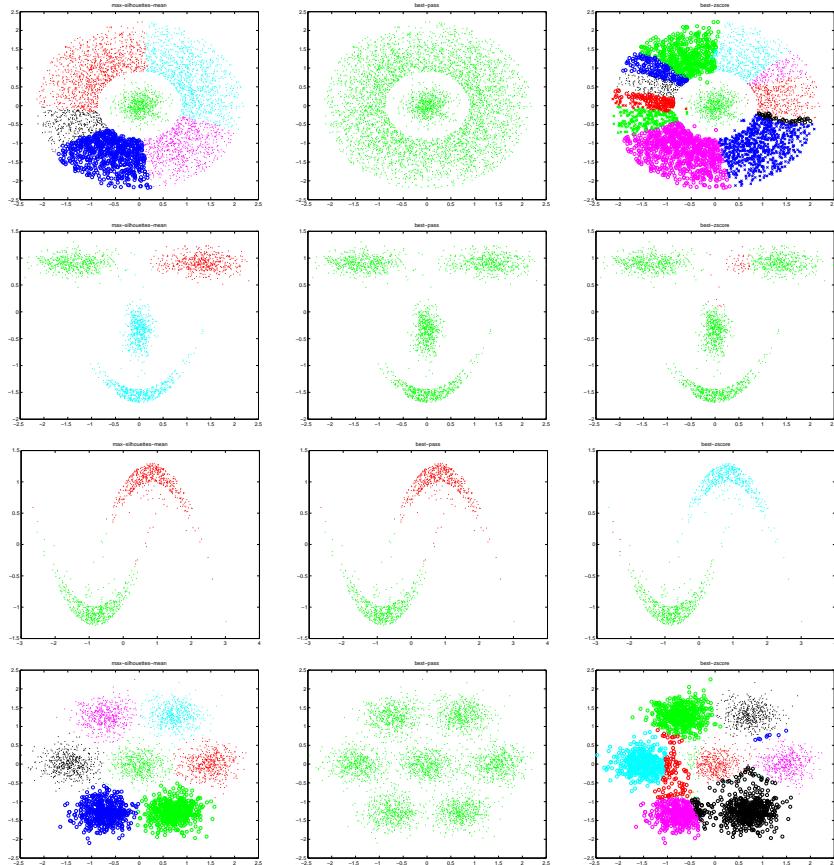


Abbildung A.3.: Turing-Test (Fortsetzung der Bewertungsfunktionen): Silhouetten-Koeffizienten, Isolation-und-Ausprägung und schließlich Isolation-und-Kompaktheit

Datensatz	Stabilität		Likelihood		Least Squares		Silhouette		Iso&Ausprägung		Iso&Kompaktheit	
	N/ C /S/D	C	FM-Index	C	FM-Index	C	FM-Index	C	FM-Index	C	FM-Index	C
bullseye/004560/02	1	0.89	1	0.89	1	0.89	6	0.49	1	0.89	14	0.38
sickels/001050/02	2	0.97	2	0.97	3	0.97	2	0.97	2	0.97	3	0.97
smile/002000/02	4	0.98	4	0.98	4	0.99	3	0.81	1	0.5	2	0.49
wheel/003500/02	7	0.97	1	0.38	1	0.38	7	0.97	1	0.38	11	0.88

Tabelle A.4.: Turing-Test: Ermittelte Ergebnisse für 50 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 2.0]

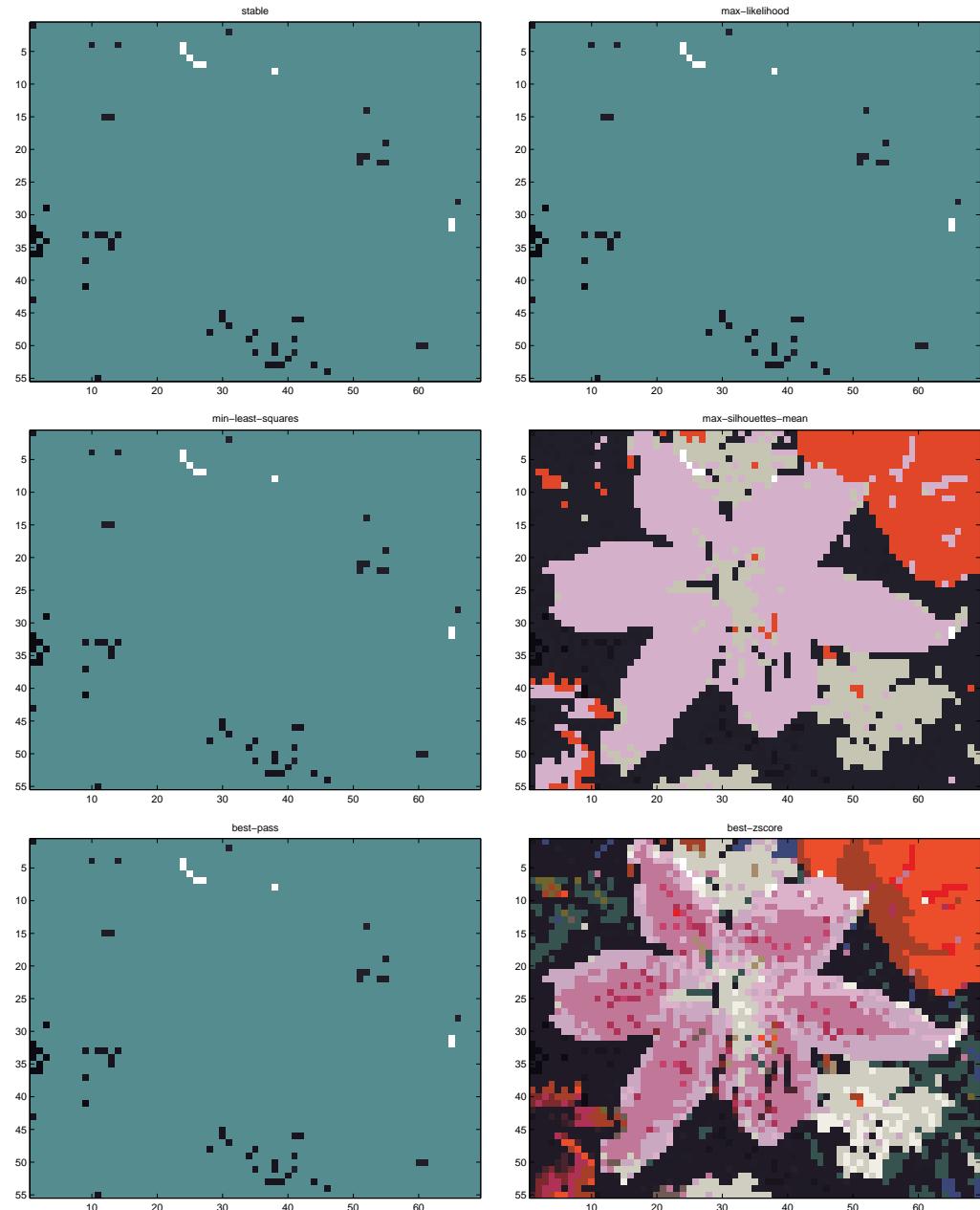
Datensatz	Stabilität		Likelihood		Least Squares		Silhouette		Iso&Ausprägung		Iso&Kompaktheit	
	N/ C /S/D	C	FM-Index	C	FM-Index	C	FM-Index	C	FM-Index	C	FM-Index	C
synth/2/001000/02	1	0.72	1	0.72	1	0.72	3	0.76	1	0.72	7	0.69
synth/2/001000/03	2	0.94	1	0.73	1	0.73	2	0.95	1	0.73	11	0.72
synth/2/001000/04	2	0.87	1	0.73	1	0.73	891	0.046	1	0.73	1	0.73
synth/2/001000/05	1000	0	2	0.68	2	0.68	1000	0	2	0.71	2	0.66
synth/2/001000/06	1	0.72	1	0.72	1	0.72	1000	0	1	0.72	1	0.72
synth/3/001000/02	3	0.97	1	0.63	3	0.84	3	0.96	1	0.63	32	0.45
synth/3/001000/03	2	0.83	1	0.62	2	0.79	4	0.89	1	0.62	10	0.88
synth/3/001000/04	2	0.8	2	0.79	2	0.79	2	0.8	2	0.79	2	0.79
synth/3/001000/05	3	0.85	3	0.65	3	0.65	1000	0	76	0.52	4	0.93
synth/4/001000/02	3	0.89	3	0.79	3	0.89	3	0.89	1	0.57	7	0.93
synth/4/001000/03	4	0.95	1	0.62	1	0.62	4	0.96	1	0.62	5	0.97
synth/4/001000/04	4	0.92	2	0.71	2	0.71	792	0.11	3	0.79	2	0.71
synth/4/001000/05	4	0.84	4	0.71	4	0.81	812	0.12	179	0.49	6	0.87
synth/5/001000/02	4	0.86	1	0.51	4	0.86	4	0.86	1	0.51	27	0.75
synth/5/001000/03	5	0.94	2	0.6	2	0.6	2	0.62	2	0.61	13	0.67
synth/5/001000/04	5	0.99	3	0.75	3	0.75	3	0.8	3	0.75	3	0.8
synth/5/001000/05	3	0.88	3	0.88	3	0.88	959	0.048	3	0.88	4	0.89

Tabelle A.5.: Synthetische Daten: Ergebnisse der Bewertungsstrategien für 50 logarithmisch gewählte Bandweiten aus dem Intervall [0.1, 2.0]

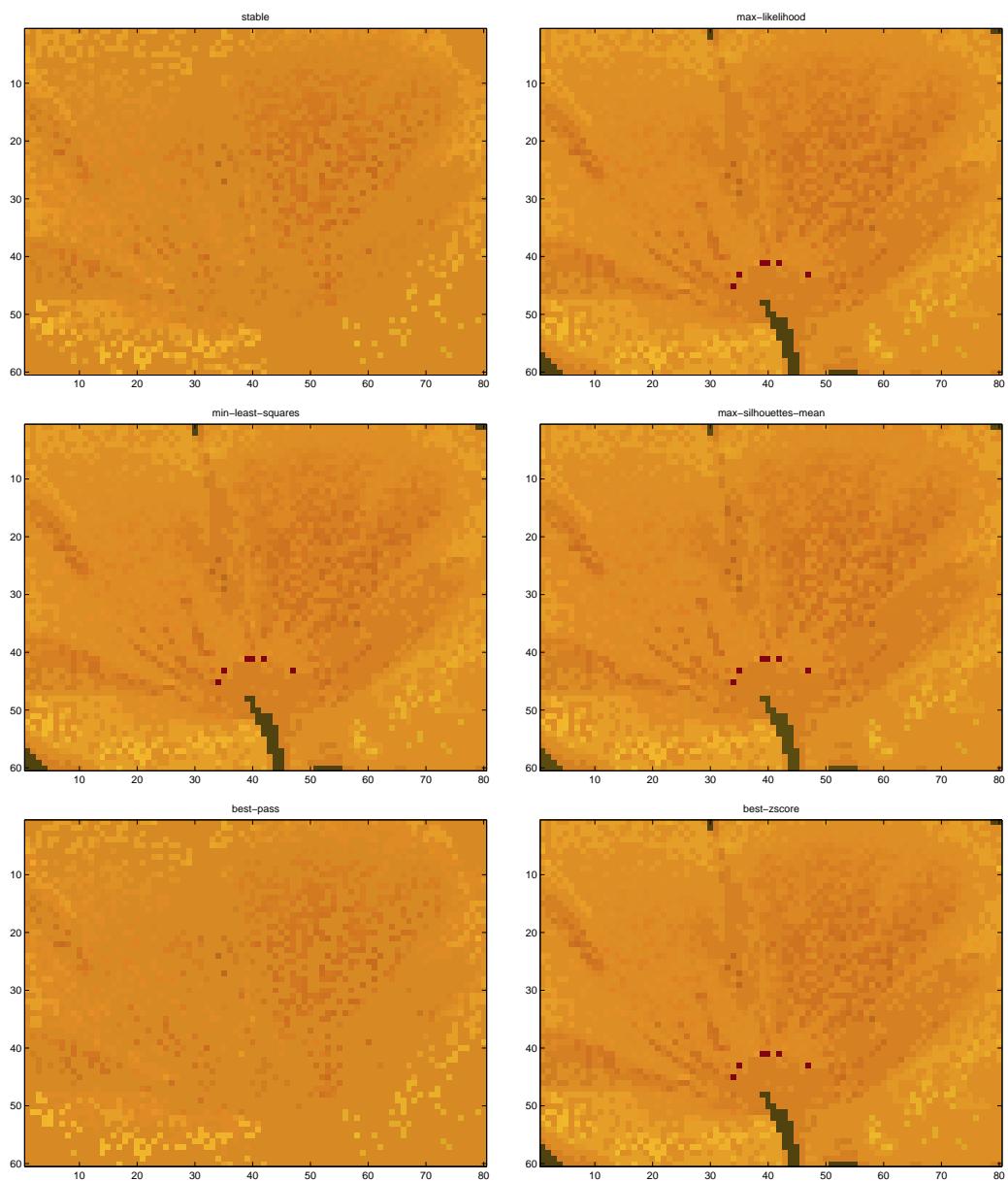
Datensatz	Stabilität		Likelihood		Least Squares		Silhouette		Iso&Ausprägung		Iso&Kompaktheit	
	N/S/D	\mathcal{C}	FM-Index	\mathcal{C}	FM-Index	\mathcal{C}	FM-Index	\mathcal{C}	FM-Index	\mathcal{C}	FM-Index	\mathcal{C}
abalone/004177/03	2	0.25	1	0.32	1	0.32	2	0.25	1	0.32	14	0.24
glass/000214/03	8	0.53	1	0.51	1	0.51	4	0.58	1	0.51	2	0.53
hayes-roth/000132/03	10	0.19	1	0.59	1	0.59	4	0.32	1	0.59	1	0.59
iris/000150/03	2	0.77	1	0.57	1	0.57	125	0.11	1	0.57	1	0.57
liver-disorders/000345/03	1	0.72	1	0.72	1	0.72	2	0.71	1	0.72	3	0.7
liver-disorders/000345/04	1	0.72	1	0.72	1	0.72	3	0.7	1	0.72	5	0.68
tic-tac-toe/000958/03	4	0.43	4	0.45	4	0.45	4	0.42	1	0.74	1	0.74
wine/000178/03	1	0.58	1	0.58	1	0.58	133	0.14	1	0.58	2	0.56
yeast/001484/03	1	0.47	1	0.47	1	0.47	2	0.46	1	0.47	5	0.43
abalone/004177/04	1	0.32	1	0.32	1	0.32	2	0.25	1	0.32	4	0.29
glass/000214/04	6	0.56	1	0.51	1	0.51	3	0.48	1	0.51	5	0.49
hayes-roth/000132/04	9	0.33	29	0.35	1	0.59	3	0.36	1	0.59	1	0.59
iris/000150/04	1	0.57	77	0.29	1	0.57	2	0.5	1	0.57	1	0.57
liver-disorders/000345/04	1	0.72	1	0.72	1	0.72	3	0.7	1	0.72	5	0.68
tic-tac-toe/000958/04	192	0.08	1	0.74	1	0.74	14	0.31	1	0.74	6	0.67
wine/000178/04	178	0	178	0	1	0.58	178	0	1	0.58	1	0.58
yeast/001484/04	1	0.47	1	0.47	1	0.47	2	0.46	1	0.47	3	0.45

Tabelle A.6.: Reale Datensätze: Resultate der Bewertungsstrategien

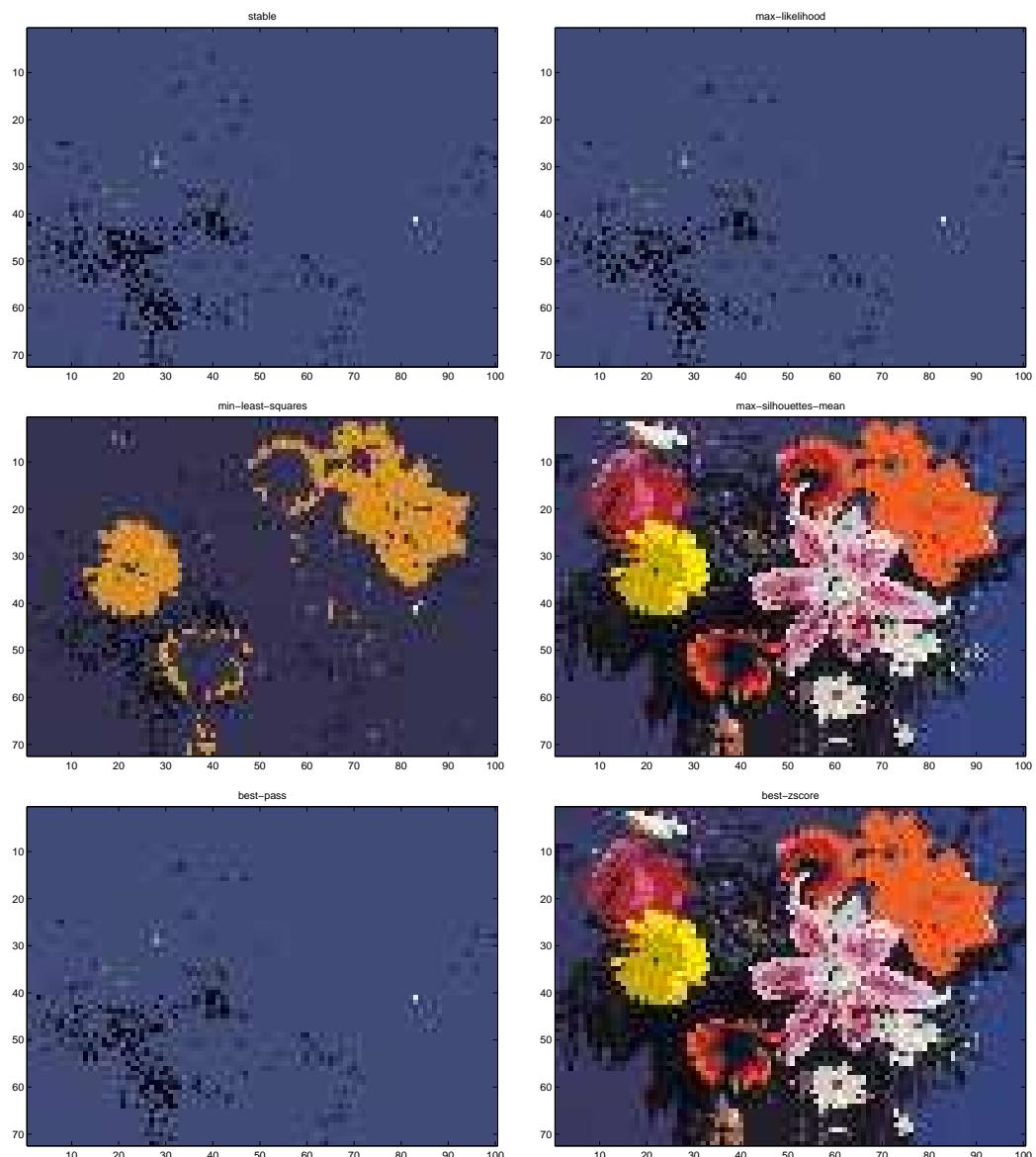
A. Ergebnisprotokolle



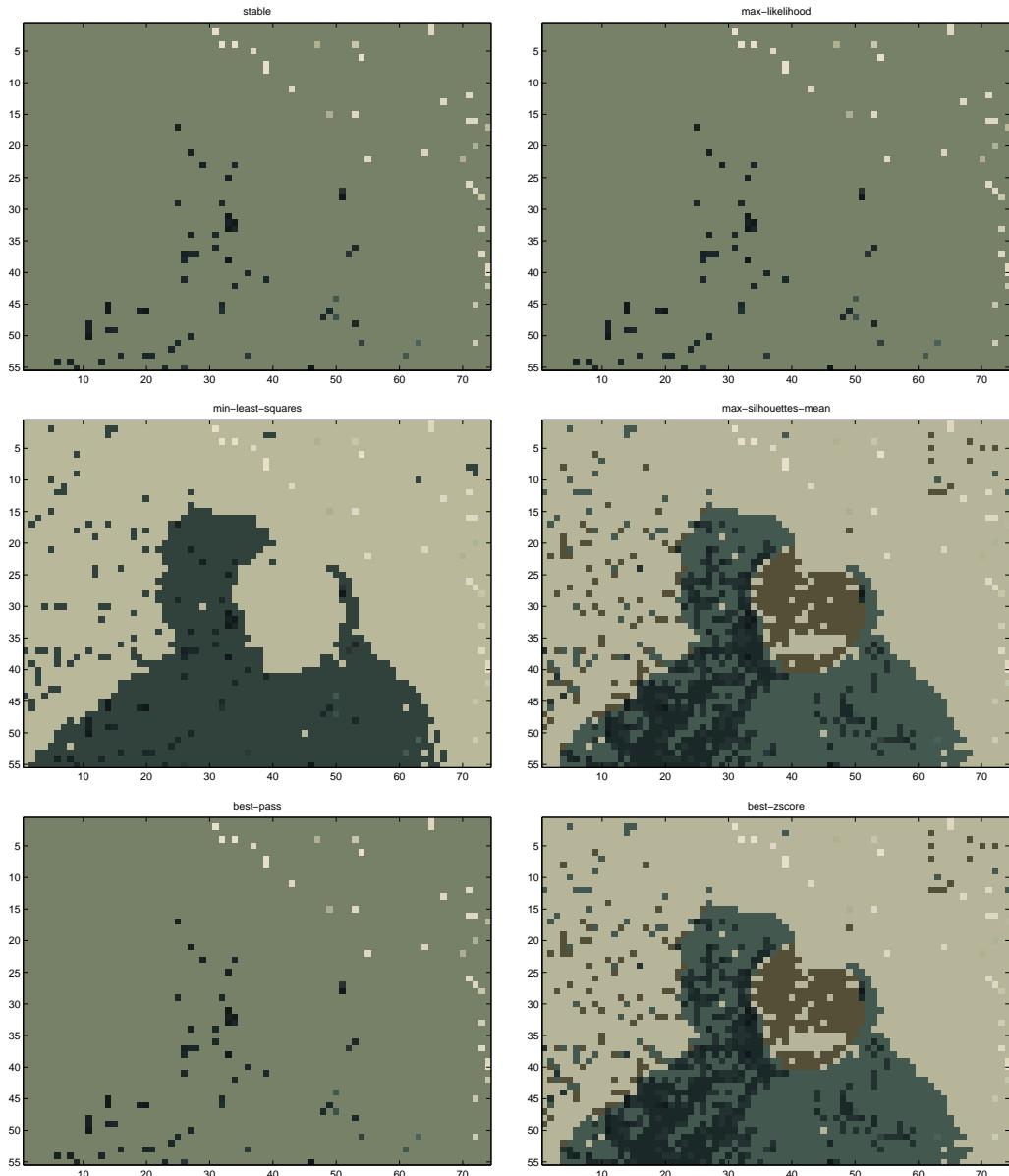
A.2. Bewertungsfunktionen und stochastische Modelle



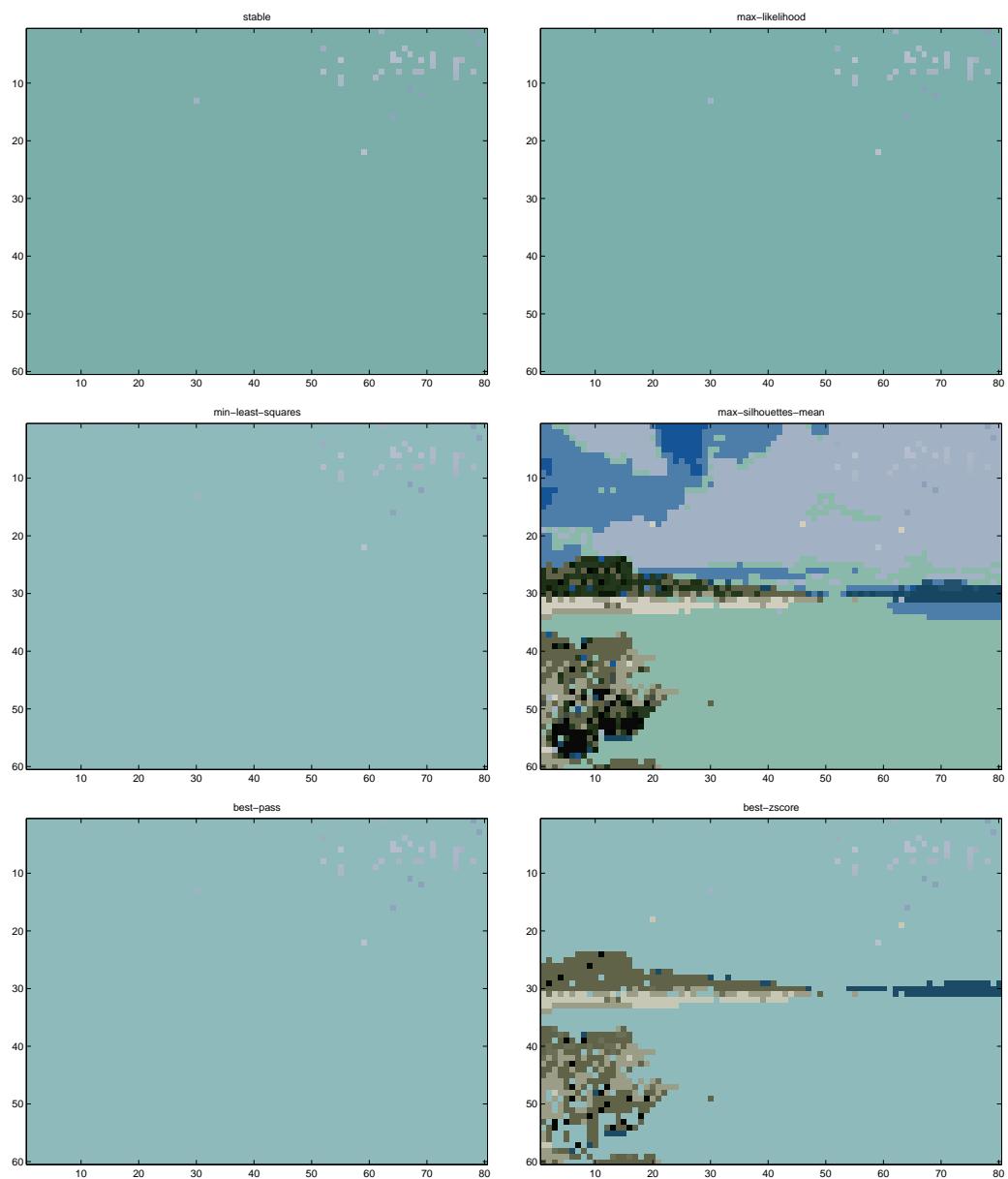
A. Ergebnisprotokolle



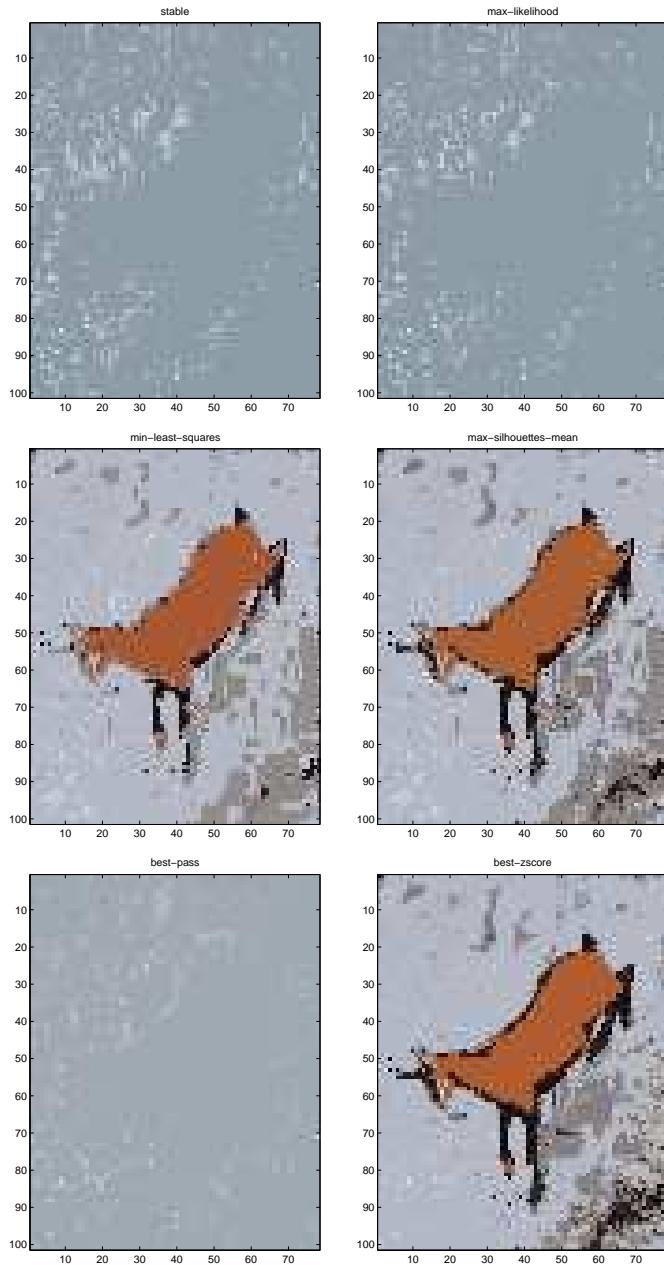
A.2. Bewertungsfunktionen und stochastische Modelle



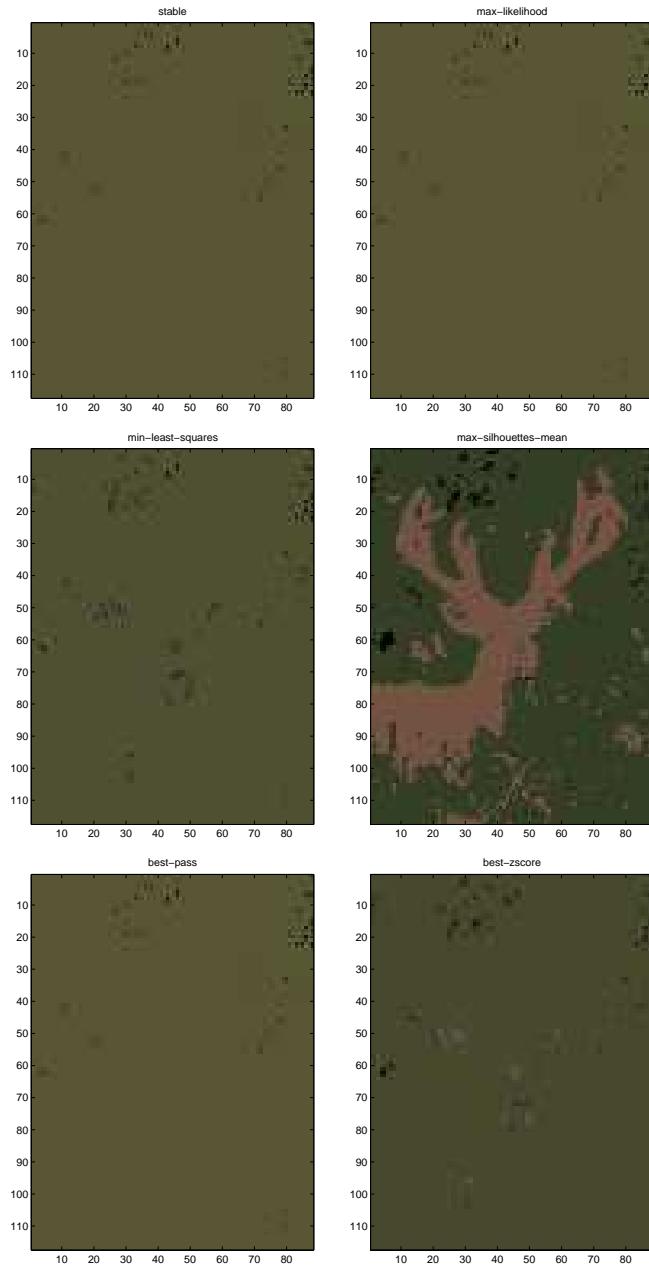
A. Ergebnisprotokolle



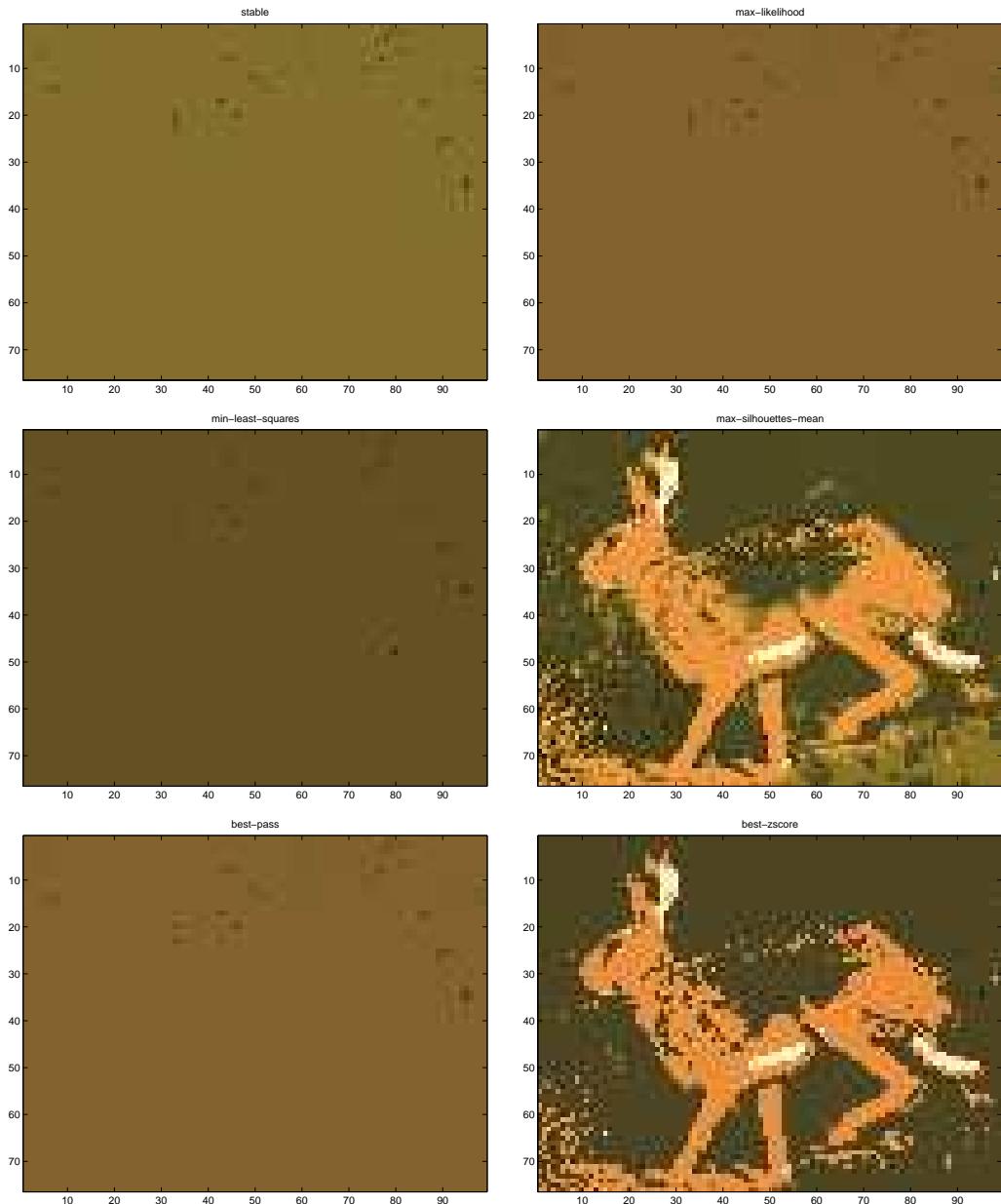
A.2. Bewertungsfunktionen und stochastische Modelle



A. Ergebnisprotokolle



A.2. Bewertungsfunktionen und stochastische Modelle



A. Ergebnisprotokolle

A.3. Clustermerging

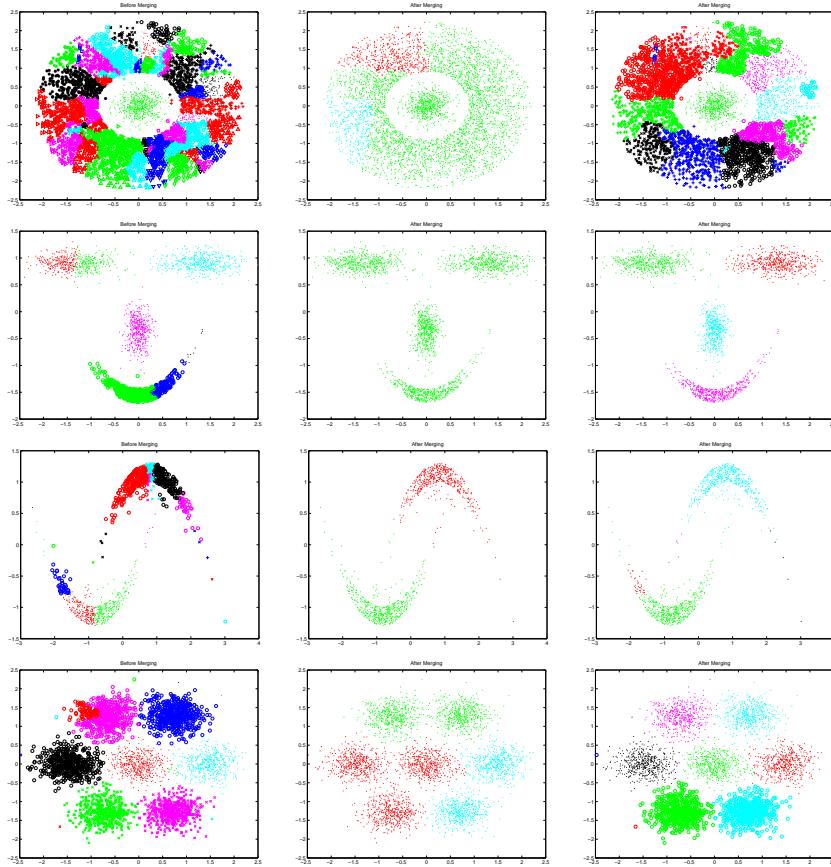


Abbildung A.4.: Turing-Test (v.l.n.r.): Initiale Lösung, Merging nach Ausprägungskriterium, Merging nach Sattelpunkt-Kriterium

Datensatz		initiale Lösung	resultierende Lösung		
Name/Samples/Dimension		$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
bullseye/004560/02		62	0.24	3	0.69
sickels/001050/02		18	0.61	2	0.98
smile/002000/02		7	0.88	1	0.5
wheel/003500/02		16	0.94	3	0.63

Tabelle A.7.: Turing-Test: Ergebnisse des Clustermergings mit dem Ausprägungskriterium. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.

A.3. Clustermerging

Datensatz	initiale Lösung		resultierende Lösung		
	Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
bullseye/004560/02	62	0.24	19	19	0.33
sickels/001050/02	18	0.61	5	5	0.93
smile/002000/02	7	0.88	4	4	1
wheel/003500/02	16	0.94	9	9	0.97

Tabelle A.8.: Turing-Test: Ergebnisse, die mittels Clustermerging Methode nach dem Sattelpunkt-Kriterium erzielt wurden. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.

Datensatz	initiale Lösung		resultierende Lösung		
	Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
synth/2/001000/02	77	0.36	6	6	0.73
synth/2/001000/03	396	0.18	11	11	0.76
synth/2/001000/04	910	0.035	24	24	0.43
synth/2/001000/05	987	0.011	12	12	0.61
synth/3/001000/02	39	0.54	3	3	0.86
synth/3/001000/03	229	0.36	7	7	0.62
synth/3/001000/04	659	0.23	5	5	0.76
synth/3/001000/05	946	0.033	15	15	0.56
synth/4/001000/02	19	0.78	2	2	0.66
synth/4/001000/03	222	0.35	4	4	0.74
synth/4/001000/04	729	0.12	6	6	0.88
synth/4/001000/05	715	0.15	4	4	0.61
synth/5/001000/02	52	0.76	1	1	0.51
synth/5/001000/03	224	0.28	3	3	0.66
synth/5/001000/04	604	0.16	4	4	0.94
synth/5/001000/05	915	0.041	9	9	0.71

Tabelle A.9.: Synthetische Daten: Clustermerging nach dem Ausprägungskriterium. Die initiale Lösung wurde mit der Bandweite $h = 0.25$ generiert.

A. Ergebnisprotokolle

Datensatz		initiale Lösung	resultierende Lösung	
Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
synth/2/001000/02	77	0.36	18	0.54
synth/2/001000/03	396	0.18	34	0.41
synth/2/001000/04	910	0.035	21	0.53
synth/2/001000/05	987	0.011	12	0.69
synth/3/001000/02	39	0.54	10	0.71
synth/3/001000/03	229	0.36	15	0.53
synth/3/001000/04	659	0.23	21	0.7
synth/3/001000/05	946	0.033	15	0.73
synth/4/001000/02	19	0.78	10	0.85
synth/4/001000/03	222	0.35	15	0.56
synth/4/001000/04	729	0.12	20	0.86
synth/4/001000/05	715	0.15	7	0.53
synth/5/001000/02	52	0.76	9	0.83
synth/5/001000/03	224	0.28	26	0.41
synth/5/001000/04	604	0.16	18	0.55
synth/5/001000/05	915	0.041	11	0.51

Tabelle A.10.: Synthetische Daten: Clustermerging mit dem Sattelpunkt-Kriterium.
Die initiale Lösung entstand mit der Bandweite $h = 0.25$.

A.3. Clustermerging

Datensatz		initiale Lösung		resultierende Lösung
Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $	FM-Index
abalone/004177/03	201	0.13	3	0.32
glass/000214/03	59	0.43	2	0.49
hayes-roth/000132/03	109	0.079	7	0.31
iris/000150/03	106	0.18	5	0.43
liver-disorders/000345/03	147	0.19	5	0.63
tic-tac-toe/000958/03	514	0.052	8	0.37
wine/000178/03	147	0.084	7	0.47
yeast/001484/03	370	0.14	7	0.47
abalone/004177/04	708	0.086	7	0.32
glass/000214/04	108	0.3	3	0.45
hayes-roth/000132/04	110	0.099	2	0.45
iris/000150/04	144	0.057	5	0.68
liver-disorders/000345/04	280	0.066	5	0.49
tic-tac-toe/000958/04	822	0.026	16	0.32
wine/000178/04	174	0.018	4	0.53
yeast/001484/04	921	0.059	7	0.43

Tabelle A.11.: Realer Datensatz: Ergebnisse, die vermöge des Clustermergings mit dem Ausprägungskriterium erzielt wurden.

A. Ergebnisprotokolle

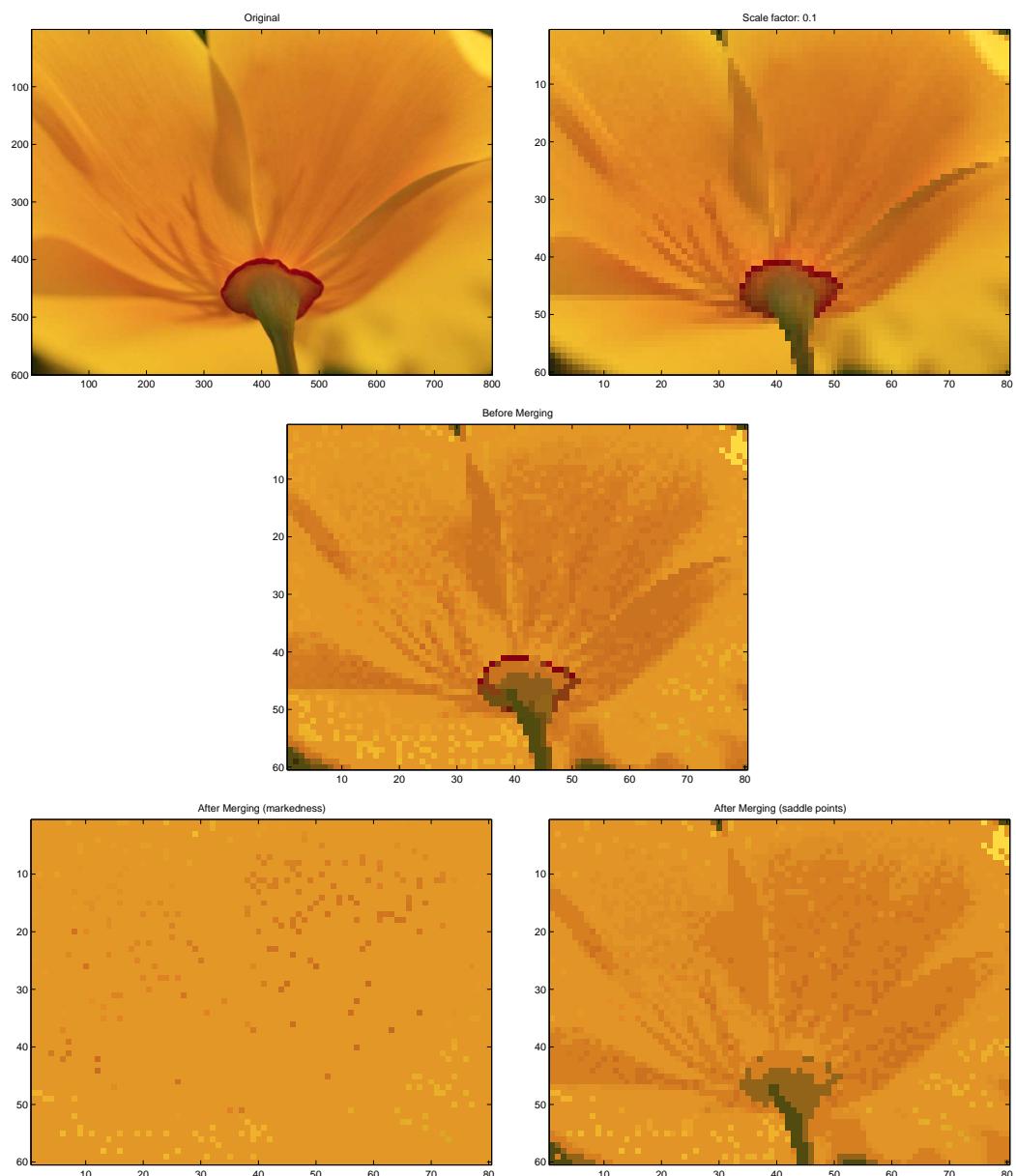
Datensatz	initiale Lösung		resultierende Lösung	
	Name/Samples/Dimension	$ \mathcal{C} $	FM-Index	$ \mathcal{C} $
abalone/004177/03	201	0.13	28	0.17
glass/000214/03	59	0.43	1	0.51
hayes-roth/000132/03	109	0.079	7	0.37
iris/000150/03	106	0.18	6	0.51
liver-disorders/000345/03	147	0.19	11	0.41
tic-tac-toe/000958/03	514	0.052	32	0.36
wine/000178/03	147	0.084	7	0.64
yeast/001484/03	370	0.14	31	0.2
abalone/004177/04	708	0.086	37	0.16
glass/000214/04	108	0.3	1	0.51
hayes-roth/000132/04	110	0.099	6	0.5
iris/000150/04	144	0.057	5	0.54
liver-disorders/000345/04	280	0.067	9	0.39
tic-tac-toe/000958/04	822	0.026	25	0.76
wine/000178/04	174	0.018	4	0.56
yeast/001484/04	922	0.06	30	0.27

Tabelle A.12.: Realer Datensatz: Diese Ergebnisse sind mittels Clustermerging mit Sattelpunkt-Kriterium berechnet.

A.3. Clustermerging



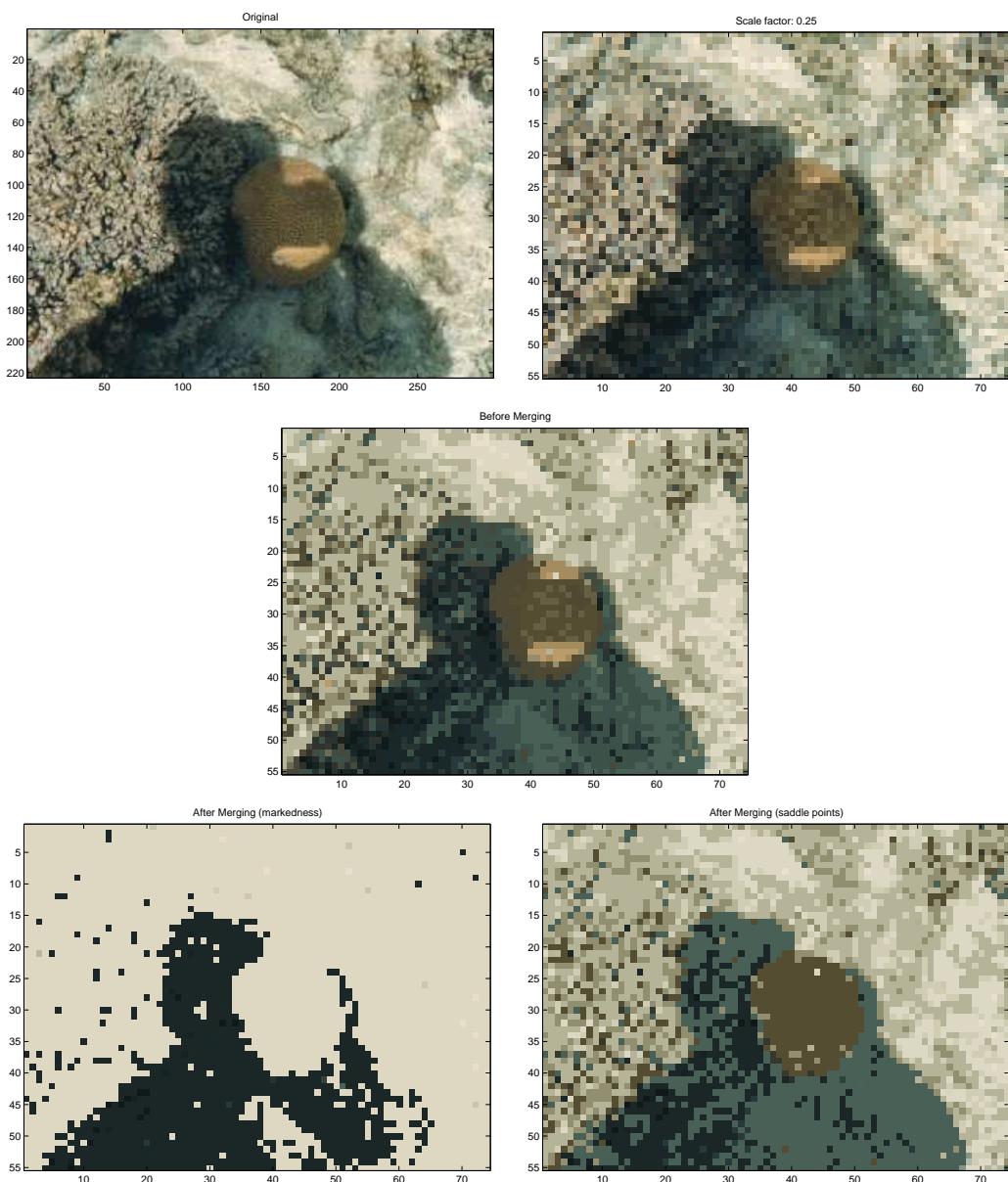
A. Ergebnisprotokolle



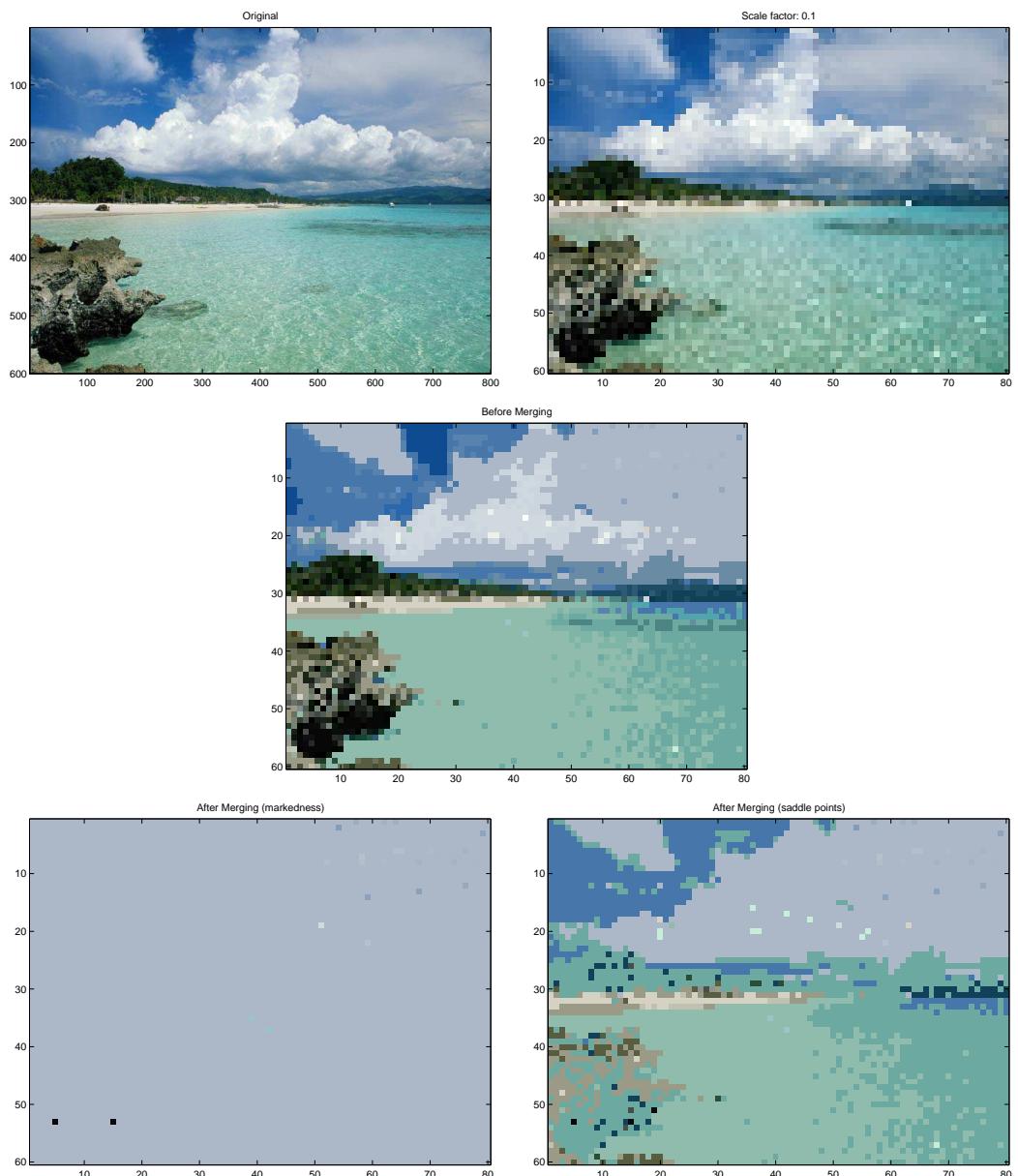
A.3. Clustermerging



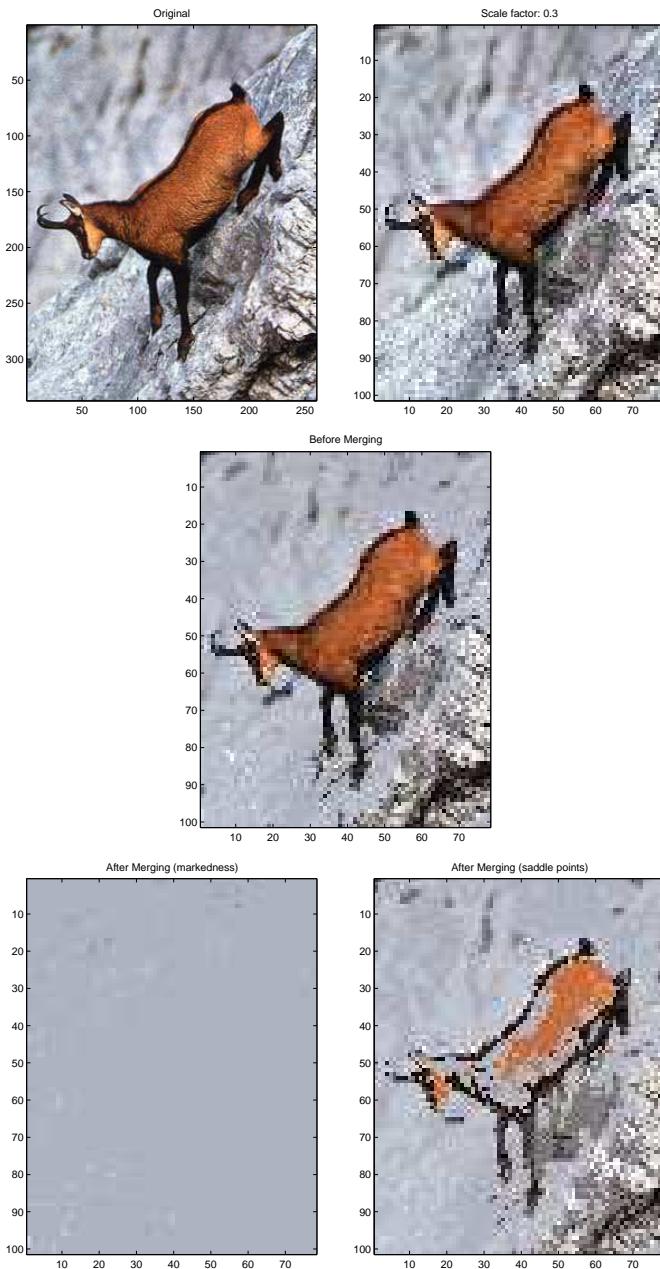
A. Ergebnisprotokolle



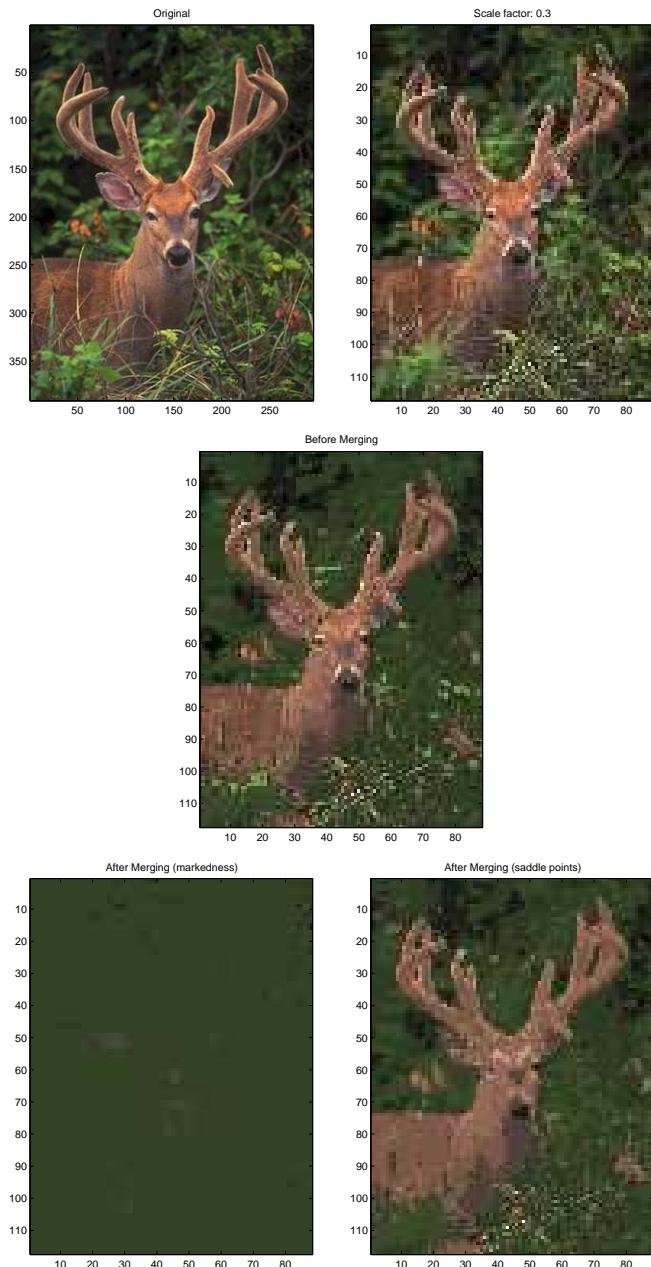
A.3. Clustermerging



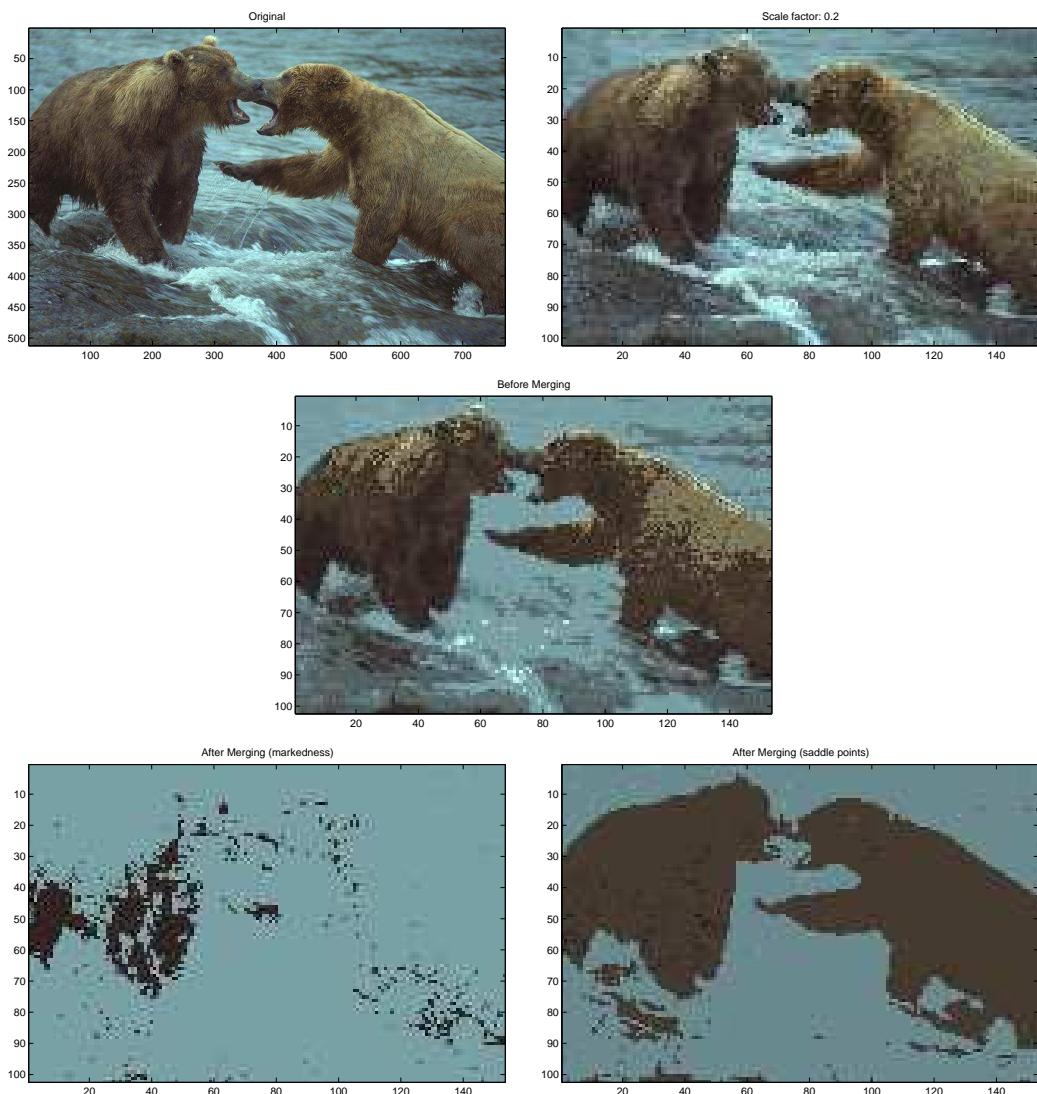
A. Ergebnisprotokolle



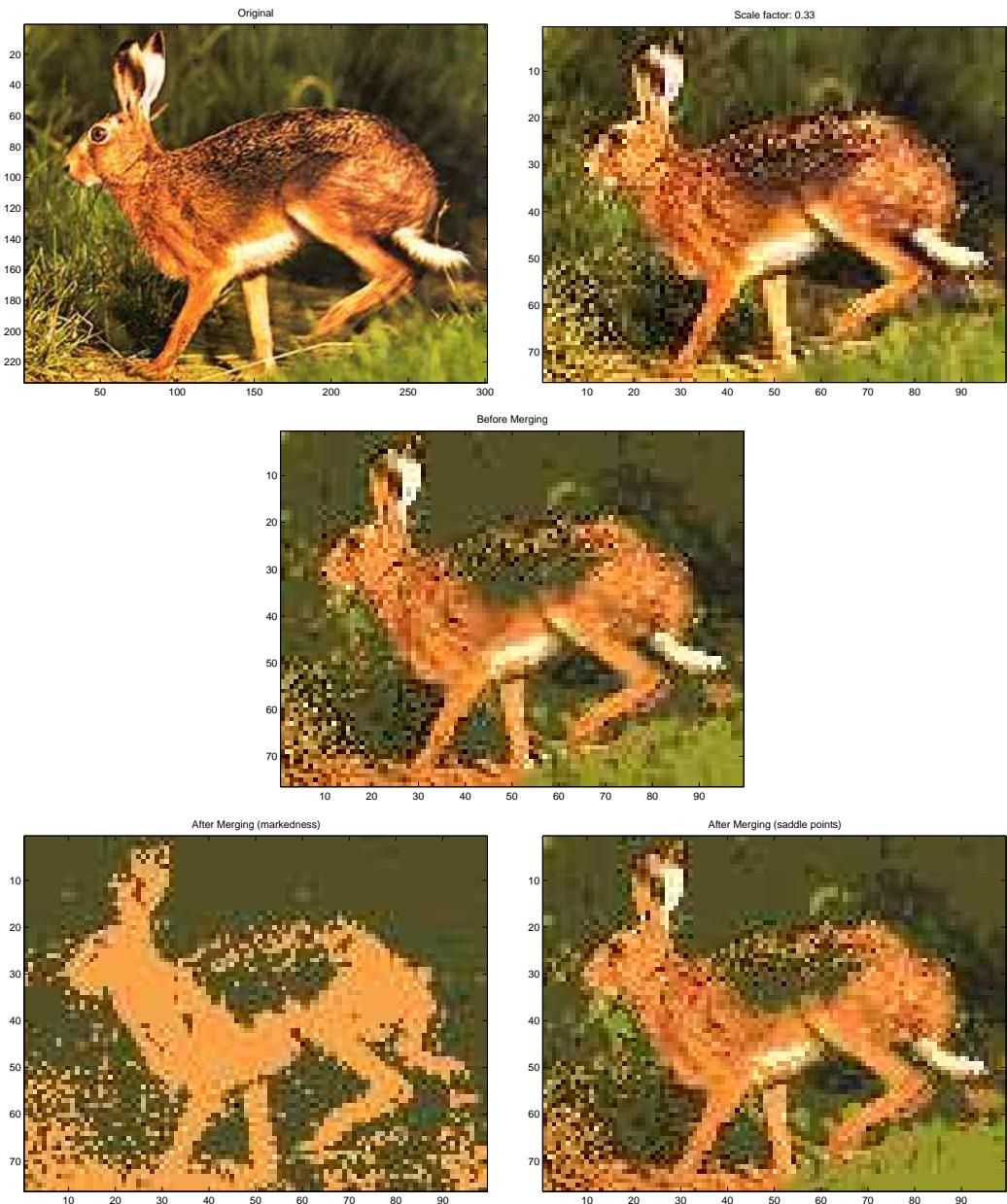
A.3. Clustermerging



A. Ergebnisprotokolle



A.3. Clustermerging



A. Ergebnisprotokolle

A.4. Spritzererkennung

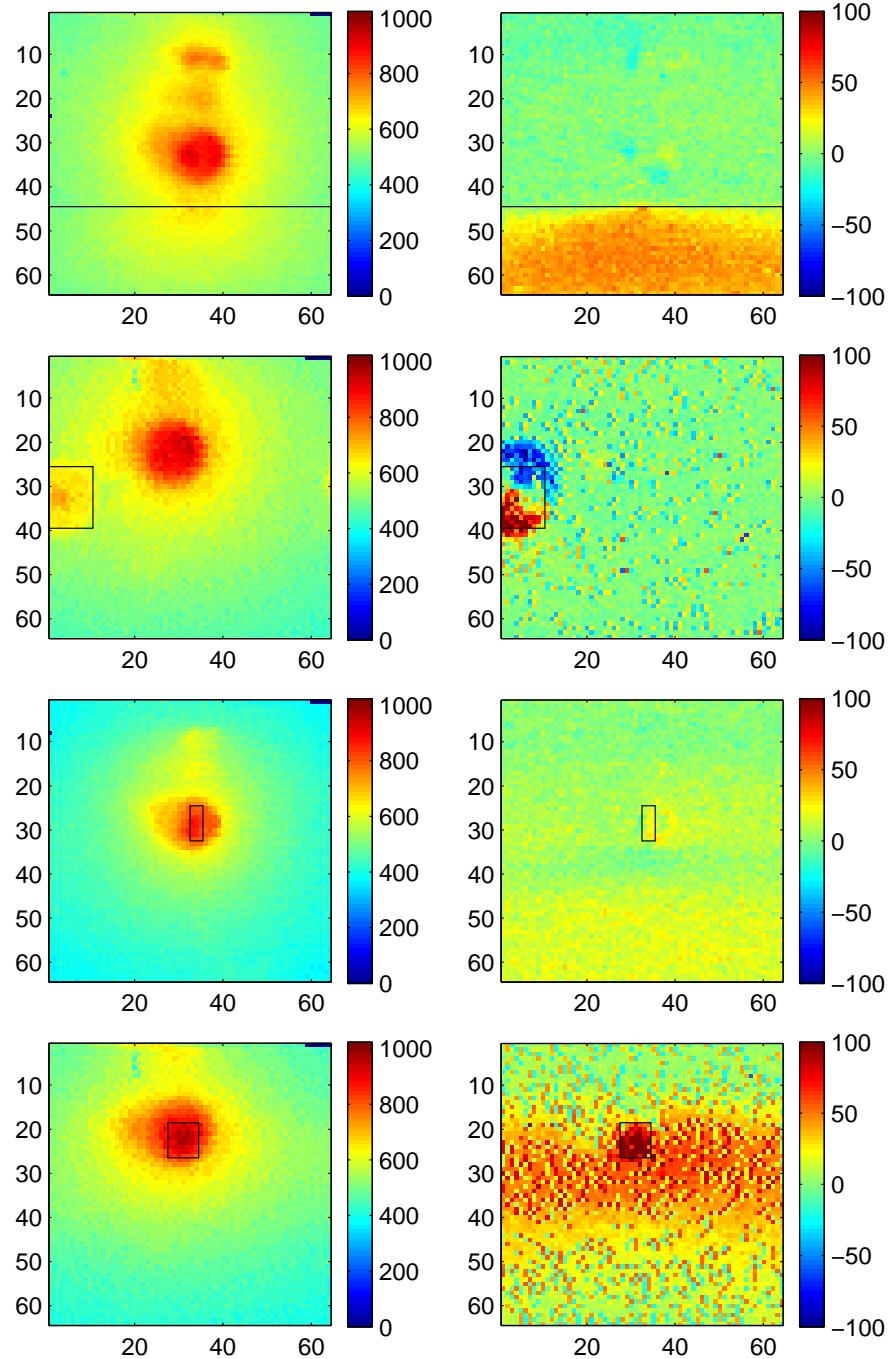


Abbildung A.5.: Links: Originalbild. Rechts: Differenzbild + gekennzeichnetes Objekt (Bounding Box). Das Clustering wurde mittels Silhouetten-Koeffizienten-Bewertungsfunktion bestimmt.

B. Notation

\mathcal{S}	Samples, die Menge aller Punkte eines d diensionalen Raumes
C_i	Cluster i der Zerlegung C
\mathcal{C}	Clustering, Zerlegung der Daten, $\mathcal{C} \equiv \{C_1, \dots, C_k\}$
z_i	Zentrum des Clusters C_i
\mathcal{Z}	Gesamtmenge aller Clusterzentren $\mathcal{Z} \equiv \{z_1, \dots, z_k\}$
ρ	Größe eines Clusterings
$\Delta(i, j)$	Distanz zwischen den Knoten i und j unter der gegebenen Metrik L_q
L_1	Metrik, Manhattan Distanz
L_2	Metrik, Euklid Distanz
L_∞	Metrik, Maximum Distanz
$\mathcal{G} := (V, E)$	Graph G mit der Knotenmenge V und Kanten E
$w(e)$	$w: E \rightarrow \mathbb{R}^{\geq 0}$ Kantengewichtsfunktion
$\mathcal{N}(v)$	Nachbarschaft des Knoten v , $\mathcal{N}(v) \equiv \{w w \in V \wedge (v, w) \in E\}$
f	Dichtefunktion
\hat{f}	Dichteschätzung
K	Kernfunktion
X_i	Beobachtung, Ereignis, Datum
Z_i	Modus der Dichte f
μ	Mittelwert einer Zufallsgröße X
σ^2	Streuung einer Zufallsgröße X
σ	Varianz der Zufallsgröße X , auch $\text{Var}(X)$
\mathcal{X}	Die Menge aller Ereignisse X_i , $\mathcal{X} \equiv \{X_1, \dots, X_n\}$
$I(\cdot)$	Die Indikatorfunktion

Literatur

- Anderberg 1973** ANDERBERG, Michael R.: *Clusteranalysis for Applications*. Academic Press, 1973
- Barber u. a. 1996** BARBER, C. B. ; DOBKIN, D. P. ; HUHDANPAA, H.: The Quickhull algorithm for convex hulls. In: *ACM Trans. Math. Soft.* 22 (1996), Nr. 4, S. 469–483. – URL citeseer.nj.nec.com/barber96quickhull.html
- Bellmann 1961** BELLMANN, R. E.: *Adaptive Control Process*. Princeton University Press. 1961
- Bock 1974** BOCK, H. H.: *Automatische Klassifikation*. Vandenhoeck & Ruprecht, Göttingen, 1974
- Brucker 1977** BRUCKER, P.: On the complexity of clustering problems. In: *Optimization and Operations Research* (1977)
- Cheng 1995** CHENG, Y.: Mean shift, mode seeking and clustering. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (1995), S. 790–799
- Colantoni und Trémeau 2003** COLANTONI, P. ; TRÉMEAU, A.: 3d visualization of color data to analyze color images. In: *The PICS Conference*. Rochester, USA, 2003
- Comaniciu 2003** COMANICIU, Dorin: An algorithm for data-driven bandwidth selection. In: *IEEE transactions on pattern analysis and machine intelligence* 25 (2003), February, Nr. 2, S. 1–8. – URL www-2.cs.cmu.edu/~misc-read/talks-2003/comaniciu-pami-2003.pdf
- Comaniciu und Meer 2002** COMANICIU, Dorin ; MEER, Peter: Mean shift: A robust approach toward feature space analysis. In: *IEEE transactions on pattern analysis and machine intelligence* 24 (2002), May, Nr. 5, S. 603–619. – URL www.cse.msu.edu/~cse902/S03/MeanShift.pdf
- Comaniciu u. a. 2002** COMANICIU, Dorin ; RAMESH, Visvanathan ; BUE, Alessio D.: Multivariate Saddle Detection for Statistical Clustering. In: SPARR, A. H. and G. (Hrsg.) ; NIELSEN, M. (Hrsg.) ; P. JOHANSEN (Hrsg.): *Computer Vision – ECCV 2002* Bd. 2352, Springer-Verlag Heidelberg, 2002, S. 561. – URL www.caip.rutgers.edu/~comanici/Papers/ImageSegmentationClustering.pdf

- Domeniconi und Gunopulos 2001** DOMENICONI, C. ; GUNOPULOS, D.: An efficient approach for approximating multi-dimensional range queries and nearest neighbor classification in large datasets. In: *Proc. 18th International Conf. on Machine Learning*. San Francisco : Morgan Kaufmann, 2001, S. 98–105. – URL citeseer.nj.nec.com/domeniconi01efficient.html
- Drezner 1984** DREZNER, Z.: The p-centres problems — Heuristic and optimal algorithms. In: *J. Oper. Res. Soc* 35 (1984), S. 741–748
- Duda u. a. 2001** DUDA, Richard O. ; HART, Peter E. ; STORK, David G.: *Pattern Classification*. Second. John Wiley & Sons, Inc., 2001
- Duin 1976** DUIN, R. P. W.: On the choice of smoothing parameters for Parzen estimators of probability density functions. In: *IEEE Trans. Comp.* C-25 (1976)
- Dyer 1986** DYER, M. E.: On a multidimensional search technique and its application to the Euclidean one-centre problem. In: *SIAM J. Comput.* 15 (1986), S. 725–738
- Feder und Greene 1988** FEDER, T. ; GREENE, D. H.: Optimal algorithms for approximate clustering. In: *Proc. 20th Annu. AC; Sympos. Theory Comput.*, 1988, S. 434–444
- Fix und Hodges 1951** FIX, E. ; HODGES, J. L.: Discriminatory analysis, nonparametric estimation: consistency properties / USAF School of Aviation Medicine. 1951 (Report No. 4, Project no. 21-49-004). – Forschungsbericht
- Fowler u. a. 1981** FOWLER, R. J. ; PATERSON, M. S. ; TANIMOTO, S. L.: Optimal packing and covering in the plane are NP-complete. In: *Inform. Process. Lett.* 12 (1981), S. 133–137
- Fowlkes und Mallows 1983** FOWLKES, E. B. ; MALLOWS, C. L.: A method for comparing two hierarchical clusterings. In: *Journal of the American Statistical Association* (1983), Nr. 42, S. 578–588
- Fukunaga 1990** FUKUNAGA, Keinosuke: *Introduction to statistical pattern recognition*. Second. Academic Press, 1990
- Gabler und Borg 1996** GABLER ; BORG: Unimodalität und Unimodalitätstests. In: *ZUMA-Nachrichten*, 1996, S. 33–44
- Golub u. a. 1999** GOLUB, T. R. ; SOLNIM, D. K. ; TAMAYO, P. ; HUARD, C.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression. In: *Science* 286 (1999). – URL www-genome.wi.mit.edu/mpr/publications/projects/Leukemia/Golub_et_al_1999.pdf
- Gonzalez 1985** GONZALEZ, T.: Clustering to minimize the maximum intercluster distance. In: *Theoret. Comput. Sci.* 23 (1985), S. 293–306

Literatur

- Hader und Hamprecht 2003** HADER, Sören ; HAMPRECHT, Fred A.: Efficient Density Clustering Using Basin Spanning Trees. In: SCHÄDER, M. (Hrsg.) ; GAUL, W. (Hrsg.) ; VICHI, M. (Hrsg.): *Between Data Science and Applied Data Analysis*, Springer, 2003 (Studies in Classification, Data Analysis and Knowledge Organization), S. 39–48. – URL klimt.iwr.uni-heidelberg.de/mip/fhamprecht/efficient-density-clustering.pdf
- Hinneburg und Keim 1998** HINNEBURG, Alexander ; KEIM, Daniel A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In: *Knowledge Discovery and Data Mining*, URL citeseer.nj.nec.com/hinneburg98efficient.html, 1998, S. 58–65
- Hochbaum und Shmoys 1985** HOCHBAUM, D. S. ; SHMOYS, D.: A best possible heuristic for the k -center problem. In: *Math. Oper. Res.* 10 (1985), S. 180–184
- Hochbaum und Shmoys 1986** HOCHBAUM, D. S. ; SHMOYS, D.: A unified approach to approximation algorithms for bottleneck problems. In: *J. ACM* 33 (1986), S. 533–550
- Hsu und Nemhauser 1979** HSU, W. L. ; NEMHAUSER, G. L.: Easy and hard bottleneck location problems. In: *Discr. Appl. Math.* 1 (1979), S. 209–215
- Hwang u. a. 1993** HWANG, R. Z. ; LEE, R. C. T. ; CHANG, R. C.: The slab dividing approach to solve the Euclidean p -center problem. In: *Algorithmica* 1 (1993), Nr. 1-22
- Jain und Dubes 1988** JAIN, A. K. ; DUBES, R. C.: *Algorithms for Clustering Data*. Prentice Hall, 1988
- Jain u. a. 1999** JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Computing Surveys* 31 (1999), Nr. 3, S. 264–323. – URL citeseer.nj.nec.com/jain99data.html
- Kauffmann und Rousseeuw 1990** KAUFFMANN, L. ; ROUSSEEUW, P.: *Finding Groups in Data: An Introduction to Cluster Analysis*. J. Wiley & Sons, 1990
- Ko u. a. 1990** KO, M. T. ; LEE, R. C. ; CHANG, J. S.: An optimal approximation algorithm for the rectilinear m -center problem. In: *Algorithmica* 5 (1990), S. 341–352
- Kowalewski 1995** KOWALEWSKI, F.: A gradient procedure for determining clusters of relatively high point density. In: *Pattern Recognition* 28 (1995), S. 1973–1984
- Krengel 2000** KRENGEL, Ulrich: *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. 5 Auflage. Vieweg, 2000

- Lachenbruch 1968** LACHENBRUCH, P. A.: On expected probabilities of misclassification in discriminant analysis, necessary sample size, and a relation with the multiple correlation coefficient. In: *Biometetrics* (1968)
- Megiddo 1983** MEGIDDO, N.: Linear-time algorithms for linear programming in R^3 and related problems. In: *SIAM J. Comput.* 12 (1983), Nr. 759–776
- Megiddo und Supowit 1983** MEGIDDO, N. ; SUPOWIT, K. J.: On the complexity of some common geometric location problems. In: *SIAM J. Comput.* 12 (1983), S. 182–196
- Miguel Á. Carreira-Perpiñán 1997** MIGUEL Á. CARREIRA-PERPIÑÁN: A Review of Dimension Reduction Techniques / Dept. of Computer Science, University of Sheffield. URL citeseer.nj.nec.com/126333.html, January 1997 (CS-96-09). – Forschungsbericht
- Nussbaum 1997** NUSSBAUM, D.: Rectilinear p -piercing problems. In: *Proceedings of the Annual International Symposium on Symbolic and Algebraic Computation*, 1997
- Parzen 1962** PARZEN, E.: On estimation of a probability density function and mode. In: *Ann. Math Statist.* 33 (1962)
- Pauwels und Frederix 1999** PAUWELS, E. J. ; FREDERIX, G.: Finding Salient Regions in Images. In: *Computer Vision and Image Understanding* 85 (1999), S. 73–85. – URL citeseer.nj.nec.com/pauwels98finding.html
- Procopiuc** PROCOPIUC, Cecilia M.: *Clustering Problems and their Applications*. – URL <http://www.cs.duke.edu/~magda/clustering-survey.ps.gz>
- Rao 1983** RAO, Prakasa: *Nonparametric Functional Estimation*. New York: Academic Press, 1983
- Rudemo 1982** RUDEMO, M.: Empirical choice of histograms and kernel density estimators. In: *Scand. J. Statist.* 9 (1982)
- Scott und Thompson 1983** SCOTT, D. W. ; THOMPSON, J. R.: Probability density estimation in higher dimesion. In: GENTLE, J. E. (Hrsg.): *Proceedings of the Fifteenth Symposium on the Interface*. Amsterdam, New York, Oxford : North Holland-Elsevier Science Publishers, 1983, S. 173–179
- Sharir und Welzl 1996** SHARIR, M. ; WELZL, E.: Rectilinear and polygonal p -piercing and p -center problems. In: *Proc. 12th Annu. AC; Sympos. Comput. Geom.*, URL citeseer.nj.nec.com/sharir96rectilinear.html, 1996, S. 122–132
- Silverman 1986** SILVERMAN, B. W.: *Density Estimation for Statistics and Data Analysis*. 2. 29 West 35th Street, New York NY 1001 : Chapman and Hall, 1986

Literatur

Überhuber und Katzenbeisser 2002 ÜBERHUBER, Christoph ; KATZENBEISER, Stefan: MATLAB 6.5. Eine Einführung. Springer-Verlag/Wien, 2002

Ward 1963 WARD, J. H.: Hierarchical groupings to optimize an objective function. In: *Journal of the American Statistical Association* 59 (1963), S. 234–244

Wei u. a. 2001 WEI, Javed Kahnand Jun S. ; RINGNER, Markus ; SAAL, Lao H. ; LADANYI, Marc ; WESTERMANN, Frank ; BERTHOLD, Frank ; SCHWAB, Manfred ; ANTONESCU, Christina R. ; PETERSON, Carsten ; MELTZER, Paul S.: Classification and diagnostic prediction of cancer using gene expression profiling and artificial neural networks. In: *Nature Medicine* (2001). – URL www.theplu.se/pub/Preprints/01/lu_tp_01_06.pdf

Erklärung

Hiermit versichere ich, dass diese Diplomarbeit eigenständig und ohne fremde Hilfe angefertigt wurde. Ebenso wurden keine anderen, als die angegebenen Quellen und Hilfsmittel, die zitiert sind, benutzt.

*Paderborn, den 13. September 2004
Eduard Wiebe*
