Moderne Clusteralgorithmen – eine vergleichende Analyse auf zweidimensionalen Daten

Marcus Josiger, Kathrin Kirchner
Friedrich–Schiller–Universität Jena
07743 Jena
m.josiger@gmx.de, k.kirchner@wiwi.uni-jena.de

Abstract

Ziel der vorliegenden Arbeit ist die Analyse ausgewählter moderner Clusteralgorithmen. Als modern werden dabei Algorithmen angesehen, die sich durch neue bzw. durch Kombination traditioneller Vorgehensweisen auszeichnen. Anhand von Testdatensätzen werden Eigenschaften, Leistungsfähigkeit sowie Stärken und Schwächen dieser Algorithmen auf zweidimensionalen Punktmengen evaluiert.

1 Einführung

Die Clusteranalyse ist eine Data Mining Methode, mit der nach natürlich auftretenden Gruppen im Datenbestand gesucht wird. Ziel dabei ist, möglichst ähnliche Objekte in einem Cluster zusammenzufassen, so dass die Ähnlichkeit der Objekte innerhalb eines Clusters maximiert und zwischen den Clustern mininmiert wird [Chen, 2001]. Oftmals wird die Clusteranalyse als Vorstufe zur Datenverdichtung für weitere Analysen verwendet. Im Data Mining existieren zahllose Clusteralgorithmen. Nach [Han und Kamber, 2001] können diese in partitionierende, hierarchische, dichtebasierende, gitterbasierende und modellbasierende Methoden eingeteilt werden.

Gegenstand der Untersuchung bildet die Qualität solcher Clusteralgorithmen auf zweidimensionalen Daten. Darunter werden hier Punkte in der Ebene verstanden, zwischen denen ein euklidisches Abstandsmaß definiert werden kann.

Für die Untersuchung wurden drei modernere Clusteralgorithmen ausgewählt und in Bezug auf die Clusterqualität bei verschieden gestalteten Punktmengen und bei Rauschen untersucht. Die ausgewählten Algorithmen sollten sich dabei von traditionellen Algorithmen, wie z.B. dem seit den sechziger Jahren bekannten k–means, abgrenzen. Diese Bedingung wurde als erfüllt angesehen, wenn die Algorithmen eine neuartige Vorgehensweise oder eine Kombination vorhandener Methoden verwendeten. In den Vergleich wurden endgültig eine Auswahl von Algorithmen einbezogen, die auf Grund ihrer Leistungskriterien in der Literatur gelobt wurden und unterschiedlichen Grundlogiken folgen.

Diese Arbeit ist wir folgt aufgebaut: Abschnitt 2 beschreibt zunächst kurz die für die Untersuchung ausgewählten Algorithmen, während Abschnitt 3 die einzelnen Untersuchungen und deren Ergebnisse erläutert. In Abschnitt 4 werden die Untersuchungsergebnisse ausgewertet. Abschnitt 5 enthält eine Zusammenfassung.

2 Ausgewählte Clusteralgorithmen

2.1 BIRCH

BIRCH[1] ist ein Vertreter der hierarchischen Clustermethoden. Es wird eine hierarchische Datenstruktur, der Clustering–Feature–Tree (CF–Tree, [Zhang et. al, 1996]) verwendet. Dadurch können multidimensionale Objekte inkrementell und dynamisch zu Clustern zusammengefaßt werden. Dabei läuft BIRCH in zwei Phasen ab. In einer ersten Phase werden zunächst alle Daten durchsucht und ein CF–Tree aufgebaut. Somit befinden sich nun alle Clusterinformationen im Hauptspeicher des Rechners. In der zweiten Phase wird die eigentliche Clusteranalyse auf Grundlage des CF–Trees durchgeführt. Dabei ist die Anzahl k der gewünschten Cluster ein vom Anwender festzulegender Parameter.

2.2 DBSCAN

DBSCAN[2] [Ester et. al, 1996] ist ein dichtebasierender Algorithmus. Dabei wird davon ausgegangen, dass die Dichte von Objekten innerhalb eines Clusters höher ist als außerhalb desselben. Grundbedingung zur Bildung und zum Wachsen eines Clusters ist, dass eine Mindestanzahl von Objekten (MinPts) innerhalb eines definierten Bereiches (der ϵ–Umgebung) vorhanden sein muss.

Die Parameter ϵ und MinPts werden entsprechend der Aufgabenstellung vom Anwender festgelegt. Je nach Festlegung können sich die gefundenen Cluster extrem unterscheiden. Für die Parameterbestimmung wird daher auf eine effektive heuristische Methode, den sortierten k–Distanz-Graphen, zurückgegriffen. Grundgedanke dabei ist es, eine „Grenzdichte“ zu bestimmen, anhand derer der am dünnsten besiedelte Cluster bestimmt wird. Die Objekte, die diesen Grenzwert in ihrer Umgebung nicht erreichen, werden als Rauschen identifiziert. Dazu wird eine Funktion k–Distanz ($k \geq 1$) definiert, mit der der Abstand jedes Objekts zu seinem k–nächsten Nachbarn bestimmt wird. Diese Abstände werden dann in einem Graph dargestellt, der Aufschluss über die Dichteverteilung innerhalb des Datenbestandes gibt. Dies ermöglicht eine einfachere Festlegung der benötigten Parameter für DBSCAN.

2.3 CHAMELEON*

Aus der Data Mining Anwendung CLUTO[3] wurde ein dem CHAMELEON–Algorithmus [Karypis et. al, 1999] nach-

[1] Balanced Iterative Reducing and Clustering using Hierarchies

[2] Density Based Spatial Clustering of Applications with Noise

[3] Clustering Toolkit – Data Mining Softwarepaket

empfunder Algorithmus ausgewählt[4]. Die Clusterung läuft bei diesem Algorithmus in zwei Phasen ab. Zunächst wird dabei in der ersten Phase ein Graph–partitionierendes Verfahren zur Vorverdichtung eingesetzt, anschließend ein agglomeratives hierarchisches Verfahren.

In der ersten Phase wird dabei ein nächster–Nachbar–Graph erstellt, wobei jedes Objekt mit den Objekten im Datenraum verbunden wird, die diesem am ähnlichsten sind. Dieser Graph wird dann in M Cluster geteilt, wobei M>K gilt. K ist dabei die vom Anwender endgültig gewünschte Clusteranzahl. In der zweiten Phase wird ein agglomeratives hierarchisches Clusterverfahren eingesetzt, wobei die M Cluster verschmolzen werden, bis die gewünschte Clusteranzahl K entstanden ist. M und K werden dabei als Parameter vom Anwender vorgegeben.

3 Analyse der Clusteralgorithmen

Untersuchungen ausgewählter Clusteralgorithmen finden sich beispielsweise in [Kolatch, 2001]. Hier wird ein Benchmark für Clusteralgorithmen auf räumlichen Daten vorgestellt. Anhand dieses Benchmarks und der vorliegenden Daten werden für die ausgewählten Clusteralgorithmen nachfolgende Kriterien überprüft:

1. Identifikation von Clustern einfacher Gestalt
2. Identifikation von Clustern nicht–sphärischer Gestalt
3. Umgang mit verrauschten Daten
4. Festlegung von Parametern durch den Anwender
5. Effizienz der Behandlung großer Datenmengen (auf Grund der verwendeten Daten nur eingeschränkt prüfbar)

3.1 Punktmengen einfacher Gestalt

Für Untersuchung der Clusterung von Punktmengen einfacher Gestalt wurden drei Testdatensätze verwendet, die kreis– und ellipsenförmige Gestalt haben. Datensatz 1 (siehe Abbildung 1, links) besteht aus 300 Objekten und ist dadurch gekennzeichnet, dass die zu ermittelnden Cluster aus einer ähnlichen Anzahl an Objekten bestehen, die die gleiche Dichte aufweisen. Hierbei soll überprüft werden, ob die Algorithmen in der Lage sind, die Cluster richtig voneinander abzugrenzen. Beim Datensatz 2 (Abbildung

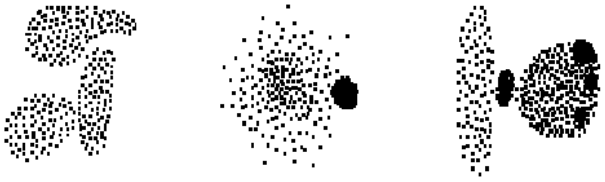

Abbildung 1: Testdatensätze 1–3 zur Untersuchung der Clusterung einfacher Punktmengen

1 Mitte) sollen zwei Cluster unterschiedlicher Größe und Dichte identifiziert werden. Er enthält ebenfalls 300 Objekte. Der dritte Datensatz (Abbildung 1 rechts) umfasst 630 Objekte und enthält sechs Cluster, die sich in Form, Größe und Dichte unterscheiden. Eine Besonderheit stellen dabei die drei kleinen Cluster rechts dar, da diese sich innerhalb eines größeren, weniger dichten Clusters befinden. Die Datensätze 2 und 3 wurden der Arbeit von [Ertöz et. al., 2003] entnommen.

Ergebnisse von BIRCH

Abbildung 2 zeigt die Ergebnisse von BIRCH auf den Testdatensätzen 1–3. Bei Datensatz 1 (Abbildung 2 links) konn-

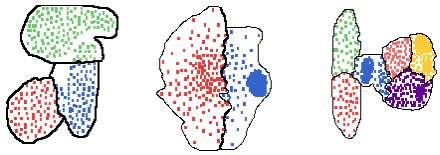

Abbildung 2: Ergebnisse von BIRCH für Punktmengen einfacher Gestalt

te keine korrekte Abgrenzung zwischen den drei Clustern gefunden werden. Nur die Objekte, die zum obersten Cluster gehören, wurden korrekt zugewiesen, mehrere Objekte des vertikalen Clusters wurden jedoch fälschlicherweise ebenfalls zugeordnet. Ähnlich verhält es sich mit dem runden Cluster links unten.

Bei Datensatz 2 (Abbildung 2, Mitte) wurden die Cluster ebenfalls nicht korrekt voneinander abgegrenzt. Objekte des großen Clusters mit geringer Dichte wurden dem kleineren Cluster mit hoher Dichte zugeordnet. Abbildung 2 (rechts) zeigt, dass für den dritten Datensatz ebenfalls kein korrektes Ergebnis erzielt werden konnte. Es wurden zwar sechs Cluster gefunden, jedoch keines der angedachten Cluster wurde korrekt erkannt. Die beiden großen Cluster mit geringer Dichte wurden in mehrere kleine Cluster aufgeteilt.

Ergebnisse von DBSCAN

Die Qualiät der Cluster hängt von der Wahl der Eingabeparameter ϵ und MinPts ab. Wird mit den Werten für ϵ=2,368 und MinPts=3[5] gearbeitet, ergibt sich das in Abbildung 3 links dargestellte Ergebnis. Lediglich der obere Cluster wurde hier korrekt identifiziert. Durch Veränderung der Eingabeparameter auf ϵ=3,7462 und MinPts=10 konnte die Clusterlösung verbessert werden (ähnliches Ergebnis wie bei CHAMELEON*, Abbildung 4 links). Bei Testdaten-

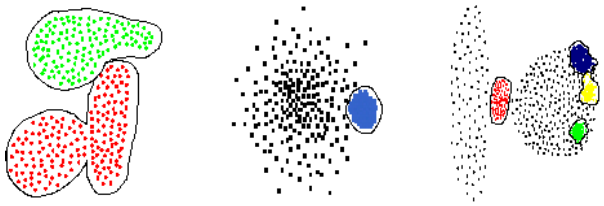

Abbildung 3: Ergebnisse von DBSCAN für Punktmengen einfacher Gestalt

satz 2 (Abbildung 3 Mitte) wurden alle zum kleinen Cluster gehörenden Punkte korrekt zugeordnet (bei ϵ=1,3007 und MinPts=3). Alle anderen Objekte werden dabei jedoch als Rauschen interpretiert (schwarze Punkte). Bei Testdatensatz 3 (Abbildung 3 rechts) wurden vier von sechs Clustern korrekt erkannt (ϵ=1,5775, MinPts=3). Die verbleibenden

[4]Der Code von CHAMELEON ist gegenwärtig nicht öffentlich verfügbar, hier wird zur Unterscheidung die Bezeichnung CHAMELEON* gewählt

[5]Die Werte für ϵ wurden jeweils mit Unterstützung des k–Distanz–Graphen bestimmt, der Wert für MinPts wird in [Sander, 1998] mit 2*d mit d Anzahl der Dimensionen vorgeschlagen, was jedoch nicht immer günstig war.

Punkte wurden als Rauschen identifiziert (schwarz dargestellt).

Ergebnisse von CHAMELEON*
CHAMELEON* lieferte bei allen Testdatensätzen gute Ergebnisse (siehe Abbildung 4). In den Grenzbereichen der Cluster höherer und geringerer Dichten wurden allerdings wiederholt einzelne Objekte Clustern höhrer Dichte zugeordnet. Beim ersten Datensatz wurden fünf Objekte falsch zugeordnet, bei Datensatz 2 waren es vier.

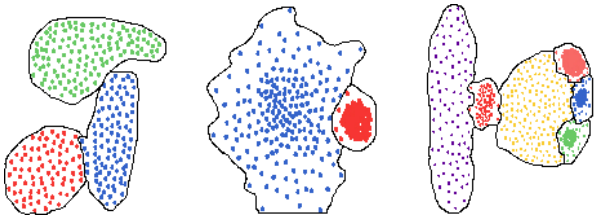

Abbildung 4: Ergebnisse von CHAMELEON* für Punktmengen einfacher Gestalt

3.2 Punktmengen nicht–sphärischer Gestalt

Neben den Clustern einfacher Gestalt werden hier Punktmengen untersucht, die als schwierig zu clustern gelten. Die gewählten Formen sind dabei abstrakt. Die zu untersuchenden Datensätze sind in Abbildung 5 dargestellt. Die Besonderheit des Testdatensatzes 4 (links) besteht darin, dass sich ein kleinerer Cluster innerhalb eines größeren befindet. Testdatensatz 5 (rechts) hat die Form eines Krähenfußes.

Abbildung 5: Testdatensätze 4 und 5 zur Untersuchung von Punktmengen nicht–sphärischer Gestalt

Ergebnisse von BIRCH
Die Clusterbildung bei Testdatensatz 4 (vergleiche Abbildung 6 links) ist mit einem Schnitt durch die Mitte des Datenraumes vergleichbar, wobei jede Hälfte als ein Cluster identifiziert wurde. Bei Datensatz 5 (Abbildung 6 rechts) wurde ausschließlich die untere Hälfte des Ypsilon als ein Cluster entdeckt.

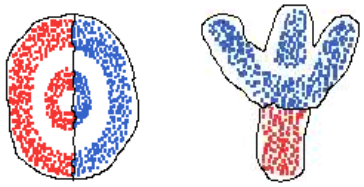

Abbildung 6: Ergebnisse von BIRCH für Punktmengen nicht–sphärischer Gestalt

Ergebnisse von DBSCAN und CHAMELEON*
DBSCAN und CHAMELEON* waren in der Lage, die Clusterung korrekt durchzuführen.

3.3 Verrauschte Daten

Um die untersuchten Algorithmen auf ihre Clusterqualität prüfen zu können, wurden zum Testdatensatz 2 (vergleiche Abbildung 1, Mitte) zufällig weitere Objekte als Ausreißer bzw. Rauschen hinzugefügt. Sie gehören nicht zum Cluster und können z.B. natürlich auftretende Ausreißer im Datenbestand repräsentieren oder durch fehlerhafte Datenaufnahme entstanden sein. Dabei soll untersucht werden, inwieweit die Clusteranalyse durch diese Objekte beeinträchtigt wird. Abbildung 7 zeigt die zu Testdatensatz 2 hinzugefügten 1% und 40% Rauschanteil. Nach

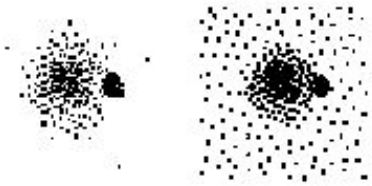

Abbildung 7: Testdatensatz 6 zur Untersuchung der Clusterbildung bei Rauschen (Testdatensatz 2 jeweils mit 1% und 40% Rauschanteil)

Durchführung der Clusteranalyse mit diesen Daten werden die entdeckten Cluster mit dem Originalergebnis verglichen. Je weniger Einfluss das Rauschen auf das Clusterergebnis hat, desto besser ist der Algorithmus zu beurteilen. Rauschen besitzt keinen Einfluss auf die Analyse, wenn die hinzugefügten Ausreißer entweder als solche gekennzeichnet werden oder diese einen oder mehrere selbständige Cluster bilden. Wenn ein Ausreißer dem Originalcluster hinzugefügt wird, liegt eine Fehlklassifikation vor.

Ergebnisse von BIRCH
Bei einprozentigem Rauschen erkannte BIRCH zwei Cluster (Abbildung 8, links). Der Ausreißer auf der linken Seite wurde zusammen mit einigen Punkten des weniger dichten Clusters zu einem Cluster zusammengefaßt. Die übrigen Punkte bilden den zweiten Cluster. Ein Vergleich dieses

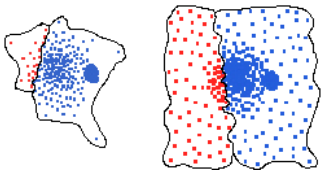

Abbildung 8: Ergebnisse von BIRCH für verrauschte Daten

Ergebnisses mit dem Ergebnis bei diesen Daten ohne Rauschen (Abbildung 2, Mitte) zeigt, dass bei Rauschen weniger Punkte zum linken Cluster zugeordnet wurden. Weiterhin ist der gedachte Schnitt zwischen den Clustern nach links verschoben. Ein Grund dafür könnte sein, dass der linke Ausreißer den Mittelpunkt des linken Clusters verlagert hat und dies in der Verschiebung der Clustergrenzen resultierte. Ähnliches ist bei 40–prozentigem Rauschen (Abbildung 8) zu beobachten.

Ergebnisse von DBSCAN
Die Clusteranalyse wurde hier mit MinPts=3 und und zwei verschiedenen Werten für ϵ durchgeführt. Bei ϵ=1,3007 hat die Erhöhung des Rauschanteils keinerlei Auswirkungen

auf den kleinen Originalcluster (Abbildung 9 erstes und zweites Bild von links). Bis zu einem Rauschanteil von 40% verändert sich die Clusterung nicht.

Bei einem ϵ–Wert von 7,365 wird für 1%-iges Rauschen jedes neue Objekt als Rauschen (schwarze Objekte in Abbildung 9 drittes Bild von links) identifiziert. Bei einem 40%-igen Rauschanteil wurden 13 Cluster gebildet (Abbildung 9 ganz rechts). Dies waren ein großer Cluster (317 Objekte) und 12 kleine Cluster, die zwischen 3 und 10 Objekten enthielten. Grund für die Bildung dieser kleineren Cluster ist, dass die hinzugefügten Ausreißer zufällig die Bedingung MinPts=3 und ϵ=7,365 erfüllten.

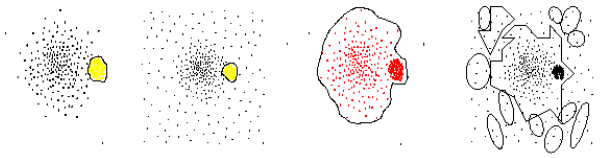

Abbildung 9: Ergebnisse von DBSCAN für verrauschte Daten

Ergebnisse von CHAMELEON*

Bei einprozentigem Rauschen konnten alle Ausreißer als solche identifiziert werden (Abbildung 10 links, Ausreißer schwarz gekennzeichnet). Bei 40–prozentigem Rauschen wurden drei Clus/-ter gefunden (Abbildung 10, rechts). Dabei wurden das Originalcluster geteilt. Zwei Objekte wurden als Ausreißer gekennzeichnet (schwarze Punkte), alle übrigen Objekte wurden einem großen Cluster zugeordnet.

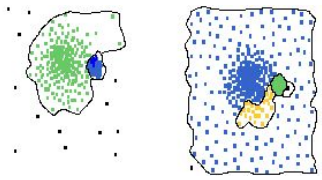

Abbildung 10: Ergebnisse von CHAMELEON* für verrauschte Daten

4 Auswertung der Ergebnisse

BIRCH

Die von BIRCH generierten Cluster einfacher und nicht–sphärischer Gestalt entsprachen nicht den natürlichen Clustern, die durch die Visualisierungen als vermeintlich offensichtlich galten.

Der Grund dafür ist in der Vorgehensweise von BIRCH zu suchen. Nach Erstellung des CF–Baumes wird ein zentroid–basiertes Verfahren zur Clusterbildung genutzt. Zunächst werden die Abstände zwischen den Zentroiden zweier Cluster berechnet. Dann werden die am weitesten entfernten Objekte im CF–Baum als Startpunkte definiert. Von diesen Punkten ausgehend kommt es dann zur schrittweisen Zuordnung der Objekte zum nächstgelegenen Startpunkt. Dies erklärt die seltsame Clusterbildung der zwei Kreisringe im Datensatz 4 (vergleiche Abbildung 6, links). Auch wenn die Clusterresultate inkorrekt scheinen, so können sie unter Betrachtung der Vorgehensweise von BIRCH durchaus dem Verständnis eines Clusters als Repräsentant von Objekten möglichst großer Ähnlichkeit entsprechen.

Weiterhin fällt auf, dass die von BIRCH generierten Cluster stets eine ähnliche Anzahl von Objekten beinhalteten. Der Grund dafür ist in der ersten Phase des Clusterprozesses, der Erstellung des CF–Baumes, zu suchen. In jedem Blattknoten des Baumes darf es nur eine maximale Zahl an Einträgen geben. Wenn diese begrenzte Anzahl durch die Größe eines natürlichen Clusters überschritten wurde, werden diese auf mehrere Blattknoten verteilt. Dies kommt einer Splittung der natürlichen Cluster gleich.

Bei den Untersuchungen im Rahmen der vorliegenden Arbeit konnte BIRCH auch bei verrauschten Daten nicht überzeugen. Es ist möglich, schon beim Aufbau des CF–Baumes die Handhabung von Rauschen festzulegen. Dabei sind hier Ausreißer als Blattknoten definiert, der weniger als 25% der Durchschnittszahl an Objekten der anderen Blattknoten beinhaltet. In der zweiten Phase des Algorithmus werden die Objekte als Ausreißer identifiziert, deren Abstand zum nahegelegenen Startpunkt mehr als dem doppelten Radius des Clusters entspricht.

DBSCAN

DBSCAN ist gut einsetzbar zur Erkennung jeglicher Clusterformen. Eine Einschränkung stellen jedoch Cluster unterschiedlicher Dichte dar. Bei der Identifikation von Clustern gleicher Dichte, die relativ nah beieinander liegen und dadurch schwer abgrenzbar sind, traten ebenfalls Probleme auf. Der k–Distanz–Graph erwies sich dabei als hilfreiches Instrument zur Bestimmung von ϵ. Die in [Sander, 1998] genannte Regel zur Ermittlung von MinPts ist jedoch nicht bei allen Daten sinnvoll. Nur durch wiederholte Veränderung der Parameter konnte die Clusterlösung verbessert werden. Bei verrauschten Daten wurden die Cluster korrekt identifiziert. Die Rechenzeit ist $O(NlogN)$.

CHAMELEON*

Bei allen durchgeführten Untersuchungen wurden gute Ergebnisse erzielt. Die kreis– und ellipsenförmigen Cluster konnten trotz unterschiedlicher räumlicher Ausdehnung und unterschiedlicher Dichte korrekt identifiziert werden.

Auch die Ergebnisse bei verrauschten Daten können als sehr gut beschrieben werden. Bis zu einem Rauschanteil von 20% wurden fast alle Objekte korrekt zugeordnet, erst oberhalb von 20% kam es zu einem starken Leistungseinbruch.

Dies legt den Schluss nahe, dass die genutzte Kombination partitionierender und hierarchischer Vorgehensweisen sehr erfolgreich ist. Ein Nachteil ist allerdings die ungünstige Skalierung von $O(nm + nlogn + m^2logm)$, wobei n die Anzahl der Objekte und m die Anzahl der Cluster ist.

5 Zusammenfassung

In dieser Arbeit wurden drei Clusteralgorithmen hinsichtlich definierter Kriterien untersucht. Jeder der Algorithmen nutzte eine andere Vorgehensweise zur Ermittlung der Cluster. Dadurch war es möglich, einen Querschnitt über verschiedene Methoden der Clusteranalyse zu geben.

Im direkten Vergleich der Clusterergebnisse aller Algorithmen bei Rauschen indentifizierte CHAMELEON* die am nahesten an den Originalclustern angelehnten Cluster. Auch DBSCAN erreichte sehr gute Clusterergebnisse, vor allem bei klar abgegrenzten Clustern. Sind die Cluster jedoch nicht klar getrennt, sind umfangreiche Analysen und Variationen der Paramter ϵ und MinPts erforderlich.

BIRCH konnte unter Rauschen keine guten Ergebnisse erzielen. In Abbildung 11 sind die erreichten Ergebnisse bei Rauschen zusammenfassend dargestellt.[6]

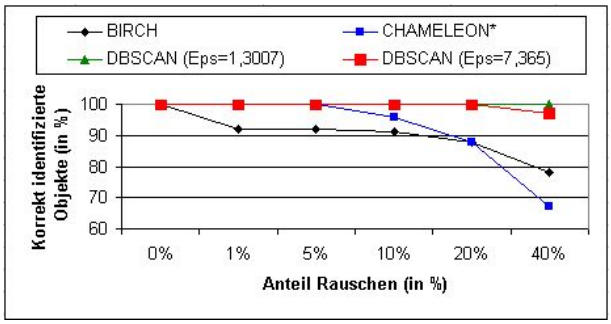

Abbildung 11: Clusterergebnisse unter Rauschen

Bezüglich der Rechenzeiten konnte festgestellt werden, dass BIRCH durch den Einsatz des CF–Trees schneller als die beiden anderen Clusteralgorithmen arbeitet. Allerdings ist erst ab einer Anzahl von 10.000 Punkten oder wenn die Daten nicht mehr im Hauptspeicher gehalten werden können ein deutlicher Unterschied zwischen den Rechenzeiten der Algorithmen zu beobachten. Die hier vorgestellten Datenmengen umfassten jedoch deutlich weniger Punkte.

In Tabelle 1 sind die Ergebnisse für alle Untersuchungskriterien noch einmal zusammengefaßt.

	Algorithmus		
Untersuchte Kriterien	**BIRCH**	**DBSCAN**	**CHAMELEON***
Identifikation von Clustern einfacher Gestalt	akzeptabel	sehr geeignet	sehr geeignet
Identifikation von Clustern nicht–sphärischer Gestalt	ungeeignet	sehr geeignet	sehr geeignet
Identifikation von Clustern bei Rauschen	geeignet	geeignet	sehr geeignet
Rechenzeit	$O(N)$	$O(NlogN)$	$O(nm+nlogn+m^2logm)$
Anzahl benötigter Inputparameter	k - Anzahl der Cluster	MinPts, ϵ	k, M (Clusteranzahl)

Tabelle 1: Zusammenfassende Übersicht der Ergebnisse

Literatur

[Chen, 2001] Chen, Z.: Data Mining and Uncertain Reasoning - An Integrated Approach. John Wiley & Sons, Inc. New York, 2001.

[Ertöz et. al., 2003] Ertöz, L., Steinbach, M. und Kumar, V.: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. AHPCRC Technical Report 2003–102.

[Ester et. al, 1996] Ester, M., Kriegel, H.-P., Sander, J. und Xu,X.: A density–based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Data Mining. Seite 226–231, 1996.

[Han und Kamber, 2001] Han, J. und Kamber, M.: Data Mining. Concepts and Techniques. Academic Press. San Diego, 2001.

[Karypis et. al, 1999] Karypis, G., Han, E.-U. und Kumar, V.: CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer, 32:8, S. 68–75. August 1999.

[Kolatch, 2001] Kolatch, E.: Clusterig Algorithms for Spatial Databases: A Survey. Departement of Computer Science. University of Maryland, College Park. 2001.

[Sander, 1998] Sander, J.: Generalized Density–Based Clustering for Spatial Data Mining. Dissertation an der Ludwig–Maximilians–Universität München. 1998.

[Zhang et. al, 1996] Zhang, T., Ramakrishnan, R. und Livny, M.: BIRCH: An efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, 1996, S. 103–114.

[6] 100% korrekt identifizierte Objekte heißt hier, dass ein Algorithmus unter Rauschen exakt die gleiche Clusterlösung generiert wie ohne Rauschen.