SUBSCALE Algorithmus

William Mendat[1] Max Ernst[2] Steven Schall[3] Matthias Reichenbach[4]

1 Einleitung

Anders, als konventionelle Ansätze bei Clustering-Algorithmen, die sämtliche Features auf einmal vergleichen [KD14], zielt der Subscale Algorithmus darauf ab, die hochdimensionalen Daten in Teilen zu effizient verarbeiten. Dabei möchte der Algorithmus das durch hohe Dimensionalität bedingte, so genannte Problem *Curse of Dimensionality* lösen, indem es die Abgeschlossenheit des Apriori-Prinzips [Bu92] nutzt und die Teilmengen der Datensatzfeatures sukzessive, bottom-up aufbaut.

Das Apriori-Prinzips ermöglicht es aus der Gesamtmenge der Dimensionen in Teilmengen davon, den so genannten Subspaces, so zu verarbeiten, dass anstatt $2^k - 1$ *möglichen Achsenparallele Subspaces in* k *Dimensionen* [KD14], nur die benötigten Subspaces berechnet werden.

2 Daten Aufbereitung

Der Subscale Algorithmus beginnt damit die Daten aufzubereiten. Dazu werden die einzelnen Punkte, die in jeder Dimension enthalten sind, mit einem eindeutigen Index versehen. Die Idee hinter dem Index besteht daraus, dass jeder Punkt eine eindeutige, hohe, zufällig gewählte Ganzzahl als Schlüssel erhält. Später werden die Punkte zu Partitionen zusammengefasst. Dabei bildet die Summe der Schlüssel die Signatur ab. Da jeder Schlüssel einen hohen Wert hat, besitzt die Summe der Schlüssel ebenfalls einen hohen Wert. Laut [KD14] wird für eine sehr hohe Ganzzahl, bei sehr kleiner Partitionsgröße, die Anzahl der einzigartigen, Partitionen bestimmter Größe astronomisch hoch. Dadurch ist die Wahrscheinlichkeit, dass zwei Partitionen die gleiche Zahl als Signatur bilden sehr gering.

Die Signaturen werden verwendet, um paarweise identische *Dense Units* (siehe 5) $U_1^{d_a}$, $U_2^{d_b}$ zwischen den Dimensionen d_a, d_b zu ermitteln.

[1] Hochschule Offenburg, Offenburg, Deutschland wmendat@stud-hs.offenburg.de
[2] Hochschule Offenburg, Offenburg, Deutschland wmendat@stud-hs.offenburg.de
[3] Hochschule Offenburg, Offenburg, Deutschland sschall@stud-hs.offenburg.de
[4] Hochschule Offenburg, Offenburg, Deutschland mreichen@stud-hs.offenburg.de

3 Daten Projektion

Ein Datensatz besteht aus n Zeilen x k Spalten, wobei eine Zeile jeweils ein Punkt P^k als k-Dimensionaler Vektor in k Spalten auftritt. Somit besteht der gesamte Datensatz aus P_n Punkten. Der Datensazu wird zu einem Subspace S von der Größe einer Dimension projeziert, sodass sämtliche Punkte in einer Dimension verglichen werden können. Für die Nachbarschaftsbeziehung kann als Distanzmaß z.B. die euklidische Distanz angenommen werden.

4 CoreSets Erzeugen

Die CoreSets werden durch die einzigen beiden frei wählbaren Parameter des Algorithmus, die vom DBSCAN-Algorithmus inspiriert wurden [KD14], berechnet: ϵ und τ. Ein Referenzpunkt P_i in einem Subspace S ist genau dann mit einem anderen Punkt P_j in S benachbart ($N^S(P_i)$), wenn $dist(P_i^S, P_j^S) < \epsilon$ und $P_i \neq P_j$ gillt. Außerdem muss gelten, dass $|N^S(P_i)| >= \tau$. Es muss also eine Mindestanzahl entsprechend τ an Nachbarn vorhanden sein, damit ein CoreSet gebildet wird. CoreSets können Schnittpunkte in gemeinsamen Punkten bilden.

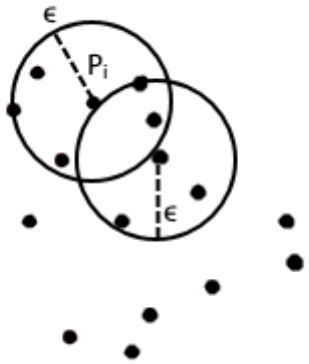

Abb. 1: CoreSet Erzeugung

Auf diese Weise werden für jede Dimension CoreSets gebildet.

5 Berechnung der Dense Units

Clusters von benachbarten Punkten müssen zu einem minPoint Großen subsets kombiniert werden. Diese subsets werden Dense Units genannt. Dense Units werden in verschiedene Funktionen mit unterschiedlichen Zwecken berechnet. Das Hauptziel dieser Berechnung,

ist die Bestimmung einer Teilmenge aus allen möglichen Kombinationen in einem subset. Dabei dürfen keine Wiederholungen der Dense Units auftreten. Die Kombinationen aus einem subset können mittels dem Binominialkoeffizient berechnen, $\binom{n}{k}$ dabei ist n die Anzahl der Elemente in dem subset und k die minimale Anzahl an punkten in einem subset (e.g. minPoints). Daraus entstehen k-Elemente großes subsets von einem n-Elemente set ohne Wiederholungen der Kombinationen. Wenn zum Beispiel ein Core Set aus folgenden Punkten besteht: [1, 5, 7, 9, 22] und die minimale Anzahl an punkten in einem Subset Drei ist, sind die ersten Drei Dense Units folgende: [1, 5, 7] und [1, 5, 9] und [1, 5, 22]. Die Formel zur Berechnung der Dense Units kann also folgendermaßen betrachtet werden: $\binom{|CS|}{minPoints}$ [KD14] Abbildung 2 zeigt ein weiteres Beispiel für n-Punkte in einem Core Set mit minPoints von 4.

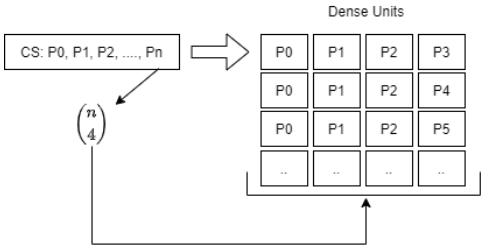

Abb. 2: Berechnung der Dense Units

6 Kollision von Dense Units

Da die Dense Units auch in unterschiedlichen Dimensionen existieren können, müssen diese als solche gekennzeichnet werden. In diesem Schritt ist das subspace clustering erkennbar. Direkt nach Berechnung der Dense Units, werden diese mit anderen Dense Units verglichen um Höhere subspaces zu finden. Um Kollisionen in den Dense Units festzustellen, muss als erstens jedem Punkt in dem Datenset eine hohen Zufallszahl zugewiesen werden. Dieser schritt wird vor dem ausführen des SUBSCALE Algorithmus durchgeführt. Für alle Dense Units, werden die Signaturen gebildet, diese Signatur ist die Summer aller Punkte in der Dense Unit. Anhand dieser Signatur werden die Dense Units in eine Tabelle eingetragen, mit den dazugehörigen Punkten und der Dimension. Bei einer Kollision der Signaturen, wird die Dimension zu dem bereits vorhanden Eintrag hinzugefügt (siehe Abbildung 3). [Ra]

7 Abbildung Dense-Units auf Subspaces

Für das endgültige Clustering muss die Ausgabe des SUBSCALE-Algorithmus in die Struktur von den Clusterkandidaten abgebildet werden. Jeder Kandidat besteht aus Dimensionen und eindeutigen Punkt IDs. Dabei werden Punkte ausgesucht, welche in mehreren Dense-Units

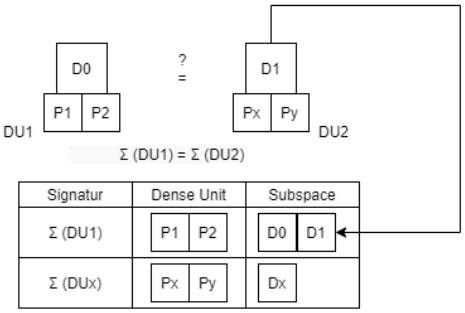

Abb. 3: Kollisionsauflösung von Dense Units

vertreten sind. Diese werden dann zusammen innerhalb eines Subspaces definiert, wobei mehrfach vorkommende Punkte nur einmal eingetragen werden. Die hier zusammengeführten Punkte haben die Eigenschaft, dass sie ziemlich wahrscheinlich auch in dem jeweiligen Unterraum geclustert sind. Diese Eigenschaft macht diese Punkte zu günstigen Kandidaten für das endgültige Clustering, welches im folgenden Unterkaptiel 8 näher beschrieben wird.

8 Abschließendes Clustering mit DBSCAN

Nach dem alle maximalen Subspaces identifiziert worden sind, wird zuletzt das Clustering durchgeführt, welches die maximalen Supspace Cluster finden soll. Zur Bewältigung dieser Aufgabe wird ein volldimensionaler Clustering-Algorithmus verwendet. Der Algorithmus, welcher in der von uns verwendeten Implementierung verwendet wird, nennt sich DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN ist ein auf die Dichteverbundenheit basierender Clustering-Algorithmus, der Cluster mit beliebiger Form findet und Rauschpunkte separat zurückliefert.

9 Verteilungsmöglichkeiten des SUBSCALE Algorithmus

[PLK21]

Literaturverzeichnis

[Bu92] Buprenorphine: An alternative treatment for opioid dependence ; [based on the papers and discussions from a Technical Review on "Buprenorphine: an Alternative Treatment for Opioid Dependence", held on March 16 - 17, 1989, in Rockville, Md, Jgg. 92,1912 in DHHS publication (ADM). U.S. Dep. of Health and Human Services Publ. Health Service Alcohol Drug Abuse and Mental Health Administration National Inst. on Drug Abuse, Rockville, Md., 1992.

[KD14] Kaur, Amardeep; Datta, Amitava: SUBSCALE: Fast and Scalable Subspace Clustering for High Dimensional Data. In: 2014 IEEE International Conference on Data Mining Workshop. S. 621–628, 2014.

[PLK21] Prinzbach, Jürgen; Lauer, Tobias; Kiefer, Nicolas: Accelerating Density-Based Subspace Clustering in High-Dimensional Data. In: 2021 International Conference on Data Mining Workshops (ICDMW). S. 474–481, 2021.

[Ra] Ramin, Stanislav: Python Implementation of the SUBSCALE Algorithm. Bachelor thesis, Offenburg.