Abhängigkeits-Clustering über verschiedende Messskalen

Lilas Chakhachirou
Betreuer: Collin Leiber, Walid Durani, Mauritius Klein
LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR DATENBANKSYSTEME UND DATA MINING
Email: l.chakhachirou@campus.lmu.de

***Abstract*—Wie lassen sich numerische und kategoriale Daten in einem Cluster zusammenfassen? Trotz unterschiedlicher Messskalen verschiedener Datentypen gibt es bereits Lösungen für dieses Problem. Allerdings bergen diese oft Herausforderungen wie die Festlegung einer bestimmten Clusteranzahl oder mangelnde Unterstützung von Attributabhängigkeiten, was die Interpretation erschwert. Zudem sind viele Techniken auf einen Datentyp beschränkt. Wir präsentieren den Algorithmus Scenic: Ein parameterfreier Ansatz, der heterogene Daten über unterschiedliche Messskalen hinweg integriert. Mittels des Minimum Description Length-Prinzips wandelt Scenic hochdimensionale Daten in einen niedrigdimensionalen Raum um und verhindert dabei Overfitting.**

1. Motivation

Der Begriff Big Data“ ist inzwischen weit verbreitet. Daten werden überall im Alltag gesammelt und analysiert, sei es in sozialen Medien, Börsenvorhersagen oder medizinischen Diagnosen und Prognosen. Solche Anwendungen sehen sich mit komplexen und heterogenen Daten konfrontiert. Diese Datenstrukturen beinhalten häufig unterschiedliche Datentypen wie numerische, kategoriale und binäre Typen, die gemeinsam in einem Datensatz vorkommen und aufgrund ihrer verschiedenen Messskalen verglichen werden müssen. Ein gutes Beispiel hierfür findet sich im Bereich der Biomedizin, wo numerische Attribute wie Laborparameter neben kategorialen Attributen wie Genotypen oder binären Attributen wie dem Geschlecht stehen.

Die zunehmende Dimensionalität stellt eine große Herausforderung für Clustering-Algorithmen dar. Sie erschwert die Erkennung von Mustern zwischen den Attributen, was zu einer schlechten Interpretierbarkeit der Datensätze führt und Overfitting sowie längere Lernzeiten für Algorithmen zur Folge haben kann. Dieses Problem wurde von Bellman als Curse of Dimensionality“ [1] bezeichnet und findet sich in zahlreichen wissenschaftlichen Arbeiten wieder. Ein guter Clustering-Algorithmus ermöglicht eine präzise und klare Datenanalyse, indem er das volle Potenzial der verfügbaren Daten ausschöpft. Daher unterstützen viele Data-Mining-Verfahren die Integration heterogener Daten über verschiedene Messskalen hinweg. In solch komplexen Datensätzen können jedoch einige Attribute durch Kombinationen anderer Attribute repräsentiert werden. Diese Attributabhängigkeiten beeinflussen die sogenannte ”intrinsische Dimensionalität” des Datensatzes. Einige Attributabhängigkeiten treten in bestimmten Clustern auf, andere nicht. Die intrinsische Dimensionalität gibt die Anzahl der erforderlichen Variablen an, um eine Information minimal darzustellen. Jedoch tragen nicht alle Attribute gleichermaßen zur Informationsgewinnung bei.

Die Gliederung meiner Arbeit ist wie folgt: Kapitel 2 erklärt die für den Algorithmus Scenic erforderlichen Methoden, einschließlich des MDL-Prinzips. Kapitel 3 präsentiert den effizienten und parameterfreien Algorithmus Scenic. Kapitel 4 beschreibt die durchgeführten Experimente sowohl mit synthetischen als auch mit realen Datensätzen und vergleicht die Ergebnisse von Scenic mit anderen Algorithmen für gemischte Datentypen wie K-Means Mixed und INCONCO. Kapitel 5 fasst die Arbeit abschließend zusammen.

2. Methoden

In der Clusteranalyse numerischer Daten hat sich das Korrelation Clustering“ als effizient erwiesen. Viele Data-Mining-Verfahren integrieren die Hauptkomponentenanalyse (PCA), um Muster in numerischen Attributen über ihre kontinuierlichen Messskalen zu erkennen. Diese erkannten Abhängigkeiten unterstützen dann das Gruppieren von Daten nach ihrer Korrelation in spezifischen Clustern , wodurch die Interpretierbarkeit des Ergebnisses verbessert wird.

Allerdings sind diese Ansätze lediglich für Vektordaten bzw. für numerische Attribute geeignet. Aus diesem Grund schlägt Scenic vor, die Daten zunächst in einen niedrigdimensionalen Attribut-Objekt Vektorraum (AO-Vektorraum) einzubetten. Dabei ist es wichtig, sowohl die Reihenfolge als auch die Abstandsbeschränkungen zu berücksichtigen. Ein Hauptziel besteht darin, eine hohe Vorhersagegenauigkeit zu erzielen, was durch eine Maximierung der Clusteranzahl erreicht werden kann. Dies könnte jedoch zur Gefahr der Überanpassung an die Daten führen. Um diesem

1. Dimensionality Reduction for Handwritten Digit Recognition, Ankita; Kundu, Tuhin; Chandran Saravanan: https://www.proquest.com/docview/2305560343?pq-origsite=primo

Problem entgegenzuwirken, kombiniert Scenic die Idee der Datenkompression mit unüberwachter Klassifikation. Dadurch können wir Cluster identifizieren, die sowohl eine hohe Accuracy als auch eine angemessene Modellkomplexität aufweisen. Dieses Prinzip ist bekannt als Minimum Description Length“, kurz beschrieben MDL.

2.1. Attribut-Objekt Vektorraum

Einen guten Objekt-Attributen Vektorraum ist entscheidend unüberwachtes Klassifikation. Dieser hochdimensionale Vektorraum setzt sich wie folgt zusammen:

- Eine n×d_v Matrix (x): Sie enthält die niedrigdimensionalen Objektkoordinaten (x) für jedes Objekt x in Form eines Zeilenvektors.
- Eine c×d_v Matrix (a): Sie enthält die niedrigdimensionalen Attributkoordinaten (a) für jedes Attribut A und den zugehörigen Wert a.
  Dabei bezeichnet c die Anzahl der unterschiedlichen Attributwerte von allen numerischen und kategorialen Attributen im Datensatz.

Wir befinden uns in einem hochdimensionalen Vektorraum und möchten unsere Daten in einen niedrigdimensionalen Raum transformieren. Die Positionierung der Objekte in diesem Raum sollte unter Beibehaltung der Reihenfolge und Abstandsbeschränkungen erfolgen. Auf diese Weise wird durch die Einbettung der Objekte und ihrer zugehörigen numerischen Attribute weder die Datenintegrität noch die Relevanz von Abhängigkeiten beeinträchtigt. Jedoch wird in einem solchen Vektorraum zwischen kontinuierlichen und nominalen Attributen unterschieden. Kontinuierliche Attribute lassen sich durch Reihenfolge und Abstandsbeschränkungen als Linien repräsentieren, während nominale Attribute keine Reihenfolge oder Abstandsbeschränkung haben. Ihre Positionierung im Vektorraum ist daher nicht eingeschränkt.

Für jedes numerische Attribut A gibt es Reihenfolge- und Abstandsbeschränkungen $\pi(a)$:

- Reihenfolgebeschränkung: Für alle Merkmalswerte a1, a2 eines numerischen Attributs gilt:
  $a_1 < a_2 \Rightarrow \pi(a_1)_d < \pi(a_2)_d$
- Abstandsbeschränkung: Für die drei Werte a_1,a_2, a_3 gilt:
  $a_1 = a_2 \cdot a_3 \Rightarrow \pi(a_1)_d = \pi(a_2)_d \cdot \pi(a_3)_d$

Der Ziel so einen AO-Vektorraum die Hauptaspekte einer komplexen Hochdimensionalen Ähnlichkeit zwischen gemischttypischen Datenobjekten und Attributen in kompakter Formdazustellen.2 Als Beispiel betrachten wir Abbildung 1. Jede Instanz wird durch fünf Attribute repräsentiert: die numerischen Attribute x und y sowie drei kategoriale Attribute. Das Attribut ”Farbe” hat die Ausprägungen Rot, Grün und Blau; das Attribut ”Symbol” die Ausprägungen Kästchen und Dreieck; und das Attribut ”Füllung” die Ausprägungen offen und gefüllt.

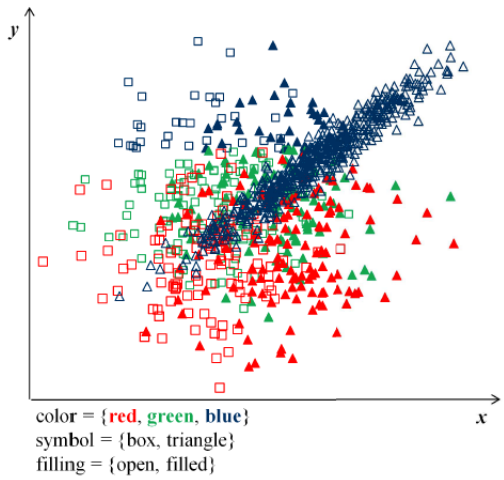

Figure 1. laufendes Beispiel

Mithilfe des Algorithmus Scenic analysierten wir den in Abbildung 1 dargestellten Datensatz und identifizierten zwei relevante Cluster, dargestellt in Abbildung 2. Im Cluster 1 von Abbildung 2(a) zeigt sich eine signifikante Korrelation zwischen den numerischen Koordinaten x und y. Abbildungen 2(b) und 2(c) visualisieren die räumliche Anordnung der Objekte und ihrer zugehörigen Attribute in diesen Clustern. Um eine klare Übersicht zu gewährleisten, wurden sie in separaten Sub-Figuren präsentiert. Der erste Cluster charakterisiert ein Objekt, genauer gesagt ein offenes blaues Dreieck. Deshalb sind die kategorialen Attribute zentral positioniert. In Abbildung 2(c) verdeutlicht ein kleiner Winkel zwischen zwei Linien die intensive Korrelation. Cluster 2 in Abbildung 2(c) zeigt, wie Scenic komplexe Abhängigkeiten, speziell zwischen den kategorialen Werten offenes Kästchen und gefülltes Dreieck, visualisiert. Hierbei wird keine direkte Beziehung zwischen den numerischen Koordinaten erkannt. Dennoch gibt es in Abbildung 2(b) eine Korrelation zwischen zwei kategorialen Attributen und der x-Koordinate: Mit steigendem x-Wert wechselt das Attribut ”Symbol” von offenem Kästchen zu gefülltem Dreieck. Eine zusätzliche Abhängigkeit ist entlang der Y-Achse sichtbar: Bei wachsendem y-Wert ändert sich die Farbe von Rot zu Blau.

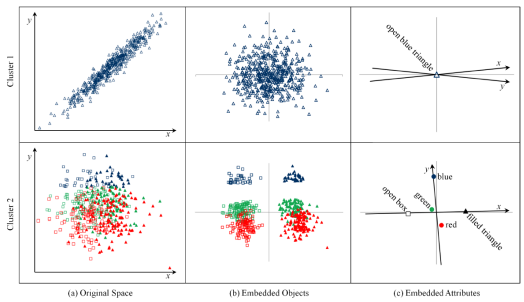

Figure 2. Einzige Cluster des laufenden Beispiels: (a) Originalraum; (b,c) Objekte und Attribute im Attribut-Objekt (AO) Raum.

2.2. Minimum Beschreibungslänge MDL

Im zweiten Kapitel wurde betont, dass ein gut konstruierter AO-Raum alleine nicht die Garantie für eine präzise Vorhersage bietet. Es besteht das Overfitting-Risiko, wenn jedes Cluster ausschließlich aus Objekten mit ihren zugehörigen Attributwerten eingebettet sind. Unter solchen Bedingungen würde jedes Cluster, unabhängig von seiner Klassifizierungsmethode, stets optimale Ergebnisse liefern. Deswegen hilft das MDL-Prinzip, eine Balance zwischen Datenkompression und einer geringen Modellkomplexität herstellt.

Scenic nutzt MDL, um eine möglichst kurze Darstellung von Daten mit erkennbaren Gemeinsamkeiten oder Abhängigkeiten zu zeigen. Dabei werden diese Informationen in einem Cluster zusammengefasst, wodurch eine optimale Anzahl von Clustern angestrebt wird. Es gilt folgendes:

$$\min \sum DL(Ci)$$

Somit sorgt Scenic dafür, dass es parameterfrei läuft und ohne Eingabe einer Clusteranzahl. Die Laufzeit des Algorithmus wird auch von der Beschreibungslänge abhängig.

Um die Beschreibungslänge für ein solches Modell zu bestimmen, müssen wir zunächst die Rekonstruktionsfehler und Modellkomplexität berechnen.

2.2.1. Rekonstruktionsfehler RE_{ci}. Bei der Daten- Einbettung soll es möglich sein, von den eingebetteten Daten zurück zu den ursprünglichen Attributswerten zu gelangen, um die Ergebnisse korrekt interpretieren zu können. Daher repräsentiert RE_{ci} die Anzahl der benötigten Bits für die Rekonstruktion der ursprüngliche Daten und misst dabei die Abweichung zwischen den ursprünglichen Attributswerten und den rekonstruierten Werten im AO-Raum. Da wir uns in einem Vektorraum aus heterogenen Daten befinden, müssen wir den Rekonstruktionsfehler sowohl für kategoriale als auch für numerische Attribute berechnen.

Kategoriale Attribute: Für kategoriale Attribute nutzt MDL in solchen probabilistischen Modellen das Bayes'sche Theorem, um den Rekonstruktionsfehler zu berechnen. Es gibt die Wahrscheinlichkeit an, die Repräsentation $\pi(x)$ zu beobachten, wenn die Kategorie a gegeben ist. $p(a)$ wird als Prior-Wahrscheinlichkeit und $p(a|\pi(x))$ als korrespondierende posteriori Wahrscheinlichkeit interpretiert. $p(\pi(x))$ wird als Summe über alle Kategorien a von $p(\pi(x)|a)$ definiert.

Um $p(\pi(x)|a)$ zu berechnen, wird eine geeignete Wahrscheinlichkeitsverteilung für die Kategorie a im AO-Raum bzw. für kontinuierliche Daten verwendet. Zunächst wählen wir die passende probabilistische Verteilung. Hierbei wird eine Gaußsche Wahrscheinlichkeitsdichtefunktion (PDF) $N_a(\mu_a, \Sigma_a)$ mit diagonalen Kovarianzen verwendet, die für die Maximierung der Entropie vorteilhaft ist. Der genaue Grund hierfür wird später erläutert. Eine Kovarianzmatrix ist eine quadratische Matrix, bei der die Elemente auf der Diagonale die Varianzen repräsentieren und die Elemente außerhalb der Hauptdiagonale die Kovarianzen zwischen den Attributen darstellen. Dies hilft, die Korrelation zwischen den Attributen zu bestimmen:

$$p(\pi(x)|a) = (2\pi)^{-d/2} \cdot |\Sigma_a|^{-1/2} \cdot \exp\left((\pi(x) - \mu_a)^T \cdot \Sigma_a^{-1} \cdot (\pi(x) - \mu_a)\right)$$

Wobei μ_a für den Mittelwert der Kategorie a steht und $p(a)$ als $|a|/n$ definiert wird, wobei $|a|$ die Anzahl der Elemente in der Kategorie a und n die Gesamtzahl der Kategorien ist. $(\pi(x)-\mu_a)^T$ repräsentiert den transponierten Spaltenvektor und $(\pi(x) - \mu_a)$ den Zeilenvektor.

Ein höherer Rekonstruktionsfehler weist auf eine größere Unsicherheit der kategorialen Attributswerte nach der Einbettung im AO-Raum hin. Diese Unsicherheit entsteht, weil die Einbettung in den AO-Raum möglicherweise nicht alle Informationen über die kategorialen Attribute bewahrt. Da wir die richtige Kategorie für das Objekt x nicht kennen, erwarten wir uns eine hohe Wahrscheinlichkeit für $p(a|\pi(x))$. Das Maximieren dieser Wahrscheinlichkeit ist äquivalent zur Minimierung ihres negativen Logarithmus. Somit minimieren wir eventuell entstandene Unsicherheiten und Fehler. Weiterhin nutzen wir das Huffman-Coding-Prinzip, um den Rekonstruktionsfehler in Bits zu quantifizieren, indem wir den Logarithmus der Summe der wahren Kategorie a_x über alle Objekte x in einem Cluster C_i und über alle kategorialen Attribute A_{cat} berechnen:

$$RE_{x,A} = \sum_{x \in C_i} \sum_{A_{cat}} -\log_2(p(a_x|\pi(x)))$$

Nummerische Attribute: Um den Rekonstruktionsfehler für ein numerisches Attribut B in einem Cluster C_i zu quantifizieren, wird eine multivariate Regression angewendet. Hierbei dienen die AO-Koordinaten der Objekte $\pi(C_i)$ als unabhängige Variablen (Regressoren) und die ursprünglichen Attributwerte $C_{i,B}$ als abhängige Variable betrachtet. Bei diesem Klassifikationsproblem versuchen wir, den Regressionskoeffizienten zu schätzen, um das Modell bestmöglich an die Daten anzupassen. Dabei analysieren wir ebensfalls, wie die AO-Koordinaten auf die ursprünglichen Attributwerte einen Einfluss nehmen.

Die Regression wird durch die folgende Gleichung ausgedrückt:

$$C_{i,B} = \pi(C_i) \cdot \beta_B + \varepsilon_B$$

Dabei sind β_B die Regressionskoeffizienten und ε_B ist der Fehlerterm. Es handelt sich hierbei um eine lineare Regression mit einer Gaußschen Fehlerverteilung.

Die Kodierungskosten für den Rekonstruktionsfehler ergeben sich aus dem negativen Log-Likelihood der Fehlerverteilung berechnet. Dieser Wert wird über alle Objekte in Cluster C_i und über alle numerischen Attribute summiert. Das heißt, je geringer der negative Log-Likelihood, desto besser passt die Modellierung zu den Daten.

$$RE_{num} = \sum_{x \in C_i} \sum_{A_{num}} -\log_2\left(\frac{1}{\sigma_A\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_A)^2}{\sigma_A^2}\right)\right)$$

2.2.2. Modellkomplexität MCci. Die Modellkomplexität ergibt sich aus der Summe der Parameter-Kodierungskosten Pcost und den ID-Kosten. Das Ziel dahinter ist es, abzuschätzen, wie viele Informationen oder Bits benötigt werden, um das Modell in komprimierter Form zu speichern. **Berechnung der ID-Kosten:** ID-Kosten werden mithilfe der Huffman-Kodierung als Identifikator für den Cluster der jeweiligen Objekte berechnet. Sie werden wie folgt berechnet:

$$IDcost = |Ci| \cdot \log_2 \left(\frac{n}{|Ci|} \right) \tag{1}$$

Bezüglich **der Kodierungskosten für die Parameter** des Modells werden diese wie folgt geschätzt:

$$Pcost = \frac{|m|}{2} \cdot \log_2(|Ci|) \tag{2}$$

Ein wichtiger Punkt ist die Berechnung der Anzahl der Modellparameter. Sie besteht aus der Summe der benötigten Kodierungskosten der Objektrepräsentationen (1-ter Term), dem Mittelwert und der Varianz der Kategorien (2-ter Term) sowie dem für die linearen Modelle der numerischen Attribute benötigten Modellkoeffizienten β (3-ter Term).

$$|m| = dv \cdot n + 2 \cdot dv \cdot |cat| + dv \cdot dn \tag{3}$$

3. Algorithm Scenic

algorithm Scenic (): set of clusters
  Cluster $\mathcal{C}_S := k$-Scenic (1);
  return REC-SPLIT ($\mathcal{C}_R$) ;
  determine best AO of clusters with MDL ;

algorithm k-Scenic (k): set of k clusters
  $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$:= INITIALIZATION (k);
  repeat
    assign every object to $\mathcal{C}_i$ with minimum coding cost;
    update AO space as in Section 3.1;
  until convergence;
  return $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$;

procedure INITIALIZATION (k): set of k clusters
  k-d Princals;
  obtain l object clusters for each category combination;
  while $l > k$ **do**
    merge most similar category combination;
  return $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$;

procedure REC-SPLIT (Cluster $\mathcal{C}$): set of clusters
  $\{\mathcal{C}_L, \mathcal{C}_R\} := k$-Scenic (2);
  if $MDL(\mathcal{C}_L) + MDL(\mathcal{C}_R) \geq MDL(\mathcal{C})$ **then**
    return $\{\mathcal{C}\}$;
  else
    return REC-SPLIT ($\mathcal{C}_L$)$\cup$ REC-SPLIT ($\mathcal{C}_R$);

Figure 3. Die Scenic Algorithm

Die beigefügten Abbildung (Figure3) zeigt den Pseudo-Code für den Scenic-Algorithmus. In vorherigen Kapiteln wurde Wie in früheren Kapiteln diskutiert, zeichnet sich Scenic durch seinen parameterfreien Ansatz aus. Dabei greift er auf den K-Means-ähnlichen Algorithmus K-Scenic mit festgelegter Clusteranzahl zurück und kombiniert diesen mit einer effizienten Top-Down Splitting-Strategie. Anfänglich befinden sich alle Objekte in einem Cluster (K-Scenic(1)). Anschließend wendet Scenic rekursiv 2-Scenic an, solange eine Verbesserung der Kodierungskosten feststellbar ist.

Für die Initialisierung von K-Scenic wird der Algorithmus Princals verwendet, der ähnlich wie K-Means verfährt, um einen geeigneten AO-Vektorraum zu erzeugen. Princals wählt Koordinaten zufällig aus, um sicherzustellen, dass die Positionierung der eingebetteten Objekte vairiert und somit sich von geteilten Clusterobjekten unterscheidet. Der Algorithmus iteriert in zwei Schritten bis zur Konvergenz:

1) Die Koordinaten der Attributwerte werden als Mittelwert aller Objekte genutzt. Dabei werden die Attributwerte basierend auf den Objekten, die diesen Wert besitzen, durchschnittlich positioniert. Um Raum- und Reihenfolgebeschränkungen zu beachten, wird die Position der numerischen Attributwerte mittels linearer Regression angepasst.

2) Bestimmung der Koordinaten eines jeden Objekts durch den Mittelwert der zugehörigen Kategoriekoordinaten.

Nach Erreichen der Konvergenz sind die Objektkoordinaten orthogonal und zeigen keine lineare Abhängigkeit mehr. Damit befinden sich alle Kategoriezentroide im selben Bereich des Einbettungsraums und teilen eine gemeinsame Varianz.

Der Algorithmus überprüft zunächst die Bedingung, dass, falls mehr als k Kombinationen von Kategorien existieren, in jedem Schritt die ähnlichsten Kombinationen verschmolzen werden, bis k Cluster erreicht sind. Die Verschmelzung zweier Kombinationskategorien erfolgt durch Ersetzen ihrer multivariaten Gaußverteilungen durch einen Repräsentanten, wie in Abbildung 4 (Figure 4) dargestellt. Dieser Repräsentant besitzt die minimale Kullback-Leibler-Divergenz der beiden Kombinationen. Dies definiert als ein Maß für den Unterschied zwischen zwei Wahrscheinlichkeitsverteilungen[2]. Das bedeutet, dass die repräsentative Gaußverteilung so gewählt wird, dass sie möglichst geringe Abweichungen zu den ursprünglichen Verteilungen aufweist.

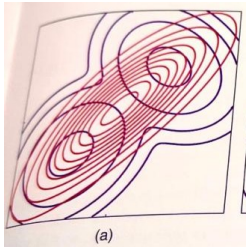

Figure 4. Die blauen Konturen zeigen eine bimodale Verteilung, die durch eine Mischung von zwei Gaußschen Verteilungen gegeben ist, und die roten Konturen entsprechen der neue Repräsentant

2. Kullback-Leibler-Divergenz: https://de.wikipedia.org/wiki/Kullback-Leibler-Divergenz

Nach der Initialisierung erzeugt Princals Clustern für jeweilige Kategoriekombinationen. K-Scenic iteriert in zwei Phasen, bis Konvergenz erreicht ist:

1) Objektzuweisung zu Clustern basierend auf minimalen Kodierungskosten, wie in Kapitel 2.2 bereits erklärt.
2) Aktualisierung des Clustermodells bzw. des AO-Raums.

Abschließend führt Scenic in der Aufteilungsphase ein Clustering im 2-dimensionalen AO-Raum durch. Am Ende wird die beste Dimensionalität des AO-Raums für jeden Cluster mit MDL (Minimum Description Length) ermittelt.

Die Laufzeitkomplexität von Scenic, referenziert auf Claudia Plants Paper, hängt von der Anzahl der Iterationen in den K-Scenic-Aufrufen und der Anzahl der Iterationen in Princals ab. Die Laufzeit für K-Scenic wird mit $n \cdot |iter_{kSc}| \cdot |(iter_P \cdot nd^2)|$ angegeben, wobei der letzte Term die Zeit für die Gram-Schmidt-Orthogonalisierung repräsentiert. Typischerweise liegt dieser Wert zwischen 10 und 50, sodass der Algorithmus als effizient betrachtet werden kann.

4. Experimente

In dem folgenden Abschnitt vergleichen wir die Clustering-Leistung von Scenic mit anderen Clustering-Methoden für heterogene Daten, wie K-Means-Mixed, K-Modes und INCONCO. Als Qualitätsmaß verwenden wir die Normalisierte Gegenseitige Information (NMI - Normalized Mutual Information), die Werte zwischen 0 und 1 annimmt. Je höher dieser Wert ist, desto besser ist die Clustering-Methode. Unsere Experimente führen wir sowohl mit synthetischen als auch mit realen Datensätzen durch.

4.1. Synthtische Daten

Hier betrachten wir unser laufendes Beispiel und vergleichen das Ergebnis von Scenic mit den Methoden K-Means-Mixed, K-Modes und INCONCO. Der zu untersuchende Datensatz umfasst insgesamt 1.000 Objekte. Diese werden durch drei nominale und zwei numerische Attribute charakterisiert. Dabei enthält jedes Cluster genau 500 Objekte.

In der vorliegenden Abbildung (Figure 5) konnten mit Scenic zwei Cluster identifiziert werden. Der erste Cluster zeigt eine deutliche numerische Korrelation, während im zweiten Cluster eine Attributabhängigkeit zu erkennen ist. Dies resultiert in einem perfekten NMI-Wert von 1 für Scenic. Als Nächstes betrachten wir K-Means-Mixed. Dieser Algorithmus besitzt ein Optimierungsschema, das ihm hilft, die Relevanz eines Attributs für den Cluster bestimmt. Deshalb erkennen wir, dass Cluster 1 durch das Attribut Farbe gesteuert wird. Allerding fehlt ihm die Fähigkeit, komplexe Abhängigkeiten wie die zwischen Farbe und Füllung zu erfassen. Dies führt zu einem NMI von 0,71. Weiter geht es mit der K-Modes-diskretisierten

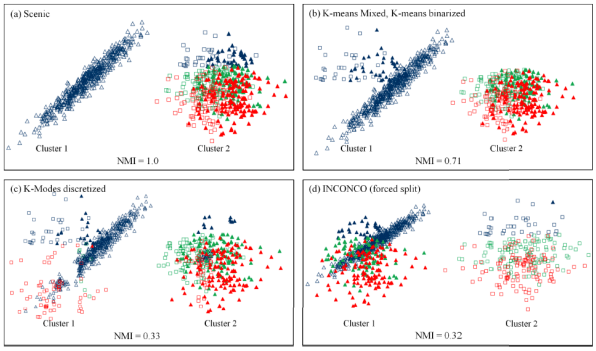

Figure 5. Resultat des laufendes Besipiel

Methode, bei der numerische Daten in diskrete Werte umgewandelt werden. Hierbei sehen wir jedoch kein interpretierbares oder klares Clusterergebnis, weshalb K-Modes einen niedrigen NMI-Wert von 0,33 aufweist. Schließlich betrachten wir den Algorithmus INCONCO. Obwohl es leistungsfähig im Erkennen gemischter Abhängigkeitsmuster ist, hat seine eigenen Einschränkungen. Es erfordert eine eindeutige numerische Datenverteilung für alle Kategorien nominaler Variablen. Diese Limitation zeigt sich in der Interpretation des Übergangs von offenen Kisten zu gefüllten Dreiecken in Abhängigkeit vom x-Wert. Anders als Scenic, das auf dem MDL-Prinzip basiert, bestimmt INCONCO die Clusteranzahl automatisch. Zusammenfassend zeigte Scenic in diesem Experiment eine starke Leistung bei der Erkennung von Abhängigkeiten sowohl zwischen kategorialen als auch numerischen Attributen und wählte zudem die optimale Anzahl an Clustern automatisch. Andere Algorithmen taten sich schwerer dabei, sich auf Abhängigkeiten zwischen mehreren Attributen zu konzentrieren. Zudem zeigte Scenic eine schnellere Laufzeit als K-Means-Mixed, war jedoch nicht schneller als INCONCO, K-Means und K-Modes.

4.2. Reale Daten

Hier führen wir zwei Experimente mit zwei Datensätzen durch und vergleichen die Ergebnisse mit den Methoden INCONCO und K-Means-Mixed.

4.2.1. Abalone DAtensatz. Dieser Datensatz besteht aus 4.177 Instanzen. Er wird durch ein kategoriales Geschlechtsattribut mit den Werten ”männlich”, ”weiblich” und ”jung (infant)” charakterisiert. Die weiteren acht numerischen Attribute repräsentieren verschiedene Messungen der Abalone-Muschel. Das Attribut ”Ringe” gibt die Anzahl der Ringe der Muschel an und ermöglicht Rückschlüsse auf das Alter des Tieres.

Ziel: Die Population der Abalone-Muscheln in Tasmanien auf eine unüberwachte Art und Weise zu untersuchen.

Wir haben die folgenden Ergebnisse erzielt und in Tabelle 1 zusammengefasst. Dabei steht ”n” für die Anzahl der erkannten Cluster:

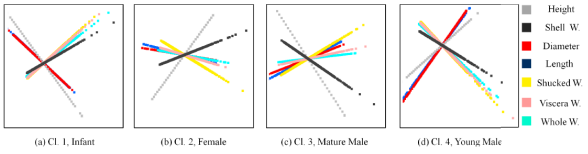

Figure 6. Cluster-spezifische Einbettung von Attributen auf Abalone-Daten

TABLE 1. RESULTAT FÜR ABALONE DATENSATZ

Scenic	K-Mixed	INCONCO
n= 4	n= 4	n= 25

Scenic korreliert gut mit dem Attribut ”Geschlecht” und verwendet dieses, um die Daten in 4 Clustern zu gruppieren. Dabei repräsentiert jeder Cluster genau eine Geschlechtskategorie. Es erkennt sogar ein Abhängigkeitsmuster in den männlichen Instanzen, wodurch diese in zwei Cluster unterteilt werden.

K-Mixed gruppiert die Daten ebenfalls in vier Cluster. Es liefert auch plausible Ergebnisse in Bezug auf das Attribut ”Ringe”, aber aufgrund seiner eingeschränkten Fähigkeit, Abhängigkeiten zwischen Attributen zu berücksichtigen, ist es schwerer zu interpretieren. Im Gegensatz dazu erkennt INCONCO 25 Cluster, darunter vier große Cluster mit jeweils 300 Objekten. Es generiert viele kleine Cluster, ohne eine klare Repräsentation der Geschlechter oder signifikante Unterschiede im Bezug auf das Attribut ”Ringe” zu zeigen.

4.2.2. akuter Entzündungen Datensatz. Hierbei geht es um die Diagnose akuter Entzündungen der Harnblase und akuter Nephritis. Unser Datensatz besteht aus 120 Instanzen, wobei jede Instanz einen Patienten repräsentiert. Die erhaltenen Ergebnisse haben wir in Tabelle 2 zusammengefasst; dabei steht n für die Anzahl der erkannten Cluster.

Zusammenfassend wird bemerkt, dass sowohl INCONCO als auch Scenic ähnliche Ergebnisse in Bezug auf die Bewertungsattribute erzielen. Jedoch weist INCONCO eine etwas höhere NMI auf. Scenic erkennt ein zusätzliches Cluster, was sich negativ auf die NMI auswirkt, da die Bewertungsattribute binär sind. K-means Mixed liefert im Vergleich weniger aussagekräftige Clusterergebnisse bezogen auf die Bewertungsattribute. Scenic bleibt jedoch die beste Methode, da sie Ergebnisse liefert, die am besten mit den tatsächlichen Diagnosen übereinstimmen.

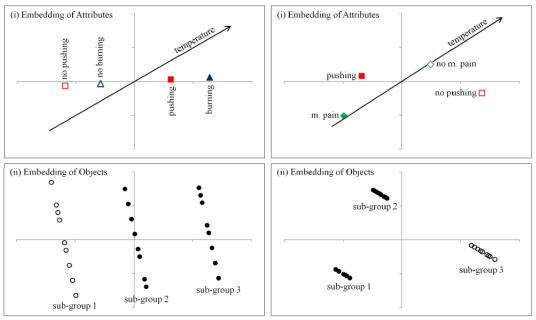

Figure 7. Einbettung des ausgewählten Clusters akuter Entzündung.

TABLE 2. RESULTAT DES AKUTE ENTZÜNDUNG DATENSATZ

Scenic	K-Mixed
n = 5	n = 4
NMI = 0,24 bezüglich Evaluation Attribute 1 NMI = 0,43 bezüglich Evaluation Attribute 2	NMI = 0,007 bezüglich Evaluation Att. NMI = 0,20 bezüglich Evaluation Att.2

INCONCO
n = 4
NMI = 0,11 bezüglich Evaluation Attribute 1 NMI = 0,52 bezüglich Evaluation Attribute 2

5. Conclusion

In diesem Paper wurde das Abhängigkeits-Clustering für heterogene Daten vorgestellt sowie die Herausforderungen, auf die man in einer unüberwachten Umgebung trifft. Es wurde erläutert, wie man einen geeigneten AO-Vektorraum erstellt, sodass die Daten ohne Verlust effizient eingebettet werden können. Anschließend wurde der Algorithmus Scenic vorgestellt und mit anderen Methoden verglichen. Dabei hat er eine gute Leistung beim Clustering von Daten mit unterschiedlichen Messskalen gezeigt. Scenic betrachtet das Clustering als Klassifikationsproblem und erzeugt eine niedrigdimensionale Einbettung von Objekten und ihren Attributen, um die ursprünglichen Attributwerte mit hoher Genauigkeit vorherzusagen. Das Problem des Overfittings wird vermieden, indem Scenic die Konzepte von Datenkompression und unüberwachter Klassifikation verwendet. Die Experimente verdeutlichten die gute Interpretierbarkeit der Clusterergebnisse. Scenic ermöglicht eine klare Visualisierung, die dazu beiträgt, weitere Attributmuster zu entdecken. Dies ist besonders wichtig bei medizinischen Diagnosen , was im Kapitel "Motivation" diskutiert wurde. Der Algorithmus nutzt diese Attributabhängigkeit zur Trennung der Daten in sinnvolle Cluster, was am Beispiel der Abalone-Muscheln und deren Altersbestimmung gezeigt wurde. Ein weiterer Vorteil von Scenic ist, dass die Anzahl der Cluster automatisch mithilfe des MDL-Prinzips bestimmt wird, wodurch er parameterfrei ist.

Es bestehen natürlich weiterhin offene Herausforderungen beim Clustering von heterogenen Daten, wie beispielsweise Subgruppen-Clustering oder Fuzzy-Clustering, bei denen möglicherweise keine linearen Abhängigkeiten zwischen den Attributen und den Objekten existieren. Zukünftig sollte auch die Erkennung von interessanten, nicht-redundanten Clustern sowie die Ausreißererkennung unter Berücksichtigung der Abhängigkeiten zwischen numerischen und kategorialen Attributen untersucht werden.

References

[1] P. E. Alpaydin, *Maschinelles Lernen*, 3., aktualisierte und erweiterte Auflage , 2021.

[2] A. Das, T. Kundu, and C. Saravanan, *Dimensionality Reduction for Handwritten Digit Recognition* . [Online]. Available: https://www.proquest.com/docview/2305560343?pq-origsite=primo

[3] C. M. Bishop, *Pattern Recognition and Machine Learning* , 2006.

[4] P. D. Grunwald, *The Minimum Description Length Principle* , 2019.

[5] C. Plan, *Dependency Clustering Across Measurement Scales* .

[6] A. Zimek, *Correlation Clustering* , 2008.

[7] M. J. Zaki, J. X. Yu, B. Ravindran, and V. Pudi (Eds.), *Advances in Knowledge Discovery and Data Mining* , 2010.