technische universität
dortmund

Masterarbeit

Veröffentlichung von häufig vorkommenden Mustern in relationalen Datenbanken unter Vertraulichkeitsanforderungen

Andrej Dudenhefner
28. Oktober 2013

Gutachter:
Prof. Dr. Joachim Biskup
Dipl.-Inf. Cornelia Tadros

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl VI Informationssysteme und Sicherheit (ISSI)
http://ls6-www.informatik.uni-dortmund.de/

Inhaltsverzeichnis

Kapitel 1

Einleitung

Wenn auf **sensible Daten** Methoden des **Data Mining** angewendet und deren Ergebnisse veröffentlicht werden, dann müssen diese Ergebnisse den **Sicherheitsinteressen** [5] der Beteiligten genügen. Um **Vertraulichkeit** zu gewährleisten, dürfen nur autorisierte Personen zweckgebunden auf sensible Daten zugreifen. Um **Anonymität** zu gewährleisten, darf trotz Autorisation für Teile der Daten keine Möglichkeit bestehen, einzelne Individuen einer Anonymitätsklasse zu identifizieren. Die Berücksichtigung dieser Sicherheitsinteressen bei der Anwendung der Methoden des **Data Mining** wird unter dem Begriff **Privacy Preserving Data Mining** [1] aufgefasst.

Diese Arbeit beschäftigt sich mit der Anonymisierung von häufig vorkommenden Mustern im Zusammenhang mit Assoziationsregeln in relationalen Datenbanken. Dabei werden Datenbanken betrachtet, die sensible Daten über Personen beinhalten. Genauer gesagt repräsentiert jedes Tupel einer Tabelle innerhalb der betrachteten Datenbanken das Verhalten bzw. Eigenschaften einer Person.

Die Anonymisierung der gesamten Datenbank vor der Ausführung des Data Mining, auch als »**Anonymize-and-Mine**« [1] bezeichnet, wird hier nicht betrachtet. Stattdessen werden nur die zu veröffentlichenden Data Mining Ergebnisse anonymisiert. Diese als »**Mine-and-Anonymize**« [1] bezeichnete Vorgehensweise hat den Vorteil, dass nur die mit den Data Mining Ergebnissen im Zusammenhang stehenden Teile der Datenbank betrachtet werden müssen.

Ausgehend von der Anonymisierung klassischer Assoziationsregeln [3] und bisherigen Methoden des Privacy Preserving Data Mining werden in dieser Arbeit mehrere Arten von Assoziationsregeln für relationale Datenbanken in Bezug auf deren Anonymisierung untersucht. Insbesondere spielt bei dieser Betrachtung der Begriff der k-**Anonymität** [11] eine zentrale Rolle, da diese Eigenschaft eine Identifikation von Gruppen mit weniger als k Personen verhindern soll. Potentielle Verletzungen von Vertraulichkeitsanforderungen werden dabei über einen entsprechenden Begriff des k-Inferenzkanals beschrieben. Dabei

soll a priori Wissen eines potentiellen Angreifers über die zugrundeliegende Datenbank berücksichtigt werden.

Aufbau der Arbeit

Diese Arbeit besteht aus zwei Teilen.

Im ersten Teil (Kapitel 1-2) werden die in dieser Arbeit benötigten Grundlagen zusammengestellt. In Abschnitt 2.1 werden vorhandene Ergebnisse zur Anonymisierung klassischer Assoziationsregeln in binären Datenbanken genannt. In Abschnitt 2.2 werden betrachtete Typen von Assoziationsregeln in relationalen Datenbanken beschrieben. In Abschnitt 2.2.3 werden Konzepte, die im Zusammenhang mit der Identifizierung von Personen im relationalen Datenbankmodell stehen, zusammengetragen.

Im zweiten Teil (Kapitel 3-4) werden die im ersten Teil beschriebenen Begriffe und Konzepte zur Anonymisierung auf Assoziationsregeln in relationalen Datenbanken übertragen sowie neue Ansätze dafür formuliert. In Abschnitt 3.1 werden Annahmen über einen potentiellen Angreifer sowie über betrachtete Datenbanken zusammengestellt. In Abschnitt 3.2 werden intra-dimensionale Assoziationsregeln auf ihre Anonymisierbarkeit untersucht. Dabei werden Ergebnisse aus dem binären Datenbankmodell direkt übertragen und ihre Übertragbarkeit bewiesen. In Abschnitt 3.3 werden inter-dimensionale Assoziationsregeln auf ihre Anonymisierbarkeit untersucht. Dabei werden neue Ansätze (inter-dimensionaler Ableitungsoperator) formuliert und untersucht.

Der Beitrag dieser Arbeit ist die Übertragung der Begriffe und Konzepte zur Anonymisierung von dem binären in das relationale Modell in Abschnitt 3.2 sowie die Formulierung und Untersuchung eigener Ansätze in Abschnitt 3.3.

Kapitel 2

Grundlagen und Stand der Technik

2.1 Muster auf binären Datenbanken

2.1.1 Klassische Assoziationsregelanalyse

Im Bereich des Data Mining arbeitet die **klassische Assoziationsregelanalyse** [2] auf **binären Datenbanken**. Dabei werden Zusammenhänge zwischen Itemmengen erkannt und mittels **Assoziationsregeln** beschrieben. Dieser Abschnitt bietet einen Überblick über die zentralen Begriffe der klassischen Assoziationsregelanalyse.

Definition 2.1.1 (Item, Transaktion, binäre Datenbank).
Sei $\mathcal{I} = \{i_1, \ldots, i_p\}$ eine Menge von binären Variablen,
$\mathcal{T} = \{t_1, \ldots, t_n\}$ eine nullvektorfreie Multimenge von p-dimensionalen Binärvektoren.

Bezeichne
eine binäre Variable $i \in \mathcal{I}$ als **Item**,
einen p-dimensionalen Binärvektor über den Variablen $\mathcal{I}$ als
Transaktion bzw. **Itemmenge**,
$\mathcal{D} = (\mathcal{I}, \mathcal{T})$ als **binäre Datenbank**.

2.1.2 Bemerkung. Die Bedingung, dass eine binäre Datenbank keine leeren Transaktionen enthalten darf, ist keine semantische Einschränkung. Mit Hilfe eines neuen Items i und dem Erweitern jeder Transaktion um i kann die leere Transaktion durch die Transaktion, die nur i enthält, modelliert werden.

Für eine Transaktion $t = (1, 0, 1)$ (auf Variablen a,b,c) oder die entsprechende Itemmenge wird im Folgenden die äquivalente Notationen als Menge $\{a, c\}$ verwendet, um je nach Kontext unnötige Syntax zu minimieren.

Definition 2.1.3 (Assoziationsregel).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
$X, Y \in 2^{\mathcal{I}}$ Itemmengen mit $X \cap Y = \emptyset$.
Bezeichne $X \to Y$ als **Assoziationsregel**.

Intuitiv gesprochen, soll die Assoziationsregel $X \to Y$ den folgenden Zusammenhang beschreiben:
wenn alle Items der Menge X in einer Transaktion vorkommen, dann liegt es nahe zu vermuten, dass alle Items der Menge Y in ihr ebenfalls vorkommen.
Relevanz und Stärke einer Assoziationsregel werden durch ihren **Support** und **Confidence** beschrieben.

2.1.4 Bemerkung. Um unnötige Komplikationen mit der Semantik von Multimengen zu vermeiden, wird im Folgenden statt einer Multimenge $\mathcal{T}$ die Menge $\text{set}(\mathcal{T}) = \{(i,t) \in \mathbb{N} \times \mathcal{T} \mid t \text{ kommt in } \mathcal{T} \text{ genau } n\text{-mal vor und } 0 < i \leq n\}$ verwendet, um Transaktionen mit bestimmten Eigenschaften zu zählen. Dabei wird der Index i zum Durchnummerieren gleicher Transaktionen verwendet.

Definition 2.1.5 (Support, Confidence).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
$X, Y \in 2^{\mathcal{I}}$ Itemmengen, $X \to Y$ eine Assoziationsregel.
Definiere

Support (absolut):	$\text{supp}_{\mathcal{D}}(X) = \lvert\{(i,t) \in \text{set}(\mathcal{T}) \mid X \subseteq t\}\rvert$,
	$\text{supp}_{\mathcal{D}}(X \to Y) = \text{supp}_{\mathcal{D}}(X \cup Y)$,
Confidence:	$\text{conf}_{\mathcal{D}}(X \to Y) = \frac{\text{supp}_{\mathcal{D}}(X \cup Y)}{\text{supp}_{\mathcal{D}}(X)}$.

Der Support einer Regel $\text{supp}_{\mathcal{D}}(X \to Y)$ entspricht der Anzahl beitragender Transaktionen der binären Datenbank $\mathcal{D}$, und ist damit ein Maß für die **Relevanz der Regel**. Die Confidence einer Regel $\text{conf}_{\mathcal{D}}(X \to Y)$ entspricht der Wahrscheinlichkeit für das Vorkommen aller Items aus Y in einer Transaktion aus $\mathcal{D}$, die alle Items aus X enthält. Die Confidence ist deshalb ein Maß für die **Stärke der Regel**.

Da für eine relevante Assoziationsregel $X \to Y$ ein ausreichend hoher Support s notwendig ist, d. h. $\text{supp}_{\mathcal{D}}(X \to Y) \geq s$, sind **$s$-häufige Itemmengen**, d. h. Itemmengen X' mit $\text{supp}_{\mathcal{D}}(X') \geq s$, Grundbausteine von Assoziationsregeln.

Definition 2.1.6 (s-häufige Itemmenge).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank, $s \in \mathbb{N}$ minimaler Support.
Bezeichne eine Itemmenge $X \in 2^{\mathcal{I}}$ als **s-häufig in $\mathcal{D}$**, falls $\text{supp}_{\mathcal{D}}(X) \geq s$.

Für eine Assoziationsregel $X \to Y$ mit $\text{supp}_{\mathcal{D}}(X \to Y) \geq s$, ist die Itemmenge $X \cup Y$ s-häufig in $\mathcal{D}$, da nach Definition 2.1.5 $\text{supp}_{\mathcal{D}}(X \cup Y) = \text{supp}_{\mathcal{D}}(X \to Y) \geq s$ gilt. Da jede Transaktion in $\mathcal{D}$, die $X \cup Y$ enthält, die Itemmenge X enthält, folgt $\text{supp}_{\mathcal{D}}(X) \geq \text{supp}_{\mathcal{D}}(X \cup Y) \geq s$. Damit ist X auch s-häufig in $\mathcal{D}$. Nach Definition 2.1.5 setzt sich der Support und die Confidence der Assoziationsregel $X \to Y$ aus den Supports der Itemmengen X und $X \cup Y$ zusammen. Damit lässt sich aus $\text{supp}_{\mathcal{D}}(X \to Y)$ und $\text{conf}_{\mathcal{D}}(X \to Y)$ nicht mehr Wissen gewinnen als aus den s-häufigen Itemmengen X und $X \cup Y$. Daher werden s-häufige Itemmengen betrachtet, um Sicherheitsanforderungen bei der Veröffentlichung von Assoziationsregeln mit einem Mindestsupport von s zu erfüllen.

2.1.2 k-Anonymität

Für Daten über Personen ist die **k-Anonymität** [11] eine Eigenschaft, die eine Identifikation von Gruppen mit weniger als k Personen verhindern soll. Abhängig davon, in welcher Form Daten veröffentlicht werden, z.B. als relationale Datenbank oder als Assoziationsregeln, sind sowohl die konkrete Definition der k-Anonymität als auch die Mittel, mit welchen sie erreicht werden kann, unterschiedlich. In diesem Abschnitt werden zentrale Definitionen und Ergebnisse zur Sicherstellung der k-Anonymität bei der Veröffentlichung häufiger Itemmengen [3] zusammengefasst.

Um den Begriff der Anonymität im Kontext klassischer Assoziationsregeln zu definieren, werden zunächst **Muster** eingeführt, die Transaktionsmengen einer binären Datenbank beschreiben und damit Itemmengen verallgemeinern.

Definition 2.1.7 (Muster).
Sei $\mathcal{I}$ eine Menge von binären Variablen.
Ein **Muster** auf $\mathcal{I}$ ist eine aussagenlogische Formel mit Variablen nur aus $\mathcal{I}$.

Definition 2.1.8 (Support auf Mustern).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
p ein Muster auf $\mathcal{I}$.
Definiere
Support: $\text{supp}_{\mathcal{D}}(p) = |\{(i,t) \in \text{set}(\mathcal{T}) \mid [\![p]\!]_t = true\}|$,
wobei $[\![p]\!]_t$ die Auswertung von p mit Belegung t ist.

Während eine Itemmenge lediglich diejenigen Transaktionen beschreibt, die alle Elemente aus dieser Itemmenge enthalten, kann ein Muster mittels der Negation auch Transaktionen beschreiben, die bestimmte Elemente nicht enthalten.

Definition 2.1.9 (k-anonymes Muster).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
$k \in \mathbb{N}$ eine Anonymitätsschwelle,
p ein Muster auf $\mathcal{I}$.
Bezeichne p als **k-anonym** in $\mathcal{D}$, falls $\text{supp}_{\mathcal{D}}(p) = 0$ oder $\text{supp}_{\mathcal{D}}(p) \geq k$.

Die Anonymitätsschwelle $k \in \mathbb{N}$ beschreibt, intuitiv gesprochen, die Mindestgröße der Anonymitätsklasse, hinter der sich einzelne, vom Muster beschriebene Transaktionen verbergen können. Ein nicht k-anonymes Muster garantiert diese Mindestgröße nicht und kann potentiell zur Verletzung der Anonymität führen.

Da die Datenbank nicht zusammen mit dem Ergebnis des Data Mining Vorgangs veröffentlicht wird, muss vor der Definition des Inferenzkanals eine Abstraktion von der vorliegenden Datenbank etabliert werden.

Definition 2.1.10 (Datenbankkompatibilität).
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
$S = \{(X_1, n_1), \ldots, (X_s, n_s)\} \subseteq 2^{\mathcal{I}} \times \mathbb{N}$ eine Menge von Itemmenge-Support-Paaren.
Bezeichne $\mathcal{D}$ als **mit S kompatibel**, falls
für alle $(X, n) \in S$ gilt: $\text{supp}_{\mathcal{D}}(X) = n$.

Auf diese Weise lassen sich alle Datenbanken betrachten, deren Analyse zu den gegebenen Itemmenge-Support-Paaren geführt haben könnte.

Definition 2.1.11 (Supportinferenz).
Sei p ein Muster auf $\mathcal{I}$,
$k \in \mathbb{N}$ eine Supportschwelle,
$S = \{(X_1, n_1), \ldots, (X_s, n_s)\} \subseteq 2^{\mathcal{I}} \times \mathbb{N}$ eine Menge von Itemmenge-Support-Paaren.
Schreibe $S \models 0 < \text{supp}(p) < k$, falls für alle binären Datenbanken $\mathcal{D}$ gilt:
wenn $\mathcal{D}$ mit S kompatibel ist, dann gilt $0 < \text{supp}_{\mathcal{D}}(p) < k$.

Mit dem Begriff der Supportinferenz ist es möglich, bei gegebenen Itemmenge-Support-Paaren über den Support von Mustern zu sprechen, ohne über die tatsächlich zugrundeliegende Datenbank zu verfügen.

Definition 2.1.12 (k-Inferenzkanal).
Sei $k \in \mathbb{N}$ eine Anonymitätsschwelle,
$S = \{(X_1, n_1), \ldots, (X_s, n_s)\} \subseteq 2^{\mathcal{I}} \times \mathbb{N}$ eine Menge von Itemmenge-Support-Paaren.
Bezeichne
S als **k-Inferenzkanal**, falls gilt:
es existiert ein Muster p auf $\mathcal{I}$ mit $S \models 0 < \text{supp}(p) < k$.

Ein k-Inferenzkanal lässt eine Aussage über ein nicht k-anonymes Muster zu, ohne Zugriff auf die tatsächliche Datenbank zu benötigen. Dies soll genau der Verletzung der k-Anonymität entsprechen. Die Anonymitätsschwelle k ist in der Praxis viel kleiner als jeder positive Support-Wert n_i eines Itemmenge-Support-Paares (X_i, n_i), sonst würde (X_i, n_i) bereits den Anforderungen der k-Anonymität nicht genügen.

Es sind sowohl Bedingungen an die Existenz von Inferenzkanälen innerhalb gegebener Itemmenge-Support-Paare als auch Methoden zu deren Erkennung und Beseitigung untersucht worden [3]. Bei der **Sanitisierung**, d. h. der Beseitigung von Inferenzkanälen, ist es wichtig, dass das sanitisierte Ergebnis sich aus einer hypothetischen binären Datenbank ergeben kann. Dieses Ziel wird erreicht, indem bei der Sanitisierung nur Änderungen vorgenommen werden, die dem Hinzufügen oder Löschen von Transaktionen aus der ursprünglichen Datenbank entsprechen.

2.2 Muster auf relationalen Datenbanken

2.2.1 Relationale Algebra

Eine **relationale Datenbank** ist eine Menge von **Tabellen**. Jede Tabelle $R(A_1, \ldots, A_n)$ besitzt einen eindeutigen **Namen** R und eine Menge von **Attributen** $\{A_1, \ldots, A_n\}$, wobei jedes Attribut A_i mit einer Menge $\mathrm{dom}(A_i)$, der **Domäne** des Attributs, verknüpft ist. Die Domäne eines Attributs A_i hängt nicht von Tabellen, in den A_i vorkommt, ab und kann sowohl endlich als auch unendlich sein.

Zudem beschreibt die Tabelle $R(A_1, \ldots, A_n)$ eine Menge der Tupel $\{t_1, \ldots, t_m\} \subseteq \mathrm{dom}(A_1) \times \ldots \times \mathrm{dom}(A_n)$ über den Domänen ihrer Attribute. Insbesondere sind im Folgenden keine Duplikate innerhalb der Menge der Tupel der Tabelle erlaubt. Falls die Attribute einer Tabelle im aktuellen Kontext nicht wesentlich sind, wird lediglich der Tabellenname verwendet. Die Notation $t \in R$ ($t \notin R$) soll beschreiben, dass das Tupel t in (nicht in) der Menge der Tupel von R enthalten ist.

Für Tabellen sind neben Operatoren der Mengenlehre weitere Operatoren, die zur Abfrage von Informationen dienen, definierbar. Die Menge dieser Operatoren wird als **relationale Algebra** bezeichnet. Die in dieser Arbeit verwendeten Operatoren der relationalen Algebra werden in diesem Abschnitt definiert.

Definition 2.2.1 (Vereinigung, Schnitt, Differenz, Teilmengenbeziehung).
Seien $R(A_1, \ldots, A_n), S(A_1, \ldots, A_n)$ Tabellen.

Definiere
- **Vereinigung:** $R \cup S = \{t \mid t \in R \vee t \in S\}$,
- **Schnitt:** $R \cap S = \{t \mid t \in R \wedge t \in S\}$,
- **Differenz:** $R \setminus S = \{t \mid t \in R \wedge t \notin S\}$.

Schreibe $R \subseteq S$, falls für alle $t \in R$ gilt: $t \in S$.

Definition 2.2.2 (Einschränkung).
Seien $A_1, \ldots, A_n$ Attribute,
$t = (d_1, \ldots, d_n) \in \mathrm{dom}(A_1) \times \ldots \times \mathrm{dom}(A_n)$ ein Tupel,
$A_j \in \{A_1, \ldots, A_n\}$ ein Attribut,
$X = (A_{i_1}, \ldots, A_{i_m})$ Attribute aus $\{A_1, \ldots, A_n\}$.

Definiere
- **Komponentenauswahl:** $t(A_j) = d_j$,
- **Einschränkung:** $t\big|_X = (t(A_{i_1}), \ldots, t(A_{i_m}))$.

Definition 2.2.3 (Projektion).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$X = (A_{i_1}, \ldots, A_{i_m})$ Attribute aus $\{A_1, \ldots, A_n\}$.

Definiere **Projektion:** $\pi_X(R) = \{t\big|_X \mid t \in R\}$.

Da die Reihenfolge der Attribute einer Tabelle keine Rolle spielt, werden im Folgenden auch Mengen von Attributen als Parameter der Einschränkung bzw. Projektion verwendet.

Definition 2.2.4 (A=c-Selektion).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$X = (A_{i_1}, \ldots, A_{i_m})$ Attribute aus $\{A_1, \ldots, A_n\}$,
$x = (c_{i_1}, \ldots, c_{i_m}) \in \prod_{j=1}^{m} \mathrm{dom}(A_{i_j})$ Elemente der Domänen der Attribute aus X.
Definiere **A=c-Selektion:** $\sigma_{X=x}(R) = \sigma_{A_{i_1}=c_{i_1},\ldots,A_{i_m}=c_{i_m}}(R)$
$= \{t \mid t \in R \wedge t(A_{i_1}) = c_{i_1} \wedge \ldots \wedge t(A_{i_m}) = c_{i_m}\}$.

2.2.2 Assoziationsregelanalyse

Während die klassische Assoziationsregelanalyse ausschließlich mit binären Datenbanken arbeitet, wurde das Konzept der **häufigen Muster** und **Assoziationsregeln** auf relationale Datenbanken übertragen [10, 8]. Es lassen sich verschiedene Assoziationsregeltypen auf relationalen Datenbanken definieren. Jeder Assoziationsregeltyp besitzt eine eigene Syntax und konkrete Definitionen des **Supports** und der **Confidence**, die analog zu klassischen Assoziationsregeln die Relevanz und Stärke der Regel beschreiben. Innerhalb einzelner Assoziationsregeltypen existieren wiederum verschiedene Beschreibungsansätze und Maße für diesen Assoziationsregeltypen. In Folgenden wird eine Auswahl verschiedener Assoziationsregeltypen zusammenfassend vorgestellt.

Zur beispielhaften Darstellung verschiedener Assoziationsregeltypen wird im Folgenden die Tabelle Markt (Abbildung 2.1) verwendet.

Abbildung 2.1: Tabelle Markt($TID, Kunde, Objekt, Jahreszeit$)

TID	Kunde	Objekt	Jahreszeit
0	0	a	s
1	0	b	s
2	0	d	s
3	0	c	f
4	1	a	w
5	1	b	w
6	1	d	w
7	1	c	f
8	2	c	s
9	3	c	f
10	3	c	f

Intra-dimensionale Assoziationsregeln

Analog zu klassischen Assoziationsregeln beschreiben **intra-dimensionale Assoziationsregeln** [8] Zusammenhänge innerhalb eines Attributs.

Definition 2.2.5 (Intra-dimensionale Assoziationsregel).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,

$Z \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen, bezüglich welchen die Relevanz der Regel bestimmt wird,

$A \in \{A_1, \ldots, A_n\} \setminus Z$ ein Attribut,

$X, Y \subseteq \mathrm{dom}(A)$ Mengen von Ausprägungen von A.

Bezeichne $X \rightarrow_Z^A Y$ als **intra-dimensionale Assoziationsregel auf A bezüglich Z**.

Definition 2.2.6 (Support, Confidence für intra-dimensionale Assoziationsregeln).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,

$Z \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen,

$A \in \{A_1, \ldots, A_n\} \setminus Z$ ein Attribut,

$X \rightarrow_Z^A Y$ eine intra-dimensionale Assoziationsregel auf A bezüglich Z.

Definiere

Support: $\mathrm{supp}^A_{(R,Z)}(X) = |\bigcap_{x \in X} \pi_Z(\sigma_{A=x}(R))|$,

$\mathrm{supp}_R(X \rightarrow_Z^A Y) = \mathrm{supp}^A_{(R,Z)}(X \cup Y)$,

Confidence: $\mathrm{conf}_R(X \rightarrow_Z^A Y) = \frac{\mathrm{supp}^A_{(R,Z)}(X \cup Y)}{\mathrm{supp}^A_{(R,Z)}(X)}$.

Der Support und die Confidence einer intra-dimensionalen Assoziationsregel werden bezüglich einer Menge Z von Attributen bestimmt. Für eine intra-dimensionale Assoziationsregel auf A entspricht diese Menge in der Literatur [10] allen Attributen der Tabelle bis auf A.

2.2.7 Beispiel. Die Regel »$\{a, b\} \rightarrow^{Objekt}_{\{Kunde\}} \{d\}$« auf *Objekt* bezüglich $\{Kunde\}$ am Beispiel der Tabelle Markt (Abbildung 2.1) drückt aus, dass Kunden, die Objekte a und b gekauft haben, auch das Objekt d gekauft haben. Es gilt:
$\mathrm{supp}_{\mathrm{Markt}}(\{a, b\} \rightarrow^{Objekt}_{\{Kunde\}} \{d\}) = |\{0, 1\}| = 2$,
$\mathrm{conf}_{\mathrm{Markt}}(\{a, b\} \rightarrow^{Objekt}_{\{Kunde\}} \{d\}) = \frac{|\{0,1\}|}{|\{0,1\}|} = \frac{2}{2}$.

Klassische Assoziationsregeln auf binären Datenbanken sind ein Spezialfall intra-dimensionaler Assoziationsregeln. Die Begründung dazu liefert die folgende Bemerkung 2.2.8.

2.2.8 Bemerkung.
Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,
$X, Y \in 2^{\mathcal{I}}$ Itemmengen, $X \to Y$ eine Assoziationsregel.
Konstruiere
ein Attribut T mit der Domäne $\mathrm{dom}(T) = \mathrm{set}(\mathcal{T})$,
ein Attribut I mit der Domäne $\mathrm{dom}(I) = \mathcal{I}$,
eine Tabelle $R(T, I)$ mit der Tupelmenge $\{((i, t), j) \in \mathrm{set}(\mathcal{T}) \times \mathcal{I} \mid j \in t\}$.
Es gilt
$\mathrm{supp}_{\mathcal{D}}(X \to Y) = \mathrm{supp}_R(X \to^I_{\{T\}} Y)$.

Der Beweis der in der Bemerkung 2.2.8 aufgestellten Behauptung wird erst durch Lemma 3.2.16 (ii) mit $\mathcal{U} = \mathrm{set}(\mathcal{T})$ und f =Identität geliefert, und ist an dieser Stelle nicht von Bedeutung.

2.2.9 Beispiel. Die binäre Datenbank $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ mit $\mathcal{I} = \{a, b, c\}, \mathcal{T} = [\{a, b\}, \{a, b\}, \{a\}]$ und $\mathrm{set}(\mathcal{T}) = \{(1, \{a, b\}), (2, \{a, b\}), (1, \{a\})\}$ kann als Tabelle $R(T, I)$ (Abbildung 2.2) beschrieben werden.

Abbildung 2.2: Tabelle $R(T, I)$

T	I
$(1, \{a, b\})$	a
$(1, \{a, b\})$	b
$(2, \{a, b\})$	a
$(2, \{a, b\})$	b
$(1, \{a\})$	a

Die klassische Assoziationsregel $\{a\} \to \{b\}$ mit $\mathrm{supp}_{\mathcal{D}}(\{a\} \to \{b\}) = 2$ kann als intra-dimensionale Assoziationsregel $\{a\} \to^I_{\{T\}} \{b\}$ mit $\mathrm{supp}_R(\{a\} \to^I_{\{T\}} \{b\}) = 2$ beschrieben werden.

Intra-dimensionale Assoziationsregeln werden analog zu klassischen Assoziationsregeln mittels **häufiger Itemmengen** bestimmt. Insbesondere kann aus einer Menge intra-dimensionaler Assoziationsregeln mit einem Mindestsupport von s nicht mehr Wissen über die zugrundeliegenden Daten inferiert werden als aus den zugrundeliegenden s-häufigen Itemmengen.

Definition 2.2.10 (s-häufige Menge).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$Z \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen, bezüglich welchen die Relevanz bestimmt wird,
$A \in \{A_1, \ldots, A_n\} \setminus Z$ ein Attribut,
$X \subseteq \mathrm{dom}(A)$ eine Menge von Ausprägungen von A,
$s \in \mathbb{N}$ minimaler Support.
Bezeichne X als **s-häufig bezüglich Z in R**, falls $\mathrm{supp}^A_{(R,Z)}(X) \geq s$.

Inter-dimensionale Assoziationsregeln

Inter-dimensionale Assoziationsregeln beschreiben Zusammenhänge zwischen verschiedenen Attributen.

Definition 2.2.11 (Inter-dimensionale Assoziationsregel).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$\{A_{i_1}, \ldots, A_{i_{m+l}}\} \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$c_1 \in \mathrm{dom}(A_{i_1}), \ldots, c_{m+l} \in \mathrm{dom}(A_{i_{m+l}})$ Ausprägungen der entsprechenden Attribute,
$Z \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen, bezüglich welchen die Relevanz der Regel bestimmt wird.
Bezeichne $\bigwedge_{j=1}^{m} (A_{i_j} = c_j) \rightarrow_Z \bigwedge_{j=m+1}^{m+l} (A_{i_j} = c_j)$
als **inter-dimensionale Assoziationsregel bezüglich Z**.

Definition 2.2.12 (Support, Confidence für inter-dimensionale Assoziationsregeln).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$X \rightarrow_Z Y$ eine inter-dimensionale Assoziationsregel bezüglich Z mit
$X = \bigwedge_{j=1}^{m} (A_{i_j} = c_j)$ und $Y = \bigwedge_{j=m+1}^{m+l} (A_{i_j} = c_j)$.
Definiere
Support: $\mathrm{supp}_R(X \rightarrow_Z Y) = |\pi_Z(\sigma_{A_{i_1}=c_1,\ldots,A_{i_{m+l}}=c_{m+l}}(R))|$,
Confidence: $\mathrm{conf}_R(X \rightarrow_Z Y) = \frac{\mathrm{supp}_R(X \rightarrow_Z Y)}{|\pi_Z(\sigma_{A_{i_1}=c_1,\ldots,A_{i_m}=c_m}(R))|}$.

2.2.13 Beispiel. Die Regel »$Objekt = c \rightarrow_{\{Kunde\}} Jahreszeit = f$« am Beispiel der Tabelle Markt (Abbildung 2.1) drückt aus, dass das Objekt c im Frühling gekauft wird. Es gilt $\mathrm{supp}_{\mathrm{Markt}}(Objekt = c \rightarrow_{\{Kunde\}} Jahreszeit = f) = |\{0,1,3\}| = 3$ und $\mathrm{conf}_{\mathrm{Markt}}(Objekt = c \rightarrow_{\{Kunde\}} Jahreszeit = f) = \frac{|\{0,1,3\}|}{|\{0,1,2,3\}|} = \frac{3}{4}$.

Im Gegensatz zu intra-dimensionalen Assoziationsregeln sind mehrfache Vorkommen eines Attributs mit unterschiedlichen Ausprägungen innerhalb einer inter-dimensionalen Assoziationsregel gemäß Definition 2.2.11 nicht sinnvoll, da die A=c-Selektion in diesem Fall eine widersprüchliche Eingabe erhalten würde.

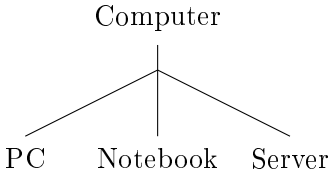
Erweiterungen

Im Gegensatz zu inter-dimensionalen Assoziationsregeln, die mehrfaches Vorkommen eines Attributes innerhalb einer Assoziationsregel verbieten, dürfen Attribute in **hybriden Assoziationsregeln** [10, 6] mehrfach vorkommen.

Mit Hilfe einer Taxonomie, die Domänen einzelner Attribute hierarchisch klassifiziert, können **mehrstufige Assoziationsregeln** [8] Klassen von Ausprägungen beschreiben statt lediglich einzelne Ausprägungen. Dazu wird der Selektionsoperator der relationalen Algebra bezüglich der jeweiligen Taxonomie erweitert. Eine Aussage $Attribut = Klasse$ im Parameter der Selektion bedeutet, dass die Ausprägung von $Attribut$ des selektierten Tupels ein Blatt im Teilbaum der Taxonomie mit der Wurzel $Klasse$ ist.

2.2.14 Beispiel. Sei eine Taxonomie für Computer (Abbildung 2.3) gegeben.
Für eine Tabelle, die von Kunden gekaufte Produkte mit Angabe ihres Alters beschreibt, lässt sich eine hybride mehrstufige Assoziationsregel
»$Alter = [20..30]$, $Produkt = Computer \rightarrow_{\{Kunde\}} Produkt = Notebook$«
formulieren. Diese Regel sagt aus, dass die meisten Computer, die von Zwanzig- bis Dreißigjährigen gekauft werden, Notebooks sind. Diese Regel ist »hybrid«, da das Attribut »Produkt« innerhalb der Regel mehrfach vorkommt. Diese Regel ist »mehrstufig«, da (mehrere) Abstraktionsebenen eines Attributs innerhalb der Regel vorkommen. Der Support einer solchen Regel entspricht der Anzahl der zwanzig- bis dreißigjährigen Kunden, die Computer, d. h. PCs, Notebooks oder Server, gekauft haben. Die Confidence entspricht dem Verhältnis der zwanzig- bis dreißigjährigen Kunden, die Notebooks gekauft haben, zu denjenigen zwanzig- bis dreißigjährigen Kunden, die Computer gekauft haben.

Abbildung 2.3: Taxonomie für Computer



2.2.3 Identifizierung von Personen

In dieser Arbeit werden Tabellen betrachtet, die vertrauliche Informationen über Personen beinhalten. Genauer gesagt werden Tabellen betrachtet, deren Tupel jeweils das Verhalten bzw. Eigenschaften einer Person repräsentieren. Daher soll jedes Tupel durch Experten wieder derjenigen Person, über die Informationen innerhalb des Tupels enthalten sind, zugeordnet werden können. Einer Person dürfen innerhalb einer Tabelle beliebig viele Tupel zugeordnet werden können. Die betrachtete Menge von Personen $\mathrm{dom}(\mathcal{U})$ (des dazugehörigen Attributs $\mathcal{U}$) wird als **Population** bezeichnet. Die Menge $\mathrm{dom}(\mathcal{U})$ kann unendlich sein

und es wird nicht verlangt, dass eine Tabelle Informationen über jede Person $u \in \text{dom}(\mathcal{U})$ enthält.

Im Folgenden werden alle Tabellen im Kontext einer festen Population $\text{dom}(\mathcal{U})$ betrachtet. Deshalb wird auf die explizite Erwähnung von $\text{dom}(\mathcal{U})$ in Prämissen verzichtet.

Tabellen können sowohl in der Form $R(A_1, \ldots, A_n)$ als auch in der Form $R(\mathcal{U}, A_1, \ldots, A_n)$ vorliegen. Dabei entspricht die Ausprägung von $\mathcal{U}$ für ein Tupel genau derjenigen Person, über die Informationen innerhalb des Tupels enthalten sind.

Für eine Population wird das a priori Wissen eines potentiellen Angreifers über eine Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$ als eine Menge von Tabellen, die er für möglich hält, über den Attributen $\mathcal{U}, A_1, \ldots, A_n$ repräsentiert.

Bei der Definition von Mengen von Tabellen wird ebenfalls die Notation $R(\mathcal{U}, A_1, \ldots, A_n)$ verwendet. In diesem Fall ist R lediglich ein lokaler Platzhalter für einen eindeutigen Namen der Tabelle.

Definition 2.2.15 (A priori Wissen).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen.
Bezeichne

$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}} = \{R(\mathcal{U}, A_1, \ldots, A_n) \mid$ Ein potentieller Angreifer hält $R(\mathcal{U}, A_1, \ldots, A_n)$ für möglich$\}$ als **a priori Wissen über** $\{A_1, \ldots, A_n\}$ **in** $\mathcal{U}$.

Es wird angenommen, dass ein potentieller Angreifer die Domänen $\text{dom}(\mathcal{U}), \text{dom}(A_1), \ldots, \text{dom}(A_n)$ kennt. Über die Definition 2.2.15 hinaus werden Einschränkungen an das a priori Wissen formuliert, um im Folgenden Unwissenheit über Teile der ursprünglichen Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$ zu modellieren.

Der potentielle Angreifer hat kein Wissen über welche Personen Informationen in der ursprünglichen Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$ vorhanden sind, d. h. für jede Tabelle $R'(\mathcal{U})$ existiert eine Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $R' = \pi_{\mathcal{U}}(R'')$. Insbesondere ist $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ nicht leer, da ein potentieller Angreifer eine leere Tabelle für möglich hält.

Attribute von Tabellen werden im Folgenden in drei disjunkte Gruppen unterteilt: **sensible Attribute**, **identifizierende Attribute** und **sonstige Attribute**.

Sensible Attribute sind genau die »zu schützenden« Attribute, d. h. der Zusammenhang zwischen Personen und Ausprägungen sensibler Attribute soll Vertraulichkeitsanforderungen genügen. Es wird angenommen, dass ein potentieller Angreifer kein a priori Wissen über die Ausprägungen sensibler Attribute von Personen bzw. Personengruppen besitzt. Mit anderen Worten sind für ein sensibles Attribut A alle Zuordnungen zwischen $\text{dom}(\mathcal{U})$ und $\text{dom}(A)$ aus der Sicht eines potentiellen Angreifers möglich. Für das a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $A \in \{A_1, \ldots, A_n\}$ gilt daher: für jede Tabelle $R'(\mathcal{U}, A)$ existiert eine Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $R' = \pi_{(\mathcal{U},A)}(R'')$.

Identifizierende Attribute (z. B. Name, Postleitzahl oder Geschlecht) tragen Informationen, die nicht nur zur Identifizierung von Personen genutzt werden können, sondern

auch über deren Ausprägungen ein potentieller Angreifer a priori Wissen besitzen kann [9]. Mit anderen Worten besitzt ein potentieller Angreifer zu jedem identifizierenden Attribut A die Tabelle $R'(\mathcal{U}, A)$ über die gesamte Population $\text{dom}(\mathcal{U})$. Für das a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $A \in \{A_1, \ldots, A_n\}$ gilt daher: für jede Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ gilt $\pi_{(\mathcal{U},A)}(R'') \subseteq R'$. Es ist sowohl erlaubt, dass einer Person innerhalb einer Tabelle mehrere Ausprägungen eines identifizierendes Attributs zugeordnet werden, als auch dass eine Ausprägung eines identifizierenden Attributs innerhalb einer Tabelle mehreren Personen zugeordnet wird.

Sonstige Attribute sind diejenigen, in die in keine der beiden anderen Kategorien eingeordnet werden können. Es kann angenommen werden, dass ein potentieller Angreifer über a priori Wissen über Ausprägungen sonstiger Attribute verfügt, jedoch es nicht zur Identifizierung von Personen benutzen kann.

Es wird angenommen, dass die Charakterisierung der Attribute für eine konkrete Population für jede Tabelle von Experten vorgenommen wird.

Die genannten einschränkenden Annahmen für das a priori Wissen werden in Abschnitt 3.1.1 aufgelistet.

Identifikator

Eine Menge von identifizierenden Attributen, über deren Ausprägungen einzelne Personen im Kontext einer Tabelle und des entsprechenden a priori Wissens eindeutig identifiziert werden können, wird als **Identifikator** bezeichnet. Beispiele für Identifikatoren sind unter anderem Kreditkartennummer oder Mobiltelefonnummer.

Definition 2.2.16 (Identifikator).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,

$X = \{A_{i_1}, \ldots, A_{i_m}\} \subseteq A_1, \ldots, A_n$ eine Menge von identifizierenden Attributen,

$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$

unter Annahmen aus Abschnitt 3.1.1.

Bezeichne

X als **Identifikator im Kontext** $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, falls

für jede Tabelle $R'(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, gilt:

$$\forall (d_1, \ldots, d_m) \in \prod_{j=1}^{m} \text{dom}(A_{i_j}) : |\pi_{\mathcal{U}}(\sigma_{A_{i_1}=d_1,\ldots,A_{i_m}=d_m}(R'))| \leq 1.$$

Die Eigenschaft, ein Identifikator zu sein, ist oft stark von der konkreten Population abhängig. Beispielweise ist die Matrikelnummer innerhalb der Population der Studierenden einer Universität eindeutig, während sie innerhalb der Population der Studierenden mehrerer Universitäten bereits kein Identifikator mehr ist. Zudem lassen sich mit Hilfe der Population Randfälle künstlich ausschließen. So ist die Mobiltelefonnummer wohl kein Identifikator für die deutsche Bevölkerung, da sich womöglich zwei Personen eine Mobil-

telefonnummer teilen. Durch den Ausschluss dieser Randfälle aus der Population wird die Mobiltelefonnummer zum Identifikator für den Großteil der Bevölkerung.

Im Hinblick auf Anonymität spielen Identifikatoren eine zentrale Rolle, da deren Ausprägungen einen direkten Bezug zu Personen sicherstellen.

Quasi-Identifikator

In der Literatur [11, 9, 4] werden Attributmengen, die zwar keine Identifikatoren sind, aber deren Ausprägungen einzelne Personen identifizieren könnten, als **Quasi-Identifikatoren** bezeichnet. Studien zufolge kann 87% der amerikanischen Bevölkerung anhand der Kombination von Postleitzahl, Geburtsdatum und Geschlecht eindeutig identifiziert werden.

Quasi-Identifikatoren ermöglichen eine Gefährdung der Anonymität, falls sie sowohl in öffentlich verfügbaren Informationsquellen in Verbindung mit Identifikatoren als auch in Quellen mit sensiblen Daten auftauchen und damit eine Verknüpfung zwischen sensiblen Daten und Identifikatoren ermöglichen.

Definition 2.2.17 (Quasi-Identifikator).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,

$X = \{A_{i_1}, \ldots, A_{i_m}\} \subseteq A_1, \ldots, A_n$ eine Menge von identifizierenden Attributen,
$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$
unter Annahmen aus Abschnitt 3.1.1.

Bezeichne

X als **Quasi-Identifikator im Kontext** $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, falls
für jede Tabelle $R'(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, gilt:
$\exists (d_1, \ldots, d_m) \in \prod_{j=1}^{m} \mathrm{dom}(A_{i_j}) : |\pi_{\mathcal{U}}(\sigma_{A_{i_1}=d_1,\ldots,A_{i_m}=d_m}(R'))| = 1.$

In der Literatur wird zur Definition eines Quasi-Identifikators eine konkrete Zuordnung zwischen der Population und den Ausprägungen der Attribute des Quasi-Identifikators [11] oder eine Reidentifikationsfunktion [4], die Ausprägungen der Attribute Personen zuordnet, verlangt. Die Definition 2.2.17 orientiert sich an [11, Definition 2]. Die Zuordnung zwischen der Population und den Ausprägungen der Attribute des Quasi-Identifikators wird durch das a priori Wissen berücksichtigt.

Die Bedingung der Existenz mindestens einer Ausprägung der Attribute des Quasi-Identifikators zur Identifizierung einer Person ist im Allgemeinen zu schwach. So kann bereits durch einen theoretischen Randfall ein Attribut als Quasi-Identifikator bezeichnet werden. Umgangssprachliche Formulierungen wie »... für den Großteil der Population gilt ... « lassen sich jedoch schwer innerhalb eines formalen Modells realisieren und sind zusätzlich abhängig vom Grad der Sicherheitsanforderungen. Daher werden Quasi-Identifikatoren in der Praxis von Experten bestimmt und die Definition 2.2.17 soll mindestens diese Quasi-Identifikatoren charakterisieren.

Jeder Identifikator ist im gleichen Kontext ein Quasi-Identifikator, daher werden im Folgenden zur Untersuchung der Anonymität Quasi-Identifikatoren nicht aber speziell Identifikatoren betrachtet.

2.2.4 k-Anonymität

In Kapitel 2.1.2 wurde bereits der Begriff der k-Anonymität im Kontext binärer Datenbanken definiert. Im Fall der Veröffentlichung einer relationalen Datenbank unter Vertraulichkeitsanforderungen wird k-Anonymität mittels Generalisierung und Unterdrückung [11] erreicht. Dabei heißt eine relationale Datenbank k-anonym, falls jede in ihr vorkommende Kombination der Ausprägungen eines Quasi-Identifikators in mindestens k Tupeln enthalten ist. Die Anonymisierung der gesamten Datenbank vor der Ausführung des Data Mining, auch als »Anonymize-and-Mine« [1] bezeichnet, wird hier nicht betrachtet. Stattdessen sollen Vertraulichkeitsanforderungen durch »Mine-and-Anonymize«, d. h. die Anonymisierung von Data Mining Ergebnissen, erreicht werden.

In der Literatur [1] werden bei einer Anonymisierung von Assoziationsregeln lediglich inter-dimensionale Assoziationsregeln ohne Projektion auf eine Attributmenge betrachtet. Diese Betrachtung erlaubt eine Darstellung der relationalen Datenbank als eine binäre Datenbank und macht damit die Ergebnisse aus [3] anwendbar. Dabei wird jede Kombination eines Attributs A mit dessen Ausprägung c als Item »$A = c$« und jedes Tupel $(c_1, \ldots, c_n)$ der Datenbank über den Attributen $(A_1, \ldots, A_n)$ als Transaktion $\{A_1 = c_1, \ldots, A_n = c_n\}$ dargestellt.

Definition 2.2.18.
Sei $R(A_1, \ldots, A_n)$ eine Tabelle.
Definiere

$\text{bin}(R(A_1, \ldots, A_n)) = (\mathcal{I}, \mathcal{T})$, wobei
$\mathcal{I} = \{A = c \mid A \in \{A_1, \ldots, A_n\}, c \in \text{dom}(A)\}$,
$\mathcal{T} = \{\{A_1 = c_1, \ldots, A_n = c_n\} \mid (c_1, \ldots, c_n) \in R\}$.

Das folgende Beispiel 2.2.19 zeigt, dass dieser Ansatz nicht zur Anonymisierung von Data Mining Ergebnissen auf relationalen Datenbanken genügt.

2.2.19 Beispiel. Für eine Tabelle $R(Kunde, Alter, Geschlecht, Produkt)$, die von Kunden gekaufte Produkte unter Angabe ihres Alters und Geschlechts beinhaltet mit $\{[0..39], [40..]\} = \text{dom}(Alter)$, $\{m, w\} = \text{dom}(Geschlecht)$ und $Computer \in \text{dom}(Produkt)$, seien folgende inter-dimensionale Assoziationsregeln mit jeweiligem Support verfügbar:

(I) $\top \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 200,

(II) $Geschlecht = m \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 100,

(III) $Geschlecht = w \wedge Alter = [0..39] \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 99.

Umgangssprachlich lassen sich diese Daten auf folgende Weise interpretieren:

(i) 200 Kunden haben einen Computer gekauft,

(ii) 100 männliche Kunden haben einen Computer gekauft,

(iii) 99 weibliche Kunden im Alter von 0 bis 39 Jahren haben einen Computer gekauft.

Die Interpretation dieser Daten als Transaktionen einer binären Datenbank gemäß Definition 2.2.18 führt zu folgenden Itemmengen-Support-Paaren:
$(\{Produkt = Computer\}, 200)$,
$(\{Produkt = Computer, Geschlecht = m\}, 100)$,
$(\{Produkt = Computer, Geschlecht = w, Alter = [0..39]\}, 99)$.
Aus gegebenen Itemmengen-Support-Paaren folgt mittels der in [3] beschriebenen Inferenzmöglichkeiten nur ein nichttriviales Muster: »$Produkt = Computer \wedge \neg(Geschlecht = m)$« mit einem Support von 100.
Im binären Datenbankmodell ist keine semantische Beziehung zwischen den Items »$Geschlecht = m$« und »$Geschlecht = w$« vorhanden. Im relationalen Datenbankmodell mit $\mathrm{dom}(Geschlecht) = \{m, w\}$ gilt jedoch für ein Tupel $t \in R$ die Aussage:
$t(Geschlecht) = m$ genau dann, wenn $t(Geschlecht) \neq w$.
Unter der Annahme, dass ein Kunde entweder männlich oder weiblich sein muss, lässt sich »logisch« aus den gegebenen Daten mehr inferieren. Aus (i) und (ii) folgt, dass 100 weibliche Kunden einen Computer gekauft haben. Wenn laut (iii) 99 von ihnen im Alter von 0 bis 39 Jahren waren, dann hat **genau eine** Frau einen Computer gekauft, die mindestens 40 Jahre alt war. Das Vorhandensein dieser Schlussfolgerung würde die Vertraulichkeitsanforderung verletzen.

Dieses Beispiel zeigt, dass die Vernachlässigung von Semantik im binären Datenbankmodell dazu führen kann, dass vorhandene Inferenzkanäle nicht erkannt werden. Dieses Problem wird in Kapitel 3.3 näher untersucht.

Kapitel 3

Betrachtung unterschiedlicher Inferenzkanaltypen

In diesem Kapitel werden unterschiedliche Inferenzkanaltypen betrachtet, um potentielle Verletzungen der Anonymität zu definieren und ihnen entgegenzuwirken. Intra-dimensionale Assoziationsregeln unterscheiden sich wesentlich von inter-dimensionalen Assoziationsregeln. Daher werden **intra-dimensionale Muster** und **inter-dimensionale Muster** im Kontext der entsprechenden Assoziationsregeln als Übertragung des Begriffs des Muster für binäre Datenbanken eingeführt, um anschließend einen entsprechenden Begriff der k-Anonymität zu definieren.

3.1 Zusammenstellung der Annahmen

In diesem Abschnitt werden im Kapitel 3 getroffene Annahmen über gegebene Tabellen zusammengefasst. Dabei sollen einige Annahmen für das gesamte Kapitel 3, einige nur für Abschnitt 3.2 und einige nur für Abschnitt 3.3 gelten, was entsprechend kenntlich gemacht wird. Dabei erweitern die zusätzlichen Annehmen die globalen Annahmen und gelten jeweils nur für die entsprechenden Abschnitte.

3.1.1 Globale Annahmen für relationale Datenbanken

(i) $\mathcal{U}$ ist das Attribut, das die betrachtete Population $\mathrm{dom}(\mathcal{U})$ repräsentiert.

(ii) Die betrachtete Population $\mathrm{dom}(\mathcal{U})$ kann unendlich sein.

(iii) Die dem Data Mining Vorgang zugrundeliegende Tabelle hat die Form $R(A_1, \ldots, A_n)$.

(iv) Domänen der Attribute $A_1, \ldots, A_n$ sind endlich (gegebenenfalls diskretisiert).

(v) Die Attribute $A_1, \ldots, A_n$ sind in disjunkte Gruppen »sensibel«, »identifizierend« und »sonstig« eingeteilt.

(vi) Jedes Tupel $t \in R(A_1, \ldots, A_n)$ enthält Informationen zu einer Person $u \in \mathrm{dom}(\mathcal{U})$, sodass die Tabelle $R(A_1, \ldots, A_n)$ zu $R(\mathcal{U}, A_1, \ldots, A_n)$ mit Tupeln $(u, t(A_1), \ldots, t(A_n))$ erweitert werden kann.

(vii) Das a priori Wissen des potentiellen Angreifers wird durch $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ (Definition 2.2.15) modelliert.

(viii) Der potentielle Angreifer hat kein Wissen darüber, welche Personen in $R(\mathcal{U}, A_1, \ldots, A_n)$ vorkommen, d. h. für jede Tabelle $R'(\mathcal{U})$ existiert eine Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $R' = \pi_{\mathcal{U}}(R'')$. Insbesondere gilt dies für die leere Tabelle $R' = \emptyset$. Algebraisch ausgedrückt gilt: $\{\pi_{\mathcal{U}}(R'') \mid R'' \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}\} = 2^{\mathrm{dom}(\mathcal{U})}$.

(ix) Der potentielle Angreifer hat kein Wissen darüber, welche Ausprägungen eines sensiblen Attributs $A \in \{A_1, \ldots, A_n\}$ einzelne Personen haben, d. h. für jede Tabelle $R'(\mathcal{U}, A)$ existiert eine Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $R' = \pi_{(\mathcal{U},A)}(R'')$. Algebraisch ausgedrückt gilt: $\{\pi_{(\mathcal{U},A)}(R'') \mid R'' \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}\} = 2^{\mathrm{dom}(\mathcal{U})\times\mathrm{dom}(A)}$.

(x) Der potentielle Angreifer hat Wissen darüber, welche Ausprägungen eines identifizierenden Attributs $A \in \{A_1, \ldots, A_n\}$ einzelne Personen haben und kennt die Tabelle $R'(\mathcal{U}, A)$, d. h. für jede Tabelle $R''(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ gilt $\pi_{(\mathcal{U},A)}(R'') \subseteq R'$.

Zusätzliche Annahmen für Abschnitt 3.2

(xi) $R(\mathcal{U}, A_1, \ldots, A_n)$ darf für jede Person $u \in \mathrm{dom}(\mathcal{U})$ beliebig viele Tupel t mit $t(\mathcal{U}) = u$ enthalten.

Zusätzliche Annahmen für Abschnitt 3.3

(xii) $R(\mathcal{U}, A_1, \ldots, A_n)$ darf für jede Person $u \in \mathrm{dom}(\mathcal{U})$ höchstens ein Tupel t mit $t(\mathcal{U}) = u$ enthalten.

3.1.2 Globale Annahmen für binäre Datenbanken

(i) Die Transaktionen einer Transaktionsmenge $\mathcal{T}$ stehen in Beziehung zu konkretem Verhalten von Personen der betrachteten Population, d. h. es existiert eine Funktion $f\colon \mathrm{set}(\mathcal{T}) \to \mathrm{dom}(\mathcal{U})$, die eine Transaktion derjenigen Personen zuordnet, deren Verhalten die Transaktion repräsentiert.

(ii) Eine Person der betrachteten Population beeinflusst höchstens eine Transaktion der Transaktionsmenge [3], d. h. für die in (i) beschriebene Funktion $f\colon \mathrm{set}(\mathcal{T}) \to \mathrm{dom}(\mathcal{U})$ gilt: für alle $u \in \mathrm{dom}(\mathcal{U}) : |f^{-1}(u)| \leq 1$. Beispielsweise bedeutet es, dass jeder höchstens einmal einkauft. Diese Annahme impliziert $|\mathrm{dom}(\mathcal{U})| \geq |\mathrm{set}(\mathcal{T})|$.

3.2 Intra-dimensionale Muster

3.2.1 Übertragung der Grundbegriffe

Aufgrund der Ähnlichkeit zu klassischen Assoziationsregeln können Definitionen im Kontext der k-Anonymität aus Abschnitt 2.1.2 direkt auf die intra-dimensionale Assoziationsregelanalyse übertragen werden.

Ein **intra-dimensionale Muster** auf einem Attribut A ist eine aussagenlogische Formeln mit Variablen, die Ausprägungen von A entsprechen. Um die Syntax zu entlasten, werden Ausprägungen von Attributen mit gleichnamigen Variablen identifiziert.

Definition 3.2.1 (Intra-dimensionales Muster).
Sei A ein Attribut.
Ein **intra-dimensionales Muster auf** A ist eine aussagenlogische Formel mit Variablen aus $\mathrm{dom}(A)$.
Definiere $\mathbf{Pat}_{\mathrm{intra}}(\mathbf{A}) = \{p \mid p \text{ ist ein intra-dimensionales Muster auf } A\}$.

Definition 3.2.2 (Support von intra-dimensionalen Mustern).
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$Z = \{A_{i_1}, \ldots, A_{i_m}\}$ Attribute, bezüglicher welcher der Support bestimmt wird,
$A \in \{A_1, \ldots, A_n\} \setminus Z$ ein Attribut,
$p \in Pat_{\mathrm{intra}}(A)$ ein intra-dimensionales Muster auf A.
Definiere

$$\textbf{Auswertung:} \quad \mathrm{eval}^A_{(R,Z)}(p) = \begin{cases} \pi_Z(\sigma_{A=d}(R)) & \text{wenn } p = d \in \mathrm{dom}(A) \\ \mathrm{eval}^A_{(R,Z)}(p_1) \cap \mathrm{eval}^A_{(R,Z)}(p_2) & \text{wenn } p = p_1 \wedge p_2 \\ \mathrm{eval}^A_{(R,Z)}(p_1) \cup \mathrm{eval}^A_{(R,Z)}(p_2) & \text{wenn } p = p_1 \vee p_2 \\ \pi_Z(R) \setminus \mathrm{eval}^A_{(R,Z)}(p_1) & \text{wenn } p = \neg p_1 \end{cases},$$

$$\textbf{Support:} \quad \mathrm{supp}^A_{(R,Z)}(p) = |\mathrm{eval}^A_{(R,Z)}(p)|.$$

3.2.3 Bemerkung. Nach Definition 2.2.6 ist der Support einer Ausprägungsmenge D gleich dem Support desjenigen Musters, das der Konjunktion der Elemente aus D entspricht, d. h.
$\mathrm{supp}^A_{(R,Z)}(D) = \mathrm{supp}^A_{(R,Z)}(\bigwedge_{d \in D} d)$. Dabei spielt die Attributmenge Z in der Definition 3.2.2 dieselbe Rolle wie in der Definition 2.2.6.

Lemma 3.2.4.
Sei $R(A_1, \ldots, A_n)$ eine Tabelle,
$Z = \{A_{i_1}, \ldots, A_{i_m}\}$ Attribute, bezüglicher welcher der Support bestimmt wird,
$A \in \{A_1, \ldots, A_n\} \setminus Z$ ein Attribut,
$p \in Pat_{intra}(A)$ ein intra-dimensionales Muster auf A.
Es gilt

$$\mathrm{supp}^A_{(R,Z)}(p) = \mathrm{supp}^A_{(\pi_{(A_{i_1}, \ldots, A_{i_m}, A)}(R), Z)}(p).$$

Beweis.
Induktion nach der Struktur des intra-dimensionalen Musters mit Induktionsanfang
$\pi_Z(\sigma_{A=d}(R)) = \{z \in \prod_{A_i \in Z} \mathrm{dom}(A_i) \mid \exists t \in \pi_{Z \dot\cup \{A\}}(R) : t\Big|_Z = z \text{ und } t(A) = d\}$
$= \pi_Z(\sigma_{A=d}(\pi_{Z \dot\cup \{A\}}(R)))$. □

Der Support eines Musters wird bezüglich einer Attributmenge Z bestimmt und entspricht deshalb zwar der Anzahl erfasster Ausprägungen von Z, jedoch nicht immer der genauen Anzahl der vom Muster erfassten Personen. Die Problematik der Wahl von Z wird an Beispielen 3.2.5 und 3.2.6 erläutert.

3.2.5 Beispiel. Man betrachte eine Tabelle $R(Vorname, Nachname, Datum, Symptom)$, in der Symptome von Patienten gespeichert sind. Dabei kann ein Patient an einem Tag mehrere Symptome haben. Die Attribute $Q = \{Vorname, Nachname\}$ sind identifizierend, das Attribut $Symptom$ mit $\{a, b\} \subseteq \mathrm{dom}(Symptom)$ ist sensibel und das Attribut $Datum$ ist sonstig.
Es soll bekannt sein, dass Patienten das Symptom a noch zwei weitere Tage nach der ersten Beschwerde haben. Insbesondere soll gelten:
$(v, n, d, a) \in R$ impliziert $(v, n, d', a), (v, n, d'', a) \in R$ für zwei untereinander und von d unterschiedliche $d', d'' \in \mathrm{dom}(Datum)$.
Zusätzlich haben Patienten, die das Symptom a haben, das Symptom b entweder an jedem oder an keinem Datum, an dem sie das Symptom a haben. Insbesondere soll gelten:
$(v, n, d, a), (v, n, d', a), (v, n, d, b) \in R$ impliziert $(v, n, d', b) \in R$ und
$(v, n, d, a), (v, n, d', a) \in R, (v, n, d, b) \notin R$ impliziert $(v, n, d', b) \notin R$.
Darüber hinaus hat ein Patient genau einen Vor- und Nachnamen und teilt ihre Kombination mit keinem anderen Patienten.

Das Wissen über den Zusammenhang zwischen $Datum$ und $Symptom$ ist bezüglich Annahme (ix) in Abschnitt 3.1.1 erlaubt, da es bei keinem Patienten eine der möglichen Kombinationen der Symptome ausschließt. Mit anderen Worten gehören zum a priori Wissen des Angreifers $\mathcal{W}^{\mathcal{U}}_{\{Vorname, Nachname, Datum, Symptom\}}$ nur Tabellen, die die genannten Eigenschaften erfüllen.

Sei Folgendes für $Z = \{Vorname, Nachname, Datum\}$ bekannt:

- 15 Ausprägungen von Z stehen mit dem Symptom a in R,
  d. h. $\mathrm{supp}^{Symptom}_{(R,Z)}(\{a\}) = 15$.
- 12 Ausprägungen von Z stehen mit den Symptomen a und b in R,
  d. h. $\mathrm{supp}^{Symptom}_{(R,Z)}(\{a, b\}) = 12$.

Daraus lässt sich bereits folgendes ableiten:

- $15 - 12 = 3$ Ausprägungen von Z stehen mit dem Symptom a aber nicht mit b in R. Dies entspricht dem Muster $p_1 = a \wedge \neg b$ mit $\mathrm{supp}^{Symptom}_{R,Z}(p_1) = 3$.

Man könnte vermuten, dass die vom Muster p_1 betroffenen Patienten aufgrund des Supports von p_1 sich innerhalb einer Anonymitätsklasse der Größe 3 befinden. Dies ist unter Berücksichtigung des a priori Wissens jedoch nicht der Fall. Aus dem Support vom p_1 folgt, dass 3 Ausprägungen von $\{Vorname, Nachname, Datum\}$, die sich in mindestens einer Komponente voneinander unterschieden, mit dem Symptom a und nicht dem Symptom b in R stehen. Mit anderen Worten existieren paarweise mindestens in einer Komponente unterschiedliche $(v_1, n_1, d_1), (v_2, n_2, d_2), (v_3, n_3, d_3) \in \text{dom}(Vorname) \times \text{dom}(Nachname) \times \text{dom}(Datum)$ mit $(v_1, n_1, d_1, a) \in R, (v_2, n_2, d_2, a) \in R, (v_3, n_3, d_3, a) \in R$ und $(v_1, n_1, d_1, b) \notin R, (v_2, n_2, d_2, b) \notin R, (v_3, n_3, d_3, b) \notin R$.

Zusammen mit dem a priori Wissen folgt unmittelbar, dass **genau ein** Patient mit dem Vor- und Nachnamen (v_1, n_1) an zwei weiteren von d_1 unterschiedlichen Tagen das Symptom a und nicht das Symptom b hatte.

Insbesondere beschreibt das Muster p_1 genau einen Patienten.

Der Support von p_1 bezüglich $Z' = \{Vorname, Nachname\}$ ist $\text{supp}^{Symptom}_{R,Z'}(p_1) = 1$. Das macht deutlich, dass eine bessere Wahl der Attribute, bezüglich welcher der Support bestimmt wird, zu einer besseren Approximation der Anzahl der vom Muster betroffenen Personen führen kann.

Dieses Beispiel zeigt, dass der Support eines Musters p bezüglich einer »schlecht gewählten« Attributmenge Z nicht der Anzahl der tatsächlich vom Muster betroffenen Personen entsprechen muss. Gleichzeitig kann ein potentieller Angreifer mit Hilfe von a priori Wissen aus Supportwerten bezüglich Z die genaue Anzahl der vom Muster betroffenen Personen bestimmen.

Das Beispiel 3.2.5 lässt vermuten, dass die Wahl einer Menge von identifizierenden Attributen als diejenige Attributmenge, bezüglich welcher der Support bestimmt wird, bei der Berechnung der von einem Muster erfassten Personen hilfreich ist. Dies ist jedoch im allgemeinen nicht der Fall.

3.2.6 Beispiel. Man betrachte eine Tabelle $R(Geschlecht, L, Produkt)$, in der von Kunden unter Angabe ihres Geschlechts mit $\{m, w\} = \text{dom}(Geschlecht)$ gekaufte Produkte mit $\{a, b, c\} \subseteq \text{dom}(Produkt)$ gespeichert sind. Bei jedem Einkauf der Produkte $a_1, \ldots, a_m$ eines Kunden mit dem Geschlecht g wird eine neue Zufallszahl $l \in \{1, \ldots, 2^{32}\}$ erzeugt. Darauf werden die Tupel $(g, a_1, l), \ldots (g, a_m, l)$ in der Tabelle gespeichert.

Das Attribut L ist ist gemäß Abschnitt 2.2.3 nicht identifizierend. Ausprägungen von L sind keine Eigenschaften von Kunden, sondern zufällig gewählt und sind auch nicht Teil des a priori Wissens eines potentiellen Angreifers. Mit anderen Worten trägt eine zufällige Zahl keine Informationen über einen Kunden.

Das Attribut *Geschlecht* ist identifizierend und das Attribut *Produkt* ist sensibel.

Sei Folgendes für $Z = \{Geschlecht, L\}$ bekannt:

- 10 Ausprägungen von Z stehen mit dem Produkt a in R, darunter waren sowohl Männer als auch Frauen,
  d. h. $\text{supp}^{Produkt}_{(R,Z)}(\{a\}) = 10$ und $\text{supp}^{Produkt}_{(R,\{Geschlecht\})}(\{a\}) = 2$.
- 9 Ausprägungen von Z stehen mit den Produkten a und b in R, darunter waren sowohl Männer als auch Frauen,
  d. h. $\text{supp}^{Produkt}_{(R,Z)}(\{a, b\}) = 9$ und $\text{supp}^{Produkt}_{(R,\{Geschlecht\})}(\{a, b\}) = 2$.

Daraus lässt sich bereits folgendes ableiten:

- $10 - 9 = 1$ Ausprägung von Z steht mit dem Produkt a aber nicht mit b in R. Dies entspricht dem Muster $p_1 = a \wedge \neg b$ mit $\text{supp}^{Produkt}_{(R,Z)}(p_1) = 1$.

Offensichtlich gibt es genau einen Kunden, der bei genau einem Einkauf das Produkt a und nicht das Produkt b gekauft hat. Folglich kann die Veröffentlichung der obigen Informationen eine Verletzung von Vertraulichkeitsanforderungen darstellen.

Der Support von p_1 bezüglich der Menge identifizierender Attribute $Z' = \{Geschlecht\}$ ist $\text{supp}^{Produkt}_{(R,Z')}(p_1) = 2 - 2 = 0$. Es gibt also kein Geschlecht, das von p_1 beschrieben wird. Daraus folgt aber nicht, dass das Muster keine Person beschriebt.

Dieses Beispiel zeigt, dass der Support eines Musters p bezüglich identifizierender Attribute im Allgemeinen nicht der besten Approximation für die Anzahl der tatsächlich vom Muster betroffenen Personen entspricht. Darüber hinaus folgt aus einem Supportwert von 0 bezüglich identifizierender Attribute nicht, dass tatsächlich keine Person vom Muster betroffen ist.

Eine Lösung des an Beispielen 3.2.5 und 3.2.6 dargestellten Problems der Approximation der vom einem Muster erfassten Personen ist es, eine Tabelle der Form $R(\mathcal{U}, A_1, \ldots, A_n)$ zu verlangen. Die Grundidee besteht darin, den Support eines Musters $p \in Pat_{\text{intra}}(A)$ in $R(\mathcal{U}, A_1, \ldots, A_n)$ bezüglich $\mathcal{U}$ zu bestimmen, d. h. $\text{supp}^{A}_{(R,\mathcal{U})}(p)$ als die Anzahl der vom Muster p erfassten Personen zu verwenden.

Definition 3.2.7 (k-anonymes intra-dimensionales Muster).
Sei $R(\mathcal{U}, A_1, \ldots, A_n)$ eine Tabelle,
$k \in \mathbb{N}$ eine Anonymitätsschwelle,
$A \in \{A_1, \ldots, A_n\}$ ein Attribut,
p ein intra-dimensionales Muster auf A.
Bezeichne p als **k-anonym** in R, falls $\text{supp}^{A}_{(R,\mathcal{U})}(p) = 0$ oder $\text{supp}^{A}_{(R,\mathcal{U})}(p) \geq k$.

Ein k-anonymes intra-dimensionales Muster stellt einen Bezug zu 0 oder mindestens k unterschiedlichen Personen sicher. Damit wird eine Zuordnung des Musters zu einer nicht leeren Gruppe mit weniger als k Personen verhindert. Ein intra-dimensionales Muster p,

das nicht k-anonym ist, ist eine notwendige Bedingung für das Vorhandensein einer nicht leeren Gruppe mit weniger als k Personen, die mit Hilfe des Musters p beschrieben werden können. Falls alle intra-dimensionalen Muster in R k-anonym sind, dann kann mit deren Hilfe unabhängig von a priori Wissen des Angreifers keine nicht leere Gruppe mit weniger als k Personen beschrieben werden.

Ein potentieller Angreifer kann aus eine Menge von intra-dimensionalen Assoziationsregeln nicht mehr Wissen inferieren, als aus den zugrundeliegenden Ausprägungsmengen mit dem jeweiligen Support. Deshalb wird im Folgenden eine Menge von Ausprägungsmenge-Support-Paaren statt einer Menge von intra-dimensionalen Assoziationsregeln als ein möglicher intra-dimensionaler Inferenzkanal betrachtet.

Definition 3.2.8 (Datenbankkompatibilität).
Sei $R(\mathcal{U}, A_1, \ldots, A_n)$ eine Tabelle,
$A \in \{A_1, \ldots, A_n\}$ ein Attribut,
$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\mathrm{dom}(A)} \times \mathbb{N}$ eine Menge von Ausprägungsmenge-Support-Paaren.
Bezeichne R als **mit S kompatibel**, falls für alle $(D, m) \in S$ gilt: $\mathrm{supp}^A_{(R,\mathcal{U})}(D) = m$.

Mit Hilfe der Datenbankkompatibilität lassen sich alle Datenbanken betrachten, deren Analyse zu den gegebenen Ausprägungsmenge-Support-Paaren geführt haben könnte. Dabei ist zu beachten, dass der Support bezüglich $\mathcal{U}$ berechnet wird, um die Anzahl betroffener Personen zu bestimmen.

Definition 3.2.9 (Intra-dimensionale Supportinferenz bezüglich $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$A \in \{A_1, \ldots, A_n\}$ ein Attribut,
$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$,
$p \in Pat_{\mathrm{intra}}(A)$ ein intra-dimensionales Muster auf A,
$k \in \mathbb{N}$ eine Supportschwelle,
$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\mathrm{dom}(A)} \times \mathbb{N}$ eine Menge von Ausprägungsmenge-Support-Paaren.
Schreibe
$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\mathrm{intra}} 0 < \mathrm{supp}^A(p) < k$, falls
für alle Tabellen $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, die mit S kompatibel sind, gilt:
$0 < \mathrm{supp}^A_{(R,\mathcal{U})}(p) < k$;
$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\mathrm{intra}} \mathrm{supp}^A(p) = k$, falls
für alle Tabellen $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, die mit S kompatibel sind, gilt:
$\mathrm{supp}^A_{(R,\mathcal{U})}(p) = k$.

Mit Hilfe der Supportinferenz lässt sich bei gegebenen Ausprägungsmenge-Support-Paaren über den Support von Mustern sprechen, ohne über die tatsächliche Tabelle, sondern lediglich das a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, zu verfügen.

Falls A ein sensibles Attribut ist, lässt sich eine einfachere Definition der Supportinferenz definieren. Der Grund dafür ist die Annahme (ix) in Abschnitt 3.1.1, dass das a priori Wissen keinen Zusammenhang zwischen Personen und Ausprägungen sensibler Attribute beschreiben darf.

Definition 3.2.10 (Intra-dimensionale Supportinferenz).

Sei A ein sensibles Attribut,

$p \in Pat_{\text{intra}}(A)$ ein intra-dimensionales Muster auf A,

$k \in \mathbb{N}$ eine Supportschwelle,

$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$ eine Menge von Ausprägungsmenge-Support-Paaren.

Schreibe

$S \models_{\text{intra}} 0 < \text{supp}^A(p) < k$, falls

für alle Tabellen $R(\mathcal{U}, A)$, die mit S kompatibel sind, gilt: $0 < \text{supp}^A_{(R,\mathcal{U})}(p) < k$;

$S \models_{\text{intra}} \text{supp}^A(p) = k$, falls

für alle Tabellen $R(\mathcal{U}, A)$, die mit S kompatibel sind, gilt: $\text{supp}^A_{(R,\mathcal{U})}(p) = k$.

Satz 3.2.11.

Es gelten die Annahmen aus Abschnitt 3.1.1.

Sei $\{A_1, \ldots, A_n\}$ *eine Menge von Attributen,*

$A \in \{A_1, \ldots, A_n\}$ *ein sensibles Attribut,*

$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ *a priori Wissen über* $\{A_1, \ldots, A_n\}$ *in* $\mathcal{U}$,

$p \in Pat_{intra}(A)$ *ein intra-dimensionales Muster auf* A,

$k \in \mathbb{N}$ *eine Supportschwelle,*

$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$ *eine Menge von Ausprägungsmenge-Support-Paaren.*

Es gilt

$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{intra} \text{supp}^A(p) = k$ *genau dann wenn* $S \models_{intra} \text{supp}^A(p) = k$.

Beweis.

Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,

$A \in \{A_1, \ldots, A_n\}$ ein sensibles Attribut,

$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$,

$p \in Pat_{\text{intra}}(A)$ ein intra-dimensionales Muster auf A,

$k \in \mathbb{N}$ eine Supportschwelle,

$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$ eine Menge von Ausprägungsmenge-Support-Paaren.

Da A sensibel ist, hält ein potentieller Angreifer nach Annahme 3.1.1 (ix) alle Tabellen $R(\mathcal{U}, A)$ für möglich.

Zeige »⇒«

$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{intra}} \text{supp}^A(p) = k$ impliziert $S \models_{\text{intra}} \text{supp}^A(p) = k$.

Sei $R(\mathcal{U}, A)$ beliebig und mit S kompatibel. Zeige $\text{supp}^A_{(R,\mathcal{U})}(p) = k$.
Da A sensibel ist und ein potentieller Angreifer deshalb die durch $R(\mathcal{U}, A)$ beschriebene Zuordnung zwischen $\text{dom}(\mathcal{U})$ und $\text{dom}(A)$ für möglich hält, existiert nach Annahme 3.1.1 (ix) ein $R'(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit $\pi_{(\mathcal{U},A)}(R') = R$. Aus s-facher Anwendung von Lemma 3.2.4 und Bemerkung 3.2.3 folgt, dass R' mit S kompatibel ist. Aus $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{intra}} \text{supp}^A(p) = k$ und Lemma 3.2.4 folgt damit, dass $\text{supp}^A_{(R,\mathcal{U})}(p) = \text{supp}^A_{(R',\mathcal{U})}(p) = k$ gilt.
Zeige »⇐«

$S \models_{\text{intra}} \text{supp}^A(p) = k$ impliziert $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{intra}} \text{supp}^A(p) = k$.

Sei $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ beliebig und mit S kompatibel. Zeige $\text{supp}^A_{(R,\mathcal{U})}(p) = k$. Definiere $R'(\mathcal{U}, A) = \pi_{(\mathcal{U},A)}(R)$. Aus s-facher Anwendung von Lemma 3.2.4 und Bemerkung 3.2.3 folgt, dass R' mit S kompatibel ist. Aus $S \models_{\text{intra}} \text{supp}^A(p) = k$ und Lemma 3.2.4 folgt damit, dass $\text{supp}^A_{(R,\mathcal{U})}(p) = \text{supp}^A_{(R',\mathcal{U})}(p) = k$ gilt. □

Der Satz 3.2.11 eliminiert die Betrachtung von a priori Wissen des Angreifers nach Definition 2.2.15 für intra-dimensionalen Muster auf sensiblen Attributen, wenn die Annahmen aus Abschnitt 3.1.1 gelten. Dabei spielt die Annahme (ix) aus Abschnitt 3.1.1 über das Unwissen des potentiellen Angreifers bezüglich des sensiblen Attributs A eine zentrale Rolle. Leider ist diese Annahme in der Praxis oftmals nicht erfüllt.

Definition 3.2.12 (Intra-dimensionaler k-Inferenzkanal).
Sei A ein sensibles Attribut,

$k \in \mathbb{N}$ eine Anonymitätsschwelle,

$S = \{(D_1, m_1), \ldots, (D_s, m_s)\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$ eine Menge von Ausprägungsmenge-Support-Paaren.

Bezeichne

S als **intra-dimensionaler k-Inferenzkanal**, falls

$\exists p \in Pat_{\text{intra}}(A) : S \models_{\text{intra}} 0 < \text{supp}^A(p) < k$.

Ein intra-dimensionaler k-Inferenzkanal stellt eine notwendige Bedingung für die Existenz eines nicht k-anonymen Musters in jeder Tabelle, die mit den gegebenen Menge-Support-Paaren kompatibel ist, dar. Da das betrachtete Attribut A sensibel ist, muss nach Satz 3.2.11 das entsprechende a priori Wissen nicht zusätzlich betrachtet werden, wenn die Annahmen aus Abschnitt 3.1.1 gelten.

3.2.2 Erkennung und Beseitigung von Inferenzkanälen

In diesem Abschnitt wird mittels einer Homomorphie zwischen Inferenzkanälen für intra-dimensionale Muster und Inferenzkanälen für häufige Itemmengen einer binären Datenbank die Inferenzkanalfreiheit sichergestellt.

In diesem Abschnitt wird die zu einer Multimenge von Transaktionen $\mathcal{T}$ äquivalente Menge $\text{set}(\mathcal{T}) = \{(i, t) \in \mathbb{N} \times \mathcal{T} \mid t \text{ kommt in } \mathcal{T} \text{ genau } n\text{-mal vor und } 0 < i \leq n\}$ verwendet.

Zunächst wird einer binären Datenbank $\mathcal{D}$ eine Menge von Tabellen $R(\mathcal{U}, A)$ zugeordnet.

Definition 3.2.13.

Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,

A ein Attribut mit $\mathrm{dom}(A) = \mathcal{I}$.

Definiere

$\mathrm{rel}(\mathcal{D},\mathcal{U}, A) = \{(R(\mathcal{U}, A), f) \mid f\colon \mathrm{set}(\mathcal{T}) \to \mathrm{dom}(\mathcal{U})$ ist eine injektive Funktion,

$R(\mathcal{U}, A)$ Tabelle derjenigen Tupel (u, a),

für die eine Transaktion $(i, t) \in \mathrm{set}(\mathcal{T})$ existiert mit $u = f((i, t))$ und $a \in t\}$.

3.2.14 Bemerkung. Die Injektivität der Funktion f modelliert die Annahme 3.1.2 (ii), laut welcher eine Person der betrachteten Population höchstens eine Transaktion beeinflusst.

3.2.15 Bemerkung. Falls $|\mathrm{dom}(\mathcal{U})| \geq |\mathrm{set}(\mathcal{T})|$ für $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ gilt, dann ist $\mathrm{rel}(\mathcal{D},\mathcal{U}, A)$ nicht leer.

Lemma 3.2.16.

Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ *eine binäre Datenbank,*

A *ein Attribut mit* $\mathrm{dom}(A) = \mathcal{I}$,

$(R(\mathcal{U}, A), f) \in \mathrm{rel}(\mathcal{D},\mathcal{U}, A)$,

$X \subseteq \mathrm{dom}(A)$ *eine Menge von Ausprägungen von* A,

p *eine aussagenlogische Formel mit Variablen aus* $\mathrm{dom}(A)$.

Es gilt

(i) $\quad \mathrm{supp}^A_{(R,\mathcal{U})}(p) = \mathrm{supp}_{\mathcal{D}}(p)$,

(ii) $\quad \mathrm{supp}^A_{(R,\mathcal{U})}(X) = \mathrm{supp}_{\mathcal{D}}(X)$.

Beweis.

Sei $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ eine binäre Datenbank,

A ein Attribut mit $\mathrm{dom}(A) = \mathcal{I}$,

$(R(\mathcal{U}, A), f) \in \mathrm{rel}(\mathcal{D},\mathcal{U}, A)$,

$X \subseteq \mathrm{dom}(A)$ eine Menge von Ausprägungen von A,

p eine aussagenlogische Formel mit Variablen aus $\mathrm{dom}(A)$.

Zeige (i)

$$\mathrm{supp}^A_{(R,\mathcal{U})}(p) \overset{\text{Def. 3.2.2}}{=} |\,\mathrm{eval}^A_{(R,\mathcal{U})}(p)| \overset{!}{=} |\{(i, t) \in \mathrm{set}(\mathcal{T}) \mid [\![p]\!]_t = \mathit{true}\}| \overset{\text{Def. 2.1.8}}{=} \mathrm{supp}_{\mathcal{D}}(p).$$

Zeige dazu $(*)$ mittels einer Induktion nach der Tiefe des Syntaxbaums von p, dass für ein $(i, t) \in \mathrm{set}(\mathcal{T})$ und $u = f((i, t))$ die Aussage $u \in \mathrm{eval}^A_{(R,\mathcal{U})}(p)$ genau dann gilt, wenn $[\![p]\!]_t = \mathit{true}$ gilt.

Seien $(i, t) \in \mathrm{set}(\mathcal{T})$ und $u = f((i, t))$.

Induktionsanfang (Tiefe 0):

Eine aussagenlogische Formel mit einem Syntaxbaum der Tiefe 0 besteht aus genau einer

aussagenlogischen Variablen, d. h. $p = a \in \mathrm{dom}(A)$.
In diesem Fall gilt die folgende Äquivalenz:

$$
\begin{array}{ll}
u \in \mathrm{eval}^A_{(R,\mathcal{U})}(a) & \\
\overset{\text{Def. 3.2.2}}{\iff} & u \in \pi_{\mathcal{U}}(\sigma_{A=a}(R)) \\
\iff & (u,a) \in \pi_{\mathcal{U},A}(R) \\
\overset{\text{Def. 3.2.13}}{\iff} & \exists (i',t') \in \mathrm{set}(\mathcal{T}) \text{ mit } u = f((i',t')) \text{ und } a \in t' \\
\overset{u=f((i,t)),f \text{ injektiv}}{\iff} & (i,t) = f^{-1}(u) = (i',t') \in \mathrm{set}(\mathcal{T}) \text{ mit } a \in t' \\
\overset{u=f((i,t)),f \text{ injektiv}}{\iff} & a \in t \\
\iff & [\![a]\!]_t = true
\end{array}
$$

Induktionsvoraussetzung (Tiefe $\leq t$):
Für $(i,t) \in \mathrm{set}(\mathcal{T})$ und $u = f((i,t))$ gilt $u \in \mathrm{eval}^A_{(R,\mathcal{U})}(p)$ genau dann, wenn $[\![p]\!]_t = true$ gilt.

Induktionsschritt (Tiefe $t+1$):
Der letzte Operator eines Syntaxbaums von p der Tiefe $t+1$ ist entweder $\wedge$, $\vee$ oder $\neg$. Diese Fälle werden einzeln betrachtet.

Fall $p = p_1 \wedge p_2$:

$$
\begin{array}{ll}
q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1 \wedge p_2) & \\
\overset{\text{Def. 3.2.2}}{\iff} & q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \cap \mathrm{eval}^A_{(R,\mathcal{U})}(p_2) \\
\iff & q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \text{ und } q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_2) \\
\overset{\text{IV}}{\iff} & [\![p_1]\!]_t = true \text{ und } [\![p_2]\!]_t = true \\
\iff & [\![p_1 \wedge p_2]\!]_t = true
\end{array}
$$

Fall $p = p_1 \vee p_2$:

$$
\begin{array}{ll}
q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1 \vee p_2) & \\
\overset{\text{Def. 3.2.2}}{\iff} & q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \cup \mathrm{eval}^A_{(R,\mathcal{U})}(p_2) \\
\iff & q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \text{ oder } q \in \mathrm{eval}^A_{(R,\mathcal{U})}(p_2) \\
\overset{IV}{\iff} & [\![p_1]\!]_t = true \text{ oder } [\![p_2]\!]_t = true \\
\iff & [\![p_1 \vee p_2]\!]_t = true
\end{array}
$$

Fall $p = \neg p_1$:

$$
\begin{array}{ll}
q \in \mathrm{eval}^A_{(R,\mathcal{U})}(\neg p_1) & \\
\overset{\text{Def. 3.2.2}}{\iff} & \pi_{\mathcal{U}}(R) \setminus \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \\
\iff & q \notin \mathrm{eval}^A_{(R,\mathcal{U})}(p_1) \\
\overset{IV}{\iff} & [\![p_1]\!]_t \neq true \\
\iff & [\![\neg p_1]\!]_t = true
\end{array}
$$

Insgesamt ergibt sich

$$\begin{aligned}
&\operatorname{supp}^A_{(R,\mathcal{U})}(p)\\
\overset{\text{Def. 3.2.2}}{=}\ &|\operatorname{eval}^A_{(R,\mathcal{U})}(p)|\\
\overset{\operatorname{eval}^A_{(R,\mathcal{U})}(p)\subseteq\pi_{\mathcal{U}}(R)}{=}\ &|\{u\in\pi_{\mathcal{U}}(R)\mid u\in\operatorname{eval}^A_{(R,\mathcal{U})}(p)\}|\\
\overset{\text{Def. 3.2.13}}{=}\ &|\{u\in\pi_{\mathcal{U}}(R)\mid \exists(i,t)\in\operatorname{set}(\mathcal{T}): u=f((i,t)), u\in\operatorname{eval}^A_{(R,\mathcal{U})}(p)\}|\\
\overset{(*)}{=}\ &|\{u\in\pi_{\mathcal{U}}(R)\mid \exists(i,t)\in\operatorname{set}(\mathcal{T}): u=f((i,t)), [\![p]\!]_t=true\}|\\
\overset{f\text{ injektiv}}{=}\ &|\{(i,t)\in\operatorname{set}(\mathcal{T})\mid [\![p]\!]_t=true\}|\\
\overset{\text{Def. 2.1.8}}{=}\ &\operatorname{supp}_{\mathcal{D}}(p)
\end{aligned}$$

Zeige (ii)

$$\operatorname{supp}^A_{(R,\mathcal{U})}(X)\overset{\text{Bemerkung 3.2.3}}{=}\operatorname{supp}^A_{(R,\mathcal{U})}(\bigwedge_{x\in X}x)\overset{(i)}{=}\operatorname{supp}_{\mathcal{D}}(\bigwedge_{x\in X}x)=\operatorname{supp}_{\mathcal{D}}(X).$$ □

Satz 3.2.17.

Es gelten die Annahmen aus Abschnitten 3.1.1 und 3.1.2.

Sei A ein sensibles Attribut,

$k\in\mathbb{N}$ eine Anonymitätsschwelle,

$S=\{(D_1,m_1),\dots,(D_s,m_s)\}\subseteq 2^{\operatorname{dom}(A)}\times\mathbb{N}$ Menge von Ausprägungsmenge-Support-Paaren.

Wenn S ein intra-dimensionaler k-Inferenzkanal ist, dann ist S ein k-Inferenzkanal für binäre Datenbanken.

Beweis.

Sei A ein sensibles Attribut,

$k\in\mathbb{N}$ eine Anonymitätsschwelle,

$S=\{(D_1,m_1),\dots,(D_s,m_s)\}\subseteq 2^{\operatorname{dom}(A)}\times\mathbb{N}$ ein intra-dimensionaler k-Inferenzkanal.

Da S ein intra-dimensionaler k-Inferenzkanal ist, existiert nach Definition 3.2.12 ein intra-dimensionales Muster $p\in Pat_{\text{intra}}(A)$ mit $S\models_{\text{intra}}0<\operatorname{supp}^A(p)<k$.

Sei $\mathcal{D}=(\mathcal{I},\mathcal{T})$ eine mit S kompatible binäre Datenbank. Nach Annahme 3.1.2 (ii) gilt $|\operatorname{dom}(\mathcal{U})|\geq|\operatorname{set}(\mathcal{T})|$.

Zeige $0<\operatorname{supp}_{\mathcal{D}}(p)<k$.

Da $|\operatorname{dom}(\mathcal{U})|>|\operatorname{set}(\mathcal{T})|$ gilt, folgt $\operatorname{rel}(\mathcal{D},\mathcal{U},A)\neq\emptyset$. Wähle ein $(R(\mathcal{U},A),f)\in\operatorname{rel}(\mathcal{D},\mathcal{U},A)$.

Da $\mathcal{D}$ mit S kompatibel ist, folgt aus der s-fachen Anwendung von Lemma 3.2.16 (ii), dass $R(\mathcal{U},A)$ mit S kompatibel ist. Zusammen mit $S\models_{\text{intra}}0<\operatorname{supp}^A(p)<k$ und der Definition 3.2.10 folgt $0<\operatorname{supp}^A_{(R,\mathcal{U})}(p)<k$. Aus Lemma 3.2.16 (i) folgt damit $0<\operatorname{supp}^A_{(R,\mathcal{U})}(p)=\operatorname{supp}_{\mathcal{D}}(p)=\operatorname{supp}^A_{(R,\mathcal{U})}(p)<k$. □

Aus diesem Satz folgt, dass zur Erkennung und Beseitigung von intra-dimensionalen k-Inferenzkanälen Ergebnisse für binäre Datenbanken aus [3] genutzt werden können. Die Vorgehensweise zur Erkennung und Beseitigung von intra-dimensionalen k-Inferenzkanälen ist in Algorithmus 3.1 festgehalten. Die Grundidee besteht darin, Ausprägungsmengen mit

dem dazugehörigen Support bezüglich $\mathcal{U}$ als Itemmenge-Support-Paare im binären Datenbankmodell zu betrachten. Wenn diese Menge kein k-Inferenzkanal für binäre Datenbanken mit einem entsprechenden nicht k-anonymen Muster p ist, dann ist sie nach Satz 3.2.17 auch kein intra-dimensionaler k-Inferenzkanal. Anderenfalls kann die k-Anonymität des Musters p durch das Entfernen der von p betroffenen Personen aus der Tabelle sichergestellt werden. Dies bewirkt gleichzeitig, dass der Support von p im binären Datenbankmodell nach Lemma 3.2.16 (i) auf 0 sinkt und p entsprechend k-anonym wird.

Eingabe:
Population $\mathcal{U}$,
Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$,
sensibles Attribut $A \in \{A_1, \ldots, A_n\}$,
Anonymitätsschwelle $k \in \mathbb{N}$,
Attributmenge $Z \subseteq \{A_1, \ldots, A_n\}$, bezüglich welcher die Relevanz bestimmt wird,
Menge von Ausprägungsmenge-Support-Paaren
$\{(D_1, \text{supp}^A_{(R,Z)}(D_1)), \ldots, (D_m, \text{supp}^A_{(R,Z)}(D_m))\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$
Ausgabe:
Sanitisierte Tabelle $R'(\mathcal{U}, A_1, \ldots, A_n)$,
sanitisierte Menge von Ausprägungsmenge-Support-Paaren
$\{(D_1, \text{supp}^A_{(R',Z)}(D_1)), \ldots, (D_m, \text{supp}^A_{(R',Z)}(D_m))\} \subseteq 2^{\text{dom}(A)} \times \mathbb{N}$.

$R' \leftarrow R$
$S \leftarrow \{(D_1, \text{supp}^A_{(R',\mathcal{U})}(D_1)), \ldots, (D_m, \text{supp}_{(R',\mathcal{U})}(D_m))\}$
while S ist ein k-Inferenzkanal im binären Datenbankmodell **do**
  $p \leftarrow$ Muster auf $\text{dom}(A)$ mit $S \models 0 < \text{supp}(p) < k$
  $R' \leftarrow R' \setminus \{t \in R' \mid t(\mathcal{U}) \in \text{eval}^A_{(R',\mathcal{U})}(p)\}$
  $S \leftarrow \{(D_1, \text{supp}^A_{(R',\mathcal{U})}(D_1)), \ldots, (D_m, \text{supp}^A_{(R',\mathcal{U})}(D_m))\}$
end while
return $R', \{(D_1, \text{supp}^A_{(R',Z)}(D_1)), \ldots, (D_m, \text{supp}^A_{(R',Z)}(D_m))\}$

Algorithmus 3.1: Beseitigung von intra-dimensionalen k-Inferenzkanälen

Der Algorithmus 3.1 terminiert, da bei jedem Schleifendurchlauf mindestens ein Tupel aus R' entfernt wird. Der Algorithmus 3.1 stellt die k-Inferenzkanalfreiheit her, da bei jedem Schleifendurchlauf der Support des nicht k-anonymen Musters p (sowohl für R' als intra-dimensionales Muster, als auch für alle mit S kompatiblen binären Datenbanken als Muster in binären Datenbankmodell durch Lemma 3.2.16) auf 0 gesenkt wird. Die Schleifenabbruchbedingung garantiert durch Satz 3.2.17, dass S kein intra-dimensionaler k-Inferenzkanal ist. Dabei werden Routinen zur Erkennung von k-Inferenzkänalen im binären Datenbankmodell und zur Erzeugung eines nicht k-anonymen Musters im binären

Datenbankmodell benötigt. Da das Attribut A sensibel ist, muss nach Satz 3.2.11 das entsprechende a priori Wissen nach Definition 2.2.15 nicht zusätzlich betrachtet werden, wenn die Annahmen aus Abschnitt 3.1.1 gelten.

Es ist wichtig zu beachten, dass der Algorithmus 3.1 intern mit Supportwerten bezüglich $\mathcal{U}$ und nicht bezüglich Z arbeitet, und Supportwerte bezüglich Z lediglich als Schnittstelle zu Data Mining Algorithmen in der Ein- und die Ausgabe benutzt werden. Dafür sprechen mehrere Gründe. Zum einen sind Supportwerte bezüglich $\mathcal{U}$ die beste Approximation für die Anzahl der von einem Muster betroffenen Personen in der Tabelle. Zum anderen kann nicht ausgeschlossen werden, dass ein potenzieller Angreifer aus gegebenen Supportwerten bezüglich Z mittels in Kapitel 2.2.3 erlaubtem a priori Wissen auf Supportwerte bezüglich $\mathcal{U}$ schließen kann.

Ein weiterer Aspekt von Algorithmus 3.1 ist die Entfernung bestimmter Tupel bei jedem Schleifendurchlauf. Die k-Anonymität eines intra-dimensionalen Musters lässt sich auch durch das Hinzufügen neuer Tupel erreichen. Die Ausprägungen dieser Tupel auf den Attributen Z sind jedoch unklar und können semantische Bedingungen an eine Tabelle verletzen. Auch lassen sich nicht beliebig viele Ausprägungen bestimmter Attribute finden, wie z. B. »*Geschlecht*«.

3.3 Inter-dimensionale Muster

Bei der Veröffentlichung von inter-dimensionalen Data Mining Ergebnissen, bei welchen sich die in Abschnitt 3.2.2 beschriebene Homomorphie nicht nutzen lässt, reicht die Betrachtung einer Tabelle als eine binäre Datenbank (Definition 2.2.18) zur Sicherstellung der k-Anonymität aufgrund des Verlustes an Semantik nicht aus. Dieses Problem wurde bereits anhand des Beispiels 2.2.19 demonstriert.

Durch den Verlust an Semantik werden bei der Sicherstellung der k-Anonymität Domänen der Attribute nicht berücksichtigt. Beispielsweise hat das Attribut »Geschlecht« genau zwei Ausprägungen $\{m, w\}$. Dies führt dazu, dass aus dem Support des Musters »$\neg(Geschlecht = m)$« der Support des Musters »$Geschlecht = f$« gefolgert werden kann. Dieser Zusammenhang wird von Algorithmen, die für binäre Datenbanken konzipiert sind, nicht beachtet.

Zusätzlich schließt die Darstellung der Tupel einer Tabelle als Transaktionen Muster der Form »$A = a \wedge A = b$« für $a \neq b$ nicht aus. Der Support solcher Muster ist offensichtlich 0, da in keinem Tupel an einer Stelle zwei unterschiedliche Ausprägungen stehen können. Algorithmen, die für binäre Datenbanken konzipiert sind, verwenden solches Wissen nicht.

Ferner werden Quasi-Identikatoren nicht berücksichtigt. So werden Muster der Form »$Vorname = Max \wedge Nachname = Mustermann \wedge \ldots$« unter anderem dadurch anonymisiert, dass mehrere Personen mit dem Namen »Max Mustermann« und gleichen Aus-

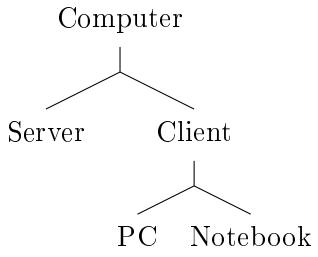
prägungen sensibler Attribute erzeugt werden. Die Veröffentlichung eines solchen Musters genügt nicht den Vertraulichkeitsanforderungen.

3.3.1 Beispiel. Für eine Tabelle, die von Kunden gekaufte Produkte unter Angabe ihres Geschlechts beinhaltet, seien eine Taxonomie für Computer (Abbildung 3.1) sowie folgende Daten verfügbar:

- 100 männliche Kunden haben Computer gekauft,
- 99 männliche Kunden haben einen Client gekauft.

Aus diesen Daten folgt unmittelbar, dass 1 männlicher Kunde einen Server gekauft hat. Bei der Betrachtung der Tabelle als eine binäre Datenbank zur Sicherstellung der k-Anonymität (Abschnitt 2.2.4) entsprechen die gegebenen Daten den beiden Mustern »$Geschlecht = m \wedge (Produkt = Server \vee Produkt = PC \vee Produkt = Notebook)$« bzw. »$Geschlecht = m \wedge Produkt = Computer$« und »$Geschlecht = m \wedge (Produkt = PC \vee Produkt = Notebook)$« bzw. »$Geschlecht = m \wedge Produkt = Client$« mit Supports 100 und 99. Nur mit Mitteln aus [3] lässt sich weder der Support des Musters »$Geschlecht = m \wedge Produkt = Server$« noch des semantisch äquivalenten Musters »$Geschlecht = m \wedge \neg(Produkt = PC) \wedge \neg(Produkt = Notebook)$« herleiten.

Abbildung 3.1: Taxonomie für Computer



Das Beispiel 3.3.1 macht deutlich, dass die alleinige Betrachtung von Tabellen als binäre Datenbanken mit Items der Form »Attribut=Ausprägung« zur Sicherstellung der Anonymität nicht ausreicht, wenn darüber hinaus weitere semantische Information über Ausprägungen der Attribute zur Verfügung steht und nicht genutzt wird.

In diesem Abschnitt werden **inter-dimensionale Muster** eingeführt, um semantische Aspekte von Tabellen und mehrstufigen Assoziationsregeln zu berücksichtigen. Dabei werden gemäß Annahme Annahme (xii) in Abschnitt 3.1.1 Tabellen betrachtet, die keine zwei Tupel mit Informationen über dieselbe Person enthalten, d. h. $R(\mathcal{U}, A_1, \ldots, A_n)$ darf für jede Person $u \in \mathrm{dom}(\mathcal{U})$ höchstens ein Tupel t mit $t(\mathcal{U}) = u$ enthalten.

3.3.1 Übertragung der Grundbegriffe

Die Intention hinter einem **inter-dimensionalen Muster** ist es, die in Beispiel 2.2.19 beschriebene Vorgehensweise um die Semantik von Tabellen zu erweitern. Insbesondere sollen Muster, die syntaktisch zwar erlaubt, semantisch aber ausgeschlossen sind, vermieden werden. Deshalb werden inter-dimensionale Muster mit einer eigenen Syntax und Semantik eingeführt.

Es ist zu beachten, dass bei der Übertragung der Grundbegriffe immer die Gültigkeit der Annahmen aus Abschnitt 3.1.1 vorausgesetzt wird.

Definition 3.3.2 (Inter-dimensionales Muster).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$Z_1 \subseteq \mathrm{dom}(A_1), \ldots, Z_n \subseteq \mathrm{dom}(A_n)$ Mengen von Ausprägungen.
Bezeichne $(A_1 \in Z_1, \ldots, A_n \in Z_n)$ als **inter-dimensionales Muster**.
Definiere $Pat_{\mathrm{inter}}(A_1, \ldots, A_n)$
$= \{p \mid p \text{ ist inter-dimensionales Muster mit Attributen aus } \{A_1, \ldots, A_n\}\}$.

Ein inter-dimensionales Muster $p \in Pat_{\mathrm{inter}}(A_1, \ldots, A_n)$ muss nicht alle Attribute aus $\{A_1, \ldots, A_n\}$ beinhalten. Insbesondere gilt $() \in Pat_{\mathrm{inter}}(A_1, \ldots, A_n)$.

Ein inter-dimensionales Muster $(A_1 \in Z_1, \ldots, A_n \in Z_n)$ beschreibt die Menge der Tupel t einer Tabelle mit mindestens den Attributen $A_1, \ldots, A_n$, für die gilt:
$t(A_1) \in Z_1, \ldots, t(A_n) \in Z_n$. Der Support von inter-dimensionalen Mustern spiegelt diese Eigenschaft wieder.

Definition 3.3.3 (Support von inter-dimensionalen Mustern).
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$\{A_{i_1}, \ldots, A_{i_m}\} \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$p = (A_{i_1} \in Z_1, \ldots, A_{i_m} \in Z_m)$ ein inter-dimensionales Muster.
Definiere **Support:** $\mathrm{supp}_R(p) = |\bigcup_{(z_1,\ldots,z_m) \in Z_1 \times \ldots \times Z_m} \sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)|$.

Zur Approximation der Anzahl der vom Muster betroffenen Personen ist das Attribut $\mathcal{U}$ bzw. eine Projektion darauf nicht notwendig, da bei betrachteten Tabellen nach Annahme (xii) in Abschnitt 3.1.1 unterschiedliche Tupel Informationen über unterschiedliche Personen enthalten und damit die Anzahl der vom Muster betroffenen Tupel bereits der Anzahl der vom Muster erfassten Personen entspricht.

Mehrstufige Assoziationsregeln lassen sich mit inter-dimensionalen Mustern berücksichtigen, da eine Aussage der Form »*Attribut* = *Klasse*« sich mittels des Musters (*Attribut* ∈ Menge der Blätter im Teilbaum der Taxonomie mit der Wurzel »*Klasse*«) darstellen lässt.

Definition 3.3.4 (k-anonymes inter-dimensionales Muster).
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$k \in \mathbb{N}$ eine Anonymitätsschwelle,
$p \in Pat_{\text{inter}}(A_1, \ldots, A_n)$ ein inter-dimensionales Muster.
Bezeichne p als **k-anonym** in R, falls $\text{supp}_R(p) = 0$ oder $\text{supp}_R(p) \geq k$.

Analog zu intra-dimensionalen Mustern werden die Begriffe Datenbankkompatibilität, Supportinferenz und Inferenzkanal definiert.

Definition 3.3.5 (Datenbankkompatibilität).
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von Muster-Support-Paaren.
Bezeichne R als **mit S kompatibel**, falls für alle $(p, m) \in S$ gilt: $supp_R(p) = m$.

Definition 3.3.6 (Inter-dimensionale Supportinferenz bezüglich $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$,
$p \in Pat_{\text{inter}}(A_1, \ldots, A_n)$ ein inter-dimensionales Muster,
$k \in \mathbb{N}$ eine Supportschwelle,
$S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von Muster-Support-Paaren.
Schreibe
$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} 0 < \text{supp}(p) < k$, falls
für alle Tabellen $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, die mit S kompatibel sind, gilt:
$0 < \text{supp}_R(p) < k$;
$S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} \text{supp}(p) = k$, falls
für alle Tabellen $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$, die mit S kompatibel sind, gilt:
$\text{supp}_R(p) = k$.

Mit Hilfe der Supportinferenz lässt sich bei gegebenen Muster-Support-Paaren über den Support von Mustern sprechen, ohne über die tatsächliche Tabelle, sondern lediglich über das a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ zu verfügen.

Definition 3.3.7 (Inter-dimensionaler k-Inferenzkanal bezüglich $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen über $\{A_1, \ldots, A_n\}$ in $\mathcal{U}$,
$k \in \mathbb{N}$ eine Anonymitätsschwelle,
$S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von Muster-Support-Paaren.
Bezeichne
S als **inter-dimensionaler k-Inferenzkanal bezüglich $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$**, falls gilt:
$\exists p \in Pat_{\text{inter}}(A_1, \ldots, A_n) : S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} 0 < \text{supp}(p) < k$.

3.3.2 Inter-dimensionales Supportinferenzproblem

In diesem Abschnitt wird eine untere Komplexitätsschranke für die Entscheidung, ob eine Menge von inter-dimensionalen Muster-Support-Paaren mittels der Supportinferenz den Support eines weiteren Musters bestimmt, bewiesen. Dazu wird das **inter-dimensionale Supportinferenzproblem** definiert, um dann das Komplement eines NP-schweren Problems (eins-in-drei SAT) darauf zu reduzieren.

1 Problem (Inter-dimensionales Supportinferenzproblem).
Es gelten die Annahmen aus Abschnitt 3.1.1.
Gegeben:
$\text{dom}(\mathcal{U})$ eine endliche Population,
$\{A_1, \ldots, A_n\}$ eine Menge von Attributen,
$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen,
$S = \{(p_1, n_1), \ldots, (p_s, n_s)\} \subseteq Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von inter-dimensionalen Muster-Support-Paaren,
$p \in Pat_{\text{inter}}(A_1, \ldots, A_n)$ ein inter-dimensionales Muster,
$k \in \mathbb{N}$ ein Supportwert.
Frage:
Gilt $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} \text{supp}(p) = k$?

Die Endlichkeit der betrachteten Population $\text{dom}(\mathcal{U})$ und der Größen aller anderen Eingaben durch Annahme (iv) in Abschnitt 3.1.1 garantiert, dass das inter-dimensionale Supportinferenzproblem entscheidbar ist.

Zur Bestimmung der Komplexität des inter-dimensionalen Supportinferenzproblems muss die Eingabegröße genauer definiert werden. Es ist nicht sinnvoll, die Kardinalität von $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ auch nur linear einfließen zu lassen, da das Supportinferenzproblem damit mittels einer linearen Iteration über $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ lösbar wäre.
Wenn $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ beispielsweise der Menge aller Tabellen $R(\mathcal{U}, A_1, \ldots, A_n)$ entspricht, d.h. ein potentieller Angreifer hat bis auf die vorkommenden Attribute $\{A_1, \ldots, A_n\}$ keine

Informationen, ist die Kardinalität von $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ exponentiell in der Summe der Kardinalitäten der Domänen von $A_1, \ldots, A_n$. Eine geeignete Kodierung von $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ wäre in diesem Fall eine durch eine Turingmaschine kodierte Funktion, die jede syntaktisch korrekte Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$ auf $true$ und alle anderen Tabellen auf $false$ abbildet. Die Größe dieser Kodierung wäre polynomiell in der Summe der Kardinalitäten der Domänen von $A_1, \ldots, A_n$.

Im Folgenden wird als Eingabegröße für das Problem 1 der Term

$$\underbrace{c(|\operatorname{dom}(\mathcal{U})|)}_{\text{Kodierung von } \mathcal{U}} + \underbrace{c(n) + \sum_{i=1}^{n} c(|\operatorname{dom}(A_i)|)}_{\text{Kodierung von } A_1,\ldots,A_n}$$

$$+ s * \underbrace{\Big(\underbrace{c(n) + \sum_{i=1}^{n} c(|\operatorname{dom}(A_i)|)}_{\text{Kodierung des größten Musters}} + \underbrace{\log_2(|\operatorname{dom}(\mathcal{U})| * \prod_{i=1}^{n} |\operatorname{dom}(A_i)|)}_{\text{Kodierung des größten Supports}} \Big)}_{\text{Kodierung von } s \text{ Muster-Support-Paaren}} + \underbrace{\log_2(k)}_{\text{Kodierung von } k}$$

mit $c(x) = x * \log_2(x)$ verwendet.

Das Problem **eins-in-drei SAT** [7] ist eine Einschränkung des **SAT** Problems. Um in diesem Abschnitt über ein einheitliches Vokabular zu verfügen, werden folgende Begriffe definiert:

- Ein **Literal** ist eine aussagenlogische Variable x oder deren Negation $\neg x$.
- Eine **Klausel** ist eine Menge von Literalen.
- Eine **Variablenbelegung** ist eine Abbildung von aussagenlogischen Variablen auf $\{0, 1\}$.
- Eine Variablenbelegung b **macht ein Literal** l **wahr**, falls $l = x$ eine aussagenlogische Variable ist und $b(x) = 1$ gilt, oder falls $l = \neg x$ eine negierte aussagenlogische Variable ist und $b(x) = 0$ gilt.

2 Problem (Eins-in-drei SAT).
Gegeben:
$C = \{c_1, \ldots, c_n\}$ eine Menge von Klauseln mit genau drei Literalen pro Klausel.
Frage:
Existiert eine Variablenbelegung, die genau ein Literal jeder Klausel aus C wahr macht?

Satz 3.3.8.
Das inter-dimensionale Supportinferenzproblem ist Co-NP-schwer.

Beweis.

Das eins-in-drei SAT Problem ist NP-schwer [7]. Da für die polynomielle Reduktion ($\leq_p$)

und $L \in$ NP die Aussage $L \leq_p$ eins-in-drei SAT genau dann gilt, wenn Komplement$(L) \leq_p$ Komplement(eins-in-drei SAT) gilt, ist das Komplement von eins-in-drei SAT Co-NP-schwer. Um zu zeigen, dass das inter-dimensionale Supportinferenzproblem Co-NP-schwer ist, wird das Komplement von eins-in-drei SAT darauf polynomiell reduziert.
Sei $C = \{c_1, \ldots, c_n\}$ eine Menge von Klauseln mit genau drei Literalen pro Klausel,
$X = \{x_1, \ldots, x_m\}$ die Menge der in C vorkommenden aussagenlogischen Variablen,

Betrachte eine Population dom$(\mathcal{U})$ mit $2 * m + 1$ Personen. Konstruiere
ein Attribut A mit der Domäne dom$(A) = \{y, x_1, \ldots, x_m, \neg x_1, \ldots, \neg x_m\}$,
a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A\}}$ = Menge aller Tabellen $R(\mathcal{U}, A)$, kodiert als Funktion (Turingmaschine), die für alle syntaktisch korrekten Tabellen $R(\mathcal{U}, A)$ den Wert *true* liefert und $false$ sonst,
eine Menge von Muster-Support-Paaren $S_1 = \{((A \in \{x_i, \neg x_i\}), 1) \mid x_i \in X\}$,
eine Menge von Muster-Support-Paaren $S_2 = \{((A \in c), 1) \mid c \in C\}$,
eine Menge von Muster-Support-Paaren $S_3 = \{((A \in \{y\}), 1)\}$,
eine Menge von Muster-Support-Paaren $S = S_1 \cup S_2 \cup S_3$,
ein Muster $p = (A \in \{y\})$,
ein Supportwert $k = 2$.
Offensichtlich kann diese Konstruktion deterministisch in Polynomialzeit in der Größe der Eingabe C berechnet werden. Zeige
(i) wenn keine Variablenbelegung existiert, die genau ein Literal jeder Klausel aus C wahr macht, dann gilt $S \models_{\text{inter}} \text{supp}(p) = 2$.
(ii) wenn eine Variablenbelegung existiert, die genau ein Literal jeder Klausel aus C wahr macht, dann gilt $S \models_{\text{inter}} \text{supp}(p) = 2$ nicht.
Erst wird gezeigt $(*)$, dass wenn eine Tabelle existiert, die mit S kompatibel ist, dann existiert eine Variablenbelegung, die genau ein Literal jeder Klausel aus C wahr macht.

Beweis von $(*)$ Angenommen es existiert eine (syntaktisch korrekte) Tabelle $R(\mathcal{U}, A)$, die mit S kompatibel ist.
Definiere Variablenbelegung
$b\colon X \cup \{y\} \to \{0, 1\}$,
$$b(x) = \begin{cases} 0 & \text{wenn } |\sigma_{A=x}(R)| = 0 \\ 1 & \text{sonst} \end{cases}$$
Da R mit S_1 kompatibel ist, enthält R zu jeder Variablen $x \in X$ genau ein Tupel, in dem entweder sie oder ihre Negation als Ausprägung des Attributs A vorkommt. Also gilt durch die Kompatibilität zu S_1 für $x \in X$ die Aussage $|\sigma_{A=x}(R)| = 0$ genau dann, wenn $|\sigma_{A=\neg x}(R)| = 1$ gilt. Analog gilt für $x \in X$ die Aussage $|\sigma_{A=x}(R)| = 1$ genau dann, wenn $|\sigma_{A=\neg x}(R)| = 0$ gilt. Damit macht b genau diejenigen Literale l wahr, für die gilt $|\sigma_{A=l}(R)| = 1$.

Da R mit S_2 kompatibel ist, gibt es für jede Klausel $\{l_1, l_2, l_3\}$ genau ein Literal l_i für das $|\sigma_{A=l_i}(R)| = 1$ gilt. Damit ist b eine Variablenbelegung, die genau ein Literal jeder Klausel aus C wahr macht. Damit ist $(*)$ gezeigt.

Beweis von (i) Da aus der Existenz einer mit S kompatiblen Tabelle gemäß $(*)$ die Existenz einer Variablenbelegung folgt, die genau ein Literal jeder Klausel aus C wahr macht, gibt es im Fall der Nichtexistenz einer solchen Variablenbelegung auch keine mit S kompatible Tabelle. Folglich erfüllt jede mit S kompatible Tabelle R die Bedingung $\text{supp}_R(p) = 2$. Damit ist (i) gezeigt.

Beweis von (ii) Da $|\text{dom}(\mathcal{U})| = 2 * m + 1 = |\text{dom}(A)|$ gilt, existiert eine injektive Funktion $f\colon \text{dom}(A) \to \text{dom}(\mathcal{U})$. Wähle eine solche Funktion f.
Angenommen es existiert eine Variablenbelegung b, die genau ein Literal jeder Klausel aus C wahr macht.

Definiere eine Tabelle $R(\mathcal{U}, A)$ mit der Tupelmenge
$\{(f(x), x) \mid x \in X, b(x) = 1\} \cup \{(f(\neg x), \neg x) \mid x \in X, b(x) = 0\} \cup \{(f(y), y)\}$.

Die Injektivität von f stellt die Gültigkeit der Annahme (xii) in Abschnitt 3.1.1 sicher.

$R(\mathcal{U}, A)$ ist syntaktisch korrekt, daher gilt $R(\mathcal{U}, A) \in \mathcal{W}^{\mathcal{U}}_{\{A\}}$.

R ist mit S_1 kompatibel, da jede Variable $x \in X$ auf entweder 0 oder 1 durch b abgebildet wird, d. h. R enthält jede Variable aus X als Ausprägung von A genau einmal entweder positiv oder negiert.

R ist mit S_2 kompatibel, da b in jeder Klausel aus C genau ein Literal wahr macht und R genau die Literale der Variablen aus X, die b wahr macht, enthält, d. h. R enthält genau ein Literal aus jeder Klausel als Ausprägung von A.

R ist mit S_3 kompatibel, da R die Variable y genau ein mal als Ausprägung von A enthält.

Folglich ist R mit S kompatibel, aber der Support des Musters p ist 1, also gilt $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A\}}}_{\text{inter}} \text{supp}(p) = 2$ nicht. Damit ist (ii) gezeigt. □

3.3.3 Inter-dimensionaler Ableitungsoperator

In Abschnitt 3.3.2 wurde gezeigt, dass das inter-dimensionales Supportinferenzproblem Co-NP schwer ist. Nichtsdestotrotz lässt sich in vielen Fällen bei gegebenen Muster-Support-Paaren der Support anderer Muster bestimmen. Das kann ein potentieller Angreifer ausnutzen, um nicht k-anonyme inter-dimensionale Muster zu entdecken.

In diesem Abschnitt wird ein **inter-dimensionale Ableitungsoperator** vorgestellt, mit dessen Hilfe syntaktisch aus einer Menge von Muster-Support Paaren S weitere Muster-

Support Paare, die gemäß Definition 3.3.6 semantisch aus S folgen, bestimmt werden können.

Zum einfacheren Umgang mit inter-dimensionalen Mustern werden einige Zugriffs- und Vergleichsoperatoren definiert.

Definition 3.3.9 (Definitionsbereich, Attributauswahl, Erweiterung).
Sei $p = (A_1 \in Z_1, \ldots, A_m \in Z_m)$ ein inter-dimensionales Muster,
A ein Attribut,
$Z \subseteq \mathrm{dom}(A)$ eine Menge von Ausprägungen von A.
Definiere

Definitionsbereich: $\mathrm{dom}(p) = \{A_1, \ldots, A_m\}$,

Attributauswahl:
$$p(A) = \begin{cases} Z_i & \text{wenn } A = A_i \text{ für } 1 \leq i \leq m \\ \mathrm{dom}(A) & \text{sonst, d. h. } A \notin \mathrm{dom}(p) \end{cases},$$

Veränderung:
$$p[A \in Z] = \begin{cases} (A_1 \in Z_1, \ldots, A_i \in Z, \ldots, A_m \in Z_m) \\ \quad \text{wenn } A = A_i \text{ für } 1 \leq i \leq m \\ (A \in Z, \underbrace{A_1 \in Z_1, \ldots, A_m \in Z_m}_{p}) \\ \quad \text{sonst, d. h. } A \notin \mathrm{dom}(p) \end{cases}.$$

Lemma 3.3.10.
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$p = (A_{i_1} \in Z_1, \ldots, A_{i_m} \in Z_m)$ ein inter-dimensionales Muster,
$A \in \{A_1, \ldots, A_n\} \setminus \mathrm{dom}(p)$ ein Attribut.
Es gilt
$\mathrm{supp}_R(p) = \mathrm{supp}_R(p[A \in \mathrm{dom}(A)])$.

Beweis.
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$p = (A_{i_1} \in Z_1, \ldots, A_{i_m} \in Z_m)$ ein inter-dimensionales Muster,
$A \in \{A_1, \ldots, A_n\} \setminus \mathrm{dom}(p)$ ein Attribut.
Es wird Gleichheit der Mengen bewiesen, deren Kardinalitäten den Supportwerten von p und $p[A \in \mathrm{dom}(A)]$ gemäß Definition 3.3.3 entsprechen. Aufgrund von $A \in \{A_1, \ldots, A_n\} \setminus \mathrm{dom}(p)$ gilt nach Definition 3.3.3 und Definition 3.3.9
$$\mathrm{supp}_R(p[A \in \mathrm{dom}(A)]) = |\bigcup_{(z,z_1,\ldots,z_m) \in \mathrm{dom}(A) \times Z_1 \times \ldots \times Z_m} \sigma_{A=z, A_{i_1}=z_1, \ldots, A_{i_m}=z_m}(R)|.$$
Zeige
$$\bigcup_{(z_1,\ldots,z_m) \in Z_1 \times \ldots \times Z_m} \sigma_{A_{i_1}=z_1, \ldots, A_{i_m}=z_m}(R) = \bigcup_{(z,z_1,\ldots,z_m) \in \mathrm{dom}(A) \times Z_1 \times \ldots \times Z_m} \sigma_{A=z, A_{i_1}=z_1, \ldots, A_{i_m}=z_m}(R)$$
Zeige »$\subseteq$«

Sei $t \in \bigcup_{(z_1,\ldots,z_m)\in Z_1\times\ldots\times Z_m} \sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)$,
dann existiert ein $(z'_1,\ldots,z'_m) \in Z_1 \times \ldots \times Z_m$ mit $t \in \sigma_{A_{i_1}=z'_1,\ldots,A_{i_m}=z'_m}(R)$ und es folgt
$t \in \sigma_{A=t(A),A_{i_1}=z'_1,\ldots,A_{i_m}=z'_m}(R) \subseteq \bigcup_{(z,z_1,\ldots,z_m)\in\mathrm{dom}(A)\times Z_1\times\ldots\times Z_m} \sigma_{A=z,A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)$.

Zeige »$\supseteq$«

Sei $t \in \bigcup_{(z,z_1,\ldots,z_m)\in\mathrm{dom}(A)\times Z_1\times\ldots\times Z_m} \sigma_{A=z,A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)$,
dann existiert ein $(z', z'_1,\ldots,z'_m) \in \mathrm{dom}(A) \times Z_1 \times \ldots \times Z_m$ mit
$t \in \sigma_{A=z',A_{i_1}=z'_1,\ldots,A_{i_m}=z'_m}(R)$.
Damit ist $t \in \sigma_{A_{i_1}=z'_1,\ldots,A_{i_m}=z'_m}(R) \subseteq \bigcup_{(z_1,\ldots,z_m)\in Z_1\times\ldots\times Z_m} \sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)$.

Insgesamt folgt

$$\begin{aligned}
&\mathrm{supp}_R(p)\\
&= |\bigcup_{(z_1,\ldots,z_m)\in Z_1\times\ldots\times Z_m} \sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)|\\
&= |\bigcup_{(z,z_1,\ldots,z_m)\in\mathrm{dom}(A)\times Z_1\times\ldots\times Z_m} \sigma_{A=z,A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)|\\
&= \mathrm{supp}_R(p[A \in \mathrm{dom}(A)]).
\end{aligned}$$

□

Definition 3.3.11 (Unterschied).
Seien p_1, p_2 inter-dimensionale Muster.
Definiere

Unterschied $\quad \mathrm{diff}(p_1,p_2) = \{A \in \mathrm{dom}(p_1) \cup \mathrm{dom}(p_2) \mid p_1(A) \neq p_2(A)\}$

Definition 3.3.12 (Inter-dimensionaler Ableitungsoperator).
Der **inter-dimensionale Ableitungsoperator** $(\vdash) \subseteq 2^{Pat_{\mathrm{inter}}\times\mathbb{N}} \times (Pat_{\mathrm{inter}} \times \mathbb{N})$ ist die kleinste Relation, die die Ableitungsregeln (AX), (SUB), (ADD) und (HALF) erfüllt, d. h. für $S \subseteq Pat_{\mathrm{inter}} \times \mathbb{N}$ und $(p,n) \in Pat_{\mathrm{inter}} \times \mathbb{N}$ gilt $S \vdash (p,n)$ genau dann, wenn ein endlicher Ableitungsbaum aus entsprechenden Regeln mit der Wurzel $S \vdash (p,n)$ existiert. Für $S \vdash (p,n)$ ist dabei gleichbedeutend mit $(S,(p,n)) \in (\vdash)$.

$$\text{(AX)}\ \frac{(p,n) \in S}{S \vdash (p,n)}$$

$$\text{(SUB)}\ \frac{S \vdash (p_1,n_1) \qquad S \vdash (p_2,n_2) \qquad \mathrm{diff}(p_1,p_2) = \{A\} \text{ mit } p_1(A) \supset p_2(A)}{S \vdash (p_1[A \in p_1(A) \setminus p_2(A)], n_1 - n_2)}$$

$$\text{(ADD)}\ \frac{S \vdash (p_1,n_1) \qquad S \vdash (p_2,n_2) \qquad \mathrm{diff}(p_1,p_2) = \{A\} \text{ mit } p_1(A) \cap p_2(A) = \emptyset}{S \vdash (p_1[A \in p_1(A) \cup p_2(A)], n_1 + n_2)}$$

$$\text{(HALF)}\ \frac{S \vdash (p_1, n_1) \qquad S \vdash (p_2, n_2) \qquad S \vdash (p_3, n_3) \qquad \begin{array}{l}\text{diff}(p_1, p_2) = \\ \text{diff}(p_1, p_3) = \{A\} \text{ mit} \\ p_1(A) \triangle p_2(A) = p_3(A)\end{array}}{S \vdash (p_1[A \in p_1(A) \cap p_2(A)], \lfloor \frac{n_1 + n_2 - n_3}{2} \rfloor)}$$

Der mengentheoretische Operator $\triangle$ steht dabei für die symmetrische Differenz, d. h. $A \triangle B = (A \cup B) \setminus (A \cap B)$.

Die Intention hinter dem inter-dimensionalen Ableitungsoperator ist es, alle Mengen S von Muster-Support-Paaren und weitere Muster-Support-Paare (p, n), die syntaktisch aus S inferiert werden können, in Beziehung $S \vdash (p, n)$ zu setzen und den Zusammenhang zu der in Definition 3.3.6 beschriebenen semantischen Inferenz zu untersuchen.

Das Beispiel 3.3.13 demonstriert die Korrektheit der Anwendung jeder Ableitungsregel.

3.3.13 Beispiel. Für eine Population, die mindestens 6 Personen beinhaltet, beschreibt die Tabelle $R(\mathcal{U}, A, B)$ (Abbildung 3.2) Ausprägungen der Attribute A mit $\text{dom}(A) = \{a, b, c\}$ und B mit $\text{dom}(B) = \{x, y, z\}$.

Abbildung 3.2: Tabelle $R(\mathcal{U}, A, B)$

$\mathcal{U}$	A	B
u_1	a	x
u_2	a	z
u_3	b	y
u_4	c	x
u_5	c	y
u_6	c	z

Sei S als die mit R nach Definition 3.3.5 kompatible Menge folgender inter-dimensionale-Muster-Support-Paare gegeben:
$(p_1, m_1) = ((B \in \{x, y\}), 4)$,
$(p_2, m_2) = ((A \in \{c\}, B \in \{x, y\}), 2)$,
$(p_3, m_3) = ((A \in \{a\}, B \in \{x, z\}), 2)$,
$(p_4, m_4) = ((A \in \{b\}, B \in \{x, z\}), 0)$,
$(p_5, m_5) = ((A \in \{a, b\}, B \in \{y, z\}), 2)$.

Der inter-dimensionale Ableitungsoperator erlaubt folgende Ableitungen der Tiefe 1:

$$\begin{array}{l}\text{(AX)} \\ \text{(SUB)}\end{array}\ \frac{\dfrac{(p_1, m_1) \in S}{S \vdash (p_1, m_1)} \quad \text{(AX)} \dfrac{(p_2, m_2) \in S}{S \vdash (p_2, m_2)} \qquad \begin{array}{l}\text{diff}(p_1, p_2) = \{A\} \text{ mit} \\ p_1(A) = \text{dom}(A) \supset p_2(A)\end{array}}{S \vdash (p_6, m_6) = ((A \in \{a, b\}, B \in \{x, y\}), 4 - 2 = 2)}$$

$$\text{(ADD)}\ \frac{\text{(AX)}\ \dfrac{(p_3, m_3) \in S}{S \vdash (p_3, m_3)} \quad \text{(AX)}\ \dfrac{(p_4, m_4) \in S}{S \vdash (p_4, m_4)} \quad \begin{array}{l}\operatorname{diff}(p_3, p_4) = \{A\} \text{ mit} \\ p_3(A) \cap p_4(A) = \emptyset\end{array}}{S \vdash (p_7, m_7) = ((A \in \{a, b\}, B \in \{x, z\}), 2 + 0 = 2)}$$

Die Ableitbarkeit der Muster p_6 und p_7 erlaubt die folgende Ableitung:

$$\text{(HALF)}\ \frac{\text{(AX)}\ \dfrac{(p_5, m_5) \in S}{S \vdash (p_5, m_5)} \quad S \vdash (p_6, m_6) \quad S \vdash (p_7, m_7) \quad \begin{array}{l}\operatorname{diff}(p_5, p_6) = \\ \operatorname{diff}(p_5, p_7) = \{B\} \text{ mit} \\ p_5(B) \triangle p_6(B) = p_7(B)\end{array}}{S \vdash (p_8, m_8) = ((A \in \{a, b\}, B \in \{y\}), \lfloor \frac{2+2-2}{2} \rfloor = 1)}$$

Die Permutation der Rollen von p_5, p_6 und p_7 führt analog zu
$S \vdash (p_9, m_9) = ((A \in \{a, b\}, B \in \{x\}), 1)$ und
$S \vdash (p_{10}, m_{10}) = ((A \in \{a, b\}, B \in \{z\}), 1)$.

Mit Hilfe von p_8 lässt sich eine weitere Ableitung bilden:

$$\text{(ADD)}\ \frac{S \vdash (p_7, m_7) \quad S \vdash (p_8, m_8) \quad \begin{array}{l}\operatorname{diff}(p_7, p_8) = \{B\} \text{ mit} \\ p_7(B) \cap p_8(B) = \emptyset\end{array}}{S \vdash (p_{11}, m_{11}) = ((A \in \{a, b\}, B \in \{x, y, z\}), 2 + 1 = 3)}$$

Da $p_{11}(B) = \{x, y, z\} = \operatorname{dom}(B)$ gilt, entspricht das Muster
$p_{11} = (A \in \{a, b\}, B \in \{x, y, z\})$ nach Lemma 3.3.10 dem Muster $(A \in \{a, b\})$.

Dieses Beispiel demonstriert die kombinierte Anwendung aller Regeln des inter-dimensionalen Ableitungsoperators. Die Supportwerte der abgeleiteten Muster entsprechen zudem den tatsächlichen Supportwerten in R.

Um die Korrektheit des inter-dimensionalen Ableitungsoperators zu beweisen, wird erst der Support von Mustern näher untersucht.

Lemma 3.3.14.
Sei $R(A_1, \ldots, A_n)$ (bzw. $R(\mathcal{U}, A_1, \ldots, A_n)$) eine Tabelle,
$p = (A_{i_1} \in Z_1, \ldots, A_{i_m} \in Z_m)$ ein inter-dimensionales Muster.
Es gilt

$$\begin{aligned}\operatorname{supp}_R(p) &= \sum_{(z_1,\ldots,z_m) \in Z_1 \times \ldots \times Z_m} |\sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)| \\ &= \sum_{(z_1,\ldots,z_m) \in Z_1 \times \ldots \times Z_m} \operatorname{supp}_R((A_{i_1} \in \{z_1\}, \ldots, A_{i_m} \in \{z_m\})) \\ &= \sum_{(z_1,\ldots,z_n) \in p(A_1) \times \ldots \times p(A_n)} \operatorname{supp}_R((A_1 \in \{z_1\}, \ldots, A_n \in \{z_n\})).\end{aligned}$$

Beweis.
Die erste Gleichheit von Lemma 3.3.14 ergibt sich direkt aus der Tatsache, dass für unterschiedliche $(z_1, \ldots, z_m), (z'_1, \ldots, z'_m) \in Z_1 \times \ldots \times Z_m$ die Mengen $\sigma_{A_{i_1}=z_1,\ldots,A_{i_m}=z_m}(R)$

und $\sigma_{A_{i_1}=z'_1,\ldots,A_{i_m}=z'_m}(R)$ disjunkt sind. Damit entspricht nach Definition 3.3.3 der Support eines Musters der Kardinalität der Vereinigung disjunkter Mengen und damit der Summe der Kardinalitäten dieser Mengen.

Die zweite Gleichheit ergibt sich aus der Definition 3.3.3 des Supports.

Die dritte Gleichheit ergibt sich direkt aus bis zu n-facher Anwendung von Lemma 3.3.10 zur Erweiterung des Definitionsbereiches von p auf $A_1, \ldots, A_n$ und anschließender Anwendung der ersten beiden Gleichheiten. □

Lemma 3.3.15.
Sei A ein Attribut,

$X_1, X_2, X_3 \subseteq \mathrm{dom}(A)$ *Mengen von Ausprägungen von A,*

$f\colon \mathrm{dom}(A) \to \mathbb{N}$ *eine Funktion.*

Es gilt

$$(i) \quad X_1 \supset X_2 \text{ impliziert} \quad \sum_{x\in(X_1\setminus X_2)} f(x) = \sum_{x\in X_1} f(x) - \sum_{x\in X_2} f(x),$$

$$(ii) \quad X_1 \cap X_2 = \emptyset \text{ impliziert} \quad \sum_{x\in(X_1\cup X_2)} f(x) = \sum_{x\in X_1} f(x) + \sum_{x\in X_2} f(x),$$

$$(iii) \quad X_1 \triangle X_2 = X_3 \text{ impliziert} \quad 2 * \sum_{x\in(X_1\cap X_2)} f(x) = \sum_{x\in X_1} f(x) + \sum_{x\in X_2} f(x) - \sum_{x\in X_3} f(x)$$

Beweis.
Offensichtliche mengentheoretische Beobachtungen. □

Satz 3.3.16 (Korrektheit des inter-dimensionalen Ableitungsoperators).
Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,

$S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{inter}(A_1, \ldots, A_n) \times \mathbb{N}$ *eine Menge von Muster-Support-Paaren,*

$(p, k) \in Pat_{inter}(A_1, \ldots, A_n) \times \mathbb{N}$ *ein Muster-Support-Paar.*

Der inter-dimensionale Ableitungsoperator ist für alle a priori Wissen korrekt, d. h.
für alle a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ gilt: $S \vdash (p, k)$ impliziert $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{inter} \mathrm{supp}(p) = k$.

Beweis.

Sei $\{A_1, \ldots, A_n\}$ eine Menge von Attributen,

$S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von Muster-Support-Paaren,

$(p, k) \in Pat_{\text{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ ein Muster-Support-Paar,

$\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ a priori Wissen.

Falls $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ keine mit S kompatiblen Tabellen enthält, dann gilt die Behauptung nach Definition 3.3.6. Sonst sei $R(\mathcal{U}, A_1, \ldots, A_n) \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ eine mit S kompatible Tabelle.

Angenommen es existiert ein Ableitungsbaum für $S \vdash (p, k)$. Der Beweis des Satzes 3.3.16 erfolgt mittels einer Induktion nach der Tiefe des Ableitungsbaums.

Induktionsanfang (Tiefe 0):

Ein Ableitungsbaum der Tiefe 0 für $S \vdash (p, k)$ ist nur durch eine Anwendung der Regel (AX) konstruierbar. Daraus folgt $(p, k) \in S$. Da R mit S kompatibel ist, gilt nach Definition 3.3.5 $\text{supp}_R(p) = k$.

Induktionsvoraussetzung (Tiefe $\leq t$):

Der inter-dimensionale Ableitungsoperator ist korrekt.

Induktionsschritt (Tiefe $t + 1$):

Die letzte Ableitungsregel eines Ableitungsbaums für $S \vdash (p, k)$ der Tiefe $t + 1$ ist entweder (SUB), (ADD) oder (HALF). Diese Fälle werden einzeln betrachtet. Die Prämisse Ableitungsregel (AX) bedingt deren Anwendung ausschließlich in Blättern von Ableitungsbäumen.

Fall (SUB):

Die Anwendbarkeit der Regel (SUB) impliziert die Existenz von Muster-Support-Paaren (p_1, m_1) und (p_2, m_2) mit $S \vdash (p_1, m_1)$, $S \vdash (p_2, m_2)$ und eines Attributs A mit $\text{diff}(p_1, p_2) = \{A\}$ und $p_1(A) \supset p_2(A)$. O. B. d. A. sei $A = A_1$. Da die Tiefe der Ableitungsbäume für $S \vdash (p_1, m_1)$ und $S \vdash (p_2, m_2)$ höchstens t ist, folgt aus der Induktionsvoraussetzung $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_1) = m_1$ und $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_2) = m_2$.
Da $R \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit S kompatibel ist, folgt damit $\text{supp}_R(p_1) = m_1$ und $\text{supp}_R(p_2) = m_2$.

Insgesamt folgt:

$$
\begin{aligned}
&\text{supp}_R(p) \\
&= \quad \text{supp}_R(p_1[A_1 \in p_1(A_1) \setminus p_2(A_1)]) \\
&\overset{\text{Lemma 3.3.14}}{=} \quad \sum_{(z_1,\ldots,z_m)\in(p_1(A_1)\setminus p_2(A_1))\times p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
&= \quad \sum_{z_1\in(p_1(A_1)\setminus p_2(A_1))} \left(\sum_{(z_2,\ldots,z_m)\in p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
&\overset{(*)}{=} \quad \sum_{z_1\in p_1(A_1)} \left(\sum_{(z_2,\ldots,z_m)\in p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) - \\
&\qquad \sum_{z_1\in p_2(A_1)} \left(\sum_{(z_2,\ldots,z_m)\in p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
&= \quad \sum_{(z_1,z_2,\ldots,z_m)\in p_1(A_1)\times p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) - \\
&\qquad \sum_{(z_1,z_2,\ldots,z_m)\in p_2(A_1)\times p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
&\overset{\text{Lemma 3.3.14}}{=} \quad \text{supp}_R(p_1) - \\
&\qquad \sum_{(z_1,z_2,\ldots,z_m)\in p_2(A_1)\times p_1(A_2)\times\ldots\times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
&\overset{\text{diff}(p_1,p_2)=A_1}{=} \quad \text{supp}_R(p_1) - \\
&\qquad \sum_{(z_1,z_2,\ldots,z_m)\in p_2(A_1)\times p_2(A_2)\times\ldots\times p_2(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
&\overset{\text{Lemma 3.3.14}}{=} \quad \text{supp}_R(p_1) - \text{supp}_R(p_2) = m_1 - m_2 = k
\end{aligned}
$$

Die Begründung $(*)$ ergibt sich aus $p_1(A) \supset p_2(A)$ und Lemma 3.3.15 (i).

Fall (ADD):

Die Anwendbarkeit der Regel (ADD) impliziert die Existenz von Muster-Support-Paaren (p_1, m_1) und (p_2, m_2) mit $S \vdash (p_1, m_1)$, $S \vdash (p_2, m_2)$ und eines Attributs A mit $\text{diff}(p_1, p_2) = \{A\}$ und $p_1(A) \cap p_2(A) = \emptyset$. O.B.d.A. sei $A = A_1$. Da die Tiefe der Ableitungsbäume für $S \vdash (p_1, m_1)$ und $S \vdash (p_2, m_2)$ höchstens t ist, folgt aus der Induktionsvoraussetzung $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_1) = m_1$ und $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_2) = m_2$. Da $R \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit S kompatibel ist, folgt damit $\text{supp}_R(p_1) = m_1$ und $\text{supp}_R(p_2) = m_2$. Insgesamt folgt:

$$
\begin{aligned}
& \text{supp}_R(p) \\
= \quad & \text{supp}_R(p_1[A_1 \in p_1(A_1) \cup p_2(A_1)]) \\
\overset{\text{Lemma 3.3.14}}{=} \quad & \sum_{(z_1,\ldots,z_m) \in (p_1(A_1) \cup p_2(A_1)) \times p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
= \quad & \sum_{z_1 \in (p_1(A_1) \cup p_2(A_1))} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
\overset{(**)}{=} \quad & \sum_{z_1 \in p_1(A_1)} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) + \\
& \sum_{z_1 \in p_2(A_1)} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
= \ldots = \quad & \text{supp}_R(p_1) + \text{supp}_R(p_2) = m_1 + m_2 = k
\end{aligned}
$$

Die Begründung $(**)$ ergibt sich aus $p_1(A) \cap p_2(A) = \emptyset$ und Lemma 3.3.15 (ii). Die mit »...« abgekürzten Zwischenschritte verlaufen analog zum Fall (SUB).

Fall (HALF):

Die Anwendbarkeit der Regel (HALF) impliziert die Existenz von Muster-Support-Paaren (p_1, m_1), (p_2, m_2) und (p_3, m_3) mit $S \vdash (p_1, m_1)$, $S \vdash (p_2, m_2)$, $S \vdash (p_3, m_3)$ und eines Attributs A mit $\text{diff}(p_1, p_2) = \text{diff}(p_1, p_3) = \text{diff}(p_2, p_3) = \{A\}$ und $p_1(A) \triangle p_2(A) = p_3(A)$. O. B. d. A. sei $A = A_1$. Da die Tiefe der Ableitungsbäume für $S \vdash (p_1, m_1)$, $S \vdash (p_2, m_2)$ und $S \vdash (p_3, m_3)$ höchstens t ist, folgt aus der Induktionsvoraussetzung $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_1) = m_1$, $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_2) = m_2$ und $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}} \text{supp}(p_3) = m_3$. Da $R \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ mit S kompatibel ist, folgt damit $\text{supp}_R(p_1) = m_1$, $\text{supp}_R(p_2) = m_2$ und $\text{supp}_R(p_3) = m_3$. Insgesamt folgt:

$$
\begin{aligned}
& 2 * \text{supp}_R(p) \\
= \quad & 2 * \text{supp}_R(p_1[A_1 \in p_1(A_1) \cap p_2(A_1)]) \\
\overset{\text{Lemma 3.3.14}}{=} \quad & 2 * \sum_{(z_1,\ldots,z_m) \in (p_1(A_1) \cap p_2(A_1)) \times p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \\
= \quad & 2 * \sum_{z_1 \in (p_1(A_1) \cap p_2(A_1))} \\
& \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
\overset{(***)}{=} \quad & \sum_{z_1 \in p_1(A_1)} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) + \\
& \sum_{z_1 \in p_2(A_1)} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) - \\
& \sum_{z_1 \in p_3(A_1)} \left(\sum_{(z_2,\ldots,z_m) \in p_1(A_2) \times \ldots \times p_1(A_m)} \text{supp}_R(A_1 \in \{z_1\}, \ldots, A_m \in \{z_m\}) \right) \\
= \ldots = \quad & \text{supp}_R(p_1) + \text{supp}_R(p_2) - \text{supp}_R(p_3) = m_1 + m_2 - m_3
\end{aligned}
$$

Die Begründung $(***)$ ergibt sich aus $p_1(A) \triangle p_2(A) = p_3(A)$ und Lemma 3.3.15 (*iii*). Die mit »...« abgekürzten Zwischenschritte verlaufen analog zum Fall (SUB), wobei das Muster p_3 analog zum Muster p_2 behandelt wird. Da m_1, m_2, m_3 und $\text{supp}_R(p)$ definitionsgemäß natürliche Zahlen sind, folgt aus $2 * \text{supp}_R(p) = m_1 + m_2 - m_3$, dass $m_1 + m_2 - m_3$ durch 2 teilbar ist und $\text{supp}_R(p) = \lfloor \frac{m_1 + m_2 - m_3}{2} \rfloor = k$ gilt. Die Gaußklammern stellen lediglich die Typkorrektheit im Fall illegaler Eingaben her. □

3.3.17 Beispiel (Erweiterung des Beispiels 3.3.13).

Aus der Korrektheit des inter-dimensionalen Ableitungsoperators folgt, dass in Beispiel 3.3.13 ein inter-dimensionaler 2-Inferenzkanal erkannt wurde. Der Support jedes Musters in S beträgt 0 oder mindestens 2, dennoch sind die aus S ableitbaren Muster p_8, p_9 und p_{10} nicht 2-anonym. Da der inter-dimensionale Ableitungsoperator nach Satz 3.3.16 korrekt ist, folgt für $p \in \{p_8, p_9, p_{10}\}$ und jedes $\mathcal{W}^{\mathcal{U}}_{\{A,B\}}$ die Aussage $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A,B\}}} \text{supp}(p) = 1$ und damit insbesondere $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A,B\}}} 0 < \text{supp}(p) < 2$. Damit ist S ein inter-dimensionaler 2-

Inferenzkanal zu dessen Erkennung die Anwendung aller Regeln des inter-dimensionalen Ableitungsoperators nötig ist.

3.3.18 Bemerkung. Da der inter-dimensionale Ableitungsoperator das a priori Wissen nicht berücksichtigt, kann er nicht für alle a priori Wissen vollständig sein, d. h. wenn für ein a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ die Aussage $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} \text{supp}(p) = k$ gilt, lässt sich nicht folgern, dass $S \vdash (p,k)$ gilt.

Um dies zu demonstrieren, konstruiere

eine Population $\text{dom}(\mathcal{U})$ mit mindestens einer Person,

ein nicht sensibles Attribut A mit $\{a,b\} \subseteq \text{dom}(A)$,

ein a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A\}}$, das nur Tabellen $R(\mathcal{U},A)$ enthält mit $\pi_{\{A\}}(R) = \{a\}$ oder $\pi_{\{A\}}(R) = \{\}$,

eine Menge von Muster-Support-Paaren $S = \{((A \in \{a,b\}),1)\}$.

Da alle Tabellen $R(\mathcal{U},A) \in \mathcal{W}^{\mathcal{U}}_{\{A\}}$ nur Personen mit a als Ausprägung von A enthalten und die Kompatibilität zu S garantiert, dass genau eine Person mit a oder b als Ausprägung von A vorkommt, folgt $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A\}}}_{\text{inter}} \text{supp}((A \in \{a\})) = 1$.

Bis auf (AX), lässt sich keine Ableitungsregel des inter-dimensionalen Ableitungsoperators anwenden. Insbesondere gilt nicht $S \vdash ((A \in \{a\}),1)$.

3.3.19 Bemerkung. Der inter-dimensionale Ableitungsoperator ist sogar für alle a priori Wissen (inklusive des Randfalls »kein Wissen«) nicht vollständig, d. h. wenn für alle a priori Wissen $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ die Aussage $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}}_{\text{inter}} \text{supp}(p) = k$ gilt, lässt sich nicht folgern, dass $S \vdash (p,k)$ gilt.

Um dies zu demonstrieren, konstruiere

eine Population $\text{dom}(\mathcal{U})$ mit mindestens 4 Personen,

ein Attribut A mit $\{a,b,c,d\} \subseteq \text{dom}(A)$,

eine Menge von Muster-Support-Paaren $S = \{((A \in \{a,b\}),2),((A \in \{a,c\}),2),((A \in \{a,d\}),2),((A \in \{b,c,d\}),3)\}$,

ein Muster-Support-Paar $(p,k) = ((A \in \{a\}),1)$.

Sei $\mathcal{W}^{\mathcal{U}}_{\{A\}}$ ein beliebiges a priori Wissen. Es gilt $S \models^{\mathcal{W}^{\mathcal{U}}_{\{A\}}}_{\text{inter}} \text{supp}(p) = k$, denn für eine mit S kompatible Tabelle $R(\mathcal{U},A) \in \mathcal{W}^{\mathcal{U}}_{\{A\}}$ gilt:

(i) Aus $supp_R((A \in \{a,b\})) = 2$ folgt, dass R genau 0,1 oder 2 Tupel mit a als Ausprägung von A enthält,

(ii) R kann nicht genau 0 Tupel mit a als Ausprägung von A enthalten, da sonst für Ausprägungen b,c und d jeweils genau 2 Tupel in R enthalten sein müssten. Damit würde $supp_R((A \in \{b,c,d\})) = 6$ gelten und die Bedingung $supp_R((A \in \{b,c,d\})) = 3$ wäre verletzt.

(iii) R kann nicht genau 2 Tupel mit a als Ausprägung von A enthalten, da sonst für Ausprägungen b,c und d jeweils genau 0 Tupel in R enthalten sein müssten. Damit würde $supp_R((A \in \{b,c,d\})) = 0$ gelten und die Bedingung $supp_R((A \in \{b,c,d\})) = 3$ wäre verletzt.

Aus (i), (ii), (iii) folgt entweder $supp_R((A \in \{a\})) = 1$ oder die Nichtexistenz von R. Insbesondere gilt $S \models_{\text{inter}}^{\mathcal{W}^{\mathcal{U}}_{\{A\}}} \text{supp}(p) = k$.

Bis auf (AX), lässt sich keine Ableitungsregel des inter-dimensionalen Ableitungsoperators anwenden. Insbesondere gilt nicht $S \vdash (p,k)$.

3.3.20 Bemerkung. Aus der Korrektheit des inter-dimensionalen Ableitungsoperators folgt, dass er zur Erkennung von einigen Widersprüchen in einer Menge S von Muster-Support-Paaren verwendet werden kann. Wenn für ein Muster p sowohl $S \vdash (p,k)$ als auch $S \vdash (p,k')$ mit $k \neq k'$ abgeleitet werden kann, dann folgt aus der Korrektheit des inter-dimensionalen Ableitungsoperators und der Definition 3.3.6 $\forall R \in \mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}} : k = \text{supp}_R(p) = k'$. Falls $\mathcal{W}^{\mathcal{U}}_{\{A_1,\ldots,A_n\}}$ nicht leer ist, ist dies ein Widerspruch.

Analog führt $S \vdash (p,k)$ mit $k < 0$ oder $k > |\text{dom}(\mathcal{U})|$ zum Widerspruch.

Darüber hinaus wurde im Beweis des Satzes 3.3.16 gezeigt, dass falls die Prämisse der Regel (HALF) in Definition 3.3.12 zutrifft, d. h. $S \vdash (p_1,n_1)$, $S \vdash (p_2,n_2)$, $S \vdash (p_3,n_3)$ und $\text{diff}(p_1,p_2) = \text{diff}(p_1,p_3) = \{A\}$ mit $p_1(A) \triangle p_2(A) = p_3(A)$ gilt, dann $n_1 + n_2 - n_3$ gerade sein muss. Falls $n_1 + n_2 - n_3$ ungerade ist, existiert keine mit S kompatible Tabelle.

3.3.4 Erkennung und Beseitigung von Inferenzkanälen

In diesem Abschnitt wird ausgehend vom inter-dimensionalen Ableitungsoperator eine Möglichkeit vorgestellt, einige inter-dimensionale k-Inferenzkanäle zu erkennen und zu beseitigen. Da der inter-dimensionale Ableitungsoperator für kein a priori Wissen vollständig ist, lassen sich mit seiner Hilfe nicht alle inter-dimensionalen k-Inferenzkanäle erkennen. Aufgrund der Co-NP-Schwierigkeit des inter-dimensionalen Supportinferenzproblems (Satz 3.3.8) kann an dieser Stelle kein korrekter und vollständiger Polynomialzeitalgorithmus zur Erkennung und Beseitigung von inter-dimensionalen k-Inferenzkanälen angegeben werden.

Zunächst müssen vorliegende Data Mining Ergebnisse als inter-dimensionale Muster repräsentiert werden. Dies ist nicht immer möglich. Sowohl inter-dimensionalen Assoziationsregeln als auch deren hybriden Erweiterungen liegen in der Literatur [6, 8, 10] Kardinalitäten von **Projektion-Selektion-Anfragen** zugrunde. Im Fall von mehrstufigen inter-dimensionalen Assoziationsregeln ist der Selektionsoperator bezüglich einer Taxonomie erweitert. Analog zur Vorgehensweise in Abschnitt 3.2.2 werden an dieser Stelle Projektionen auf $\mathcal{U}$ statt beliebiger Attributmengen Z betrachtet, um die genaue Anzahl der von einer Selektion-Anfrage beschriebenen Personen zu erhalten. Solche Projektion-Selektion-Anfragen lassen sich mittels inter-dimensionaler Muster modellieren, was die folgende Bemerkung 3.3.21 demonstriert.

3.3.21 Bemerkung.

Sei T eine Taxonomie,

$R(\mathcal{U}, A_1, \ldots, A_n)$ eine Tabelle,

$\{A_{i_1}, \ldots, A_{i_{m+t}}\} \subseteq \{A_1, \ldots, A_n\}$ eine Menge von Attributen,

$c_1 \in \mathrm{dom}(A_{i_1}), \ldots, c_m \in \mathrm{dom}(A_{i_m})$ Ausprägungen der entsprechenden Attribute,

$c_{m+1}, \ldots, c_{m+t}$ Klassen in T,

$q = \pi_{\mathcal{U}} \circ \sigma_{A_{i_1}=c_1,\ldots,A_{i_{m+t}}=c_{m+t}}$ eine auf T erweiterte Projektion-Selektion-Anfrage.

Konstruiere das inter-dimensionale Muster

$p = (A_{i_1} \in \{c_1\}, \ldots, A_{i_m} \in \{c_m\}, A_{i_{m+1}} \in T(c_{m+1}), \ldots, A_{i_{m+t}} \in T(c_{m+t}))$, wobei

$T(c)$ die Menge der Blätter des Teilbaums in T mit Wurzel c ist.

Für Tabellen, die die Annahme 3.1.1 (xii) erfüllen, entspricht jedes Tupel t in der $\mathrm{supp}_R(p)$ nach Definition 3.3.3 zugrundeliegenden Menge genau einer Person $t(\mathcal{U})$ in $q(R)$, da zum einen t die Selektionsbedingung erfüllt und zum anderen alle die Selektionsbedingung erfüllenden Tupel in der $\mathrm{supp}_R(p)$ zugrundeliegenden Menge enthalten sind. Daher gilt $|q(R)| = \mathrm{supp}_R(p)$.

Insgesamt folgt, dass aus der Projektion-Selektion-Anfrage q und dem Wert von $|q(R)|$ nicht mehr Wissen inferiert werden kann als aus dem entsprechenden Muster-Support-Paar $(p, \mathrm{supp}_R(p))$.

Eine wesentliche Eigenschaft einer Menge S von Muster-Support-Paaren ist es, dass sie keine zwei Muster-Support-Paare mit gleichen Mustern, aber unterschiedlichen Supports besitzt. Anderenfalls wäre sie mit keiner Tabelle kompatibel. Diese Eigenschaft in der folgenden Definition 3.3.22 festgehalten und lässt sich in Polynomialzeit überprüfen.

Definition 3.3.22 (Unmittelbare Widerspruchsfreiheit).

Sei $S = \{(p_1, m_1), \ldots, (p_s, m_s)\} \subseteq Pat_{\mathrm{inter}}(A_1, \ldots, A_n) \times \mathbb{N}$ eine Menge von Muster-Support-Paaren.

Bezeichne

S als **unmittelbar widerspruchsfrei**, falls

für alle $(p, n), (p', n') \in S$ mit $p = p'$ gilt: $n = n'$.

Wenn die Data Mining Ergebnisse als eine Menge von Muster-Support-Paaren modelliert sind, lässt sich der naive Algorithmus 3.2 zur Erkennung und Beseitigung einiger inter-dimensionaler k-Inferenzkanäle anwenden.

Der Algorithmus 3.2 senkt den Support aller Muster p mit $S \vdash (p, i)$ und $0 < i < k$ auf 0, indem er alle von p beschriebenen Personen aus der Tabelle entfernt. Deshalb kann ein potentieller Angreifer keine inter-dimensionalen k-Inferenzkanäle in den sanitisierten Muster-Support-Paaren mittels des inter-dimensionalen Ableitungsoperators finden.

Der Algorithmus 3.2 terminiert, da bei jedem Schleifendurchlauf der inneren Schleife mindestens ein Tupel aus R' entfernt wird und ein unmittelbar widerspruchsfreier

Eingabe:

Population $\mathcal{U}$,

Tabelle $R(\mathcal{U}, A_1, \ldots, A_n)$,

Anonymitätsschwelle $k \in \mathbb{N}$,

Menge von Muster-Support-Paaren

$\{(p_1, \text{supp}_R(p_1)), \ldots, (p_m, \text{supp}_R(p_m))\} \subseteq 2^{Pat_{\text{inter}}(A_1,\ldots,A_n) \times \mathbb{N}}$

Ausgabe:

Sanitisierte Tabelle $R'(\mathcal{U}, A_1, \ldots, A_n)$,

sanitisierte Menge von Muster-Support-Paaren

$\{(p_1, \text{supp}_{R'}(p_1)), \ldots, (p_m, \text{supp}_{R'}(p_m))\} \subseteq 2^{Pat_{\text{inter}}(A_1,\ldots,A_n) \times \mathbb{N}}$

$R' \leftarrow R$

repeat

$S \leftarrow \{(p_1, \text{supp}_{R'}(p_1)), \ldots, (p_m, \text{supp}_{R'}(p_m))\}$

$S^* \leftarrow$ unmittelbar widerspruchsfreier Abschluss von S bezüglich der iterativen Anwendung der Ableitungsregeln des inter-dimensionalen Ableitungsoperators $\vdash$

for all $(p, i) \in S^*$ mit $0 < i < k$ **do**

$R' \leftarrow R' \setminus \{t \in R' \mid \text{supp}_{\{t\}}(p) = 1\}$

end for

until R' hat sich nicht verändert

return $R', \{(p_1, \text{supp}_{R'}(p_1)), \ldots, (p_m, \text{supp}_{R'}(p_m))\}$

Algorithmus 3.2: Beseitigung von inter-dimensionalen k-Inferenzkanälen

transitiver Abschluss von S bezüglich der Anwendung der Ableitungsregeln des inter-dimensionalen Ableitungsoperators höchstens $2^n * 2^{|\operatorname{dom}(A_1)|} * \ldots * 2^{|\operatorname{dom}(A_n)|}$ Elemente besitzt.

Zwar kann ein Inferenzkanal beseitigt werden, indem der Support eines Muster auf mindestens k angehoben wird, dies würde in der Praxis jedoch dazu führen, dass für ein sensibles Attribut A und einen (Quasi-)Identifikator $Name$ Muster wie $p = (Name \in \{\text{Max Mustermann}\}, A \in \{a\})$ mit einem relativ hohen Support aus S inferiert werden könnten. Die Möglichkeit einer solchen Inferenz verletzt offensichtlich Vertraulichkeitsanforderungen, da der Name »Max Mustermann« bereits eine kleine Gruppe von Personen identifizieren kann und das Muster p diese Gruppe mit der Ausprägung a des sensiblen Attributs A in Beziehung setzt.

Der Algorithmus 3.2 beseitigt nicht alle inter-dimensionalen k-Inferenzkanäle, da der inter-dimensionale Ableitungsoperator nicht vollständig ist. Nichtsdestotrotz ist er in der Lage Inferenzkanäle zu beseitigen, die bisherige Methoden nicht erfassen. Dazu werden Beispiele 2.2.19 und 3.3.1 noch einmal zusammen mit Algorithmus 3.2 betrachtet.

3.3.23 Beispiel (Erweiterung des Beispiels 3.3.1).
Für eine Tabelle $R(Kunde, Geschlecht, Produkt)$, die von Kunden gekaufte Produkte unter Angabe ihres Geschlechts beinhaltet, seien eine Taxonomie für Computer (Abbildung 3.3) sowie folgende Daten verfügbar:

- 100 männliche Kunden haben Computer gekauft,
- 99 männliche Kunden haben einen Client gekauft.

Diese Daten lassen sich als eine Menge von Muster-Support-Paaren $S = \{(p_1, m_1), (p_2, m_2)\}$ interpretieren mit

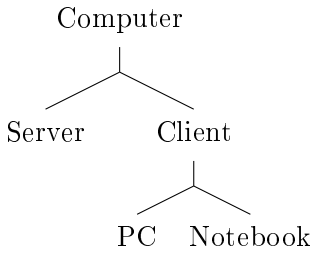
(i) $(p_1, m_1) = ((Geschlecht \in \{m\}, Produkt \in \{Server, PC, Notebook\}), 100)$,

(ii) $(p_2, m_2) = ((Geschlecht \in \{m\}, Produkt \in \{PC, Notebook\}), 99)$.

Da $\operatorname{diff}(p_1, p_2) = \{Produkt\}$ und
$p_1(Produkt) = \{Server, PC, Notebook\} \supset \{PC, Notebook\} = p_2(Produkt)$ gilt,
lässt sich die Regel (SUB) des inter-dimensionalen Ableitungsoperators anwenden und es folgt $S \vdash (p_3, m_3)$ mit
$(p_3, m_3) = (p_1[Produkt \in \{Server, PC, Notebook\} \setminus \{PC, Notebook\}], 100 - 99)$
$= ((Geschlecht \in \{m\}, Produkt \in \{Server\}), 1)$.

Damit wurde ein nicht 2-anonymes Muster gefunden. Nach der Entfernung der entsprechenden Tupel gilt $\operatorname{supp}_R(p_3) = 0$. Aus der Sicht eines potentiellen Angreifers ist wiederum unklar, ob Tupel aus der Tabelle entfernt worden sind oder kein Mann einen Server gekauft hat.

Abbildung 3.3: Taxonomie für Computer



3.3.24 Beispiel (Erweiterung des Beispiels 2.2.19).
Für eine Tabelle $R(Kunde, Alter, Geschlecht, Produkt)$, die von Kunden gekaufte Produkte unter Angabe ihres Alters und Geschlechts beinhaltet mit
$\{[0..39], [40..]\} = \mathrm{dom}(Alter), \{m, w\} = \mathrm{dom}(Geschlecht)$ und $Computer \in \mathrm{dom}(Produkt)$, seien folgende inter-dimensionale Assoziationsregeln mit jeweiligem Support verfügbar:

(I) $\top \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 200,

(II) $Geschlecht = m \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 100,

(III) $Geschlecht = w \wedge Alter = [0..39] \rightarrow_{\{Kunde\}} Produkt = Computer$, Support: 99.

Diese Daten lassen sich als eine Menge von Muster-Support-Paaren
$S = \{(p_1, m_1), (p_2, m_2), (p_3, m_3)\}$ interpretieren mit

(i) $(p_1, m_1) = ((Produkt \in \{Computer\}), 200)$,

(ii) $(p_2, m_2) = ((Geschlecht \in \{m\}, Produkt \in \{Computer\}), 100)$,

(iii) $(p_3, m_3) = ((Geschlecht \in \{w\}, Alter \in \{[0..39]\}, Produkt \in \{Computer\}), 99)$.

Da $\mathrm{diff}(p_1, p_2) = \{Geschlecht\}$ und
$p_1(Geschlecht) = \mathrm{dom}(Geschlecht) \supset \{m\} = p_2(Geschlecht)$ gilt, lässt sich die Regel (SUB) des inter-dimensionalen Ableitungsoperators anwenden und es folgt $S \vdash (p_4, m_4)$ mit $(p_4, m_4) = (p_1[Geschlecht \in \mathrm{dom}(Geschlecht) \setminus \{m\}], 200 - 100) = ((Geschlecht \in \{w\}, Produkt \in \{Computer\}), 100)$.

Da $\mathrm{diff}(p_4, p_3) = \{Alter\}$ und $p_4(Alter) = \mathrm{dom}(Alter) \supset \{[0..39]\} = p_3(Alter)$ gilt, lässt sich die Regel (SUB) des inter-dimensionalen Ableitungsoperators anwenden und es folgt $S \vdash (p_5, m_5)$ mit $(p_5, m_5) = (p_4[Alter \in \mathrm{dom}(Alter) \setminus \{[0..39]\}], 100 - 99) = ((Geschlecht \in \{w\}, Alter \in \{[40..]\}, Produkt \in \{Computer\}), 1)$.

Damit wurde ein nicht 2-anonymes Muster gefunden. Nach der Entfernung derjenigen Tupel, die durch das Muster p_5 beschrieben werden, und einer Anpassung der Supportwerte in S ist $\mathrm{supp}_R(p_5) = 0$ und es kann nicht mehr inferiert werden, dass eine Frau, die mindestens 40 Jahre alt war, einen Computer gekauft hat. Aus der Sicht eines potentiellen Angreifers ist somit unklar, ob diese Kundin etwas anderes gekauft hat, ihre Daten aus der Tabelle entfernt wurden oder ihre Daten in der Tabelle gar nicht enthalten waren.

3.3.5 Implementierung des Ableitungsoperators

Der inter-dimensionale Ableitungsoperator wurde zu Testzwecken mit Hilfe der funktionalen Programmiersprache **Haskell**[1] implementiert. Der Quellcode ist in Listing A.1 angeführt. Die Implementierung wurde mit der Intention entworfen, eine möglichst verständliche und erweiterbare Umgebung für Experimente im Zusammenhang mit dem Ableitungsoperator zu schaffen.

Der Zeichenkettentyp `String` wurde sowohl für Namen von Attributen als auch für Elemente der Domänen der Attribute gewählt. Damit entspricht ein inter-dimensionales Muster einer Funktion, die Zeichenketten auf Mengen von Zeichenketten abbildet.

Die Ableitungsregeln (ADD), (SUB) und (HALF) wurden in Methoden `applyPatternUnion`, `applyPatternDifference` und `applyPatternSymmetricDifference` realisiert. Dabei werden Muster mit Hilfe der Methode `comparePatterns` miteinander vergleichen, um die Anwendbarkeit der jeweiligen Regel zu überprüfen.

Die Methode `deducePatternsStep` leitet neue Muster-Support-Paare ab, die sich nach einmaliger Regelanwendung von (ADD), (SUB) oder (HALF) ergeben. Dabei wird die Anwendbarkeit jeder Regel für jedes Tupel bzw. Tripel von vorhandenen Muster-Support-Paaren überprüft. Neue Regeln können implementiert werden, indem die Anwendung einer entsprechenden Ableitungsregel zu `deducePatternsStep` hinzugefügt wird.

Die Methode `deducePatterns` wendet die Methode `deducePatternsStep` iterativ an und fügt neue Muster-Support-Paare zur Eingabe hinzu bis keine neuen Muster-Support-Paare abgeleitet werden können. Die Methode `patternsWithSupportsToList` stellt eine Menge von Muster-Support-Paaren als eine formatierte Liste dar.

Das Beispiel 3.3.25 demonstriert die Funktionsweise der Methoden `deducePatternsStep` und `deducePatterns`.

3.3.25 Beispiel (Erweiterung des Beispiels 3.3.13). Für eine Population, die mindestens 6 Personen umfasst, beschreibt die Tabelle $R(\mathcal{U}, A, B)$ (Abbildung 3.2) Ausprägungen der Attribute A mit $\mathrm{dom}(A) = \{a, b, c\}$ und B mit $\mathrm{dom}(B) = \{x, y, z\}$.

Sei die Menge S folgender inter-dimensionale-Muster-Support-Paare gegeben:
$(p_1, m_1) = ((B \in \{x, y\}), 4)$,
$(p_2, m_2) = ((A \in \{c\}, B \in \{x, y\}), 2)$,
$(p_3, m_3) = ((A \in \{a\}, B \in \{x, z\}), 2)$,
$(p_4, m_4) = ((A \in \{b\}, B \in \{x, z\}), 0)$,
$(p_5, m_5) = ((A \in \{a, b\}, B \in \{y, z\}), 2)$.

Das Listing 3.1 beschriebt die Kodierung der Muster aus S in Haskell.

[1] http://www.haskell.org/

Listing 3.1: Kodierung der Eingaben in Haskell

```
attributes = Map.fromList [("A", domainA), ("B", domainB)]
  :: Map.Map String Domain where
    domainA = Set.fromList ["a","b","c"] :: Domain
    domainB = Set.fromList ["x","y","z"] :: Domain

p1 = Map.fromList [("B", Set.fromList ["x","y"])] :: Pattern
p2 = Map.fromList [("A", Set.fromList ["c"]),
  ("B", Set.fromList ["x","y"])] :: Pattern
p3 = Map.fromList [("A", Set.fromList ["a"]),
  ("B", Set.fromList ["x","z"])] :: Pattern
p4 = Map.fromList [("A", Set.fromList ["b"]),
  ("B", Set.fromList ["x","z"])] :: Pattern
p5 = Map.fromList [("A", Set.fromList ["a", "b"]),
  ("B", Set.fromList ["y","z"])] :: Pattern
```

Nach Beispiel 3.3.13 erlaubt der inter-dimensionale Ableitungsoperator folgende Ableitungen der Tiefe 1:
$S \vdash (p_6, m_6) = ((A \in \{a, b\}, B \in \{x, y\}), 2)$,
$S \vdash (p_7, m_7) = ((A \in \{a, b\}, B \in \{x, z\}), 2)$.

Das formatierte Ergebnis der Methode `deducePatternsStep` ist in Listing 3.2 und entspricht diesen Ergebnissen.

Listing 3.2: Formatiertes Ergebnis der Methode `deducePatternsStep`

```
patternsWithSupportsToList $ deducePatternsStep attributes
  (Set.fromList [(p1, 4),(p2, 2),(p3, 2),(p4, 0),(p5, 2)])
==
[([("A",["a","b"]),("B",["x","y"])],2), --(p6,m6)
([("A",["a","b"]),("B",["x","z"])],2)] --(p7,m7)
```

Die Liste aller aus S ableitbarer Muster-Support-Paare mit Ableitungen der Tiefe größer als 1 ist:
$S \vdash (p_8, m_8) = ((A \in \{a, b\}, B \in \{x\}), 1)$,
$S \vdash (p_9, m_9) = ((A \in \{a, b\}, B \in \{y\}), 1)$,
$S \vdash (p_{10}, m_{10}) = ((A \in \{a, b\}, B \in \{z\}), 1)$,
$S \vdash (p_{11}, m_{11}) = ((A \in \{a, b\}), 3)$.

Das formatierte Ergebnis der Methode `deducePatterns` ist in Listing 3.3 angegeben und spiegelt dieses Resultat wieder.

Listing 3.3: Formatiertes Ergebnis der Methode `deducePatterns`

```
patternsWithSupportsToList $ deducePatterns attributes
  (Set.fromList [(p1, 4),(p2, 2),(p3, 2),(p4, 0),(p5, 2)])
==
[([("B",["x","y"])],4), --(p1,m1)
([("A",["c"]),("B",["x","y"])],2), --(p2,m2)
([("A",["a"]),("B",["x","z"])],2), --(p3,m3)
([("A",["b"]),("B",["x","z"])],0), --(p4,m4)
([("A",["a","b"]),("B",["y","z"])],2), --(p5,m5)
([("A",["a","b"]),("B",["x","y"])],2), --(p6,m6)
([("A",["a","b"]),("B",["x","z"])],2), --(p7,m7)
([("A",["a","b"]),("B",["x"])],1), --(p8,m8)
([("A",["a","b"]),("B",["y"])],1), --(p9,m9)
([("A",["a","b"]),("B",["z"])],1), --(p10,m10)
([("A",["a","b"])],3)] --(p11,m11)
```

Um das »worst case« Verhalten der naiven Implementierung zu untersuchen, wurde für n Attribute $A_1, \ldots, A_n$ mit m-elementigen Domänen $\{1, \ldots, m\}$ die Menge der Muster-Support-Paare $S = \{((A_1 \in \{a_1\}, \ldots, A_n \in \{a_n\}), 1) \mid (a_1, \ldots, a_n) \in \prod_{i=1}^{n} \{1, \ldots, m\}\}$ als Eingabe für `deducePatterns` verwendet. Aus S lassen sich alle $(2^m - 1)^n$ Muster-Support-Paare ableiten. Die Ausführungszeit in Abhängigkeit von n und m auf einem 3.4GHz Prozessor ist in der Tabelle 3.4 festgehalten. Es ist zu erkennen, dass die in der Anzahl der abgeleiteten Muster-Support-Paare kubische Laufzeit des naiven Algorithmus ihn für Größenordnungen von über 500 Muster-Support-Paaren praktisch untauglich macht.

Abbildung 3.4: Laufzeit von `deducePatterns` mit n Attributen mit m-elementigen Domänen. Format: Zeit in Sekunden ($(2^m - 1)^n$)

n \ m	1	2	3	4	5
1	<1s (1)	<1s (3)	<1s (7)	<1s (15)	<1s (31)
2	<1s (1)	<1s (9)	<1s (49)	27s (225)	4502s (961)
3	<1s (1)	<1s (27)	221s (343)	>7200s (3375)	
4	<1s (1)	2s (81)	>7200s (2401)		
5	<1s (1)	89s (243)	>7200s (16807)		

Kapitel 4

Fazit

In dieser Arbeit wurden mehrere Vorgehensweisen behandelt, potentielle Verletzungen von Vertraulichkeitsanforderungen bei der Veröffentlichung von häufig vorkommenden Mustern in relationalen Datenbanken zu erkennen und zu beseitigen. Eine potentielle Verletzung von Vertraulichkeitsanforderungen wurde über einen entsprechenden Begriff des k-Inferenzkanals beschrieben. Dabei wurde über das a priori Wissen in Abschnitt 2.2.3 ein Angreifermodell mit erlaubtem Vorwissen bzw. Nichtwissen über die zugrundeliegende Datenbank beschrieben. Das erlaubte Vorwissen steht in direktem Zusammenhang mit Quasi-Identifikatoren, die bei der k-Anonymität bezüglich relationaler Datenbanken [11] eine zentrale Rolle spielen.

Die Form der betrachteten Muster orientiert sich an Konzepten von Assoziationsregeln für relationale Datenbanken.

Da in der Literatur mehrere Arten von Assoziationsregeln für relationale Datenbanken beschrieben sind, wurden in dieser Arbeit zwei Arten von Mustern definiert und separat betrachtet.

Im Fall der intra-dimensionalen Assoziationsregeln wurden Ausprägungsmenge-Support-Paare als die zu veröffentlichenden Data-Mining-Ergebnisse betrachtet. Aufgrund der großen Ähnlichkeit zu klassischen Assoziationsregeln, die in dieser Arbeit in Lemma 3.2.16 aufgestellt und bewiesen wird, wurden die Ergebnisse aus [3] zur Beseitigung von k-Inferenzkanälen verwendet. Mit Hilfe von Satz 3.2.11 wurde ein Bezug zum a priori Wissen hergestellt. Anhand der Beispiele 3.2.5 und 3.2.6 wurde in dieser Arbeit gezeigt, dass die Betrachtung von a priori Wissen eine Anpassung der Supportwerte der Eingabe erfordert. Der potentielle Angreifer verfügt in der Literatur [3] über kein a priori Wissen, daher findet dort keine Anpassung der Supportwerte statt. Die gesamte Vorgehensweise zur Beseitigung von intra-dimensionalen k-Inferenzkanälen wurde in Algorithmus 3.1 festgehalten.

Im Fall der inter-dimensionalen Assoziationsregeln wurden in dieser Arbeit inter-dimensionale Muster definiert, um zum einen Data Mining Ergebnisse zu repräsentieren und zum anderen eine potentielle Verletzung von Vertraulichkeitsanforderungen zu

beschreiben. Anhand von Beispielen 2.2.19 und 3.3.1 wurden Schwächen der bisherigen Vorgehensweise [1], die durch die Vernachlässigung der Semantik von Tabellen entstehen, demonstriert.

Zur Erkennung der von der bisherigen Vorgehensweise nicht erfassten inter-dimensionalen k-Inferenzkanäle wurde in dieser Arbeit der inter-dimensionale Ableitungsoperator (Definition 3.3.12) definiert und untersucht. Die Korrektheit des inter-dimensionalen Ableitungsoperators im Kontext von a priori Wissen wurde mit Satz 3.3.16 bewiesen. Die Vorgehensweise zur Beseitigung einiger inter-dimensionaler k-Inferenzkanäle wurde in Algorithmus 3.2 festgehalten. Da der inter-dimensionale Ableitungsoperator nicht vollständig ist, lassen sich mit seiner Hilfe in Algorithmus 3.2 nicht alle Inferenzkanäle erkennen und beseitigen.

In dieser Arbeit wurde mit Satz 3.3.8 gezeigt, dass das Supportinferenzproblem, das im direkten Zusammenhang mit der Erkennung von inter-dimensionalen k-Inferenzkanälen steht, Co-NP als eine untere Komplexitätschranke besitzt.

4.1 Ausblick

Sowohl im Bereich der Anonymisierung von intra-dimensionalen als auch von inter-dimensionalen Assoziationsregeln bestehen weiterhin offene Fragen.

1) Die Homomorphie zum binären Datenbankmodell macht im Fall der intra-dimensionalen Assoziationsregeln Ergebnisse des binären Datenbankmodells anwendbar. Es wäre sinnvoll, ohne den Umweg über das binäre Datenbankmodell Inferenzkanäle zu erkennen und zu beseitigen.

2) Der in dieser Arbeit definierte inter-dimensionale Ableitungsoperator besitzt mehrere Schwächen. Seine größte Schwäche ist die fehlende Vollständigkeit. Daher kann er im Allgemeinen nicht zur Sicherstellung der Inferenzkanalfreiheit genutzt werden. Hierfür wäre es interessant, eine Vorgehensweise zu entwickeln, die durch Überapproximation zwar zu viele Inferenzkanäle erkennt, dafür jedoch vollständig ist.

Das Beispiel 3.3.25 hat zwar gezeigt, dass jede formulierte Ableitungsregel nützlich ist, es ist jedoch möglich, dass zusätzliche oder andere Regeln zur besseren Erkennung von Inferenzkanälen führen.

Darüber hinaus muss die Beziehung des Ableitungsoperators zum in [3] zur Bestimmung von Inferenzkanälen genutzten Prinzips der Inklusion und Exklusion näher untersucht werden. Es ist möglich, dass bereits die Regeln (AX), (ADD) und (SUB) ausreichen, um dieses Prinzip zu erfassen.

3) Es ist sinnvoll, das inter-dimensionale Supportinferenzproblem bezüglich einer eingeschränkten Form inter-dimensionaler Muster zu untersuchen. Da in nicht-erweiterten inter-dimensionalen Assoziationsregeln jedes Attribut höchstens einmal vorkommen darf, ließe die Betrachtung einer entsprechenden Einschränkung der Eingabe des Supportinferenzproblems eine bessere Aussage über die Anonymisierung solcher Assoziationsregeln treffen.

Analog kann für mehrstufige Assoziationsregeln gelten, dass alle Ausprägungsmengen eines Attributs Blätter eines Teilbaums einer Taxonomie sind. Diese Eigenschaft schließt eine teilweise-Überlappung von Ausprägungsmengen aus und kann zu einer anderen Komplexitätsschranke bei der Anonymisierung von mehrstufigen Assoziationsregeln führen. Darüber hinaus wäre die Betrachtung von Mengen von Muster-Support-Paaren, die mit mindestens einer Tabelle kompatibel sind, für das Supportinferenzproblem interessant, denn in der Praxis sind diese Mengen mindestens mit der ursprünglichen Tabelle kompatibel.

4) Für die Praxis ist es relevant, Abschwächungen der Annahmen (viii) und (ix) in Abschnitt 3.1.1, die das a priori Wissen des potentiellen Angreifers stark einschränken, zu betrachten. Zusätzlich wäre eine intensionale Darstellung von a priori Wissen nützlich, z.B. durch eine Menge prädikatenlogischer Formeln, die genau die zum a priori Wissen gehörenden Tabellen charakterisieren.

5) Sowohl Algorithmus 3.1 als auch Algorithmus 3.2 sind naive Umsetzungen der theoretischen Ergebnisse. Sowohl ihre Performance als auch ihre Skalierbarkeit können für einen praktischen Einsatz verbessert werden.

Anhang A

Anhang

Listing A.1: Implementierung des inter-dimensionalen Ableitungsoperators in Haskell

```
module Main where

import qualified Data.Map as Map
import qualified Data.Set as Set
import Data.Maybe

--domain ~ set of strings
type Domain = Set.Set String
--attributes ~ (attribute name -> attribute domain)
type Attributes = Map.Map String Domain

--get domain of attribute attributeName in the map attrs
getDomain attrs attributeName =
 Map.findWithDefault (Set.empty::Set.Set String) attributeName attrs

--inter dimensional pattern ~ map attribute name -> attribute domain subset
type Pattern = Map.Map String (Set.Set String)

--convert pattern to a list of tuples (attribute name, domain seubset)
patternToList :: Pattern -> [(String,[String])]
patternToList pattern =
  map (\(attributeName,domainSubset)->
    (attributeName, Set.toList domainSubset)) $ Map.toList pattern
--convert patterns with support
--to a list of tuples ((attribute name, domain seubset), support)
patternsWithSupportsToList patternsWithSupports =
  map (\(pattern, support)->(patternToList pattern, support)) $
   Set.toList patternsWithSupports

--possible ways two sets can be in relation to each other
data SetRelation =
```

```
  Equal | Subset | Superset | Disjoint | Incomparable deriving (Eq, Show,
    Read)
34
--remove statements of the form "attribute in whole domain" from pattern
normalisePattern :: Attributes -> Pattern -> Pattern
normalisePattern attrs p =
  Map.filterWithKey (\attributeName domainSubset->
    (getDomain attrs attributeName) /= domainSubset) p
40


--given the map attribute name -> attribute domain and two patterns p1 and
    p2
--compare p1 and p2 and return a map with set relations of each domain
    subset
comparePatterns :: Map.Map String Domain -> Pattern -> Pattern ->
  Map.Map String SetRelation
comparePatterns attrs p1 p2 = Map.mapWithKey compareDomainSubsets attrs
    where
  compareDomainSubsets :: String -> Domain -> SetRelation
  compareDomainSubsets attributeName attributeDomain =
    case (Map.lookup attributeName p1, Map.lookup attributeName p2) of
      (Nothing, Nothing) -> Equal
      (Just domainSubset1, Nothing) ->
        if domainSubset1 == attributeDomain then Equal else Subset
      (Nothing, Just domainSubset2) ->
        if domainSubset2 == attributeDomain then Equal else Superset
      (Just domainSubset1, Just domainSubset2)
        | domainSubset1 == domainSubset2 -> Equal
        | Set.isProperSubsetOf domainSubset1 domainSubset2 -> Subset
        | Set.isProperSubsetOf domainSubset2 domainSubset1 -> Superset
        | Set.null $ Set.intersection domainSubset1 domainSubset2 ->
    Disjoint
        | otherwise -> Incomparable

--apply rule ADD if possible
applyPatternUnion attrs (p1, n1) (p2, n2) =
  case Map.toList $ Map.filter (/= Equal) $ comparePatterns attrs p1 p2 of
    [(attributeName, Disjoint)] ->
      Just (if domain == domainSubsetUnion
        then Map.delete attributeName p1
        else Map.insert attributeName domainSubsetUnion p1, n1+n2) where
          domain = getDomain attrs attributeName
          domainSubset1 = Map.findWithDefault domain attributeName p1
          domainSubset2 = Map.findWithDefault domain attributeName p2
          domainSubsetUnion = Set.union domainSubset1 domainSubset2
    _ -> Nothing
74

--apply rule SUB if possible
```

```
76 applyPatternDifference attrs (p1, n1) (p2, n2) =
   case Map.toList $ Map.filter (/= Equal) $ comparePatterns attrs p1 p2 of
78   [(attributeName, Superset)] ->
       Just (Map.insert attributeName
80       (Set.difference domainSubset1 domainSubset2) p1, n1-n2) where
         domain = Map.findWithDefault (Set.empty::Set.Set String)
82         attributeName attrs
         domainSubset1 = Map.findWithDefault domain attributeName p1
84       domainSubset2 = Map.findWithDefault domain attributeName p2
     _ -> Nothing
86
--apply rule HALF if possible
88 applyPatternSymmetricDifference attrs (p1, n1) (p2, n2) (p3, n3) = let
   comparePatterns12 =
90   Map.toList $ Map.filter (/= Equal) $ comparePatterns attrs p1 p2
   comparePatterns13 =
92   Map.toList $ Map.filter (/= Equal) $ comparePatterns attrs p1 p3
   comparePatterns23 =
94   Map.toList $ Map.filter (/= Equal) $ comparePatterns attrs p2 p3 in
       case (comparePatterns12,comparePatterns13,comparePatterns23) of
96     ([(attributeName1, Incomparable)], [(attributeName2, Incomparable)],
         [(attributeName3, Incomparable)]) ->
98       if attributeName1 == attributeName2 &&
           attributeName1 == attributeName3 &&
100        domainSubset3 == domainSubsetSymmetricDifference
         then Just (Map.insert attributeName1 domainSubsetIntersection p1,
102        quot (n1+n2-n3) 2)
         else Nothing where
           domain = getDomain attrs attributeName1
           domainSubset1 = Map.findWithDefault domain attributeName1 p1
106        domainSubset2 = Map.findWithDefault domain attributeName1 p2
           domainSubset3 = Map.findWithDefault domain attributeName1 p3
108        domainSubsetUnion = Set.union domainSubset1 domainSubset2
           domainSubsetIntersection =
110          Set.intersection domainSubset1 domainSubset2
           domainSubsetSymmetricDifference =
112          Set.difference domainSubsetUnion domainSubsetIntersection
       _ -> Nothing
114
--apply rules ADD, SUB, HALF and return new (pattern, support) tuples
116 deducePatternsStep :: Attributes -> Set.Set (Pattern, Int) ->
   Set.Set (Pattern, Int)
118 deducePatternsStep attrs patterns = Set.difference allPatterns patterns
     where
   patternsList = Set.toList patterns
120 unionPatterns = Set.fromList $ catMaybes
     [applyPatternUnion attrs p1 p2 |p1<-patternsList, p2<-patternsList]
122 differencePatterns = Set.fromList $ catMaybes
```

```
    [applyPatternDifference attrs p1 p2 |p1<-patternsList, p2<-patternsList]
  symmetricDifferencePatterns = Set.fromList $ catMaybes
    [applyPatternSymmetricDifference attrs p1 p2 p3 |
      p1<-patternsList, p2<-patternsList, p3<-patternsList]
  allPatterns = Set.unions
    [unionPatterns, differencePatterns, symmetricDifferencePatterns]

--iteratively apply rules ADD, SUB, HALF as long as new tuples are created
deducePatterns :: Attributes -> Set.Set (Pattern, Int) -> Set.Set (Pattern,
    Int)
deducePatterns attrs patterns | Set.null newPatterns = patterns
                              | otherwise = deducePatterns attrs
                                (Set.union patterns newPatterns) where
  newPatterns = deducePatternsStep attrs patterns
```

Algorithmenverzeichnis

Literatur

[1] Charu C. Aggarwal und Philip S. Yu, Hrsg. *Privacy-Preserving Data Mining - Models and Algorithms*. Bd. 34. Advances in Database Systems. Springer, 2008. ISBN: 978-0-387-70991-8.

[2] Rakesh Agrawal, Tomasz Imielinski und Arun N. Swami. »Mining Association Rules between Sets of Items in Large Databases«. In: *SIGMOD Conference*. 1993, S. 207–216.

[3] Maurizio Atzori, Francesco Bonchi, Fosca Giannotti und Dino Pedreschi. »Anonymity preserving pattern discovery«. In: *VLDB J.* 17.4 (2008), S. 703–727.

[4] Claudio Bettini, Xiaoyang Sean Wang und Sushil Jajodia. »How Anonymous Is k-Anonymous? Look at Your Quasi-ID«. In: *Secure Data Management*. 2008, S. 1–15.

[5] Joachim Biskup. *Security in Computing Systems - Challenges, Approaches and Solutions*. Springer, 2009, S. I–XXVII, 1–694. ISBN: 978-3-540-78441-8.

[6] Bart Goethals, Wim Le Page und Heikki Mannila. »Mining Association Rules of Simple Conjunctive Queries«. In: *SDM*. 2008, S. 96–107.

[7] Venkatesan Guruswami und Luca Trevisan. »The Complexity of Making Unique Choices: Approximating 1-in- k SAT«. In: *APPROX-RANDOM*. 2005, S. 99–110.

[8] Jiawei Han, Micheline Kamber und Jian Pei. *Data Mining: Concepts and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123814790, 9780123814791.

[9] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer und Muthuramakrishnan Venkitasubramaniam. »l-Diversity: Privacy Beyond k-Anonymity«. In: *ICDE*. 2006, S. 24.

[10] Thi Kim Ngan Nguyen. »Generalizing Association Rules in N-ary Relations: Application to Dynamic Graph Analysis«. en. Diss. INSA de Lyon, Okt. 2012. URL: `http://liris.cnrs.fr/publis/?id=5804`.

[11] Latanya Sweeney. »k-Anonymity: A Model for Protecting Privacy«. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (2002), S. 557–570.

Selbstständigkeitserklärung

Hiermit versichere ich, Andrej Dudenhefner, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 28. Oktober 2013

Andrej Dudenhefner