Vertiefungsprüfung Datenmanagement und Datenexploration

Datum 22.03.2007

Prüfling Thorben Keller

Note 1.3

Fächer Datenbanken, Data Mining, Datenexploration

Prüfer Prof. Seidl

Womit wollen sie denn anfangen?

Ich denke Datenbanken wär gut, ist mir aber eigentlich auch egal.

Ja gut. Dann machen wir Datenbanken, Daten Exploration und Data Mining.

Jo.

Datenbanken

Wenn sie eine Datenbank entwerfen, wie gehen sie dabei vor?

Zuerst machen wir eine Anforderungsanalyse, dazu befragen wir dann eine Menge Leute und finden raus, welchen Teil der realen Welt wir eigentlich modellieren wollen. Die Modellierung findet dann in der konzeptuellen Modellierungsphase statt, wo auch die verschiedenen Sichten konsolidiert werden um ein globales Schema zu erhalten, dann folgt die Auswahl eines Datenbankmodells, am Schluss dann die Implementierung.

Genau. Und welche Datenmodelle gibt es?

Da gibts einige,

- das Netzwerkmodell,
- das relationale Modell,
- das Objektorientierte Modell,
- das Objekt-relationale Modell und
- das deduktive Modell.

Ja genau. Wie unterscheiden sich denn das relationale und das objektorientierte Modell?

Da gibts so einiges,

- beim relationalen Modell werden Objekte stark segmentiert und auf viele Relationen aufgeteilt. Will man dann ein komplettes Objekt haben, dann müssen wir erst umständlich durch die Gegend joinen.
- Für die Einbettung in externe Programmiersprachen bietet sich da eher das objektorientierte Modell an (kurz noch den Begriff *Impendance Mismatch* eingeworfen).
- Bei der objektorientierten Modellierung haben wir einen systemweit eindeutigen und automatisch generierten Schlüssel, beim relationalen Modell müssen wir das umständlich mit Fremdschlüsseln realisieren.
- Dumm ist auch, das beim relationalen Modell keine Operationen machbar sind, das muss dann alles extern gemacht werden.

Aha. Was ist denn eigentlich so ein Schlüssel?

Ein Schlüssel ist eine Menge von Attributen α so dass die Relation voll funktional abhängig von α ist. Aufgeschrieben:

$$\alpha \dot{\rightarrow} R$$

Wenn α nicht minimal ist, dann sprechen wir von einem Superschlüssel.

Was bedeutet denn *funktional abhängig?*

Am obigen Beispiel bedeutet es, dass alle Attributewerte aus R durch die Attributwerte von α bestimmt sind.

Wie wird denn die konzeptuelle Ebene modelliert?

Das machen wir mit dem ER-Modell oder mit UML.

Wie unterscheiden sich die Beiden?

Bei UML werden auch noch Operationen auf den Objekten mit reingenommen.

Ich male ihnen jetzt einmal folgendes Schema auf:

$$\{[MatrNr, Name, VorlNr, Titel]\}$$

Ist das gut oder eher nicht?

Nee, das ist voll gar nicht gut, da werden Informationen zu mehreren Entities in einer Relation vermischt, dass Ding ist also noch nichteinmal in 2NF. Dabei können dann auch verschiedene Anomalien auftreten.

Wie 2NF? Die Begründung ist ein bisschen schwammig, wie sieht denn 2NF formal aus?

Ein Schema ist in 2NF, wenn jedes nicht-Schlüsselattribut voll funktional abhängig ist von jedem Schlüsselkandidaten.

Was sind das denn für Anomalien, die da auftreten können?

Also an obigem Beispiel sähe das dann wie folgt aus:

- Wollen wir jetzt z.B. eine neue Vorlesung anlegen, dann würden wir nicht wissen, welche Informationen wir für den Studenten eintragen sollen. Da müssten man dann dumm mit $NULL$ Werten rumwerfen. (Einfüge-Anomalie)
- Wenn wir z.B. den Titel einer Vorlesung ändern möchten, dann müssen wir die ganze Datenbank entlang laufen um alle Tupel zu finden, die auch mit dieser Vorlesung zu tun haben, das können evtl. ziemlich viele werden. Vom Speicherplatz will ich da gar nicht erst anfangen... (Update-Anomalie)
- Und wenn wir jetzt einen Studenten löschen und es ist der einzige, der eine bestimmte Vorlesung hört, dann gehen auch die Informationen zu der Vorlesung verloren. (Lösch-Anomalie)

Ja genau. Und kann man das jetzt irgendwie verbessern?

Ja, wir können das in 3NF bringen, das machen wir mit dem Synthesealgorithmus.

Dann machen sie mal.

Zuerst müssen wir die kanonische Überdeckung machen. Dazu machen wir Linksreduktion, Rechtsreduktion und noch ein paar andere Sachen. Dann schmeissen wir einige Relationen weg und bauen evtl. eine neue.

(Guckt mich ziemlich skeptisch an) Nee... machen sie bitte mal richtig!

Also gut. Zuerst brauchen wir die funktionalen Abhängigkeiten. Das sind im Einzelnen

- $MatrNr \rightarrow Name$
- $VorlNr \rightarrow Titel$
- $Matr, VorlNr \rightarrow Name, Titel$

Bei der Rechtsreduktion fliegt auf jeden Fall die letzte raus. Das sieht man ja direkt. Dann machen wir aus den beiden anderen jeweils eine Relation also

- Student: $\{[Matr, Name]\}$
- Vorlesung: $\{[VorlNr, Titel]\}$

Da jetzt keine dieser beiden Relationen eine Schlüssel bzgl. der kompletten Relation enthält brauchen wir noch eine neue:

- Hören: $\{[MatrNr, VorlNr]\}$

Ok. Was kennen sie denn für Anfragsprachen für relationale Datenbanken?

- Relationale Algebra
- Tupelkalkül
- Domänenkalkül
- SQL

Wie sieht denn allgemein eine SQL Anfrage aus?

SELECT $A_1, \ldots, A_n$ FROM $R_1, \ldots, R_m$ WHERE P;
Der WHERE Teil ist dabei optional.

Schön. Kennen sie noch andere SQL Befehle?

Ja, z.B. GROUP BY mit HAVING.

Ja die gibts auch aber das meinte ich nicht. Muss das denn immer ein SELECT sein?

Achso. Nein natürlich nicht, das kann z.B auch SELECT DISTINCT sein.

Das meinte ich jetzt auch nicht. Was kann denn da anstatt von SELECT stehen?

Achso. Wir können natürlich nicht nur Sachen suche, sondern auch einfügen und so. Das macht man dann mit INSERT. Sonst gibts auch noch Dinge wie CREATE TABLE, DROP TABLE, SHOW TABLES und so weiter.

Wie unterscheiden sich denn z.B. SELECT und CREATE TABLE?

Hmm... die arbeiten ja auf ner unterschiedlichen Ebene. SELECT arbeitet auf den Tabellen, CREATE auf der Datenbankebene.

Ja schon. Aber...

(hab voll keinen Plan)

Sagt ihnen DML und DDL was?

Öhh.... nee!

Also DML ist die Datenmanipulationssprache, DDL die Datenbeschreibungssprache!

Achso, ja genau. Also CREATE arbeitet auf dem Schema und SELECT auf den Relationen.

Genau. Und toll ist vorallem, dass beim relationalen Modell das Schema auch in Relationen steht, man braucht da also keine extra Sachen. Dann machen wir jetzt mal Datenexploration.

Modelle der Datenexploration

Was machen wir eigentlich bei der Datenexploration?

Also wir haben ein Anfrageobjekt und wollen dann entweder alle Objekte finden die einen gewissen Grad von Ähnlichkeit aufweisen oder wir wollen z.B. das ähnlichste Objekt haben.

Und wie funktioniert das mit der Ähnlichkeitsbestimmung?

Also wir haben unsere Objekte immer in einen n-dimensionalen Vektorraum abgebildet, und dann haben wir da den Abstand gemessen.

Wie kann man die Ähnlichkeit denn sonst noch bestimmen?

Wir können z.B. auch den Winkel zwischen den Vektoren als Ähnlichkeitsmaß verwenden. Dazu nehmen wir dann die Kosinus-Distanz.

Und wie sieht jetzt formal so eine Bereichsanfrage aus?

$$range(q, \varepsilon) = \{o \in DB | d(o, q) \leq \varepsilon\}$$

Und wie findet man jetzt solche Objekte?

Dazu machen wir einen sequenziellen Scan über der Datenbank. Das läuft dann in $O(n)$, wächst also linear in der Anzahl der Objekte.

Was gibt es denn sonst noch für Anfragen?

- Also bei der Bereichsanfrage finden wir ja alle Objekte die bzgl. q eine gewisse Ähnlichkeit aufweisen.
- Mit einer Nächste-Nachbar Anfrage finden wir dann **das** ähnlichste Objekt.
- Die Nächste-Nachbar Anfrage können wir auch verallgemeinern auf k Nächste Nachbarn.

Gibts da sonst noch was?

Jo stimmt.

- Wenn wir weder einen guten Wert für ε oder q kennen, dann machen wir eine Ranking Anfrage. Dabei liefert uns die Funktion $getnext(k)$ jeweils die nächsten k Objekte. Das machen wir dann so lange, bis wir entweder genug Objekte gefunden haben, oder uns die gelieferten Objekte zu unähnlich geworden sind.

Mal abgesehen von der euklidischen Distanz. Wie kann man Ähnlichkeit denn noch bestimmen?

Also abgesehen von der euklidischen Distanz und der Kosinus DIstanz kann man auch Quadratische Formen verwenden. Dabei werden dann auch noch Korrelationen zwischen den einzelnen Dimensionen berücksichtigt.

Schreiben sie die Quadratische Form mal auf!

$$d_A(q, o) = \sqrt{(p - q) \cdot A \cdot (p - q)^T}$$

Und was ist daran jetzt besser als bei der euklidischen Distanz?

Öhh... naja wie gesagt da berücksichtigen wir auch Dimensionkorrelationen.

Ja schon. Aber warum machen wir das denn?

Hmm... also ohne Quadratische Formen können wir ja eigentlich auch Objekte finden, die eigentlich gar nicht ähnlich zueinander sind, aber von der euklidischen Distanz trotzdem als ähnlich eingestuft werden.

Wie aufwendig ist denn die Berechnung der Quadratischen Form?

Also Vektor $\times$ Matrix $\times$ Vektor: Das ist natürlich kubisch, also $O(N^3)$.

???

???

Die Quadratische Form kann man ja auch anders aufschreiben. Machen sie das mal!

Jo:

$$d_A(p,q) = \sqrt{\sum_{i,j} a_{ij}(p_i - q_i)(p_j - q_j)}$$

Ja gaaaar nicht. Da sieht man ja direkt das dass quadratisch ist, also in $O(n^2)$ liegt. (Super, Thorben. Das Grundstudium lässt grüßen)

Na gut. Was für Objekte haben wir denn eigentlich alles betrachtet in der Vorlesung?

So einiges:

- Zeitreihen und allgemeine Sequenzen
- Bilder
- Formen

Da gabs noch was anderes!

Jo, stimmt. Wie wärs mit

- Graphen?

Genau. Wie vergleicht man denn jetzt z.B. Bilder?

Bilder kann man anhand der folgenden Dinge unterscheiden:

- Anhand der Farbe, daraus bauen wir uns dann ein Farbhistogramm,
- Texturen und
- Formen

Was ist denn ein Histogramm?

Also um mal bei unserem Bildbeispiel zu bleiben: Wir teilen uns denn Farbraum in n Teile, Partitionen oder Bins ein. Das kann man dann als Vektor auffassen, wobei jeder Bin einer Vektordimension entspricht. Die kann man dann direkt vergleichen.

Aha. Und wie viele Dimensionen nimmt man dann so bei Bildern?

Also z.B. 256 oder 16 oder so. Das ist halt abhängig von der Farbtiefe im Bild. Haben wir nämlich ein Schwarzweiss Bild, dann benötigen wir natürlich keine 256 Dimensionen.

Und wie baut man sich jetzt so ein Histogramm?

Also wie gesagt haben wir den Farbraum ja bereits partitioniert. Jetzt laufen wir pixelweise durch das Bild und erhöhen dann den Zähler des Bins des nächstgelegenden Repräsentanten um 1.

So. Und wenn wir dann jetzt mal zurück zu den Quadratischen Formen kommen. Was ist denn jetzt an diesem Beispiel besser, wenn man die verwendet?

Also wenn wir uns die Farben angucken, dann können wir halt berücksichtigen, dass Rot ähnlicher ist zu Pink als zu Schwarz.

Na gut. Also das ist ja wie gesagt alles ziemlich aufwendig. Wie kann man das denn verbessern?

Dazu können wir dann z.B. eine Filter verwenden.

Aha. Und was macht denn so ein Filter?

Also der findet eine Obermenge der gesuchten Objekte. Dazu verwenden wir eine Filterdistanz die kleiner ist als die exakte Distanz.

Schreiben sie das doch mal formal auf!

$$d_f(q, o) \leq d_e(q, o)$$

d_f ist dann die Filterdistanz und d_e ist die exakte Distanz.

Richtig. Und wie bekommt man jetzt den Link zwischen der Teilmengen Beziehung und der Kleiner Relation hin?

Watt?!?

Also wir haben ja eine Obermenge und die Kleiner Beziehung. Wo ist denn da der Zusammenhang?

Öhh... Uff... (hier haben wir uns dann Ewigkeiten aufgehalten. Ich hab gesagt, dass die Filterdistanz ja gerade eine Obermenge liefert WEIL sie Objekte als näher dran betrachtet. Das war aber wohl nicht so richtig. Ich muss zugeben, dass ich bis jetzt nicht weiss, worauf Prof. Seidl hinaus wollte...)

Also gut, machen wir mal weiter. Wie kann man denn die Güte eines Filter bestimmen?

Dafür haben wir die ICES Kriterien:

- Ein Filter sollte indexierbar sein, sprich wir wollen mit der Filterdistanz die Dimensionen so weit drücken, dass wir einen Index darüber schmeissen können.
- Natürlich muss er auch vollständig sein. Das beweist man dann in der Regel formal mit der Lower Bounding Property.
- Effizient zu berechnen liegt ja sowieso in der Natur der Sache und
- eine gute Selektivität sollte er aufweisen. Sprich die Obermenge sollte so klein wie möglich sein.

Ok, dann kommen wir jetzt mal zu Data Mining!

Yup.

Data Mining

Was ist denn überhaupt der KDD Prozess?

Ja also, der setzt sich wie folgt zusammen:

1. Sammeln der Daten,
2. Integrieren, Transformieren und Säubern der Daten
3. Das eigentliche Data Mining, sprich Clustering, Klassifikation oder Assoziationsregeln
4. Visualisieren der Ergebnisse (das wollte er unbedingt hören)

Gut. Wir hatten da ja noch was anderes, oder?

Ja, den ganzen Generalisierungskram.

Wenn sie jetzt mal an die Relation von vorhin denken, da will der Dekan jetzt nicht alle Matrikelnummern haben, sondern das soll auf ein Blatt passen. Wie machen sie das denn?

Attribute Oriented Induction.

Richtig. Was ist denn AOI?

Also das ist so wie OLAP, nur dass da die Generalisierung nicht manuell sondern automatisch vorgenommen wird. Der Benutzer kann also vorgeben, wie weit man die Dinge konzeptuell modelliert haben will oder wie weit man Daten diskretisieren möchte.

Ok, aber der Benutzer kann ja noch was anderes vorgeben. Wie parametrisiert man denn jetzt den ganzen AOI Prozess?

??? (Kein Plan... hier haben wir uns auch wieder länger aufgehalten, bin nicht von selber darauf gekommen, dass der Benutzer auch z.B. vorgeben kann wieviele Daten man am Ende haben will, den Hint mit dem dass soll auf ein Blatt passen habe ich einfach mal gekonnt ignoriert, haha)

So. Wir hatten ja auch Klassifikation und Clustering. Wie unterscheiden sich die Beiden Dinge denn?

Also beim Clustering geht es darum Objekte so in Cluster einzuteilen, dass sie ähnlich zu den Objekten im eigenen Cluster sind und unähnlich zu den Objekten in den anderen Clustern. Klassifikation ist etwas völlig anderes, da haben wir eine Menge von Objekten gegeben, die bereits eine Klassenzugehörigkeit haben und wir wollen nun einen Klassifikator finden, der neue Objekte auf eine Klasse abbildet.

Was hatten wir denn für das Clustering so für Ansätze?

Voll viel. Wir hatten Ansätze mit nem festen k, das ist aber öde, weil wir ja nicht immer wissen, wieviele Cluster wir benötigen …

... welche kennen sie da?

- Erwartungsmaximierung,
- k-Means und
- k-Medoid.

Wie funktionieren denn k-Means und k-Medoid?

Also zuerst bilden wir eine nicht-leere Partitionierung des Datenraums in k Partitionen. Dann bestimmen wir bei k-Means die Mittelwertvektoren und weisen alle Objekte der Partition mit dem nächstgelegenden Vektor zu. Bei k-Medoid funktioniert dass dann so, dass der Medoid gefunden werden muss. Dazu haben wir 2 Algorithmen kennengelernt die das machen. Der ganze Prozess läuft dann iterativ ab, und wir hören auf, wenn sich die Clusteringgüte nicht mehr sonderlich verbessert...

... wie kann man denn die Güte eines Clusterings bestimmen?

Das machen wir mit dem Silhouetten Koeffizient, der bestimmt die durchschnittliche Distanz der Objekte zu den jeweiligen Clusterzentren.

Stimmt. Und warum bestimmen wir die Güte nicht anhand der Kompaktheit?

Die Clusterkompaktheit nähert sich mit steigendem k ja der Null, wir würden also dazu tendieren, die Clusteranzahl immer weiter zu erhöhen.

Genau. Was wäre denn z.B. bei $k = n$ oder $k = 1$?

Also bei $k = 1$ sind alle Objekte im gleichen Cluster. Das ist nicht so schön und bei $k = n$ ist die Gesamtdistanz Null, weil ja jedes Objekt durch sich selber repräsentiert wird.

Und mit welchem k fangen wir dann jetzt an?

(Keine Ahnung wie ich hier argumentiert habe)

Kann man denn eigentlich immer beide anwenden?

Nein, bei k-Means muss der Mittelwert definiert sein...

... wann ist das denn nicht so?

Bei kategorischen Werten wird es eher schwierig. (hier ging es dann noch darum welcher denn jetzt eigentlich allgemeiner ist und warum dem so ist, das waren viele kleine Fragen an die ich mich aber nicht mehr erinnern kann.)

So. Sie sagten ja, dass mit dem k ist nicht so toll. Wie geht es denn anders?

Wir haben uns ja noch mit dichtebasiertem und hierarchischem Clustering beschäftigt.

Wie funktioniert denn dichtebasiertes Clustering?

Also ausgehend von einem Kernobjekt bilden wir die Dichte-Erreichbarkeit, dass ist die transitive Hülle der direktendichte-erreichbarkeit. Das Ganze hat dann natürlich noch den Vorteil, dass wir Rauschen erkennen, das sind nämlich die Objekte die selber kein Kernobjekt sind und auch nicht in der Hülle liegen. Ausserdem sind wir auch viel flexibler, was die Form der Cluster angeht.

Ok. Und was ist ein Kernobjekt bzw. wie parametrisiert man das Ganze jetzt?

Ein Kernobjekt ist ein Objekt in dessen ε-Umgebung mindestens $MinPts$ viele Objekte liegen. Diese beiden Werte müssen wir am Anfang natürlich vorgeben.

Genau aber was sind denn gute Werte dafür?

Also dazu nehmen wir uns den am wenigsten dichten Cluster und bestimmen die Werte daraus.

Und wie funktioniert hierarchiches Clustering?

Das Ganze basiert auf der Tatsache, dass dichte Cluster vollständig in weniger dichten Clustern enthalten sind. Wir bauen uns dann ein Dendogramm, aus welchem wir dann manuell Knoten auswählen um eine Clusterstruktur zu erhalten.

Was ist denn ein Dendogramm?

Das ist ein Baumstruktur, alle Objekte in den Kinder Knoten sind geometrisch im darüber liegenden Knoten enthalten. Das ist so ähnlich wie beim R-Baum. Die Kanten sind dann noch mit den entsprechenden Distanzen beschriftet.

Und wie bestimmt man jetzt so eine hierarchie? Was macht denn eigentlich Optics?

Also Optics ist eine Erweiterung des DBSCAN Algorithmus. Wir durchsuchen hier mit mehreren ε Werten gleichzeitg den Datenraum...

... Richtig. Und was liefert der Algorithmus für eine Ausgabe?

Hmm... (hier stand ich voll aufm Schlauch. Bin nicht sofort darauf gekommen, habe irgendwas von Strukturen gefaselt, war alles nicht so ganz richtig...)

Also der liefert uns ja so ein Bild hier. Aber das ist jetzt auch egal, warten sie bitte einen Moment draussen.

Nach der Prüfung

Als ich wieder reinkam, versuchte Prof. Seidl die Note zu verdecken, aber ich hab sie trotzdem erspäht, ich adler auge. Ich hätte zwar eigentlich alles gewusst, aber bin halt nicht immer direkt drauf gekommen. Explizit erwähnt hat er die

Sache mit der Teilmengenrelation und der Optics Ausgabe. Sie haben wohl überlegt mir noch eine 1.0 zu geben, aber habens dann halt doch nicht getan, die 1.3 fand ich aber auch angemessen.
Wie alle anderen Prüflinge muss auch ich sagen, dass die Atmospähre super locker war. Da braucht keiner Angst zu haben. Allerdings wurden in meiner Prüfung Sachen behandelt die in keinem anderen Protokoll zu lesen waren, also wie immer gilt dann auch hier, dass wer eine gute Note haben will, nicht allein nach Protokollen lernen sollte. Anhand der obigen Fragen kann man ja nicht die Themenschwerpunkte auslesen, deswegen mach ich das jetzt nochmal explizit:

- Datenbanken: Kein erkennbarer Schwerpunkt, hat viel in die Breite gefragt.
- Dateneploration: Hier lag der Fokus definitiv auf den Filtern und den Quadratischen Formen, wir haben fast über nichts anderes gesprochen.
- Data Mining: Im Grossen und Ganzen ging es nur um Clustering und AOI.

So. Das war meine letzte Prüfung und ich wünsch euch allen viel Erfolg bei den eurigen.