Prüfungsprotokoll zur Vertiefungsprüfung

Thema: Einführung in Datenbanken, Data Mining, Modelle der Datenexploration
Prüfer: Prof. Seidl
Termin: März 2006
Dauer: ca. 50 min
Note: 1,3

Prof. Seidl hat am Anfang nachgefragt mit welchem Thema ich am liebsten anfangen würde. Ich hatte EiDB genommen und schon ging es los. Die Einstiegsfragen in die jeweiligen Vorlesungen waren am Anfang recht allgemein, so dass man erstmal ein wenig Sicherheit bekam und die Nervosität abgebaut wurde. Die Prüfung war wirklich sehr angenehm, eher wie ein Dialog und nicht reines Fragen und Antworten. Prof. Seidl ist sehr nett und hilft bei kleinen Problemen auch mal nach. Kann man eine Frage nicht beantworten so wird diese mehr oder weniger so stehengelassen und zur nächsten Frage übergegangen bzw. etwas Hilfestellung gegeben. Es wurde nicht explizit noch mal „nachgebohrt“ so dass man überhaupt nicht mehr weiterwusste. Die Atmosphäre während der Prüfung war echt gut, ich kann Prof. Seidl als Prüfer nur empfehlen.

Ich gebe hier einen Fragenkatalog, die genauen Antworten zu den Fragen schlage man in den Skripten zu den Vorlesungen von Prof. Seidl nach. Als Vorbereitung für EiDB wurde das Buch „Datenbanksysteme“ von Kemper/Eickler abgesprochen.
Zur Korrektheit / Qualität der Antworten kann ich nichts sagen, auch sind mir mit Sicherheit einige Zwischenfragen wieder entfallen. Des weiteren geben ich nur die groben Ideen zur Antwort. Ich hoffe die Aufstellung hilft trotzdem weiter.

EIDB (ca. 15min)

Q: Was ist eine Transaktion?
A: Menge von Befehlen, die im Mehrbenutzersystem als Einheit fehlerfrei ausgeführt wird. Hier auch BOT; COMMIT etc. erwähnt.
Q: Was sind die ACID - Eigenschaften?
A: alle aufzählen...
Q: Was muss der Benutzer denn explizit machen?
A: Hier wusste ich nicht ganz bescheit, es ist wohl so, dass SQL alles schon implizit regelt, nur die Transaktion muss eben vom Benutzer explizit gestartet werden (bestätigt werden).
Q: Kommen wir zu Datenbanken, was gibt es denn da für Systeme?
A: OODB, Rel.DB, veraltet: satzorientiert.
Q: Was ist denn am OO Modell so besonders?
A: Methoden, globaler Objektidentifier, Objekte werden als ganzes betrachtet und nicht auf mehrere Relationen aufgespalten, Vererbung
Q: OK, konzentrieren wir uns auf Rel. Datenbanken. Was ist eine Relation?
A: Teilmenge vom Kreuzprodukt
Q: Was muss denn so alles ins DB-Schema?
A: Domänen bestimmen, Namen vergeben, Funktionalitäten bestimmen, Schlüssel bestimmen, hier auch irgendwie kurz Fremdschlüssel erwähnt und kurz erklärt.
Q: Wie kann man eine DB Anfrage stellen?
A: zum Bsp. mit SQL.
Q: Wie sieht so eine Anfrage aus? Schreiben Sie mal ein Beispiel auf
A: SELECT FROM WHERE mit einem kleinen Bsp.

Q: Was gibt es denn da noch?
A: RA, TK, DK
Q: Wo liegen denn die Gemeinsamkeiten von SQL und RA/TK?
A: Bindung der Variablen an Tupel und Gemeinsamkeiten der Operationen Projektion/Selektion zu SELECT/WHERE
Q: In der RA gibt es 6 Basisoperationen, schreiben Sie die mal auf.
A: ...hingeschrieben...
Q: Und jetzt machen Sie mal alle in SQL!
A: ...mach...
Q: OK, ich habe jetzt mal ein Schema R:{[A],[B]} und S:{[B],[C]}, was ist ein Join und schreiben Sie den mal hin für SQL/RA/TK/DK
A: Hingeschrieben, noch was über die Anzahl der Tupel erzählt

DM (ca. 20min)

Q: OK, kommen wir zu DM. Wie werden denn die Daten in Data Warehouses gespeichert?
A: Data Cubes und Sternschema erklärt
Q: Ja was ist denn ein Data Cube? Was ist denn die Idee dahinter?
A: Hier was von concept hierarchy erzählt, drill-down, roll-up, slice + dice
Q: Ich habe folgende Relation: Studenten:{[MatNr],[Name],[Fach],[Semester]}, stellen Sie mal eine Anfrage in SQL wie viel Studenten pro Fach gibt es?
A: SELECT Fach, count(Fach) FROM Studenten group by Fach
Q: Was ist group by, was macht es wo muss man da aufpassen?
A: Group by erklärt, gut für Aggregation (drill-down + roll-up), alles was im group by steht muss auch im SELECT stehen
Q: Machen Sie mal ein paar roll-up und drill-down Anfragen auf der Relation von oben
A: mehr oder weniger Attribute ins select schreiben
Q: und der apex-Cube und das lowest Level?
A: ...erklärt...
Q: OK, genug zum Data Warehousing, kommen wir zu einem wichtigen Thema im Data Mining, der Klassifikation. Was ist das?
A: Klassifikator/Modell bestimmen, aus Train + Test errechnen
Q: Was ist der Bayes-Klassifikator? Schreiben Sie auch mal die Formel hin!
A: Erklärt, auf naiv und nicht-naiv eingegangen, Unabhängigkeitsannahme, Formel hingeschrieben
Q: Wieso kann man im Bruch das P(O) weglassen?
A: weil man durch das „argmax C_i“ nur an den Klassen interessiert ist (die Begründung war etwas dünn. Prof. Seidl „erklärte“ es mir daraufhin genauer, leider sofort wieder vergessen ☺)
Q: Was muss man dabei schätzen?
A: d-dim Kov.Matrix, Mittelwert und die $P(C_i)$ bekommt man aus Train
Q: Was ist der k-NN Klassifikator? Was muss man hier schätzen und wie sehen die Gewichtungen aus?
A: Erklärt, benötigt nur Mittelwert, Wichtungen anhand Häufigkeit in Train, Häufigkeit im Decision Set und nach Distanz erklärt

DE (ca. 15min)

Q: hier war wohl die Überleitung zu DE, und ich wusste nicht ganz so genau was jetzt gefragt war: Funktioniert das Ganze nur mit Vektoren? Benötigt man immer einen Vektorraum?

A: (verwirrt) ööööhhhh, najaaaa, nein, geht auch ohne Vektoren ☺ meine Begründung war etwas schwammig und es wurde einfach mal so stehengelassen
Q: In DE gibt es ja ähnlich zum k-NN Klassifikator einige Nachbarschaftsanfragen, zählen Sie die mal auf und sagen Sie mal was dazu!
A: (k-)NN Suche, e-Range-Query, und grob erklärt (ohne Algorithmus aufzuschreiben).
Q: Hier weiss ich die Frage nicht mehr so genau, irgendwas zu dem k der k-NN Query und zu Inkrementellem Ranking
A: ganz kurz Ink. Ranking erläutert
Q: Was für Ähnlichkeitssuchen hatten wir denn?
A: Für Bilder, Formen, Sequenzen...
Q: Ah, Sequenzen! Schön! Was ist denn die Edit-Dist? Funktioniert die auch mit Graphen?
A: Edit-Dist erklärt (ohne Code), Basisoperationen (Löschen, Einfügen, Umbenennen) und Zusammenhänge zu Graphen erklärt.
Q: Was für andere Ähnlichkeitsmasse kennen Sie denn?
A: Lp-Normen aller Art, Quadratische Formen, Earth Movers Dist
Q: Schreiben Sie mal die Formel der QF auf, was ist die Matrix A?
A: hingeschrieben, Ähnlichkeitsmatrix erklärt
Q: und jetzt erklären Sie mal die EMD und vergleichen Sie das mal
A: EMD erklärt, bei QF werden Ähnlichkeiten und bei EMD Distanzen zwischen Bins genutzt.
Q: Bins? Histogramme? Was ist das?
A: Histogramme Erklärt, wie werden Pixel da einsortiert (nächster Repräsentant wird gewählt), Transformation in Feature Raum, Ähnlichkeiten im Featureraum sollen Ähnlichkeiten der Objekte darstellen.
Q: Ja also sagen wir mal die QF sei recht teuer in der Berechnung. Wie kann man das Beschleunigen?
A: Durch Index, durch Filter.
Q: Hm, Index hatte ich heute morgen schon soviel, lassen wir das Mal. Sagen Sie mal was zum Filter
A: lower-bounding, Selektivität, möglichst gute Laufzeit
Q: Geben Sie mal einen Filter für die QF an
A: z.B. eine Dimension weglassen
Q: Was geht denn da noch?
A: wusste hier nix rechtes, er wollte hören: QF stellt ja geometrisch eine Ellipsoid-Anfrage dar, diese kann durch die infinity-Norm (einem Quadrat) approx. werden
Q: Was wäre denn wenn der Filter 0 wäre? Ginge das? Ist das korrekt und komplett?
A: ist komplett und korrekt, schlechte Selektivität (gar keine), dadurch müssten alle Objekte im Refinementschritt bearbeitet werden.
Q: Was wäre denn wenn man die Anfrage der QF mit QF Filtern würde?
A: quasi das Gegenteil: Refinementschritt wäre überflüssig, max. Selektivität (nämlich genau auf die korrekten und erwünschten Ergebnisse). Macht beides keinen Sinn.
Q: OK, wir haben ja auch schon überzogen, das reicht mir dazu. Warten Sie bitte kurz draussen.

Fazit:

Danach hat mir Prof. Seidl die Note mitgeteilt und begründet. Ich wusste soweit auf alles eine korrekte und ausführliche Antwort, bis eben die Frage mit dem P(O) und den Vektoren. Des weiteren hatte ich mich in der GROUP BY SQL Anfrage ziemlich verzettelt und nur sehr stockend hingeschrieben, auf Nachfrage dann korrigiert. Deshalb eben 1,3 und nicht 1,0.

Damit war ich persönlich sehr zufrieden und die Benotung ist meiner Meinung nach sehr fair.