

DAIS Qualifying Examination  
*Fall 2006*

Department of Computer Science  
University of Illinois at Urbana-Champaign

Sept. 25, 2006  
Time Limit: 180 minutes

Please read the following instructions carefully before starting the test:

- The examination has 14 questions, including one basic concept question and 13 topic questions. You are required to answer the basic concept question and any **5** of the 13 topic questions of your choice. If you answer more than five topic questions, the committee will randomly select five to grade.
- The basic concept question has **13** subquestions. You are required to answer **10** of these 13 subquestions. If you answer more than 10 subquestions, the committee will randomly select 10 to grade.
- The 13 topic questions are distributed by areas as follows:
  - **Database Systems:** 4 questions
  - **Data Mining:** 3 questions
  - **Information Retrieval:** 3 questions
  - **Bioinformatics:** 3 questions

Each of these 13 questions generally has three parts. Make sure that you answer all the three parts of any topic question that you have chosen to answer.

- Use a separate booklet for each question that you answer. That is, you will use a total of six booklets (1 for the basic concept question and the rest for the 5 chosen topic questions).
- To ensure the fairness of grading, all the exams will be kept anonymous. Your answer sheets should thus only bear the assigned ID but *no names*.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Succinctness does count!*
- The questions have been designed to avoid any ambiguity. However, in case you are not sure how to interpret a question, please assume one interpretation, write down your assumption, and solve the problem accordingly.
- On average, you will have 30 minutes for each of the six questions that you have chosen to answer. So plan your time accordingly, and keep in mind that some questions/subquestions may demand more writing than others.

## Required Question (Basic Concepts): Problem 0

You are required to answer 10 out of the following 13 subquestions. Each subquestion is worth 1 point. Please focus on the major point and keep your answer concise; in general, an answer is expected to be no more than two sentences.

- (1) (*querying mechanism*) Relational algebra and SQL can both be used as a query language. How are they different for this purpose?
- (2) (*buffer management*) It is argued that buffer management should be done inside a DBMS (instead of the operating system layer), for both *performance* and *correctness*. Give one reason for each aspect.
- (3) (*normalization*) Schema normalization has no impact on query performance. Do you agree? Explain why or why not.
- (4) (*query optimization*) Query optimization can result in orders of magnitude difference in query performance. Give one example to illustrate that a good plan A can be 1000 times better than plan B. Then, change your parameters so that now B becomes better than A.

**Note: Please start with a new answer sheet.**

- (5) (*data cube and OLAP*) Briefly point out three major differences between the following two data cube computation methods: (1) BUC, and (2) MultiWay Array Cube computation.
- (6) (*sequential-pattern mining*) Name three sequential pattern mining algorithms, one from each of the following categories: (1) candidate generation-and-test, (2) vertical data format, and (3) pattern-growth approach.
- (7) (*clustering*) List the names of four algorithms that can perform effective clustering, each from the following categories: (1) partitioning method, (2) density-based method, (3) high-dimensional clustering, and (4) hierarchical clustering.

**Note: Please start with a new answer sheet.**

- (8) (*TF-IDF weighting*) Let  $D$  be a document in a text collection. Suppose we add a copy of  $D$  to the collection (i.e.,  $D$  is duplicated). How would this affect the IDF (Inverse Document Frequency) values of all the words in the collection? Why?
- (9) (*Language model smoothing*) Why is smoothing of a document language model necessary in the query likelihood retrieval method?
- (10) (*Average precision*) Suppose a retrieval system returns, in response to a query, a ranked list of 10 documents with the following relevance status:  $(R, N, N, R, N, N, N, N, N, N)$ , where  $R$  ( $N$ ) indicates that the document at the corresponding rank is relevant (non-relevant). For example, the top-ranked document is relevant. Assume that there are only two relevant documents for the query in the whole collection. What is the (non-interpolated) average precision of this system on this query?

**Note: Please start with a new answer sheet.**

- (11) (*alignment*) What does an affine gap penalty model mean in the context of pairwise sequence alignment? What is the time complexity of a dynamic programming algorithm for pairwise sequence alignment with an affine gap penalty? (Assume length of each sequence is  $n$ .)

- (12) (*Hidden Markov models*) What does the Viterbi algorithm for an HMM maximize ? What are two important applications of HMMs to biological sequence analysis ?
- (13) (*Evolution*) What is a molecular clock hypothesis or assumption ?

# Database Systems: Problem 1. Extensible Indexing

## Part 1

What is the purpose of GiST? Why does a DBMS need a mechanism like GiST? (*2 points*)

## Part 2

Suppose we want to index a data type *circle*: A circle instance with center  $c$  and radius  $r$  is represented by a tuple  $(c, r)$ .

- (a) What are interesting queries to ask about circles? (*1 point*)
- (b) Design a GiST to support such queries. You only need to concisely describe the design of the *key* predicate and the *consistent* method. (*2 points*)

## Part 3

GiST claims to be *general*. Do you agree with this claim of generality? Explain why or why not. (*5 points*)

*Note:* The answer really can go either way. What we want to see is your insightful arguments.

## Database Systems: Problem 2. Physical Database Design

### Part 1

In the "AutoAdmin What-if" paper, what is *hypothetical configuration simulation*? Define it in terms of its input, output, and purpose. (2 points)

### Part 2

The concept of hypothetical configuration analysis appears to be similar to *cost estimation* in standard cost-based query optimization.

- What are their similarities? Give one point. (1 point)
- What are their differences? Give two points. (2 points)

### Part 3

The system as reported in this paper accepts a *workload* as a *set* of (query, frequency) tuples, with the command of the form:

```
CREATE WORKLOAD workloadname AS (Q1, f1), (Q2, f2), ..., (Qn, fn)
```

This simple workload definition may not fully capture real world situations and requirements. We ask you to identify such limitations and propose solutions.

- Identify one limitation: i.e., give a practical situation, in which the above workload specification may not be general enough to capture, and explain what issues may arise. (2 points)
- Propose your strategy to solve the limitation: i.e., specify how the workload specification can be extended, and sketch how the simulation process can be generalized accordingly (or explain why such generalization might be impossible). (3 points)

## Database Systems: Problem 3. Focused Crawling

### Part 1

There have been different techniques proposed for focused crawling. Give two types of such techniques that are different from the “critic-apprentice” paper. (2 points)

**Part 2** The setting of *critic-apprentice* is unique, which enables online-training of the *apprentice*, through a pre-trained *critic*.

- Give *one* reason, to argue that this setting is advantageous for deploying a focused crawler. (1 point)
- Give *two* reasons, to argue that this setting is not necessarily effective. (2 points)

**Part 3** In the setting of this paper, the apprentice is trained with, for each pair of  $(u, v)$  link collected online, only the features *within*  $u$  and not the larger contexts (such as paths) leading to  $u$ . If such “context” features can be used, so that the apprentice can recognize a path leading to the target, the focused crawling may be more effective.

To incorporate such context-based training, how shall we do it? Sketch your solution. (5 points)

## Database Systems: Problem 4. Supporting New Data Types

### Part 1

Give two reasons why supporting new data types is important in a DBMS. (2 points)

### Part 2

The paper “Inclusion of New Types in Relational Data Base Systems” describes how B-tree can be used as an access method for a new data type. To see how this idea works, let’s assume a new data type *Weight*: Each instance  $w$  of *Weight* can be represented with a pair (value  $v$ , unit  $u$ ), e.g.,  $w_1 = (10, \textit{pound})$ ,  $w_2 = (20, \textit{kilogram})$ .

For indexing the *Weight* type with B-tree, describe what you need to do. (3 points)

### Part 3

Using the methodology as proposed in this paper, we can support indexing of new data types (such as *Weight* above) with existing methods such as B-tree, Hash, or R-tree. We ask you to identify one *significant* limitation of this approach:

- Identify one limitation. Explain why it is significant. (2 points)
- How would you address it? Sketch your thoughts for developing a solution. (3 points)

# Data Mining and Data Warehousing: Problem 1

## Part 1

Assume a base cuboid of 20 dimensions contains 3 base cells, as shown below,

- (1)  $(a_1, a_2, a_3, \dots, a_{20})$
- (2)  $(a_1, b_2, b_3, \dots, b_{20})$
- (3)  $(a_1, a_2, b_3, \dots, b_{20})$

where  $a_i \neq b_i$  (for any  $i$ ). The measure of the cube is *count*. Let the condition of iceberg cube be “*count*  $\geq 2$ ”.

- (a) How many nonempty cuboids will this full data cube contain? (1 point)
- (b) How many **nonempty** aggregated cells are there for the (full) data cube? (1 point)
- (c) How many **nonempty** aggregated cells are there for the iceberg cube? (1 point)

## Part 2

Data warehouse has been used in many applications. Suppose one wants to record the changes of the climate in a country so that one can check how the temperature changes in the last 10 years by region, by month, etc., and drill down to any particular region, month, etc.

- (a) Design such a data warehouse by drawing star schema with concise explanation. (1 point)
- (b) Show how to efficiently compute data cubes and perform OLAP operations in such a data warehouse. (2 points)

## Part 3

In some applications, one may need to perform OLAP analysis in high dimensional databases, such as over 50 dimensions.

- (a) Present one such application example. (1 point)
- (b) Based on the example you give, show how to efficient implement OLAP operations in such a high-dimensional space. (1 point)
- (c) Can you extend your implementation to handle data streams? If your answer is “yes”, how to do it? and if “no”, why not? (2 points)

## Data Mining and Data Warehousing: Problem 2

### Part 1 (2 points)

Someone says: “In a transaction database, two items that are strongly associated, such as  $a \rightarrow b[s, c]$ ,” with high  $s$  and  $c$  values, can still be strongly positively correlated, independent or strongly negatively correlated.”

If you agree with the statement, present one example for each case.

If you disagree with it, present your reasoning.

### Part 2

A frequent-pattern mining algorithm may generate a large set of frequent patterns, under a low support threshold.

- (a) Present two definitions of compressed frequent patterns and discuss their advantages and disadvantages. (2 points)
- (b) Outline an efficient algorithm that mines the set of compressed frequent patterns directly. (2 points)

### Part 3

One often needs to mine frequent itemsets in data streams. However, due to the limited main memory space, it is often only realistic to mine *approximate* frequent itemsets in data streams.

- (a) Outline one efficient algorithm that uses limited space to mine approximate frequent itemsets in long data streams, with a guaranteed error bound. (2 points)
- (b) Discuss how such an algorithm can be extended to find the *evolution* (with time) of approximate frequent itemsets in data streams. (2 points)

## Data Mining and Data Warehousing: Problem 3

### Part 1

- (a) There are many well-known classification methods. List six most influential classification algorithms. (*1 point*)
- (a) A typical decision-tree algorithm may not be scalable in really huge (say high-order gigabytes) datasets. Present one refined decision-tree induction method that is highly scalable. (*2 points*)

### Part 2

It is well-known that different models are suited for different datasets. Suppose a dataset can be organized as a multi-dimensional database. Proposed one efficient and effective method that can find the most suitable models for such a dataset in a multidimensional space. (*3 points*)

### Part 3

Classification can be used for outlier detection. Suppose one has collected data about the movements of ships in an ocean, their corresponding time, movement paths, ship information, weather information, etc. Moreover, some ships' movements have been marked as suspicious and some others as normal.

Outline an efficient classification method that may induce effective models for detecting suspicious ships in the ocean. (*4 points*)

# Information Retrieval: Problem 1. Language Models

## Part 1

- a Let  $Q = q_1 \dots q_m$  and  $D = d_1 \dots d_n$  be a query and a document, respectively. Suppose we use Dirichlet Prior smoothing method with parameter  $\mu$  and reference (collection) language model  $p(w|C)$ . Write down the formula for scoring  $D$  w.r.t.  $Q$  using the query likelihood retrieval method. (1 point)
- b Explain how the query likelihood retrieval formula above can implement retrieval heuristics similar to TF-IDF weighting and document length normalization. (2 points)

## Part 2

The Markov Random Field Model proposed in [Metzler & Croft SIGIR 05] is a general retrieval model based on the following joint probability distribution:

$$P_{\Lambda}(Q, D) = \frac{1}{Z_{\Lambda}} \prod_{c \in C(G)} \psi(c; \Lambda)$$

- a How do you rank documents using this Markov Random Field Model? Write down the general ranking formula suggested in [Metzler & Croft SIGIR 05]. (1 point)
- b Formally show that the query likelihood retrieval formula can be expressed as a special case of the general Markov Random Field Model. (1 point)
- c How did the paper [Metzler & Croft SIGIR 05] propose to train the Markov Random Field Model? (1 point)

## Part 3

Although the paper [Metzler & Croft SIGIR 05] showed that the proposed Markov Random Field Model is quite general, the generality largely comes from the fact that the potential functions are not restricted in any way.

- a In general, how should we design/define such potential functions? (2 points)
- b Can you suggest some other interesting ways to define a potential function that has not been discussed in [Metzler & Croft SIGIR 05]? (2 points)

## Information Retrieval: Problem 2. Email Spam

### Part 1

- a Give at least three different methods for detecting email spams. Use one or two sentences to explain the basic idea of each method. (*2 points*)

### Part 2

- a Sketch the basic idea of the MailRank algorithm proposed in [Chirita et al. CIKM 05]. (*1 point*)
- b What is a biasing set? How did the paper [Chirita et al. 05] automatically determine the size of the biasing set? (*1 point*)
- c MailRank is very similar to PageRank. Briefly explain how MailRank is similar to PageRank and how it is different. (*1 point*)

### Part 3

- a Identify at least one deficiency of MailRank in detecting spam messages. (*1 point*)
- b Can you propose an improvement of MailRank to fix the identified deficiency? (*2 point*)
- c Suppose we have some examples of spams. How can you extend MailRank to take advantage of such examples? (*2 points*)

## Information Retrieval: Problem 3. Learning for Retrieval

### Part 1

- a What is pseudo feedback? (*1 point*)
- b Why does pseudo feedback often help improve retrieval performance? (*1 point*)
- c Give one scenario when pseudo feedback will unlikely help. (*1 point*)

### Part 2

The following questions are asked in the context of the paper [Radlinski & Joachims KDD 05].

- a What is a query chain? How can a query chain be detected? (*1 point*)
- b The “Ranking SVM” approach essentially learns a scoring function that can compute a score for a document w.r.t. a query based on a linear combination of several kinds of features. Traditional retrieval methods (e.g., the vector space model) also score a document w.r.t. a query with a function involving a linear combination of feature values. What are the differences between these two scoring functions? (*2 points*)
- c How can a query chain help improve the basic ranking SVM approach? (*1 point*)

### Part 3

- a Point out at least one deficiency of the ranking SVM method as a method for implicit feedback. (*1 point*)
- b Can you suggest a way to use the Markov Random Field Model proposed in [Metzler & Croft 05] for implicit feedback (i.e., learning from query chains to improve retrieval accuracy)? (*2 points*)

## Bioinformatics: Problem 1. Motif finding

### Part 1

- (i) There are many well-known motif finding algorithms that are given a set of DNA sequences, and find “motifs” *ab initio* in the sequences. List three popular motif-finding algorithms. (1 point)
- (ii) Name a motif-finding algorithm that finds motifs from orthologous sequences. What is the motif model used in this algorithm? (1 point)

### Part 2

In the Bussemaker et al paper (“Moby Dick”), a promoter sequence is modeled as the concatenation of words  $\alpha$  drawn at random with frequency  $p_\alpha$  from a “dictionary”  $D$ . Given the entries  $\alpha$  of a dictionary, the optimal  $p_\alpha$  is found by maximizing  $Z = Z(S, \mathbf{p}_\alpha)$ , the probability of obtaining sequence  $S$  for a given vector of dictionary probabilities  $p_\alpha$ .

- (i) Write down an expression for  $Z$  for their model. This expression should consider all possible ways to partition the sequence  $S$  into dictionary words. (1 point)
- (ii) The average number of times that a word  $\alpha$  is used in partitionings of the sequence (the average being over the probability distribution over partitionings induced by their model) is given by

$$\langle N_\alpha \rangle = p_\alpha \frac{\partial}{\partial p_\alpha} \log Z = \frac{p_\alpha}{Z} \frac{\partial Z}{\partial p_\alpha}$$

Use the answer from (i) to provide an expression for  $\langle N_\alpha \rangle$ . (1 point)

- (iii) Using a Lagrange multiplier  $\lambda$  for the constraint  $\sum_\alpha p_\alpha = 1$ , computing maximum likelihood values for  $p_\alpha$  amounts to maximizing  $Z - \lambda(\sum_\alpha p_\alpha - 1)$  with respect to  $\lambda$  and each  $p_\alpha$ . Show that this is equivalent to the condition

$$p_\alpha = \frac{\langle N_\alpha \rangle}{\sum_\beta \langle N_\beta \rangle}$$

(2 points)

### Part 3

- (i) One of the limitations of the MobyDick algorithm is the overly simple motif model (words over alphabet {A, C, G, T}) used. A popular motif model today is the Position Weight Matrix or the PWM model. Given a dictionary of PWM motifs, how would you modify the expression for  $Z$  (above)? Suggest an algorithmic technique to learn the optimal dictionary, if the length of each word and number of words in the dictionary is given. (3 points)
- (ii) One aspect of the MobyDick algorithm is constructing the dictionary iteratively, by comparing the frequency with which two dictionary words occur as neighbors in a partitioning of the sequence, to the frequency predicted by the model, using a z-score statistic. What are your thoughts on applying this strategy in the context of the PWM model? (1 point)

## Bioinformatics: Problem 2. Sampling

### Part 1

Sequence alignment has traditionally been performed with variants of the original Needleman-Wunsch pairwise alignment algorithm, with the objective function being a simple linear function of the numbers of matches, mismatches, and gaps. List two criticisms of this approach, which may motivate a probabilistic approach to sequence alignment. (2 points)

### Part 2

The MCAlign algorithm of Keightley and Johnson finds the alignment  $a$  of sequences  $S$  by maximizing  $Pr(a|S)$ . Their search algorithm explores the space of possible alignments by taking the current alignment  $a_1$  and generating a proposal alignment  $a_2$  by applying a randomly chosen transformation to the alignment. The proposal alignment is accepted, and becomes the current alignment, with probability

$$Pr_{\text{accept}} = \min(1, [Pr(a_2|S)/Pr(a_1|S)]^\beta)$$

where  $\beta = 1/10$  in their implementation.

- (i) What is the effect of increasing  $\beta$  above 1 ? When  $\beta \rightarrow \infty$ , what does the algorithm reduce to ? (1 point)
- (ii) What is the effect of decreasing  $\beta$  below 1 ? What happens when  $\beta = 0$  ? (1 point)
- (iii) Does it make sense to use  $\beta < 0$  ? (1 point)
- (iv) Explain why the strategy with  $\beta = 1$  ensures that the sampling procedure if run sufficiently long will produce an alignment  $a$  with probability proportional to  $Pr(a|S)$ . Assume that the number of neighboring alignments from which the proposed alignment is chosen is the same regardless of the current alignment and that the choice (proposal) is made uniformly at random. Also assume that the Markov chain defined by the sampling strategy is aperiodic and irreducible. (2 points)

### Part 3

In light of your answers above, can you suggest a modification of the sampling strategy that does not use a fixed  $\beta$ ? What is the advantage of your strategy ? What practical issues (not related to the alignment problem *per se*) do you envision in implementing your strategy ? How would you solve such issues ? (3 points)

## Bioinformatics: Problem 3. Gene expression clustering

### Part 1

(i) List three different methods that have been proposed for clustering gene expression data. (1 point)

(ii) Mention one important motivation for a model-based clustering method. (1 point)

### Part 2

The Yeung et al. paper on model based clustering compares different clustering algorithms on synthetic data sets where the “true” clustering is known. The assessment of a particular clustering algorithm is done with a measure called “Rand Index” that captures the agreement between two clusterings (e.g., the true clustering and the output clustering).

(i) Define the Rand Index. (1 point)

(ii) Consider all possible clusterings of  $N$  data points into  $k$  equal sized clusters of size  $s$  each. (i.e.,  $N = ks$ ). Now, pick two such clusterings  $U$  and  $V$  at random (sampling with replacement). Let  $n_{ij}$  be the random variable representing the number of data points in the  $i^{th}$  cluster in  $U$  and in the  $j^{th}$  cluster in  $V$ . Derive an expression for

$$E\left(\sum_{i,j} \binom{n_{ij}}{2}\right)$$

(2 points)

(iii) As mentioned in the Yeung et al. paper, the general form of an index with a constant expected value is  $(\text{index} - \text{expectedindex})/(\text{maximumindex} - \text{expectedindex})$ . Argue that this is true, i.e., this expression has constant expectation. (1 point)

### Part 3

(i) The Rand Index provides a measure for the similarity between two clusterings. Suppose you wanted to measure the similarity among more than two clusterings. Suggest a measure for this purpose. (2 points)

(ii) Suppose you were given  $d$  different clusterings of the same collection of data points. Each clustering prescribes relationships among the data points (co-clustered, or not co-clustered). Suppose you wanted to find out the data points that have the same relationships prescribed by all the clusterings. How would you go about this task? (Formulate the problem and present your thoughts on solving the problem.) (2 points)