1 An Overview (week 1)

Overview DIS

2 Semistructured Data Management

2.1 Horizontal Fragmentation

2.1.1 Relational Databases

1. At which phase of the database lifecycle is fragmentation performed ?

   √ **At database design time**

   ◯ During distributed query processing

   ◯ During updates to a distributed database

2. The reconstruction property expresses that

   ◯ In case of a node failure the data can be recovered from a fragment from another node

   √ **The original data can be fully recovered from the fragments**

   ◯ Every data value of the original data can be found in at least one fragment

2.1.2 Primary Horizontal Fragmentation (week 2)

1. Example: application A1 accesses

   1. Fragment F1: with frequency 3
   2. Fragment F2: with frequency 1

   A1 accesses the whole relation with frequency

   √ $^{13}/_{7}$

   ◯ $^{4}/_{7}$

   ◯ $^{14}/_{7}$

2. Consider the access frequencies below: How many horizontal fragments would a minimal and complete fragmentation have?

   - √ **3**
   - ○ 4
   - ○ 6

3. Which of the following sets of simple predicates is complete?

   - ○ Location = "Munich", Budget > 200000
   - ○ Location = "Munich", Location = "Bangalore"
   - ○ Location = "Paris", Budget $\leq$ 200000
   - √ **None of those**

4. Which is true for MinFrag algorithm?
   - ○ The output is independent of the order of the input
   - ○ It produces a monotonically increasing set of predicates
   - √ **It always terminates**
   - ○ All of the above statements are true

5. When deriving a horizontal fragmentation for relation S from a horizontally fragmented relation R
   - √ **Some primary key attribute in R must be a foreign key in S**
   - ○ Some primary key attribute in S must be a foreign key in R
   - ○ Both are required

2.2 Graph Databases (week 3)

2.2.1 Semi-structured Data

1. Semi-structured data

   ◯ Is always schema-less

   √ **Always embeds schema information into the data**

   ◯ Must always be hierarchically structured

   ◯ Can never be indexed

2. Why is XML a document model?

   ◯ It supports application-specific markup

   ◯ It supports domain-specific schemas

   √ **It has a serialized representation**

   ◯ It uses HTML tags

2.2.2 Graph Data Model

1. In a graph database

   ◯ There is a unique root node

   √ **Each node has a unique identifier**

   ◯ Data values in leaf nodes are unique

   ◯ The labels of edges leaving a node are different

   ◯ There is a unique path from the root to each leaf

2. The simulation relationship is a relation

   √ **Among nodes in the data and schema graph**

   ◯ Among edges in the data and schema graph

   ◯ Among sets of nodes in the data and schema graph

   ◯ Among sets of edges in the data and schema graph

3. Which is true?

   ◯ For each labelled edge in S a corresponding edge in D can be identified

   ◯ For each root node in S a corresponding root node D can be identified

   √ **For each leaf node in D a corresponding typed node in S can be identified**

   ◯ For each node in S a unique path reaching it from a root node can be identified

4. If there exists a uniquely defined simulation relationship among a graph database D and a schema graph S

   ◯ The data and schema graph are simulation equivalent

- √ **Ambiguous classification cannot occur**
- ◯ Multiple classification cannot occur

5. If schema graph S_1 subsumes S_2
   - ◯ Every graph database corresponding to S_1 corresponds also to S_2
   - √ S_2 **simulates** S_1
   - ◯ S_1 has fewer nodes than S_2

2.2.3 Schema Extraction

1. Which is wrong? In a dataguide
   - ◯ Every path in the data graph occurs only once
   - √ **Every node in the data graph occurs only in one data guide node**
   - ◯ Every data guide node has a unique set of nodes
   - ◯ A leaf node in the data graph corresponds always to a leaf node in the data guide
2. In a non-deterministic schema graph
   - √ **Every node of the data graph occurs exactly once**
   - ◯ Every path of the data graph occurs at most once
   - ◯ Every label of an outgoing edge of a node in the schema graph is unique

3 Information Retrieval and Data Mining

3.1 Information Retrieval (week 4)

3.1.1 Information Retrieval

1. A retrieval model attempts to model
   - ◯ The interface by which a user is accessing information
   - √ **The importance a user gives to a piece of information**
   - ◯ The formal correctness of a query formulation by user
   - ◯ All of the above
2. If the top 100 documents contain 50 relevant documents
   - ◯ The precision of the system at 50 is 0.5
   - √ **The precision of the system at 100 is 0.5**
   - ◯ The recall of the system is 0.5
   - ◯ None of the above
3. If retrieval system A has a higher precision than system B
   - ◯ The top k documents of A will have higher similarity values than the top k documents of B

- √ **The top k documents of A will contain more relevant documents than the top k documents of B**
- ○ A will recall more documents above a given similarity threshold than B
- ○ Relevant documents in A will have higher similarity values than in B

3.1.2 Text-based Information Retrieval

1. Full-text retrieval means that
   - ○ The document text is grammatically deeply analyzed for indexing
   - ○ The complete vocabulary of a language is used to extract index terms
   - √ **All words of a text are considered as potential index terms**
   - ○ All grammatical variations of a word are indexed
2. The term-document matrix indicates
   - √ **How many relevant terms a document contains**
   - ○ How relevant a term is for a given document
   - √ **How often a relevant term occurs in a document collection**
   - √ **Which relevant terms are occurring in a document collection**
3. Let the query be represented by the following vectors: (1, 0, -1) (0, -1, 1); the document by the vector (1, 0, 1)
   - ○ Matches the query because it matches the first query vector
   - √ **Matches the query because it matches the second query vector**
   - ○ Does not match the query because it does not match the first query vector
   - ○ Does not match the query because it does not match the second query vector
4. Which is right? The term frequency is normalized
   - √ **By the maximal frequency of a term in the document**
   - ○ By the maximal frequency of a term in the document collection
   - ○ By the maximal frequency of a term in the vocabulary
   - ○ By the maximal term frequency of any document in the collection
5. The inverse document frequency of a term can increase
   - ○ By adding the term to a document that contains the term
   - √ **By adding a document to a document collection that does not contain the term**
   - ○ By removing a document from the document collection that does not contain the term
   - ○ By adding a document to a document collection that contains the term

3.2 Advanced Retrieval Models (week 5)

3.2.1 Latent Semantic Indexing

1. In vector space retrieval each row of the matrix M^t corresponds to

   √ **A document**

   ◯ A concept

   ◯ A query

   ◯ A query result

2. Applying SVD to a term-document matrix $\mathbf{M}$. Each concept is represented

   ◯ As a singular value

   √ **As a linear combination of terms of the vocabulary**

   ◯ As a linear combination of documents in the document collection

   ◯ As a least square approximation of the matrix $\mathbf{M}$

3. The number of term vectors in the SVD for LSI

   ◯ Is smaller than the number of rows in the matrix $\mathbf{M}$

   √ **Is the same as the number of rows in the matrix $\mathbf{M}$**

   ◯ Is larger than the number of rows in the matrix $\mathbf{M}$

4. A query transformed into the concept space for LSI has

   √ s **components (number of singular values)**

   ◯ m components (size of vocabulary)

   ◯ n components (number of documents)

3.2.2 User Relevance Feedback

1. Can documents which do not contain any keywords of the original query receive a positive similarity coefficient after relevance feedback ?

   ◯ No

   ◯ Yes, independent of the values β and γ

   √ **Yes, but only if** $\beta > 0$

   ◯ Yes, but only if $\gamma > 0$

3.2.3 Link-based Ranking

2. A positive random jump value for exactly one node implies that

   √ **a random walker can leave the node even without outgoing edges**

   ◯ a random walker can reach the node multiple times even without outgoing edges

   √ **a random walker can reach the node even without incoming edges**

◯ none of the above

3. Given the graph below and an initial hub vector of $(1, 1, 1)$. The hub-authority ranking will result in the following

◯ authority vector $(0, 0, 1)$; hub vector $(1, 1, 0)$

◯ authority vector $(0, 0, 2)$; hub vector $(2, 2, 0)$

√ **authority vector** $(0, 0, 1)$ **; hub vector** $(\frac{1}{2}, \frac{1}{2}, 0)$

◯ authority vector $(0, 0, 2)$; hub vector $(1, 1, 0)$

3.2.4 Inverted Files (week 6)

1. A posting indicates

◯ The frequency of a term in the vocabulary

◯ The frequency of a term in a document

√ **The occurrence of a term in a document**

◯ The list of terms occurring in a document

2. When indexing a document collection using an inverted file, the main space requirement is implied by

◯ The access structure

◯ The vocabulary

◯ The index file

√ **The postings file**

3. Using a trie in index construction

◯ Helps to quickly find words that have been seen before

◯ Helps to quickly decide whether a word has not seen before

◯ Helps to maintain the lexicographic order of words seen in the documents

√ **All of the above**

4. Maintaining the order of document identifiers when partitioning the document collection is important

√ **In the index merging approach for single node machines**

- ◯ In the map-reduce approach for parallel clusters
- ◯ In both
- ◯ In neither of the two

3.2.5 Distributed Retrieval

1. When applying Fagin's algorithm for a query with three different terms for finding the k top documents, the algorithm will scan
   - ◯ 2 different lists
   - $\surd$ **3 different lists**
   - ◯ k different lists
   - ◯ it depends how many rounds are taken
2. Once k documents have been identified that occur in all of the lists
   - ◯ These are the top-k documents
   - $\surd$ **The top-k documents are among the documents seen so far**
   - ◯ The search has to continue in round-robin till the top-k documents are identified
   - ◯ Other documents have to be searched to complete the top-k list

Credits

Quiz questions were taken from the lecture notes of Prof. Karl Aberer.