NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Science

Examination paper for TDT4300 Data warehousing and data mining

Academic contact during examination: Kjetil Nørvåg/Heri Ramampiaro

Phone: 73596755/73591459

Examination date: May 28th 2016

Examination time (from-to): 09.00-13.00

Permitted examination support material: D: No tools allowed except approved simple calculator.

Other information:

Language: English

Number of pages (front page excluded): 4

Number of pages enclosed: 0

Checked by:

Date Signature

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig X 2-sidig □

sort/hvit X farger □

Problem 1 – Big Data – 5 %

a) When explaining Big Data, one often talks about the three (or more) V's. Describe the three most important of those.

Problem 2 – Hadoop – 20 % (all having same weight)

a) What were important goals for the Hadoop File system (HDFS), and what is HDFS *not* suitable for?

b) Describe the architecture of HDFS (feel free to use figure as help). Describe how files are stored, and node types.

c) Explain what happens when a client reads a file that is stored in HDFS (incl.Interaction between nodes).

d) Explain execution of an application on YARN, incl. description of node types and processes. Feel free to explain with the help of a figure.

Problem 3 – MapReduce and Spark– 10 % (all having same weight)

Assume a file PersonInfo.txt that contains information about name, age and salary, e.g., on format like this:

```
Kari  45  450000
Ola   30  200000
Kate 30  500000
Pål    45  550000
```

We want to find average salary for each age, e.g., a result like this (not needed to be sorted):

```
45  500000
30  350000
```

a) Show with pseudo code for *mapper* og *reducer* how this can be done in MapReduce. Assume for simplicity that *value* of map is a record with the fields *age* and *salary*, e.g., use the following as a starting point:

```
public void map(key(name), value(age,salary))
public void reduce(key, Iterable values)
```

b) We now want to find maximum salary for each age using Spark. Assume we already have read the file into an RDD of pairs (key,value)=(age,salary), e.g., RDD[(int,int)]. Show which transformation(s) have to be performed in order to get a resulting RDD where (key,value)=(age,maxSalary).
Hint: important transformations and actions in Spark includes `map, flatMap, filter, distinct, union, collect, count, countByValue, reduce, reduceByKey, groupByKey, values, sortByKey,` og `countByKey.`

Problem 4 – NoSQL – 15 % (10 % on a, 5 % on b)

a) We have a student database with the following tables in a relation schema:

```
Student(SNo, Name, Email)
Exam(ENo, CourseName, EDay, EMonth, EYear, Duration)
ExamResult(ExamNo, StudentNo, Grade)
```

Here it is interesting to find which exams a particular student have taken (grade transcripts). It is also interesting to find which students have taken a particular exam (grading list). How would you like to store this schema in HBase when you use the design principle DDI (*denormalization, duplication, intelligent keys*) and need to support the two queries given above?

b) Describe how «sharding»/partitioning are performed in both MongoDB and Apache HBase (also called «auto-sharding»).

Problem 5 – Streaming Data – 30 % (all having same weight)

Let us assume you want to analyze how many times a topic about the American election and campaigning is mentioned in messages in social media (e.g., Twitter).

a) Discuss the characteristics and / or challenges with streaming data. List two other examples where you have to handle streaming data (in addition to Twitter and social media in general).

b) We distinguish between two types of queries in connection to streaming data. Explain what these types are. Use examples to support your answer.

c) Imagine that you want to calculate the fraction of messages that are related to the topic "election" and "campaigning" in a given limited time period. For this purpose we choose to use the sliding window principle. Assume that our sliding window has a size of 1000 twitter messages (i.e., Tweets). Show how you can compute this fraction.

d) Can the problem above be seen as a variant of "bit counting"? Justify your answer.

e) Use the bloom filter principle to fill out the table below:

Streaming element	**Hash function - h_1**	**Hash function - h_2**	**Filter Content**
			00000000000
39 = 10 0111			
214 = 1101 0110			
353 = 01 0110 0001			

Hint: Use *h(x)=y* mod 11 as hash function, where *y* is generated from odd-numbered bits in *x* or even-numbered bits in x, respectively.

f) Suppose we want to analyze the last 11 messages that have arrived. On Twitter, the same user often resends many their messages to emphasize his/her view. Other users retweet messages to reach more people. Explain how we can use bloom filters to discard such messages. Make any assumptions you find necessary.

g) Now assume that when the 11 messages have arrived, we have a stream of data that looks like this: 10100101010. Could we have seen the message which can be represented by *y* = 1111011 before? Justify your answer.

Problem 6 – Recommender Systems – 20 % (6% on a.i, 4% on a.ii and 10% on b)

You are employed by a new company specializing in movie streaming. One of your tasks is to develop good recommendation algorithms and methods.

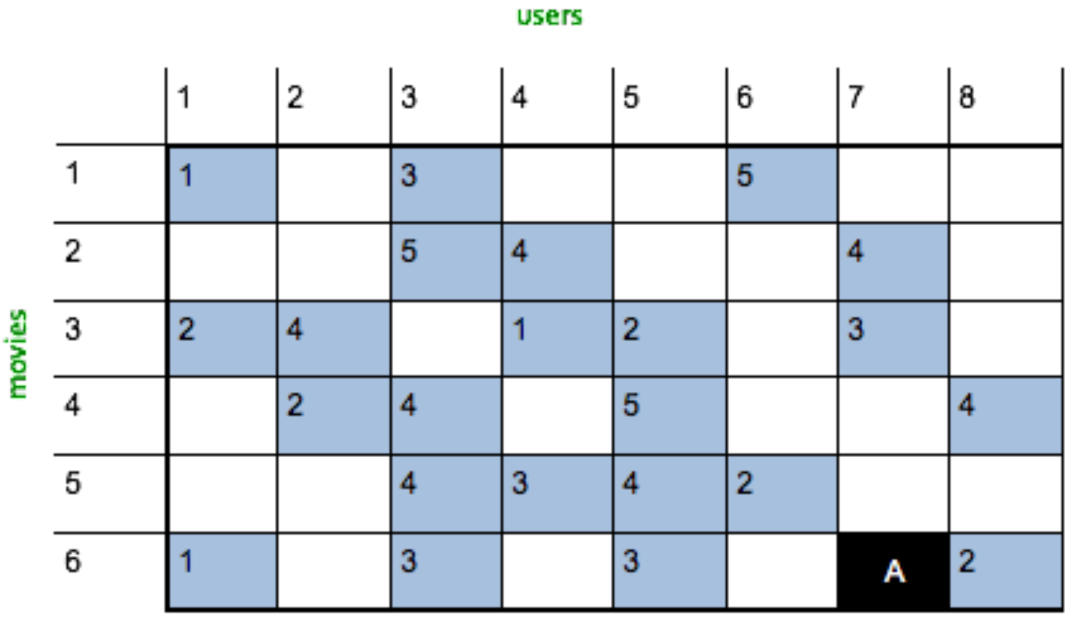
a) Part of the method you suggest is to give the user the ability to "rate" movies and then use this to find outdecide what movies your system will recommend later. Assume that your users have rated the following 10 movies with 3 or more stars:

`Jurasic Park (Fantasi/SciFi), Harry Potter (Fantasi/Adventure), ET (SciFi), Lord of the Rings (Fantasi/Adventure), Alien (SciFi), Terminator (SciFi), 101 Dalmatians (Adventure/Family), Titanic (Romantic), Sleepless in Seattle (Romantic) og Mr. Bean (Comedy).`

i. Explain how you would proceed to recommend the next movie to this user. Make any assumptions you find necessary.

ii. Would you use content-based recommender systems method collaborative filtering? Justify your answer.

b) Assume the following user-rating table:


users
movies

	1	2	3	4	5	6	7	8
1	1		3			5		
2			5	4			4	
3	2	4		1	2		3	
4		2	4		5			4
5			4	3	4	2		
6	1		3		3		A	2

Use the item-item collaborative filtering method to predict/estimate the rating for user no. 7' s rating of movie no. 6, i.e., what would the rating value A be? You must show how you compute the value.

For this task you will need the following formulas:

Pearson Correlation similarity – similarity between vector x and y:

$$\boldsymbol{sim}(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{s \in S_{xy}} (\boldsymbol{r_{xs}} - \overline{\boldsymbol{r_x}})(\boldsymbol{r_{ys}} - \overline{\boldsymbol{r_y}})}{\sqrt{\sum_{s \in S_{xy}} (\boldsymbol{r_{xs}} - \overline{\boldsymbol{r_x}})^2} \sqrt{\sum_{s \in S_{xy}} (\boldsymbol{r_{ys}} - \overline{\boldsymbol{r_y}})^2}}$$

where r_{xs} is the rating on movie *x* by a user s and $\overline{\boldsymbol{r}}_x$ is the average of all ratings on movie *x*.

Weighted average for ratings of a user:

$$\boldsymbol{r_{ix}} = \frac{\sum_{j \in N(i;x)} \boldsymbol{s_{ij}} \cdot \boldsymbol{r_{jx}}}{\sum \boldsymbol{s_{ij}}}$$

r_{ix} is here the rating of user *x* on movie *i,* while s_{ij} is the similarity between the ratings of movie *i* and *j*.