Towards inferring ratings from user behavior in BitTorrent communities

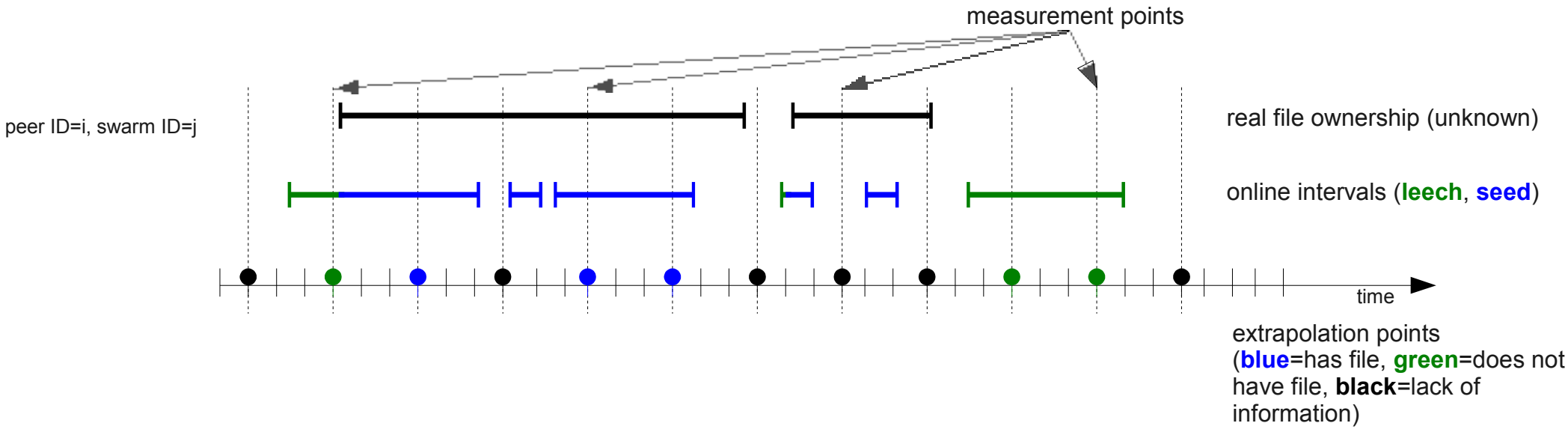
Róbert Ormándi, István Hegedűs and Márk Jelasity

The user behavior inferring problem

- Our two main observations:
  - Knowing the user preferences is an ***important*** issue
    - Numerous application like business model building, recommender systems
  - We ***cannot get*** the user preferences ***directly***, since
    - It does not exist datasource which directly contains this information
    - We cannot ask the users directly since they are unwilling to provide the sufficient information
- From these two observations naturally comes that we have to ***apply inference*** to get the necessary information
- In our case the base of the inference is BitTorrent tracker trace and the user behavior modeling is the inferring of the users like/dislike values according to the files

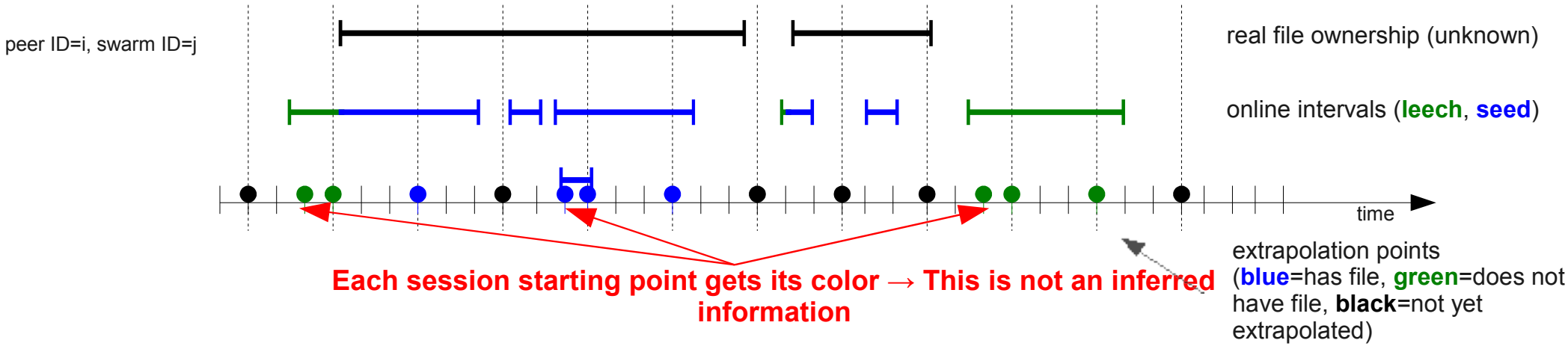
BitTorrent tracker trace

- FileList.org private community collected by TU Delft
- 93 days, 91,745 peers, 3,068 swarms
- The database contains the following information:
  - timestamp, peer ID, swarm ID, completion of the file in %
- On the figure we can see the different states of the file ownership of a user in three different aspects



Preprocessing the raw data – Coloring the beginnings

- Sequence of online and offline sessions for each user in each swarm
- The first question is how we can fill the inner parts of the online sessions
- Each measurement point according to an online session contains information from the beginning of the current session → we can add colors for the first timestamp easily

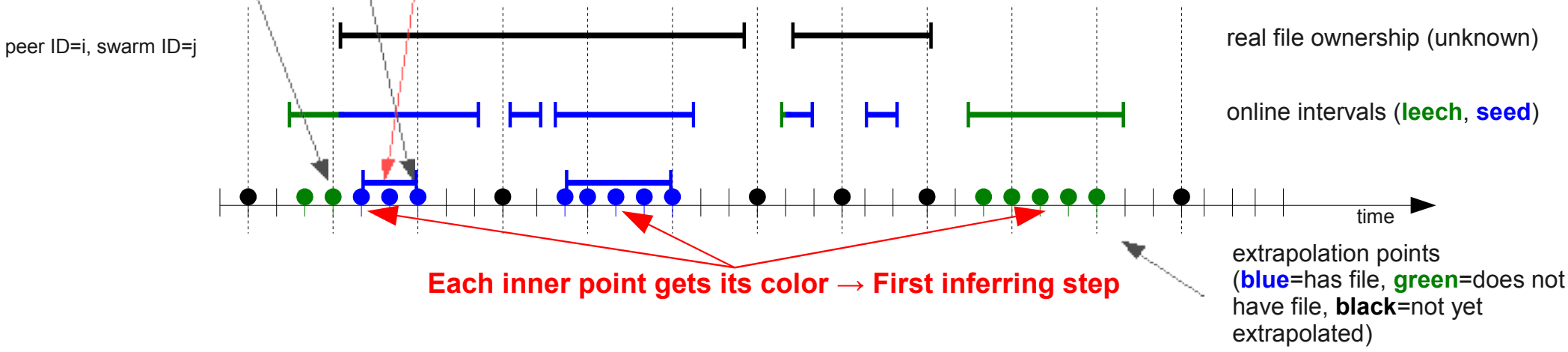


Preprocessing the raw data – Filling the inner parts

- We extrapolated the inner parts of each online session applying the following heuristic rules:

Online sessions		
front	end	fill
0	0	0
0	1	1
1	0	0
1	1	1

- Each remaining point considered as “offline session point”
- We do not know anything about those too short online sessions → we ignore them

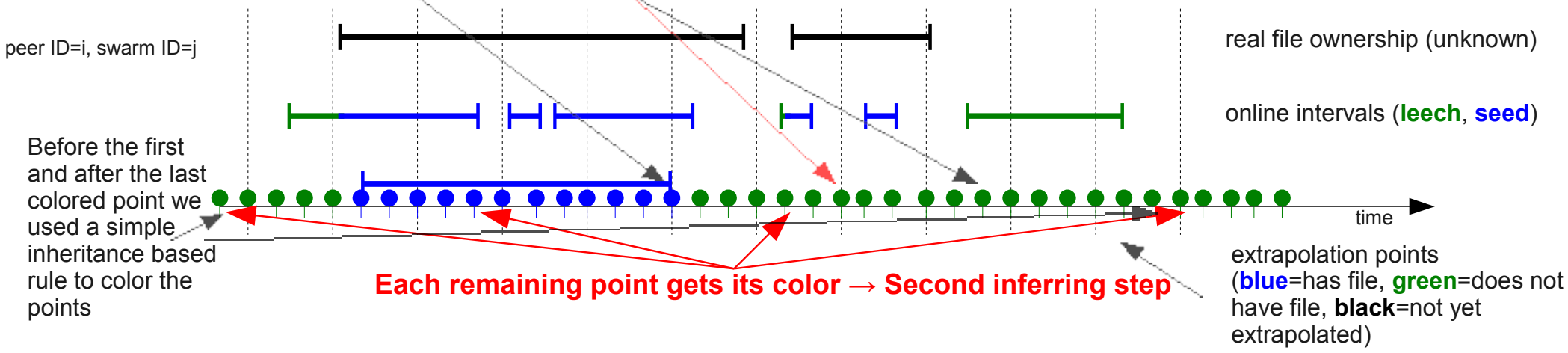


Preprocessing the raw data – Filling the outer parts

- We colored the remaining parts as well applying the following heuristic rules:

Offline sessions		
front	end	fill
0	0	0
0	1	0
1	0	0
1	1	1

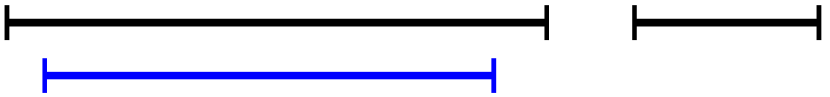
- We performed this preprocessing steps on each peer in each swarm → For each user in case of each file we can say a predicted file ownership



Inferred file ownerships

- If we create intervals from the extrapolated points, we can get the below showed approximation of the real file ownership
- The sample showed states of only one user, file pair

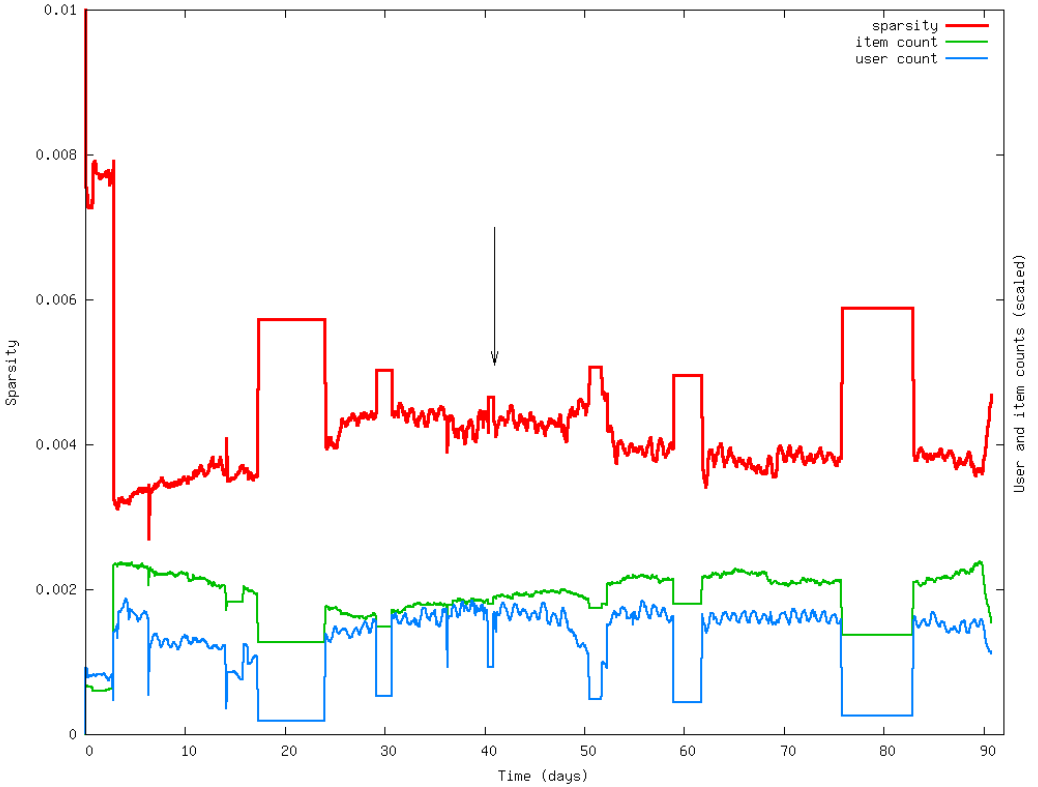
peer ID=i, swarm ID=j



real file ownership(unknown)
extrapolated intervals

Sparsity of the Dataset

- Completing the preprocessing steps we got a user-item matrix for every point in time ('has file' (1) is the known rating, 'does not have file' (0) is considered as unknown)
- Sparsity of this matrix:
  - The ratio of known and all ratings
  - Important factor in the difficulty of producing recommender model



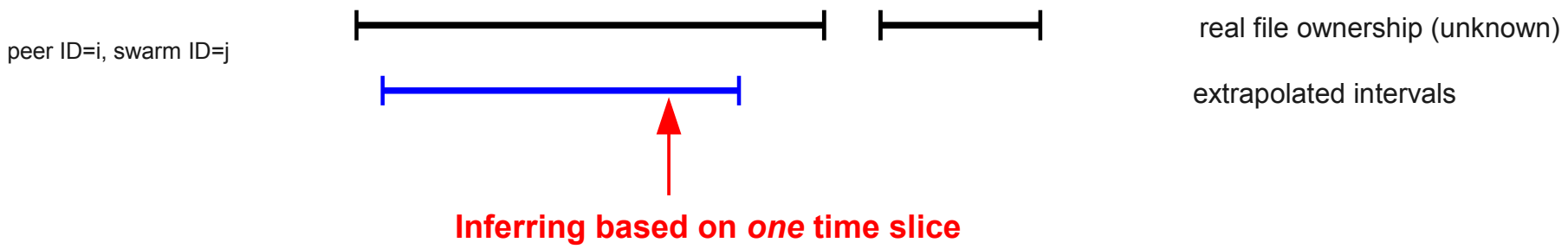
Inferring the user behavior from the preprocessed data

- After we preprocessed the data, we have to apply inferring behavior
- In our case we have to convert a series of file ownership into a single ***like/dislike*** value
- We propose two approach:
  - The naïve solution
  - The time shift based approach
- Let's look them closer

Inferring ratings – the naïve solution

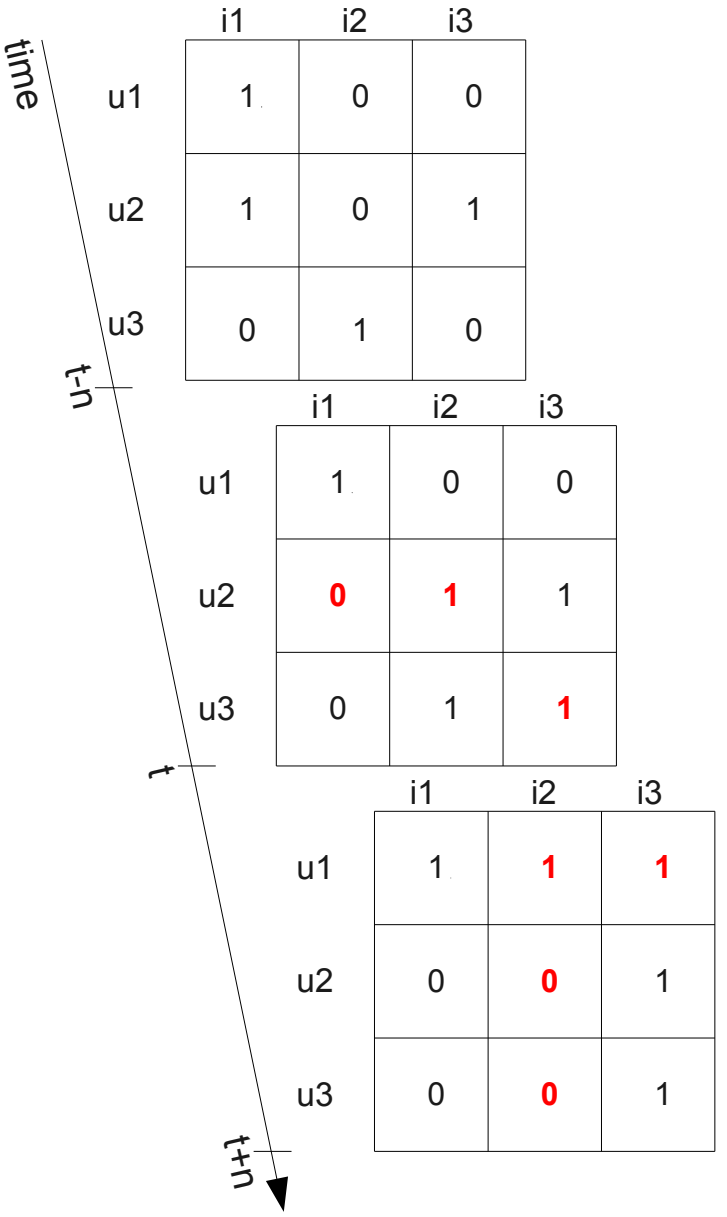
- Pick one point int time and slice series of file ownerships
- The naïve approach is:
  - 'has file' → positive rating (like)
  - 'does not have file' → lack of rating

  so we simply ignore the negative ratings, since we have no basis information



Inferring ratings- the time shifting approach

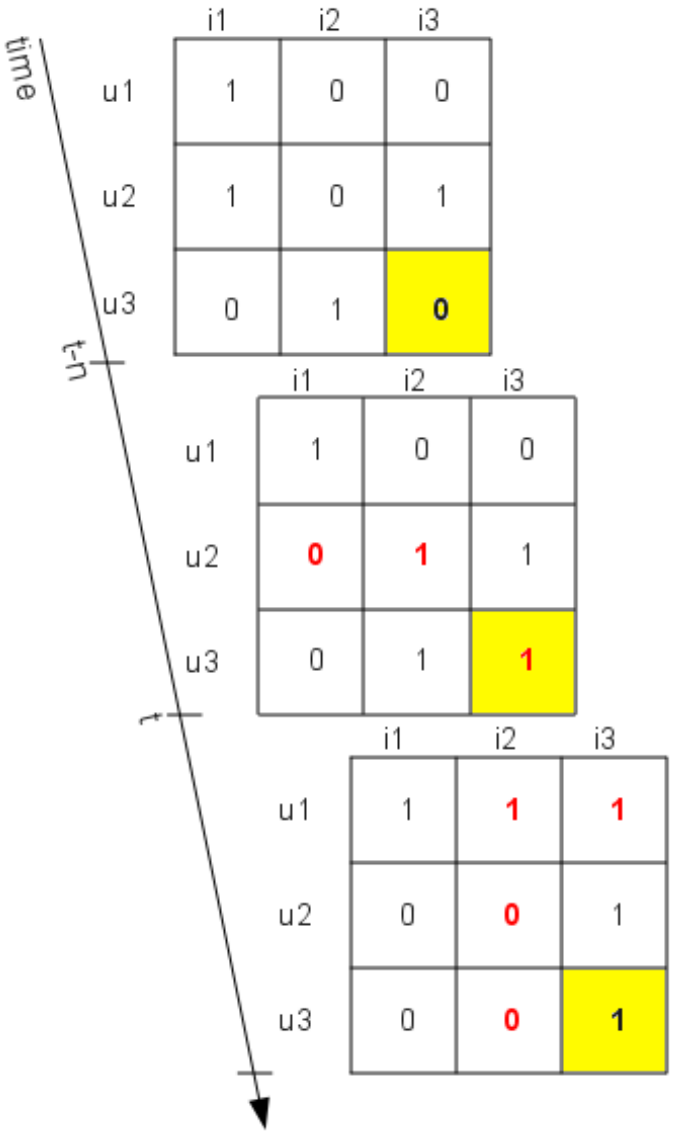
- Pick one common user-item matrix and two corresponding matrices as well
  - the first corresponding matrix selected before the timestamp of the chosen user-item matrix
  - the second one selected after the timestamp of the given user-item matrix
- Inferring ratings using the triplets



Inferring ratings- the time shifting approach

- Rating conversion rules:

Dataset labeling			
before	actual	after	inference
0	0	0	unspecified
0	0	1	unspecified
0	1	0	**0 (dislike)**
0	1	1	**1 (like)**
1	0	0	**0 (dislike)**
1	0	1	unspecified
1	1	0	unspecified
1	1	1	**1 (like)**

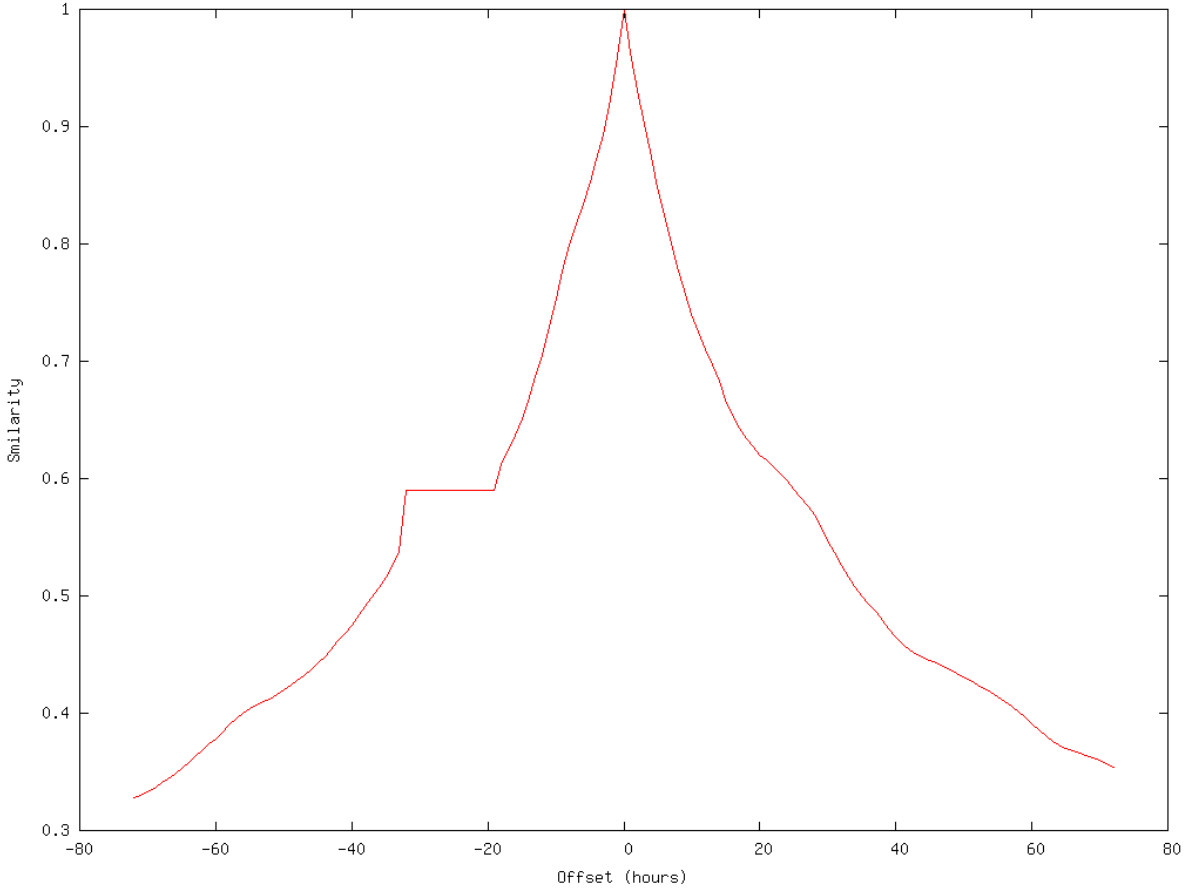


First the user u3 does not have the file i3, after n hours he downloaded the file completely (so he has the file) and after n hours he still has the file i3.

Since we infer that the user u3 likes the file i3.

Inferring ratings- the time shifting approach

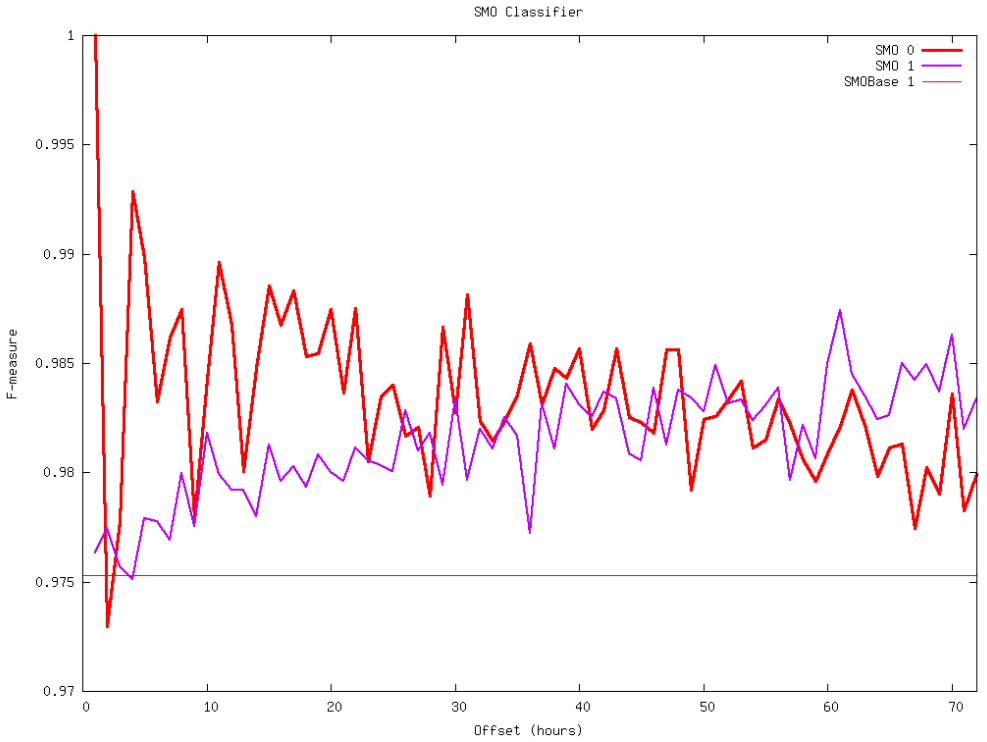
- How should we set the offset?
- The next figure shows the similarities between the base and the time shifted matrices:

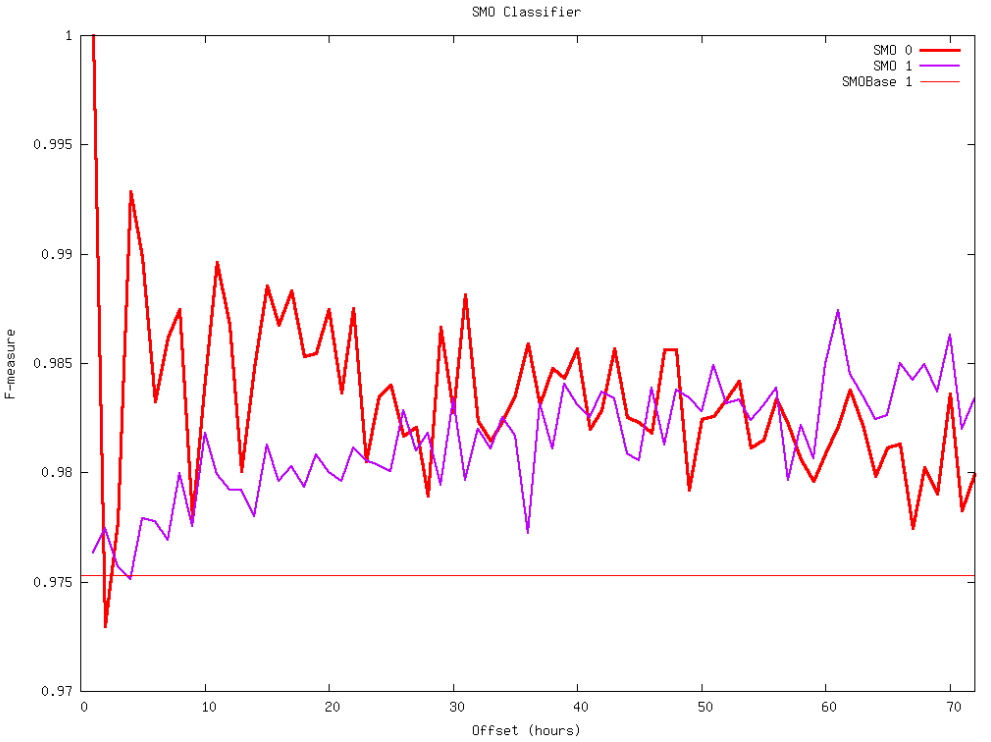


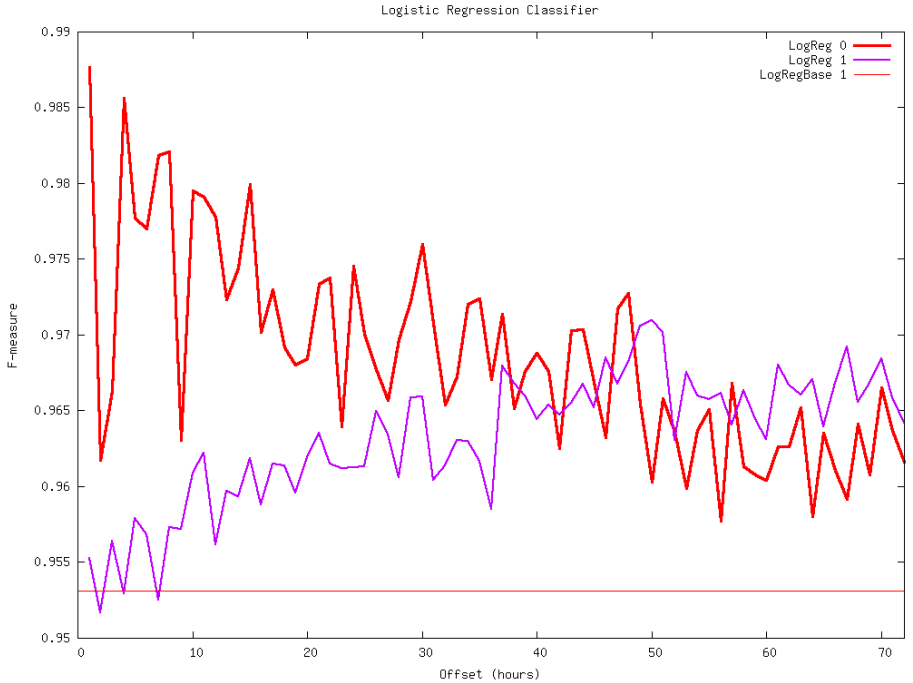
Evaluation

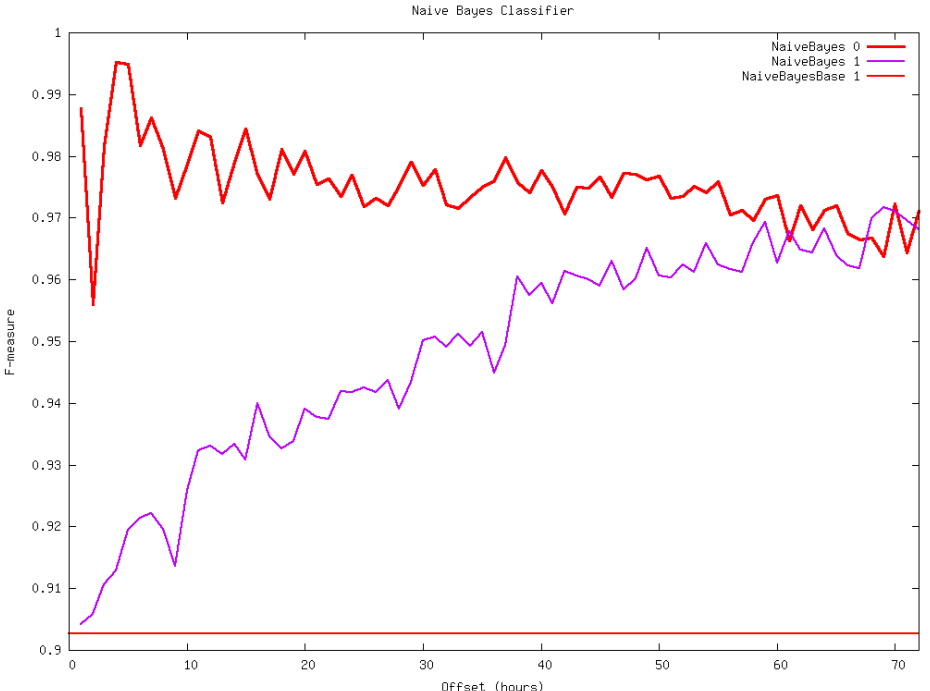
- Without any “ground truth” indirect approach
- We examined the learnability of the different labels:
  - 'Base 1': baseline inferring approach positive ratings vs. randomly selected unknown values
  - '0': time shifting based inferring approach negative ratings vs. randomly selected unknown values
  - '1': time shifting based inferring approach positive ratings vs. randomly selected unknown values
- Evaluation metric: F-measure
- Applied learning algorithms:
  - SVM
  - J48
  - Logistic Regression
  - Naïve Bayes

Evaluation


SMO Classifier
SMO 0
SMO 1
SMOBase 1
F-measure
Offset (hours)


SMO Classifier
SMO 0
SMO 1
SMOBase 1
F-measure
Offset (hours)

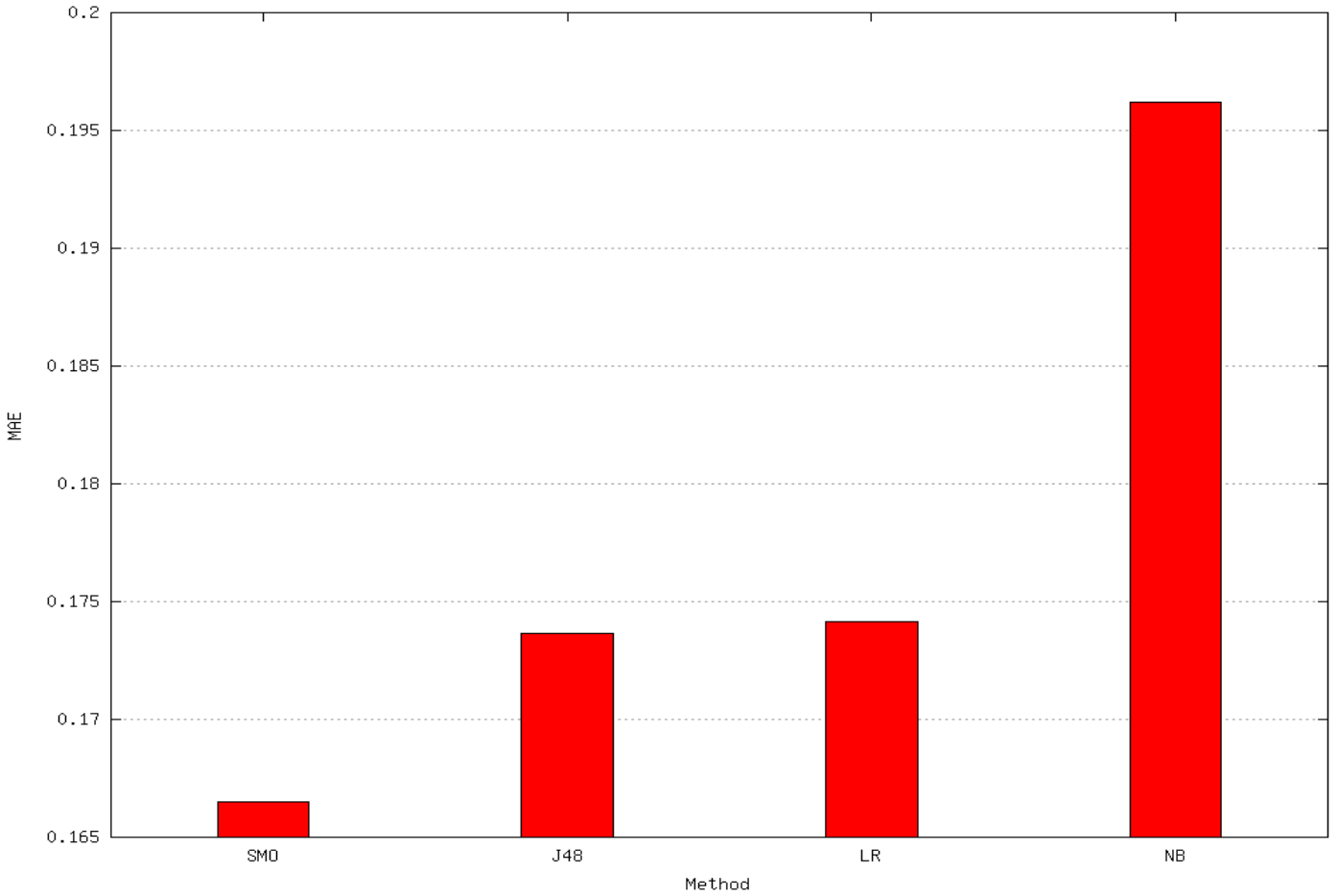

Logistic Regression Classifier
LogReg 0
LogReg 1
LogRegBase 1
F-measure
Offset (hours)


Naive Bayes Classifier
NaiveBayes 0
NaiveBayes 1
NaiveBayesBase 1
F-measure
Offset (hours)

The published dataset

- A database which was generated
  - based on the a randomly selected user-item matrix (showed the certain time point in the previous slides)
  - applying the proposed time shift based approach using offset +/- 60 hours

  is available for research purposes
- A train/test split is also available



Conclusion

- We proposed an inferring approach
- It is based on heuristics so it is far from precise
- But we could demonstrated that using a wide range of machine learning algorithms the inferred database is much more learnable than the naïve solution
- We made the inferred database publicly available

Thank you for your attention!