Nearest-Biclusters Collaborative Filtering

Philadelphia, 20 August 2006

Speaker : Panagiotis Symeonidis
PhD Candidate
Scholar of the State Scholarships Foundation
Aristotle University of Thessaloniki, Greece
symeon@delab.csd.auth.gr
http://delab.csd.auth.gr/~symeon

Authors: Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, Yannis Manolopoulos.

What is Collaborative Filtering (CF)?

- CF is a successful recommendation technique used the last decade to confront the "information overload" in the internet.
- CF helps a customer to find what he interested in.

Related work on CF

- In 1994, GroupLens implemented a CF algorithm based on users' similarities.

  It is well-known as *user-based algorithm(UB)*.

- In 2001, *item-based algorithm* (IB) is proposed. (Sarwar et al.) It is based on the items' similarities.

- Several model-based approaches (mainly k-means clustering). They develop a model of user ratings.

Basic Challenges for CF algorithms

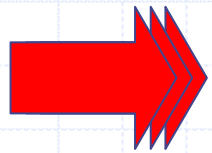
- Accuracy in recommendations: Users must be satisfied from items' suggestions.
- Scalability: Algorithms face performance problems as the volume of data increases.

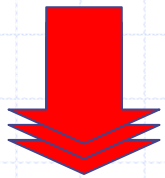
Motivation of our work(1)

Nearest Neighbors algorithms(UB, IB) cannot handle scalability to large volumes of data.

e.g.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	5	-	2	-	1	-	-
U_2	2	-	4	1	4	3	-
U_3	4	-	2	-	2	-	5
U_4	-	3	1	4	-	5	2
U_5	-	2	4	2	5	1	-
U_6	5	1	-	1	-	-	3
U_7	-	2	5	-	4	1	-
U_8	1	4	-	5	4	3	-





Motivation of our work(2)

- UB and IB are both one-sided approaches.
  (ignore the duality between users and items)

e.g.

	U1	U2	U3
U1	0	0.5	0.2
U2	0.5	0	0.1
U3	0.2	0.1	0

(User-User similarity matrix)

	I1	I2	I3
I1	0	0.1	0.2
I2	0.1	0	0.7
I3	0.2	0.7	0

(Item-Item similarity matrix)

Motivation of our work(3)

UB and IB cannot not detect partial matching. (they just find the less dissimilar users/items)

e.g.

	I1	I2	I3	I4	I5
U1	5	5	1	1	1
U2	5	5	5	5	5

(1-5 rating scale)

The above users would have negative similarity in UB and IB.
SO, WE MISS THEIR PARTIAL MATCHING..

Motivation of our work(4)

Traditional model-based algorithms (k-means, H-clustering) place each item/user in one cluster.

e.g.

	Sports			Computers	
	I1	I2	I3	I4	I5
U1	5	5	-	5	5

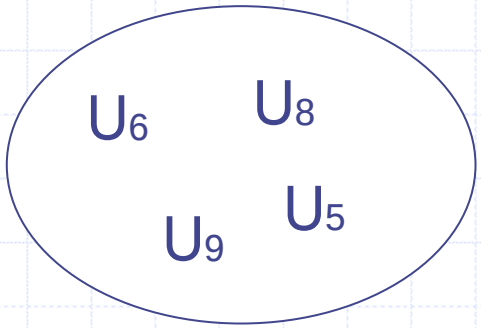
(bookstore)

The above user can have many different preferences or an item can belong in many different item categories.

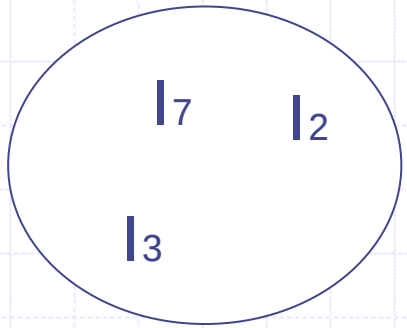
Motivation of our work(5)

- K-means and H-clustering algorithms again ignore the duality of data. (one sided approach)

e.g.



Create clusters only of users



or only of items

What we propose

- Biclustering to disclose the duality between users and items by grouping them in both dimensions simultaneously.

- a novel nearest-biclusters CF algorithm which uses a new similarity measure to achieve partial matching of users' preferences.

Related work in Biclustering

- Cheng and Church algorithm – uses mean square residue score to construct biclusters.
- xMotif algorithm - extracts motifs.
- Bimax : finds binary maximal-inclusion bicliques.

Related work in CF

- No related work has applied an exact biclustering algorithm.
- Hoffman and Puzicha proposed just a latent class model where clustering is performed seperately for users and for items.

Our Contribution

- Apply an exact biclustering algorithm in CF.
- Propose a novel nearest-biclusters CF algorithm.
- Use a new similarity measure for partial matching.
- Provide extensive experimental results.

Our Methodology

a. The data preprocessing step(optional).

b. The biclustering process.

c. The nearest-biclusters algorithm.

Running Example

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	5	-	2	-	1	-	-
U_2	2	-	4	1	4	3	-
U_3	4	-	2	-	2	-	5
U_4	-	3	1	4	-	5	2
U_5	-	2	4	2	5	1	-
U_6	5	1	-	1	-	-	3
U_7	-	2	5	-	4	1	-
U_8	1	4	-	5	4	3	-

(Training Set)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_9	5	-	4	-	1	-	2

(Test Set)

Rating scale : 1-5

a. The data preprocessing step (optional)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	5	-	-	-	-	-	-
U_2	-	-	4	-	4	3	-
U_3	4	-	-	-	-	-	5
U_4	-	3	-	4	-	5	-
U_5	-	-	4	-	5	-	-
U_6	5	-	-	-	-	-	3
U_7	-	-	5	-	4	-	-
U_8	-	4	-	5	4	3	-

Training Set with $P_t > 2$

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	1	0	0	0	0	0	0
U_2	0	0	1	0	1	1	0
U_3	1	0	0	0	0	0	1
U_4	0	1	0	1	0	1	0
U_5	0	0	1	0	1	0	0
U_6	1	0	0	0	0	0	1
U_7	0	0	1	0	1	0	0
U_8	0	1	0	1	1	1	0

Binary discretization of the Training Set.

P_t : Positive Rating Threshold

b. The biclustering process(1)

- Use Bimax algorithm : Binary inclusion-maximal algorithm.

- A bicluster $b(U_b , I_b)$ corresponds to a subset of users U_b that jointly present positively rating behavior across a subset of items I_b.

- In other words for Bimax, the pair (U_b , I_b) defines a submatrix for which all elements equal to 1 and is not entirely contained in any other bicluster.

b. The biclustering process(2)

	I_4	I_2	I_6	I_5	I_3	I_1	I_7
U_3	0	0	0	0	0	1	1
U_6	0	0	0	0	0	1	1
U_5	0	0	0	1	1	0	0
U_7	0	0	0	1	1	0	0
U_2	0	0	1	1	1	0	0
U_8	1	1	1	1	0	0	0
U_4	1	1	1	0	0	0	0
U_1	0	0	0	0	0	1	0

$$b_1: \quad U_{b_1} = \{U_3, U_6\}, \qquad I_{b_1} = \{I_1, I_7\}$$
$$b_2: \quad U_{b_2} = \{U_5, U_7, U_2\}, \qquad I_{b_2} = \{I_5, I_3\}$$
$$b_3: \quad U_{b_3} = \{U_2, U_8\}, \qquad I_{b_3} = \{I_6, I_5\}$$
$$b_4: \quad U_{b_4} = \{U_8, U_4\}, \qquad I_{b_4} = \{I_4, I_2, I_6\}$$

Applying Bimax to Training Set.

Input parameters:
1. Min. number of users
2. Min. number of items
(here is 2 for both)

- **Four biclusters found.**
- **overlapping between biclusters.**
- **well-tuning of overlapping.**

c. The nearest-Biclusters algorithm(1)

It consists of two basic operations:

- **The formation of the test user neighborhood, i.e. to find the k-nearest biclusters.**

- **The generation of the top-N recommendation list.**

c. The nearest-Biclusters algorithm(2)

To find the k-nearest biclusters of a test user:

$$sim(u,b) = \frac{|I_u \bigcap I_b|}{|I_u \bigcap I_b| + |I_b - I_u|}$$

- **We divide items they have in common to the sum of items they have in common and the number of items they differ.**
- **Similarity values range between [0,1].**

c. The nearest-Biclusters algorithm(3)

To generate the top-N recommendation list :

$$WF(i,b) = sim(u,b) * |U_b|$$

- **Weighted Frequency (WF) of an item i in a bicluster b is the product between $|U_b|$ and the similarity measure $sim(u,b)$**
- **We weight the contribution of each bicluster with its size, in addition to its similarity with the test user.**

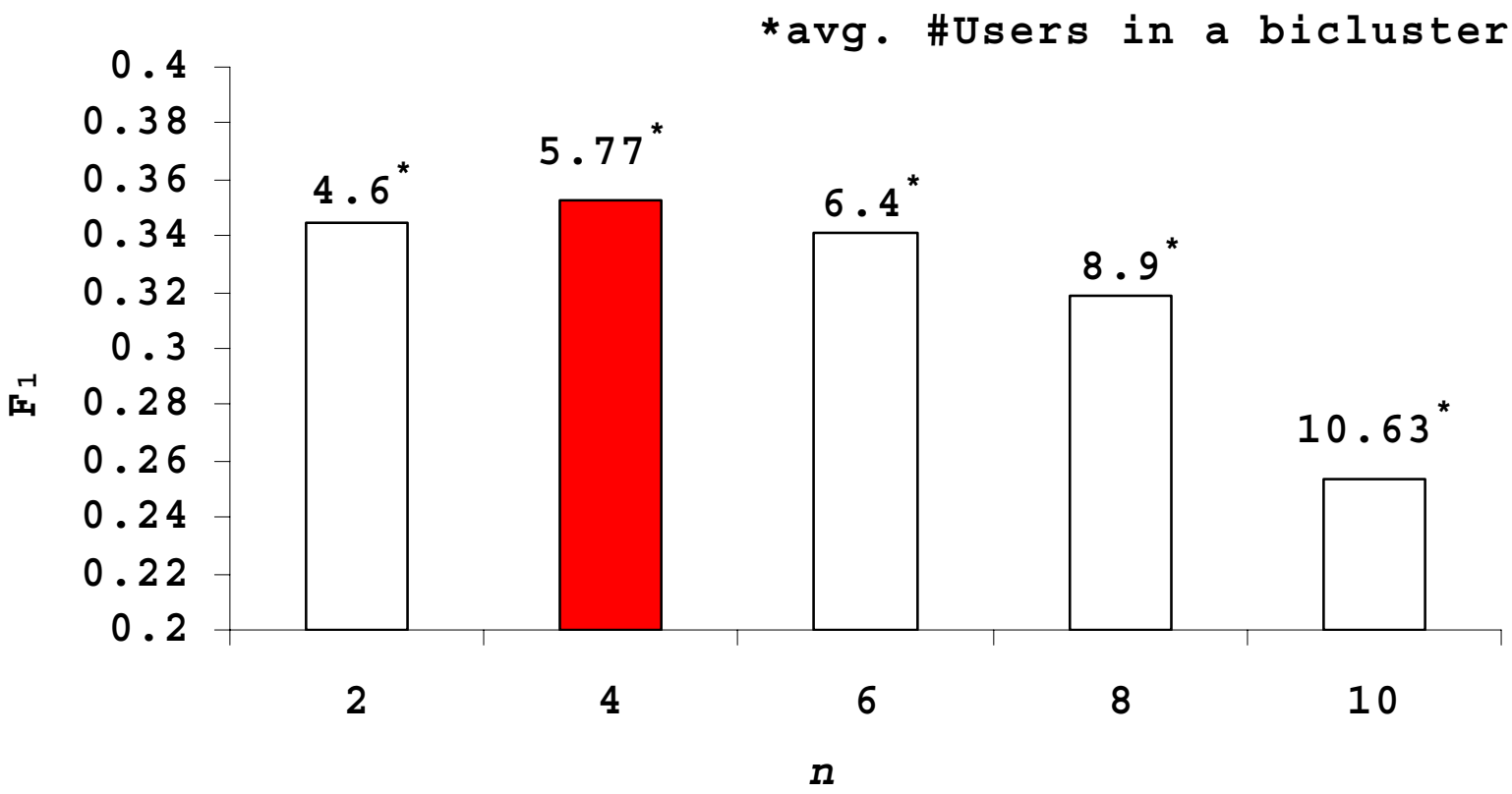
Evaluating the CF process

- Evaluation is done through Precision, Recall and F1 metric.
- Note that, MAE is not indicative for the quality of the top-N list, but only for the quality of the similarity measure.

Experimental Configuration

- Compare nearest-biclusters , UB and IB algorithms in three real datasets.
  (Movielens 100k and 1M, Eachmovie)
- We present results for Movielens 100k.
- Top-*N* list : 20 items
- *k*-nearest neighbors: 1-100

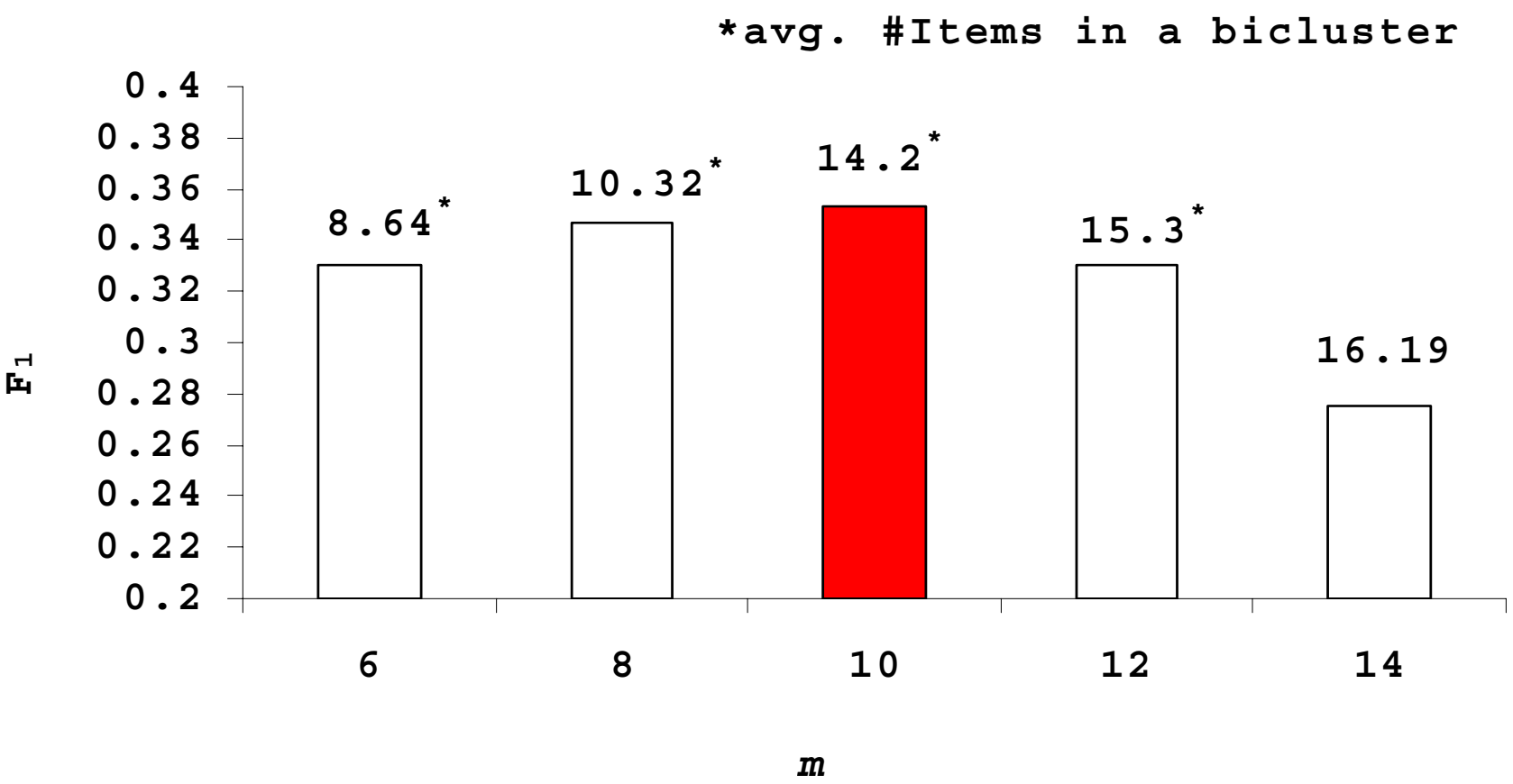
Tuning of users' initial parameter(1)



Tuning of the minimum number of users parameter in a bicluster.

(n= 4 users in a bicluster)

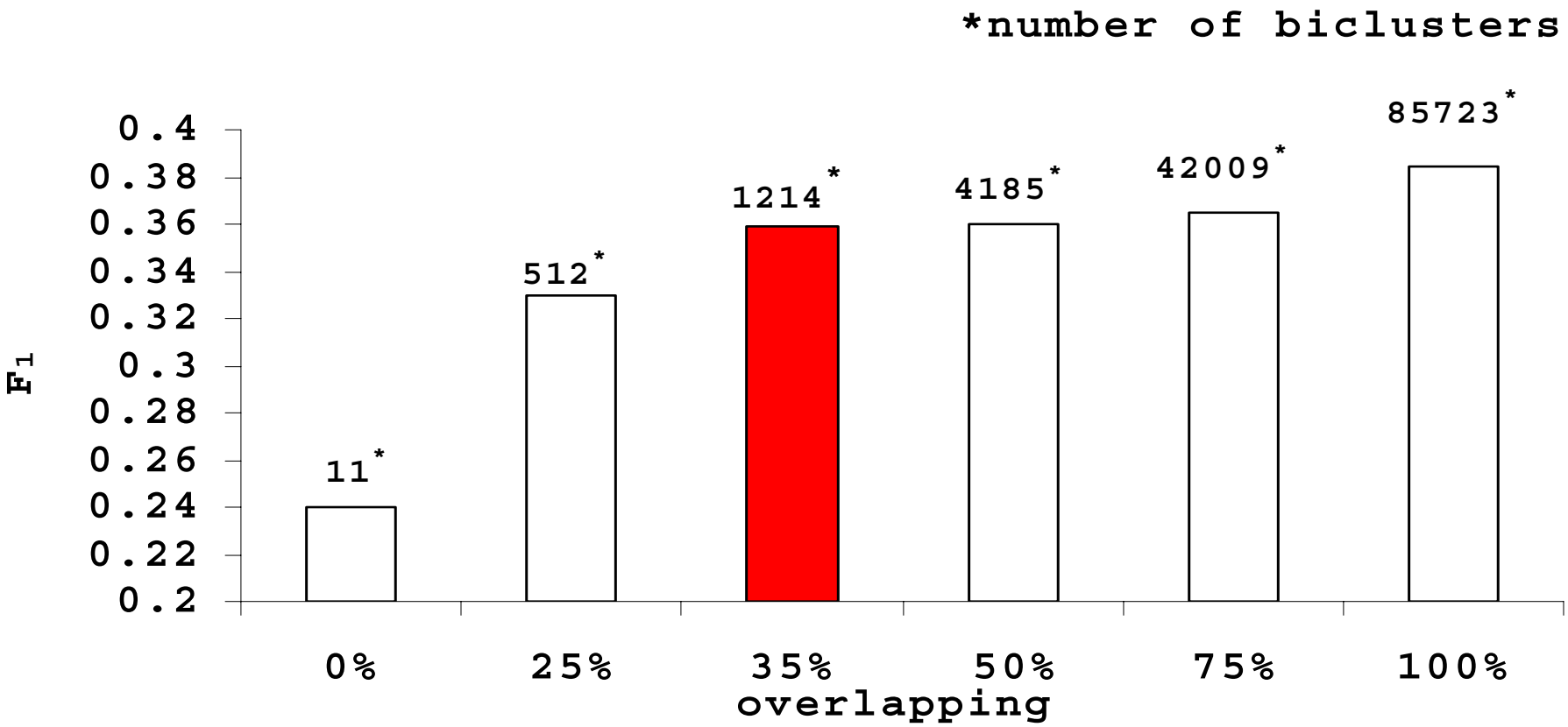
Tuning of items' initial parameter(2)



Tuning of the minimum number of items parameter in a bicluster.

(m= 10 items in a bicluster)

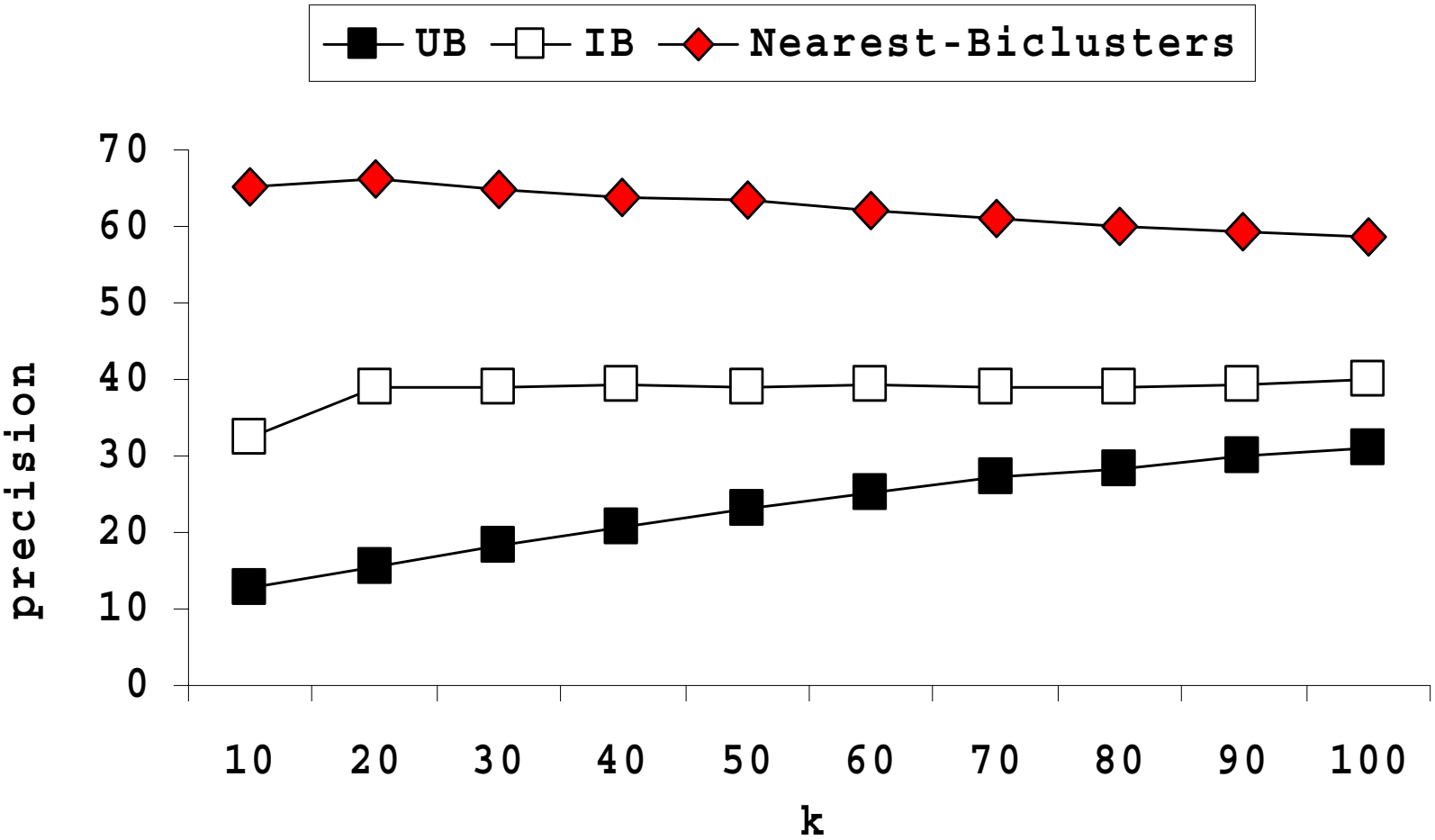
Tuning of overlapping factor(3)



Tuning of the number of overlapping biclusters.

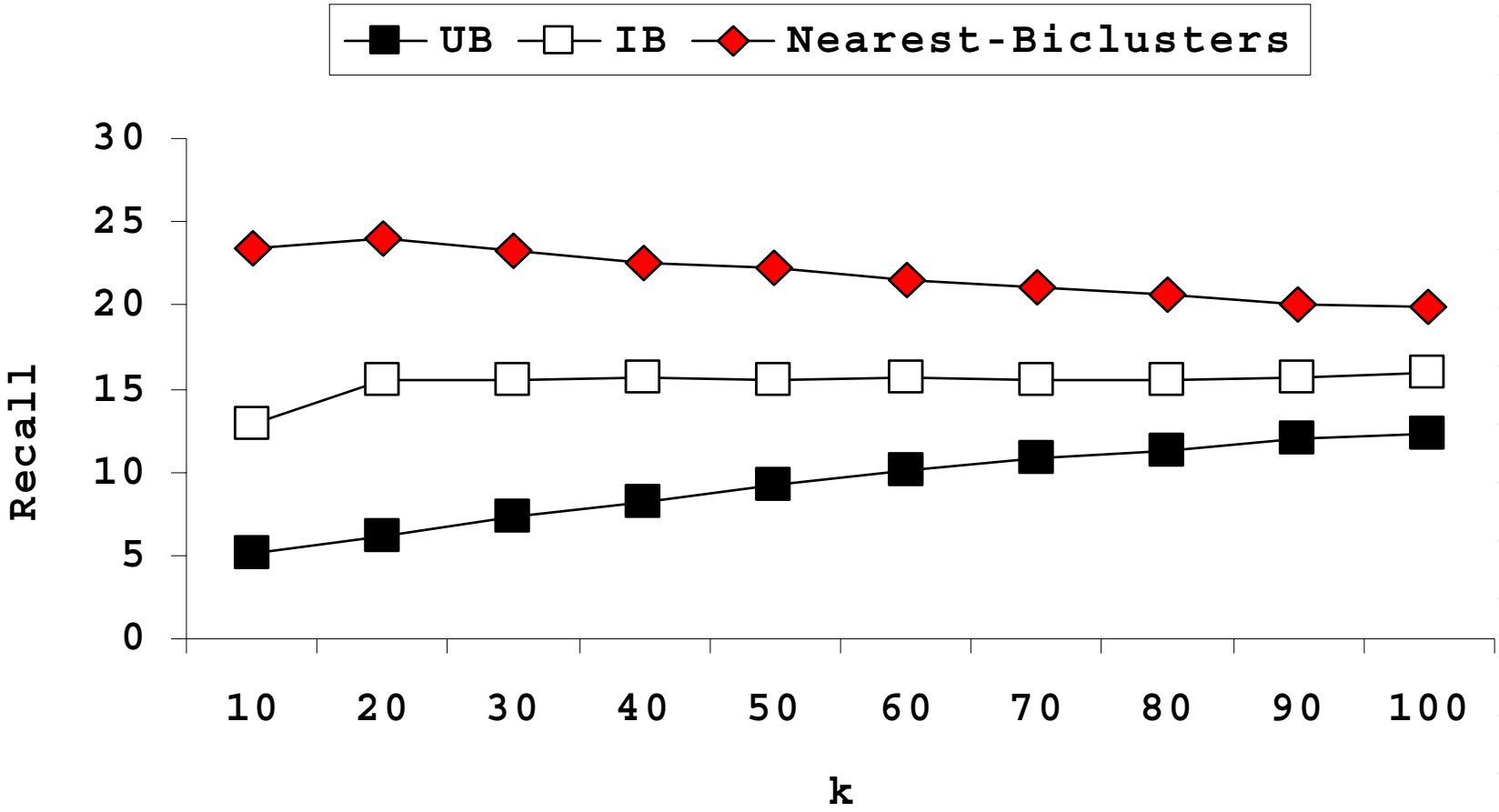
(35% overlapping)

Comparative Results for accuracy(1)



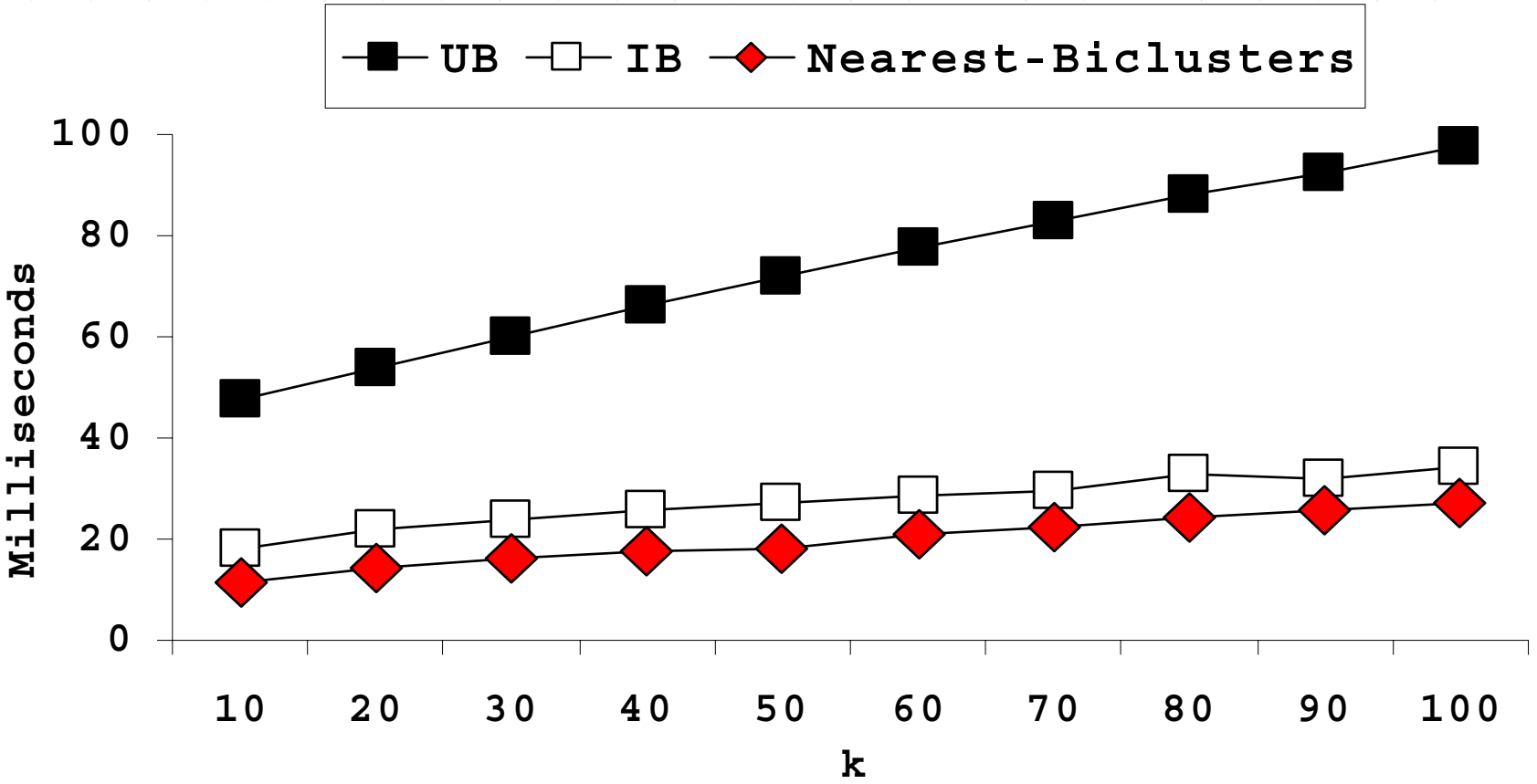
(30% more precision)

Comparative Results for accuracy(2)



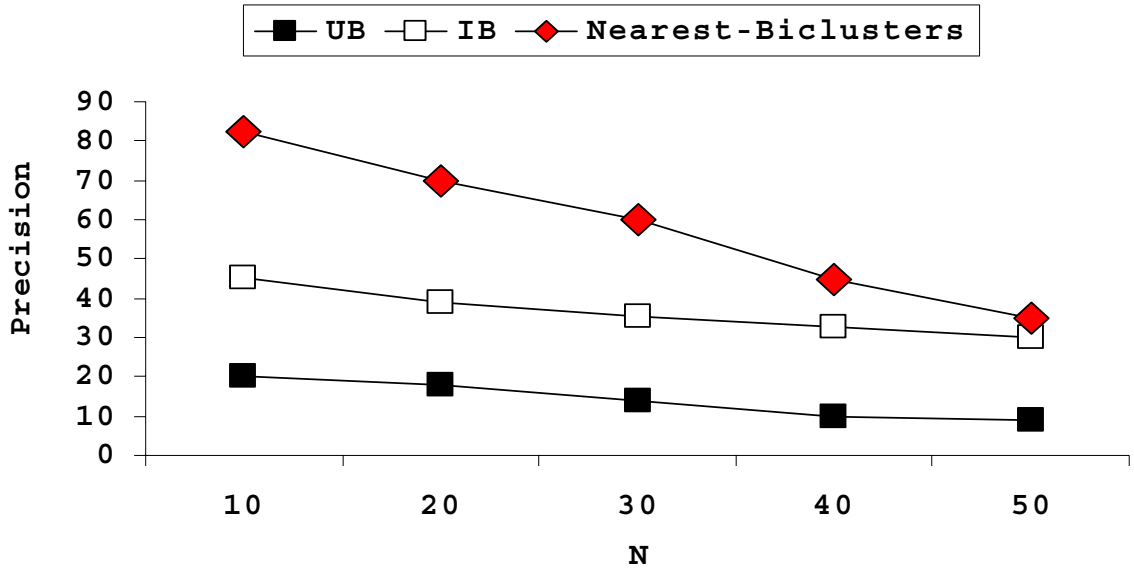
(10% more recall)

Comparative Results for execution time

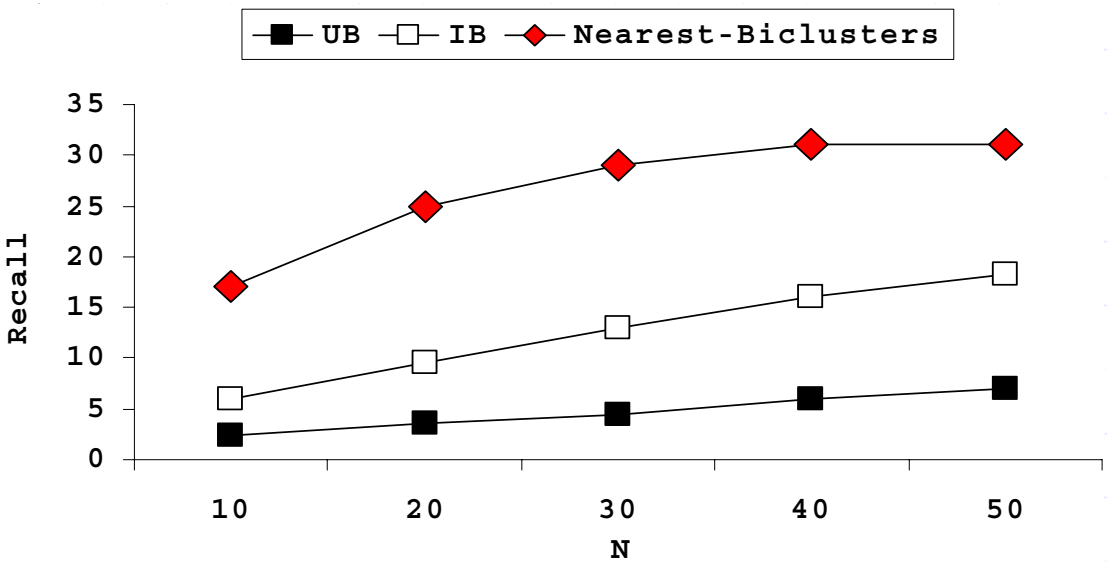


(Nearest-biclusters is faster than IB algorithm)

Examination of additional factors (1)

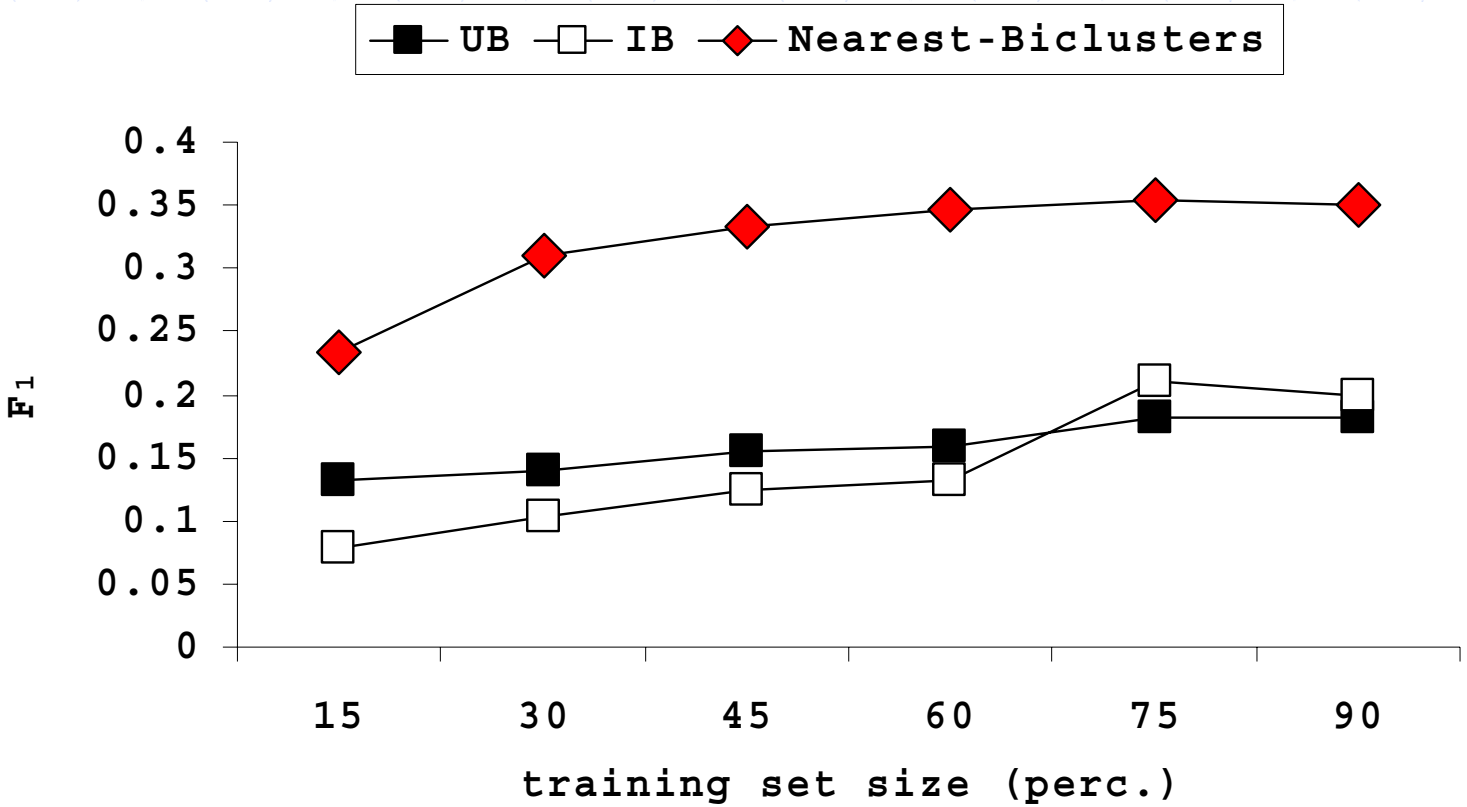


Precision vs. recommendation list size (N).



Recall vs. recommendation list size (N).

Examination of additional factors (2)



- F_1 metric vs. training set size.
- Note that a 15% of the training set of nearest-biclusters algorithm gives better F_1 than gives the 75% of the training set for the UB and IB cases.

Conclusions

- Our approach shows more than 30% improvement in terms of precision than UB and IB.
- Our approach shows improvement in terms of efficiency (beats even the IB algorithm).
- We introduced a novel similarity measure for the user's neighborhood formation and Weighted Frequency for the top-N list generation.

Future Work

- Examine other classes of biclustering algorithms as well. (coherent algorithms etc.)
- Test different similarity measures between a user and a bicluster.

THANK YOU.

symeon@delab.csd.auth.gr

http://delab.csd.auth.gr/~symeon