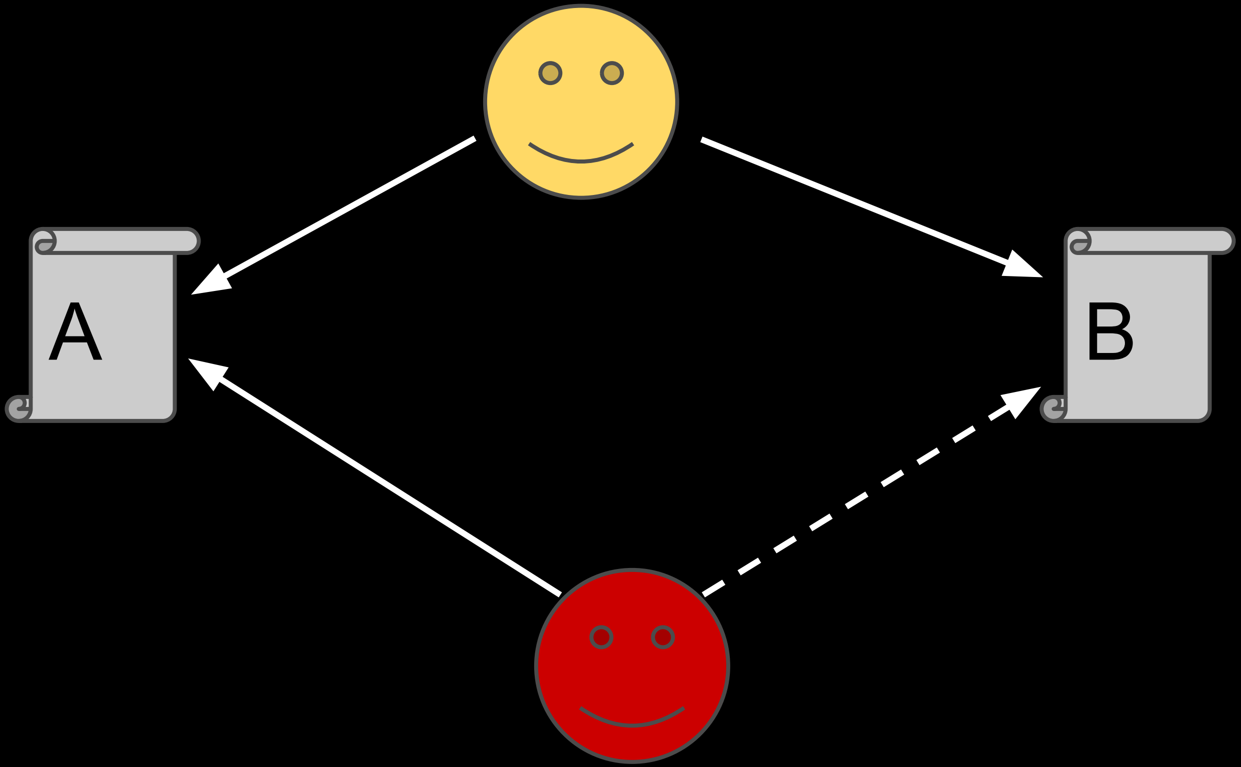
Collaborative Filtering

Patrick Heia Romstad
Dag Einar Monsen

Collaborative filtering



Research questions

- What are the ***relevant*** collaborative filtering techniques for ***news*** recommendation?

- How do ***model-based*** and ***memory-based*** filtering techniques ***compare*** for the domain of news recommendation?

Approach

1. Survey of collaborative filtering literature
2. Implement memory- and model-based collaborative filtering algorithms
3. Compare and evaluate

Memory based approach

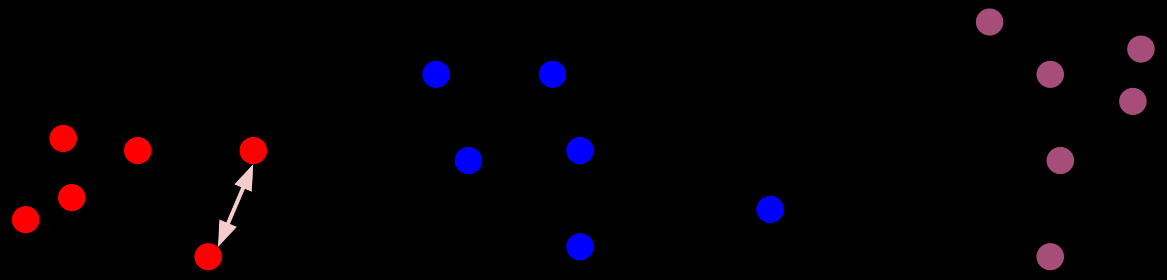
- Neighborhoods of users
- KNN, Threshold

	Article 1	Article 2	Article 3	Article 4
Dag	1	1	1	
Patrick			1	1
Edvard		1	1	1

- Experiment with different values for K and different thresholds

Model-based approach

- Cluster users into K clusters
- What value for K?
  - Experiment!
  - Threshold of >=100 users per cluster
- Evaluate neighborhood methods on each cluster



Challenges

- Performance and scale
- Data sparsity
- Evaluation
- High churn in news domain

Data set

- Click stream from Arena Partners operated news site in Finland between 15.06 - 15.07 this year

Users	Items	Ratings	Density
2123	2438	35 890	**0.69%**

Evaluation

- Top N recommendation task

Arbitrary with boolean data set, but we assume the extracted items are relevant

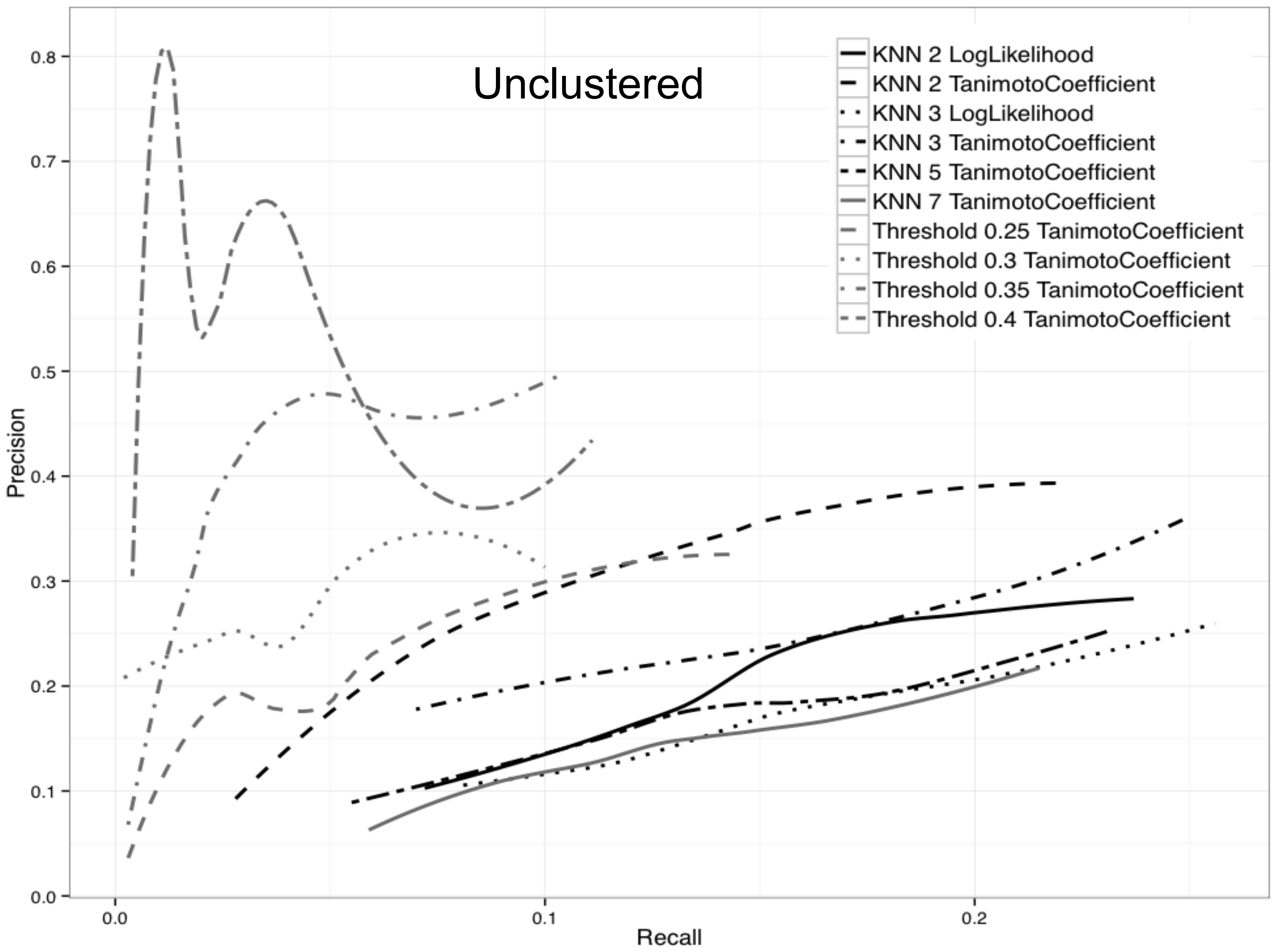
1. Extract the top N ratings for a user
2. Recommend N items to the user
3. See how many of the items extracted in step 1 that are returned

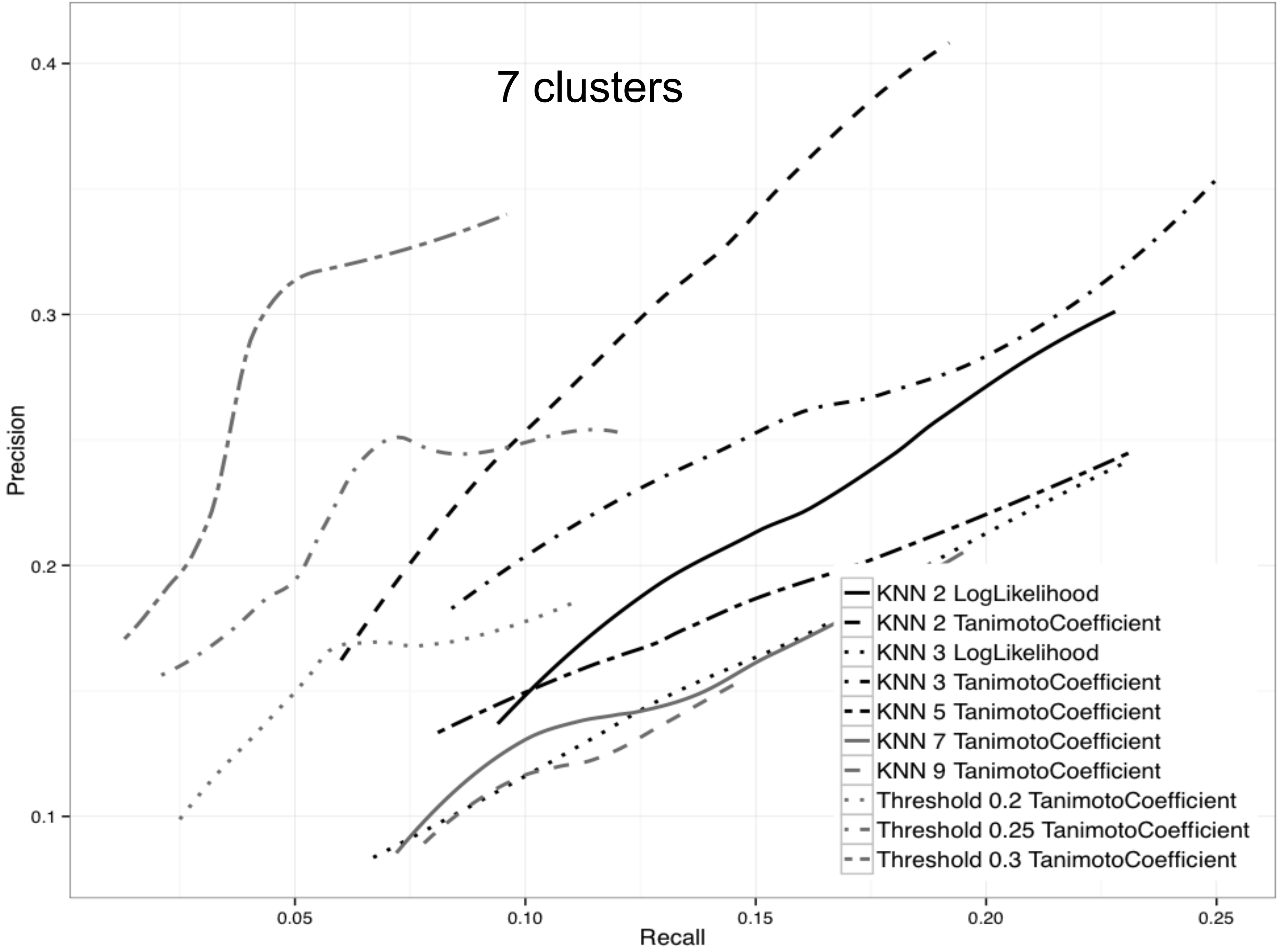
Precision / Recall

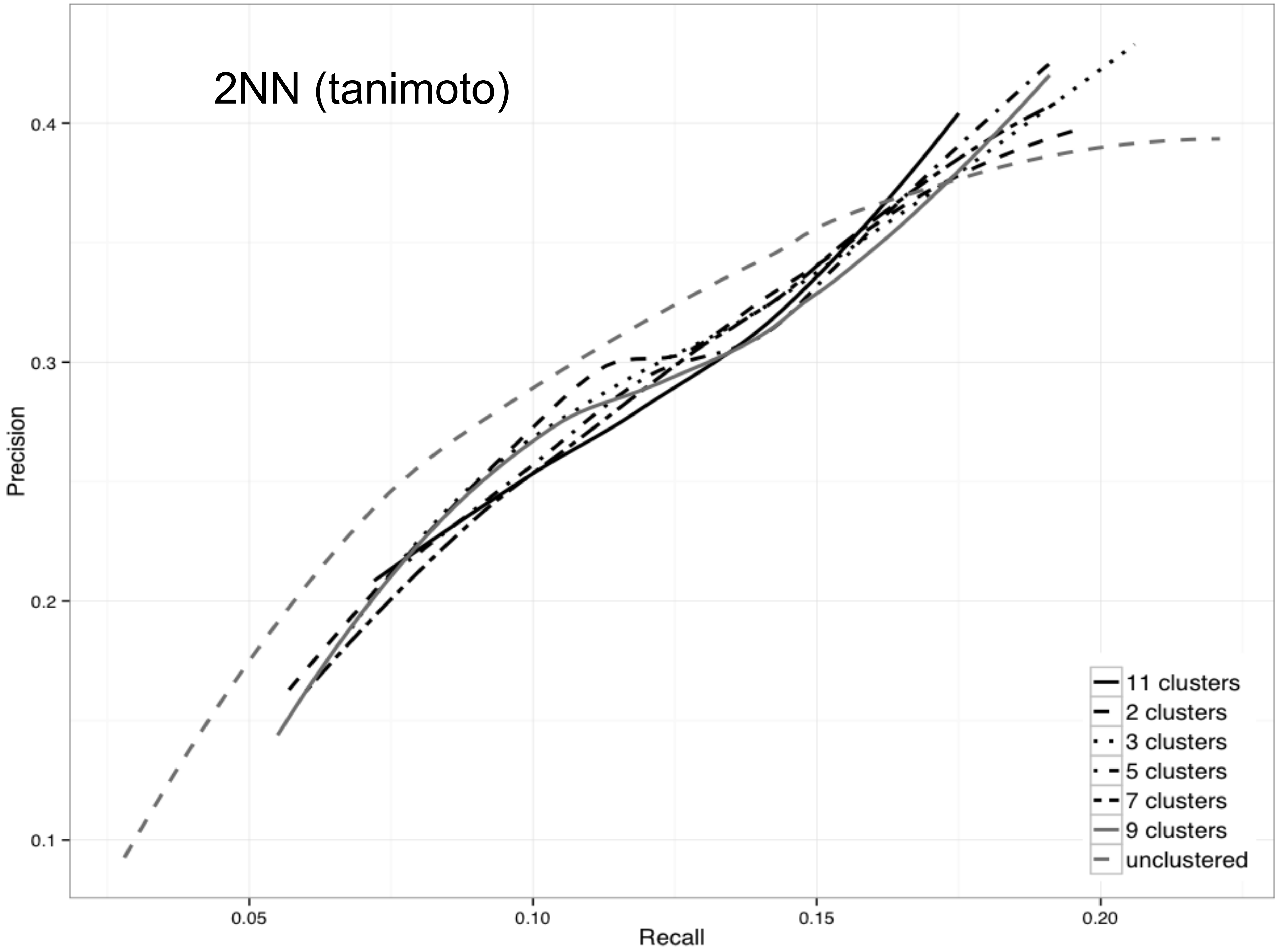
$$\text{Precision} = \frac{\text{number of } \textbf{\textit{relevant}} \text{ items retrieved}}{\text{total number of items retrieved}}$$

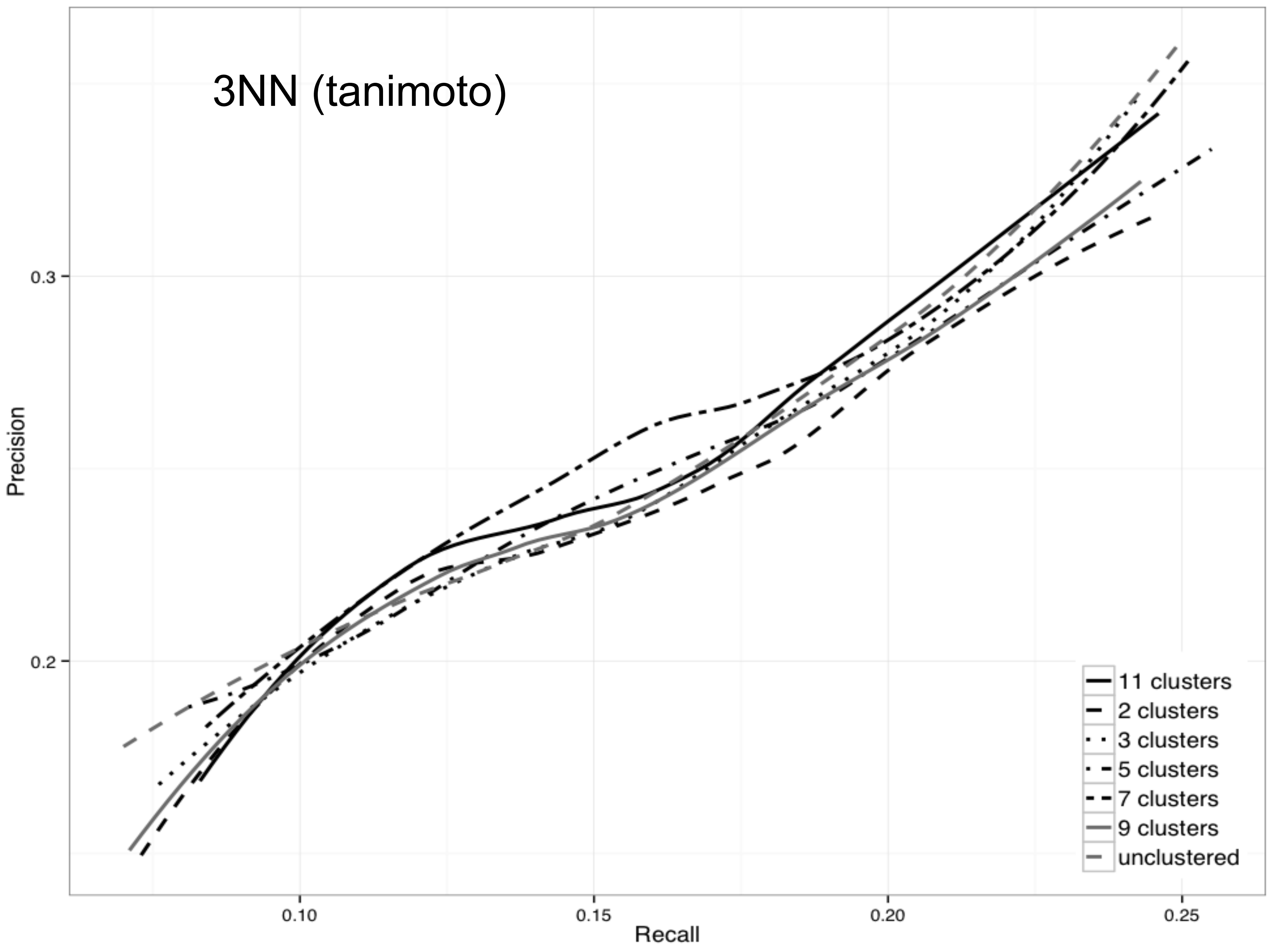
$$\text{Recall} = \frac{\text{number of } \textbf{\textit{relevant}} \text{ items retrieved}}{\text{number of } \textbf{\textit{relevant}} \text{ items in } \textbf{\textit{collection}}}$$

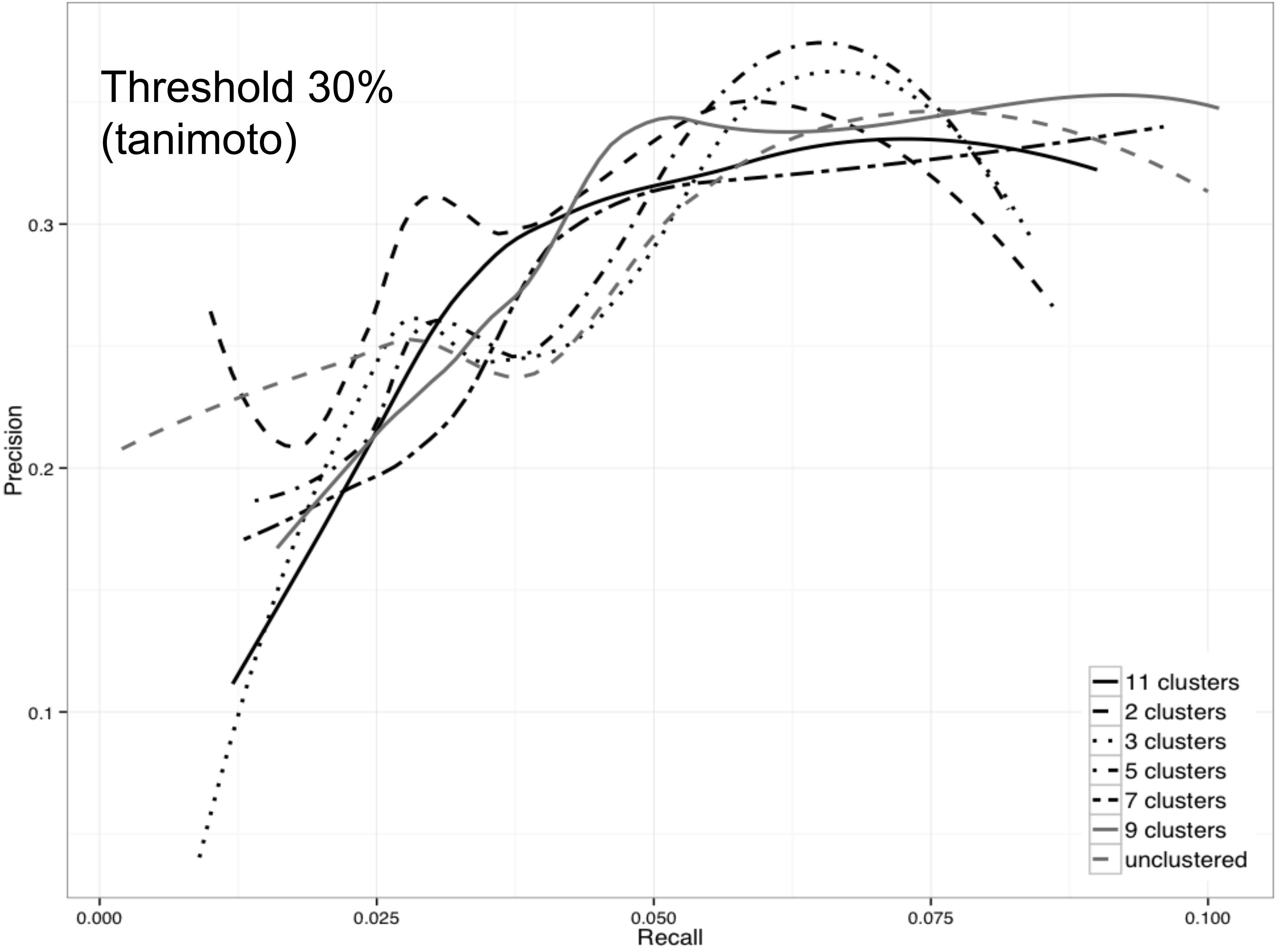
Results


Unclustered
KNN 2 LogLikelihood
KNN 2 TanimotoCoefficient
KNN 3 LogLikelihood
KNN 3 TanimotoCoefficient
KNN 5 TanimotoCoefficient
KNN 7 TanimotoCoefficient
Threshold 0.25 TanimotoCoefficient
Threshold 0.3 TanimotoCoefficient
Threshold 0.35 TanimotoCoefficient
Threshold 0.4 TanimotoCoefficient
Precision
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Recall
0.0
0.1
0.2

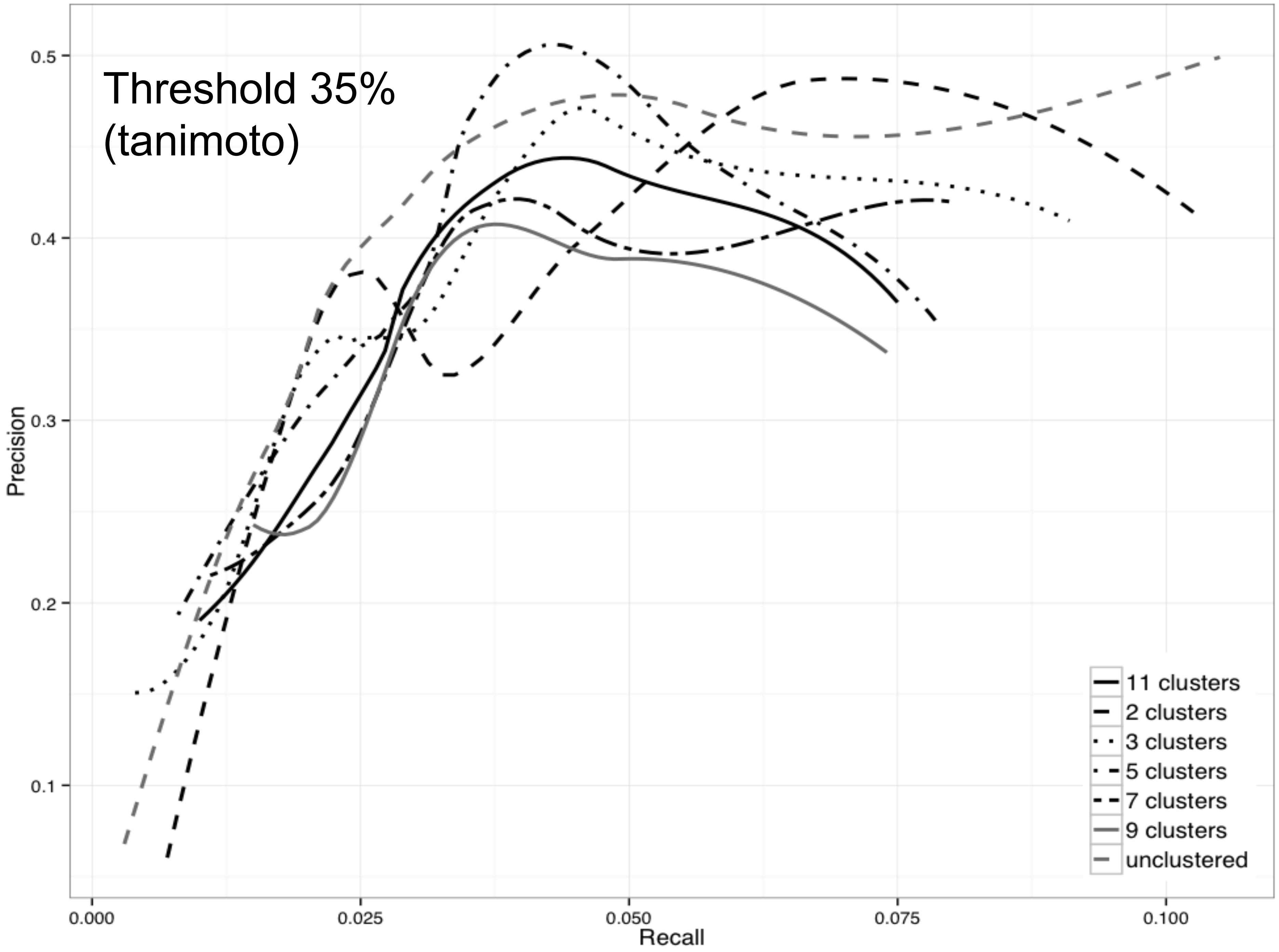

7 clusters
Precision
Recall
0.1
0.2
0.3
0.4
0.05
0.10
0.15
0.20
0.25
KNN 2 LogLikelihood
KNN 2 TanimotoCoefficient
KNN 3 LogLikelihood
KNN 3 TanimotoCoefficient
KNN 5 TanimotoCoefficient
KNN 7 TanimotoCoefficient
KNN 9 TanimotoCoefficient
Threshold 0.2 TanimotoCoefficient
Threshold 0.25 TanimotoCoefficient
Threshold 0.3 TanimotoCoefficient


2NN (tanimoto)
Precision
0.4
0.3
0.2
0.1
Recall
0.05
0.10
0.15
0.20
11 clusters
2 clusters
3 clusters
5 clusters
7 clusters
9 clusters
unclustered


3NN (tanimoto)
Precision
0.3
0.2
Recall
0.10
0.15
0.20
0.25
11 clusters
2 clusters
3 clusters
5 clusters
7 clusters
9 clusters
unclustered



Threshold 30%
(tanimoto)


Threshold 35%
(tanimoto)
Precision
0.1
0.2
0.3
0.4
0.5
Recall
0.000
0.025
0.050
0.075
0.100
11 clusters
2 clusters
3 clusters
5 clusters
7 clusters
9 clusters
unclustered

Main findings

- Trade off between precision and recall
- Optimal threshold values changes on different clusters
- KNN algorithms scales well with clusters
- Small neighborhoods are preferred
- Tanimoto coefficient is the preferred similarity metric

Future plans

- Explore *Latent Factor Models*
- Mitigate challenges with item churn

Questions?