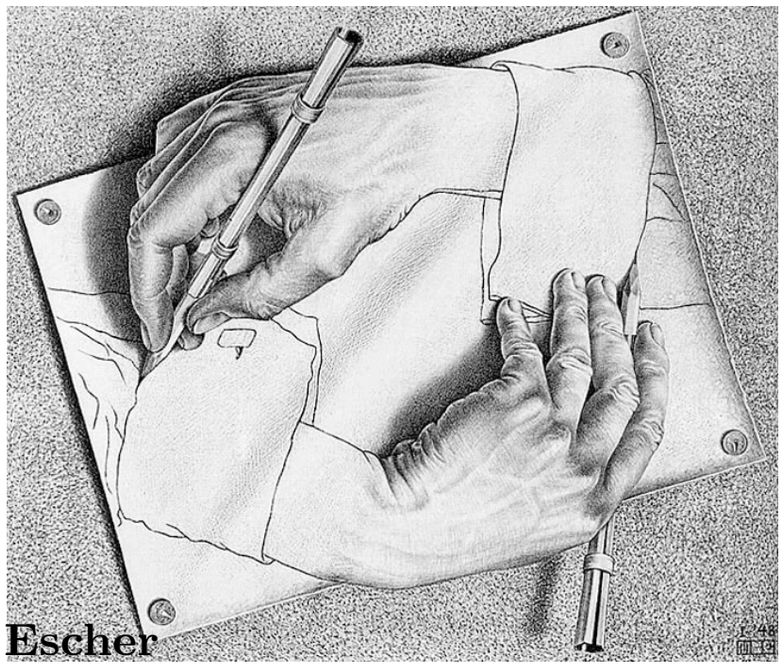
Collaborative Filtering


Escher

Onkar Dabeer
School of Technology and Computer Science
Tata Institute of Fundamental Research
Mumbai, India

Outline

Outline

- Introduction (15 mins)
  - Recommendation systems
  - The Collaborative Filtering component

Outline

- Introduction (15 mins)
  - Recommendation systems
  - The Collaborative Filtering component
- State of the Art (15 mins)
  - Neighborhood methods and latent factor models
  - Blend many schemes
  - Provably good principles?

Outline

- Introduction (15 mins)
  - Recommendation systems
  - The Collaborative Filtering component
- State of the Art (15 mins)
  - Neighborhood methods and latent factor models
  - Blend many schemes
  - Provably good principles?
- Two mathematical models (15 mins)
  - Estimation of rearranged smooth fields
  - Low-rank matrix completion

Outline (Contd.)

Outline (Contd.)

- Low-rank matrix completion (20 mins)
  - Candes, Recht (2008) - links with CS
  - Keshavan, Oh, Montanari (2009) - link with SVD
  - Many others

Outline (Contd.)

- Low-rank matrix completion (20 mins)
  - Candes, Recht (2008) - links with CS
  - Keshavan, Oh, Montanari (2009) - link with SVD
  - Many others
- Estimation of rearranged smooth processes (90 mins)
  - A "channel coding" result (Aditya, D, Dey, 2009)
  - Popularity Amongst Friends (PAF) algorithm
  - Empirical performance of PAF
  - BER analysis of PAF (Barman, D, 2010)

Outline (Contd.)

- Low-rank matrix completion (20 mins)
  - Candes, Recht (2008) - links with CS
  - Keshavan, Oh, Montanari (2009) - link with SVD
  - Many others
- Estimation of rearranged smooth processes (90 mins)
  - A "channel coding" result (Aditya, D, Dey, 2009)
  - Popularity Amongst Friends (PAF) algorithm
  - Empirical performance of PAF
  - BER analysis of PAF (Barman, D, 2010)
- Future directions (25 mins)

Introduction

Beyond Search

Social Network

Media

Shopping

Events

Services

Travel

Beyond Search

Social Network

Media

Shopping

Events

Services

Web Search:
Long list of
related items

Travel

Beyond Search

Social Network

Media

Shopping

Events

Services

Web Search:
Long list of
related items

Travel

Recommendations:
Few “likable” items
Limited domain

Recommendations in Action

- Amazon
  - People who bought this also bought...
- Google News: recommended stories
- iTunes - Genius sidebar
- Netflix
  - Suggests movies using rating matrix
- Facebook, LinkedIn
  - Suggest connections
- RichRelevance recommendation engine
  - Disney Stores, Sears, Office Depot, etc.

Challenges

Challenges

- A new kind of estimation problem

Challenges

- A new kind of estimation problem
- Massive data
  - iTunes: 10+ million songs, many more users
  - Scalability is paramount

Challenges

- A new kind of estimation problem
- Massive data
  - iTunes: 10+ million songs, many more users
  - Scalability is paramount
- Lack of good statistical models
  - Rating - subjective, context-dependent; spam

Challenges

- A new kind of estimation problem
- Massive data
  - iTunes: 10+ million songs, many more users
  - Scalability is paramount
- Lack of good statistical models
  - Rating - subjective, context-dependent; spam
- Sparse observations
  - Computational boon but makes model inference difficult

Challenges

- A new kind of estimation problem
- Massive data
  - iTunes: 10+ million songs, many more users
  - Scalability is paramount
- Lack of good statistical models
  - Rating - subjective, context-dependent; spam
- Sparse observations
  - Computational boon but makes model inference difficult
- Privacy issues

What Data Can We Use?

What Data Can We Use?

User Features

Location
Gender
Age
Contacts

What Data Can We Use?

User Features

Location
Gender
Age
Contacts

Item Features

Director
Cast
Year
Genre

What Data Can We Use?

User Features	Joint Features	Item Features
Location	Time	Director
Gender	Ratings (Netflix)	Cast
Age	Clicks (Google News)	Year
Contacts	Buy (Amazon)	Genre

Some Datasets

Some Datasets

- Movielens
  - 3900 movies, 6040 users, 1 million ratings
  - 10681 movies, 71567 users, 10 million ratings

Some Datasets

- Movielens
  - 3900 movies, 6040 users, 1 million ratings
  - 10681 movies, 71567 users, 10 million ratings
- Netflix
  - 17,770 movies, 480,189 users, about 100 million ratings, date of rating, movie names
  - Not online anymore but I have a copy

Some Datasets

- Movielens
  - 3900 movies, 6040 users, 1 million ratings
  - 10681 movies, 71567 users, 10 million ratings
- Netflix
  - 17,770 movies, 480,189 users, about 100 million ratings, date of rating, movie names
  - Not online anymore but I have a copy
- Yahoo Music data
  - 136,000 songs, 1.8 million users, 717 million ratings
  - Genre, artist, album attributes

Collaborative Filtering

Collaborative Filtering

- Content based recommendations
  - Use similarity of item contents

Collaborative Filtering

- Content based recommendations
  - Use similarity of item contents
- Collaborative filtering (CF)
  - Exploit past user-item rating data
  - He likes what she likes

Collaborative Filtering

- Content based recommendations
  - Use similarity of item contents
- Collaborative filtering (CF)
  - Exploit past user-item rating data
  - He likes what she likes
- Many practical schemes use a blend of both
  - Collaborative filter with side-information

Collaborative Filtering

- Content based recommendations
  - Use similarity of item contents
- Collaborative filtering (CF)
  - Exploit past user-item rating data
  - He likes what she likes
- Many practical schemes use a blend of both
  - Collaborative filter with side-information
- In this tutorial: CF with no side-information

The Problem

Items

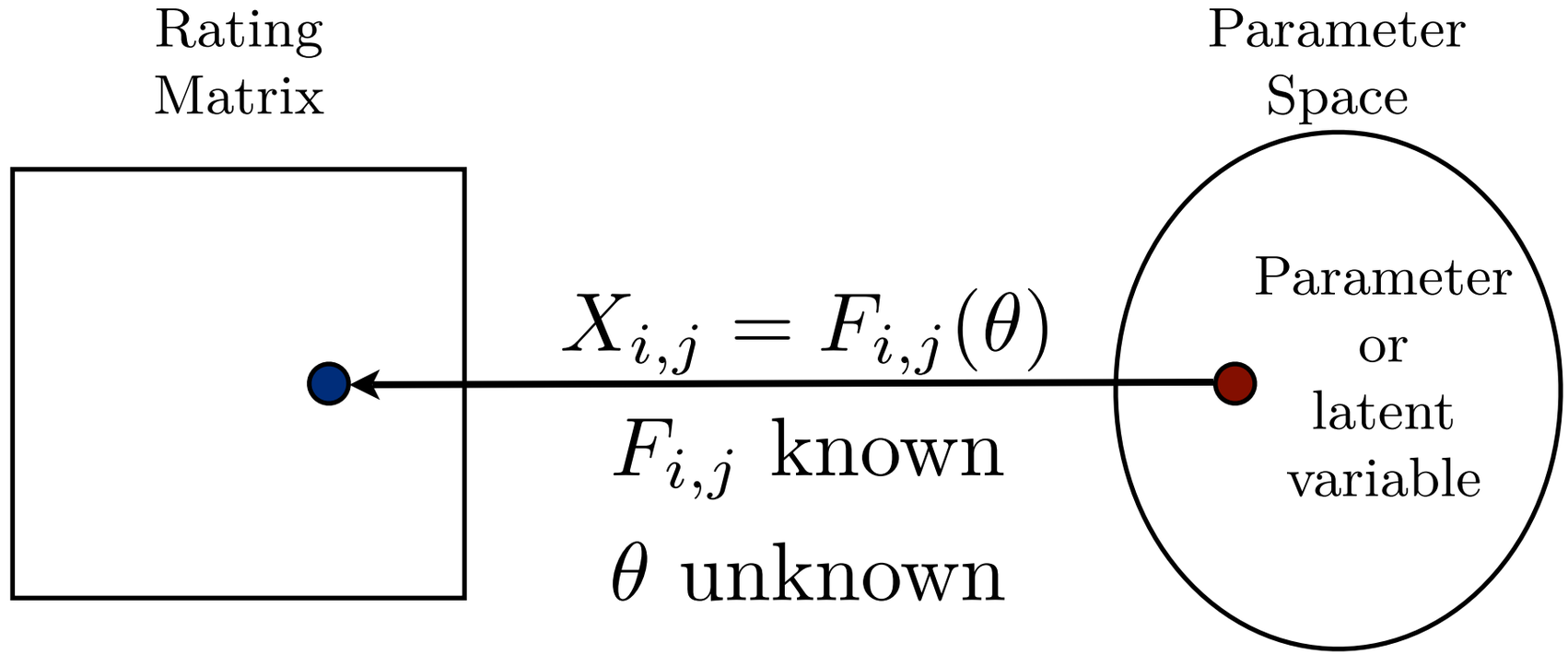
Users				
	2	?	1	3
	4	4	?	?
	?	3	1	?
	1	?	2	?

Recommend items based on available ratings

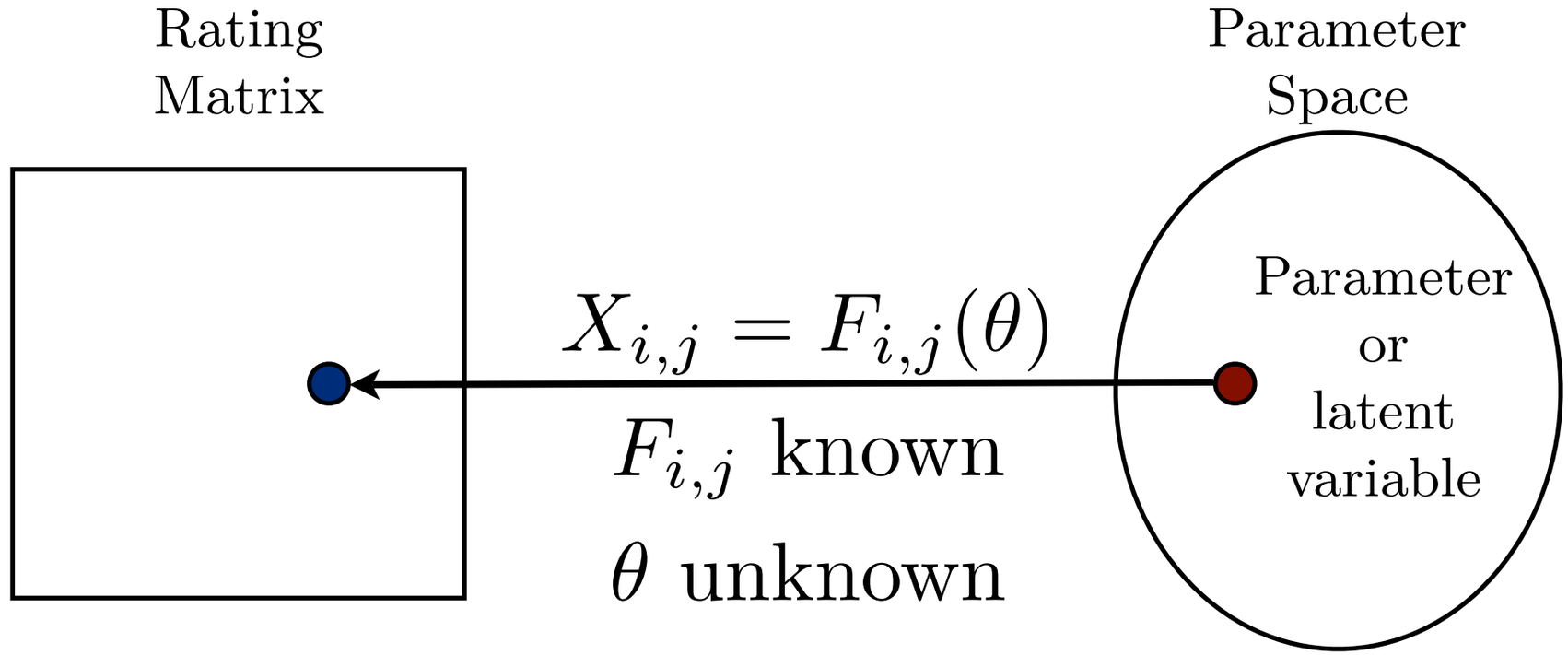
State of the Art

(Review papers: Adomavicius et al, 2005; Su et al 2009)

Latent Variable Model



Latent Variable Model



Example: $X \sim m \times n$, $U \sim m \times r$, $V \sim r \times n$,

$$X = UV, \quad \theta = (U, V)$$

Factor Analysis: An SVD Inspired Algorithm

- Factorization: $X = UV$, X is $m \times n$, U is $m \times r$, V is $r \times n$, and $r << \min\{m, n\}$.
- Alternating Least-Squares: Initialize U, solve

$$\min_V \|X - UV\|_F^2.$$

  Fix V, and solve for best U.

- For collaborative filtering: Evaluate error only over known entries, use gradient descent. (Variants in Koren, 2009; Keshavan, ISIT 2009).
- When does this method work?

Neighborhood Methods

Neighborhood Methods

- Identify a relevant "neighborhood" for any matrix entry
  - Related to clustering of users and/or items

Neighborhood Methods

- Identify a relevant "neighborhood" for any matrix entry
  - Related to clustering of users and/or items
- Predictor uses neighborhood data

Neighborhood Methods

- Identify a relevant "neighborhood" for any matrix entry
  - Related to clustering of users and/or items
- Predictor uses neighborhood data
- Parameters learnt from known data

Neighborhood Methods

- Identify a relevant "neighborhood" for any matrix entry
  - Related to clustering of users and/or items
- Predictor uses neighborhood data
- Parameters learnt from known data
- Example:
  - Neighborhood: K most similar items
  - Predictor: Affine map of average item ratings
  - Parameter estimation: Least-squares
  - A variant in Koren, 2009

Practical Schemes

Practical Schemes

- **Committee Methods**: Blend weak estimators

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters
  - The Netflix prize winner blends 100+ algorithms

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters
  - The Netflix prize winner blends 100+ algorithms
  - SVD inspired algorithms, neighborhood methods, exploiting time stamps, movie names

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters
  - The Netflix prize winner blends 100+ algorithms
  - SVD inspired algorithms, neighborhood methods, exploiting time stamps, movie names
  - Typical individual RMSE is 0.87-0.92 (on test data)

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters
  - The Netflix prize winner blends 100+ algorithms
  - SVD inspired algorithms, neighborhood methods, exploiting time stamps, movie names
  - Typical individual RMSE is 0.87-0.92 (on test data)
  - After blending, RMSE is about 0.85

Practical Schemes

- **Committee Methods**: Blend weak estimators
  - Strong empirical and theoretical backing
  - Popular in classification and regression
- Blending collaborative filters
  - The Netflix prize winner blends 100+ algorithms
  - SVD inspired algorithms, neighborhood methods, exploiting time stamps, movie names
  - Typical individual RMSE is 0.87-0.92 (on test data)
  - After blending, RMSE is about 0.85
  - RichRelevance blends about 40 algorithms

Some Questions

Some Questions

- Is a reduction of RMSE from 0.95 to 0.85 relevant on a scale of 1-5?
  - Why not probability of error?

Some Questions

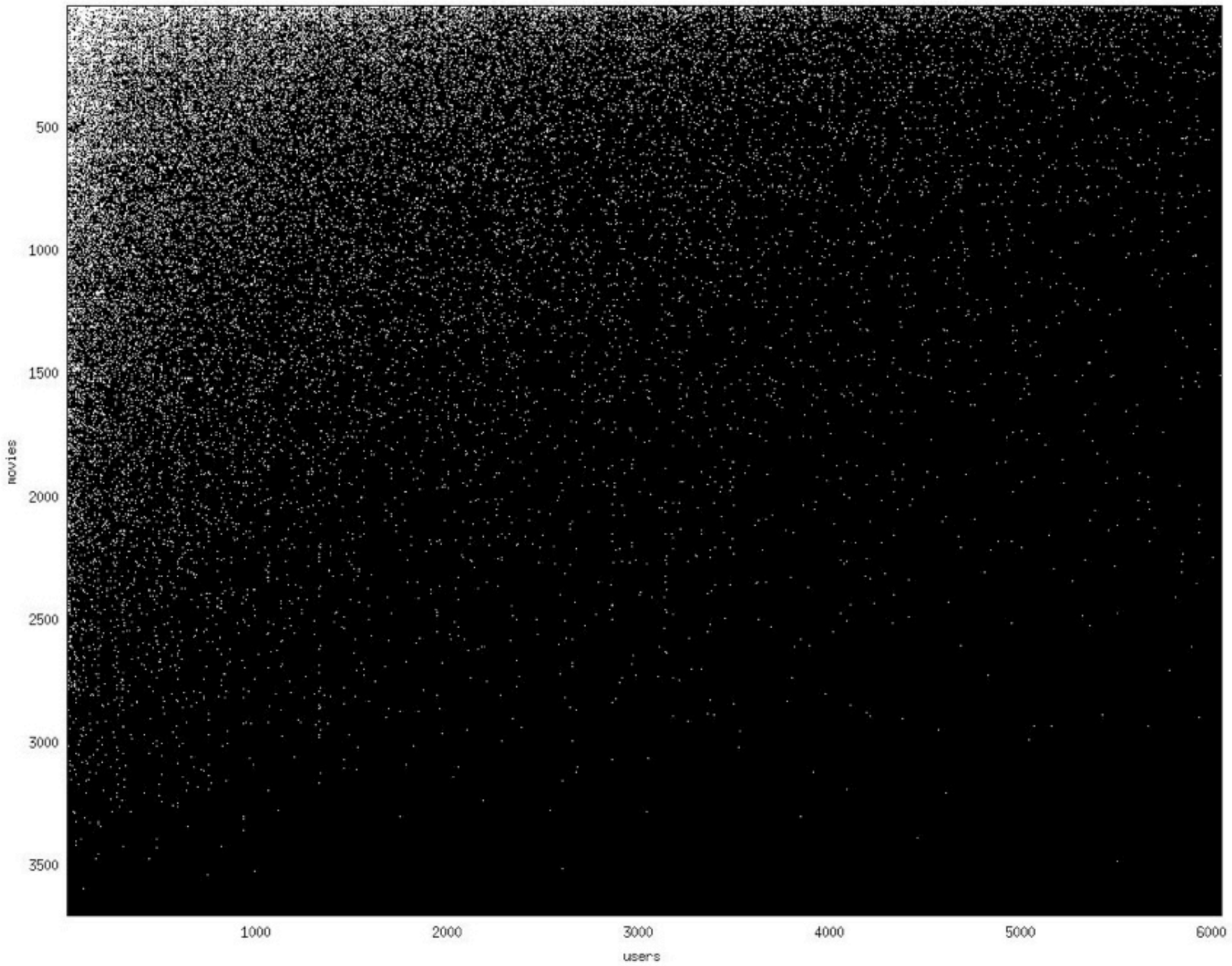
- Is a reduction of RMSE from 0.95 to 0.85 relevant on a scale of 1-5?
  - Why not probability of error?
- Do we need so many non-trivial predictors?
  - Identify a most important subset
  - Provably good principles?

Some Questions

- Is a reduction of RMSE from 0.95 to 0.85 relevant on a scale of 1-5?
  - Why not probability of error?
- Do we need so many non-trivial predictors?
  - Identify a most important subset
  - Provably good principles?
- In this tutorial, we explore these questions
  - Empirical results on Movielens, Netflix
  - Theoretical results for two mathematical models

Rating Matrix Models

Movielens Rating Matrix

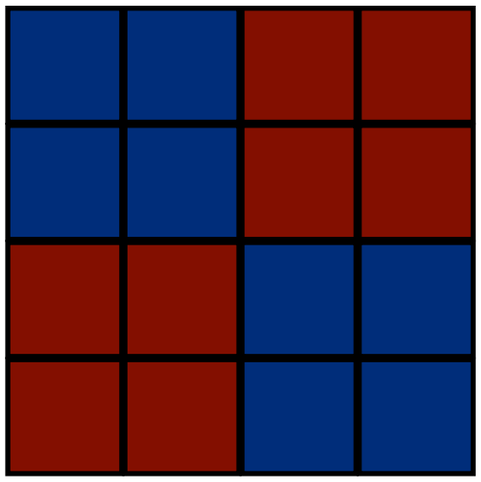


Low-Rank Rating Matrix

2	?	1	3
4	4	?	?
?	3	1	?
1	?	2	?

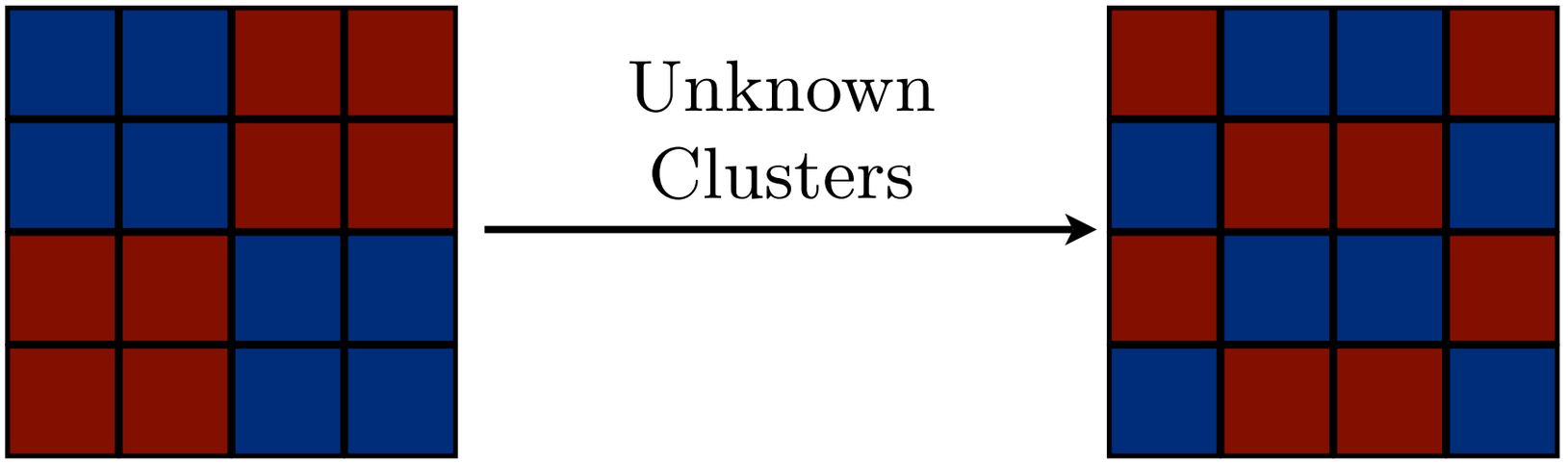
- SVD inspired algorithms
  - Koren el al (2009) and earlier
- Low-rank matrix completion
  - Candes, Recht (2008)
  - Keshavan et al (2009)
  - Neka et al (2009)
  - Lee and Bresler (2009)

Estimation of Rearranged Smooth Processes



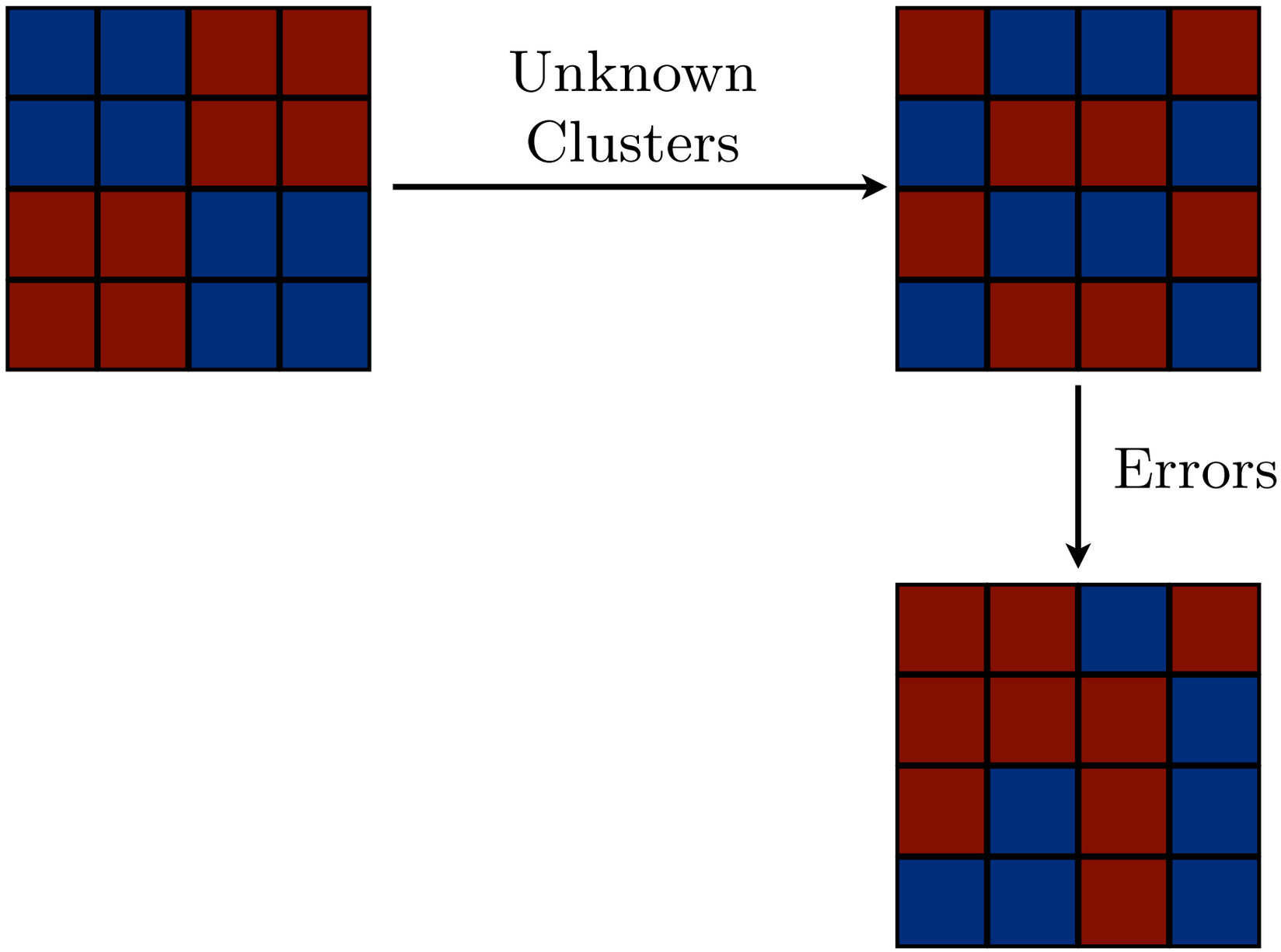
Introduced in Aditya, D, Dey (ISIT 2009)

Estimation of Rearranged Smooth Processes



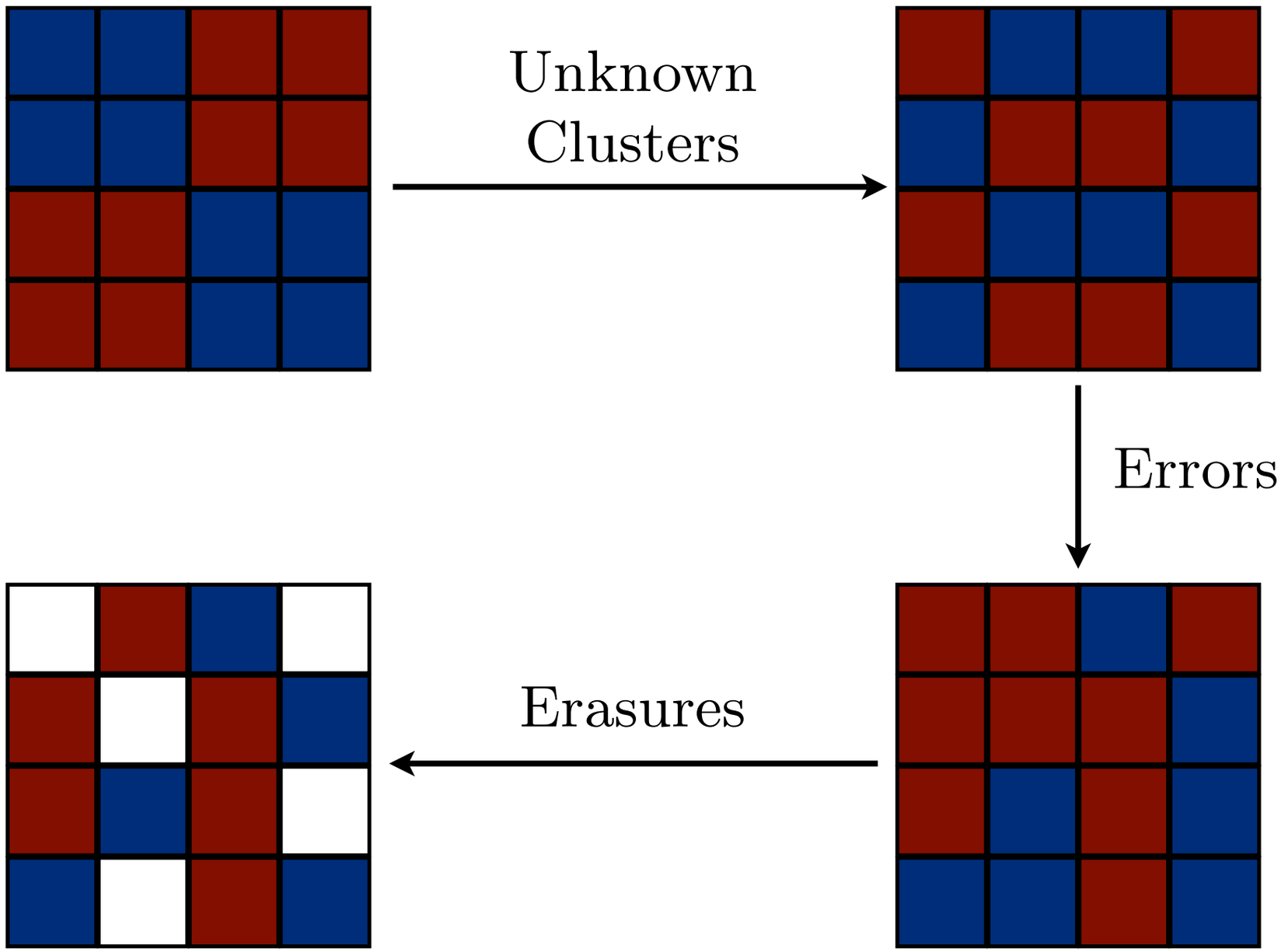
Introduced in Aditya, D, Dey (ISIT 2009)

Estimation of Rearranged Smooth Processes



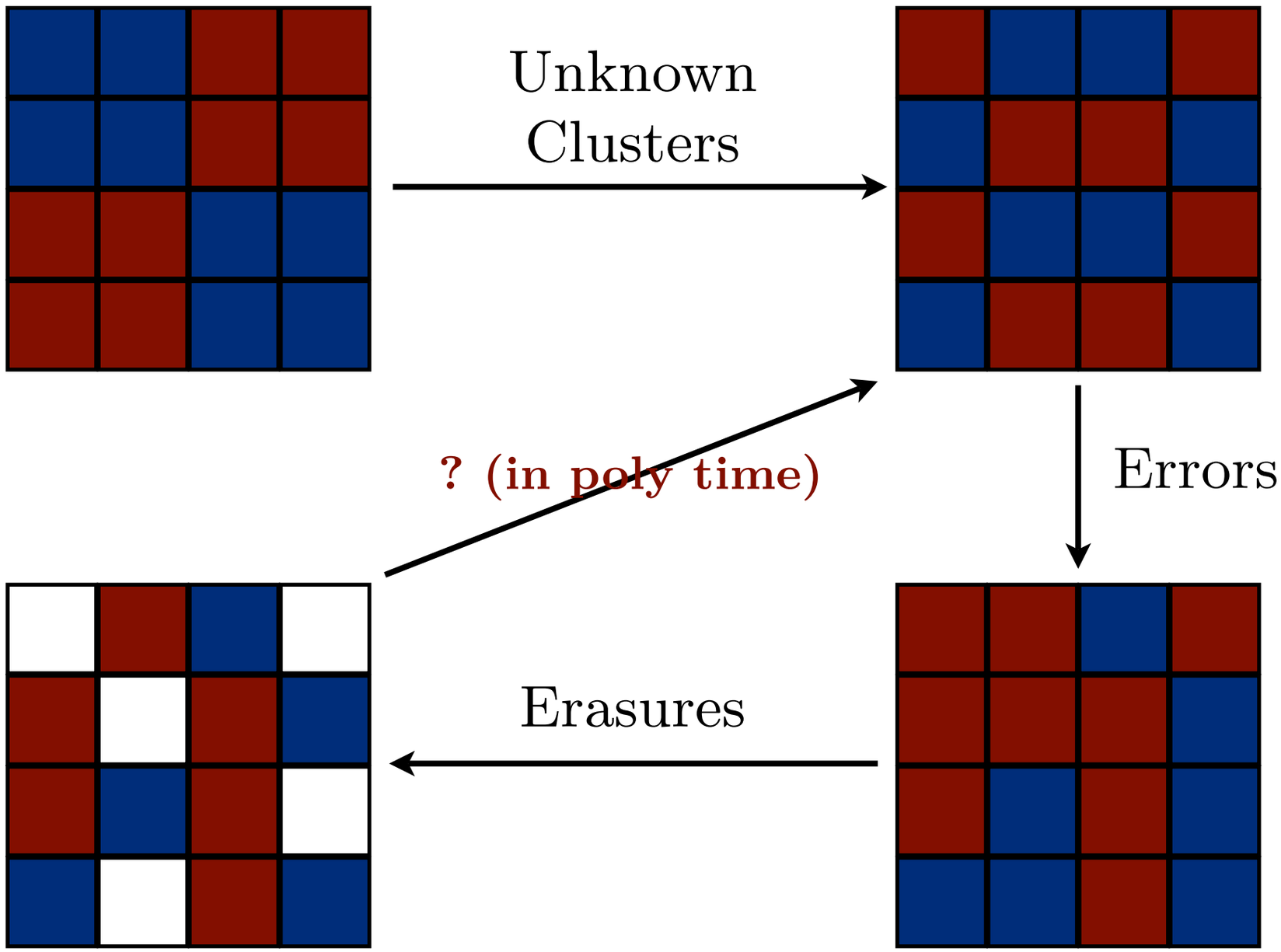
Introduced in Aditya, D, Dey (ISIT 2009)

Estimation of Rearranged Smooth Processes



Introduced in Aditya, D, Dey (ISIT 2009)

Estimation of Rearranged Smooth Processes



Introduced in Aditya, D, Dey (ISIT 2009)

The Relationship: Commonalities

The Relationship: Commonalities

- From data compression:
  - Smoothness corresponds to fast decay of coefficients in a wavelet/DCT expansion
  - Fast decay implies a sparse set of coefficients capture most signal energy

The Relationship: Commonalities

- From data compression:
  - Smoothness corresponds to fast decay of coefficients in a wavelet/DCT expansion
  - Fast decay implies a sparse set of coefficients capture most signal energy
- Hence smoothness implies approximation with low-rank structure

The Relationship: Commonalities

- From data compression:
  - Smoothness corresponds to fast decay of coefficients in a wavelet/DCT expansion
  - Fast decay implies a sparse set of coefficients capture most signal energy
- Hence smoothness implies approximation with low-rank structure
- Row-column permutations preserve rank

The Relationship: Commonalities

- From data compression:
  - Smoothness corresponds to fast decay of coefficients in a wavelet/DCT expansion
  - Fast decay implies a sparse set of coefficients capture most signal energy
- Hence smoothness implies approximation with low-rank structure
- Row-column permutations preserve rank
- Hence a low-rank approximation makes sense

The Relationship: Differences

The Relationship: Differences

- We prefer finite alphabet
  - Real data has finite alphabet
  - Allows consideration of probability of error in a recommendation (more relevant than RMSE?)
  - What is "smoothness"?

The Relationship: Differences

- We prefer finite alphabet
  - Real data has finite alphabet
  - Allows consideration of probability of error in a recommendation (more relevant than RMSE?)
  - What is "smoothness"?
- Lot of noise - not just stability analysis
  - User noise
  - Modeling noise
  - Better describes real data?
  - Keshavan et al (2010) also consider noise in low-rank model

Low-Rank Matrix Completion with In-Coherent Singular Vectors

Spectral Norm Minimization

- $S =$ Set of locations at which X is known
- **The problem:**

$$\min_{Y : Y_{i,j} = X_{i,j}, (i,j) \in S} \operatorname{rank}(Y)$$

- $\operatorname{rank}(Y) = \|\boldsymbol{\sigma}(Y)\|_{\ell_0} = \#$ non-zero singular values
- **Relaxation** (Candes, Recht, 2008):

$$\min_{Y : Y_{i,j} = X_{i,j}, (i,j) \in S} \|\boldsymbol{\sigma}(Y)\|_{\ell_1}$$

  Can be cast as a semi-definite program.

- Poor scalability, but with enough samples, original matrix can be recovered

OptSpace
(Keshavan et al, ISIT 2009)

OptSpace
(Keshavan et al, ISIT 2009)

- Inserting zeros:
  - Replace missing entries by 0
  - If a row/column has 'many' samples, then make it all 0

OptSpace
(Keshavan et al, ISIT 2009)

- Inserting zeros:
  - Replace missing entries by 0
  - If a row/column has 'many' samples, then make it all 0
- Find a low-rank approximation - initialization for next step

OptSpace
(Keshavan et al, ISIT 2009)

- Inserting zeros:
  - Replace missing entries by 0
  - If a row/column has 'many' samples, then make it all 0
- Find a low-rank approximation - initialization for next step
- Insert back pruned rows/columns

OptSpace
(Keshavan et al, ISIT 2009)

- Inserting zeros:
  - Replace missing entries by 0
  - If a row/column has 'many' samples, then make it all 0
- Find a low-rank approximation - initialization for next step
- Insert back pruned rows/columns
- Consider error restricted to known entries and apply gradient descent

OptSpace
(Keshavan et al, ISIT 2009)

- Inserting zeros:
  - Replace missing entries by 0
  - If a row/column has 'many' samples, then make it all 0
- Find a low-rank approximation - initialization for next step
- Insert back pruned rows/columns
- Consider error restricted to known entries and apply gradient descent
- Faster than Candes-Recht with similar theoretical guarantee (and MSE bound)

Other Related Works

Other Related Works

- Koren, Bell, Volinsky (IEEE Computer, 2009)
  - No theoretical guarantee

Other Related Works

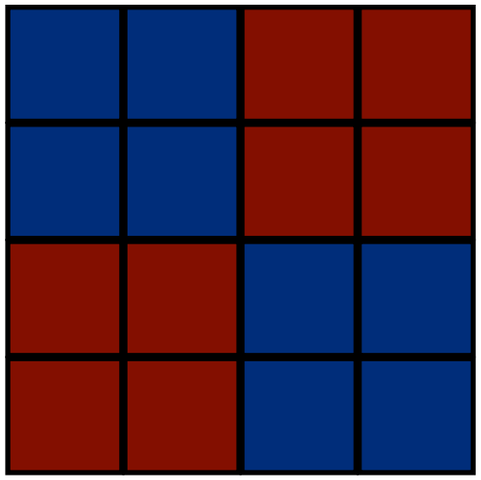
- Koren, Bell, Volinsky (IEEE Computer, 2009)
  - No theoretical guarantee
- ISIT 2009, 2010 special sessions

Other Related Works

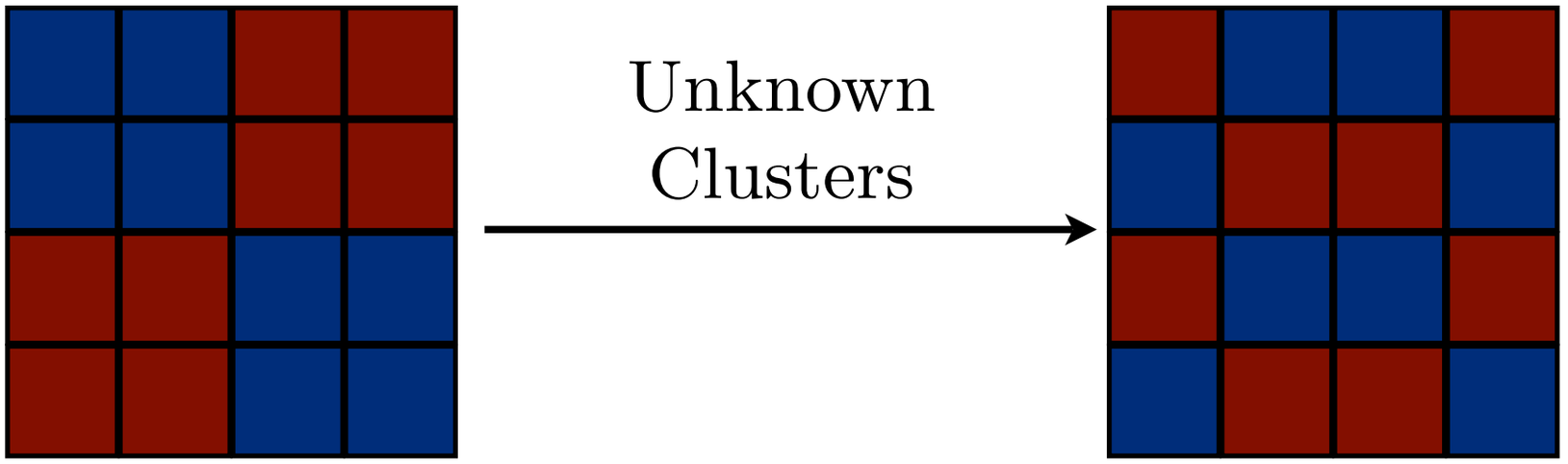
- Koren, Bell, Volinsky (IEEE Computer, 2009)
  - No theoretical guarantee
- ISIT 2009, 2010 special sessions
- For example: Lee and Bresler, ISIT 2009
  - Linear measurements
  - Close link to matching pursuits (best sparse representation of signals)
  - No theoretical guarantee

Estimating Rearranged 'Smooth' Processes

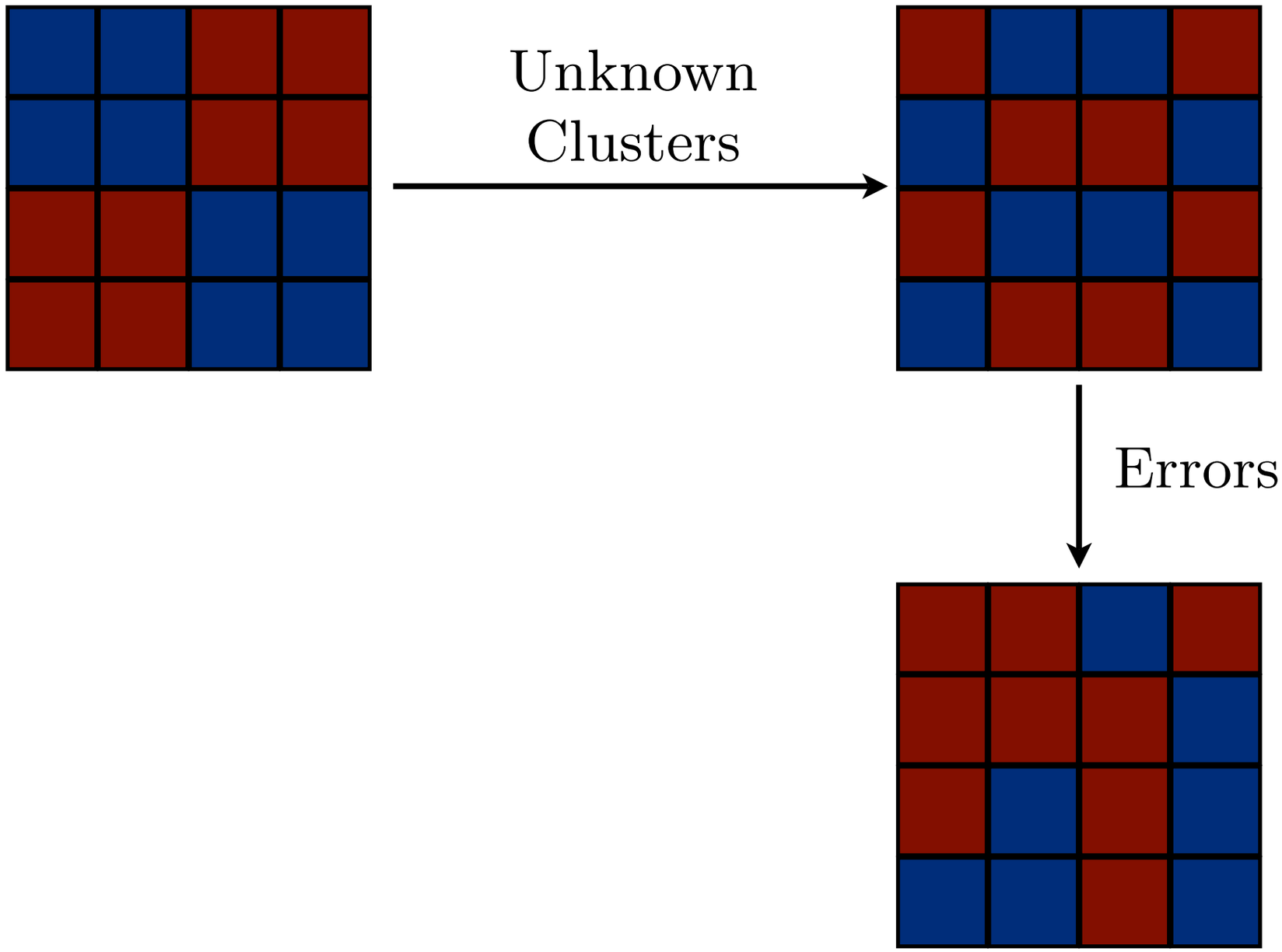
Data Model



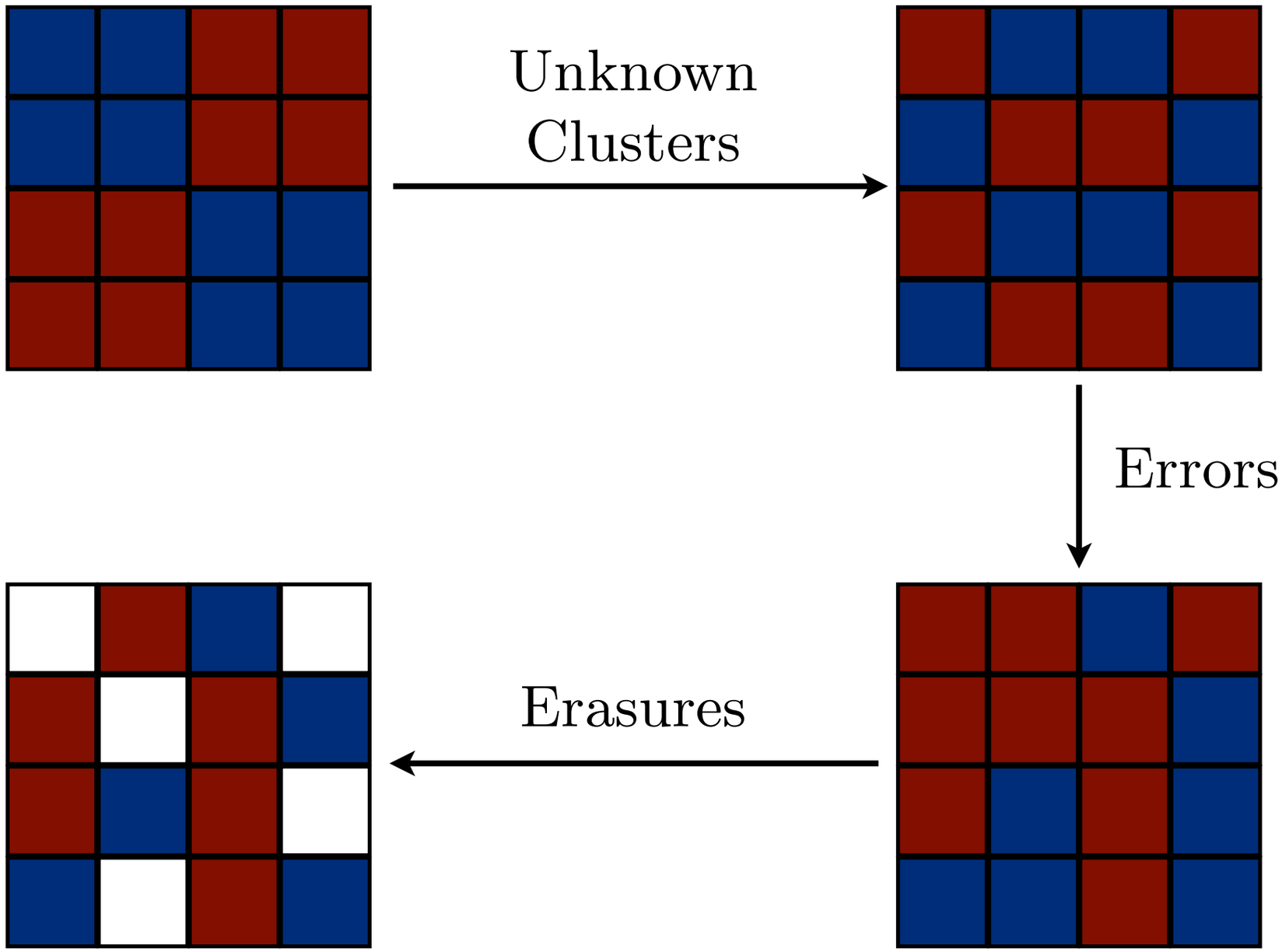
Data Model



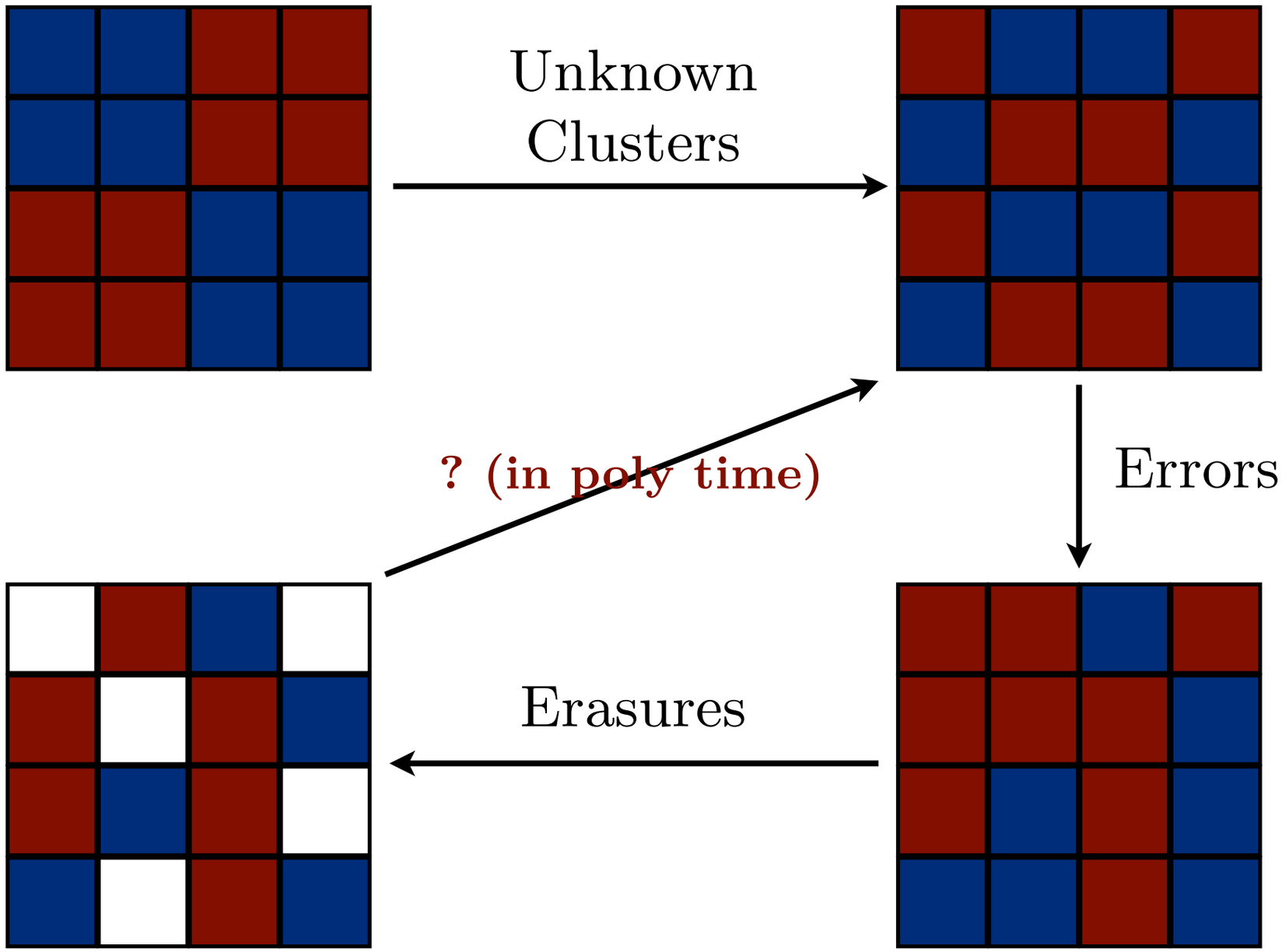
Data Model



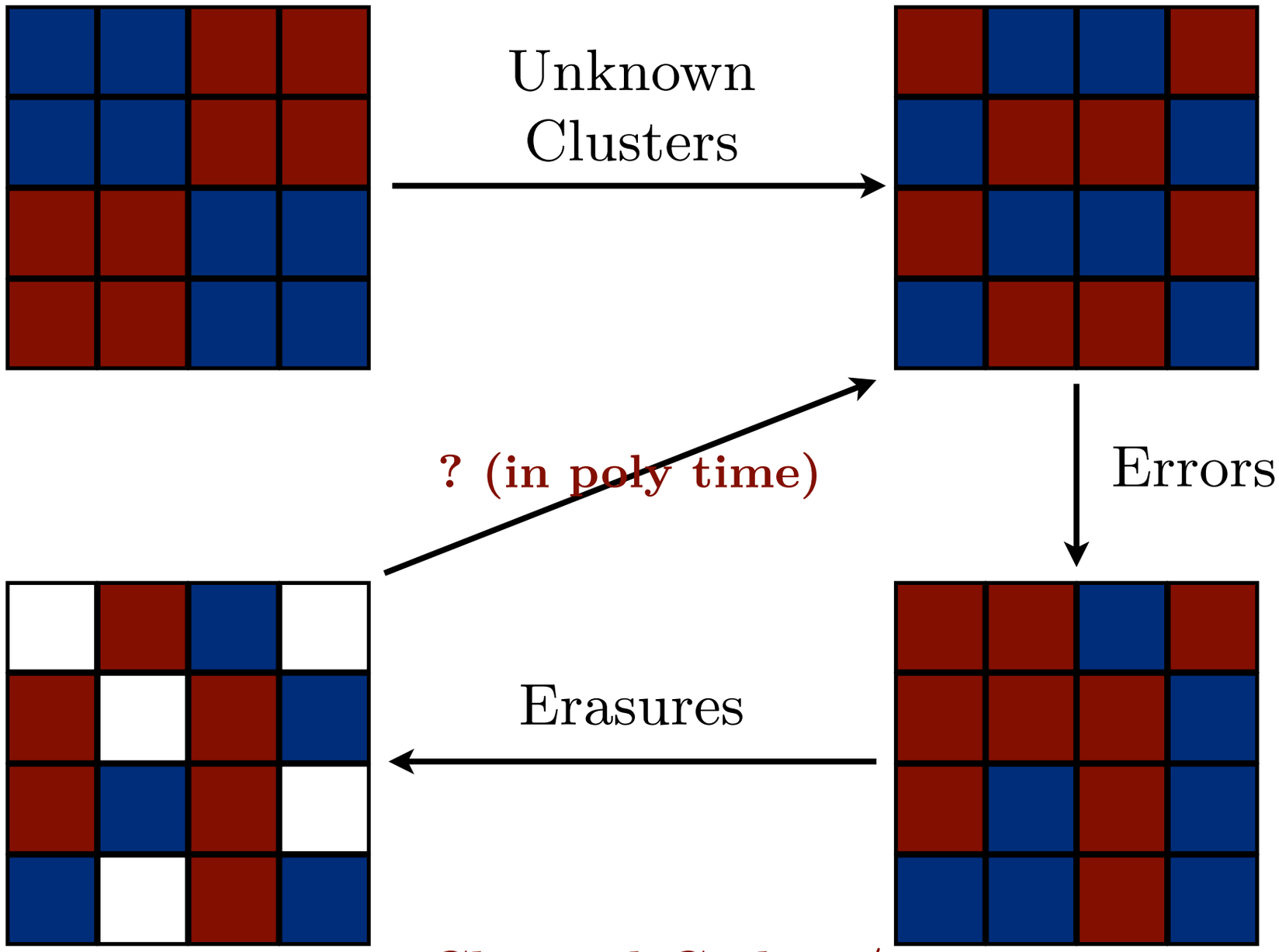
Data Model



Data Model



Data Model



Channel Coding/
Estimation of rearranged, ‘smooth’ process,
under noise and erasures

Relation with Low-Rank Literature

Relation with Low-Rank Literature

- Underlying true matrix is low-rank
  - Recall link between 'smoothness' and low-rank

Relation with Low-Rank Literature

- Underlying true matrix is low-rank
  - Recall link between 'smoothness' and low-rank
- Incorporates user and modeling noise
  - Candes and Plan (2009) consider small noise for stability analysis

Relation with Low-Rank Literature

- Underlying true matrix is low-rank
  - Recall link between 'smoothness' and low-rank
- Incorporates user and modeling noise
  - Candes and Plan (2009) consider small noise for stability analysis
- Finite alphabet
  - Hence asymptotic error free recovery possible even in presence of noise
  - Probability of error in recovering entire matrix
  - Bit error rate (BER)

The Model in Words

The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom

The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic

The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
- **Errors:** i.i.d., binary symmetric channel, represent noisy user behavior

The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
- **Errors:** i.i.d., binary symmetric channel, represent noisy user behavior
- **Erasures:** i.i.d., model missing data

The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
- **Errors:** i.i.d., binary symmetric channel, represent noisy user behavior
- **Erasures:** i.i.d., model missing data
- **Diverse opinions:** i.i.d. Bernoulli(1/2) ratings across clusters
  - No information from self data; must use collaborative filtering

Some Assumptions

- Matrix: $m \times n$, $m = \Theta(n)$
- Erasure probability

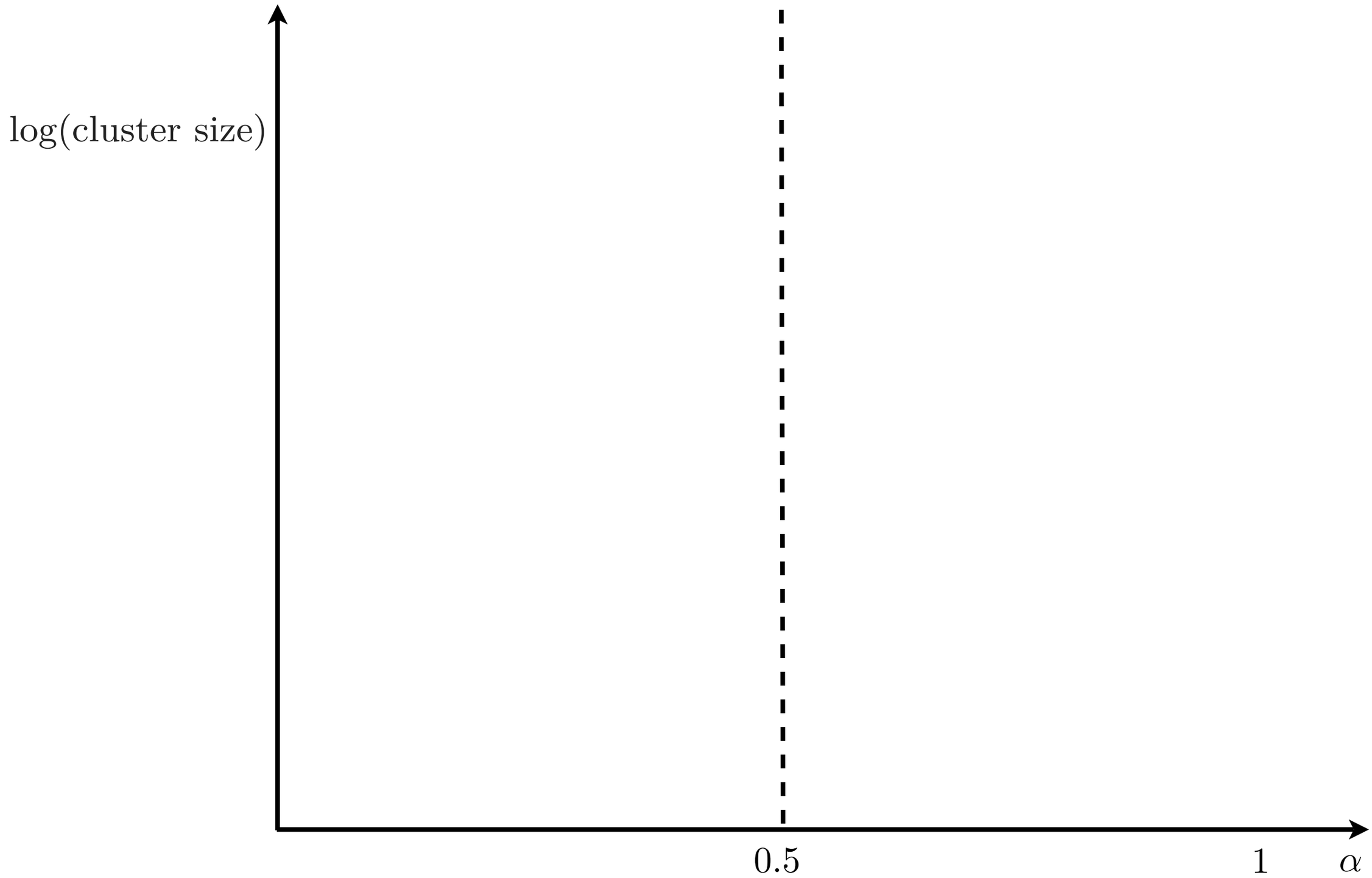
$$\epsilon = 1 - \frac{c}{n^{\alpha}}, \quad 0 \leq \alpha \leq 1$$

  $\alpha < 1/2$: Near-quadratic regime
  $\alpha > 1/2$: Near-linear regime

- All clusters of same order
  - Number of clusters $= \Omega(\log n)$ to ensure P(cluster merging) is vanishing

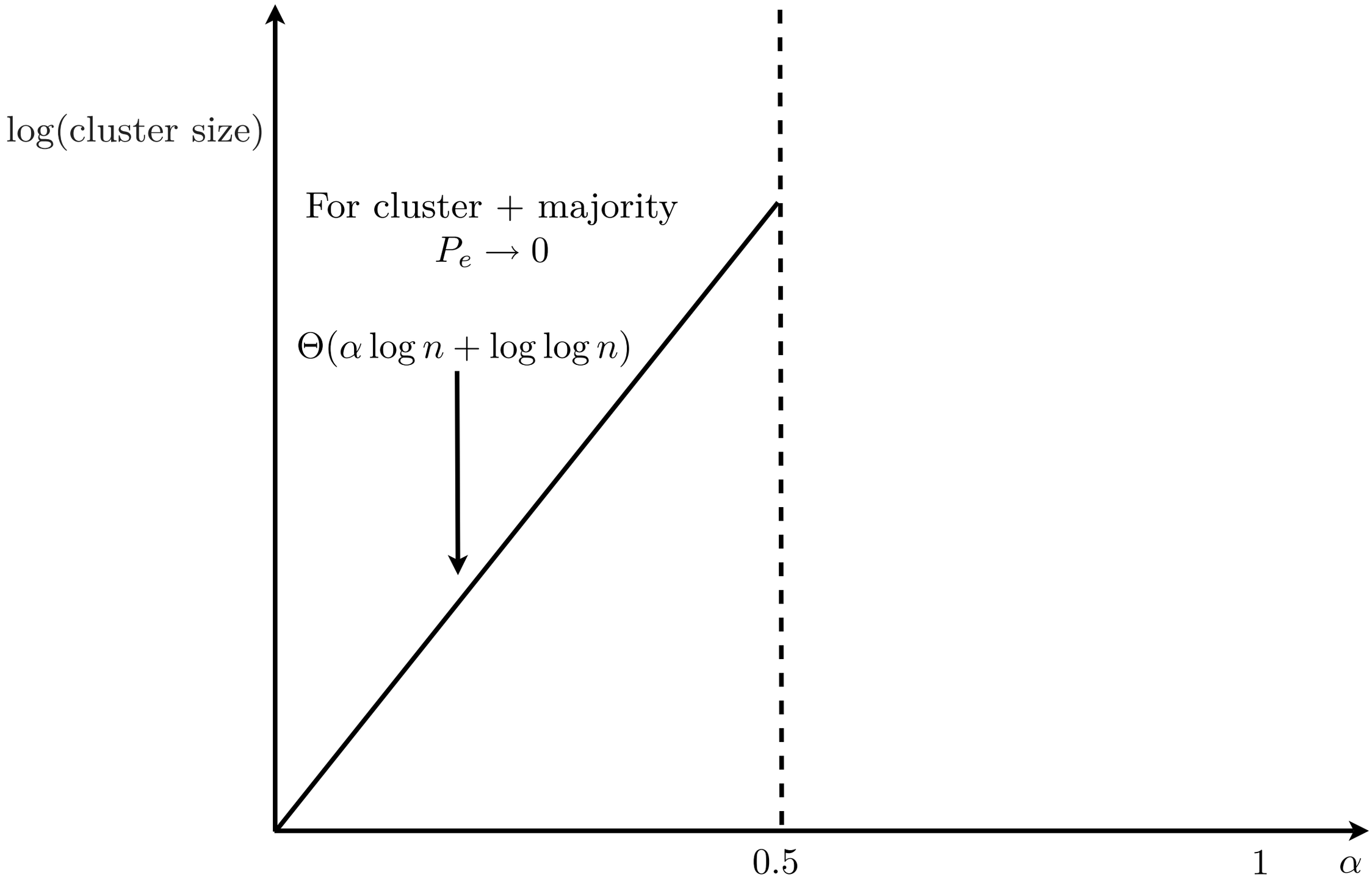
Recover Entire Matrix

(Aditya, Dabeer, Dey, ISIT 2009; to appear IEEE Trans. Inform. Theory)



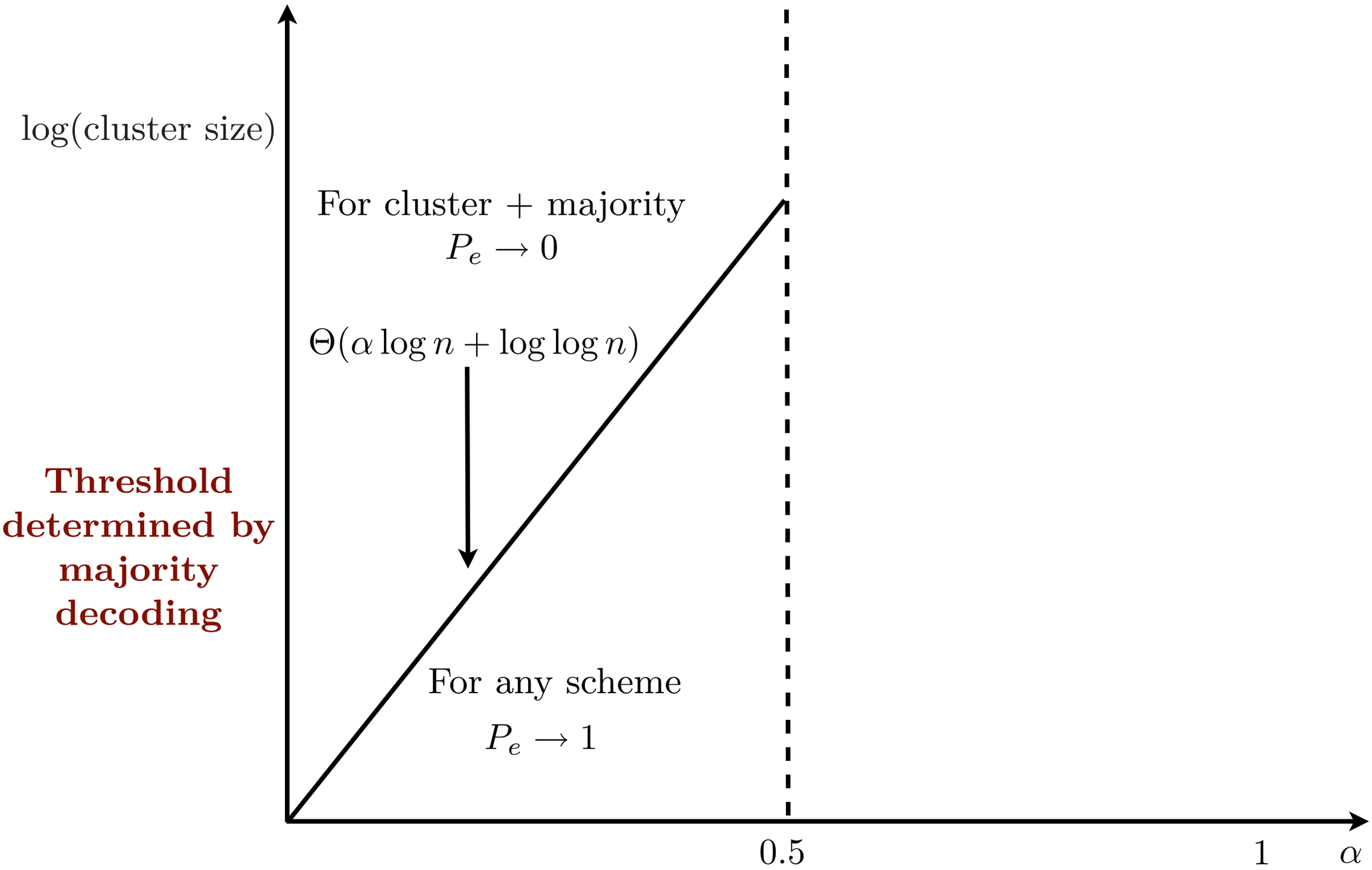
Recover Entire Matrix

(Aditya, Dabeer, Dey, ISIT 2009; to appear IEEE Trans. Inform. Theory)



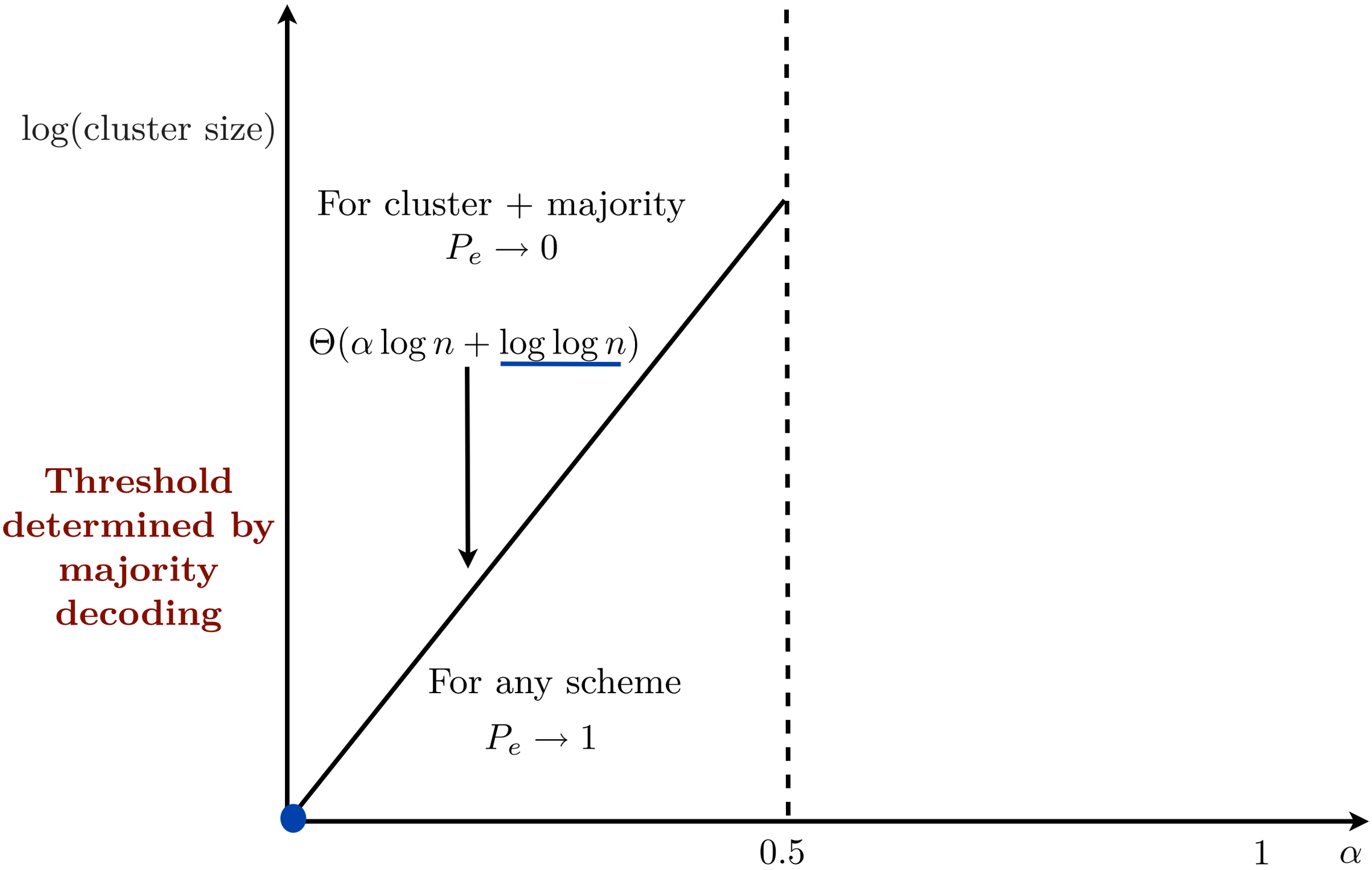
Recover Entire Matrix

(Aditya, Dabeer, Dey, ISIT 2009; to appear IEEE Trans. Inform. Theory)



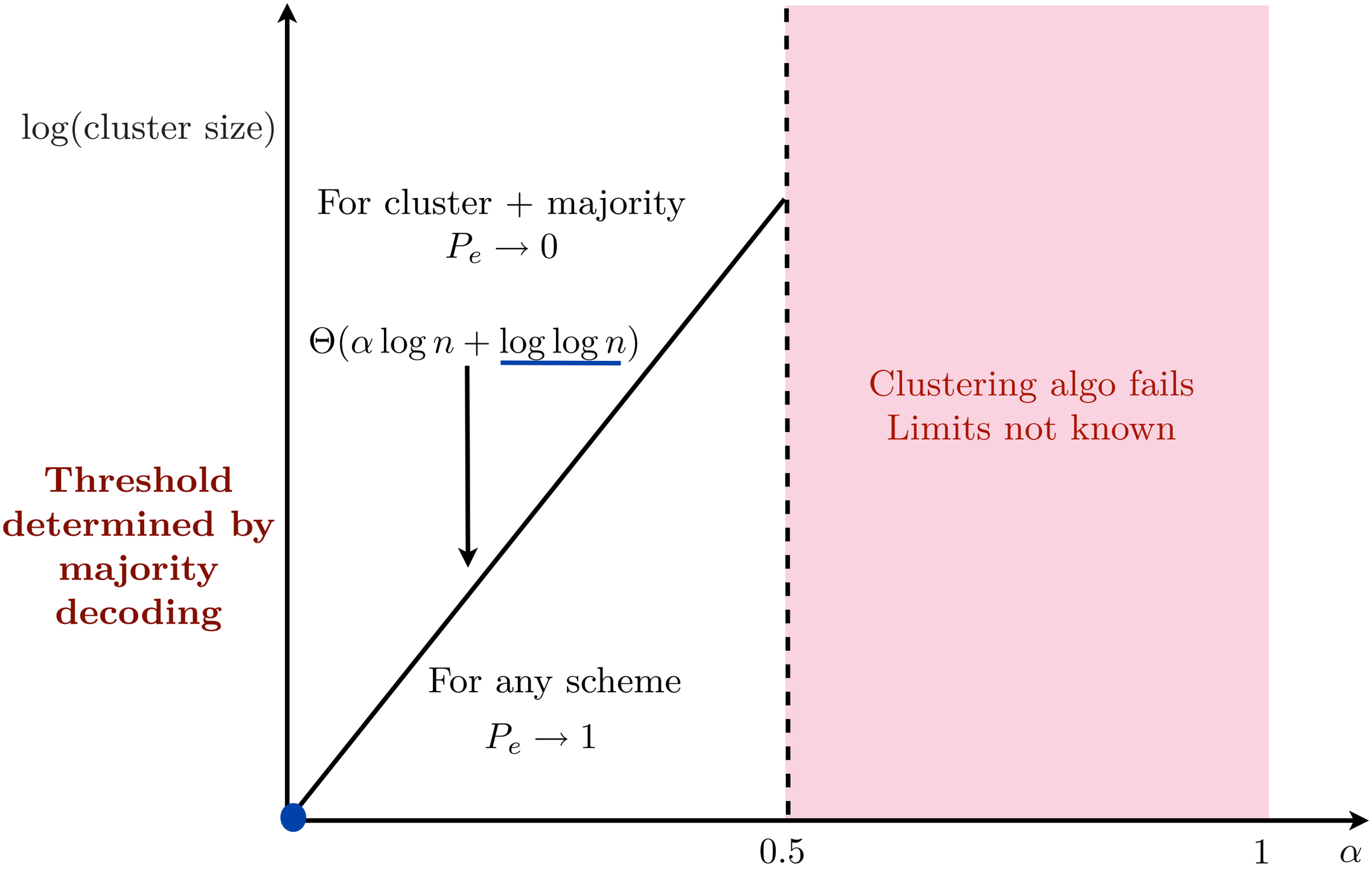
Recover Entire Matrix

(Aditya, Dabeer, Dey, ISIT 2009; to appear IEEE Trans. Inform. Theory)



Recover Entire Matrix

(Aditya, Dabeer, Dey, ISIT 2009; to appear IEEE Trans. Inform. Theory)



Implementation on Real Data

Implementation on Real Data

- Our clustering algo:
  - Similarity = #commonly agreed entries/# common entries
  - Threshold similarities

Implementation on Real Data

- Our clustering algo:
  - Similarity = #commonly agreed entries/# common entries
  - Threshold similarities
- Good for analysis

Implementation on Real Data

- Our clustering algo:
  - Similarity = #commonly agreed entries/# common entries
  - Threshold similarities
- Good for analysis
- But bad for implementation as may not lead to clusters on finite data

Implementation on Real Data

- Our clustering algo:
  - Similarity = #commonly agreed entries/# common entries
  - Threshold similarities
- Good for analysis
- But bad for implementation as may not lead to clusters on finite data
- Need a modification

Implementation on Real Data

- Our clustering algo:
  - Similarity = #commonly agreed entries/# common entries
  - Threshold similarities
- Good for analysis
- But bad for implementation as may not lead to clusters on finite data
- Need a modification
- Also clustering both rows and columns is computationally intensive

The Popularity Amongst Friends (PAF) Algorithm

PAF for Binary Ratings

PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users

PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users
- Motivated by practice (example: Amazon)

PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users
- Motivated by practice (example: Amazon)
- Not matrix completion

PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users
- Motivated by practice (example: Amazon)
- Not matrix completion
- Low complexity
  - User node degree << Total number of users
  - Simple updates

Performance Metrics

Performance Metrics

- RMSE - popular since Netflix prize

Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works

Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works
- Probability that entire matrix is recovered
  - Candes, Recht (2008) and others
  - Aditya, D, Dey (2009)

Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works
- Probability that entire matrix is recovered
  - Candes, Recht (2008) and others
  - Aditya, D, Dey (2009)
- Our focus is on bit error rate (BER)
  - Probability that a recommendation made is incorrect

Empirical Performance (Movielens and Netflix)

Setup

Setup

- Rating quantization
  - 4,5 mapped to 1 (yes), 1-3 mapped to 0 (no)

Setup

- Rating quantization
  - 4,5 mapped to 1 (yes), 1-3 mapped to 0 (no)
- Hide 30% of data per user; can compute metrics only when recommended item is in the hidden list

Setup

- Rating quantization
  - 4,5 mapped to 1 (yes), 1-3 mapped to 0 (no)
- Hide 30% of data per user; can compute metrics only when recommended item is in the hidden list
- Comparison with OptSpace (Keshavan et al, 2008)
  - Representative of matrix algorithms
  - Evaluated only on items recommended by local algorithm
  - Unquantized input; output quantized for BER

Empirical Performance

Movielens

	BER	MAE	RMSE
PAF	0.103	0.627	0.748
OptSpace	0.108	0.581	0.733

Naive Estimate:
For Local Algorithm,
to compute
RMSE, MAE
1 is mapped to 4.5

Empirical Performance

Movielens

	BER	MAE	RMSE
PAF	0.103	0.627	0.748
OptSpace	0.108	0.581	0.733

Naive Estimate:
For Local Algorithm,
to compute
RMSE, MAE
1 is mapped to 4.5

Snapshot of Netflix (2000)

	BER	MAE	RMSE
PAF	0.18	0.742	0.942
OptSpace	0.19	0.590	0.742

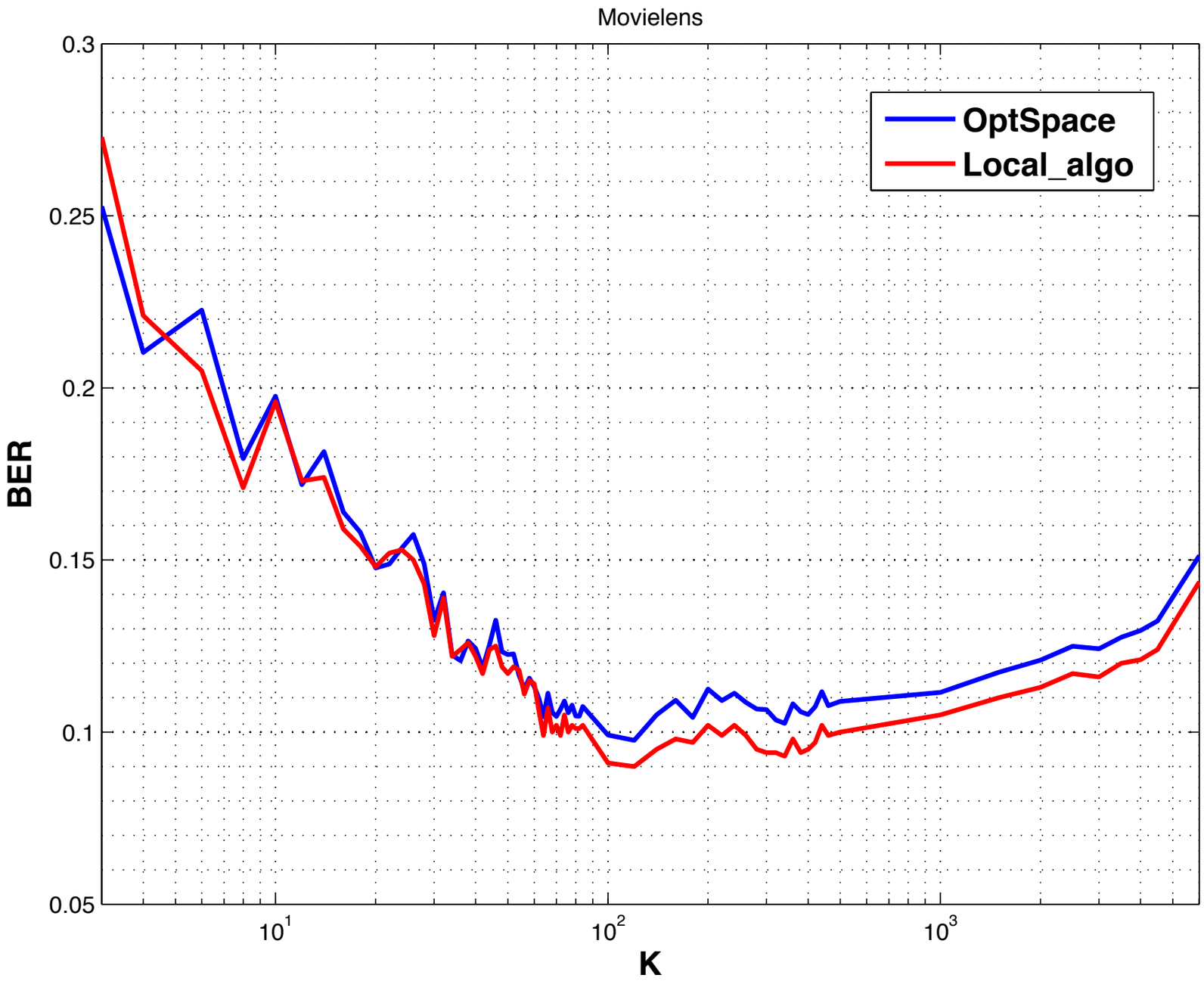
Netflix has higher
percentage of
low ratings

Empirical Performance (Contd.)

Movielens After Removing Popular Movies
(those with 60% or more 1's)

	BER	MAE	RMSE
PAF	0.335	0.709	1.010
OptSpace	0.327	0.718	0.901

Empirical Performance (Contd.)



Remarks

Remarks

- PAF competitive for BER

Remarks

- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information

Remarks

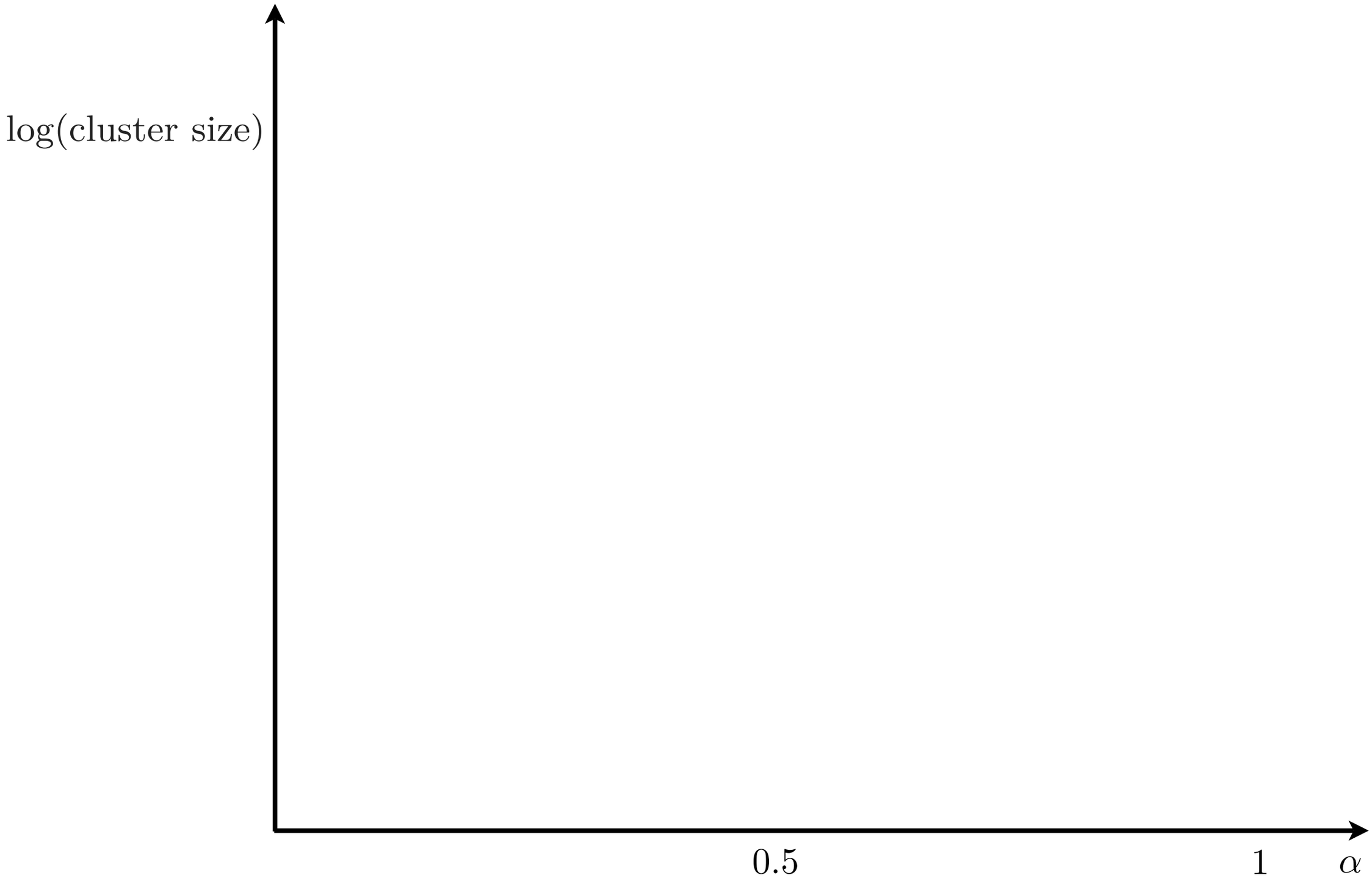
- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information
- 2-10 times faster than OptSpace
  - Also amenable to recursive update

Remarks

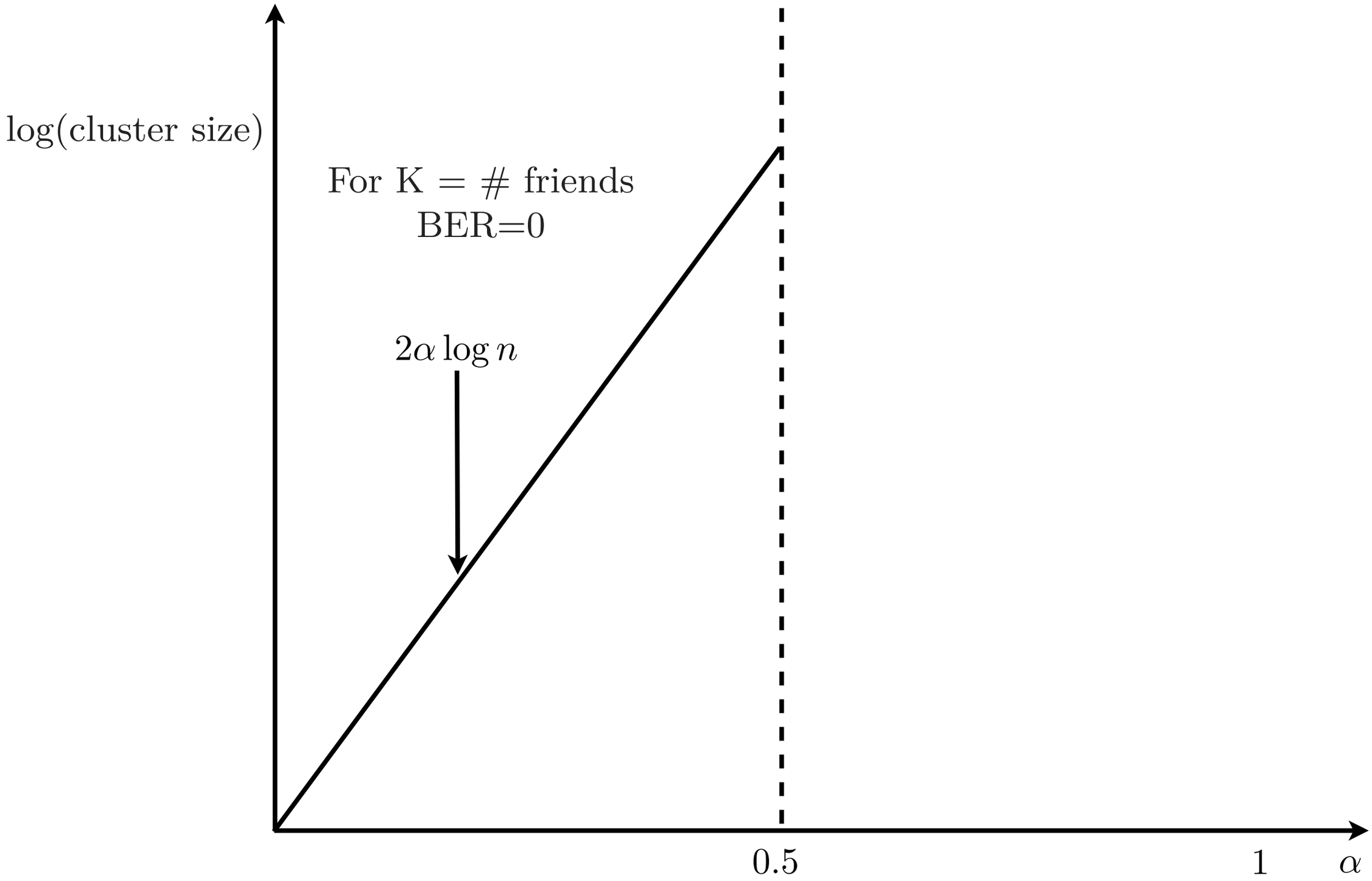
- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information
- 2-10 times faster than OptSpace
  - Also amenable to recursive update
- Any provably good properties?

BER Analysis of PAF

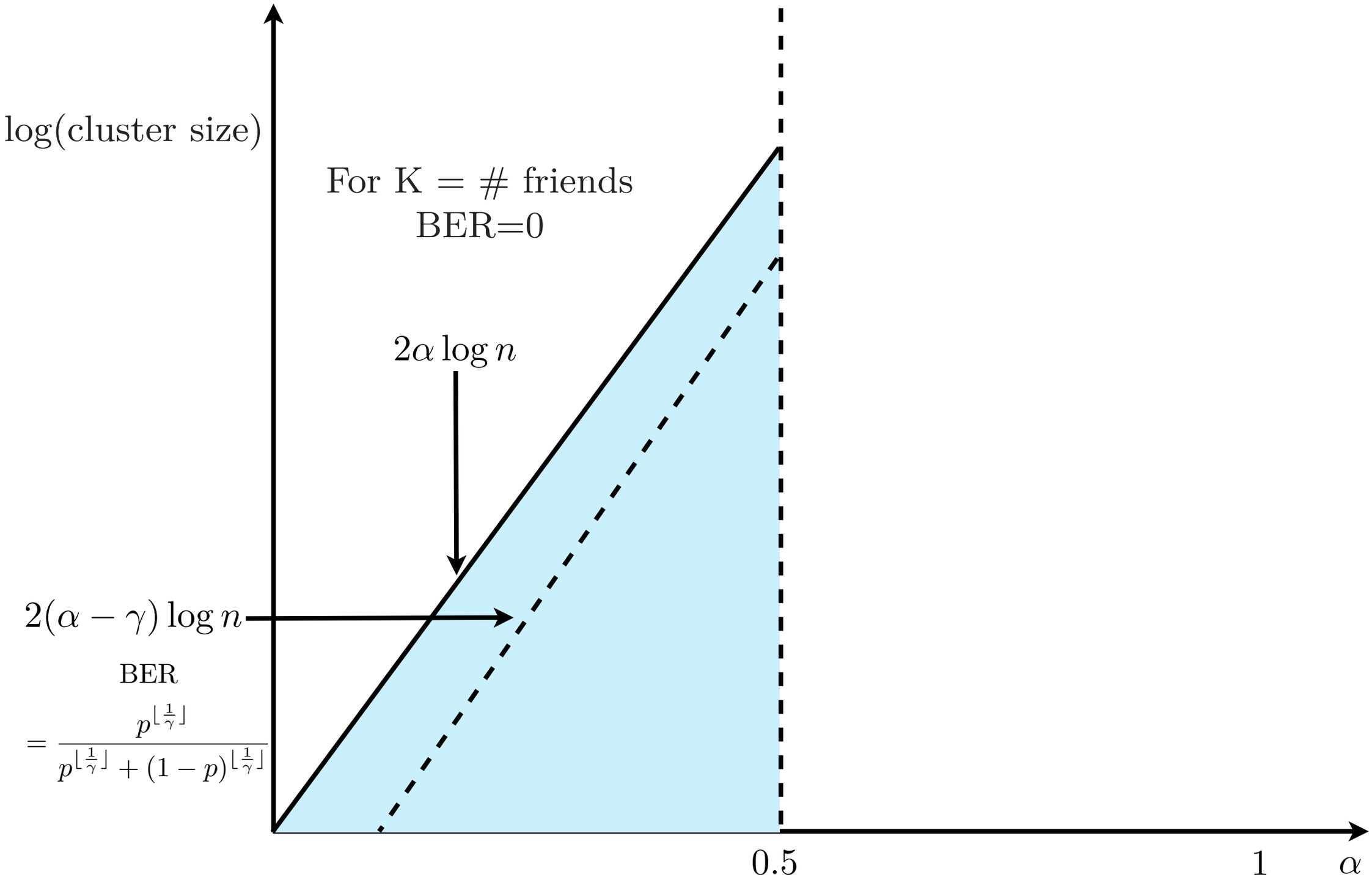
Asymptotic BER of PAF



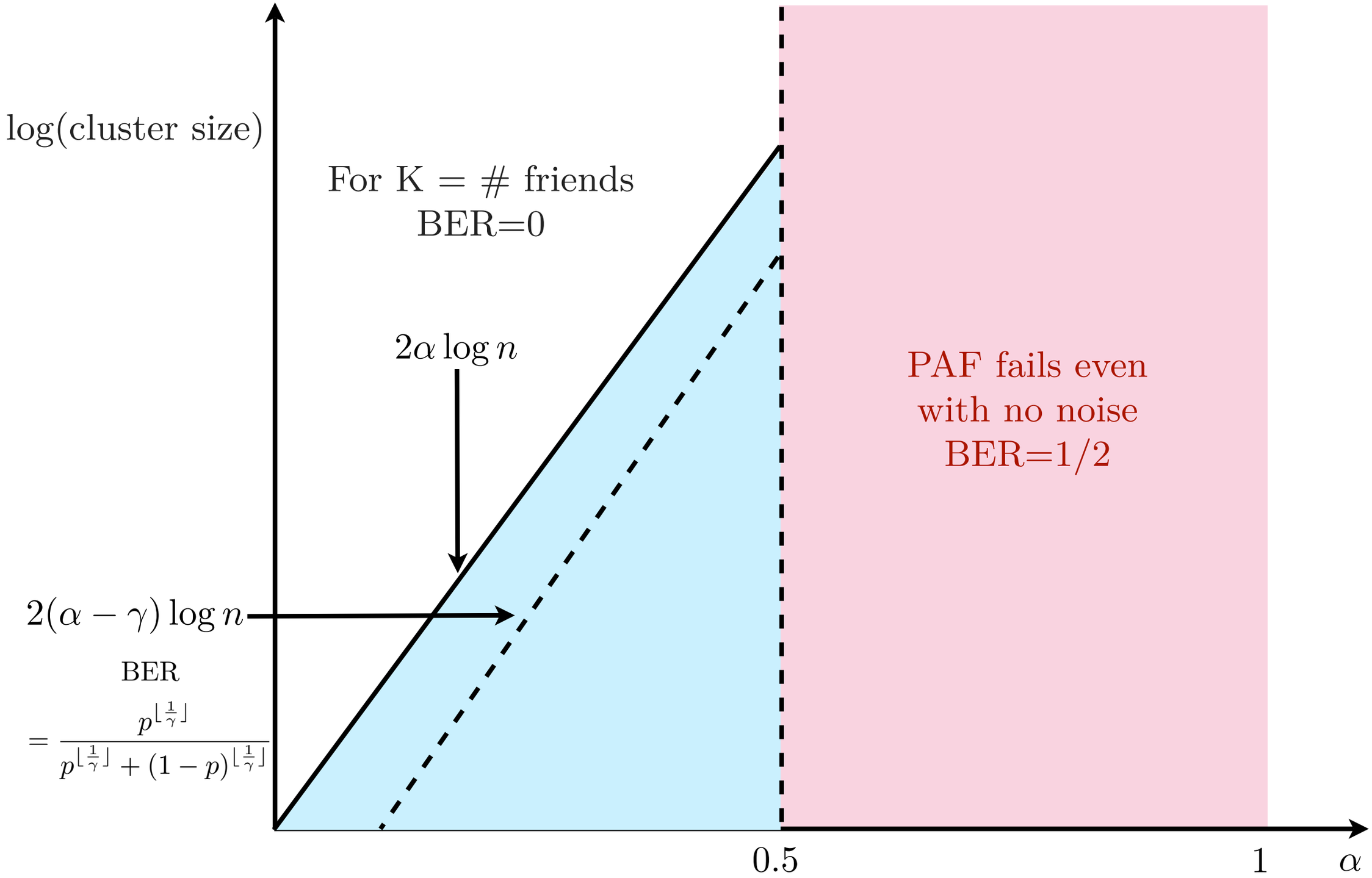
Asymptotic BER of PAF



Asymptotic BER of PAF



Asymptotic BER of PAF



Three Phases

Three Phases

- **Phase 1:** Large cluster, near-quadratic samples, BER=0
  - Top neighbors good, large cluster averages out noise

Three Phases

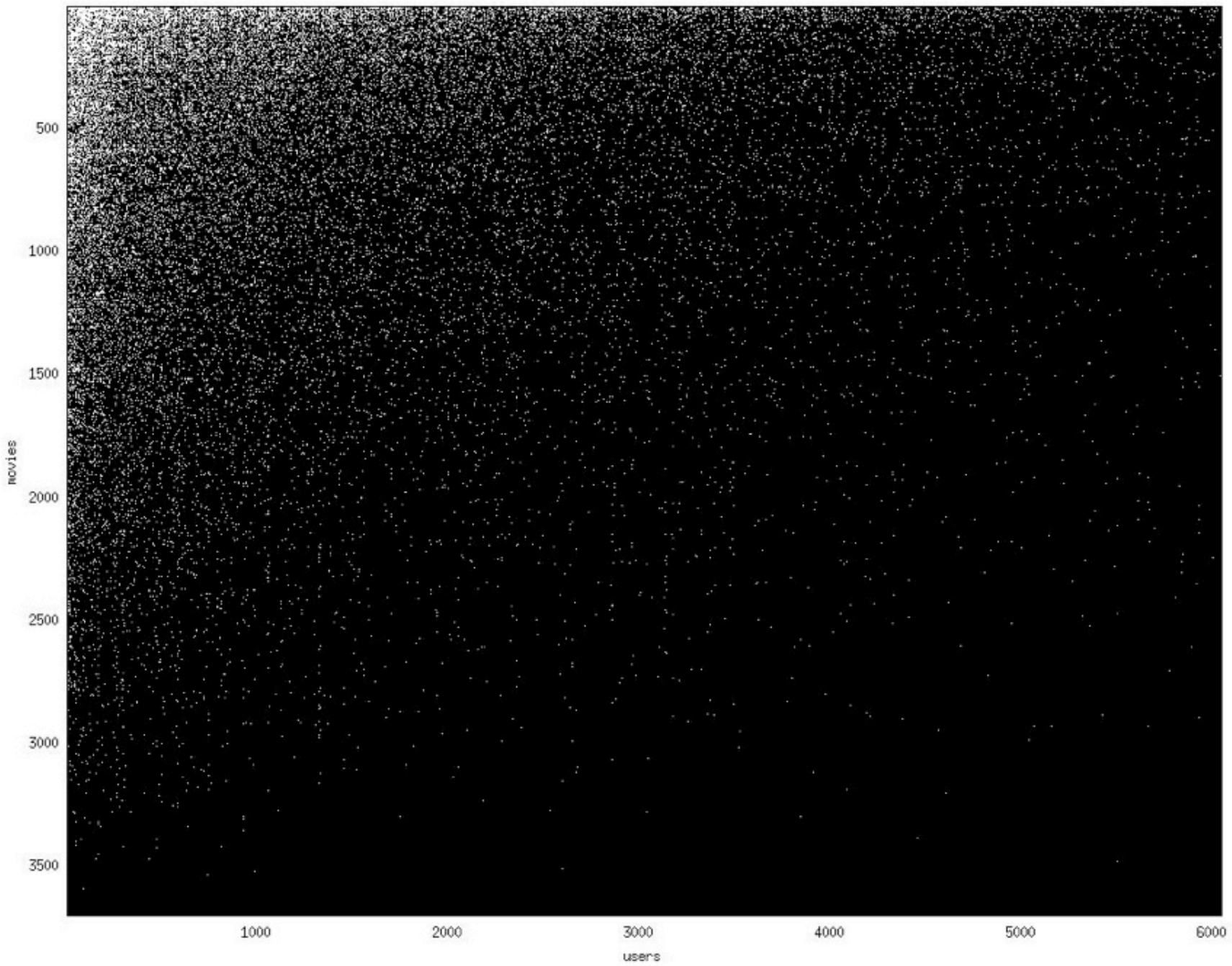
- **Phase 1:** Large cluster, near-quadratic samples, BER=0
  - Top neighbors good, large cluster averages out noise
- **Phase 2:** Small cluster, near-quadratic samples, $0 < \text{BER} < 1/2$
  - Top neighbors good
  - But cluster too small to average out noise
  - Optimum list size = # friends

Three Phases

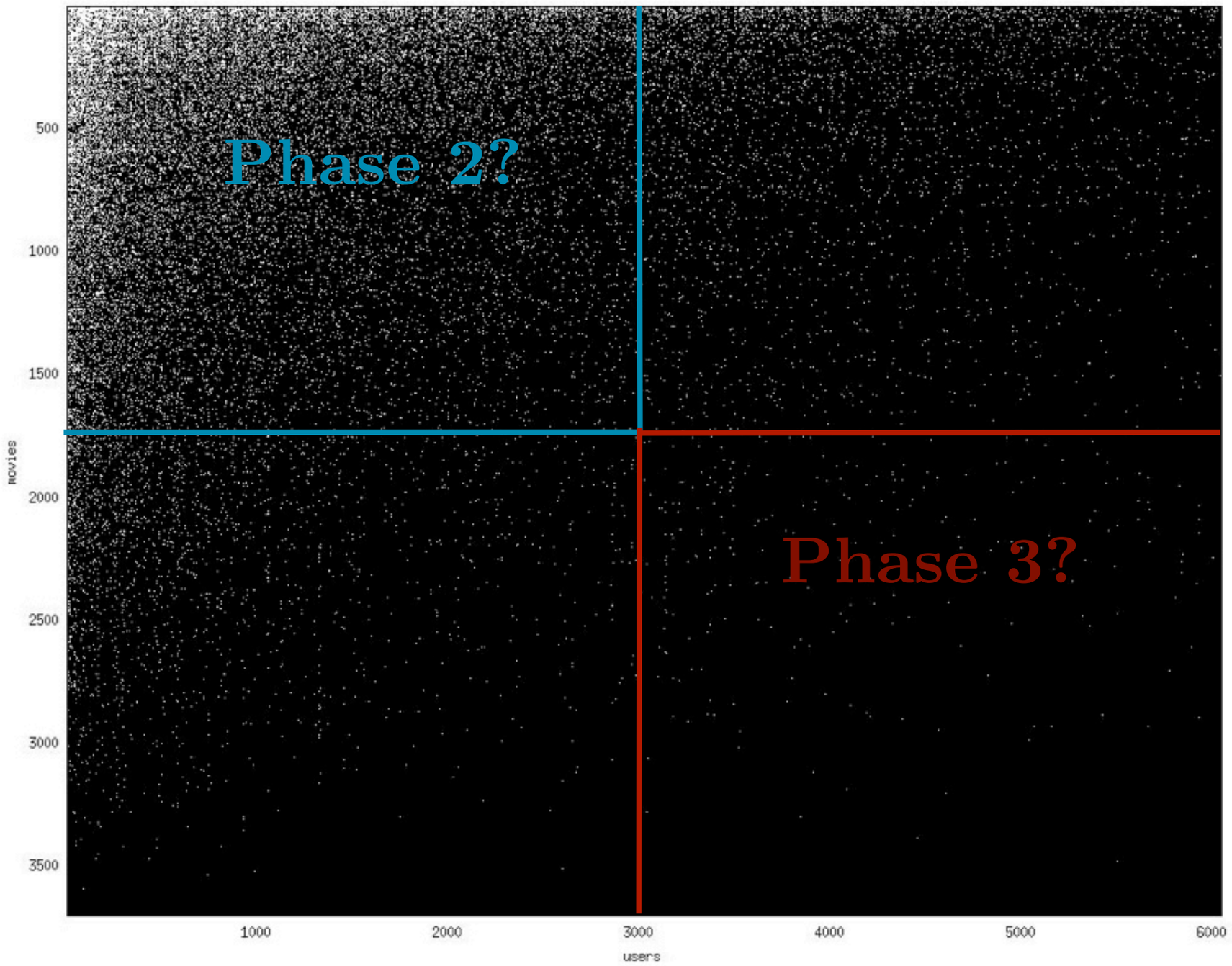
- **Phase 1:** Large cluster, near-quadratic samples, BER=0
  - Top neighbors good, large cluster averages out noise
- **Phase 2:** Small cluster, near-quadratic samples, $0 < \text{BER} < 1/2$
  - Top neighbors good
  - But cluster too small to average out noise
  - Optimum list size = # friends
- **Phase 3:** Near-linear samples
  - Most neighbors picked are bad

Where Does Real Data Live?

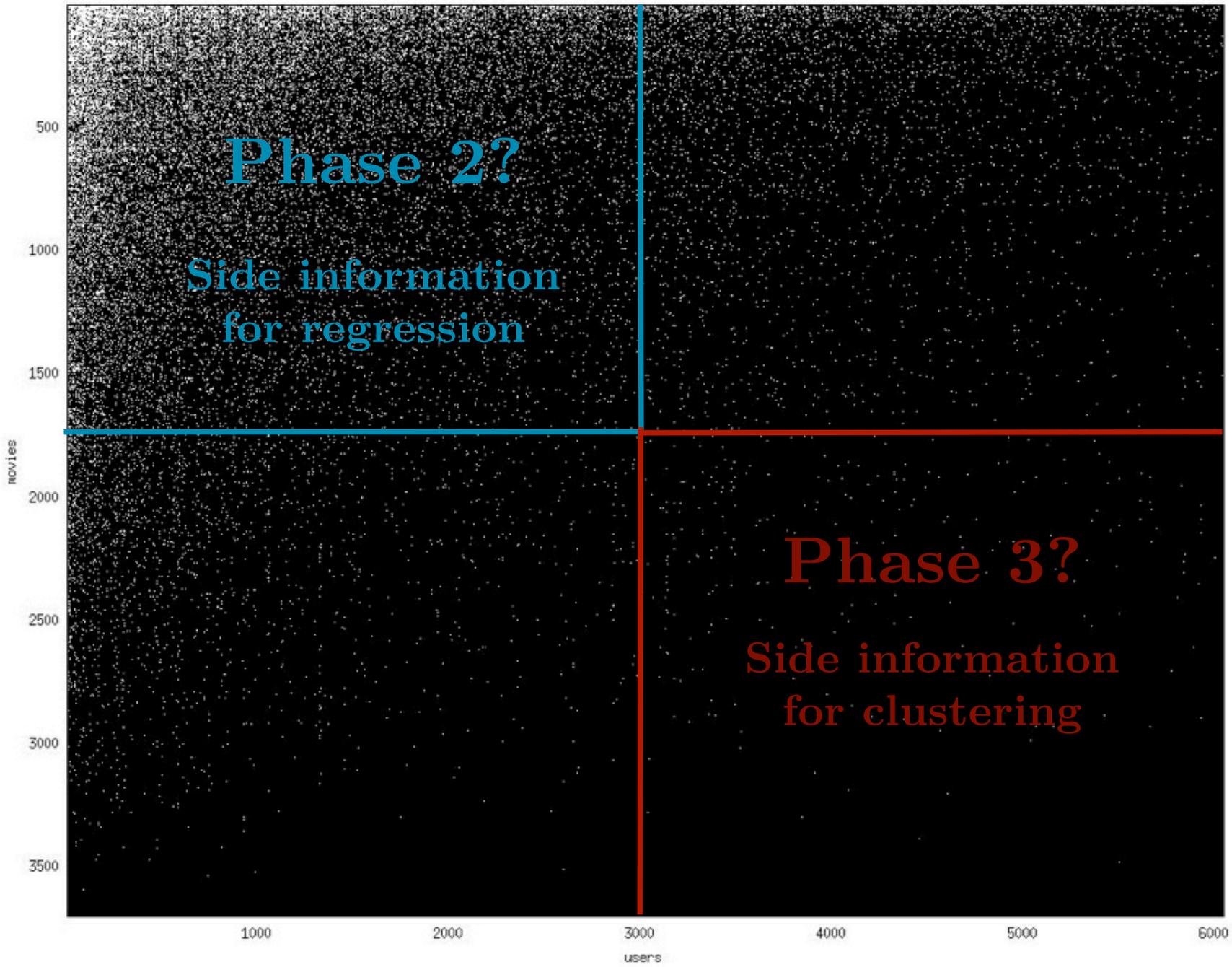
The Movielens Matrix



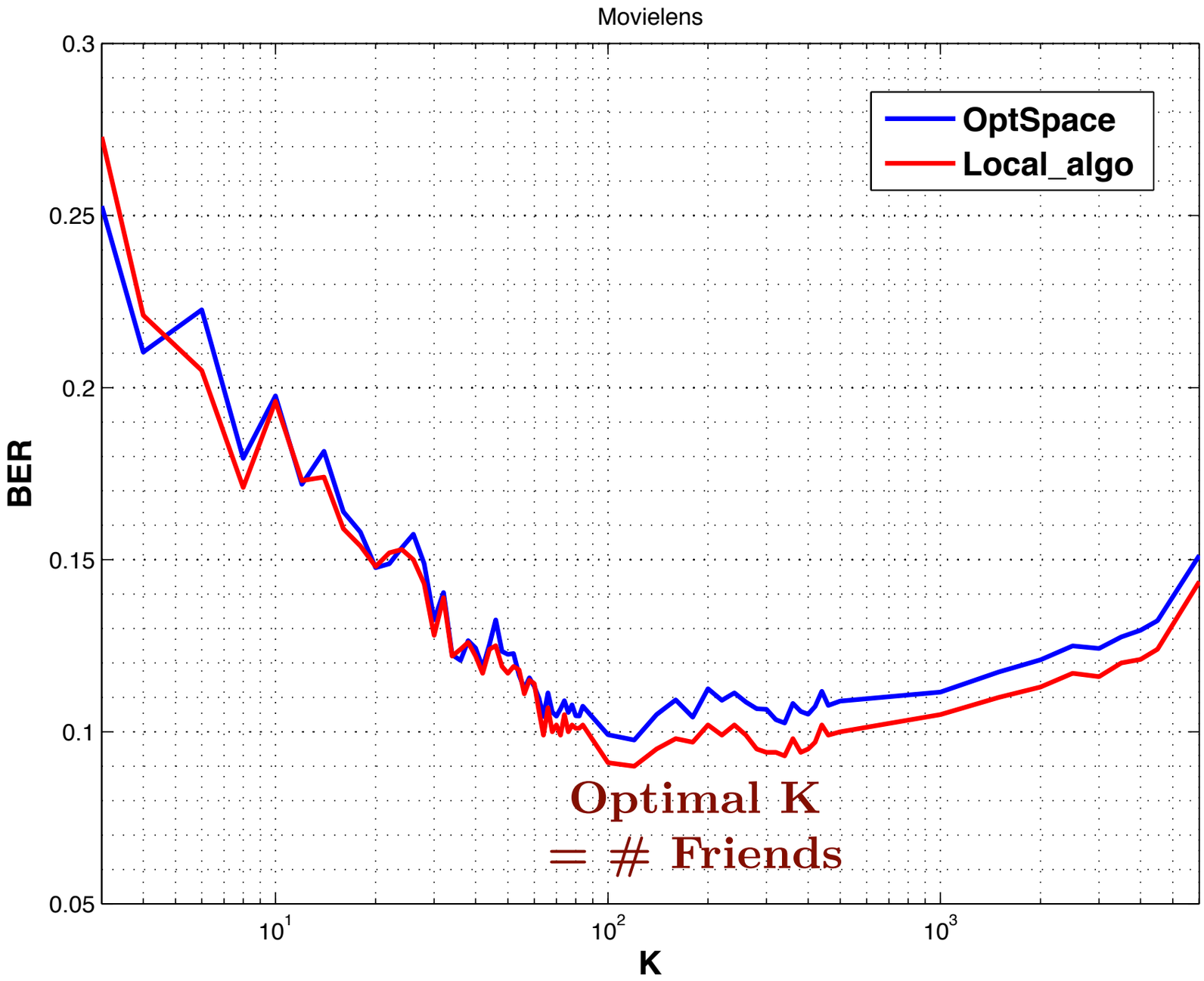
The Movielens Matrix



The Movielens Matrix



Degrees of Freedom



Remarks

Remarks

- PAF algorithm is scalable and competitive

Remarks

- PAF algorithm is scalable and competitive
- Provably good in near-quadratic regime
  - BER bounded away from 1/2

Remarks

- PAF algorithm is scalable and competitive
- Provably good in near-quadratic regime
  - BER bounded away from 1/2
- Near-linear regime: Blend side-information

Remarks

- PAF algorithm is scalable and competitive
- Provably good in near-quadratic regime
  - BER bounded away from 1/2
- Near-linear regime: Blend side-information
- Refining our simple model
  - Sampling - account for heavy tails
  - Incorporate item correlations
  - Rearranged general ‘smooth’ processes?
  - Incorporating side information?

Proof Outline

Recovering Entire Matrix: Converse

Recovering Entire Matrix: Converse

- Analyze probability of error in recovering entire matrix as matrix size grows

Recovering Entire Matrix: Converse

- Analyze probability of error in recovering entire matrix as matrix size grows
- To show that if cluster size is small, then w.h.p. no scheme can yield perfect recovery
  - In fact, a strong converse: probability of error approaches 1

Recovering Entire Matrix: Converse

- Analyze probability of error in recovering entire matrix as matrix size grows
- To show that if cluster size is small, then w.h.p. no scheme can yield perfect recovery
  - In fact, a strong converse: probability of error approaches 1
- Consider oracle that tells us the clusters

Recovering Entire Matrix: Converse

- Analyze probability of error in recovering entire matrix as matrix size grows
- To show that if cluster size is small, then w.h.p. no scheme can yield perfect recovery
  - In fact, a strong converse: probability of error approaches 1
- Consider oracle that tells us the clusters
- Once the clusters are known, the MAP decoder in each block is just majority decoder

Known Clusters

- Let $p_1 = \epsilon + 2(1-\epsilon)\sqrt{p(1-p)}$.
- If

$$s_*(\mathbf{X}) \geq \frac{\ln(mn)}{\ln(1/p_1)}$$

then $P_{e|\mathcal{A},\mathcal{B}}(\mathbf{X}) \to 0$.

- If

$$s^*(\mathbf{X}) \leq \frac{(1-\delta)\ln(mn)}{\ln(1/p_1)} \text{ for some } \delta > 0,$$

then $P_{e|\mathcal{A},\mathcal{B}}(\mathbf{X}) \to 1$.

Known Clusters

- Let $p_1 = \epsilon + 2(1-\epsilon)\sqrt{p(1-p)}$.
- If

$$s_*(\mathbf{X}) \geq \frac{\ln(mn)}{\ln(1/p_1)}$$

$\sim Cn^\alpha \ln(n)$

for $\epsilon = 1 - \dfrac{c}{n^\alpha}$

then $P_{e|\mathcal{A},\mathcal{B}}(\mathbf{X}) \to 0$.

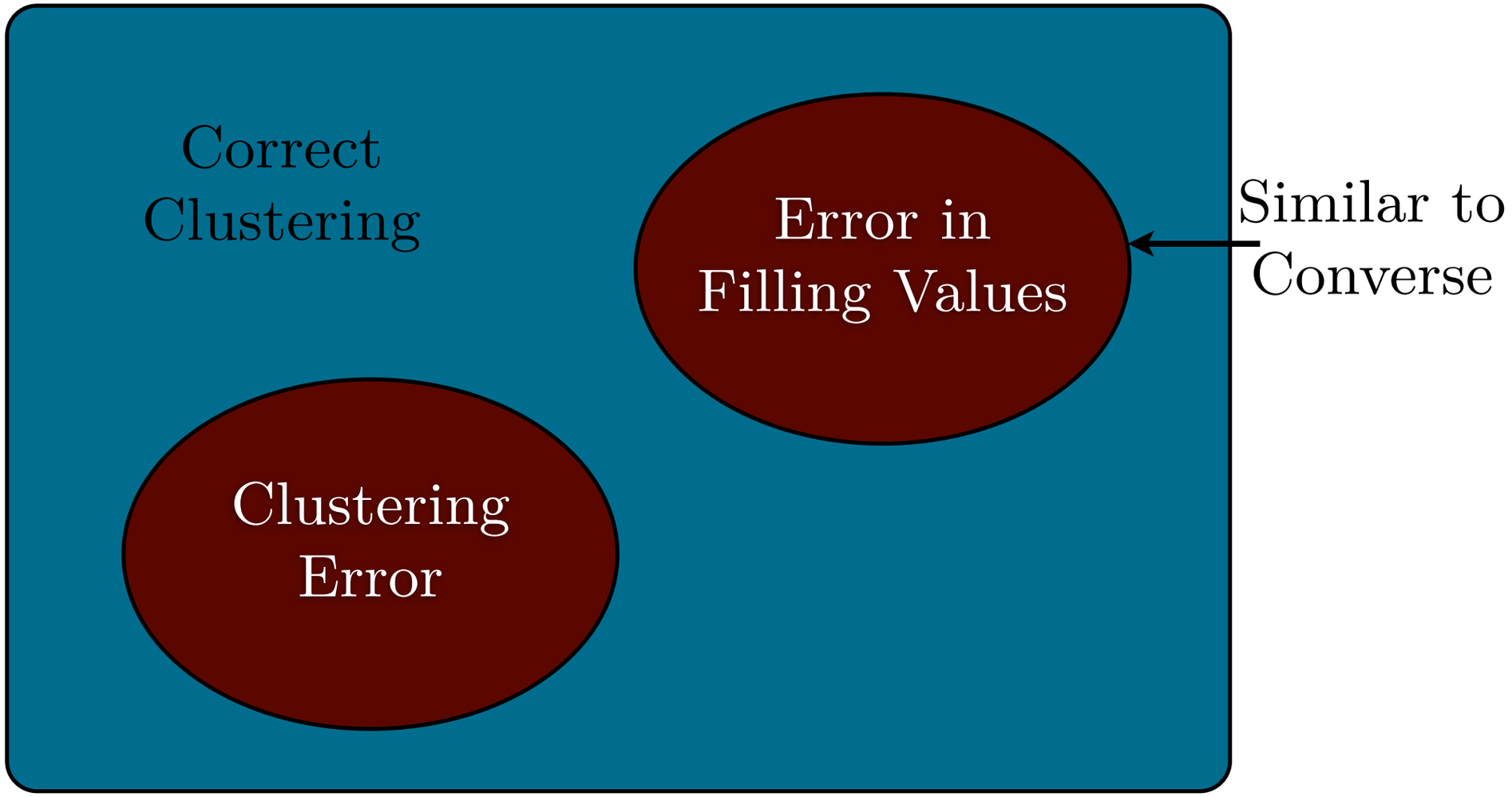
- If

$$s^*(\mathbf{X}) \leq \frac{(1-\delta)\ln(mn)}{\ln(1/p_1)} \text{ for some } \delta > 0,$$

then $P_{e|\mathcal{A},\mathcal{B}}(\mathbf{X}) \to 1$.

Achievability
(Small Clustering Error)

Achievability
(Small Clustering Error)



Achievability
(Small Clustering Error)

Achievability (Small Clustering Error)

- First cluster rows and columns, and then assuming correct clustering, apply majority decoding

Achievability (Small Clustering Error)

- First cluster rows and columns, and then assuming correct clustering, apply majority decoding
- **Row/Column Clustering:**
  - For every pair of rows, find the Hamming distance over commonly sampled positions
  - Compare with threshold to decide whether rows are in same cluster or not
  - Error in clustering if any pair is misclassified

Achievability (Small Clustering Error)

- First cluster rows and columns, and then assuming correct clustering, apply majority decoding
- **Row/Column Clustering:**
  - For every pair of rows, find the Hamming distance over commonly sampled positions
  - Compare with threshold to decide whether rows are in same cluster or not
  - Error in clustering if any pair is misclassified
- **Analysis:** Use concentration of metric around its mean to choose threshold

The Clustering Algorithm

- For rows i, j with N_{ij} commonly sampled positions

$$d_{ij} = \frac{1}{N_{ij}} \sum_{k=1}^{n} 1\left(Y_{ik} \neq e, Y_{jk} \neq e\right) 1(Y_{ik} \neq Y_{jk}).$$

- Choose threshold $d_0 \in (2p_0(1-p_0), 1/2)$.
- If $d_{ij} < d_0$, declare i, j in same cluster, else declare them to be in different clusters.

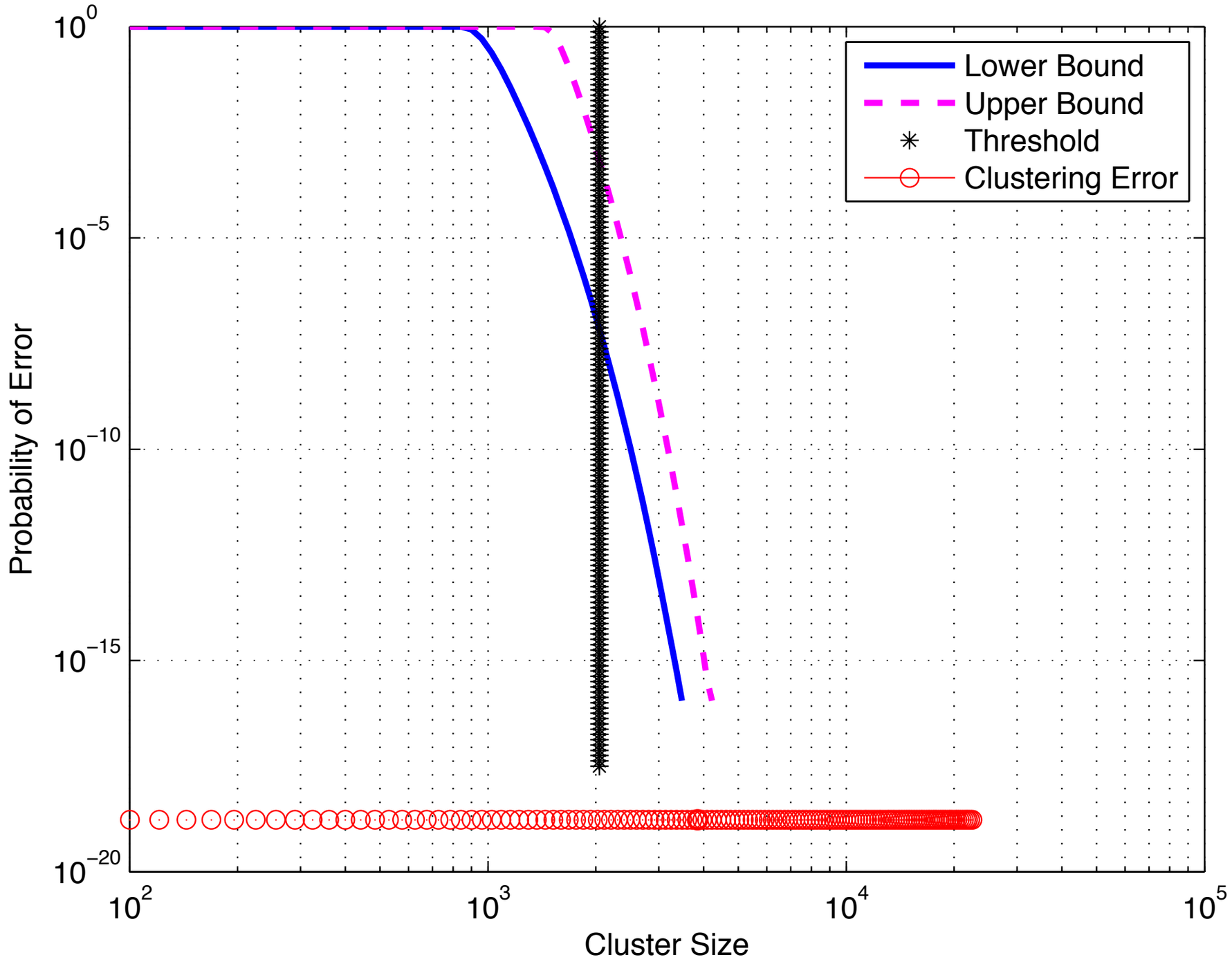
Clustering Error - Simpler Case

- **Simplifications:**
  - Square matrix, uniform cluster size, ϵ fixed
  - Suppose all channel parameters are known.
  - Normalize metric by n instead of N_{ij}.
- **Hypothesis A - Rows in same cluster:** Conditioned on $\mathbf{X}$, d_{ij} is avg. of i.i.d. Bernoulli variables with mean μ.
- **Hypothesis B - Rows in different cluster:** Conditioned on $\mathbf{X}$, d_{ij} is avg. of two groups of i.i.d. random variables and has mean $\mu + \delta s_{ij}/n$, $s_{ij} =$ Hamming distance between rows.

Clustering Error - Simpler Case (Contd.)

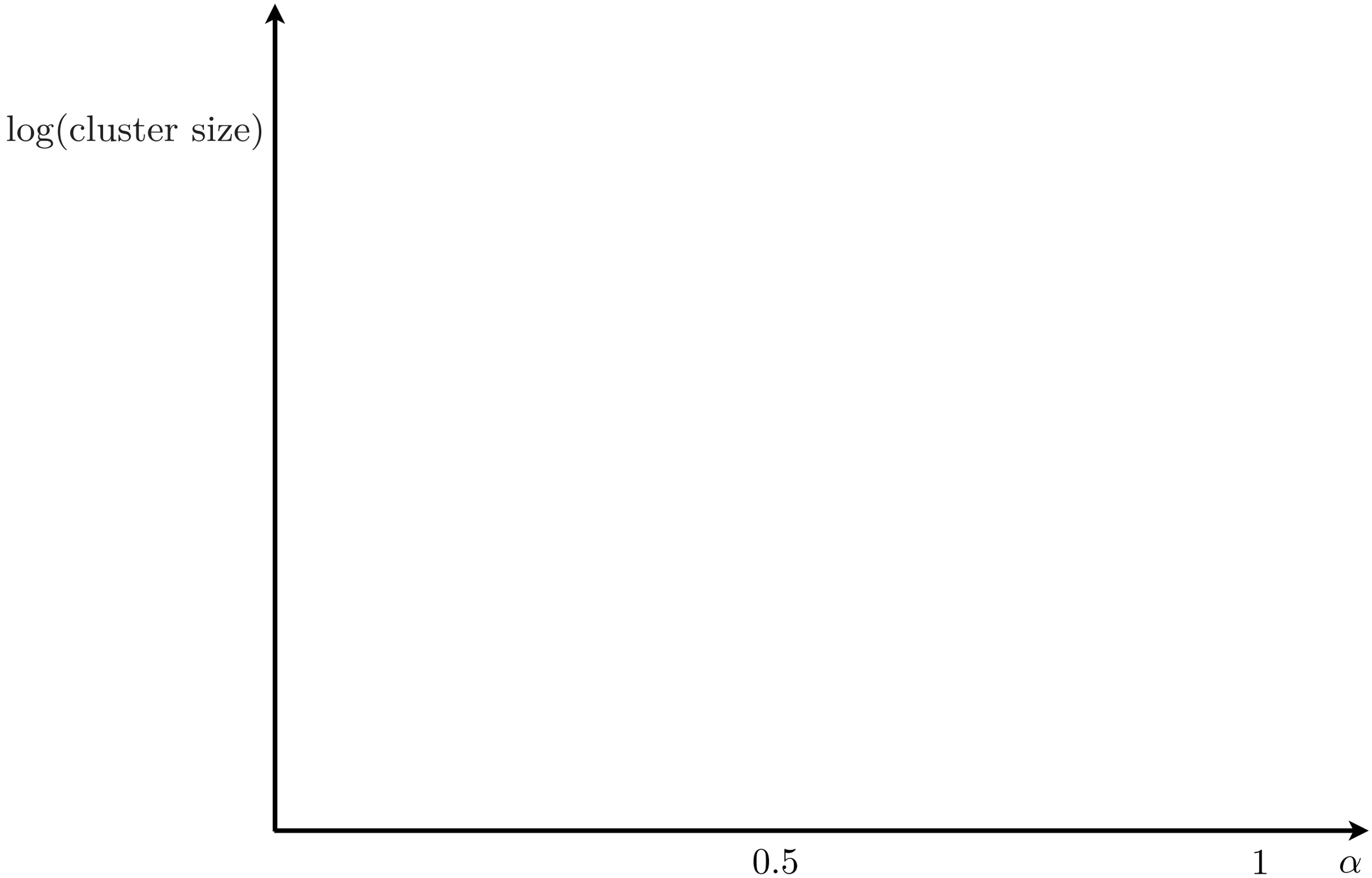
- Let t=# of clusters, n_0= size of cluster
- $s_{i,j}$ is $n_0\times$ Binomial(t,1/2); concentrates if $t \to \infty$
- Gap in means: $\delta/2$; pick d_0 in this gap.
- Errors:
  - Hypothesis A is true: $P(\text{error}|A) = O(\exp(-cn))$.
  - Hypothesis B is true: $P(\text{error}|B) = O(\exp(-ct))$.
- Clustering error: By union bound, diminishing provided $t > C\ln(n)$; otherwise does not depend on n_0.

Main Result (Illustration)

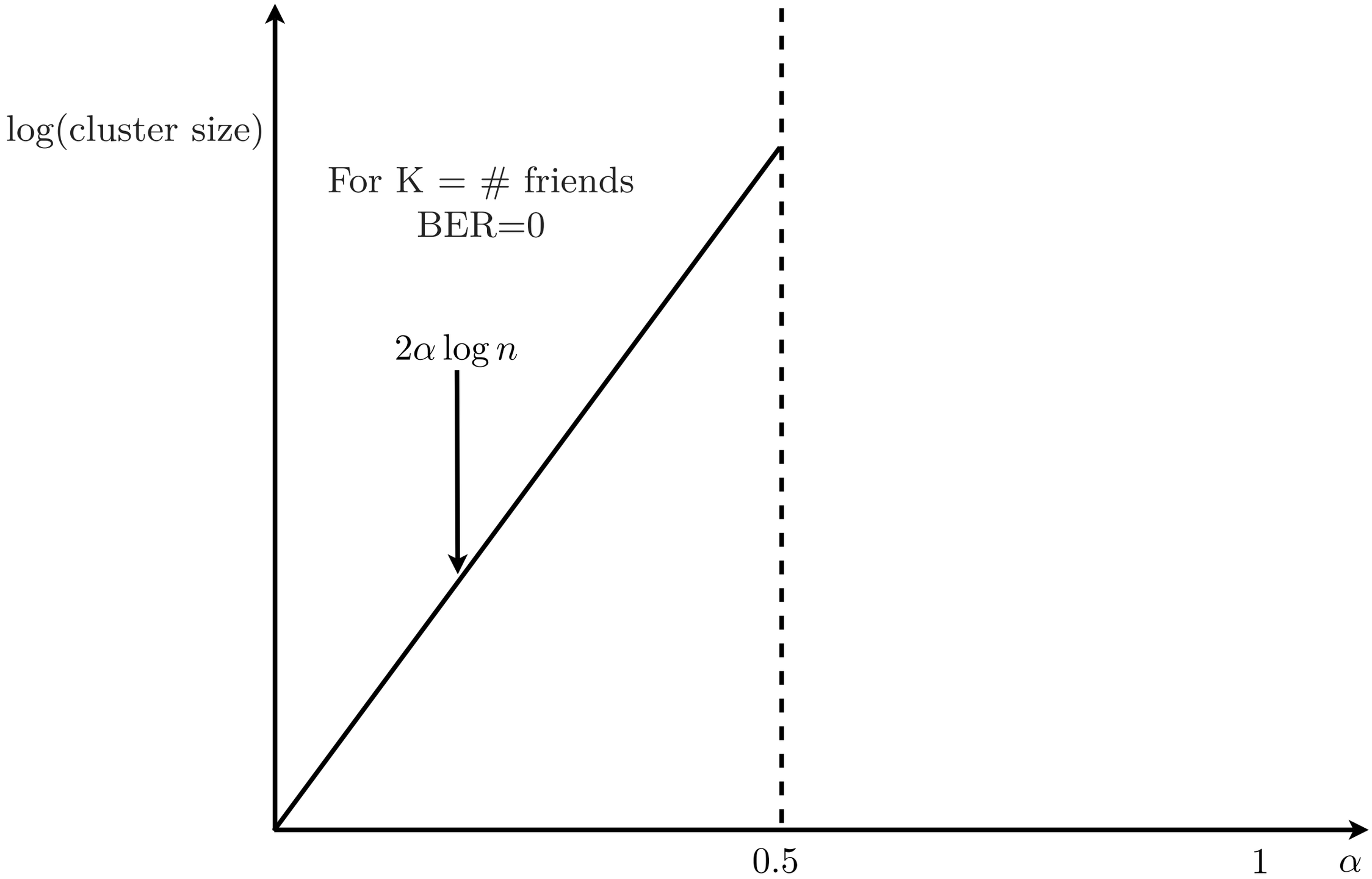


BER Analysis of PAF

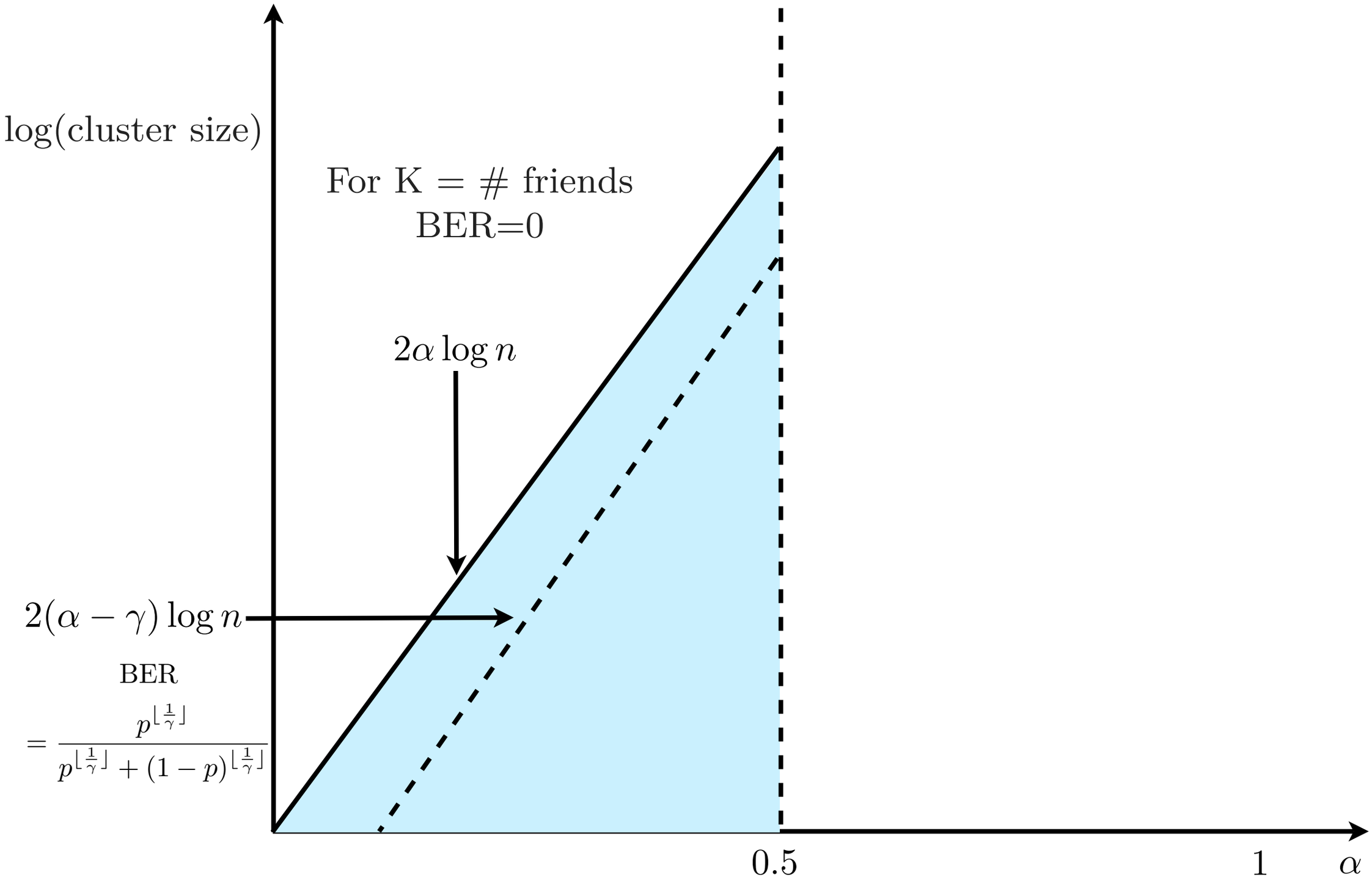
Recall: Asymptotic BER of PAF



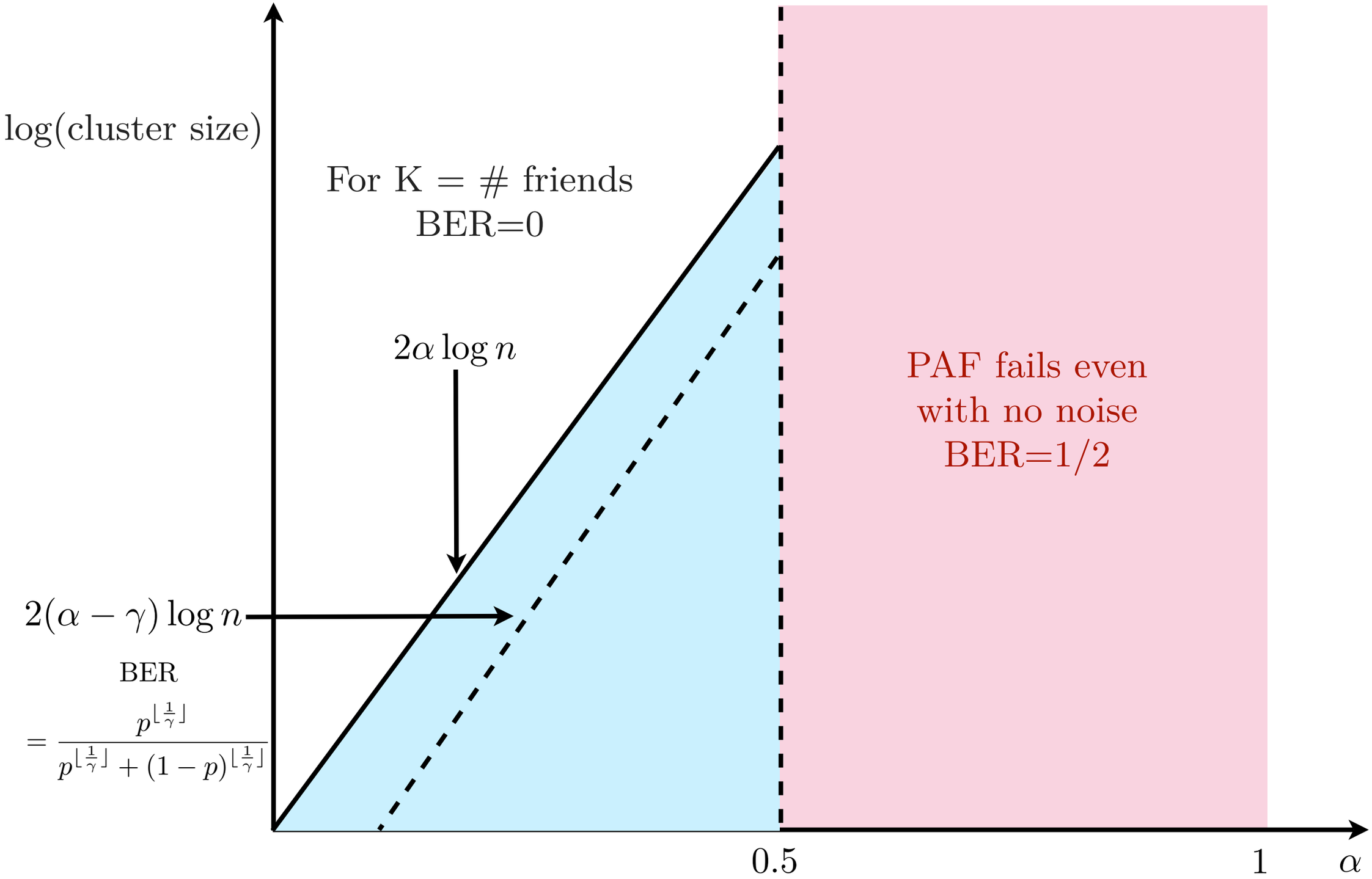
Recall: Asymptotic BER of PAF



Recall: Asymptotic BER of PAF



Recall: Asymptotic BER of PAF



Phase 2: Finding Good Neighbors

- Similarity between row 1 and another row in its cluster:

$$\text{Binomial}\left(n, c^2[p^2 + (1-p)^2]n^{-2\alpha}\right)$$

- Similarity between row 1 and a row in a different cluster:

$$\text{Binomial}\left(n, c^2 n^{-2\alpha}/2\right)$$

- Above marginals concentrate for $\alpha < 1/2$. So we hope to find good neighbors.

- Detailed calculations (accounting for dependence) confirm the hope.

Phase 2: Filtering Noise

- $K = \#$ friends - w.h.p. all neighbors are good
- Most popular column: w.h.p. # 1's $= \lfloor 1/\gamma \rfloor$, and # 0's $= 0$
  - For an arbitrary column, $E[\#\text{samples}] = \Theta(1/n^{\gamma})$
- Aposteriori probability:

$$P(X(1, j_*) = 0 | Y_K(:, j_*)) = \frac{p^{\#1-\#0}}{p^{\#1-\#0} + (1-p)^{\#1-\#0}}$$

Phase 3: Failure of PAF/ Clustering

Phase 3: Failure of PAF/ Clustering

- Most neighbors picked by PAF are bad
  - Even the clustering algorithm we considered fails

Phase 3: Failure of PAF/ Clustering

- Most neighbors picked by PAF are bad
  - Even the clustering algorithm we considered fails
- Similarity metrics do not concentrate
  - Probability that a row is not sampled is non-zero
  - Probability that distinct part of bad candidates is never sampled is non-zero

Phase 3: Failure of PAF/ Clustering

- Most neighbors picked by PAF are bad
  - Even the clustering algorithm we considered fails
- Similarity metrics do not concentrate
  - Probability that a row is not sampled is non-zero
  - Probability that distinct part of bad candidates is never sampled is non-zero
- Number of clusters increasing to infinity implies # good candidates/(# bad candidates) approaches zero

Phase 3: Failure of PAF/ Clustering

- Most neighbors picked by PAF are bad
  - Even the clustering algorithm we considered fails
- Similarity metrics do not concentrate
  - Probability that a row is not sampled is non-zero
  - Probability that distinct part of bad candidates is never sampled is non-zero
- Number of clusters increasing to infinity implies # good candidates/(# bad candidates) approaches zero
- A problem for any pairwise scheme

Future Directions

Improving PAF

Improving PAF

- PAF exploits correlations only amongst users
  - Correlation amongst items?

Improving PAF

- PAF exploits correlations only amongst users
  - Correlation amongst items?
- Gap w.r.t. optimal threshold for recovering entire matrix

Improving PAF

- PAF exploits correlations only amongst users
  - Correlation amongst items?
- Gap w.r.t. optimal threshold for recovering entire matrix
- How do we modify PAF?
  - For an item, find most similar items
  - Given top few recommendations by PAF, how do we use the list of related items to pick a recommendation?
  - Analysis and empirical evaluation

Improving PAF

- PAF exploits correlations only amongst users
  - Correlation amongst items?
- Gap w.r.t. optimal threshold for recovering entire matrix
- How do we modify PAF?
  - For an item, find most similar items
  - Given top few recommendations by PAF, how do we use the list of related items to pick a recommendation?
  - Analysis and empirical evaluation
- Note: computational load increases

General ‘Smooth’ Processes

General 'Smooth' Processes

- Block constant model: limited

General 'Smooth' Processes

- Block constant model: limited
- Extension to Markov random fields?

General 'Smooth' Processes

- Block constant model: limited
- Extension to Markov random fields?
- Upper bound on achievable BER
  - Use PAF or its variants

General 'Smooth' Processes

- Block constant model: limited
- Extension to Markov random fields?
- Upper bound on achievable BER
  - Use PAF or its variants
- Lower bound on BER?
  - Due to rearrangements, we do not know the dependence neighborhood
  - Suppose an oracle gives us the neighborhood
  - We do not know the conditional law of the MRF
  - How do we estimate the MRF?

Side-Information

Side-Information

- Social connections, item content, etc.

Side-Information

- Social connections, item content, etc.
- Side-information may or may not be correlated with rating data
  - How do we identify relevant side-information?
  - How do we use relevant side-information - compute similarities or fit a regression?

Side-Information

- Social connections, item content, etc.
- Side-information may or may not be correlated with rating data
  - How do we identify relevant side-information?
  - How do we use relevant side-information - compute similarities or fit a regression?
- Related to multi-view clustering
  - Given multiple feature similarities, how do we cluster objects?
  - Features could reflect same or different relationships

The Time Axis

The Time Axis

- Time-stamps play an important role in Netflix Grand Prize (Koren, 2009, Bell et al, 2009)

The Time Axis

- Time-stamps play an important role in Netflix Grand Prize (Koren, 2009, Bell et al, 2009)
- Temporal models and provably good schemes?

The Time Axis

- Time-stamps play an important role in Netflix Grand Prize (Koren, 2009, Bell et al, 2009)
- Temporal models and provably good schemes?
- Dynamics of recommender systems:
  - Recommend items
  - Users choose from the recommended items
  - How does the rating matrix evolve?

Matrix Completion and Compressed Sensing

Matrix Completion and Compressed Sensing

- Sparse vector, say K non-zero entries
  - Product of any (K+1) or more entries is zero
  - Multinomial constraints

Matrix Completion and Compressed Sensing

- Sparse vector, say K non-zero entries
  - Product of any (K+1) or more entries is zero
  - Multinomial constraints
- Low-rank N-by-N matrix, say rank K
  - Consider coefficients of characteristic polynomials
  - Each a multinomial of matrix entries
  - N-K+1 coefficients are zero

Matrix Completion and Compressed Sensing

- Sparse vector, say K non-zero entries
  - Product of any (K+1) or more entries is zero
  - Multinomial constraints
- Low-rank N-by-N matrix, say rank K
  - Consider coefficients of characteristic polynomials
  - Each a multinomial of matrix entries
  - N-K+1 coefficients are zero
- General problem: Recover a vector in satisfying multinomial constraints from few samples

Miscellaneous

Miscellaneous

- Dealing with spam
  - Consider a group of rogue users
  - Given PAF, what is the worst effect they can have, given a bound on the number of items they can rate?

Miscellaneous

- Dealing with spam
  - Consider a group of rogue users
  - Given PAF, what is the worst effect they can have, given a bound on the number of items they can rate?
- Privacy
  - Personalization: Need information about taste
  - But potential for misuse of information
  - Let market figure out the user-vendor trust level?
  - A peer-to-peer system?

Some Related References
(Not Exhaustive)

References

- Review Papers
  - G. Adomavicius, A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, pp. 734-749, June, 2005
  - X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009
- Datasets
  - Movielens: http://www.movielens.org/
  - Yahoo Webscope: http://webscope.sandbox.yahoo.com/
- Netflix Prize
  - R. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," KDD, 2007
  - Y. Koren, "The BellKor Solution to the Netflix Grand Prize," 2009, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.162.2118
  - A. Toscher et al, "The BigChaos Solution to the Netflix Grand Prize," 2009, http://www.stat.osu.edu/~dmsl/GrandPrize2009_BPC_BigChaos.pdf

References (Contd.)

- Low-Rank Matrix Completion
  - E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. of Comput. Math.*, **9** 717-772
  - E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inform. Theory*, **56**(5), 2053-2080
  - R. Keshavan, A. Montanari, and S. Oh, "Matrix Completion from Noisy Entries," *Journal of Machine Learning Research*, 2010
  - ISIT 2009, 2010 special sessions on matrix completion
- Estimation of rearranged smooth processes
  - S. T. Aditya, O. Dabeer, and B. K. Dey, "A channel coding perspective of recommendation systems," *ISIT* 2009, pp. 319–323
  - S. T. Aditya, O. Dabeer, and B. K. Dey, "A channel coding perspective of collaboratuve filters," to appear in *IEEE Transactions on Information Theory*, 2011
  - K. Barman, O. Dabeer, "Local Popularity Based Collaborative Filters," *ISIT*, June 2010, Austin, USA
  - K. Barman, O. Dabeer, "What is Popular Amongst Your Friends?" submitted to *IEEE Transactions on Information Theory,* July 2010 (arXiv:1006.1772)

References (Contd.)

- Committee methods
  - R. E. Schapire, “The boosting approach to machine learning: An overview,” Lecture Notes in Staitsics, New York, pp. 149-172, 2003
  - Tin Kam Ho, “The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.20, no.8, pp.832-844, Aug 1998
  - T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning,” Springer-Verlag, 2003