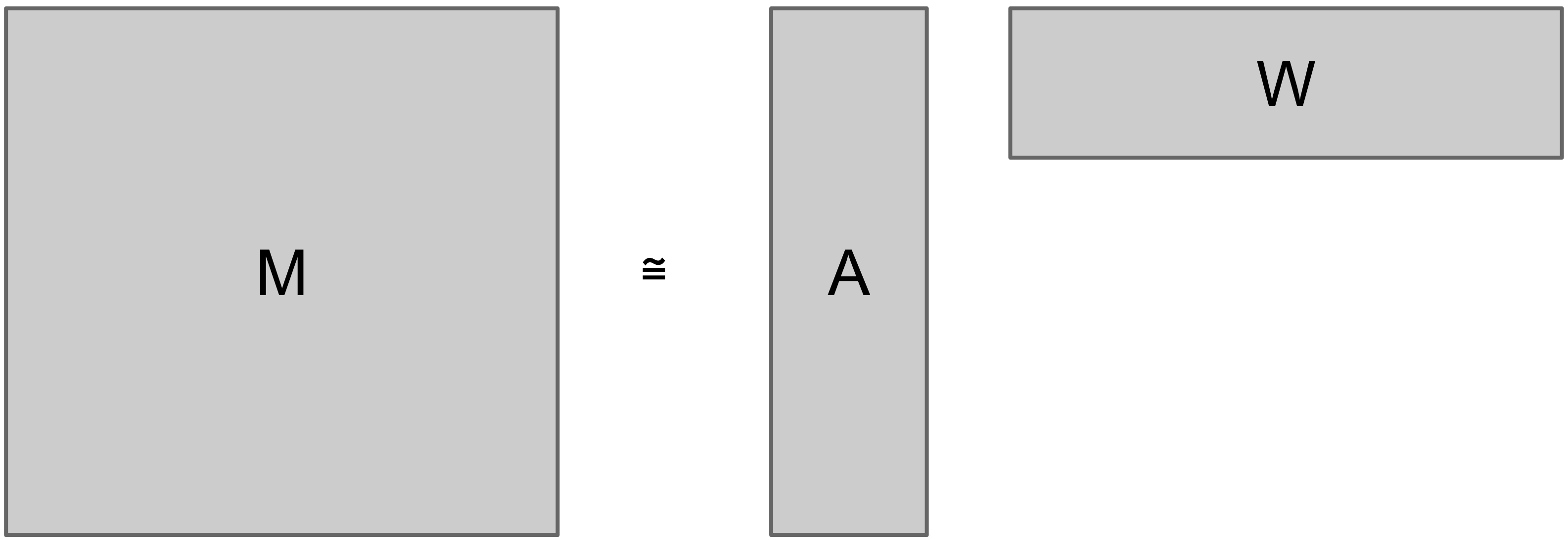
Lecture 5: NMF and Topic Modeling in Practice

NMF and Topic Modelling in Practice

- Nonnegative Matrix Factorization
  - Alternating Minimization

- Topic Models
  - EM algorithm
  - Implementing the provable algorithm
  - Evaluating topic modeling algorithms
  - Challenges and new algorithms

Nonnegative Matrix Factorization



- NP-hard in general [Vavasis]
- Solvable in polynomial time when
  - rank is constant
  - A is separable

Algorithm in Practice: Alternating Minimization [Lee Seung '00]

- Given A, can find the best W

$$\min \|M - AW\|_F$$
$$W_{i,j} \geq 0$$

- Given W, can find the best A
- Alternate between 2 steps.

- $\|M - AW\|_2$ converges
- May not converge to global OPT

Algorithm in Practice: Alternating Minimization [Lee Seung '00]

- Different objectives

$$\min D(M||AW) = \sum (M_{ij} \log M_{ij}/(AW)_{ij} - M_{ij} + (AW)_{ij}))$$

- Can still do alternating minimization
- Still may not converge to global optimum.

- Open: Why these algorithms work in practice?
  Can we prove they work for separable NMF?

Topic Models

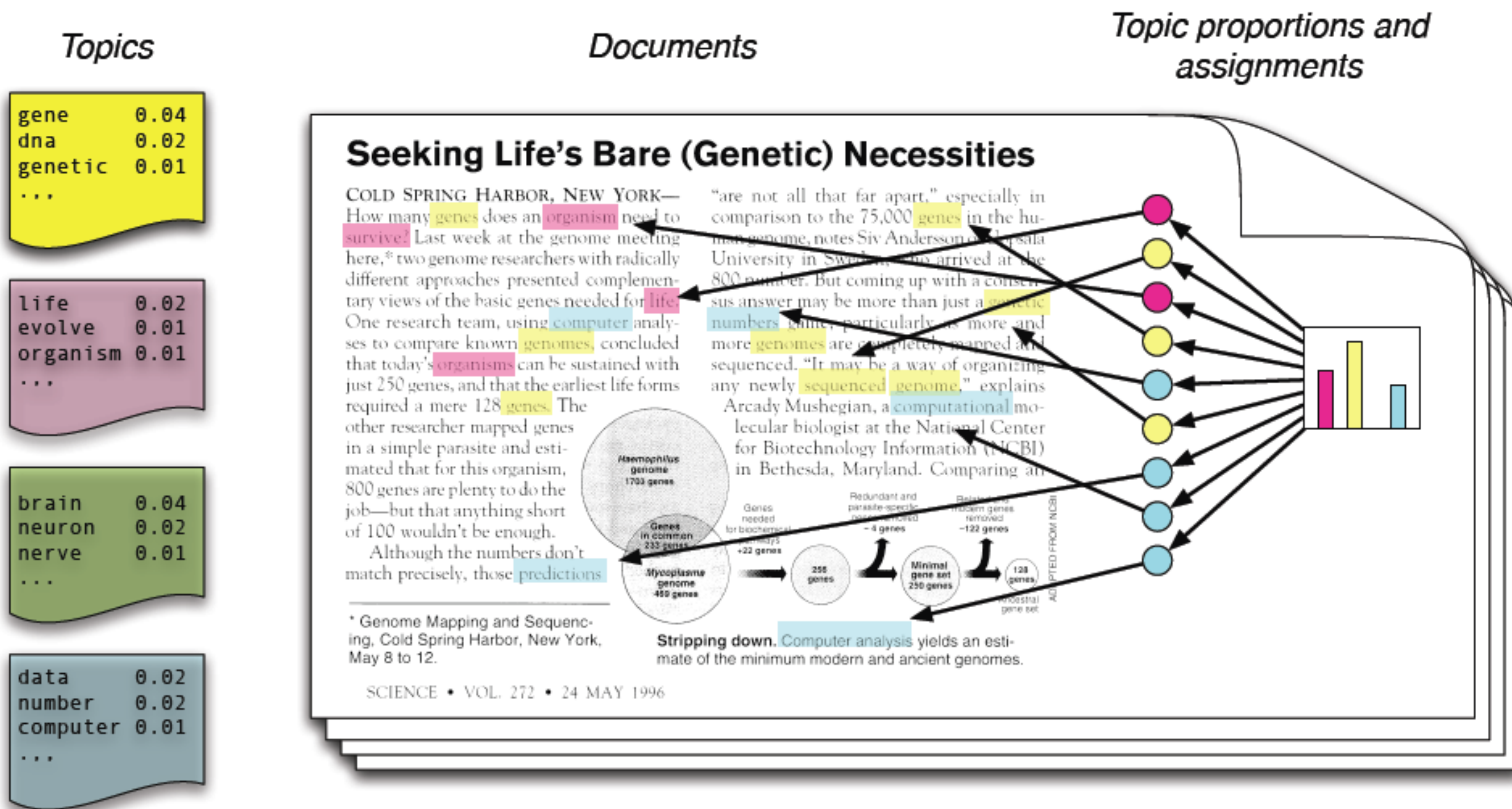


image from [Blei Ng Jordan 2012]

Recap: Probabilistic Topic Model

Known: Topic Matrix A

For each document

- Sample length of document
- Sample a mixture of topics
- For each word
  - Sample a topic
  - Sample a word from the topic

Expectation-Maximization algorithm

- Alternate between 2 steps

- E (Expectation) step

Based on current parameters (topics), estimate the (hidden) topic assigned to each word.

- M(Maximization) step

Based on the topics assigned to words, find the best (most likely) word-topic matrix.

Expectation-Maximization algorithm

- EM tries to solve the maximum likelihood problem.
- EM converges, but may not to global OPT

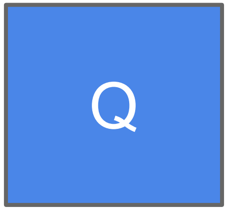
- Problem: E-step is already hard to compute
  - Use approximation (Variational EM)
  - Use sampling (Markov-Chain Monte-Carlo, Gibbs)
- Many ways to optimize/parallelize/…
- Many packages ready for applications.

Implementing Provable Algorithm

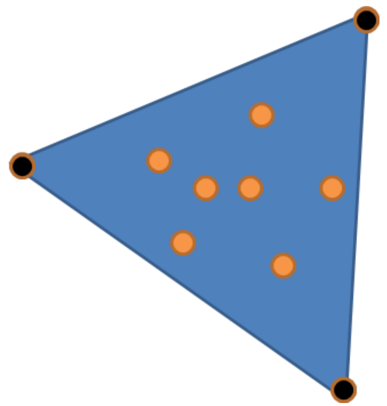
- Provable algorithms may not be practical
- Running time may be a large polynomial.
- Sample complexity may be far from optimal.
- Algorithms may not be robust to model mismatch.

Recall: Algorithm for Topic Modeling

- Estimate word-word correlation matrix



- Apply NMF Algorithm
  - Test each word (with a linear program)
  - Compute A’ matrix (again by LP)



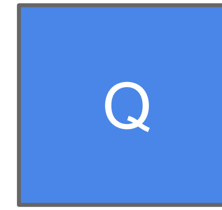
- Use Bayes’ rule to compute the topic matrix

Difficulties

- Effectively estimate the word-word correlation?
- Efficiently solve many Linear Programs?
- Real documents satisfy “anchor words” assumption?

Estimating Word-Word Correlation

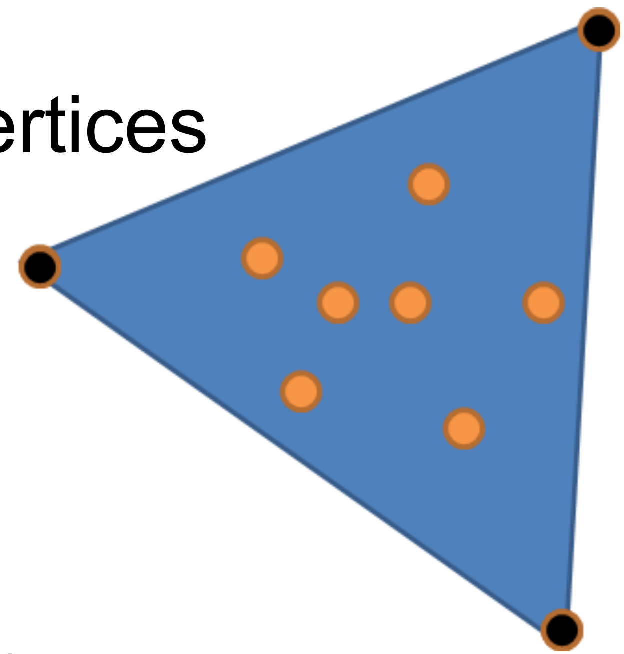
- $Q_{i,j}$ = Pr[first word is i, second word is j]



- Need to consider all N(N-1)/2 pairs for a length N document.
- Can only estimate for frequent words
- Prune stop words and rare words.

Nonnegative Matrix Factorization

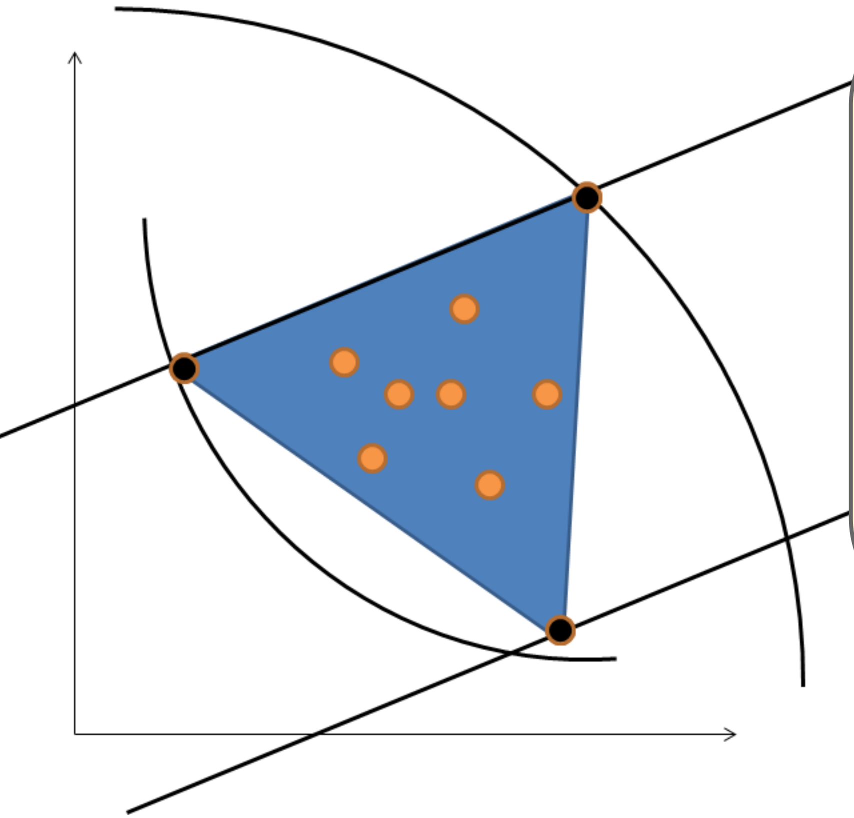
- Recall:
  Separable NMF$\Leftrightarrow$ Finding vertices



- Solving one linear program
  for each word is too slow!

- Need to find faster algorithms.

Faster Algorithm for Separable NMF



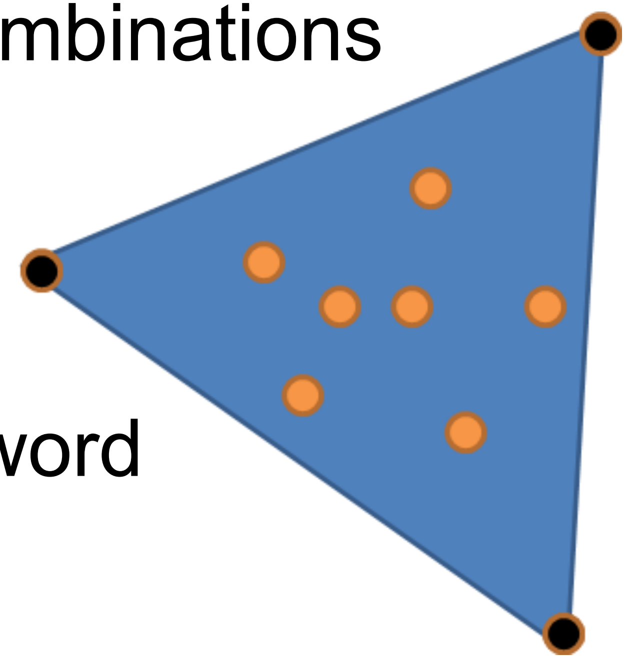
Find the farthest point to origin

REPEAT k-1 times

    Find the point farthest to affine hull of previously found points.

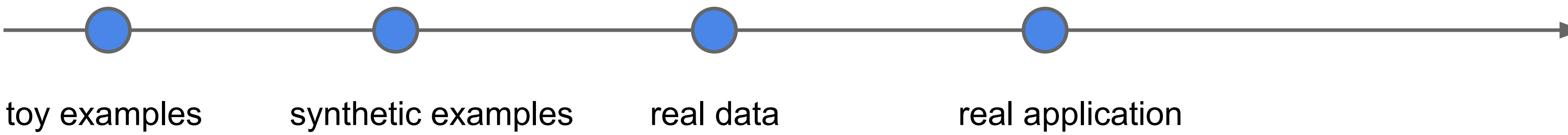
Finding Convex Combinations

- Given anchor words, represent all other words as convex combinations
- Different objectives: $\|\ \|_2$ norm, KL-Divergence



- A convex program for each word
  - Low dimensional
  - Can be solved approximately
  - Use gradient descent/exponentiated gradient

Evaluation

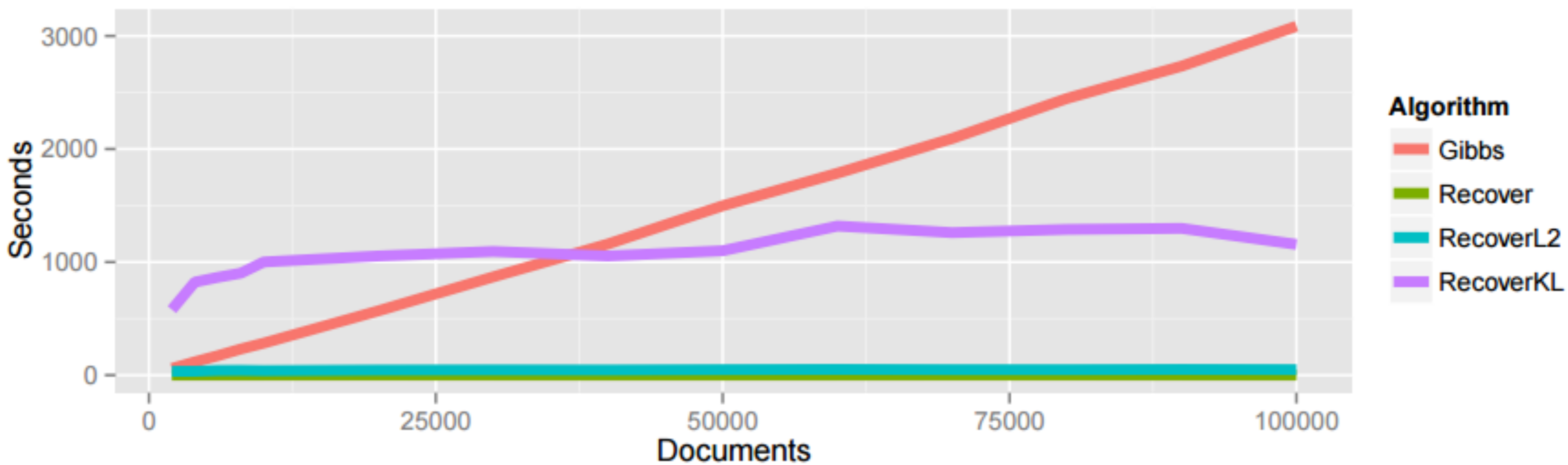


- Toy Examples: correctness.
- Synthetic Examples: running time, sample complexity, robustness
- Real Data
  - Qualitative evaluation: look at the topics found
  - Quantitative evaluation: held-out likelihood, …
- Real Application: Apply topic models to recommend articles, social science, ...

Evaluating Topic Modeling Algorithm

- Compare to MALLAT
  (package based on Gibbs sampling)
- Variants of algorithms
  - Recover: Basic algorithm
  - Recover-L2: Try to minimize $||Q-AW||_F$
  - Recover-KL: Try to minimize KL-divergence between rows of Q and AW.
- Data Set: UCI New York Times
  - 295k articles, 15k vocabulary, average length~300

Running Time

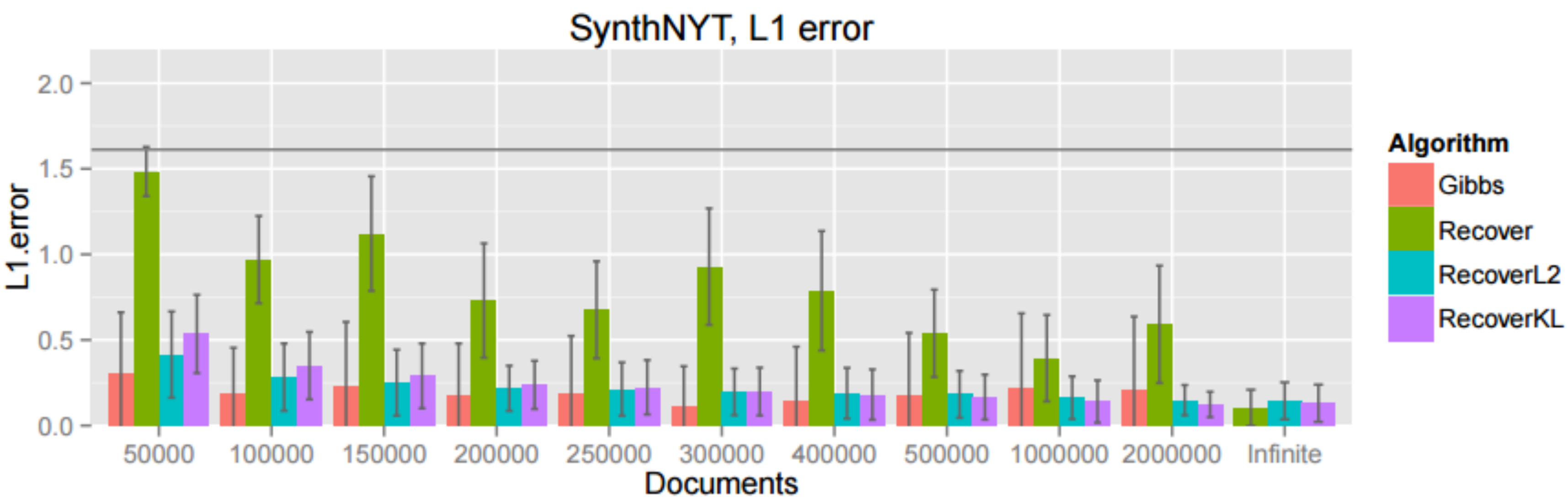


- Algorithms are faster than MALLAT, because most of the work is done on the word-word correlation matrix

Semi-synthetic Example

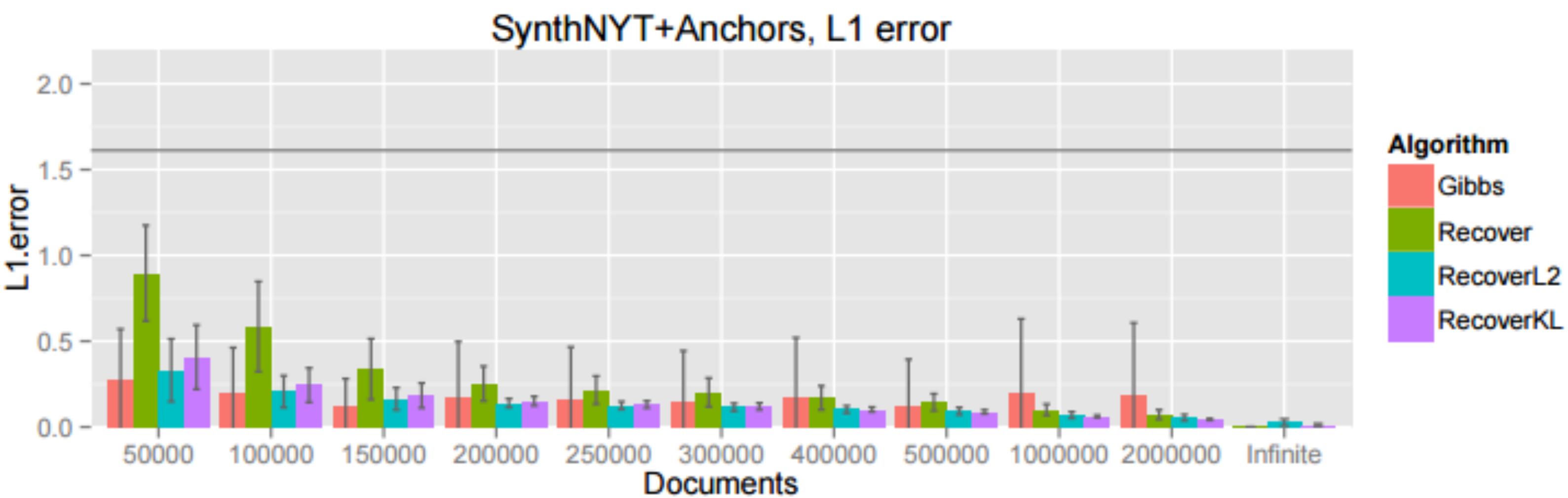
- Idea: Compute topic matrix by running MALLET on NYT data set, then generate synthetic documents.
- Benefit:
  - Has ground truth, measure error in parameter space
  - Easy to tweak parameters (different topic models, topic matrix, # documents, #words, ...)
  - Topic matrix is “natural”
- Data is still generated from the model, hard to evaluate the robustness of algorithm.

Semi-synthetic Experiments



- Performance is comparable to MALLAT, especially with more documents.
- Does not achieve 0 error with infinite data (not separable)

Anchor Words?



- Most topics have anchor words.
- Algorithms works OK even when some topics do not have anchor words.

Real Data (sample topics)

RecoverL2	president zzz_clinton zzz_white_house zzz_bush official zzz_bill_clinton
Gibbs	zzz_bush zzz_george_bush president administration zzz_white_house zzz_dick_cheney
RecoverL2	father family **zzz_elian** boy court zzz_miami
Gibbs	zzz_cuba zzz_miami cuban zzz_elian boy protest
RecoverL2	oil prices percent million market zzz_united_states
Gibbs	oil power energy gas prices plant
RecoverL2	**zzz_microsoft** company computer system window software
Gibbs	zzz_microsoft company companies cable zzz_at zzz_internet
RecoverL2	government election zzz_mexico political zzz_vicente_fox president
Gibbs	election political campaign zzz_party democratic voter
RecoverL2	fight **zzz_mike_tyson** round right million champion
Gibbs	fight zzz_mike_tyson ring fighter champion round

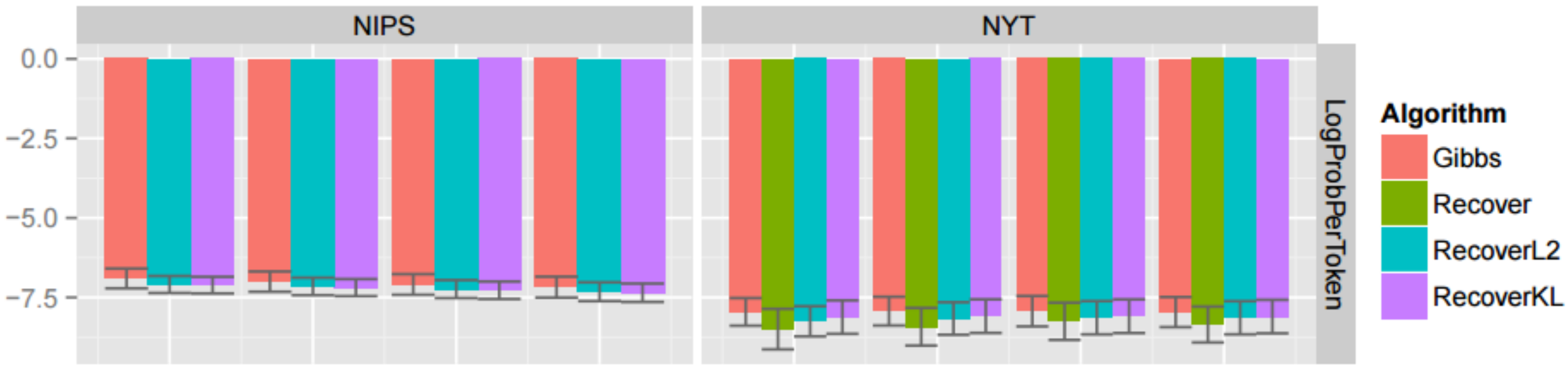
Real Data (Held-out likelihood)

- Idea: For each document, show a fraction of words, use the learned topic matrix to predict the distribution $\Pr[z = i|doc]$

- For the rest of the words $i_1, i_2, \ldots$
  Score = $\sum_j \log \Pr[z = i_j|doc]$

- Details matter (how to predict $\Pr[z=i|doc]$, fraction of held-out, smoothing…)

Real Data (Held-out likelihood)



- MALLAT is better, but RecoverKL is close.
- Recover algorithms followed by MALLAT improves held-out likelihood.

Challenges and New Algorithms

- What if anchor-word assumption is not true?
  - For LDA, can use tensor decomposition [AFHKL’12]
  - Only appear in 1 topic ⇒ Only appear in few topics (subset separable [GZ’15])
  - “Catch Words”: words that appear more frequently in one topic than all others [BBGKP’15]
- How to guess the number of topics?
  - Use low dimensional embeddings? [LeeMimno’14]
- Variants of topic models?
  - multilingual, temporal, ...

Homework

- Homework 1 is out, due in 2 weeks (9/24/2015 in class)
- Latex strongly encouraged.
- Discussions are allowed, but must acknowledge.
- Start early.
- Questions: email rongge@cs.duke.edu.

References

Codes for Recover Algorithm:

http://www.cs.nyu.edu/~halpern/code.html

MALLAT package

http://mallet.cs.umass.edu/

Papers

[Lee Seung ’00]

[AFHKL’12]

[LeeMimno’14]

[GZ’15]

[BBGKP’15] (not available yet)