# Monte Carlo MCMC
## Efficient Inference by Approximate Sampling

**Sameer Singh**, Michael Wick, Andrew McCallum

UMASS AMHERST

# Overview

- MCMC is a popular choice for inference in NLP
  - But is often slow in practice
- Existing work has focused on:
  - Modifying the model for faster sampling
  - Generating multiple samples simultaneously
  - Improving quality of each sample
- Instead, we generate “approximate samples”
  - But each sample is much faster
- Results in up to 13 times speedup!

# Background

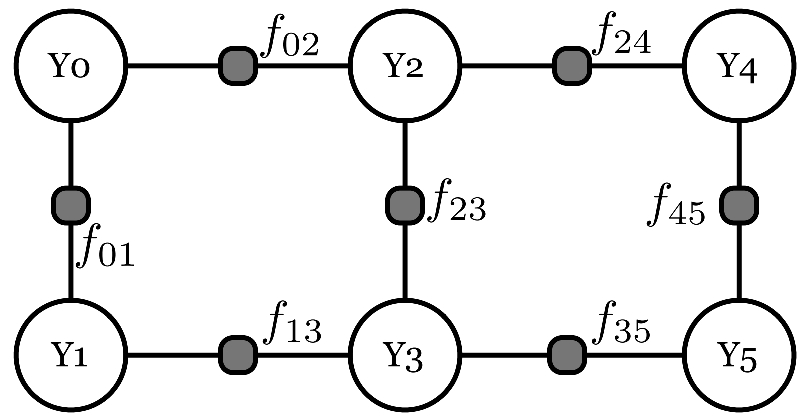
# Graphical Models

- Factor Graphs
- Variables **Y**
- Factors **F**
- Score of a configuration:

$$\psi(\mathbf{Y}{=}\mathbf{y}) = \sum_{f \in \mathbf{F}} f(\mathbf{y}_f)$$

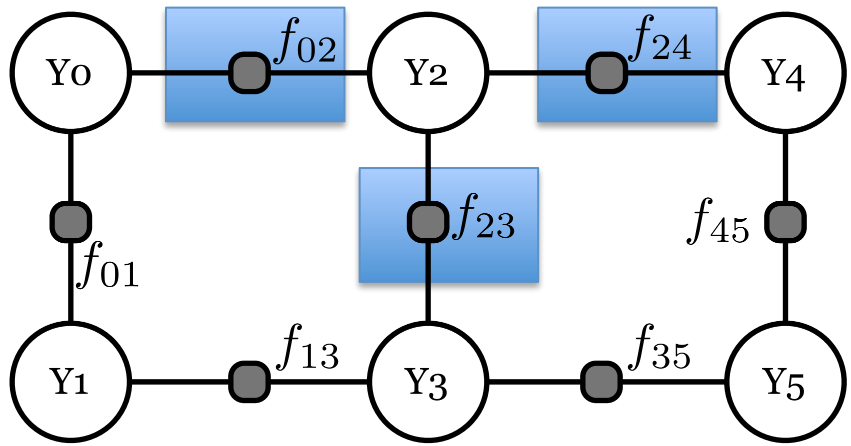
- Probability:

$$p(\mathbf{Y}{=}\mathbf{y}) = \frac{1}{Z} \exp \psi(\mathbf{y})$$



# Markov Chain Monte Carlo

1. Current Sample, **y**
2. Propose a move: **y** → **y'**
3. Accept with Probability α




$$\alpha(\mathbf{y},\mathbf{y'}) = \frac{p(\mathbf{y'})}{p(\mathbf{y})}$$

$$= \exp\ \psi(\mathbf{y'}) - \psi(\mathbf{y})$$

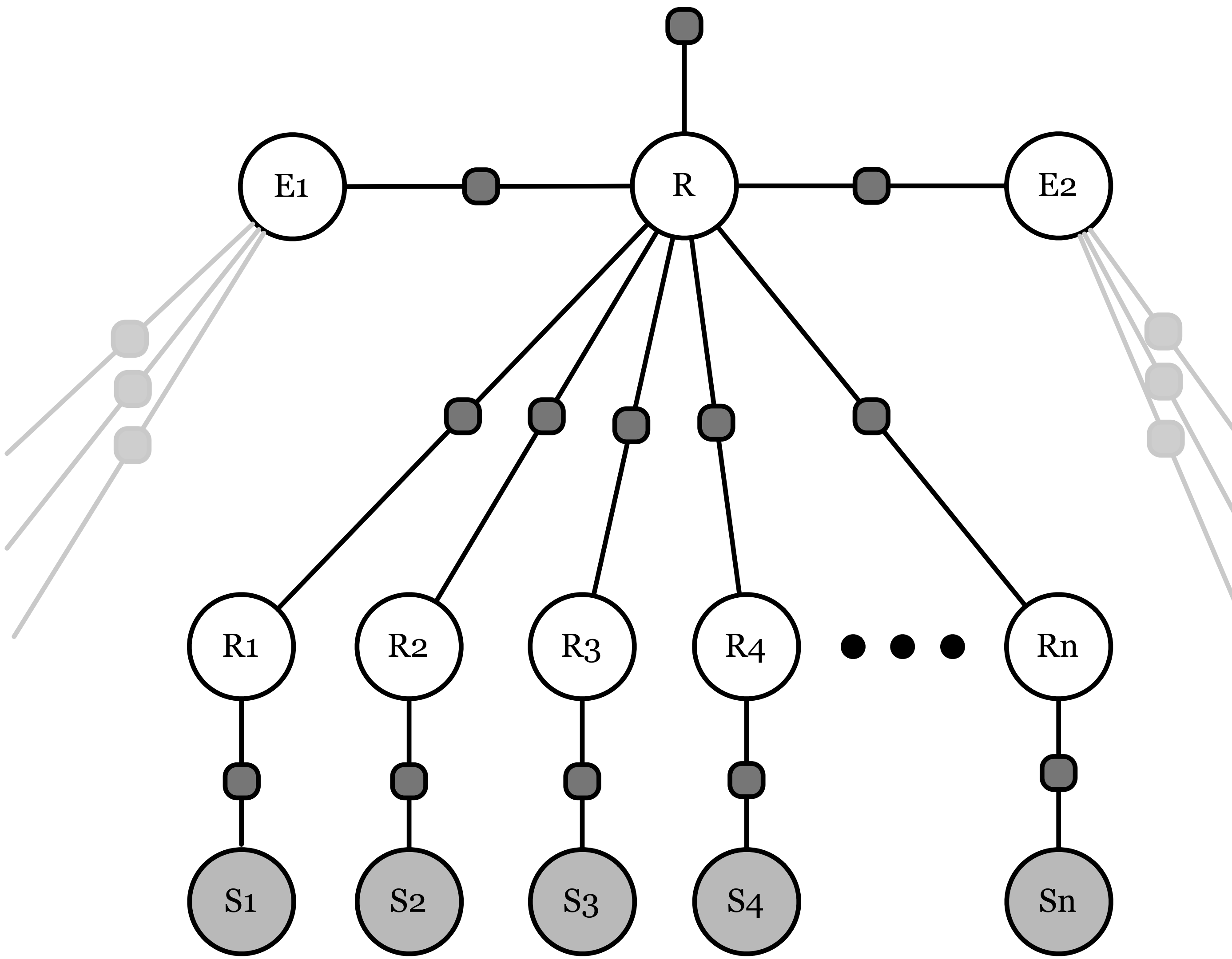
$$= \exp\ \psi(\mathbf{y'}/\mathbf{y}) - \psi(\mathbf{y}/\mathbf{y'})$$

4. Current sample ← **y'**

# Markov Chain Monte Carlo

- **Pros**: Low memory requirement, etc.
- Generating a sample is often fast
  – Depends only on factors involved in a proposal
- Unfortunately, sometimes this is a bottleneck
  1. If a variable neighbors many factors
  2. A proposal changes many variables
  3. Scoring a factor is slow (expensive features)

# Example: Relation Extraction



# Monte Carlo MCMC

# Approximating Sampling

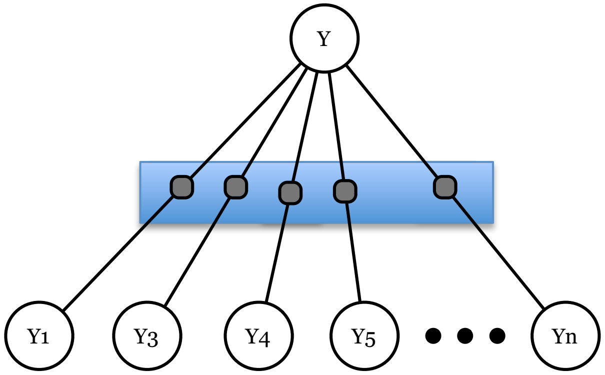
- Acceptance ratio involves partial model scores

$$\alpha(\mathbf{y},\mathbf{y}') = \exp\ \psi(\mathbf{y}'/\mathbf{y}) - \psi(\mathbf{y}/\mathbf{y}')$$

$$\psi(\mathbf{y}/\mathbf{y}') = \sum_{f\in\mathbf{F}'} f(\mathbf{y}) = |\mathbf{F}'|\mathbb{E}_{\mathbf{F}'} f(\mathbf{y})$$

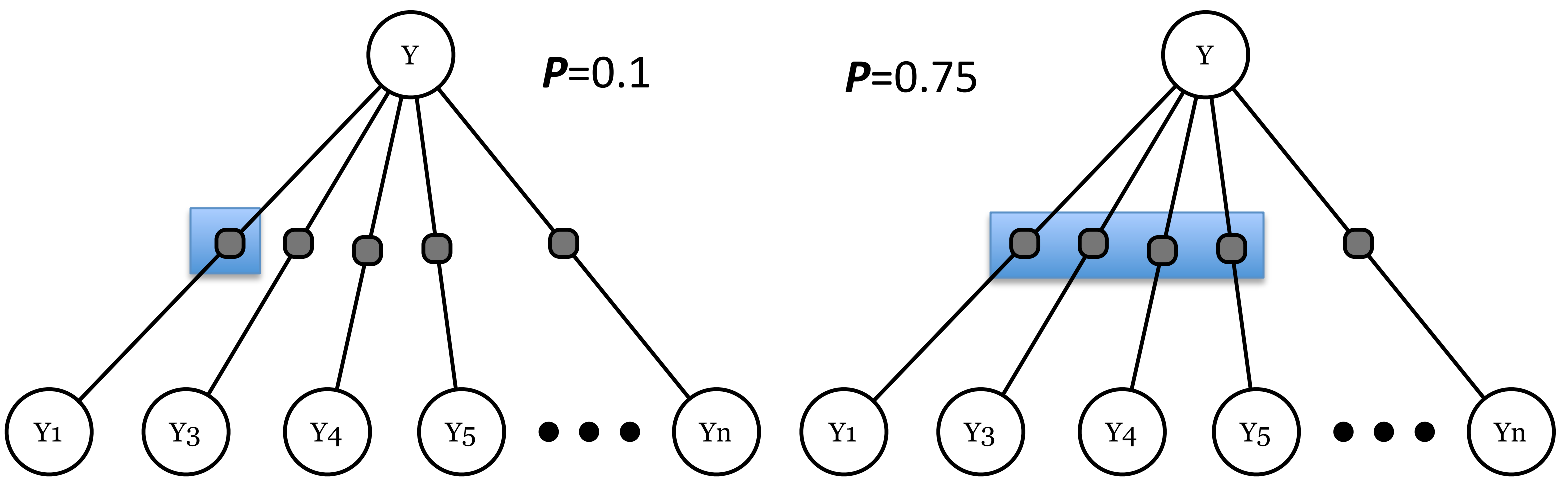
- Estimate the scores by sub-sampling the factors:

$$\mathbf{S} \subseteq \mathbf{F}';\ \psi_{\mathbf{S}}(\mathbf{y}/\mathbf{y}') = |\mathbf{F}'|\mathbb{E}_{\mathbf{S}} f(\mathbf{y})$$



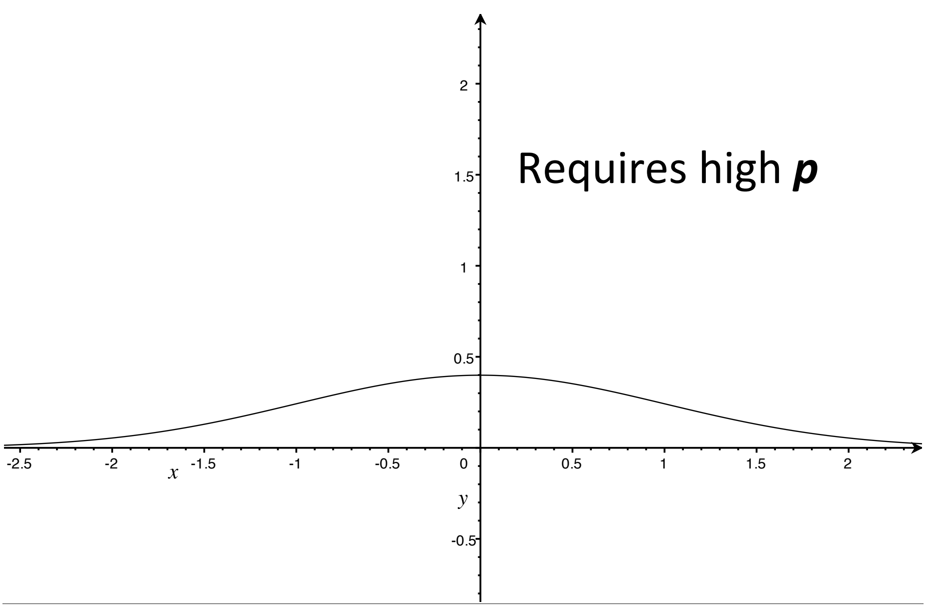
# Uniform Sampling

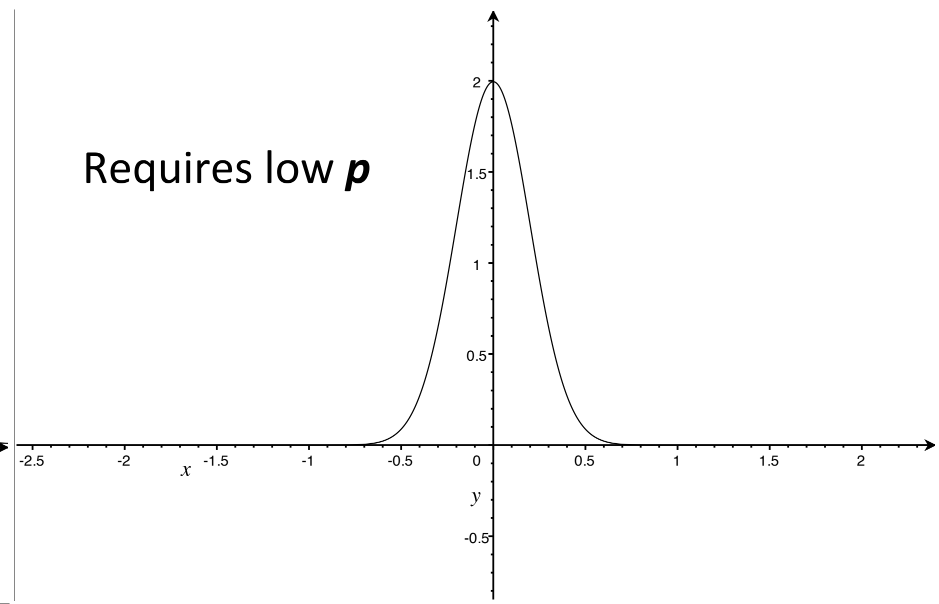
- Pick the subset **S** uniformly
  - Proportion of factors to pick is ***p***
- Scoring is ***1/p*** times faster
  - But with lower ***p***, more samples are needed



# Limitations of Uniform Sampling

- Performance is sensitive to parameter ***p***
  - Which has to be manually specified
- Different proposals may prefer different ***p***'s
  - Depends on the variance of the factor scores

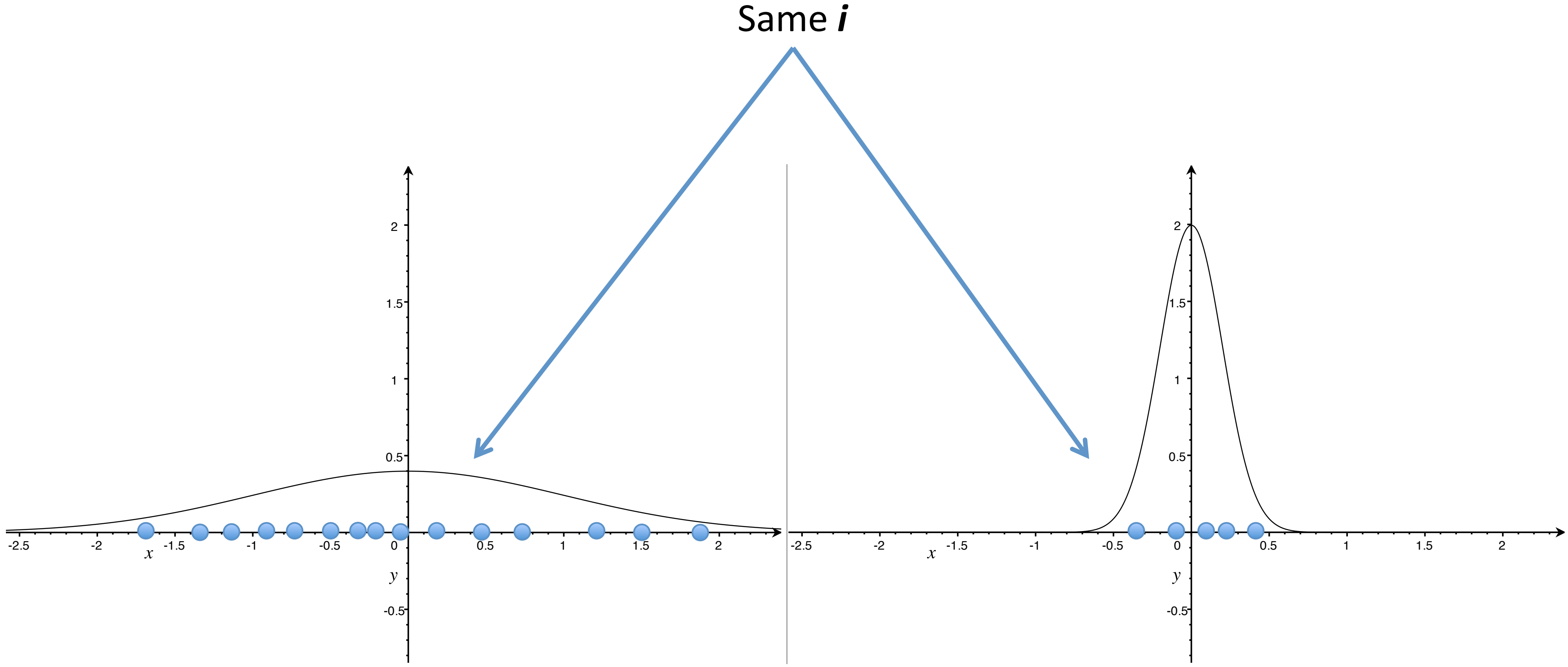


# Confidence-Based Stopping

- Sample uniformly as before
  - Compute 95% confidence interval around mean
- We want to sample till reasonably confident
  - If, width of interval < ***i***, stop.
  - Else, continue sampling
- Need to include finite population control (fpc)
  - Since ***S*** is a substantial subset of ***F'***

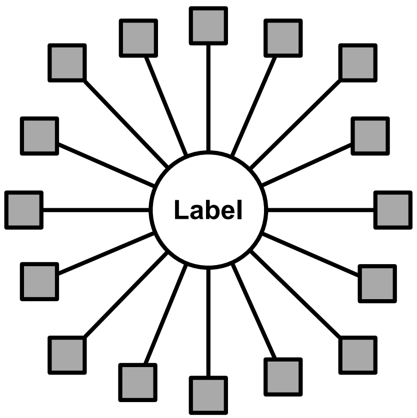
# Confidence-Based Stopping

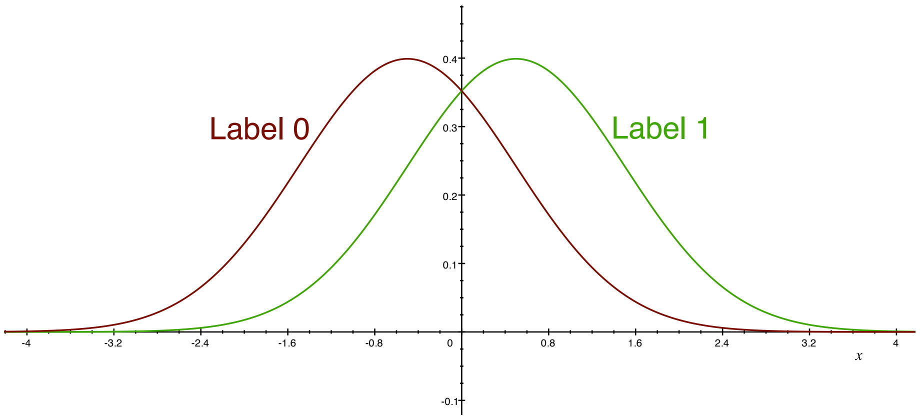


# Experiments

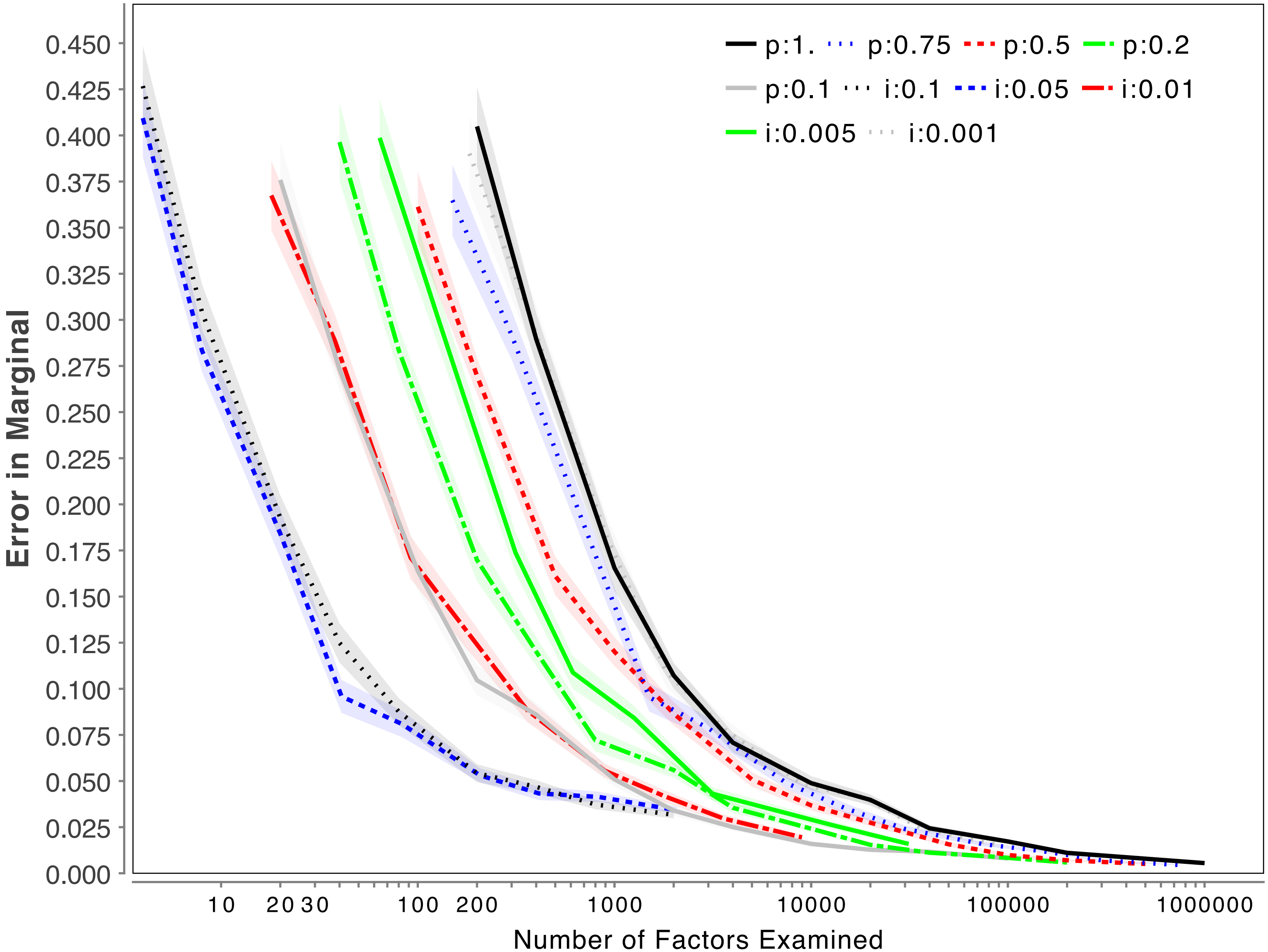
# Synthetic Data

- Binary Classification Model
  - 100 factors
- Generate Samples
  - Compute marginals from them
  - Compare error to exact
- Similar operation as Gibbs
  - Ignore Burn-in and Thinning



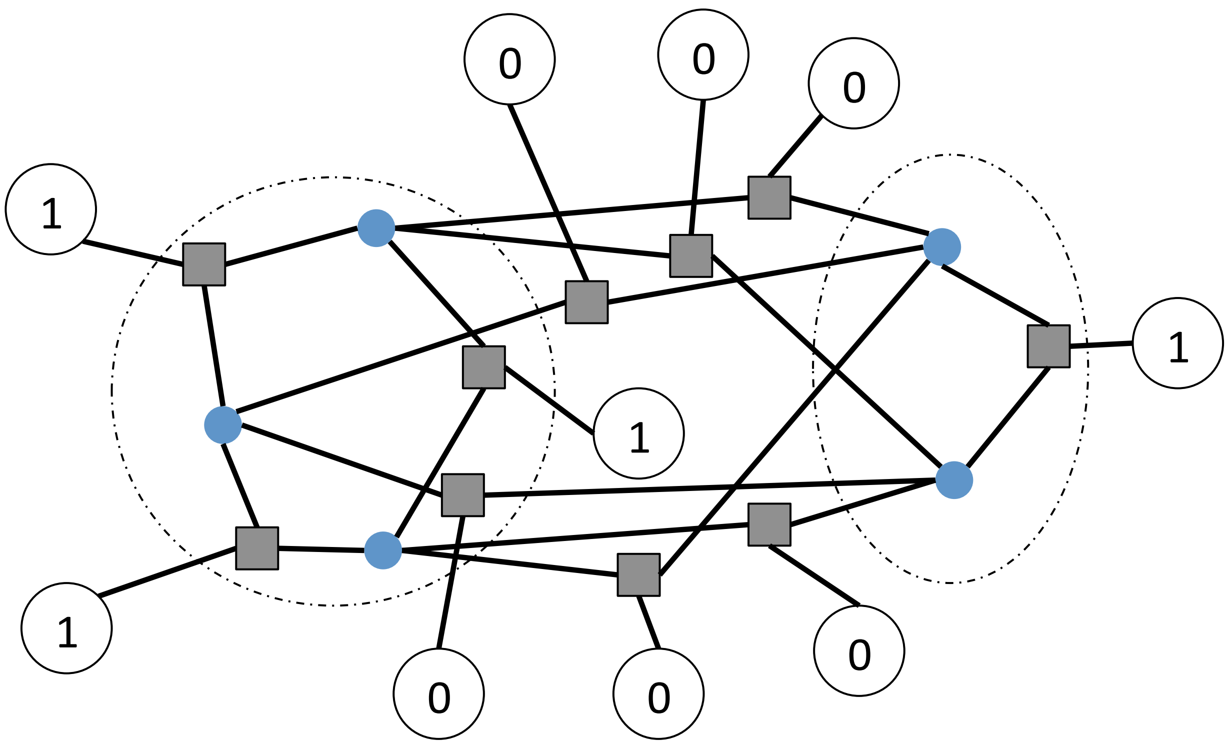



# Synthetic Data



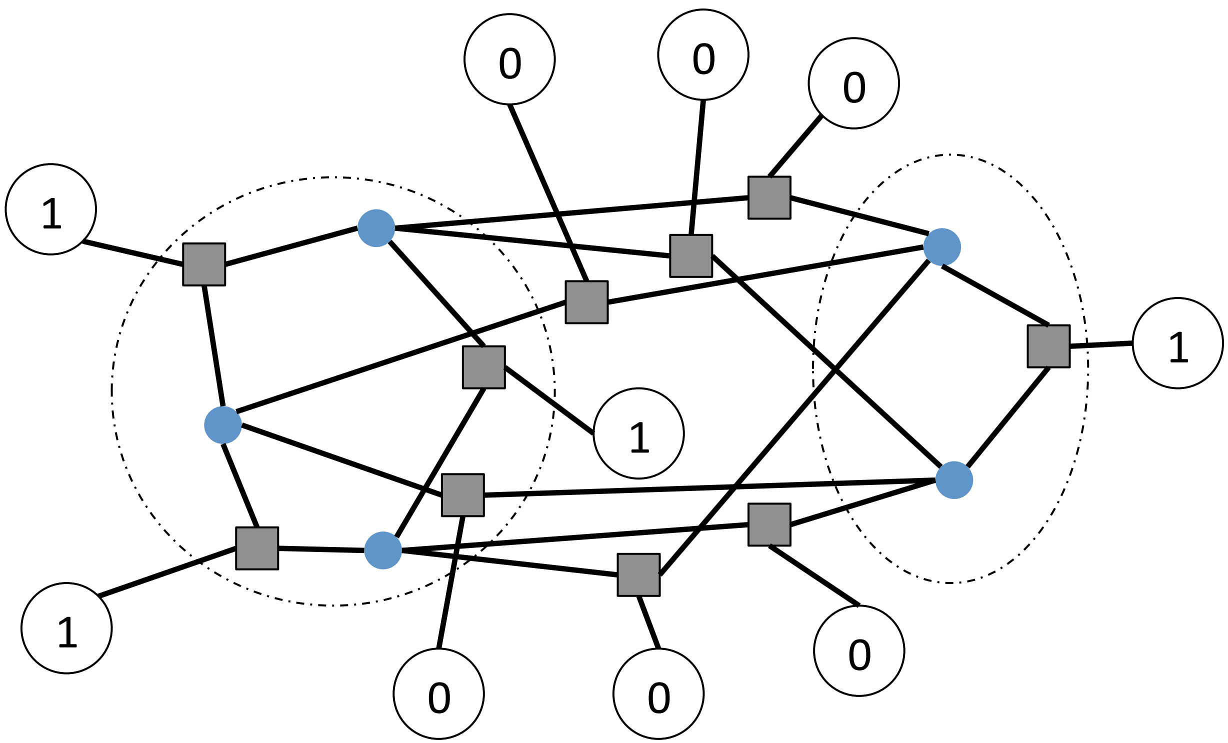
# Entity Resolution Model

- Or Clustering…
- Used for Entity Disambiguation, Coreference Resolution, Record De-duplication, etc.



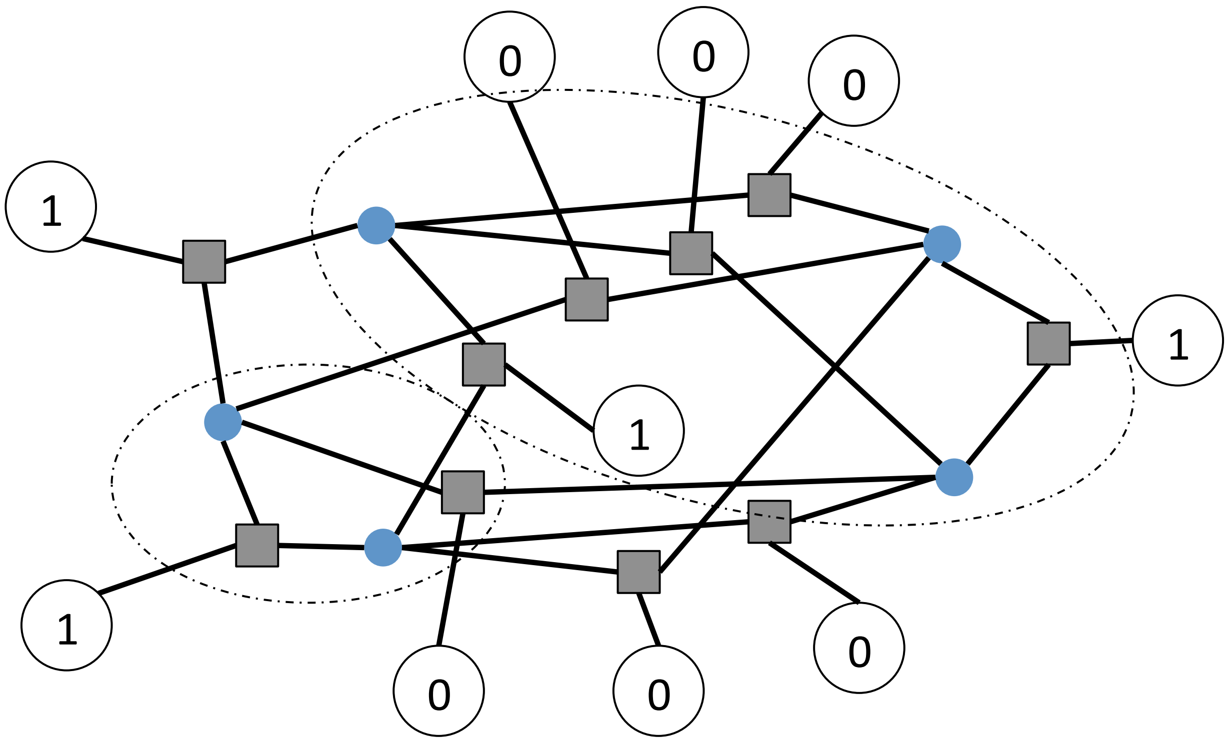
# MCMC for Entity Resolution

- Initialize to any valid configuration



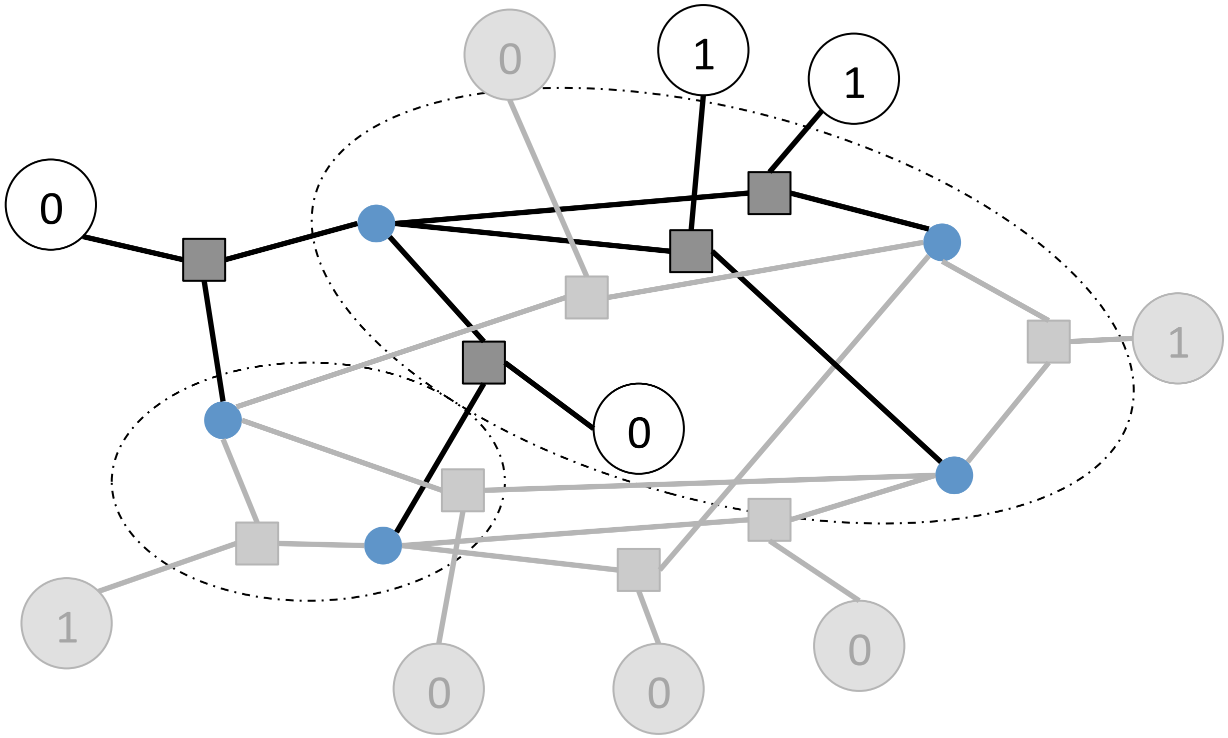
# MCMC for Entity Resolution

- Proposal moves a single data point..



# MCMC for Entity Resolution

- Score factors that neighbor the moved point
  - And the points in the old and new clusters



# MCMC for Entity Resolution

- Pros:
  - Allows us to enforce transitivity implicitly
  - May not compare all pairs of points
  - Scoring a proposal is linear in cluster size
- Cons:
  - Scoring a proposal is linear in cluster size!!!

  (Fortunately, points in a cluster are redundant)

# Cora Citation Matching

- 1295 citation strings that refer to 134 papers

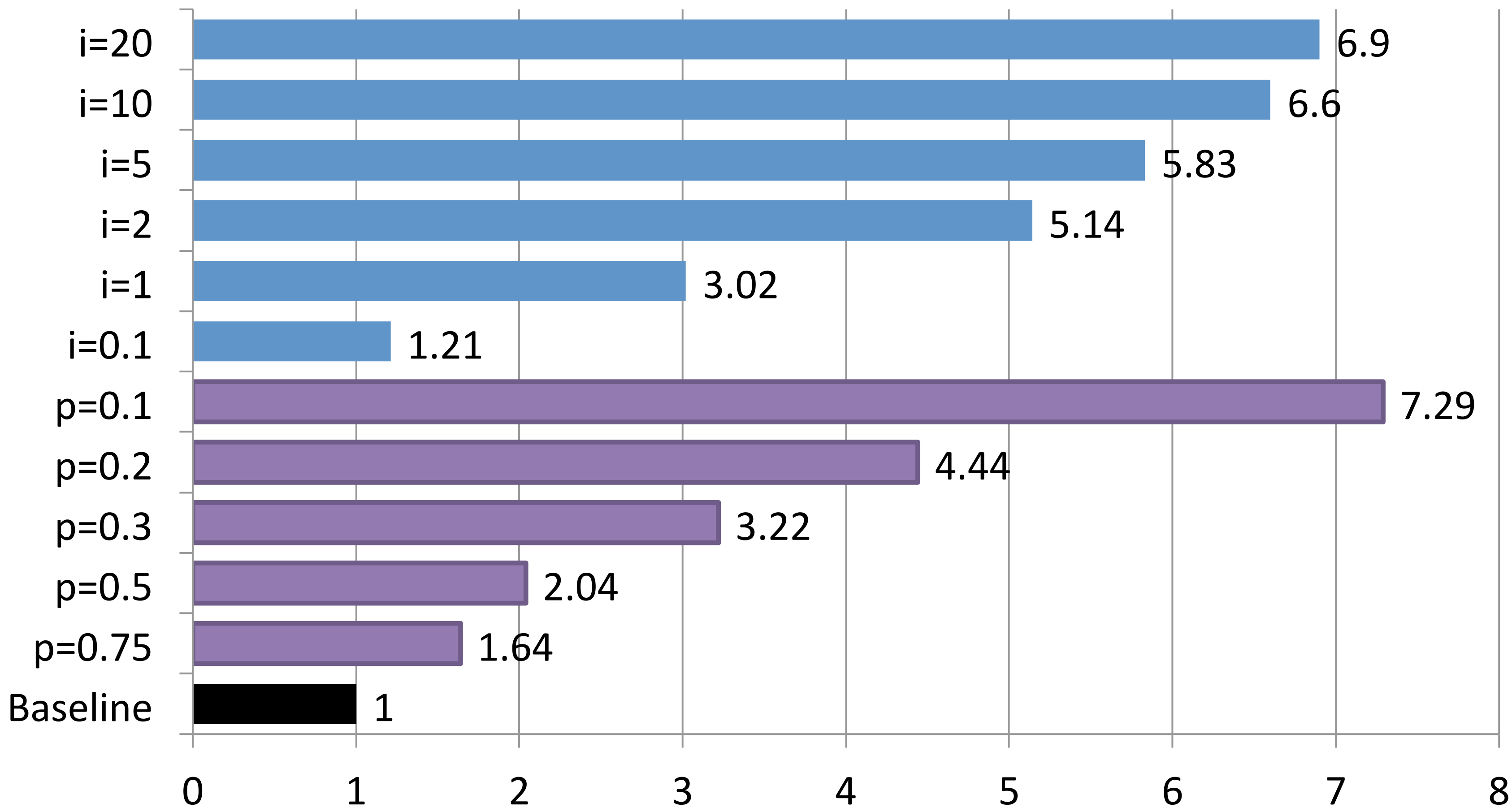
Yoav Freund, H. Sebastian Seung, Eli Shamir, Naftali Tishby. Information, prediction, and query by committee, NIPS92, p. 1993 483-490

Y. freund, H.S seung, E. shamir, and N. tishby. Accelerating learning using query by Committee. Proceedings of the 1992 conf. on neural informations processing systems (to appear), 1993

< 10 citations per paper on average

- Use features based on similarity of fields
  – Author, Title and Venue

# Speedup to obtain 90% $B^3$



# Large-Scale Author Coreference

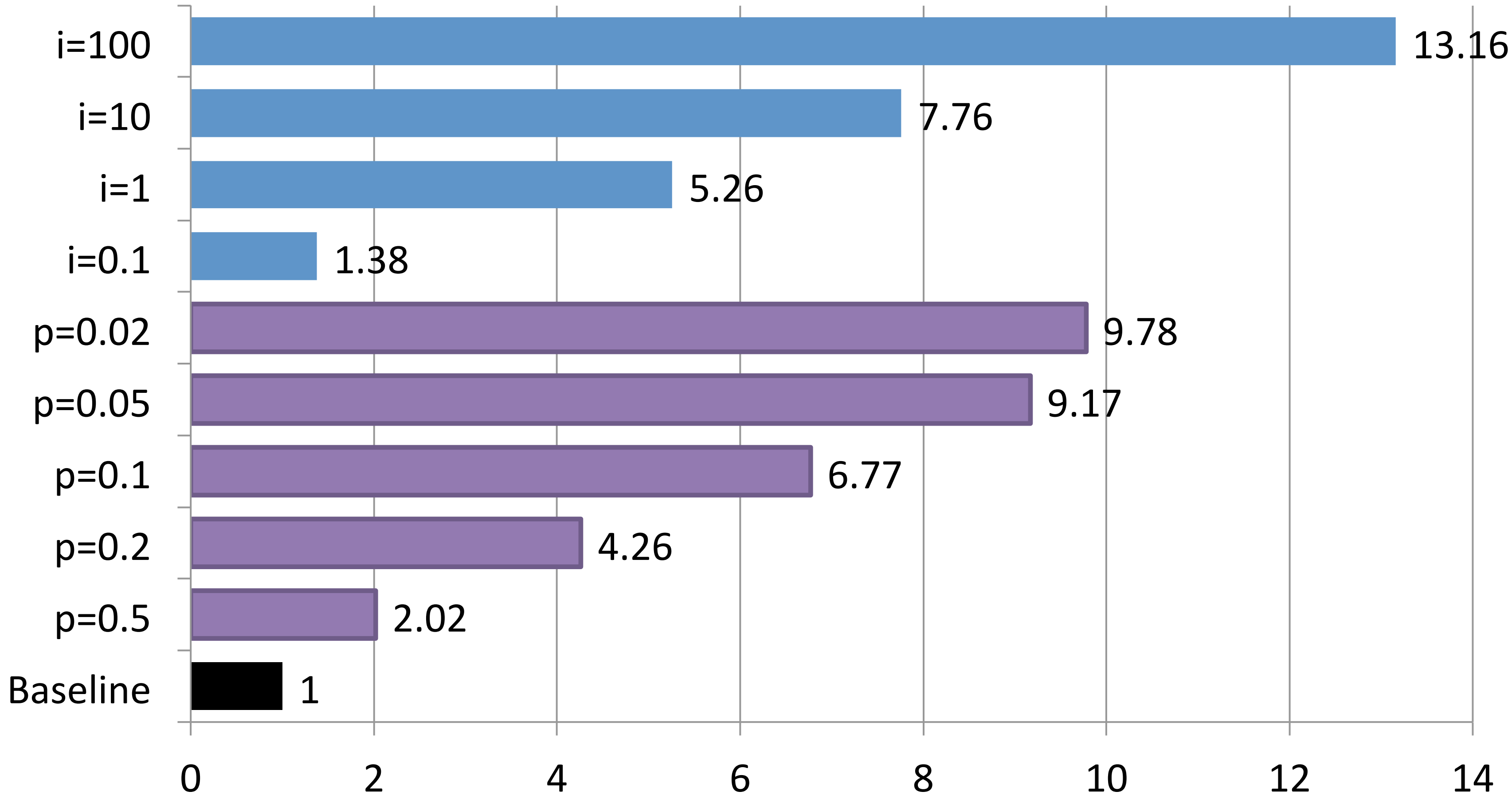
- 5 million authors from DBLP BibTex entries

*@techreport*{
author= S. Palacharia, N.P.Jouppi, **J.E.Smith**,
title= Quantifying the complexity of superscalar processors
institution= University of Wisconsin, year=1996}

@inproceedings{
author= Aggarwal, Ranganathan, Jouppi, and **Smith**,
title= Building High Availability Systems with Commodity Processors,
booktitle=Int. Symposium on Computer Architecture, year=2007}

- Include 2,833 **labeled** mentions from Rexa
- Use BibTex context as the features
  - First/last names, title BOW, title topics, coauthors

# Speedup to obtain 80% $B^3$



# Limitations and Future Work

1. Is fairly naïve about factor selection
   – Assumes factors are distributed normally
   – Does not (re)use factor scores
   – **Future**: Score-aware factor selection
2. Theoretical Issues
   – Unwanted bias in the samples, introduces error
   – **Future**: Reweight samples to remove the bias
3. Dynamic Threshold
   – *Ideal* threshold may depend on the state of inference
   – **Future**: Reduce approximation as inference proceeds
4. Evaluate on more tasks

# Summary

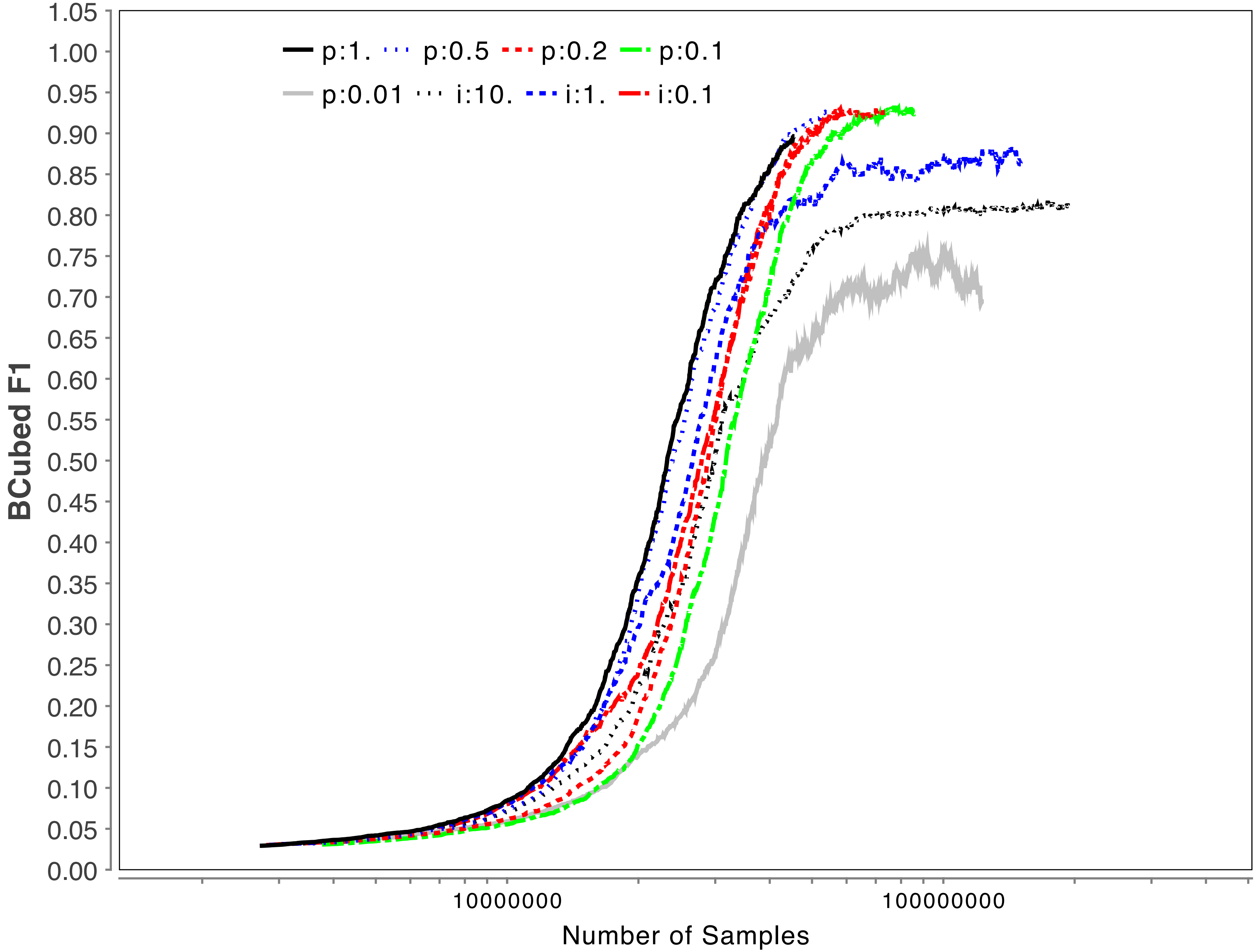
- Examined scenarios where MCMC is slow
- Proposed stochastic evaluation of samples
  - Uniform Sampling
  - Confidence-Based Sampling
- Demonstrated significant speedups
  - For marginal inference on synthetic data
  - Up to 13x speedup on large-scale entity resolution
- Approach is general and easy to code

# Thanks!

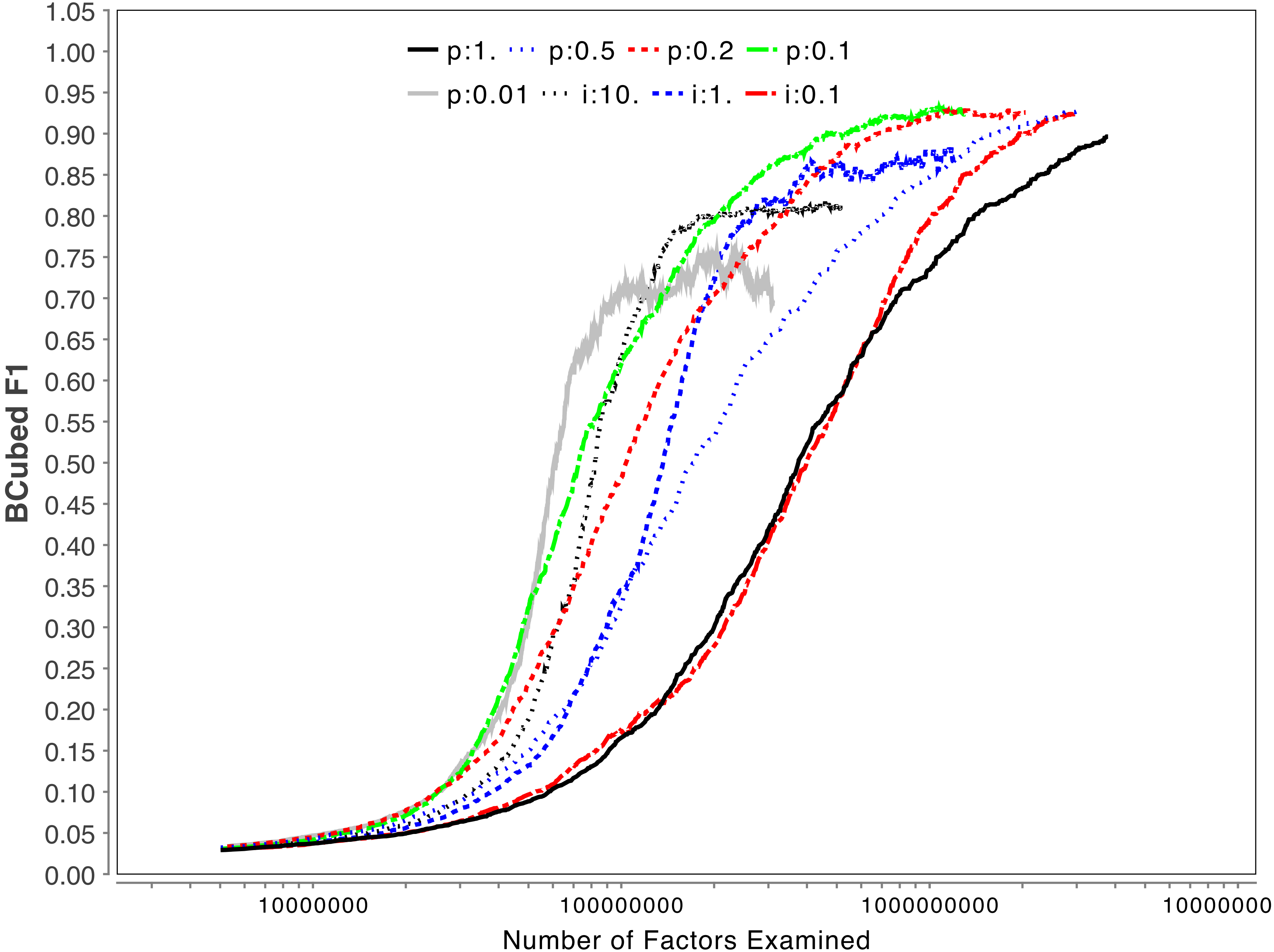
**Sameer Singh**, Michael Wick, Andrew McCallum

sameer@cs.umass.edu

# Appendix



p:1.
p:0.5
p:0.2
p:0.1
p:0.01
i:10.
i:1.
i:0.1
BCubed F1
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
10000000
100000000
Number of Samples



p:1.
p:0.5
p:0.2
p:0.1
p:0.01
i:10.
i:1.
i:0.1
BCubed F1
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
10000000
100000000
1000000000
Number of Factors Examined