Monte Carlo MCMC
Efficient Inference by Approximate Sampling

Sameer Singh, Michael Wick, Andrew McCallum
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:
  \[ \psi(Y=y) = \sum_{f \in F} f(y_f) \]
- Probability:
  \[ p(Y=y) = \frac{1}{Z} \exp \psi(y) \]
Markov Chain Monte Carlo

1. Current Sample, \( y \)
2. Propose a move: \( y \rightarrow y' \)
3. Accept with Probability \( \alpha \)

\[
\alpha(y, y') = \frac{p(y')}{p(y)} = \exp \psi(y') - \psi(y) = \exp \psi(y'/y) - \psi(y/y')
\]

4. Current sample \( \leftarrow y' \)
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
  \[ \alpha(y, y') = \exp \left( \psi(y'/y) - \psi(y/y') \right) \]
  \[ \psi(y/y') = \sum_{f \in F'} f(y) = |F'| \mathbb{E}_{F', f(y)} \]

- Estimate the scores by sub-sampling the factors:
  \[ S \subseteq F'; \quad \psi_S(y/y') = |F'| \mathbb{E}_S f(y) \]
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

Same $i$
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

![Graph showing the error in marginal distribution with varying number of factors examined. The x-axis represents the number of factors examined, ranging from 10 to 1,000,000, and the y-axis represents the error in marginal distribution, ranging from 0.000 to 0.450. Different lines represent different error rates, with labels indicating the error rate such as p:1, p:0.75, p:0.5, p:0.2, p:0.1, i:0.1, i:0.05, i:0.01, i:0.005, i:0.001.](image)
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


< 10 citations per paper on average

• Use features based on similarity of fields
  – Author, Title and Venue
Speedup to obtain 90% $B^3$

Baseline

- $p=0.75$: 1.64
- $p=0.5$: 2.04
- $p=0.3$: 3.22
- $p=0.2$: 4.44
- $p=0.1$: 7.29

- $i=20$: 6.9
- $i=10$: 6.6
- $i=5$: 5.83
- $i=2$: 5.14
- $i=1$: 3.02
- $i=0.1$: 1.21
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% $B^3$

- $i=100$: 13.16
- $i=10$: 7.76
- $i=1$: 5.26
- $i=0.1$: 1.38
- $p=0.02$: 9.78
- $p=0.05$: 9.17
- $p=0.1$: 6.77
- $p=0.2$: 4.26
- $p=0.5$: 2.02
- Baseline: 1
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
Number of Samples

BCubed F1

Number of Samples