Adaptive Importance Sampling for Network Growth Models

Adam Guetz

(joint work with Susan Holmes)

guetz@stanford.edu

Stanford University, Stanford CA

Problem Setting

- Problem – Compute $\mathrm{E}_f[h(\sigma)]$, where h is a non-negative function and $\sigma \sim f$, a distribution on the set of permutations S_n.

Problem Setting

- Problem – Compute $\mathrm{E}_f[h(\sigma)]$, where h is a non-negative function and $\sigma \sim f$, a distribution on the set of permutations S_n.
  - In our applications, f is usually the uniform distribution and h is the likelihood function $L_\phi(\sigma|D)$ for a Network Growth Model ϕ with dataset D.

Problem Setting

- Problem – Compute $\mathrm{E}_f[h(\sigma)]$, where h is a non-negative function and $\sigma \sim f$, a distribution on the set of permutations S_n.
  - In our applications, f is usually the uniform distribution and h is the likelihood function $L_\phi(\sigma|D)$ for a Network Growth Model ϕ with dataset D.
- $\mathrm{E}_f[h(\sigma)]$ may be 'dominated' by a subset of states with exponentially small measure in f ('rare events'). Crude Monte Carlo estimator based on samples from f does not work well.

Problem Setting

- Problem – Compute $\mathrm{E}_f[h(\sigma)]$, where h is a non-negative function and $\sigma \sim f$, a distribution on the set of permutations S_n.
  - In our applications, f is usually the uniform distribution and h is the likelihood function $L_\phi(\sigma|D)$ for a Network Growth Model ϕ with dataset D.
- $\mathrm{E}_f[h(\sigma)]$ may be 'dominated' by a subset of states with exponentially small measure in f ('rare events'). Crude Monte Carlo estimator based on samples from f does not work well.
- Idea – Use Importance Sampling to build lower variance estimator.

In This Talk

- Use of Adaptive Importance Sampling (AdIS) for Network Growth Models (NGM).

In This Talk

- Use of Adaptive Importance Sampling (AdIS) for Network Growth Models (NGM).
- Introduce Plackett-Luce (PL) model as family of proposal distributions.

In This Talk

- Use of Adaptive Importance Sampling (AdIS) for Network Growth Models (NGM).
- Introduce Plackett-Luce (PL) model as family of proposal distributions.
- Addressing degeneracy of AdIS with Minimum Description Length (MDL).

In This Talk

- Use of Adaptive Importance Sampling (AdIS) for Network Growth Models (NGM).
- Introduce Plackett-Luce (PL) model as family of proposal distributions.
- Addressing degeneracy of AdIS with Minimum Description Length (MDL).
- Analysis of *Mus Musculus* Protein-Protein Interaction (PPI) network.

Motivation

- Applications:

Motivation

- Applications:
  - Statistical inference for network data.

Motivation

- Applications:
  - Statistical inference for network data.
  - Likelihood computation, Model selection

Motivation

- Applications:
  - Statistical inference for network data.
  - Likelihood computation, Model selection
    - How well does the network model fit the data?

Motivation

- Applications:
  - Statistical inference for network data.
  - Likelihood computation, Model selection
    - How well does the network model fit the data?
  - Estimation of normalizing constants/partition functions for distributions on permutations.

Motivation

- Applications:
  - Statistical inference for network data.
  - Likelihood computation, Model selection
    - How well does the network model fit the data?
  - Estimation of normalizing constants/partition functions for distributions on permutations.
  - Approximate counting.

Motivation

- Applications:
  - Statistical inference for network data.
  - Likelihood computation, Model selection
    - How well does the network model fit the data?
  - Estimation of normalizing constants/partition functions for distributions on permutations.
  - Approximate counting.
  - Rare event simulation.

Models of Network Growth

- Defined by two rules:
  - Networks are grown one vertex at a time.
  - New edges are attached from new vertex to (possibly empty) set of pre-existing vertices.

Models of Network Growth

- Defined by two rules:
  - Networks are grown one vertex at a time.
  - New edges are attached from new vertex to (possibly empty) set of pre-existing vertices.
- Commonly used to model phenomena from biology, computer science, and sociology.

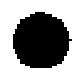
Models of Network Growth

- Defined by two rules:
  - Networks are grown one vertex at a time.
  - New edges are attached from new vertex to (possibly empty) set of pre-existing vertices.
- Commonly used to model phenomena from biology, computer science, and sociology.
- $L(G|\sigma)$ usually easy to compute.
  - G: network data.
  - σ: vertex labeling/permutation.

Models of Network Growth

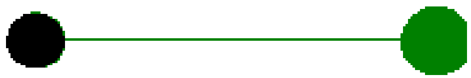
- Defined by two rules:
  - Networks are grown one vertex at a time.
  - New edges are attached from new vertex to (possibly empty) set of pre-existing vertices.
- Commonly used to model phenomena from biology, computer science, and sociology.
- $L(G|\sigma)$ usually easy to compute.
  - G: network data.
  - σ: vertex labeling/permutation.
- Examples:
  - Preferential Attachment (PA).
  - Duplication/Divergence (DD) (Vertex Copying).
  - Kronecker Delta Product Graphs.

Network Growth Model



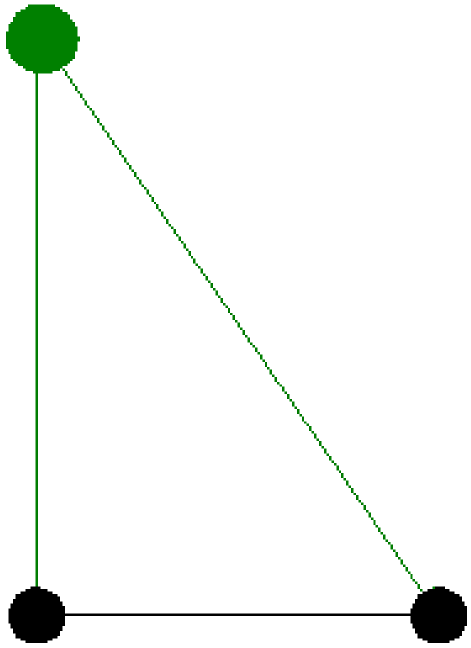
Network Growth Model

Network Growth Model

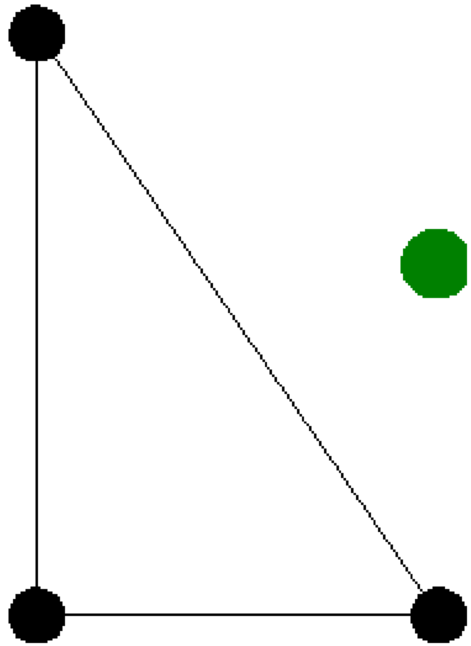


Network Growth Model

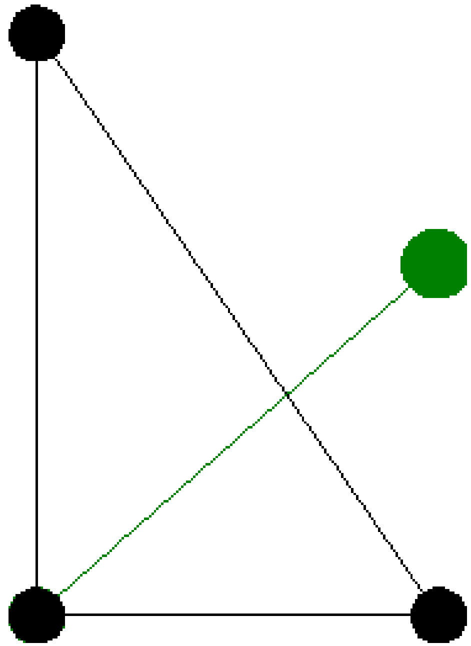
Network Growth Model



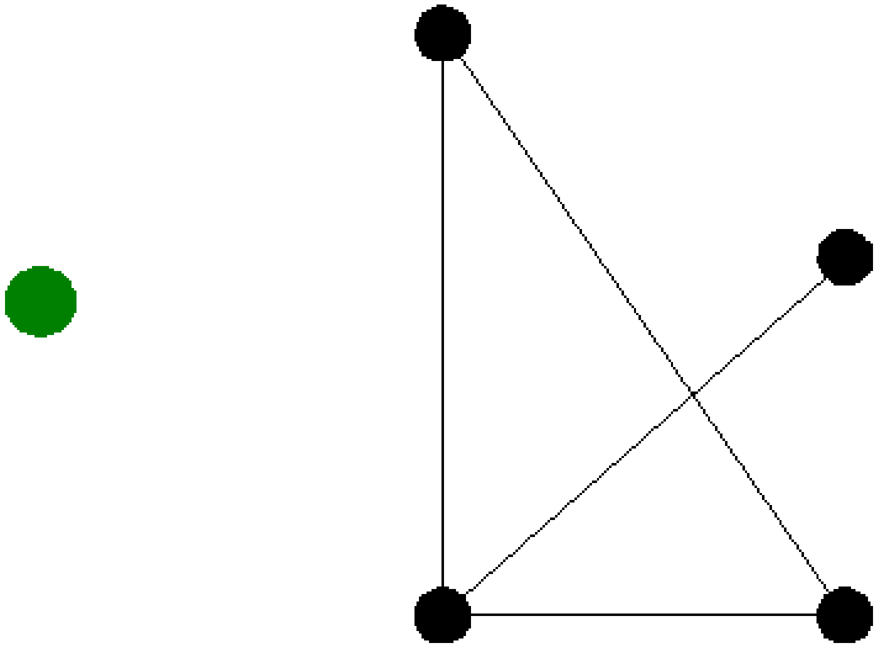
Network Growth Model



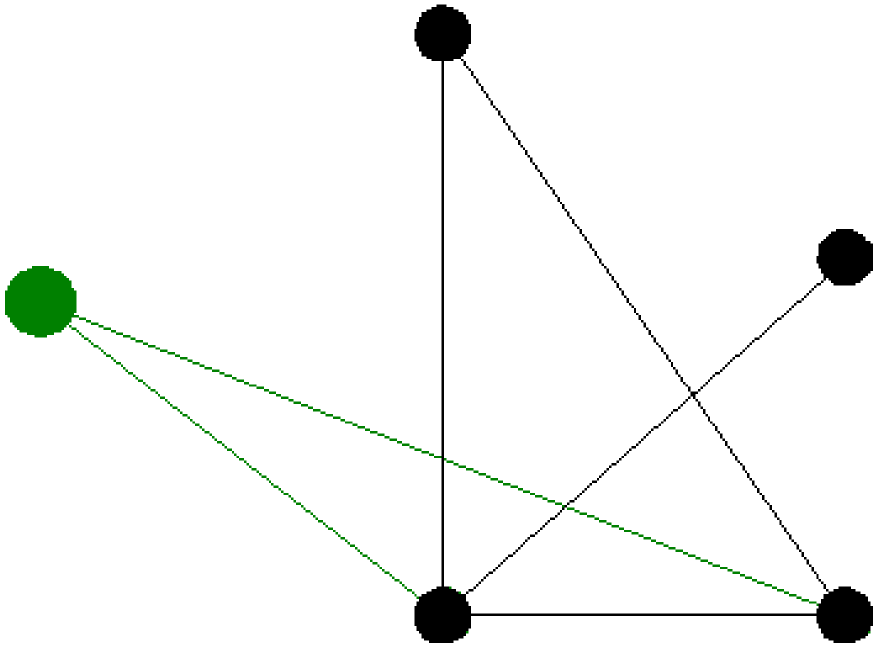
Network Growth Model



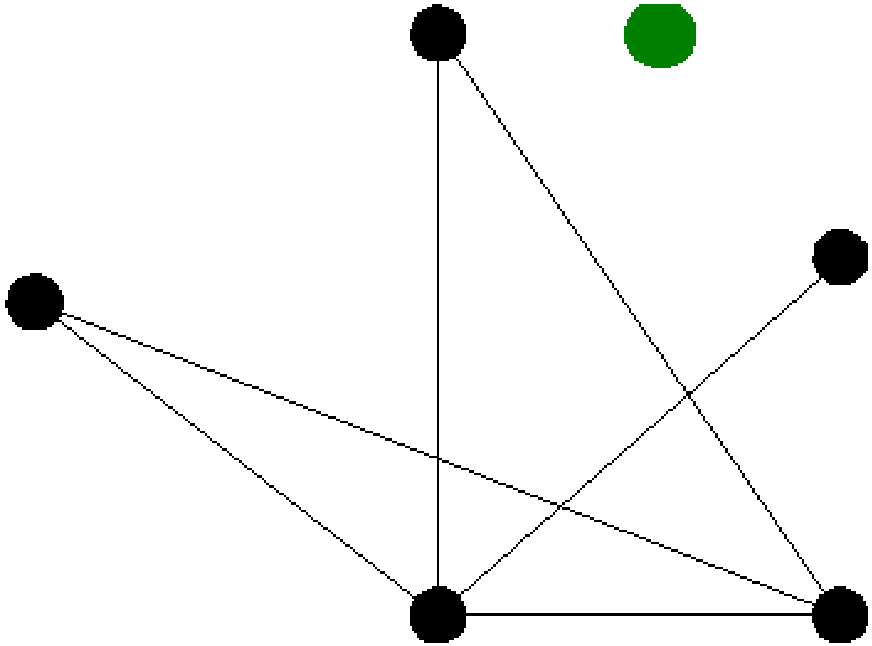
Network Growth Model



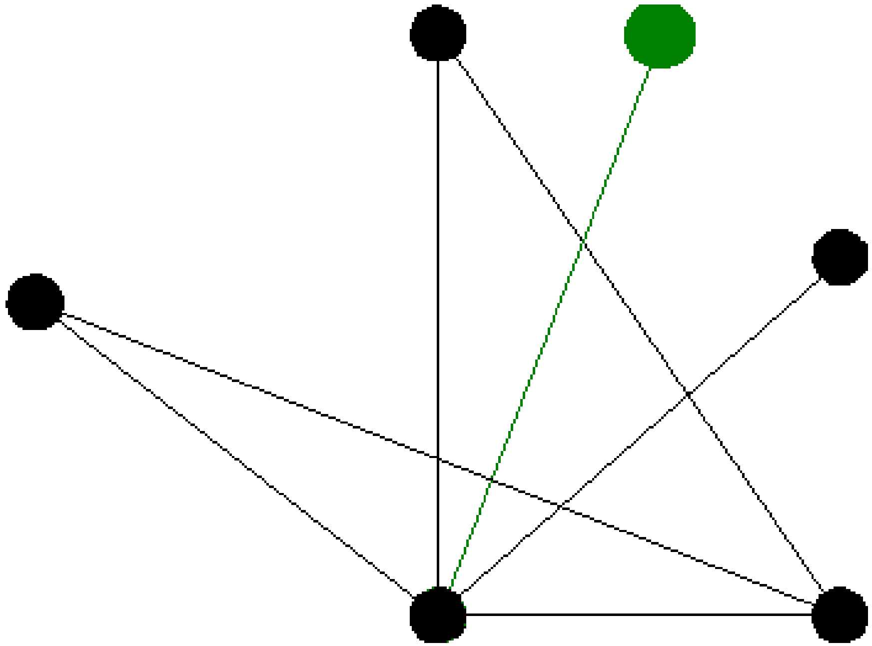
Network Growth Model



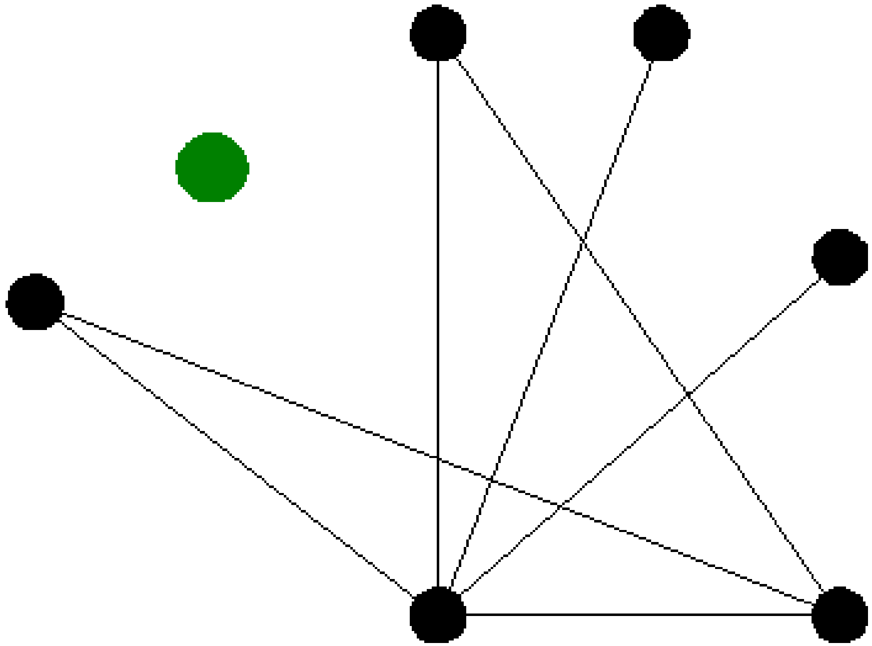
Network Growth Model



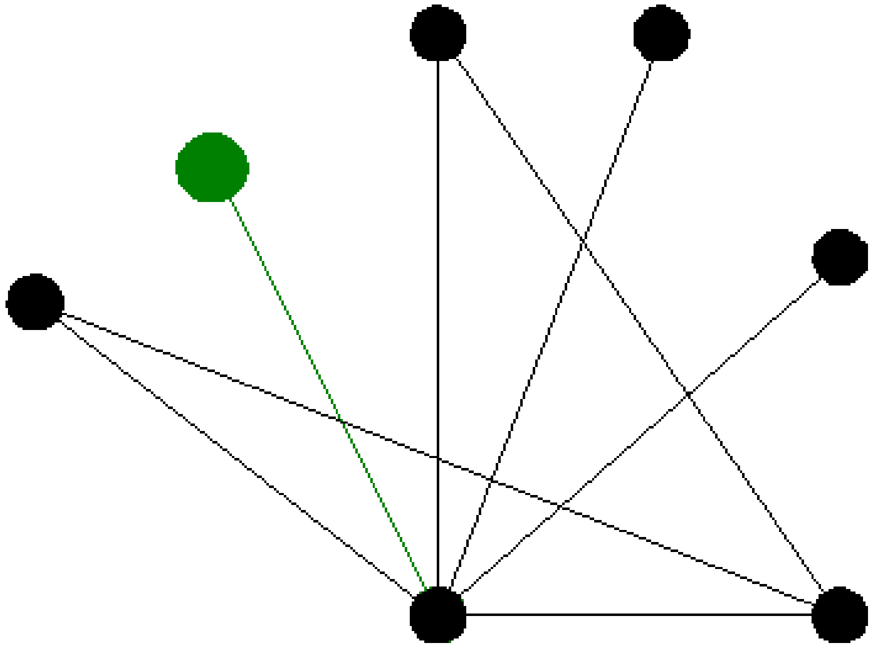
Network Growth Model



Network Growth Model



Network Growth Model



Network Growth

- We assume networks to be undirected, simple, without self-loops

Network Growth

- We assume networks to be undirected, simple, without self-loops
- Order in which vertices appear important – NGM are inherently models for labeled graphs

Network Growth

- We assume networks to be undirected, simple, without self-loops
- Order in which vertices appear important – NGM are inherently models for labeled graphs
- Many network datasets are unlabeled – age of vertices unknown or uncertain.

Network Growth

- We assume networks to be undirected, simple, without self-loops
- Order in which vertices appear important – NGM are inherently models for labeled graphs
- Many network datasets are unlabeled – age of vertices unknown or uncertain.
  - To model with NGM we must sum over all possible labelings.

Network Growth

- We assume networks to be undirected, simple, without self-loops
- Order in which vertices appear important – NGM are inherently models for labeled graphs
- Many network datasets are unlabeled – age of vertices unknown or uncertain.
  - To model with NGM we must sum over all possible labelings.
  - Infeasible – factorial number of permutations.

Network Growth

- We assume networks to be undirected, simple, without self-loops
- Order in which vertices appear important – NGM are inherently models for labeled graphs
- Many network datasets are unlabeled – age of vertices unknown or uncertain.
  - To model with NGM we must sum over all possible labelings.
  - Infeasible – factorial number of permutations.
- Use Adaptive Importance Sampling.

Adaptive Importance Sampling

- Uses IS identity: $\mathrm{E}_f\left[h(\sigma)\right] = \mathrm{E}_g\left[\frac{h(\sigma)f(\sigma)}{g(\sigma)}\right]$

Adaptive Importance Sampling

- Uses IS identity: $\mathrm{E}_f\left[h(\sigma)\right] = \mathrm{E}_g\left[\frac{h(\sigma)f(\sigma)}{g(\sigma)}\right]$
- Want to find g 'close' to the optimal min variance importance distribution:

$$g^*(x) \propto h(\sigma)f(\sigma)$$

Adaptive Importance Sampling

- Uses IS identity: $\mathrm{E}_f\left[h(\sigma)\right] = \mathrm{E}_g\left[\frac{h(\sigma)f(\sigma)}{g(\sigma)}\right]$
- Want to find g 'close' to the optimal min variance importance distribution:

$$g^*(x) \propto h(\sigma)f(\sigma)$$

- Need a family of proposal distributions $\mathcal{F}$ such that:
  - Likelihood (with normalizing constant) is easily computed.
  - MLE easy to find.
  - $\exists g \in \mathcal{F}$ that is 'close' to g^*.

Adaptive Importance Sampling (cont)

- Generic AdIS step:
  - Draw samples from current IS dist g_i.
  - Choose g_{i+1} to be 'best' $g \in \mathcal{F}$ according to previous samples and repeat.

Adaptive Importance Sampling (cont)

- Generic AdIS step:
  - Draw samples from current IS dist g_i.
  - Choose g_{i+1} to be 'best' $g \in \mathcal{F}$ according to previous samples and repeat.
- Commonly used frameworks for AdIS:
  - Cross-Entropy method
  - Variance Minimization
  - Population Monte Carlo

Cross-Entropy Method

- Use KL-divergence as “closeness” measure
  - corresponds to MLE estimator where each sample appears $w(\sigma)h(\sigma)$ times.

Cross-Entropy Method

- Use KL-divergence as “closeness” measure
  - corresponds to MLE estimator where each sample appears $w(\sigma)h(\sigma)$ times.
- Basic CE iteration:
  - Draw $[\sigma_i]_{1\dots n} \sim g_i$
  - Take $g_{i+1} = \mathrm{argmin}_g KL(g, [w(\sigma_i)g_i(\sigma_i)]_{1\dots n})$

Cross-Entropy Method

- Use KL-divergence as “closeness” measure
  - corresponds to MLE estimator where each sample appears $w(\sigma)h(\sigma)$ times.
- Basic CE iteration:
  - Draw $[\sigma_i]_{1...n} \sim g_i$
  - Take $g_{i+1} = \operatorname{argmin}_g KL(g, [w(\sigma_i)g_i(\sigma_i)]_{1...n})$
- Unfortunately, MLE may produce a ’degenerate’ importance distribution g_{i+1} if there aren’t enough samples.
  - Few samples dominate.
  - ’Entropy’ of distribution greatly decreases.
  - Importance weights blow up.

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.
- Samples expensive (score function is $O(n^2)$).

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.
- Samples expensive (score function is $O(n^2)$).
- MLE $\rightarrow$ degeneracy as 'overfitting'. Can use:

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.
- Samples expensive (score function is $O(n^2)$).
- MLE $\rightarrow$ degeneracy as 'overfitting'. Can use:
  - Cross-validation

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.
- Samples expensive (score function is $O(n^2)$).
- MLE $\rightarrow$ degeneracy as 'overfitting'. Can use:
  - Cross-validation
  - AIC, BIC.

Overcoming Degeneracy

- Common strategies to avoid degeneracy in CE-method:
  - 'elite' samples – take top ρ-percentile samples and weigh equally (min KL becomes MLE of elite sample).
  - take previous proposal distribution w/prob α, adjust sample sizes, other 'tuning' parameters.
- These techniques work well for some problems, but don't seem to work well for our applications with high dimensional parameter spaces.
- Samples expensive (score function is $O(n^2)$).
- MLE $\rightarrow$ degeneracy as 'overfitting'. Can use:
  - Cross-validation
  - AIC, BIC.
  - Bayesian priors.

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of 'Occam's Razor'

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of 'Occam's Razor'
  - Description length – tradeoff between fit and simplicity

$$L(v, \sigma) = L(v) - \log\left(\mathrm{P}(\sigma|V)\right) + const$$

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of ’Occam’s Razor’
  - Description length – tradeoff between fit and simplicity

$$L(v, \sigma) = L(v) - \log\left(\mathrm{P}(\sigma|V)\right) + const$$

    - # bits needed to describe model

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of 'Occam's Razor'
  - Description length – tradeoff between fit and simplicity

$$L(v,\sigma) = L(v) - \log\left(\mathrm{P}(\sigma|V)\right) + const$$

    - # bits needed to describe model
    - # bits needed to describe data under model

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of 'Occam's Razor'
  - Description length – tradeoff between fit and simplicity

$$L(v, \sigma) = L(v) - \log\left(\mathrm{P}(\sigma|V)\right) + const$$

    - # bits needed to describe model
    - # bits needed to describe data under model
  - We compute first term as negative model "entropy", other interpretations possible.

Minimum Description Length

- Minimum Description Length (MDL) – robust, information theoretic approach to model selection.
- MDL principle – generalization of 'Occam's Razor'
  - Description length – tradeoff between fit and simplicity

$$L(v, \sigma) = L(v) - \log\left(\mathrm{P}(\sigma|V)\right) + const$$

    - # bits needed to describe model
    - # bits needed to describe data under model
  - We compute first term as negative model "entropy", other interpretations possible.
  - "Small sample" correction – second term dominates for N large and become same as MLE

AdIS with MDL

- Sample size correction enables one to take fewer samples per iteration without encountering degeneracy.
  - More frequent,dynamic updating of proposal.

AdIS with MDL

- Sample size correction enables one to take fewer samples per iteration without encountering degeneracy.
  - More frequent,dynamic updating of proposal.
- Other modifications to AdIS:
  - Reuse old samples, increasing elite sample size as needed.

AdIS with MDL

- Sample size correction enables one to take fewer samples per iteration without encountering degeneracy.
  - More frequent,dynamic updating of proposal.
- Other modifications to AdIS:
  - Reuse old samples, increasing elite sample size as needed.
- CE-MDL algorithm:
  - Draw N samples from $[\sigma_j]_{1...N} \sim g_i$.
  - Compute $[h(\sigma_j)g_i(\sigma_j)]_{1...N}$; take ρ-elite sample.
  - Compute g_{i+1} as best MDL for elite sample.

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n + 1$ vertices with according to stochastic matrix.

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n+1$ vertices with according to stochastic matrix.
  - Quite general, but MLE estimator doesn't generalize for ranking data as only pairwise transitions are considered.

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n+1$ vertices with according to stochastic matrix.
  - Quite general, but MLE estimator doesn't generalize for ranking data as only pairwise transitions are considered.
  - $O(n^2)$ parameters

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n+1$ vertices with according to stochastic matrix.
  - Quite general, but MLE estimator doesn't generalize for ranking data as only pairwise transitions are considered.
  - $O(n^2)$ parameters
- Mallow's model – exponential family

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n+1$ vertices with according to stochastic matrix.
  - Quite general, but MLE estimator doesn't generalize for ranking data as only pairwise transitions are considered.
  - $O(n^2)$ parameters
- Mallow's model – exponential family
- Thurstonian Models – orderings of multivariate normal

Models of Rank

- Stochastic Edge Network (SEN) Markov model [Rubinstein, Kroese]
  - Picks a random Hamiltonian path in network with $n+1$ vertices with according to stochastic matrix.
  - Quite general, but MLE estimator doesn't generalize for ranking data as only pairwise transitions are considered.
  - $O(n^2)$ parameters
- Mallow's model – exponential family
- Thurstonian Models – orderings of multivariate normal
- Need Monte Carlo to compute likelihoods for both models

Proposal Family: Plackett-Luce

- Plackett-Luce Model

Proposal Family: Plackett-Luce

- Plackett-Luce Model
  - 'Urn' model.

Proposal Family: Plackett-Luce

- Plackett-Luce Model
  - 'Urn' model.
  - Each item has weight θ_i.

Proposal Family: Plackett-Luce

- Plackett-Luce Model
  - 'Urn' model.
  - Each item has weight θ_i.
  - Draw items without replacement with prob. prop. to θ.

Proposal Family: Plackett-Luce

- Plackett-Luce Model
  - 'Urn' model.
  - Each item has weight θ_i.
  - Draw items without replacement with prob. prop. to θ.
- Log-Likelihood easily computed as:

$$L(\sigma|\theta) = \sum_{i=1}^{n} \log(\theta_i) - \sum_{i=1}^{n} log \left(\sum_{j=i}^{n} \theta_{\sigma_j} \right)$$

Proposal Family: Plackett-Luce

- Plackett-Luce Model
  - 'Urn' model.
  - Each item has weight θ_i.
  - Draw items without replacement with prob. prop. to θ.
- Log-Likelihood easily computed as:

$$L(\sigma|\theta) = \sum_{i=1}^{n} \log(\theta_i) - \sum_{i=1}^{n} log\left(\sum_{j=i}^{n} \theta_{\sigma_j}\right)$$

- MLE efficiently found via deterministic majorization-minimization algorithm.

MDL and Plackett-Luce

- Computing MDL for PL model:

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]
  - Entropy of model is estimated efficiently thorugh Crude Monte Carlo.

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]
  - Entropy of model is estimated efficiently thorugh Crude Monte Carlo.
  - Heuristic univariate minimization works well in practice.

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]
  - Entropy of model is estimated efficiently thorugh Crude Monte Carlo.
  - Heuristic univariate minimization works well in practice.
- Potential problems with MDL interpretation:

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]
  - Entropy of model is estimated efficiently thorugh Crude Monte Carlo.
  - Heuristic univariate minimization works well in practice.
- Potential problems with MDL interpretation:
  - Not sampling from g^*, so not true model selection.

MDL and Plackett-Luce

- Computing MDL for PL model:
  - MDL is convex – exponential sum [Boyd and Vandenberghe 2004]
  - Entropy of model is estimated efficiently thorugh Crude Monte Carlo.
  - Heuristic univariate minimization works well in practice.
- Potential problems with MDL interpretation:
  - Not sampling from g^*, so not true model selection.
  - How much to weigh model complexity vs. fit not obvious. This is a tuning parameter.

Application: Preferential Attachment

- One of the best studied models to produce 'power-law' degree distributions.

Application: Preferential Attachment

- One of the best studied models to produce 'power-law' degree distributions.
- Yule-Simon model
  - Originally used to explain power-law frequency of word usage
  - Combination of 'Polya's Urn' processes.

Application: Preferential Attachment

- One of the best studied models to produce 'power-law' degree distributions.
- Yule-Simon model
  - Originally used to explain power-law frequency of word usage
  - Combination of 'Polya's Urn' processes.
- Barabási-Albert Model – Linear Preferential Attachment:
  - Rediscovered model in 1999 to explain internet graph.
  - Attach edges with probability (linearly) proportional to degree.
  - Add a fixed number of edges m at each step.
  - Showed that converges to a 'power-law' degree distribution with exponent 3.

Modeling Networks with PA

- For statistical applications, need PA model that

Modeling Networks with PA

- For statistical applications, need PA model that
  - is non-degenerate (with $p(G)$ bounded from 0 for $G \in \mathcal{G}_n$).

Modeling Networks with PA

- For statistical applications, need PA model that
  - is non-degenerate (with $p(G)$ bounded from 0 for $G \in \mathcal{G}_n$).
  - the likelihood given vertex ordering is easily to compute.

Modeling Networks with PA

- For statistical applications, need PA model that
  - is non-degenerate (with $p(G)$ bounded from 0 for $G \in \mathcal{G}_n$).
  - the likelihood given vertex ordering is easily to compute.
- Our PA model:

Modeling Networks with PA

- For statistical applications, need PA model that
  - is non-degenerate (with $p(G)$ bounded from 0 for $G \in \mathcal{G}_n$).
  - the likelihood given vertex ordering is easily to compute.
- Our PA model:
  - At step j add $Bin\left(\theta\binom{j}{2}\right)$ edges.

Modeling Networks with PA

- For statistical applications, need PA model that
  - is non-degenerate (with $p(G)$ bounded from 0 for $G \in \mathcal{G}_n$).
  - the likelihood given vertex ordering is easily to compute.
- Our PA model:
  - At step j add $Bin\left(\theta\binom{j}{2}\right)$ edges.
  - Edges added independently at random from new vertex v to old vertex w with probability proportional to

$$\frac{deg(i)}{\sum deg(l)}(1-\alpha)+\alpha$$

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0, 1]$ correspond to

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0, 1]$ correspond to
  - α – 'smoothing' parameter

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0, 1]$ correspond to
  - α – 'smoothing' parameter
    - $\alpha = 0$ is 'pure' preferential attachment

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0, 1]$ correspond to
  - α – ’smoothing’ parameter
    - $\alpha = 0$ is ’pure’ preferential attachment
    - $\alpha = 1$ is uniform attachment (Erdös-Rényi $G(n, p)$ model)

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0,1]$ correspond to
  - α – 'smoothing' parameter
    - $\alpha = 0$ is 'pure' preferential attachment
    - $\alpha = 1$ is uniform attachment (Erdös-Rényi $G(n,p)$ model)
  - θ – expected edge density, $\theta = \mathrm{E}[|edges(G)|]/\binom{n}{2}$.

Modeling Networks with PA (cont.)

- Parameters $\alpha, \theta \in [0, 1]$ correspond to
  - α – 'smoothing' parameter
    - $\alpha = 0$ is 'pure' preferential attachment
    - $\alpha = 1$ is uniform attachment (Erdös-Rényi $G(n, p)$ model)
  - θ – expected edge density, $\theta = \mathrm{E}[|edges(G)|]/\binom{n}{2}$.
- Similar to "Poisson Growth" model of Sheridan, Yagahara and Shimodaira [2008]. They show power-law degree distribution.

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :
  - Start a ’particle’ in known distibution.

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :
  - Start a ’particle’ in known distibution.
  - Move particle by sequence of Markov kernels f_i ending at distribution of interest.

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :
  - Start a 'particle' in known distibution.
  - Move particle by sequence of Markov kernels f_i ending at distribution of interest.
  - Compute at level t the ratio $W_t(\sigma_i) = \frac{f_{t+1}(\sigma_i)}{f_t(\sigma_i)}$

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :
  - Start a 'particle' in known distibution.
  - Move particle by sequence of Markov kernels f_i ending at distribution of interest.
  - Compute at level t the ratio $W_t(\sigma_i) = \frac{f_{t+1}(\sigma_i)}{f_t(\sigma_i)}$
  - Product $\prod_{i=1}^{\infty} W_i$ forms unbiased estimator of $\frac{Z_g}{Z_f}$

Annealed Importance Sampling

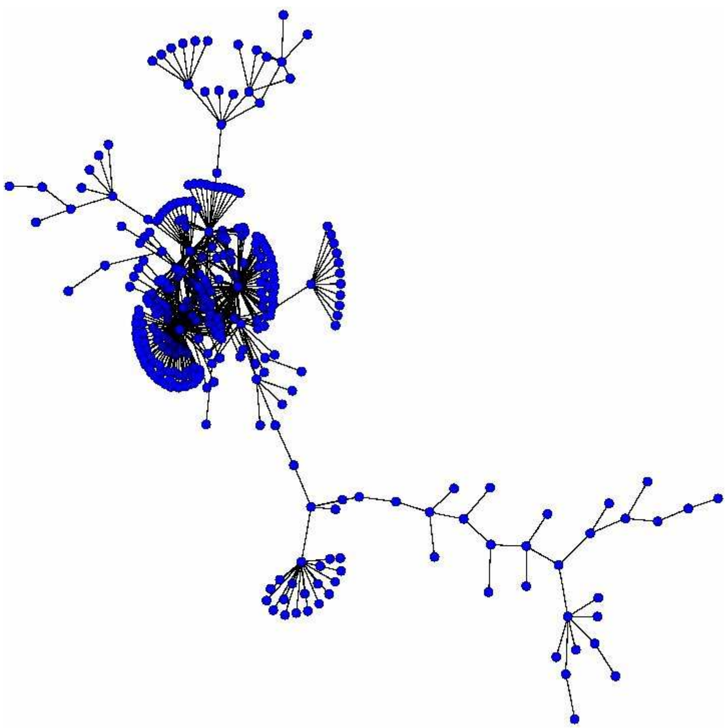
- Annealed Importance Sampling [Neal 2001] :
  - Start a 'particle' in known distibution.
  - Move particle by sequence of Markov kernels f_i ending at distribution of interest.
  - Compute at level t the ratio $W_t(\sigma_i) = \frac{f_{t+1}(\sigma_i)}{f_t(\sigma_i)}$
  - Product $\prod_{i=1}^{\infty} W_i$ forms unbiased estimator of $\frac{Z_g}{Z_f}$
- Essentially 'Umbrella Sampling' MCMC modified to produce an unbiased estimator.

Annealed Importance Sampling

- Annealed Importance Sampling [Neal 2001] :
  - Start a 'particle' in known distibution.
  - Move particle by sequence of Markov kernels f_i ending at distribution of interest.
  - Compute at level t the ratio $W_t(\sigma_i) = \frac{f_{t+1}(\sigma_i)}{f_t(\sigma_i)}$
  - Product $\prod_{i=1}^{\infty} W_i$ forms unbiased estimator of $\frac{Z_g}{Z_f}$
- Essentially 'Umbrella Sampling' MCMC modified to produce an unbiased estimator.
- Popular for applications in Physics, Chemistry, Biology.

Example: Mouse PPI Network

- Protein-Protein Interaction dataset for *Mus Musculus* (common mouse) from BioGRID (www.thebiogrid.org).
- Connected sub-network w/ 314 nodes and 503 interactions.

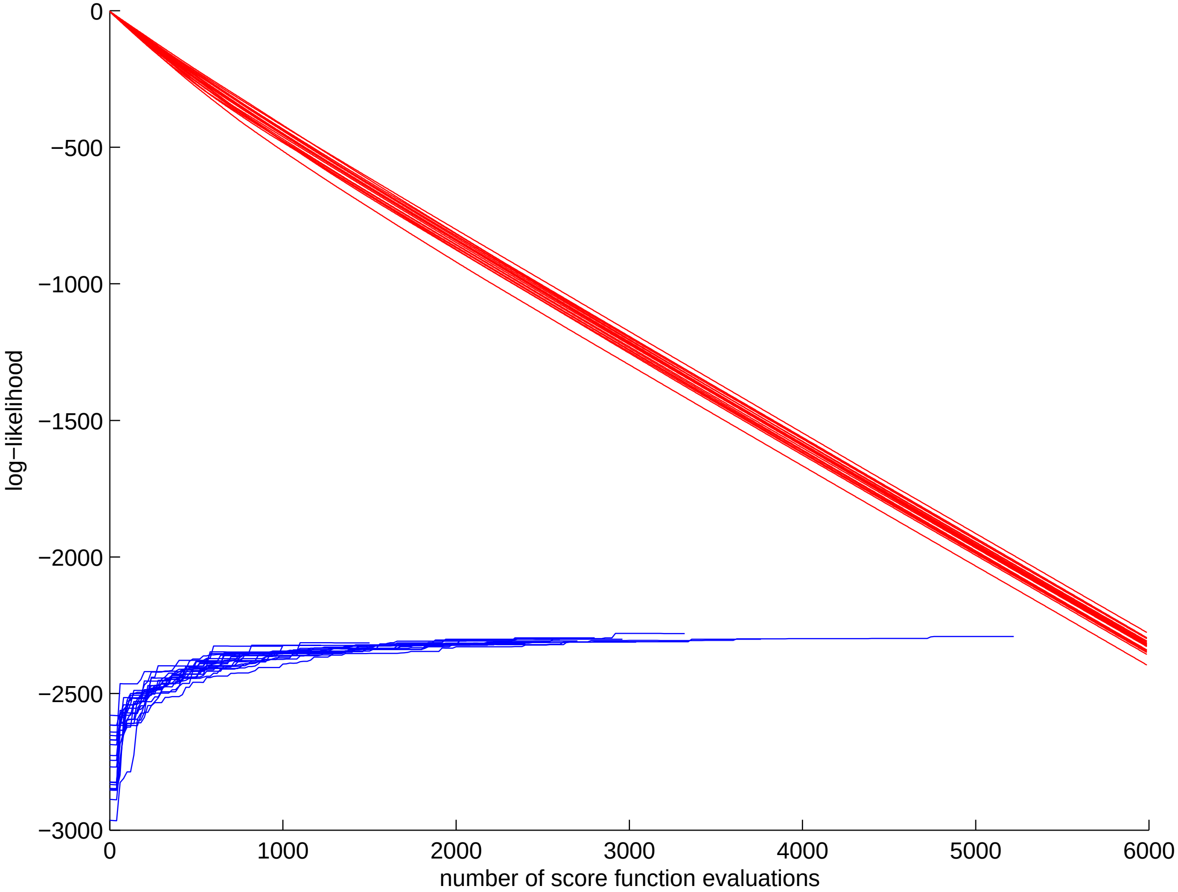


Example: Mouse PPI network

- For AnIS, I ran 20 particles, with 1000 cooling levels, with 6 Markov steps at each level.
- For AdIS, I ran 20 simulation runs, with $N = 20$ at each iteration, elite sample sizes adjusted dynamically.
- Simulation results:

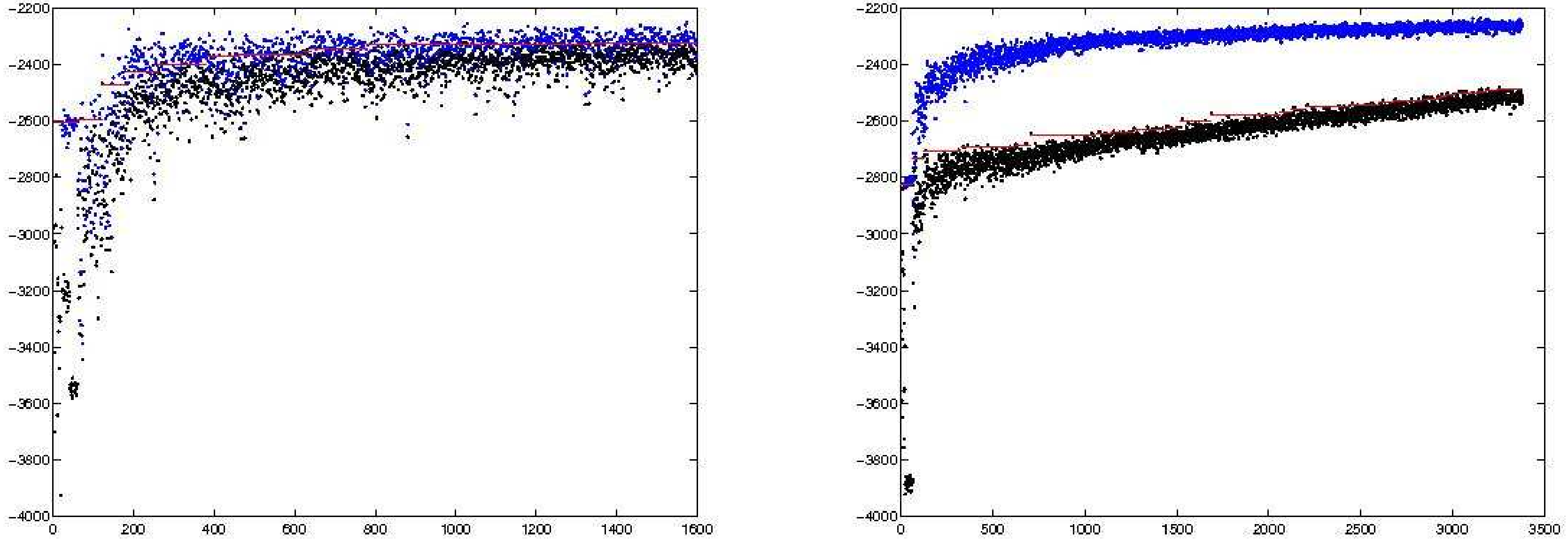
Model	log-lik	sample var. log lik
Erdös-Réni	$-3.070e3$	-
PA CE-MDL IS	$-2.280e3$	$3.41e2$
PA AnIS	$-2.276e3$	$6.80e2$

Example: Mouse PPI network



Red lines are Annealed IS simulations. Blue lines are MDL-CE Adaptive IS runs.

Example: MDL vs no MDL



Simulation run of CE-MDL AdIS on the left, CE AdIS with no MDL on the right. Blue points correspond to score function values, black points correspond to importance weights. Note the wide separation of black and blue points in AdIS without MDL.

Comparison of AnIS and AdIS

- Advantages of CE-MDL AdIS:
  - Results are interpretable; gives distribution on labelings that can be used as Bayesian prior or mixture distribution.
  - Recasts integration problem as an optimization problem.
  - Efficient for at least some classes of networks and NGMs.
- Disadvantages of CE-MDL AdIS
  - Best possible AdIS dist. $\hat{g}^*$ for proposal familty may not be close to optimal IS dist, potentially leading to poor performance and misleading results.
  - Convergence may be slow.

Comparison of AnIS and AdIS (cont.)

- Advantages of AnIS:
  - Non-parametric, easy to implement
  - Efficient in practice for many applications
- Disadvantages of AnIS:
  - Need to formulate 'cooling schedule'.
  - Works as well or poorly as simulated annealing.
  - Results not as interpretable.
- Running times comparable for our example.
- Both methods produce unbiased estimators $\rightarrow$ can run both and reliably combine results.

Future Work

- Implement for other copying models, e.g. vertex copying and Kronecker delta.
  - Use distributions on phylogenies?
- Try other models of rank as proposal distributions – Thurstonian model seems particularly promising.
- Analysis of convergence rate for simplified model.

Previous Work/References

- Network model selection
  - Kronecker delta model maximum likelihood – [Faloustous et al.]
  - Gibbs-type algorithm – [Bezakova et al.]
  - Sequential IS for growth models [Wiuf et al.]
- Adaptive Importance Sampling – [Rubinstein and Kroese, 2004], [Asmussen and Glynn, 2007]
- Models of rank – [Marden 1995], [Diaconis 1988]