StarAI 2015

- Fifth International Workshop on Statistical Relational AI
- At the 31st Conference on Uncertainty in Artificial Intelligence (**UAI**) (right after ICML)
- In **Amsterdam**, The Netherlands, on **July 16**.
- Paper Submission: **May 15**
  - Full, 6+1 pages
  - Short, 2 page position paper or abstract

What we can't do (yet, well)?

Approximate Symmetries in Lifted Inference

Guy Van den Broeck

(on joint work with Mathias Niepert and Adnan Darwiche)

KU Leuven

Overview

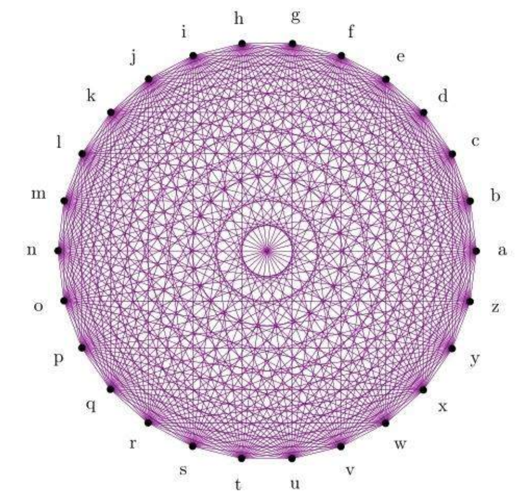
- Lifted inference in 2 slides
- Complexity of evidence
- Over-symmetric approximations
- Approximate symmetries
- Conclusions

Overview

- **Lifted inference in 2 slides**
- Complexity of evidence
- Over-symmetric approximations
- Approximate symmetries
- Conclusions

Lifted Inference

- In AI: exploiting symmetries/exchangeability
- Example: WebKB



symmetry

<u>Domain:</u>

url ∈ { “google.com”, ”ibm.com”, “aaai.org”, … }

<u>Weighted clauses:</u>

0.049 CoursePage(x) ^ Linked(x,y) => CoursePage(y)

-0.031 FacultyPage(x) ^ Linked(x,y) => FacultyPage (y)

...

0.235 HasWord(“Lecture",x) => CoursePage(x)

0.048 HasWord(“Office",x) => FacultyPage(x)

...

5000 more first-order sentences

The State of Lifted Inference

- UCQ database queries: solved

  PTIME in database size (when possible)

- MLNs and related
  - Two logical variables: solved

    Partition function PTIME in domain size (always)

  - Three logical variables: $\#P_1$-hard
- Bunch of great approximation algorithms
- Theoretical connections to **exchangeability**

Overview

- Lifted inference in 2 slides
- **Complexity of evidence**
- Over-symmetric approximations
- Approximate symmetries
- Conclusions

Problem: Prediction with Evidence

- Add evidence on links:

Linked("google.com", "gmail.com")
Linked("google.com", "aaai.org")

Symmetry google.com – ibm.com? **No!**

Linked("ibm.com", "watson.com")
Linked("ibm.com", "ibm.ca")

- Add evidence on words

HasWord("Android", "google.com")
HasWord("G+", "google.com")

Symmetry google.com – ibm.com? **No!**

HasWord("Blue", "ibm.com")
HasWord("Computing", "ibm.com")

Complexity in Size of "Evidence"

- Consider a model liftable for model counting:

3.14 FacultyPage(x) $\wedge$ Linked(x,y) $\Rightarrow$ CoursePage(y)

- Given database DB, compute P(Q|DB). Complexity in DB size?
  - Evidence on unary relations: Efficient

FacultyPage("google.com")=0, CoursePage("coursera.org")=1, …

  - Evidence on binary relations: #P-hard

Linked("google.com","gmail.com")=1, Linked("google.com",“aaai.org")=0

Intuition: Binary evidence breaks symmetries

Consequence: Lifted algorithms reduce to ground (also approx)

[Van den Broeck, Davis; AAAI'12, Bui et al., Dalvi and Suciu, etc.]

Approach

- Conditioning on **binary** evidence is hard
- Conditioning on **unary** evidence is efficient
- Solution: Represent binary evidence as unary
- Matrix notation:

$$e = \mathrm{p}(a,a) \wedge \mathrm{p}(a,b) \wedge \neg\, \mathrm{p}(a,c) \wedge \cdots \wedge \neg\, \mathrm{p}(d,c) \wedge \mathrm{p}(d,d)$$

$$\mathbf{P} = \begin{array}{c} \mathrm{p}(X,Y) \\ X=a \\ X=b \\ X=c \\ X=d \end{array} \begin{array}{c} \begin{array}{cccc} Y=a & Y=b & Y=c & Y=d \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Vector Product

- <u>Solution</u>: Represent binary evidence as unary
- Case 1: $\forall X, \forall Y, \mathrm{p}(X, Y) \Leftrightarrow \mathrm{q}(X) \wedge \mathrm{r}(Y)$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{e = \neg\, \mathrm{p}(a, a) \wedge \neg\, \mathrm{p}(a, b) \wedge \cdots \wedge \neg\, \mathrm{p}(d, c) \wedge \mathrm{p}(d, d)}$$

Vector Product

- <u>Solution</u>: Represent binary evidence as unary
- Case 1: $\forall X, \forall Y, \mathrm{p}(X, Y) \Leftrightarrow \mathrm{q}(X) \wedge \mathrm{r}(Y)$

$$\mathbf{P} = \begin{array}{cc} \begin{array}{c} 0 \\ 1 \\ 0 \\ 1 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \\ & \begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \end{array}$$

$$\boxed{e = \neg\, \mathrm{p}(a, a) \wedge \neg\, \mathrm{p}(a, b) \wedge \cdots \wedge \neg\, \mathrm{p}(d, c) \wedge \mathrm{p}(d, d)}$$

Vector Product

- <u>Solution</u>: Represent binary evidence as unary
- Case 1: $\forall X,\, \forall Y,\, \mathrm{p}(X, Y) \Leftrightarrow \mathrm{q}(X) \wedge \mathrm{r}(Y)$

$$\mathbf{P} = \begin{array}{cc} \begin{array}{c} 0 \\ 1 \\ 0 \\ 1 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \\ & \begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \end{array} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}} = \mathbf{q}\,\mathbf{r}^{\mathsf{T}}$$

$$\boxed{e = \neg\,\mathrm{p}(a, a) \wedge \neg\,\mathrm{p}(a, b) \wedge \cdots \wedge \neg\,\mathrm{p}(d, c) \wedge \mathrm{p}(d, d)}$$

Vector Product

- Solution: Represent binary evidence as unary
- Case 1: $\forall X, \forall Y, \mathrm{p}(X,Y) \Leftrightarrow \mathrm{q}(X) \wedge \mathrm{r}(Y)$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}} = \mathbf{q}\,\mathbf{r}^{\mathsf{T}}$$

$$\boxed{e = \neg\,\mathrm{p}(a,a) \wedge \neg\,\mathrm{p}(a,b) \wedge \cdots \wedge \neg\,\mathrm{p}(d,c) \wedge \mathrm{p}(d,d)}$$

$$e = \neg\,\mathrm{q}(a) \wedge \mathrm{q}(b) \wedge \neg\,\mathrm{q}(c) \wedge \mathrm{q}(d) \quad 0\,1\,0\,1$$
$$\wedge\, \mathrm{r}(a) \wedge \neg\,\mathrm{r}(b) \wedge \neg\,\mathrm{r}(c) \wedge \mathrm{r}(d) \quad 1\,0\,0\,1$$

Matrix Product

- Solution: Represent binary evidence as unary
- Case 2: $\forall X, \forall Y, \mathrm{p}(X,Y) \Leftrightarrow (\mathrm{q}_1(X) \wedge \mathrm{r}_1(Y))$

$$\begin{aligned} & \vee (\mathrm{q}_2(X) \wedge \mathrm{r}_2(Y)) \\ & \vee \dots \\ & \vee (\mathrm{q}_n(X) \wedge \mathrm{r}_n(Y)) \end{aligned}$$

Matrix Product

- Solution: Represent binary evidence as unary
- Case 2: $\forall X, \forall Y, \mathrm{p}(X,Y) \Leftrightarrow (\mathrm{q}_1(X) \wedge \mathrm{r}_1(Y))$

$$\begin{aligned} &\vee (\mathrm{q}_2(X) \wedge \mathrm{r}_2(Y)) \\ &\vee \ldots \\ &\vee (\mathrm{q}_n(X) \wedge \mathrm{r}_n(Y)) \end{aligned}$$

$$\mathbf{P} = \mathbf{q}_1\,\mathbf{r}_1^\intercal \vee \mathbf{q}_2\,\mathbf{r}_2^\intercal \vee \cdots \vee \mathbf{q}_n\,\mathbf{r}_n^\intercal = \mathbf{Q}\,\mathbf{R}^\intercal$$

where $(\mathbf{Q}\,\mathbf{R}^\intercal)_{i,j} = \bigvee_r \mathbf{Q}_{i,r} \wedge \mathbf{R}_{j,r}$

Boolean Matrix Factorization

- Decompose

$$\mathbf{P} = \mathbf{q}_1\,\mathbf{r}_1^{\intercal} \vee \mathbf{q}_2\,\mathbf{r}_2^{\intercal} \vee \cdots \vee \mathbf{q}_n\,\mathbf{r}_n^{\intercal} = \mathbf{Q}\,\mathbf{R}^{\intercal}$$

- In Boolean algebra, where 1+1=1
- Minimum *n* is the Boolean rank
- Always possible

Matrix Product

- Solution: Represent binary evidence as unary
- Example: $\mathbf{P} = \mathbf{q}_1 \, \mathbf{r}_1^\intercal \vee \mathbf{q}_2 \, \mathbf{r}_2^\intercal \vee \cdots \vee \mathbf{q}_n \, \mathbf{r}_n^\intercal = \mathbf{Q} \, \mathbf{R}^\intercal$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^\intercal \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^\intercal \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^\intercal$$

Matrix Product

- Solution: Represent binary evidence as unary
- Example: $\mathbf{P} = \mathbf{q}_1\, \mathbf{r}_1^\intercal \vee \mathbf{q}_2\, \mathbf{r}_2^\intercal \vee \cdots \vee \mathbf{q}_n\, \mathbf{r}_n^\intercal = \mathbf{Q}\, \mathbf{R}^\intercal$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^\intercal \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^\intercal \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^\intercal$$

Matrix Product

- <u>Solution</u>: Represent binary evidence as unary
- Example: $\mathbf{P} = \mathbf{q}_1\,\mathbf{r}_1^\mathsf{T} \vee \mathbf{q}_2\,\mathbf{r}_2^\mathsf{T} \vee \cdots \vee \mathbf{q}_n\,\mathbf{r}_n^\mathsf{T} = \mathbf{Q}\,\mathbf{R}^\mathsf{T}$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^\mathsf{T} \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^\mathsf{T} \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^\mathsf{T}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^\mathsf{T}$$

Matrix Product

- Solution: Represent binary evidence as unary
- Example: $\mathbf{P} = \mathbf{q}_1\,\mathbf{r}_1^{\mathsf{T}} \vee \mathbf{q}_2\,\mathbf{r}_2^{\mathsf{T}} \vee \cdots \vee \mathbf{q}_n\,\mathbf{r}_n^{\mathsf{T}} = \mathbf{Q}\,\mathbf{R}^{\mathsf{T}}$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}} \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^{\mathsf{T}} \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^{\mathsf{T}}$$

Boolean rank n=3

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

Theoretical Consequences

- Theorem:

  > Complexity of computing Pr(q|e) in SRL is polynomial in |e|, when e has bounded Boolean rank.

- Boolean rank
  - key parameter in the complexity of conditioning
  - says how much lifting is possible

[Van den Broeck, Darwiche; NIPS'13]

Analogy with Treewidth in Probabilistic Graphical Models

<u>Probabilistic graphical models:</u> ⟷ <u>SRL Models:</u>

Probabilistic graphical models:

1. Find tree decomposition

1. Perform inference

- Exponential in (**tree**)**width** of decomposition
- Polynomial in **size** of Bayesian network

SRL Models:

1. Find Boolean matrix factorization of evidence
2. Perform inference

- Exponential in **Boolean rank** of evidence
- Polynomial in **size** of evidence database
- Polynomial in **domain size**

Overview

- Lifted inference in 2 slides
- Complexity of evidence
- **Over-symmetric approximations**
- Approximate symmetries
- Conclusions

Over-Symmetric Approximation

- Approximate Pr(q|e) by Pr(q|e')

  Pr(q|e') has more symmetries, is more liftable

- E.g.: Low-rank Boolean matrix factorization

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}} \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^{\mathsf{T}} \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^{\mathsf{T}}$$

Boolean rank 3

Over-Symmetric Approximation

- Approximate Pr(q|e) by Pr(q|e')

  Pr(q|e') has more symmetries, is more liftable

- E.g.: Low-rank Boolean matrix factorization

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}} \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^{\mathsf{T}} \vee \cancel{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^{\mathsf{T}}}$$

Boolean rank 2 approximation

$$\approx \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{0} & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

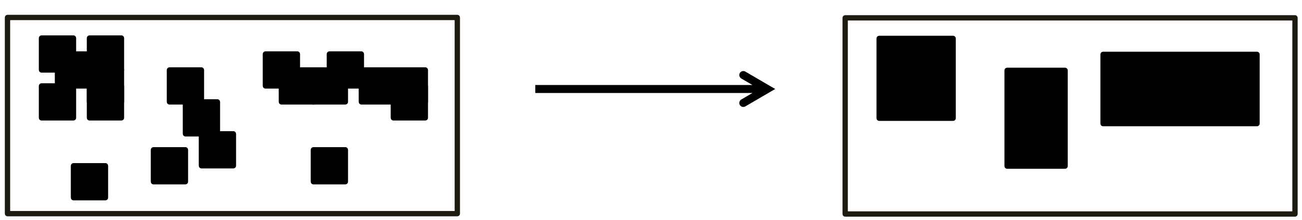
Over-Symmetric Approximations

- OSA makes model more symmetric
- E.g., low-rank Boolean matrix factorization

Link (“aaai.org”, “google.com”)
Link (“google.com”, “aaai.org”)
Link (“google.com”, “gmail.com”)
Link (“ibm.com”, “aaai.org”)

Link (“aaai.org”, “google.com”)
Link (“google.com”, “aaai.org”)
- ~~Link (“google.com”, “gmail.com”)~~
+ Link (“aaai.org”, “ibm.com”)
Link (“ibm.com”, “aaai.org”)

google.com and ibm.com become symmetric!



[Van den Broeck, Darwiche; NIPS’13]

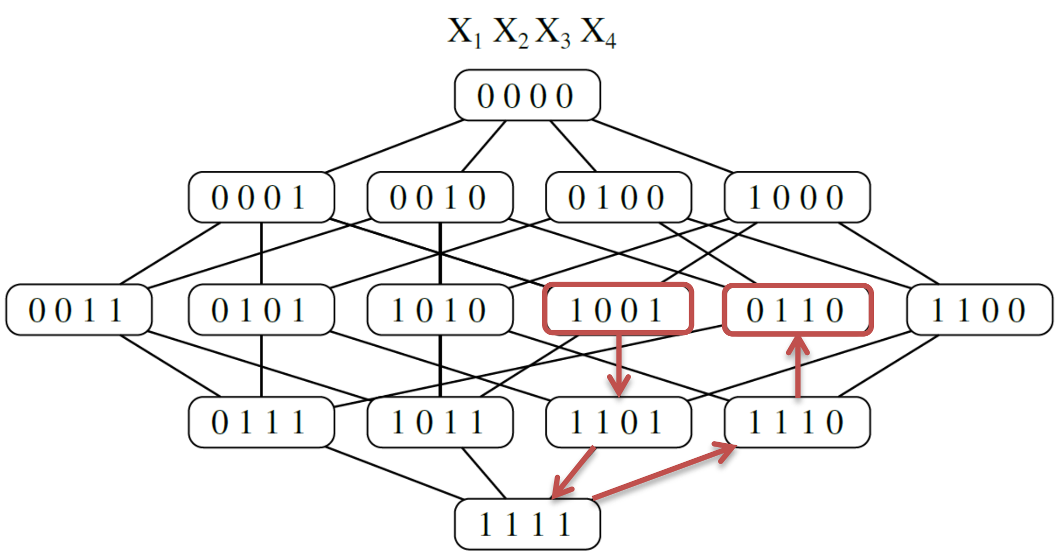
Markov Chain Monte-Carlo

Gibbs sampling or MC-SAT

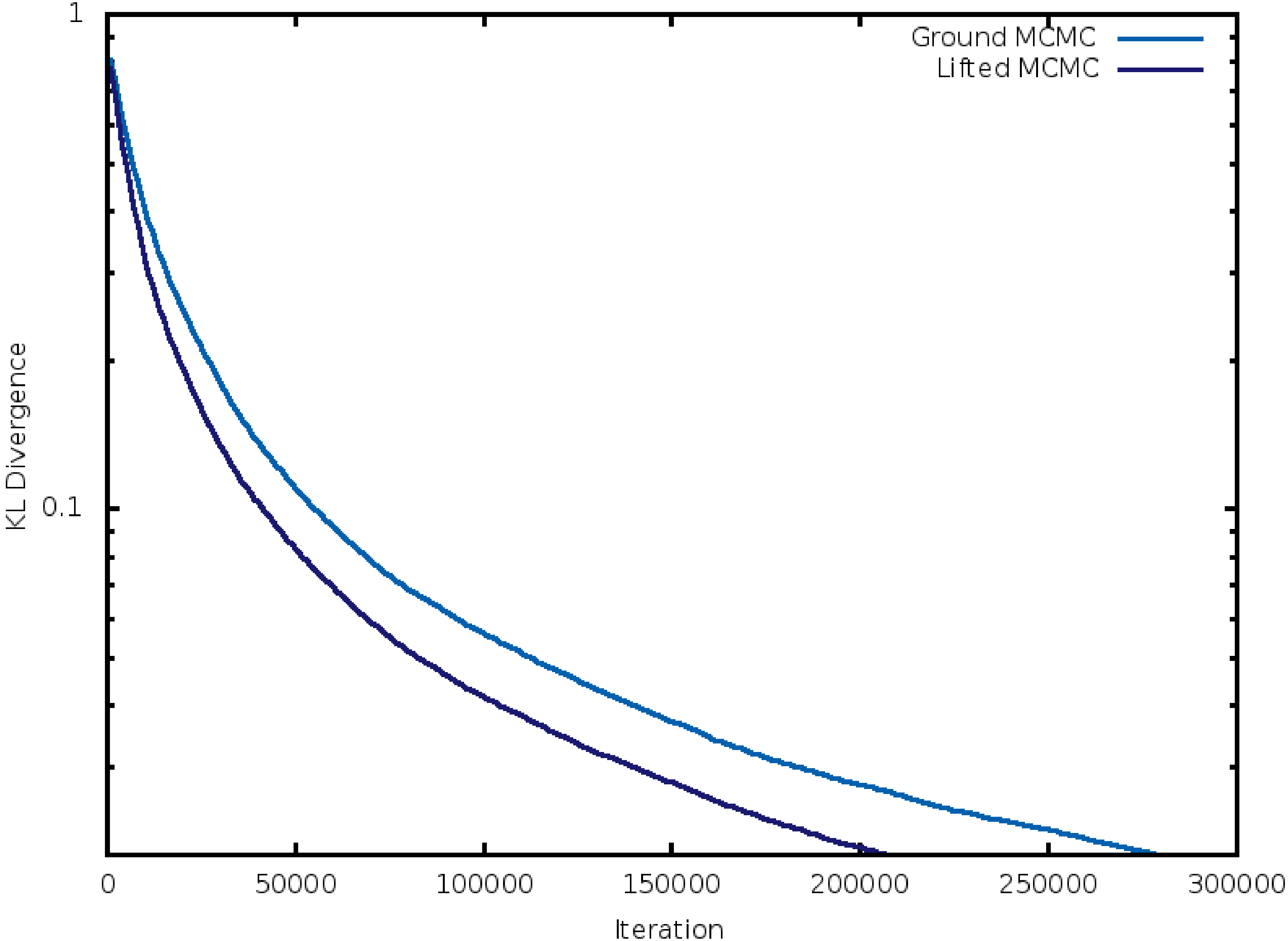
– Problem: slow convergence, one variable changed

– One million random variables: need at least one million iteration to move between two states

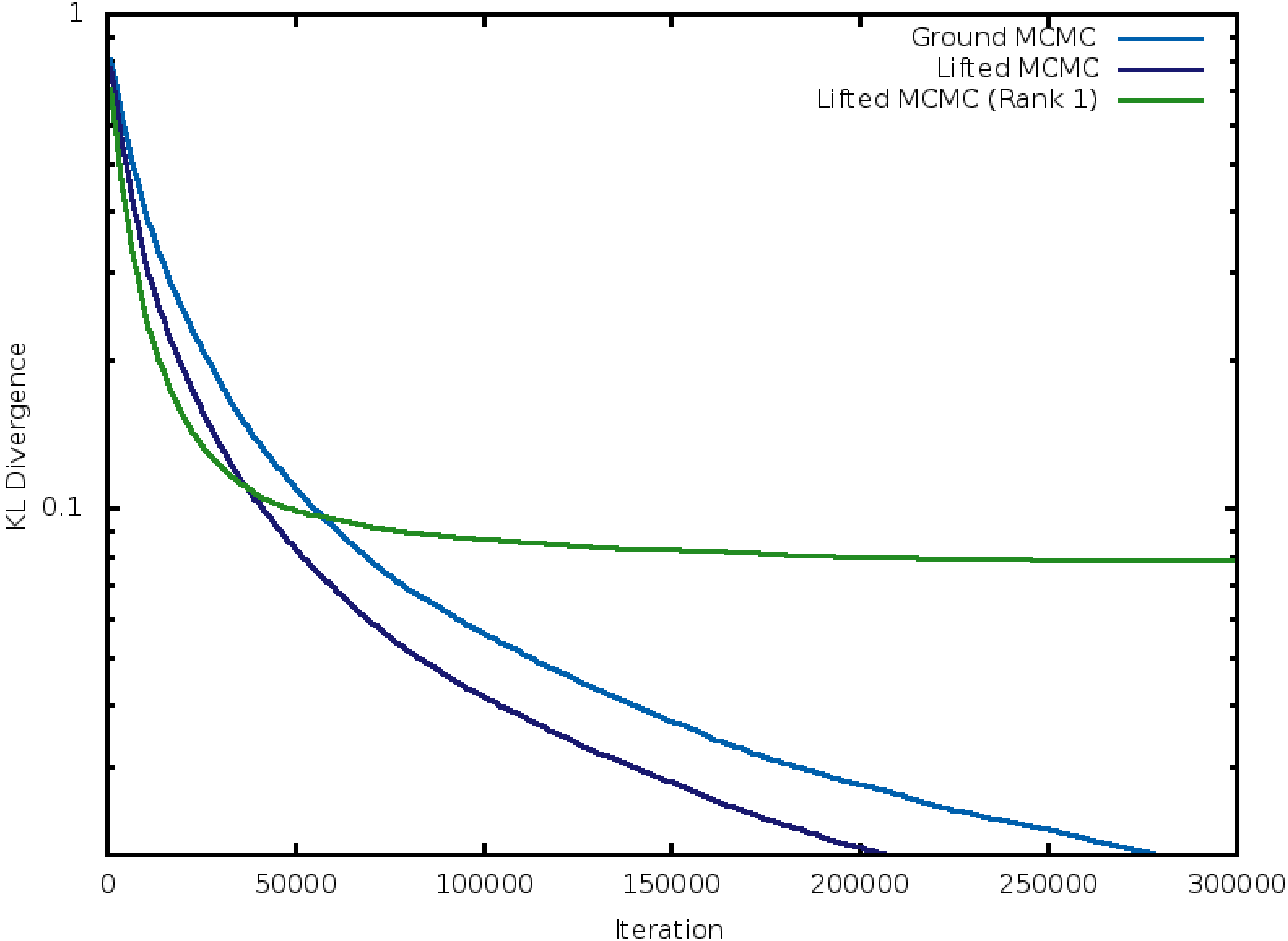
Lifted MCMC: move between symmetric states



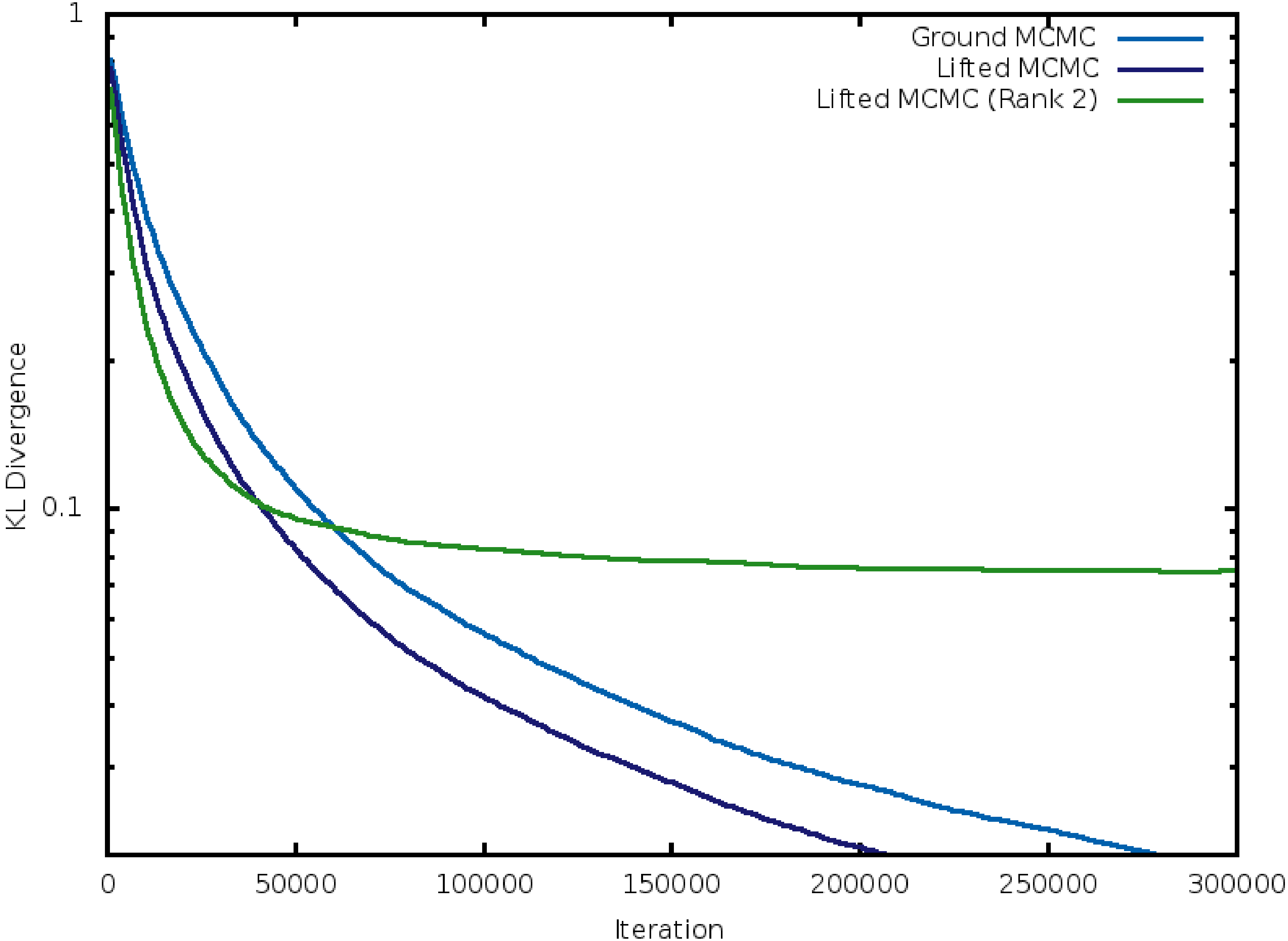
Lifted MCMC on WebKB



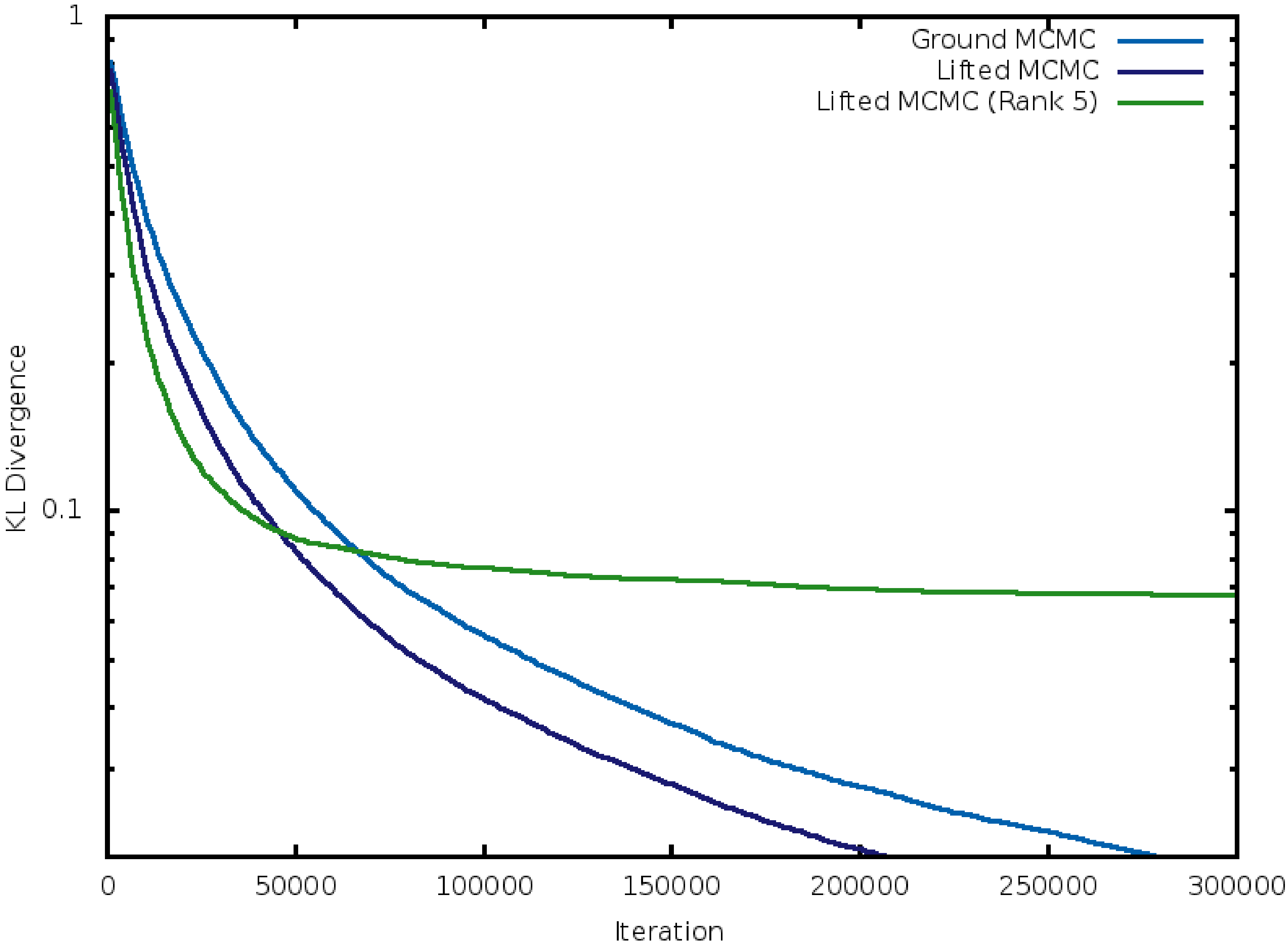
Rank 1 Approximation



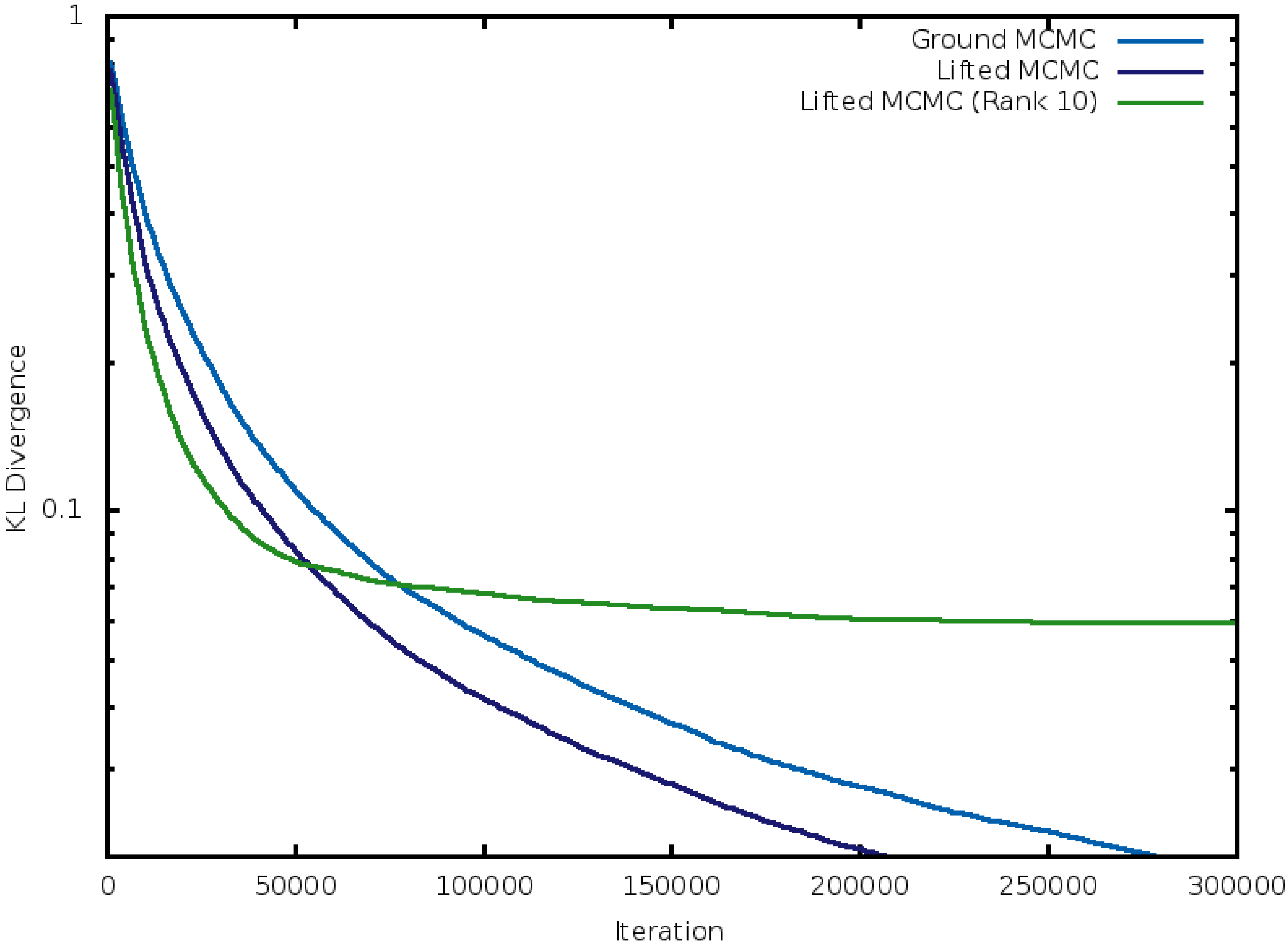
Rank 2 Approximation



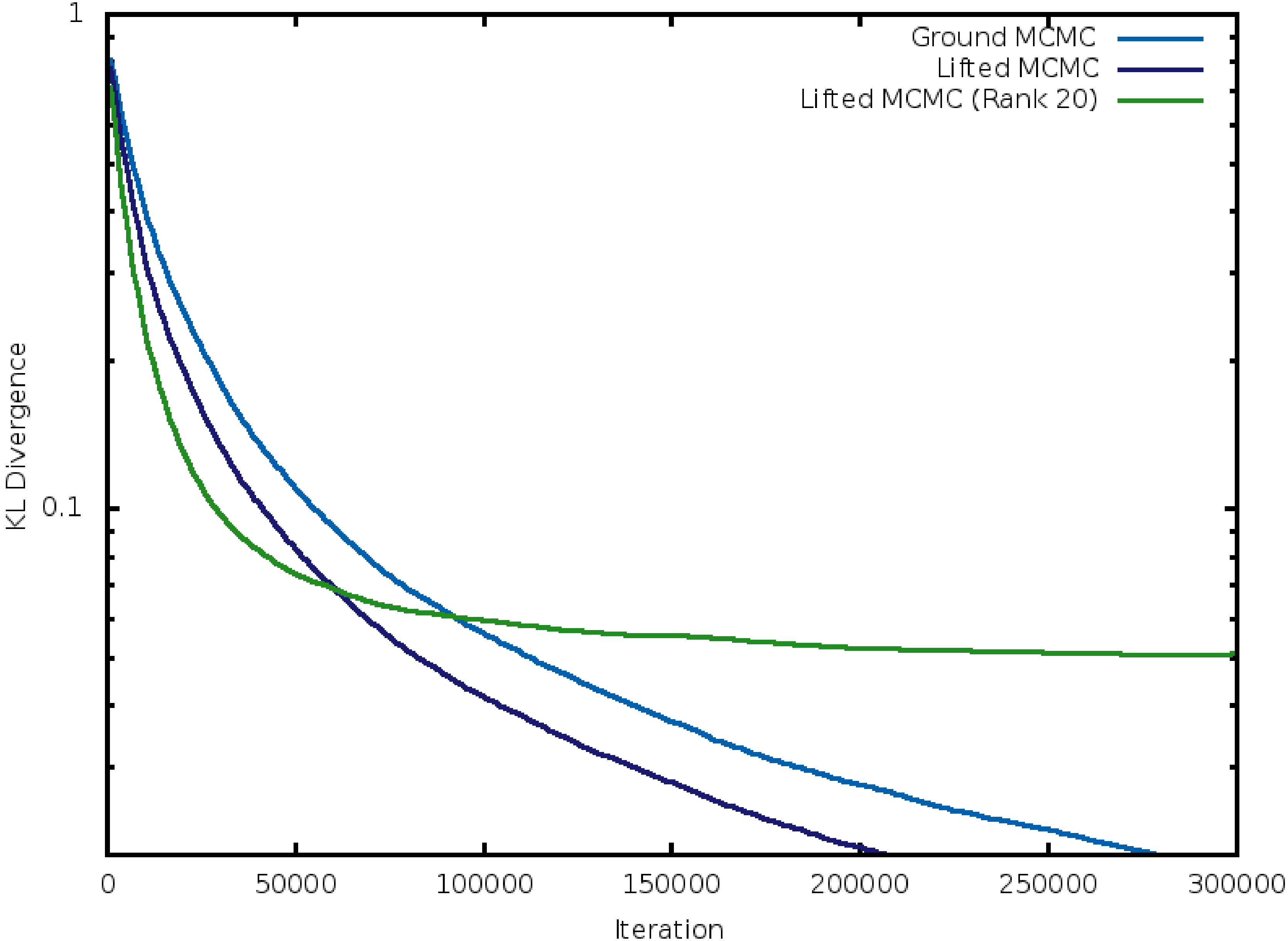
Rank 5 Approximation



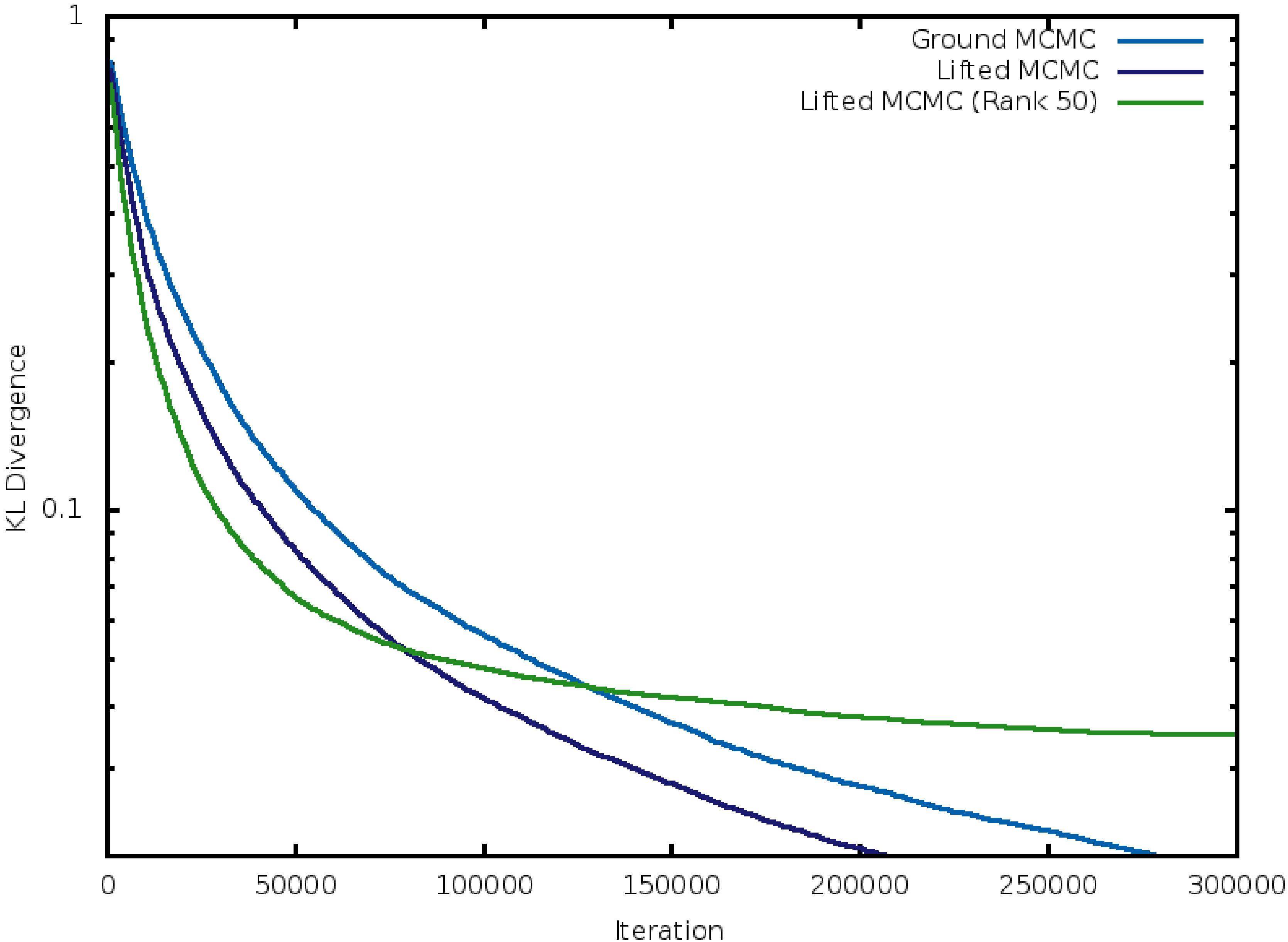
Rank 10 Approximation



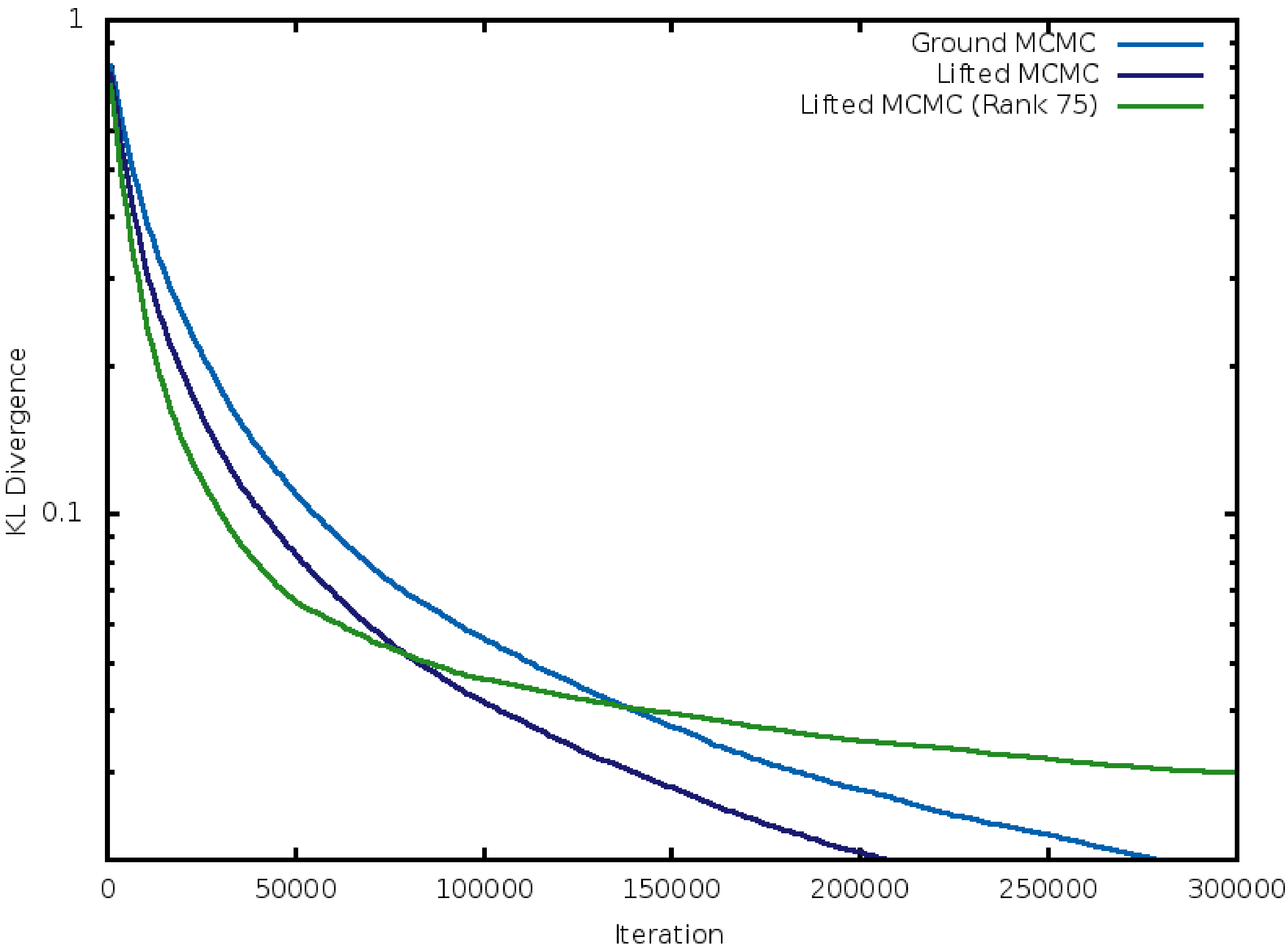
Rank 20 Approximation



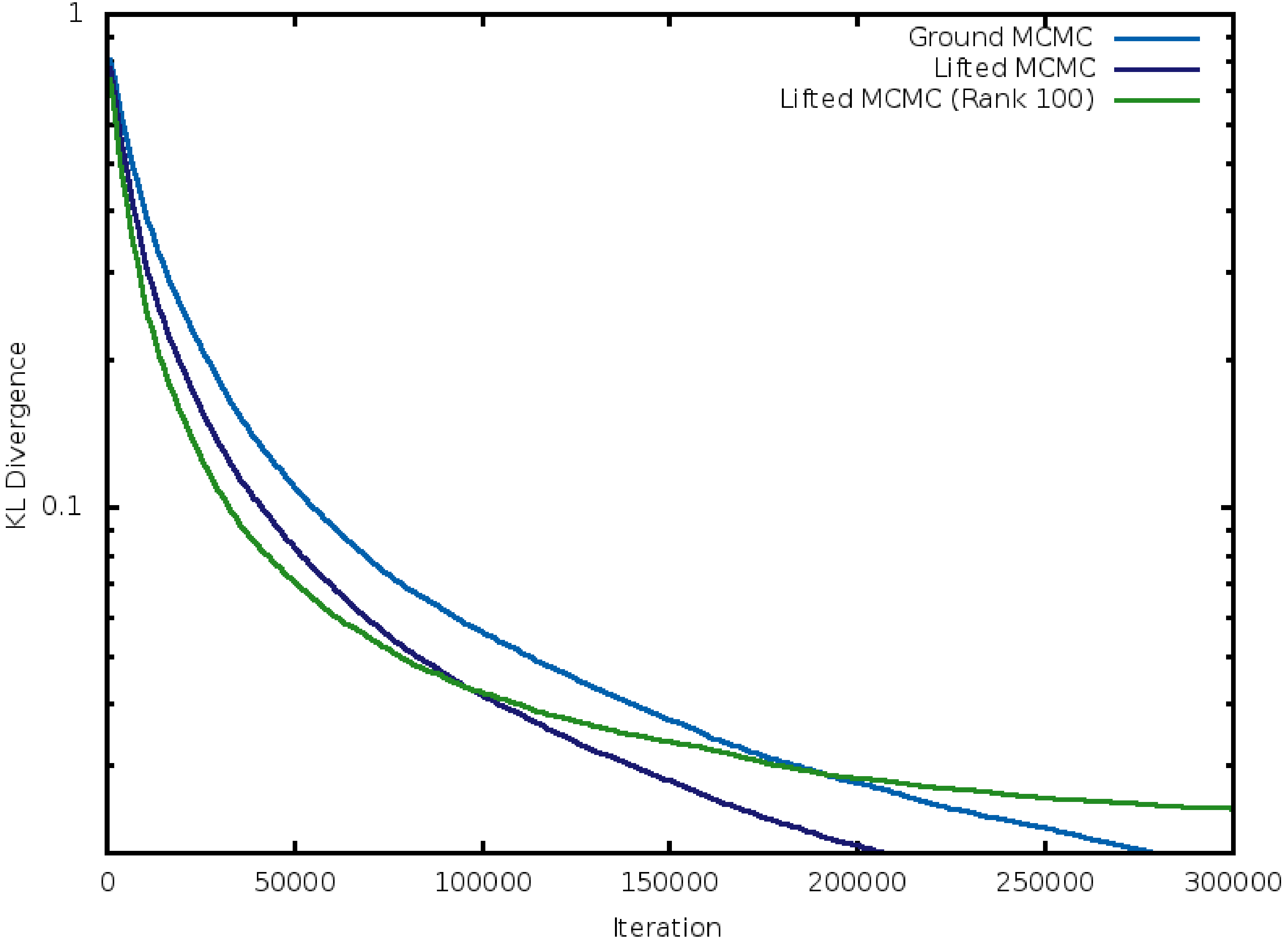
Rank 50 Approximation



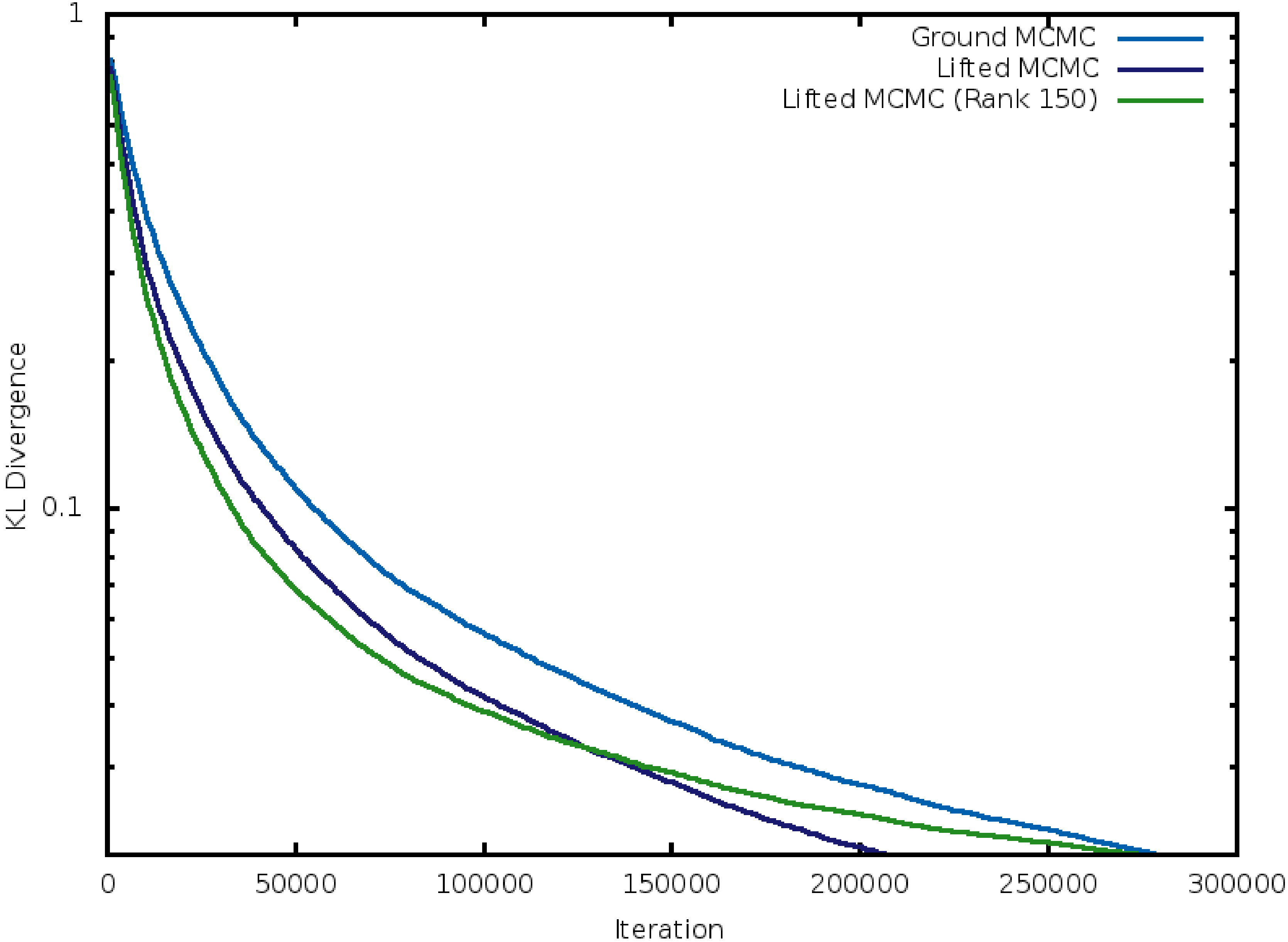
Rank 75 Approximation



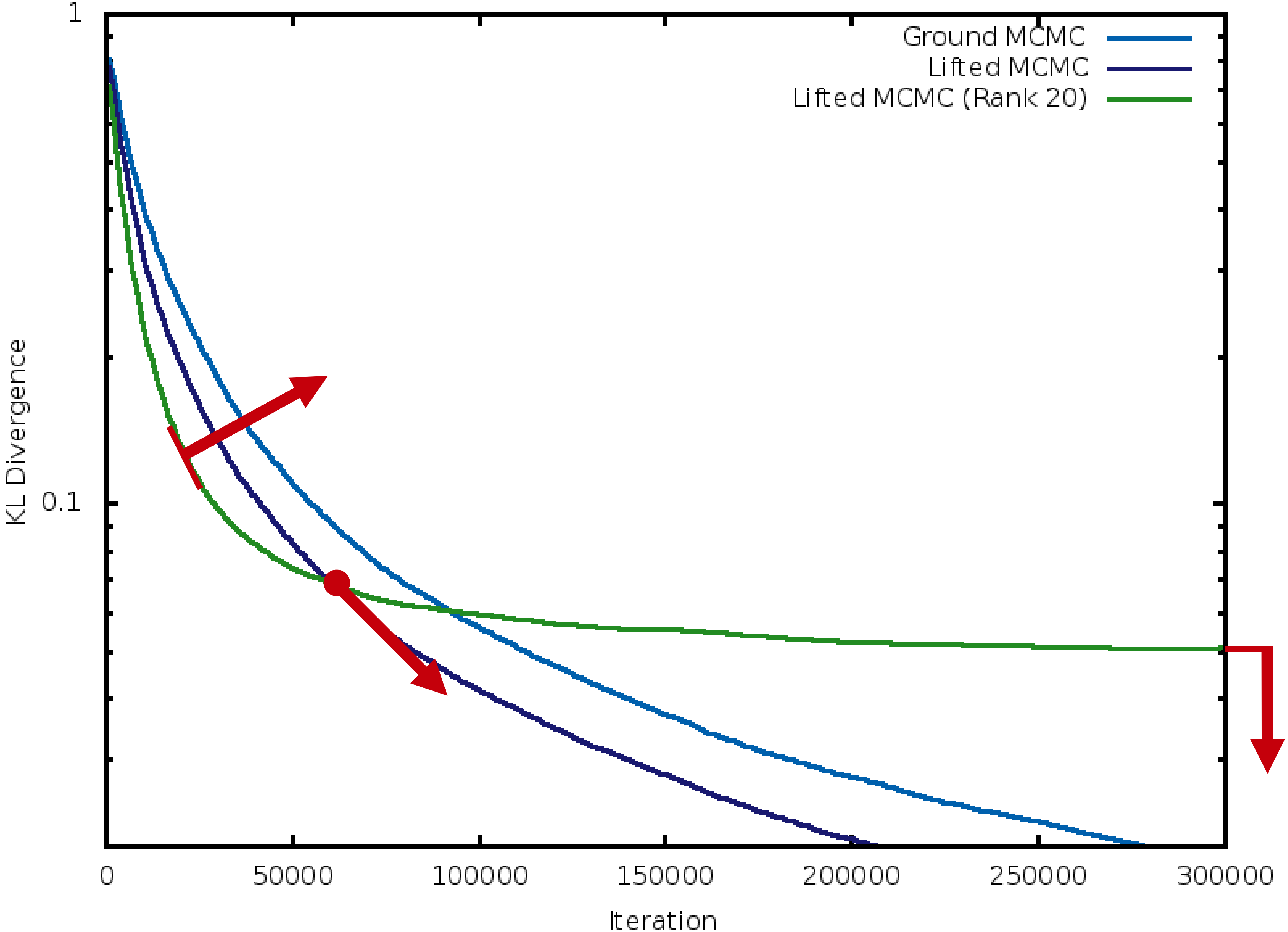
Rank 100 Approximation



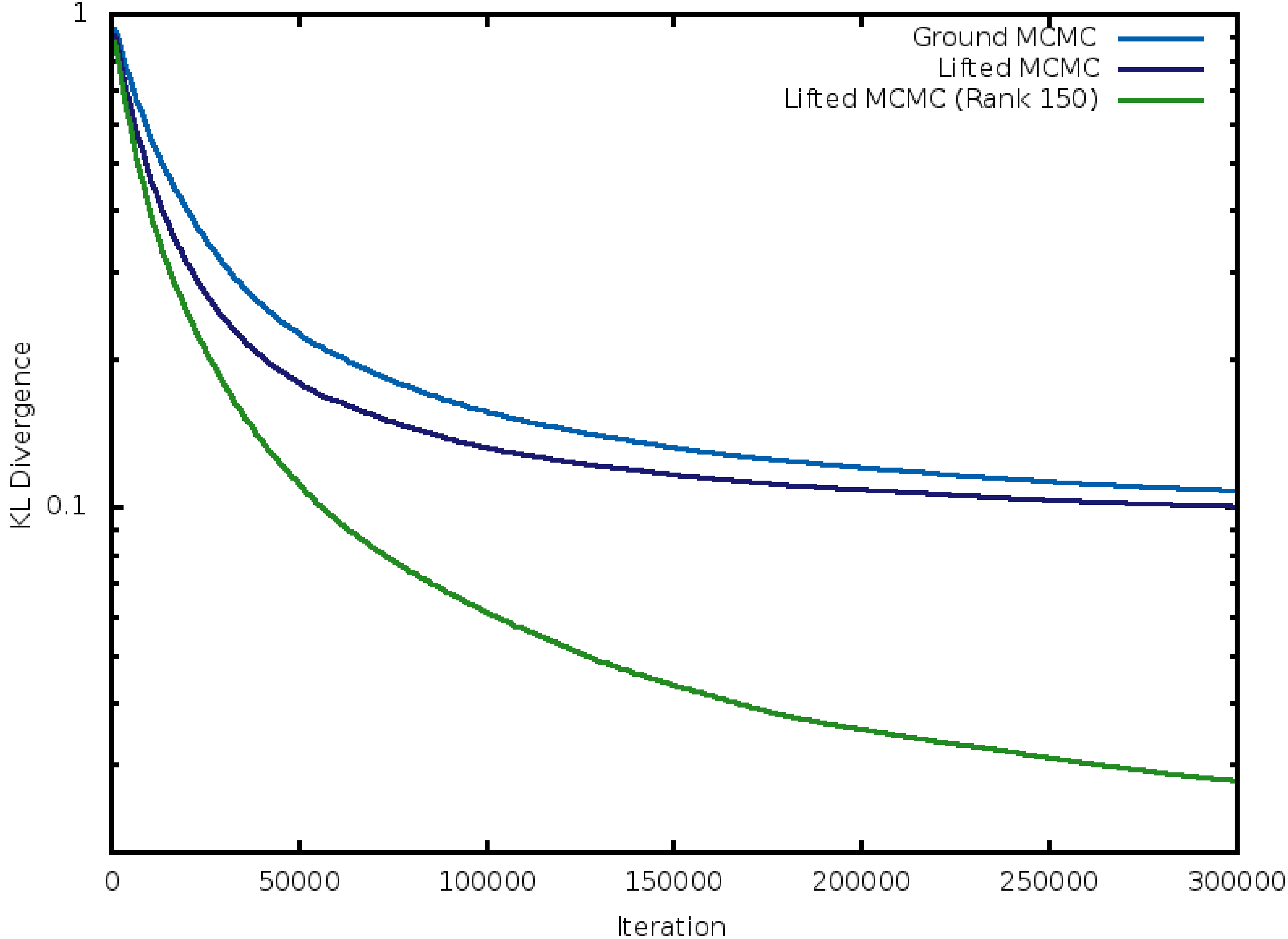
Rank 150 Approximation



Trend for Increasing Boolean Rank



Best Case



Ground MCMC
Lifted MCMC
Lifted MCMC (Rank 150)
KL Divergence
1
0.1
0
50000
100000
150000
200000
250000
300000
Iteration

Overview

- Lifted inference in 2 slides
- Complexity of evidence
- Over-symmetric approximations
- **Approximate symmetries**
- Conclusions

Problem with OSAs

- Approximation can be crude
- Cannot converge to true distribution
- Lose information about subtle differences
  - Real distribution

    Pr(PageClass(“Faculty”, “http://.../~pedro/”)) = 0.47
    Pr(PageClass(“Faculty”, “http://.../~luc/”)) = 0.53

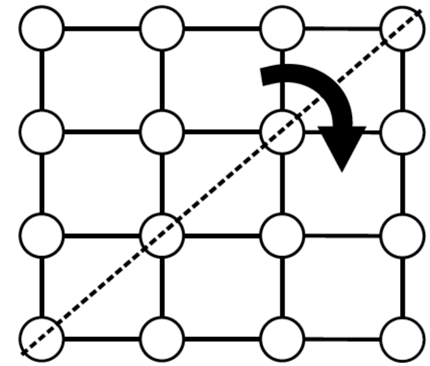
  - OSA distribution

    Pr(PageClass(“Faculty”, “http://.../~pedro/”)) = 0.50
    Pr(PageClass(“Faculty”, “http://.../~luc/”)) = 0.50

Approximate Symmetries

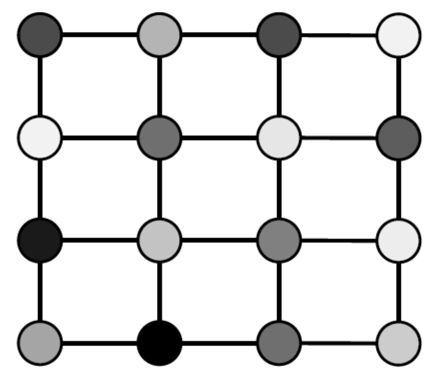
- Exploit approximate symmetries:
  - Exact symmetry g: $\Pr(\mathbf{x}) = \Pr(\mathbf{x}^g)$

    E.g. Ising model
    without external field



  - Approximate symmetry g: $\Pr(\mathbf{x}) \approx \Pr(\mathbf{x}^g)$

    E.g. Ising model with external field



P () ≈ P ()

Orbital Metropolis Chain: Algorithm

- Given symmetry group G (approx. symmetries)
- Orbit $\mathbf{x}^G$ contains all states approx. symm. to $\mathbf{x}$
- In state $\mathbf{x}$:

1. Select $\mathbf{y}$ uniformly at random from $\mathbf{x}^G$
2. Move from $\mathbf{x}$ to $\mathbf{y}$ with probability $\min\left(\frac{\Pr(\boldsymbol{y})}{\Pr(\boldsymbol{x})}, 1\right)$
3. Otherwise: stay in $\mathbf{x}$ (reject)
4. Repeat

Orbital Metropolis Chain: Analysis

- ✓ Pr(.) is stationary distribution
- ✓ Many variables change (fast mixing)
- ✓ Few rejected samples:

$$\Pr(\boldsymbol{y}) \approx \Pr(\boldsymbol{x}) \Rightarrow \ \min\left(\frac{\Pr(\boldsymbol{y})}{\Pr(\boldsymbol{x})}, 1\right) \approx 1$$

Is this the perfect proposal distribution?

Orbital Metropolis Chain: Analysis

- ✓ Pr(.) is stationary distribution
- ✓ Many variables change (fast mixing)
- ✓ Few rejected samples:

$$\Pr(\boldsymbol{y}) \approx \Pr(\boldsymbol{x}) \Rightarrow \min\left(\frac{\Pr(\boldsymbol{y})}{\Pr(\boldsymbol{x})}, 1\right) \approx 1$$

Is this the perfect proposal distribution?

✘ Not irreducible…
Can never reach 0100 from 1101.

Lifted Metropolis-Hastings: Algorithm

- Given an **orbital Metropolis chain $\mathbf{M_S}$** for Pr(.)
- Given a **base Markov chain $\mathbf{M_B}$** that
  - is irreducible and aperiodic
  - has stationary distribution Pr(.)

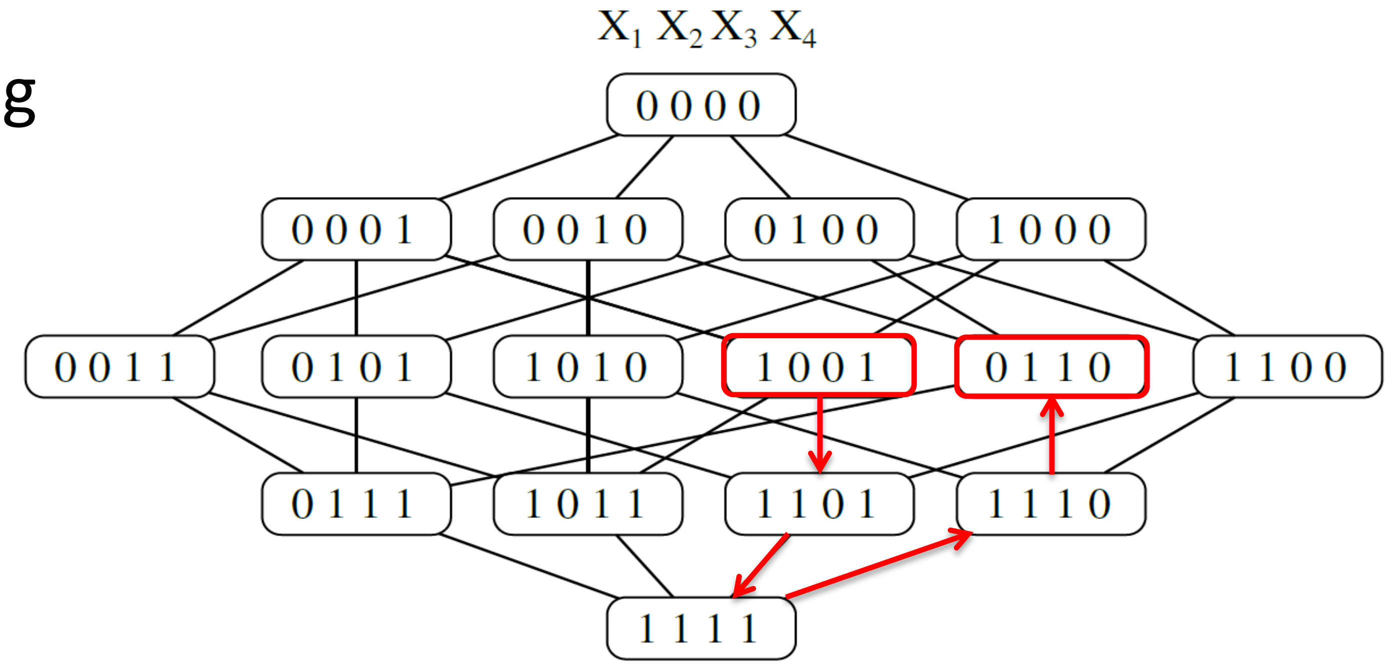
  (e.g., Gibbs chain or MC-SAT chain)
- In state **x**:
  1. With probability α, apply the kernel of M_B
  2. Otherwise apply the kernel of M_S

Lifted Metropolis-Hastings: Analysis

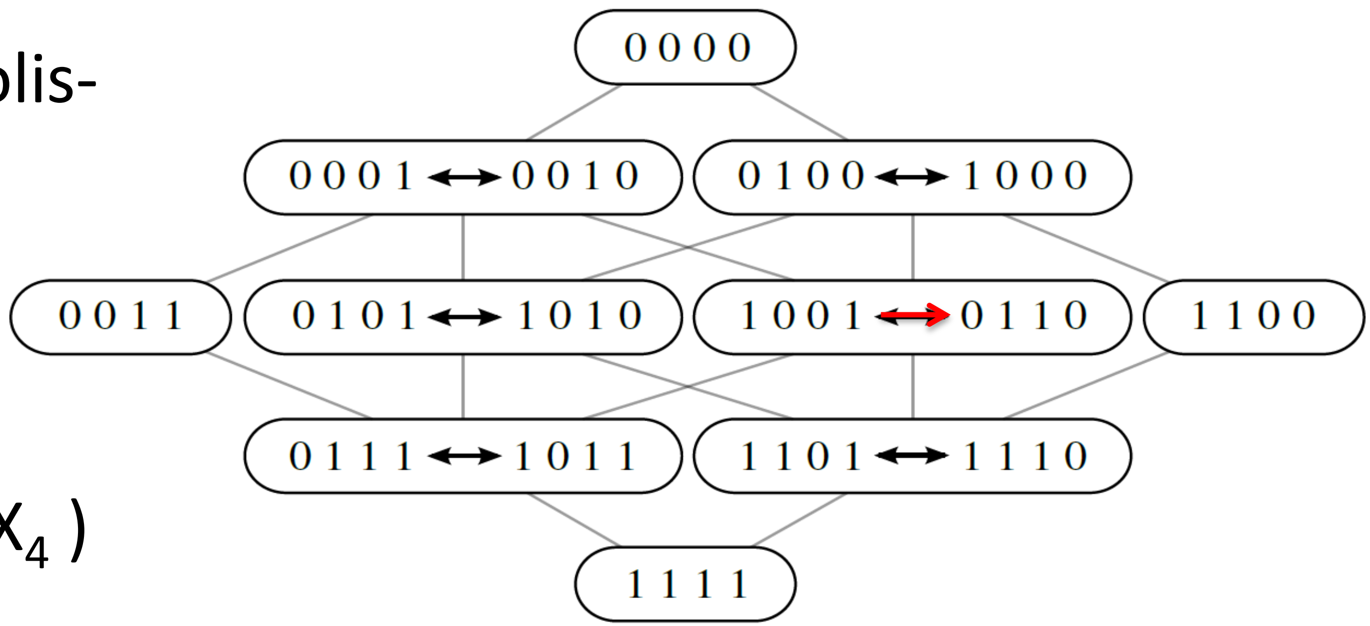
<u>Theorem [Tierney 1994]:</u>
A mixture of Markov chains is irreducible and aperiodic if at least one of the chains is irreducible and aperiodic .

- ✓ Pr(.) is stationary distribution
- ✓ Many variables change (fast mixing)
- ✓ Few rejected samples
- ✓ Irreducible
- ✓ Aperiodic

Gibbs Sampling

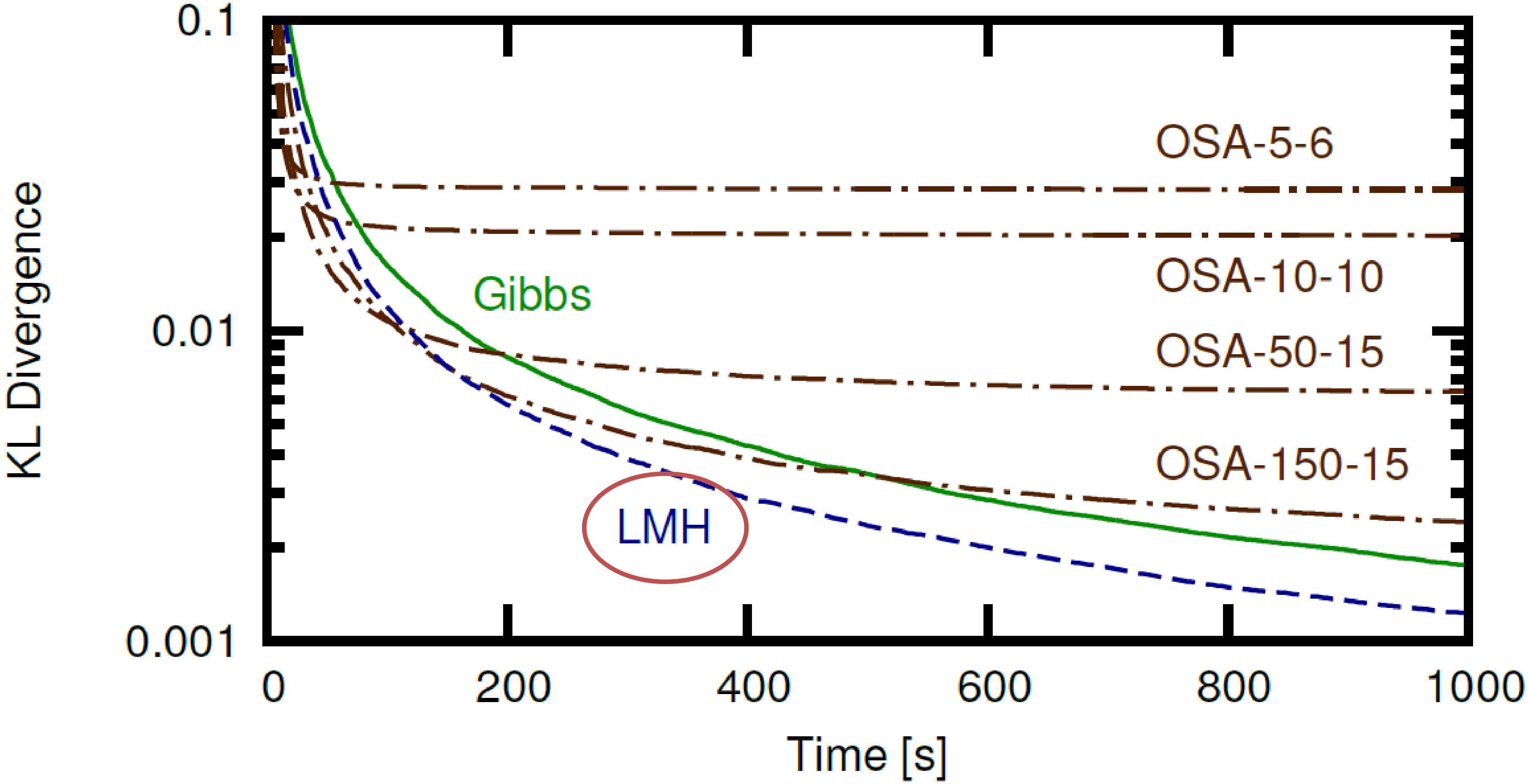


Lifted Metropolis-Hastings



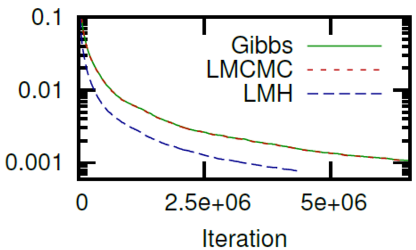
$G = (X_1\ X_2\ )(X_3\ X_4\ )$

Experiments: WebKB

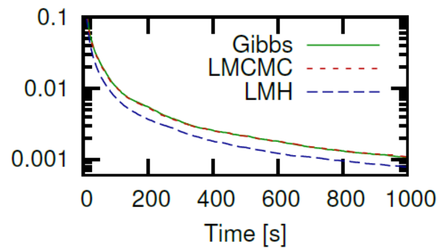


[Van den Broeck, Niepert; AAAI'15]

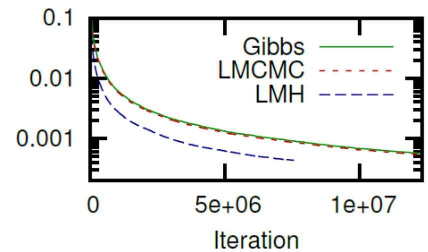
Experiments: WebKB



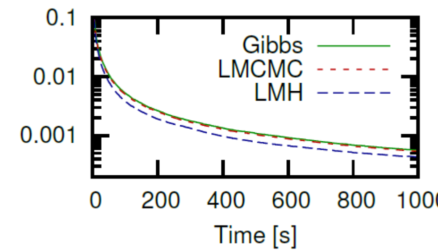
(a) Texas - Iterations



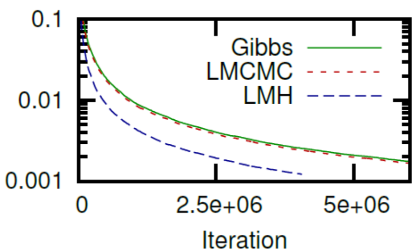
(b) Texas - Time



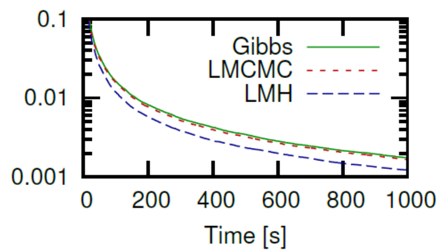
(a) Cornell - Iterations



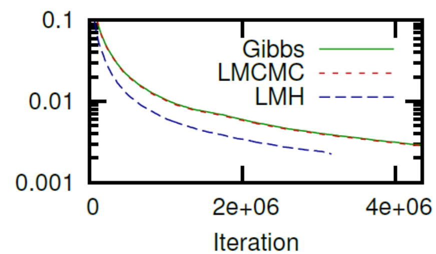
(b) Cornell - Time



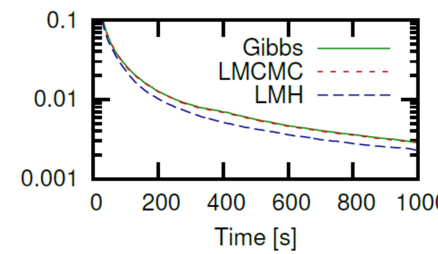
(c) Washington - Iterations



(d) Washington - Time



(c) Wisconsin - Iterations



(d) Wisconsin - Time

Overview

- Lifted inference in 2 slides
- Complexity of evidence
- Over-symmetric approximations
- Approximate symmetries
- **Conclusions**

Take-Away Message

Two problems:

1. Lifted inference gives **exponential speedups** in **symmetric** graphical models.
   But what about real-world **asymmetric** problems?

2. When there are **many variables**, MCMC is **slow**.
   How to sample quickly in large graphical models?

One solution: Exploit **approximate symmetries**!

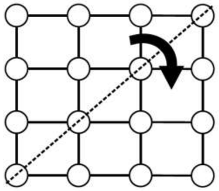
Open Problems

- Find approximate symmetries
  - Principled (theory)
  - Is a type of machine learning?
  - During inference, not preprocessing?
- Give guarantees on approximation quality/convergence speed
- Plug in lifted inference from prob. databases

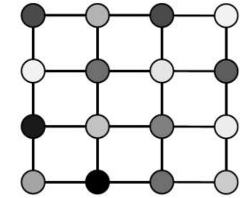
Lots of Recent Activity

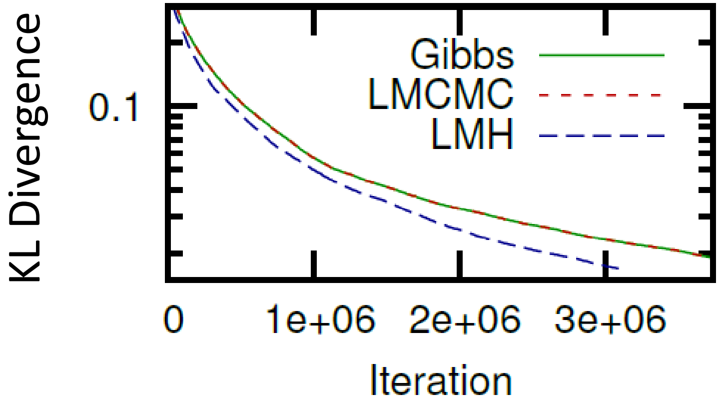
- Singla, Nath, and Domingos (2014)
- Venugopal and Gogate (2014)
- Kersting et al. (2014)

Thanks

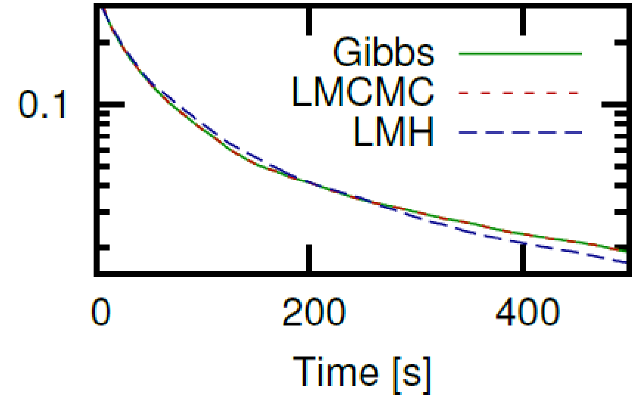


Example: Grid Models

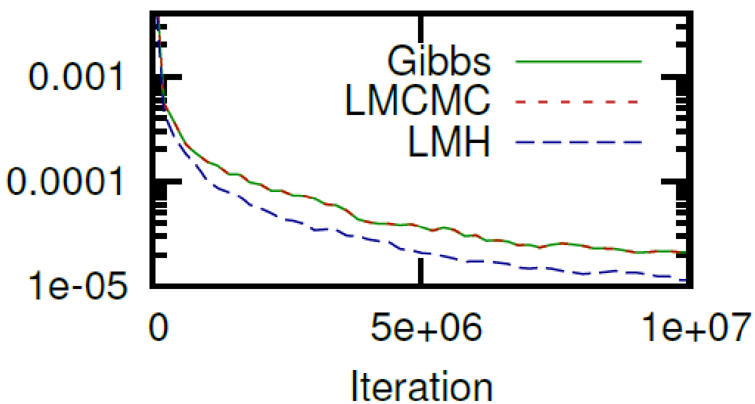




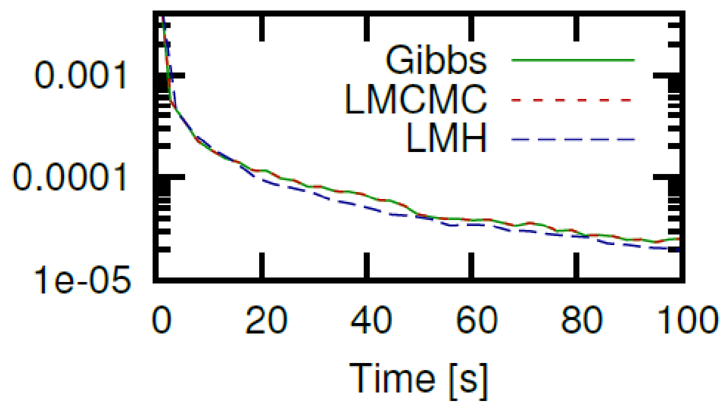
(a) Ising - Iterations



(b) Ising - Time



(c) Chimera - Iterations



(d) Chimera - Time