Improving Mixing times in Gibbs Sampling

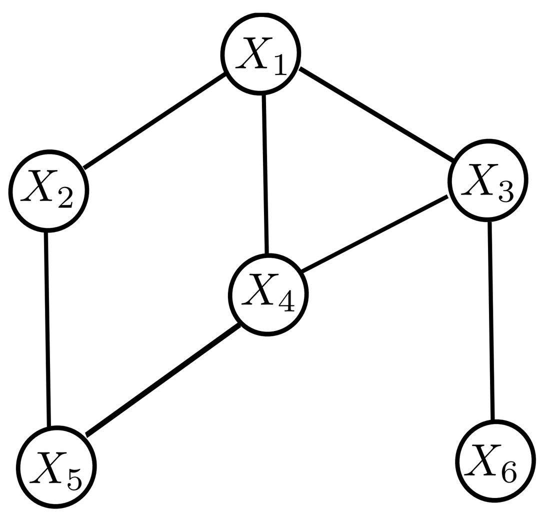
Wenlu Hu   Kirthevasan Kandasamy   Joseph Tassarotti

10-701 Introduction to Machine Learning, Course Project

Summary

Improve mixing time in Gibbs Sampling via Tree Blocks that group correlated variables together.

Introduction



Given: Graphical Model with known Parameters.

Goal: Estimate $\mathbb{E}[h(X)]$
E.g.: $1_A(x)$, $\sum_i X_i$

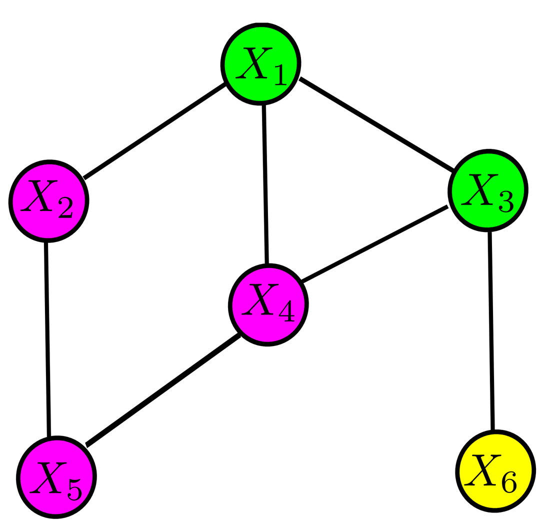
Generate Samples: $X^{(1)}, X^{(2)}, \ldots X^{(N)}$

Empirical Estimator

$$\mu_0 = \frac{1}{N}\sum_{i=1}^{N} h(X^{(i)})$$

Gibbs Sampling–one way to generate samples.

Blocked Gibbs Sampling



Current Sample: $X^{(t)}$

$X_{1,3}^{(t+1)} \mid X_{2,4,5,6}^{(t)}$

$X_{2,4,5}^{(t+1)} \mid X_{1,3}^{(t+1)}, X_6^{(t)}$

$X_6^{(t+1)} \mid X_{1,2,3,4,5}^{(t+1)}$

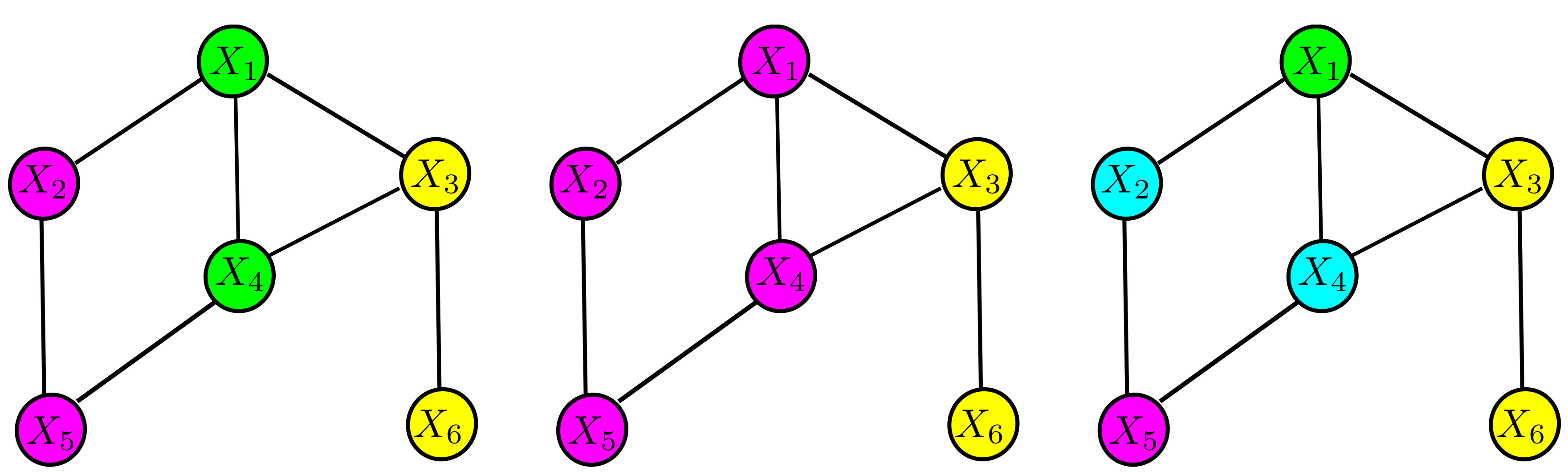
Next Sample: $X^{(t+1)}$

Why ?

Sampling is now more difficult.

BUT, Chain mixes faster $\Longrightarrow$ better samples.

How to Block ?



Strategy

- We will focus only on **Tree Partitions**.
  - Otherwise Problem is too big.
  - Inference on Trees is easy (Belief Propagation converges in linear time).
- Will consider **correlations** between variables when developing tree blocks.

Why Correlations ?

MC: $X^{(1)} \to X^{(2)} \to \cdots \to X^{(t)} \to X^{(t+1)} \to \ldots,$ Eqlbm Distribtuion: π

$L_0(\pi) = \{h : \Omega \to \mathbb{R} :\ \mathbb{E}_\pi h(X) = 0,\ \mathbb{V}_\pi h(X) < \infty\}$, $\langle h, g\rangle = \mathrm{Covar}_\pi(h(X), g(X))$
$(L_0(\pi), \langle \cdot,\cdot \rangle)$ is a Hilbert Space.

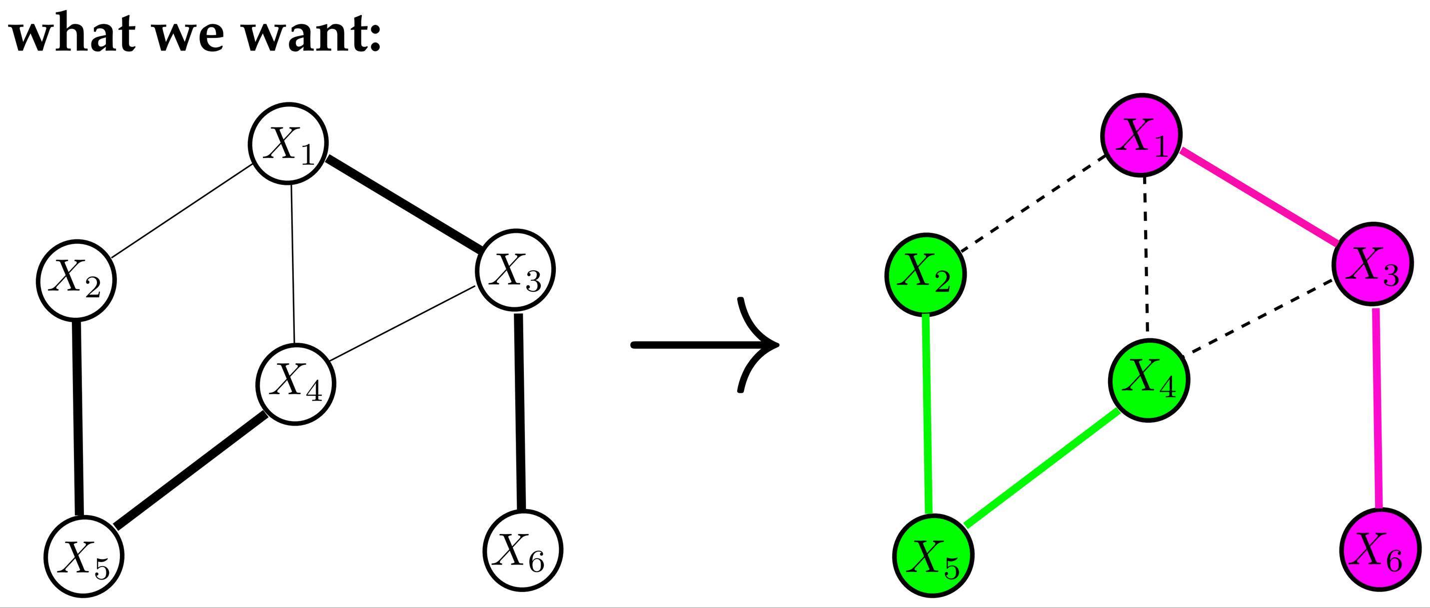
Define $F : L_0(\pi) \to L_0(\pi)$, $[Fh](z) = \mathbb{E}[h(X^{(1)})|X^{(0)} = z]$

Fact: $|\mathbb{E}^{(n)}h(X) - \mathbb{E}_\pi h(X)| \leqslant C\,\|F\|^n\,\|h\|$

$\|F\| = \sup_{f,g} \mathrm{Corr}(f(X^{(t+1)}), g(X^{(t)}))$

Correlated Variables in different blocks $\Longrightarrow$ successive samples correlated.

So this is what we want:



Algorithms

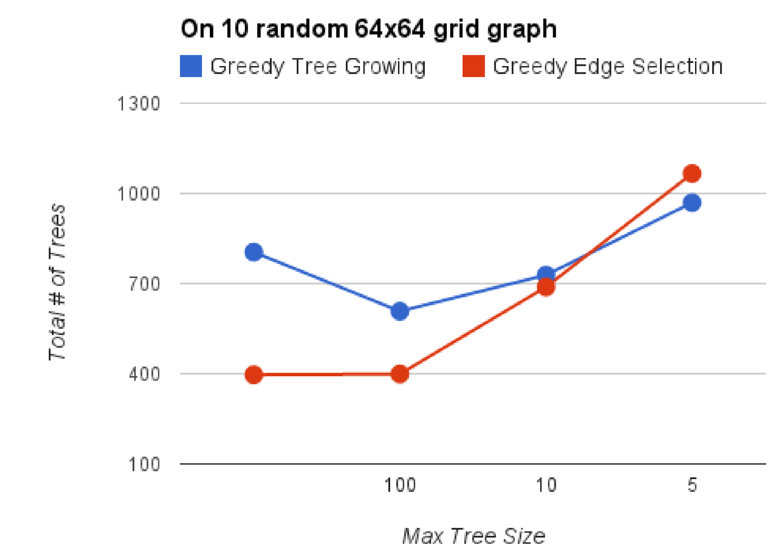
For tree splitting, we consider algorithm from []. They greedily grows trees favoring vertices of low degree. We use its slightly simplified version as our baseline. We compared our algorithm, Greedy Edge Selection Algorithm, with the baseline under two different conditions.

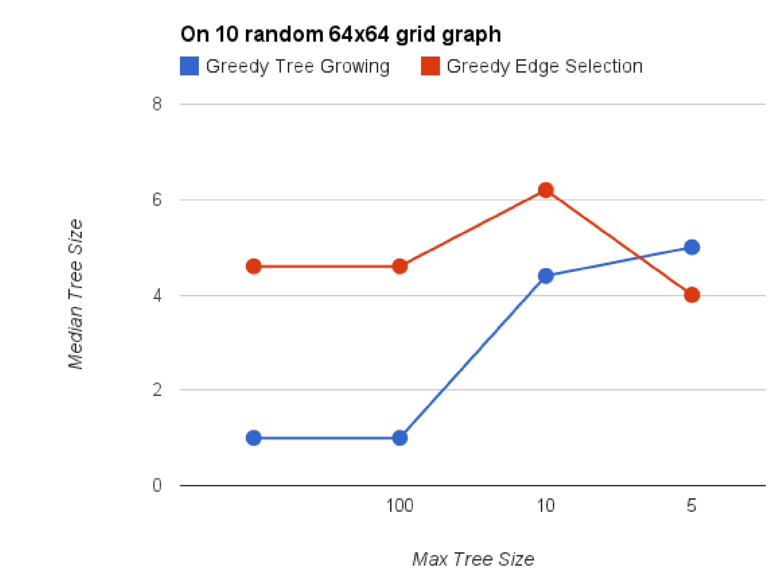
Greedy Edge Selection Algorithm

1. Construct an ordered list of edges, E, with $E[0]$ being the highest weight edge. Edges are vertex pairs (i, j).
2. Initialize an all-zero n-dimensional integer list V of vertex colors. ($V[i]$ is the color of vertex i, and $V[i] = 0$ means that vertex i has not yet been colored.)
3. Initialize n empty vertex sets: $T_1, \ldots, T_n$
(Logically, T_i is the set of vertices labeled with color i.)
4. Initialize $unusedColor = 1$.
5. For each edge $e = (i, j)$ in E,
   - If $V[i] = V[j] = 0$,
     - Set $V[i] = V[j] = unusedColor$
     - Add i, j to $T_{unusedColor}$
     - Increment $unusedColor$ by 1
   - Else if $V[i] = 0$ and $V[j] \notin getOtherNeighborColors(\{i\}, e)$,
     - Set $V[i] = V[j]$
     - Add i to $T_{V[j]}$
   - Else if $V[j] = 0$ and $V[i] \notin getOtherNeighborColors(\{j\}, e)$,
     - Set $V[j] = V[i]$
     - Add j to $T_{V[i]}$
   - Else if $V[i] \neq 0$ and $V[j] \neq 0$ and $V[i] \notin getOtherNeighborColors(T_j, e)$,
     - For each $k \in T_j$, set $V[k] = V[i]$
     - Set $T_i = T_i \cup T_j$
     - Set $T_j = \emptyset$
   - Otherwise do nothing
6. For each vertex i, if $V[i] = 0$, set $V[i] = unusedColor$, $unusedColor{+}{+}$
7. Output $\{T_j : T_j \neq \emptyset\}$

Greedy Tree Growing Algorithm

1. Initialize $i = 0$ and V to the vertex set.
2. While $V \neq \emptyset$
   - Select $v \in V$
   - Start a new tree T_i and a priority queue Q_i. Add v to Q_i
   - While $Q_i \neq \emptyset$
     - Pop u from Q_i.
     - Initialize $neighborsInT = 0$.
     - For all $v \in T_i$, if $u \in N(v)$, increment $neighborsInT$
     - If $neighborsInT \leqslant 1$,
     - Add u to T_i and remove v from V
     - Add $N(u)$ to Q_i.
3. Return $\{T_i\}$





With no limit on the max tree size (the leftmost dots), Greedy Edge Selection Algorithm performs much better than Greedy Tree Growing Algorithm. But when as the limit of max tree size goes down, Greedy Tree Growing Algorithm caught up with Greedy Edge Selection Algorithm.

Lists

- You can make
- lists, that
- allow people to see quickly

Math

Include math within the text is as simple as $1 + 1 = 2$. You can also highlight more important equations like this:

$$\int_0^1 \sin(x) + \cos^2(x) + \alpha x\, dx$$

Pictures

Experiments

Remember to put lots of figures on your poster... Nobody reads anymore!

Conclusion

Much less annoying than PowerPoint. Copy and Paste from your document. Overall, a great idea!

keywords: Gibbs Sampling, Markov Networks, Mixing Times

{wenluh, kkandasa, jtassro} @ cs.cmu.edu