Markov Chain Monte Carlo Sampling

B.Tech Project
GUIDE: PROF. SUYASH AWATE

Saurabh Garg | 140070003

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

November 28, 2017

Why Perfect Sampling

- The main issue with using standard Gibbs sampling just by itself is that, even though we run the iterations for long enough, we never know when we have reached convergence.

Why Perfect Sampling

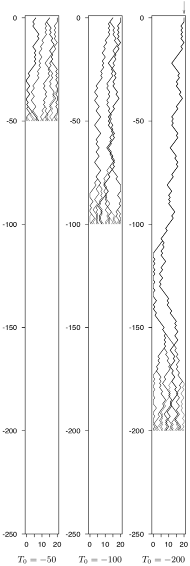
- The main issue with using standard Gibbs sampling just by itself is that, even though we run the iterations for long enough, we never know when we have reached convergence.
- All we know is as $n \to \infty$ we will reach the stationary distribution (π) but there is no mention of how to check whether it has converged to π.

Why Perfect Sampling

- The main issue with using standard Gibbs sampling just by itself is that, even though we run the iterations for long enough, we never know when we have reached convergence.
- All we know is as $n \to \infty$ we will reach the stationary distribution (π) but there is no mention of how to check whether it has converged to π.
- If we do not know when convergence occurs, we might get biased samples which will depend on the initial state.

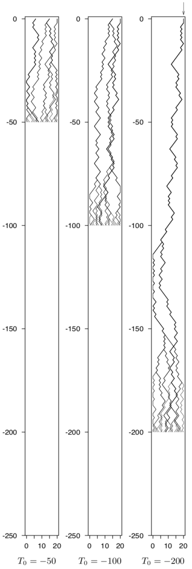
Coupling from the Past

- First idea is to check *coalescence* of coupled Markov chains.



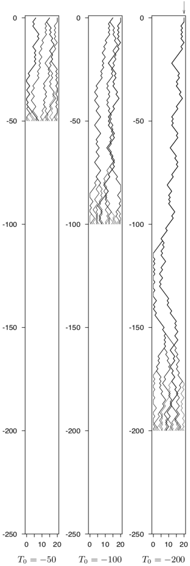
Coupling from the Past

- First idea is to check *coalescence* of coupled Markov chains.
- Second idea is to keep track of all the states and transition all of them using the same random number generated in that time step.



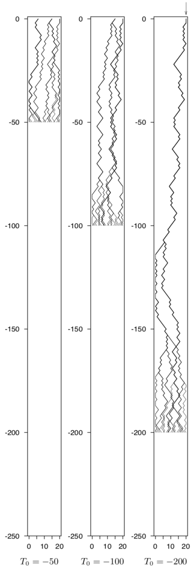
Coupling from the Past

- First idea is to check *coalescence* of coupled Markov chains.
- Second idea is to keep track of all the states and transition all of them using the same random number generated in that time step.
- If the states have not converged, we go back in time and run the sampling procedure again using the same random numbers at time points we have already seen.



Coupling from the Past

- First idea is to check *coalescence* of coupled Markov chains.
- Second idea is to keep track of all the states and transition all of them using the same random number generated in that time step.
- If the states have not converged, we go back in time and run the sampling procedure again using the same random numbers at time points we have already seen.
- Once all the states have coalesced to a single state, we know that convergence has occurred.



Motivation for Fill's Algorithm

- The running time of CFTP algorithm is an unbounded random variable whose order of magnitude is typically unknown a priori and which is not independent of the state sampled

Motivation for Fill's Algorithm

- The running time of CFTP algorithm is an unbounded random variable whose order of magnitude is typically unknown a priori and which is not independent of the state sampled
- A naive user with limited patience who aborts a long run of the algorithm will introduce bias.

Motivation for Fill's Algorithm

- The running time of CFTP algorithm is an unbounded random variable whose order of magnitude is typically unknown a priori and which is not independent of the state sampled
- A naive user with limited patience who aborts a long run of the algorithm will introduce bias.
- Claim of unbiasedness in Fills with respect to user impatience follows from the very nature of acceptance rejection sampling together with the fact that all information is erased after each iteration

Fill's Algorithm

Algorithm Fill's algorithm [1]

$t = 1$
repeat
    $x_t = z$
    Generate $X_{t-1}|x_t, X_{t-2}|x_{t-1}, ..., X_0|x_1$
    Generate $[U_1|x_0 \rightarrow x_1], [U_2|x_1 \rightarrow x_2], ..., [U_t|x_{t-1} \rightarrow x_t]$
    Begin chains $X^{0,1}, X^{0,2}, ...X^{0,k}$ in all possible initial states at time 0
    Use the common $U_1, U_2, ..., U_t$to update chains
    $t = 2t$
until Until chains coalesce
return x_0

- Choose a arbitrary state X_T and move the chain for T steps.

[1] J.A. Fill, An interruptible algorithm for perfect sampling via Markov chains

Fill's Algorithm

Algorithm Fill's algorithm [1]

```
1: t = 1
2: repeat
3:     x_t = z
4:     Generate X_{t-1}|x_t, X_{t-2}|x_{t-1}, ...., X_0|x_1
5:     Generate [U_1|x_0 → x_1], [U_2|x_1 → x_2], ..., [U_t|x_{t-1} → x_t]
6:     Begin chains X^{0,1}, X^{0,2}, ....X^{0,k} in all possible initial states at time 0
7:     Use the common U_1, U_2, ..., U_t to update chains
8:     t = 2t
9: until Until chains coalesce
10: return x_0
```

- Choose a arbitrary state X_T and move the chain for T steps.
- Begin all chains at $T = 0$ and move them in **synchrony** with the evolution of X.

[1] J.A. Fill, An interruptible algorithm for perfect sampling via Markov chains

Fill's Algorithm

Algorithm Fill's algorithm [1]

1: $t = 1$
2: **repeat**
3: $x_t = z$
4: Generate $X_{t-1}|x_t, X_{t-2}|x_{t-1}, ..., X_0|x_1$
5: Generate $[U_1|x_0 \rightarrow x_1], [U_2|x_1 \rightarrow x_2], ..., [U_t|x_{t-1} \rightarrow x_t]$
6: Begin chains $X^{0,1}, X^{0,2}, ...X^{0,k}$ in all possible initial states at time 0
7: Use the common $U_1, U_2, ..., U_t$to update chains
8: $t = 2t$
9: **until** Until chains coalesce
10: **return** x_0

- Choose a arbitrary state X_T and move the chain for T steps.
- Begin all chains at $T = 0$ and move them in **synchrony** with the evolution of X.
- If the chains have coalesced by time T, then accept z as a draw from π. Otherwise begin again, possibly with a new T and z.

[1] J.A. Fill, An interruptible algorithm for perfect sampling via Markov chains

Bounding Chains

- For monotone chains partial order exists on states. The partial order implies that there is a minimum and maximum state and the convergence can be concluded when these states meet.

Bounding Chains

- For monotone chains partial order exists on states. The partial order implies that there is a minimum and maximum state and the convergence can be concluded when these states meet.
- But, partial order doesn't exist for many systems, even if exists, it is not easy to find.

Bounding Chains

- For monotone chains partial order exists on states. The partial order implies that there is a minimum and maximum state and the convergence can be concluded when these states meet.
- But, partial order doesn't exist for many systems, even if exists, it is not easy to find.
- M' $= \{Y_t | t \geq 0\}$ is a bounding chain for M $= \{X_t | t \geq 0\}$ if there exists a coupling between M' and M such that

$$X_t(v) \in Y_t(v) \quad \forall v \implies X_{t+1}(v) \in Y_{t+1}(v) \quad \forall v$$

Bounding Chains

- For monotone chains partial order exists on states. The partial order implies that there is a minimum and maximum state and the convergence can be concluded when these states meet.
- But, partial order doesn't exist for many systems, even if exists, it is not easy to find.
- M' $= \{Y_t | t \geq 0\}$ is a bounding chain for M $= \{X_t | t \geq 0\}$ if there exists a coupling between M' and M such that

$$X_t(v) \in Y_t(v) \quad \forall v \implies X_{t+1}(v) \in Y_{t+1}(v) \quad \forall v$$

- If we can do this, and we let $Y_0(v) = C \quad \forall v$, then when we reach the state with $|Y_0(v)| = 1 \quad \forall v$,we can say for sure that the chain converged.

Algorithm Bounding chain for Gibbs sampler [2]

Choose $v \in_U \{1, 2, ...n\}$, Let N_v be the neighbours of v
Let $y(v) \leftarrow \phi$
repeat
    Choose $c \in_U \{1, 2, ..., k\}$
    Choose $u \in_U [0, 1]$
    Let b_c be the w neighbouring v with y(w) = {c}
    Let d_c be the w neighbouring v with c $\in$ y(w)
    if $u < \gamma^{f_{max}(N_v^{b_c, d_c})}$ **then**
        Let $y(v) \leftarrow y(v) \cup \{c\}$
until $u \leq \gamma^{f_{min}(N_v^{b_c, d_c})}$ or $|y(v)| > \Delta$

- Choose a arbitrary state v and initialize its set of possible colors with an empty state.

[2] M.Huber, Perfect Sampling Using Bounding Chains, 2004

Algorithm Bounding chain for Gibbs sampler [2]

Choose $v \in_U \{1, 2, ...n\}$, Let N_v be the neighbours of v
Let $y(v) \leftarrow \phi$
repeat
    Choose $c \in_U \{1, 2, ..., k\}$
    Choose $u \in_U [0, 1]$
    Let b_c be the w neighbouring v with y(w) = {c}
    Let d_c be the w neighbouring v with c ∈ y(w)
    if $u < \gamma^{f_{max}(N_v^{b_c, d_c})}$ **then**
        Let $y(v) \leftarrow y(v) \cup \{c\}$
until $u \leq \gamma^{f_{min}(N_v^{b_c, d_c})}$ or $|y(v)| > \Delta$

- Choose a arbitrary state v and initialize its set of possible colors with an empty state.
- Add colors to the set depending on the convergence of the extremal states.

[2] M.Huber, Perfect Sampling Using Bounding Chains, 2004

Algorithm Bounding chain for Gibbs sampler [2]

1: Choose $v \in_U \{1, 2, ...n\}$, Let N_v be the neighbours of v
2: Let $y(v) \leftarrow \phi$
3: **repeat**
4: Choose $c \in_U \{1, 2, ..., k\}$
5: Choose $u \in_U [0, 1]$
6: Let b_c be the w neighbouring v with y(w) = {c}
7: Let d_c be the w neighbouring v with c ∈ y(w)
8: **if** $u < \gamma^{f_{max}(N_v^{b_c,d_c})}$ **then**
9: Let $y(v) \leftarrow y(v) \cup \{c\}$
10: **until** $u \leq \gamma^{f_{min}(N_v^{b_c,d_c})}$ or $|y(v)| > \Delta$

- Choose a arbitrary state v and initialize its set of possible colors with an empty state.
- Add colors to the set depending on the convergence of the extremal states.
- If all chains for that state gets a color, continue the procedure with some randomly chosen state.

[2] M.Huber, Perfect Sampling Using Bounding Chains, 2004

Prefect Sampling with Bounding Chains

- Bounding chains helps to detect convergence of a chain, but it does not define the stopping point and so if we stop at early point and output the sample, we might get biased samples.

Prefect Sampling with Bounding Chains

- Bounding chains helps to detect convergence of a chain, but it does not define the stopping point and so if we stop at early point and output the sample, we might get biased samples.
- So we do a perfect sampling procedure above this to get unbiased or "perfect" samples.

Prefect Sampling with Bounding Chains

- Bounding chains helps to detect convergence of a chain, but it does not define the stopping point and so if we stop at early point and output the sample, we might get biased samples.
- So we do a perfect sampling procedure above this to get unbiased or "perfect" samples.
    - Coupling from the past

Prefect Sampling with Bounding Chains

- Bounding chains helps to detect convergence of a chain, but it does not define the stopping point and so if we stop at early point and output the sample, we might get biased samples.
- So we do a perfect sampling procedure above this to get unbiased or "perfect" samples.
  - Coupling from the past
  - Fill's Algorithm

Algorithm CFTP with bounding chains (Proposed)

$T = 1$
repeat
  $Y(v) = \mathcal{C} \quad \forall v \in \{1, 2..., n\}$
  for $t = 1 : T$ **do**
    if $t < T/2$ **then**
      $U_t \leftarrow$ Random numbers used in previous t^{th} run
    else
      $U_t \leftarrow U[0,1]^V$
    for each $v \in \{1, 2, ...n\}$ **do**
      Let N_v be the neighbours of v
      Let $Y(v) \leftarrow \phi$
      repeat
        Choose $c \in_U \{1, 2, ..., k\}$
        Choose $u \in_U U_t$
        Let b_c be the w neighbouring v with $Y(w) = \{c\}$
        Let d_c be the w neighbouring v with c $\in Y(w)$
        if $u < \gamma^{f_{max}(N_v^{b_c, d_c})}$ **then**
          Let $Y(v) \leftarrow Y(v) \cup \{c\}$
      until $u \leq \gamma^{f_{min}(N_v^{b_c, d_c})}$ or $|Y(v)| > \Delta$
  $T = 2T$
until $|Y(v)| = 1 \quad \forall v \in \{1, 2...., n\}$
return x_T s.t if $c \in Y(v)$ then $x_T(v) = c$

Algorithm Fill's algorithm with bounding chains (Proposed)

$T = 1$
repeat
  $Y(v) = \mathcal{C} \quad \forall v \in \{1, 2..., n\}$
  $x_T = z \in_U \{1, 2..., k\}^V$
  Generate $X_{T-1}|x_T, X_{T-2}|x_{T-1}, ..., X_0|x_1$
  Generate $[U_1|x_0 \to x_1], [U_2|x_1 \to x_2], ..., [U_T|x_{T-1} \to x_T]$
  for $t = 1 : T$ **do**
    for each $v \in \{1, 2, ...n\}$ **do**
      Let N_v be the neighbours of v
      Let $Y(v) \leftarrow \phi$
      repeat
        Choose $c \in_U \{1, 2, ..., k\}$
        Choose $u \in_U U_t^c$
        Let b_c be the w neighbouring v with $Y(w) = \{c\}$
        Let d_c be the w neighbouring v with c $\in Y(w)$
        if $u < \gamma^{f_{max}(N_v^{b_c, d_c})}$ **then**
          Let $Y(v) \leftarrow Y(v) \cup \{c\}$
      until $u \leq \gamma^{f_{min}(N_v^{b_c, d_c})}$ or $|Y(v)| > \Delta$
  $T = 2T$
until $|Y(v)| = 1 \quad \forall v \in \{1, 2...., n\}$
return x_0

Note Points

- Fill's algorithm for generation of perfect samples can be interrupted at any time, and yet, the samples generated are not biased because, the running time of the algorithm and the returned values are independent.

Note Points

- Fill's algorithm for generation of perfect samples can be interrupted at any time, and yet, the samples generated are not biased because, the running time of the algorithm and the returned values are independent.
- Fill's algorithm simulates reversed Markov chains using a updating function ϕ s.t. $x^{(t+1)} = \phi(x^{(t)}, R^{(t-1)})$

Note Points

- Fill's algorithm for generation of perfect samples can be interrupted at any time, and yet, the samples generated are not biased because, the running time of the algorithm and the returned values are independent.
- Fill's algorithm simulates reversed Markov chains using a updating function ϕ s.t. $x^{(t+1)} = \phi(x^{(t)}, R^{(t-1)})$
- Thus, Fill's algorithm is similar to CFTP in that it uses Markov chains to produce perfect samples, but it is based on the idea of rejection sampling instead of the concept of coupling.

Applications of Perfect Sampling

Perfect Sampling from Posteriors

- **Potts Model** is defined over a graph with vertex set V and a set of possible colors $\mathcal{C}$ such that the probability of being in state $X \in \mathcal{C}^V$ is given by,

$$\pi(X) = \frac{e^{\beta \sum_i J(X_i)}}{Z_\beta} \quad \textit{where} \quad J(X_i) = \sum_{j \in V} W_{i,j} \mathbb{I}_{X_i = X_j}$$

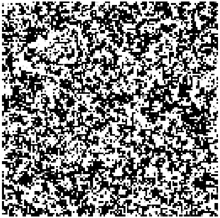
where Z_β is normalization constant and β is a free parameter.

Perfect Sampling from Posteriors

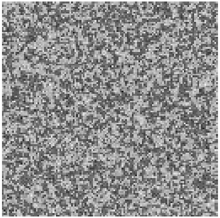
- **Potts Model** is defined over a graph with vertex set V and a set of possible colors $\mathcal{C}$ such that the probability of being in state $X \in \mathcal{C}^V$ is given by,

$$\pi(X) = \frac{e^{\beta \sum_i J(X_i)}}{Z_\beta} \quad where \quad J(X_i) = \sum_{j \in V} W_{i,j} \mathbb{I}_{X_i = X_j}$$

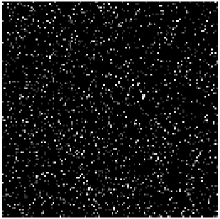
  where Z_β is normalization constant and β is a free parameter.
- To sample from a posterior $P(X_i|Z_i)$ with an i.i.d likelihood model and a Potts model prior, we use bounding chains with transition function defined using the posterior probabilities.



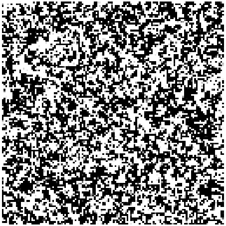
(a) Prior Image



(b) Data Image



(c) Variance Image



(d) Mean Posterior Sample

Figure 1: Posterior Sampling: Image Samples when $\mu_1 = 10$, $\mu_2 =$ -10, $\sigma_1 = 5$, and $\sigma_2 = 5$. Number of samples generated 20. Error of 3.3%.

Expectation Maximization with perfect sampling

- In the problem of soft image segmentation with MRF prior on label image and Gaussian mixture mode likelihood, the E step expectation is analytically intractable

[3] Zhang, Brady, and Smith. "Segmentation of brain MR images through a hidden Markov random field model and the EM algorithm.", 2001 IEEE TMI

Expectation Maximization with perfect sampling

- In the problem of soft image segmentation with MRF prior on label image and Gaussian mixture mode likelihood, the E step expectation is analytically intractable
- Instead of approximating the E-step with an approximate distribution whose expectation is easy to obtain, we retain the original distribution and perform perfect sampling on this distribution[3]

$$E_{P(x_i, x_{\sim i}|y,\theta^t)}[\log P(y_i|x_i,\theta)] \approx E_{P(x_i|x_{\sim i}y,\theta^t)}[\log P(y_i|x_i,\theta)]$$

[3] Zhang, Brady, and Smith. "Segmentation of brain MR images through a hidden Markov random field model and the EM algorithm.", 2001 IEEE TMI

Expectation Maximization with perfect sampling

- In the problem of soft image segmentation with MRF prior on label image and Gaussian mixture mode likelihood, the E step expectation is analytically intractable
- Instead of approximating the E-step with an approximate distribution whose expectation is easy to obtain, we retain the original distribution and perform perfect sampling on this distribution[3]

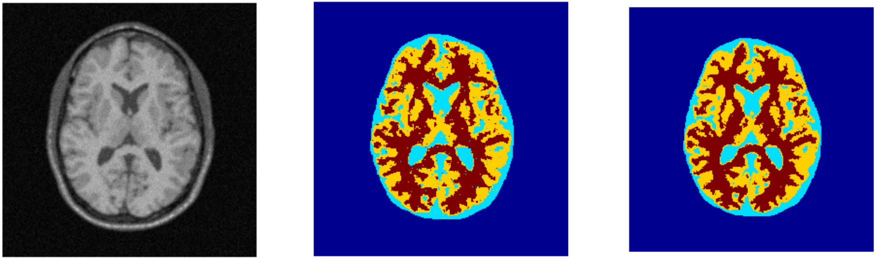
$$E_{P(x_i, x_{\sim i} | y, \theta^t)}[\log P(y_i | x_i, \theta)] \approx E_{P(x_i | x_{\sim i} y, \theta^t)}[\log P(y_i | x_i, \theta)]$$

- We estimate the expectation using the average of a few samples that are obtained using the perfect sampling procedure.

$$E_{P(x_i, x_{\sim i} | y, \theta^t)}[\log P(y_i | x_i, \theta)] \approx \frac{1}{\mathcal{S}} \sum_{\substack{s=1 \\ x^s \sim P(x|y,\theta^t)}}^{\mathcal{S}} \log P(y_i | x_i^s, \theta)$$

[3] Zhang, Brady, and Smith. "Segmentation of brain MR images through a hidden Markov random field model and the EM algorithm.", 2001 IEEE TMI

Brain Image Segmentation



(a) Brain MRI Image (b) Approximations in E step (c) Perfect Sampling in E step

Figure 2: Label MAP for Segmentation on Brain MRI

Uncertainty Estimation with Perfect Sampling

- Need for perfect sampling from a posterior distribution for uncertainty estimation in various real life scenarios like lesion detection.

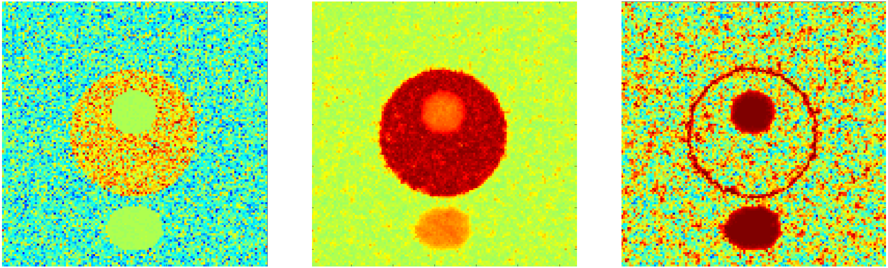
Uncertainty Estimation with Perfect Sampling

- Need for perfect sampling from a posterior distribution for uncertainty estimation in various real life scenarios like lesion detection.
- Insufficient burn-in for Gibbs sampling artificially inflates the variance and hence incorrect uncertainty estimations.

Uncertainty Estimation with Perfect Sampling

- Need for perfect sampling from a posterior distribution for uncertainty estimation in various real life scenarios like lesion detection.
- Insufficient burn-in for Gibbs sampling artificially inflates the variance and hence incorrect uncertainty estimations.
- Conservatively estimating burn-in can make the burn-in iterations too large making the Gibbs sampling too slow.

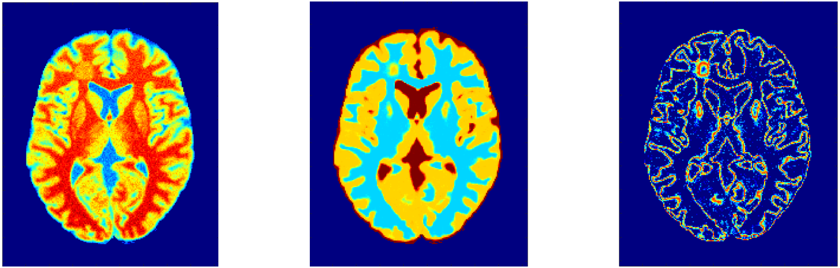
Uncertainty Estimation on Simulated Data



(a) Original Image (b) Mean Label Image (c) Variance Image

Figure 3: Uncertainty Estimation over samples from estimated posterior on simulation of lesion

Uncertainty Estimation on Brain Image Data



(a) MRI with noise and bias (b) Mean Label Image (c) Variance Image

Figure 4: Uncertainty Estimation over samples from estimated posterior on Brain slice with subcortical structure and simulated lesion

Future Work

- Uncertainty estimation on other realistic problem like tumor segmentation, Infant MRI segmentation, Multi-Atlas segmentation.

Future Work

- Uncertainty estimation on other realistic problem like tumor segmentation, Infant MRI segmentation, Multi-Atlas segmentation.
- Many image processing algorithms require one to sample from continuous state spaces. There is a lot of theory on perfect sampling from continuous state spaces which we are yet to explore.

Questions?

Thank You!