

Posterior Expectation of Regularly Paved Random Histograms

Raazesh Sainudiin

joint work with: Jennifer Harlow, Dominic Lee and Gloria Teng

Department of Mathematics and Statistics, University of Canterbury,
Christchurch, New Zealand

April 12 2012,

Stochastic Networks Workshop, Auckland University, New Zealand, April 12-13, 2010

Massive Metric Data Streams

Air Traffic Examples (Teng, Kuhn and S., *Jnl. Aerospace Comp., Inf. & Commun.*, [acc.] 2012)

Synthetic Examples (Teng, Harlow, Lee and S., *ACM Trans. Mod. & Comp. Sim.*, [r. 2] 2012)

Regular Pavings (RPs)

Statistical Regular Pavings (SRPs)

Adaptive Histograms

S.E.B. Priority Queue

Arithmetic on SRPs

Posterior Expectation over Histograms in $\mathbb{S}_{0:\infty}$

Examples - good, bad and ugly

Conclusions and References

Massive Metric Data Streams – Introduction

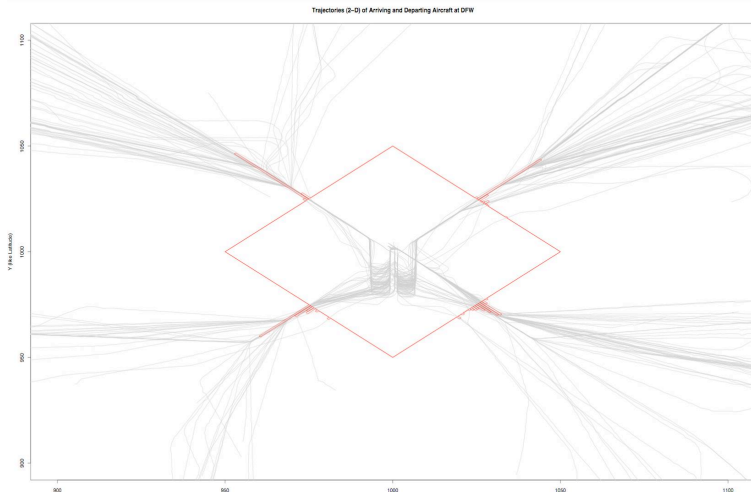
- ▶ A massive metric data stream is:

$$\dots, X_{-3}, X_{-2}, X_{-1}, X_0, \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_n, X_{n+1}, \dots \sim F, \quad X_i \in \mathbb{R}^d.$$

- ▶ Large Dimension: $1 \leq d \leq 1000$
- ▶ Huge Observations: $10^6 \leq n \leq 10^{10}$
- ▶ Need an **efficient** and **sufficient** multi-dimensional metric data-structure for non-parametric inference that is capable of:
 1. L_1 -consistent density estimation – adaptive histograms
 2. Extend Arithmetic over a dense class of Lipschitz \mathbb{M} -valued maps: $\{g : \mathbb{R}^d \rightarrow \mathbb{M}\}$ – functional estimation when $\mathbb{M} = \mathbb{R}$

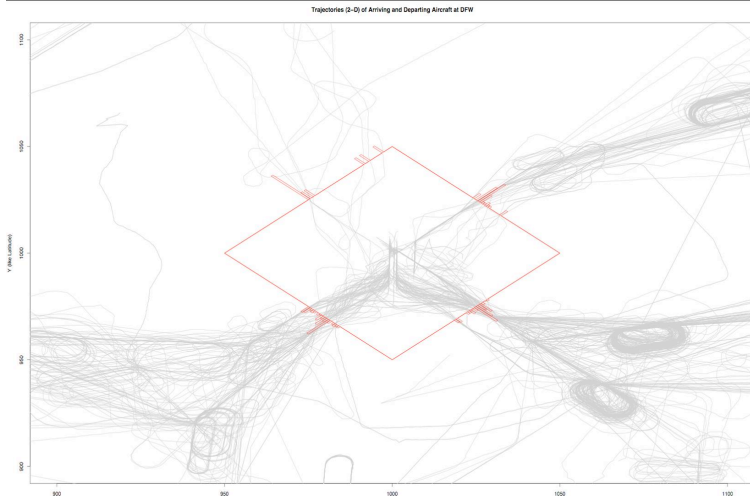
Massive Metric Data Streams – Air Traffic Example

On a Sunny Day



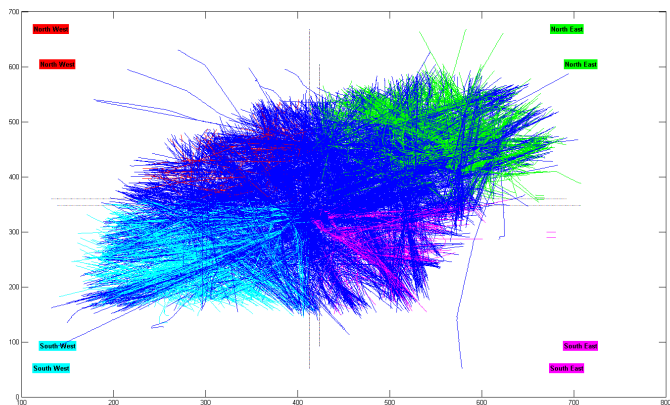
Massive Metric Data Streams – Air Traffic Example

On a Rainy Day



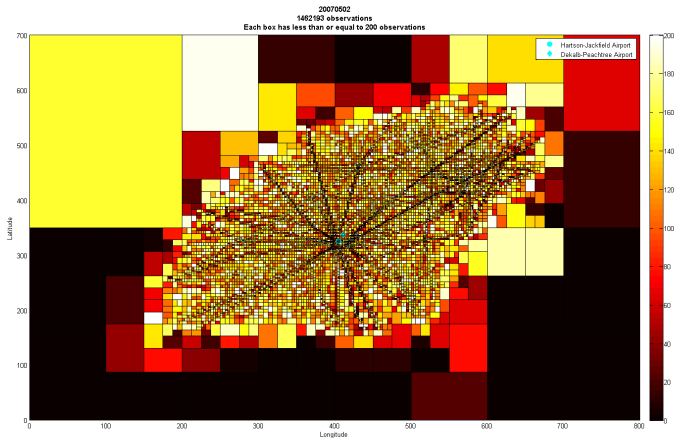
Massive Metric Data Streams – Air Traffic Example

We want to make sense of trajectories like these



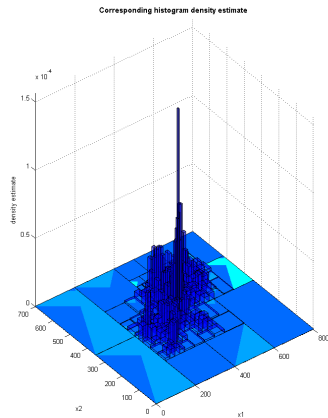
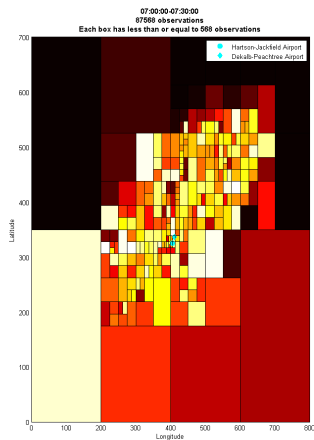
Massive Metric Data Streams – Air Traffic Example

using a picture like this



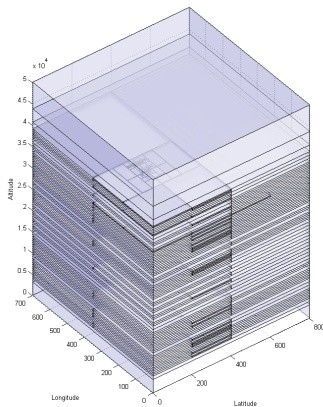
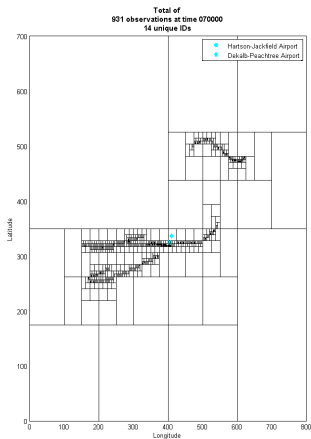
Massive Metric Data Streams – Air Traffic Example

A Histogram Estimate of Air-traffic between 0700 – 0730 hours



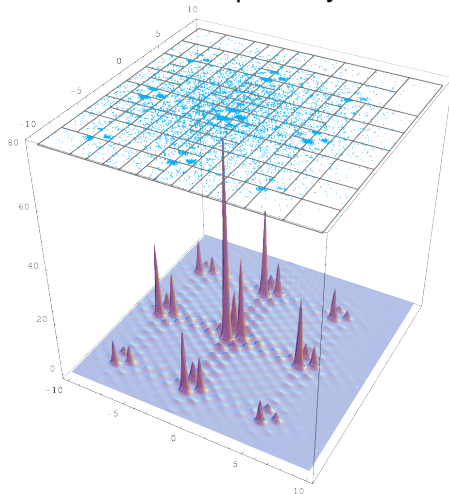
Massive Metric Data Streams – Air Traffic Example

Add the pavings of 14 flight trajectories like this



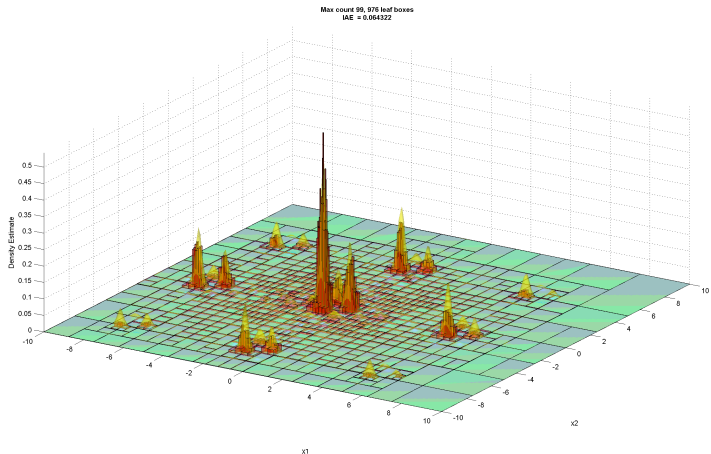
Massive Metric Data Streams – Synthetic Examples

Take millions of **realizations** of a possibly ‘challenging’ density



Massive Metric Data Streams – Synthetic Examples

and produce a consistent estimate of the density



Intervals and Boxes in \mathbb{R}^d

Intervals and *Boxes* as interval vectors:

$$\mathbf{x} = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \times \dots \times [\underline{x}_d, \bar{x}_d], \quad \underline{x}_i \leq \bar{x}_i .$$

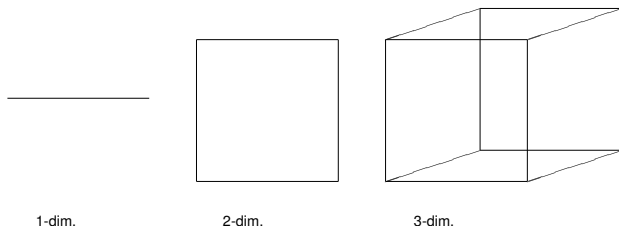


Figure: Boxes in 1D, 2D, and 3D.

Binary Tree Representation

These boxes can be represented by ordered binary trees, a.k.a.:

- *plane binary trees* of **enumerative combinatorics**
- *finite rooted binary (frb-trees)* of **geometric group theory**

An operation of bisection on a box is equivalent to performing the operation on its corresponding node in the tree:

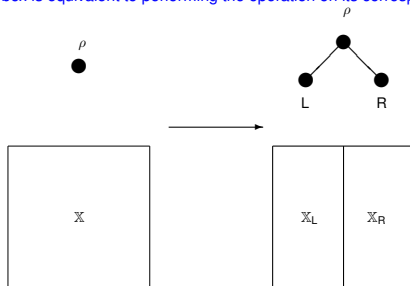


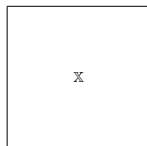
Figure: Bisecting a box or its equivalent node.

Regular Pavings (RPs)

- ▶ A sequence of bisections of boxes;
- ▶ Start from the root box;
- ▶ Along the first widest dimension.

A sequence of bisections on root box \mathbb{X} to get a 4-leaved RP.

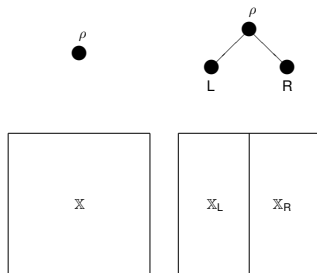
ρ



Regular Pavings (RPs)

- ▶ A sequence of bisections of boxes;
- ▶ Start from the root box;
- ▶ Along the first widest dimension.

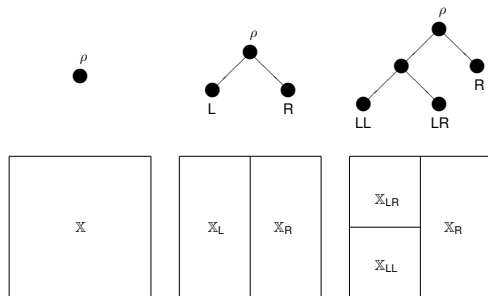
A sequence of bisections on root box \mathbb{X} to get a 4-leaved RP.



Regular Pavings (RPs)

- ▶ A sequence of bisections of boxes;
- ▶ Start from the root box;
- ▶ Along the first widest dimension.

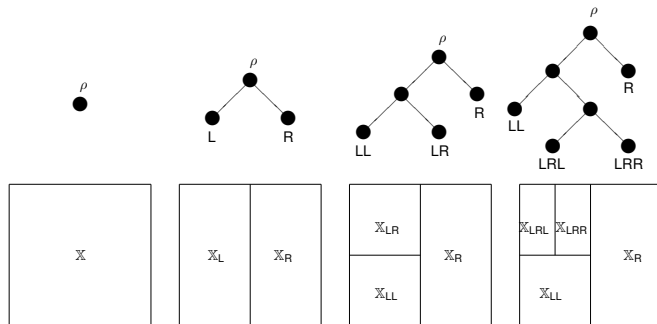
A sequence of bisections on root box \mathbb{X} to get a 4-leaved RP.



Regular Pavings (RPs)

- ▶ A sequence of bisections of boxes;
- ▶ Start from the root box;
- ▶ Along the first widest dimension.

A sequence of bisections on root box \mathbb{X} to get a 4-leaved RP.

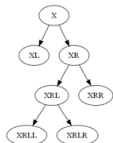


State Space of Regular Pavings

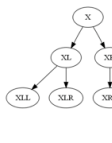
Leaf-depth encoded RPs



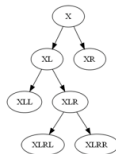
(3, 3, 2, 1)



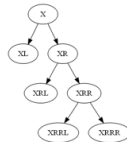
(1, 3, 3, 2)



(2, 2, 2, 2)



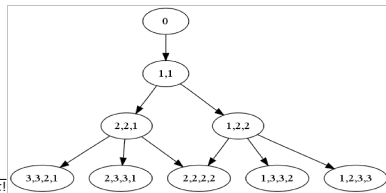
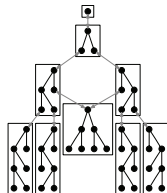
(2, 3, 3, 1)



(1, 2, 3, 3)

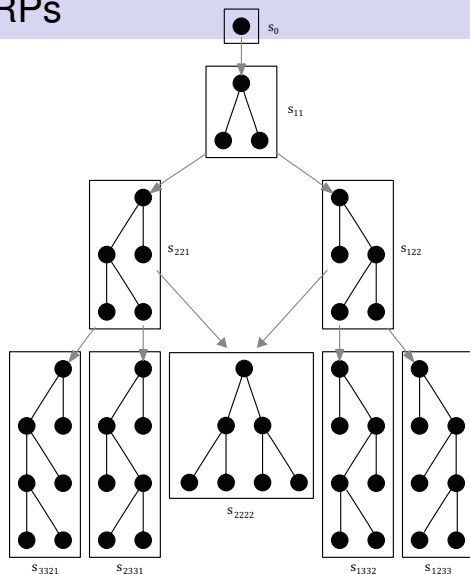
C_k RPs with k splits

- $C_0 = 1$
- $C_1 = 1$
- $C_2 = 2$
- $C_3 = 5$
- $C_4 = 14$
- $C_5 = 42$
- $C_k = \frac{(2k)!}{(k+1)!k!}$



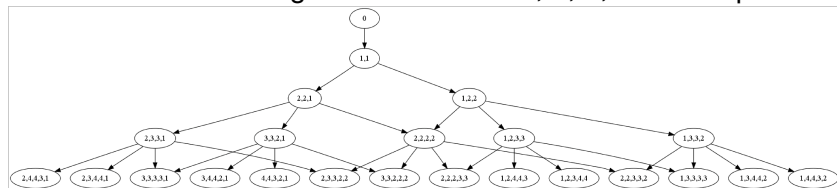
The Space of All Possible RPs

- ▶ Let \mathbb{S}_i be the set of all RPs of \mathbb{X} made of i splits and
- ▶ Let $\mathbb{S}_{i:j}$ be the set of RPs with k splits where $k \in \{i, i+1, \dots, j\}$
- ▶ The space of all RPs is $\mathbb{S}_{0:\infty} := \lim_{j \rightarrow \infty} \mathbb{S}_{0:j}$
- ▶ RPs are closed under pair-wise union (or overlay) operations (Thompson's Group)
- ▶ can get as m_∞ -close as desired to any subset of \mathbb{X}



State Transition Diagram of Regular Pavings

State Transition Diagram of RPs with 0, 1, 2, 3 and 4 splits.



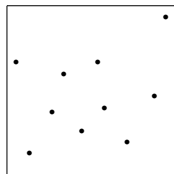
1. The above state space is denoted by $\mathbb{S}_{0:4}$
2. Number of RPs with k splits is the Catalan number C_k
3. There is more than one way to reach a RP by k splits
4. Randomized algorithms here are Markov chains on $\mathbb{S}_{0:\infty}$

Statistical Regular Pavings (SRPs)

- ▶ Extended from the RP;
- ▶ Caches recursively computable statistics at each box or node as data falls through;
- ▶ These statistics include:
 - ▶ the sample count;
 - ▶ the sample mean vector;
 - ▶ the sample variance-covariance matrix;
 - ▶ and the volume of the box.

Caching the sample count in each node (or box).

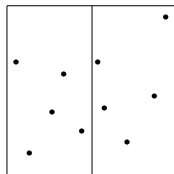
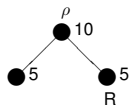
ρ
● 10



Statistical Regular Pavings (SRPs)

- ▶ Extended from the RP;
- ▶ Caches recursively computable statistics at each box or node as data falls through;
- ▶ These statistics include:
 - ▶ the sample count;
 - ▶ the sample mean vector;
 - ▶ the sample variance-covariance matrix;
 - ▶ and the volume of the box.

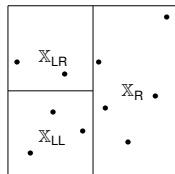
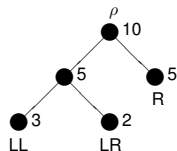
Caching the sample count in each node (or box).



Statistical Regular Pavings (SRPs)

- ▶ Extended from the RP;
- ▶ Caches recursively computable statistics at each box or node as data falls through;
- ▶ These statistics include:
 - ▶ the sample count;
 - ▶ the sample mean vector;
 - ▶ the sample variance-covariance matrix;
 - ▶ and the volume of the box.

Caching the sample count in each node (or box).



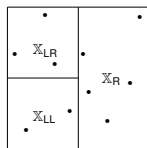
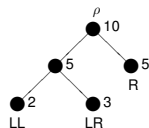
SRPs as Adaptive Histograms

SRP estimate of f from random vectors $X_1, X_2, \dots, X_n \stackrel{iid}{\sim} f$ is

$$f_{n,\hat{s}}(x) = \frac{1}{n} \sum_{i=1}^n \frac{\mathbb{1}(X_i \in \mathbf{x}(x))}{\text{vol}(\mathbf{x}(x))},$$

$\mathbf{x}(x) \in \ell(\hat{s})$ is the leaf box containing x with volume $\text{vol}(\mathbf{x}(x))$

Figure: A SRP as a histogram estimate.



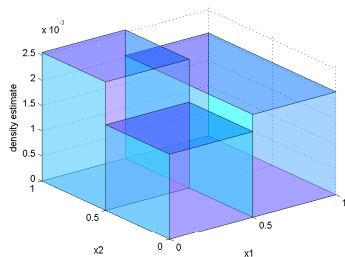
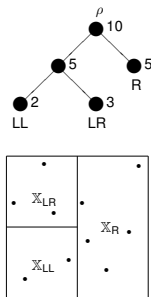
SRPs as Adaptive Histograms

SRP estimate of f from random vectors $X_1, X_2, \dots, X_n \stackrel{iid}{\sim} f$ is

$$f_{n,\hat{s}}(x) = \frac{1}{n} \sum_{i=1}^n \frac{\mathbb{1}(x_i \in \mathbf{x}(x))}{\text{vol}(\mathbf{x}(x))},$$

$\mathbf{x}(x) \in \ell(\hat{s})$ is the leaf box containing x with volume $\text{vol}(\mathbf{x}(x))$

Figure: A SRP as a histogram estimate.

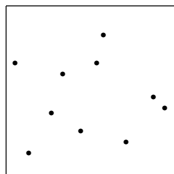


A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

ρ
● 10



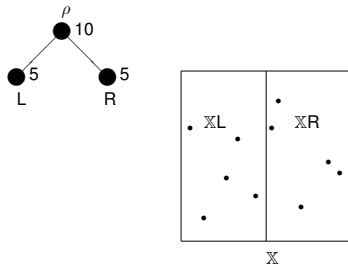
X

A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

Split the root box.

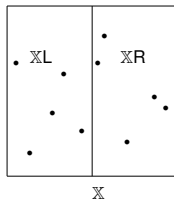
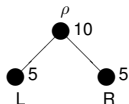


A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

Two or more boxes with the most number of points?

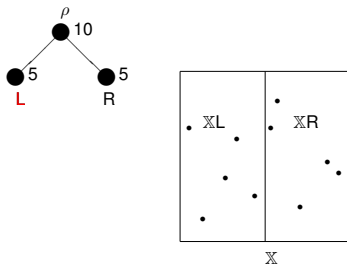


A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

Break such ties by randomising the next bisection.

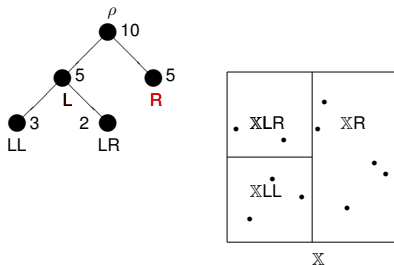


A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

Bisect until each box has $\leq k_n$ points (let $k_n = 3$ here).

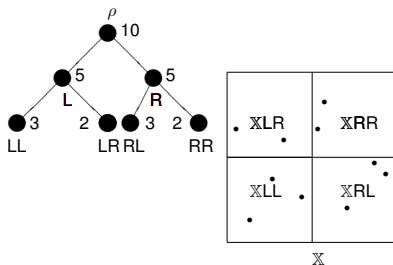


A Prioritized Queue based Algorithm

Algorithm `SplitMostCounts`

As data arrives, order the leaf boxes of the SRP so that the leaf box with **the most number of points** will be chosen for the next bisection.

Final state



The SplitMostCounts Algorithm

Input: (i) data: $x_1, \dots, x_n \subseteq \mathbb{R}^d$; (ii) root box: \mathbb{X} // optional;
 (iii) padding to handle pulsed data: $\psi \geq 0$ // optional;
 (iv) S.E.B. max: \bar{k}_n ; (v) maximum partition size: \bar{m}_n .

Output: histogram estimate $f_{n,s}$.

initialize $i \leftarrow 1$; $s \leftarrow \mathbb{X} + \psi$;

repeat until

$\# \mathbf{x} \leq \bar{k}_n$ for each $\mathbf{x} \in \ell(s)$ and $i \leq \bar{m}_n$ // $\ell(s) = \{\text{leaf boxes}\}$

$\mathbf{x} \leftarrow \text{Uniform}(\hat{\ell}(s))$ // randomized PQ on leaf boxes

$s \leftarrow \text{bisect}(s, \mathbf{x})$ // bisect leaf box \mathbf{x} of s

recursively update counts in s ;

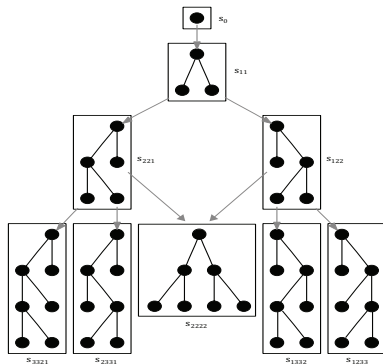
$i \leftarrow i + 1$;

return $f_{n,s}$.

- └ Adaptive Histograms
- └ S.E.B. Priority Queue

Transition Diagram of Randomized PQ Markov chain

Let \mathcal{S}_i be the set of all RPs of \mathbb{X} made of i splits and for $i, j \in \mathbb{N}$ with $i \leq j$, let $\mathcal{S}_{i:j}$ be the set of RPs with k splits, $i \leq k \leq j$.



All possible RP partitions in $\mathcal{S}_{0:4}$.

Proposition: L_1 -Consistency of Histogram Estimates from SplitMostCounts

Let X_1, X_2, \dots be independent and identical random vectors in \mathbb{R}^d whose common distribution μ has a non-atomic density f , i.e., $f \ll \lambda^d$. Let $\{S_n(i)\}_{i=0}^j$ on $\mathbb{S}_{0:\infty}$ be the Markov chain formed using SplitMostCounts with terminal state \dot{s} and histogram estimate $f_{n,\dot{s}}$ over the collection of partitions \mathcal{L}_n .

As $n \rightarrow \infty$, if $\bar{k}_n \rightarrow \infty$, $n^{-1}\bar{k}_n \rightarrow 0$, $\bar{m}_n \geq n/\bar{k}_n$, and $\bar{m}_n/n \rightarrow 0$ then the density estimate $f_{n,\dot{s}}$ is strongly consistent in L_1 , i.e.

$$\int |f(x) - f_{n,\dot{s}}(x)| dx \rightarrow 0 \text{ with probability 1.}$$

Proof Sketch

We will assume that $\bar{k}_n \rightarrow \infty$, $n^{-1}\bar{k}_n \rightarrow 0$, $\bar{m}_n \geq n/\bar{k}_n$, and $\bar{m}_n/n \rightarrow 0$, as $n \rightarrow \infty$, and show that the three conditions:

- (a) $n^{-1}m(\mathcal{L}_n) \rightarrow 0$,
- (b) $n^{-1} \log \Delta_n^*(\mathcal{L}_n) \rightarrow 0$, and
- (c) $\mu(\mathbf{x} : \text{diam}(\mathbf{x}(\mathbf{x})) > \gamma) \rightarrow 0$ with probability 1 for every $\gamma > 0$,

are satisfied. Then by Theorem 1 of Lugosi and Nobel, 1996 our density estimate $f_{n,\hat{s}}$ is strongly consistent in L_1 .

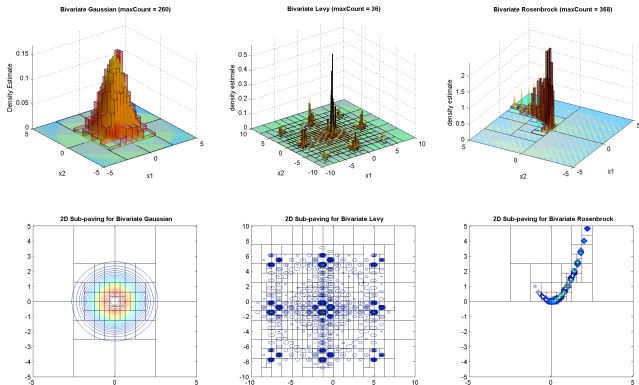
These conditions mean:

- (a) sub-linear growth of the number of leaf boxes
- (b) sub-exponential growth of a combinatorial complexity measure of the growth of the partition
- (c) shrinking leaf boxes in the partition

- └ Adaptive Histograms
- └ S.E.B. Priority Queue

Some Examples

Figure: Histogram density estimates their corresponding pavings for the bivariate Gaussian, Levy and Rosenbrock densities.



Choice of k_n

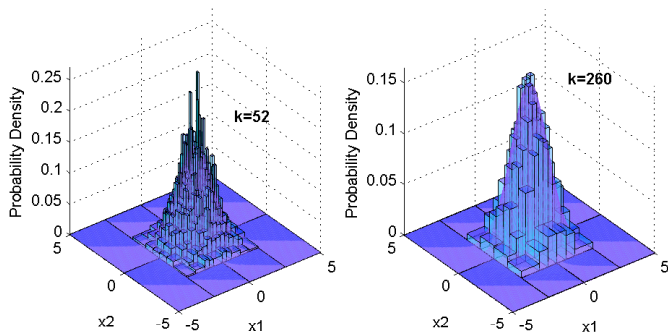
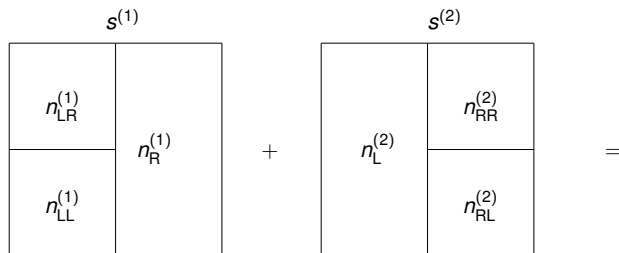


Figure: Two histogram density estimates for the standard bivariate gaussian density with different choices of k_n . The histogram is under-smoothed when k_n is relatively smaller than n and over-smoothed when k_n is relatively larger.

Adding and Averaging SRPs

Do a minimal union (or overlay) operation of $s^{(1)}$ and $s^{(2)}$ and adjust counts:



Adding and Averaging SRPs

Do a minimal union (or overlay) operation of $s^{(1)}$ and $s^{(2)}$ and adjust counts:

$$\begin{array}{|c|c|} \hline & s^{(1)} \\ \hline n_{LR}^{(1)} & \\ \hline n_{LL}^{(1)} & n_R^{(1)} \\ \hline \end{array} + \begin{array}{|c|c|} \hline & s^{(2)} \\ \hline & n_{RR}^{(2)} \\ \hline n_L^{(2)} & n_{RL}^{(2)} \\ \hline \end{array} = \begin{array}{|c|c|} \hline & s^{(1)} + s^{(2)} \\ \hline n_{LR}^{(1)} + \frac{n_L^{(2)}}{2} & \frac{n_R^{(1)}}{2} + n_{RR}^{(2)} \\ \hline n_{LL}^{(1)} + \frac{n_L^{(2)}}{2} & \frac{n_R^{(1)}}{2} + n_{RL}^{(2)} \\ \hline \end{array}$$

Adding and Averaging SRPs

Adding m SRP histogram density estimates

$$\begin{aligned}\sum_{i=1}^m f_{n,s^{(i)}} &= f_{n,s^{(1)}} + f_{n,s^{(2)}} + f_{n,s^{(3)}} + \dots + f_{n,s^{(m)}} \\ &= \left(\left(\left(f_{n,s^{(1)}} + f_{n,s^{(2)}} \right) + f_{n,s^{(3)}} \right) + \dots + f_{n,s^{(m)}} \right) .\end{aligned}$$

Adding and Averaging SRPs

Adding m SRP histogram density estimates

$$\begin{aligned}\sum_{i=1}^m f_{n,s^{(i)}} &= f_{n,s^{(1)}} + f_{n,s^{(2)}} + f_{n,s^{(3)}} + \dots + f_{n,s^{(m)}} \\ &= \left(\left(\left(f_{n,s^{(1)}} + f_{n,s^{(2)}} \right) + f_{n,s^{(3)}} \right) + \dots + f_{n,s^{(m)}} \right) .\end{aligned}$$

Averaging m SRP histogram density estimates recursively yields the sample mean SRP histogram

$$\bar{f}_{n,m} = \frac{1}{m} \sum_{i=1}^m f_{n,s^{(i)}} .$$

An Example

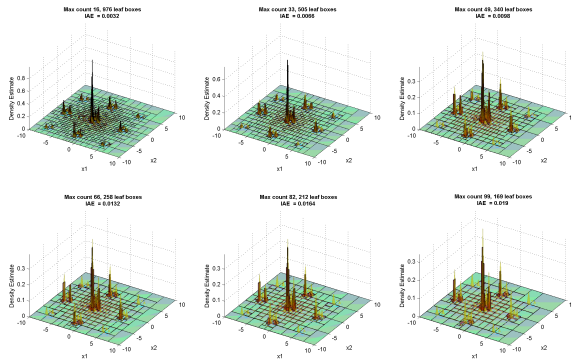


Figure: Histogram density estimates of the bivariate Levy using different values of k_n .

An Example

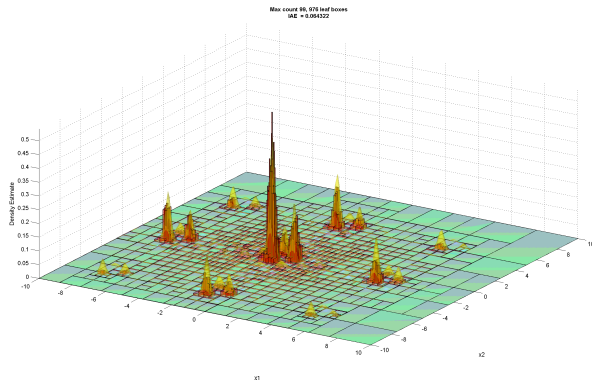


Figure: The averaged histogram density estimate.

Posterior Distribution over Histograms in $\mathbb{S}_{0:\infty}$

- ▶ Let \hat{f}_s be a histogram with partition $\ell(s)$ given by the leaves of RP s with k splits and $k + 1$ leaves in \mathbb{S}_k

Posterior Distribution over Histograms in $\mathbb{S}_{0:\infty}$

- ▶ Let \hat{f}_s be a histogram with partition $\ell(s)$ given by the leaves of RP s with k splits and $k + 1$ leaves in \mathbb{S}_k
- ▶ Then for this partition, the most likely histogram estimate is

$$\hat{f}_s(x; \text{data}) = \frac{1}{n} \hat{f}_s(x; X_{1:n}) = \sum_{i=1}^n \frac{\mathbf{1}(x_i \in \mathbf{x}(x))}{\text{vol}(\mathbf{x}(x))}$$

Posterior Distribution over Histograms in $\mathbb{S}_{0:\infty}$

- ▶ Let \hat{f}_s be a histogram with partition $\ell(s)$ given by the leaves of RP s with k splits and $k + 1$ leaves in \mathbb{S}_k
- ▶ Then for this partition, the most likely histogram estimate is

$$\hat{f}_s(x; \text{data}) = \frac{1}{n} \hat{f}_s(x; X_{1:n}) = \sum_{i=1}^n \frac{\mathbf{1}(x_i \in \mathbf{x}(x))}{\text{vol}(\mathbf{x}(x))}$$

- ▶ Let the prior probability be $P(s) \propto \frac{1}{C_k^2}$, $s \in \mathbb{S}_{0:\infty}$

Posterior Distribution over Histograms in $\mathbb{S}_{0:\infty}$

- ▶ Let \hat{f}_s be a histogram with partition $\ell(s)$ given by the leaves of RP s with k splits and $k + 1$ leaves in \mathbb{S}_k
- ▶ Then for this partition, the most likely histogram estimate is

$$\hat{f}_s(x; \text{data}) = \frac{1}{n} \hat{f}_s(x; X_{1:n}) = \sum_{i=1}^n \frac{\mathbf{1}(x_i \in \mathbf{x}(x))}{\text{vol}(\mathbf{x}(x))}$$

- ▶ Let the prior probability be $P(s) \propto \frac{1}{C_k^2}$, $s \in \mathbb{S}_{0:\infty}$
- ▶ Then the posterior density of histogram \hat{f}_s with k splits is

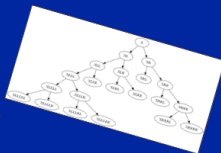
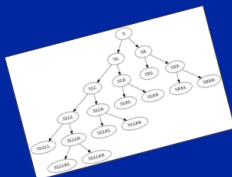
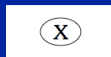
$$P(\hat{f}_s | X_{1:n}) \propto P(X_{1:n} | s) P(s) = \prod_{\mathbf{x} \in \ell(s)} \left(\frac{n_{\mathbf{x}}}{n \text{vol}(\mathbf{x})} \right)^{n_{\mathbf{x}}} \frac{1}{C_k^2}$$

Metropolis-Hastings Algorithm

- ▶ Use a proposal density $q(s^{prime}|s^{(i)})$ which depends on current state $s^{(i)}$, to generate a new proposed state s'
- ▶ We propose uniformly at random to split a leaf or merge a cherry of current SRP state $s^{(i)}$
- ▶ **Repeat**
 - ▶ **Draw** $u \sim U(0, 1)$
 - ▶ **If** $u < \frac{P(\hat{f}_{s'}|X_{1:n})}{P(\hat{f}_{s^{(i)}}|X_{1:n})} \frac{q(s^{(i)}|s')}{q(s^{prime}|s^{(i)})}$ **then** $s^{(i+1)} \leftarrow s'$
 - ▶ **else** $s^{(i+1)} \leftarrow s^{(i)}$
- ▶ With a “long enough” burn-in time, this Markov chain will be at the desired stationary distribution $P(\hat{f}_s|X_{1:n})$ over $\mathbb{S}_{0:\infty}$

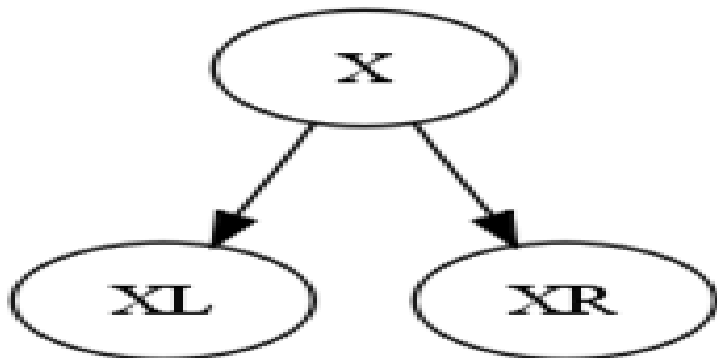
Metropolis-Hastings Algorithm

- Start from some initial state m^0
- Burn-in: run until initial state is 'forgotten'
- States after burn-in are sample histograms

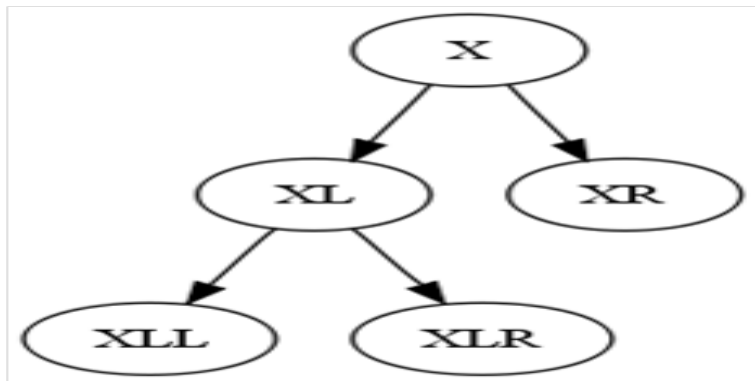


etc...

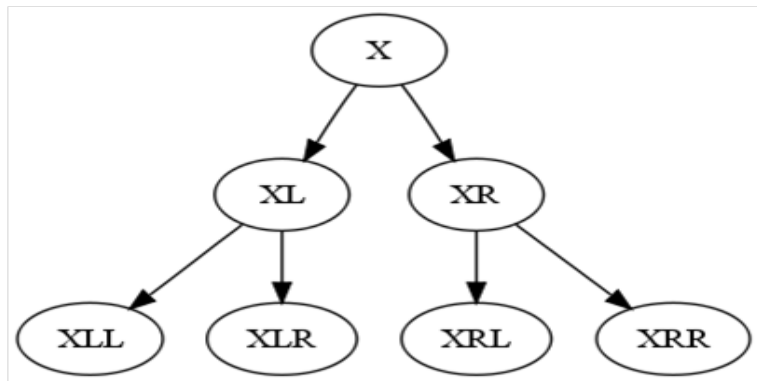
Monte Carlo Markov Chain over Histograms in $\mathcal{S}_{0:\infty}$



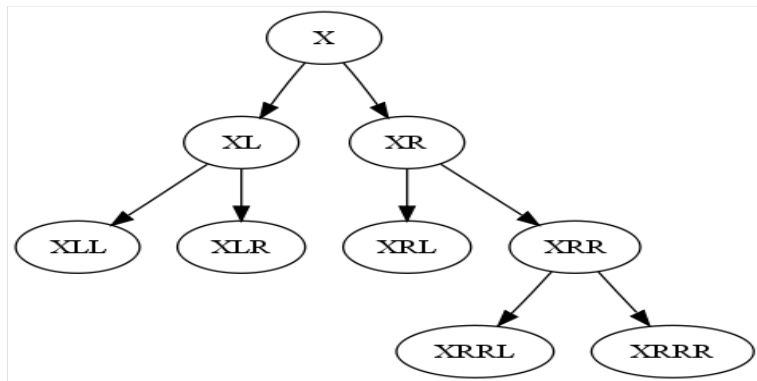
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



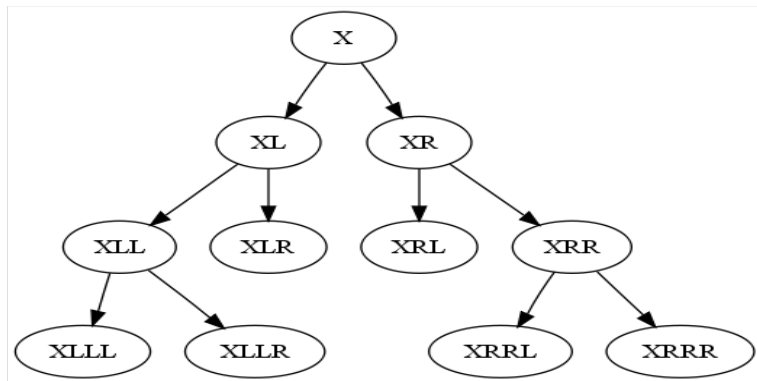
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

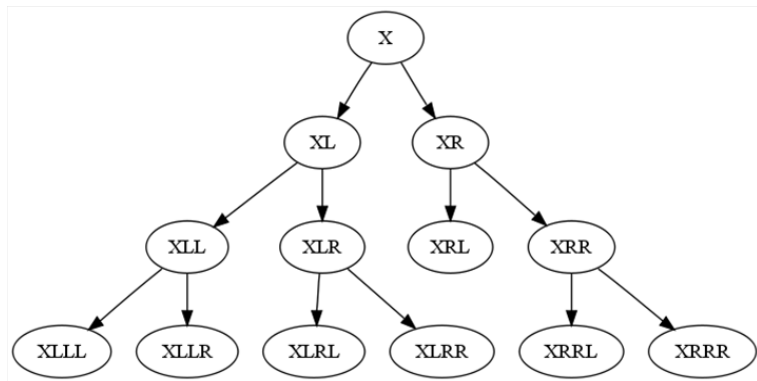


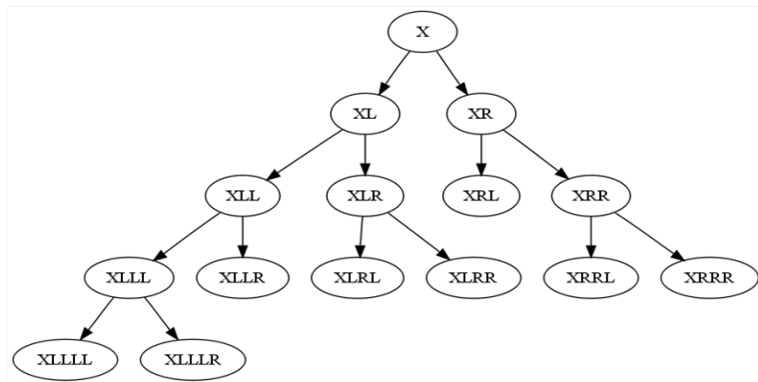
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



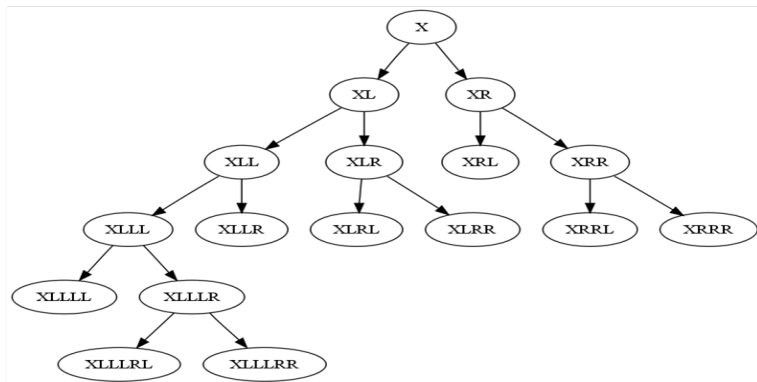
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

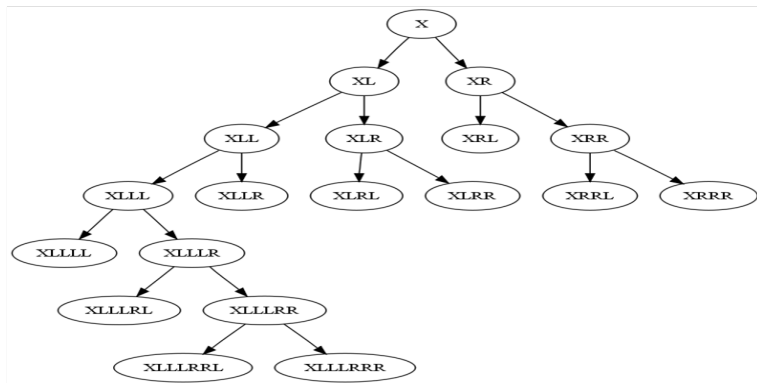


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$ 

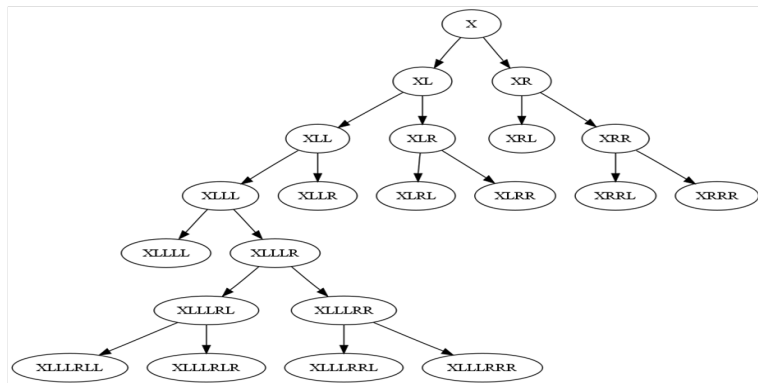
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$ 

Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

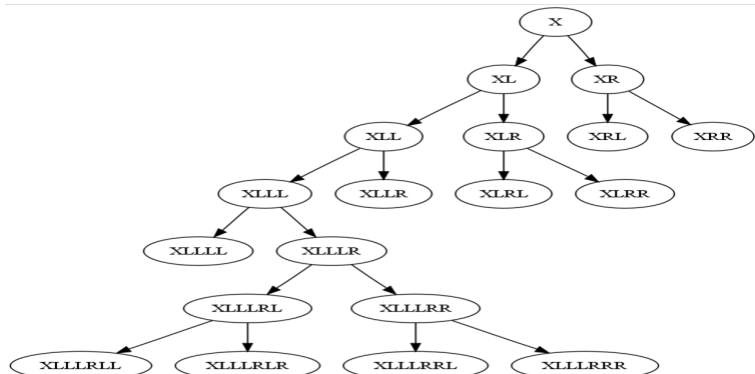


Monte Carlo Markov Chain over Histograms in $\mathcal{S}_{0:\infty}$ 

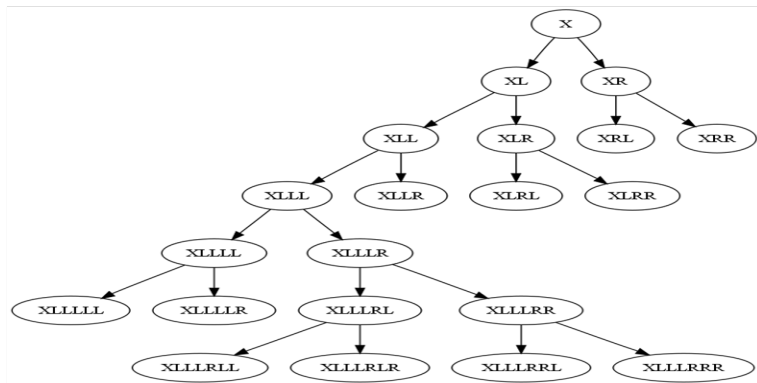
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



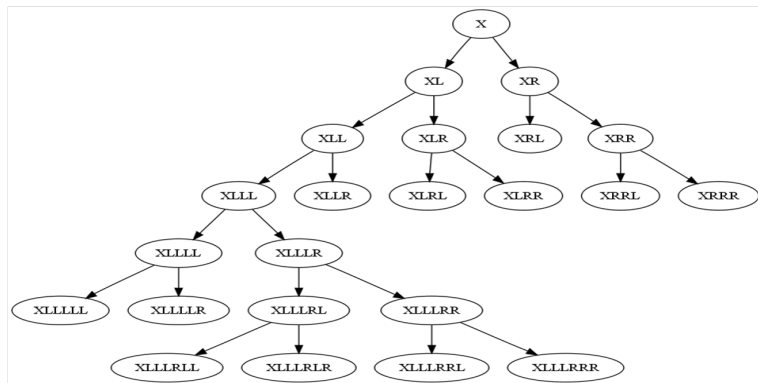
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



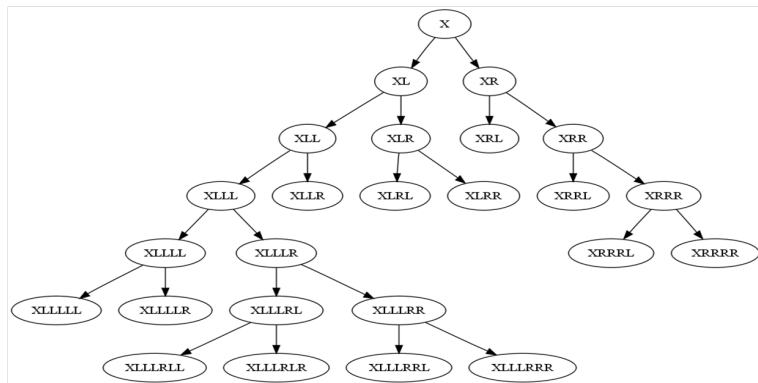
Monte Carlo Markov Chain over Histograms in $\mathcal{S}_{0:\infty}$



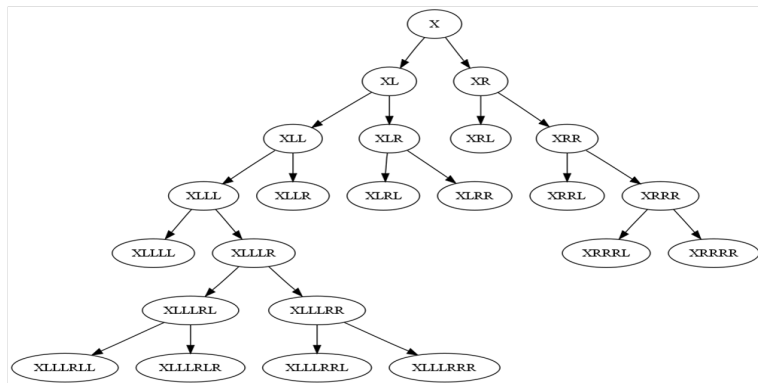
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



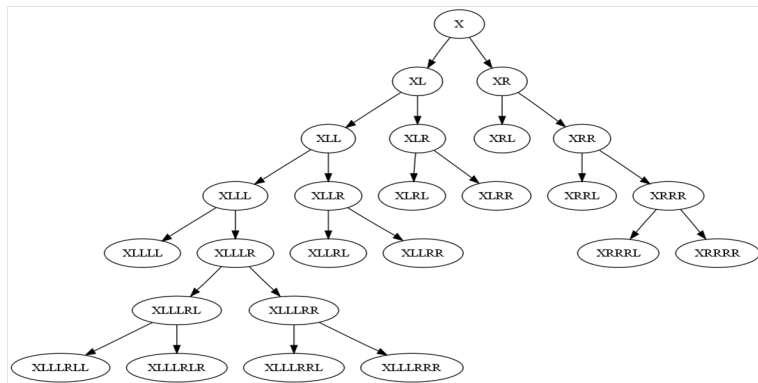
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



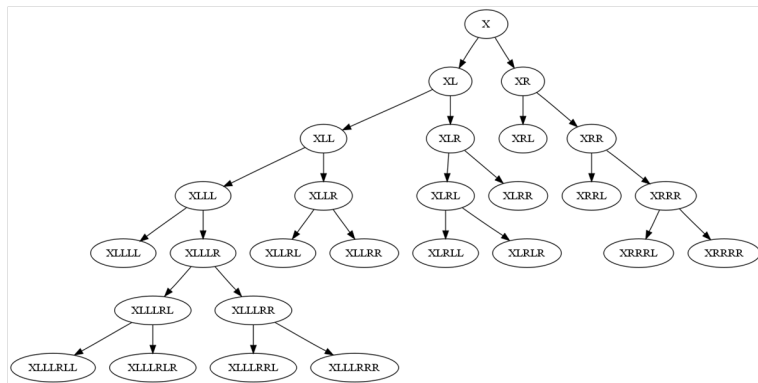
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

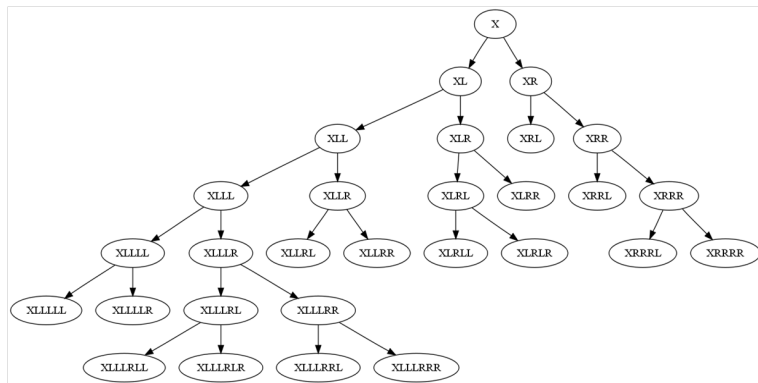


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

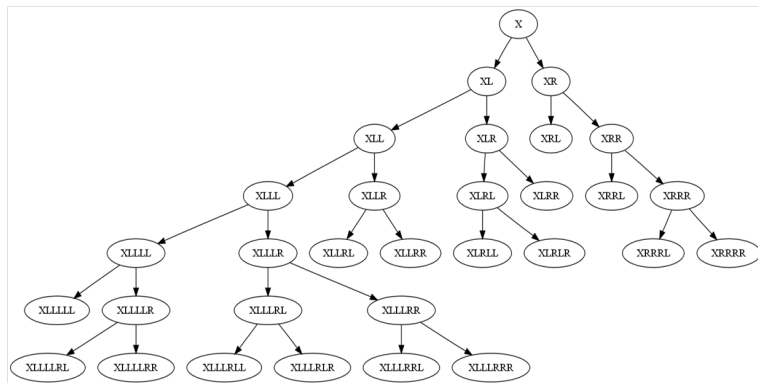


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

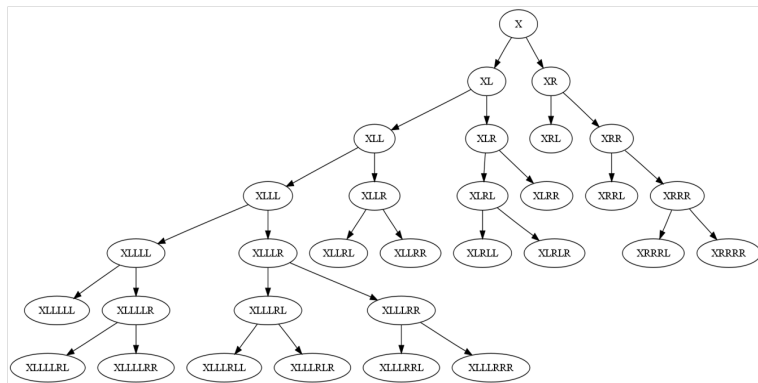


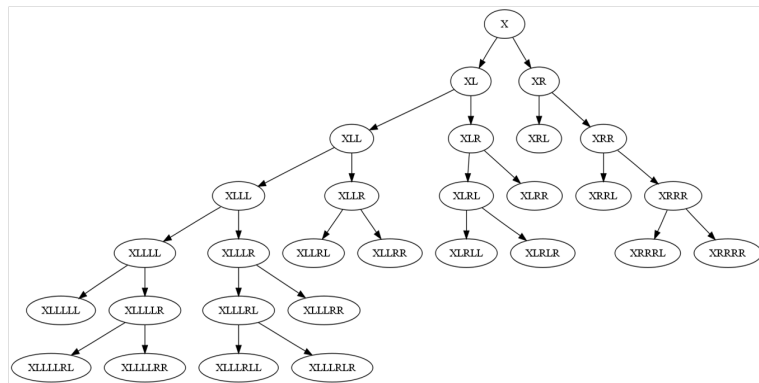
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$ 

Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

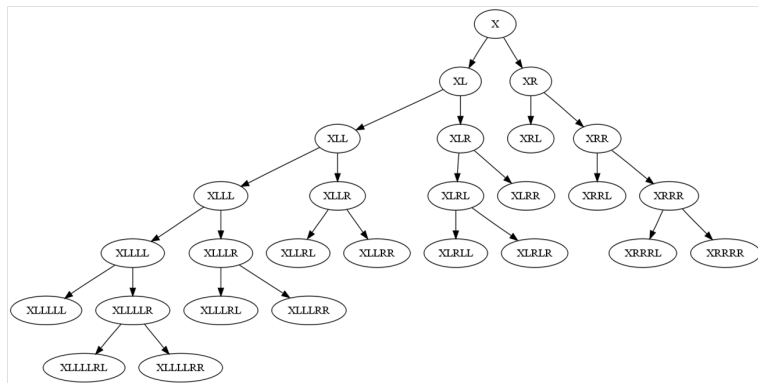


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

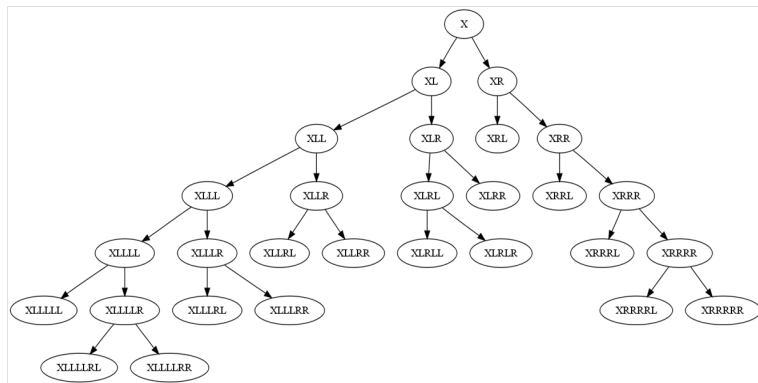


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$ 

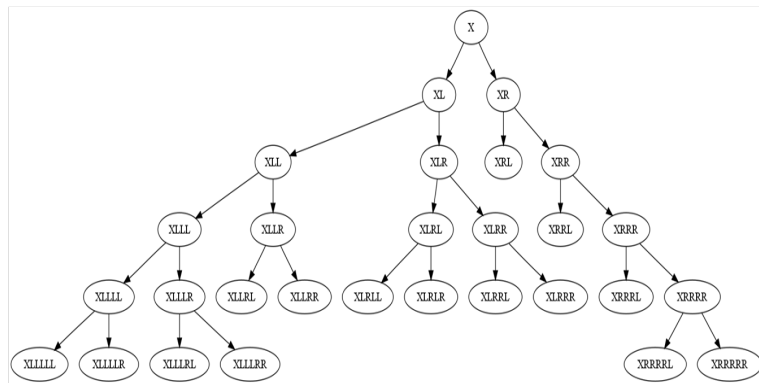
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



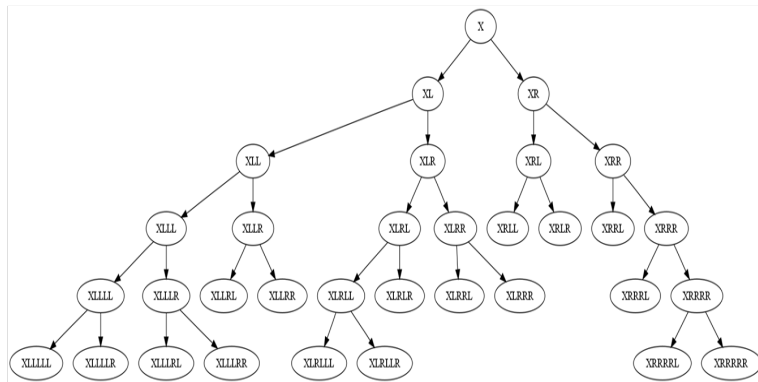
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



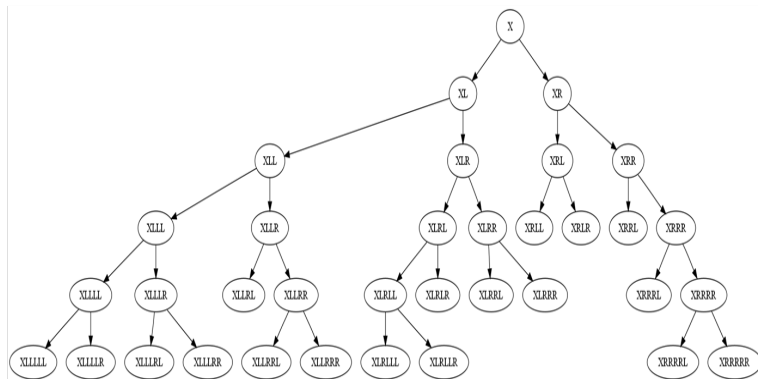
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

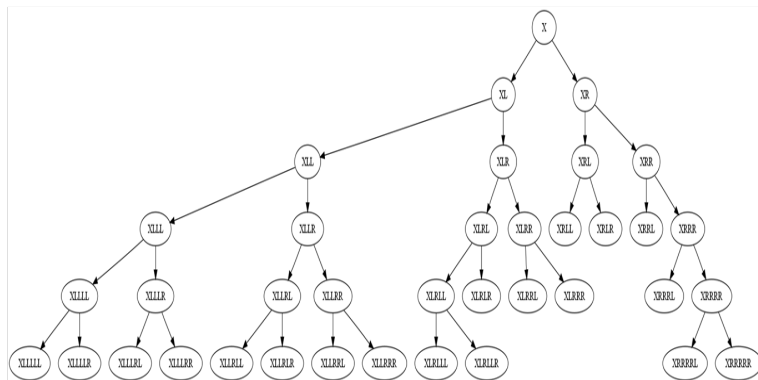


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

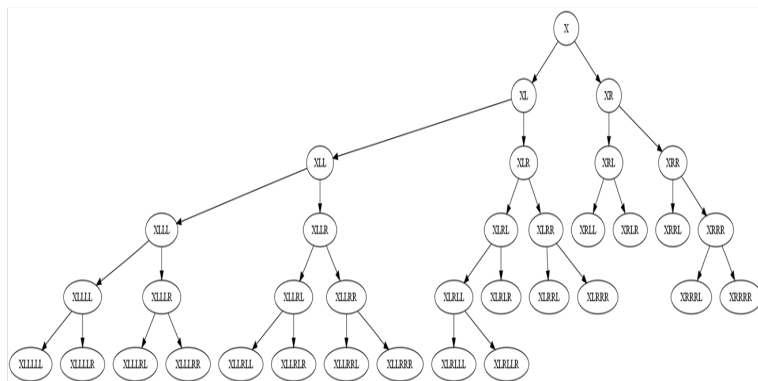


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

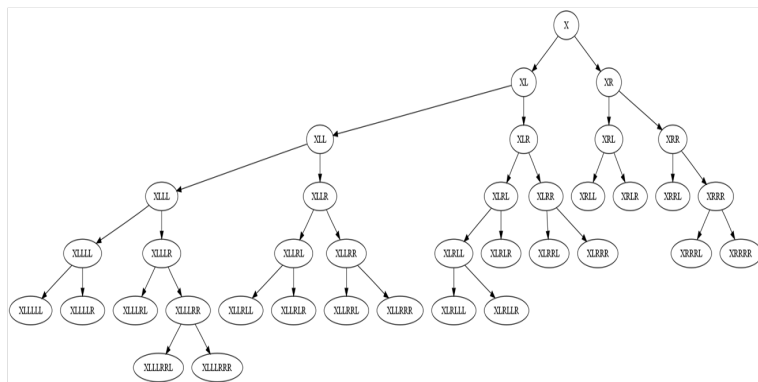


Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$ 

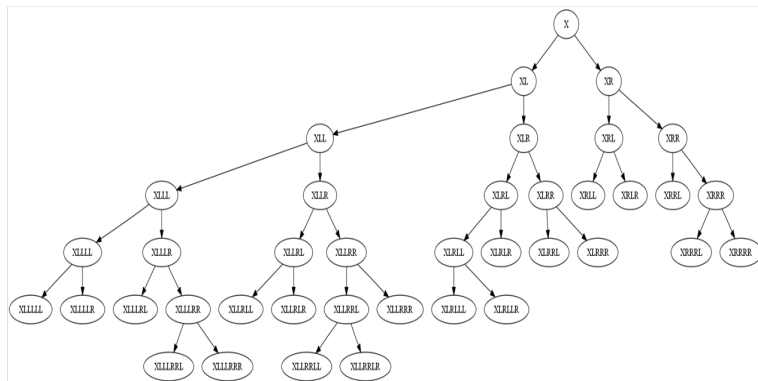
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



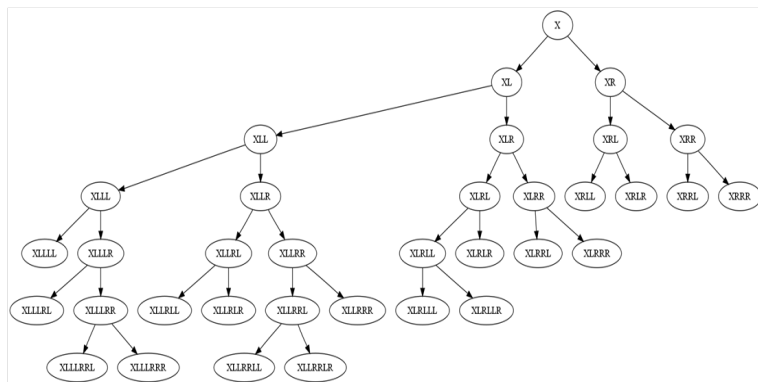
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



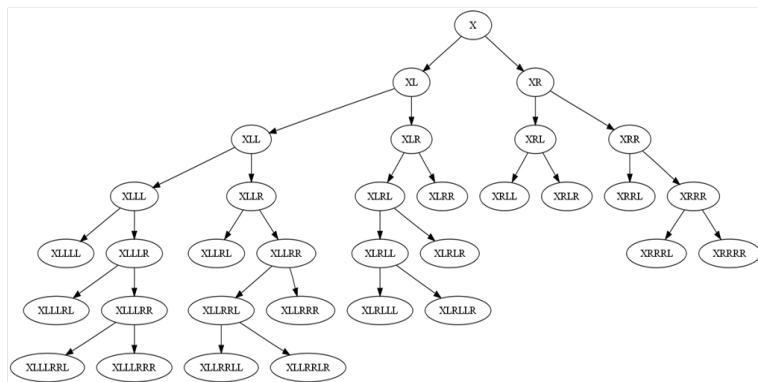
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



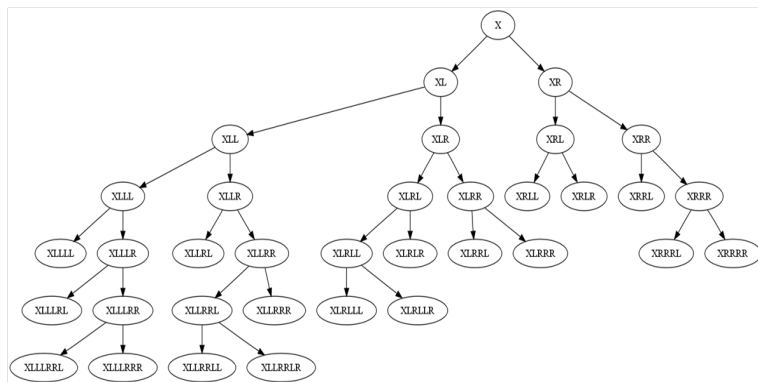
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



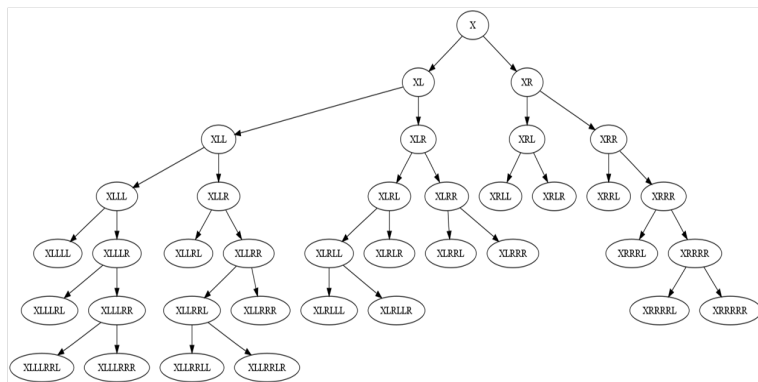
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



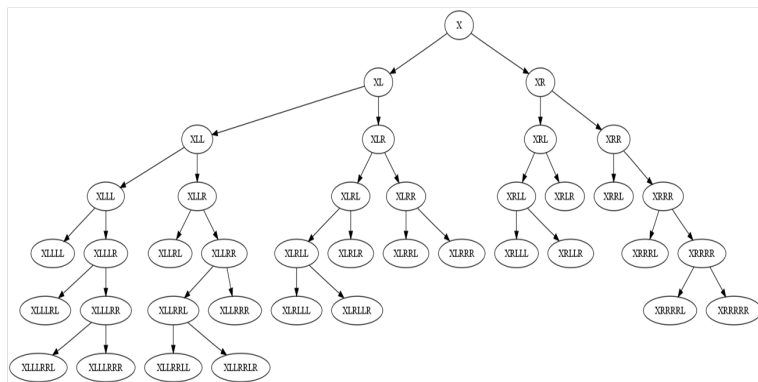
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



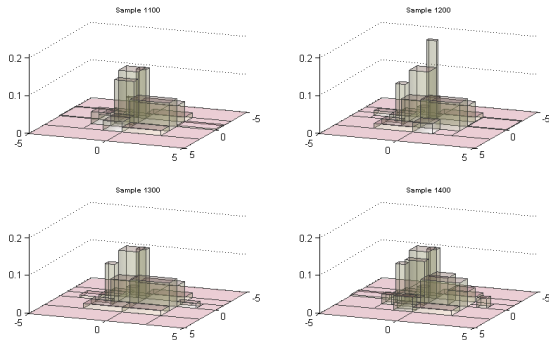
Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$



Monte Carlo Markov Chain over Histograms in $\mathbb{S}_{0:\infty}$

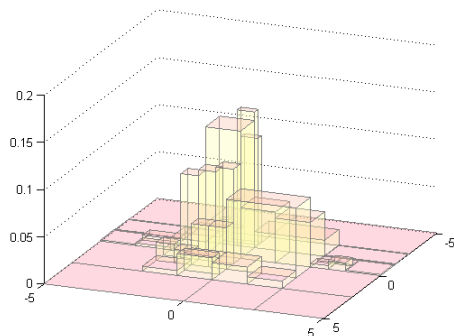


Histogram Estimates - Standard Bivariate Gaussian



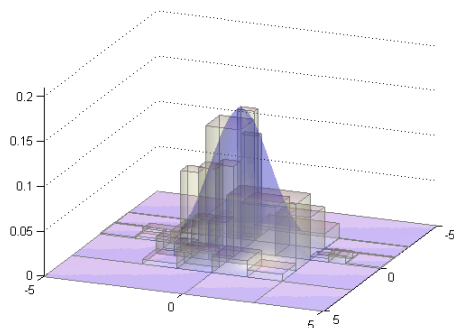
Four sample histograms

Histogram Estimates - Standard Bivariate Gaussian



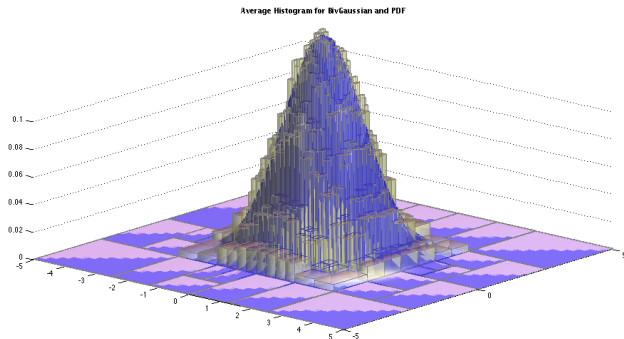
Average of the four sampled histograms

Histogram Estimates - Standard Bivariate Gaussian



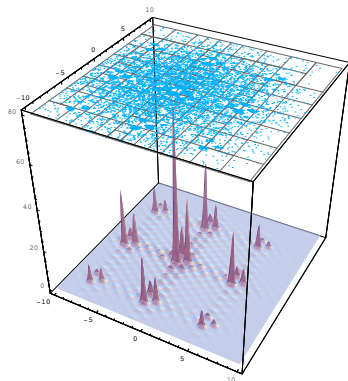
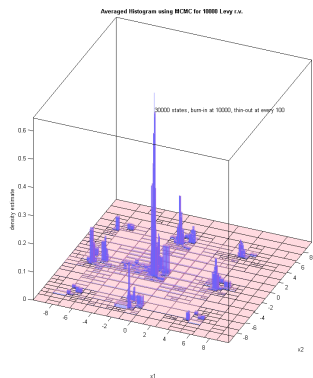
Average of the four sampled histograms with Gaussian PDF

Histogram Estimates - Standard Bivariate Gaussian



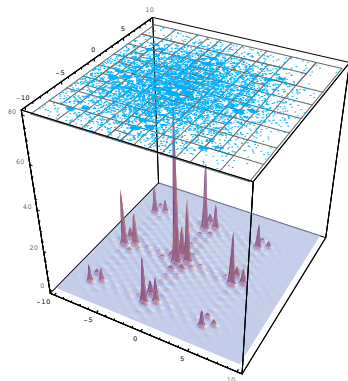
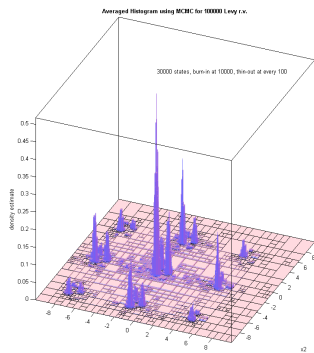
A much better estimate

Histogram Estimates - Bivariate Levy Density



Data points = 10000, Number of states = 30000, Burn-in = 10000,
Thin-out = 100, Averaged over 201 states, Time taken = 14.16s

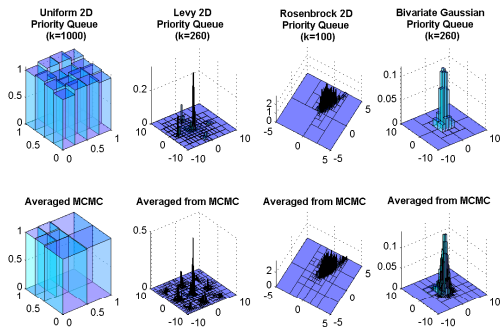
Histogram Estimates - Bivariate Levy Density



Data points = 100000, Number of states = 30000, Burn-in = 10000,
Thin-out = 100, Averaged over 201 states, Time taken = 50.59s

Simulations for MCMC and `SplitMostCounts` PQ

Density	Time (s)	MIAE (root box)	MIAE (PQ)	Density	Time (s)	MIAE (root box)	MIAE (PQ)
U(0,1) 1D	5.2940	0.0112	0.0115	U(0,1) 2D	4.96	0.0125	0.0123
N(0,1) 1D	0.4857	0.0663	0.0651	Rosen. 10D	2.2900	NA	NA
Gaussian 2D	0.6206	0.2444	0.2702	U(0,1) 10D	14.775	0.0127	0.0119
Levy 2D	4.3200	0.4187	0.3272	U(0,1) 100D	107.3963	0.0108	0.0116
Rosen. 2D	9.5672	0.3273	0.4307	U(0,1) 1000D	970.4471	0.0117	0.0108



Simulations for MCMC and `SplitMostCounts` PQ

MIAE (std. err.) for n samples from uniform density in various dimensions (CPU Times $< O(1\text{minute})$).

n	1D	2D	10D	100D	1000D
10^2	0.1112 (0.0707)	0.1425 (0.0882)	0.1170 (0.0723)	0.0958 (0.0605)	0.1111 (0.0524)
10^3	0.0366 (0.0192)	0.0363 (0.0219)	0.0442 (0.0275)	0.0413 (0.0196)	0.0305 (0.0195)
10^4	0.0164 (0.0095)	0.0124 (0.0073)	0.0115 (0.0070)	0.0111 (0.0083)	0.0089 (0.0065)
10^5	0.0041 (0.0020)	0.0040 (0.0026)	0.0041 (0.0028)	0.0050 (0.0030)	0.0043 (0.0025)
10^6	0.0011 (0.0005)	0.0016 (0.0007)	0.0010 (0.0006)	0.0012 (0.0001)	0.0010 (0.0004)
10^7	0.0004 (0.0003)	0.0003 (0.0002)	0.0003 (0.0002)	0.0002 (0.0001)	-
10^8	0.0001 (0.0009)	0.0002 (0.0002)	0.0001 (0.0001)	-	-

Simulations for MCMC and `SplitMostCounts` PQ

MIAE (std. err.) for n samples from approximated 1D-, 2D- and 10D-Gaussian densities, and 2D- and 10D-Rosenbrock densities (L_1 -minimal Simple function approximation in \mathbb{S}_Λ).

Λ	n	Standard Gaussian densities			Rosenbrock densities	
		1D	2D	10D	2D	10D
10^2	10^2	0.2665 (0.0415)	0.4856 (0.0491)	0.1192 (0.0662)	0.5089 (0.0924)	0.0323 (0.0511)
	10^3	0.1390 (0.0192)	0.2558 (0.0127)	0.0543 (0.0172)	0.1712 (0.0224)	0.0095 (0.0191)
	10^4	0.0620 (0.0047)	0.0992 (0.0067)	0.0382 (0.0036)	0.0498 (0.0081)	0.0025 (0.0050)
	10^5	0.0262 (0.0016)	0.0279 (0.0019)	0.0259 (0.0017)	0.0143 (0.0025)	0.0009 (0.0015)
	10^6	0.0099 (0.0008)	0.0086 (0.0006)	0.0073 (0.0009)	0.0045 (0.0005)	0.0004 (0.0005)
	10^7	0.0026 (0.0002)	0.0027 (0.0003)	0.0025 (0.0004)	0.0017 (0.0010)	0.0001 (0.0003)
	10^3	10^2	0.2946 (0.0678)	0.6046 (0.1299)	0.1702 (0.0907)	1.0027 (0.0437)
10^3		0.1418 (0.0226)	0.2973 (0.0174)	0.0739 (0.0183)	0.4747 (0.0191)	0.0039 (0.0075)
10^4		0.0648 (0.0052)	0.1586 (0.0067)	0.0555 (0.0045)	0.2139 (0.0054)	0.0013 (0.0028)
10^5		0.0292 (0.0014)	0.0768 (0.0016)	0.0295 (0.0020)	0.0789 (0.0023)	0.0004 (0.0006)
10^6		0.0136 (0.0006)	0.0297 (0.0006)	0.0108 (0.0005)	0.0267 (0.0058)	0.0001 (0.0002)
10^7		0.0061 (0.0002)	0.0091 (0.0003)	0.0045 (0.0003)	0.0082 (0.0011)	0.0001 (0.0002)
10^4		10^2	0.2864 (0.0487)	0.5508 (0.0590)	0.5210 (0.0799)	1.1391 (0.0545)
	10^3	0.1380 (0.0152)	0.3301 (0.0120)	0.2719 (0.0251)	0.6018 (0.0139)	0.0791 (0.0223)
	10^4	0.0664 (0.0062)	0.1736 (0.0038)	0.1157 (0.0047)	0.3163 (0.0047)	0.0391 (0.0041)
	10^5	0.0293 (0.0017)	0.0957 (0.0014)	0.0870 (0.0014)	0.1691 (0.0053)	0.0209 (0.0021)
	10^6	0.0138 (0.0005)	0.0495 (0.0005)	0.0788 (0.0009)	0.0882 (0.0048)	0.0123 (0.0012)
	10^7	0.0063 (0.0001)	0.0244 (0.0008)	0.0563 (0.0018)	0.0479 (0.0057)	0.0096 (0.0017)

Examples of Application

- ▶ Web Log Data (Link to SAGE server)
- ▶ Air Traffic Data (Link to SAGE server)
- ▶ Earthquake Data (Link to SAGE server)

Conclusions

- ▶ Statistical Regular Paving (SRP) is a sufficient statistical data-structure for density estimation and many decisions in massive IID experiments.

Conclusions

- ▶ Statistical Regular Paving (SRP) is a sufficient statistical data-structure for density estimation and many decisions in massive IID experiments.
- ▶ We can quickly grow or prune the SRP tree data-adaptively

Conclusions

- ▶ Statistical Regular Paving (SRP) is a sufficient statistical data-structure for density estimation and many decisions in massive IID experiments.
- ▶ We can quickly grow or prune the SRP tree data-adaptively
- ▶ Arithmetic can be efficiently extended to SRPs - averaging histograms and other *arithmetics* on $\mathbb{S}_{0:\infty}$.

Conclusions

- ▶ Statistical Regular Paving (SRP) is a sufficient statistical data-structure for density estimation and many decisions in massive IID experiments.
- ▶ We can quickly grow or prune the SRP tree data-adaptively
- ▶ Arithmetic can be efficiently extended to SRPs - averaging histograms and other *arithmetics* on $\mathbb{S}_{0:\infty}$.
- ▶ Smoother posterior mean from MCMC samples on the space of adaptive multi-variate histograms with partitions in $\mathbb{S}_{0:\infty}$. NFL: MCMC convergence issues exist!

Conclusions

- ▶ Statistical Regular Paving (SRP) is a sufficient statistical data-structure for density estimation and many decisions in massive IID experiments.
- ▶ We can quickly grow or prune the SRP tree data-adaptively
- ▶ Arithmetic can be efficiently extended to SRPs - averaging histograms and other *arithmetics* on $\mathbb{S}_{0:\infty}$.
- ▶ Smoother posterior mean from MCMC samples on the space of adaptive multi-variate histograms with partitions in $\mathbb{S}_{0:\infty}$. NFL: MCMC convergence issues exist!
- ▶ Higher (1000) dimensional densities can be estimated fast and rough (but L_1 -consistent) with the approach especially with `SplitMostCounts` PQ and further decisions can be done with appropriate *mapped RP arithmetic* over $\mathbb{S}_{0:\infty}$.

References

Jaulin, L., Kieffer, M., Didrit, O. & Walter, E. (2001). *Applied interval analysis*. London: Springer-Verlag.

Lugosi, G. and Nobel, A. (1996). Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics* **24** 687–706.

Sainudiin, R. and York, T. L. (2005). *An Auto-validating Rejection Sampler*. BSCB Dept. Technical Report BU-1661-M, Cornell University, Ithaca, New York.

Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16** 264–280.

Gratitude for Operating Fiscal Environment

2012, 2-day Visitor, [Stochastic Networks Workshop](#) (organizer Ilze Ziedins), [Auckland, New Zealand](#)

2011, 2-week Visitor, Advanced Learning and Evolutionary Algorithms Group (ALEA), [INRIA, Bordeaux, France](#)

2010, 6-month Sabbatical Award, [College of Engineering, U. of Canterbury, Christchurch, New Zealand](#)

2010, 6-month Visiting Scientist, Theoretical Stats. & Maths. Unit, [Indian Statistical Institute, Bengaluru, India](#)

2009, 2-week Visitor, Computer-Aided Proofs in Analysis Group (CAPA), [Angström Laboratory, Uppsala, Sweden](#)

2009, [NZD 8000](#), [External Consulting Revenues](#) from [New Zealand Ministry of Tourism](#)

Remark: [8 unsuccessful applications to NZ funds \(3 Marsden fast-starts\)](#) :(P – due to lack of “track record”...

This work in Computational Mathematics and Statistics is based on: [1,800,101 ASCII characters](#) ~ 21 days of [non-stop coding at 1 character per second](#) in GPL licensed *mrs: a C++ class library for statistical set processing* as of Thu Apr 12 08:34:44 NZST 2012 by Bycroft, Harlow, Sainudiin, Teng and York)