Accelerating Bayesian Inference on Structured Graphs Using Parallel Gibbs Sampling

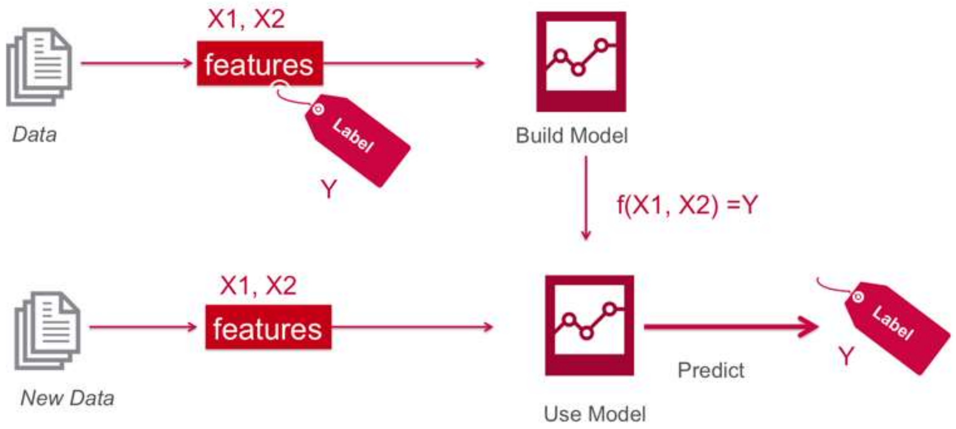
Glenn G. Ko

gko@seas.harvard.edu

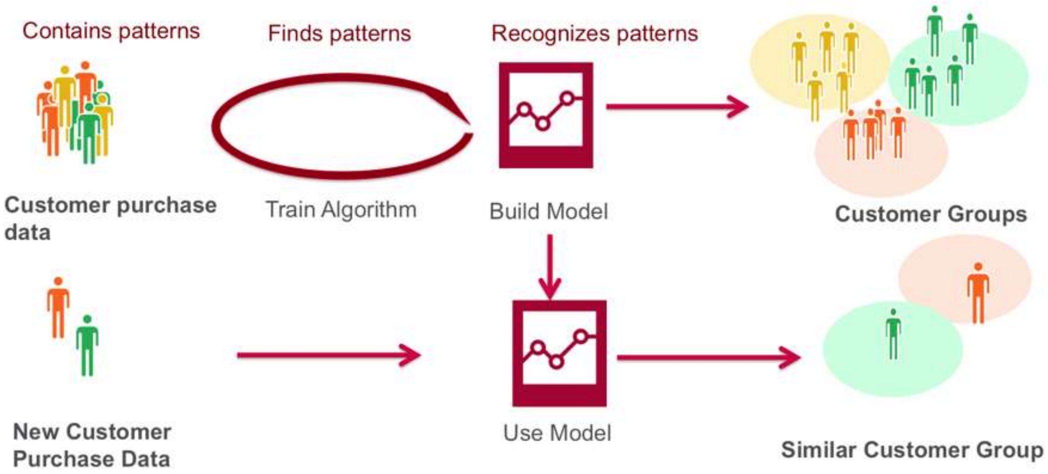
Harvard University

September 10, 2019

Supervised vs. Unsupervised Machine Learning



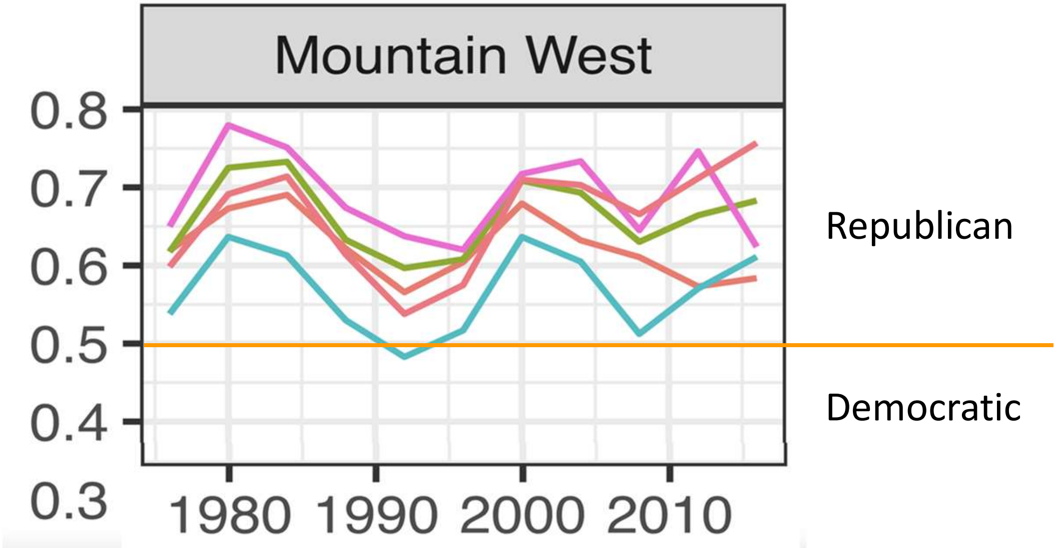
Supervised

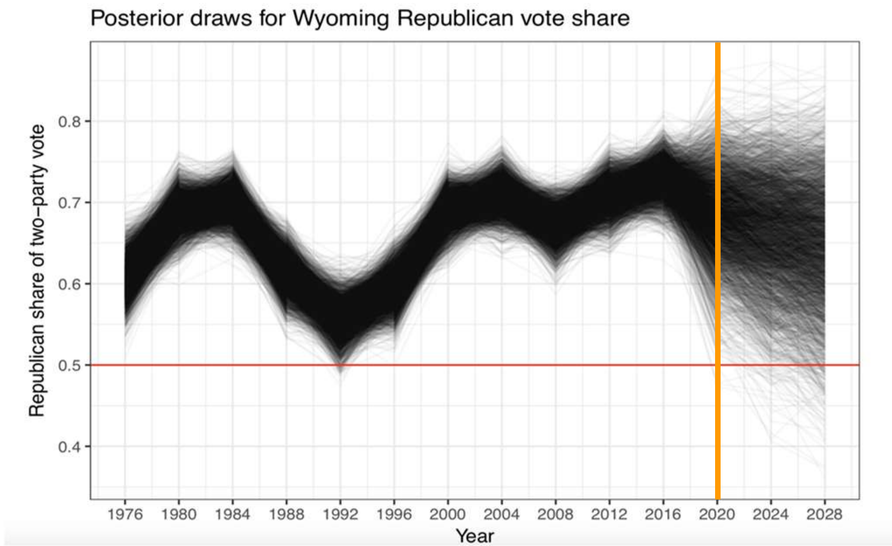


Unsupervised

[https://mapr.com/blog/demystifying-ai-ml-dl]

Why Bayesian Machine Learning





- Predict a probability distribution not a point estimate
- Quantify uncertainty

[https://github.com/stan-dev/stancon_talks/blob/master/2017/Contributed-Talks/08_trangucci/hierarchical_GPs_in_stan.pdf]

Deep Learning vs. Bayesian ML

	Deep Learning	Bayesian Inference
Data Type / Size	Needs large labeled data	Scarce or no labeled data
Interpretability	Black-box	Interpretable models
Prior Knowledge	No	Prior + new observations
Scalability	Parallelizable	Limited parallelism
Generalizability	Generalizable	Hand-crafted models
Unsupervised	Good at supervised	Good at unsupervised
...	...	...

Combining the two: Variational autoencoder, Generative Adversarial Networks, Bayesian neural networks, and etc.

Bayesian Models and Inference

- Unsupervised learning
- Scarce or no labeled data for training
- Ability to represent and manipulate uncertainty
- Generative models

Bayes' Rule:

$$P(X|Y) = \frac{\overbrace{P(Y|X)}^{\text{Likelihood}}\overbrace{P(X)}^{\text{Prior}}}{\underbrace{P(Y)}_{\text{Evidence}}}$$

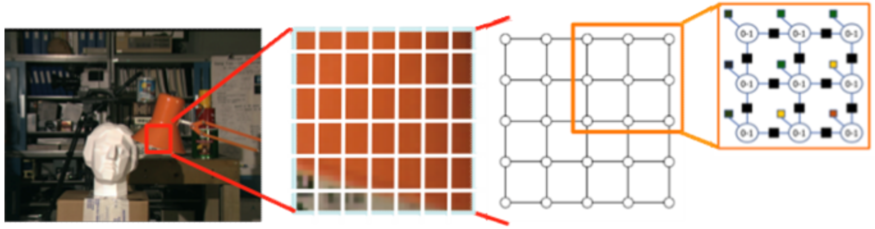
X: Hidden Parameters
Y: Observed Data

Markov Random Fields and Inference

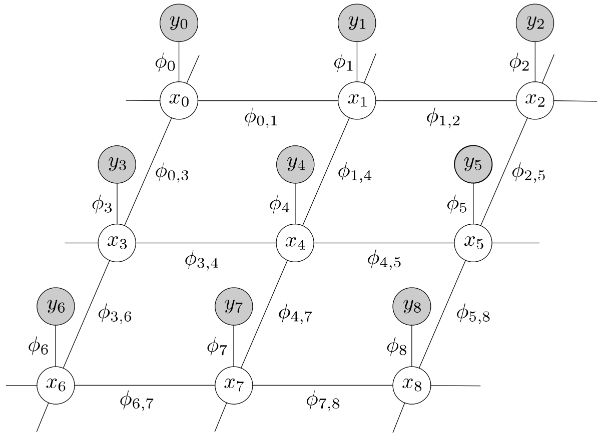
Pixel-labeling problems on MRF:

- Stereo matching
- Image restoration
- Image segmentation
- Sound source separation

Stereo matching



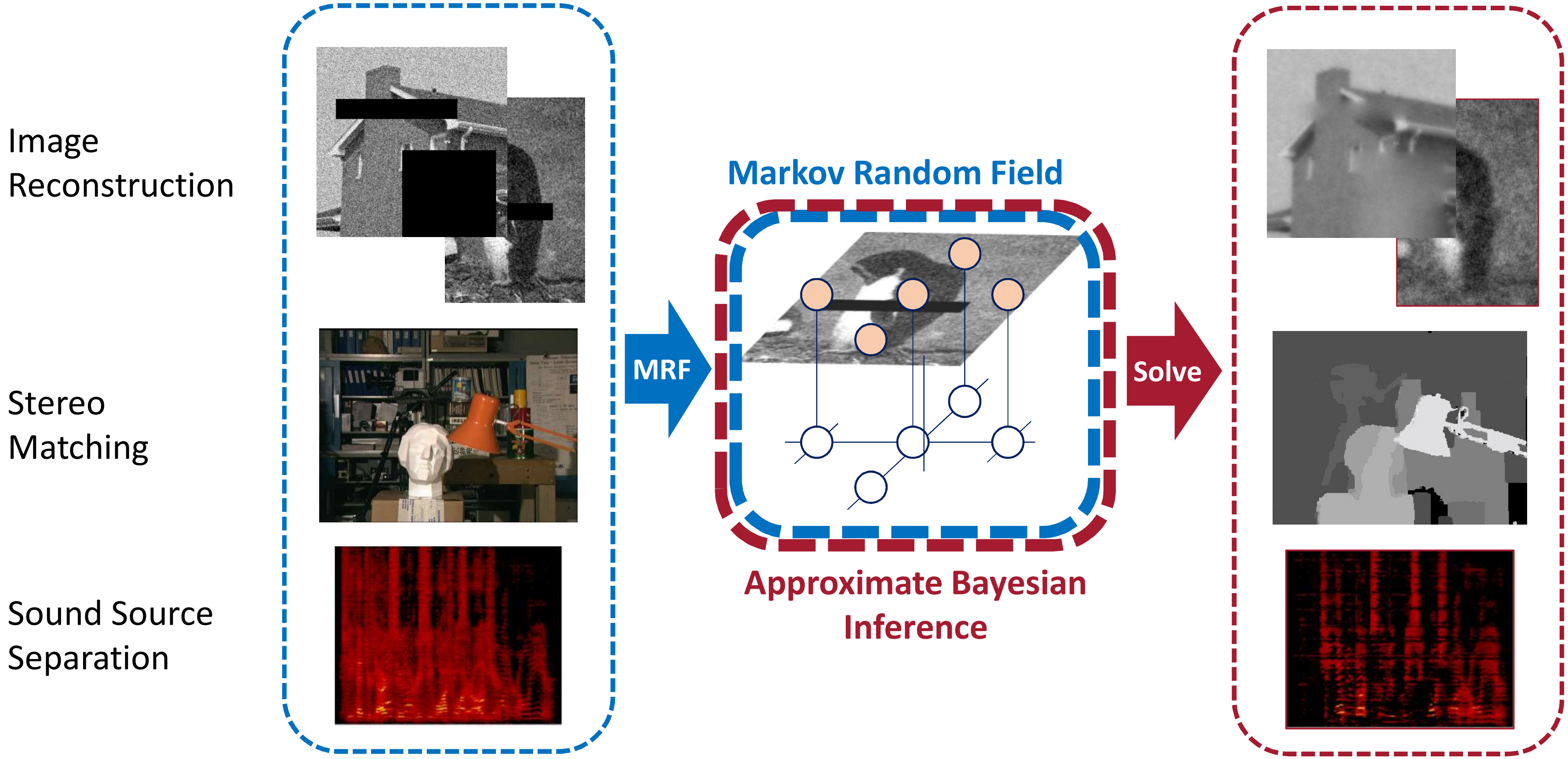
Pixels = **nodes**
Edges to neighbors
Inference for best set of new **labels**



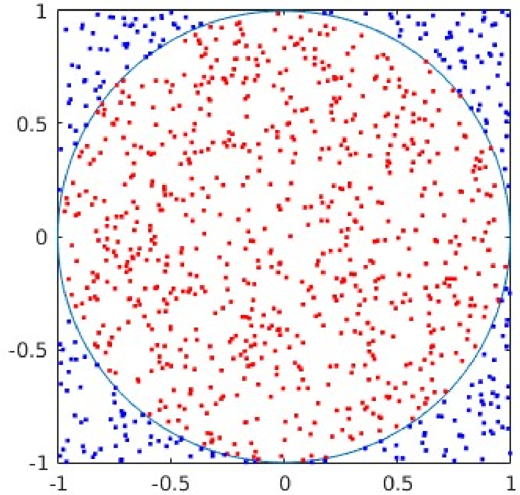
y: input pixels
x: labels for each pixel

$$P(x, y) = \frac{1}{Z} \underbrace{\prod_{i \in V} \phi_i(x_i, y_i)}_{\text{Likelihood (Data cost)}} \underbrace{\prod_{i,j \in E} \phi_{i,j}(x_i, x_j)}_{\text{Prior (Smoothness cost)}}$$

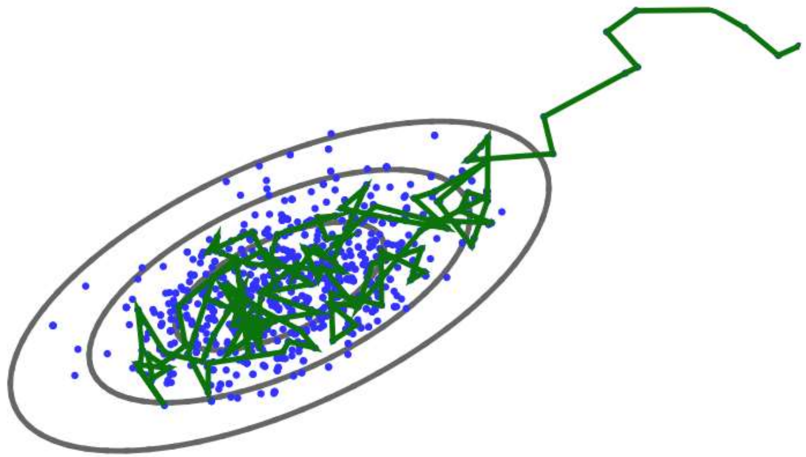
Unsupervised Learning Tasks on MRF



Markov Chain Monte Carlo Methods



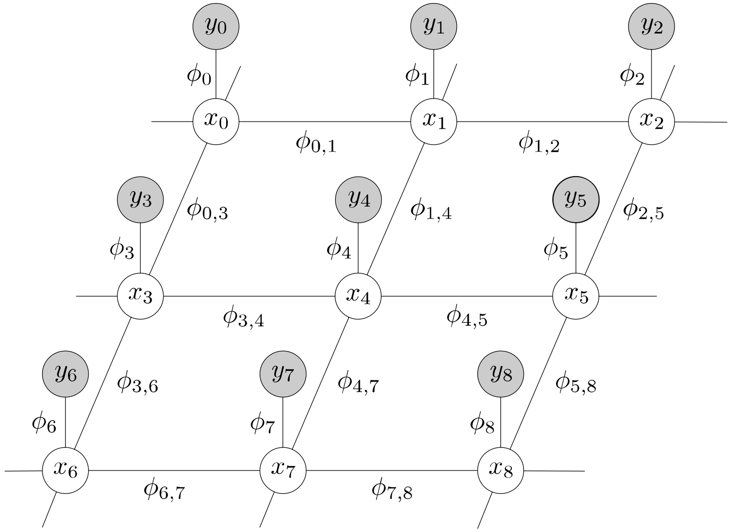
Approximating pi



A biased random walk that explores
the target distribution P

[https://wiki.ubc.ca/Course:CPSC522/MCMC]

Gibbs Sampling Inference



Maximum A Posteriori Inference:

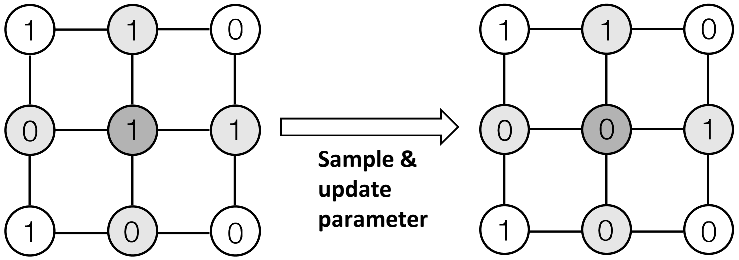
$$P(x,y) = \frac{1}{Z}\prod_{i\in V}\phi_i(x_i,y_i)\prod_{i,j\in E}\phi_{i,j}(x_i,x_j)$$

$$\begin{aligned} x^* &= \arg\max P(x|y) = \arg\min_x E(x|\theta) \\ &= \arg\min_x \left\{ \sum_{i\in\nu}\theta_i(x_i) + \sum_{i,j\in\varepsilon}\theta_{i,j}(x_i,x_j) \right\} \end{aligned}$$

Algorithm 1 Gibbs Sampler

1: *Initialize* x^0
2: **for** $t = 1$ to T **do**
3:     **for** $i = 1$ to n **do**
4:         $x_i^{(t+1)} \sim P(x_i | x_1^{(t+1)}, ..., x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, ..., x_n^{(t)})$
5:     **end for**
6: **end for**
7: **return** x =0

Gibbs sampling on Markov Random Field

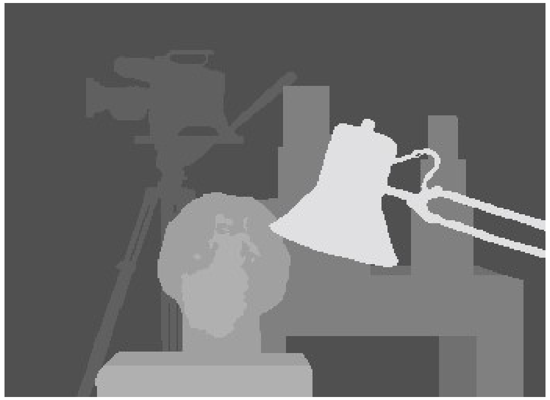


$$\forall x \in V : x \perp V \backslash x \mid neighbors(x)$$

Stereo Matching Using Gibbs Sampling

Input

Ground Truth



Parallelizing Gibbs Sampling

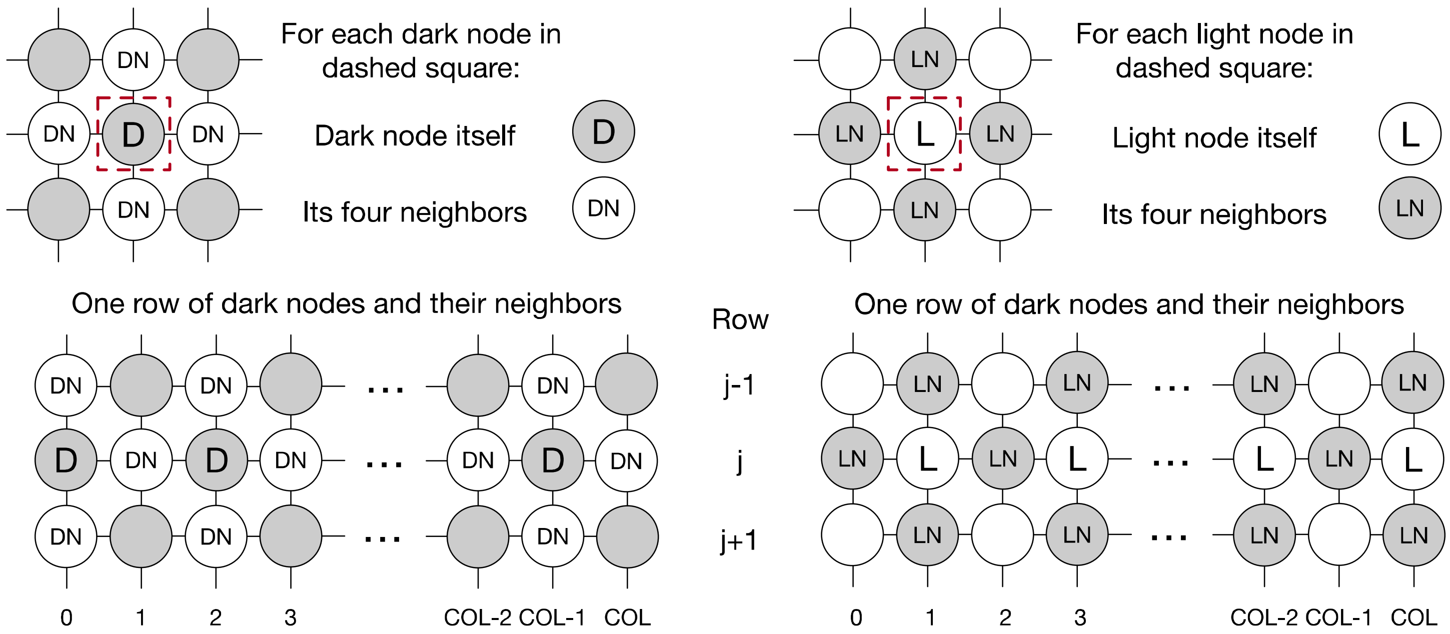
Geman & Geman stated,

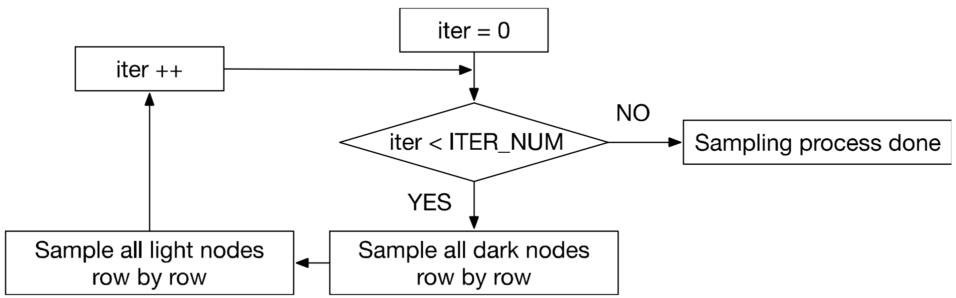
“the MRF can be divided into collections of variables with each collection assigned to an independently running asynchronous processor.”

Three types of parallelism:

- **Naïve**: Run multiple parallel chains independently
- **Algorithmic**: Graph-coloring and blocking:
  Blocked, Chromatic (Gonzalez), Splash (Gonzalez)
- **Empirical**: Asynchronous (Hogwild!) updates of partitioned graphs
  Newman et al. (AD-LDA), De Sa et al. (2016 ICML best paper)

Chromatic Gibbs Sampling

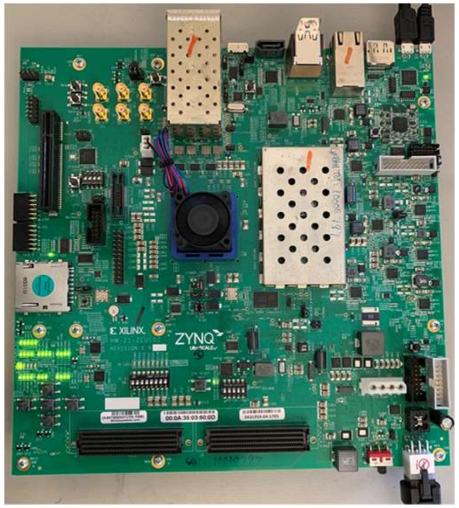




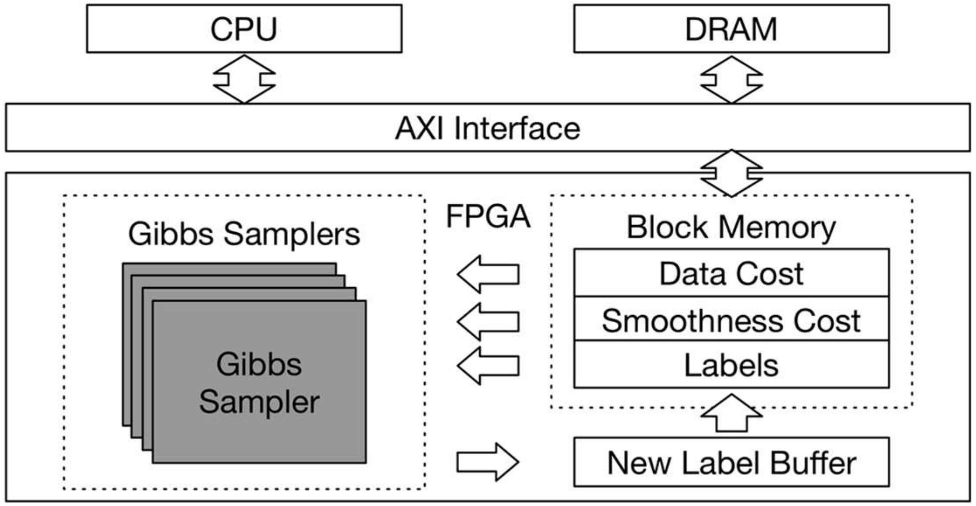
Conditional Independence via Local Markov Property

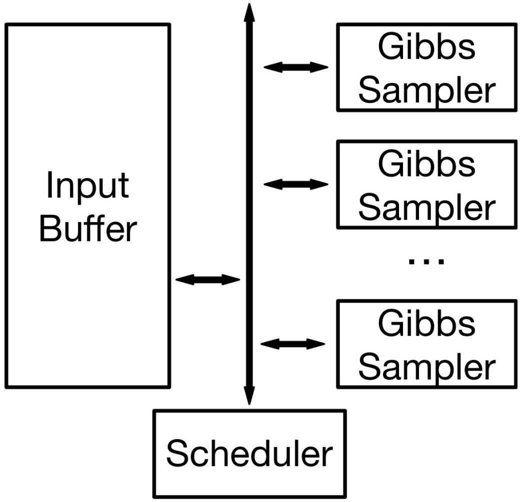
$$\forall x \in V : x \perp V \backslash x \mid neighbors(x)$$

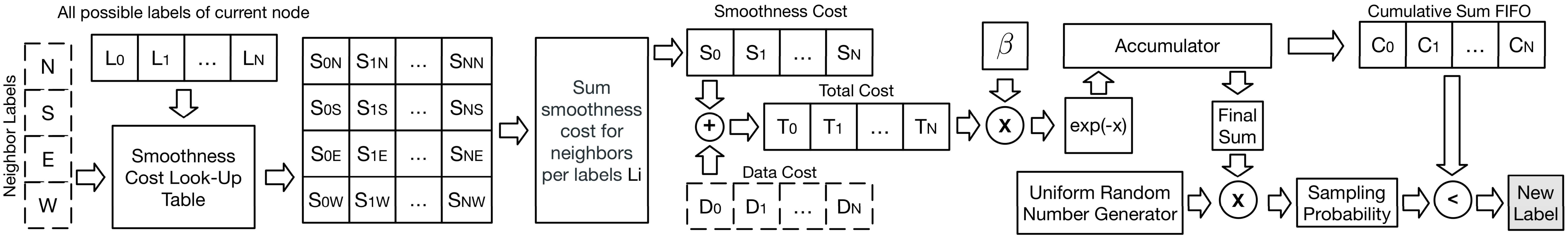
Hybrid CPU-FPGA Architecture



Xilinx Zynq UltraScale+ ZCU102-ES2

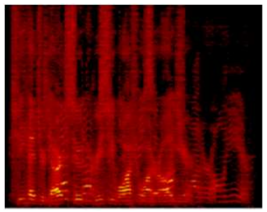




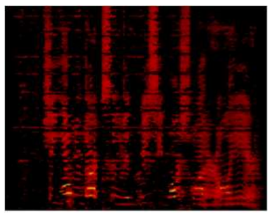


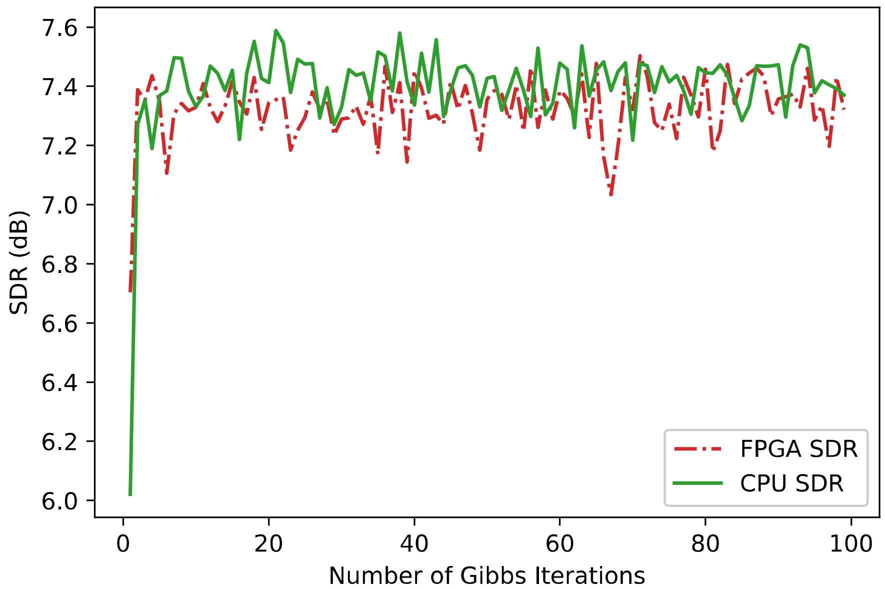
Running Sound Source Separation

Noisy mixture

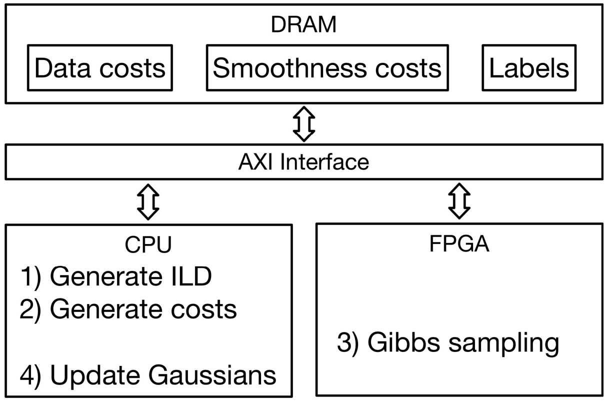


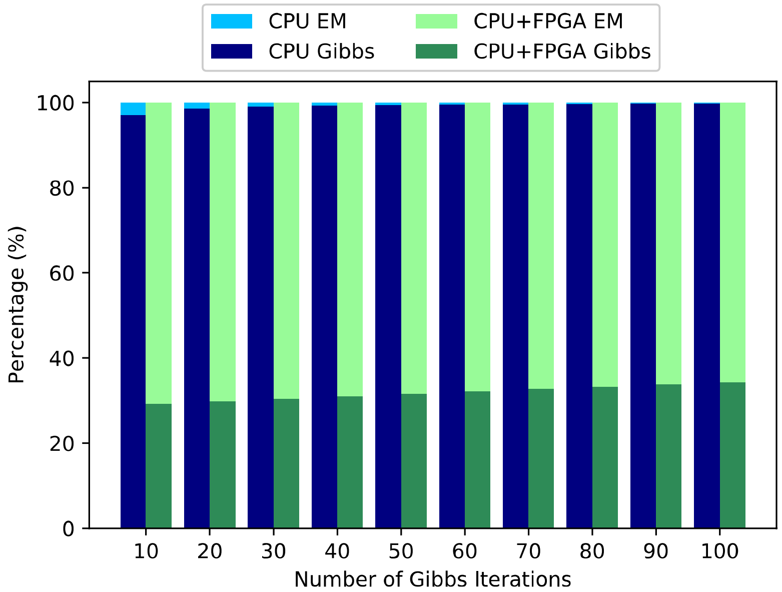
Separated source





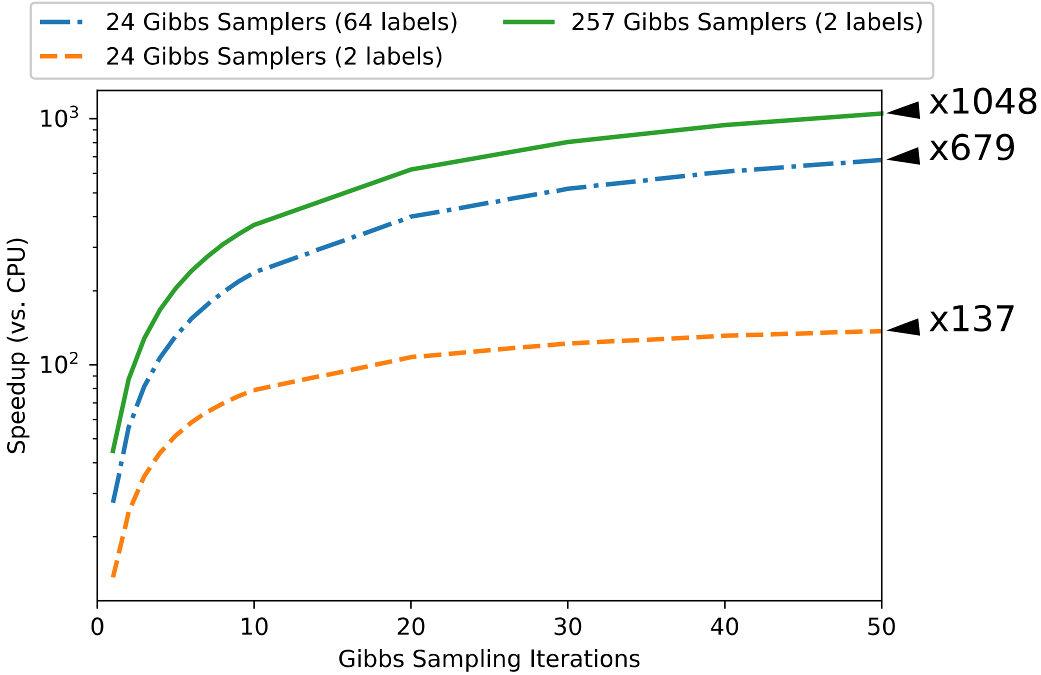
Compute Partition





230x speedup over ARM Cortex-A53

Speedups

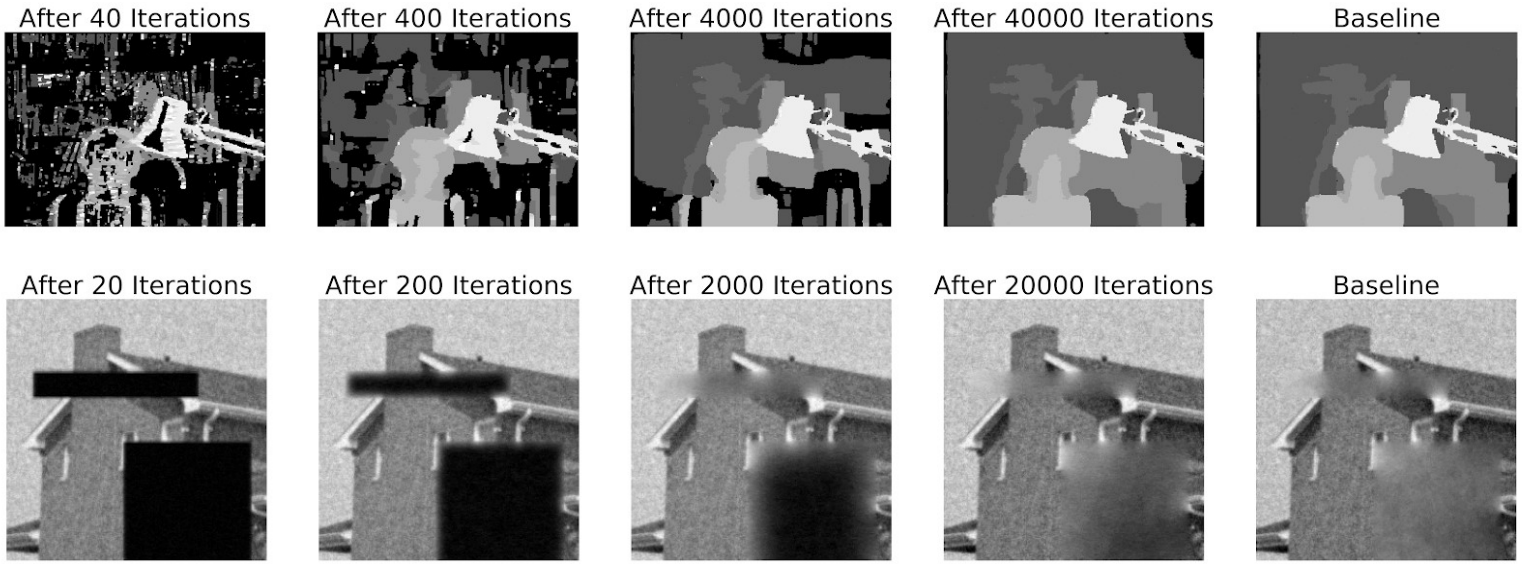


1048x speedup and 99.8% energy reduction vs. ARM Cortex A53 for binary label MRF Gibbs sampling

Number of Iterations vs. Quality of the Solution

Stereo matching: *tsukuba*

Image restoration: *house*



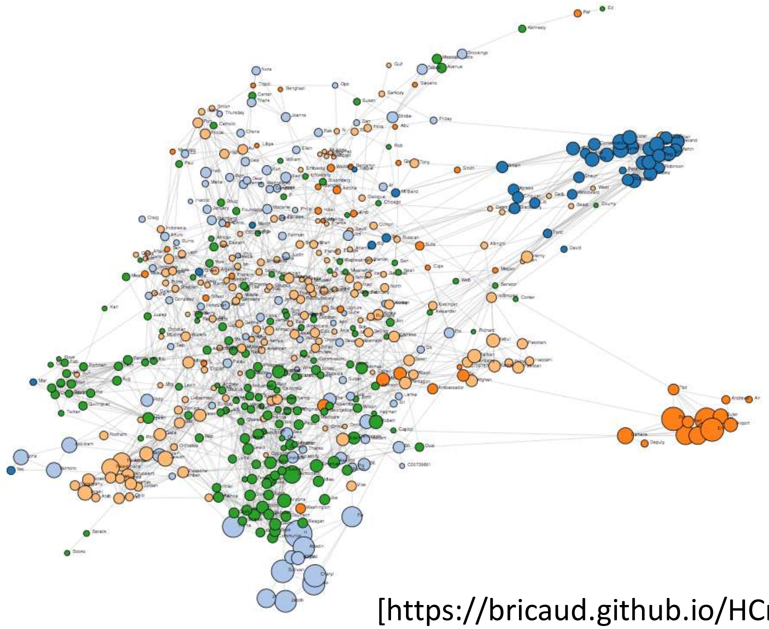
Sound source separation

Iterations	SDR
2	4.6266
4	5.0489
8	6.0176
16	6.8822

Future Work

- Asynchronous Gibbs Sampling
- Accelerating more complex graphs
  - More complex structured graphs
  - Unstructured graphs
- Challenges
  - Programmable inference architecture
  - Probabilistic programming languages
  - Compilers, IR

Hilary Clinton's emails



[https://bricaud.github.io/HCmails]

THANK YOU

This work is supported by the Semiconductor Research Corporation (SRC) and DARPA.

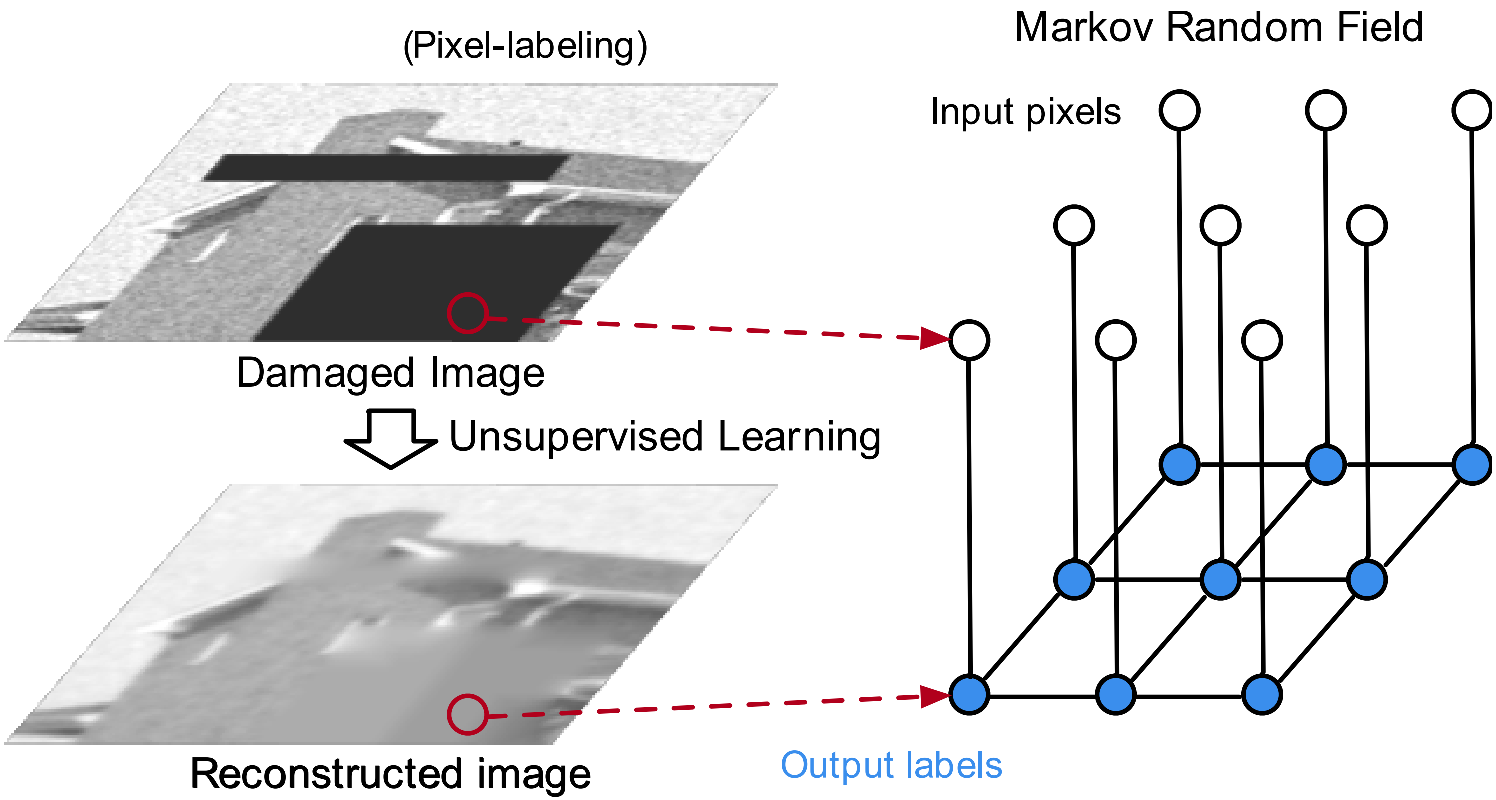
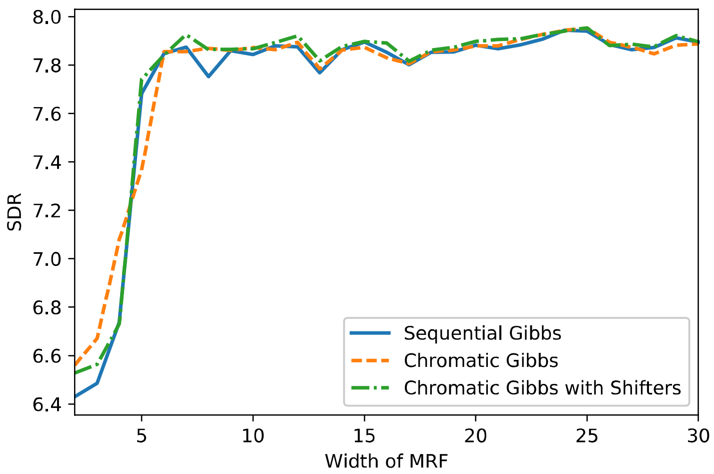


Image reconstruction : 256 labels using 64
Image segmentation : 2 labels
Stereo matching : 16-64 labels
Sound source separation : 2 labels

Nodes represents random variables corresponding to input pixels and output labels and edges encode a probability distribution over them.

Gibbs Sampler Optimization for Source Separation



Optimizations:
Multipliers -> Shifters

VERSION	LUT	LUTRAM	FF	BRAM	DSP	BUFG
MULTIPLIER	834	26	825	0	16	1
SHIFTER	929	27	852	0	8	1

MRF size: 513x24