Markov chain Monte Carlo Basics

Frank Dellaert

References

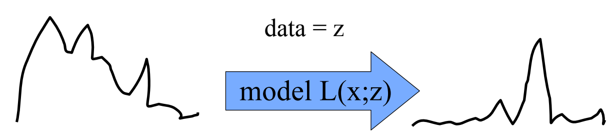
- **Smith & Gelfand**, Bayesian Statistics Without Tears
- **MacKay**, Introduction to Monte Carlo Methods
- **Gilks et al**, Introducing MCMC
- **Gilks et al**, MCMC in Practice
- **Neal**, Probabilistic Inference using MCMC Methods
- **Robert & Casella,** Monte Carlo Statistical Methods

Outline

- **Inference and Estimation via Sampling**
- **Ways to Sample**
- **Markov Chains**
- **Metropolis-Hastings**
- **Metropolis & Gibbs**

Recap: Bayes Law

$$P(x|z) \sim L(x;z)P(x)$$



Belief before = P(x)

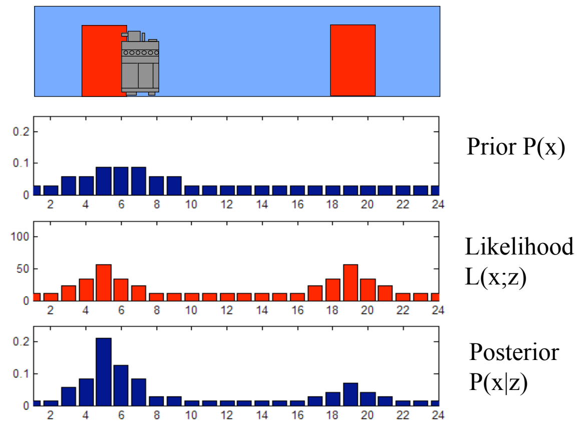
Belief after = P(x|z)

Prior Distribution of x

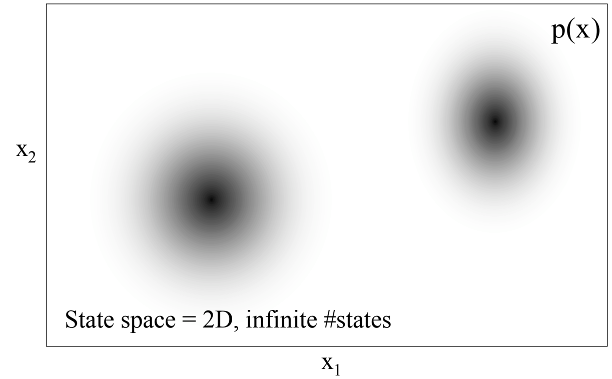
Likelihood of x given Z

Posterior Distribution of x given Z

Example: 1D Robot Localization



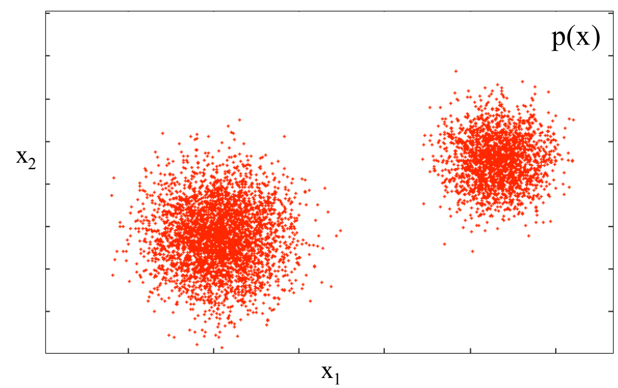
Example: 2D Robot Location



Various Density Representations

- Gaussian centered around mean x,y
- Mixture of Gaussians
- Finite element i.e. histogram
- Does not scale to large state spaces encountered in computer vision

Sampling as Representation



Sampling Advantages

- Arbitrary densities
- Memory = O(#samples)
- Only in "Typical Set"
- Great visualization tool !

- minus: Approximate

Inference = Monte Carlo Estimates

- Estimate expectation of any function f:

$$E_{P(x)}[f(x)] = \int_x f(x)P(x)d^N x$$

$$E_{P(x)}[f(x)] \approx \frac{1}{R}\sum_{r=1}^{R} f(x^{(r)})$$

Outline

- **Inference and Estimation via Sampling**
- **Ways to Sample**
- **Markov Chains**
- **Metropolis-Hastings**
- **Metropolis & Gibbs**

How to Sample ?

- Target Density π(x)
- Assumption: we can evaluate π(x) up to an arbitrary multiplicative constant

- Why can't we just sample from π(x) ??

How to Sample ?

- Numerical Recipes in C, Chapter 7
- Transformation method: Gaussians etc…
- Rejection sampling
- Importance sampling

Rejection Sampling

- Target Density $\pi(x)$
- Proposal Density $q(x)$
- π and q need only be known up to a factor

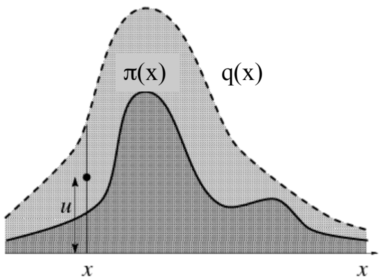


Image by MacKay

Importance Sampling

- Sample $x^{(r)}$ from $q(x)$
  - $w_r = \pi(x^{(r)})/q(x^{(r)})$

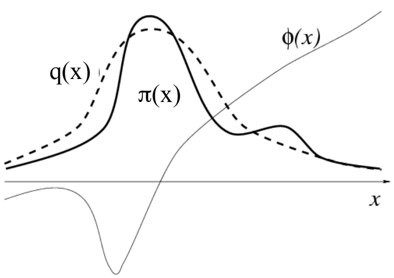
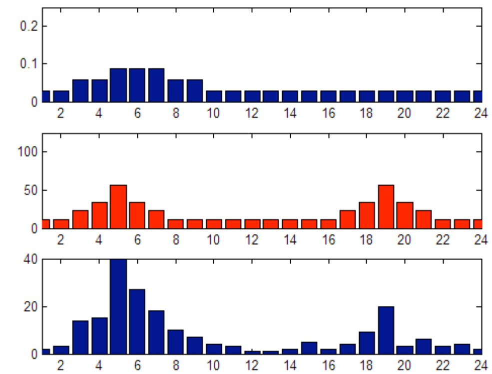


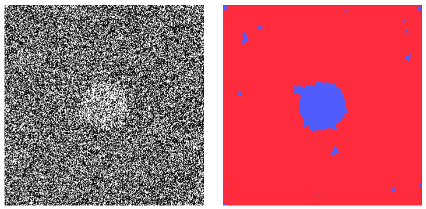
Image by MacKay

1D Importance Sampling



Segmentation Example

- Binary Segmentation of image



Probability of a Segmentation

- Very high-dimensional
- 256*256 pixels = 65536 pixels
- Dimension of state space N = 65536 !!!!

- # binary segmentations = finite !
- $2^{65536} = 2*10^{19728} >> 10^{79}$ = atoms in universe

Representation P(Segmentation)

- Histogram ? No !
- Assume pixels independent
  $P(x_1x_2x_2...)=P(x_1)P(x_2)P(x_3)...$
- Approximate solution: mean-field methods
- Approximate solution: samples !!!

Sampling in High-dimensional Spaces

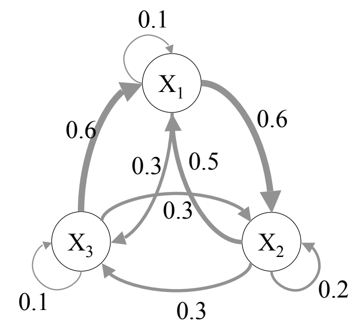
Standard methods fail:

- Rejection Sampling
  - Rejection rate increase with N -> 100%
- Importance Sampling
  - Same problem: vast majority weights -> 0

Outline

- Inference and Estimation via Sampling
- Ways to Sample
- **Markov Chains**
- Metropolis-Hastings
- Metropolis & Gibbs

A simple Markov chain

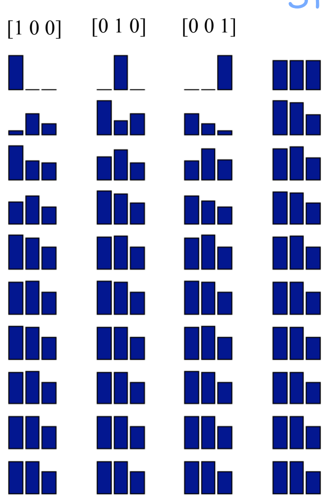


K= [

0.1	0.5	0.6
0.6	0.2	0.3
0.3	0.3	0.1

]

Stationary Distribution π



q_0

$q_1 = K\, q_0$

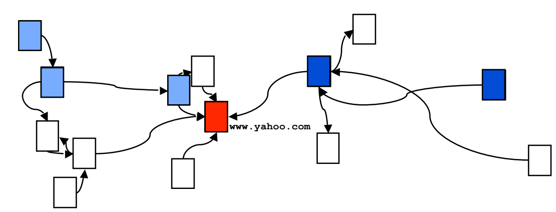
$q_2 = K\, q_1 = K^2\, q_0$

$q_3 = K\, q_2 = K^2\, q_1 = K^3\, q_0$

$q_{10} = K\, q_9 = \ldots K^{10}\, q_0$

The Web as a Markov Chain

Where do we end up if we click hyperlinks randomly ?



Answer: stationary distribution !

Eigen-analysis

K =

$$\begin{matrix} 0.1000 & 0.5000 & 0.6000 \\ 0.6000 & 0.2000 & 0.3000 \\ 0.3000 & 0.3000 & 0.1000 \end{matrix}$$

E =

$$\begin{matrix} 0.6396 & 0.7071 & -0.2673 \\ 0.6396 & -0.7071 & 0.8018 \\ 0.4264 & 0.0000 & -0.5345 \end{matrix}$$

D =

$$\begin{matrix} 1.0000 & 0 & 0 \\ 0 & -0.4000 & 0 \\ 0 & 0 & -0.2000 \end{matrix}$$

$KE = ED$

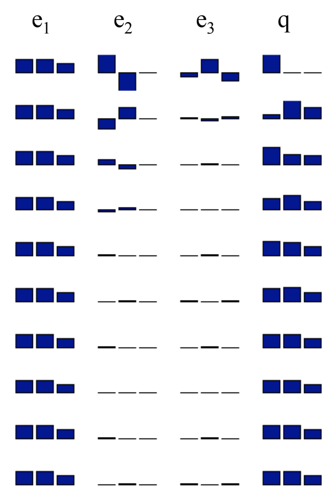
Eigenvalue v_1 always 1

Stationary distribution

$\pi = e_1/\text{sum}(e_1)$

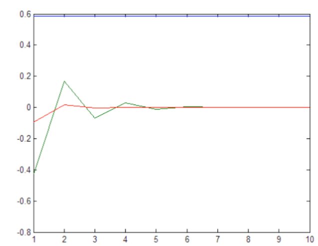
i.e. $K \pi = \pi$

Eigen-analysis



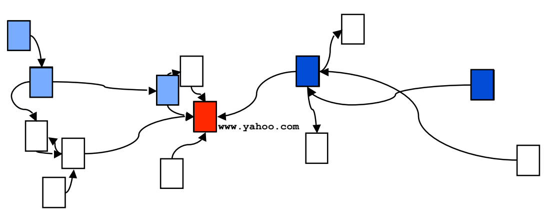
$q_n = K^n q_0 = E D^n c$

$= \pi + c_2 v_2^n e_2 + c_3 v_3^n e_{3+\dots}$



Google Pagerank

Pagerank == First Eigenvector of the Web Graph !



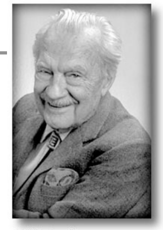
Computation assumes a 15% "random restart" probability

Sergey Brin and Lawrence Page , The anatomy of a large-scale hypertextual {Web} search engine, Computer Networks and ISDN Systems, 1998

Outline

- **Inference and Estimation via Sampling**
- **Ways to Sample**
- **Markov Chains**
- **Metropolis-Hastings**
- **Metropolis & Gibbs**

Brilliant Idea!



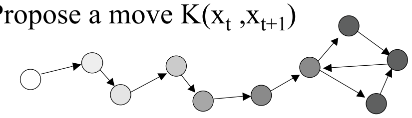
Nick Metropolis

- Published June 1953
- Top 10 algorithm !

- Set up a Markov chain
- Run the chain until stationary
- All subsequent samples are from stationary distribution

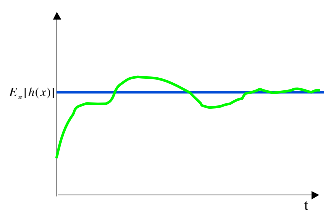
Markov chain Monte Carlo

- In high-dimensional spaces:
  - Start at $x_0 \sim q_0$
  - Propose a move $K(x_t, x_{t+1})$



- K never stored as a big matrix ☺
- K as a function/search operator

MCMC Inference



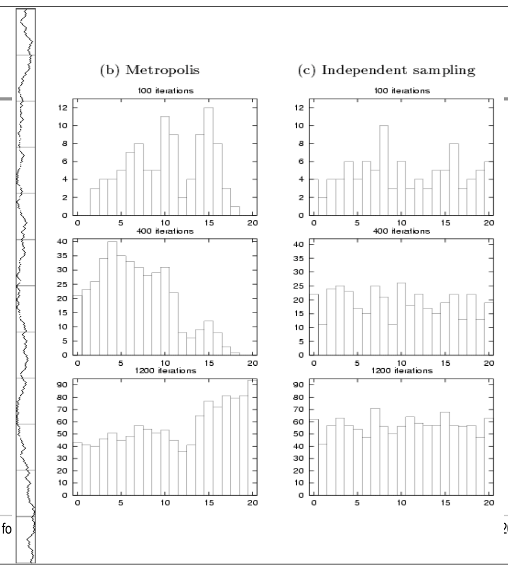
- Empirical average

$$\frac{1}{T}\sum_{t=1}^{T} h(x^{(t)})$$

converges to the expectation $E_\pi[h(x)]$
where π is the stationary distribution

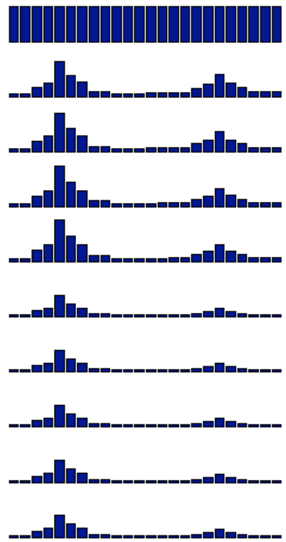
- Reason: chain is ergodic: forgets initial x_0
- In theory: no need to run multiple chains

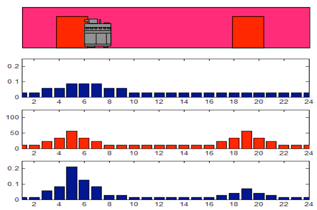
Example



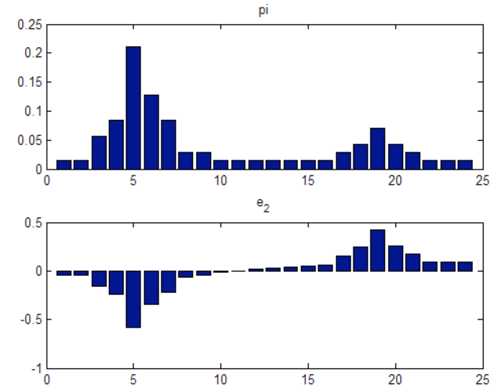
1D Robot Localization

Chain started at random
Converges to posterior





Localization Eigenvectors

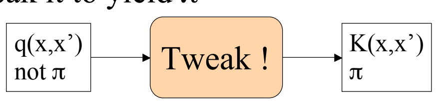


1.0000

0.9962

How do get the right chain ?

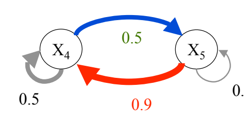
- How do we construct a transition kernel K such that π is the stationary distribution ?
- Idea: take a proposal distribution q(x,x') that is irreducible and recurrent
- Tweak it to yield π



  - Similar idea as importance/rejection sampling
  - Irreducible: you can get anywhere from anywhere
  - Recurrent: you will visit any state infinitely often

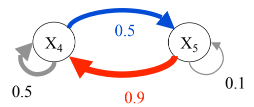
Detailed Balance

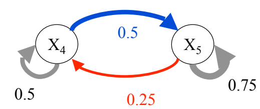
- A sufficient condition to converge to π(x) :
  K(x,x') π(x) = K(x',x) π(x')
  "Detailed Balance"
- Example that works:
  0.5 * 9/14 = 0.9 * 5/14



Tweak: Reject fraction of moves !

- Detailed balance not satisfied:
  - q(x,x') 1/3 = q(x',x) 2/3
- Tweak: insert factor a:
- 0.5 * 1/3 = a * 0.9 * 2/3
- a = 0.5 * 1/3 / (0.9 * 2/3)
  = 5/18





Metropolis-Hastings Algorithm

This leads to the following algorithm:

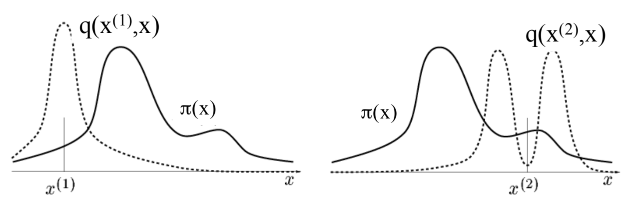
0. Start with $x^{(0)}$, then iterate:
1. propose x' from $q(x^{(t)},x')$
2. calculate ratio

$$a = \frac{\pi(x')q(x',x^{(t)})}{\pi(x^{(t)})q(x^{(t)},x')}$$

3. if a>1 accept $x^{(t+1)}=x'$
   else accept with probability a
   if rejected: $x^{(t+1)}=x^{(t)}$

Proposal Density q(x,x')

- The proposal density $q(x^{(t)},x)$ depends on $x^{(t)}$



Outline

- **Inference and Estimation via Sampling**
- **Ways to Sample**
- **Markov Chains**
- **Metropolis-Hastings**
- **Metropolis & Gibbs**

The Metropolis Algorithm

When q is symmetric, i.e., q(x,x')=q(x',x) :

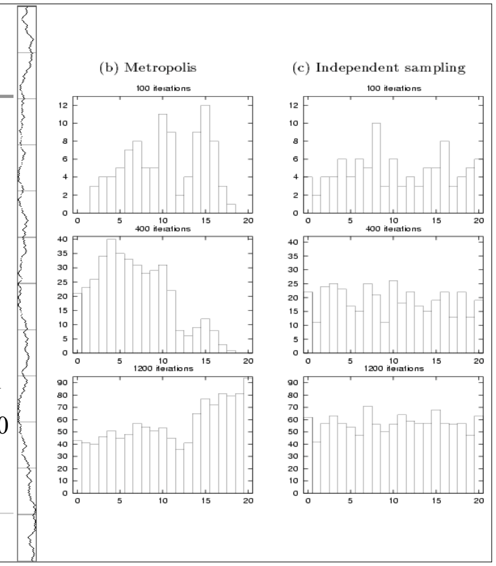
0. Start with $x^{(0)}$, then iterate:
1. propose x' from $q(x^{(t)},x')$
2. calculate ratio

$$a = \frac{\pi(x')}{\pi(x^{(t)})}$$

3. if a>1 accept $x^{(t+1)}=x'$
   else accept with probability a
   if rejected: $x^{(t+1)}=x^{(t)}$

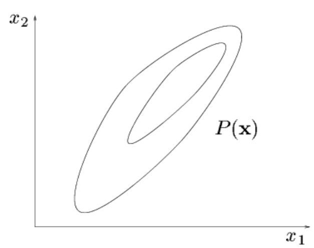
Example

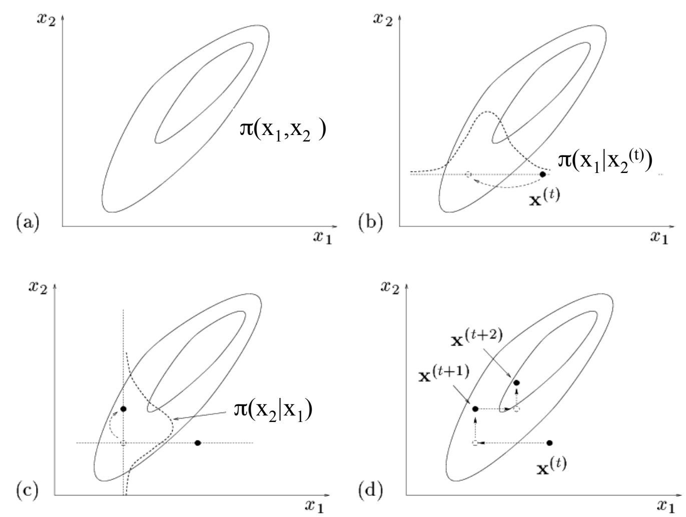
1. $x^{(0)}=10$
2. Proposal:
   x'=x-1 with Pr 0.5
   x'=x+1 with Pr 0.5
3. Calculate a:
   a=1 if x' in [0,20]
   a=0 if x'=-1 or x'=21
4. Accept if 1, reject if 0
5. Goto 2



Gibbs Sampling

- Example: target $\pi(x_1,x_2)$
- Algorithm:
  - alternate between x_1 and x_2
  - 1. sample from $x_1 \sim P(x_1|x_2)$
  - 2. sample from $x_2 \sim P(x_2|x_1)$
- After a while: samples from target density !
- Sampler equivalent of "Gauss-Seidel" iterations





Gibbs = Special Case of MH

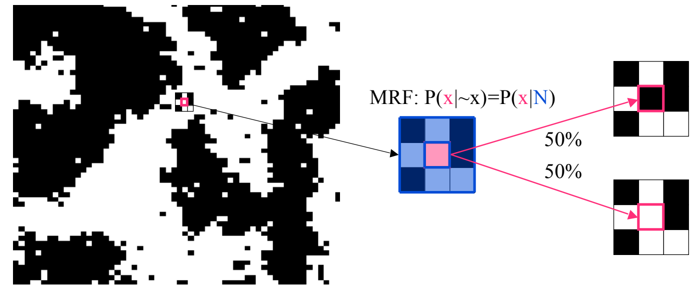
- Acceptance ratio is always 1

$$a = \frac{\pi(x')q_i(x',x^{(t)})}{\pi(x^{(t)})q_i(x^{(t)},x')}$$

$$= \frac{\pi(x')\pi(x_i^{(t)} \mid x_{\sim i}^{(t)})\delta(x_{\sim i}^{(t)},x'_{\sim i})}{\pi(x^{(t)})\pi(x'_i \mid x_{\sim i}^{(t)})\delta(x_{\sim i}^{(t)},x'_{\sim i})}$$

$$= \frac{\pi(x')\pi(x^{(t)})\pi(x_{\sim i}^{(t)})\delta(x_{\sim i}^{(t)},x'_{\sim i})}{\pi(x^{(t)})\pi(x')\pi(x_{\sim i}^{(t)})\delta(x_{\sim i}^{(t)},x'_{\sim i})} = 1$$

Gibbs Sampling in a Markov Random Field



Sampling from the Prior

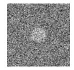
Weak Affinity to Neighbors

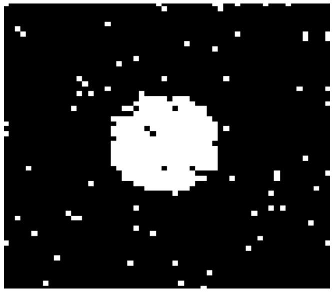
Strong Affinity to Neighbors

Sampling MRF Posterior

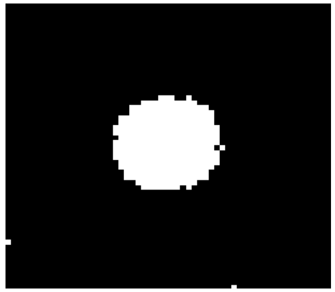
- P(x|N)
  - pulled towards 0 if data close to 0
  - pushed towards 1 if data close to 1
  - and influence of prior...

Samples from Posterior





Forgiving Prior

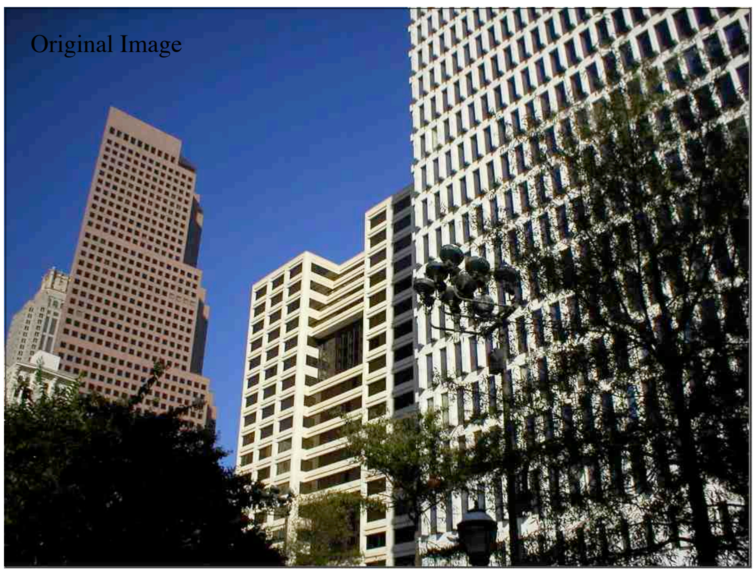


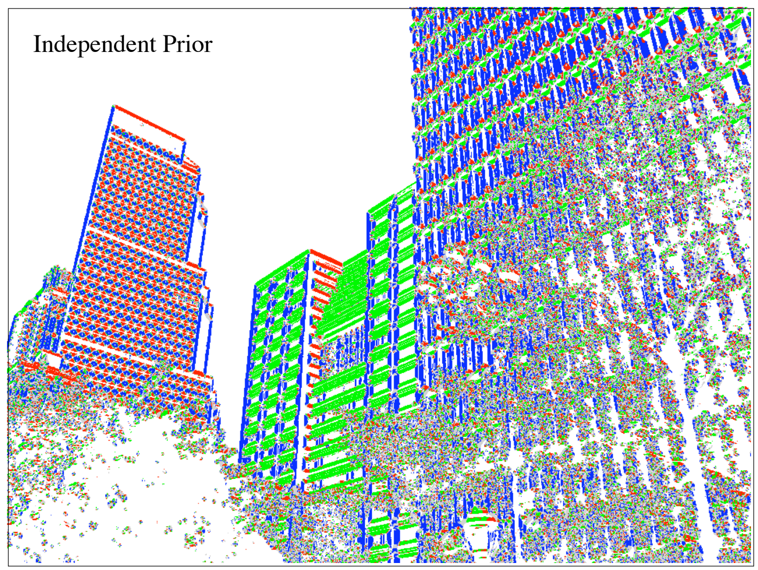
Stricter Prior

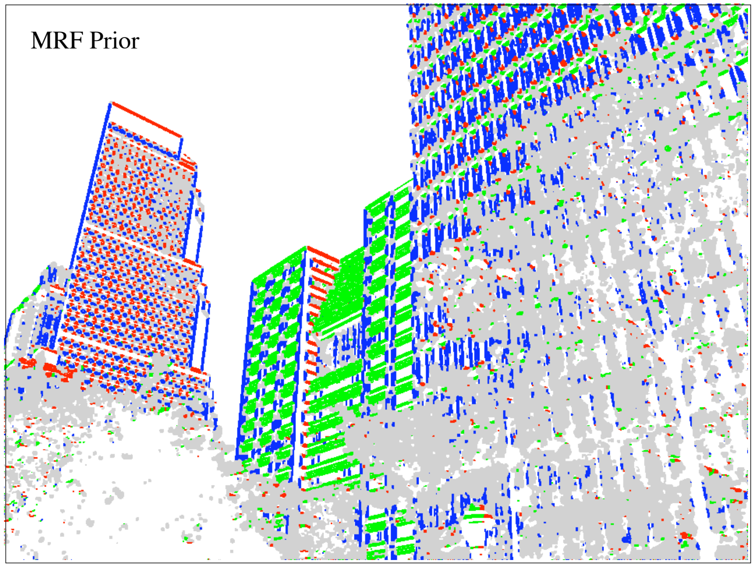
Application: Edge Classification

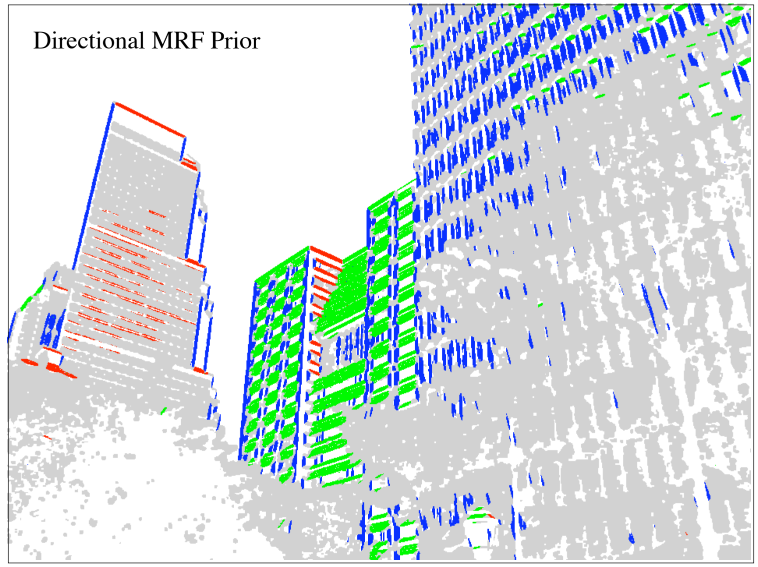
Given vanishing points of a scene, classify each pixel according to vanishing direction

ICCV05 Tutorial: MCMC for Vision. Zhu / Dellaert / Tu October 2005









Relation to Belief Propagation

- In poly-trees: BP is exact
- In MRFs: BP is a variational approximation
- Computation is very similar to Gibbs
- Difference:
  - BP Can be faster in yielding a good estimate
  - BP exactly calculates the wrong thing
  - MCMC might take longer to converge
  - MCMC approximately calculates the right thing