Motivation

How to track many INTERACTING targets?

Results: MCMC

Dancers, q=10, n=500
Probabilistic Topological Maps

Real-Time Urban Reconstruction
- 4D Atlanta, only real time, multiple cameras
- Large scale SFM: closing the loop

Current Main Effort: 4D Atlanta

Goals
- Bayesian paradigm is a useful tool to
  - Represent knowledge
  - Perform inference
- Sampling is a nice way to implement the Bayesian paradigm, e.g., Condensation
- Markov chain Monte Carlo methods are a nice way to implement sampling

References
- Neal, Probabilistic Inference using MCMC Methods
- Smith & Gelfand, Bayesian Statistics Without Tears
- MacKay, Introduction to MC Methods
- Gilks et al, Introducing MCMC
- Gilks et al, MCMC in Practice
Probability of Robot Location

\[ P(\text{Robot Location}) \]

State space = 2D, infinite #states

Density Representation

- Gaussian centered around mean \( x, y \)
- Mixture of Gaussians
- Finite element i.e. histogram
- Larger spaces \( \rightarrow \) We have a problem!

Sampling as Representation

Sampling Advantages

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

minus: Approximate

How to Sample?

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

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

How to Sample?

- Numerical Recipes in C, Chapter 7
- Transformation method: Gaussians etc...
- Rejection sampling
- Importance sampling
- Markov chain Monte Carlo
Rejection Sampling

- Target Density $P$
- Proposal Density $Q$
- $P$ and $Q$ need only be known up to a factor: $P^*$ and $Q^*$
- must exist $c$ such that $cQ^* \geq P^*$ for all $x$

The Good...

...the Bad...

...and the Ugly.

Mean and Variance of a Sample

Mean

$$\mu = \int xP(x)dx$$
$$\mu \approx \frac{1}{N} \sum_{r=1}^{N} x^{(r)}$$

Variance (1D)

$$\sigma^2 = \int (x - \mu)^2 P(x)dx$$
$$\sigma^2 \approx \frac{1}{N} \sum_{r=1}^{N} (x^{(r)} - \hat{\mu})^2$$

Monte Carlo Expected Value

$$E_{P(x)}[x] = \int \int \int \alpha(x)P(x|y)dy$$
$$E_{P(x)}[x] \approx \frac{1}{N} \sum_{r=1}^{N} (x^{(r)})$$

Expected angle = 30°
Monte Carlo Estimates (General)

- Estimate expectation of any function $f$:

$$E_{P(x)}[f(x)] = \int_x f(x)P(x)dx$$

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

Bayes Law

$$P(x|Z) \sim P(Z|x)P(x)$$

Data = $Z$

Belief before = $P(x)$

Belief after = $P(x|Z)$

Prior Distribution

Likelihood

Posterior Distribution

Inference by Rejection Sampling

- $P(\text{measured\_angle}|x,y) = N(\text{predicted\_angle},3 \text{ degrees})$

Prior($x,y$)

Posterior($x,y|\text{measured\_angle}=20^\circ$)

Importance Sampling

- Good Proposal Density would be: prior !
- Problem:
  - No guaranteed $c$ s.t. $cP(x) \geq P(x|z)$ for all $x$
- Idea:
  - Sample from $P(x)$
  - Give each sample $x^{(r)}$ a importance weight equal to $P(Z|x^{(r)})$

Example Importance Sampling

- Sample $x^{(r)}$ from $Q^*$
- $w_r = P^*(x^{(r)})/Q^*(x^{(r)})$

Importance Sampling (general)
Important Expectations

- Any expectation using weighted average:

\[ w_r = \frac{p^*(x^{(r)})}{q^*(x^{(r)})} \]

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

Particle Filtering

1D Robot Localization

- Histogram approach does not scale
- Monte Carlo Approximation
- Sample from \( P(X|Z) \) by:
  - Sample from prior \( P(x) \)
  - Weight each sample \( x^{(r)} \) using an importance weight equal to likelihood \( L(x^{(r)}, Z) \)

Importance Sampling

1D Importance Sampling

- Recursive Importance Sampling w modeled dynamics

\[ \pi_i^{(s)} = P(Z_i | X_i^{(s)}) \]

First appeared in 70's, re-discovered by Kitagawa, Isard, ...
3D Particle filter for robot pose: Monte Carlo Localization
Dellaert, Fox & Thrun ICRA 99

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 !
- $65536^2 \approx 4,294,967,296$

Representation $P(\text{Segmentation})$
- Histogram? I don’t think so!
- Assume pixels independent
  $P(x_1 x_2 x_3 \ldots) = P(x_1) P(x_2) P(x_3) \ldots$
- Markov Random Fields
  - Pixel is independent given its neighbors
- Clearly a problem!
- Giveaway: samples !!!

Sampling in High-dimensional Spaces
- Exact schemes?
  - If only we were so lucky!
- Rejection Sampling
  - Rejection rate increase with $N \to 100$
- Importance Sampling
  - Same problem: vast majority weights $\to 0$

Markov Chains
A simple Markov chain

\[
\begin{bmatrix}
0.1 & 0.5 & 0.6 \\
0.6 & 0.2 & 0.3 \\
0.3 & 0.3 & 0.1 \\
\end{bmatrix}
\]

Stationary Distribution

\[
q_{10} = K q_0 = \ldots K^{10} q_0
\]

The Web as a Markov Chain

Where do we end up if we click hyperlinks randomly?

Answer: stationary distribution!

Eigen-analysis

\[
K = \begin{bmatrix}
0.1000 & 0.5000 & 0.6000 \\
0.6000 & 0.2000 & 0.3000 \\
0.3000 & 0.3000 & 0.1000 \\
\end{bmatrix}
\]

\[
E = \begin{bmatrix}
0.6396 & 0.7071 & -0.2673 \\
0.6396 & -0.7071 & 0.8018 \\
0.4264 & 0.0000 & -0.5345 \\
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
1.0000 & 0 & 0 \\
0 & -0.4000 & 0 \\
0 & 0 & -0.2000 \\
\end{bmatrix}
\]

\[
KE = ED
\]

Eigenvalue \( \lambda_1 \) always 1

Stationary: \( q / \text{sum}(e_1) \)

i.e. \( Kp = p \)

Google Pagerank

Pagerank \( = \) First Eigenvector of the Web Graph!

Computation assumes a 15% "random restart" probability

Markov chain Monte Carlo

- Brilliant Idea!
  - 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 $K(x_{t+1} | x_t)$

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

Example

- How do we get the right chain?
  - Detailed balance:
    - $K(y | x) p(x) = K(x | y) p(y)$
  - $0.5 \times 9/14 = 0.9 \times 5/14$

- Reject fraction of moves!
  - Detailed balance:
    - $K(y | x) \frac{1}{3} = K(x | y) \frac{2}{3}$
  - $0.5 \times \frac{1}{3} = a \times 0.9 \times \frac{2}{3}$
  - $a = 0.5 \times \frac{1}{3} / (0.9 \times \frac{2}{3}) = 5/18$
Metropolis-Hastings Algorithm

- pick $x^{(0)}$, then iterate over:

1. propose $x'$ from $Q(x';x^{(t)})$
2. calculate ratio
   $$a = \frac{P^*(x')}{P^*(x^{(t)})} \frac{Q(x^{(t)};x')}{Q(x';x^{(t)})}$$
3. if $a>1$ accept $x^{(t+1)}=x'$
   else accept with probability $a$
   if rejected: $x^{(t+1)}=x^{(t)}$

Again!

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

1D Robot Localization

- Chain started at random
- Converges to posterior

Gibbs Sampling

- MCMC method that always accepts
- Algorithm:
  - alternate between $x_1$ and $x_2$
  - 1. sample from $x_1 \sim P(x_1|x_2)$
  - 2. sample from $x_2 \sim P(x_2|x_1)$
- Rationale: easy conditional distributions
- = Gauss-Seidel of samplers

Localization Eigenvectors

- $1.0000$
- $0.9962$
Sampling Segmentations

- Prior model: Markov Random Field
- Likelihood: 1 or 0 plus Gaussian noise
- Gibbs Sampling method of choice
  - Conditional densities are easy in MRF

Sampling Prior

\[ P(\text{being one}|\text{others}) = \]
- HIGH if many ones around you
- LOW if many zeroes around you

Sampling Posterior

\[ P(\text{being one}|\text{others}) = \]
- pulled towards 0 if data close to 0
- pushed towards 1 if data close to 1
- and influence of prior...

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
  - MC might take longer to converge
  - MC approximately calculates the right thing
Relation to Belief Propagation

Application: Edge Classification

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

MAP Edge Classifications

Red: VP1  Green: VP2  Blue: VP3  Gray: Other  White: Off

Bayesian Model

\[ p(M | G, V) = \frac{p(G | M, V) p(M)}{Z} \]

\( M \) = classifications, \( G \) = gradient magnitude/direction, \( V \) = vanishing points

Prior: \( p(m) \)

Likelihood: \( p(g | m, V) \)

Independent Prior  MRF Prior

Classifications w/ MRF Prior

Gibbs sampling over 4-neighbor lattice w/ clique potentials defined as: A if i=j, B if i \( \neq \) j

Gibbs Sampling & MRFs

Gibbs sampling approximates posterior distribution over classifications at each site (by iterating and accumulating statistics)
Directional MRF

Give more weight to potentials of neighbors which lie along the vanishing direction of current model

Independent Prior

MRF Prior

Directional MRF Prior

Take Home Points!

- **Bayesian paradigm** is a useful tool to
  - Represent knowledge
  - Perform inference
- **Sampling** is a nice way to implement the Bayesian paradigm, e.g. Condensation
- **Markov chain Monte Carlo** methods are a nice way to implement sampling
**World Knowledge**

Sensor Model \( P(z|x) \)

Most often analytic expression, can be learned

**Proposal Density \( Q \)**

- \( Q(x';x) \) that depends on \( x \)

**Step Size and #Samples**

- Too large: all rejected
- Too small: random walk
- \( E[d] = \sigma \sqrt{T} \)
- Rule of thumb: \( T \geq (L/e)^2 \)
- Bummer: just a lower bound

**Discussion Example**

- \( e=1 \)
- \( L=20 \)
- \( T \geq 400 \)
- Moral: avoid random walks

**MCMC in high dimensions**

- \( e=s_{\min} \)
- \( L=s_{\max} \)
- \( T = (s_{\max}/s_{\min})^2 \)
- Good news: no curse in \( N \)
- Bad news: quadratic dependence