Lecture 6: Approximate Inference – Monte Carlo Approximations

Probabilistic Graphical Models, Koller and Friedman:

Chap 12

Approximate Inference: Monte Carlo Principle, Direct Sampling, Importance Sampling, Evidence, Rejection Sampling, MCMC, Gibbs Sampling, Collapsed Importance Sampling,

Approximate Inference

- Approximate Inference comes into play whenever exact inference is not tractable.
  - E.g. the model is not tree structured
-
  -
  -
    -
    -
-
  -
  -

Approximate Inference

- Approximate Inference comes into play whenever exact inference is not tractable.
  - E.g. the model is not tree structured

- What would we like to approximate?
  - E.g. posterior distribution p(z | x)
  - Expectations:
    - continuous: integrals may be intractable
    - discrete: sum over exponentially many states is infeasible

-
  -
  -

Approximate Inference

- Approximate Inference comes into play whenever exact inference is not tractable.
  - E.g. the model is not tree structured

- What would we like to approximate?
  - E.g. posterior distribution p(z | x)
  - Expectations:
    - continuous: integrals may be intractable
    - discrete: sum over exponentially many states is infeasible

- Conceptually there are two approaches
  - Deterministic Approximation
  - Numerical Sampling (e.g. Markov Chain Monte Carlo)

Two Approaches

1. Deterministic Approximation
   - Approximate the quantity of interest, (ie everything is Gaussian, Loopy Belief Propagation, cut some edges,...)
   - Solve the approximation analytically
   - Results depends on the quality of the approximation
   - We mentioned the projection technique (a Variational Method)
2. Numerical Sampling (Monte Carlo)
   - Take the quantity of interest
   - Use random samples to approximate it
   - Results depends on the quality and amount of random samples

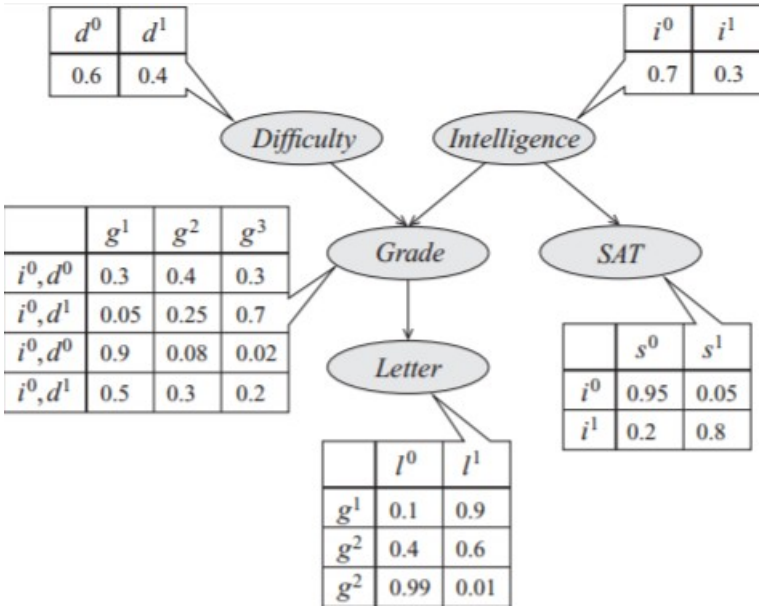
Monte Carlo Principle

$$p_N(x) \equiv \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x),$$

$$I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f\left(x^{(i)}\right) \xrightarrow[N \to \infty]{a.s.} I(f) = \int_{\mathcal{X}} f(x) p(x)\, dx.$$

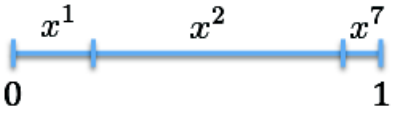
Direct Sampling

- Bayes Nets are possible to sample directly
- With evidence we can just throw away samples that do not match



$$\hat{\Phi} = \hat{E}[f(\mathbf{X})] = \frac{1}{M}\sum_{m=1}^{M} f\left(\mathbf{x}^{(m)}\right)$$

x^1	x^2	x^7
0.2	0.7	0.1



Problems with Direct Sampling

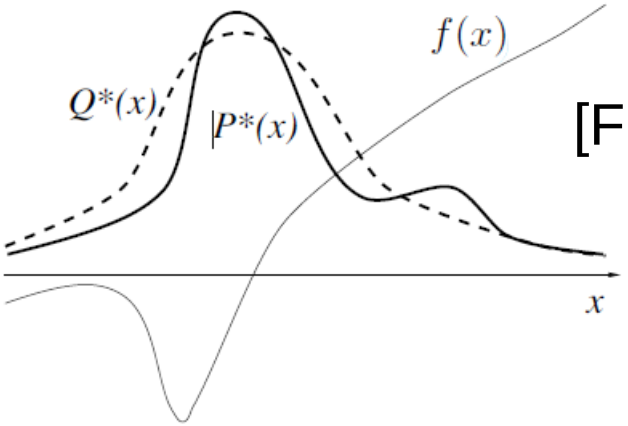
- High dimensional distributions are often impossible to sample from.
- Throw away solution for evidence can be very wasteful.

(Normalized) Importance Sampling

- Draw from a simpler distribution that wastes as few particles as possible.
- $P(x) = Q(x)\,\{P(x) / Q(x)\}$
- Importance Weight: $w_m = P(\mathbf{x}^{(m)}) / Q(\mathbf{x}^{(m)})$
- We must do inference (at a point)to do this approximate inference.

$$\hat{E}[f(\mathbf{X})] = \frac{\sum_m w_m f\left(\mathbf{x}^{(m)}\right)}{\sum_m w_m}$$

Have to pick a good Q that covers the support!



[From (3)]

'Unnormalized' uses a normalized P and Q (ie. Z is known) so no denominator.

(Normalized) Importance Sampling

- Importance Weight: $w_m = P(\mathbf{x}^{(m)}) / Q(\mathbf{x}^{(m)})$
- If there is evidence then we have to find a Q that works well for $P(\mathbf{X}\text{-}\mathbf{Y}|\mathbf{Y})$.

$$\hat{E}[f(\mathbf{X})] = \frac{\sum_m w_m f\left(\mathbf{x}^{(m)}\right)}{\sum_m w_m}$$

Have to pick a good Q that covers the support!

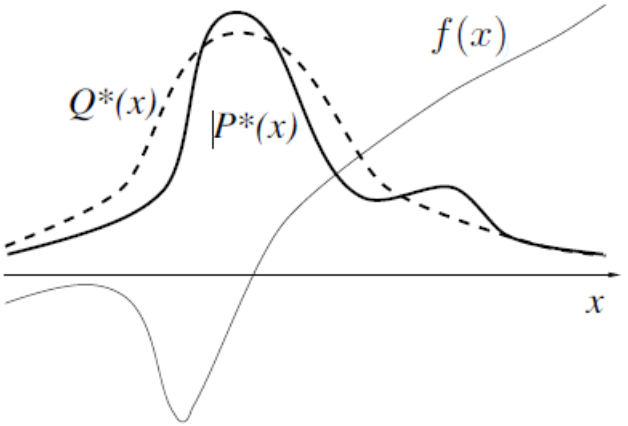
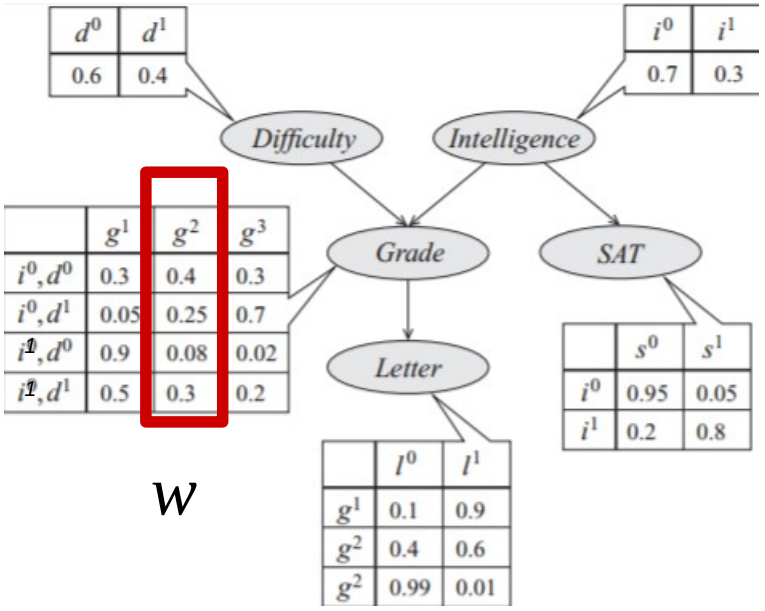
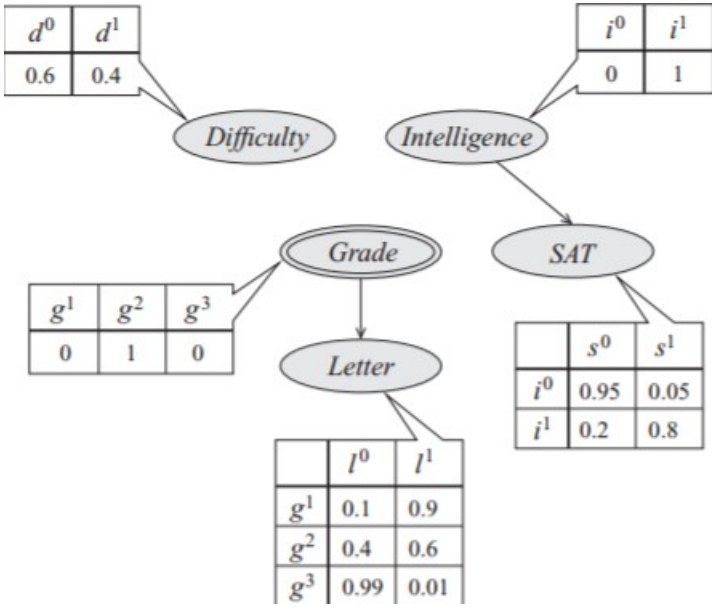


Figure from Lectures of: Veselin Stoyanov, Alexandre Klementiev and Shane Bergsman

Importance Sampling with Evidence

- If we know Grade is g^2 and the intelligence is i^1.
- Q: We modify the graph and just sample away.
- Problem is again that we are likely to sample parents that do not match well with the grade.

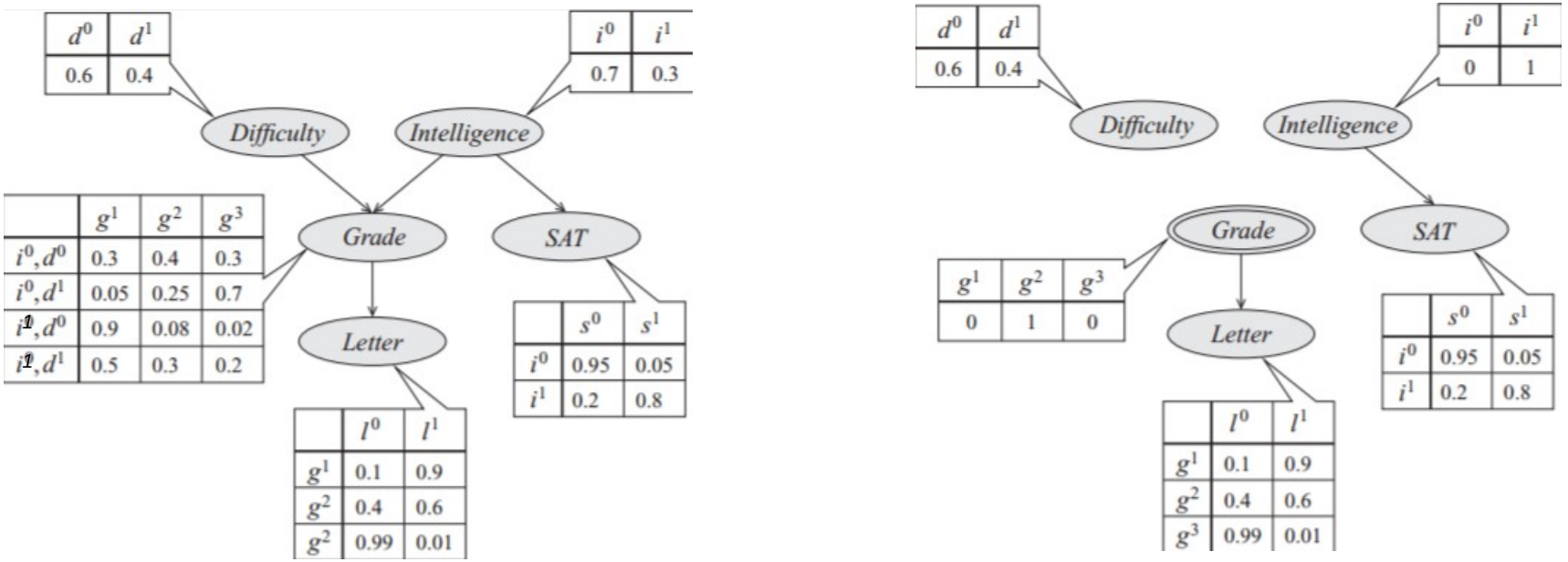




Importance Sampling with Evidence

- The normalized sampling of $p(y \mid e)$ works well if evidence is in the roots but not if it is in the leaves (for Bayes Nets) .

Mutilated BN



Importance Sampling with Evidence

- So we sample along the 'topological ordering' and when we come to our 'knowns', $\{\boldsymbol{e}\}$, we plug them in. We then multiply w_m by the conditional probability of that known.
- The weight is the conditional probability, P, since the Q here is simple 1. (Look at the multilated graph's grade table)

Ratio Importance Sampling with Evidence

An alternative is 'ratio' method, for a specific event $\boldsymbol{y}$:

$$\bar{p}(\boldsymbol{y}|\boldsymbol{e}) = \bar{p}(\boldsymbol{y}, \boldsymbol{e}) / \bar{p}(\boldsymbol{e}) = M' \Sigma_M \; w_m / (M \Sigma_{M'} \; w'_m)$$

Rejection Sampling

Same setup: $P(\boldsymbol{x})$ is too complex to sample from, and we have a simpler proposal $Q(\boldsymbol{x})$

Additional assumption:

$$\exists c = const : \forall \boldsymbol{x}, cQ^*(\boldsymbol{x}) > P^*(\boldsymbol{x})$$

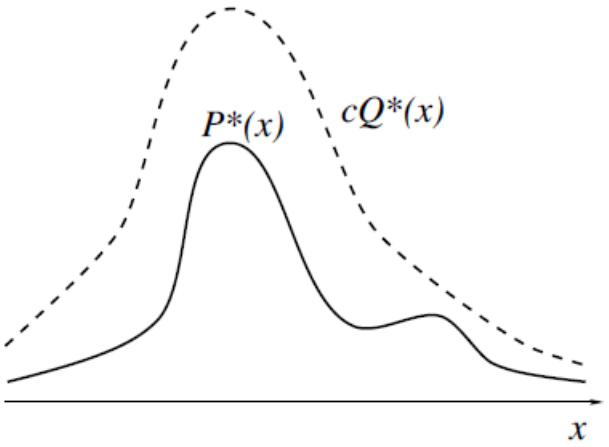


Figure from Lectures of: Veselin Stoyanov, Alexandre Klementiev and Shane Bergsman

Rejection Sampling

Generate a sample x from $Q^*(X)$. *Whats with the *?*

Evaluate $cQ^*(x)$, and sample r.v. u uniform on $[0, cQ^*(x)]$.

Evaluate $P^*(x)$, reject x if $u > P^*(x)$, else accept.

– Continue until M samples are accepted

Prob = $Q^*(x)P(u < P^*(x))$

$= Q^*(x)\, P^*(x)\, /(cQ^*(x))$

$= P^*(x)\, /c$

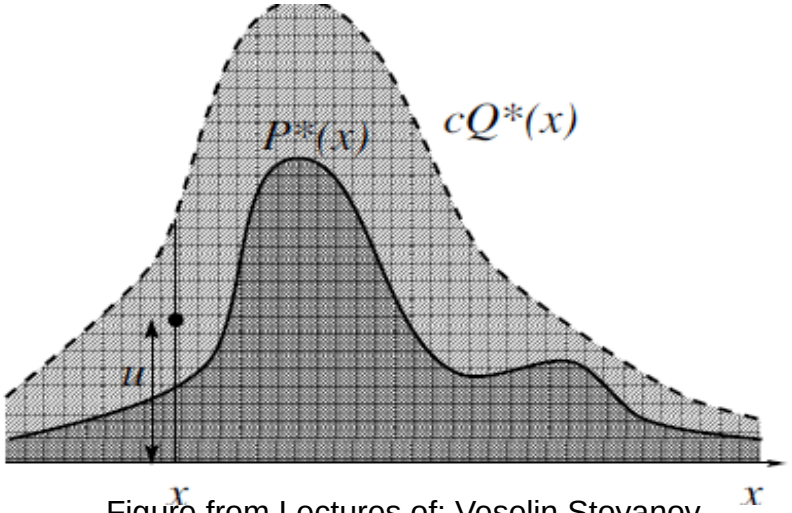
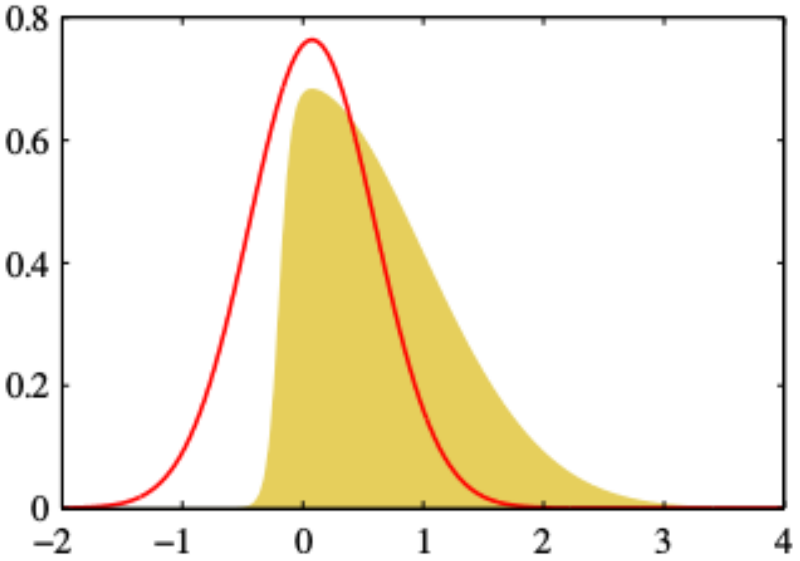


Figure from Lectures of: Veselin Stoyanov, Alexandre Klementiev and Shane Bergsman

Rejection and Importance Sampling

- Problem is that proposal distribution, Q, might not be close to P leading to many unimportant samples.
- Also P may have local maxima (modes) that are not in Q.

Laplace Approximation



Consider:

$$p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$$

Goal: Find a Gaussian approximation $q(\mathbf{z})$ which is centered on a mode of the distribution $p(\mathbf{z})$.

At a stationary point $\mathbf{z}_0$ the gradient $\bigtriangledown \tilde{p}(\mathbf{z})$ vanishes. Consider a Taylor expansion of $\ln \tilde{p}(\mathbf{z})$:

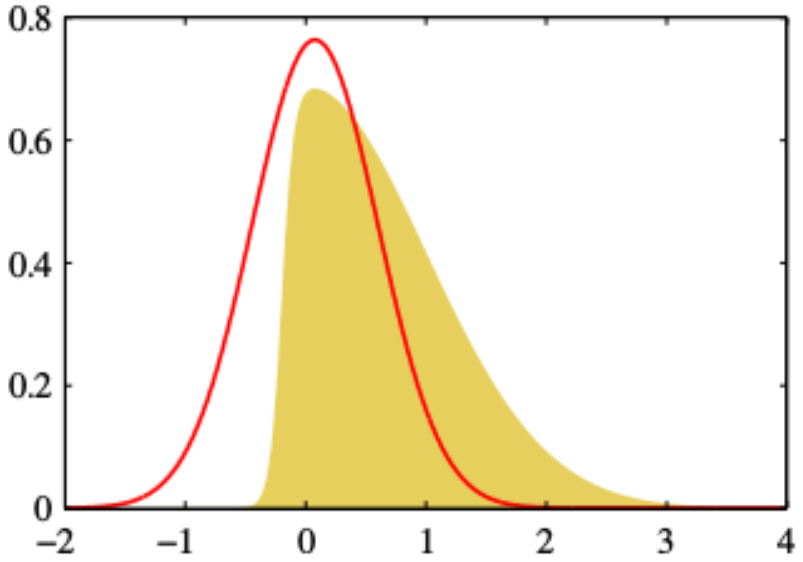
$$\ln \tilde{p}(\mathbf{z}) \approx \ln \tilde{p}(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T A (\mathbf{z} - \mathbf{z}_0)$$

where A is a Hessian matrix:

$$A = - \bigtriangledown \bigtriangledown \ln \tilde{p}(\mathbf{z})|_{z=z_0}$$

Figure: Ruslan Salakhutdinov, BCS and CSAIL, MIT

Laplace Approximation



Consider:

$$p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$$

Goal: Find a Gaussian approximation $q(\mathbf{z})$ which is centered on a mode of the distribution $p(\mathbf{z})$.

At a stationary point $\mathbf{z}_0$ the gradient $\bigtriangledown\tilde{p}(\mathbf{z})$ vanishes. Consider a Taylor expansion of $\ln\tilde{p}(\mathbf{z})$:

$$\ln\tilde{p}(\mathbf{z}) \approx \ln\tilde{p}(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T A(\mathbf{z}-\mathbf{z}_0)$$

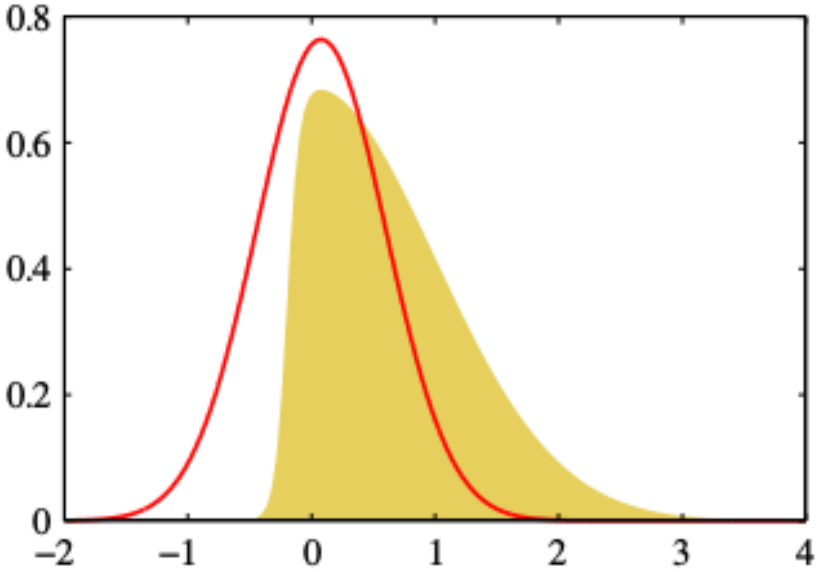
where A is a Hessian matrix:

$$A = -\bigtriangledown\bigtriangledown\ln\tilde{p}(\mathbf{z})|_{z=z_0}$$

Figure: Ruslan Salakhutdinov, BCS and CSAIL, MIT

Notice: The stationary point is the MAP point given some data.

Laplace Approximation



Consider:

$$p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$$

Goal: Find a Gaussian approximation $q(\mathbf{z})$ which is centered on a mode of the distribution $p(\mathbf{z})$.

Exponentiating both sides:

$$\tilde{p}(\mathbf{z}) \approx \tilde{p}(\mathbf{z}_0) \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T A(\mathbf{z}-\mathbf{z}_0) \right)$$

We get a multivariate Gaussian approximation:

$$q(\mathbf{z}) = \frac{|A|^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T A(\mathbf{z}-\mathbf{z}_0) \right)$$

Figure: Ruslan Salakhutdinov, BCS and CSAIL, MIT

Laplace Approximation

Remember $p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{\mathcal{Z}}$, where we approximate:

$$\mathcal{Z} = \int \tilde{p}(\mathbf{z}) d\mathbf{z} \approx \tilde{p}(\mathbf{z}_0) \int \exp\left(-\frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T A (\mathbf{z} - \mathbf{z}_0) \right) = \tilde{p}(\mathbf{z}_0) \frac{(2\pi)^{D/2}}{|A|^{1/2}}$$

Bayesian Inference: $P(\theta|\mathcal{D}) = \frac{1}{P(\mathcal{D})} P(\mathcal{D}|\theta) P(\theta)$.

Identify: $\tilde{p}(\theta) = P(\mathcal{D}|\theta)P(\theta)$ and $\mathcal{Z} = P(\mathcal{D})$:

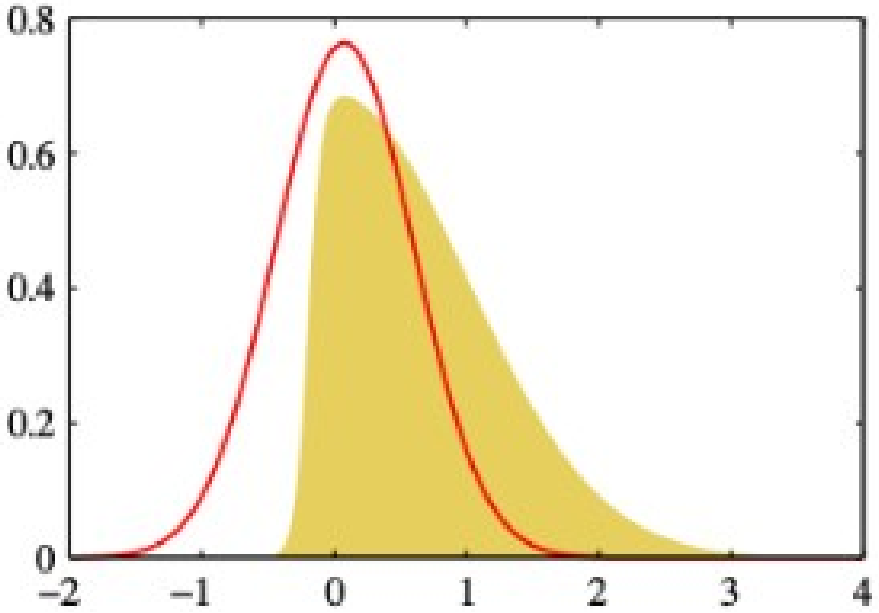
- The posterior is approximately Gaussian around the MAP estimate θ_{MAP}

$$p(\theta|\mathcal{D}) \approx \frac{|A|^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\theta - \theta_{MAP})^T A (\theta - \theta_{MAP}) \right)$$

Laplace Approximation

The posterior is approximately Gaussian around the MAP estimate θ_{MAP}

$$p(\theta|\mathcal{D}) \approx \frac{|A|^{1/2}}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}(\theta-\theta_{MAP})^T A(\theta-\theta_{MAP})\right)$$



Notice how bad a fit this is.
In Lecture 7 we will do better using variational methods.

Figure: Ruslan Salakhutdinov, BCS and CSAIL, MIT

Markov Chain Monte Carlo MCMC

- Idea is $P(X)$ is hard to sample but we can define a Markov chain that is easier.
- We have to define a transition probability:
  - $T(\boldsymbol{x}^n \rightarrow \boldsymbol{x}^{n+1}) = T(\boldsymbol{x}^{n+1} \mid \boldsymbol{x}^n)$
  - Also called a Kernel.

Markov Chain Monte Carlo MCMC

- Trick is to make the transition $T(\boldsymbol{x}^{n+1} \mid \boldsymbol{x}^{n})$ such that a sequence of samples converges to a sample from $P(X)$
- Must be stationary: $P(\boldsymbol{x}) = \Sigma_{x'}\, T(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$
  - This ensures that there is a solution
  - Still need to ensure that the solution is reached
  - And that it is unique.
- P is an 'Eigen Vector' of T
- Eigen value is λ=1 and the others are 1> λ >0,
- $\Rightarrow T^{n}Q \rightarrow P,$ (since all other components go like $\lambda^{n} \rightarrow 0$).
- Next biggest eigen value determines convergence speed.

Markov Chain Monte Carlo MCMC

- stationary: $P(\boldsymbol{x}) = \Sigma_{x'} T(\boldsymbol{x} \,|\, \boldsymbol{x}') P(\boldsymbol{x}')$
- Reducible MC: there are separate regions that one can become trapped in.
  - We do not want reducible MC.

Markov Chain Monte Carlo MCMC

- stationary: $P(\boldsymbol{x}) = \Sigma_{x'} T(\boldsymbol{x} \,|\, \boldsymbol{x}') P(\boldsymbol{x}')$
- Reducible MC: there are separate regions that one can become trapped in.
  - We do not want reducible MC.
- Periodic MC: one can become trapped in cycles.
  - We do not want periodic MC.

Markov Chain Monte Carlo MCMC

- stationary: $P(\boldsymbol{x}) = \Sigma_{x'} T(\boldsymbol{x} \mid \boldsymbol{x}') P(\boldsymbol{x}')$
- Reducible MC: there are separate regions that one can become trapped in.
  - We do not want reducible MC.
- Periodic MC: one can become trapped in cycles.
  - We do not want periodic MC.
- We want regular MC: $\exists$ k such that for $\forall$ x, x' the probability of $x \rightarrow x'$ in exactly k steps > 0.

  Finite state space + regular $\Rightarrow$ ergodic $\Rightarrow$ A stationary solution will be unique.

Ergodic

- Ergodic chain has:
  - Irreducibility: It is possible to get from any state to any other state with probability > 0 in a finite number of steps.
  - Aperiodicity: It is possible to return to any state at any time (after a finite delay), i.e. there exists an n such that for all x and all n' > n, the probability of returning to x in n' steps > 0.

Ergodic

- Ergodic chain has:
  - Irreducibility: It is possible to get from any state to any other state with probability > 0 in a finite number of steps.
  - Aperiodicity: It is possible to return to any state at any time, i.e. there exists an n such that for all x and all n' > n, the probability of returning to x in n' steps > 0.
- $T(\boldsymbol{x} \mid \boldsymbol{x}') > 0$ for all $\boldsymbol{x}, \boldsymbol{x}' \Rightarrow$ ergodic
- This guarantees that a stationary distribution is unique.
  - Still need to show that the one you want is stationary.

Markov Chain Monte Carlo MCMC

- stationary: $P(\boldsymbol{x}) = \Sigma_{x'} T(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$
- ergodic $\Rightarrow$ stationary solution is unique
- Multi-kernels: A kernel that consists of a series of kernels used one after the other.
  - Prove stationary for each single kernel
  - Prove ergodic for the composition.
- Can also use a random selector to select between a set of kernels.

Markov Chain Monte Carlo MCMC

- Stationary: $P(\boldsymbol{x}) = \Sigma_{x'} T(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$
- Reversible: Satisfies detailed balance
  - $T(\boldsymbol{x}' \mid \boldsymbol{x})P(\boldsymbol{x}) = T(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$
- Reversible $\Rightarrow$ $P(\boldsymbol{x}')$ is a stationary solution.
  - Unique if ergodic too.

MCMC Construction

- Metropolis-Hastings algorithm is a recipe to build such a Markov chain.
- Start with any transition kernel $Q(\boldsymbol{x} \mid \boldsymbol{x}')$
- Accept with probability:

$$A(\boldsymbol{x}', \boldsymbol{x}) = \min(1, P(\boldsymbol{x})Q(\boldsymbol{x}' \mid \boldsymbol{x}) / P(\boldsymbol{x}')Q(\boldsymbol{x} \mid \boldsymbol{x}'))$$

MCMC Construction

- Metropolis-Hastings algorithm is a recipe to build such a Markov chain.
- Start with any transition kernel $Q(\boldsymbol{x}|\, \boldsymbol{x}')$
- Accept with probability:

  $A(\boldsymbol{x}',\ \boldsymbol{x})=\min(1\ , P(\boldsymbol{x})Q(\boldsymbol{x}'|\, \boldsymbol{x})\ /\ P(\boldsymbol{x}')Q(\boldsymbol{x}\ |\ \boldsymbol{x}'))$

  Note: either $A(\boldsymbol{x}', \boldsymbol{x}) =1$ or $A(\boldsymbol{x},\ \boldsymbol{x}') =1$
- We do not need to compute the partition function Z.

MH Detailed Balance

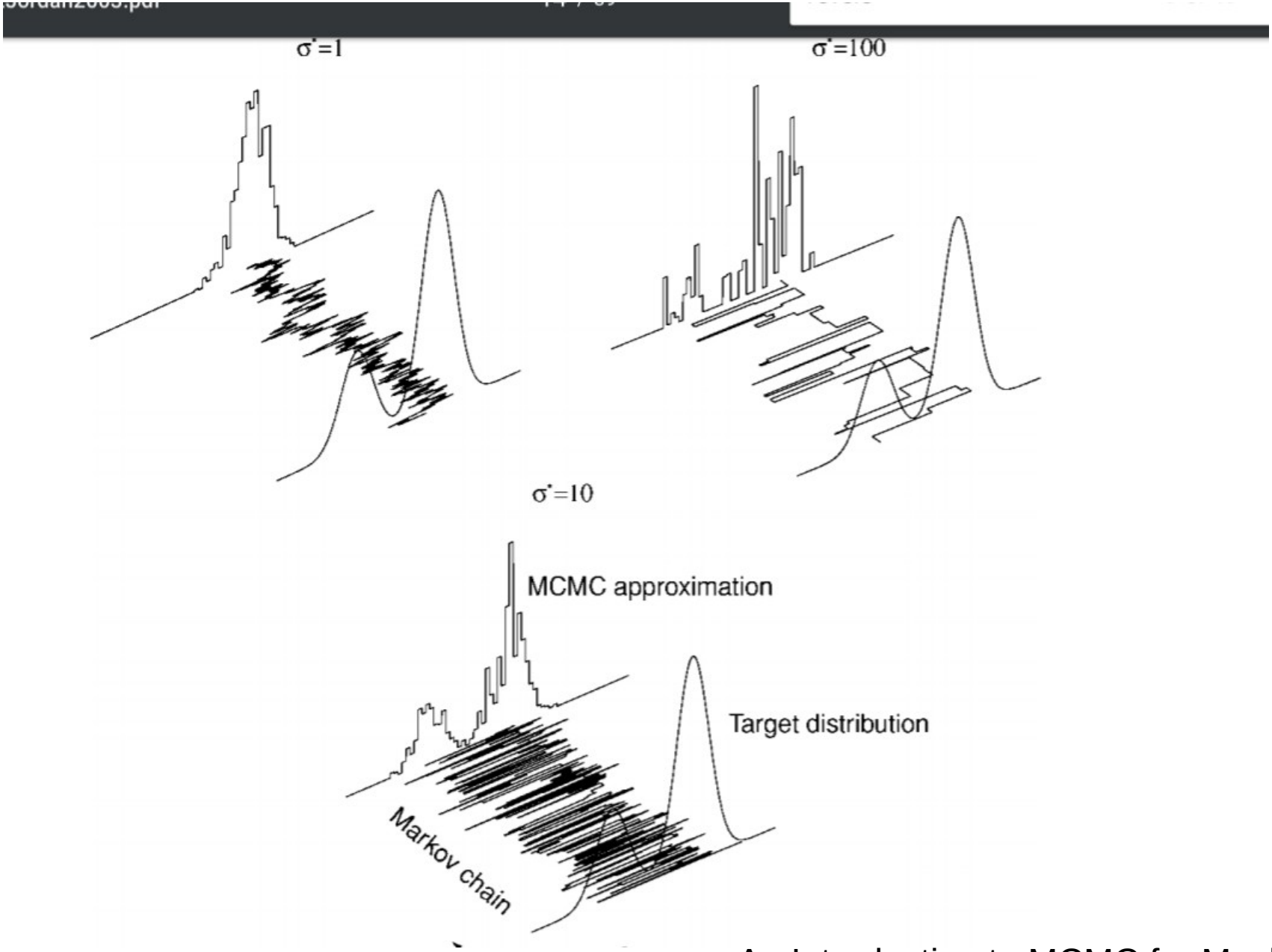
$A(\boldsymbol{x}', \boldsymbol{x}) = \min(1 , P(\boldsymbol{x})Q(\boldsymbol{x}' \mid \boldsymbol{x}) / P(\boldsymbol{x}')Q(\boldsymbol{x} \mid \boldsymbol{x}'))$

$T(\boldsymbol{x}' \mid \boldsymbol{x})P(\boldsymbol{x}) = T(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$ detail balance?

- $T(\boldsymbol{x} \mid \boldsymbol{x}') = A(\boldsymbol{x}', \boldsymbol{x})Q(\boldsymbol{x} \mid \boldsymbol{x}')$
- $A(\boldsymbol{x}, \boldsymbol{x}')Q(\boldsymbol{x}' \mid \boldsymbol{x})P(\boldsymbol{x}) = A(\boldsymbol{x}', \boldsymbol{x})Q(\boldsymbol{x} \mid \boldsymbol{x}')P(\boldsymbol{x}')$

- The side with A<1 has just the right expression to make it equal the other side (which has A=1).

MH with Gaussian Q's



An Introduction to MCMC for Machine Learning
CHRISTOPHE ANDRIEU

Gibbs Sampling

1. Sample x_i' from $p(x_i \mid x_{-i})$

2. New sample is: $\boldsymbol{x}' = (x_1, \ldots, x_i', \ldots, x_n)$

Note x_{-i} is all components of $\boldsymbol{x}$ except x_i

- Special case of MH where $A = 1$ always. (really?)
- Typically easy to sample as we only need to set the Markov blanket.
- Factor graphs and MRF are naturals for Gibbs.
- Sometimes 'blocks' of variables are sampled.

MCMC Mixing and Burn In Time

- We want good 'mixing', that is states should be able to move between all regions easily.
- We have to wait for a burn in time before we forget completely the arbitrary start state and can be really sampling from P.
- After that we can take many samples from P but they will not be independent.
  - Think about it!
- Burn in time is often polynomial in the number of dimensions, an escape from the curse of dimensionality.

Gibbs with Simulated Annealing

- We can replace P with

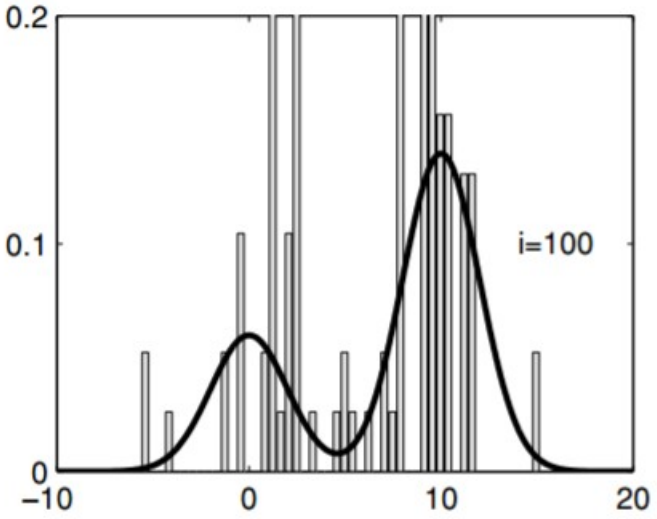
  $P'(T)=P^{(1/T)} = exp(T^{-1} ln P)/Z;$ 'T' here is now 'Temperature'.

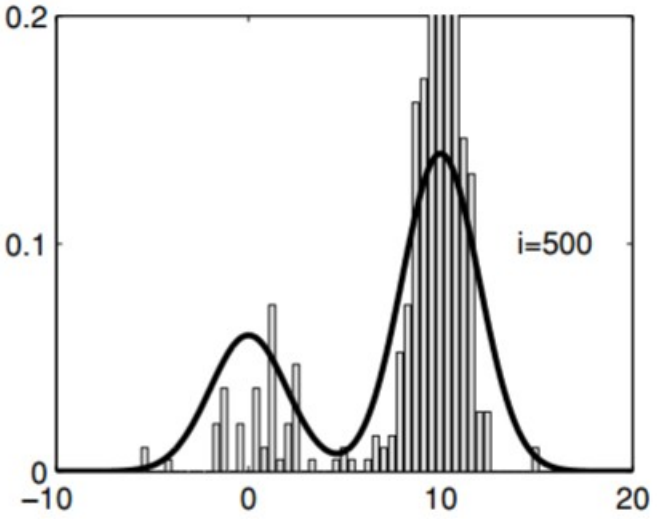
- For large T this will flatten out the peaks and cause more mixing.
- We then lower T gradually to 1 which then

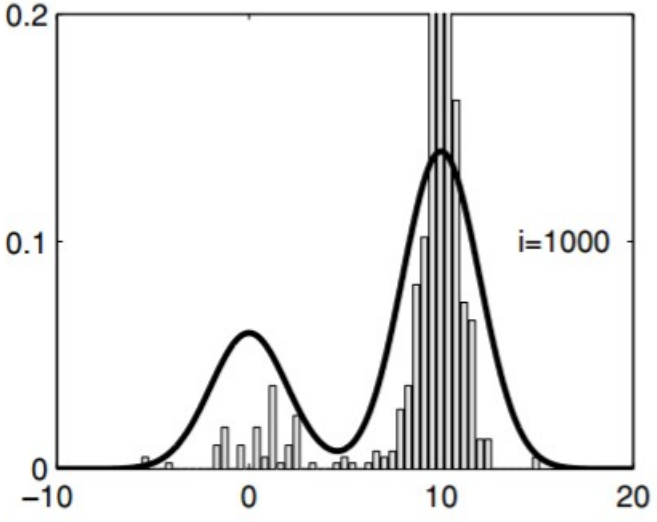
  has $P'(1)= P$

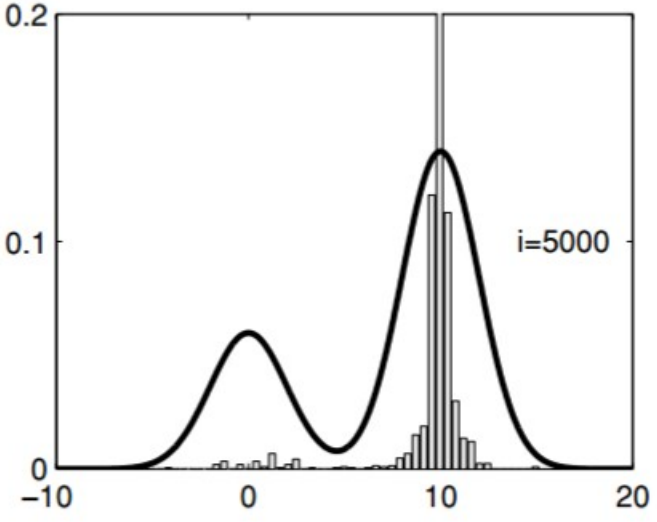
- OR go all the way to $T=0$ for a global max.
- Works on hard problems but is slow.

Simulated Annealing, $T_i \rightarrow 0$









An Introduction to MCMC for Machine Learning
CHRISTOPHE ANDRIEU, et al.

Collapsed Importance Sampling

$$p(\boldsymbol{x}) = p(\boldsymbol{x}_p, \boldsymbol{x}_d) = \Sigma_m \; w_m \, p(\boldsymbol{x}_d \mid \boldsymbol{x}_p^m)$$

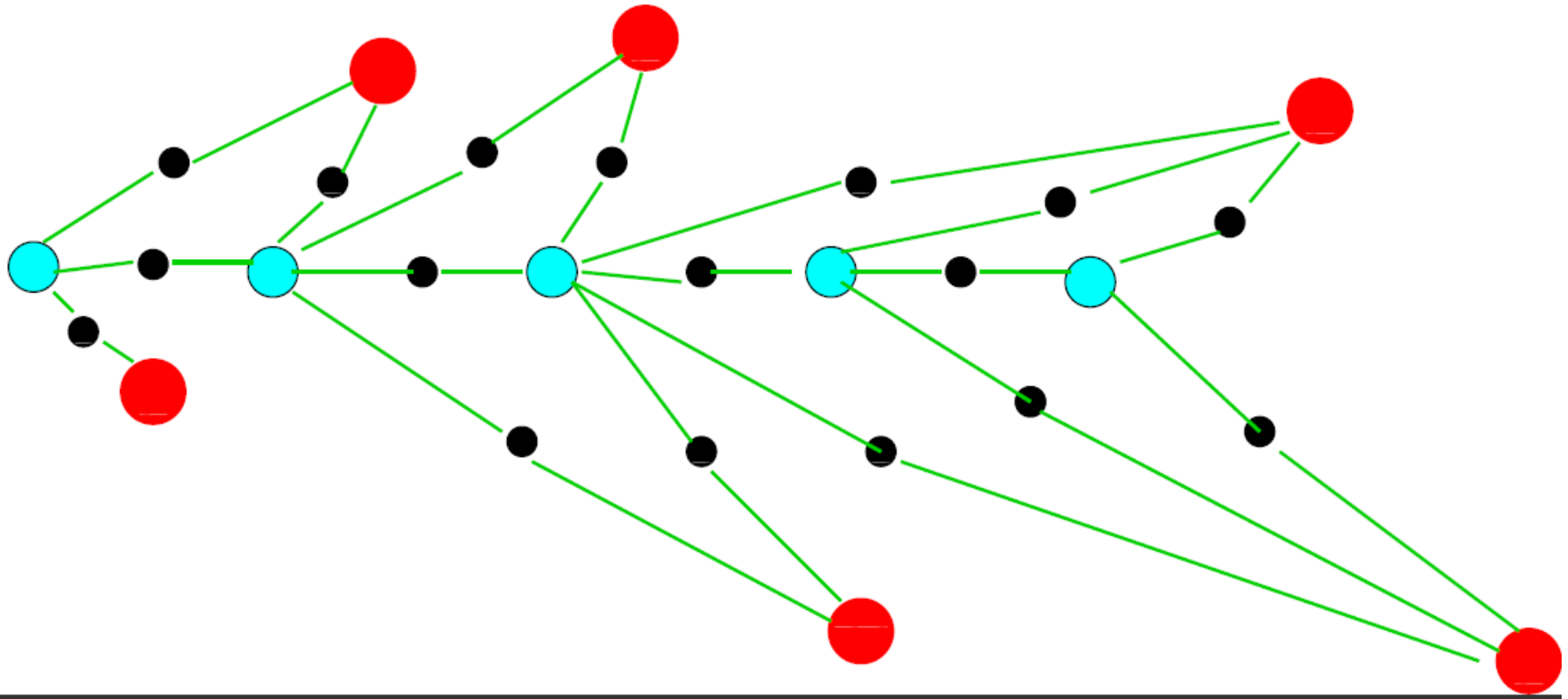
- Variables are split between ones that are estimated via samples (particles $\boldsymbol{x}_p^m$) and ones that have a parametric representation, $\boldsymbol{x}_d$.
- Weights are as usual the target / proposal.
- Can lead to huge simplifications if $p(\boldsymbol{x}_d \mid \boldsymbol{x}_p^m)$ factors nicely.
- Rao-Blackwellized Particle filters are another name for this.

FASTSLAM and its Factor Graph

= Features = Robot Poses

= Measurements or Energy Nodes

= Edge Showing dependancy



Collapsed Importance Sampling

$$p(\boldsymbol{x}) = p(\boldsymbol{x}_{\boldsymbol{p}}, \boldsymbol{x}_{\boldsymbol{d}}) = \Sigma_m \; w_m \, p(\boldsymbol{x}_{\boldsymbol{d}} \mid \boldsymbol{x}_{\boldsymbol{p}}^{\boldsymbol{m}})$$

- Can lead to huge simplifications if $p(\boldsymbol{x}_{\boldsymbol{d}} \mid \boldsymbol{x}_{\boldsymbol{p}}^{\boldsymbol{m}})$ factors nicely.
- FASTSLAM for example we represent the feature map by a giant Gaussian and the robot path as particles
- Each feature is independent of the others if the path is known.