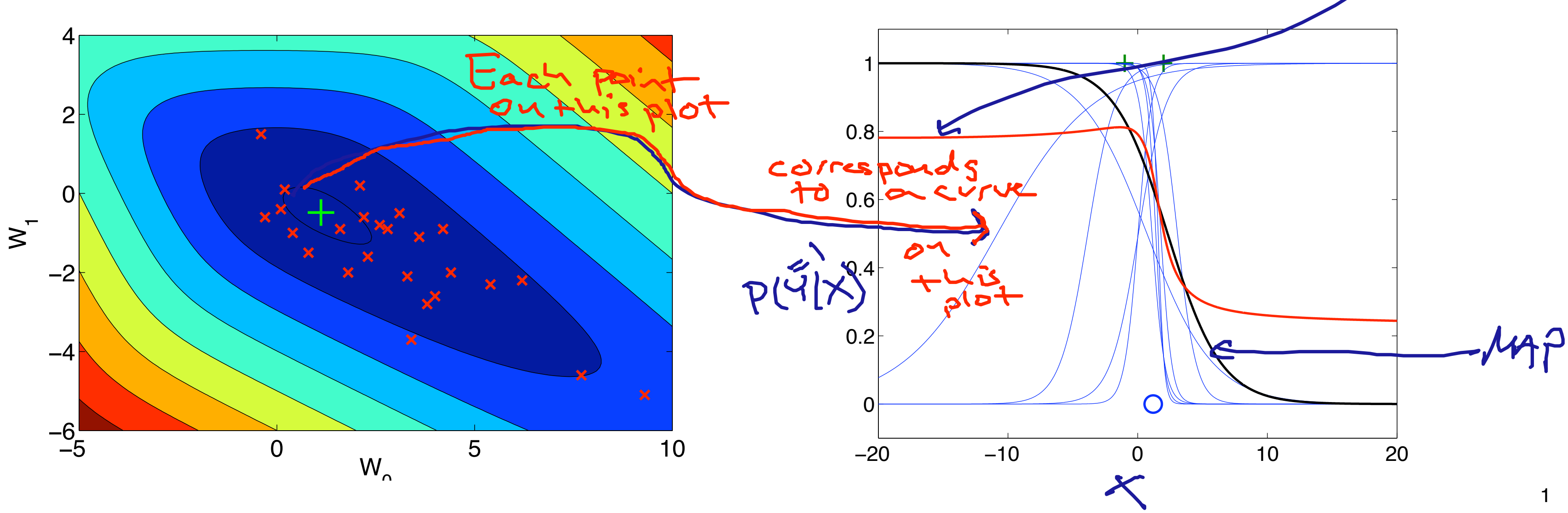
Review

- Multiclass logistic regression
- Priors, conditional MAP logistic regression
- Bayesian logistic regression
  - MAP is not always typical of posterior
  - posterior predictive can avoid overfitting



Review

- Finding posterior predictive distribution often requires numerical integration
  - uniform sampling
  - importance sampling
  - parallel importance sampling
- These are all ***Monte-Carlo algorithms***
  - another well-known MC algorithm coming up

we're the house

Application: SLAM

Eliazar and Parr, IJCAI-03

Parallel IS

set $\hat{w}_i = Z\,P(x_i)/Q(x_i)$

$$\bar{w} = \frac{1}{N}\sum_{i=1}^{N} \hat{w}_i$$

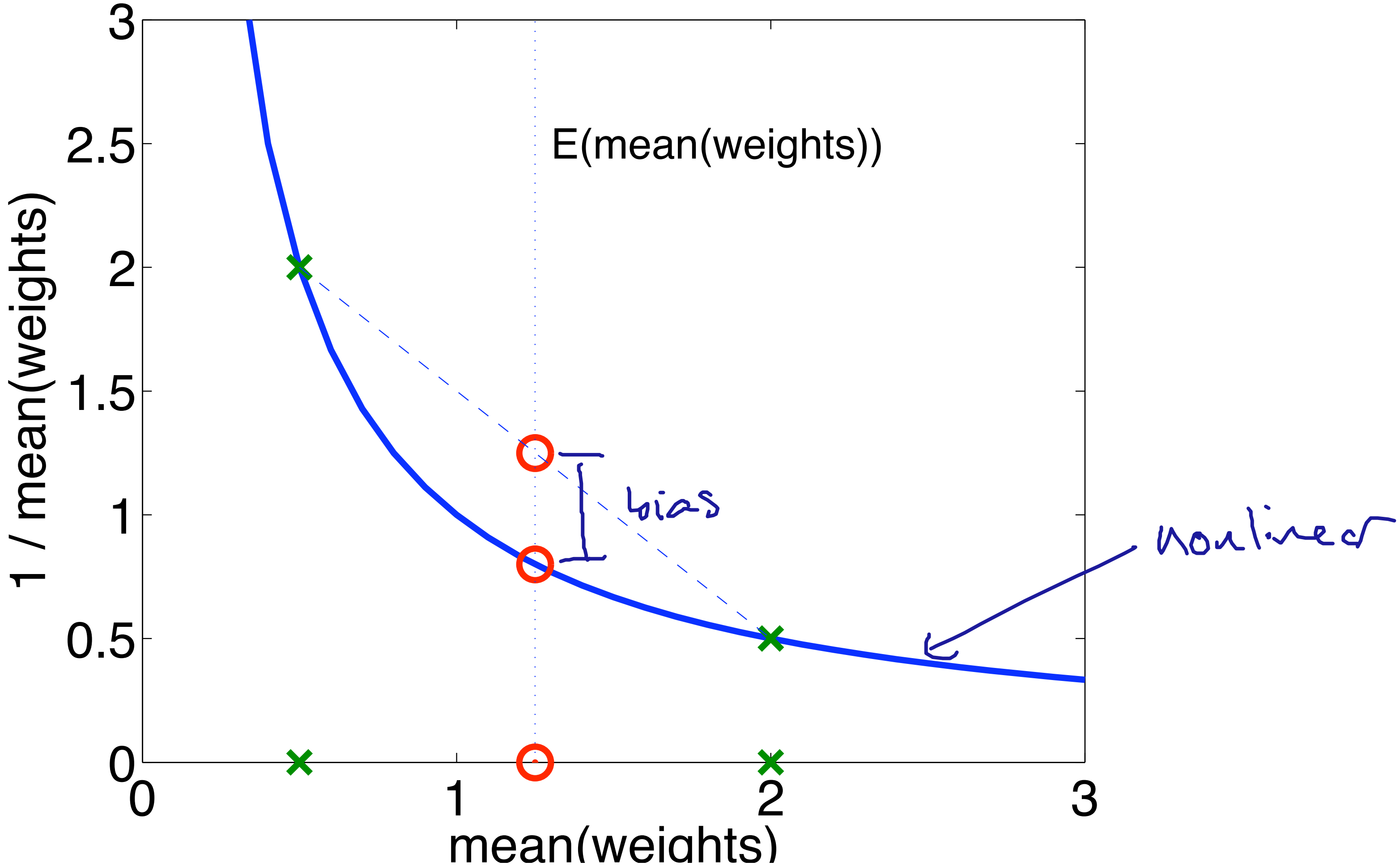
$$E(\hat{w}_i) = \int \cancel{Q(x)}\, Z\,P(x)/\cancel{Q(x)}\,dx$$

$$= Z\int P(x)\,dx = Z$$

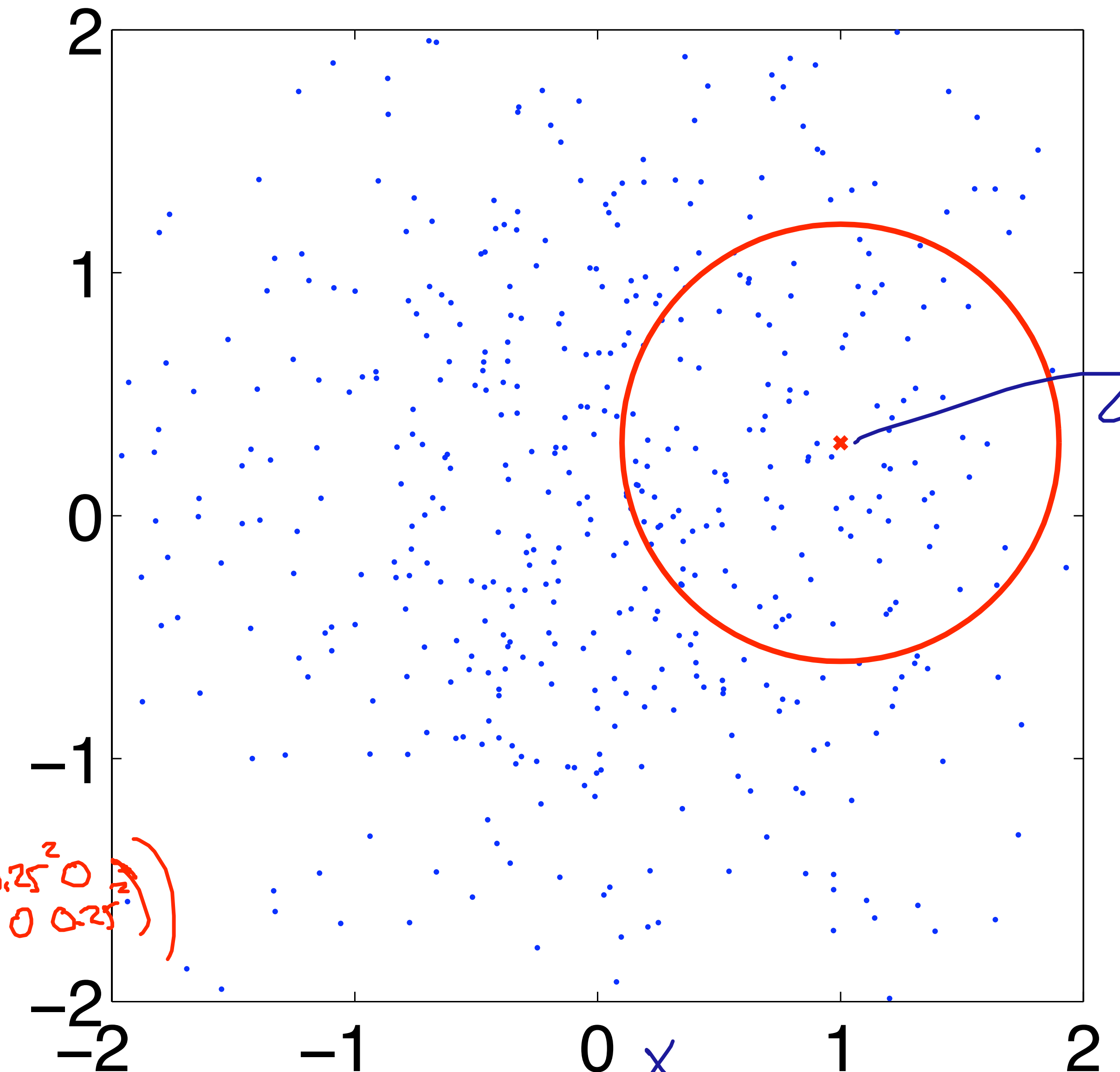
$$E(\bar{w}) = Z \quad \text{(lower variance)}$$

$$\mathbb{E}_P(g(x)) \approx \hat{G} = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\frac{\hat{w}_i}{\bar{w}}}_{\text{normalized importance wts}} g(x_i)$$

Parallel IS is biased



$$E(\bar{W}) = Z, \text{ but } E(1/\bar{W}) \neq 1/Z \text{ in general}$$



$N\left(\binom{X}{Y} \,\middle|\, \binom{0}{0}, \begin{pmatrix} 0.25^2 & 0 \\ 0 & 0.25^2 \end{pmatrix}\right)$

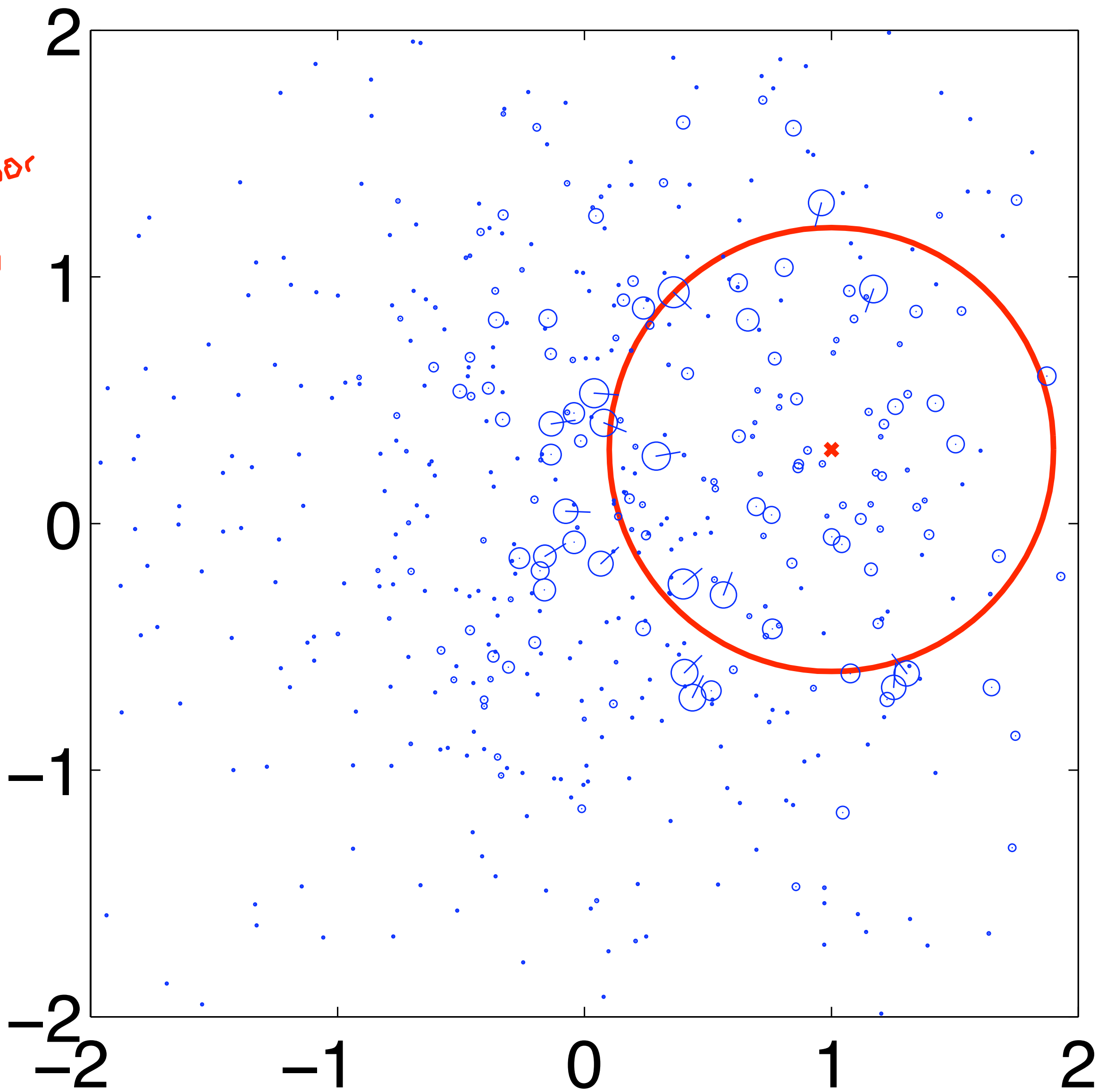
observation model

importance dist'n = prior

$$Q : (X, Y) \sim \cancel{N(1,1)} \qquad \theta \sim U(-\pi, \pi)$$

$$f(x, y, \theta) = \underbrace{Q(x, y, \theta)}_{\text{prior}} P(\underline{o = 0.8} \mid x, y, \theta)/Z$$

sensor model:
range-only sensor
w/ error N(0,1)

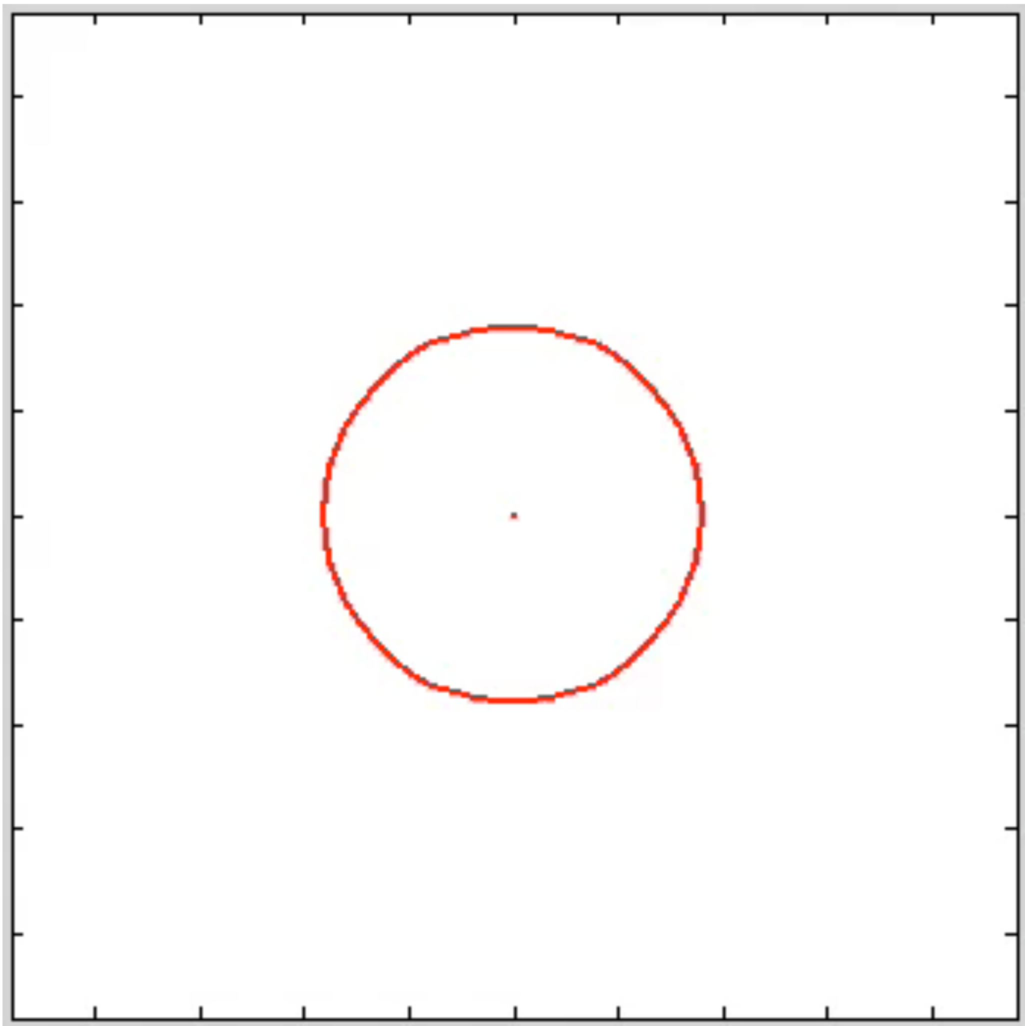


Posterior $E(X, Y, \theta) = (0.496, 0.350, 0.084)$

SLAM revisited

- Uses a recursive version of parallel importance sampling: ***particle filter***
  - each sample (particle) = trajectory over time
  - sampling extends trajectory by one step
  - recursively update importance weights and renormalize
  - resampling trick to avoid keeping lots of particles with low weights

Particle filter example



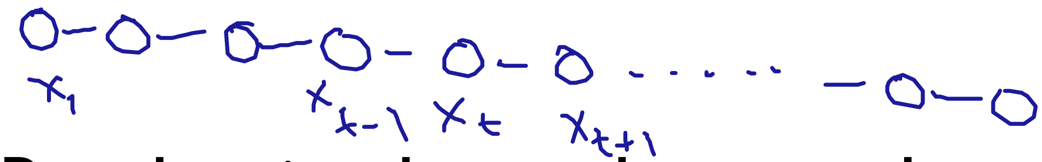
Monte-Carlo revisited

- Recall: wanted

$$E_P(g(X)) = \int g(x)P(x)dx = \int f(x)dx$$

- Would like to search for areas of high P(x)
- But searching could bias our estimates

Markov-Chain Monte Carlo



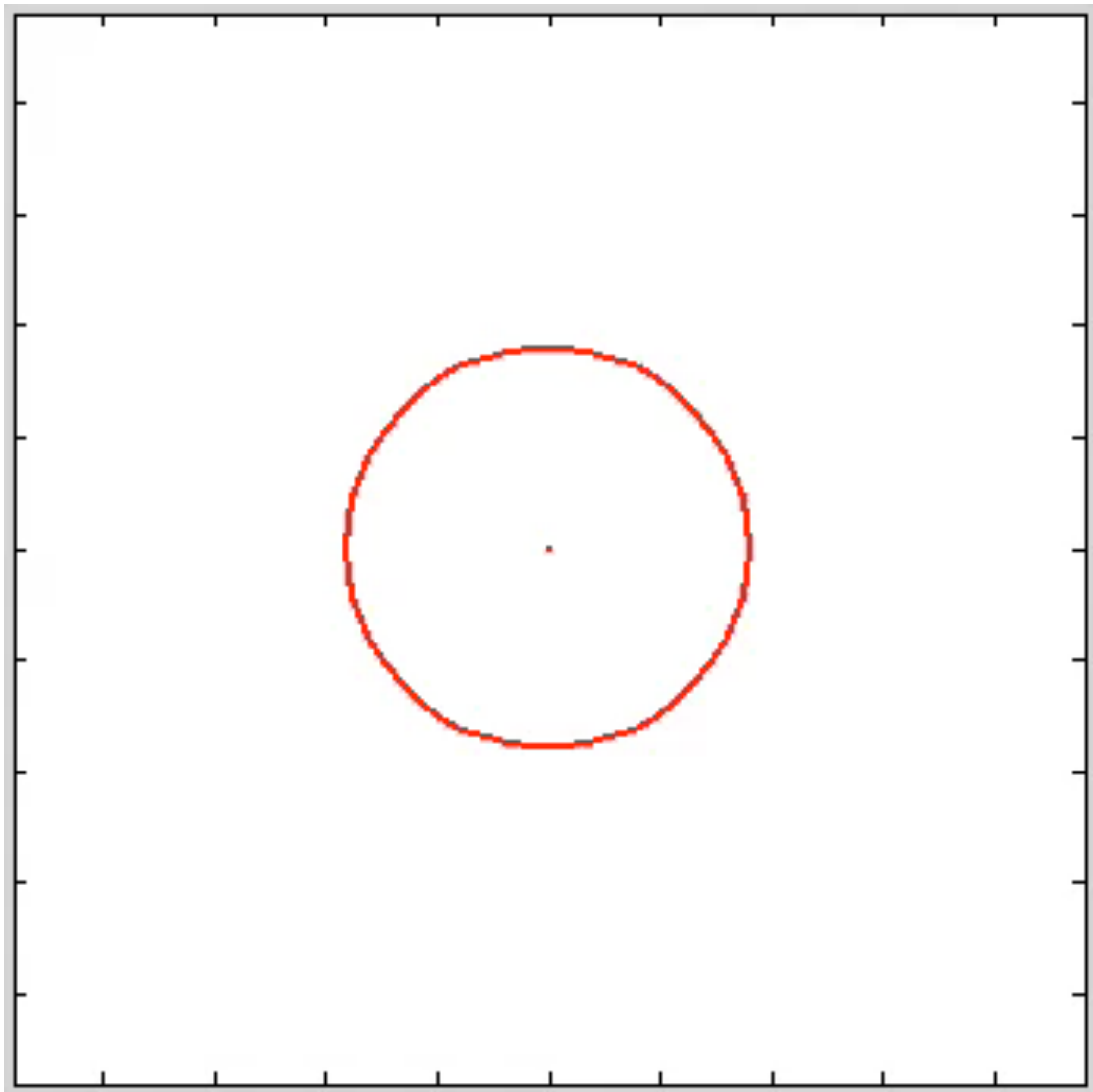
- Randomized search procedure
- Produces sequence of RVs $X_1, X_2, \dots$
  - Markov chain: satisfies Markov property

$$(x_1, x_2, \dots x_{t-1}) \perp (x_{t+1}, \dots) \mid X_t$$

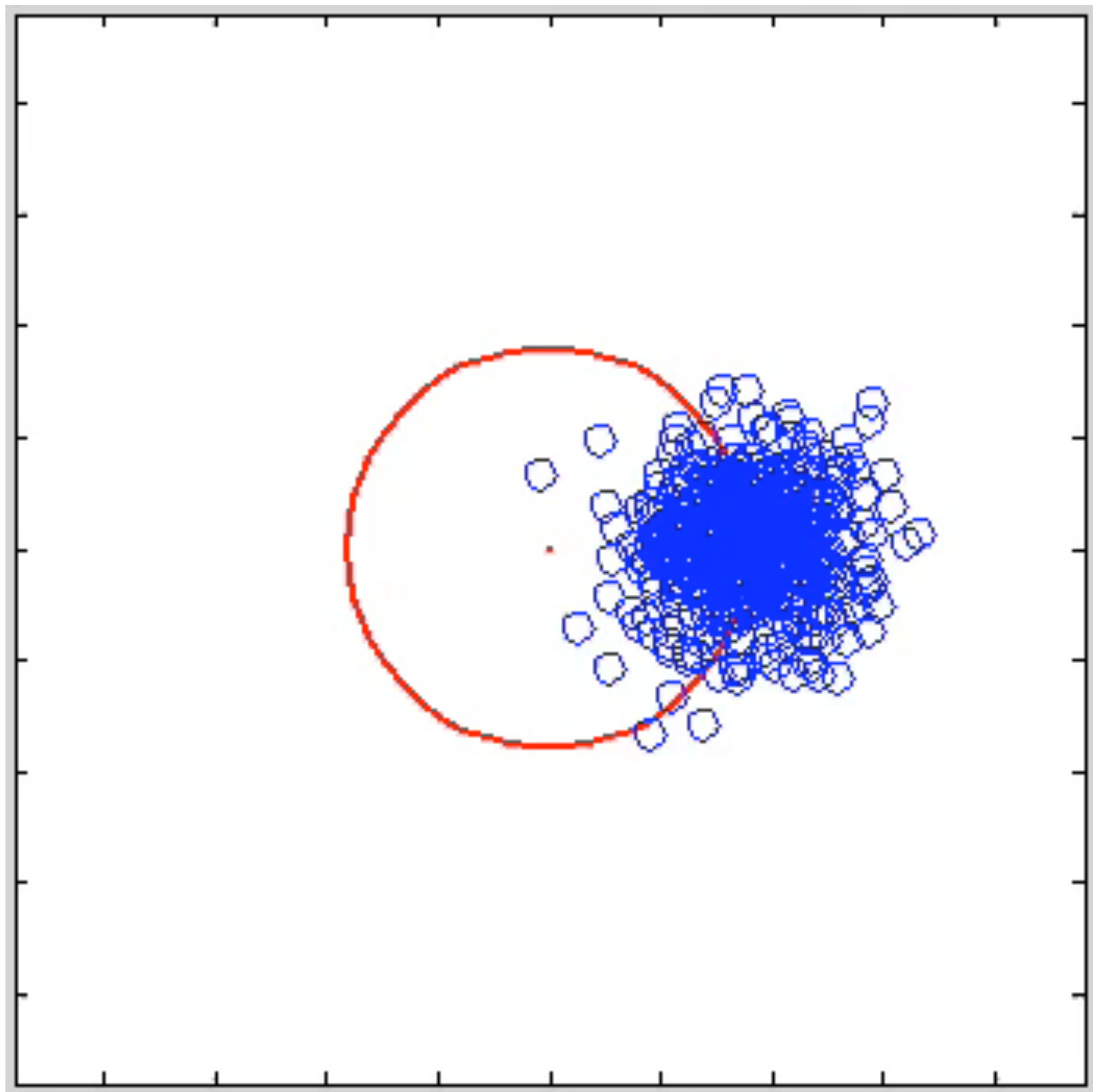
- If $P(X_t)$ small, $P(X_{t+1})$ tends to be larger
- As $t \to \infty$, $X_t \sim P(X)$ "stationary" or limiting dist'n is P
- As $\Delta \to \infty$, $X_{t+\Delta} \perp X_t$

forgetting

Markov chain



Stationary distribution



Markov-Chain Monte Carlo

- As $t \to \infty$, $X_i \sim P(X)$; as $\Delta \to \infty$, $X_{t+\Delta} \perp X_t$

  burn-in time   mixing time

- For big enough t and Δ, an approximately i.i.d. sample from P(X) is
  - $\{ X_t, X_{t+\Delta}, X_{t+2\Delta}, X_{t+3\Delta}, \ldots \}$

  better if g expensive

- Can use i.i.d. sample to estimate $E_P(g(X))$

$$\hat{G} = \frac{1}{N} \sum_{i=1}^{N} g(X_{t+(i-1)\Delta}) \qquad \hat{G}' = \frac{1}{N} \sum_{i=1}^{N} g(X_{t+1+(i-1)\Delta})$$

- Actually, don't need independence:

$$\hat{G} = \frac{1}{\Delta N} \sum_{i=1}^{\Delta N} g(X_{t+i-1})$$

better if g cheap

Metropolis-Hastings

or $ZP(X)$

i.e., don't need to know normaliz. constant

- Way to design chain w/ stationary dist'n P(X)
- Basic strategy: start from arbitrary X
- Repeatedly tweak X to get X'
  - If P(X') ≥ P(X), move to X'
  - If P(X') ≪ P(X), stay at X
  - In intermediate cases, randomize

key to getting stationary dist'n P

Proposal distribution

- Left open: what does "tweak" mean?
- Parameter of MH: Q(X' | X)

  one-step proposal dist'n

- Good proposals explore quickly, but remain in regions of high P(X)
- Optimal proposal? $Q(x' \mid x) = P(x')$

Simplest proposal

- ***Random walk*** MH: → for continuous X
  - $Q(X' \mid X) =$ $\mathcal{N}(x' \mid x, \sigma^2 I)$
  - big σ: move quickly
  - small σ: remain in regions of high p
- Not usually a great proposal, but sometimes the best we have

MH algorithm

must have

- Initialize X_1 arbitrarily → $P(X_1) > 0$
- For t = 1, 2, …:
  - Sample $X' \sim Q(X' \mid X_t)$ → current iterate
  - Compute p = $\frac{P(X')}{P(X_t)} \frac{Q(X_t \mid X')}{Q(X' \mid X_t)}$ acceptance prob.
  - With probability min(1, p), set $X_{t+1} := X'$ accept
    - else $X_{t+1} := X_t$ reject
- Note: sequence $X_1, X_2, \ldots$ will usually contain duplicates

If $Q(X' \mid X_t) = P(X')$

then mix time = 1

Acceptance rate

- Want **acceptance rate** (avg of min(1,p)) to be large, so we don't get big runs of same X
- Want Q(X' | X) to move long distances (to explore quickly)
- Tension between long moves, acceptance rate:

Random walk MH revisited

- Suppose we always accepted. Then:

$$x_t \sim x_0 + \sum_{i=1}^{t} N(0, \sigma^2 I)$$

$$x_t \sim N(x_0, t\sigma^2 I)$$

~~stdev(x_t) of one component~~ $= \sqrt{t}\,\sigma$

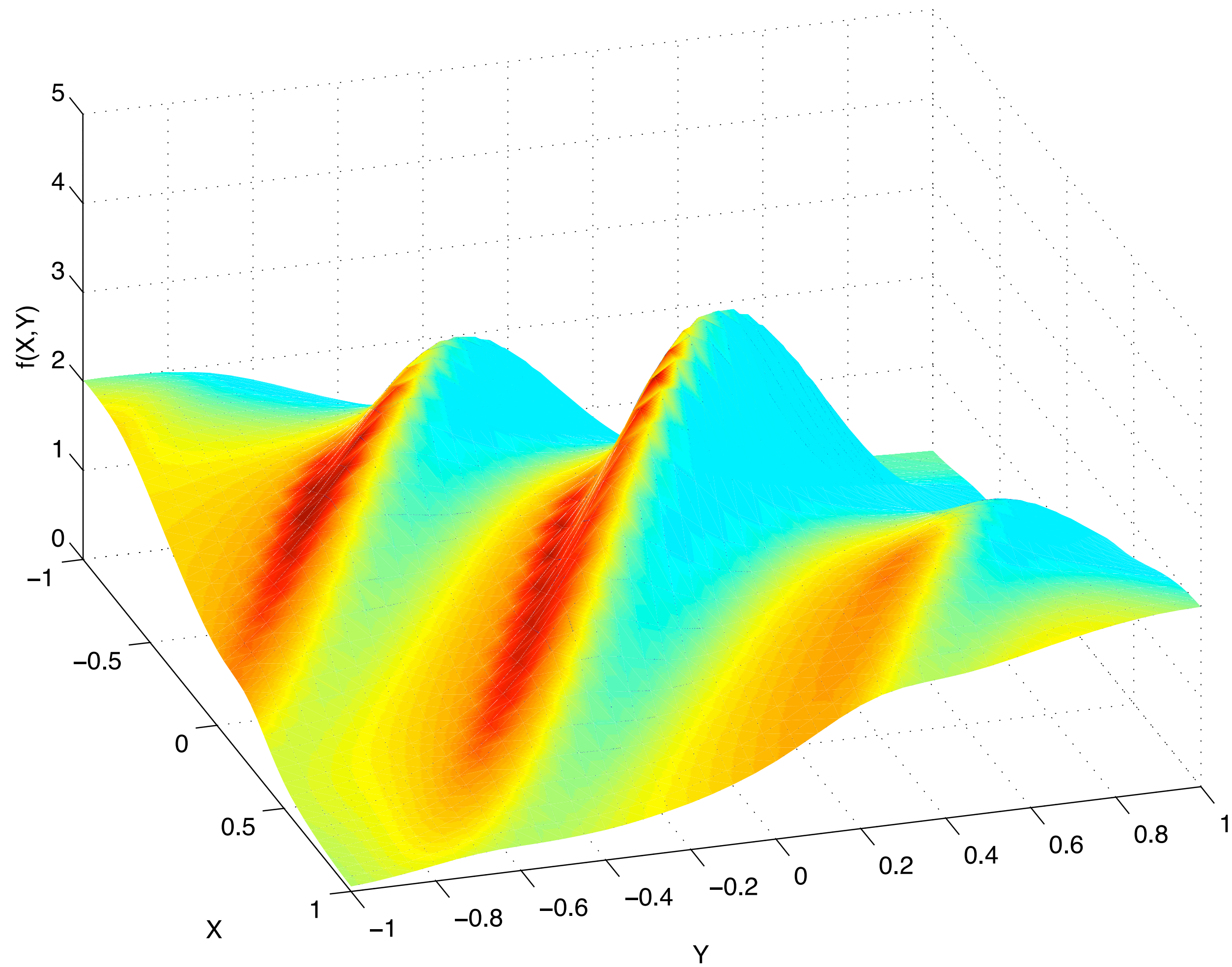
standard deviation of (any) one component of x_t
$= \sqrt{t}\,\sigma$

- Variance can only be smaller if we reject

Mixing rate, mixing time

- If we pick a good proposal, we will move rapidly around domain of P(X)
- After a short time, won't be able to tell where we started
- This is short **mixing time** = # steps until we can't tell which starting point we used
- **Mixing rate** = 1 / (mixing time)

MH example



Random walk $N(x_t, 0.25 I)$

MH example