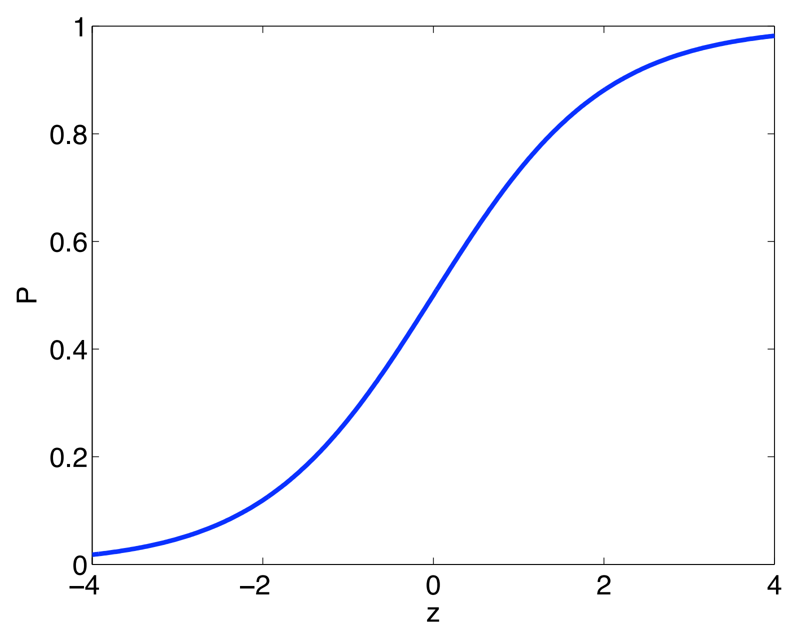
Review

- Linear separability (and use of features)
- Class probabilities for linear discriminants
  - sigmoid (logistic) function
- Applications: USPS, fMRI



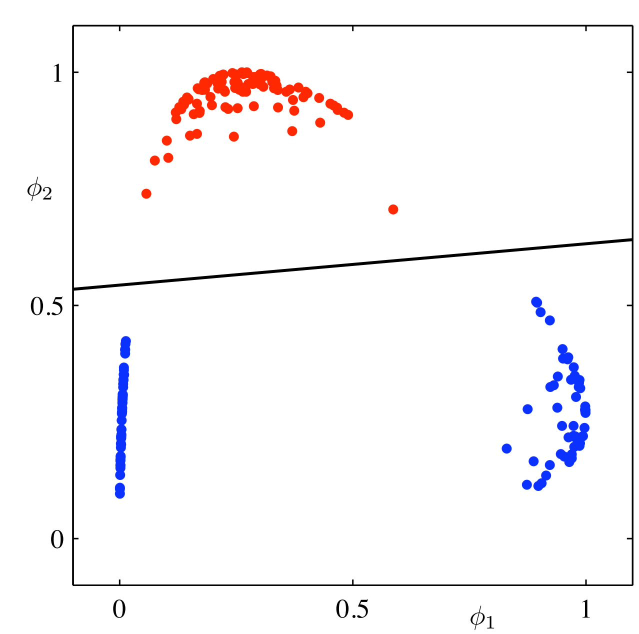
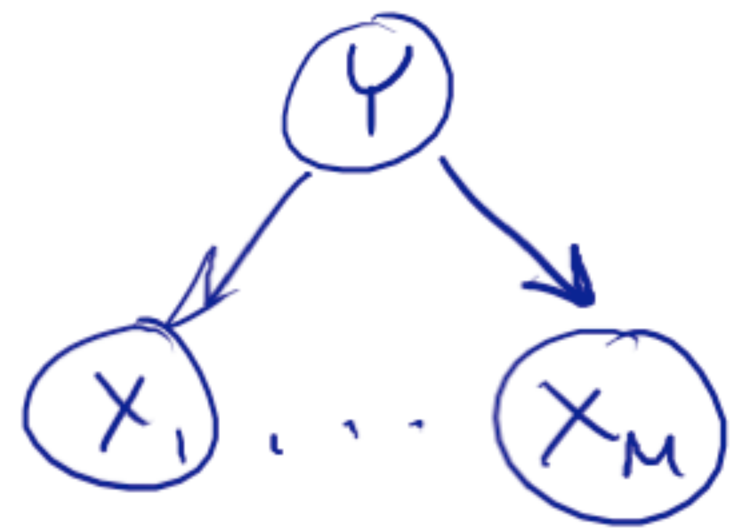
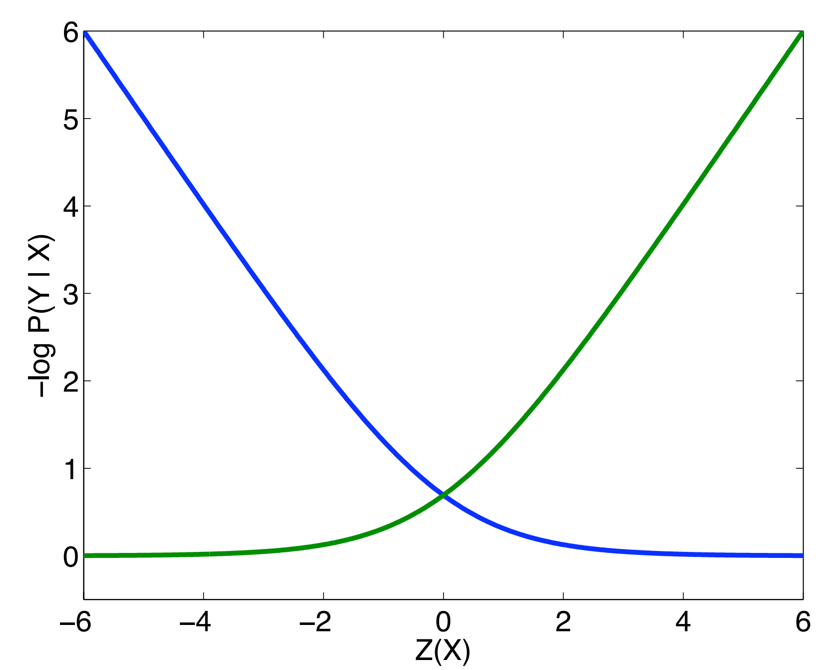


figure from book

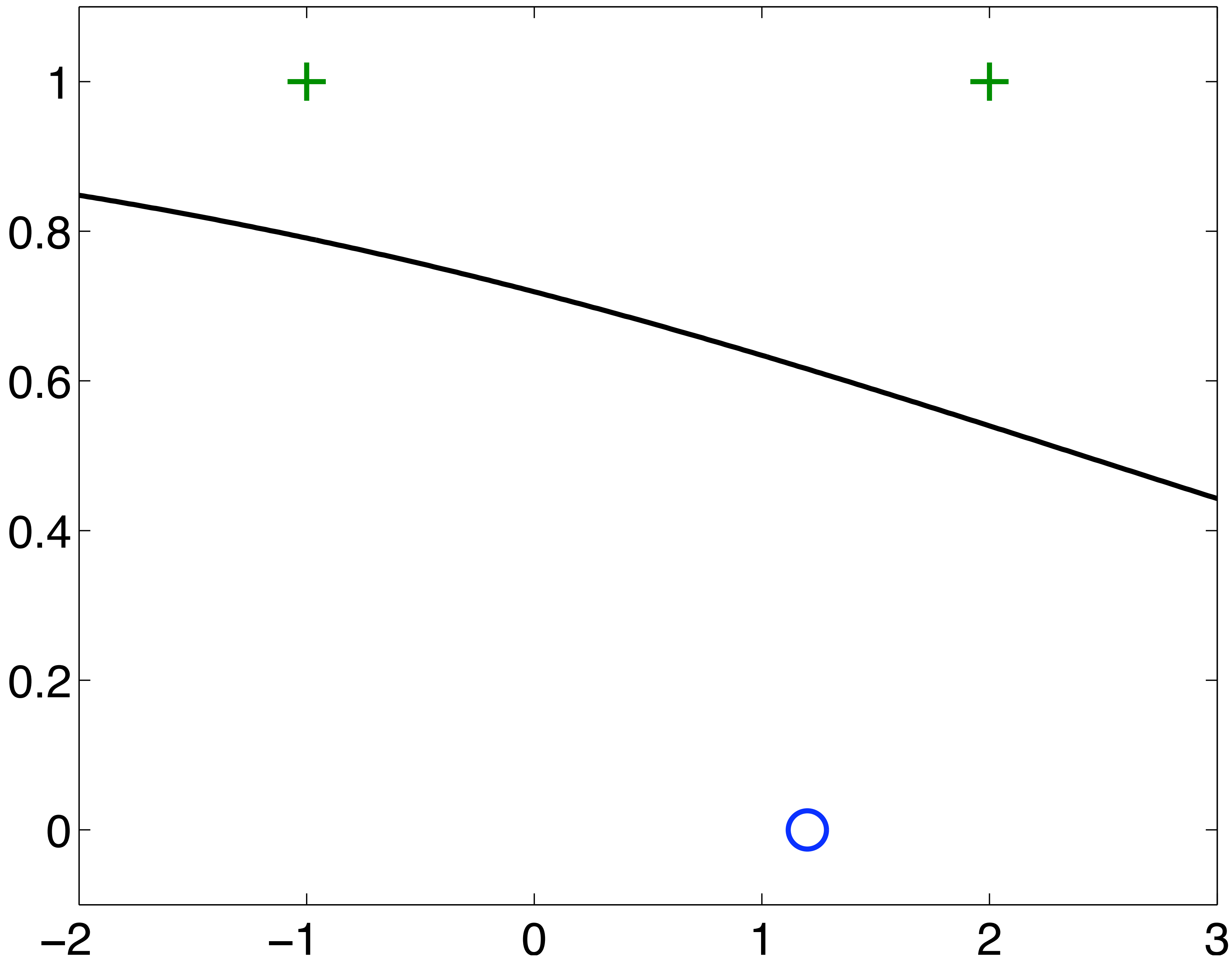
Review

- Generative vs. discriminative
  - maximum conditional likelihood
- Logistic regression
- Weight space
  - each example adds a penalty to all weight vectors that misclassify it
  - penalty is approximately piecewise linear

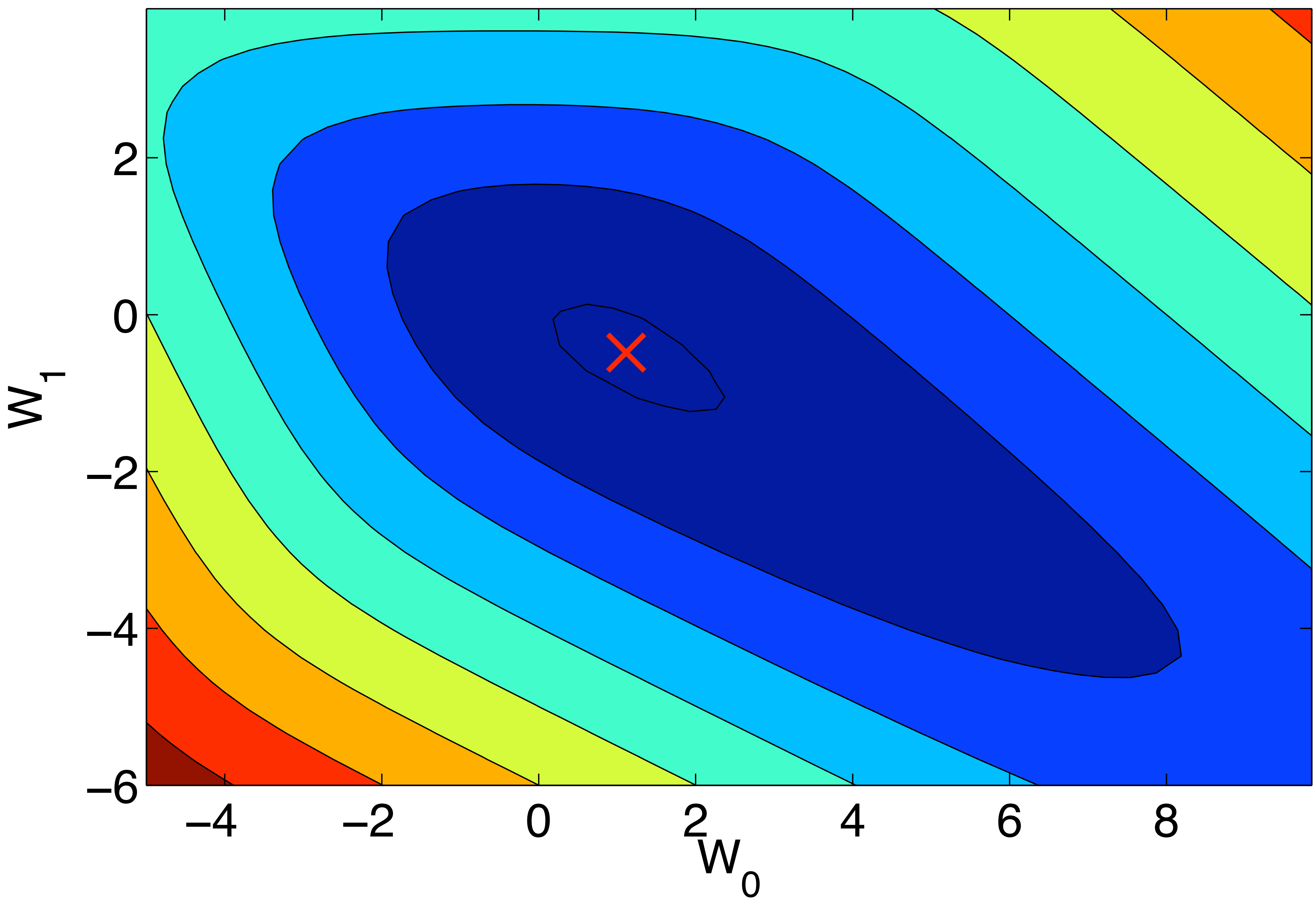




Example



$-\log(P(Y_{1..3} \mid X_{1..3}, W))$



Generalization: multiple classes

- One weight vector per class: $Y \in \{1,2,\ldots,C\}$
  - $P(Y=k) =$
  - $Z_k =$
- In 2-class case:

Multiclass example

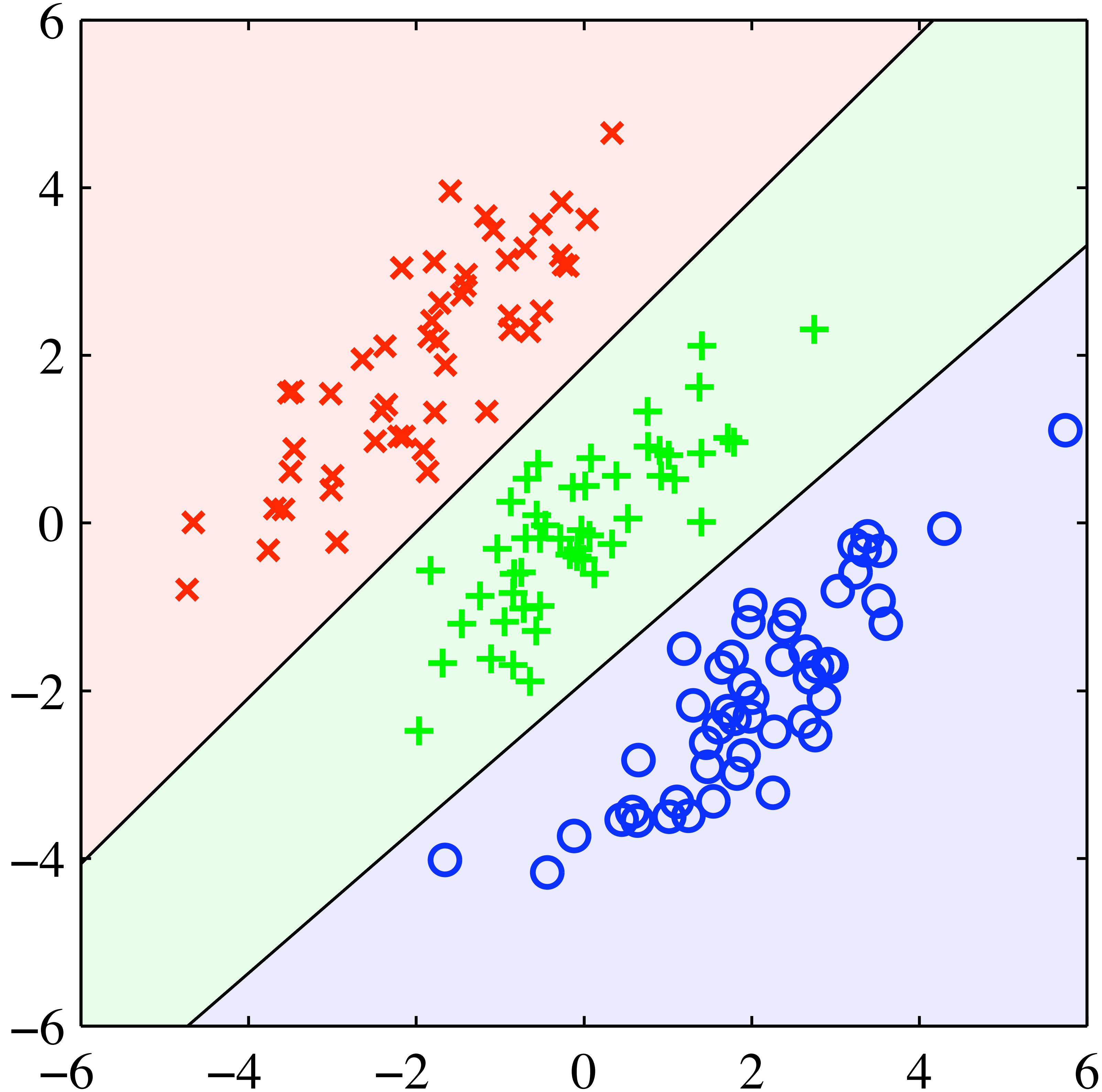


figure from book

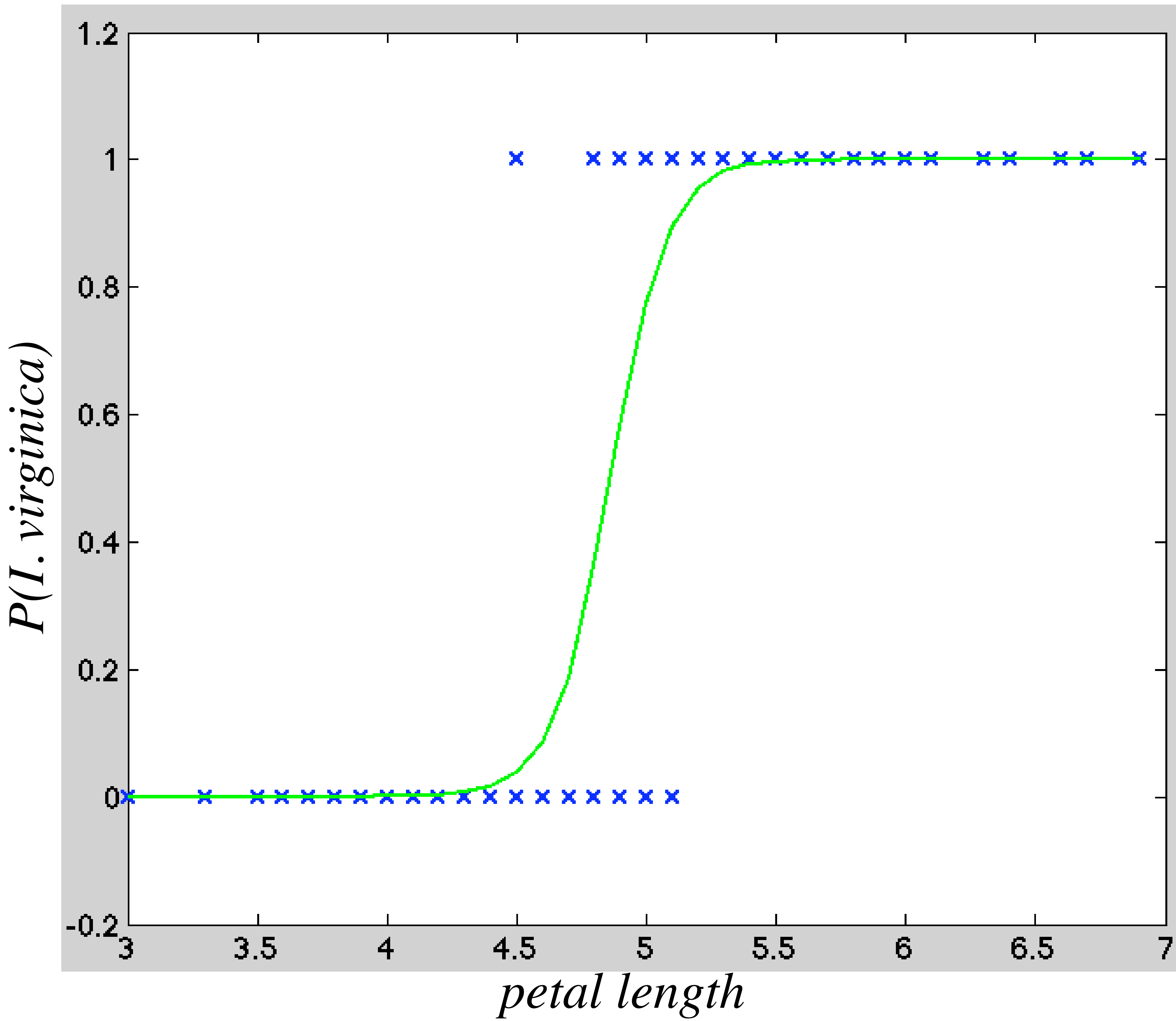
Priors and conditional MAP

- $P(Y \mid X, W) =$
  - $Z =$
- As in linear regression, can put prior on W
  - common priors: L_2 (ridge), L_1 (sparsity)
- $\max_w P(W{=}w \mid X, Y)$

Software

- Logistic regression software is easily available: most stats packages provide it
  - e.g., `glm` function in R
  - or, http://www.cs.cmu.edu/~ggordon/IRLS-example/
- Most common algorithm: Newton's method on log-likelihood (or L_2-penalized version)
  - called "iteratively reweighted least squares"
  - for L_1, slightly harder (less software available)

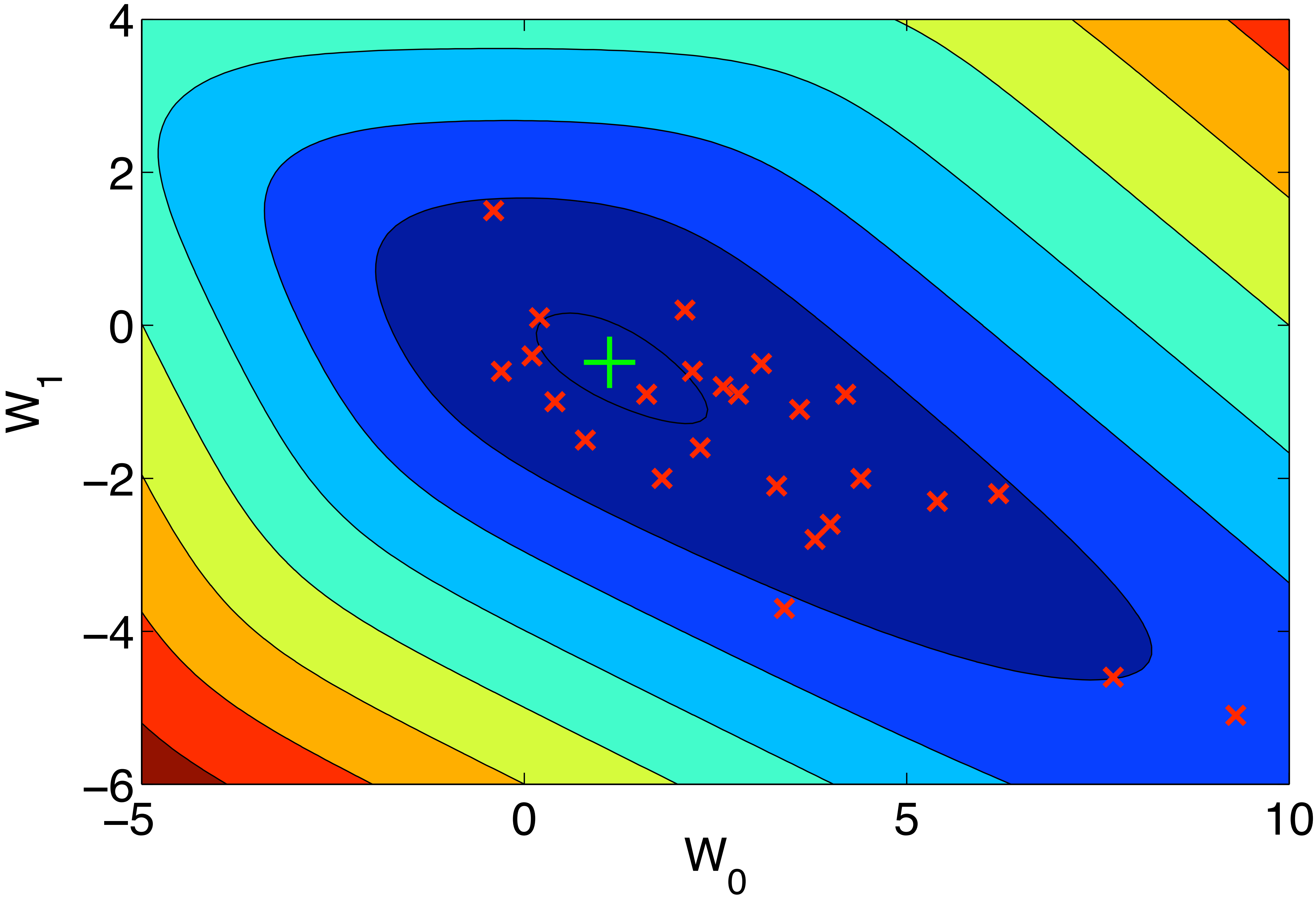
Historical application: Fisher iris data



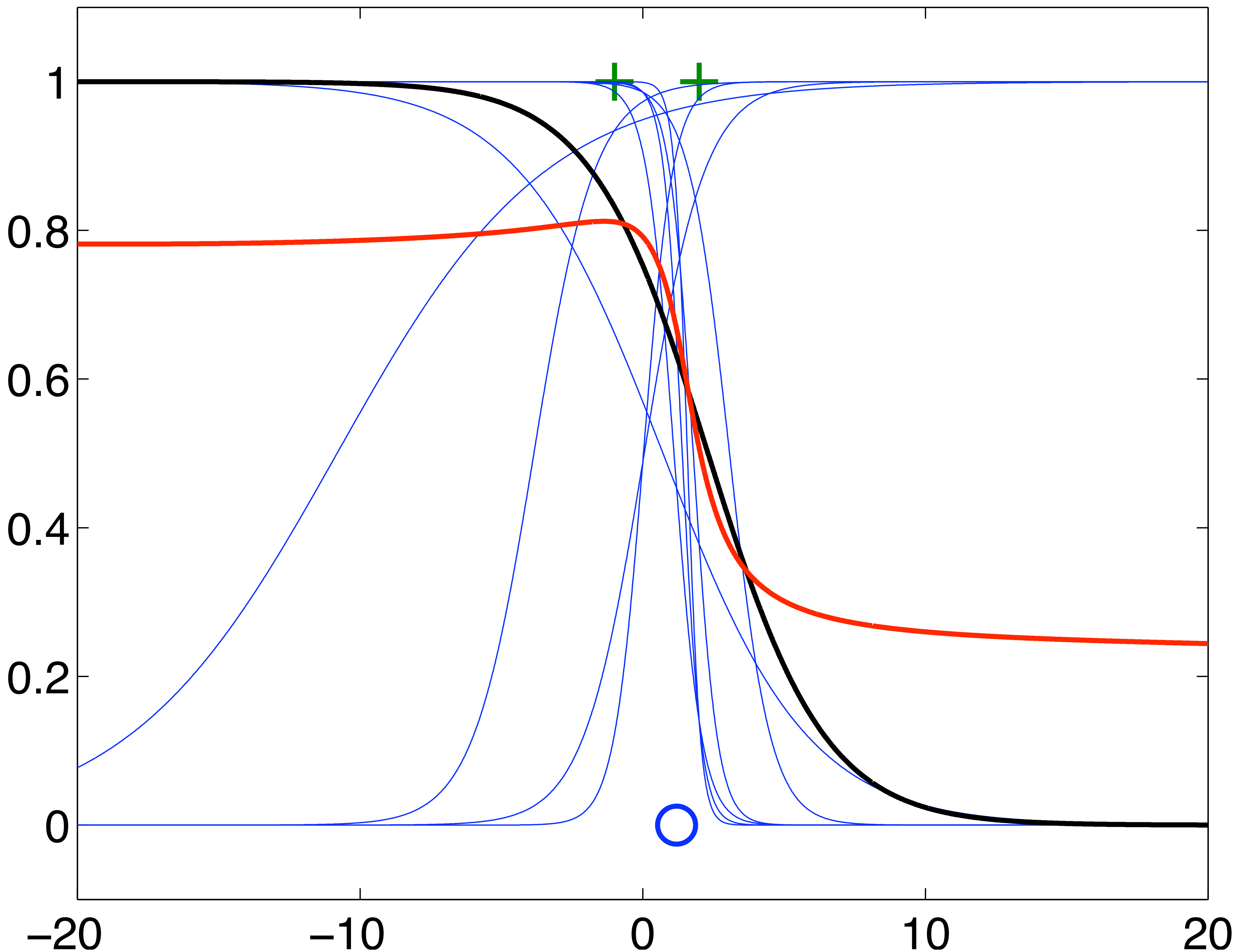
Bayesian regression

- In linear and logistic regression, we've looked at
  - conditional MLE: $\max_w P(Y \mid X, w)$
  - conditional MAP: $\max_w P(W{=}w \mid X, Y)$
- But of course, a true Bayesian would turn up nose at both
  - why?

Sample from posterior



Predictive distribution



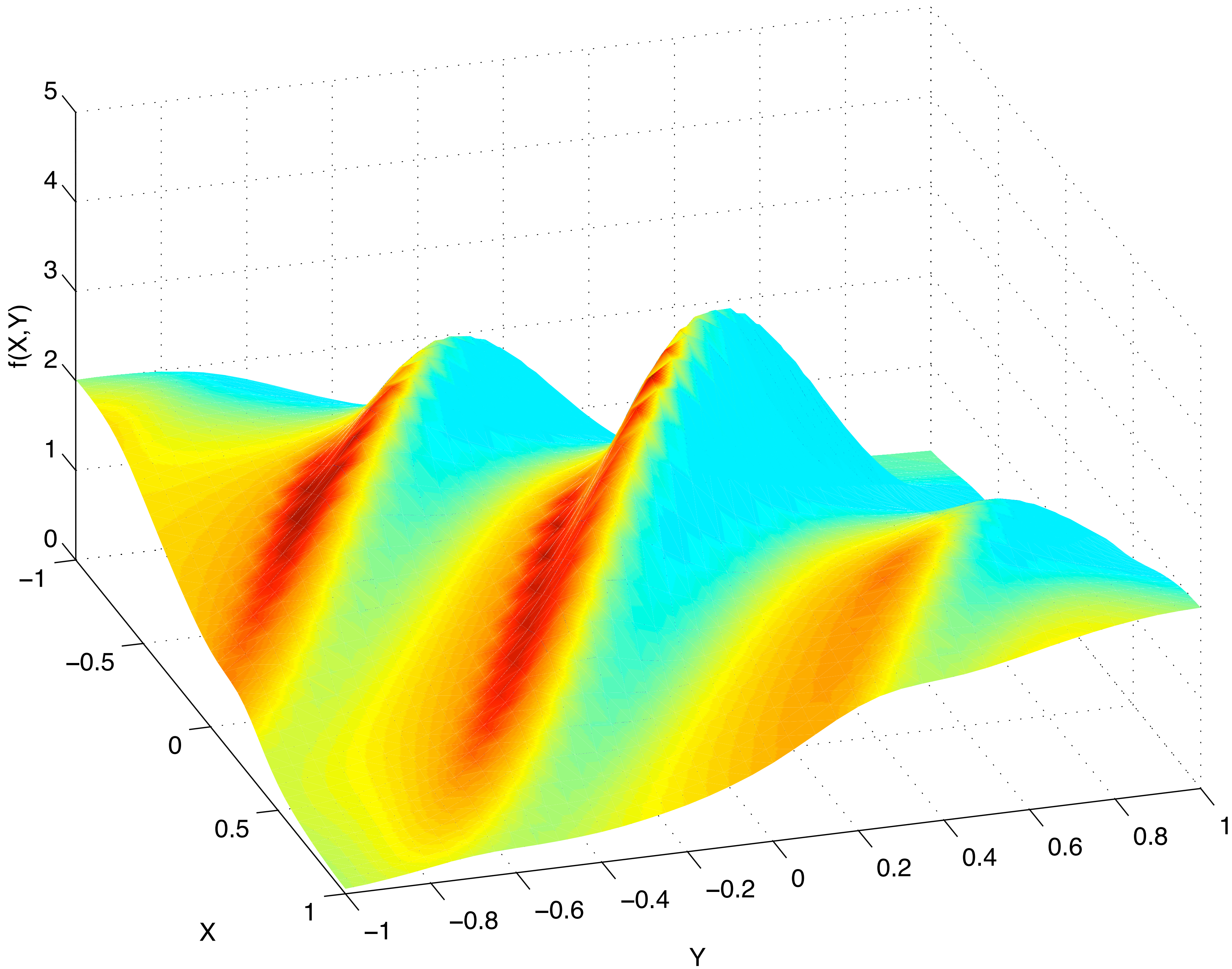
Overfitting

- Overfit: training likelihood $\gg$ test likelihood
  - often a result of overconfidence
- Overfitting is an indicator that the MLE or MAP approximation is a bad one
- Bayesian inference rarely overfits
  - may still lead to bad results for other reasons!
  - e.g., not enough data, bad model class, …

So, we want the predictive distribution

- Most of the time…
  - Graphical model is big and highly connected
  - Variables are high-arity or continuous
- Can't afford exact inference
  - Inference reduces to numerical integration (and/or summation)
  - We'll look at randomized algorithms

Numerical integration



2D is 2 easy!

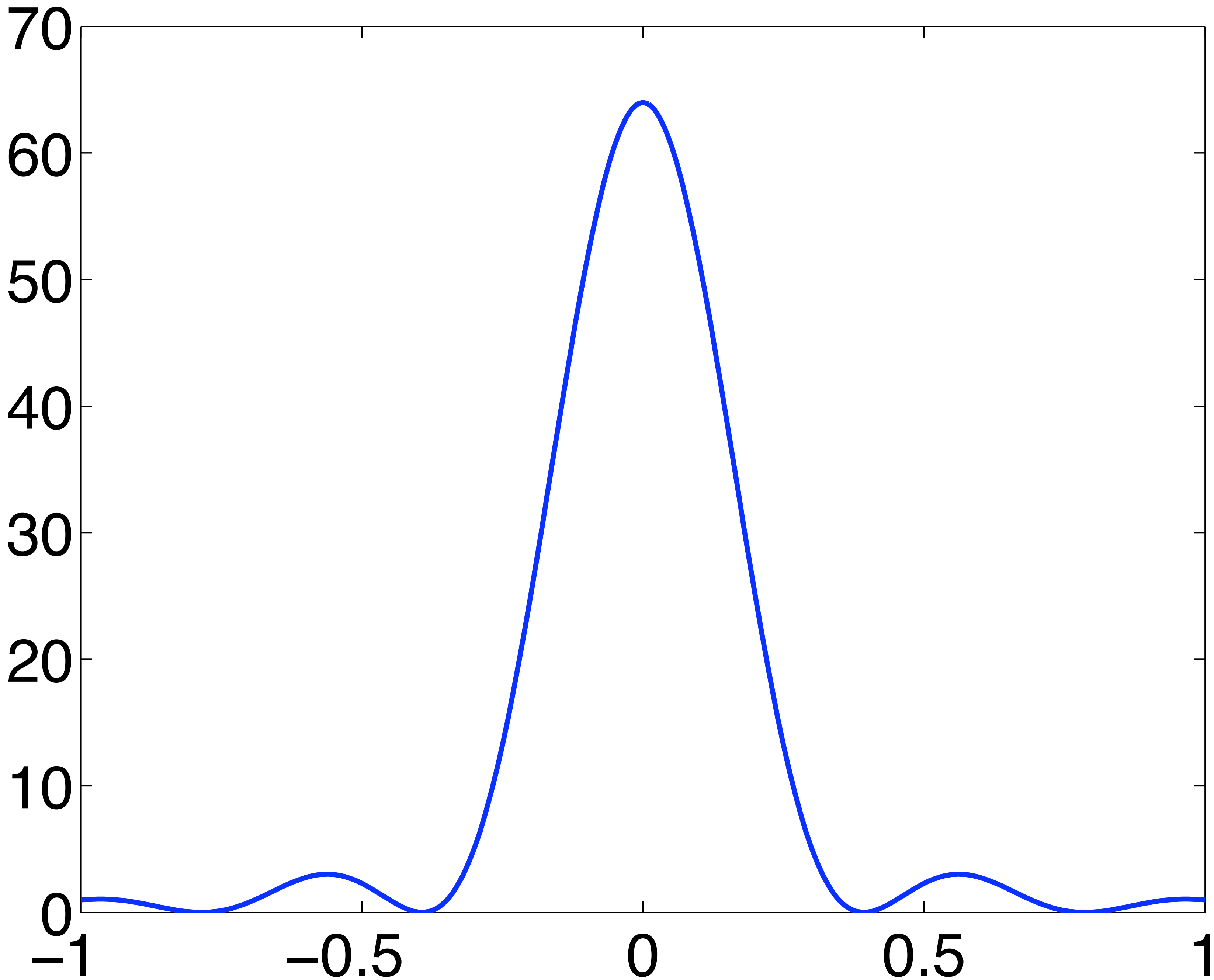
- We care about high-D problems
- Often, much of the mass is hidden in a tiny fraction of the volume
    - simultaneously try to discover it and estimate amount

Application: SLAM

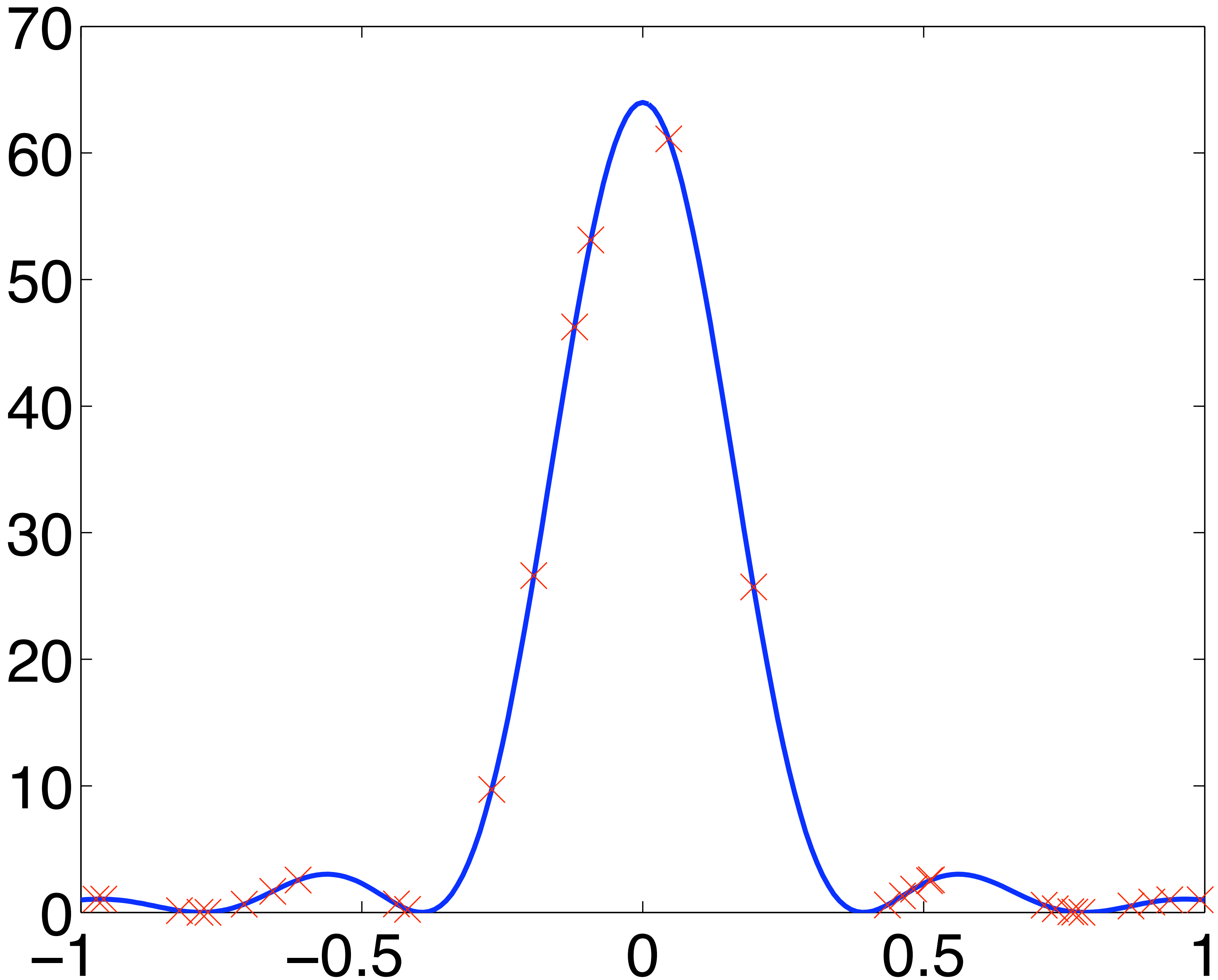
Integrals in multi-million-D

Eliazar and Parr, IJCAI-03

Simple 1D problem



Uniform sampling



Uniform sampling

E(f(X)) =

- So, is desired integral
- But standard deviation can be big
- Can reduce it by averaging many samples
- But only at rate $1/\sqrt{N}$

Importance sampling

- Instead of X ~ uniform, use X ~ Q(x)
- Q =
- Should have Q(x) large where f(x) is large
- Problem:

Importance sampling

- Instead of X ~ uniform, use X ~ Q(x)
- Q =
- Should have Q(x) large where f(x) is large
- Problem:

$$E_Q(f(X)) = \int Q(x) f(x) dx$$

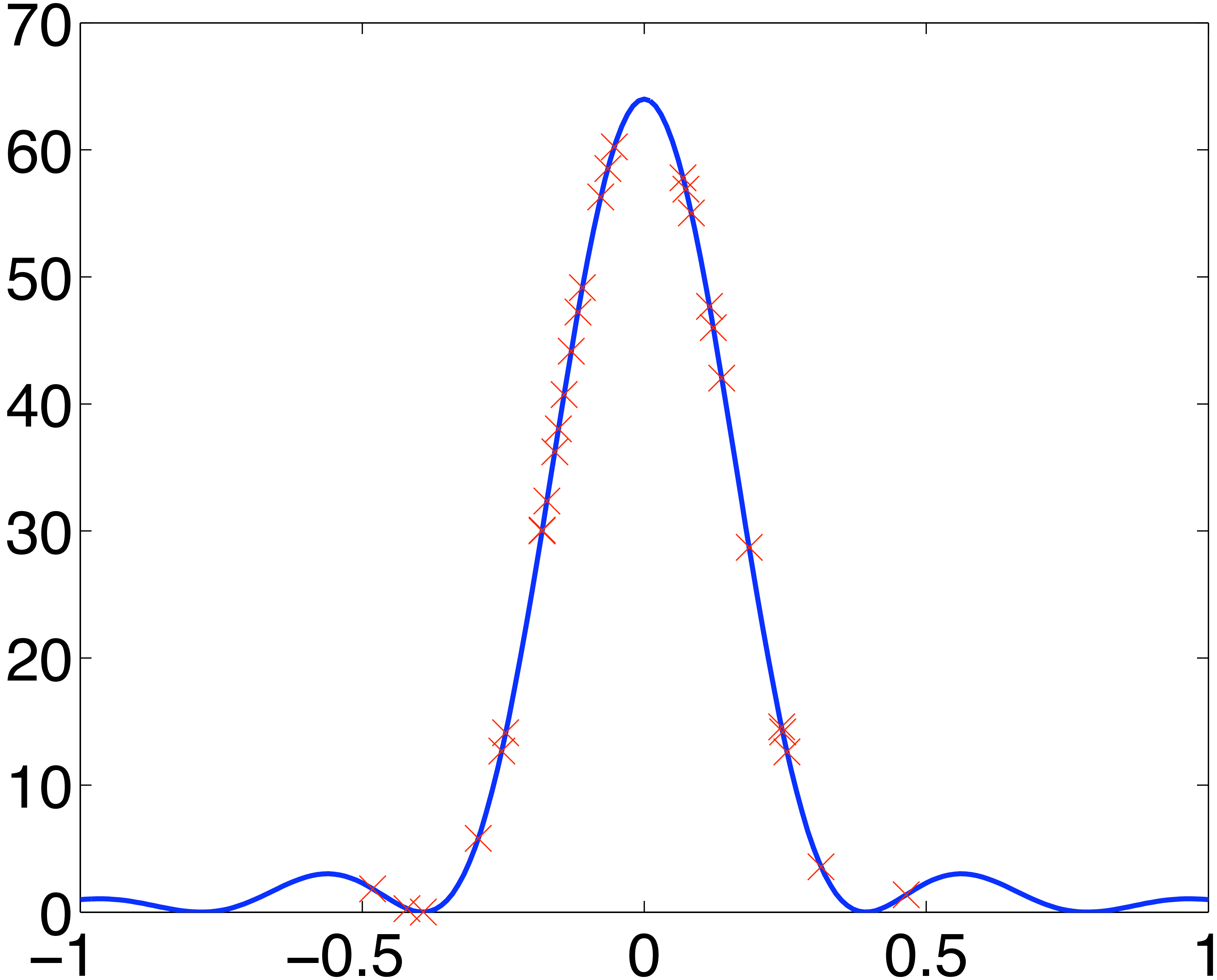
Importance sampling

$$h(x) \equiv f(x)/Q(x)$$

Importance sampling

- So, take samples of $h(X)$ instead of $f(X)$
- $W_i = 1/Q(X_i)$ is **importance weight**
- $Q = 1/V$ yields uniform sampling

Importance sampling



Variance

- How does this help us control variance?
- Suppose:
  - f big
  - Q small
- Then h = f/Q:
- Variance of each weighted sample is
- Optimal Q?

Importance sampling, part II

- Suppose we want

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

- Pick N samples X_i from proposal Q(X)
- Average W_i $g(X_i)$, where importance weight is
  - W_i =

Importance sampling, part II

- Suppose we want

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

- Pick N samples X_i from proposal $Q(X)$
- Average $W_i\ g(X_i)$, where importance weight is
  - $W_i =$

$$E_Q(Wg(X)) = \int Q(x)[P(x)/Q(x)]g(x)dx = \int P(x)g(x)dx$$

Two variants of IS

- Same algorithm, different terminology
  - want $\int f(x)\,dx$ vs. $E_P(f(X))$
  - W = 1/Q vs. W = P/Q

Parallel importance sampling

- Suppose we want

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

- But P(x) is unnormalized (e.g., represented by a factor graph)—know only Z P(x)

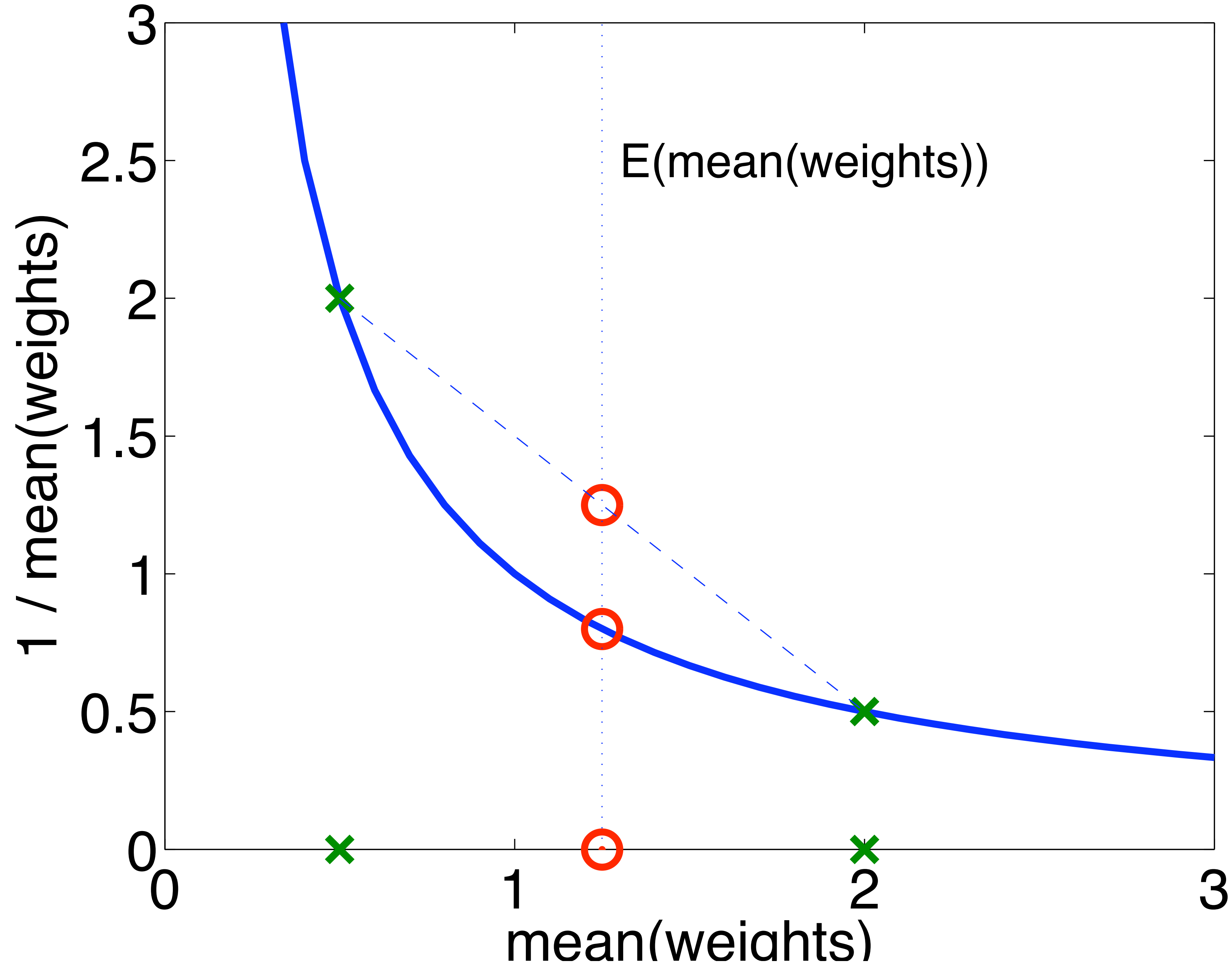
Parallel IS

- Pick N samples X_i from proposal $Q(X)$
- If we knew $W_i = P(X_i)/Q(X_i)$, could do IS
- Instead, set
  - and,
- Then:

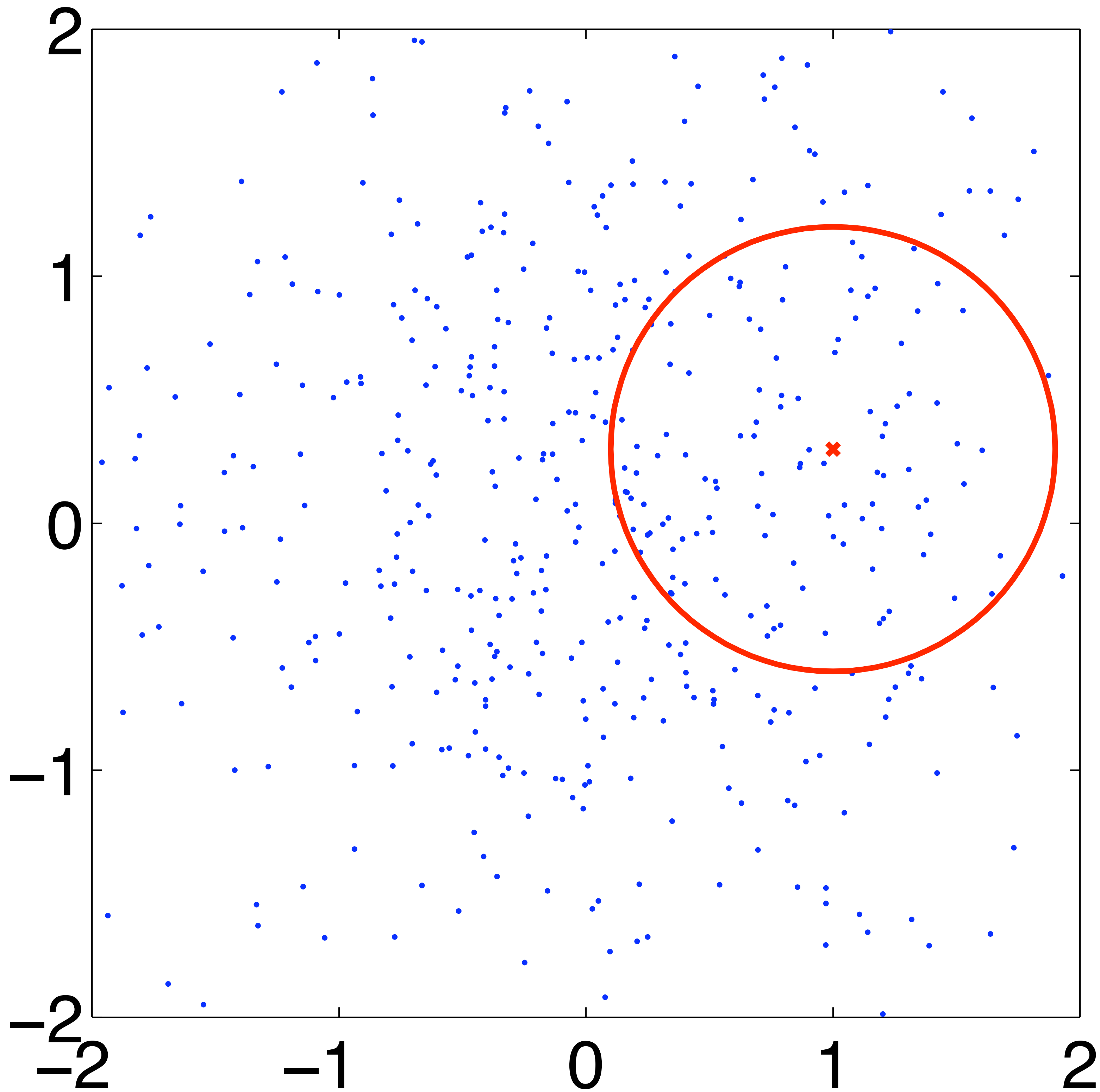
Parallel IS

- Final estimate:

Parallel IS is biased

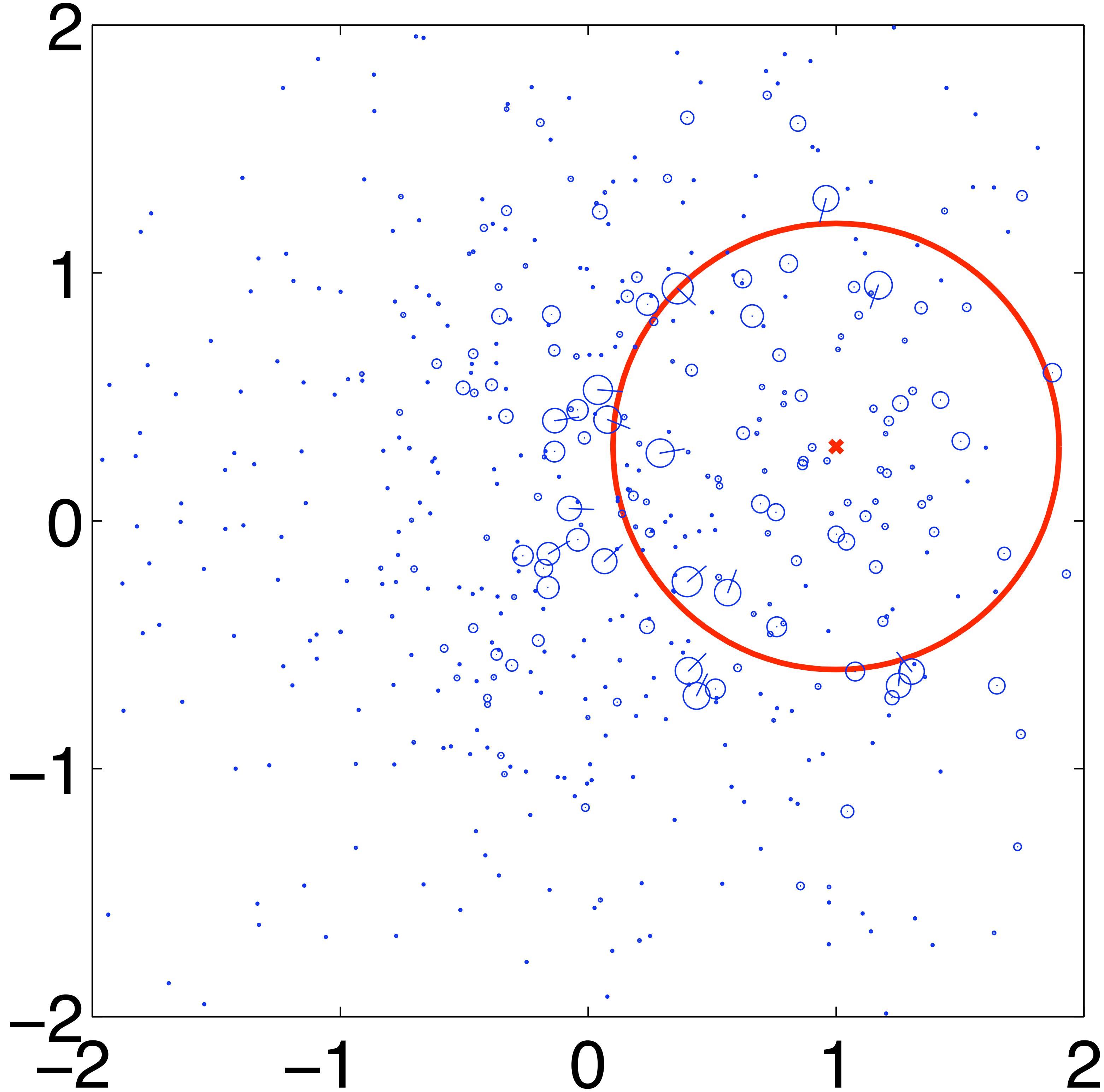


$$E(\bar{W}) = Z, \text{ but } E(1/\bar{W}) \neq 1/Z \text{ in general}$$



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

$$f(x, y, \theta) = Q(x, y, \theta)P(o = 0.8 \mid x, y, \theta)/Z$$



- Posterior

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