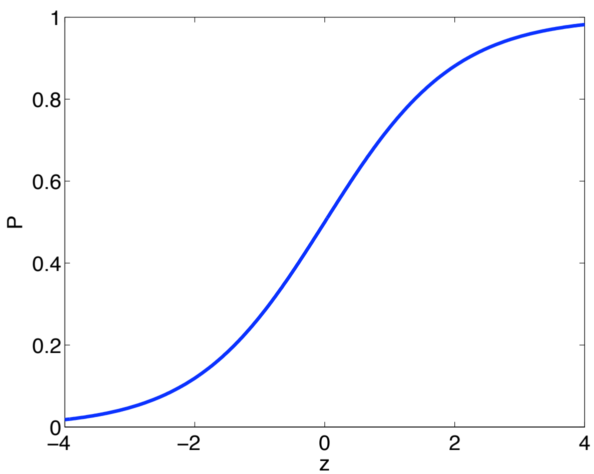
Review

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



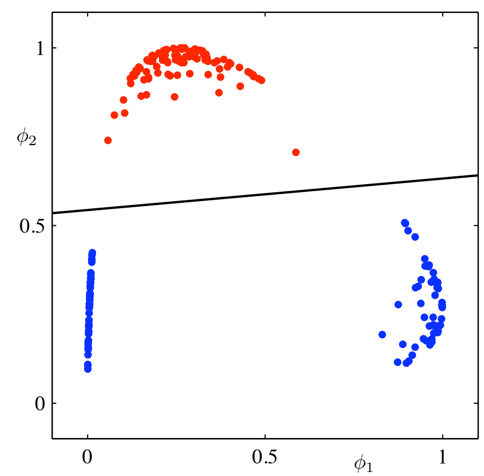


figure from book

Review

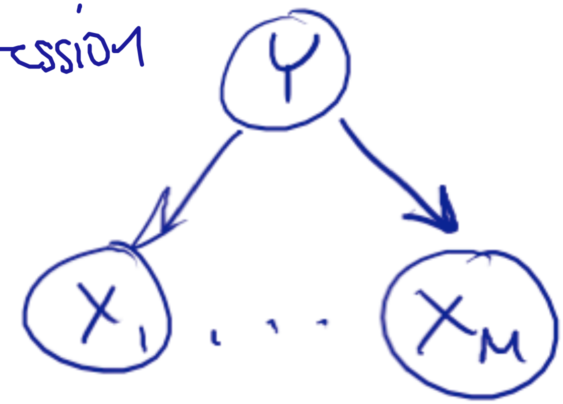
NB

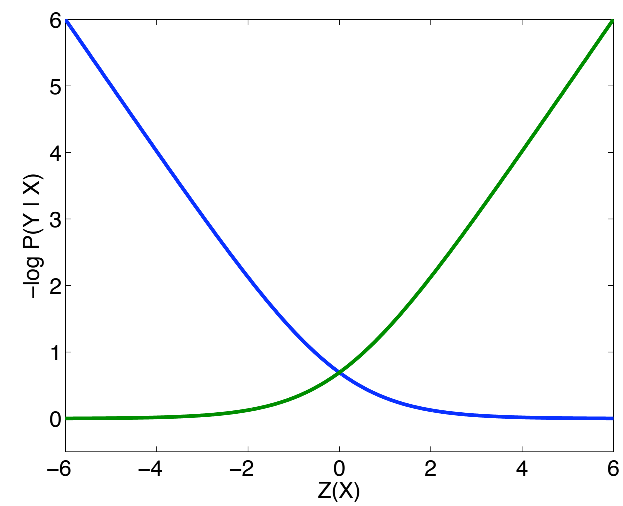
P (X,Y)

P(Y | X)

logistic regression

- 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

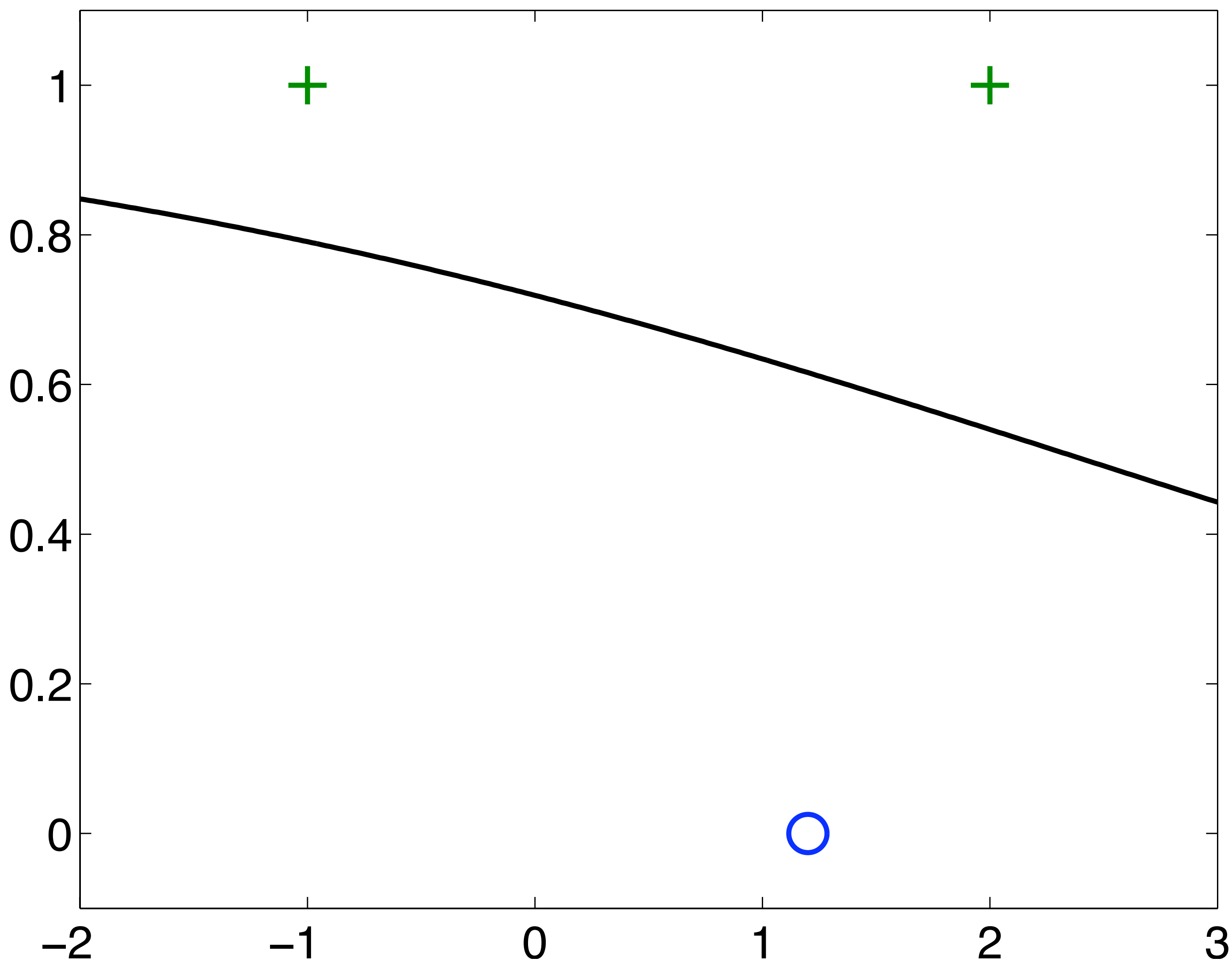




$\approx \max(0, z) \quad \max(0, -z)$

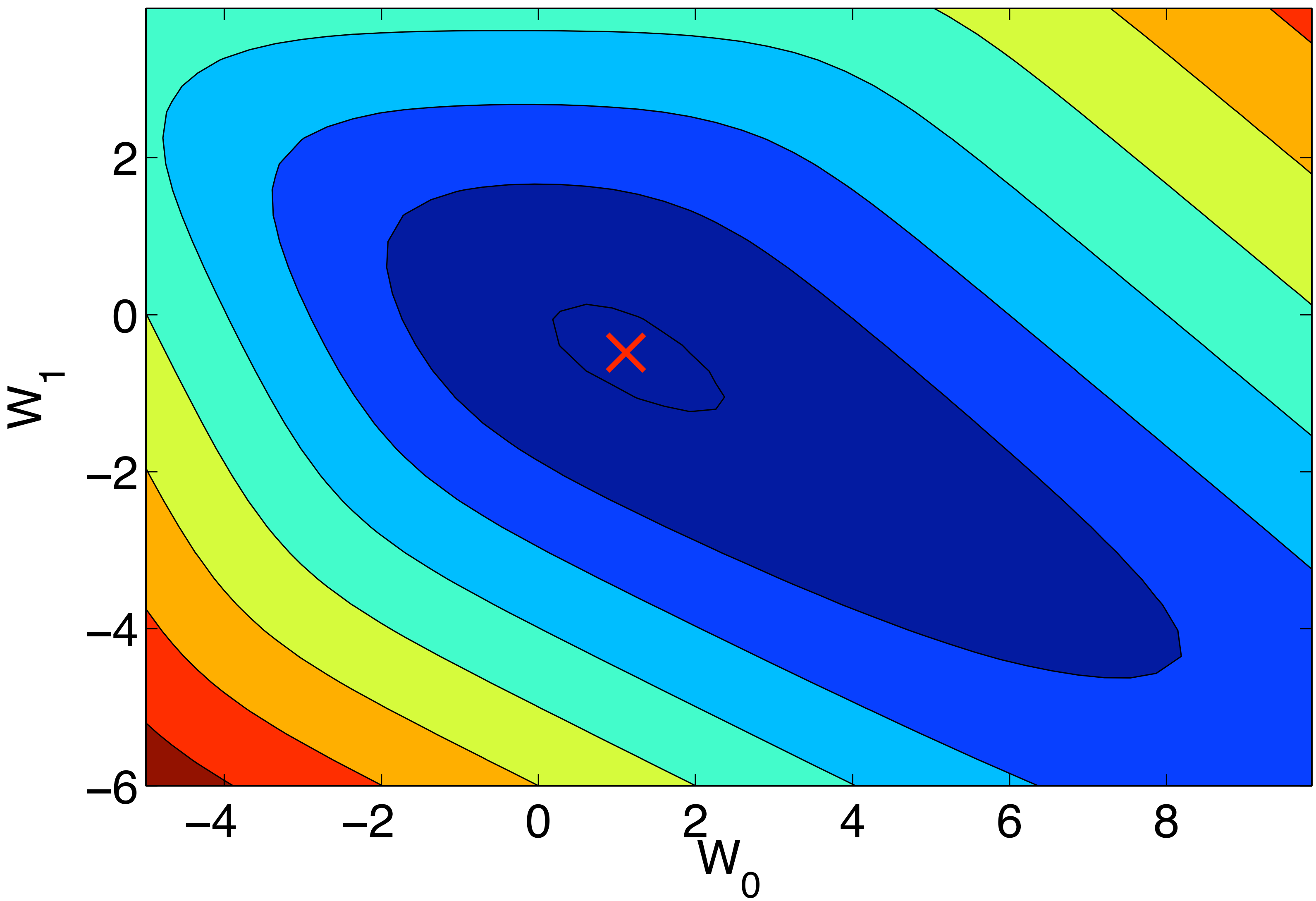
$\approx \max(0, z) \quad \max(0, -z)$

Example



$= \sum_{i=1}^{3} \log \sigma(y^i x^i \cdot w)$

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



Generalization: multiple classes

$k = \arg\max$ iff $z_k \geq z_{k'}$ $\forall k'$

$w^1 \ldots w^C \in \mathbb{R}^{M+1}$

- One weight vector per class: $Y \in \{1,2,\ldots,C\}$
  - $P(Y=k) =$ $\exp(z_k) / \sum_{k'=1}^{C} \exp(z_{k'})$
  - $Z_k =$ $w_0^k + \sum_{j=1}^{M} w_j^k X_j$

$\hat{y} = \arg\max_{y} P(y \mid X, w)$

$y \in \{0,1\}$

$y \in \{1 \ldots C\}$

- In 2-class case:

$\hat{y} = \arg\max_{y \in \{1..C\}} P(y \mid x, w)$

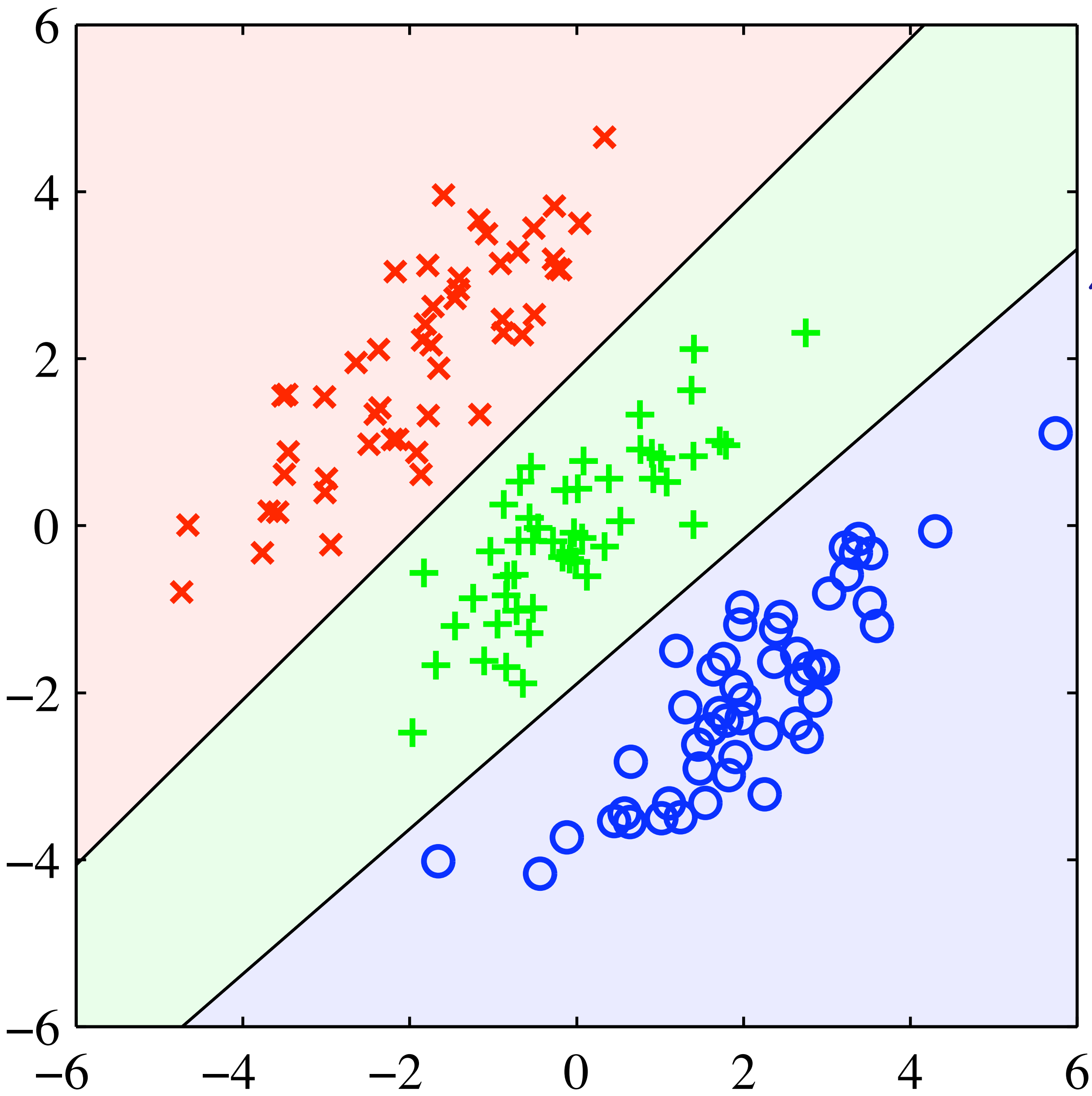
$\hat{y} = k$ iff $z_k \geq z_{k'}$ $(\forall k')$

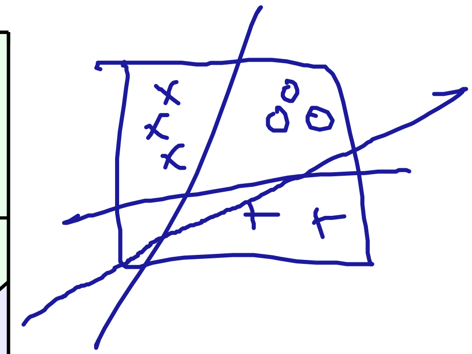
$\exp(z_0) / \exp(z_0) + \exp(z_1)$

$= 1 / 1 + \exp(z_1 - z_0)$

$z_1 - z_0 = \underbrace{w_0^1 - w_0^0}_{-w_0} + \sum_j \underbrace{(w_j^1 - w_j^0)}_{-w_j} X_j$

Multiclass example





See reading for more examples

figure from book

Priors and conditional MAP

(2 class)

- $P(Y \mid X, W) = \sigma(YZ)$
  - $Z = w_0 + \sum_j w_j x_j$
- As in linear regression, can put prior on W
  - common priors: L_2 (ridge), L_1 (sparsity)

$\cancel{-\log P(w) \propto \|w\|_2^2}$ $\quad$ $\cancel{-\log P(w) \propto \|w\|_1}$

- $\widehat{\arg\max}_w P(W=w \mid X,Y) = \widehat{\arg\max}_w P(Y \mid X, W=w) P(W=w)$

actually, $-\log P(w) = k_1 + k_2 \|w\|_2^2$

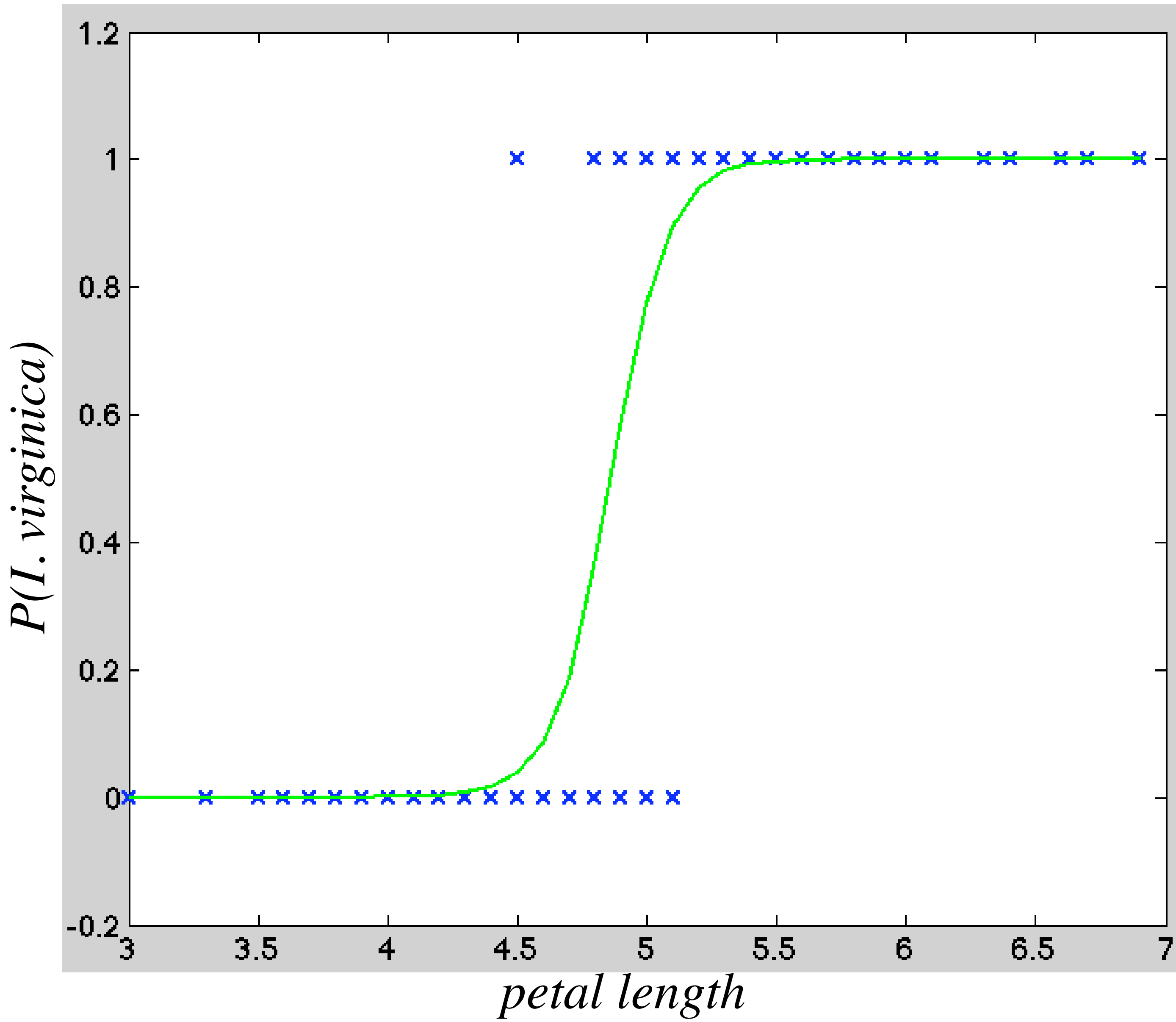
or $-\log P(w) = k_1 + k_2 \|w\|_1$ (linear w.t. affine function)

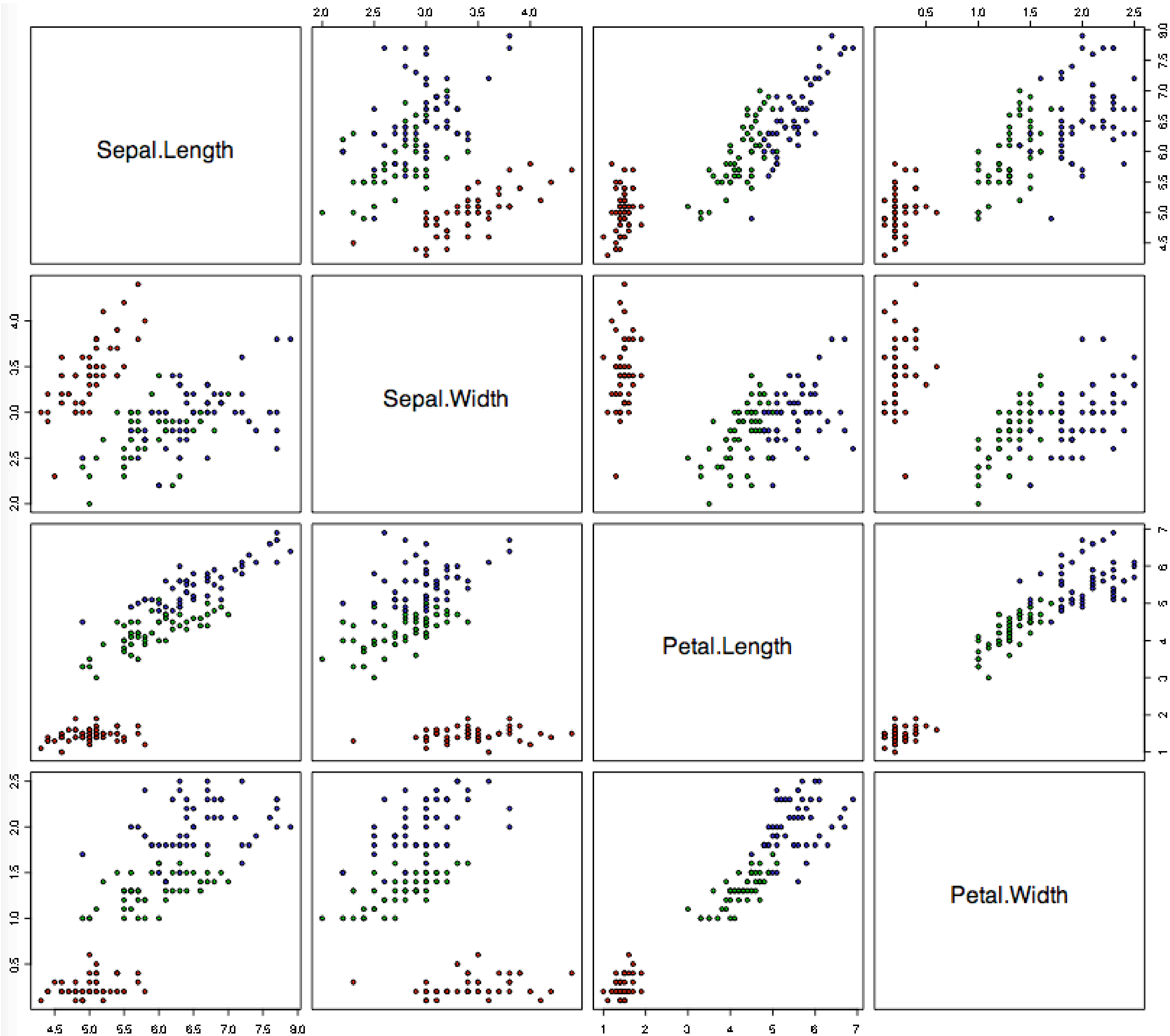
Software

- 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)

still convex

Historical application: Fisher iris data

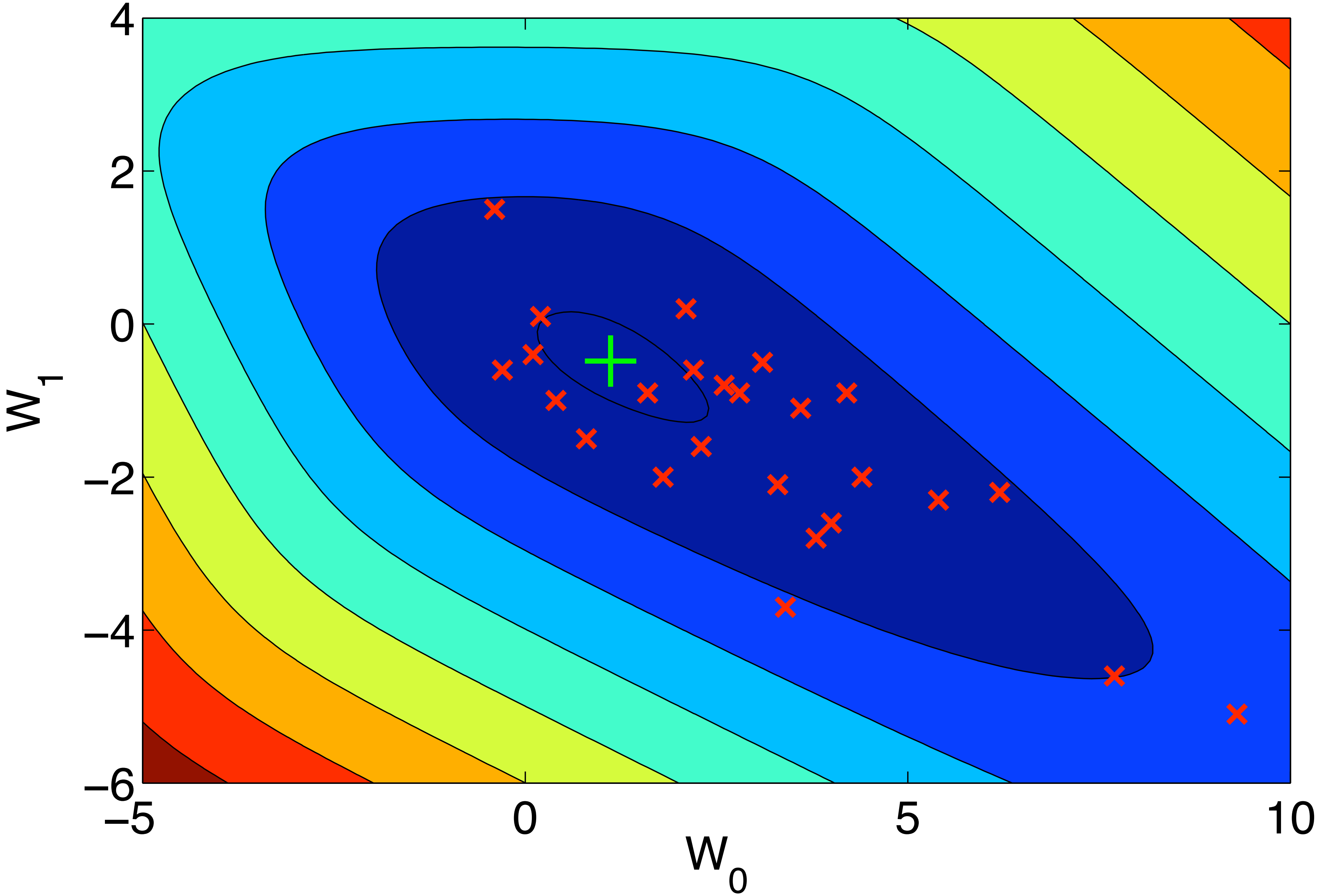



Sepal.Length
Sepal.Width
Petal.Length
Petal.Width

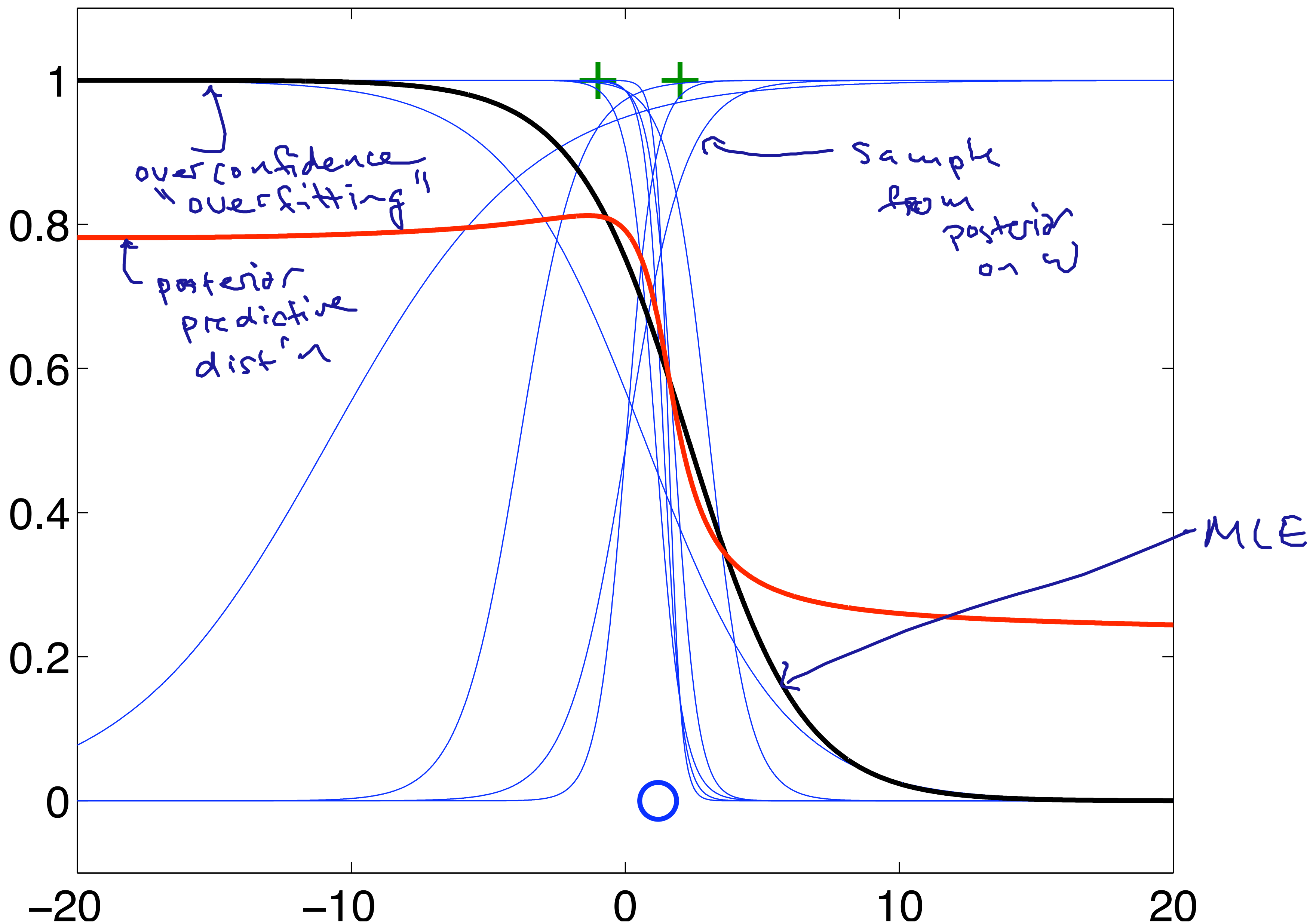
Bayesian regression

- 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



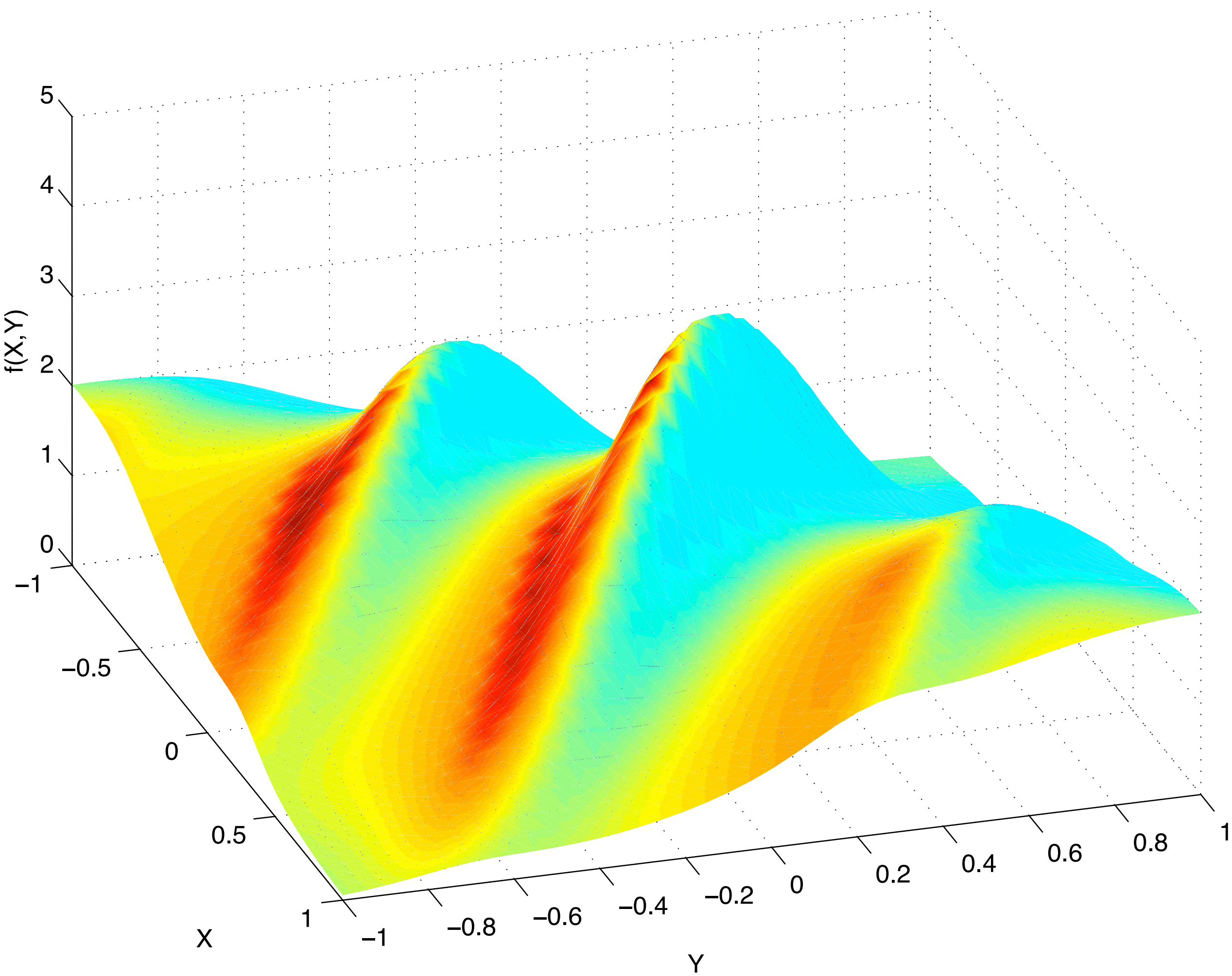
Overfitting

- 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

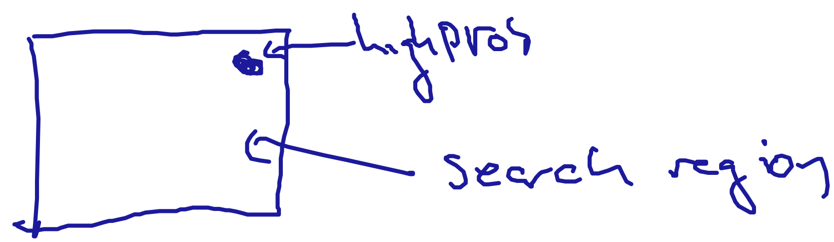
- Most of the time…
  - Graphical model is big and highly connected
  - Variables are high-arity or continuous
- Can't afford exact inference
  - Inference reduces to numerical integration (and/or summation)
  - We'll look at randomized algorithms

Numerical integration

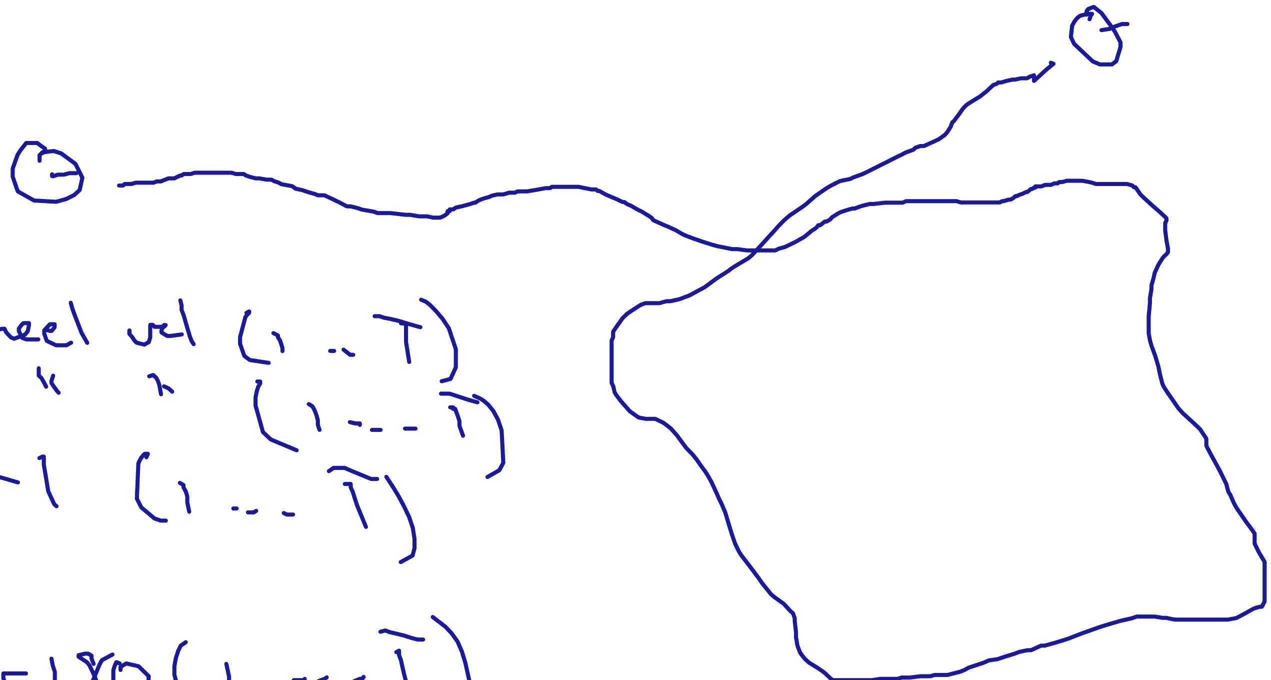


2D is 2 easy!

- 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



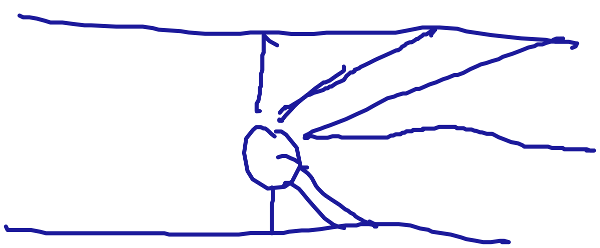
left wheel vel (1 .. T)

right " " (1 ... T)

laser1 (1 ... T)

⋮

laser180 (1 ... T)



laser range scan

true X (1 ... T)

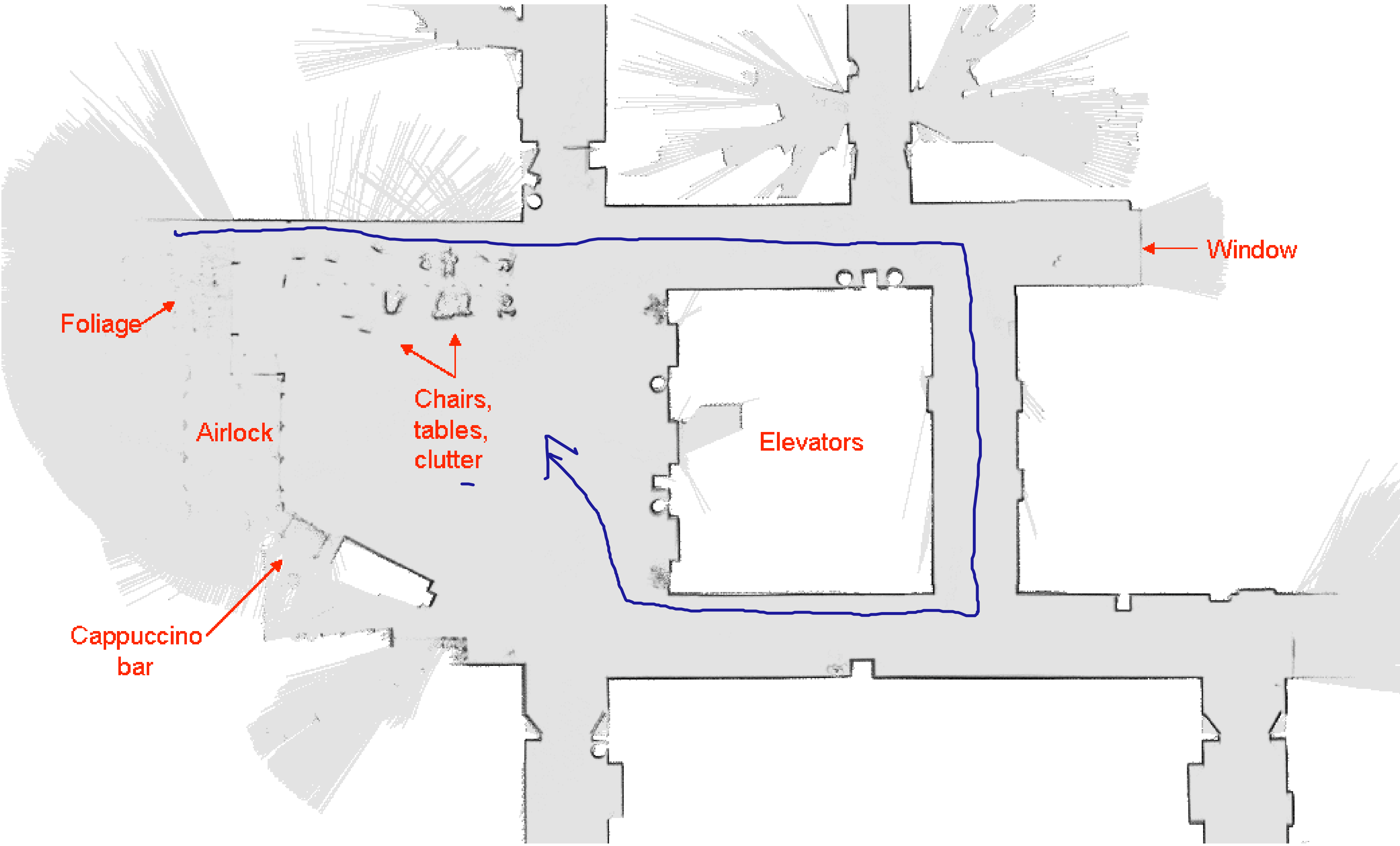
true Y (1 ... T)

true Θ (1 ... T)

true $occ_{1,1}$ (1 .. T)

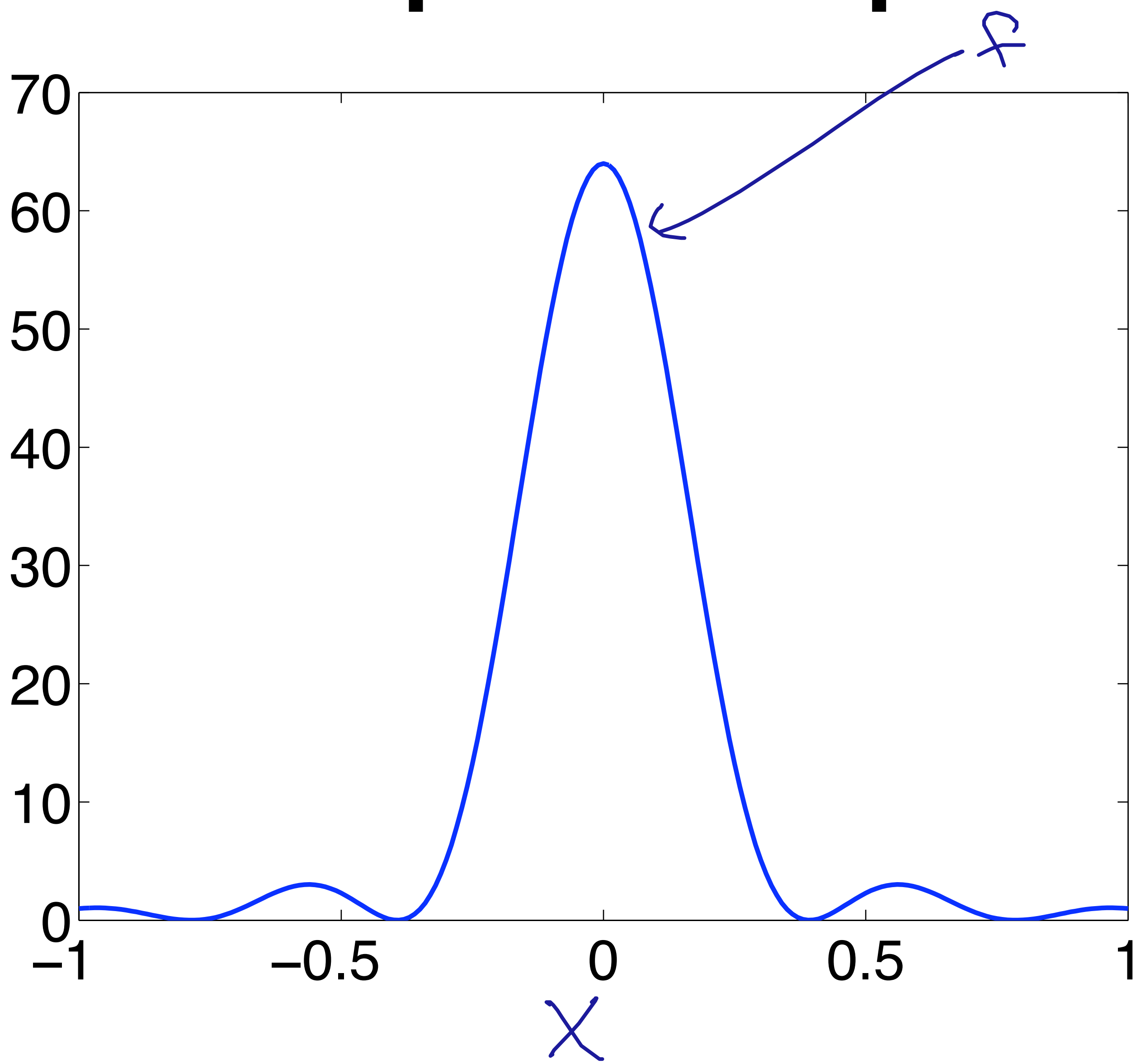
... true $occ^{100,100}$ (1 ... T)

Integrals in multi-million-D



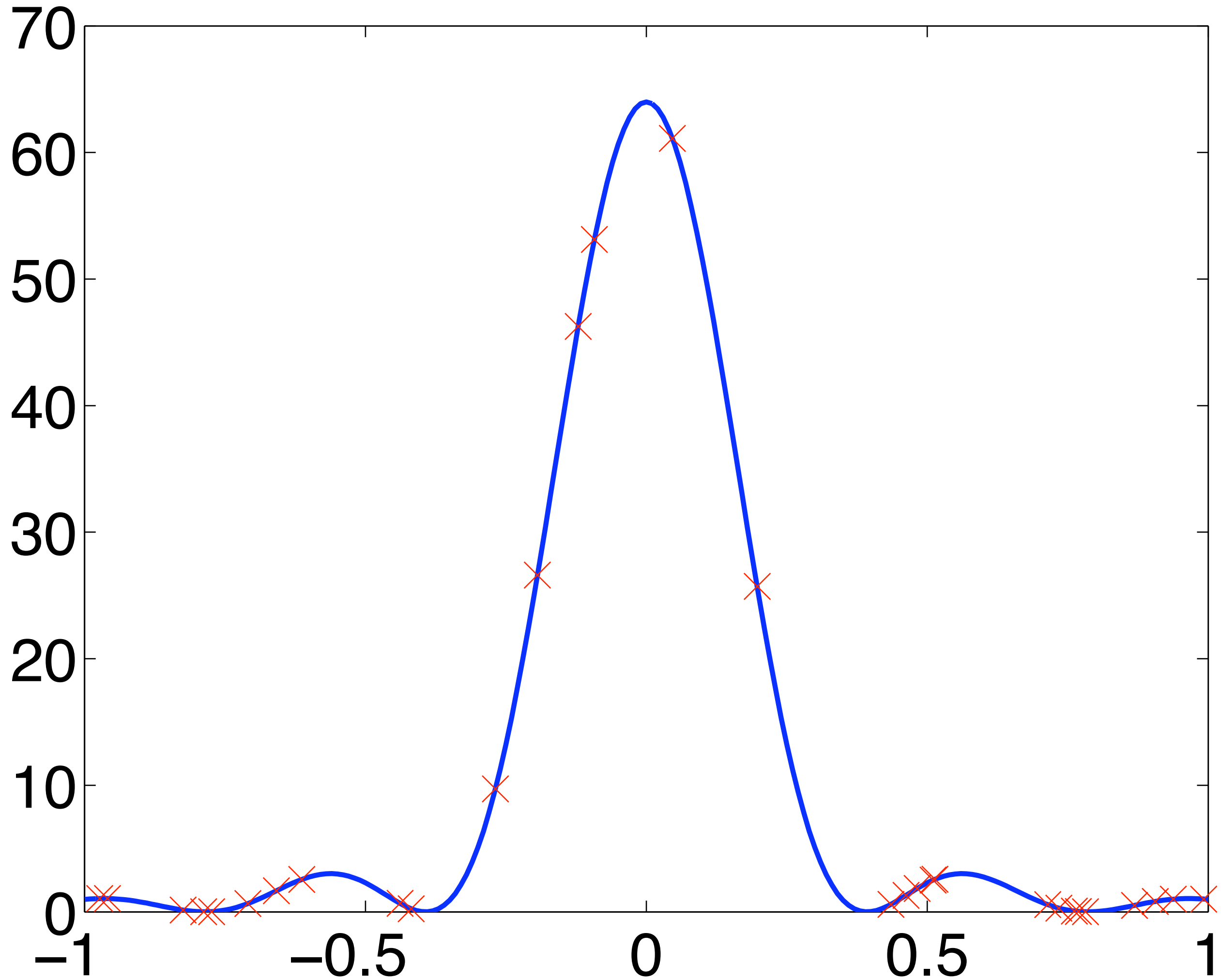
Eliazar and Parr, IJCAI-03

Simple 1D problem



$$\int_{-1}^{1} f(x)\,dx$$

Uniform sampling



$\int f \, dx \approx$

$\hat{F} = \frac{1}{N} \cdot 2 \sum_{i=1}^{N} f(x_i)$

$\int f \, dx \approx 24.0$

$\hat{F} = 14.7$

$\int_{-1}^{1} \frac{1}{2} dx = 1$

Uniform sampling

$\frac{1}{2} = 1/V = 1/\text{volume}$

$E(f(X)) = \int_{-1}^{1} p(x) f(X) dx = \frac{1}{2} \int f(x) dx$

- So, $2\,E(f(X))$ is desired integral (or, $V\,E(f(x))$)
- 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 = importance dist'n
- Should have Q(x) large where f(x) is large
- Problem: naive estimate $\frac{1}{N} \sum_i f(X_i)$ wrong!

$E(f(X_i)) = \int Q(x) f(x) dx \neq$ any simple fn of $\int f dx$

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

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

$$E_Q(h(x)) = \int Q(x)\, h(x)\, dx$$

$$= \int Q\cancel{(x)} \frac{f(x)}{Q\cancel{(x)}}\, dx$$

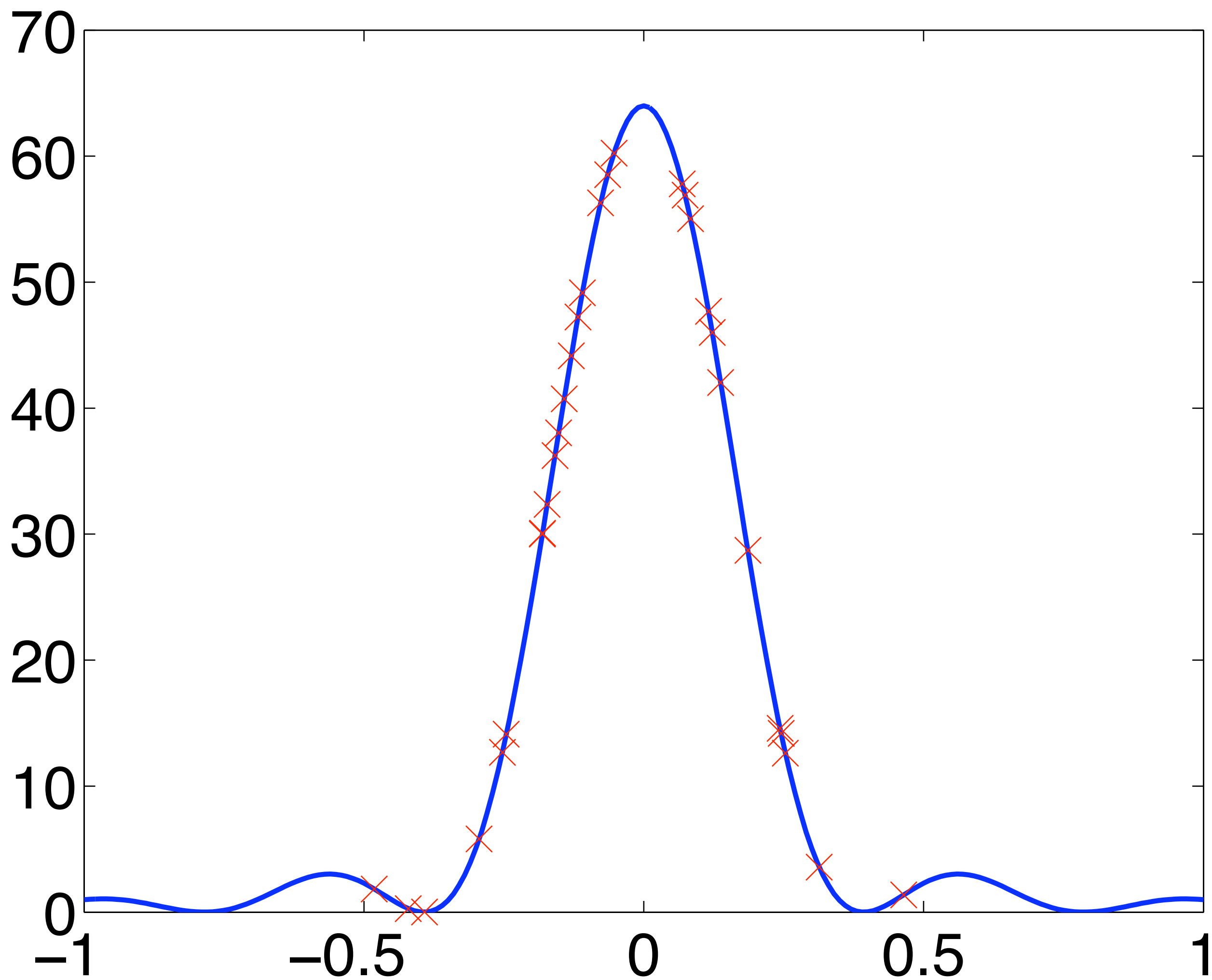
$$= \int f(x)\, dx$$

Importance sampling

$$\hat{F} = \sum_{i=1}^{N} h(X_i) = \sum_{i=1}^{N} \underbrace{\frac{1}{Q(X_i)}}_{} f(X_i)$$

- 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



$\int f \, dx = 24.0$

$\hat{F} = 25.8$

$Q(x) = N(x \mid 0, .25^2)$

30 samples

Variance

- How does this help us control variance?
- Suppose:
  - f big $\Rightarrow$ Q big
  - Q small $\Rightarrow$ f small
- Then h = f/Q: moderate values
- Variance of each weighted sample is moderate
- Optimal Q? $\frac{f(x)}{Z}$ $\leftarrow$ never happens in practice

$= f(x) / \int f \, dx$

Importance sampling, part II

- Suppose we want

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

factors

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

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)

$\underline{P(x|\text{data})} \propto P(\text{data}|X)\,P(x)$

Sample from posterior

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 $\hat{W}_i = Z\,P(X_i)\,/\,Q(X_i)$
  - and, $\bar{W} = \frac{1}{N}\sum_{i=1}^{N} \hat{W}_i$
- Then: $E(\hat{W}_i) = \int \cancel{Q(x)}\, Z\,P(x)/\cancel{Q(x)}\,dx$

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

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

Parallel IS

- Final estimate:

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