Machine Learning
Part I: Introduction

Ethem Alpaydın
alpaydin@boun.edu.tr

Ref: E. Alpaydın (2010). *Introduction to Machine Learning*, 2e, The MIT Press.

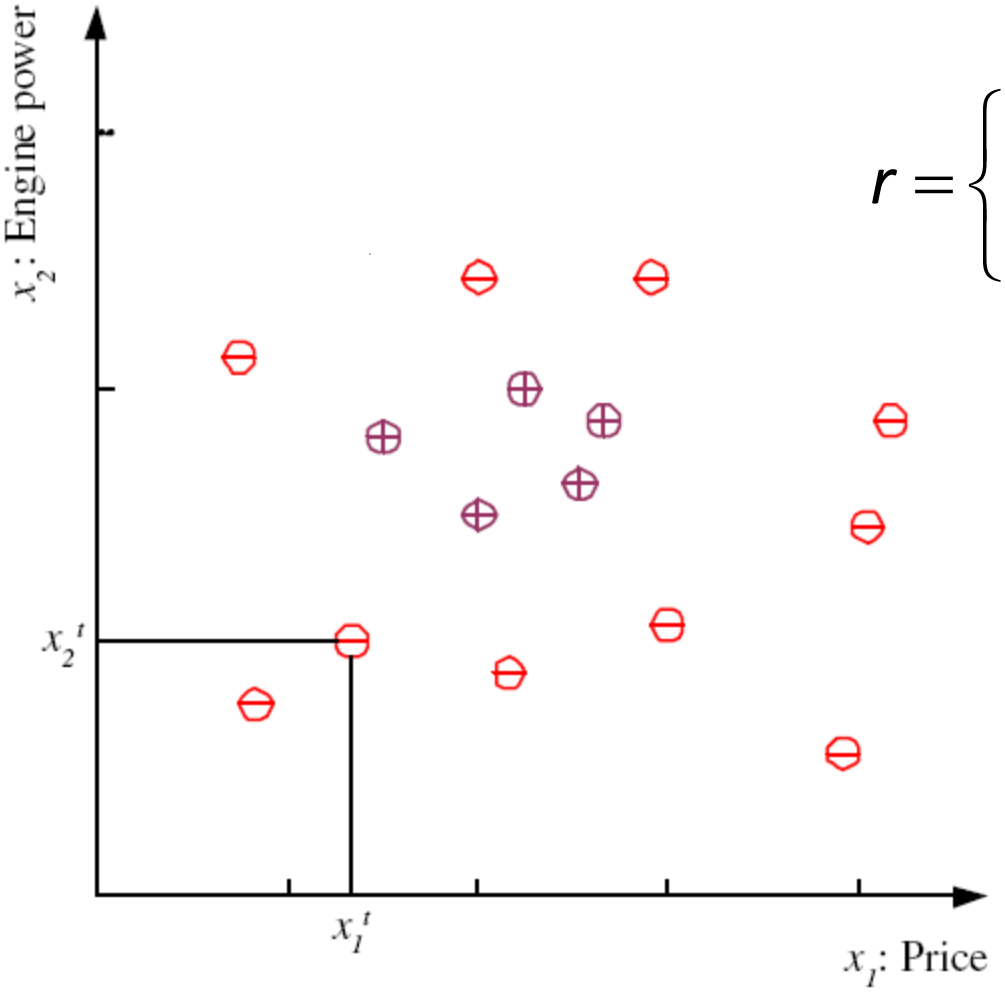
Learning a Class from Examples

- Class C of a "family car"
  - Prediction: Is car *x* a family car?
  - Knowledge extraction: What do people expect from a family car?
- Output:

  Positive (+) and negative (–) examples
- Input representation:

  x_1: price, x_2 : engine power

Training set $\mathcal{X}$

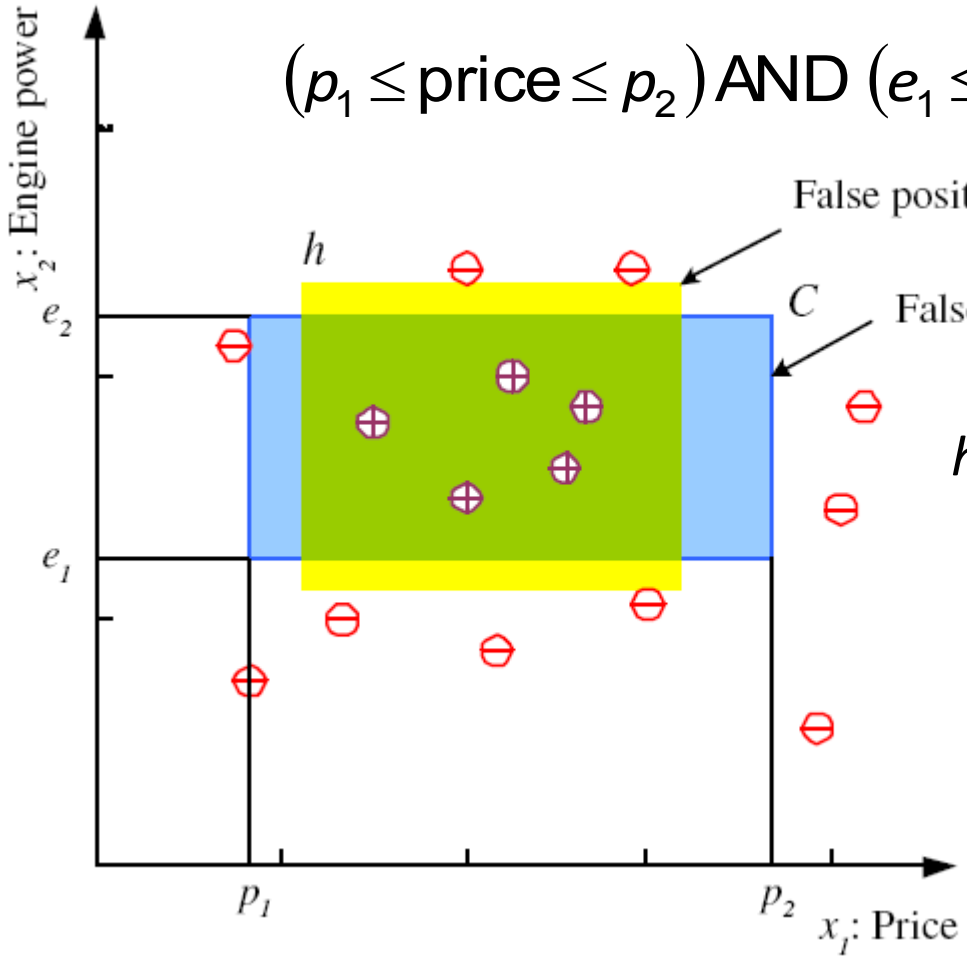


$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

$$r = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is positive} \\ 0 \text{ if } \mathbf{x} \text{ is negative} \end{cases}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Hypothesis class $\mathcal{H}$



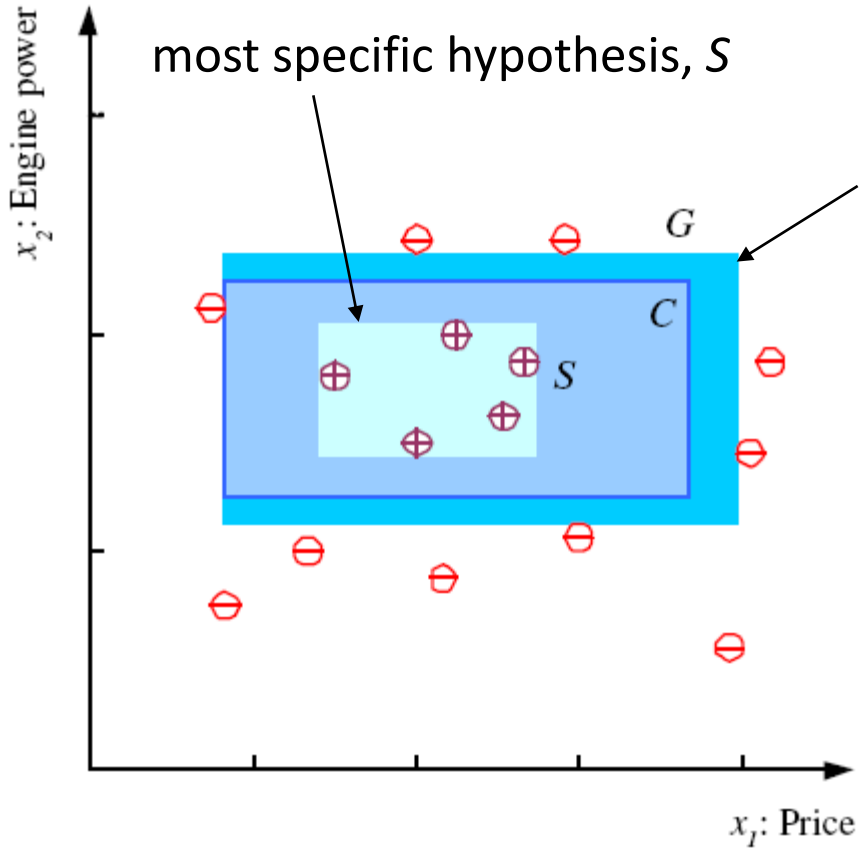
$$(p_1 \le \text{price} \le p_2) \text{ AND } (e_1 \le \text{engine power} \le e_2)$$

$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ says } \mathbf{x} \text{ is positive} \\ 0 \text{ if } h \text{ says } \mathbf{x} \text{ is negative} \end{cases}$$

Error of h on $\mathcal{H}$

$$E(h \mid \mathcal{X}) = \sum_{t=1}^{N} 1\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

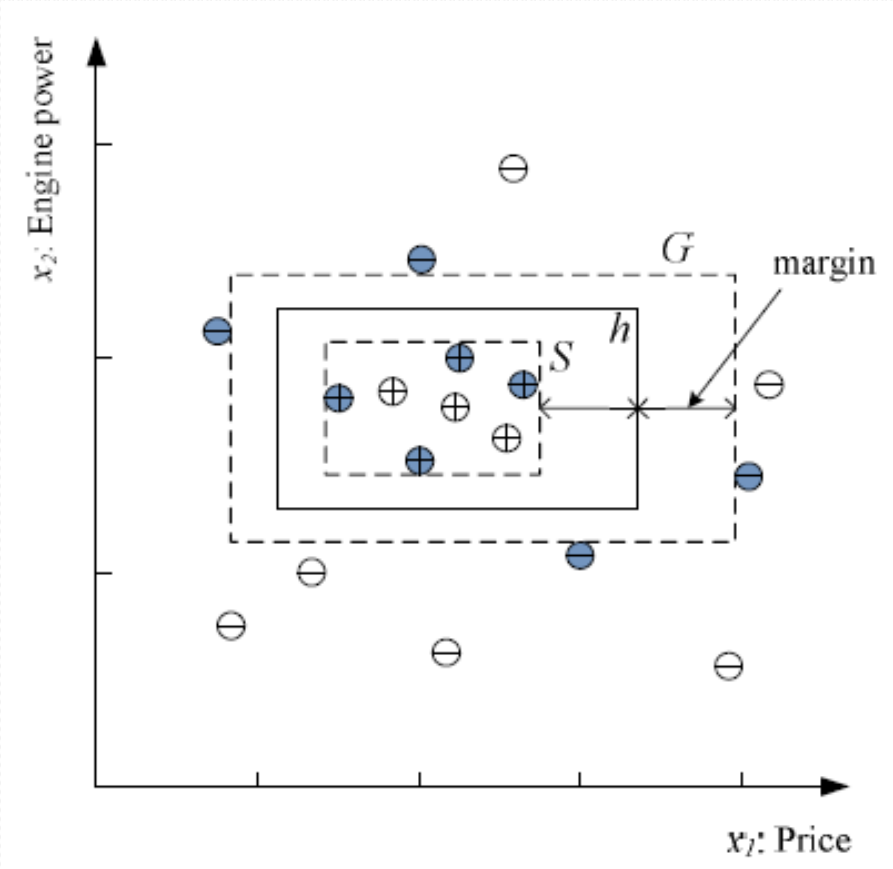
S, G, and the Version Space



$h \in$ H, between S and G is
consistent
and make up the
version space
(Mitchell, 1997)

Margin

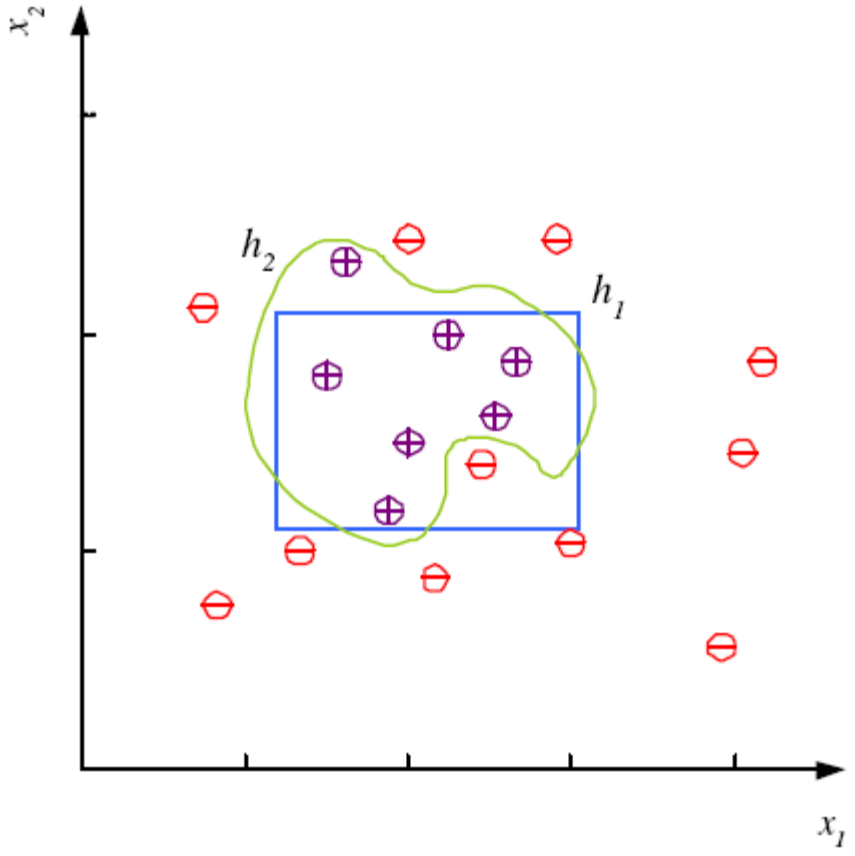
- Choose *h* with largest margin



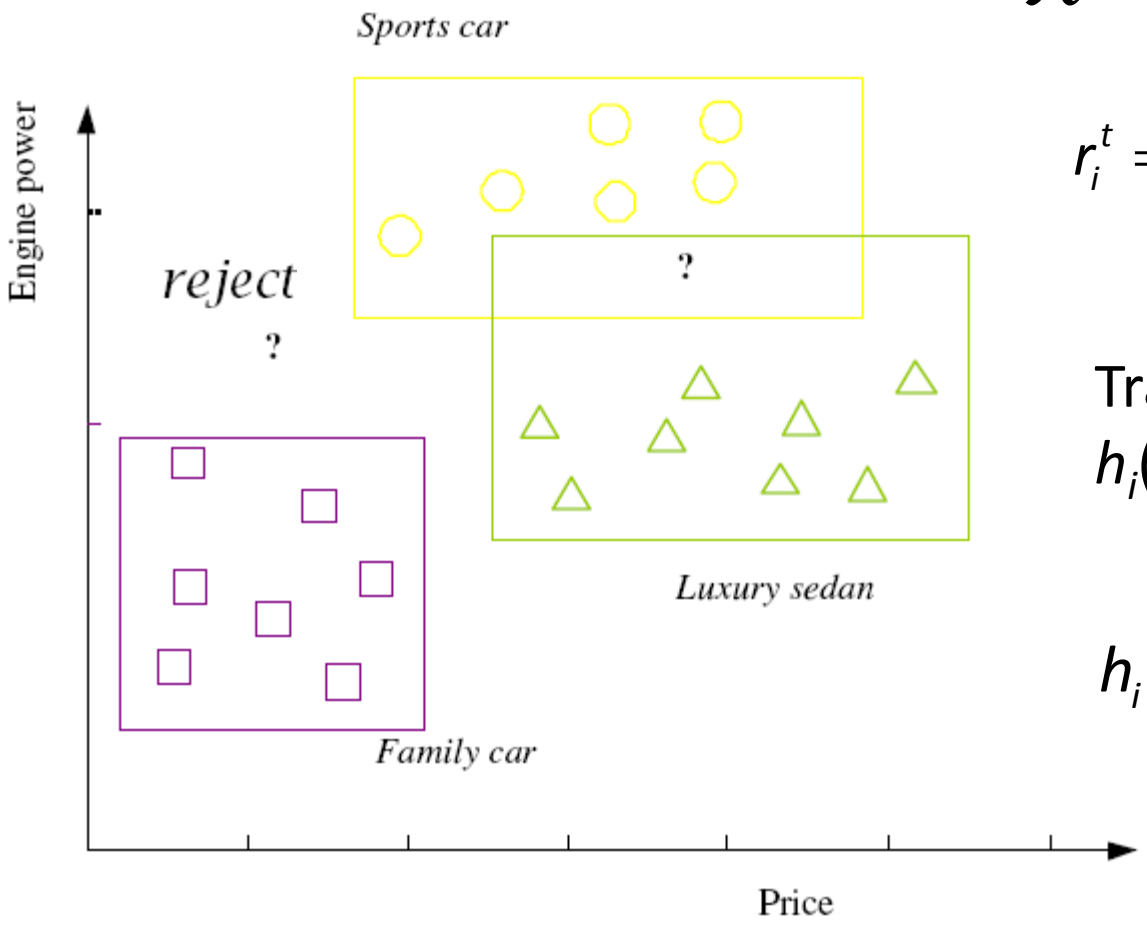
Noise and Model Complexity

Use the simpler one because

- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance - Occam's razor)



Multiple Classes, C_i i=1,...,K



$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

$$r_i^t = \begin{cases} 1 \text{ if } \mathbf{x}^t \in C_i \\ 0 \text{ if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Train hypotheses $h_i(\mathbf{x})$, $i = 1,...,K$:

$$h_i\left(\mathbf{x}^t\right) = \begin{cases} 1 \text{ if } \mathbf{x}^t \in C_i \\ 0 \text{ if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Face Recognition

Training examples of a person and negative examples

Test image

ORL dataset,
AT&T Laboratories, Cambridge UK

Regression

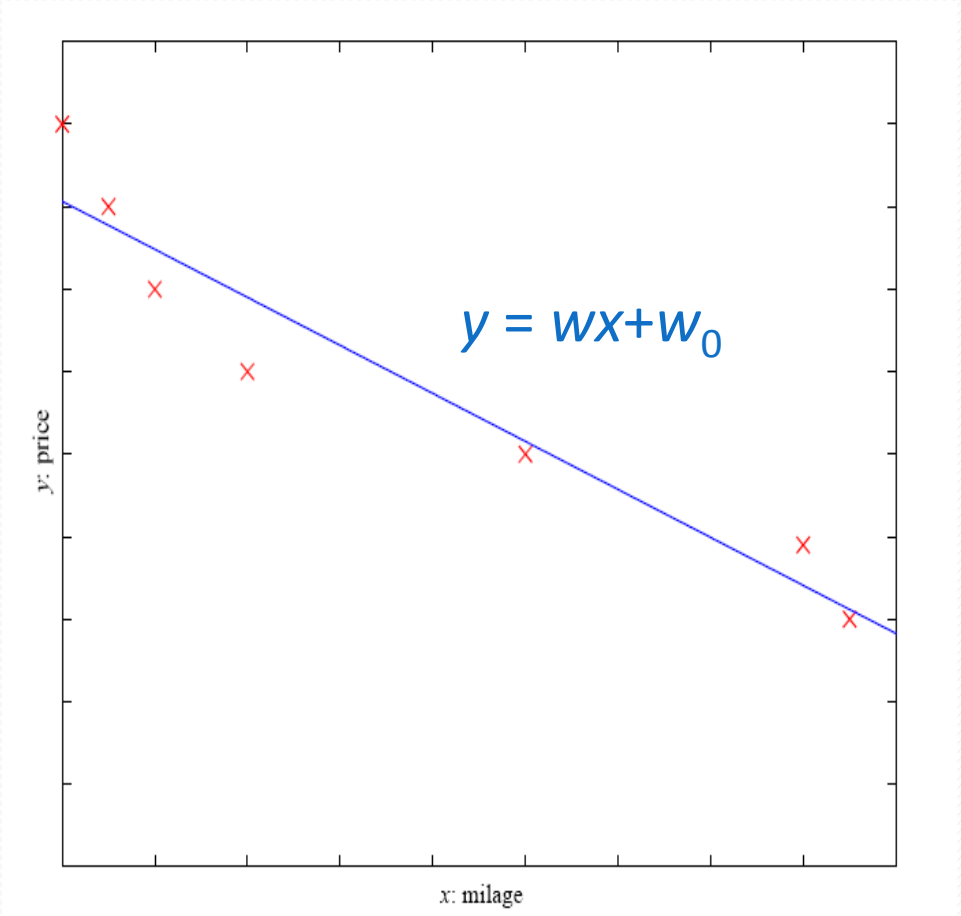
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g(x|\theta)$$

  $g(\ )$ model,

  θ parameters

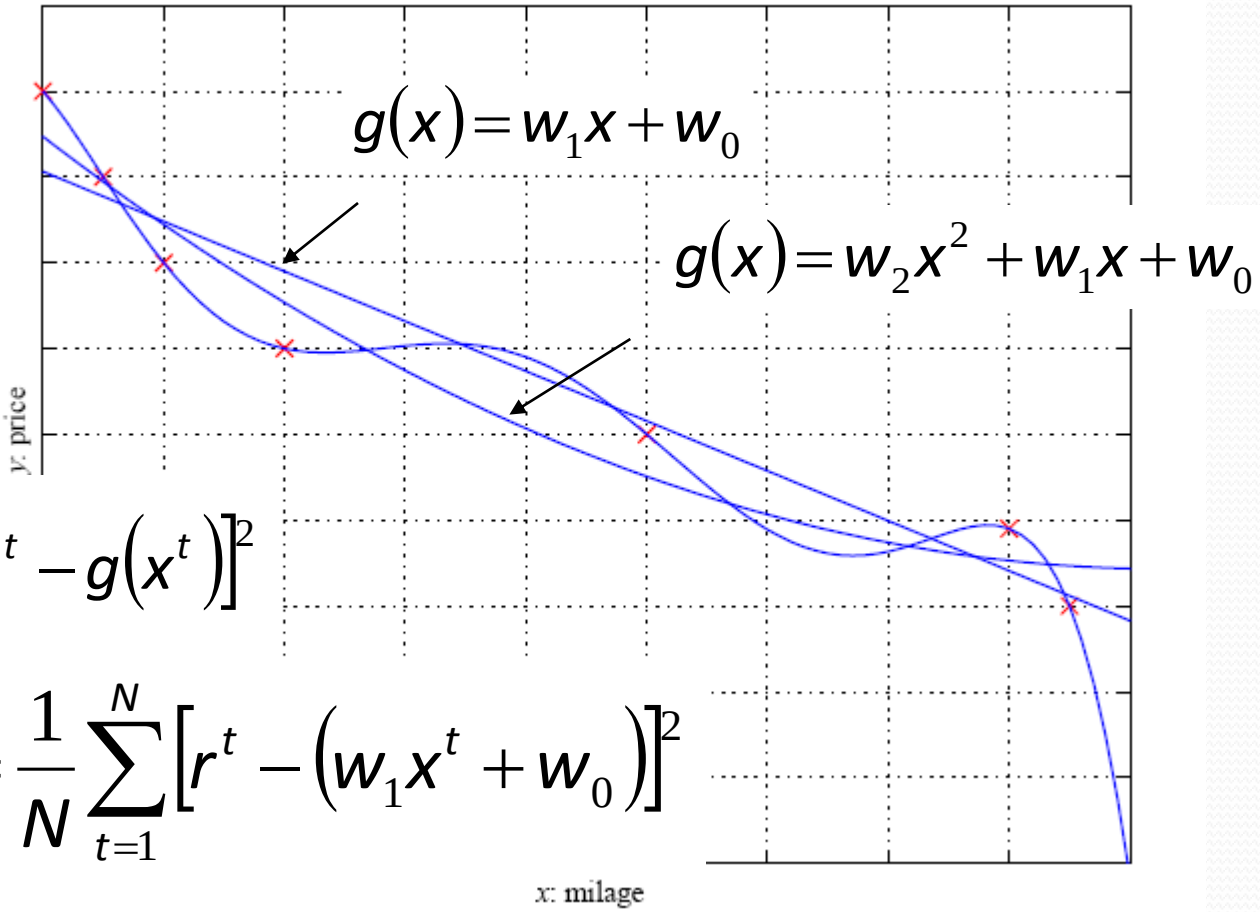


Regression

$$\mathcal{X} = \left\{x^t, r^t\right\}_{t=1}^{N}$$

$$r^t \in \Re$$

$$r^t = f\left(x^t\right) + \varepsilon$$

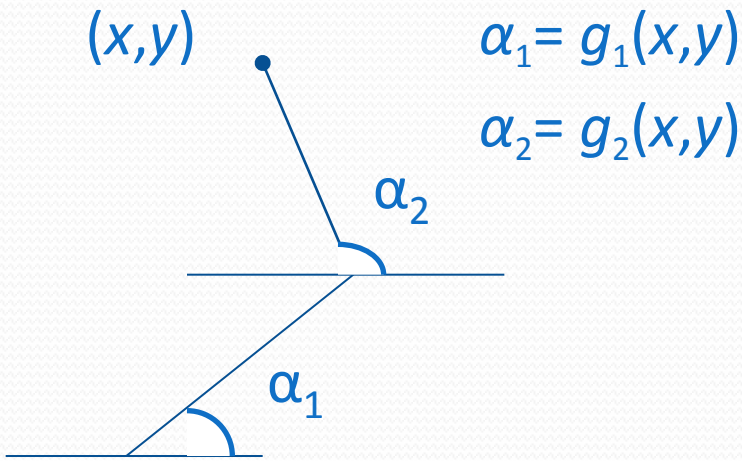


$$E(g \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\left[r^t - g\left(x^t\right)\right]^2$$

$$E(w_1, w_0 \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\left[r^t - \left(w_1 x^t + w_0\right)\right]^2$$

Regression Applications

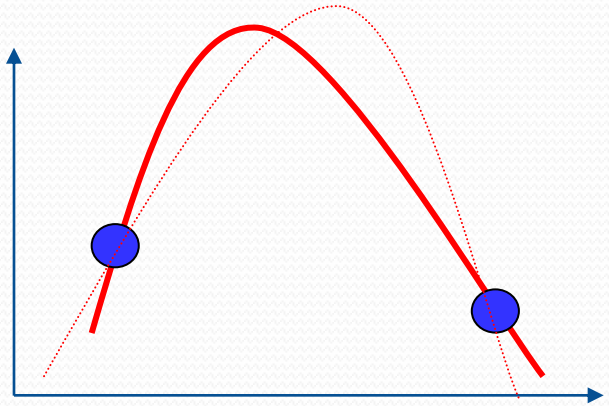
- Navigating a car: Angle of the steering wheel
- Kinematics of a robot arm



$\alpha_1 = g_1(x,y)$

$\alpha_2 = g_2(x,y)$

- Response surface design



Supervised Learning: Uses

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  1. Complexity of $\mathcal{H}$, $c(\mathcal{H})$,
  2. Training set size, N,
  3. Generalization error, E, on new data
- As N, $E\downarrow$
- As $c(\mathcal{H})$, first $E\downarrow$ and then E

Cross-Validation

- To estimate generalization error, we need data unseen during training. We split the data as
  - Training set (50%)
  - Validation set (25%)
  - Test (publication) set (25%)
- Resampling when the data set is small

Dimensions of a Supervised Learner

1. Model: $g(\mathbf{x}|\theta)$

2. Loss function: $$E(\theta|\mathcal{X}) = \sum_t L\left(r^t, g\left(\mathbf{x}^t|\theta\right)\right)$$

3. Optimization procedure:

$$\theta^* = \arg\min_\theta E(\theta|\mathcal{X})$$

Model Selection & Generalization

- Learning is an ill-posed problem; data is not sufficient to find a unique solution
- The need for inductive bias, assumptions about $\mathcal{H}$
- Generalization: How well a model performs on new data
- Overfitting: $\mathcal{H}$ more complex than C or f
- Underfitting: $\mathcal{H}$ less complex than C or f

Bayesian Decision Theory

Estimating Probabilities

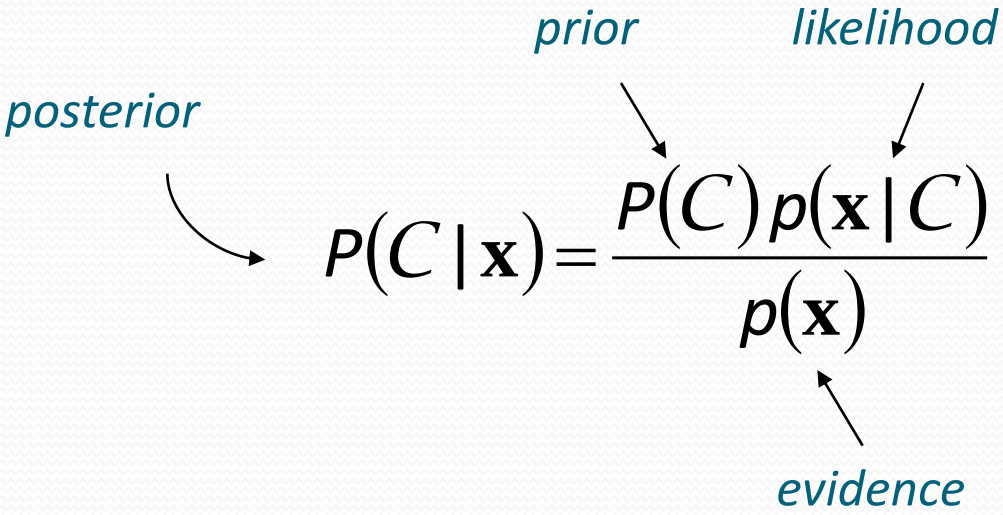
- Family car or not: Inputs are engine power and price.
  Output is family-car vs not-family-car.
- Input: $\boldsymbol{x} = [x_1, x_2]^T$,Output: $C \in \{0,1\}$
- Prediction:

$$\text{choose} \begin{cases} C = 1 \text{ if } P(C = 1 \mid x_1, x_2) > 0.5 \\ C = 0 \text{ otherwise} \end{cases}$$

or

$$\text{choose} \begin{cases} C = 1 \text{ if } P(C = 1 \mid x_1, x_2) > P(C = 0 \mid x_1, x_2) \\ C = 0 \text{ otherwise} \end{cases}$$

Bayes' Rule



$$P(C=0)+P(C=1)=1$$

$$p(\mathbf{x})=p(\mathbf{x}|C=1)P(C=1)+p(\mathbf{x}|C=0)P(C=0)$$

$$p(C=0|\mathbf{x})+P(C=1|\mathbf{x})=1$$

Bayes' Rule: $K>2$ Classes

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^{K} P(C_i) = 1$$

$$\text{choose } C_i \text{ if } P(C_i \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$$

Losses and Risks

- Actions: α_i
- Loss of α_i when the state is C_k : λ_{ik}
- Expected risk (Duda and Hart, 1973)

$$R(\alpha_i \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x})$$

$$\text{choose } \alpha_i \text{ if } R(\alpha_i \mid \mathbf{x}) = \min_k R(\alpha_k \mid \mathbf{x})$$

Losses and Risks: 0/1 Loss

$$\lambda_{ik} = \begin{cases} 0 \text{ if } i = k \\ 1 \text{ if } i \neq k \end{cases}$$

$$R(\alpha_i \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x})$$

$$= \sum_{k \neq i} P(C_k \mid \mathbf{x})$$

$$= 1 - P(C_i \mid \mathbf{x})$$

For minimum risk, choose the most probable class

Action of "reject"

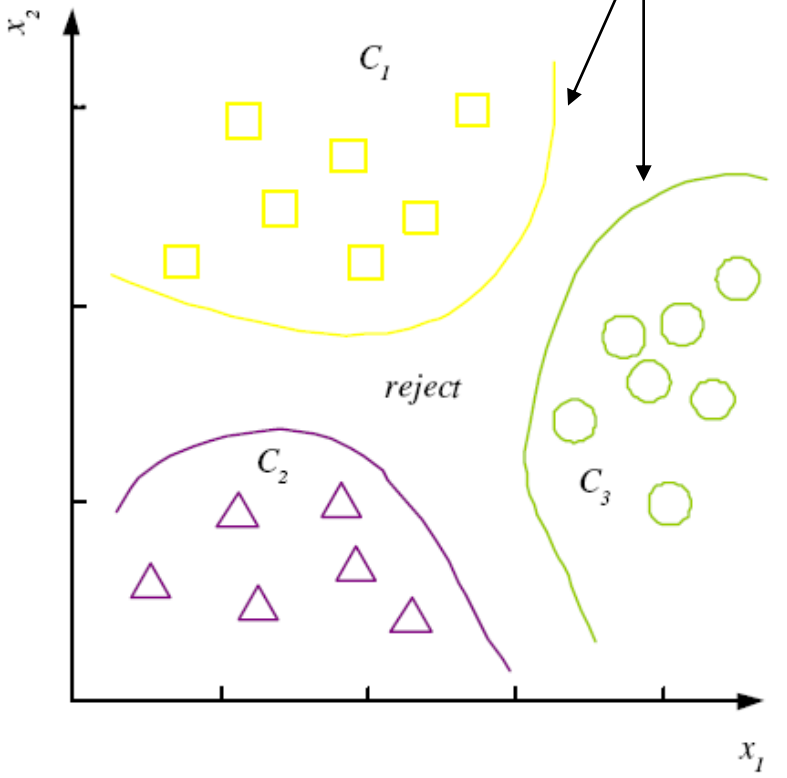
Misclassification costs may not be symmetric

Discriminant Functions

$$\text{choose } C_i \text{ if } g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})$$

$$g_i(\mathbf{x}), i = 1, \ldots, K$$

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) \\ P(C_i \mid \mathbf{x}) \\ p(\mathbf{x} \mid C_i) P(C_i) \end{cases}$$



K decision regions $\mathcal{R}_1, \ldots, \mathcal{R}_K$

$$\mathcal{R}_i = \{\mathbf{x} \mid g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})\}$$

Parametric Estimation of Densities

- $\mathcal{X} = \{ x^t \}_t$ where $x^t \sim p(x)$
- Parametric estimation:

  Assume a form for $p(x \mid \theta)$ and estimate θ, its sufficient statistics, using X

  e.g., N (μ, σ^2) where $\theta = \{ \mu, \sigma^2 \}$

Maximum Likelihood Estimation

- Likelihood of θ given the sample $\mathcal{X}$

$$l(\theta|\mathcal{X}) = p(\mathcal{X}|\theta) = \prod_t p(x^t|\theta)$$

- Log likelihood

$$L(\theta|\mathcal{X}) = \log l(\theta|\mathcal{X}) = \sum_t \log p(x^t|\theta)$$

- Maximum likelihood estimator (MLE)

$$\theta^* = \text{argmax}_\theta L(\theta|\mathcal{X})$$

Bayes' Estimator

- Treat θ as a random var with prior $p(\theta)$
- Bayes' rule: $p(\theta|\mathcal{X}) = p(\mathcal{X}|\theta)\, p(\theta) / p(\mathcal{X})$
- Full: $p(x|\mathcal{X}) = \int p(x|\theta)\, p(\theta|\mathcal{X})\, d\theta$
- Maximum a Posteriori (MAP): $\theta_{MAP} = \text{argmax}_\theta\, p(\theta|\mathcal{X})$
- Maximum Likelihood (ML): $\theta_{ML} = \text{argmax}_\theta\, p(\mathcal{X}|\theta)$
- Bayes': $\theta_{Bayes'} = E[\theta|\mathcal{X}] = \int \theta\, p(\theta|\mathcal{X})\, d\theta$

Parametric Classification

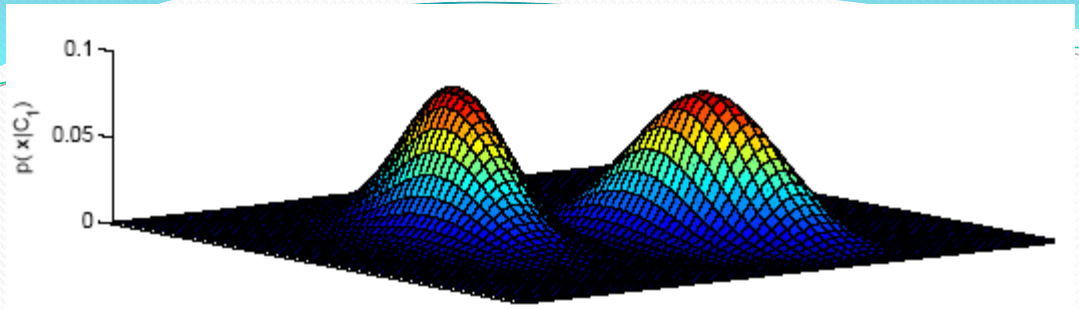
- If $p(\mathbf{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$p(\mathbf{x}|C_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right]$$

- Discriminant functions

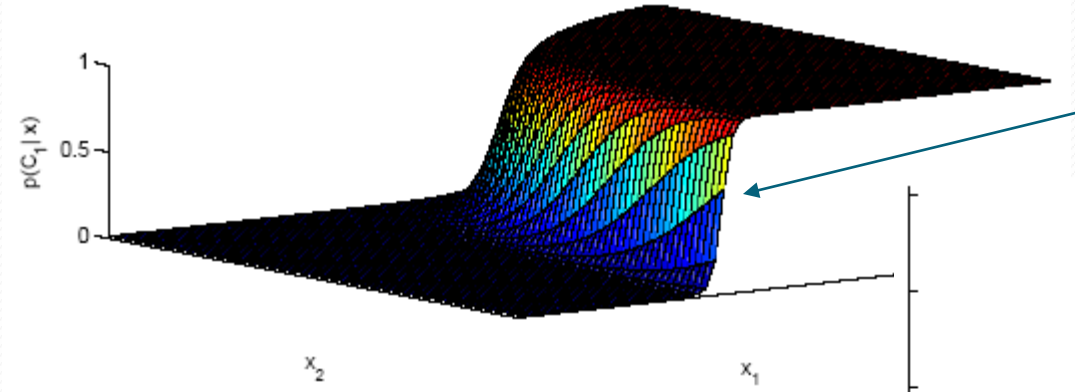
$$g_i(\mathbf{x}) = \log p(\mathbf{x}|C_i) + \log P(C_i)$$

$$= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i) + \log P(C_i)$$

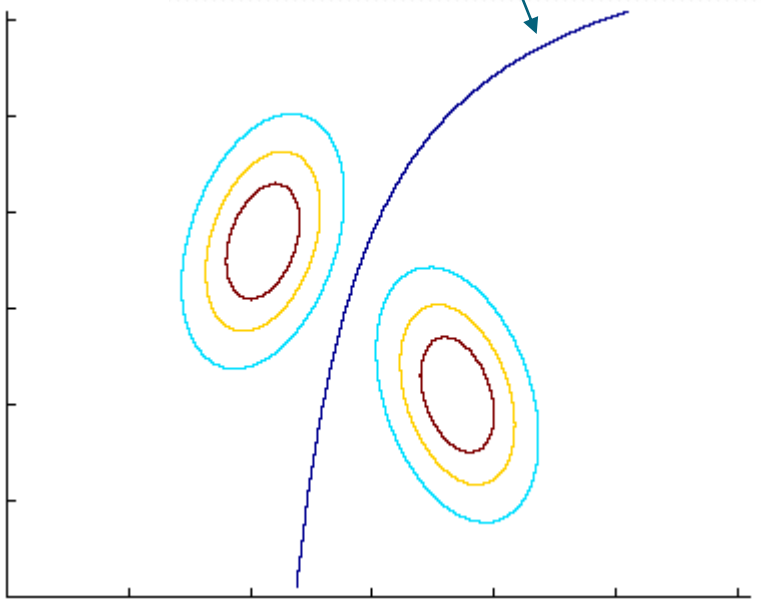


likelihoods

discriminant:
$P(C_1|\boldsymbol{x}) = 0.5$



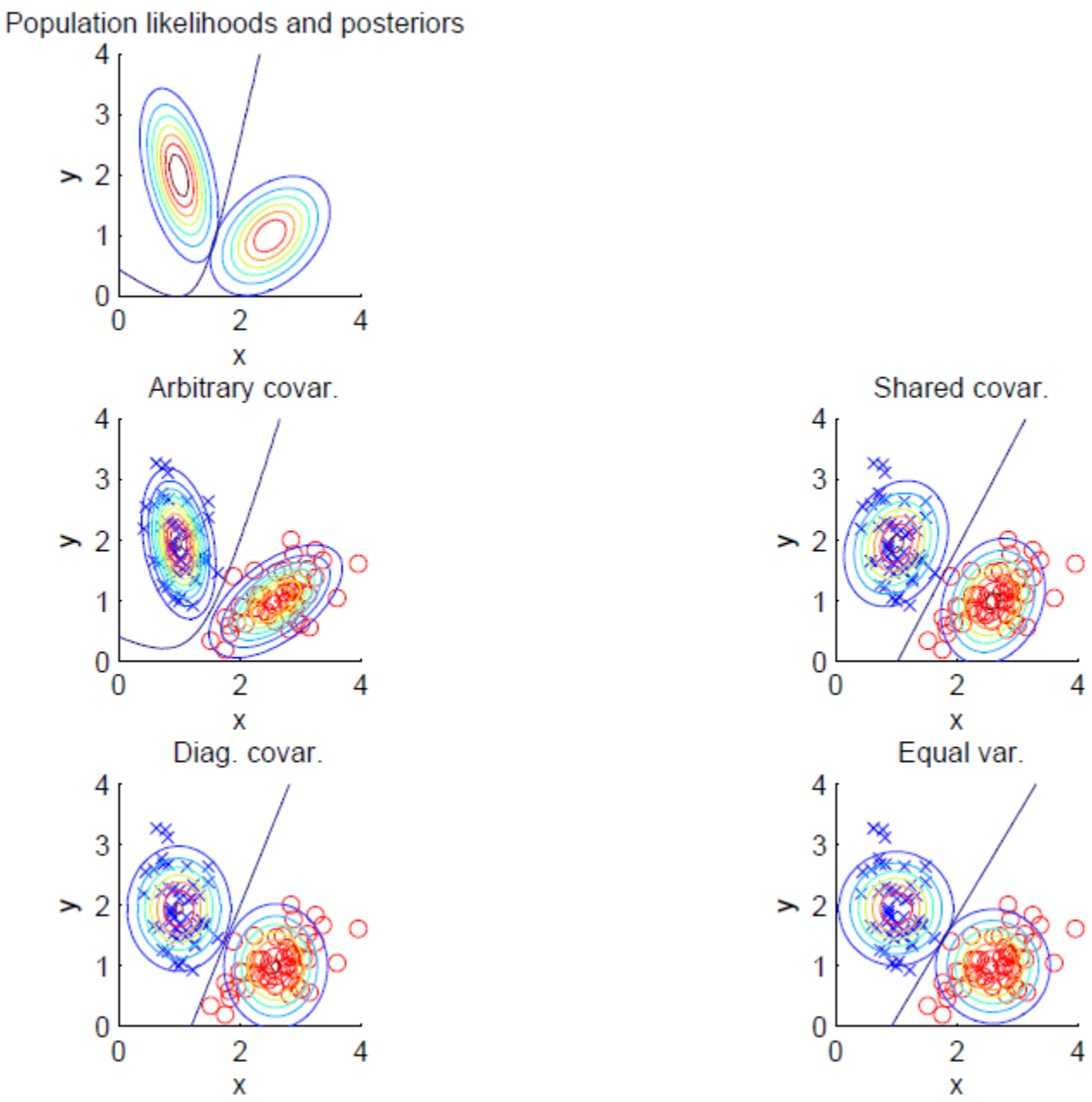
posterior for C_1



Model Selection

Assumption	*Covariance matrix*	*No of parameters*
Shared, Hyperspheric	$\mathbf{S}_i=\mathbf{S}=s^2\mathbf{I}$	1
Shared, Axis-aligned	$\mathbf{S}_i=\mathbf{S}$, with $s_{ij}=0$	d
Shared, Hyperellipsoidal	$\mathbf{S}_i=\mathbf{S}$	$d(d+1)/2$
Different, Hyperellipsoidal	$\mathbf{S}_i$	$K\,d(d+1)/2$

- As we increase complexity (less restricted **S**), bias decreases and variance increases
- Assume simple models (allow some bias) to control variance (regularization)


Population likelihoods and posteriors
Arbitrary covar.
Shared covar.
Diag. covar.
Equal var.
x
y
0
1
2
3
4

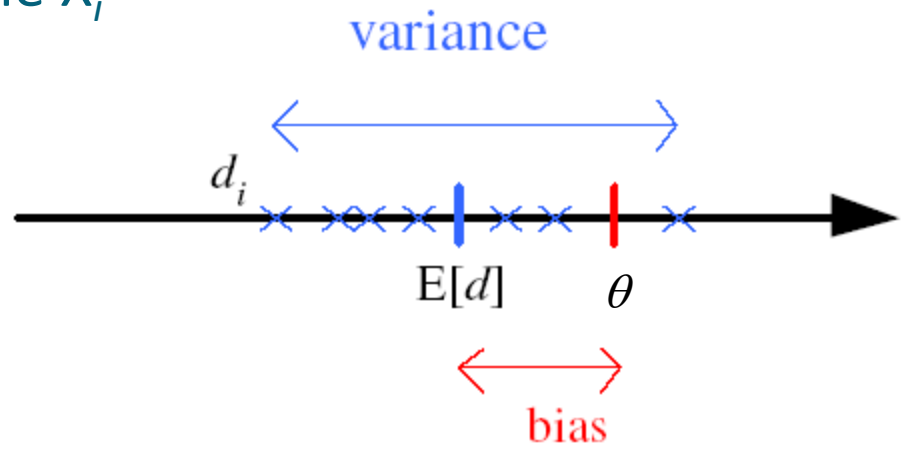
Bias and Variance

Unknown parameter θ

Estimator $d_i = d(X_i)$ on sample X_i

Bias: $b_\theta(d) = E[d] - \theta$

Variance: $E[(d - E[d])^2]$

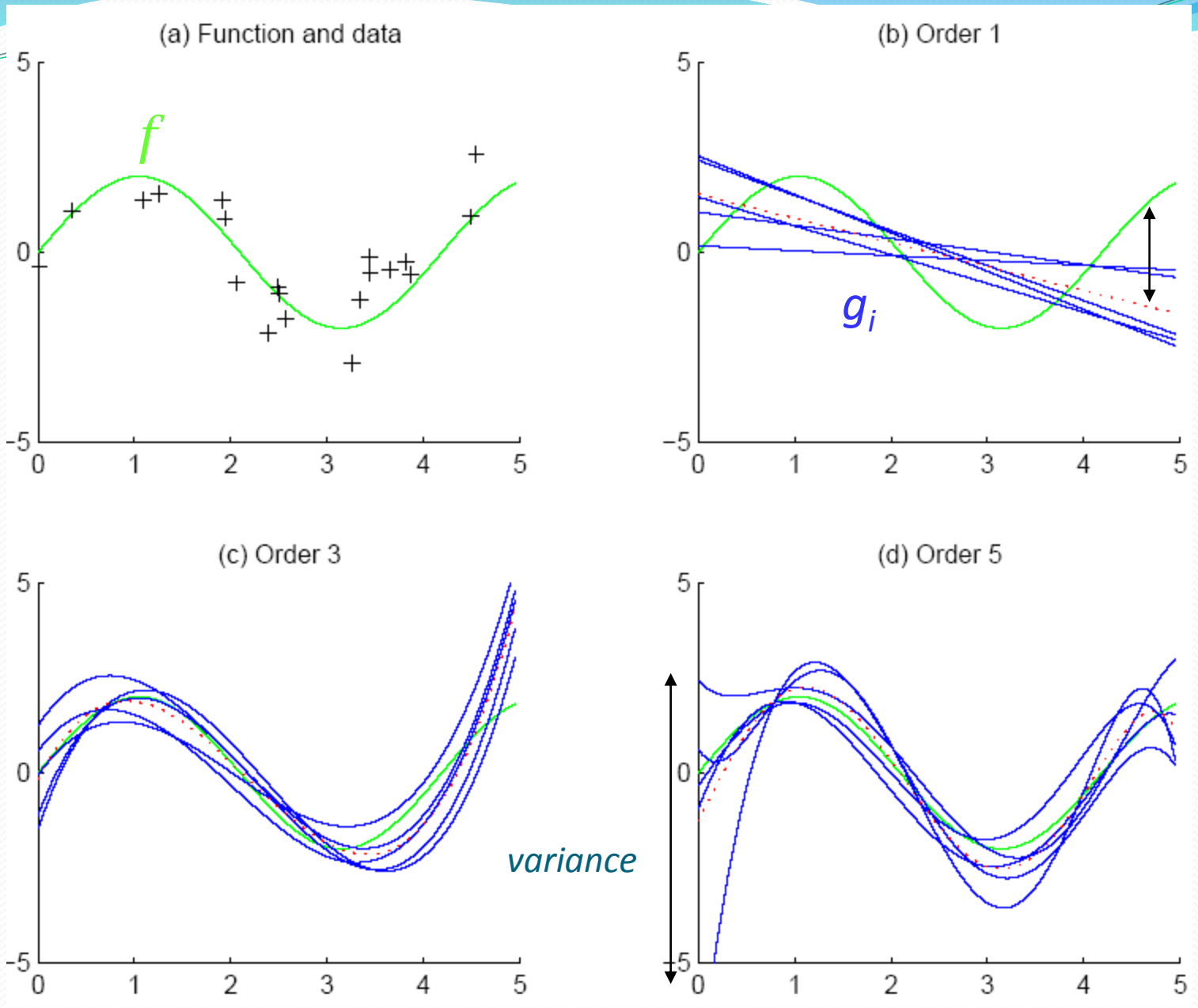


Mean square error:

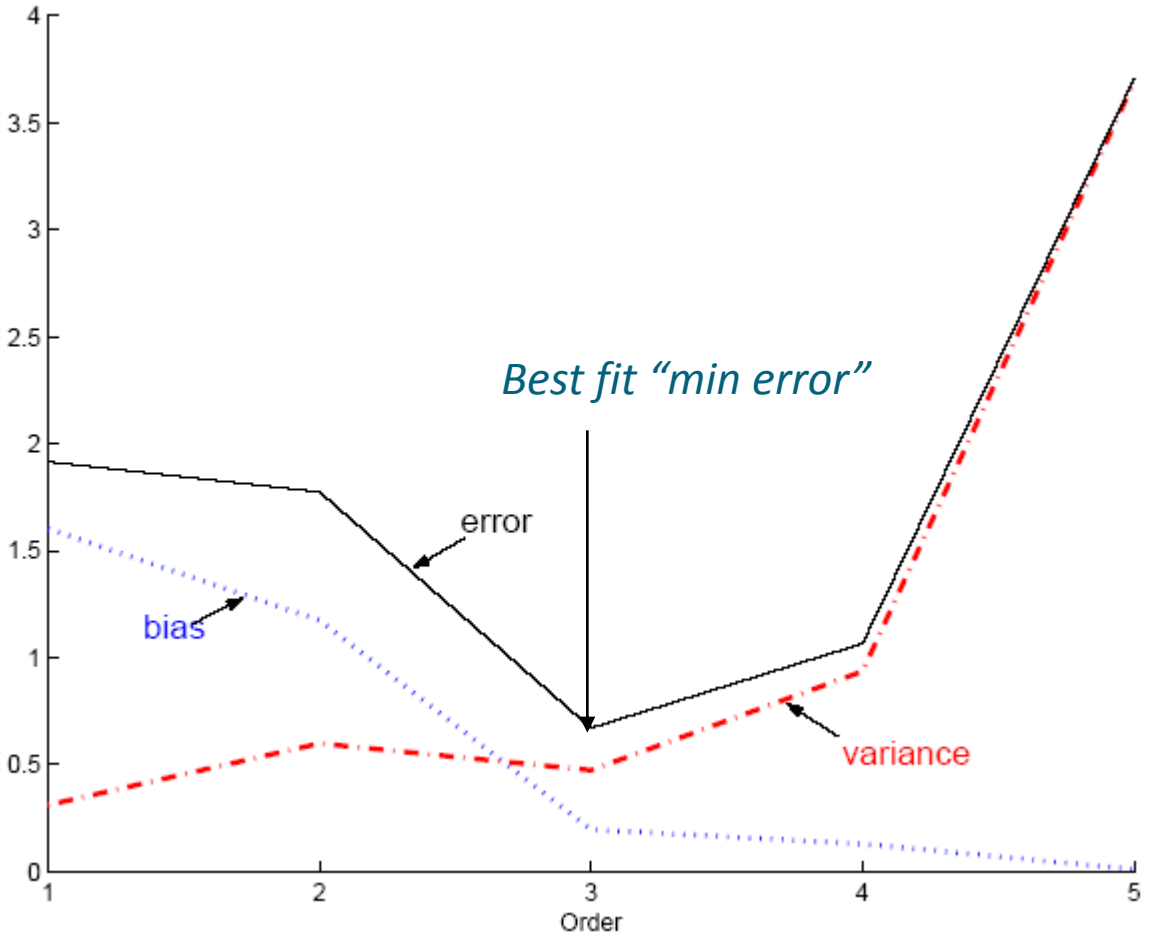
$r(d,\theta) = E[(d-\theta)^2]$

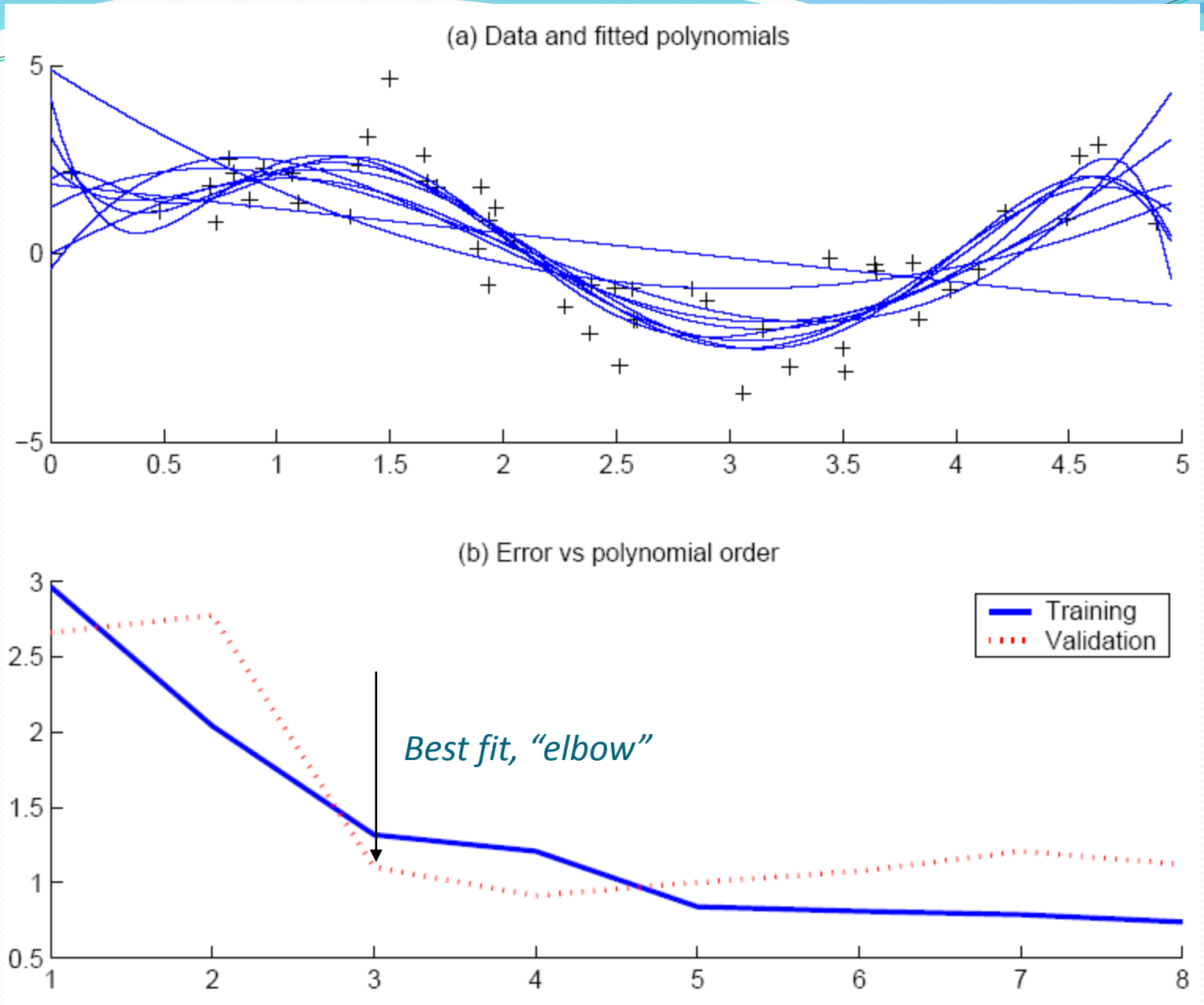
$= (E[d] - \theta)^2 + E[(d - E[d])^2]$

$=$ Bias^2 + Variance


(a) Function and data
f
(b) Order 1
f
bias
g_i
$\bar{g}$
(c) Order 3
variance
(d) Order 5

Polynomial Regression




(a) Data and fitted polynomials
5
0
−5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
(b) Error vs polynomial order
3
2.5
2
1.5
1
0.5
1
2
3
4
5
6
7
8
Training
Validation
Best fit, “elbow”

Model Selection

- Cross-validation: Measure generalization accuracy by testing on data unused during training
- Regularization: Penalize complex models

  E’=error on data + λ model complexity
- Minimum description length (MDL): Kolmogorov complexity, shortest description of data
- Structural risk minimization (SRM)

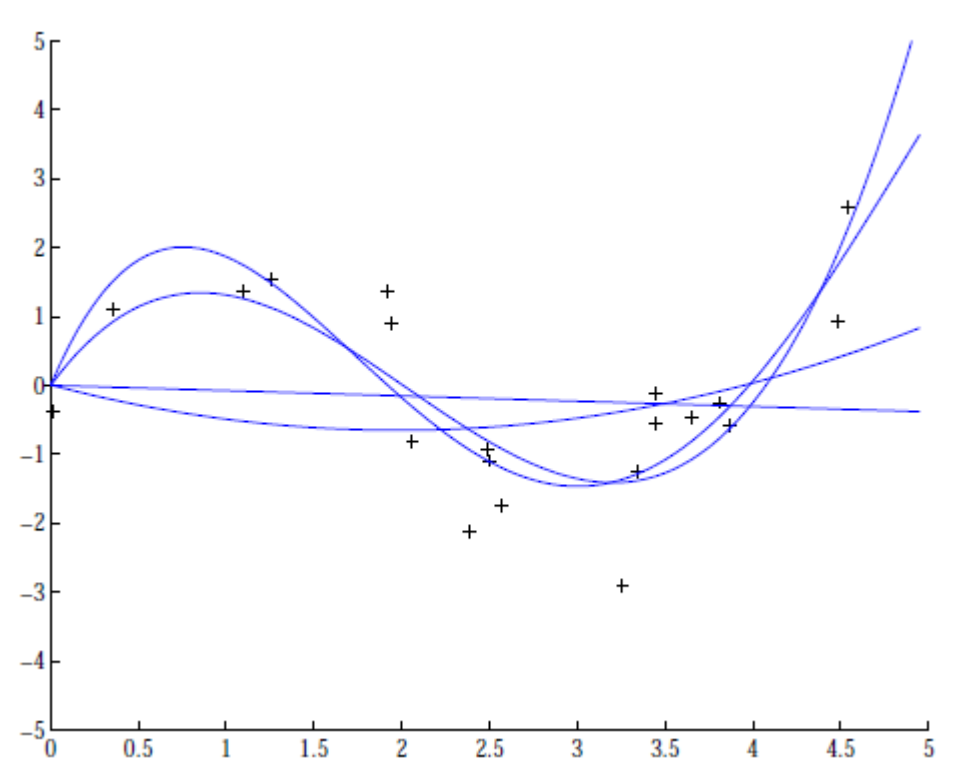
Bayesian Model Selection

- Prior on models, $p(\text{model})$

$$p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})\,p(\text{model})}{p(\text{data})}$$

- Regularization, when prior favors simpler models
- Bayes, MAP of the posterior, $p(\text{model}|\text{data})$
- Average over a number of models with high posterior (voting, ensembles: see Part II)

Regression example



Coefficients increase in magnitude as order increases:

1: [-0.0769, 0.0016]

2: [0.1682, -0.6657, 0.0080]

3: [0.4238, -2.5778, 3.4675, -0.0002

4: [-0.1093, 1.4356, -5.5007, 6.0454, -0.0019]

$$\textit{regularization}: E(\mathbf{w} \mid \mathcal{X}) = \frac{1}{2}\sum_{t=1}^{N}\left[r^t - g\left(x^t \mid \mathbf{w}\right)\right]^2 + \lambda \sum_i w_i^2$$

Nonparametric Estimation

- Parametric (single global model), semiparametric (small number of local models)
- Nonparametric: Similar inputs have similar outputs
- Functions (pdf, discriminant, regression) change smoothly
- Keep the training data;“let the data speak for itself”
- Given $\boldsymbol{x}$, find a small number of closest training instances and interpolate from these
- Aka lazy/memory-based/case-based/instance-based learning

Density Estimation

- Given the training set $X=\{x^t\}_t$ drawn iid from $p(x)$
- Divide data into bins of size h
- Histogram:

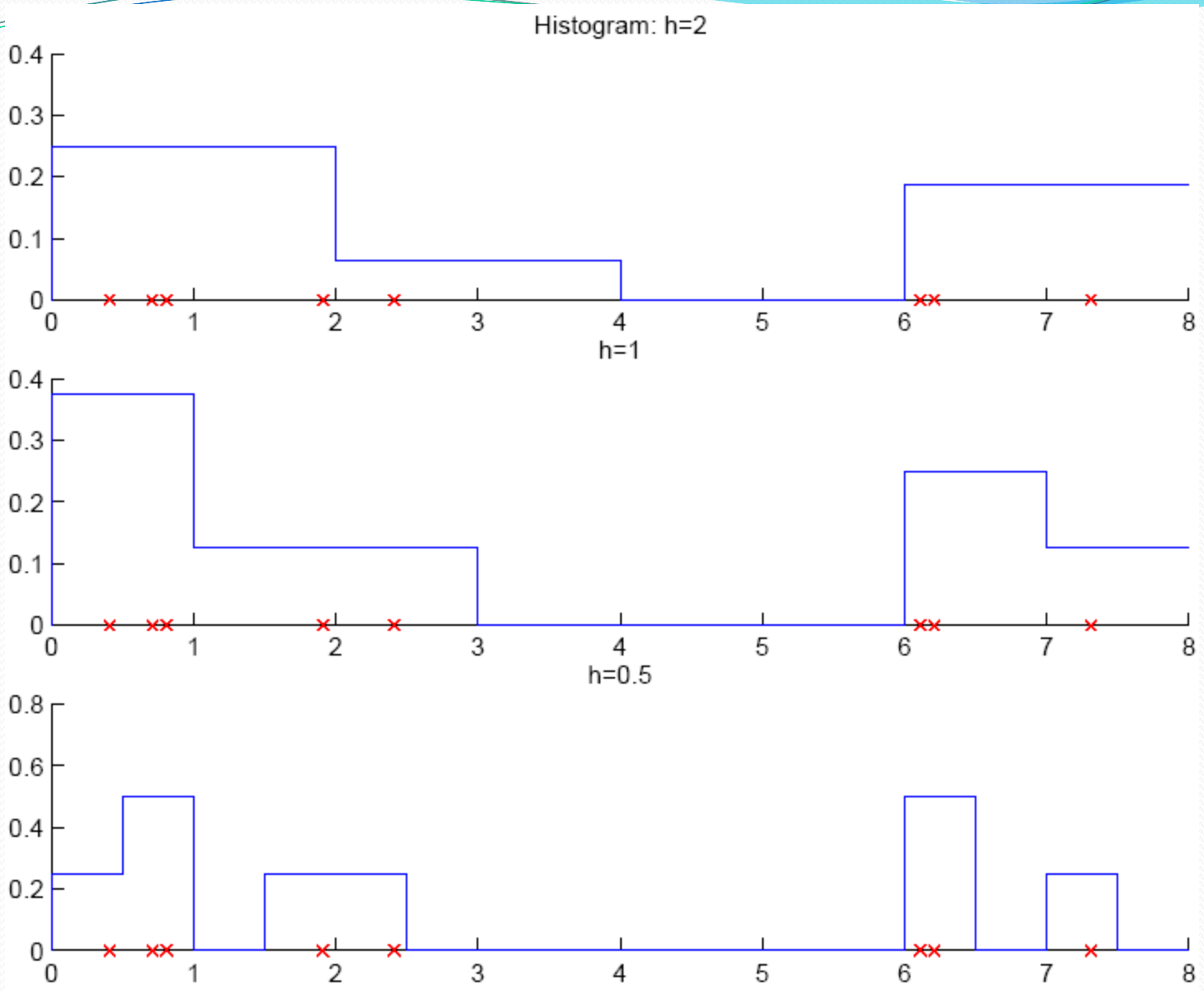
$$\hat{p}(x)=\frac{\#\left\{x^t \text{ in the same bin as } x\right\}}{Nh}$$

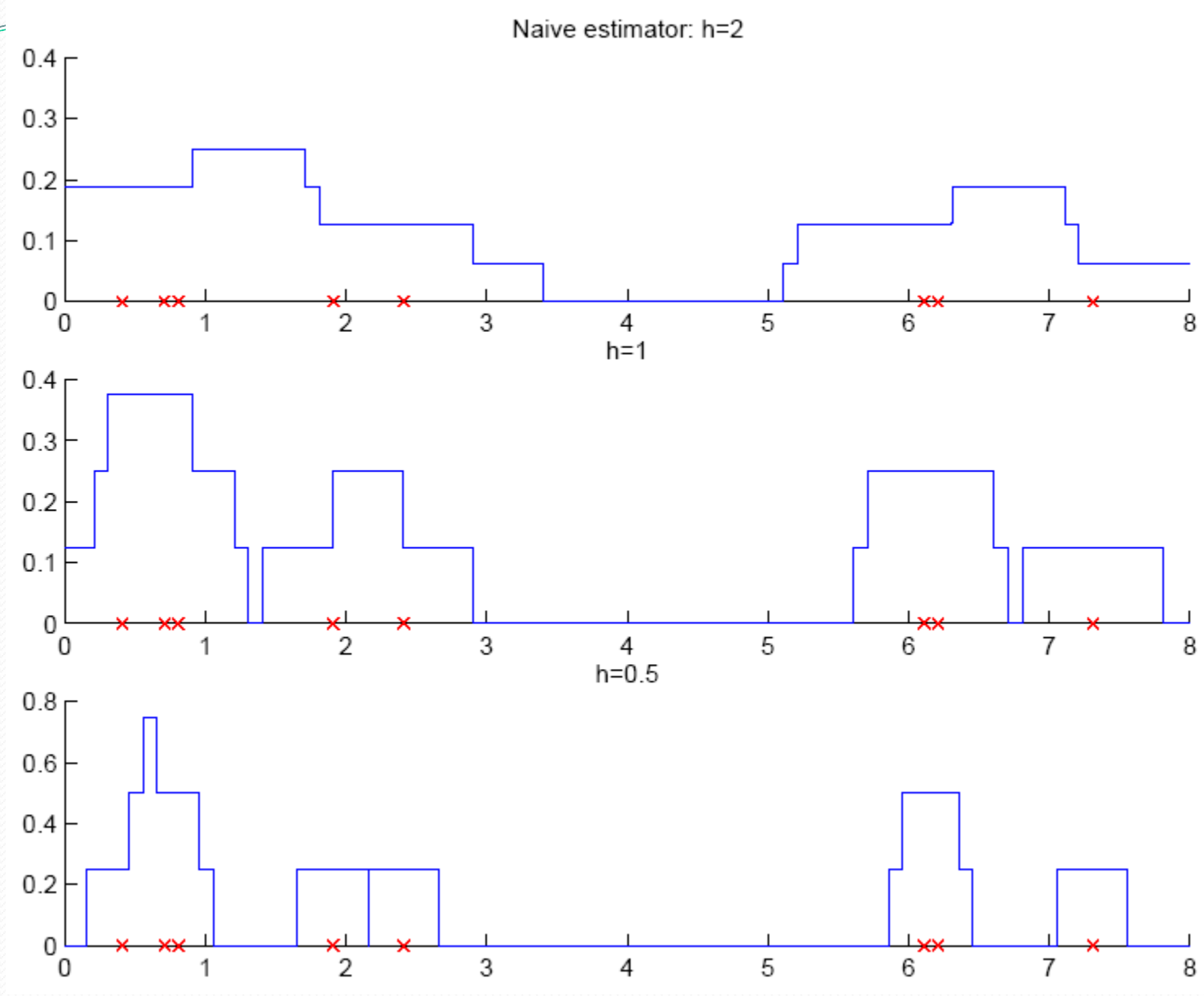
- Naive estimator:

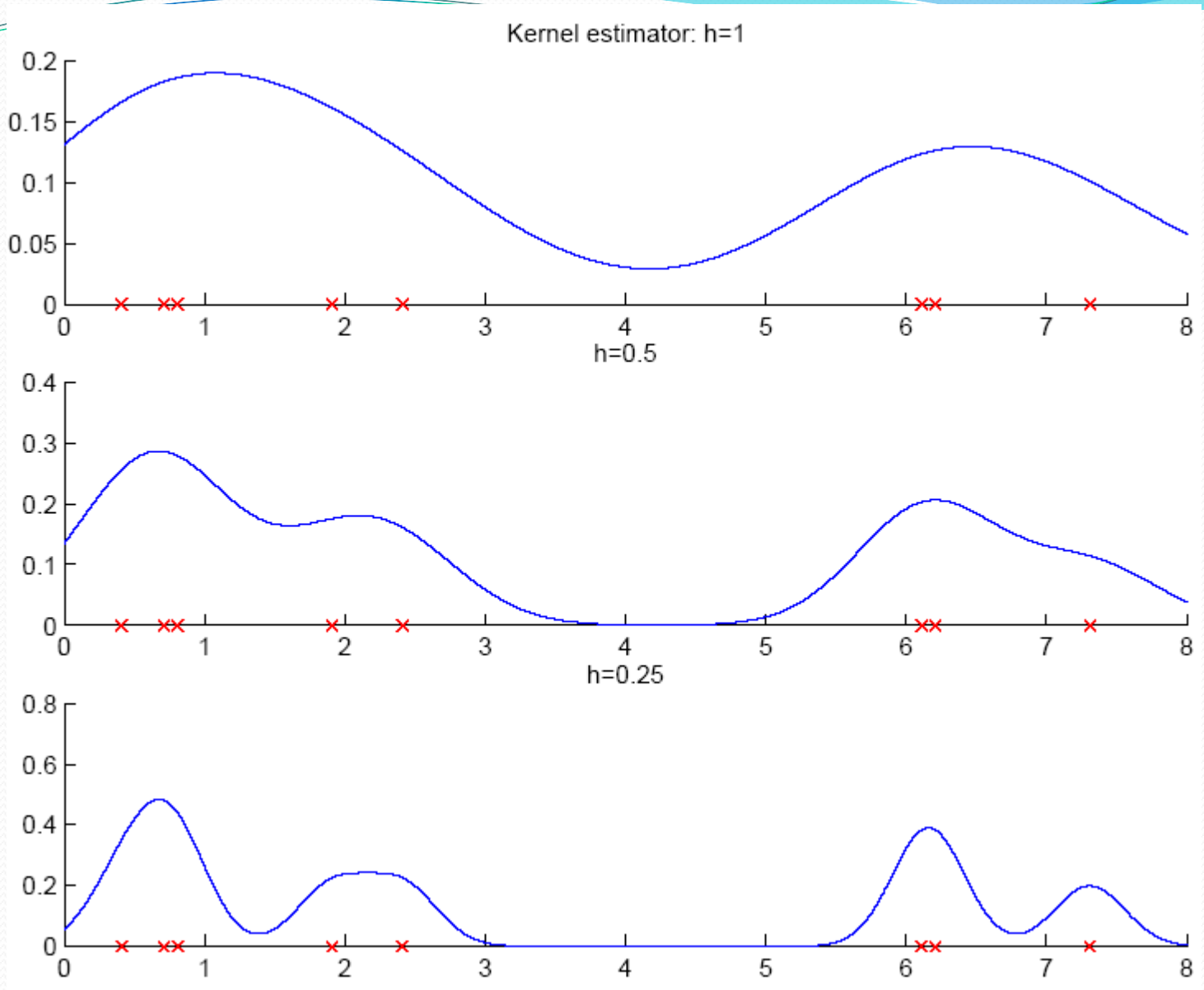
$$\hat{p}(x)=\frac{\#\left\{x-h<x^t\le x+h\right\}}{2Nh}$$

or

$$\hat{p}(x)=\frac{1}{Nh}\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right) \quad w(u)=\begin{cases}1/2 & \text{if } |u|<1\\ 0 & \text{otherwise}\end{cases}$$


Histogram: h=2
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=1
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=0.5
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8


Naive estimator: h=2
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=1
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=0.5
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8

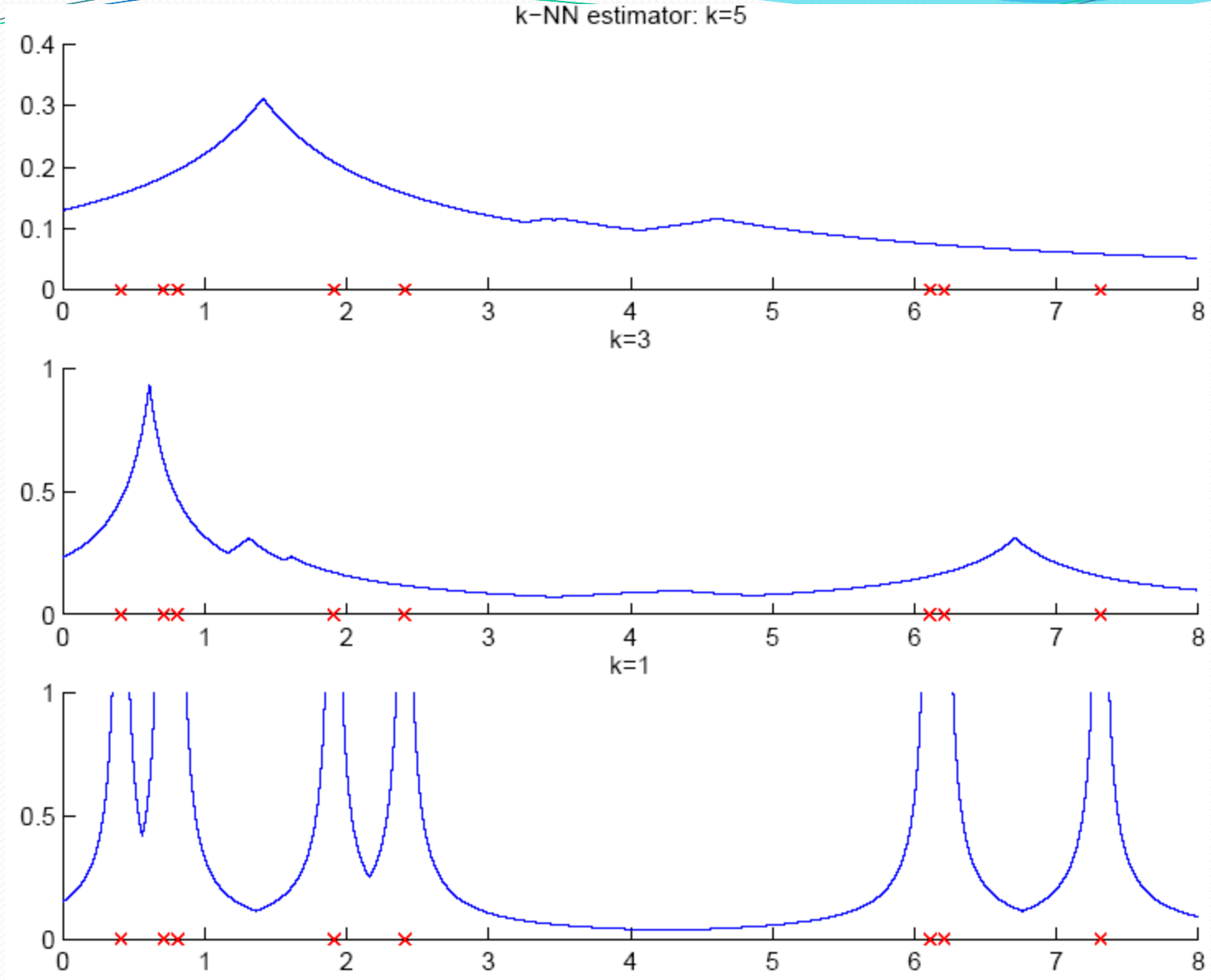

Kernel estimator: h=1
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8
h=0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=0.25
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8

k-Nearest Neighbor Estimator

- Instead of fixing bin width *h* and counting the number of instances, fix the instances (neighbors) *k* and check bin width

$$\hat{p}(x) = \frac{k}{2Nd_k(x)}$$

$d_k(x)$, distance to kth closest instance to x


k−NN estimator: k=5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
k=3
1
0.5
0
0 1 2 3 4 5 6 7 8
k=1
1
0.5
0
0 1 2 3 4 5 6 7 8

Multivariate Data

- Kernel density estimator

$$\hat{p}(\mathbf{x}) = \frac{1}{Nh^d} \sum_{t=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}^t}{h} \right)$$

Multivariate Gaussian kernel

spheric

$$K(\mathbf{u}) = \left(\frac{1}{\sqrt{2\pi}} \right)^d \exp\left[-\frac{\|\mathbf{u}\|^2}{2} \right]$$

ellipsoid

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{d/2} |\mathbf{S}|^{1/2}} \exp\left[-\frac{1}{2} \mathbf{u}^T \mathbf{S}^{-1} \mathbf{u} \right]$$

Nonparametric Classification

- Estimate $p(\boldsymbol{x}|C_i)$ and use Bayes' rule
- Kernel estimator

$$\hat{p}(\mathbf{x}|C_i)=\frac{1}{N_i h^d}\sum_{t=1}^{N}K\left(\frac{\mathbf{x}-\mathbf{x}^t}{h}\right)r_i^t \quad \hat{P}(C_i)=\frac{N_i}{N}$$

$$g_i(\mathbf{x})=\hat{p}(\mathbf{x}|C_i)\hat{P}(C_i)=\frac{1}{Nh^d}\sum_{t=1}^{N}K\left(\frac{\mathbf{x}-\mathbf{x}^t}{h}\right)r_i^t$$

- k-NN estimator

$$\hat{p}(\mathbf{x}|C_i)=\frac{k_i}{N_i V^k(\mathbf{x})} \quad \hat{P}(C_i|\mathbf{x})=\frac{\hat{p}(\mathbf{x}|C_i)\hat{P}(C_i)}{\hat{p}(\mathbf{x})}=\frac{k_i}{k}$$

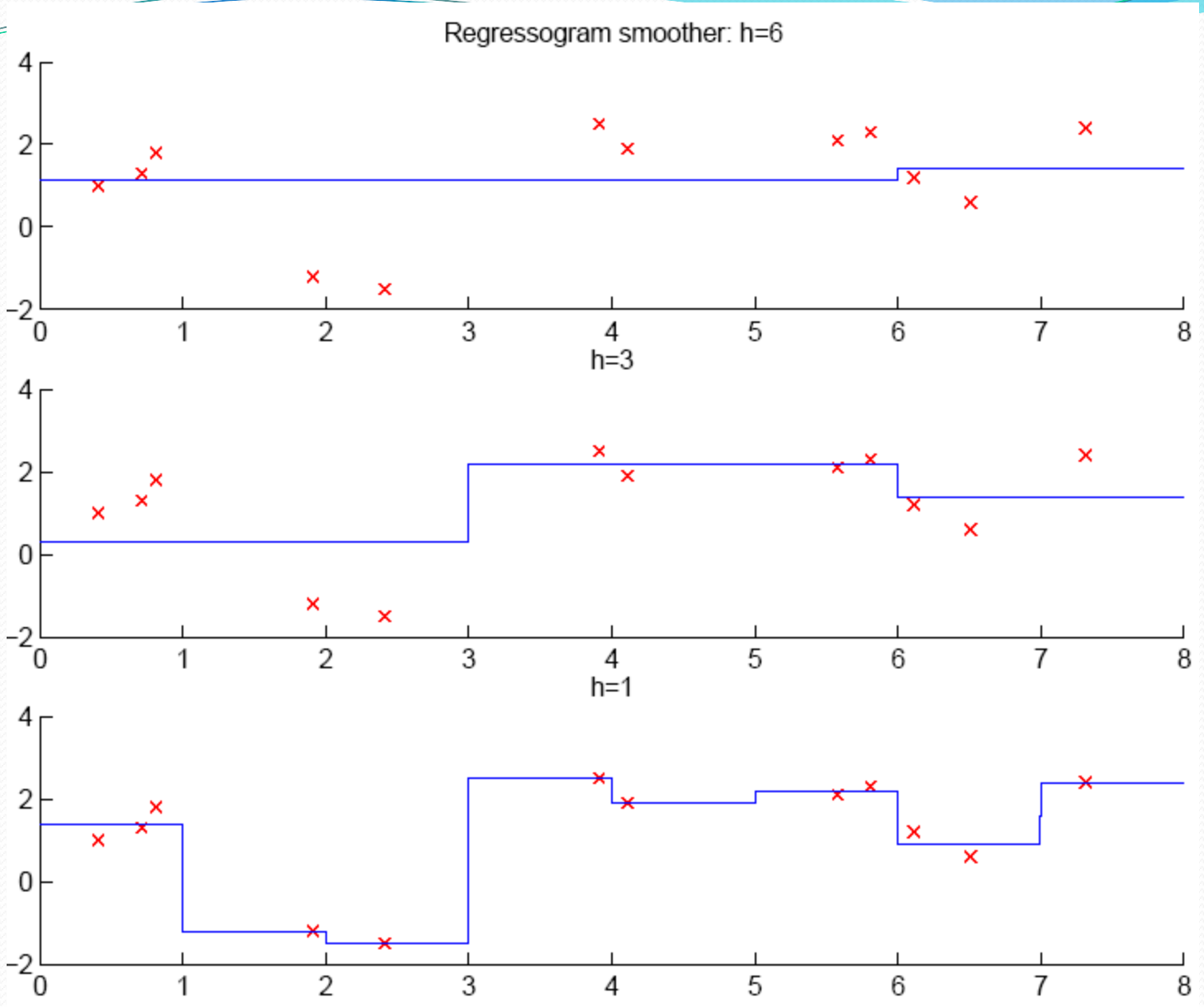
Nonparametric Regression

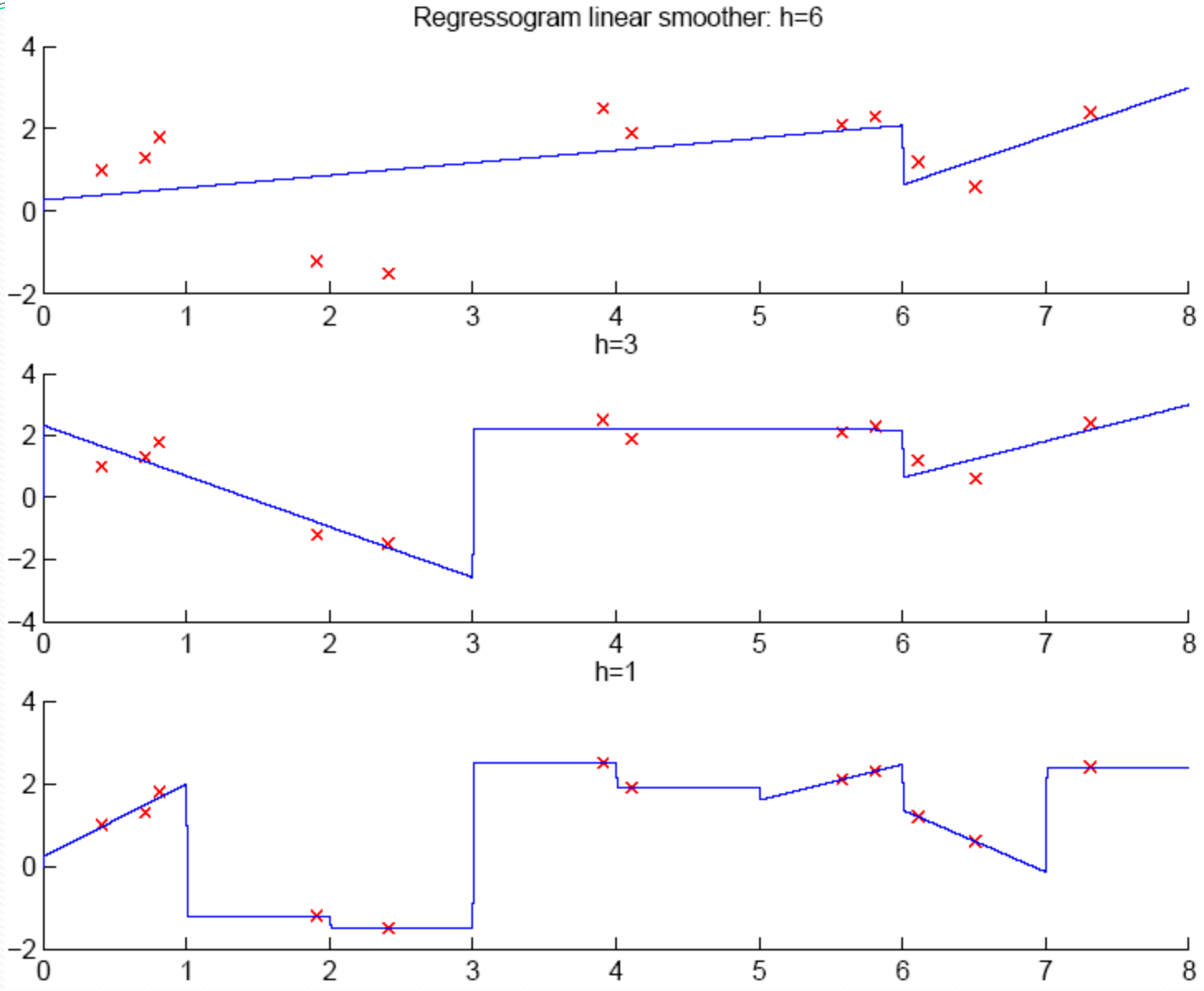
- Aka smoothing models
- Regressogram

$$\hat{g}(x)=\frac{\sum_{t=1}^{N} b\left(x, x^{t}\right) r^{t}}{\sum_{t=1}^{N} b\left(x, x^{t}\right)}$$

where

$$b\left(x, x^{t}\right)=\begin{cases}1 & \text{if } x^{t} \text{ is in the same bin with } x\\ 0 & \text{otherwise}\end{cases}$$


Regressogram smoother: h=6
h=3
h=1


Regressogram linear smoother: h=6
h=3
h=1

Running Mean/Kernel Smoother

- Running mean smoother

$$\hat{g}(x)=\frac{\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right) r^t}{\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right)}$$

where

$$w(u)=\begin{cases}1 & \text{if } |u|<1\\ 0 & \text{otherwise}\end{cases}$$

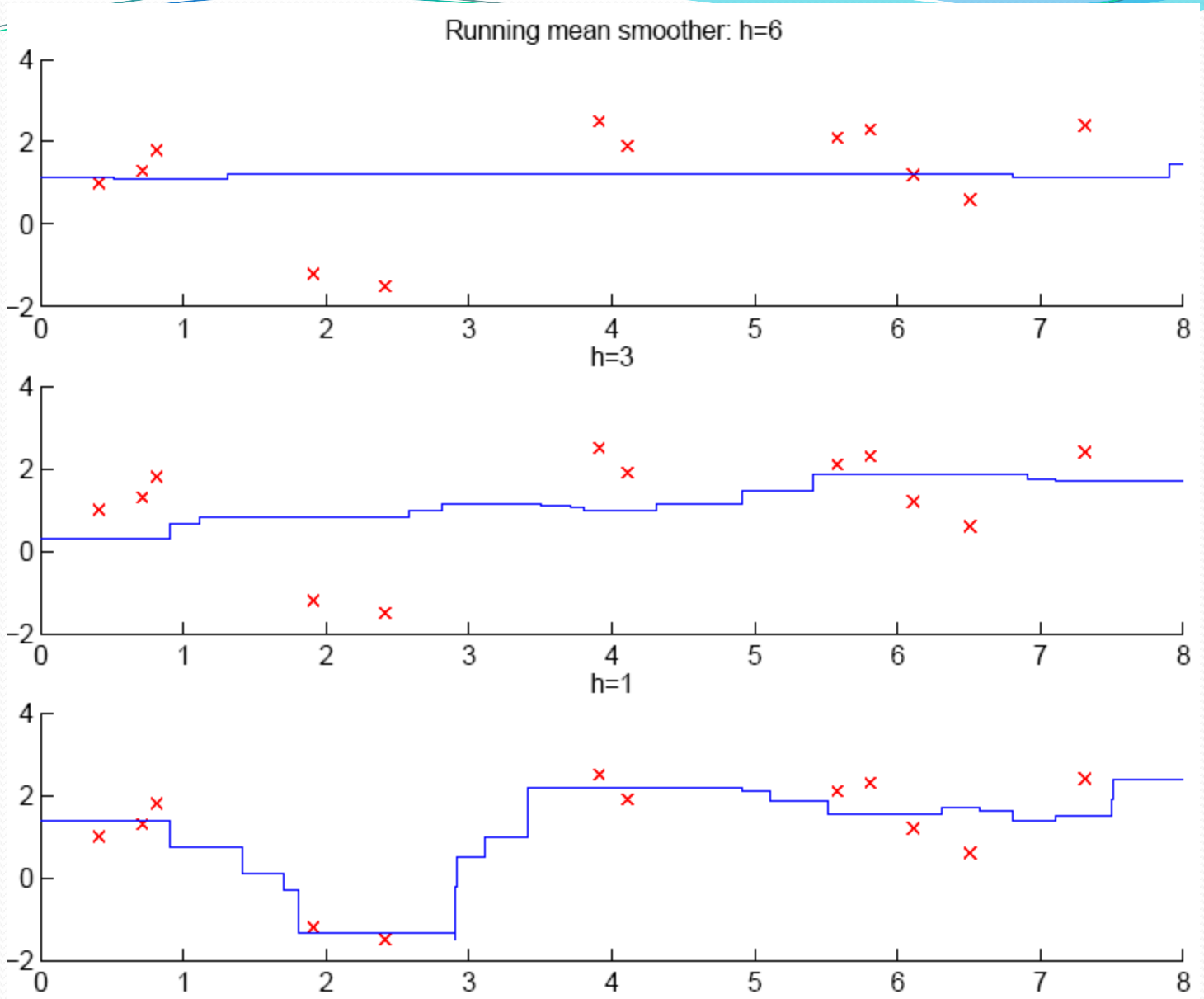
- Running line smoother

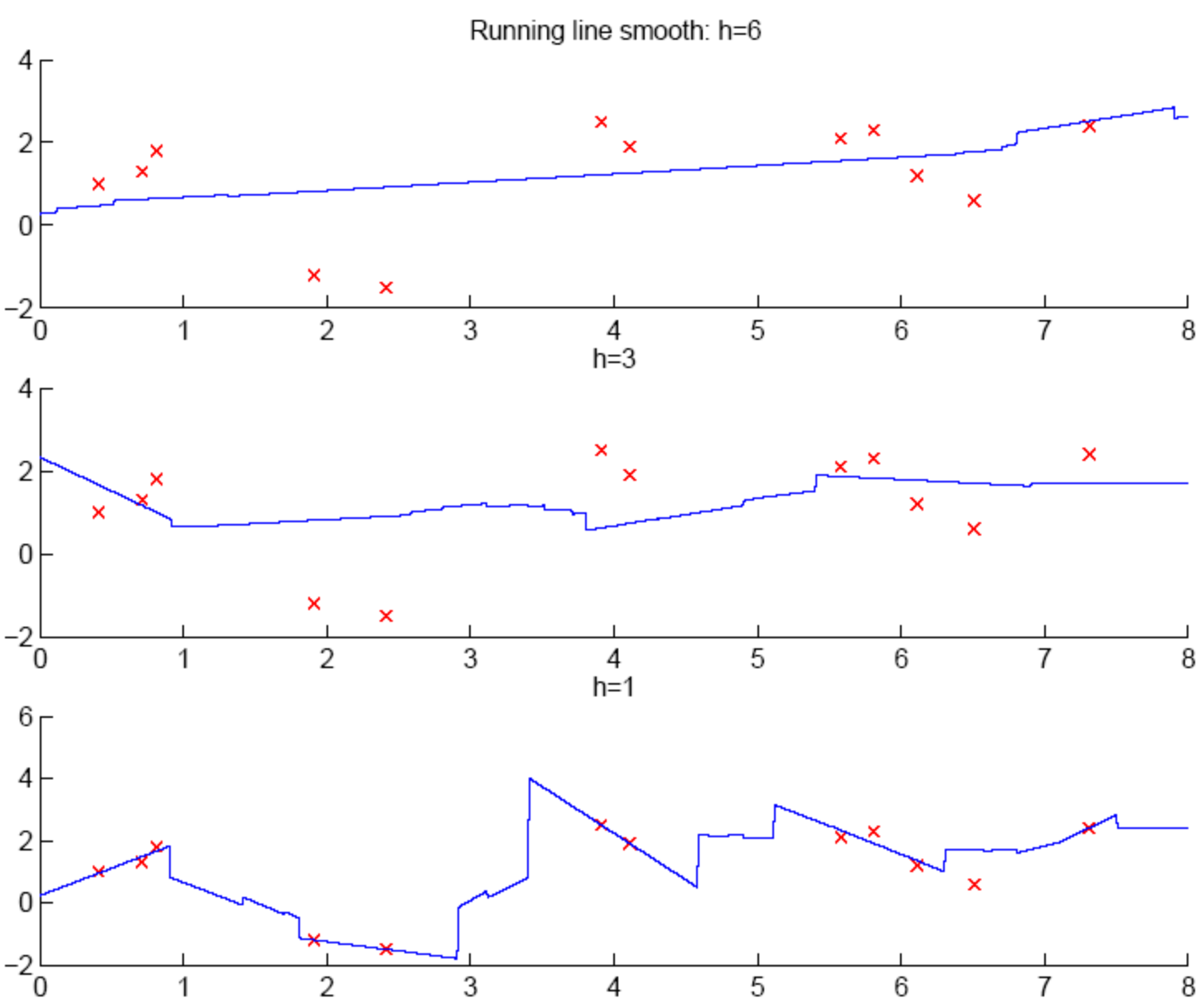
- Kernel smoother

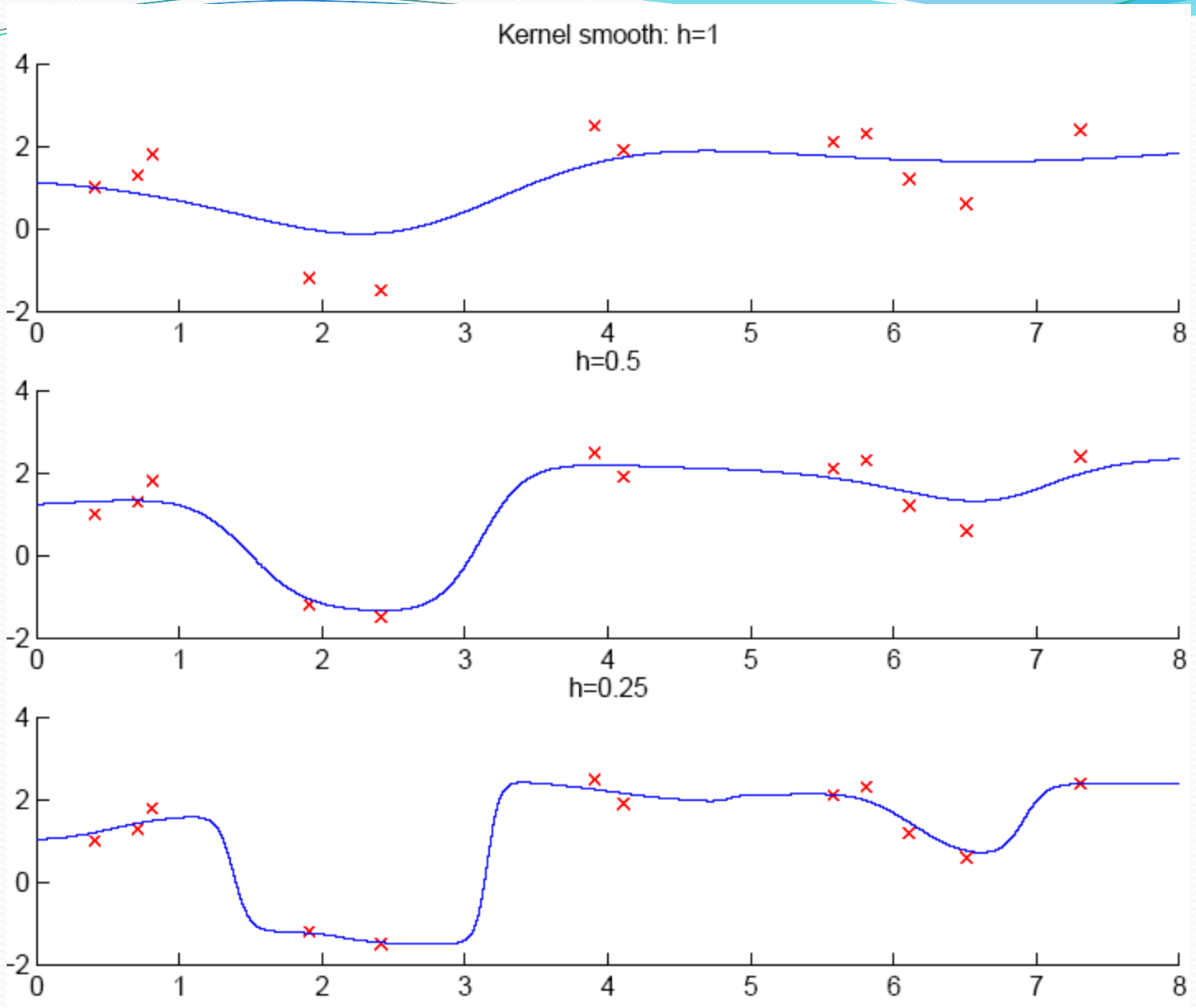
$$\hat{g}(x)=\frac{\sum_{t=1}^{N} K\left(\frac{x-x^t}{h}\right) r^t}{\sum_{t=1}^{N} K\left(\frac{x-x^t}{h}\right)}$$

where $K(\ )$ is Gaussian

- Additive models (Hastie and Tibshirani, 1990)

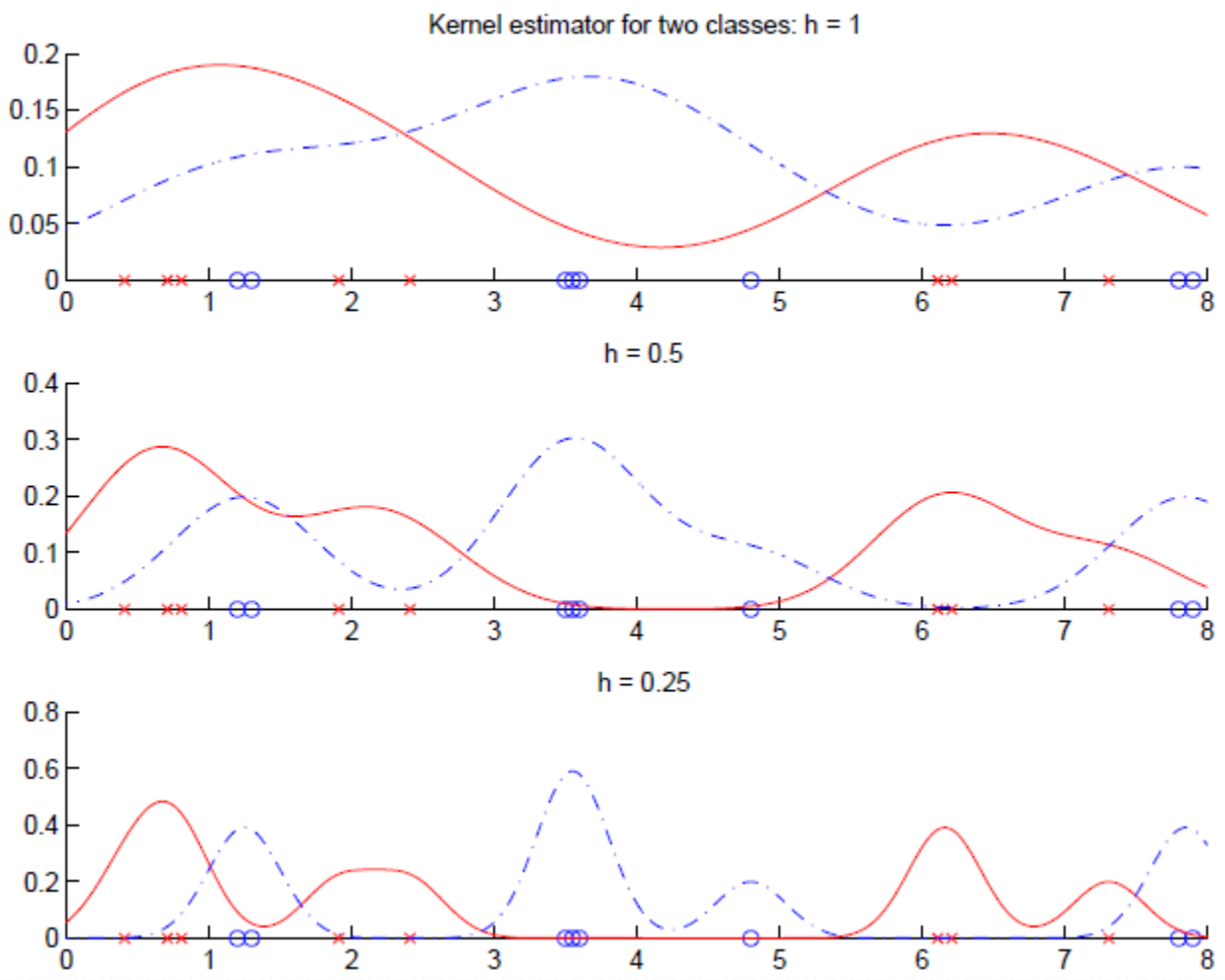

Running mean smoother: h=6
h=3
h=1
4
2
0
−2
0
1
2
3
4
5
6
7
8


Running line smooth: h=6
h=3
h=1


Kernel smooth: h=1
h=0.5
h=0.25

How to Choose *k* or *h*?

- When *k* or *h* is small, single instances matter; bias is small, variance is large (undersmoothing): High complexity
- As *k* or *h* increases, we average over more instances and variance decreases but bias increases (oversmoothing): Low complexity
- Cross-validation is used to finetune *k* or *h*.


Kernel estimator for two classes: h = 1
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8
h = 0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h = 0.25
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8

Discriminant-Based Models

Likelihood- vs. Discriminant-based Classification

- Likelihood-based: Assume a model for $p(\boldsymbol{x}|C_i)$, use Bayes' rule to calculate $P(C_i|\boldsymbol{x})$

  $$g_i(\boldsymbol{x}) = \log P(C_i|\boldsymbol{x})$$

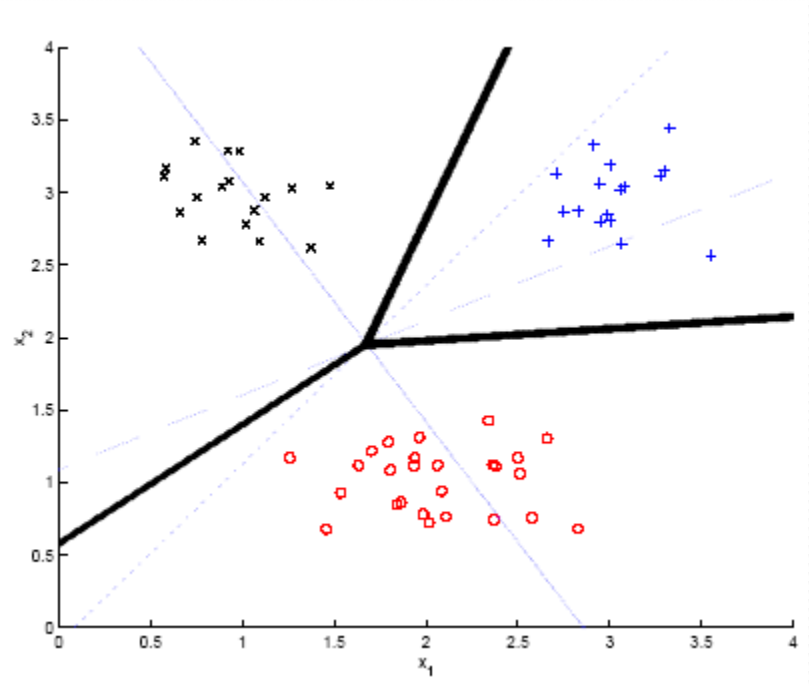
- Discriminant-based: Assume a model for $g_i(\boldsymbol{x}|\Phi_i)$; no density estimation
- Estimating the boundaries is enough; no need to accurately estimate the densities inside the boundaries

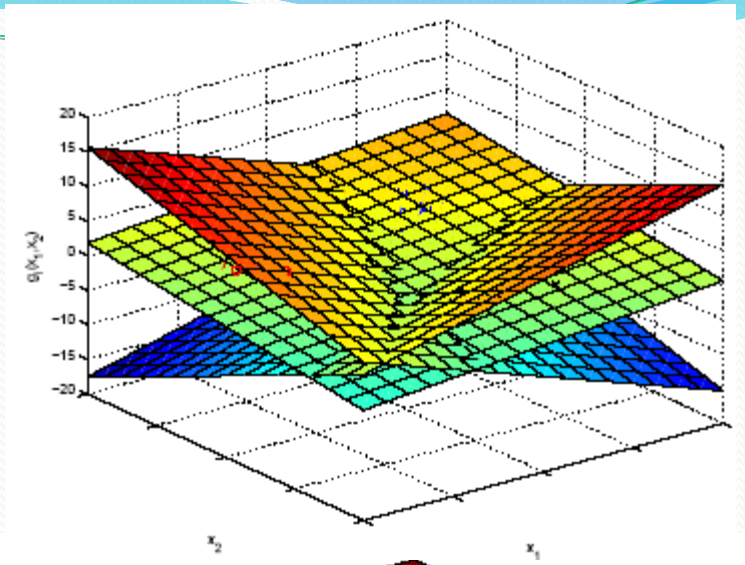
Linear Discriminant

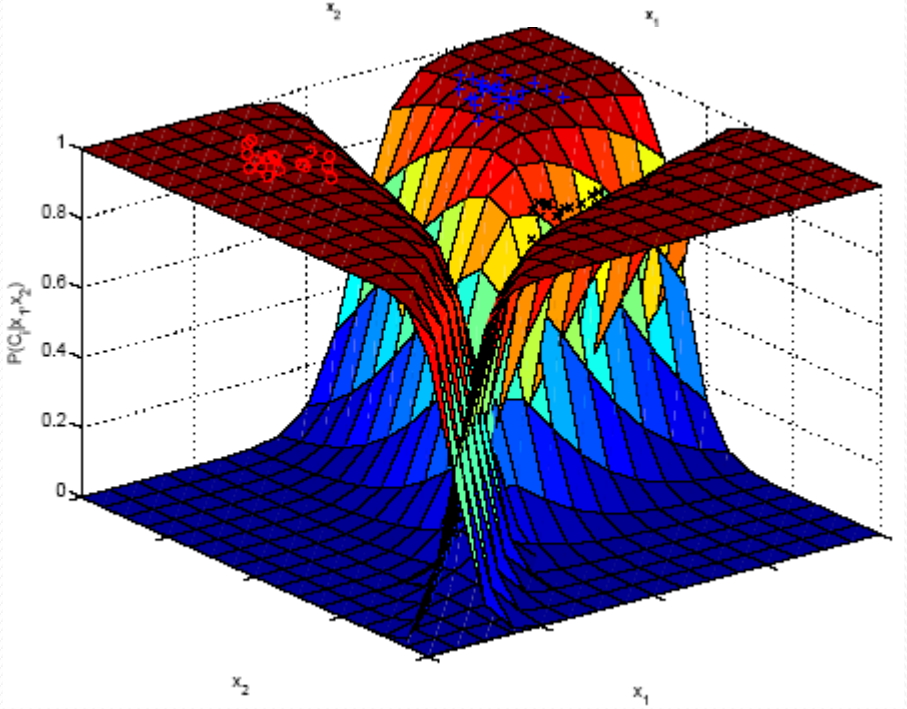
- Linear discriminant:

$$g_i\left(\mathbf{x}\,|\,\mathbf{w}_i, w_{i0}\right) = \mathbf{w}_i^T\mathbf{x} + w_{i0} = \sum_{j=1}^{d} w_{ij}x_j + w_{i0}$$

- Advantages:
  - Simple: O(d) space/computation
  - Knowledge extraction: Weighted sum of attributes; positive/negative weights, magnitudes (credit scoring)
  - Optimal when $p(\boldsymbol{x}|C_i)$ are Gaussian with shared cov matrix; useful when classes are (almost) linearly separable


4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
x2
x1


20
15
10
5
0
−5
−10
−15
−20
gi(x1,x2)
x2
x1


1
0.8
0.6
0.4
0.2
0
P(C|x1,x2)
x2
x1

Generalized Linear Model

- Quadratic discriminant:

$$g_i(\mathbf{x}|\mathbf{W}_i,\mathbf{w}_i,w_{i0})=\mathbf{x}^T\mathbf{W}_i\mathbf{x}+\mathbf{w}_i^T\mathbf{x}+w_{i0}$$

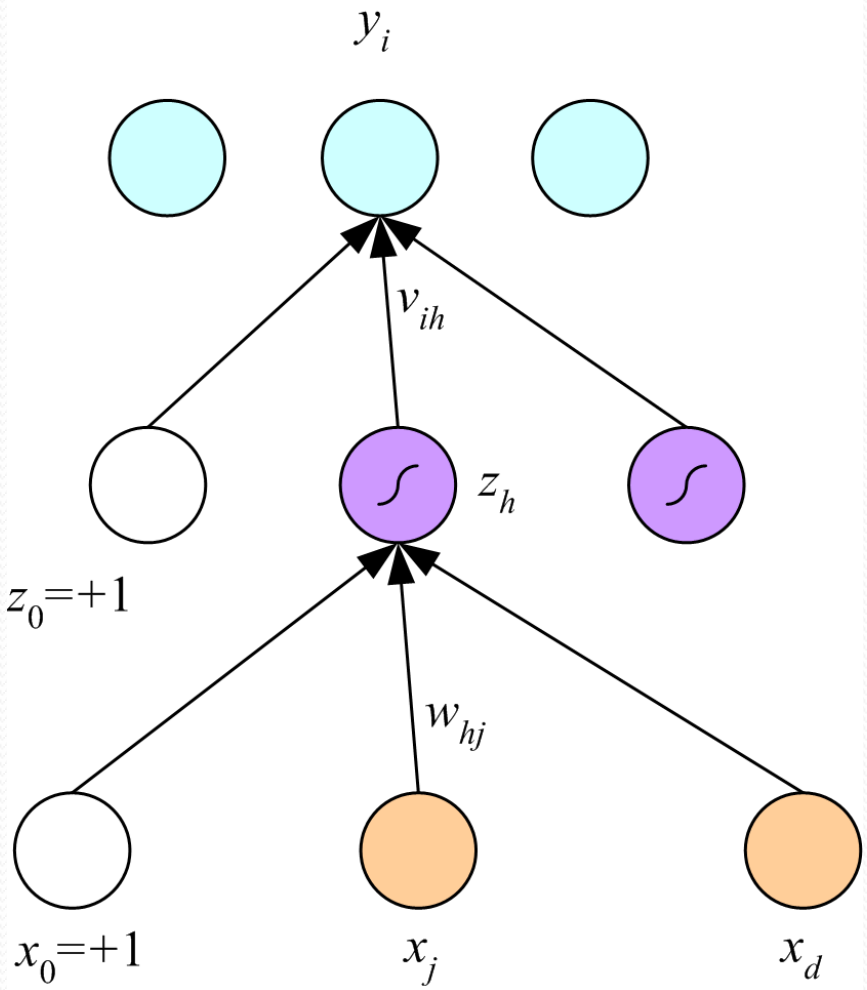
- Higher-order (product) terms:

$$z_1=x_1,\ z_2=x_2,\ z_3=x_1^2,\ z_4=x_2^2,\ z_5=x_1x_2$$

Map from ***x*** to ***z*** using nonlinear basis functions and use a linear discriminant in ***z***-space

$$g_i(\mathbf{x})=\sum_{j=1}^{k}w_{ij}\phi_j(\mathbf{x})$$

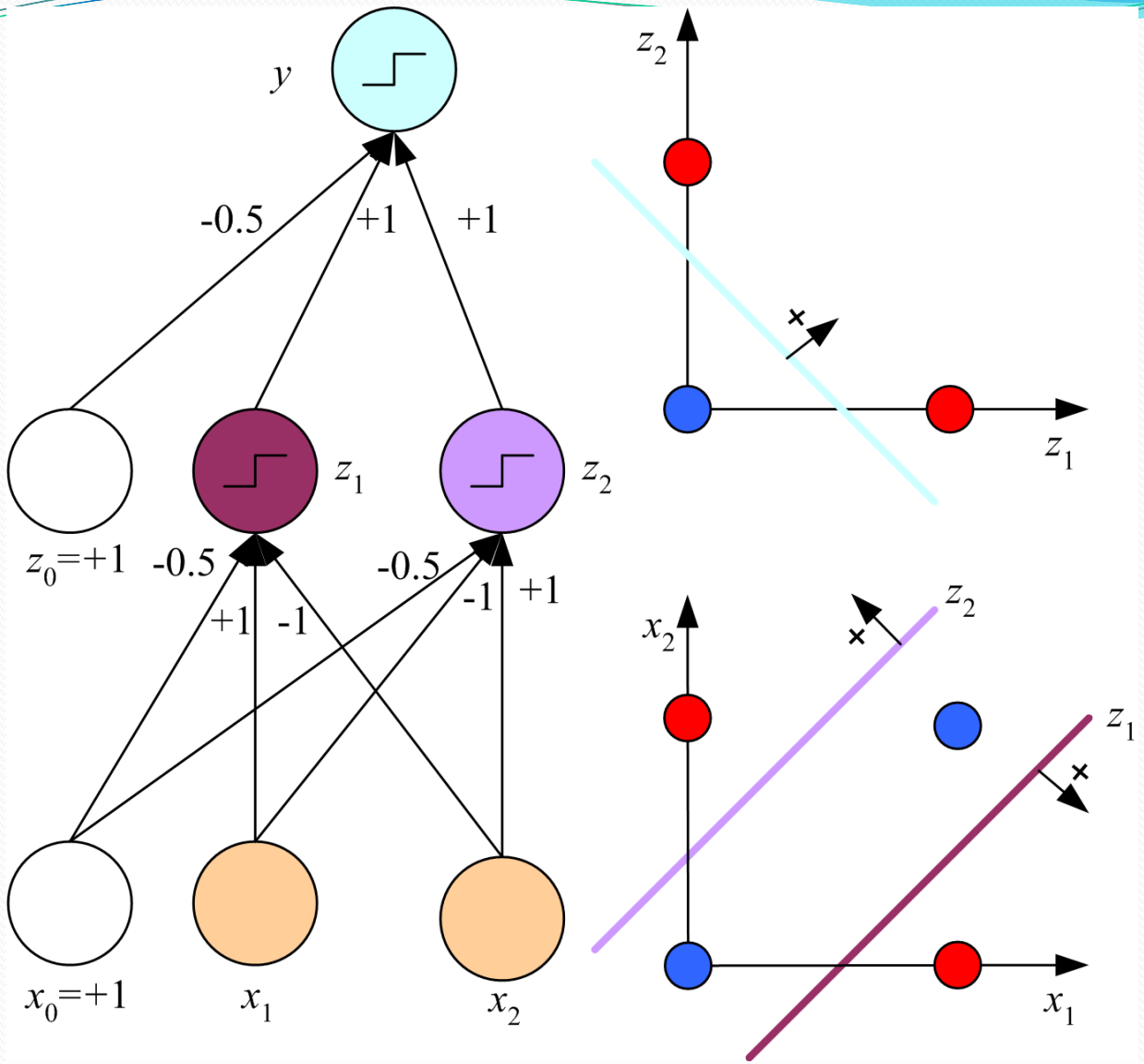
Multilayer Perceptrons



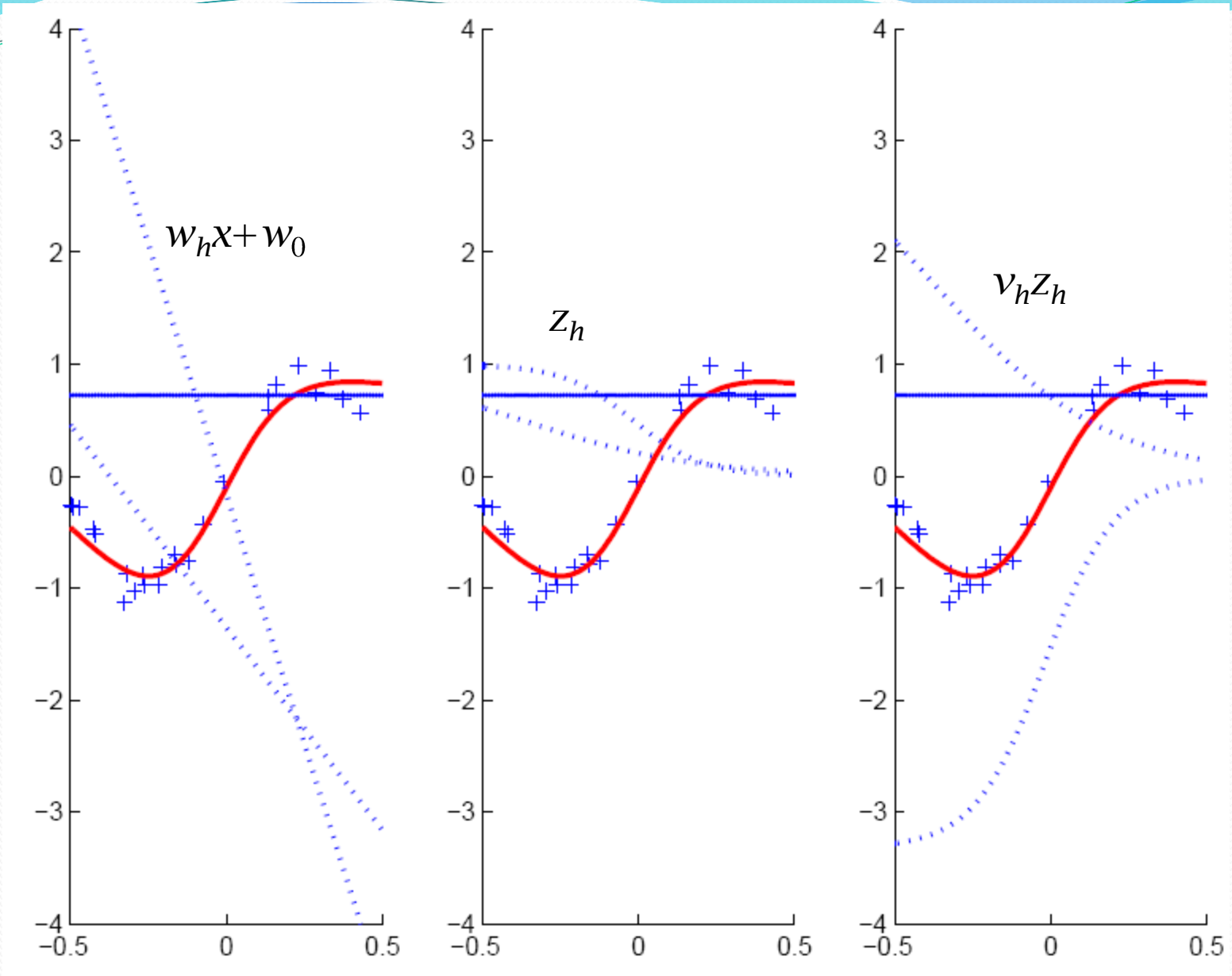
$$y_i = \mathbf{v}_i^T \mathbf{z} = \sum_{h=1}^{H} v_{ih} z_h + v_{i0}$$

$$z_h = \text{sigmoid}\left(\mathbf{w}_h^T \mathbf{x}\right) = \frac{1}{1+\exp\left[-\left(\sum_{j=1}^{d} w_{hj} x_j + w_{h0}\right)\right]}$$

(Rumelhart et al., 1986)



x_1 XOR x_2 = (x_1 AND ~x_2) OR (~x_1 AND x_2)


$w_h x + w_0$
z_h
$v_h z_h$
4
3
2
1
0
−1
−2
−3
−4
−0.5
0
0.5

Kernel Machines

- Discriminant-based: No need to estimate densities first
- Define the discriminant in terms of support vectors
- The use of kernel functions, application-specific measures of similarity
- No need to represent instances as vectors
- Convex optimization problems with a unique solution

Optimal Separating Hyperplane

$$\mathcal{X} = \left\{\mathbf{x}^t, r^t\right\}_t \text{ where } r^t = \begin{cases} +1 & \text{if } \mathbf{x}^t \in C_1 \\ -1 & \text{if } \mathbf{x}^t \in C_2 \end{cases}$$

find $\mathbf{w}$ and w_0 such that

$$\mathbf{w}^T\mathbf{x}^t + w_0 \geq +1 \text{ for } r^t = +1$$

$$\mathbf{w}^T\mathbf{x}^t + w_0 \leq +1 \text{ for } r^t = -1$$

which can be rewritten as

$$r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1$$

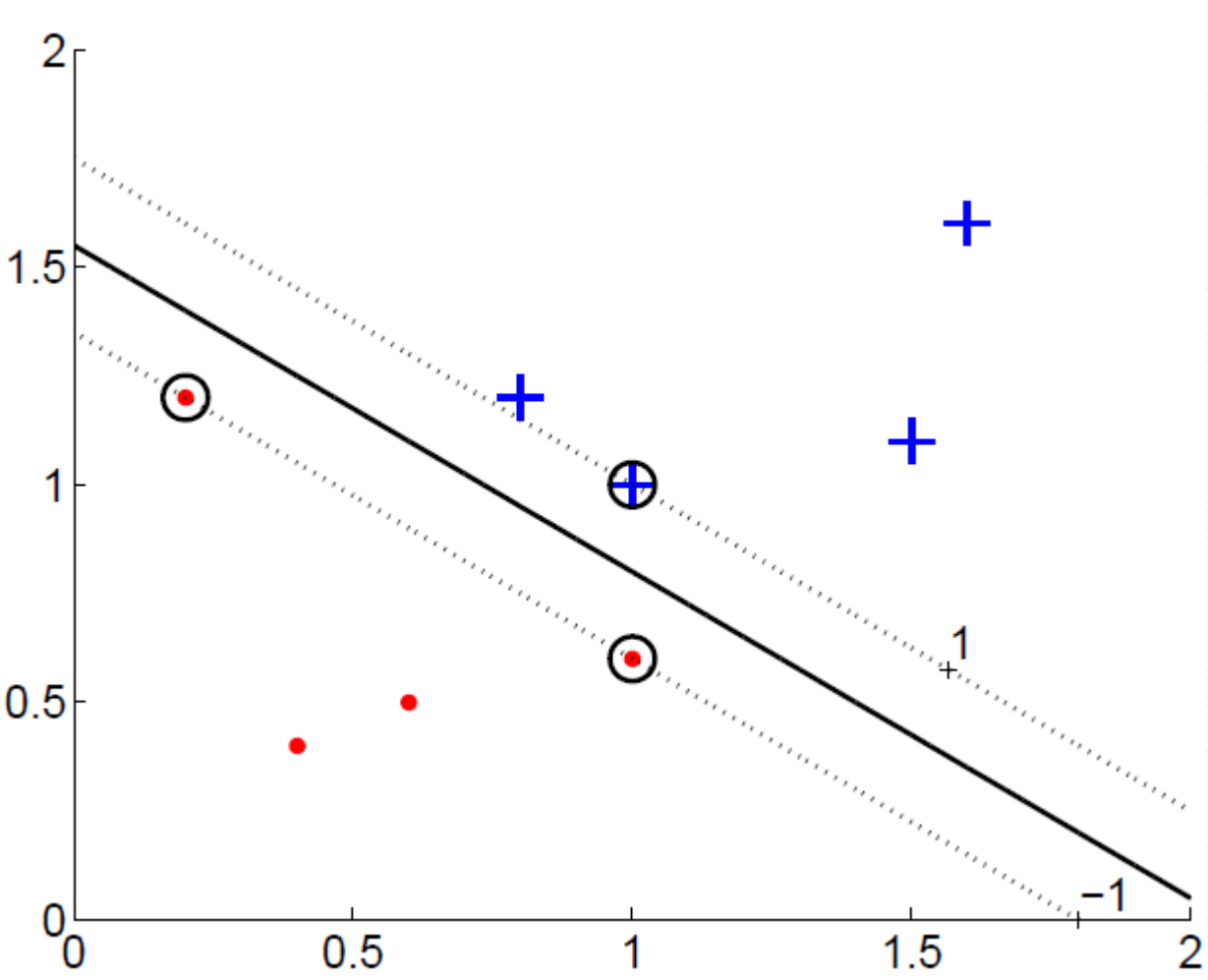
(Cortes and Vapnik, 1995; Vapnik, 1995)

Margin

- Distance from the discriminant to the closest instances on either side
- Distance of x to the hyperplane is $\frac{\left|\mathbf{w}^T\mathbf{x}^t + w_0\right|}{\|\mathbf{w}\|}$
- To max margin

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

Margin



$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t \left[r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) - 1\right]$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) + \sum_{t=1}^{N} \alpha^t$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{t=1}^{N} \alpha^t r^t \mathbf{x}^t$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_{t=1}^{N} \alpha^t r^t = 0$$

$$L_d = \frac{1}{2}\left(\mathbf{w}^T\mathbf{w}\right) - \mathbf{w}^T\sum_t \alpha^t r^t \mathbf{x}^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t$$

$$= -\frac{1}{2}\left(\mathbf{w}^T\mathbf{w}\right) + \sum_t \alpha^t$$

$$= -\frac{1}{2}\sum_t\sum_s \alpha^t \alpha^s r^t r^s \left(\mathbf{x}^t\right)^T \mathbf{x}^s + \sum_t \alpha^t$$

$$\text{subject to } \sum_t \alpha^t r^t = 0 \text{ and } \alpha^t \geq 0, \forall t$$

Most α^t are 0 and only a small number have $\alpha^t > 0$; they are the support vectors

Soft Margin Hyperplane

- Not linearly separable

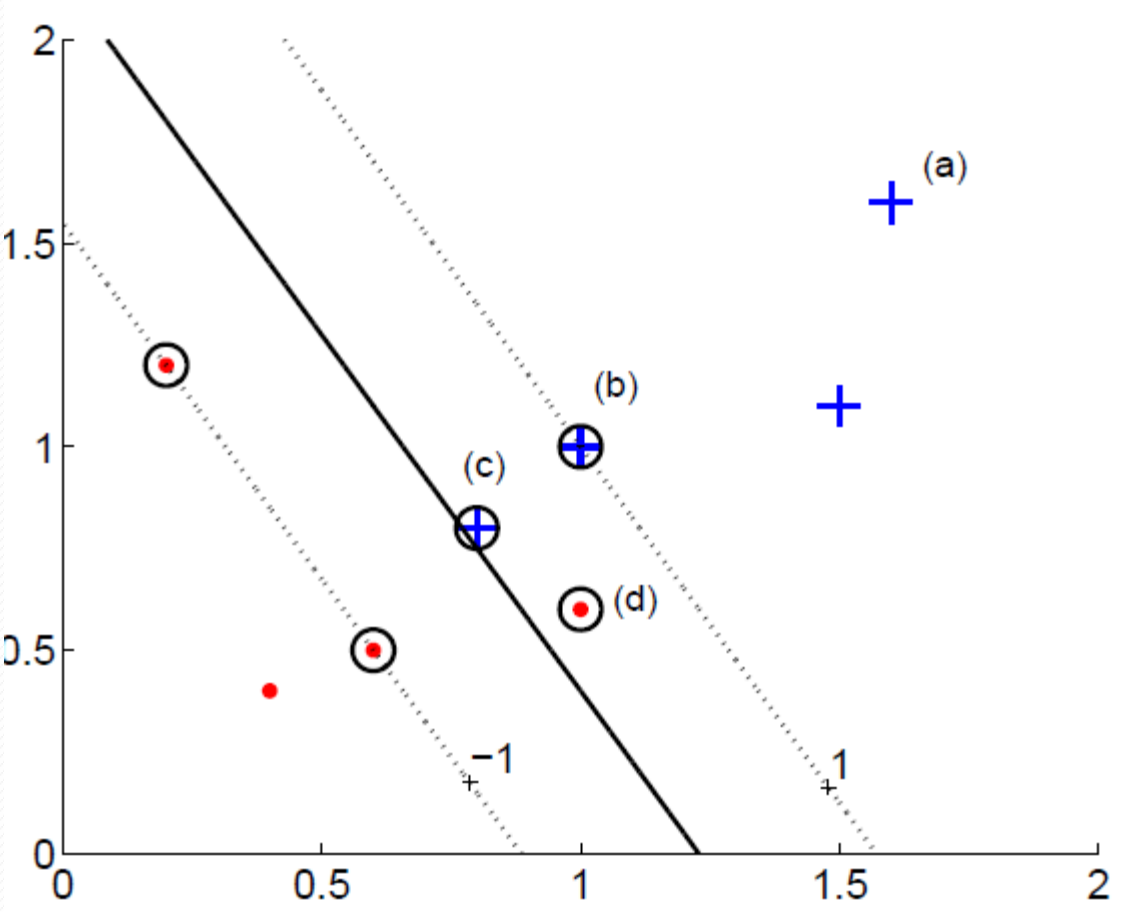
$$r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t$$

- Soft error

$$\sum_t \xi^t$$

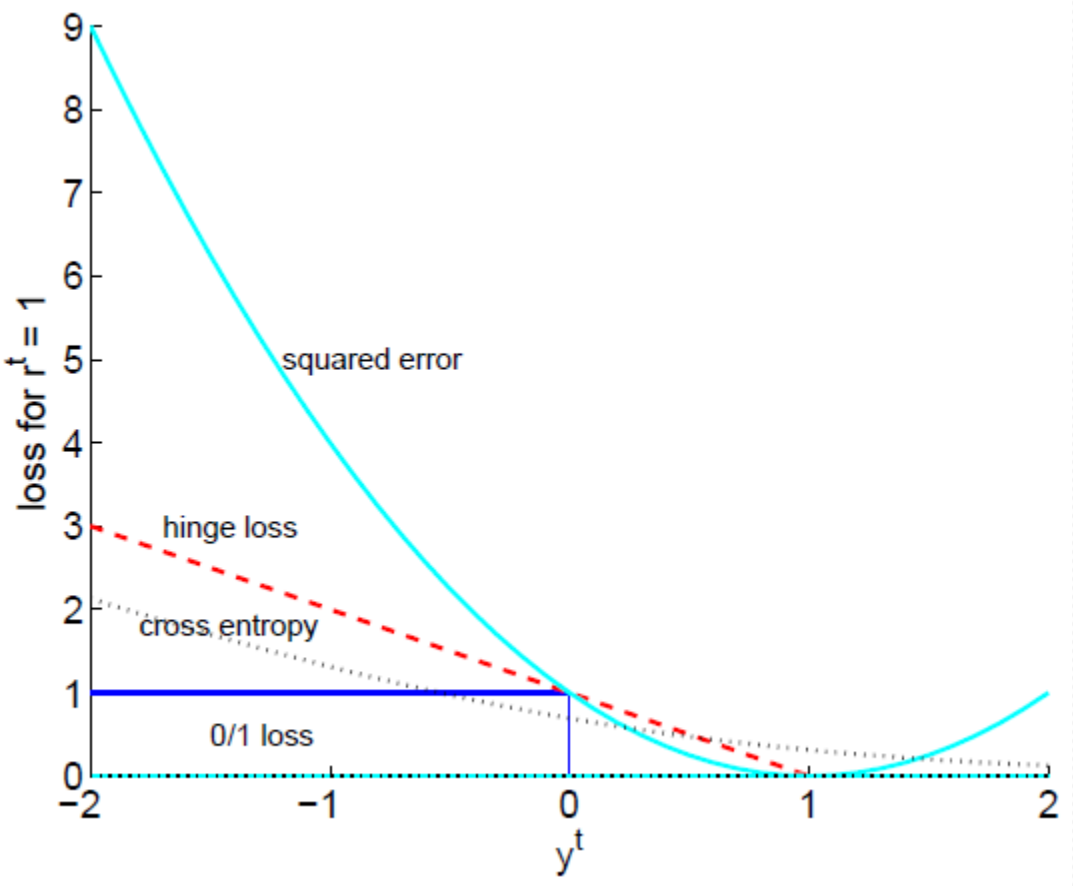
- New primal is

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t - \sum_t \alpha^t\left[r^t\left(\mathbf{w}^T x^t + w_0\right) - 1 + \xi^t\right] - \sum_t \mu^t \xi^t$$


2
1.5
1
0.5
0
0
0.5
1
1.5
2
(a)
(b)
(c)
(d)
−1
1

Hinge Loss

$$L_{hinge}(y^t, r^t) = \begin{cases} 0 & \text{if } y^t r^t \geq 1 \\ 1 - y^t r^t & \text{otherwise} \end{cases}$$



Kernel Trick

- Preprocess input $\boldsymbol{x}$ by basis functions

$$\boldsymbol{z} = \boldsymbol{\varphi}(\boldsymbol{x}) \qquad g(\boldsymbol{z})=\boldsymbol{w}^T\boldsymbol{z}$$

$$g(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x})$$

- The SVM solution

$$\mathbf{w}=\sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \boldsymbol{\varphi}\left(\mathbf{x}^t\right)$$

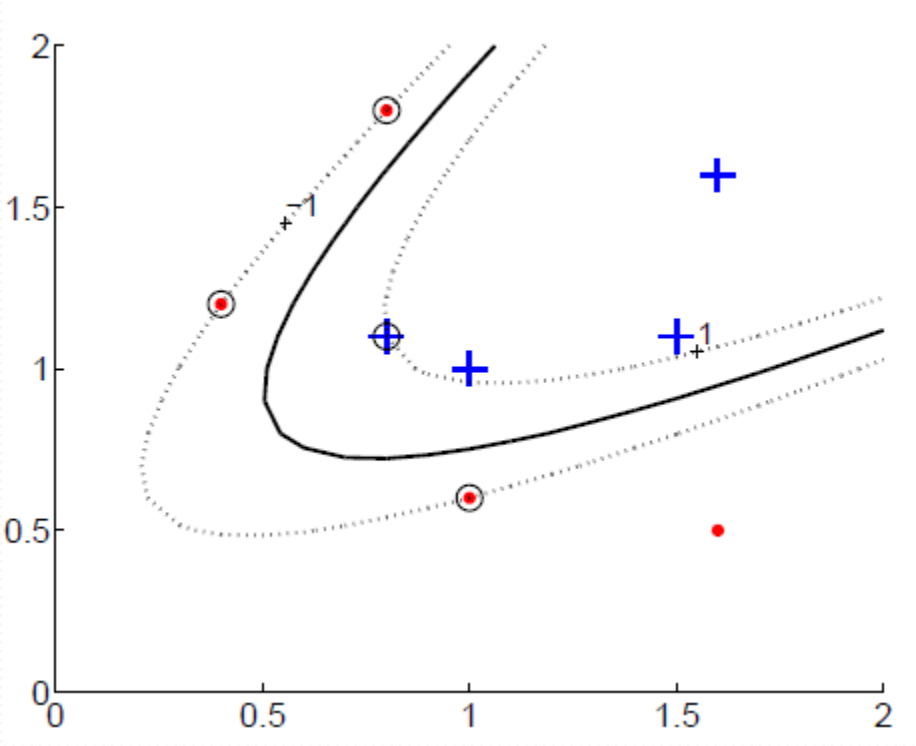
$$g\left(\mathbf{x}\right)=\mathbf{w}^T\boldsymbol{\varphi}\left(\mathbf{x}\right)=\sum_t \alpha^t r^t \boldsymbol{\varphi}\left(\mathbf{x}^t\right)^T\boldsymbol{\varphi}\left(\mathbf{x}\right)$$

$$g\left(\mathbf{x}\right)=\sum_t \alpha^t r^t K\left(\mathbf{x}^t,\mathbf{x}\right)$$

Vectorial Kernels

- Polynomials of degree q:

$$K(\mathbf{x}^t, \mathbf{x}) = (\mathbf{x}^T \mathbf{x}^t + 1)^q$$



$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^2$$

$$= (x_1 y_1 + x_2 y_2 + 1)^2$$

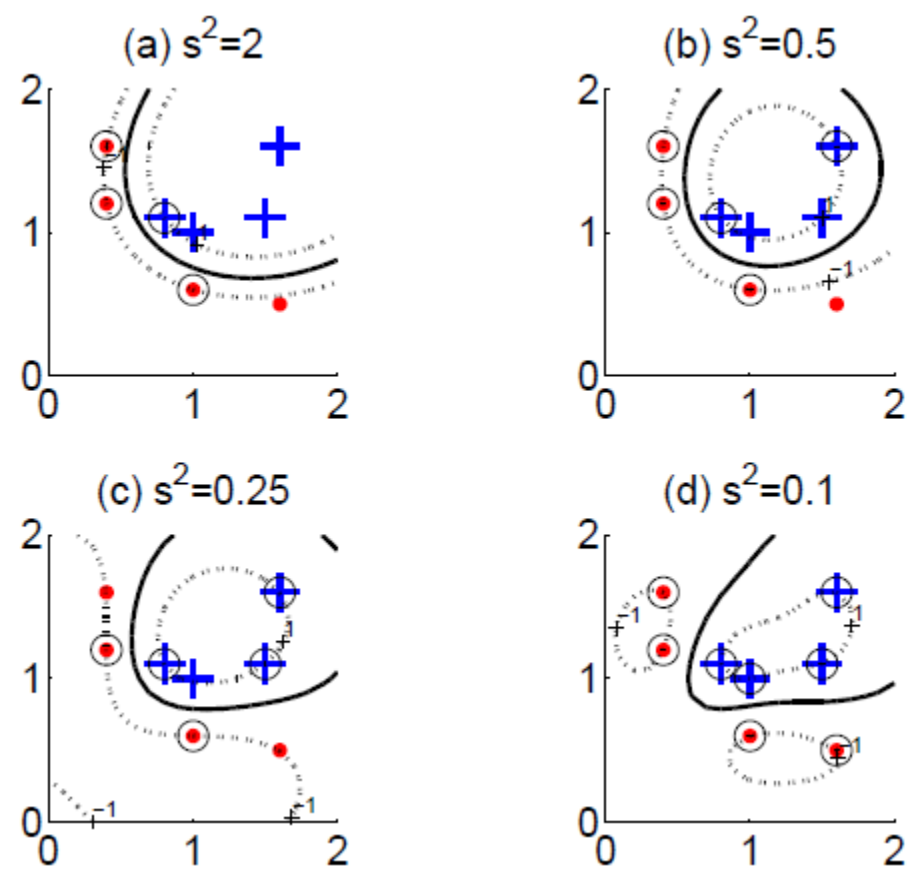
$$= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2$$

$$\phi(\mathbf{x}) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2\right]^T$$

Vectorial Kernels

- Radial-basis functions:

$$K(\mathbf{x}^t, \mathbf{x}) = \exp\left[-\frac{\left\|\mathbf{x}^t - \mathbf{x}\right\|^2}{2s^2}\right]$$



Defining kernels

- Kernel "engineering"
- Defining good measures of similarity
- String kernels, graph kernels, image kernels, ...
- Empirical kernel map: Define a set of templates $\boldsymbol{m}_i$ and score function $s(\boldsymbol{x},\boldsymbol{m}_i)$

  $\phi(\boldsymbol{x}^t)=[s(\boldsymbol{x}^t,\boldsymbol{m}_1), s(\boldsymbol{x}^t,\boldsymbol{m}_2),..., s(\boldsymbol{x}^t,\boldsymbol{m}_M)]$

  and

  $K(\boldsymbol{x},\boldsymbol{x}^t)=\phi(\boldsymbol{x})^T \phi(\boldsymbol{x}^t)$

Multiple Kernel Learning

- Fixed kernel combination

$$K(\mathbf{x},\mathbf{y}) = \begin{cases} cK(\mathbf{x},\mathbf{y}) \\ K_1(\mathbf{x},\mathbf{y}) + K_2(\mathbf{x},\mathbf{y}) \\ K_1(\mathbf{x},\mathbf{y}) K_2(\mathbf{x},\mathbf{y}) \end{cases}$$

- Adaptive kernel combination

$$K(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} \eta_i K_i(\mathbf{x},\mathbf{y})$$

$$L_d = \sum_t \alpha^t - \frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s \sum_i \eta_i K_i(\mathbf{x}^t, \mathbf{x}^s)$$

$$g(\mathbf{x}) = \sum_t \alpha^t r^t \sum_i \eta_i K_i(\mathbf{x}^t, \mathbf{x})$$

- Localized kernel combination (see Part II)

$$g(\mathbf{x}) = \sum_t \alpha^t r^t \sum_i \eta_i(\mathbf{x} \mid \theta) K_i(\mathbf{x}^t, \mathbf{x})$$

SVM for Regression

- Use a linear model (possibly kernelized)

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}+w_0$$

- Use the ϵ-sensitive error function

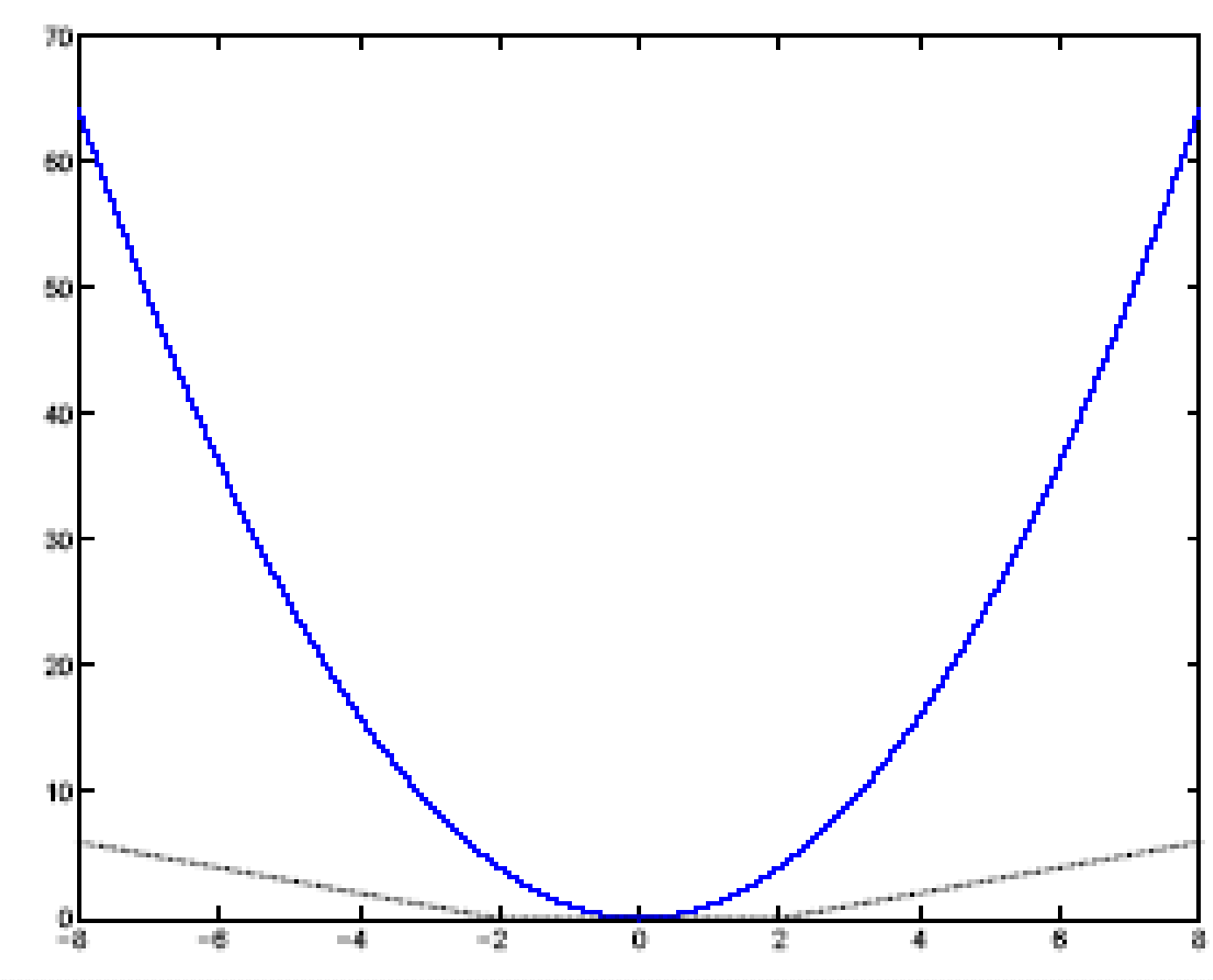
$$e_{\varepsilon}\left(r^t, f\left(\mathbf{x}^t\right)\right)=\begin{cases}0 & \text{if } \left|r^t-f\left(\mathbf{x}^t\right)\right|<\varepsilon \\ \left|r^t-f\left(\mathbf{x}^t\right)\right|-\varepsilon & \text{otherwise}\end{cases}$$

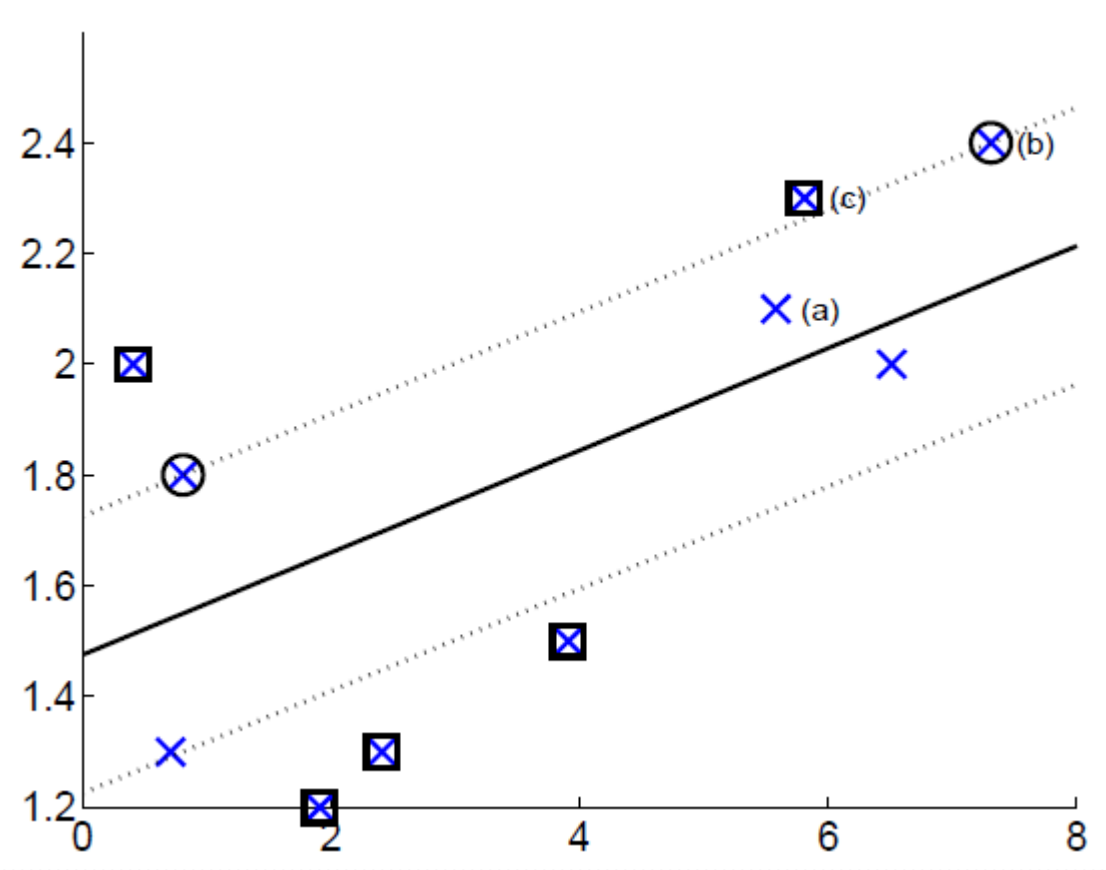
- $$\min \frac{1}{2}\|\mathbf{w}\|^2+C\sum_t\left(\xi_+^t+\xi_-^t\right)$$

$$r^t-\left(\mathbf{w}^T\mathbf{x}+w_0\right)\le\varepsilon+\xi_+^t$$

$$\left(\mathbf{w}^T\mathbf{x}+w_0\right)-r^t\le\varepsilon+\xi_-^t$$

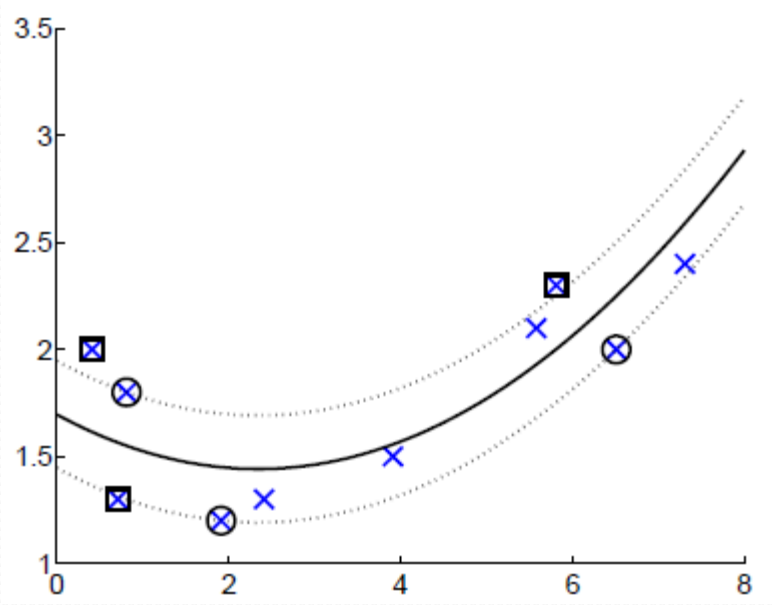
$$\xi_+^t,\xi_-^t\ge 0$$


70
60
50
40
30
20
10
0
-8
-6
-4
-2
0
2
4
6
8

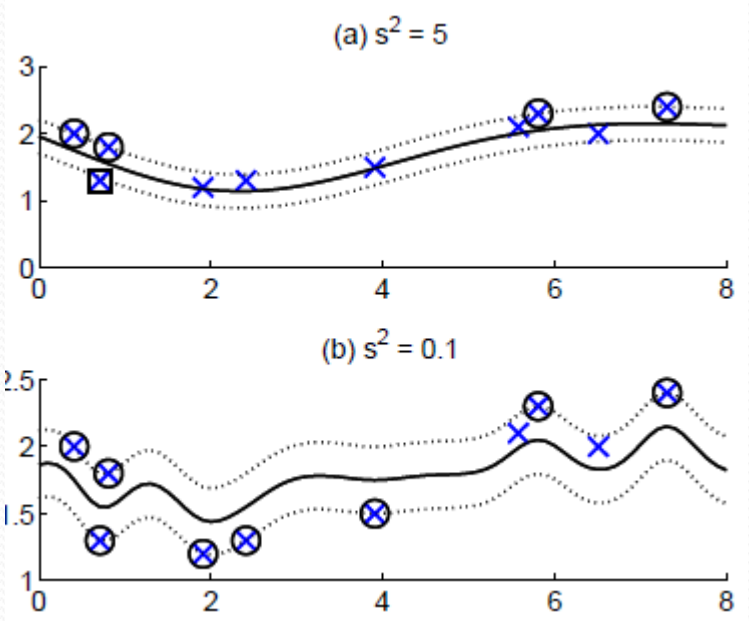

2.4
2.2
2
1.8
1.6
1.4
1.2
0
2
4
6
8
(a)
(b)
(c)

Kernel Regression

- Polynomial kernel



- Gaussian kernel



One-Class Kernel Machines

- Consider a sphere with center $\boldsymbol{a}$ and radius R

$$\min R^2 + C\sum_t \xi^t$$

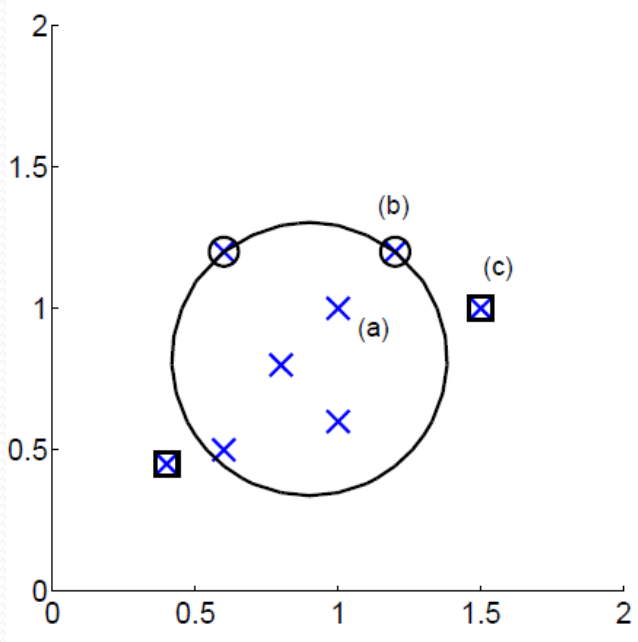
$$\text{subject to}$$

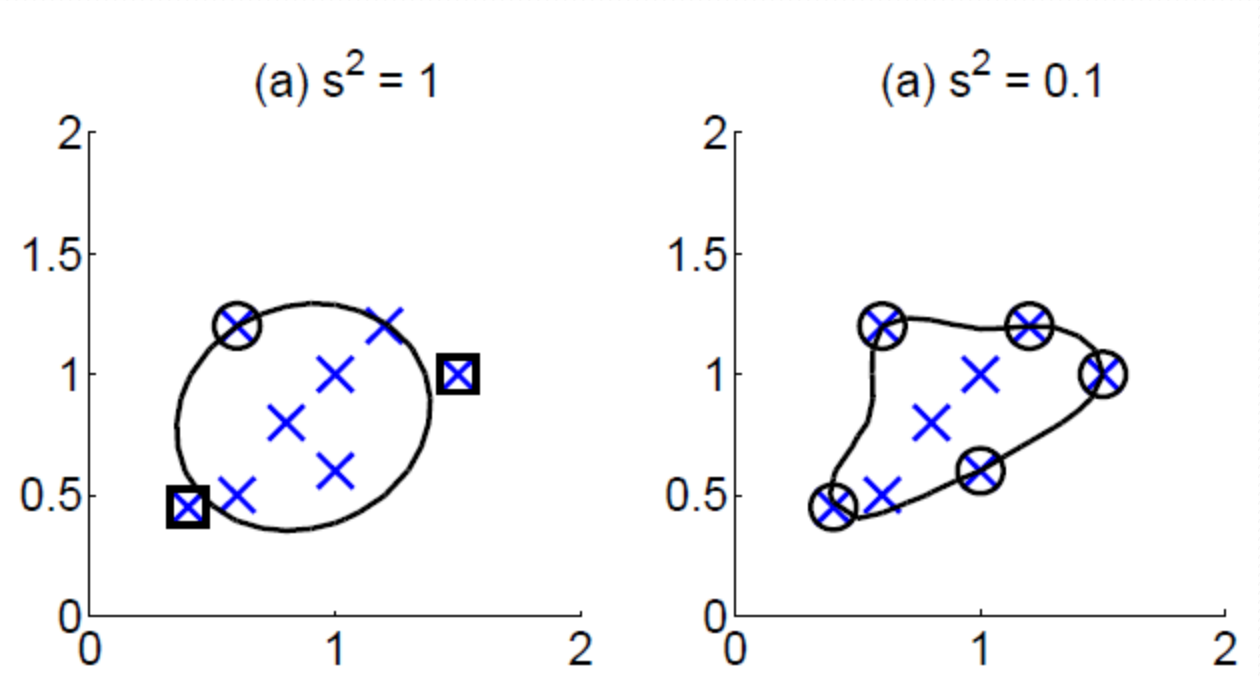
$$\left\|\mathbf{x}^t - a\right\| \le R^2 + \xi^t, \xi^t \ge 0$$

$$L_d = \sum_t \alpha^t \left(x^t\right)^T x^s - \sum_{t=1}^{N}\sum_s \alpha^t \alpha^s r^t r^s \left(x^t\right)^T x^s$$

$$\text{subject to}$$

$$0 \le \alpha^t \le C, \sum_t \alpha^t = 1$$




(a) $s^2 = 1$
(a) $s^2 = 0.1$
2
1.5
1
0.5
0
0
1
2

Conclusions

- So many algorithms, so little time
- Choosing the best model; statistical tests.
- No Free Lunch theorem
- Do different methods make different errors? See Part II.