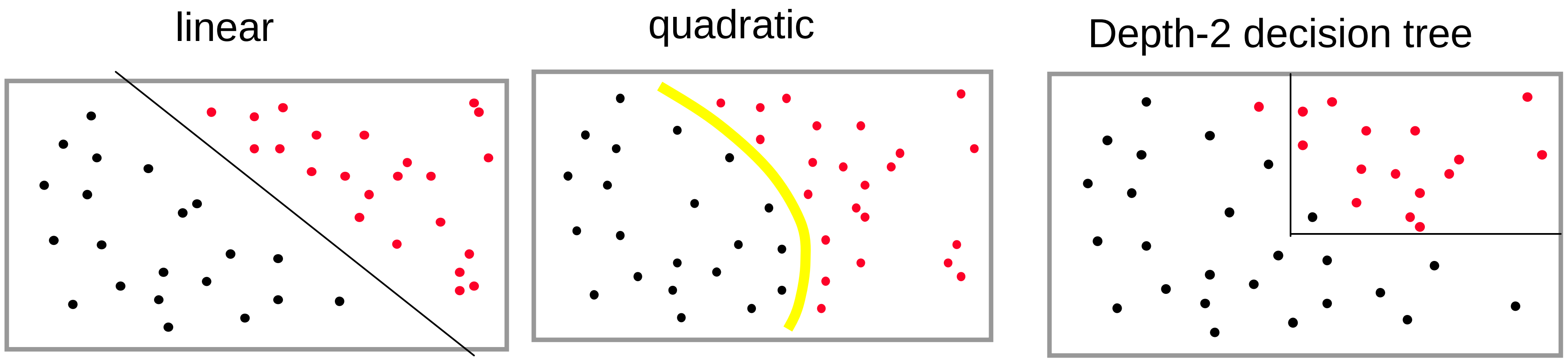
CS340 Machine learning
Lecture 2
Classification and generalization error

Summary of last lecture

- Given training data D = { (x_1.y_1), …, (x_N, y_N) }
- Choose right hypothesis class H



- Fit parameters θ of function given H and D

$$f(x,\theta) = sgn(\theta^T x) = sgn(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

- Applications

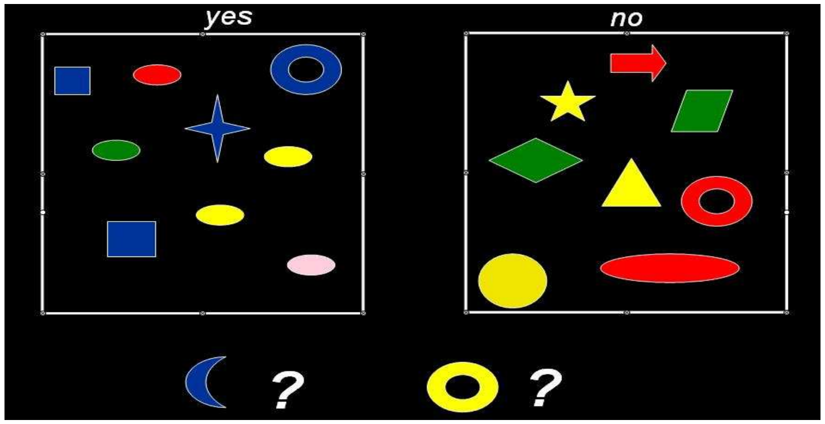
Notation for inputs (covariates)

- Input is a d-dimensional feature vector $\mathbf{x} \in \mathcal{X}$
- We will frequently omit boldface notation $\mathbf{x}$
- Sometimes we assume each component (attribute) is a real-valued quantity, so $\mathcal{X} = R^d$
- In general, some components may be categorical (eg male, female) or ordinal (eg low, medium, high) or an integer (eg number of arrivals) etc
- We can also have structured inputs, like text documents, DNA sequences or web pages
- In statistics, they use p instead of d, and say “covariate” or “explanatory variable” instead of “input”.

Notation for output (response/ target)

- The output is $y \in \mathcal{Y}$
- For classification problems with K mutually exclusive categorical classes, we have $\mathcal{Y} = \{1, 2, \ldots, K\}$
- If K=2, this is binary classification.
- We will consider ordinal classes later.
- Sometimes we will use a 1-of-K binary encoding, e.g., for 3 classes, class 1 = (1,0,0), class 2 = (0,1,0), class 3 = (0,0,1) . This allows us to represent non mutually exclusive classes.

Notation for training data



d features (attributes)

Training set:

X: N x d

y: N x 1

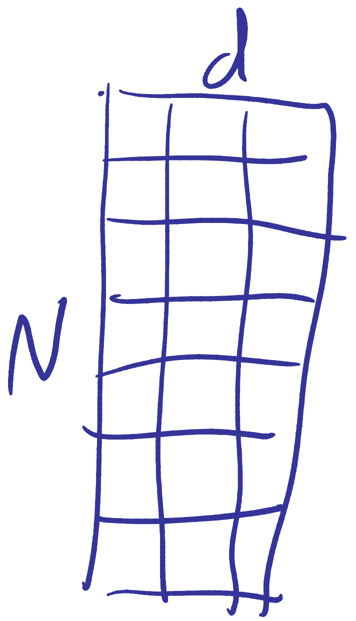
N cases

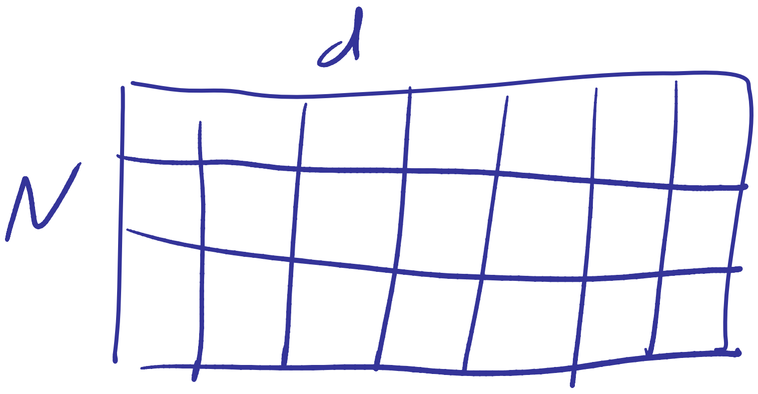
Color	Shape	Size (cm)	Label
Blue	Square	10	Yes
Red	Ellipse	12.3	Yes
Red	Ellipse	8.7	No

X y

Design matrix

Design matrices





$N >> d$

$d >> N$

Tall & skinny

Short & fat

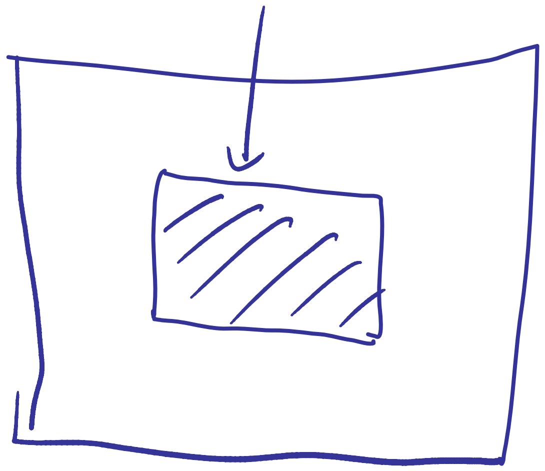
Notation for training data

- N = number of training examples, x_n = n'th training input, for n=1:N. (Notation used by Bishop's book.)
- In statistics, more common to use n = number of training examples, x_i = i'th example, i=1:n.
- Hastie book uses x_i, i=1:N.

H=rectangles in the R^d plane

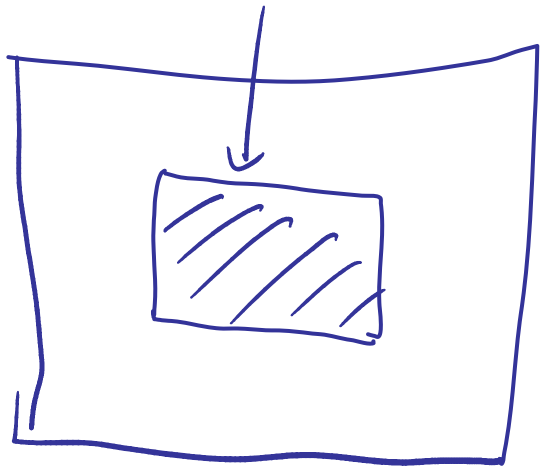
learn concept of “healthy levels” of cholestrol x_1 and insulin x_2 from positive and negative examples

True concept C (hidden)

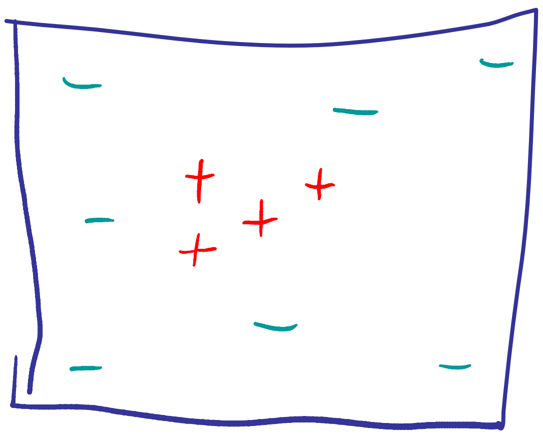


Training data D

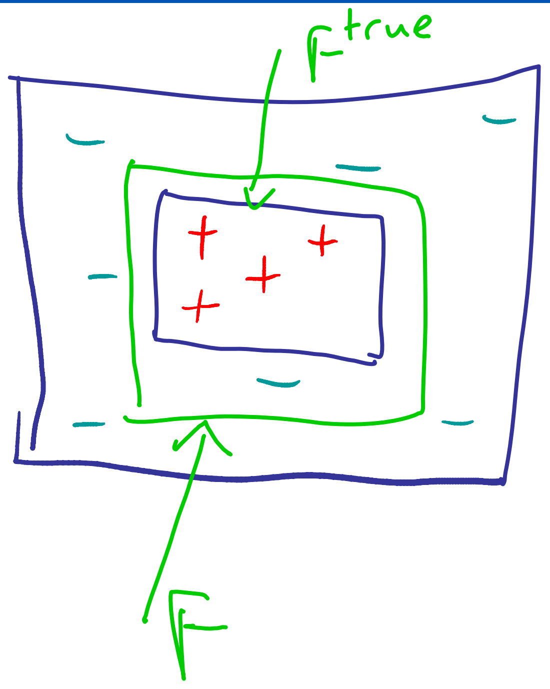
True concept C (hidden)



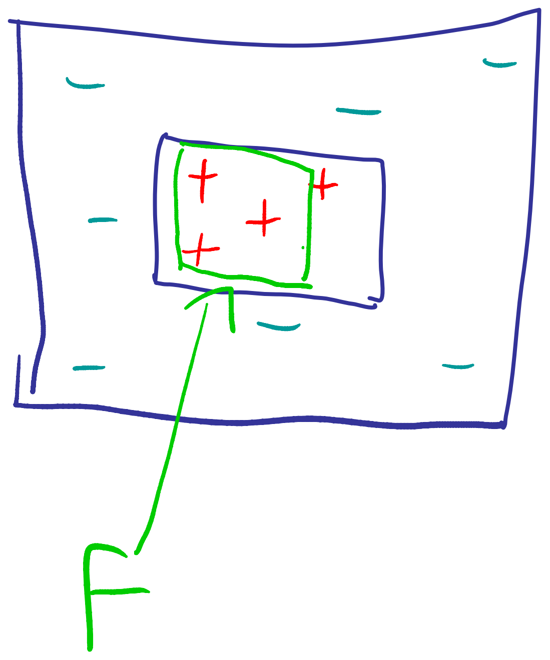
Training data D sampled from C



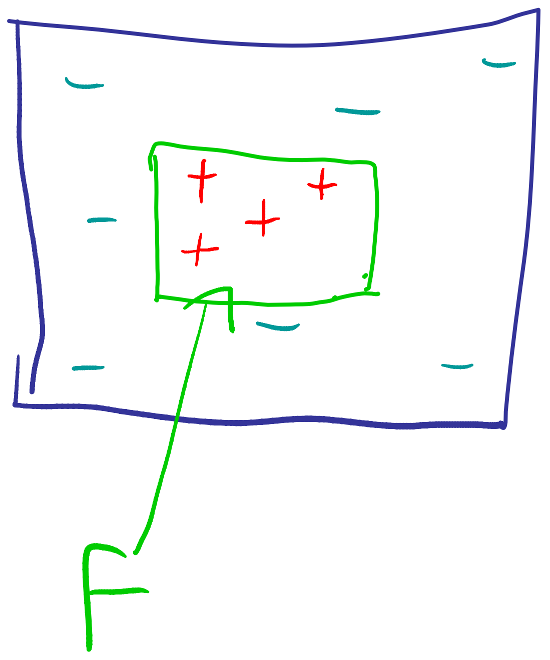
Learning = inferring hidden concept (function)



Too big

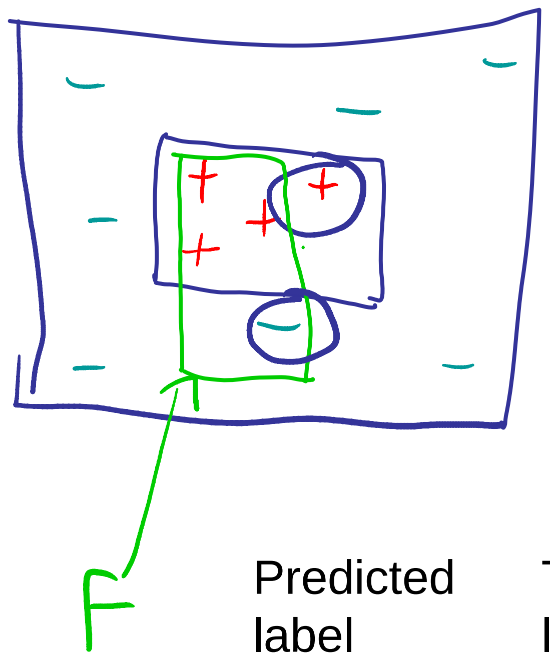


Too small



Just right

Empirical error



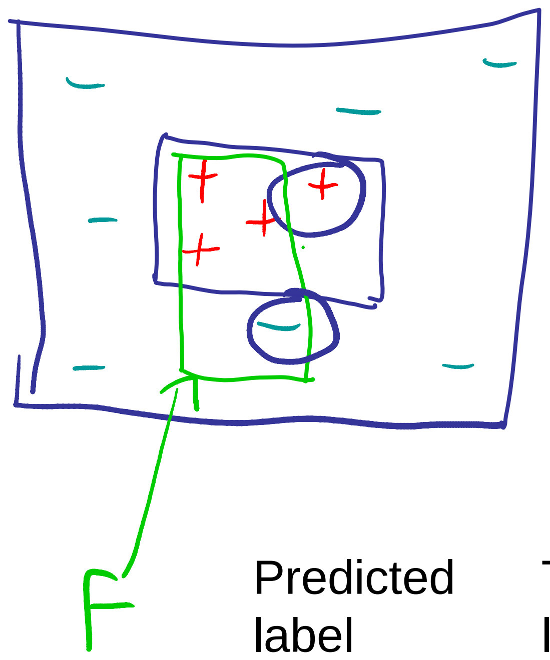
Predicted label

True label

$$N_{err} = \sum_{n=1}^{N} I(\hat{y}(x_n) \neq y_n)$$

$$I(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

Empirical error



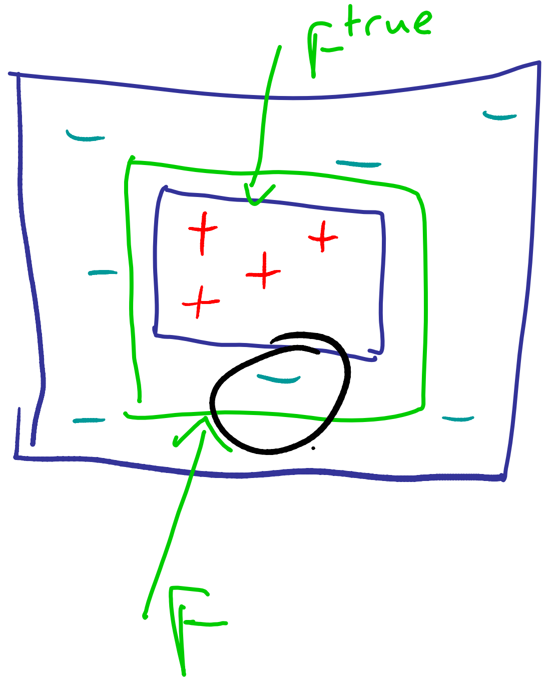
Predicted label

True label

$$N_{err} = \sum_{n=1}^{N} \delta(\hat{y}(x_n) - y_n)$$

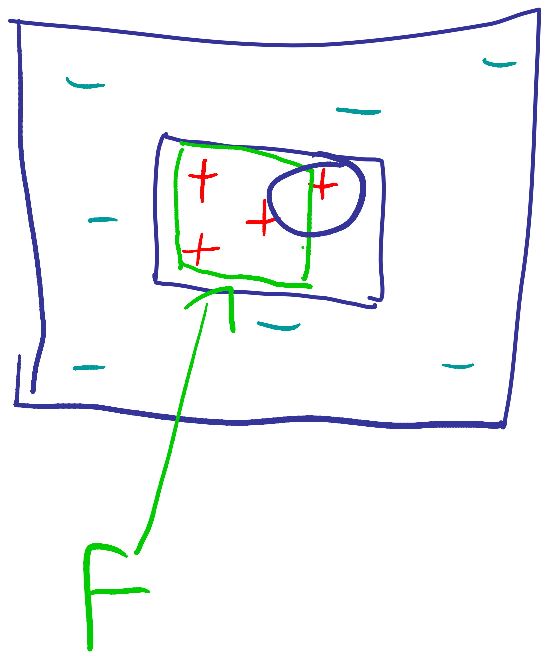
$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

False positive



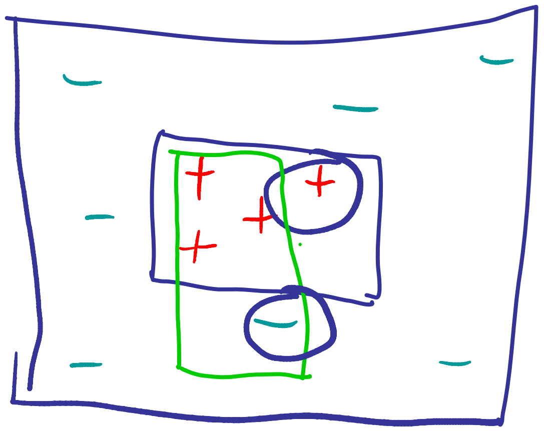
$$N_{fp} = \sum_{n=1}^{N} I(\hat{y}(x_n) = 1 \ \wedge \ y_n = 0)$$

False negative



$$N_{fn} = \sum_{n=1}^{N} I(\hat{y}(x_n) = 0 \ \wedge \ y_n = 1)$$

Generalization error

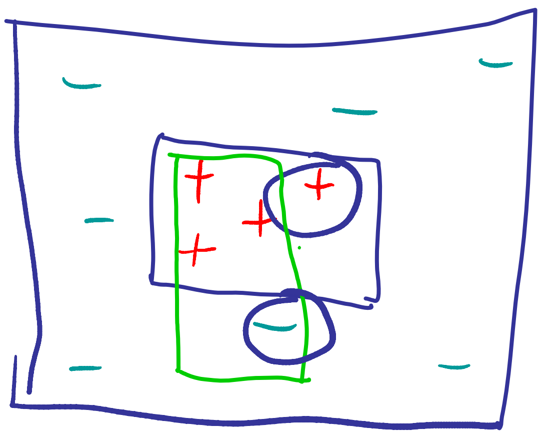


We want to minimize the average error rate,
where the test cases are assumed to be sampled from C

$$
\begin{aligned}
E[err] &= E_{x,y} I(\hat{y}(x) \neq y) \\
&= \int_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} I(\hat{y}(x) \neq y) p(x,y)
\end{aligned}
$$

But p(x,y) (which defines C) is unknown

Empirical risk minimization



In ERM, we choose f so as to
minimize the number of errors on the training set

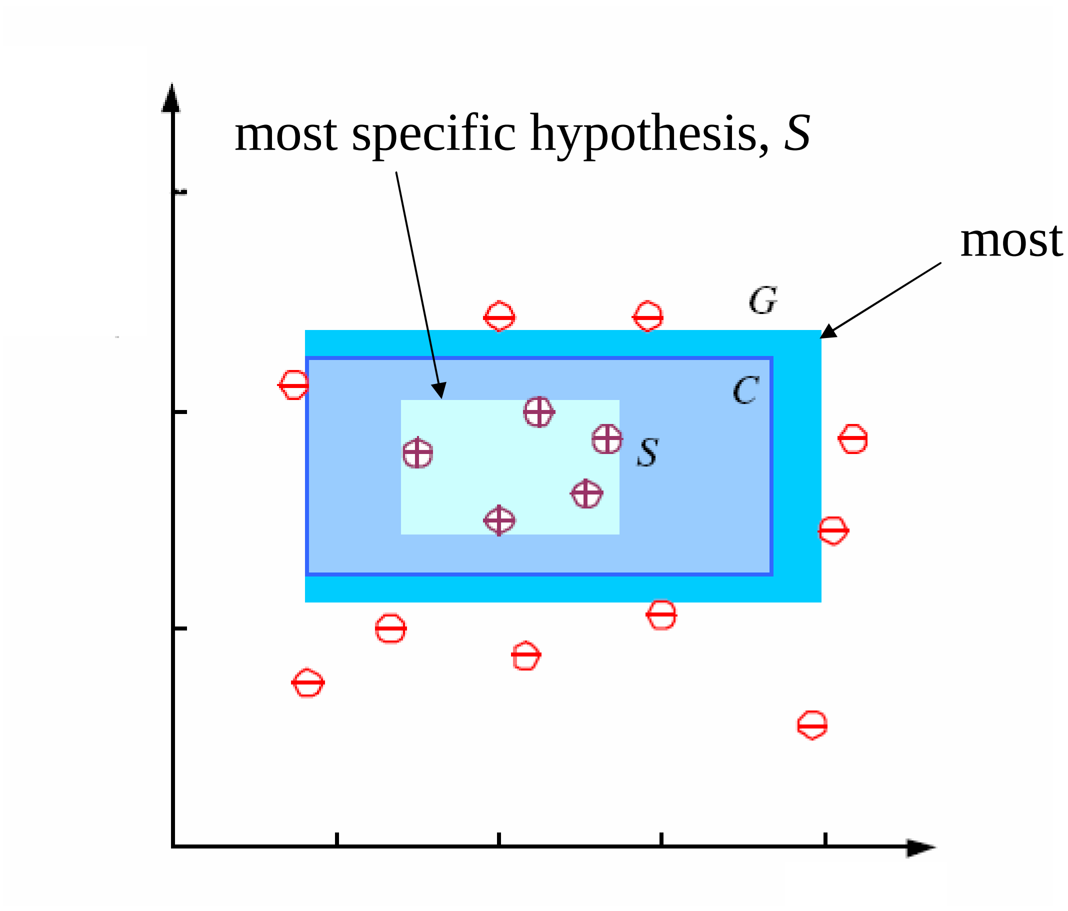
$$\hat{err} = \frac{1}{N} \sum_{n=1}^{N} I(\hat{y}(x_n) \neq y_n)$$

We are approximating p(x,y) by just a finite set of samples

$$p(x,y) = \frac{1}{N} \sum_{n} I(\ (x,y) = (x_n, y_n)\ )$$

Version space

There may be many functions which have zero training error, ranging from the most specific hypothesis to the most general. Which one we pick depends on our prior knowledge.



CNF example of version space

- Let H = formula in conjuctive normal form eg

$$f = (x_1 \wedge \neg x_3) \vee (x_2 \wedge x_4 \wedge \neg x_5)$$

- Suppose we have 3 variables and D is given by

x_1	x_2	x_3	y
1	0	0	1
1	1	0	1
0	0	0	0
0	1	0	1

- Then the most specific and general hyps are

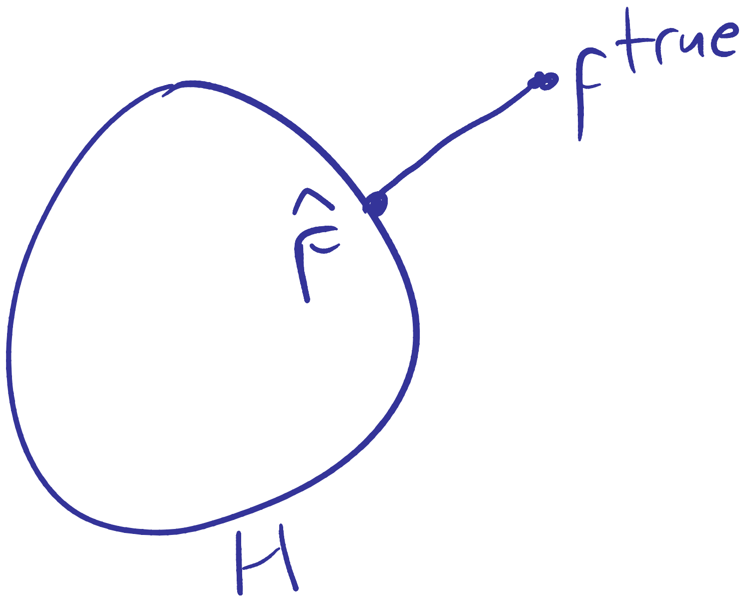
$$S = (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3)$$

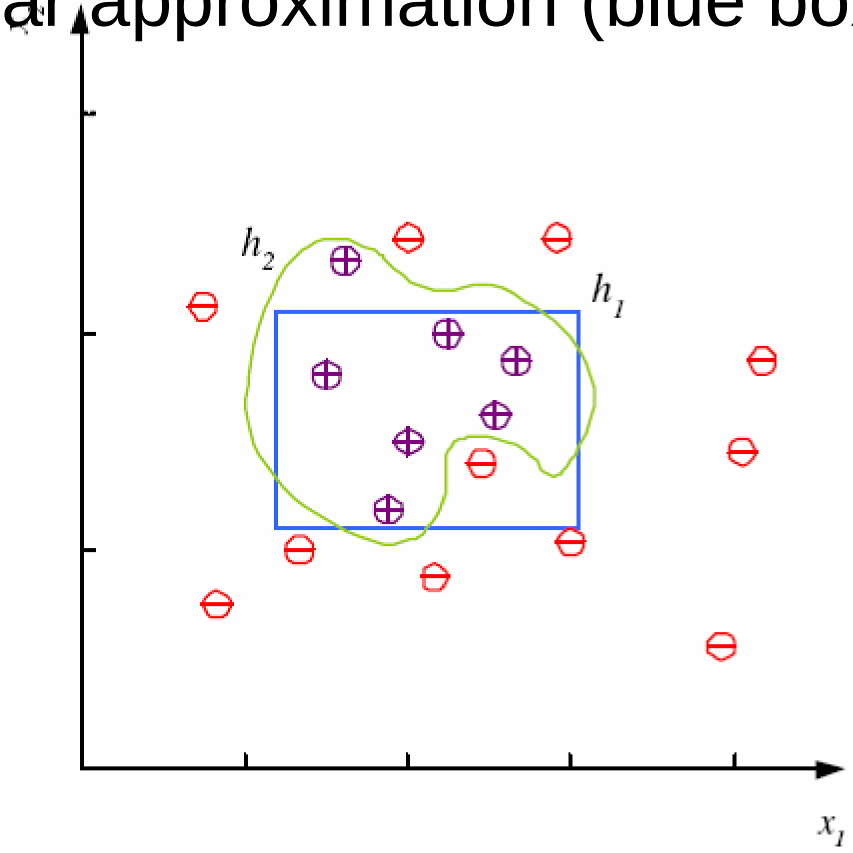
$$G = x_1 \vee x_2$$

Lower bound on achievable error rate

If the true concept (green blob) is a rectangle, we can fit it perfectly, and thus get 0 training error.

But if the truth is more complex, we will just choose the best-fitting rectangular approximation (blue box) and so Nerr ≠ 0





Overfitting in rectangle land

- We can always make the empirical error be 0 by putting a little rectangle around every +ve training example.
- But this may not lead to good generalization performance
- Hence we cannot use empirical error to select between models of different complexity