

# Lecture 11: Learning 2

# Previously...



## Learning

Decision tree learning

Nearest Neighbors

Naive Bayes

Question:

*why we can learn?*

# Classification



what can be observed:

on examples/training data:

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \quad y_i = f(\mathbf{x}_i)$$

e.g. training error

$$\epsilon_t = \frac{1}{m} \sum_{i=1}^m I(h(\mathbf{x}_i) \neq y_i)$$

what is expected:

over the whole distribution: generalization error

$$\begin{aligned} \epsilon_g &= \mathbb{E}_{\mathbf{x}}[I(h(\mathbf{x}) \neq f(\mathbf{x}))] \\ &= \int_{\mathcal{X}} p(\mathbf{x}) I(h(\mathbf{x}) \neq f(\mathbf{x})) d\mathbf{x} \end{aligned}$$

# Regression

what can be observed:

on examples/training data:

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \quad y_i = f(\mathbf{x}_i)$$

e.g. training mean square error/MSE

$$\epsilon_t = \frac{1}{m} \sum_{i=1}^m (h(\mathbf{x}_i) - y_i)^2$$

what is expected:

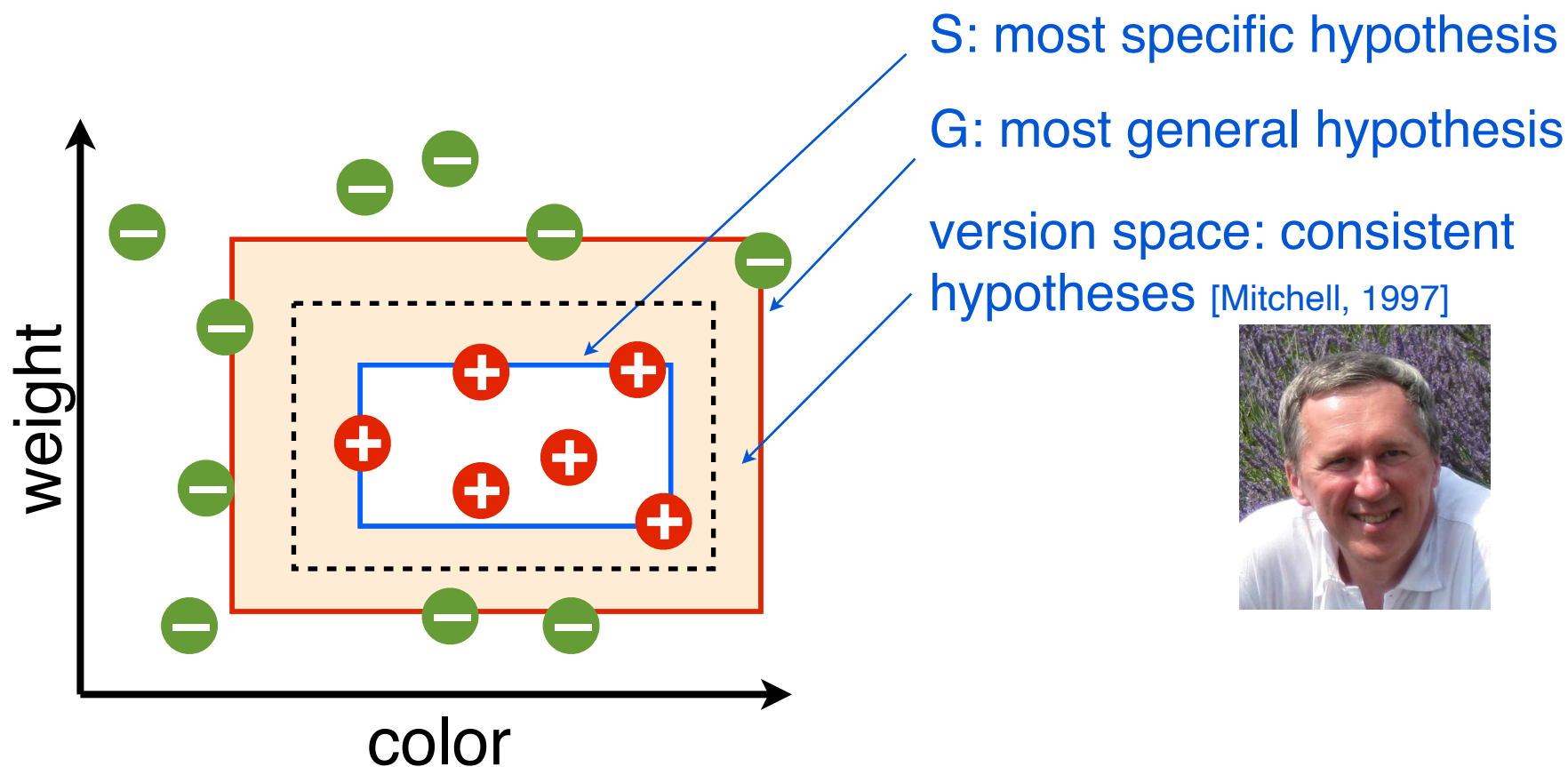
over the whole distribution: generalization MSE

$$\begin{aligned} \epsilon_g &= \mathbb{E}_{\mathbf{x}} (h(\mathbf{x}) - f(\mathbf{x}))^2 \\ &= \int_{\mathcal{X}} p(\mathbf{x}) (h(\mathbf{x}) - f(\mathbf{x}))^2 d\mathbf{x} \end{aligned}$$



# The version space algorithm

an abstract view of learning algorithms

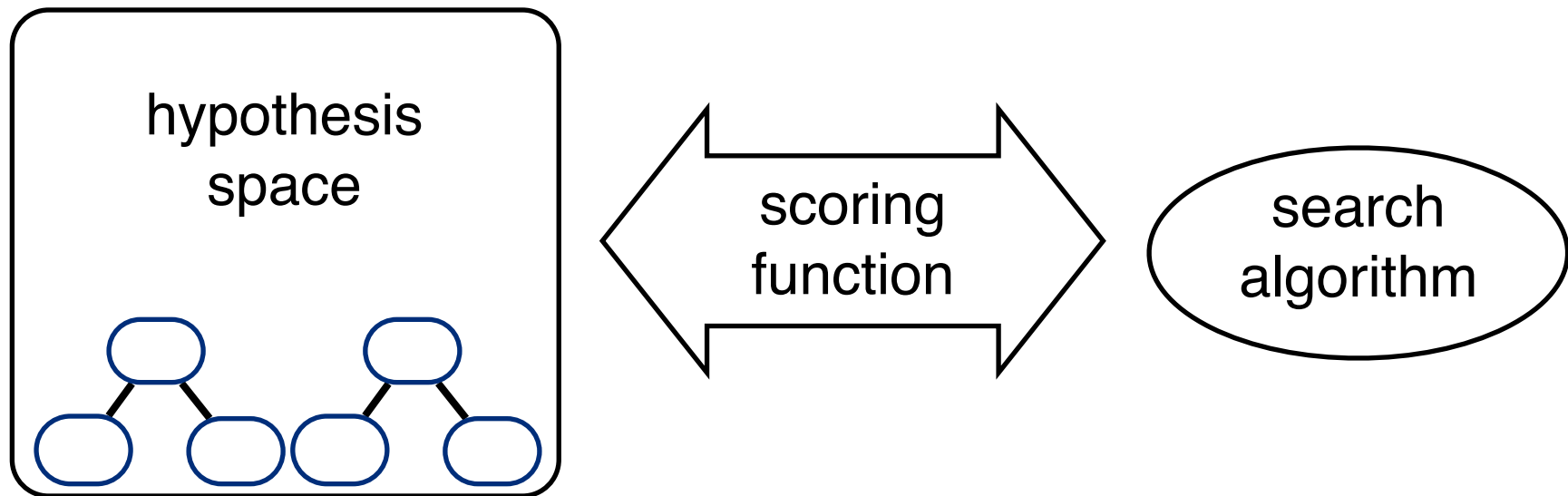


*remove the hypothesis that are inconsistent with the data,  
select a hypothesis according to learner's bias*

# The version space algorithm

an abstract view of learning algorithms

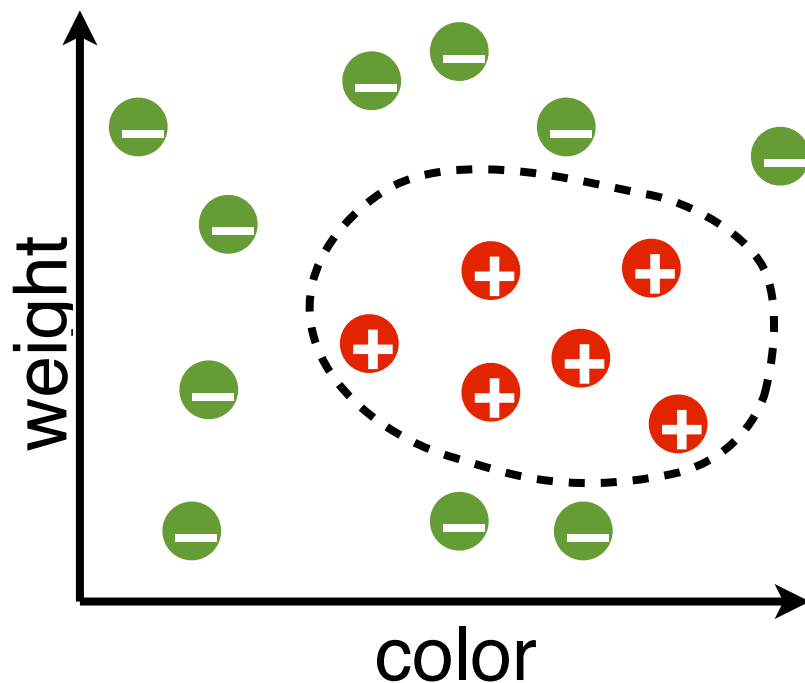
three components of a learning algorithm



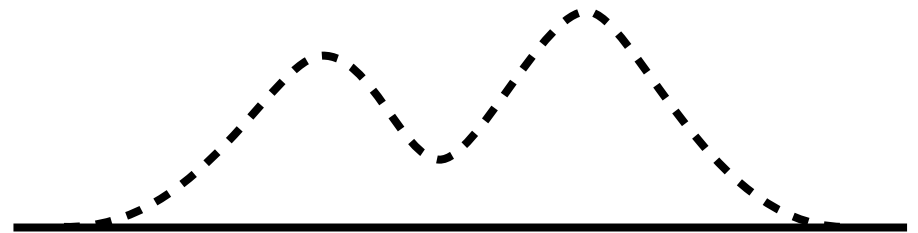
# Theories

The i.i.d. assumption:

all training examples and future (test) examples are drawn *independently* from an *identical distribution*, the label is assigned by a *fixed ground-truth function*



unknown but fixed  
distribution  $D$



# Bias-variance dilemma



Suppose we have 100 training examples  
but there can be different training sets

Start from the expected training MSE:

$$E_D[\epsilon_t] = E_D \left[ \frac{1}{m} \sum_{i=1}^m (h(\mathbf{x}_i) - y_i)^2 \right] = \frac{1}{m} \sum_{i=1}^m E_D [(h(\mathbf{x}_i) - y_i)^2]$$

(assume no noise)

$$\begin{aligned} & E_D [(h(\mathbf{x}) - f(\mathbf{x}))^2] \\ &= E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})] + E_D[h(\mathbf{x})] - f(\mathbf{x}))^2] \\ &= E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2] + E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2] \\ &\quad + E_D [2(h(\mathbf{x}) - E_D[h(\mathbf{x})])(E_D[h(\mathbf{x})] - f(\mathbf{x}))] \\ &= E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2] + E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2] \end{aligned}$$

variance bias<sup>2</sup>

# Bias-variance dilemma

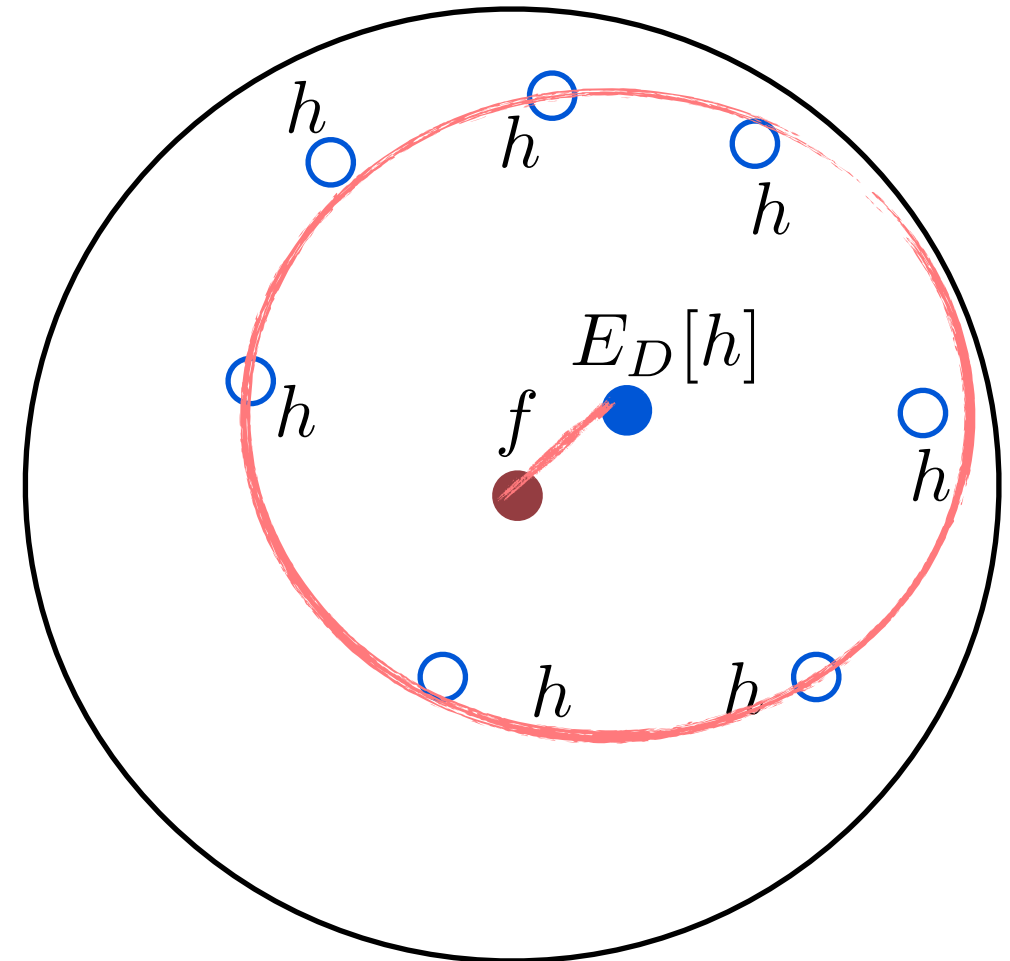
$$E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2]$$

variance

$$E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2]$$

bias<sup>2</sup>

larger hypothesis space  
 $\Rightarrow$   
 lower bias  
 but higher variance



hypothesis space

# Bias-variance dilemma

$$E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2]$$

variance

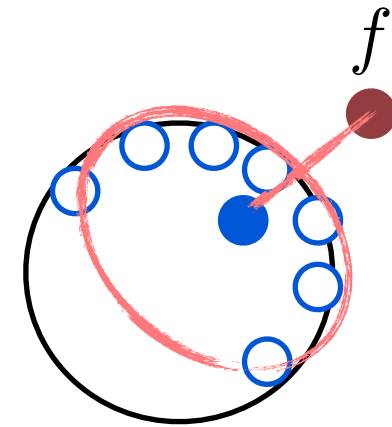
$$E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2]$$

bias<sup>2</sup>

smaller hypothesis space

=>

smaller variance  
but higher bias



hypothesis space

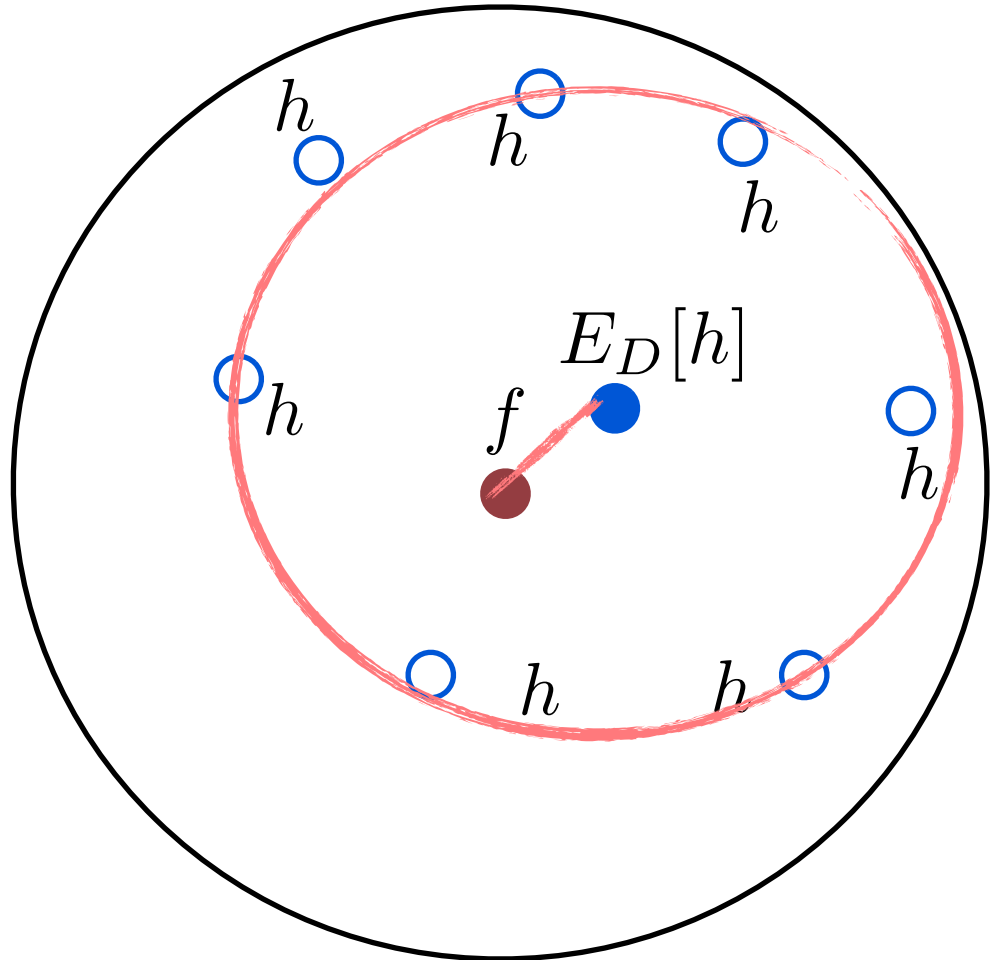
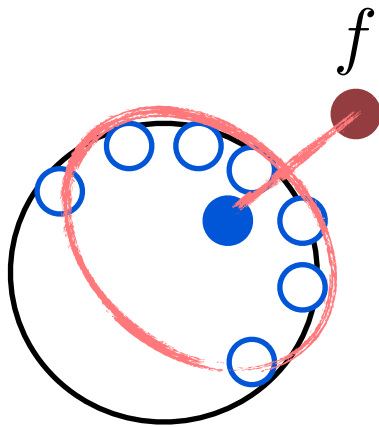
# Bias-variance dilemma

$$E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2]$$

variance

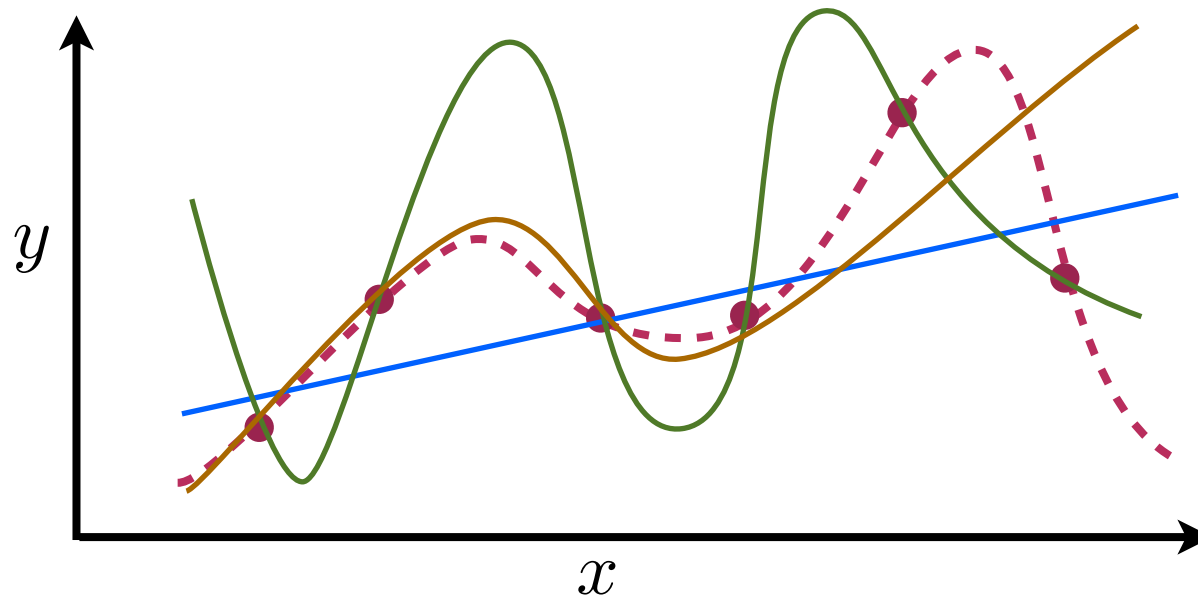
$$E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2]$$

bias<sup>2</sup>



# Overfitting and underfitting

training error v.s. hypothesis space size



linear functions: high training error, small space

$$\{y = a + bx \mid a, b \in \mathbb{R}\}$$

higher polynomials: moderate training error, moderate space

$$\{y = a + bx + cx^2 + dx^3 \mid a, b, c, d \in \mathbb{R}\}$$

even higher order: no training error, large space

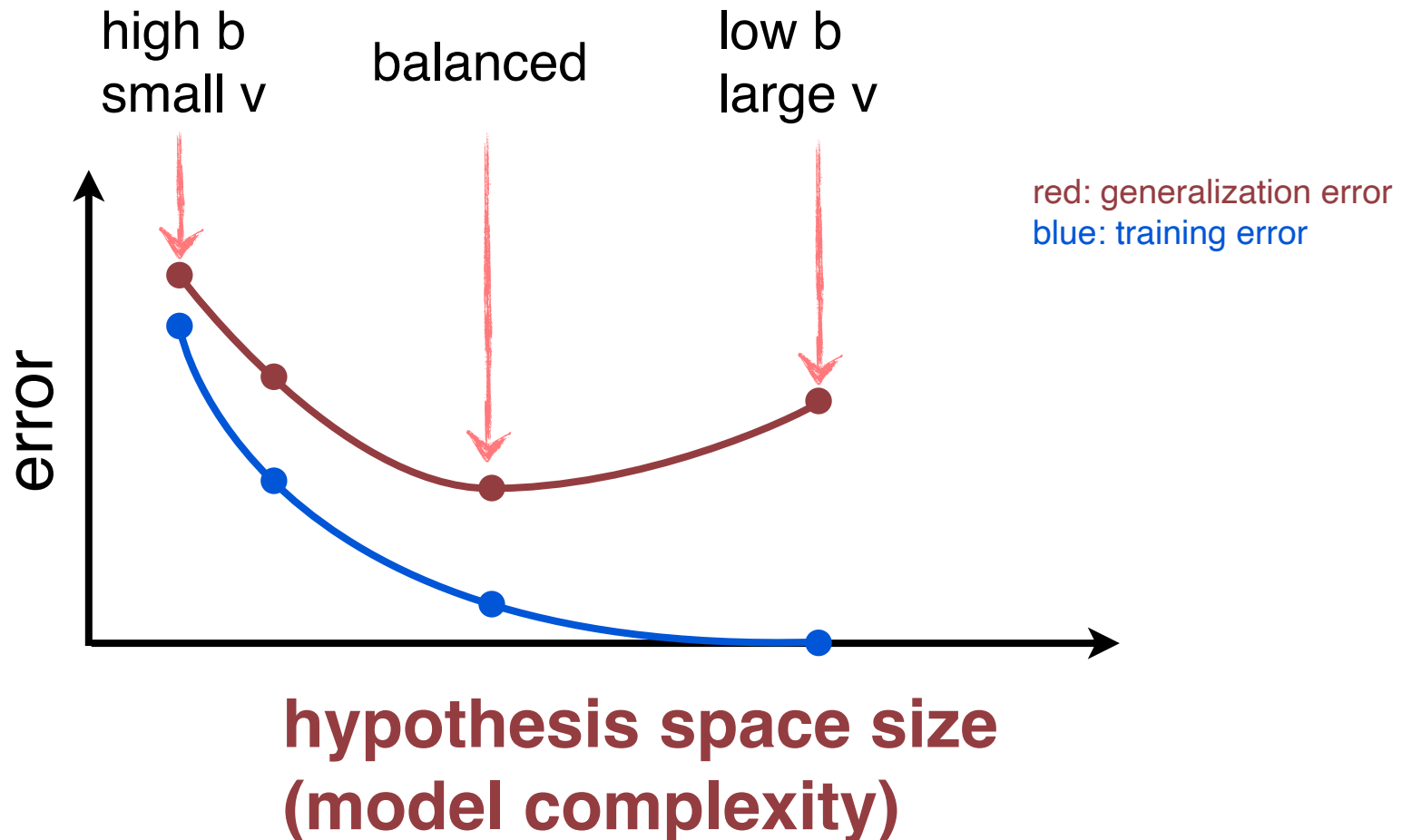
$$\{y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 \mid a, b, c, d, e, f \in \mathbb{R}\}$$



# Overfitting and bias-variance dilemma

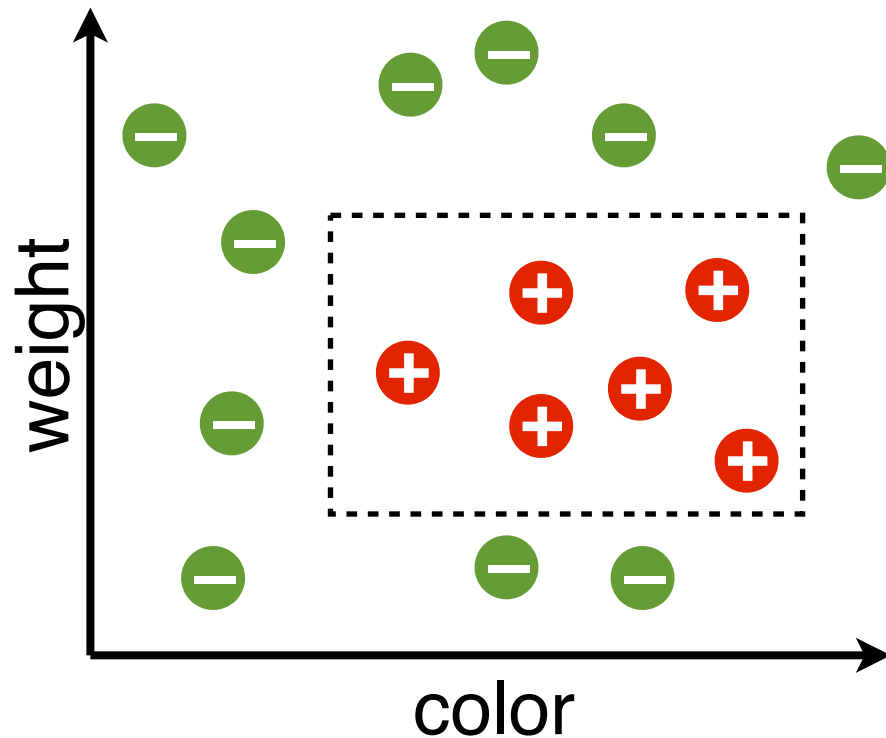
$$E_D [(h(\mathbf{x}) - E_D[h(\mathbf{x})])^2] \quad E_D [(E_D[h(\mathbf{x})] - f(\mathbf{x}))^2]$$

variance bias<sup>2</sup>



# Generalization error

assume i.i.d. examples, and the ground-truth hypothesis is a box



the error of picking a consistent hypothesis:

with probability at least  $1 - \delta$

$$\epsilon_g < \frac{1}{m} \cdot (\ln |\mathcal{H}| + \ln \frac{1}{\delta})$$

smaller generalization error:

- ▶ more examples
- ▶ smaller hypothesis space

# Generalization error

for one  $h$

What is the probability of  $h$  is consistent  
 $\epsilon_g(h) \geq \epsilon$

assume  $h$  is **bad**:  $\epsilon_g(h) \geq \epsilon$

$h$  is consistent with 1 example:

$$P \leq 1 - \epsilon$$

$h$  is consistent with  **$m$**  example:

$$P \leq (1 - \epsilon)^m$$

# Generalization error

$h$  is consistent with  $m$  example:

$$P \leq (1 - \epsilon)^m$$

There are  $k$  consistent hypotheses

Probability of choosing a bad one:

$h_1$  is chosen and  $h_1$  is bad  $P \leq (1 - \epsilon)^m$

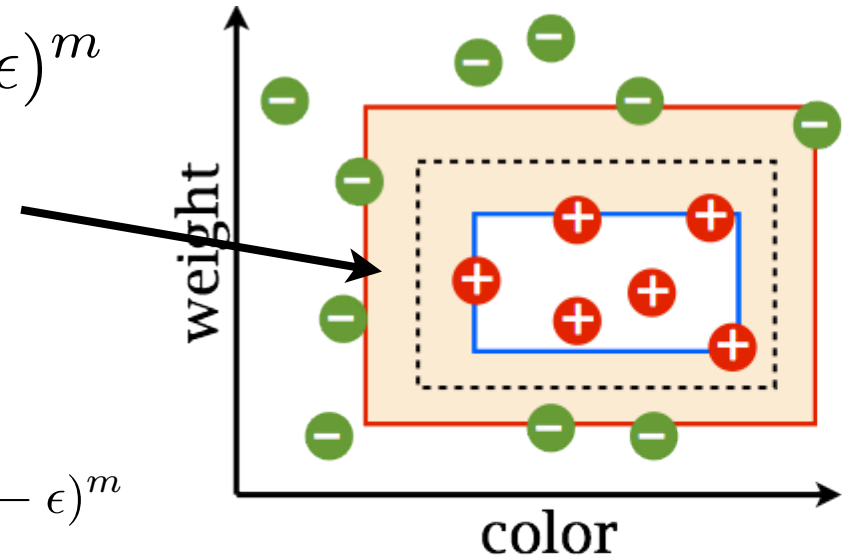
$h_2$  is chosen and  $h_2$  is bad  $P \leq (1 - \epsilon)^m$

...

$h_k$  is chosen and  $h_k$  is bad  $P \leq (1 - \epsilon)^m$

overall:

$\exists h$ :  $h$  can be chosen (consistent) but is bad



# Generalization error



$h_1$  is chosen and  $h_1$  is bad  $P \leq (1 - \epsilon)^m$

$h_2$  is chosen and  $h_2$  is bad  $P \leq (1 - \epsilon)^m$

...

$h_k$  is chosen and  $h_k$  is bad  $P \leq (1 - \epsilon)^m$

overall:

$\exists h$ :  $h$  can be chosen (consistent) but is bad

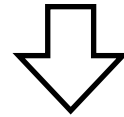
Union bound:  $P(A \cup B) \leq P(A) + P(B)$

$$P(\exists h \text{ is consistent but bad}) \leq k \cdot (1 - \epsilon)^m \leq |\mathcal{H}| \cdot (1 - \epsilon)^m$$

# Generalization error



$$P(\exists h \text{ is consistent but bad}) \leq k \cdot (1 - \epsilon)^m \leq |\mathcal{H}| \cdot (1 - \epsilon)^m$$



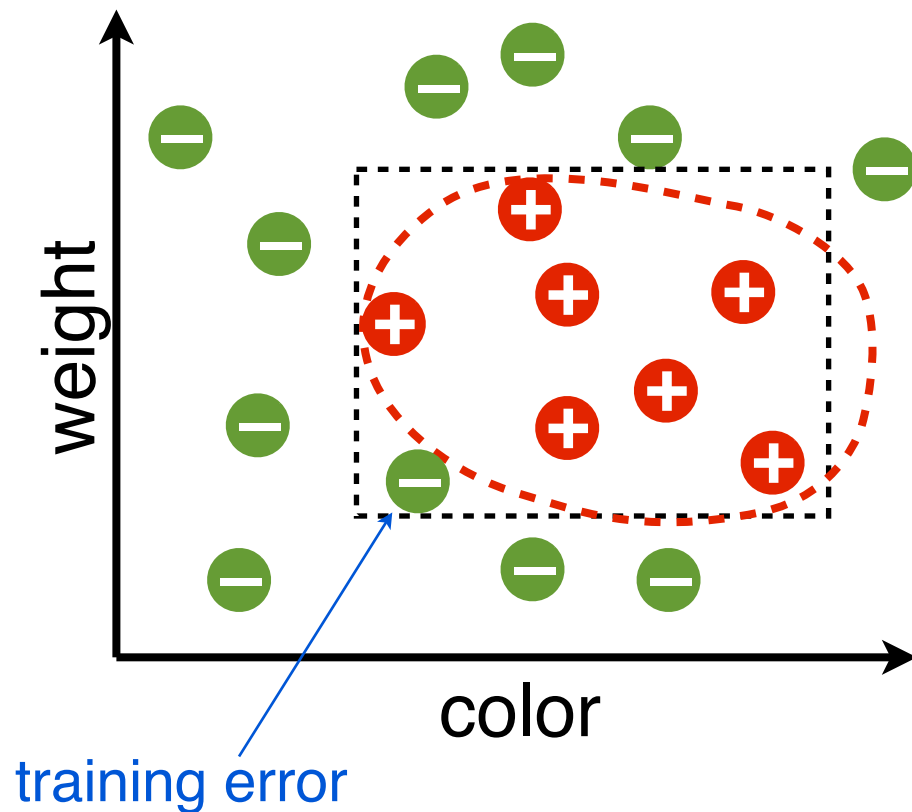
$$P(\epsilon_g \geq \epsilon) \leq \frac{|\mathcal{H}| \cdot (1 - \epsilon)^m}{\delta}$$

with probability at least  $1 - \delta$

$$\epsilon_g < \frac{1}{m} \cdot (\ln |\mathcal{H}| + \ln \frac{1}{\delta})$$

# Inconsistent hypothesis

What if the ground-truth hypothesis is  
NOT a box: **non-zero training error**



smaller generalization error:

with probability at least  $1 - \delta$

$$\epsilon_g < \epsilon_t + \sqrt{\frac{1}{m} (\ln |\mathcal{H}| + \ln \frac{1}{\delta})}$$

- ▶ more examples
- ▶ smaller hypothesis space
- ▶ **smaller training error**

# Hoeffding's inequality

$X$  be an i.i.d. random variable

$X_1, X_2, \dots, X_m$  be  $m$  samples  $X_i \in [a, b]$

$\frac{1}{m} \sum_{i=1}^m X_i - \mathbb{E}[X] \leftarrow$  difference between sum and expectation

$$P\left(\frac{1}{m} \sum_{i=1}^m X_i - \mathbb{E}[X] \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2 m}{(b-a)^2}\right)$$



# Generalization error



for one  $h$

$$X_i = I(h(x_i) \neq f(x_i)) \in [0, 1]$$

$$\frac{1}{m} \sum_{i=1}^m X_i \rightarrow \epsilon_t(h) \qquad \mathbb{E}[X_i] \rightarrow \epsilon_g(h)$$

$$P(\epsilon_t(h) - \epsilon_g(h) \geq \epsilon) \leq \exp(-2\epsilon^2 m)$$

$$\begin{aligned} &P(\epsilon_t - \epsilon_g \geq \epsilon) \\ &\leq P(\exists h \in |\mathcal{H}| : \epsilon_t(h) - \epsilon_g(h) \geq \epsilon) \leq \frac{|\mathcal{H}| \exp(-2\epsilon^2 m)}{\delta} \end{aligned}$$

with probability at least  $1 - \delta$

$$\epsilon_g < \epsilon_t + \sqrt{\frac{1}{2m} \cdot (\ln |\mathcal{H}| + \ln \frac{1}{\delta})}$$

# Generalization error: Summary



assume i.i.d. examples

consistent hypothesis case:

with probability at least  $1 - \delta$

$$\epsilon_g < \frac{1}{m} \cdot (\ln |\mathcal{H}| + \ln \frac{1}{\delta})$$

inconsistent hypothesis case:

with probability at least  $1 - \delta$

$$\epsilon_g < \epsilon_t + \sqrt{\frac{1}{m} (\ln |\mathcal{H}| + \ln \frac{1}{\delta})}$$

generalization error:

number of examples  $m$

training error  $\epsilon_t$

hypothesis space complexity

$\ln |\mathcal{H}|$

## Probably approximately correct (PAC):

with probability at least  $1 - \delta$

$$\epsilon_g < \epsilon_t + \sqrt{\frac{1}{2m} \cdot (\ln |\mathcal{H}| + \ln \frac{1}{\delta})}$$



**Leslie Valiant**

Turing Award (2010)  
EATCS Award (2008)  
Knuth Prize (1997)  
Nevanlinna Prize (1986)

## **PAC-learnable:** [Valiant, 1984]

A concept class  $\mathcal{C}$  is PAC-learnable if exists a learning algorithm  $A$  such that for all.  $f \in \mathcal{C}$ ,  $\epsilon > 0$ ,  $\delta > 0$  and distribution  $D$

$$P_D(\epsilon_g \leq \epsilon) \geq 1 - \delta$$

using  $m = \text{poly}(1/\epsilon, 1/\delta)$  examples and polynomial time.

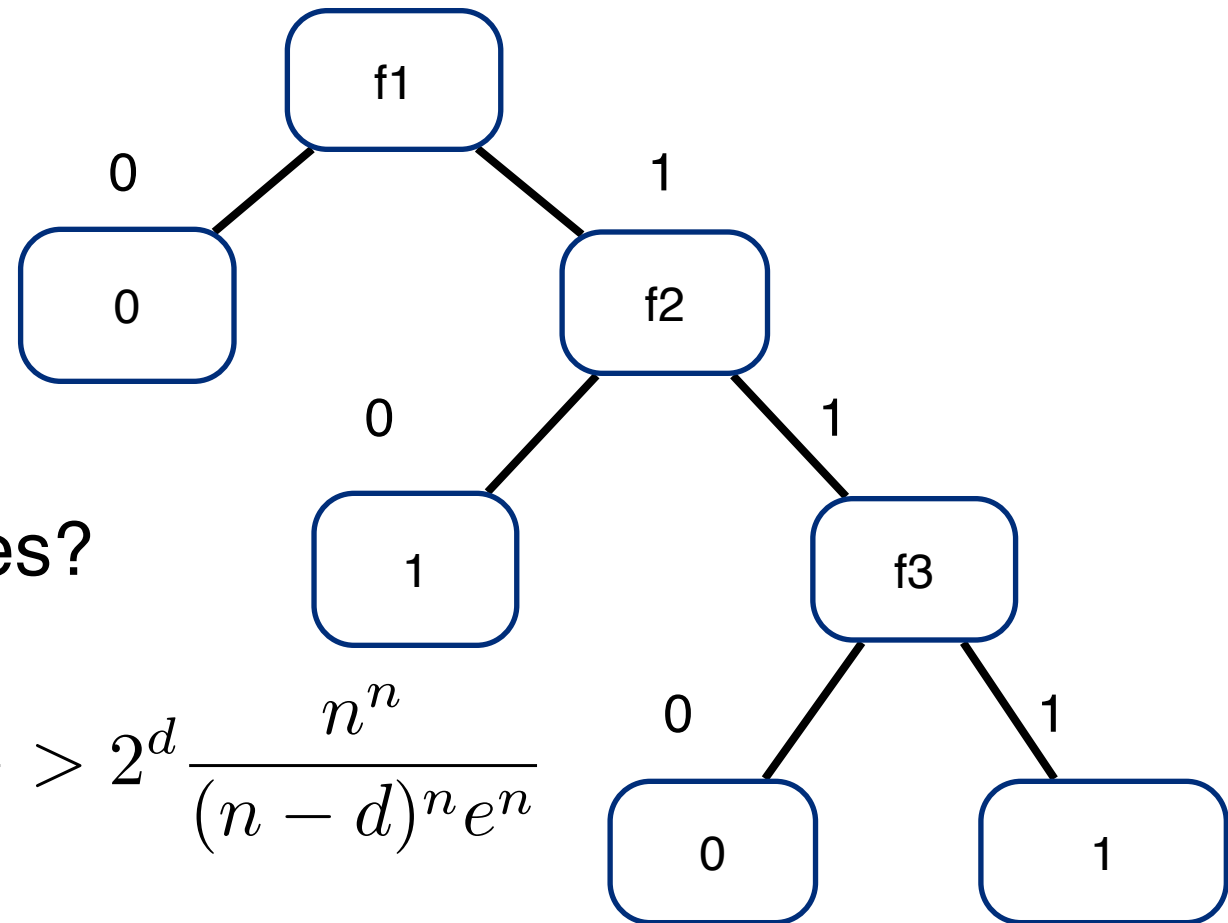
# Learning algorithms revisit



## Decision Tree

# Tree depth and the possibilities

features:  $n$   
feature type: binary  
depth:  $d < n$



How many different trees?

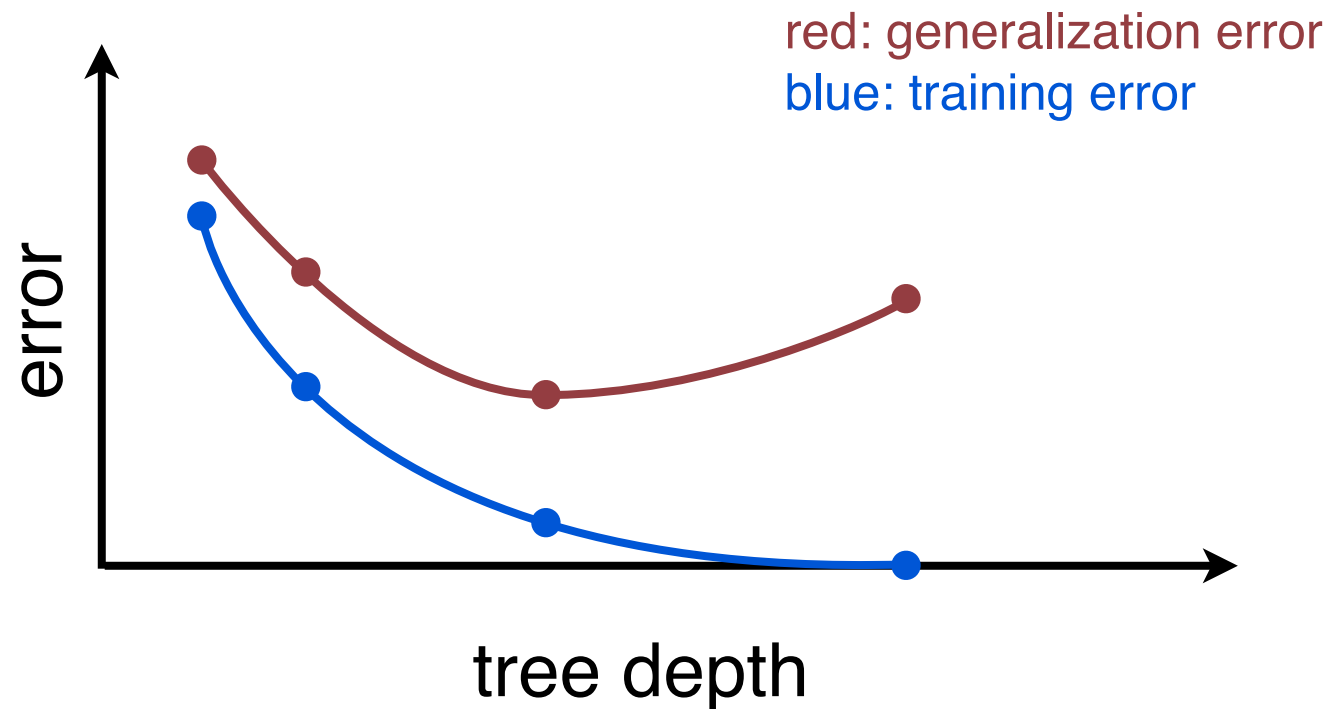
one-branch:  $2^d \frac{n!}{(n-d)!} > 2^d \frac{n^n}{(n-d)^n e^n}$

full-tree:  $2^{2^d} \prod_{i=0}^{d-1} \frac{(n-i)!}{(n-d-i)!}$

the possibility of trees grows very fast with  $d$

# The overfitting phenomena

-- the divergence between infinite and finite samples



To make decision tree less complex

**Pre-pruning:** early stop

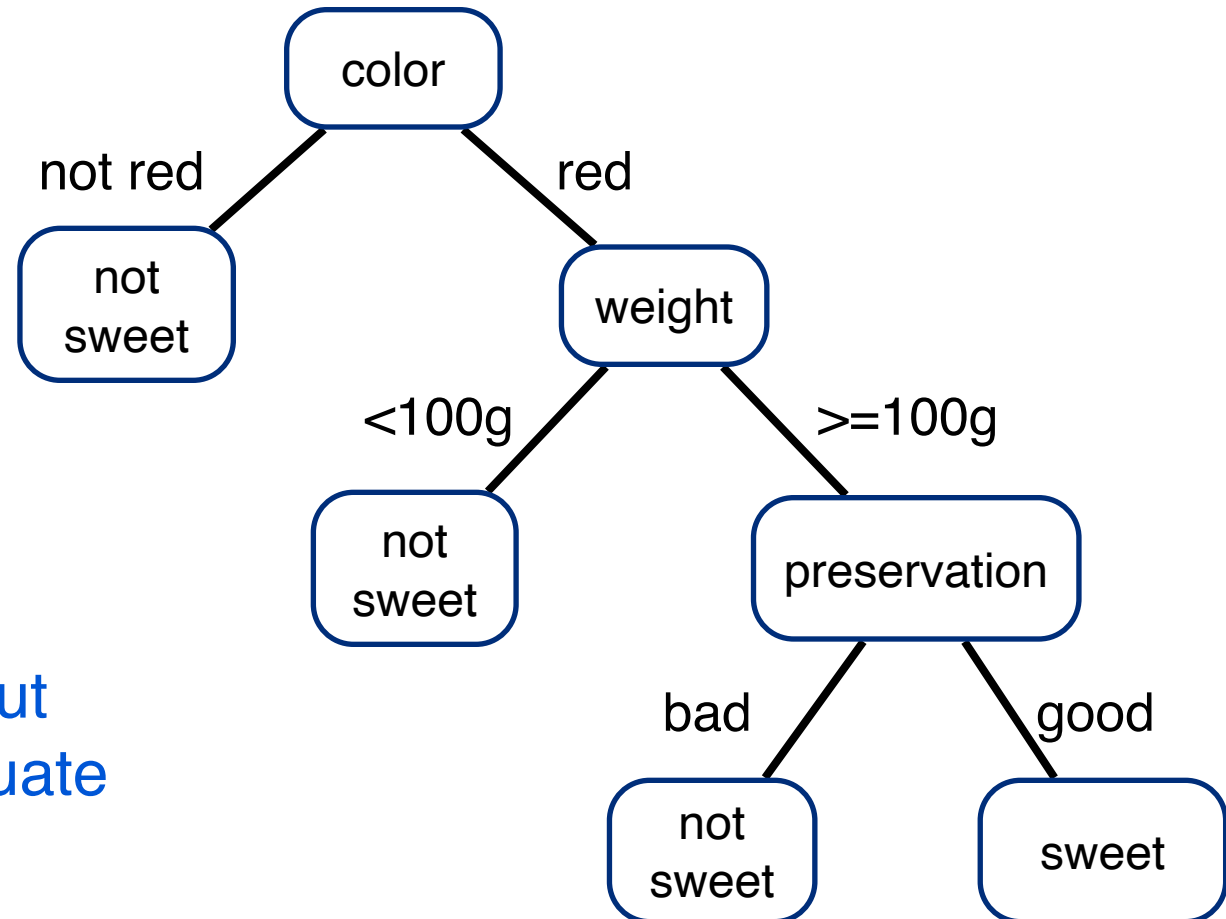
- ▶ minimum data in leaf
- ▶ maximum depth
- ▶ maximum accuracy

**Post-pruning:** prune full grown DT

reduced error pruning

# Reduced error pruning

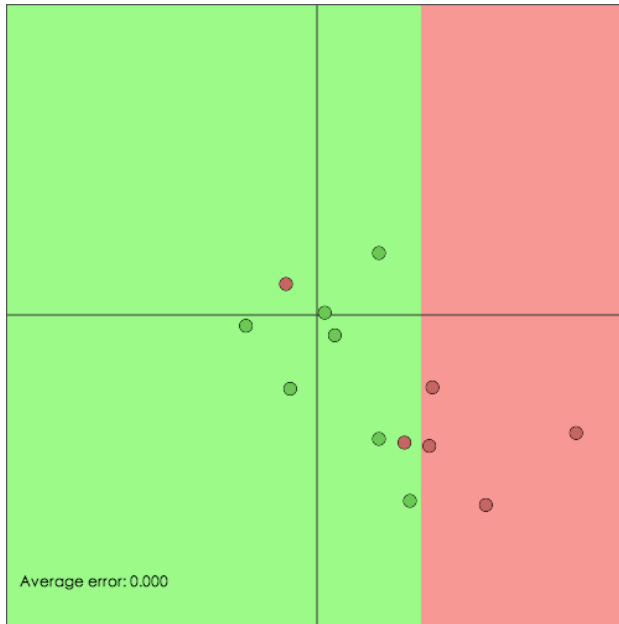
1. Grow a decision tree
2. For every node starting from the leaves
3. Try to make the node leaf, if does not increase the error, keep as the leaf



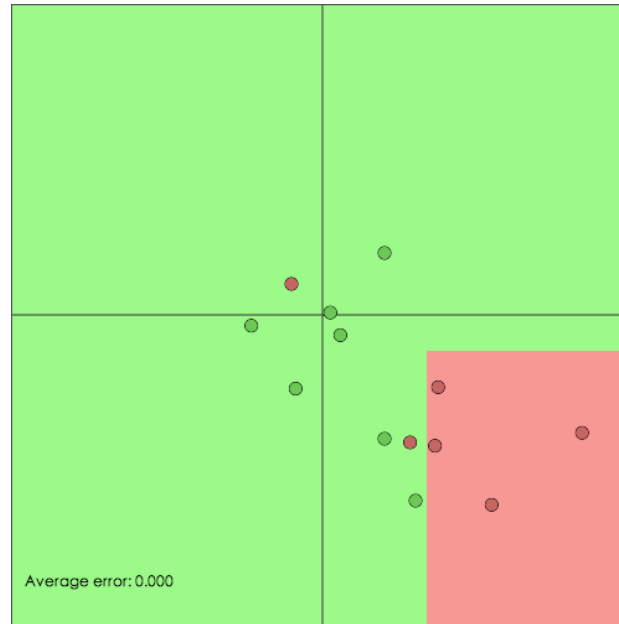
could split a validation set out from the training set to evaluate the error



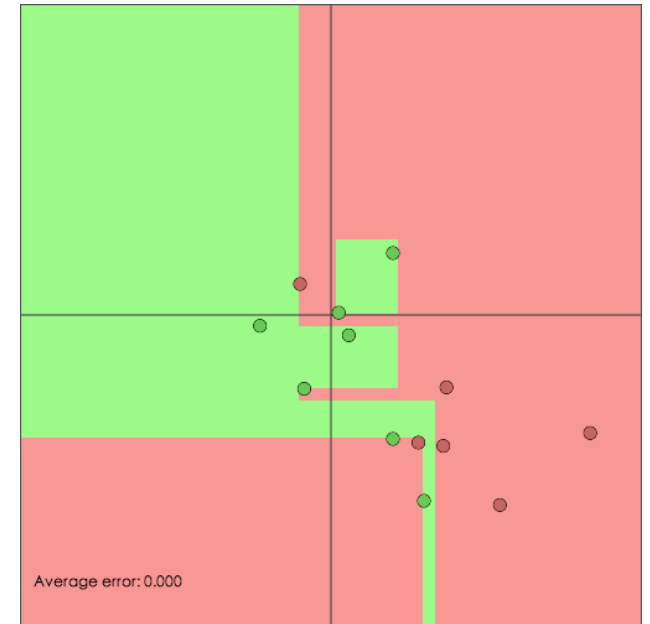
# DT boundary visualization



decision stump



max depth=2



max depth=12

# Oblique decision tree

choose a linear combination in each node:

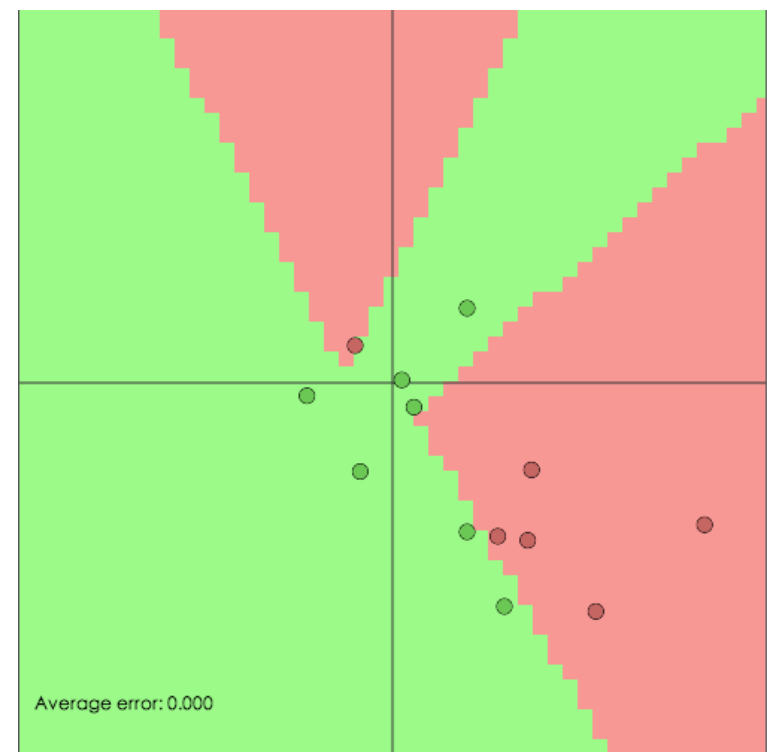
axis parallel:

$$X_1 > 0.5$$

oblique:

$$0.2 X_1 + 0.7 X_2 + 0.1 X_3 > 0.5$$

*was hard to train*



# Linear Models

# Linear model

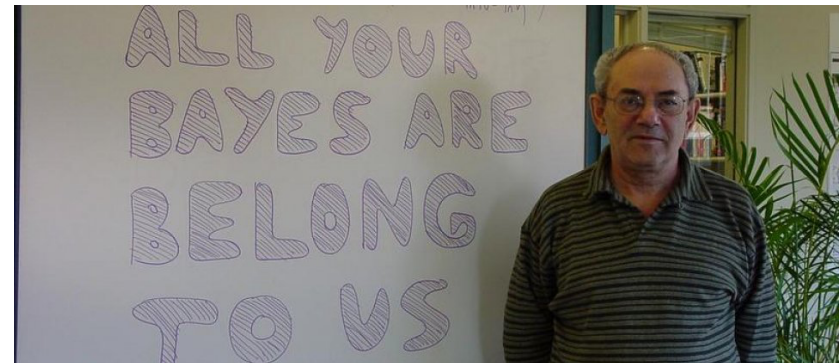
$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

$$\mathbf{w} = w_1, w_2, \dots, w_n \quad b$$



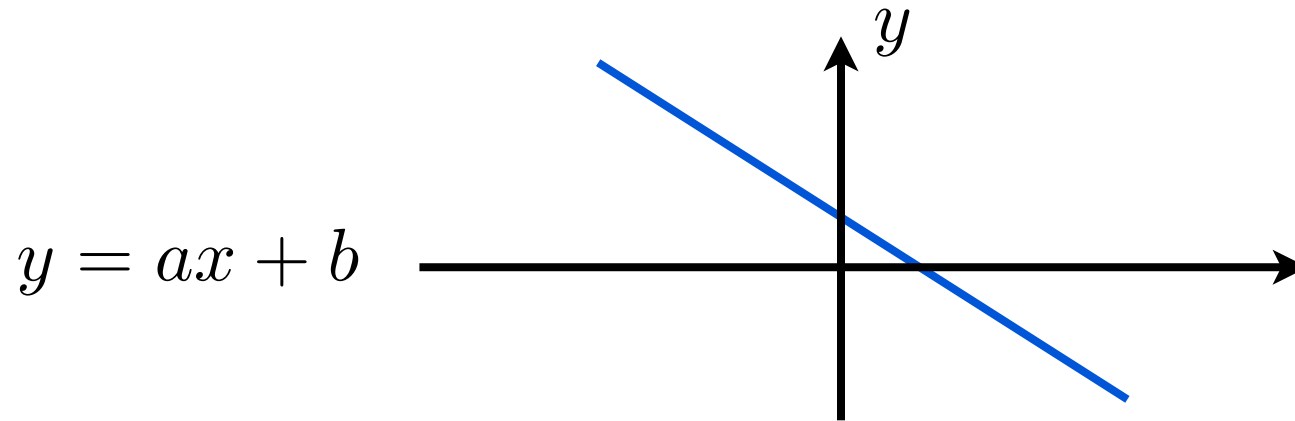
$$w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + b$$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

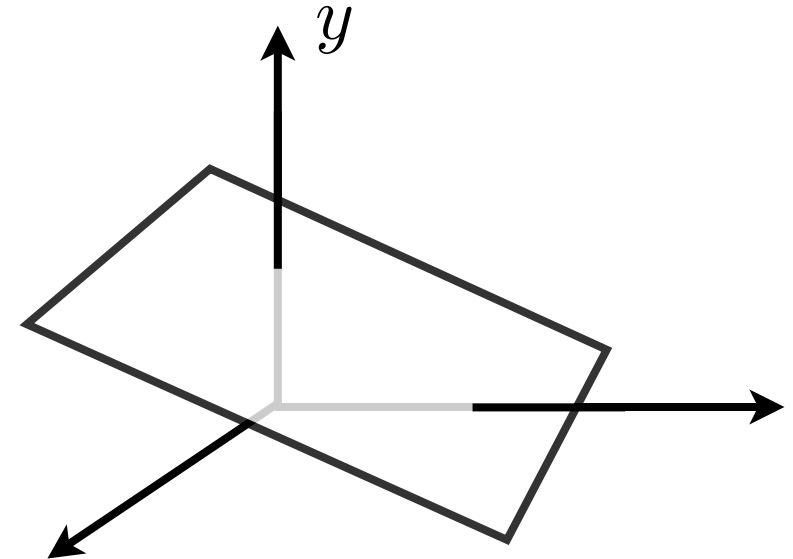


Vladimir Vapnik

# Linear model



$$y = w_1 \cdot x_1 + w_2 \cdot x_2 + b$$



is the following a linear model?

$$y = w_1 \cdot x + w_2 \cdot x^2 + b$$

yes, the parameters  
are linear

# Least square regression

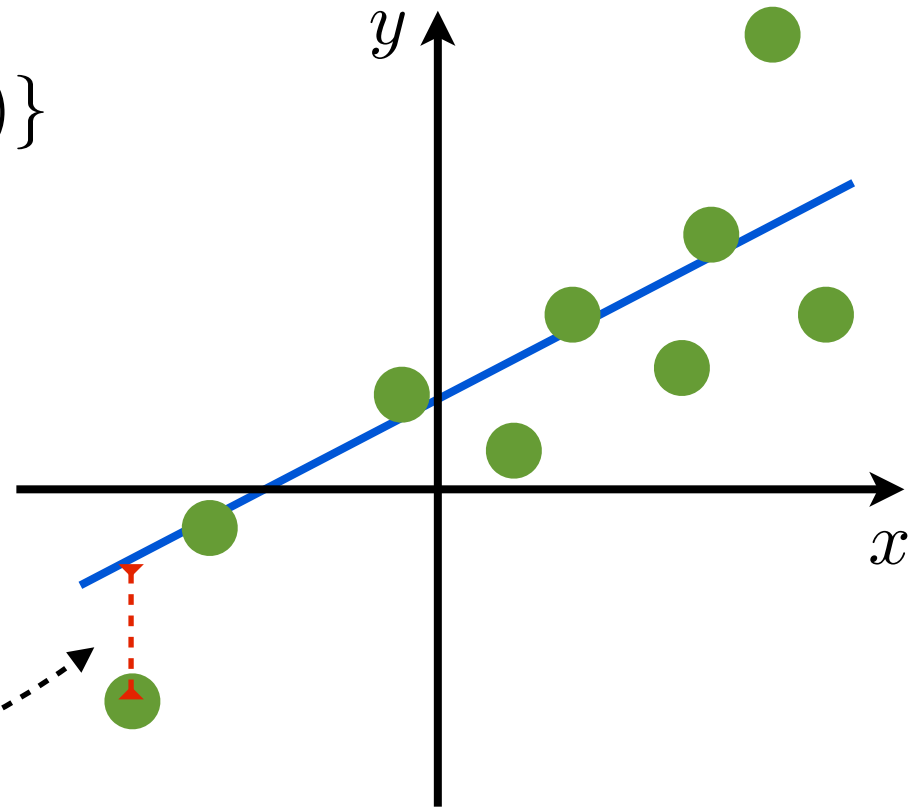
Regression:  $y \in \mathbb{R}$

Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_m, y_m)\}$$

Least square loss:

$$\frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$



# Least square regression

$$L(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

$$\frac{\partial L(\mathbf{w}, b)}{\partial b} = \frac{1}{m} \sum_{i=1}^m 2(\mathbf{w}^\top \mathbf{x}_i + b - y_i) = 0$$

$$\frac{\partial L(\mathbf{w}, b)}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=1}^m 2(\mathbf{w}^\top \mathbf{x}_i + b - y_i) \mathbf{x}_i^\top = 0$$

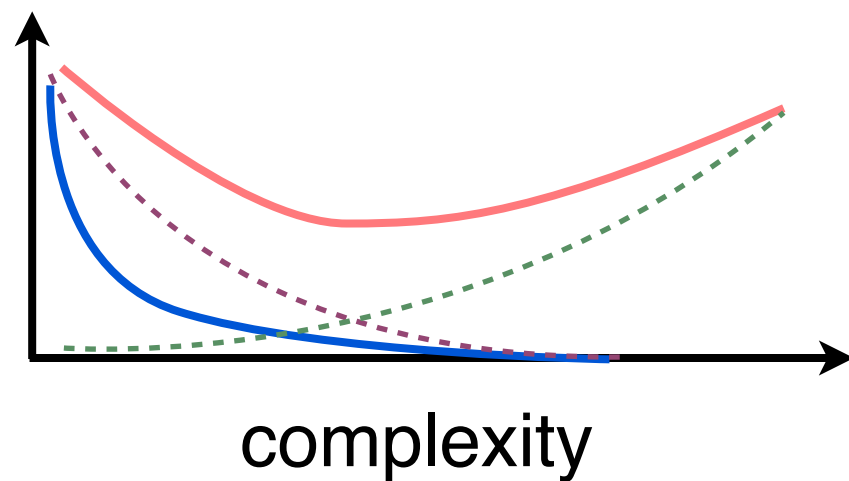
$$b = \frac{1}{m} \sum_{i=1}^m (y_i - \mathbf{w}^\top \mathbf{x}_i) = \bar{y} - \mathbf{w}^\top \bar{\mathbf{x}}$$

$$\mathbf{w} = \left( \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top - \bar{\mathbf{x}} \bar{\mathbf{x}}^\top \right)^{-1} \left( \frac{1}{m} \sum_{i=1}^m (y_i \mathbf{x}_i) - \bar{y} \bar{\mathbf{x}} \right)$$

$$= \text{var}(\mathbf{x})^{-1} \text{cov}(\mathbf{x}, y) = (X^\top X)^{-1} X^\top Y$$

*closed  
form  
solution*

# Complexity of linear models



$$f(x) = w^{\top} x$$

possibility of  $w$



# Regularization

make hypothesis space small  
→ better generalization ability  
make numerical analysis stable

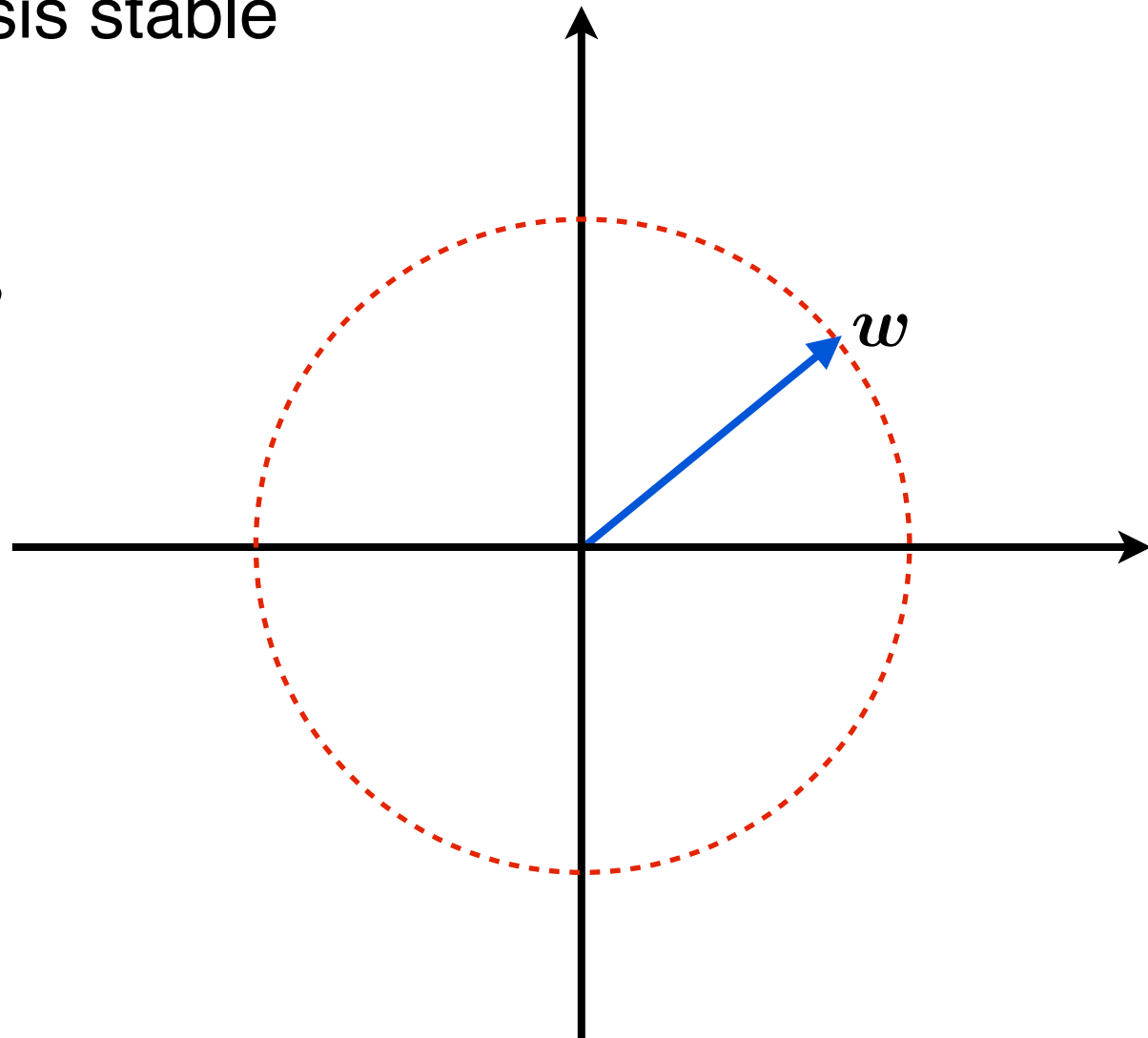
restrict the norm of  $w$

$$\|w\|_p = \left( \sum_{i=1}^n |w_i|^p \right)^{1/p}$$

$$\|w\|_2 = \sqrt{\sum_{i=1}^n w_i^2}$$

$$\|w\|_1 = \sum_{i=1}^n |w_i|$$

$$\|w\|_\infty = \max_{i=1, \dots, n} |w_i|$$



# Ridge regression

Regression:  $y \in \mathbb{R}$

Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_m, y_m)\}$$

objective:

$$\arg \min_{\mathbf{w}, b} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

$$s.t. \quad \|\mathbf{w}\|_2 \leq \theta$$

or:

$$\arg \min_{\mathbf{w}, b} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 + \lambda \|\mathbf{w}\|_2$$

# Ridge regression

centered data, no bias:

$$\arg \min_{\mathbf{w}} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2$$

closed form solution:

$$\mathbf{w} = \left( \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top - \bar{\mathbf{x}} \bar{\mathbf{x}}^\top + \lambda \mathbf{I} \right)^{-1} \left( \frac{1}{m} \sum_{i=1}^m (y_i \mathbf{x}_i) - \bar{y} \bar{\mathbf{x}} \right)$$

$$= (\text{var}(\mathbf{x}) + \lambda \mathbf{I})^{-1} \text{cov}(\mathbf{x}, y)$$

$$= (X^\top X + \lambda I)^{-1} X^\top Y$$

$I$  is the identity matrix

# Least square v.s. ridge regression

$$\begin{aligned} \mathbf{w} &= \left( \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top - \bar{\mathbf{x}} \bar{\mathbf{x}}^\top \right)^{-1} \left( \frac{1}{m} \sum_{i=1}^m (y_i \mathbf{x}_i) - \bar{y} \bar{\mathbf{x}} \right) \\ &= \text{var}(\mathbf{x})^{-1} \text{cov}(\mathbf{x}, y) = (X^\top X)^{-1} X^\top Y \end{aligned}$$

$$\begin{aligned} \mathbf{w} &= \left( \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top - \bar{\mathbf{x}} \bar{\mathbf{x}}^\top + \lambda \mathbf{I} \right)^{-1} \left( \frac{1}{m} \sum_{i=1}^m (y_i \mathbf{x}_i) - \bar{y} \bar{\mathbf{x}} \right) \\ &= (\text{var}(\mathbf{x}) + \lambda \mathbf{I})^{-1} \text{cov}(\mathbf{x}, y) \\ &= (X^\top X + \lambda \mathbf{I})^{-1} X^\top Y \end{aligned}$$



stable solution

# Least absolute shrinkage and selection operator (LASSO)

Regression:  $y \in \mathbb{R}$

Training data:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_m, y_m)\}$$

objective:

$$\arg \min_{\mathbf{w}, b} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

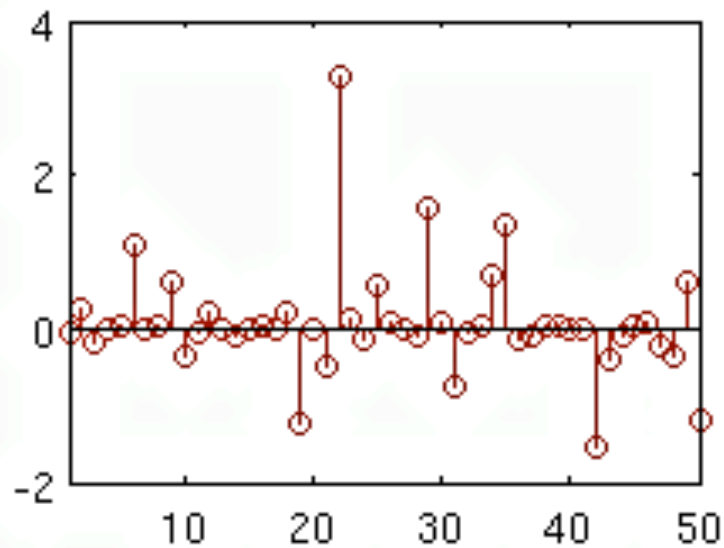
$$s.t. \quad \|\mathbf{w}\|_1 \leq \theta$$

or:

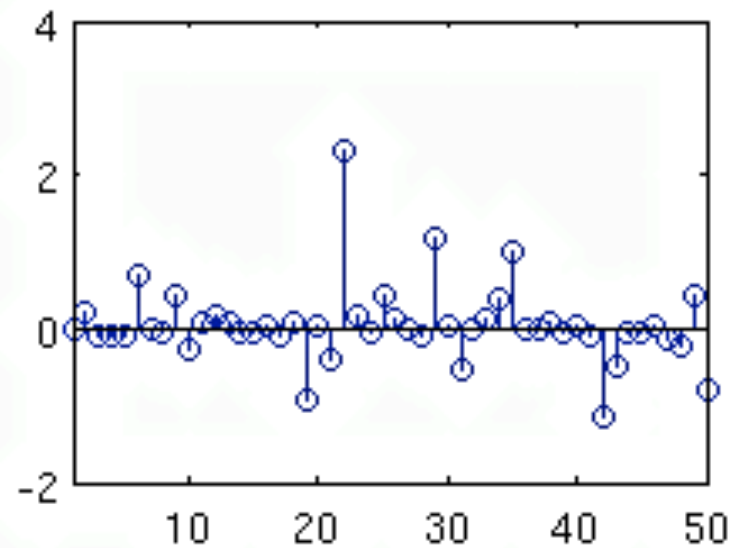
$$\arg \min_{\mathbf{w}, b} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 + \lambda \|\mathbf{w}\|_1$$

# Comparing different regressions

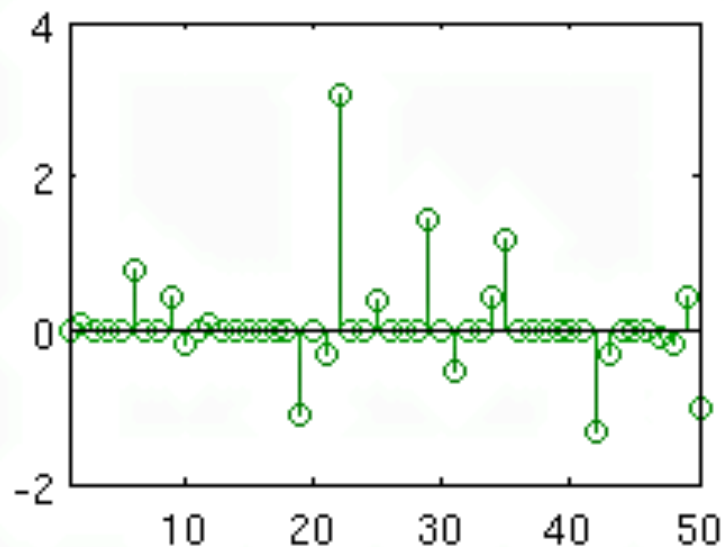
Least Squares



Ridge Regression



LASSO



[Pictures from [www.cs.ubc.ca/~schmidtm/Software/L1General/examples.html](http://www.cs.ubc.ca/~schmidtm/Software/L1General/examples.html)]

# A general framework

objective function:

$$\arg \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b) + \|\boldsymbol{w}\|_p$$

how to solve the parameters?

a generally applied technique: **gradient-descent**

# Gradient descent

(steepest descent)

for a differentiable function  $f$

$$\arg \min_w f(w)$$

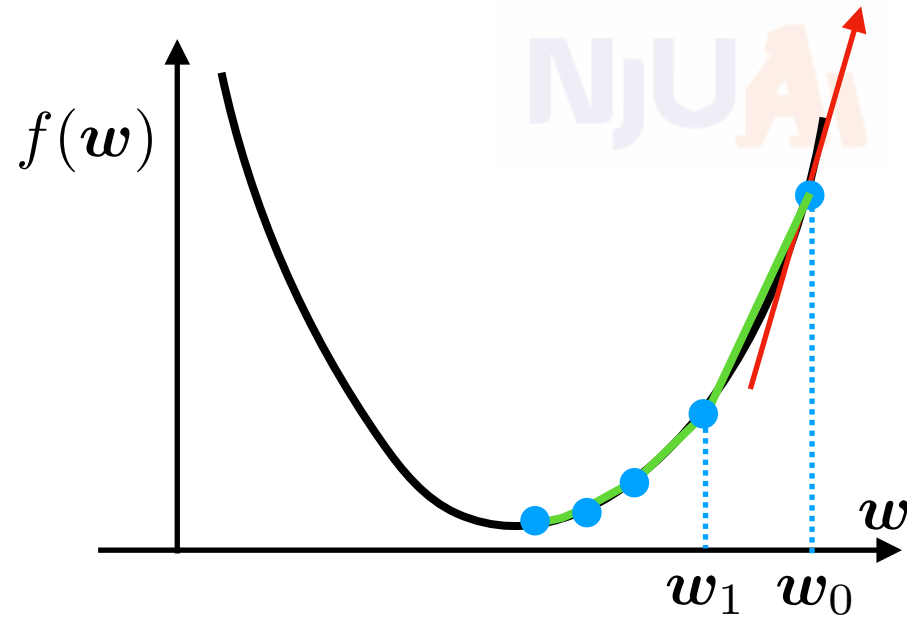
can be solved by

1. start from an arbitrary initial point
2. loop from  $t=0$

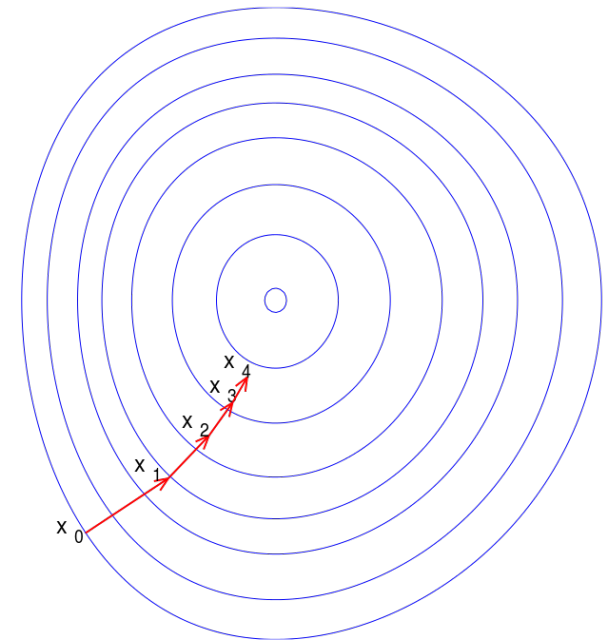
3. 
$$w_{t+1} = w - \eta \frac{\partial f(w)}{\partial w}$$

or 
$$w_{t+1} = w - \eta \nabla_w f(w)$$

4. until convergence  $\|\nabla_w f(w)\| < \epsilon$



$w_0$

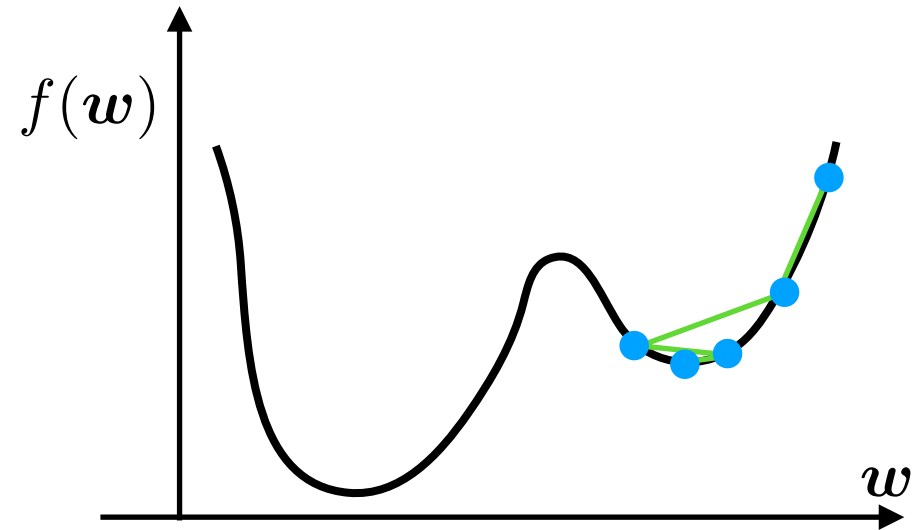
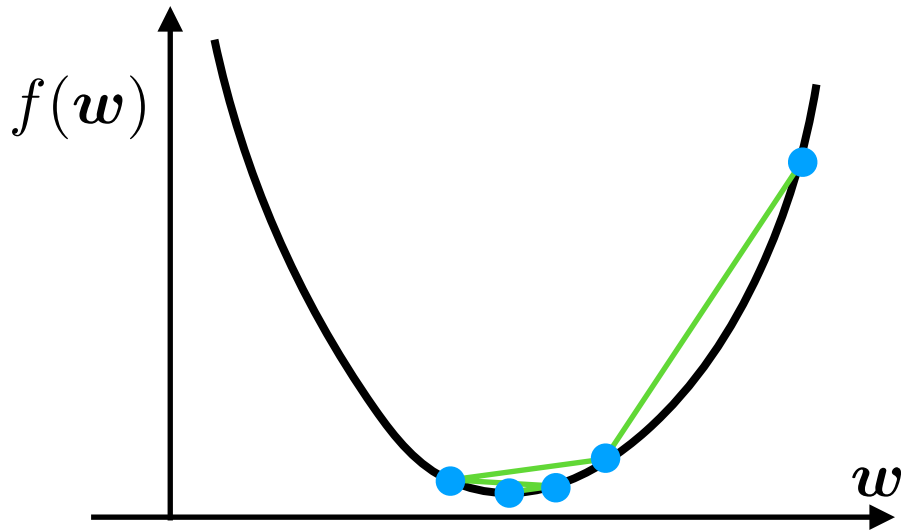


[image from wikipedia]



# Gradient descent

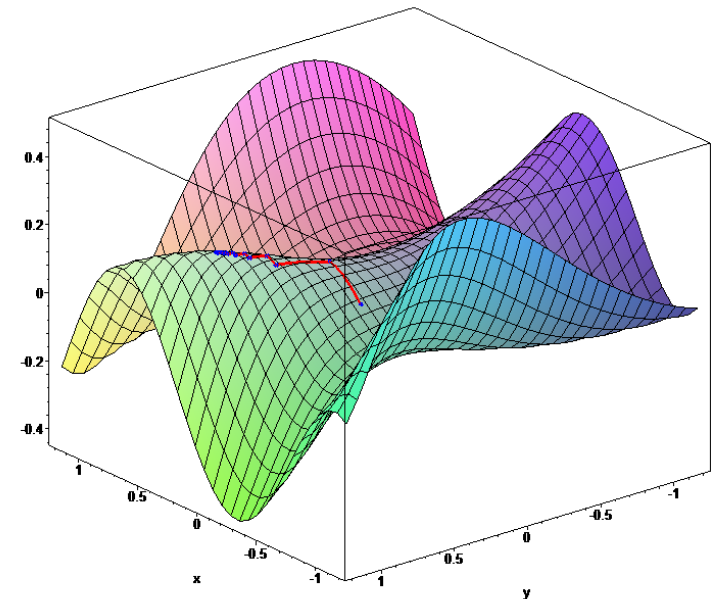
$$\mathbf{w}_{t+1} = \mathbf{w} - \eta \nabla_{\mathbf{w}} f(\mathbf{w})$$



for convex functions: converge  
to global optima

$$f(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2) \geq \alpha f(\mathbf{w}_1) + (1 - \alpha) f(\mathbf{w}_2)$$

for other functions:  
converge to stationary points



[image from wikipedia]

# A general framework

objective function:

$$\arg \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b) + \|\boldsymbol{w}\|_p$$

how to solve the parameters?

general optimization: gradient descent

$$(\boldsymbol{w}, b)_{-} = \eta \frac{\partial (L(\boldsymbol{w}, b) + \|\boldsymbol{w}\|_p)}{\partial (\boldsymbol{w}, b)}$$

# Linear classifier

model space:  $\mathbb{R}^{n+1}$

$$f(x) = w^\top x + b$$

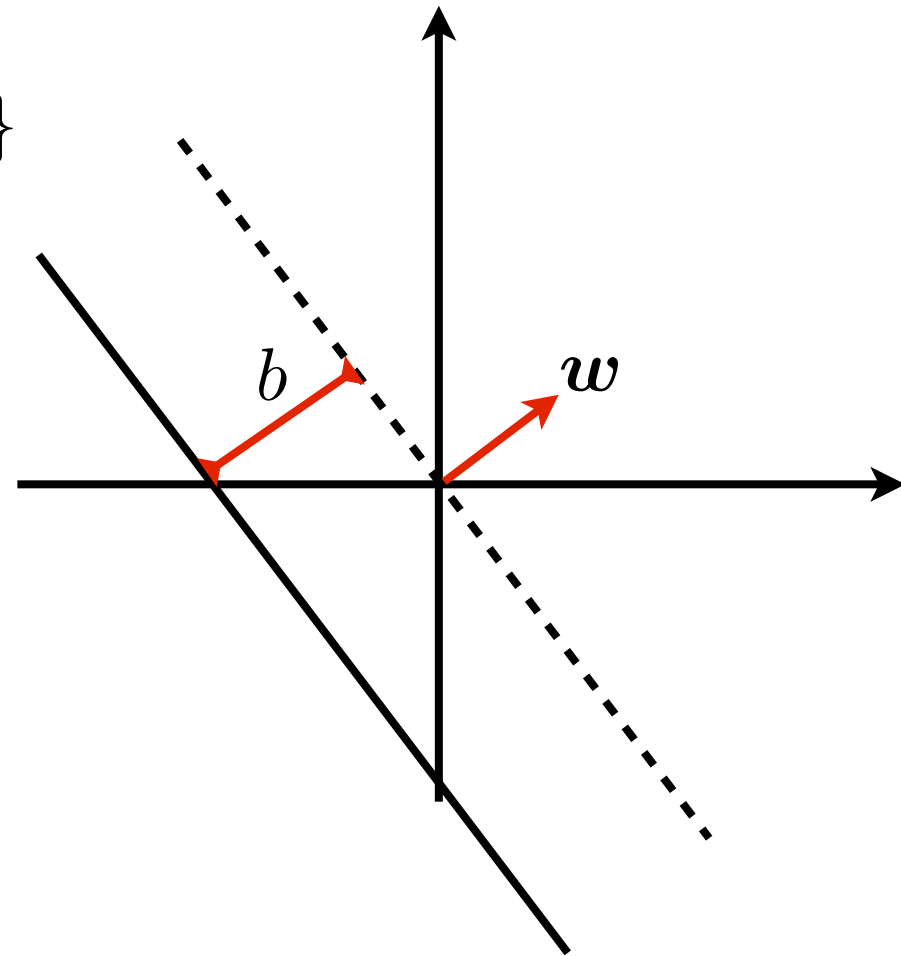
for classification  $y \in \{-1, +1\}$

we predict an instance by

$$\begin{aligned} & \text{sign}(w^\top x + b) \\ &= \begin{cases} +1, & w^\top x + b > 0 \\ -1, & w^\top x + b < 0 \\ \text{random}, & \text{otherwise} \end{cases} \end{aligned}$$

for an example  $(x, y)$  a correct prediction means

$$y(w^\top x + b) > 0$$

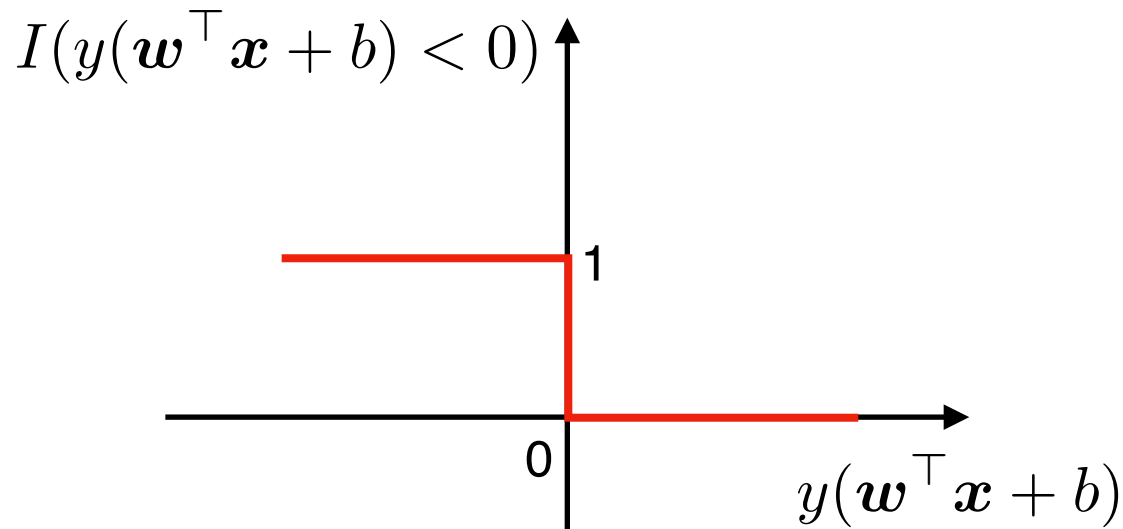


# Ideal classifier

$$\arg \min_{\mathbf{w}, b} \sum_i I(y(\mathbf{w}^\top \mathbf{x} + b) \leq 0)$$

non-differentiable

hard to solve by gradient descent



# Prototype

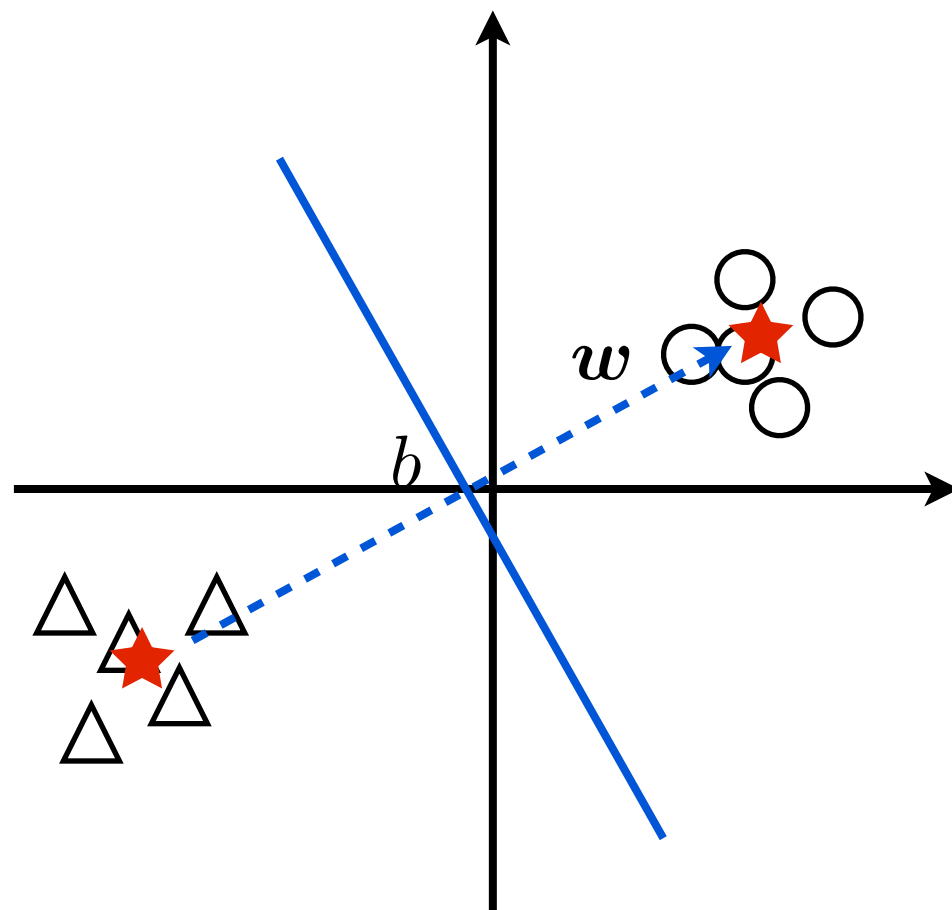
simple, but too restricted

$$\bar{\mathbf{x}}^+ = \frac{1}{\sum_{i:y_i=+1} 1} \sum_{i:y_i=+1} \mathbf{x}_i$$

$$\bar{\mathbf{x}}^- = \frac{1}{\sum_{i:y_i=-1} 1} \sum_{i:y_i=-1} \mathbf{x}_i$$

$$\mathbf{w} = \bar{\mathbf{x}}^+ - \bar{\mathbf{x}}^-$$

$$b = -\mathbf{w}^\top \cdot \frac{\bar{\mathbf{x}}^+ + \bar{\mathbf{x}}^-}{2}$$



# Perceptron

perception loss

$$\arg \min_{\mathbf{w}, b} \sum_i \max\{-y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0\}$$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

gradient ascent

$$\frac{\partial y \mathbf{w}^\top \mathbf{x}}{\partial \mathbf{w}} = y \mathbf{x}$$

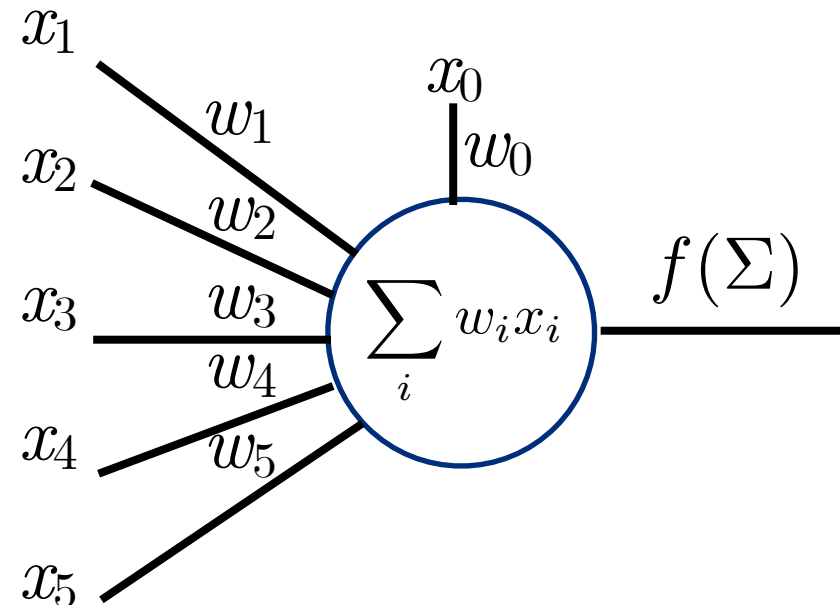
feed training examples one by one

1.  $\mathbf{w} = 0$

2. for each example  $(\mathbf{x}, y)$

if  $\text{sign}(y \mathbf{w}^\top \mathbf{x}) < 0$

$$\mathbf{w} = \mathbf{w} + y \mathbf{x}$$

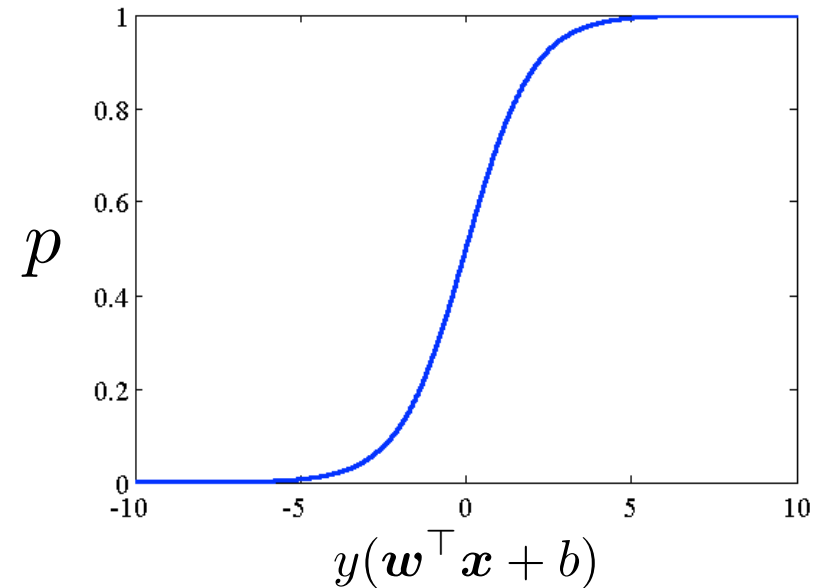


# Logistic regression

assume logit model: for a positive example

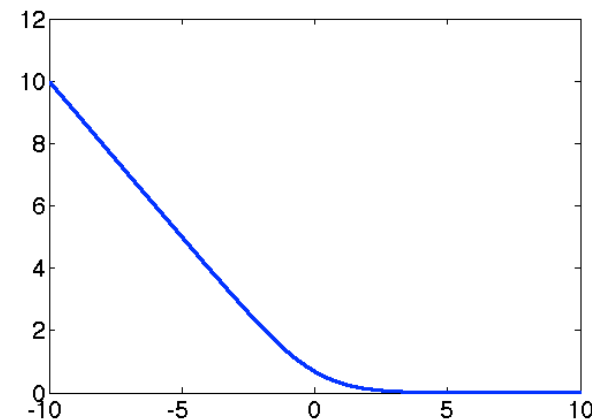
$$\mathbf{w}^\top \mathbf{x} = \log \frac{p(+1 \mid \mathbf{x})}{1 - p(+1 \mid \mathbf{x})}$$

so that  $p(y \mid \mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-y(\mathbf{w}^\top \mathbf{x})}}$



minimize negative log-likelihood:

$$\begin{aligned} \arg \min_{\mathbf{w}, b} -\log \prod_{i=1}^m p(y_i \mid \mathbf{x}_i, \mathbf{w}) &= -\sum_i \log p(y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \sum_i \log \left( 1 + e^{-y_i(\mathbf{w}^\top \mathbf{x}_i)} \right) \end{aligned}$$



convex

# Linear classifier revisit



model space:  $\mathbb{R}^{n+1}$

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

for classification  $y \in \{-1, +1\}$

Original objective:

$$\arg \min_{\mathbf{w}, b} \sum_i I(y(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0)$$

0-1 loss  
hard to optimize

Surrogate objective:

$$\arg \min_{\mathbf{w}, b} \sum_i \log \left( 1 + e^{-y_i(\mathbf{w}^\top \mathbf{x}_i + b)} \right)$$

logistic regression

$$\arg \min_{\mathbf{w}, b} \sum_i \max\{-y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0\}$$

perceptron



# Linear classifier revisit

## 0-1 loss

$$I(y(\mathbf{w}^\top \mathbf{x} + b) \leq 0)$$

## logistic regression

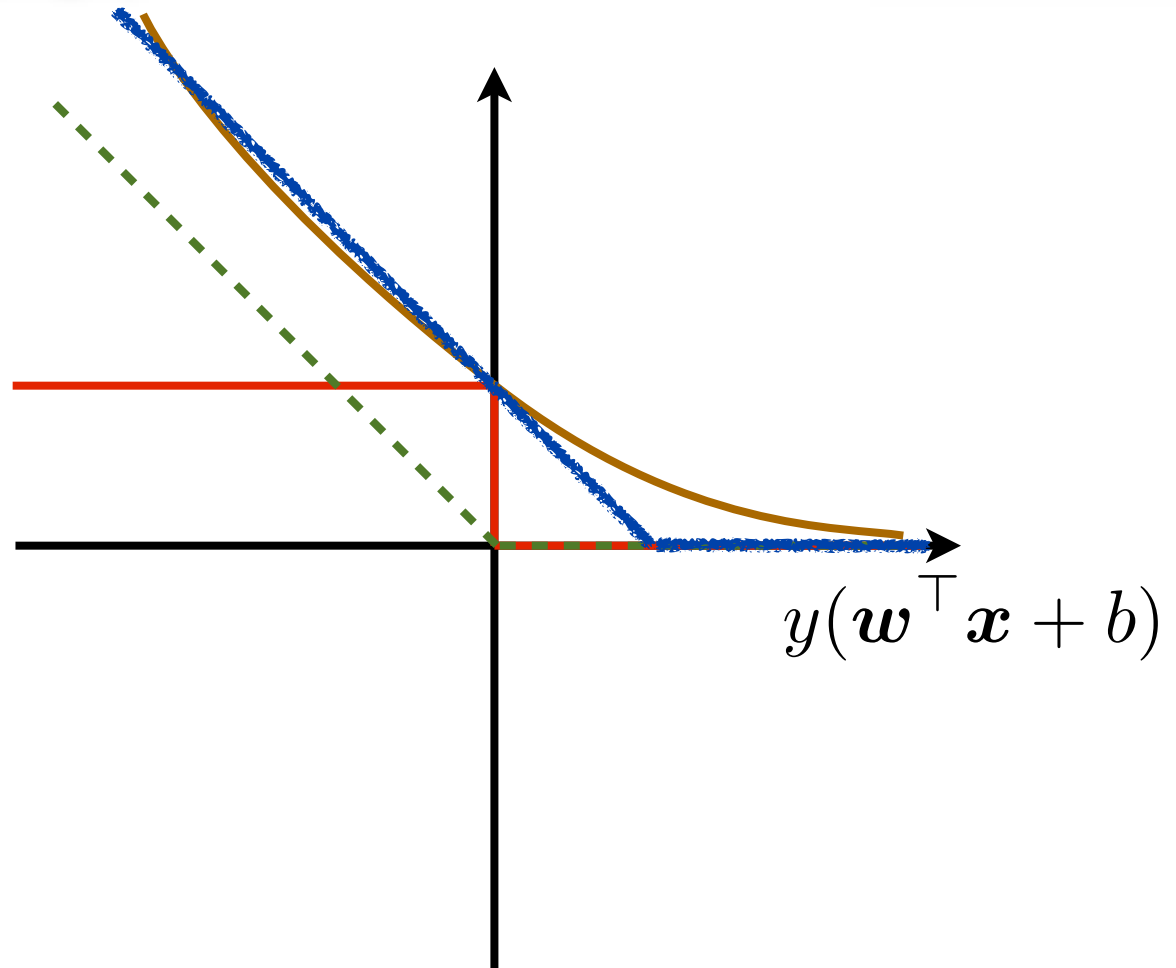
$$\log_2(1 + e^{-y(\mathbf{w}^\top \mathbf{x} + b)})$$

## perceptron

$$\max\{-y(\mathbf{w}^\top \mathbf{x} + b), 0\}$$

## hinge loss

$$\max\{1 - y(\mathbf{w}^\top \mathbf{x} + b), 0\}$$



# Support vector machines (SVM)

hinge loss + L2-norm

$$\arg \min_{\mathbf{w}, b} \sum_i \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0) + \lambda \|\mathbf{w}\|_2$$

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2 + C \sum_i \xi_i$$

$$\begin{aligned} s.t. \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

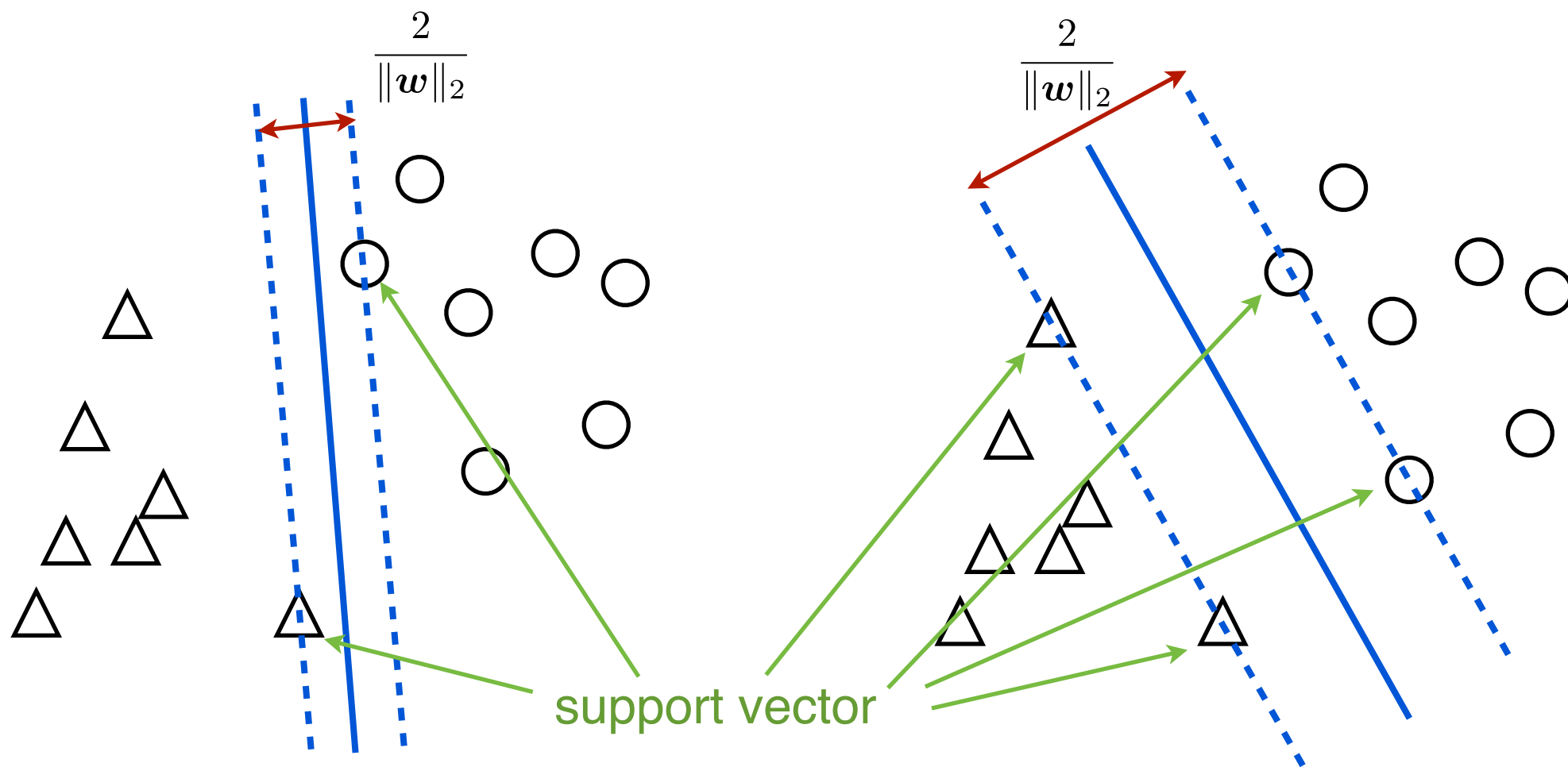
$$\begin{aligned} \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0) &= \xi_i \\ \xi_i &\geq 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \\ \xi_i &\geq 0 \end{aligned}$$

quadratic

# Support vector machines (SVM)

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2$$

$$s.t. \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$



# Scoring functions

$$\frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 \quad \text{least square regression}$$

$$\frac{1}{m} \sum_{i=1}^m |\mathbf{w}^\top \mathbf{x}_i + b - y_i| \quad \text{LAD regression}$$

$$\frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 + \lambda \|\mathbf{w}\|_2 \quad \text{ridge regression}$$

$$\frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 + \lambda \|\mathbf{w}\|_1 \quad \text{LASSO}$$

# Scoring functions

$$\sum_i I(y(\mathbf{w}^\top \mathbf{x} + b) > 0)$$

0-1 loss

$$\sum_i \max\{-y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0\}$$

perceptron

$$\sum_i \log \left( 1 + e^{-y_i(\mathbf{w}^\top \mathbf{x}_i + b)} \right)$$

logistic regression

$$\sum_i \log \left( 1 + e^{-y_i(\mathbf{w}^\top \mathbf{x}_i + b)} \right) + \lambda \|\mathbf{w}\|_2$$

regularized LR

$$\sum_i \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0) + \lambda \|\mathbf{w}\|_2$$

SVM

minimize loss + regularization