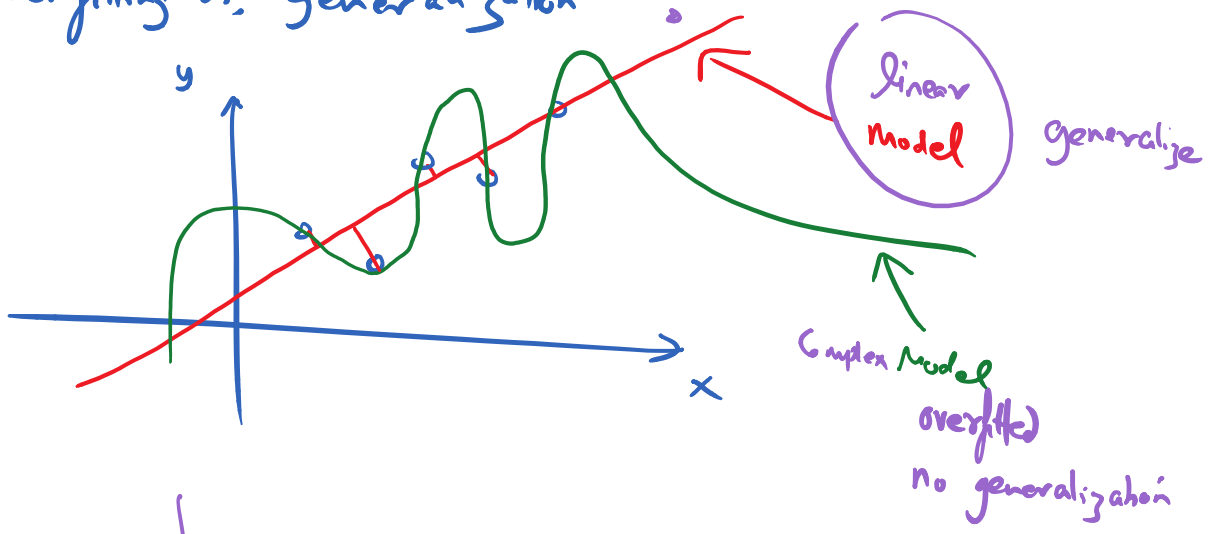
Bias - Variance Decomposition

Overfitting vs. generalization



Loss function : classifier M

$\hat{y} \leftarrow M(x)$

↑ Predicted class

y vs. $\hat{y}$

y ↑ true class

$\hat{y}$ ↑ prediction

Classification : 0-1 loss function

$$L(y_i, \hat{y}_i) = \mathbb{I}(y_i \neq \hat{y}_i)$$

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

↑ Indicator Function

expected loss:

expected loss:

$$\frac{1}{n}\sum L(y_i, \hat{y}_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq \hat{y}_i)$$

$$= \frac{1}{n}\{\#\text{of errors}\}$$

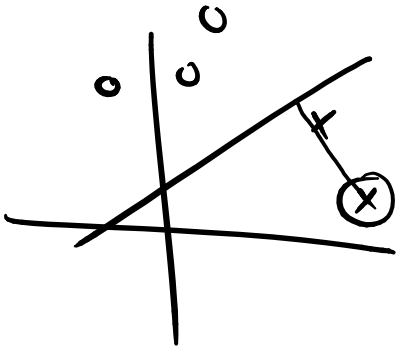
$$= \text{error-rate}$$

Indicator Function

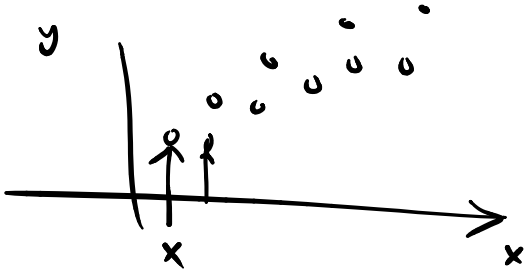
Squared loss:

$$L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

y_i: true value; $\hat{y}_i$: predicted value



Regression problem



given the points $\vec{x}_i \in \mathbb{R}^d$

predict a "real" variable $y \in \mathbb{R}$

Expected loss?

$$E[L(y_i, \hat{y}_i)]$$

$$= E[(y_i - \hat{y}_i)^2]$$

$$= E\left[\left(y_i - E[y_i] + E[y_i] - \hat{y}_i\right)^2\right]$$

One test case x_i

$$= E\left[\left((y_i - E[y_i]) + (E[y_i] - \hat{y}_i)\right)^2\right]$$

$$= E\left[(y_i - E[y_i])^2 + (\hat{y}_i - E[y_i])^2 + 2(y_i - E[y_i])(E[y_i] - \hat{y}_i)\right]$$

$$= E\left[(y_i - E[y_i])^2\right] + E\left[(\hat{y}_i - E[y_i])^2\right] + 0$$

$2E(y_i - E[y_i])(\quad)$

$2(E[y_i] - E[y_i]) = 0$

true — expectation — Variance of y_i — avg noise

predict — expected true value — sq. error term

$E_{y_i}[(\hat{y}_i - E[y_i])^2]$ — Over all possible test cases

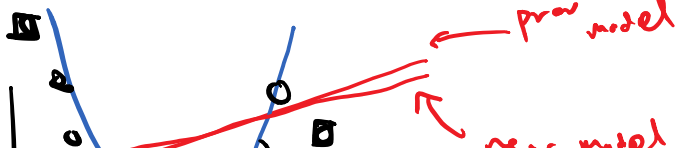
$$E\left[(\hat{y}_i - E[\hat{y}_i] + E[\hat{y}_i] - E[y_i])^2\right]$$

$$\text{avg loss} = \left[E\left[(\hat{y}_i - E[\hat{y}_i])^2\right] + E\left[(E[\hat{y}_i] - E[y_i])^2\right] + E\left[(y_i - E[y_i])^2\right]\right]$$

Variance of the model — avg bias of model — avg noise of data

simple model

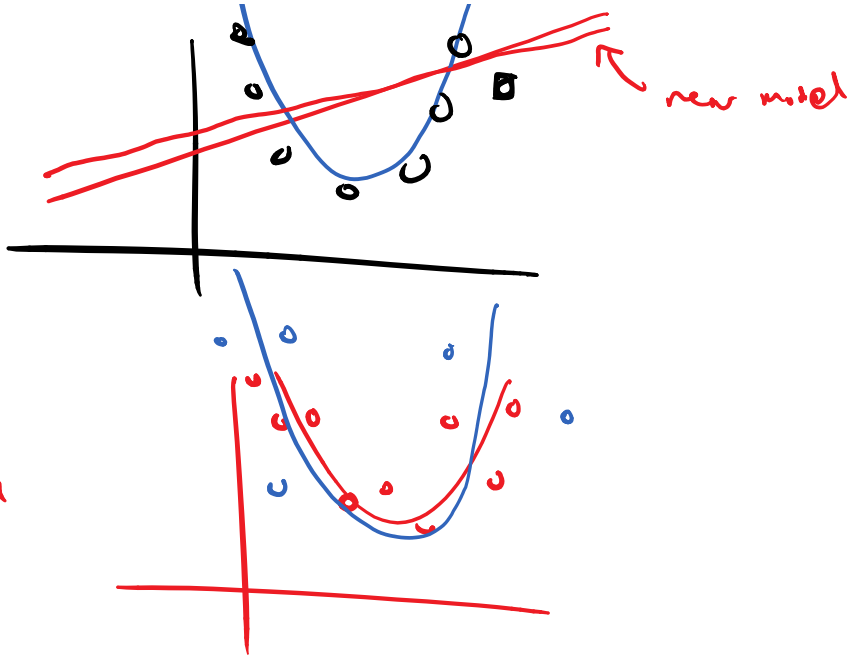
linear classifier



linear classifier

→ high bias

→ low variance



Complex model

Quadratic decision/model

→ low bias

→ high variance

bias - variance tradeoff

more complex the model → low bias → can increase variance

keep the simplest model with low variance

How to set low variance & low bias?

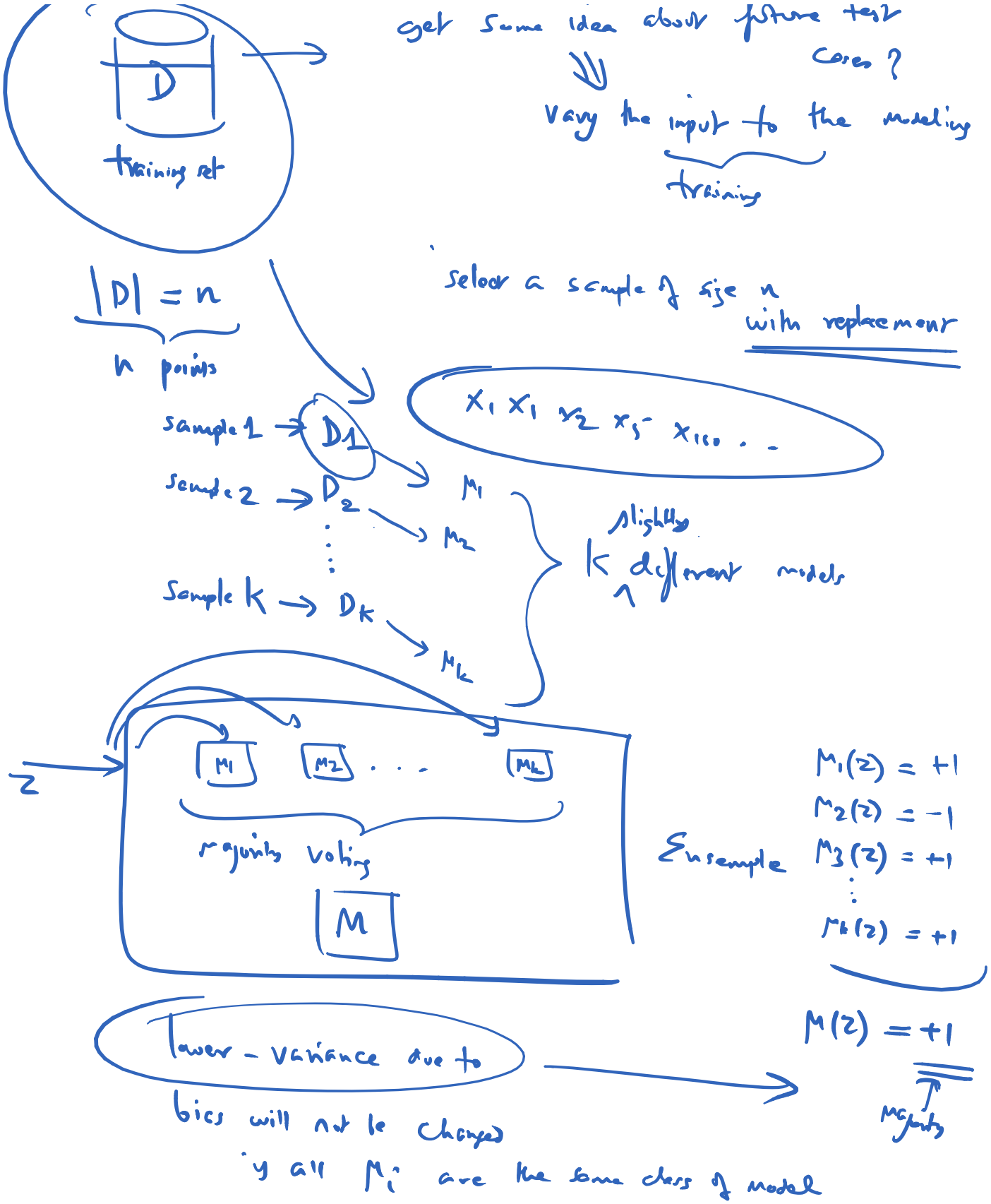
Ensemble classifiers

① Bagging → Bootstrapped Aggregation

→ lower variance due to averaging.

Bootstrap sampling

get some idea about future test cases?


D
training set
get some idea about future test cases ?
vary the input to the modeling training
|D| = n
n points
select a sample of size n with replacement
sample 1 → D1
sample 2 → D2
sample k → Dk
x1 x1 x2 x5 x100 . .
M1
M2
Mk
slightly k different models
z
M1
M2 . . .
Mk
majority voting
M
Ensemble
M1(z) = +1
M2(z) = -1
M3(z) = +1
Mk(z) = +1
M(z) = +1
Majority
lower - variance due to
bias will not be changed
by all Mi are the same class of model
lower bias ?

Boosting → weighted sampling + quality of classifier

↓

boost or bump up the weight for misclassified points

α_i ← weighted vote.

$w_0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$ ← Uniform distribution

$D_{training} \rightarrow D_1 \rightarrow M_1$

① pick a sample A of size n from D_{train} using $\vec{w}$ as the prob vector (with replacement)

② Build classifier M_1

③ ε_1 = error rate of M_1 ← $0 < \varepsilon_1 < 0.5$ ← error rate

$S_1 = \{$ set of all misclassified points, $\hat{y}_i \neq y_i \}$

④ $\alpha_1 = \ln\left(\frac{1}{\varepsilon_1} - 1\right)$ ← weight of classifier

⑤ $w_i' = \begin{cases} w_i & \text{if } y_i = \hat{y}_i \\ w_i \cdot e^{\alpha_i} & \text{if } y_i \neq \hat{y}_i \end{cases}$ ← new weight vector

$\ln\left(\frac{1}{0.5} - 1\right)$

$\ln(2 \cdot 1) = \ln(1) = 0$

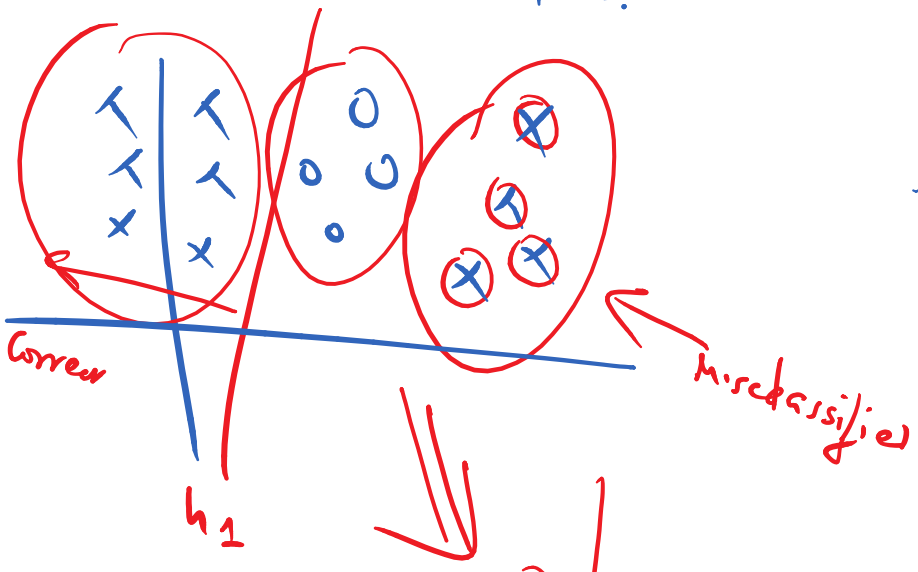
$\ln\left(\frac{1}{0.1} - 1\right) = \ln(9)$

⑥ normalize to be prob. vector

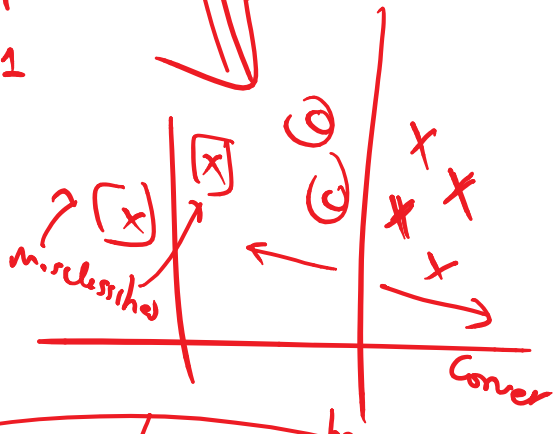
$w_i' = \frac{b_i'}{\sum_i b_i'}$

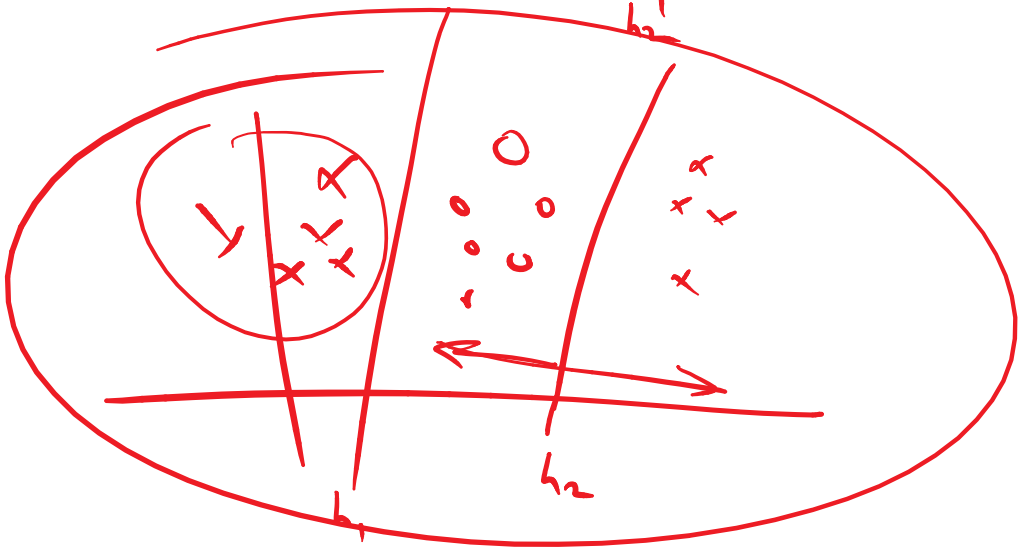
$$w_i = \frac{w_i}{\sum w_i}$$

repeat K times.



Linear SVM





$$\boxed{h_1}_{\alpha_1} + \boxed{h_2}_{\alpha_2}$$

Ensemble: $z \rightarrow$

$h_1(z)$ $h_2(z)$

$+1 = 10$ $-1 = 5$

5

$h_1(z), h_2(z) \ldots h_k(z)$

$\alpha_1 \quad \alpha_2 \ldots \alpha_k$

weighted vote for the majority class

Regression

$x_1 x_2 \ldots x_d$ y

$\vec{x}_i$ $x_1 x_2 \ldots x_d$ y_i

ground truth

d - Independent variable

point $\vec{x}_i \in \mathbb{R}^d$

$y_i \in \mathbb{R}$

dependent variable response var.

$$\hat{y}_i = h(\vec{x})$$

linear regression.

$$\hat{y}_i = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d = w_0 + \vec{w}^T \vec{x}$$

bias bias dot product

$\vec{x}_i = (1, x_1 x_2 \ldots x_d)$

$w = (w_0 w_1 w_2 \ldots w_d)$

d+1 dimensions

$\hat{y}_i = w^T x$

weight vector

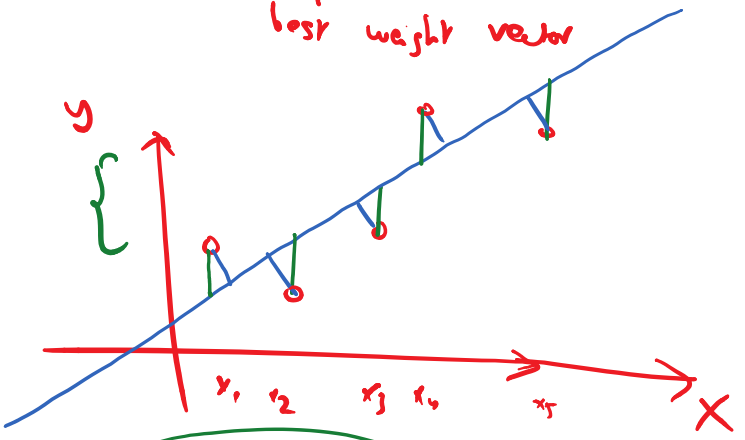
$$J(w) = \min \sum^{n} (y_i - \hat{y}_i)^2$$

squared error

$$J(w) = \min_{w} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Squared error

best weight vector



PCA: best line (orthogonal sq. error) — Min error along all attributes

Regression best line = "vertical" sq. error : along y only.

$\hat{y}_i = w^T \vec{x}_i$ — scalar, dot product

$\vec{w} \in \mathbb{R}^{d+1}$

$\vec{x}_i \in \mathbb{R}^{d+1}$

n-equations; one per point

Combine into one equation

$$n \begin{cases} \hat{\vec{y}} = X\vec{w} \end{cases}$$

$X = D$

$n \times (d+1)$ matrix of points

$\hat{\vec{y}}$ = $n \times 1$ vector of predicted values

$\vec{y}$ = $n \times 1$ vector of true values

$\vec{w} = \mathbb{R}^{d+1}$

$$J(w) = \min_{w} \underbrace{\| \vec{y} - \hat{\vec{y}} \|^2}$$

$$= \min_{w} \| \vec{y} - X\vec{w} \|^2$$

$$= (\vec{y} - X\vec{w})^T (\vec{y} - X\vec{w})$$

$$J(\omega) = \vec{y}^T\vec{y} + \vec{\omega}^T X^T X \vec{\omega} - 2(X^T\omega)y$$

$$\frac{\partial J(\omega)}{\partial \omega} = 2X^T X\omega - 2X^T y = 0$$

$$\Rightarrow \quad X^T X\omega = X^T y$$

$$\vec{\omega} = (X^T X)^{-1} X^T y$$

Closed form solution

scatter matrix

$$y = M(x) = \vec{\omega}^T \vec{x}$$

regression model