A physical scientist's introduction to Machine Learning

Physics from data

Galileo

676 EPITOMES ASTRONOMIÆ

LIBRI V.

PARS ALTERA.

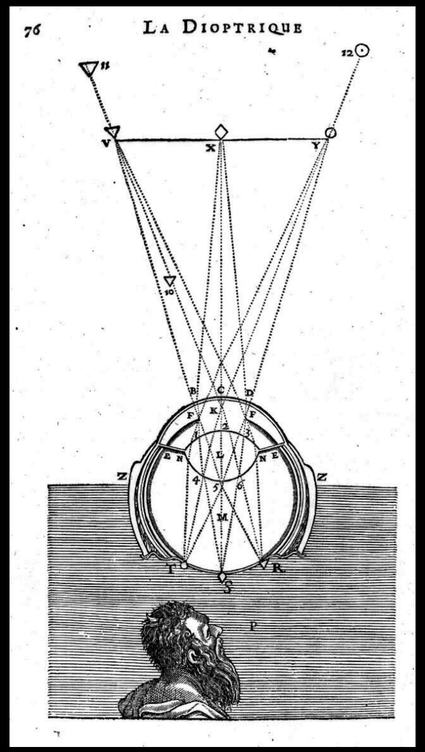
DE TERMINIS ASTRONOMICIS ET CALCVLO, EX ORBITA ECcentrica orientibus.

Quomodo appellatur Orbita cuiusque Planetæ?

APpellatur veteri voce Eccentricus subaudi, Circulus. Etsi enim orbitæ sunt Ellipticæ, vt hic PERI. quæ habent duo quasi centra A. L. quæ physicè Focos dicimus; & in eorum altero A. Sol ipse vt centrum mundi, ineſt: tamen etiam

punctum inter focos medium, vt B. à scriptoribus Conicis centrum figuræ, peculiari iure dicitur; & præterca

Kepler



76 LA DIOPTRIQUE

Descartes

Finding ***constants*** of nature that ***generalize*** in space and time

Laws are linear

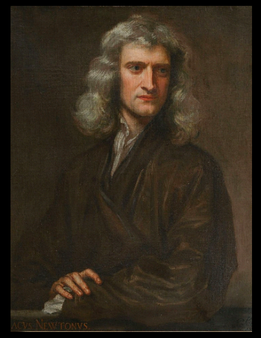
Pascal's law (1653)

$$\Delta p = \rho g \Delta h$$

Hooke's law (1678)

$$F = -kx$$

Newton's law of viscosity (1701)



$$\tau = \mu \frac{du}{dy}$$

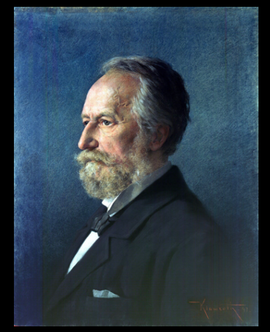
Ohm's law (1781)

$$I = V/R$$

Fourier's law (1822)

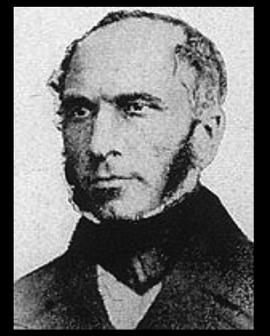
$$q = -k \frac{dT}{dx}$$

Fick's law (1855)



$$J = -D \frac{dC}{dx}$$

Darcy's law (1856)



$$Q = \frac{kA}{\mu L} \Delta p$$

Ideal gas law (1834)

Amonton's law (1808)

Boyle's law (1662)

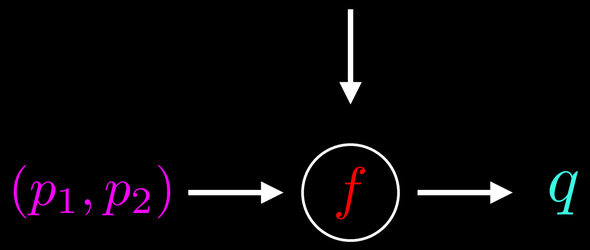
$$\frac{PV}{TN} = k_B$$

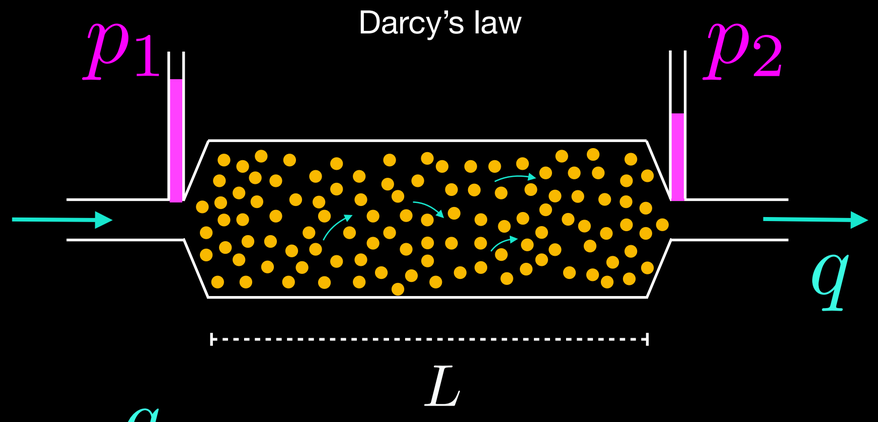
Charles's law (1787)

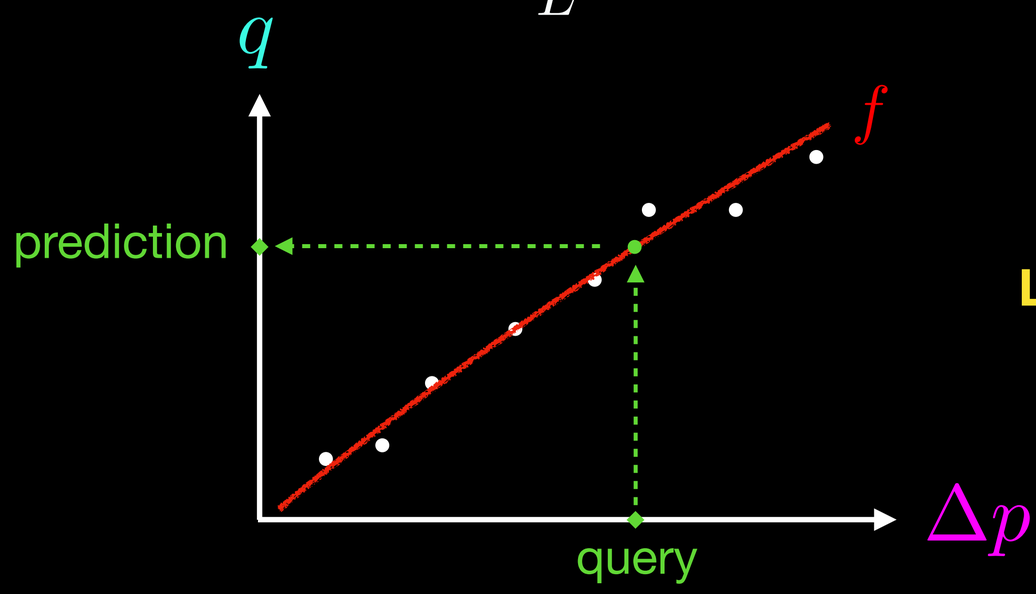
Avogadro's law (1811)

From experiment to Law

p_1	p_2	q
1.3	1.0	22
1.6	1.5	23
3.4	2.4	46
4.8	3.5	67
6.7	4.5	83
…		
2.3	**1.4**	**?**







Linear regression

Machine or human learning

Training data: $\mathcal{D}_{\text{train}}$

p_1	p_2	q
1.3	1.0	22
1.6	1.5	23
3.4	2.4	46
4.8	3.5	67
6.7	4.5	83
...		

Example →

Training

Feature extractor

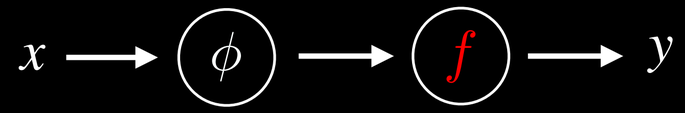
$(p_1, p_2) \rightarrow \phi \rightarrow \Delta p \rightarrow f \rightarrow q$

Predictor *Labels*

- Which predictors are possible?
- How good is the predictor?
- How can we find the best predictor?

Linear predictors

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
$\vdots$
(x_n, y_n)



feature vector

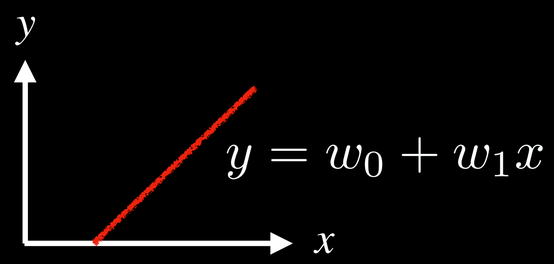
$$\phi(x) = [\phi_1(x),\ \phi_2(x),\ \ldots,\ \phi_d(x)]$$

linear predictor

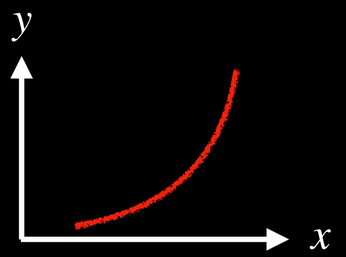
$$f_{\mathbf{w}} = \mathbf{w} \cdot \phi(x)$$
$$= w_1\phi_1(x) + w_2\phi_2(x) + \cdots + w_d\phi_d(x)$$

What is linear?

Adding a bias: $\phi(x) = [1,\ x]$

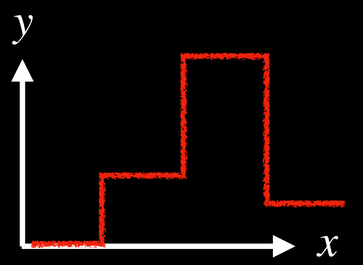


A polynomial predictor: $\phi(x) = \left[1, x, x^2, x^3\right]$



Sines and cosines: $\phi(x) = [1,\ x,\ \sin(3x)]$

Indicator functions: $\phi(x) = [\mathbf{1}[0 < x \leq 1],\ \mathbf{1}[1 < x \leq 2],\ \mathbf{1}[2 < x \leq 3]]$



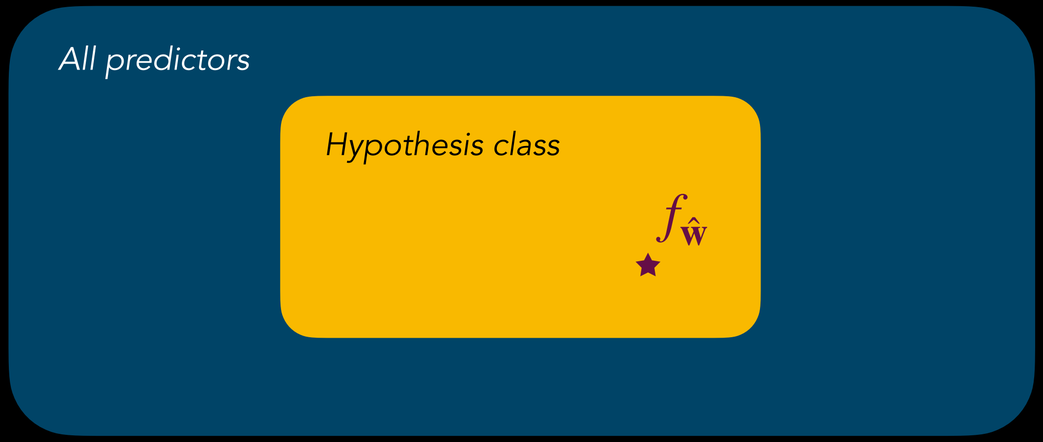
Derivatives: $\phi(x, t) = \left[x^2,\ \frac{\Delta x}{\Delta t}\right]$

Linear in $\phi(x)$ and $\mathbf{w}$ but not in x

Brunton, S. L., Joshua L. P., and J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems."

Hypothesis Class

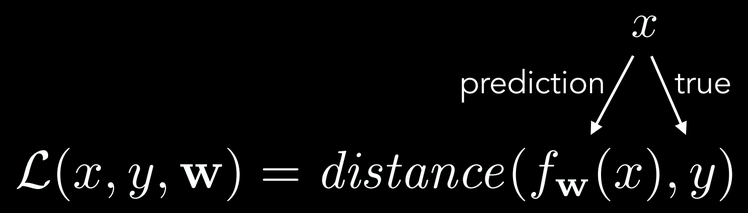
$$\mathcal{F} = \{ f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) \mid \mathbf{w} \in \mathbb{R}^d \}$$

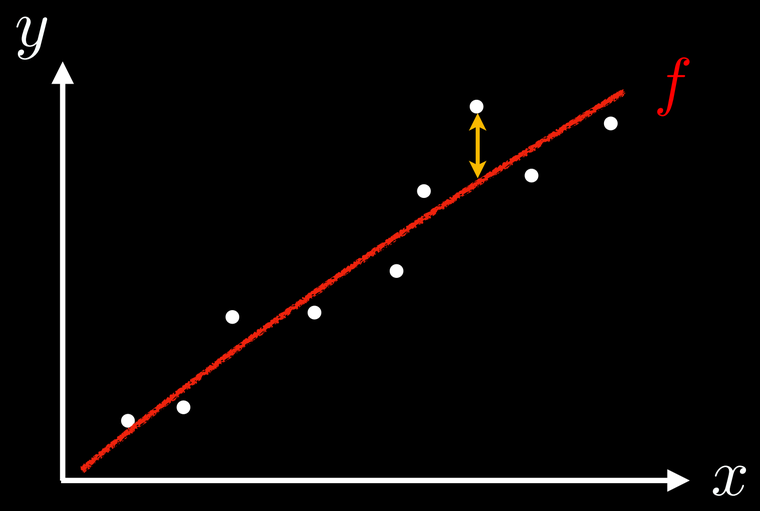


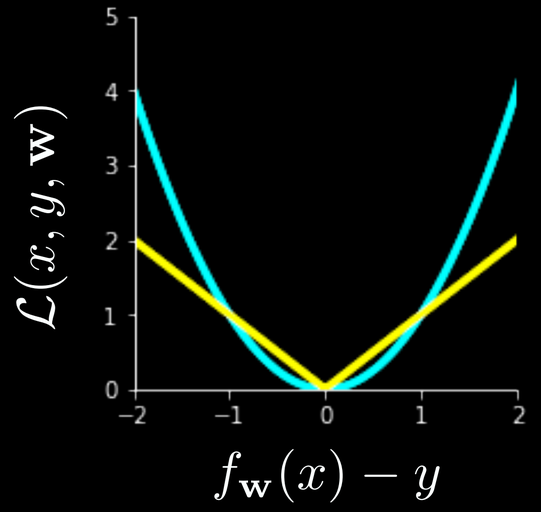
- Choose $\mathcal{F}$ based on domain knowledge
- Find the best predictor $f_{\hat{\mathbf{w}}}(x)$ based on data

The loss function

A **loss function** quantifies how *bad* the predictor is







$$\mathcal{L}_{\text{sq}}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$$

$$\mathcal{L}_{\text{abs}}(x, y, \mathbf{w}) = |f_{\mathbf{w}}(x) - y|$$

Minimize the loss

The **training loss** is the average loss over the data set

$$\mathcal{L}_{\text{train}}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \mathcal{L}(x, y, \mathbf{w})$$

objective

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \mathcal{L}_{\text{train}}(\mathbf{w})$$

optimal predictor

$$f_{\hat{\mathbf{w}}}(x) = \hat{\mathbf{w}} \cdot \phi(x)$$

Gradient Descent

Training loss gradient

$$\nabla_{\mathbf{w}} \mathcal{L}_{\text{train}}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \overbrace{2(\mathbf{w} \cdot \phi(x) - y)\phi(x)}^{\nabla_{\mathbf{w}} \mathcal{L}_{\text{sq}}(x, y, \mathbf{w})}$$

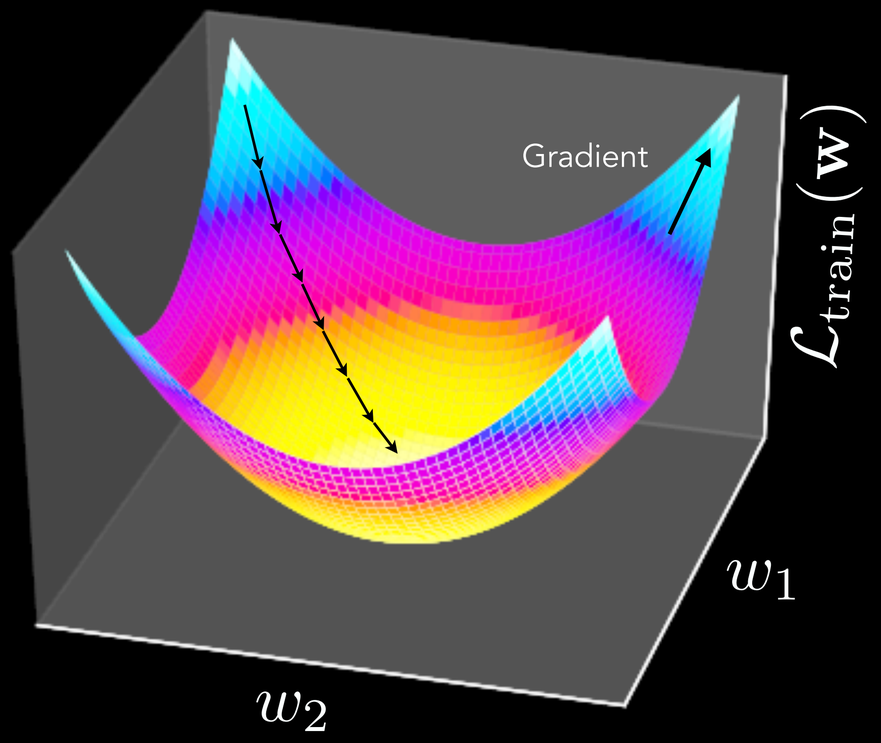
Prediction — True value

Gradient descent algorithm

epochs

initialize $\mathbf{w} = [0, \ldots, 0]$;
for $t = 1, \ldots, T$ **do**
  $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{\text{train}}$
end

step size



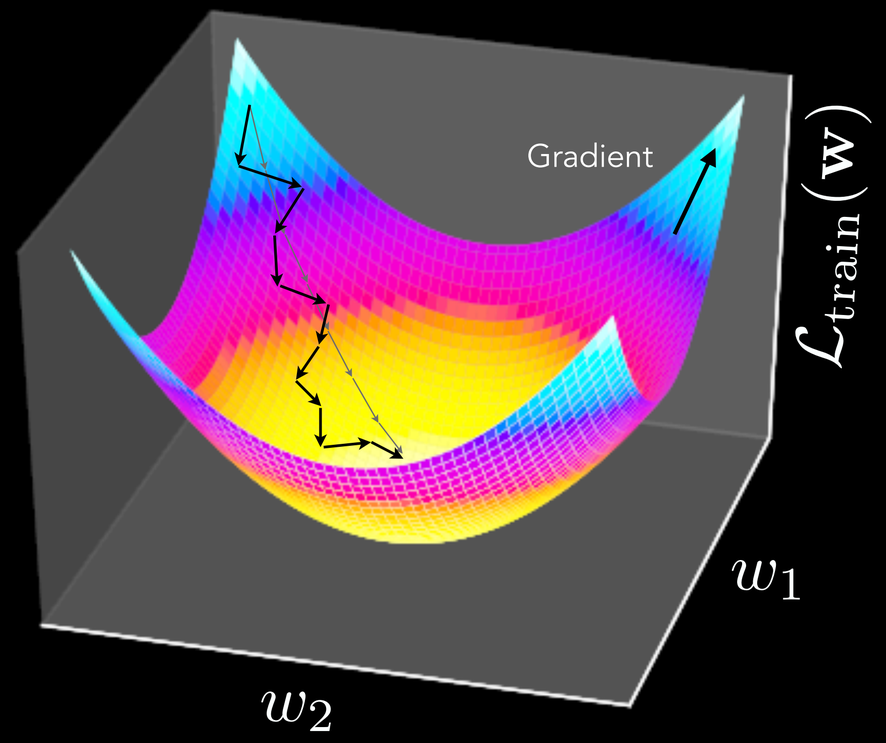
Stochastic Gradient Descent

Gradient descent is too expensive

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{\text{train}}$$

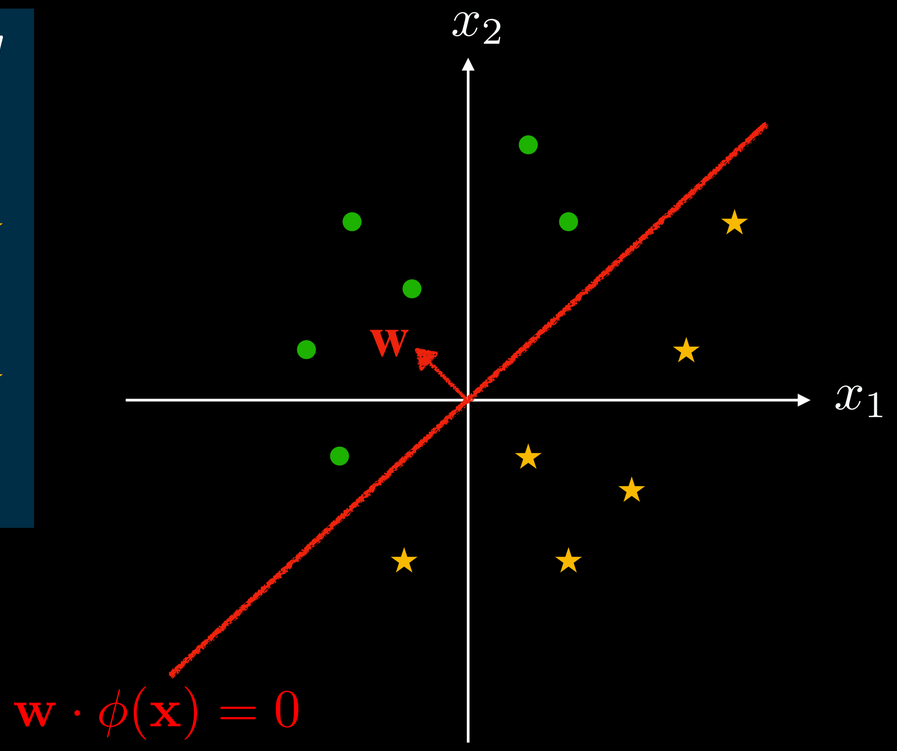
Stochastic gradient descent

initialize $\mathbf{w} = [0, \ldots, 0]$;
for $t = 1, \ldots, T$ **do**
  for $(x, y) \in \mathcal{D}_{train}$ **do**
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}(x, y, \mathbf{w})$
  end
end



Classification

x_1	x_2	y
-2	-1	●
3	1	★
2	3	●
1	-1	★
	⋮	



$$(x_1, x_2) \longrightarrow \phi \longrightarrow f \longrightarrow y$$

$$y \in [-1,\ 1]$$

$$f_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \phi(\mathbf{x}))$$

$$\mathcal{L}_{\text{0-1}} = \mathbf{1}\left[f_{\mathbf{w}}(\mathbf{x}) \neq y\right]$$

how confident?

score

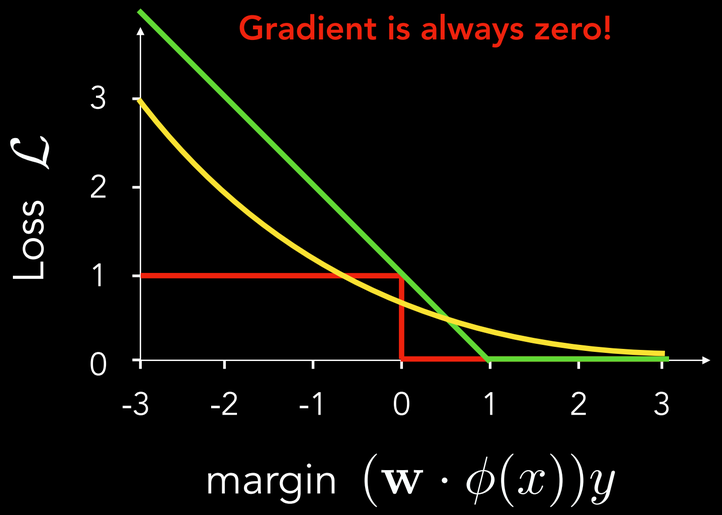
$$\mathbf{w} \cdot \phi(\mathbf{x})$$

how correct?

margin

$$(\mathbf{w} \cdot \phi(\mathbf{x}))\, y$$

Classification losses



$$\mathcal{L}_{\text{0-1}} = \mathbf{1}\left[(\mathbf{w} \cdot \phi(x))\, y \leq 0\right]$$

$$\mathcal{L}_{\text{hinge}} = \max\left\{1 - (\mathbf{w} \cdot \phi(x))\, y,\ 0\right\}$$ Support Vector Machines

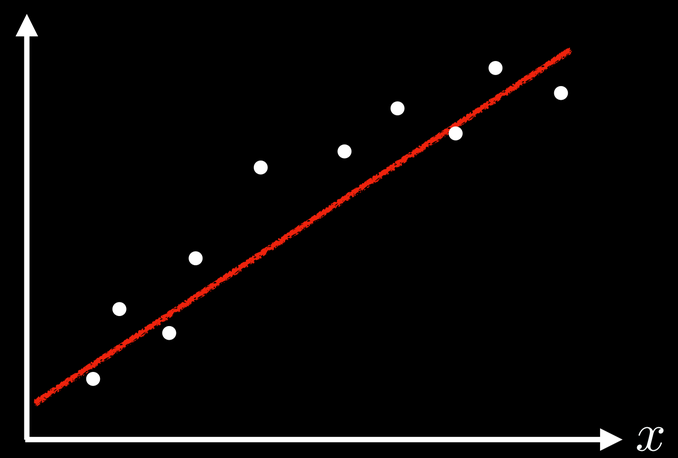
$$\mathcal{L}_{\text{logistic}} = \log\left(1 + e^{-(\mathbf{w} \cdot \phi(x))y}\right)$$ Logistic Regression

Variance vs. Bias

How do you choose $\phi(\cdot)$?

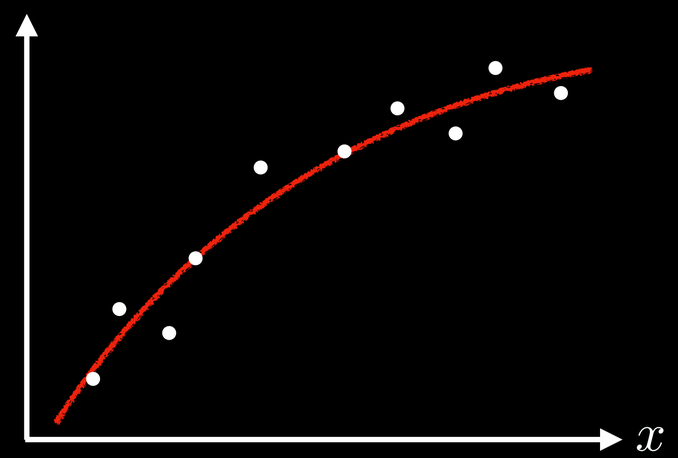
under-fitting

high bias



$\phi(x) = [1, x]$

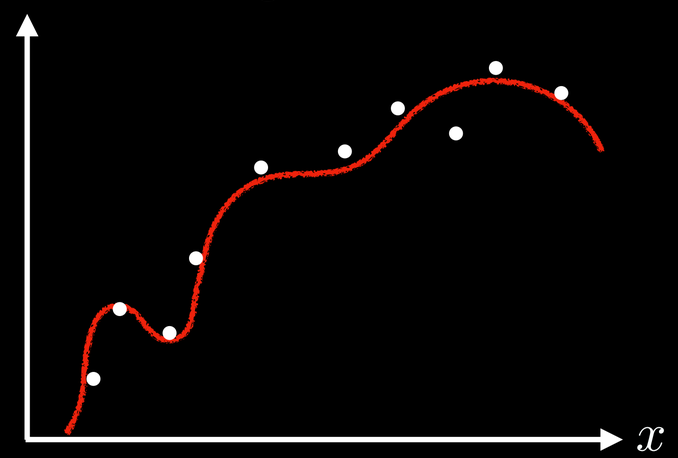
just right



$\phi(x) = [1, x, x^2]$

over-fitting

high variance



$\phi(x) = [1, x, x^2, \ldots, x^n]$

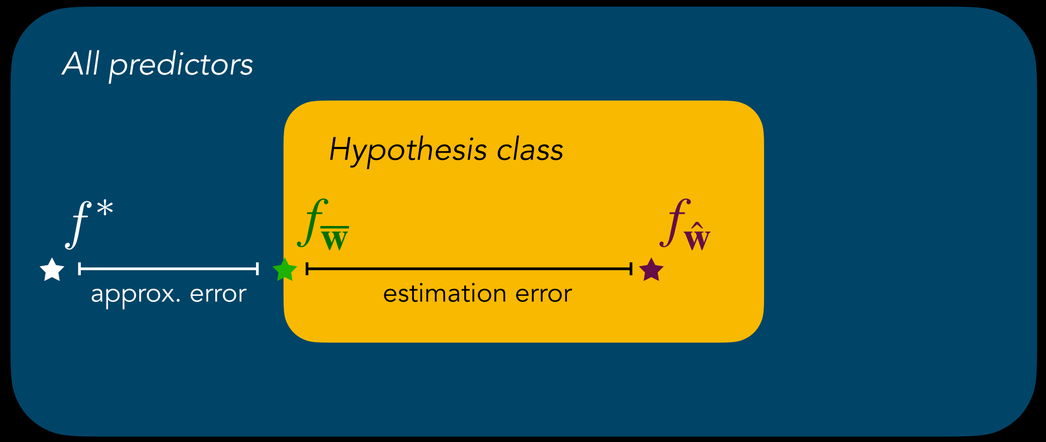
Two sources of error

Estimation error

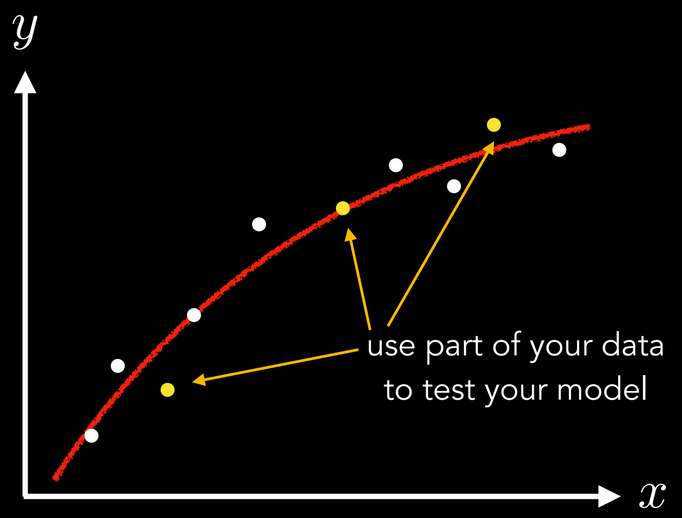
How good is the predictor, given the hypothesis class?

Approximation error

How good is the hypothesis class?

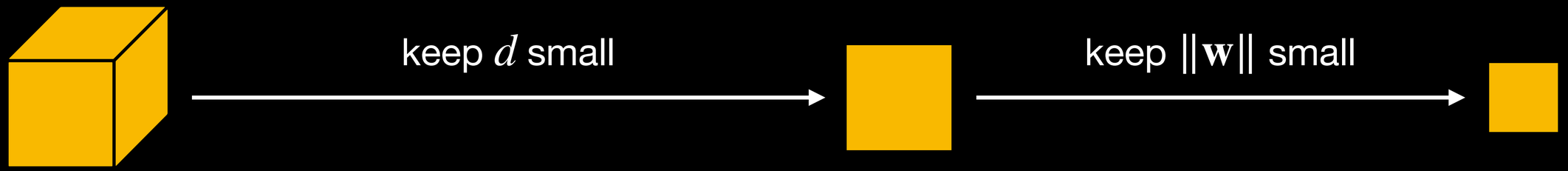


Use a test set



Designing the hypothesis class

$\mathbf{w} \in \mathbb{R}^d$



Remove features if they don't help

Use automatic feature selection

$$\mathcal{L} = \mathcal{L}_{\text{sq}} + \lambda \|\mathbf{w}\|_0$$

number of nonzero elements

Use L_2 regularization

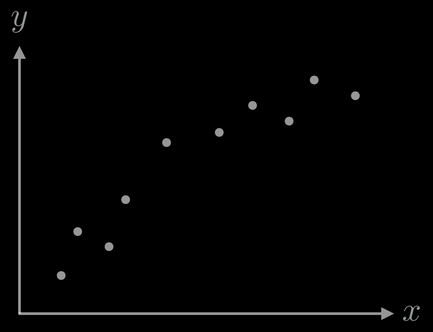
$$\mathcal{L} = \mathcal{L}_{\text{sq}} + \lambda \|\mathbf{w}\|^2$$

Hyperparameters

How do you choose the

- Regularization parameter
- Number of iterations
- Steps size
- …

The ML workflow



Can we automate feature extraction?

$$\phi(x) = [1, x, \ldots, ?]$$

not satisfied

η

T λ

$\mathcal{L}_{\text{val}}$

satisfied

$\mathcal{L}_{\text{test}}$