Regression

Sewoong Oh

CSE/STAT 416
University of Washington

Predictors

Data fitting

- goal: predicting "How much is my house worth?"

- data

$$
\begin{aligned}
(x_1, y_1) &= (\,2318\, sq.ft.\,,\ \$\,315k) \\
(x_2, y_2) &= (\,1985\, sq.ft.\,,\ \$\,295k) \\
(x_3, y_3) &= (\,2861\, sq.ft.\,,\ \$\,370k) \\
&\ \vdots \qquad\qquad \vdots \\
(x_n, y_n) &= (\,2055\, sq.ft.\,,\ \$\,320k)
\end{aligned}
$$

← data pair or example

- hope/belief: We think $y \in \mathbf{R}$ and $x \in \mathbf{R}^d$ are approximately related by

$$y \approx f_0(x)$$

- x is called the input data
  y is called the outcome, response, target, label, or dependent variable

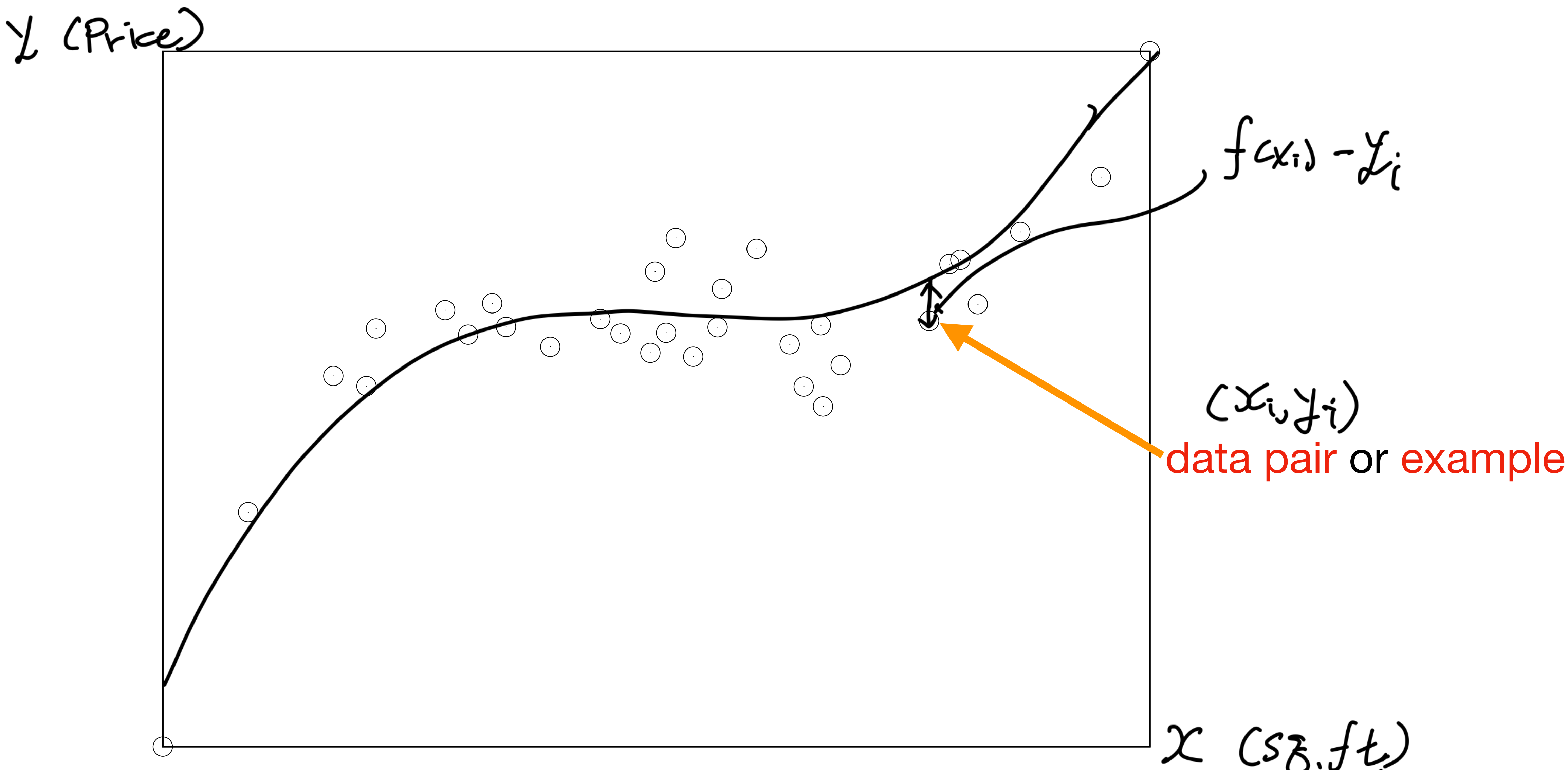
- y is what we want to predict

Predictor

- we seek a predictor or model $f : \mathbf{R}^d \to \mathbf{R}$

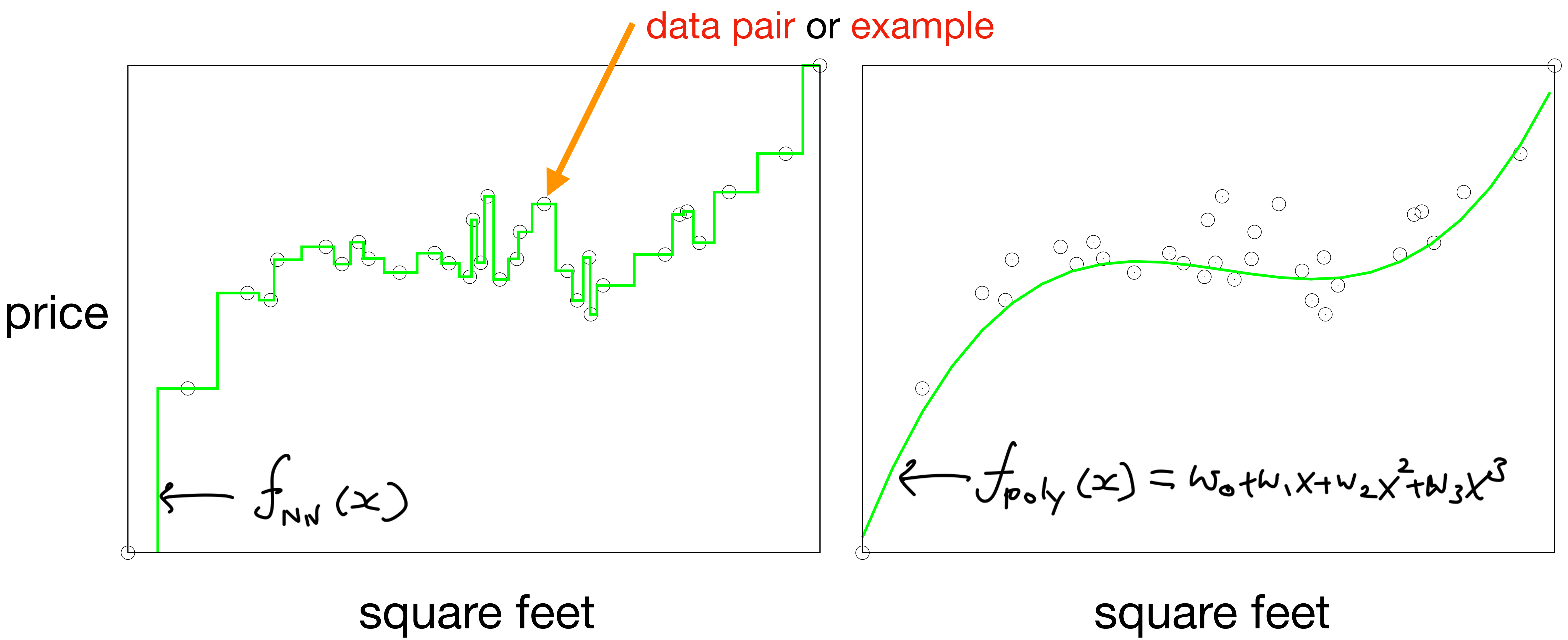
- for an input data x, our prediction of the label y is

$$\hat{y} = f(x)$$



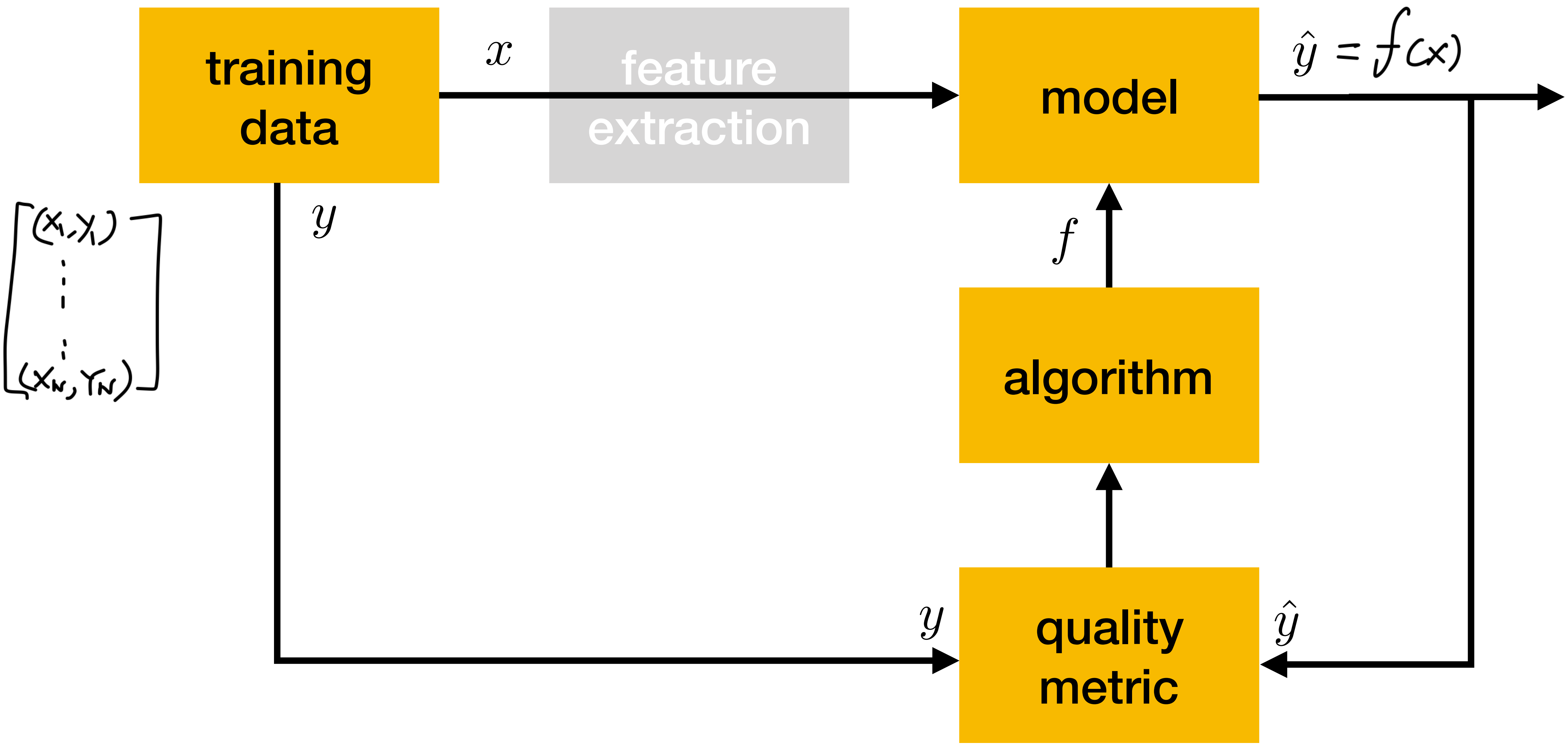
- small error on an example, $f(x_i) \approx y_i$, implies that we have a good prediction on the ith pair (x_i, y_i)

a machine learning algorithm is a principled recipe for producing a predictor, given data



- left plot shows nearest neighbor prediction
- right plot shows cubic polynomial fit
- we want a good prediction on pairs we have not seen

Machine learning pipeline



Model (linear regression)

Model

- our belief in the real world data
- linear regression model

$$y = w_0 + w_1 x + \varepsilon$$

↳ random noise with zero mean

$\mathbb{E}[\varepsilon] = 0$

- linear predictor

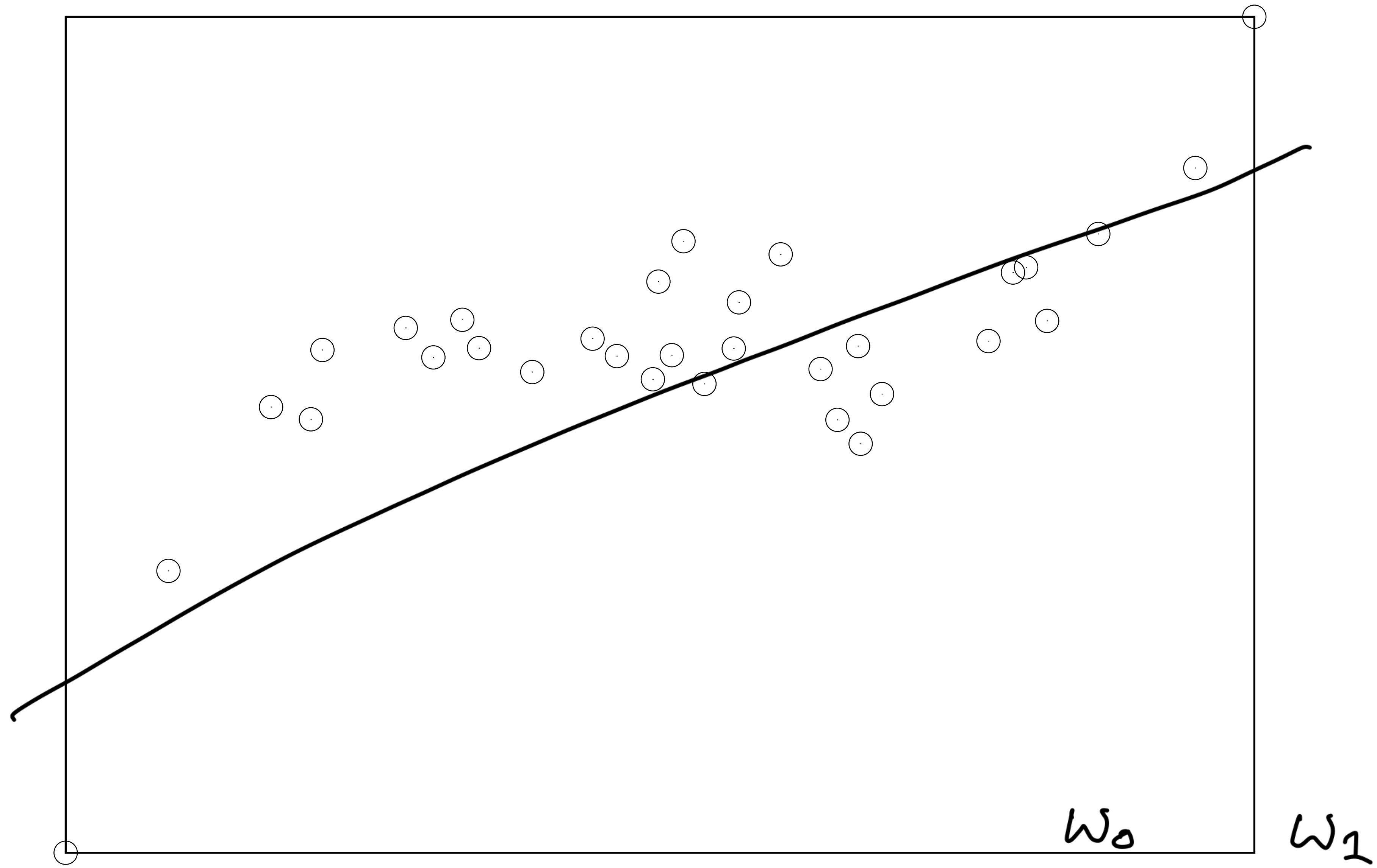
$$\hat{y} = f(x) = w_0 + w_1\, x$$

- strictly speaking, this is an affine model
- in general, linear regression model can be multi-dimensional

$$f(x) = w_0 + w_1\, x[1] + w_2\, x[2] + \cdots + w_d\, x[d]$$

- $w_0, w_1, \ldots, w_d$ are the model parameters

$$\hat{y} = f(x) = w_0 + w_1 x$$



- once you fit a model to the data, e.g. $f(x) = 10,000 + 141\,x$
  - a seller with a house $x = 2511$ sq.ft. can predict the price
  - a buyer with money $y = \$364k$ can predict the size
- interpretation of the parameters
  - w_0 is the shift: price of land with no house
  - w_1 is the slope: how much price goes up per sq.ft.

Interpreting a linear model

- In general,

$$\hat{y} \;=\; f(x) \;=\; w_0 + w_1\, x[1] + w_2\, x[2] + \cdots + w_d\, x[d]$$

- w_3 is how much the (predicted) price increase when $x[3]$ increases by 1
- $w_7 = 0$ means the price does not depend on $x[7]$
- the constant term w_0 predicts when all features are zero
- for notational consistency, sometimes we say $x[0] = 1$ is a constant feature

Quality metric

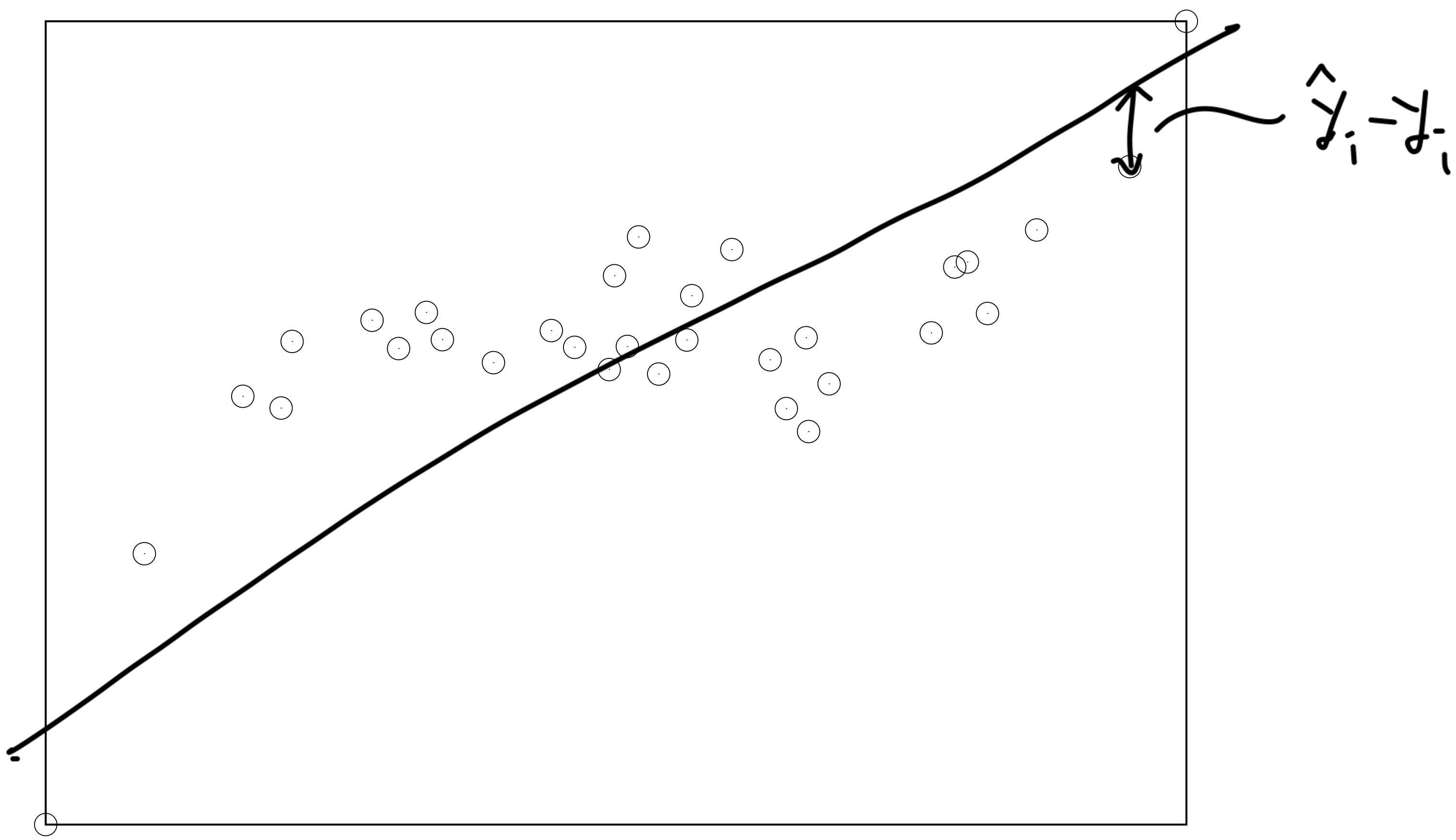
Quality metric

- cost or loss determining which model is a better fit

- mean square error (MSE) or residual sum of squares (RSS)

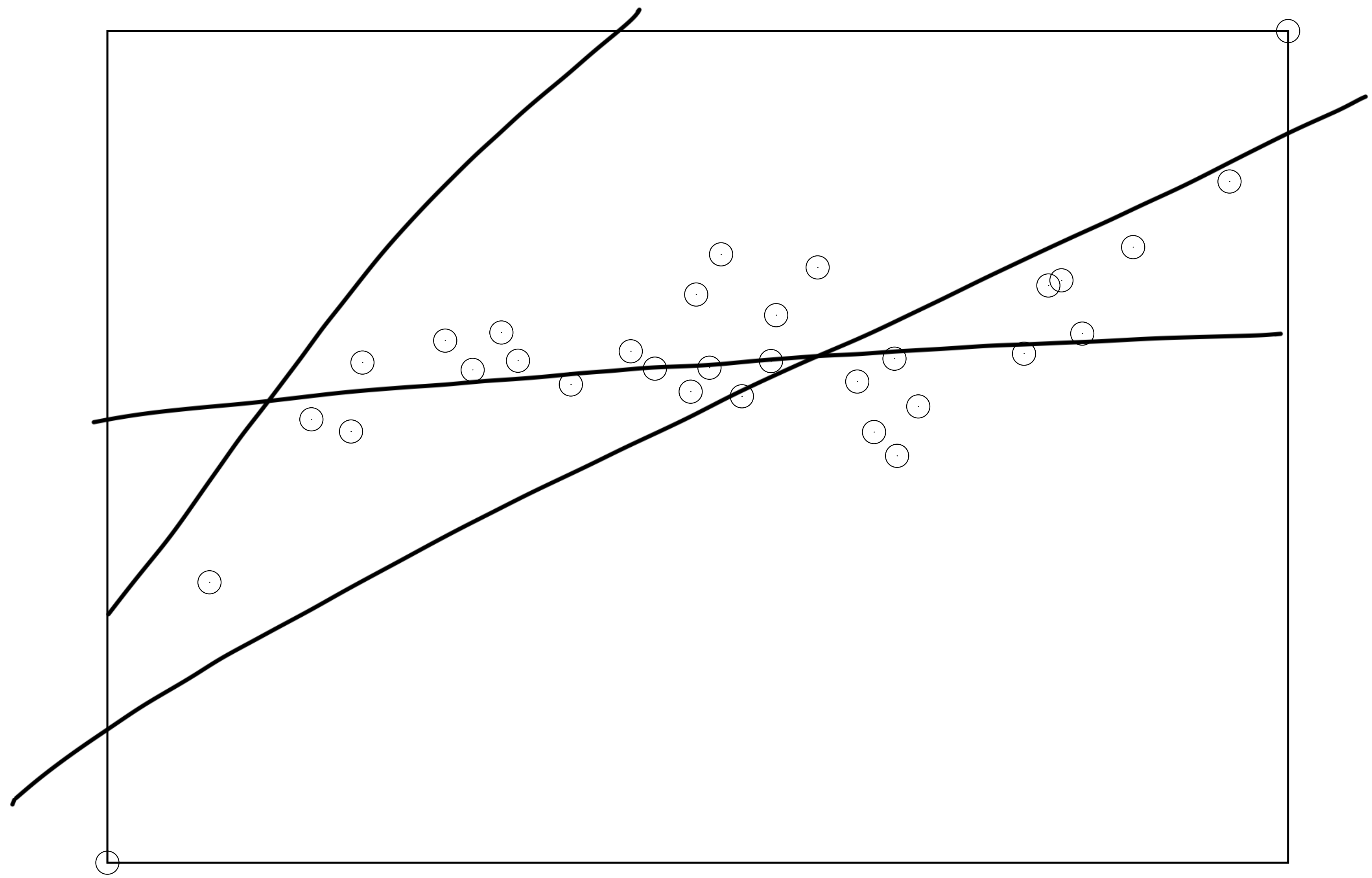
$$\text{RSS} = \sum_{i=1}^{n} \left(w_0 + w_1 x_i - y_i \right)^2$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(w_0 + w_1 x_i - y_i \right)^2$$



Training a model is finding the best parameters

$$\text{find } (w_0, w_1) \text{ that minimizes } \mathrm{RSS}(\mathrm{w}_0, \mathrm{w}_1) = \sum_{i=1}^{n} (w_0 + w_1 x_i - y_i)^2$$



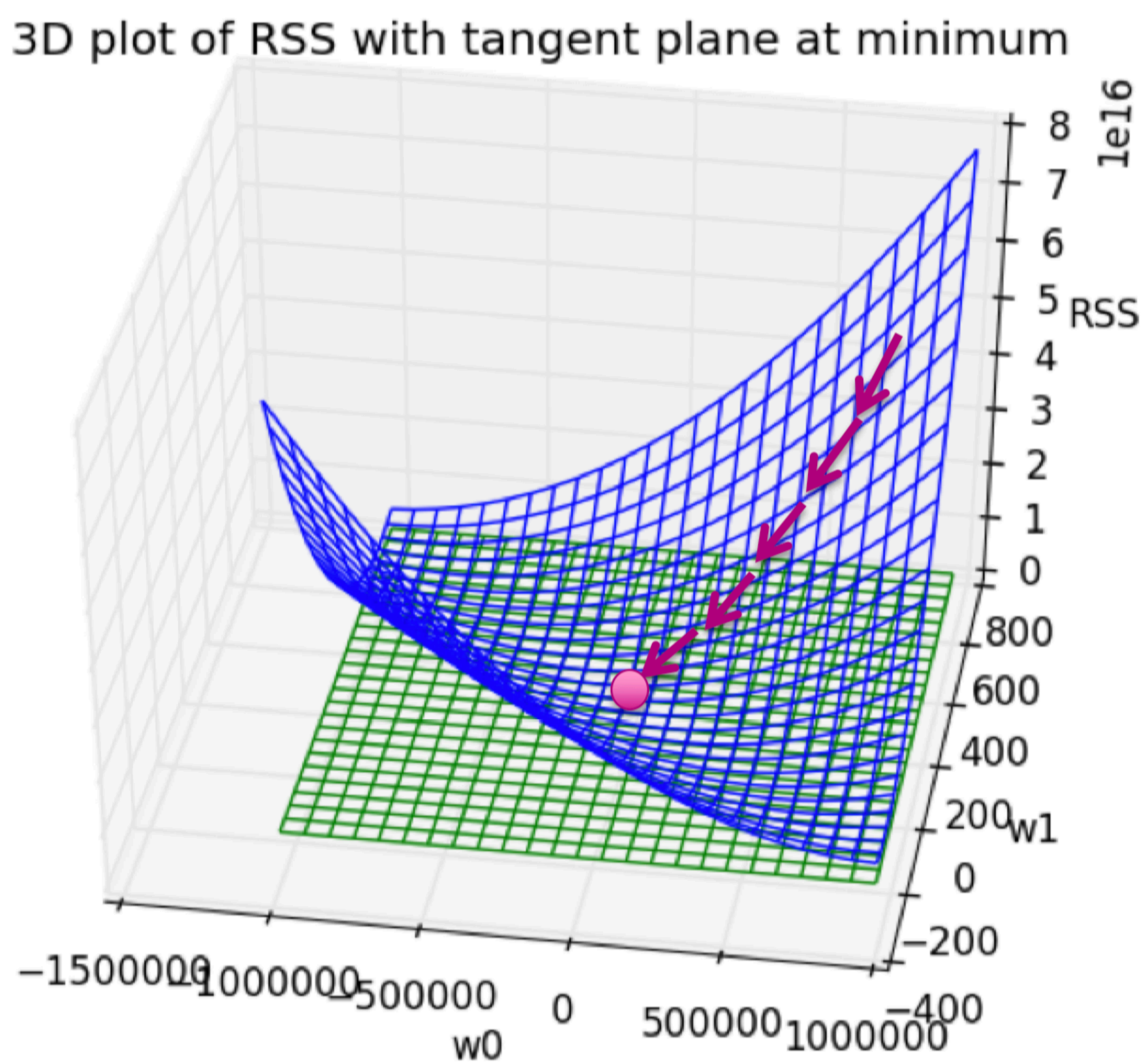
- how does it change if we use MSE = (1/n) RSS ?
- RSS is particularly sensitive to outliers

Algorithm

- find the best fit: $\mathrm{RSS}(\mathrm{w}_0, \mathrm{w}_1) = \sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)^2$

- gradient descent method

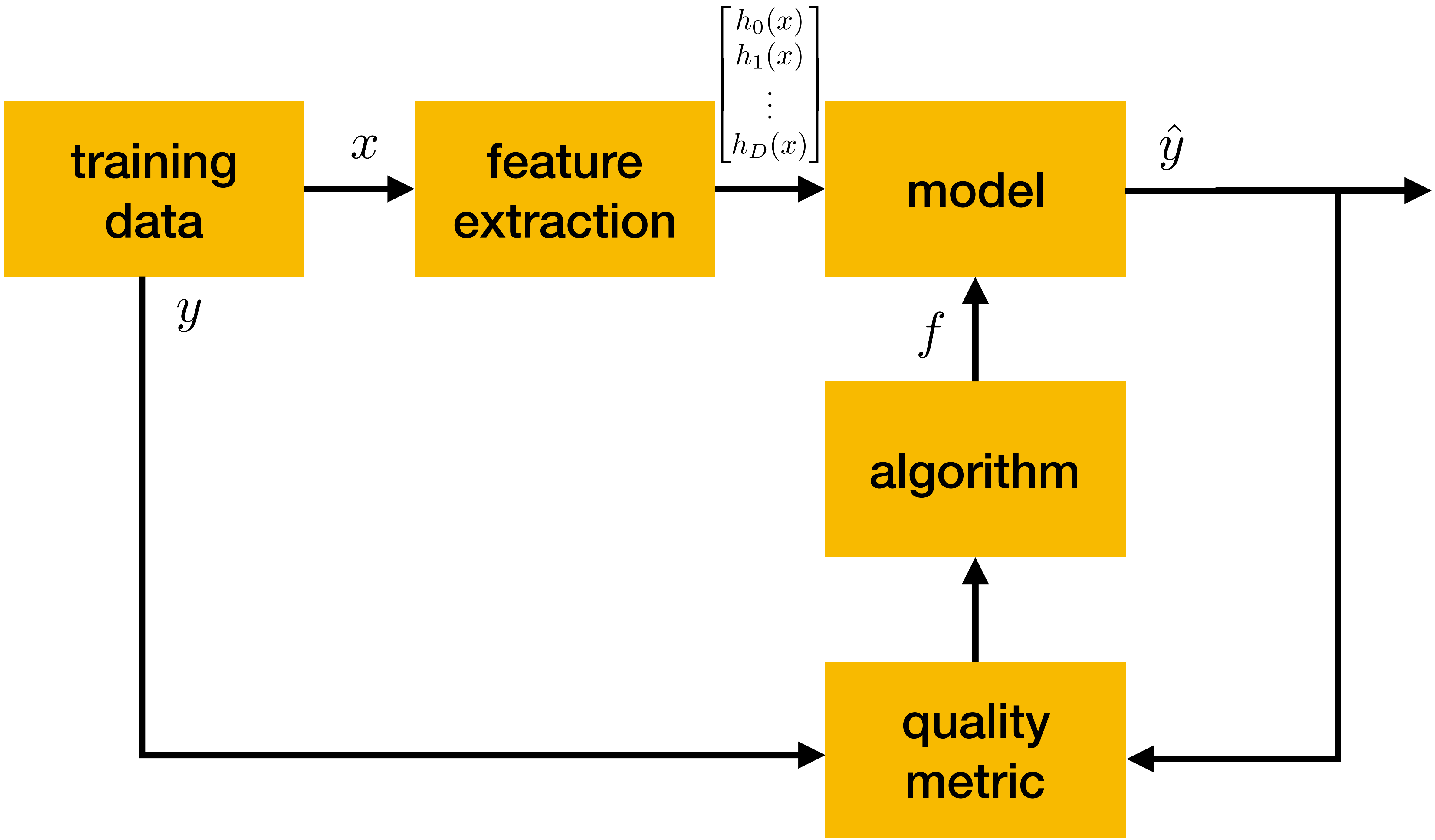
$$w_0^{(t+1)} \leftarrow w_0^{(t)} - \eta \nabla_{w_0} RSS(w^{(t)})$$
$$w_1^{(t+1)} \leftarrow w_1^{(t)} - \eta \nabla_{w_1} RSS(w^{(t)})$$



Linear models with higher order features

so far, $f(x) = w_0 + w_1 \cdot x$

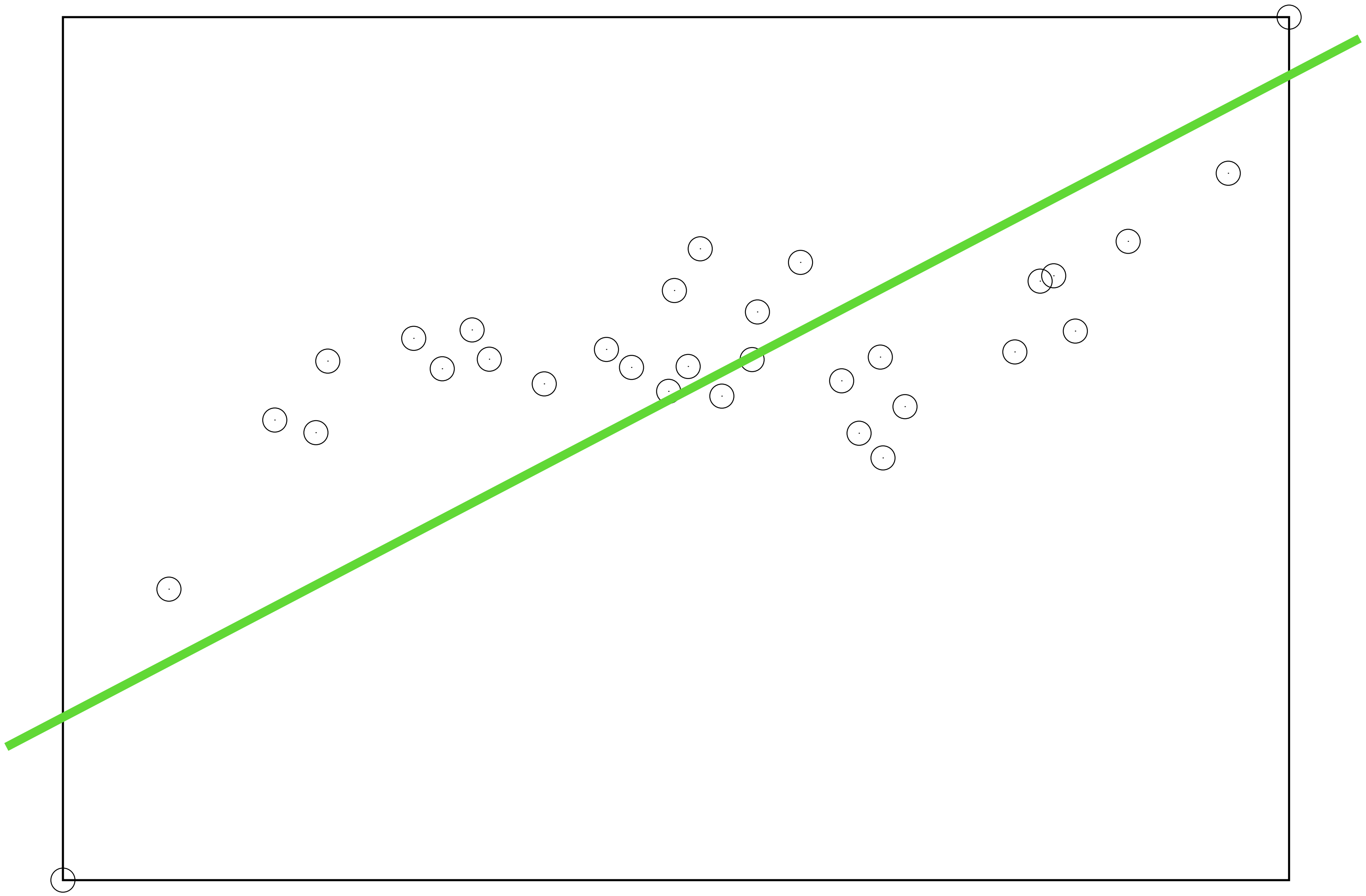
$$f(x) = w_0 + w_1 \cdot h_1(x) + w_2 \cdot h_2(x) + \cdots$$



features are functions that encode the patterns we want to find.

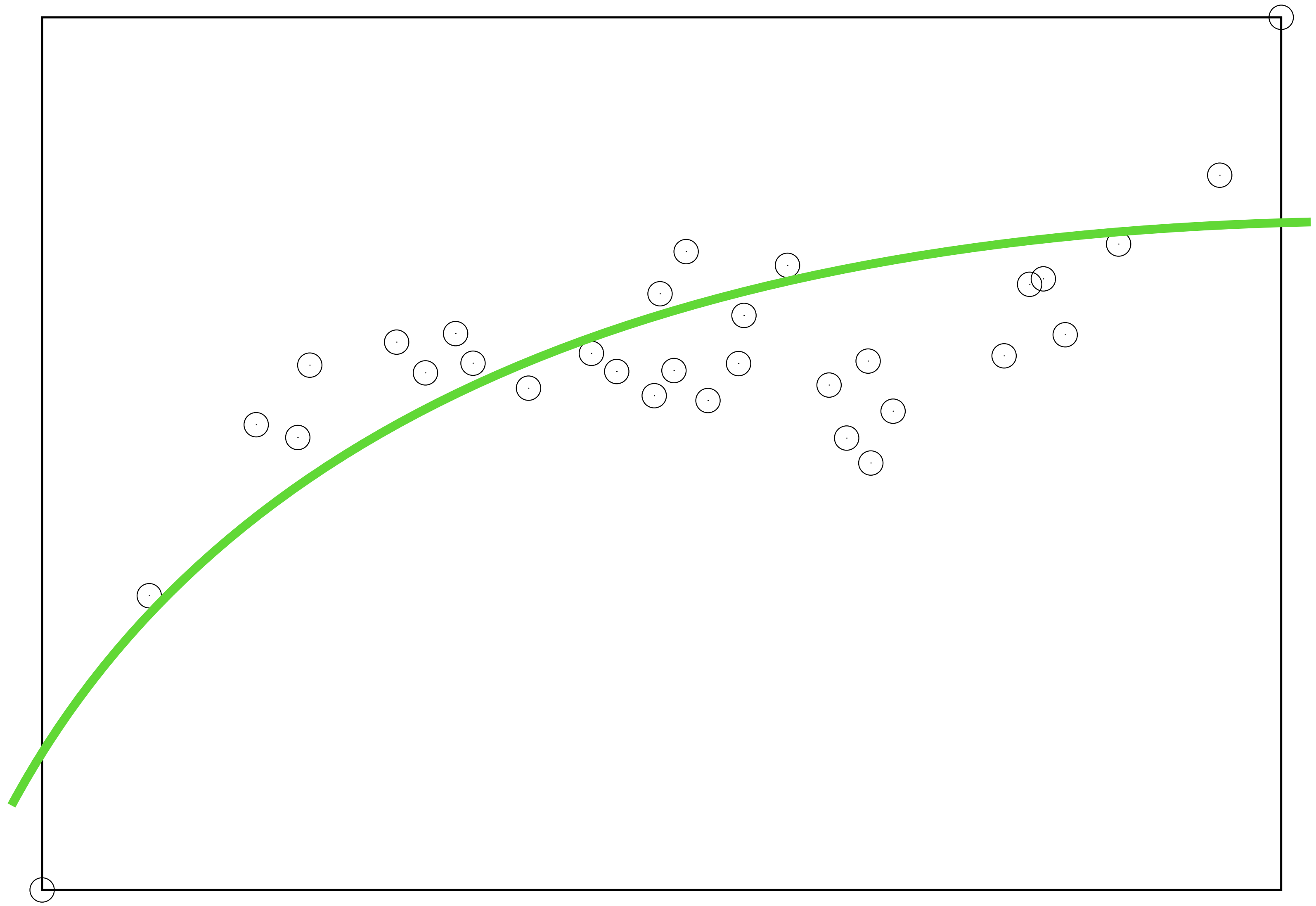
Polynomial features

- linear features: $f(x) = w_0 + w_1 \cdot x$
  fails to capture the relations



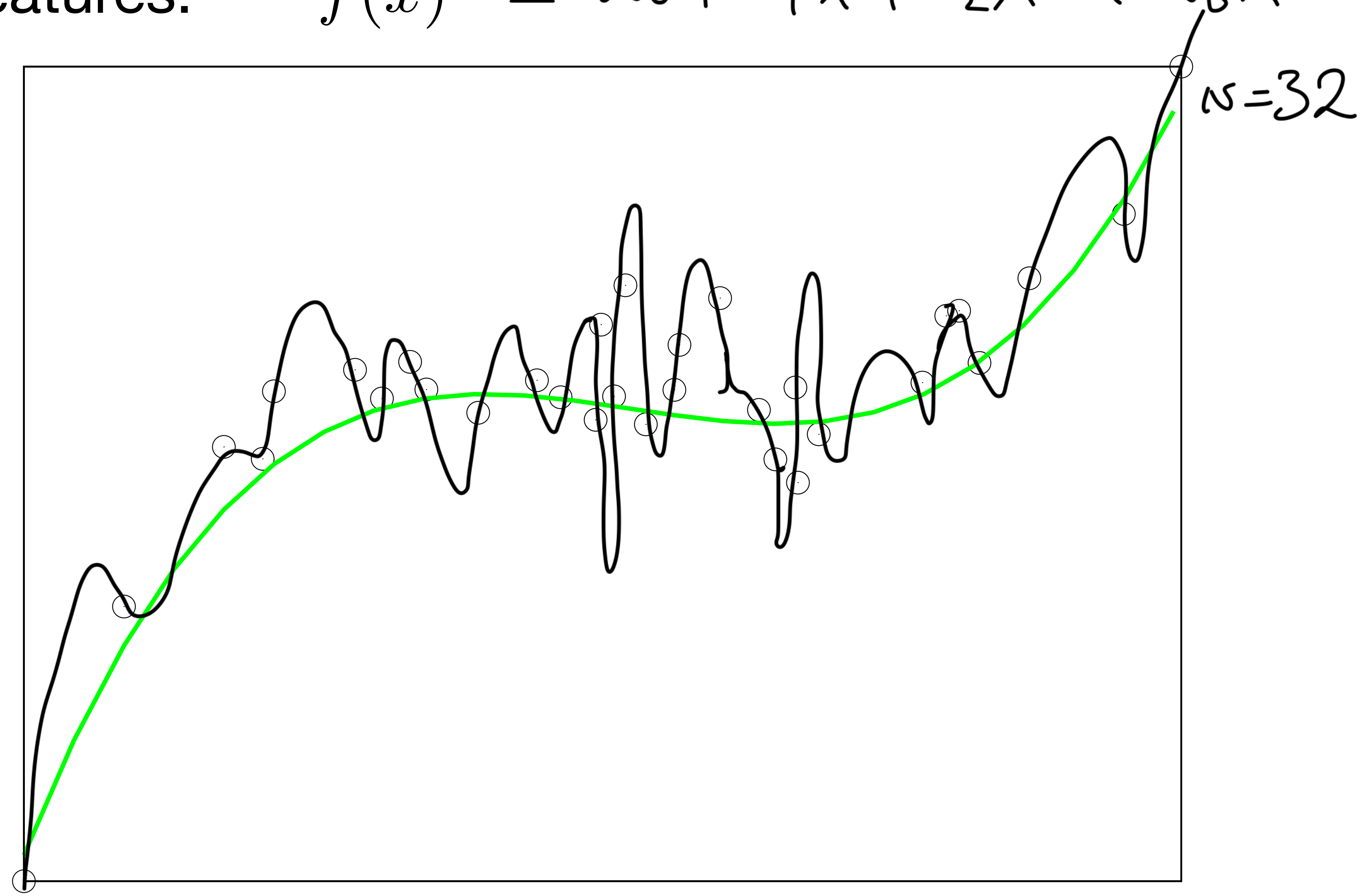
Polynomial features

- quadratic features: $f(x) = w_0 + w_1 x + w_2 x^2$



Polynomial features

- cubic features: $f(x) = w_0 + w_1 X + w_2 X^2 + w_3 X^3$



- in general: $f(x) = w_0 + w_1 x + \cdots + w_p x^p$
- why use polynomial features?
  - good for approximating any functions (what if p=32?)

Polynomial features

- model: linear regression with polynomial features

$$\hat{y} \;=\; f(x) \;=\; w_0 + w_1 x + w_2 x^2 + \cdots + w_p x^p$$

- terminology:

$$\begin{aligned}
&\text{feature } 1 = 1, && \text{parameter } 1 = w_0, \\
&\text{feature } 2 = x, && \text{parameter } 2 = w_1, \\
&\vdots && \vdots \\
&\text{feature } p+1 = x^p, && \text{parameter } p+1 = w_p,
\end{aligned}$$

- but, low degree polynomials might not capture the true relations
  - domain knowledge help

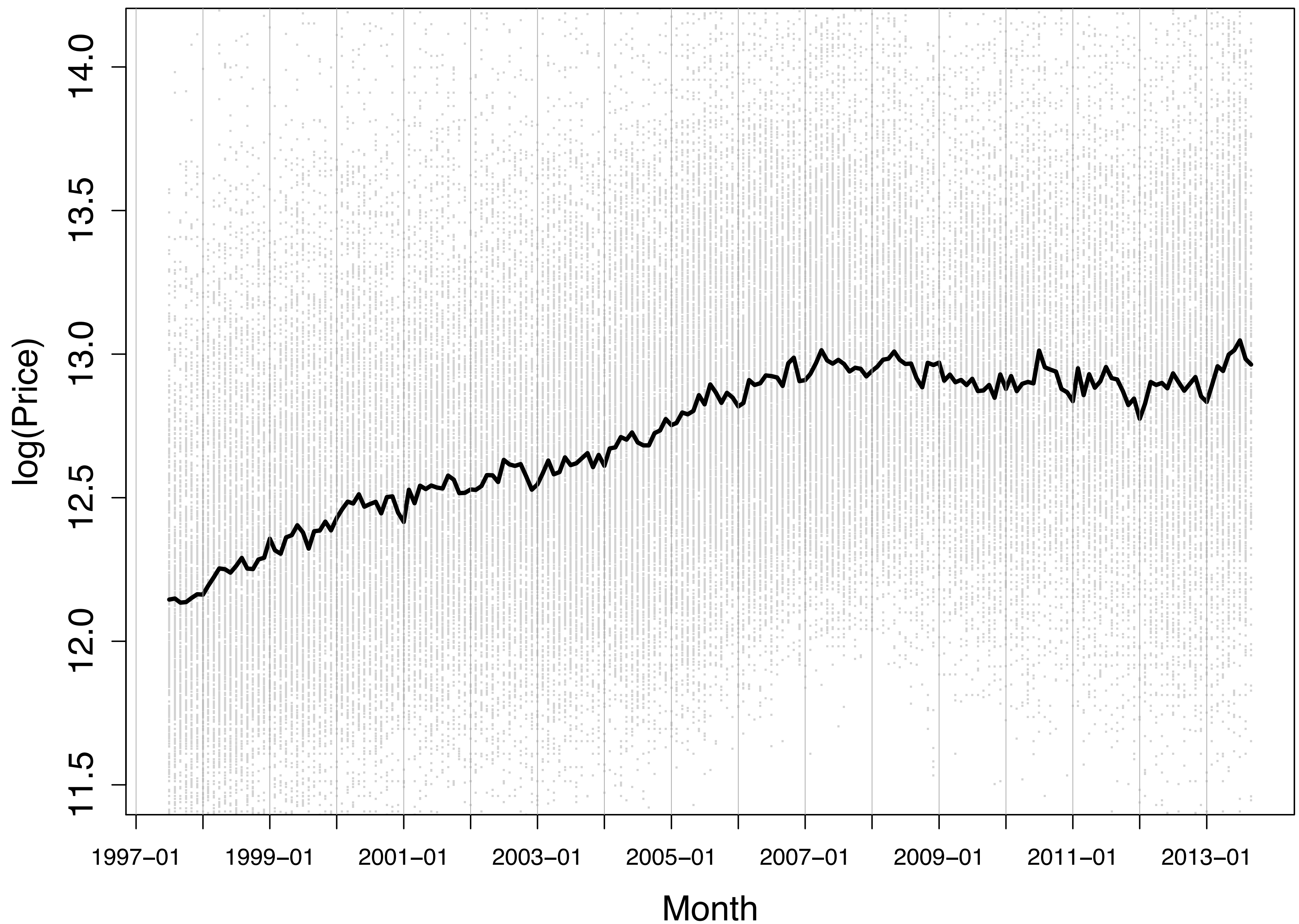
Seasonal features

$$(x_i, y_i) \;= \text{(month-year, average house price)}$$

$$(\text{Jan } 2001, \$255k)$$

$$(\text{Feb } 2001, \$268k)$$

$$\vdots$$



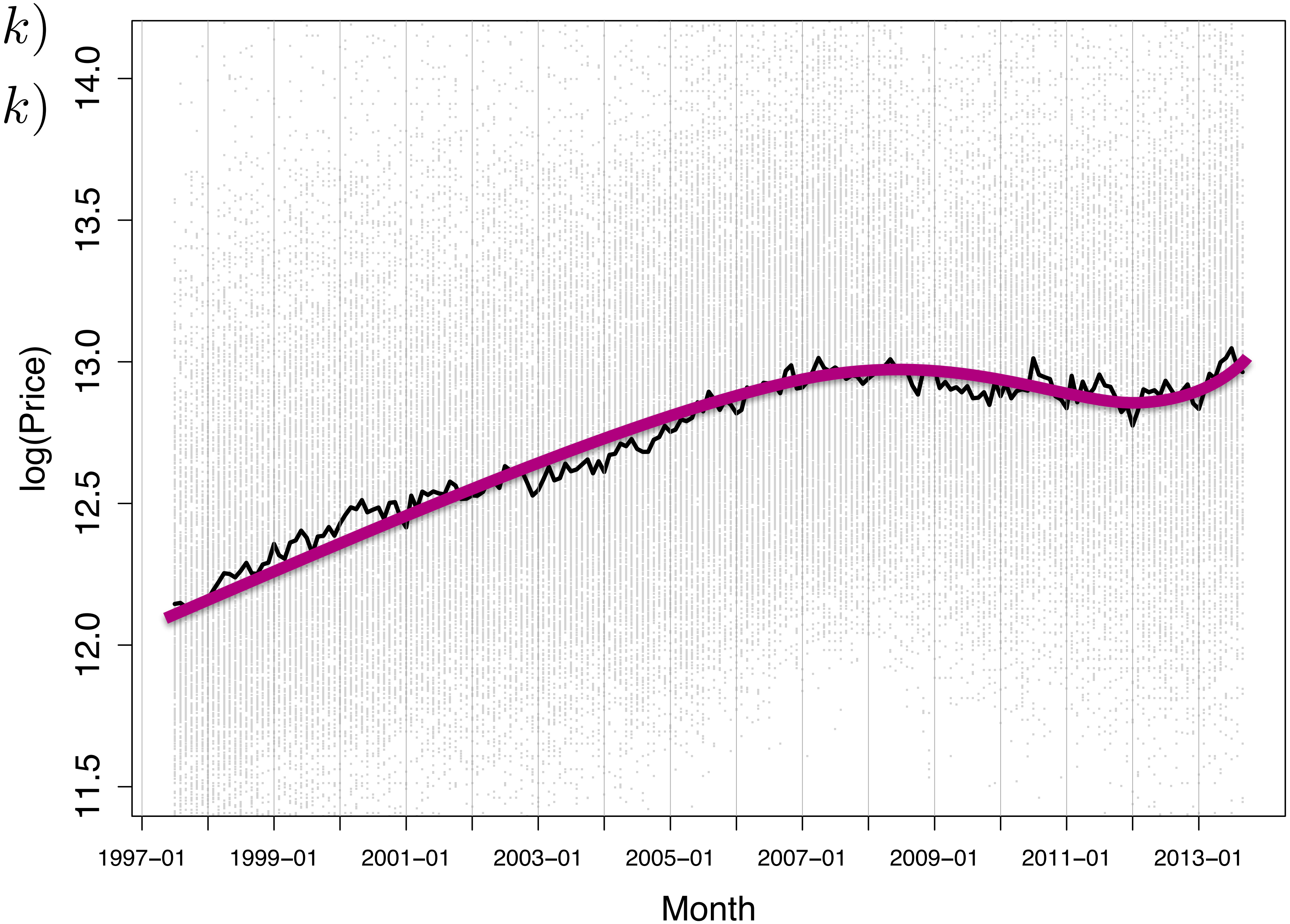
Seasonal features

$$(x_i, y_i) = (\text{month-year, average house price})$$

$$(\text{Jan } 2001, \$255k)$$

$$(\text{Feb } 2001, \$268k)$$

$$\vdots$$



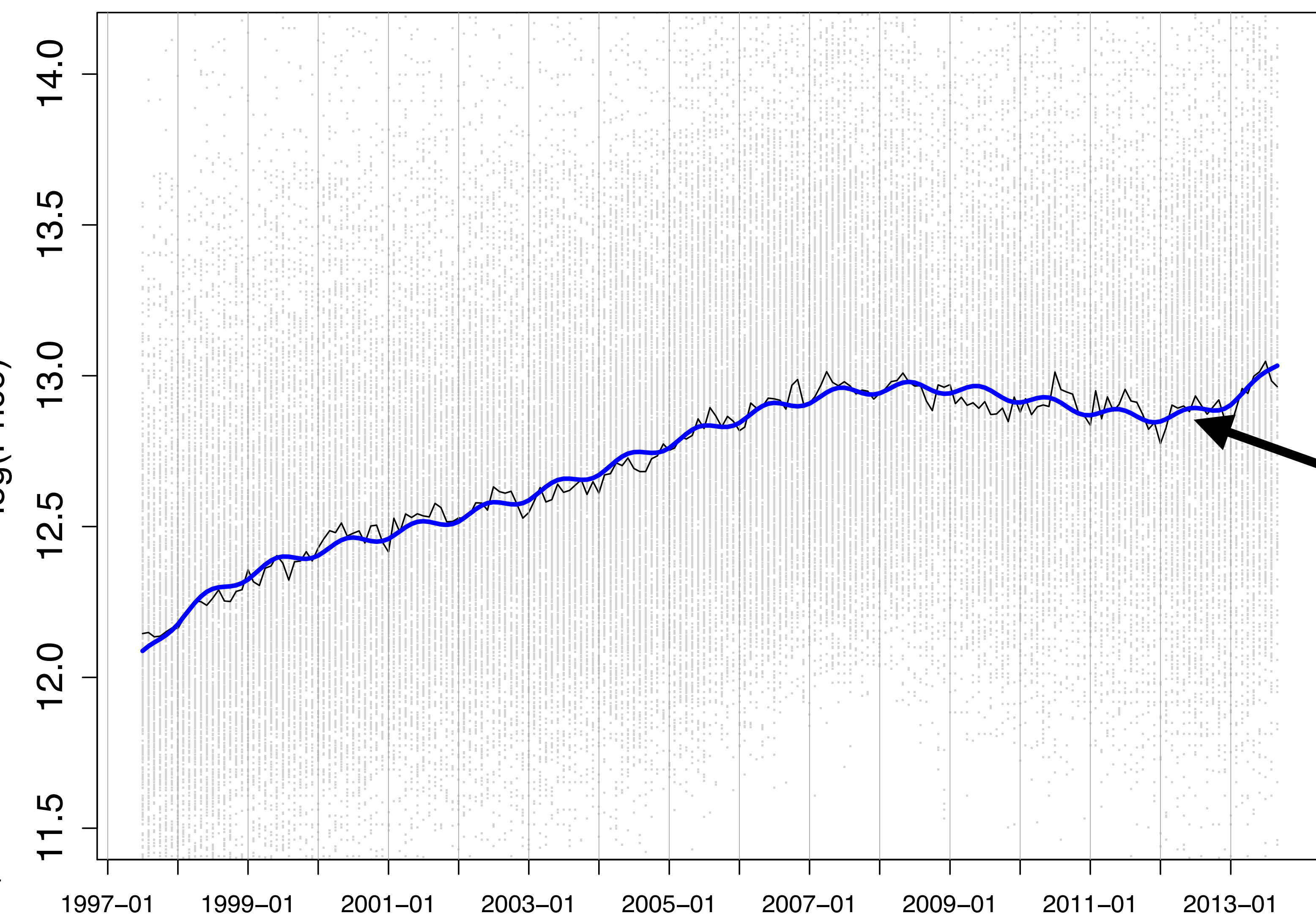
- more buyers in summer drive price higher
- but, best (low-degree) polynomial fit misses the seasonality

Seasonal features

- known relations like seasonality can be manually added as new features

magnitude → w_4, phase → w_5

$$f(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + {\color{red}w_4} \sin\left(\frac{2\pi x}{12} + {\color{red}w_5}\right)$$



best polynomial + sinusoidal fit
but, it is not linear model anymore

Seasonal features

- reparametrization from a sinusoidal model to linear model

magnitude (w_4), phase (w_5)

$$f(x) \;=\; w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 \sin\left(\frac{2\pi x}{12} + w_5\right)$$

$$\text{trigonometric identity} : \sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

$$w_4 \sin\left(\frac{2\pi x}{12} + w_5\right) \;=\; \underbrace{w_4 \cos(w_5)}_{\tilde{w}_4} \sin\left(\frac{2\pi x}{12}\right) + \underbrace{w_4 \sin(w_5)}_{\tilde{w}_5} \cos\left(\frac{2\pi x}{12}\right)$$

$$f(x) \;=\; w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \tilde{w}_4 \sin\left(\frac{2\pi x}{12}\right) + \tilde{w}_5 \cos\left(\frac{2\pi x}{12}\right)$$

feature 5: $\sin\left(\frac{2\pi x}{12}\right)$, feature 6: $\cos\left(\frac{2\pi x}{12}\right)$

- why use sinusoidal features?

Linear models with higher order features

- compact notation of the model

$$\begin{aligned} f(x) &= w_0 h_0(x) + w_1 h_1(x) + \cdots + w_D h_D(x) \\ &= w^T\, h(x) \end{aligned}$$

- vector notation of the model parameters *w* and features *h(x)*

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix} \qquad h(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \sin(2\pi x/12) \\ \cos(2\pi x/12) \end{bmatrix}$$

- as the features are hard coded, human ingenuity/insight needed in feature engineering with domain knowledge
- study guide lines in lectures 5 & 6

Linear models with multi-dimensional input

Input is multi-dimensional for most data

- house price input data:
  area of living space
  garage (no:0, yes:1)
  year built
  area of lot
  year of last remodel
  area of basement
  area of first floor
  area of second floor
  number of bedrooms (above ground)
  number of kitchens (above ground)
  number of fireplaces
  area of garage
  area of wooden deck
  number of half bathrooms
  overall condition (1-10)
  overall quality of materials and finish (1-10)
  number of rooms (above ground)

Input is multi-dimensional for most data

- goal: predicting "How much is my house worth?"

- data

  input $x \in \mathbb{R}^d$ is a d-dimensional vector

  we write it as $x = (x[1], x[2], \cdots, x[d])$

samples	$x[1]$	$x[2]$	$x[3]$	$\cdots$	$x[d]$	y
$1st$ sample	$2571\ sq.ft.$	1	2001	$\cdots$	3	$\$238k$
$\vdots$						
ith sample	$3942\ sq.ft.$	1	2018	$\cdots$	5	$\$451k$
$\vdots$						
Nth sample	$3690\ sq.ft.$	0	1987	$\cdots$	5	$\$362k$

y_i

$x_i[j]$

we use N to denote the number of samples

outcome can be multi-dimensional also

Linear model with multi-dimensional input

- D-dimensional feature extraction

$$h(x) : \mathbf{R}^d \rightarrow \mathbf{R}^{D+1}$$

$$h(x) = (h_0(x), h_1(x), \ldots, h_D(x))$$

- model

$$\begin{aligned} f(x) &= \sum_{j=0}^{D} w_j \, h_j(x) \\ &= w^T h(x) \end{aligned}$$

- quality metric

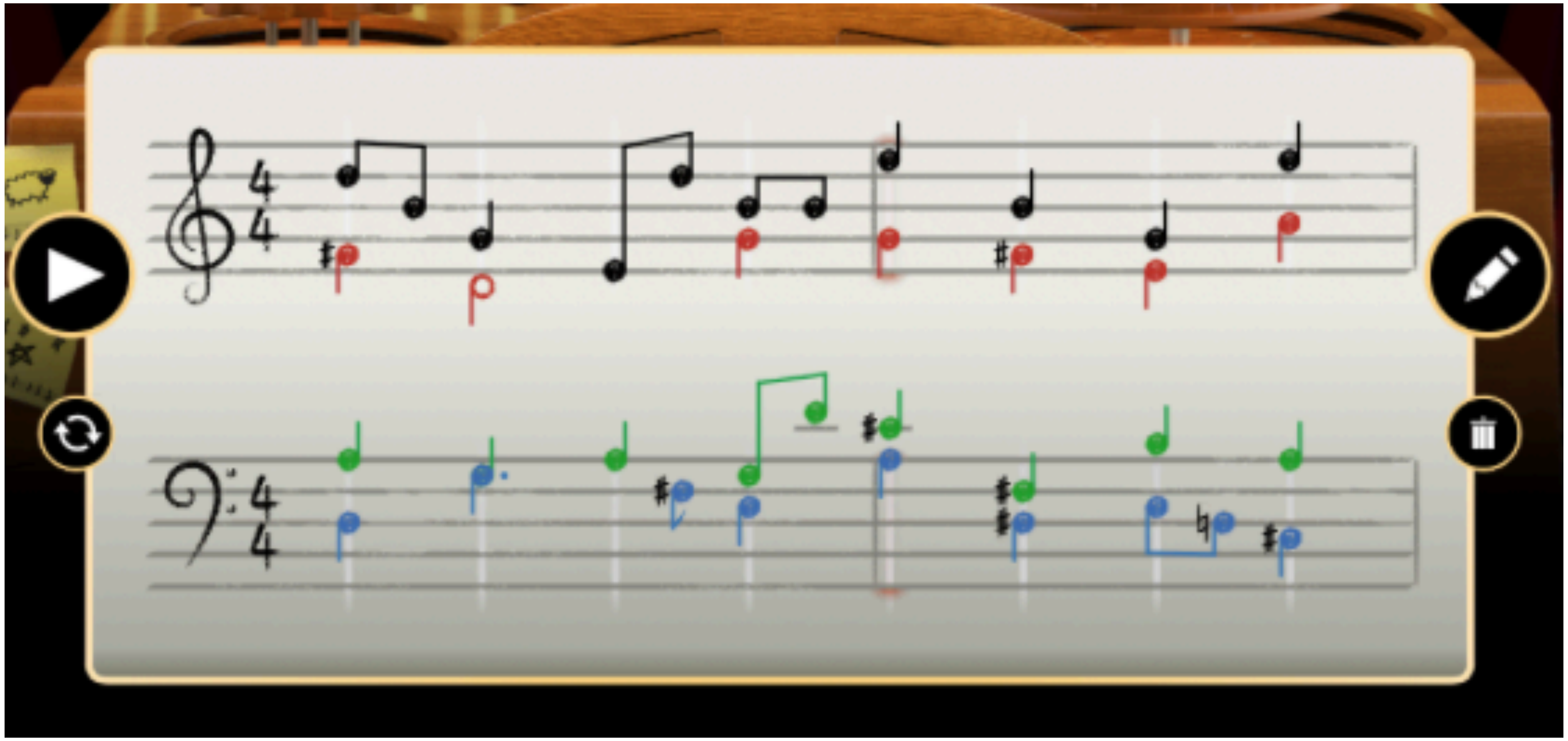
$$\mathrm{RSS}(w) = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

Modern machine learning tasks are complex

- predict “How old is this person?”

- how do we know which feature to use?

- study automated feature extraction using deep neural networks in lectures 18-20

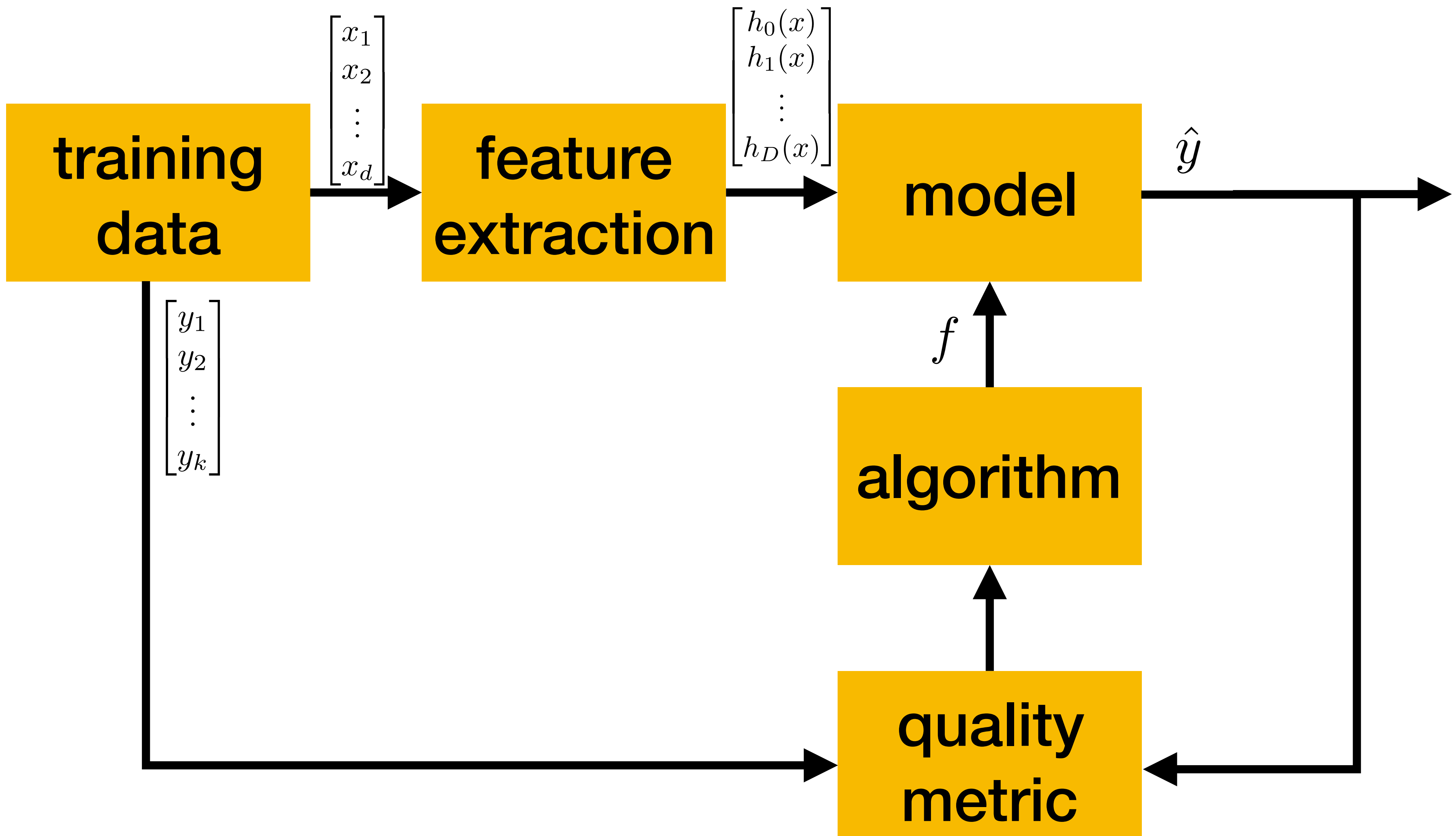


soprano (S)

alto (A)

tenor (T)

bass (B



$d = 16$

$K = 16 \times 3 = 48$