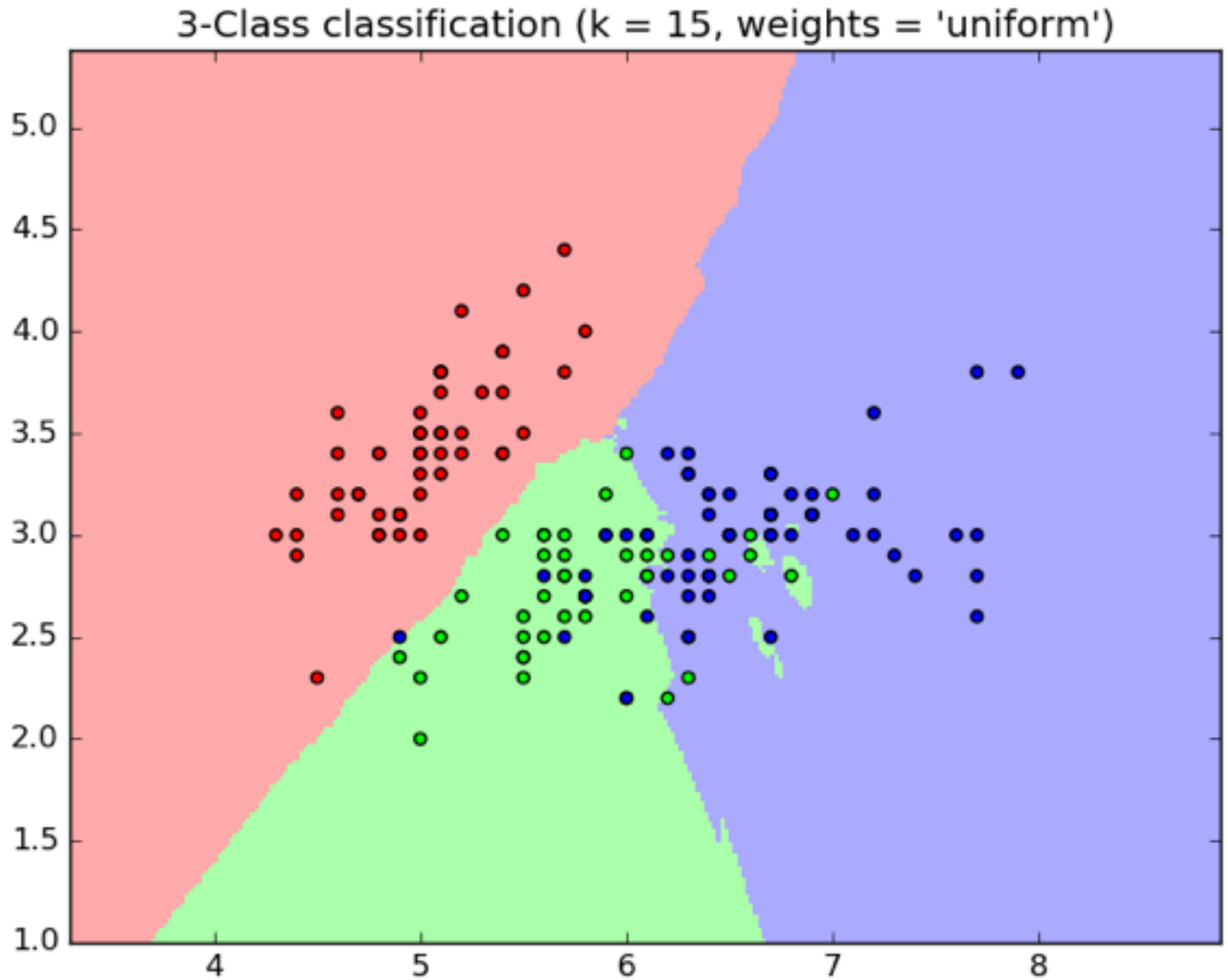


Machine Learning

EE382V Activity Sensing and Recognition

UT Austin • Dept. Electrical and Computer Engineering • Fall 2016

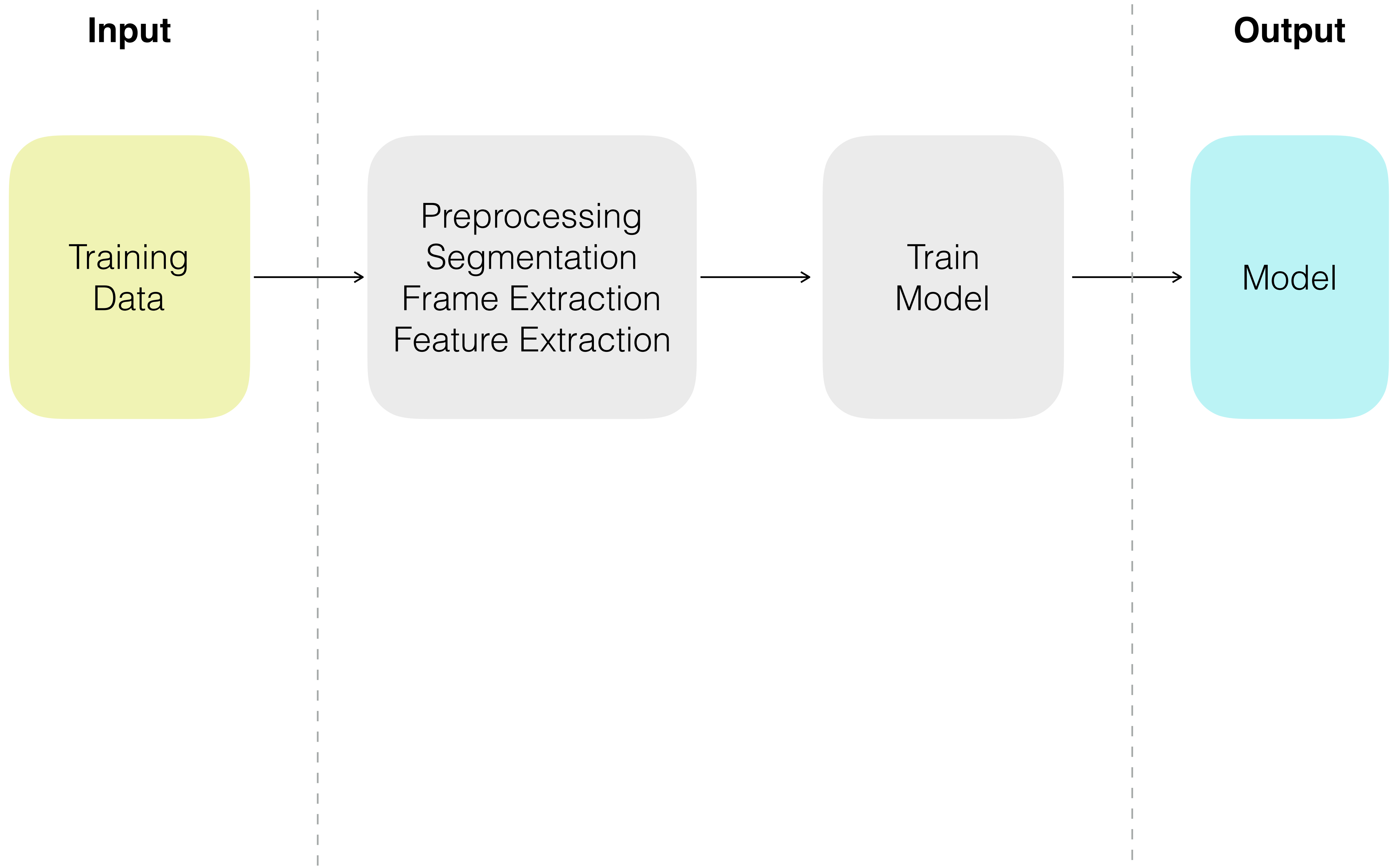
Today

Machine Learning (45mins)

- Supervised Learning & Terminology
- Linear Regression & Decision Trees
- Discriminative and Generative Models
- Bias and Variance
- Overfitting & Underfitting

Projects (15mins)

Activity Recognition Pipeline



What is ML

Field of study that gives computers the ability to learn without being explicitly programmed.
— Arthur Samuel, 1959

Study of algorithms that:

- improve their performance P
- at some task T
- with experience E

well-defined learning task: <P,T,E>

Types of ML

- Based on *information available*
  - Supervised – true labels provided
  - Reinforcement – Only indirect labels provided (reward/punishment)
  - Unsupervised – No feedback & no labels
- Based on the *role of the learner*
  - Passive – given a set of data, produce a model
  - Online – given one data point at a time, update model
  - Active – ask for specific data points to improve model
- Based on *type of output*
  - Concept Learning – Binary output based on +ve/-ve examples
  - Classification – Classifying into one among many classes
  - Regression – Numeric, ordered output

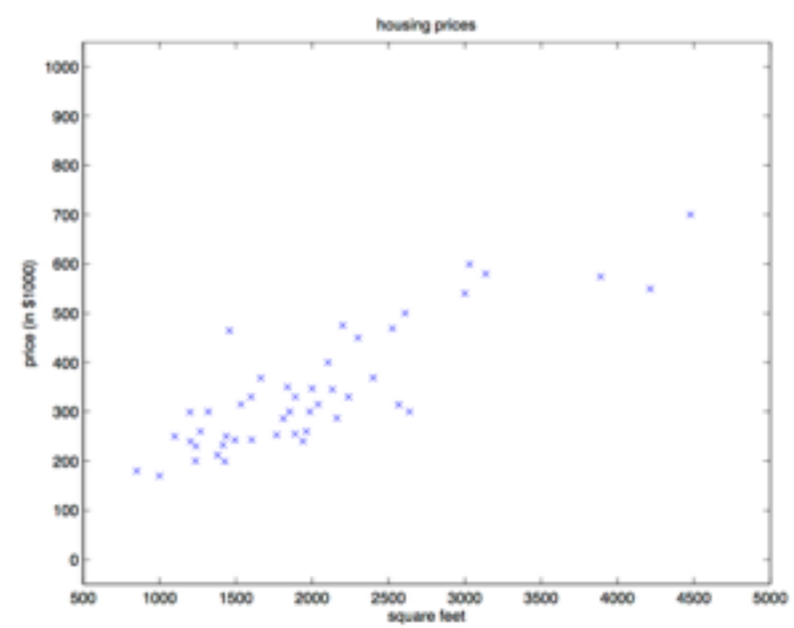
Source: Sofus Macskassy CS567 Lecture Notes

Supervised Learning

Dataset of home sale prices in Austin

Dataset includes the living area of homes in sq. feet

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



Goal: Predict future home prices as a function of living area size

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

Supervised Learning

Dataset of breast tumors

Dataset includes characteristic of tumors

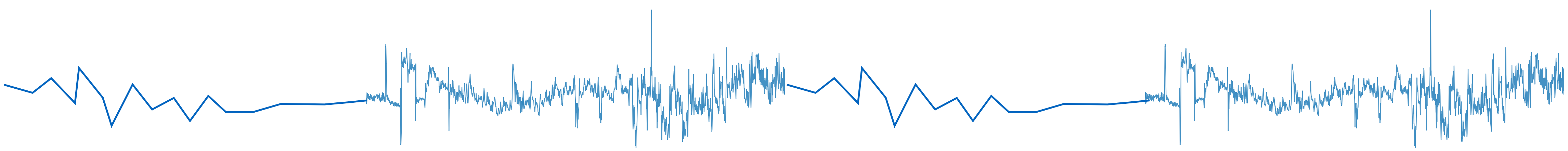
tumor size	texture	perimeter	...	outcome	time
18.02	27.60	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

Goal: Predict outcomes of future tumors

Source: (Based in part on) Sofus Macskassy CS567 Lecture Notes

Supervised Learning

Dataset of sensor features



Mean	Variance	Kurtosis	Running
0.34	0.76	0.67	No
0.75	0.54	0.99	Yes
0.02	0.23	0.79	Yes
…	…	…	…

Goal: Predict future instances of running activity

Supervised Learning

Terminology

Mean	Variance	Kurtosis	Running
0.34	0.76	0.67	No
0.75	0.54	0.99	Yes
0.02	0.23	0.79	Yes
…	…	…	…

Features / Attributes

Supervised Learning

Terminology

Mean	Variance	Kurtosis	Running
0.34	0.76	0.67	No
0.75	0.54	0.99	Yes
0.02	0.23	0.79	Yes
...	...	...	...

Labels

Supervised Learning

Terminology

Mean	Variance	Kurtosis	Running
0.34	0.76	0.67	No
0.75	0.54	0.99	Yes
0.02	0.23	0.79	Yes
…	…	…	…

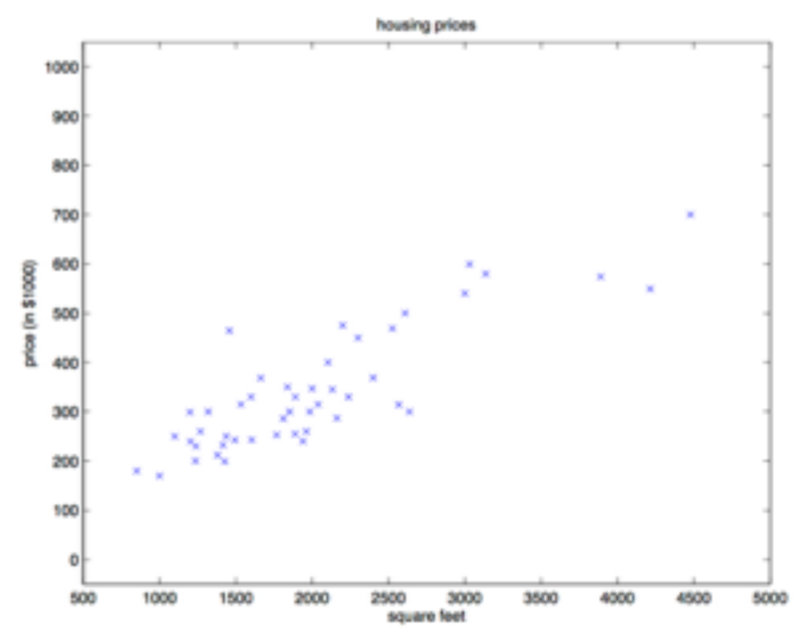
Instance / Training Example

Supervised Learning

Dataset of home sale prices in Austin

Dataset includes the living area of homes in sq. feet

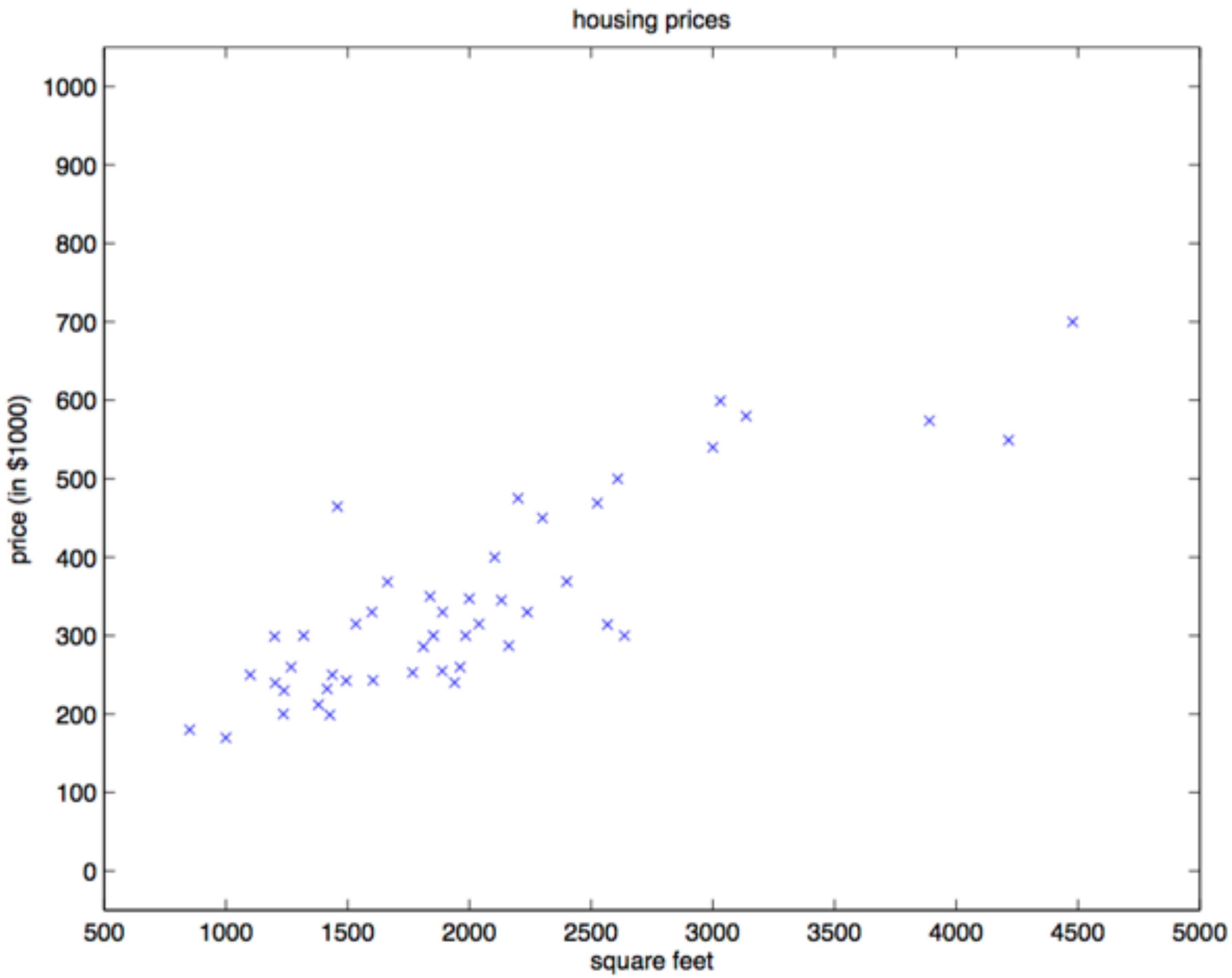
Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



Goal: Predict future home prices as a function of living area size

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

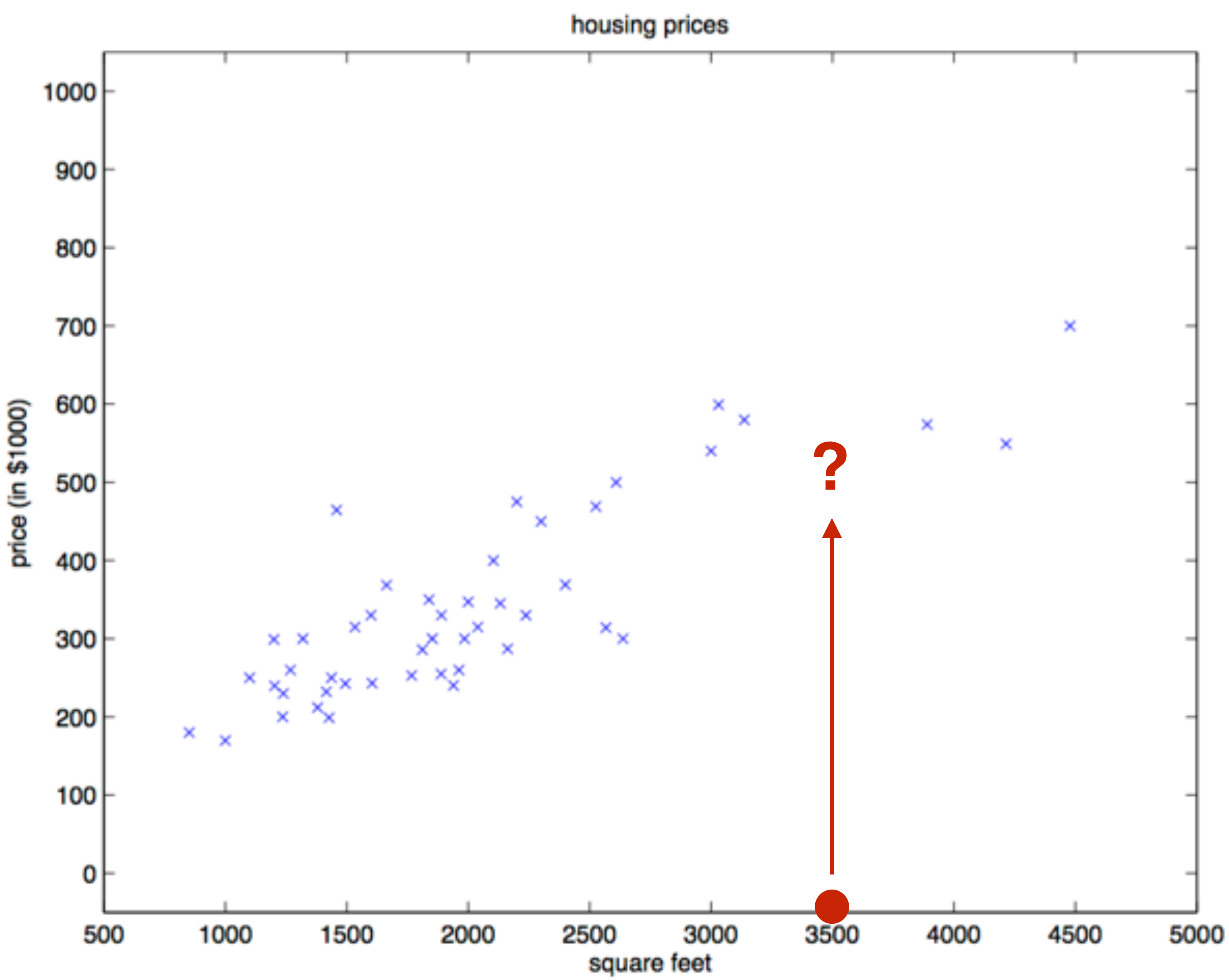
Supervised Learning



Goal: Predict future home prices as a function of living area size

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

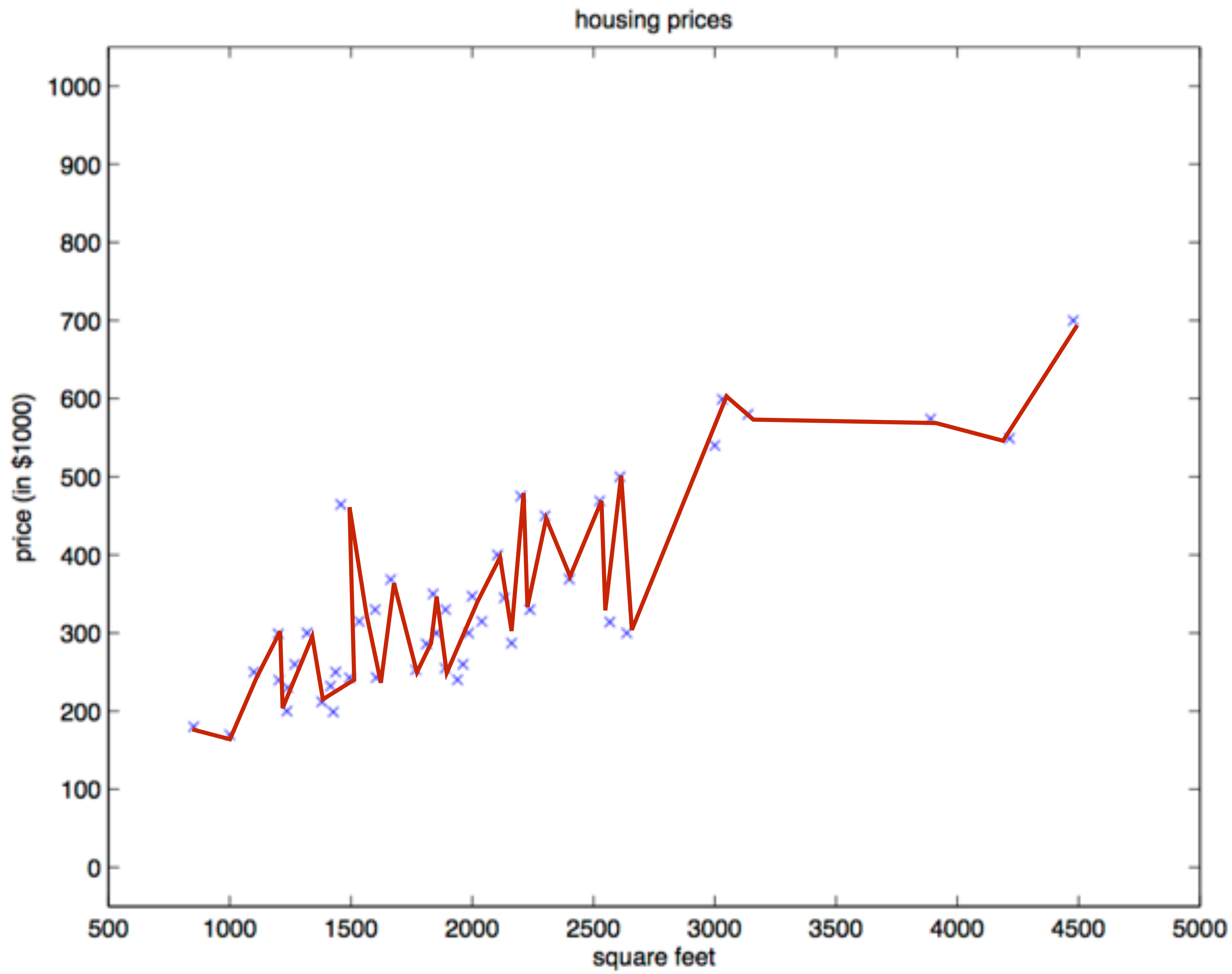
Supervised Learning



Function (f) : X → Y

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

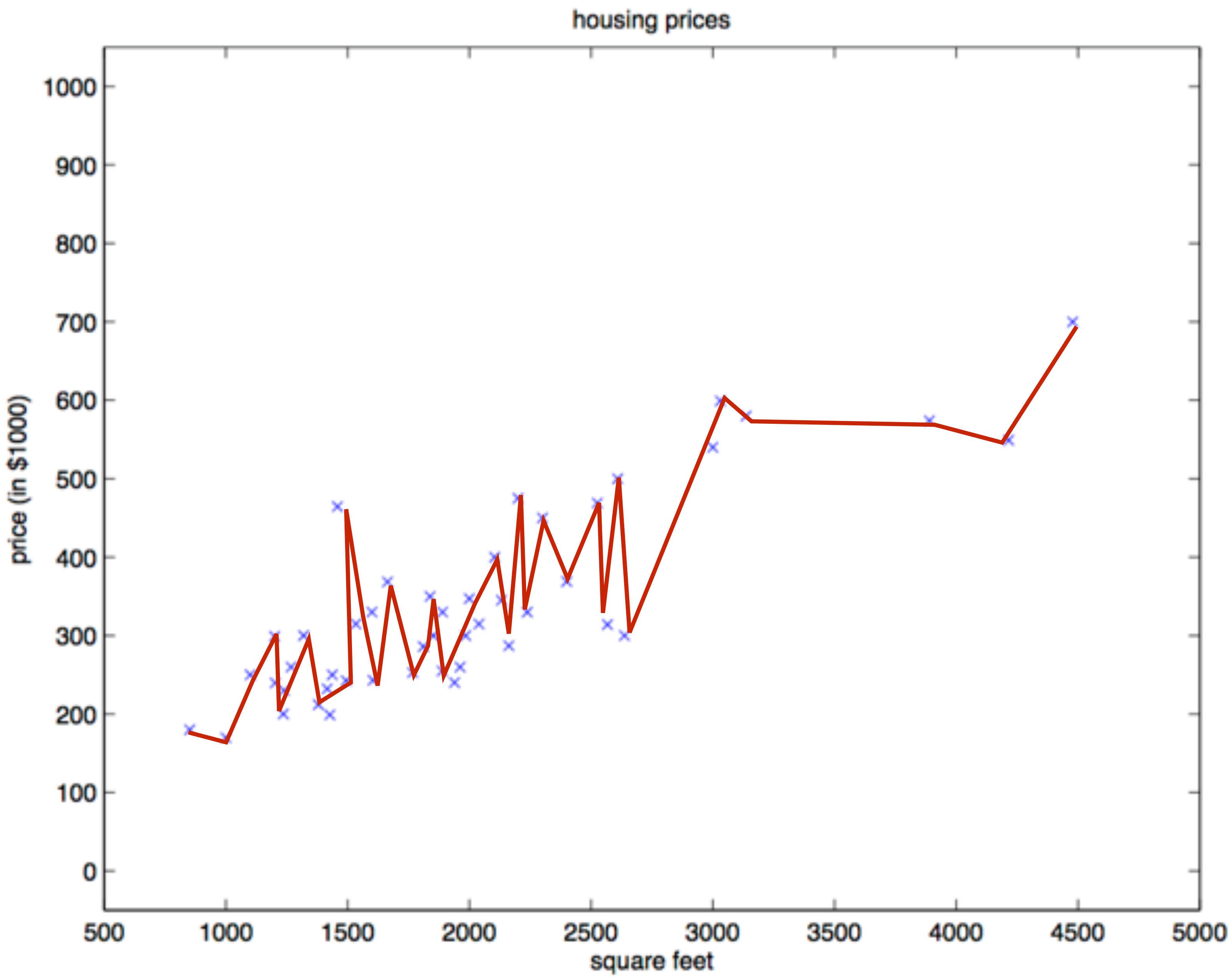
Supervised Learning



There is an unknown target function (f) mapping the X,Y relationship

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

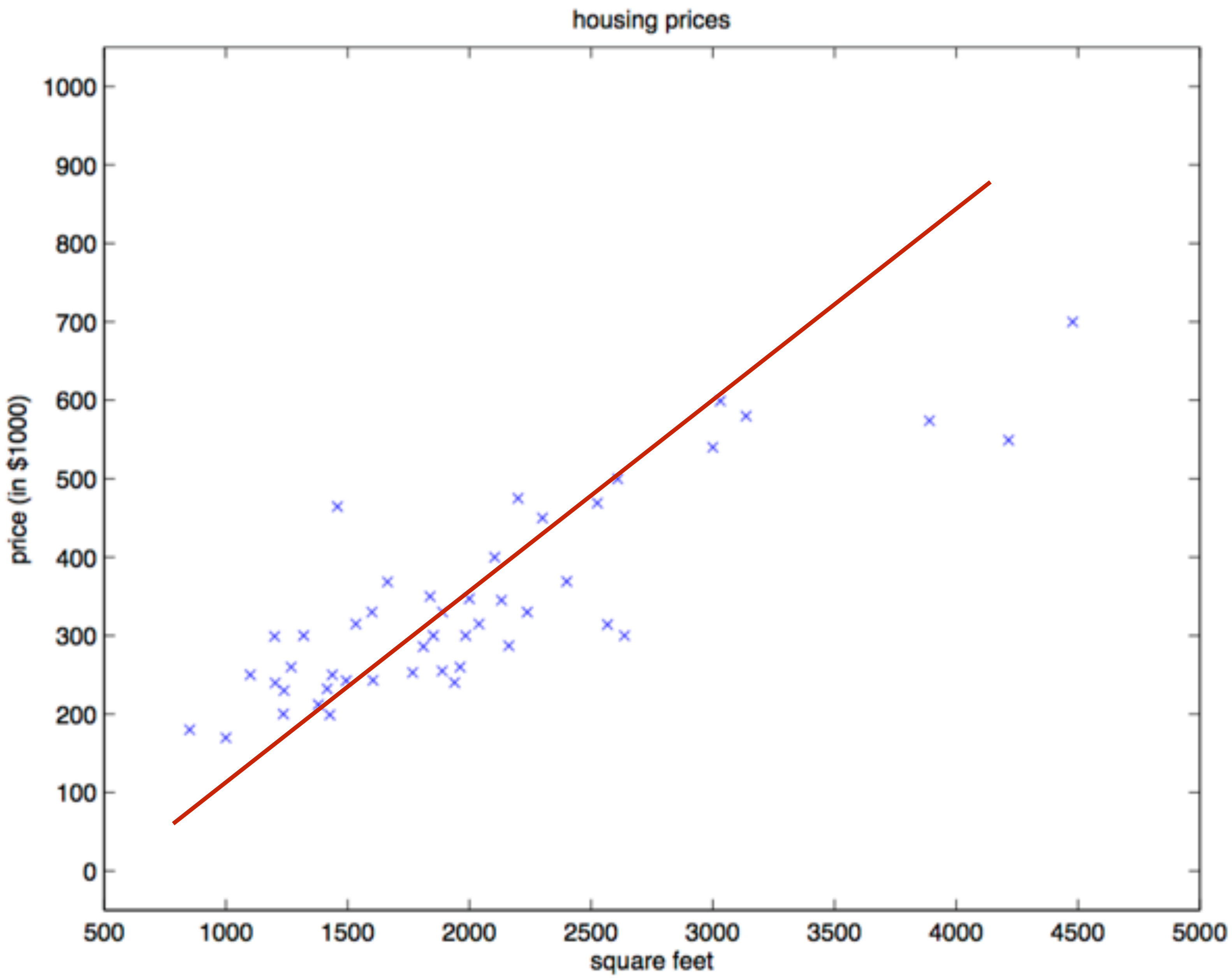
Supervised Learning



But often not useful to predict future values for Y

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

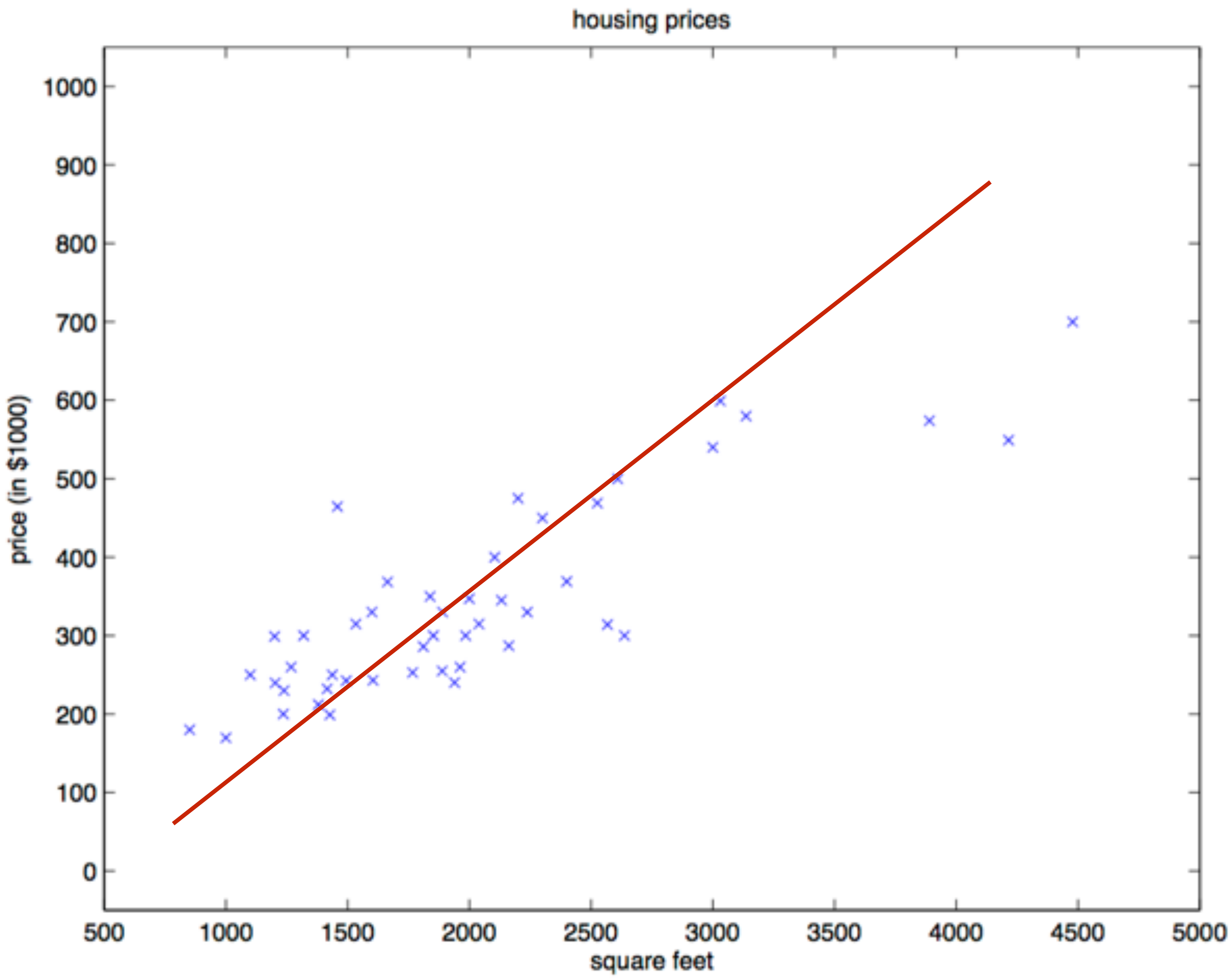
Supervised Learning



Our goal: learn a function $h : X \rightarrow Y$ that approximates f

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

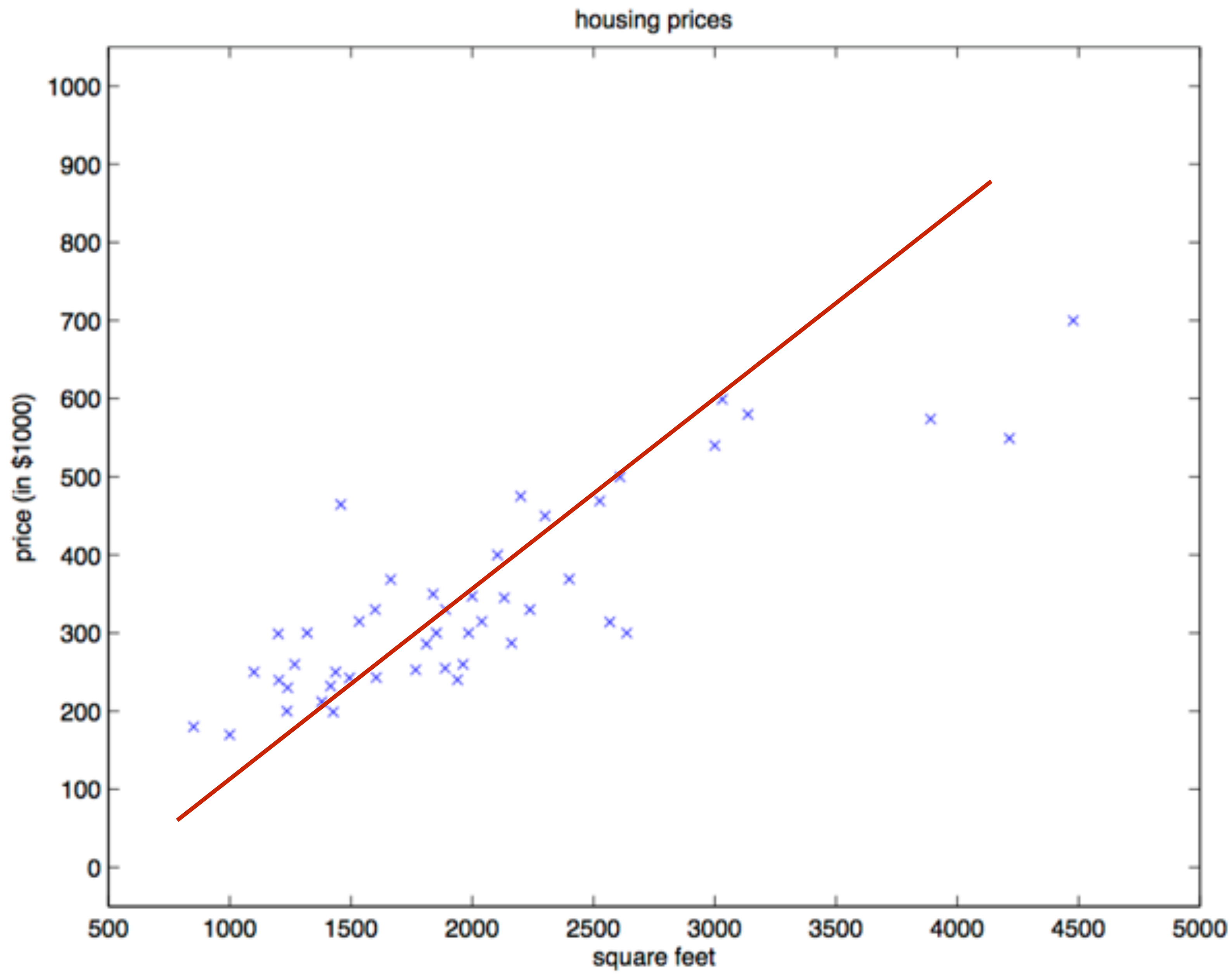
Supervised Learning



Need to use a learning algorithm!

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

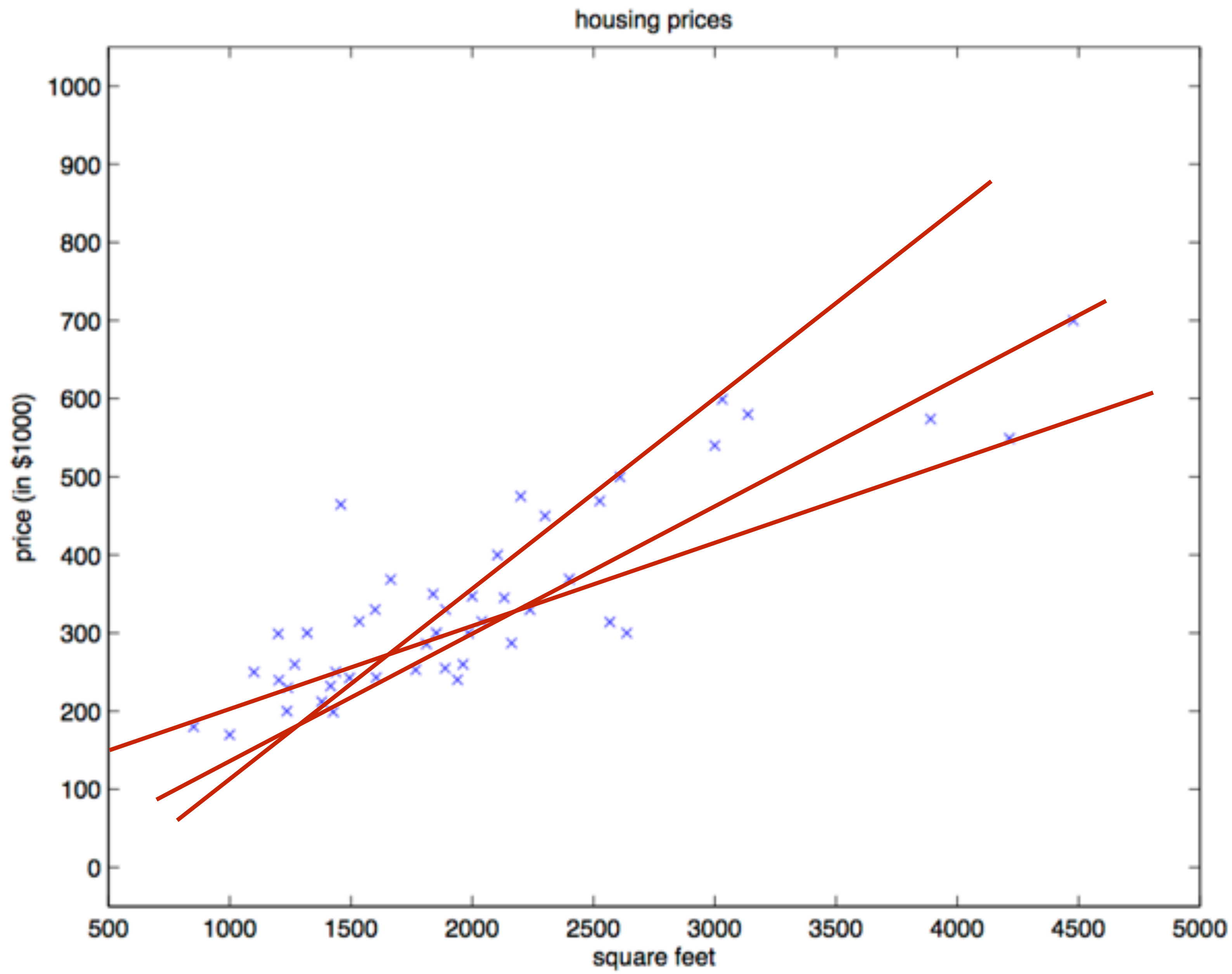
Learning Algorithm



e.g., linear regression outputs linear model

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

Supervised Learning

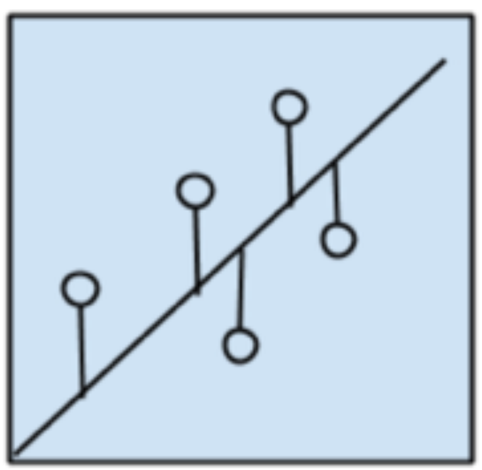


Many forms of h(x): Hypothesis Space

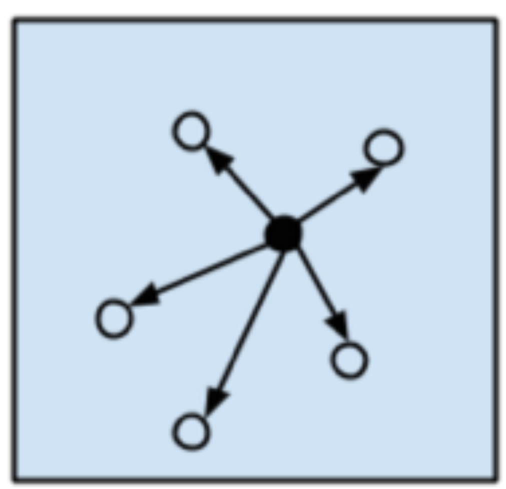
Source: (Based in part on) Andrew Ng CS229 Lecture Notes

Learning Algorithms

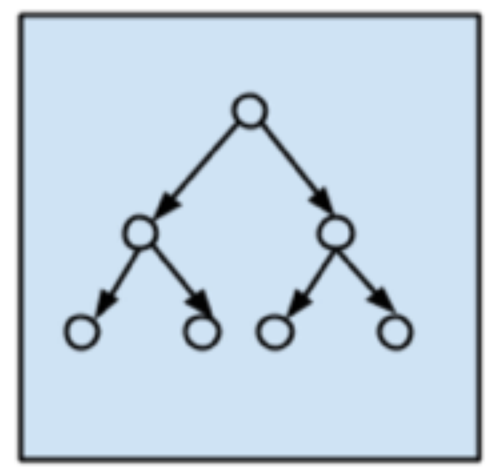
Different learning algorithms output different types of hypothesis spaces. Many representations possible



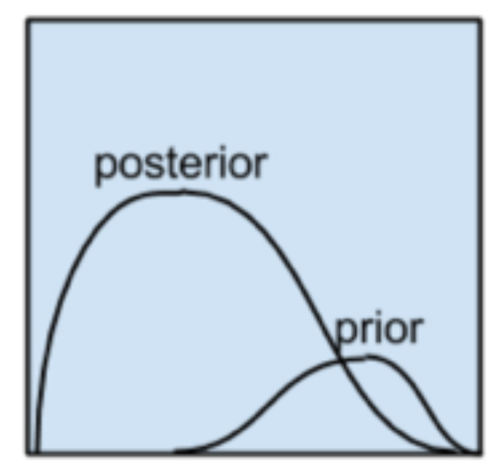
Regression Algorithms



Instance-based Algorithms



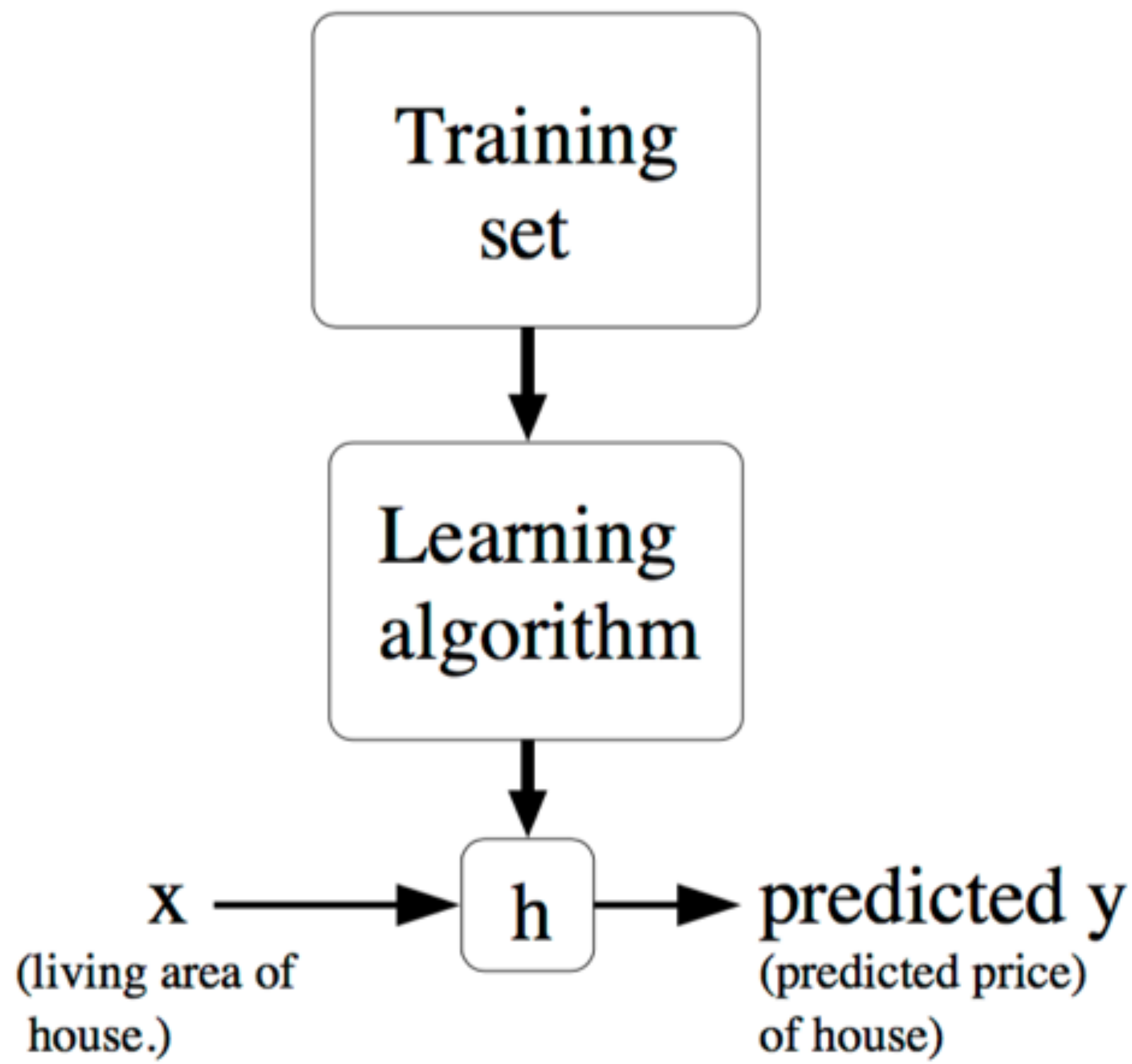
Decision Tree Algorithms



Bayesian Algorithms

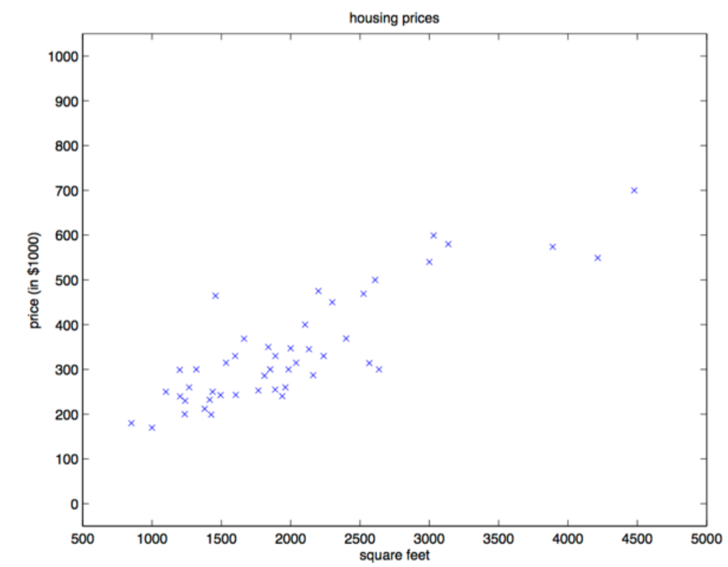
Source: http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Learning Algorithm



Source: Andrew Ng CS229 Lecture Notes

Function Approximation



Problem Setting:

- Set of possible instances X
- Unknown target function $f : X \rightarrow Y$
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$

superscript: i^{th} training example

Input:

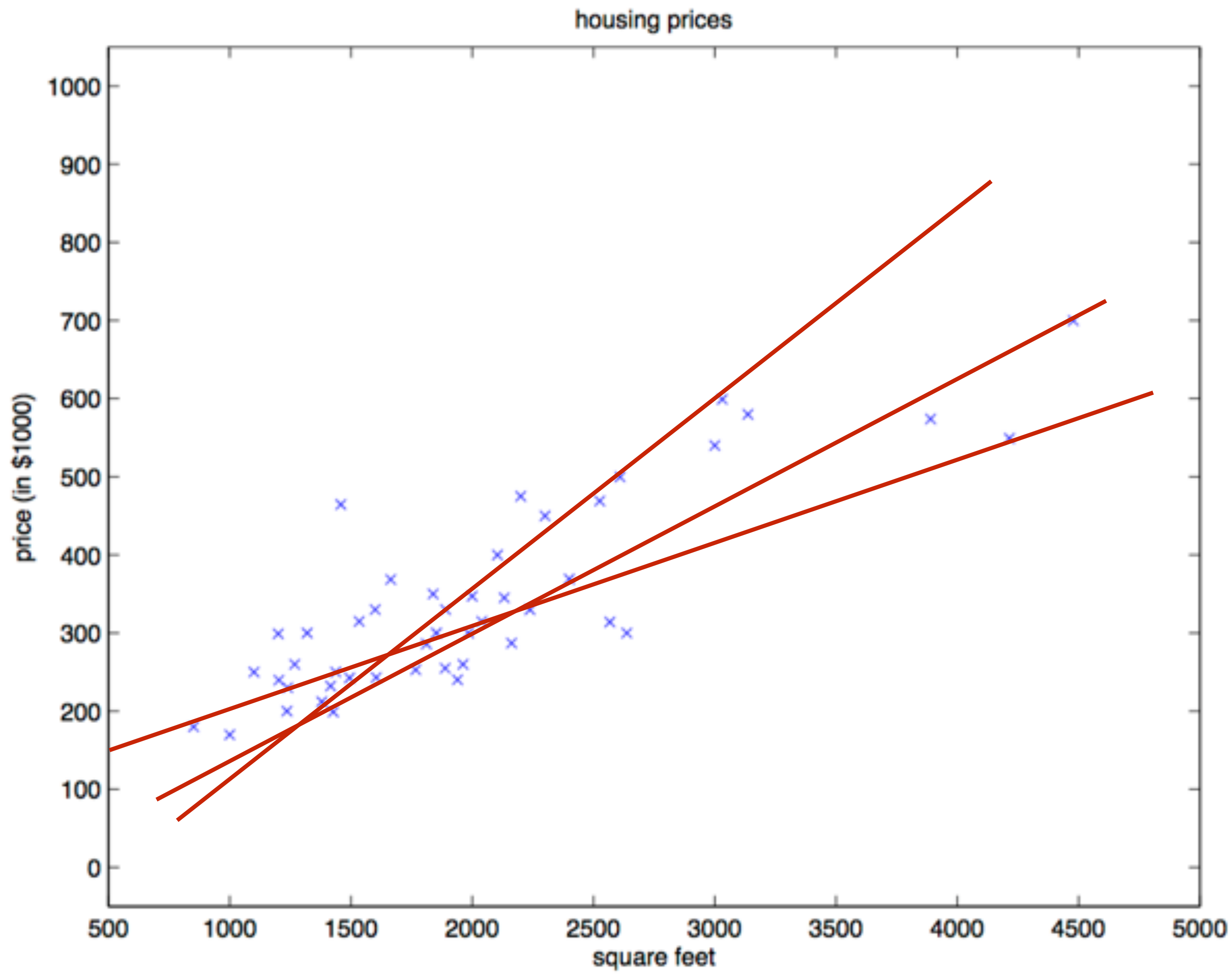
- Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function f

Output:

- Hypothesis $h \in H$ that best approximates target function f

Source: Tom Mitchell ML 10-701 Lecture Notes

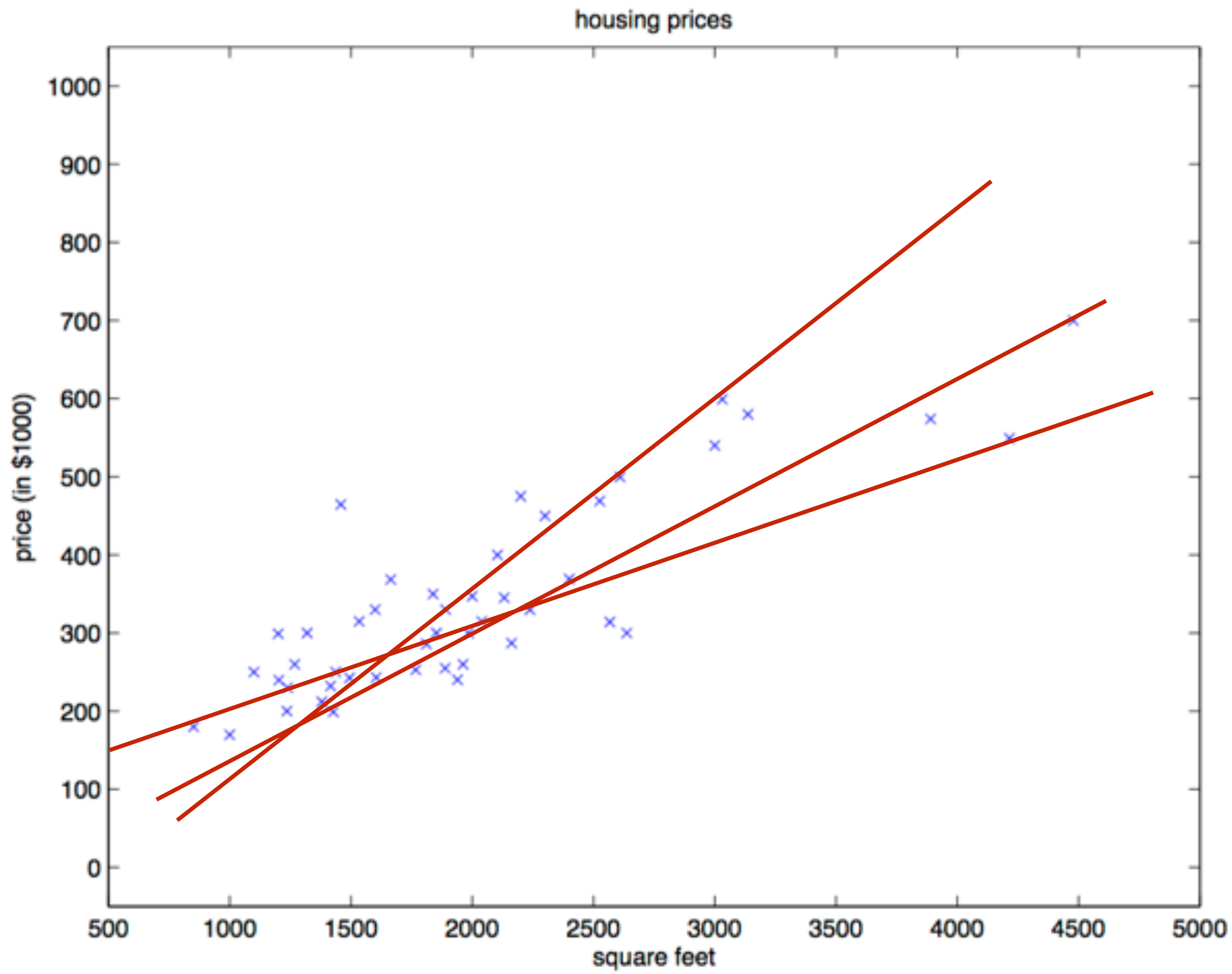
Learning Algorithms



How to choose a hypothesis?

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

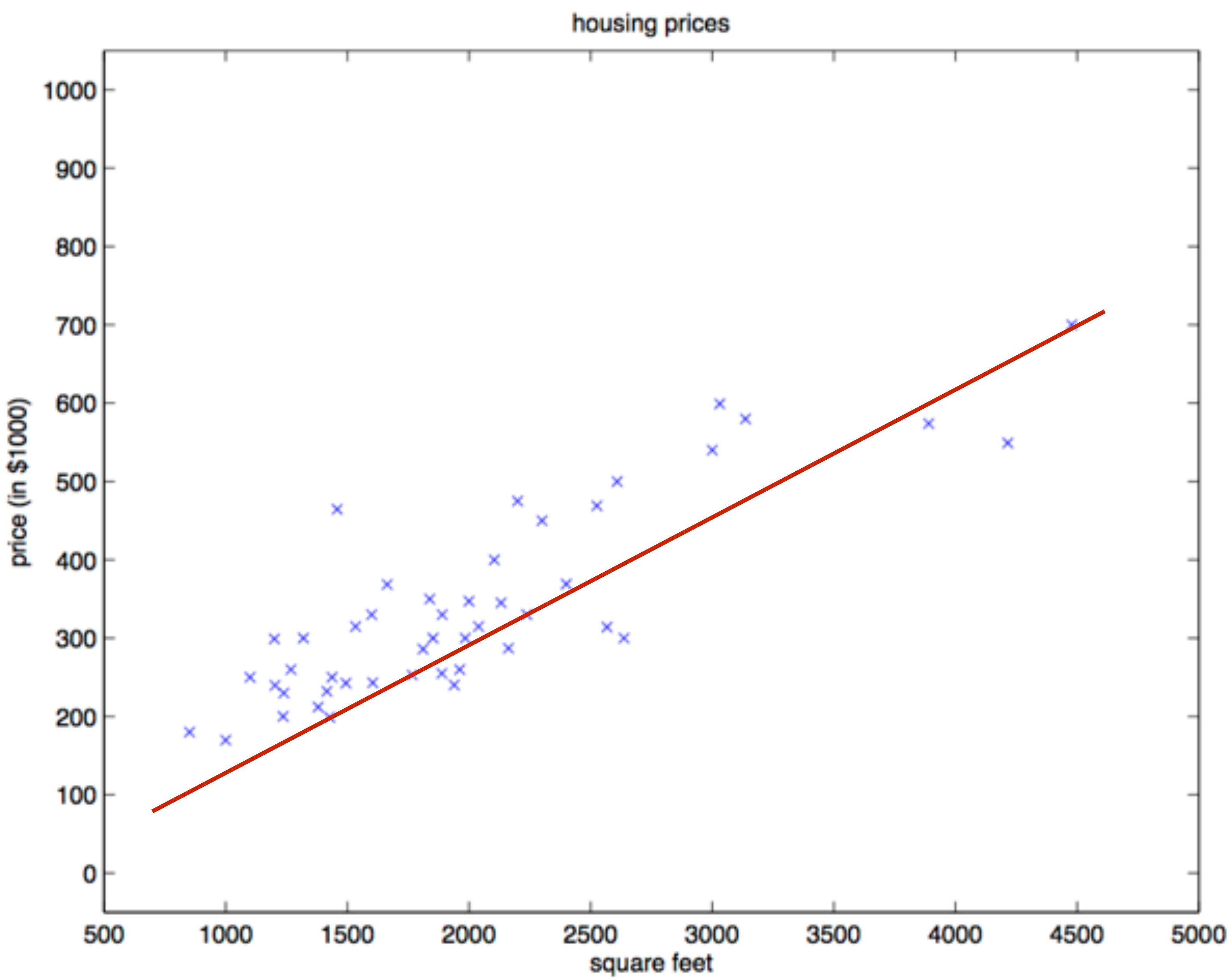
Learning Algorithms



Identify parameters that minimize error

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

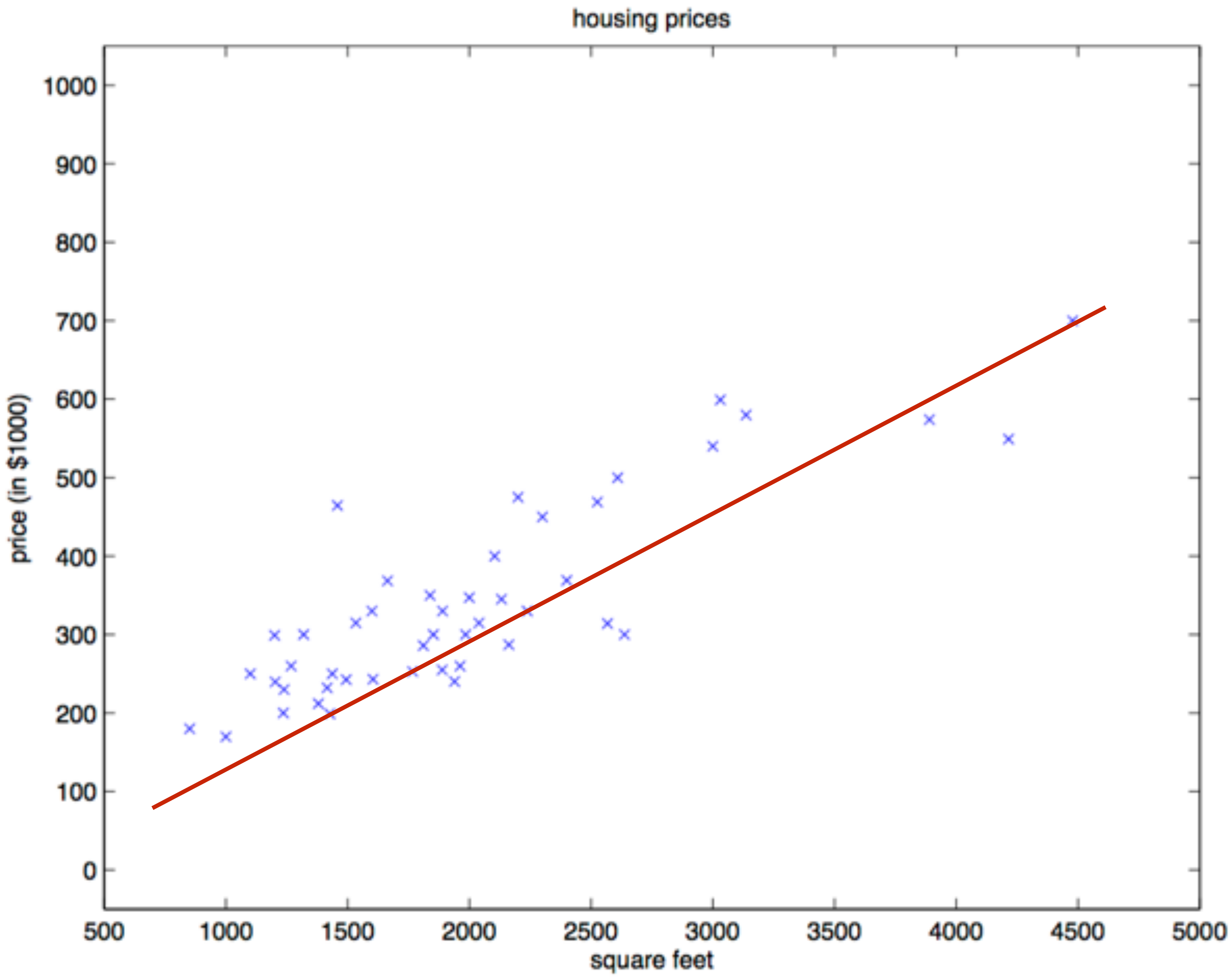
Linear Model



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

Linear Model

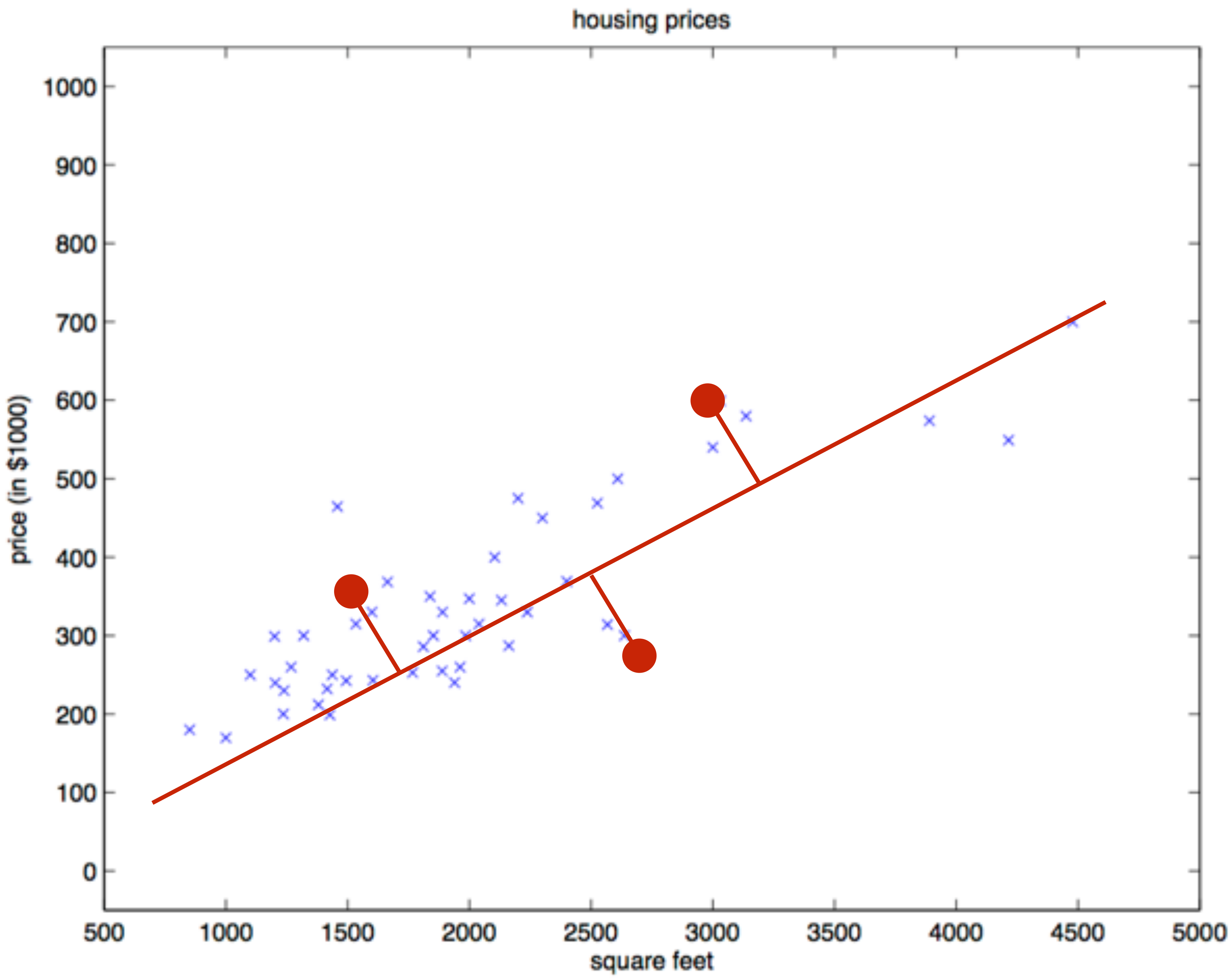


$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

parameters

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

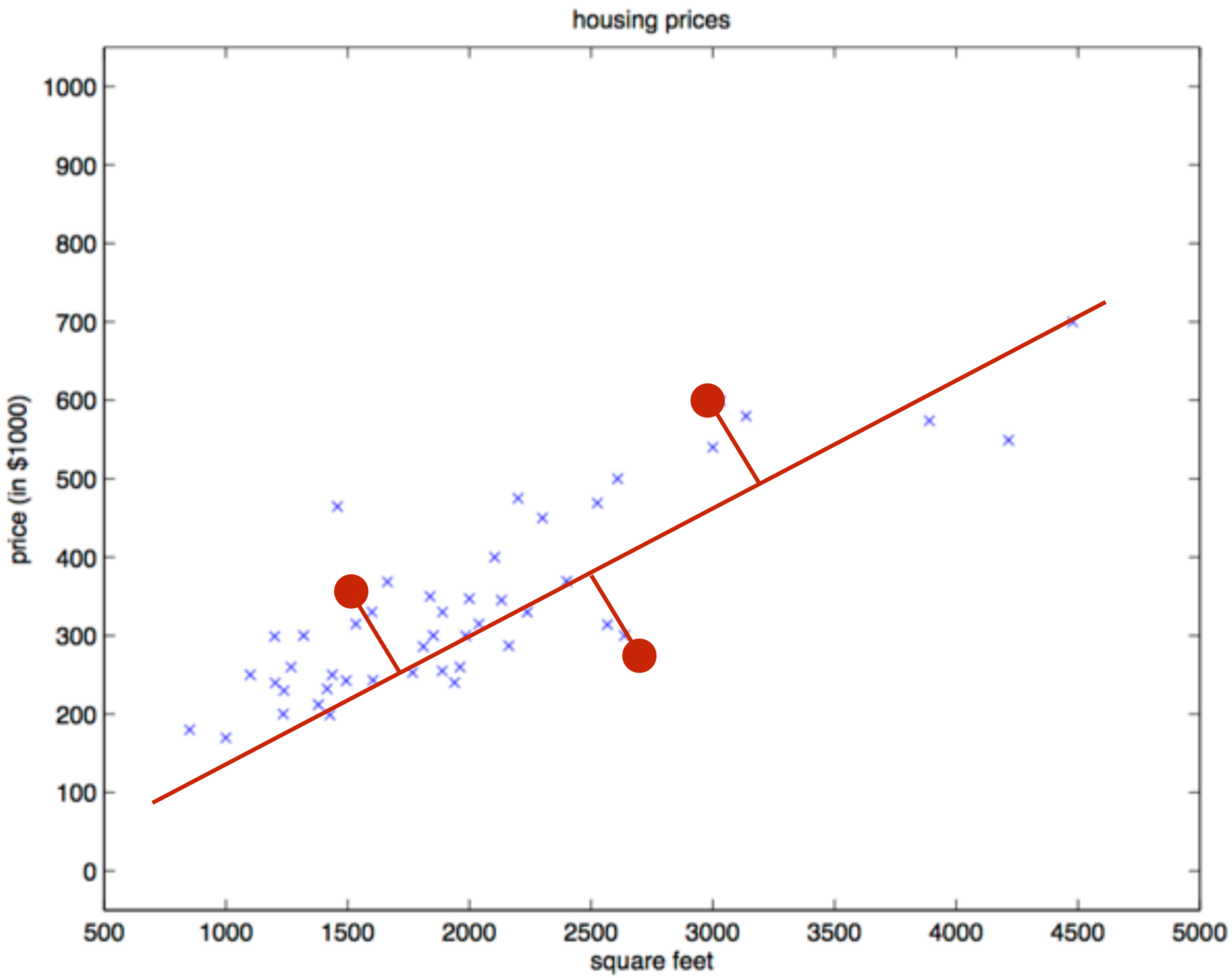
Linear Model



Error

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

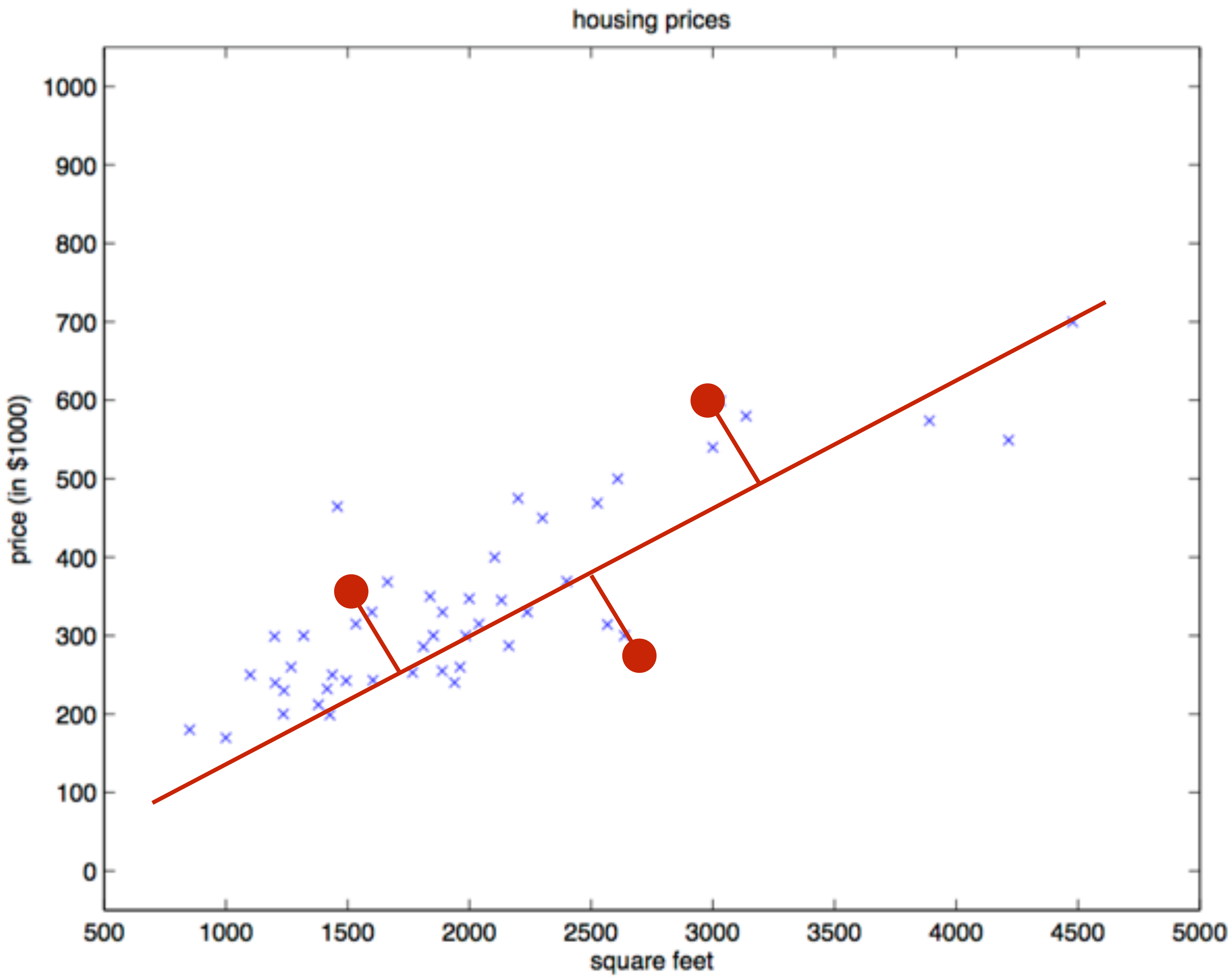
Least-Squares Cost Function



$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

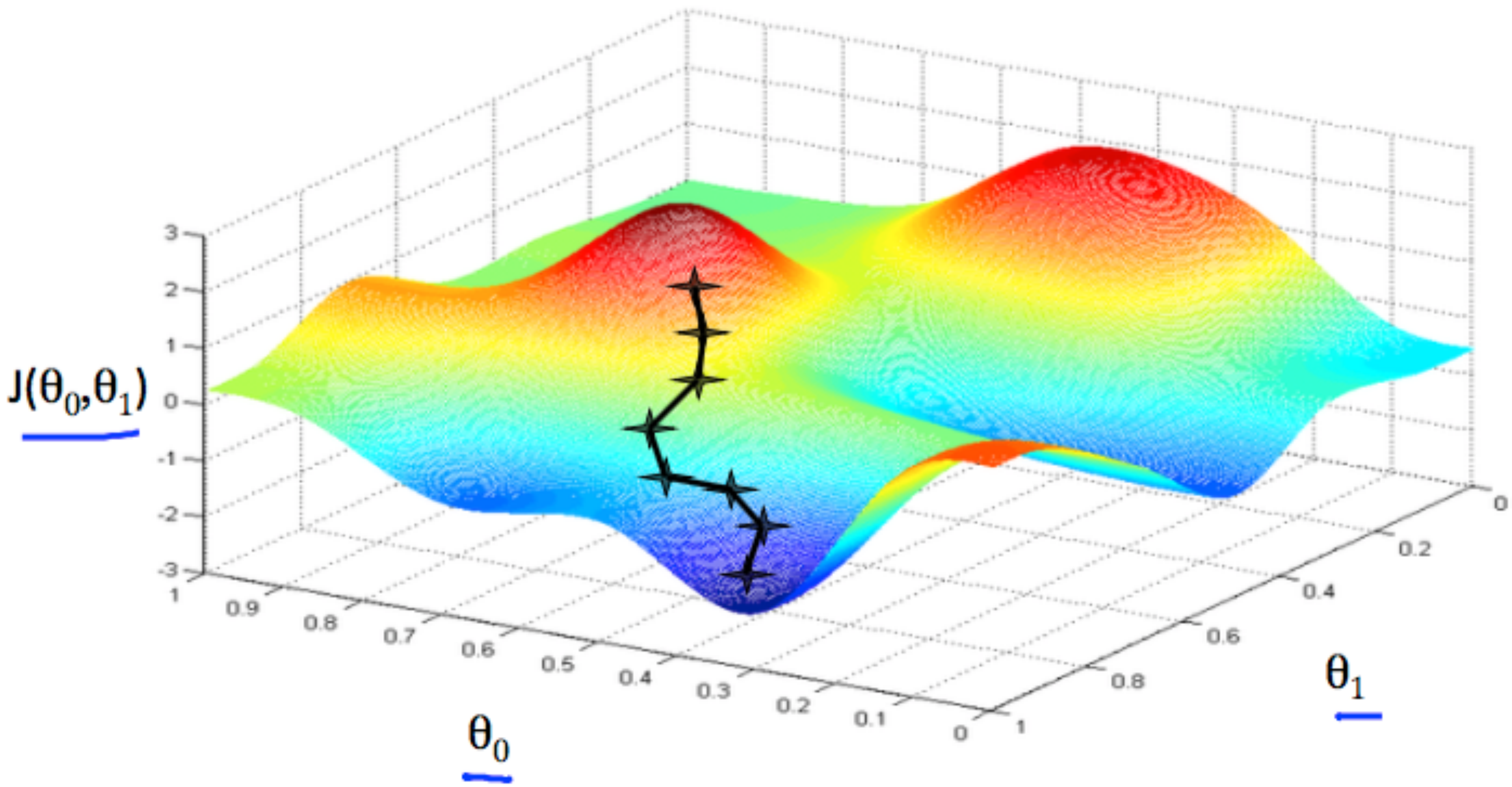
Least-Squares Cost Function



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

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

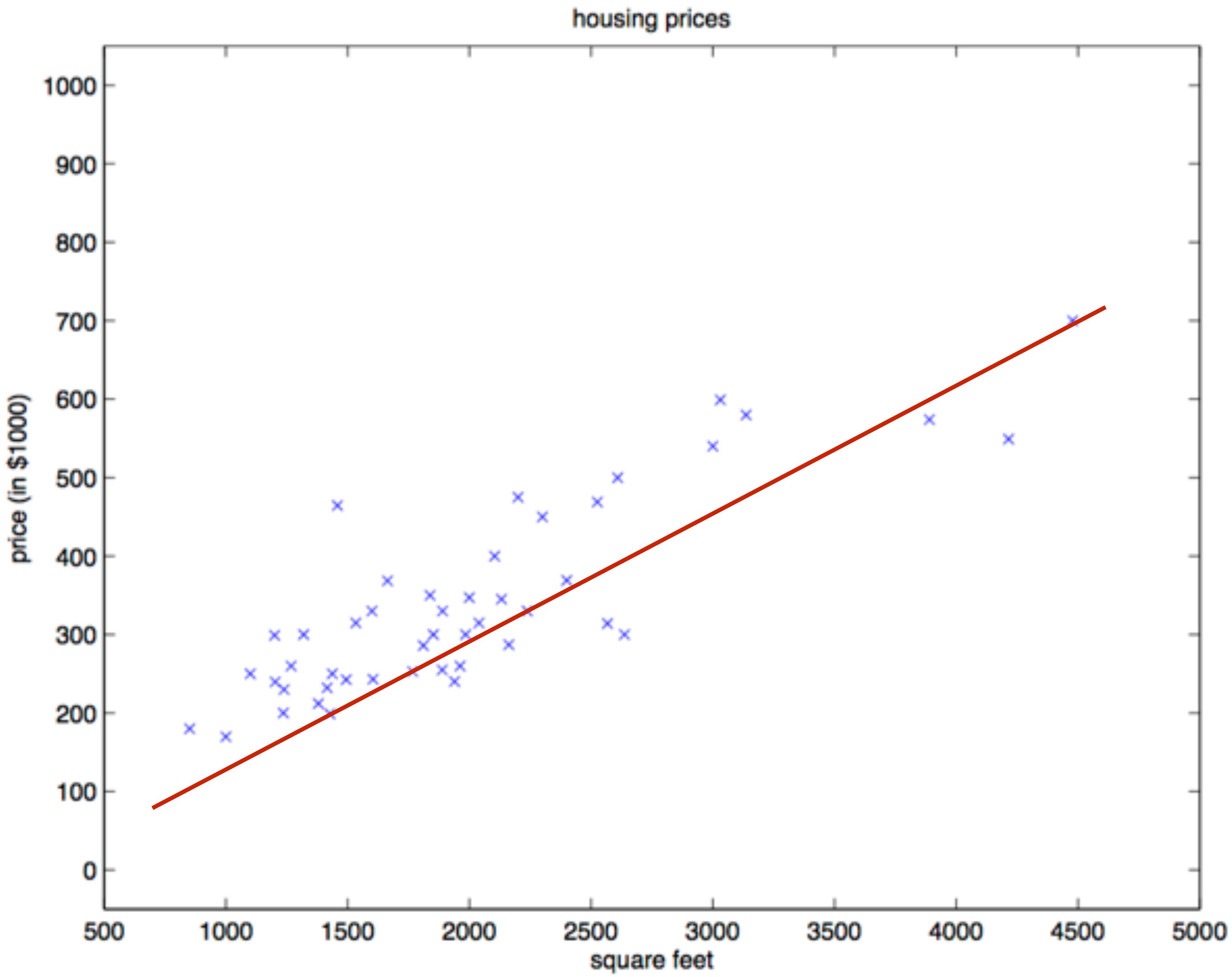
Least-Squares Cost Function



Gradient Descent

Source: Andrew Ng CS229 Lecture Notes

Linear Model



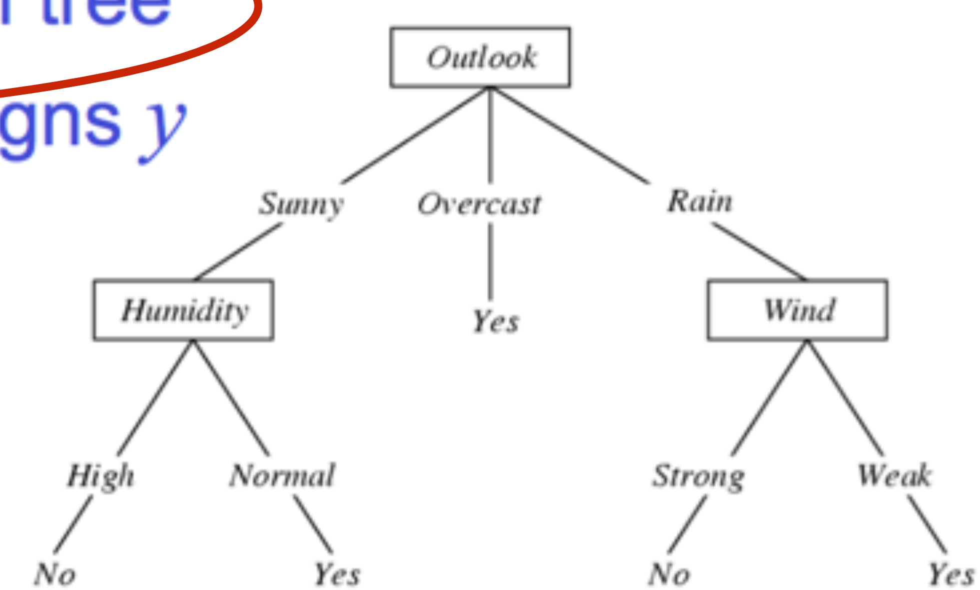
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

Decision Tree

Problem Setting:

- Set of possible instances X
  - each instance x in X is a feature vector
  - e.g., <*Humidity=low, Wind=weak, Outlook=rain, Temp=hot*>
- Unknown target function $f : X \rightarrow Y$
  - Y is discrete valued
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree
  - trees sorts x to leaf, which assigns y

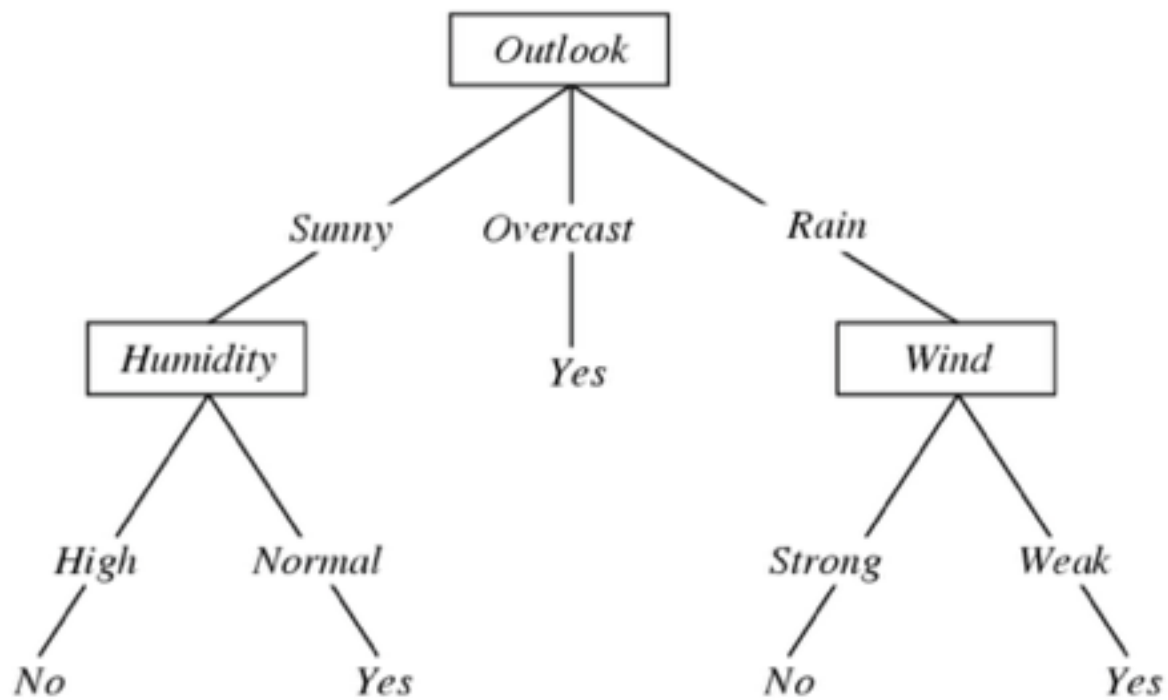


Source: (Based in part on) Tom Mitchell ML 10-701 Lecture Notes

Decision Tree

A Decision tree for

F: <Outlook, Humidity, Wind, Temp> → PlayTennis?



Each internal node: test one attribute X_i

Each branch from a node: selects one value for X_i

Each leaf node: predict Y (or $P(Y|X \in \text{leaf})$)

Decision Tree

Top-Down Induction of Decision Trees

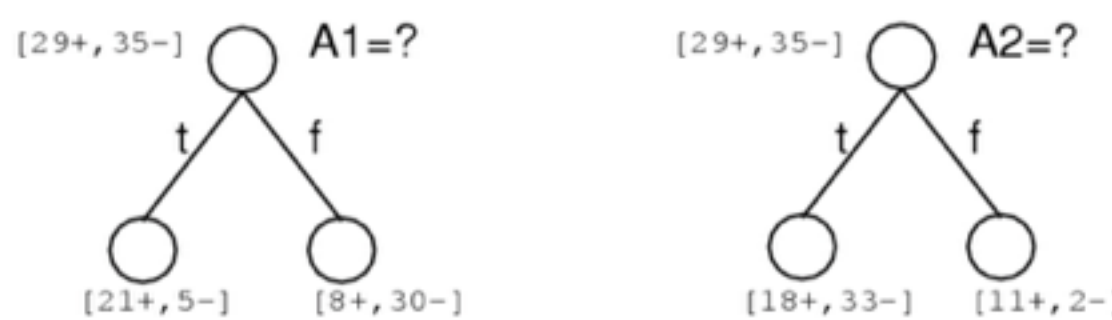
[ID3, C4.5, Quinlan]

node = Root

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



Source: (Based in part on) Tom Mitchell ML 10-701 Lecture Notes

Discriminative vs. Generative

Discriminative Models

Hypothesis defines a decision boundary between classes

e.g., Linear Models, Decision Tree

Generative Models

Build a (probabilistic) model for each class, and match against each model

e.g., Bayesian Networks, Naive Bayes

Generative Models

- Consider hypothesis space H

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$ = prior prob. of hypothesis $h \in H$
- $P(D)$ = prior prob. of training data D
- $P(h|D)$ = probability of h given D
- $P(D|h)$ = probability of D given h

Source: Sofus Macskassy CS567 Lecture Notes

Generative Models

Natural choice is most probable hypothesis given the training data, or *maximum a posteriori* hypothesis h_{MAP}:

$$\begin{aligned} h_{\text{MAP}} &= \text{argmax}_{h \in H} P(h|D) \\ &= \text{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \text{argmax}_{h \in H} P(D|h)P(h) \end{aligned}$$

P(*y*) P(*sun*| *y*) P(*cool*| *y*) P(*high*| *y*) P(*strong*| *y*) = .005

P(*n*) P(*sun*| *n*) P(*cool*| *n*) P(*high*| *n*) P(*strong*| *n*) = .021

- So, y_{NB} = *n*

Source: Sofus Macskassy CS567 Lecture Notes

Generative Models

Natural choice is most probable hypothesis given the training data, or *maximum a posteriori* hypothesis h_{MAP}:

$$\begin{aligned} h_{\text{MAP}} &= \text{argmax}_{h \in H} P(h|D) \\ &= \text{argmax}_{h \in H} \frac{P(D|h)P(h)}{\boxed{P(D)}} \\ &= \text{argmax}_{h \in H} P(D|h)P(h) \end{aligned}$$

?

P(*y*) P(*sun*| *y*) P(*cool*| *y*) P(*high*| *y*) P(*strong*| *y*) = .005

P(*n*) P(*sun*| *n*) P(*cool*| *n*) P(*high*| *n*) P(*strong*| *n*) = .021

- So, $y_{\text{NB}} = n$

Source: Sofus Macskassy CS567 Lecture Notes

Naive Bayes

f_1	f_2	f_3	c: yes I no
0.34	0.76	0.67	?
0.75	0.54	0.99	?
0.02	0.23	0.79	?
…	…	…	…

$p(\text{yes} \mid f_1,f_2,f_3)$

$p(\text{no} \mid f_1,f_2,f_3)$

argmax $p(c \mid f_1,f_2,f_3)$

$$p(\text{yes} \mid f_1,f_2,f_3) = \frac{p(f_1,f_2,f_3 \mid \text{yes})\ p(\text{yes})}{p(f_1,f_2,f_3)}$$

Bayes Rule

$$p(\text{no} \mid f_1,f_2,f_3) = \frac{p(f_1,f_2,f_3 \mid \text{no})\ p(\text{no})}{p(f_1,f_2,f_3)}$$

Naive Bayes

f_1	f_2	f_3	c: yes l no
0.34	0.76	0.67	?
0.75	0.54	0.99	?
0.02	0.23	0.79	?
…	…	…	…

$p(yes \mid f_1,f_2,f_3)$

$p(no \mid f_1,f_2,f_3)$

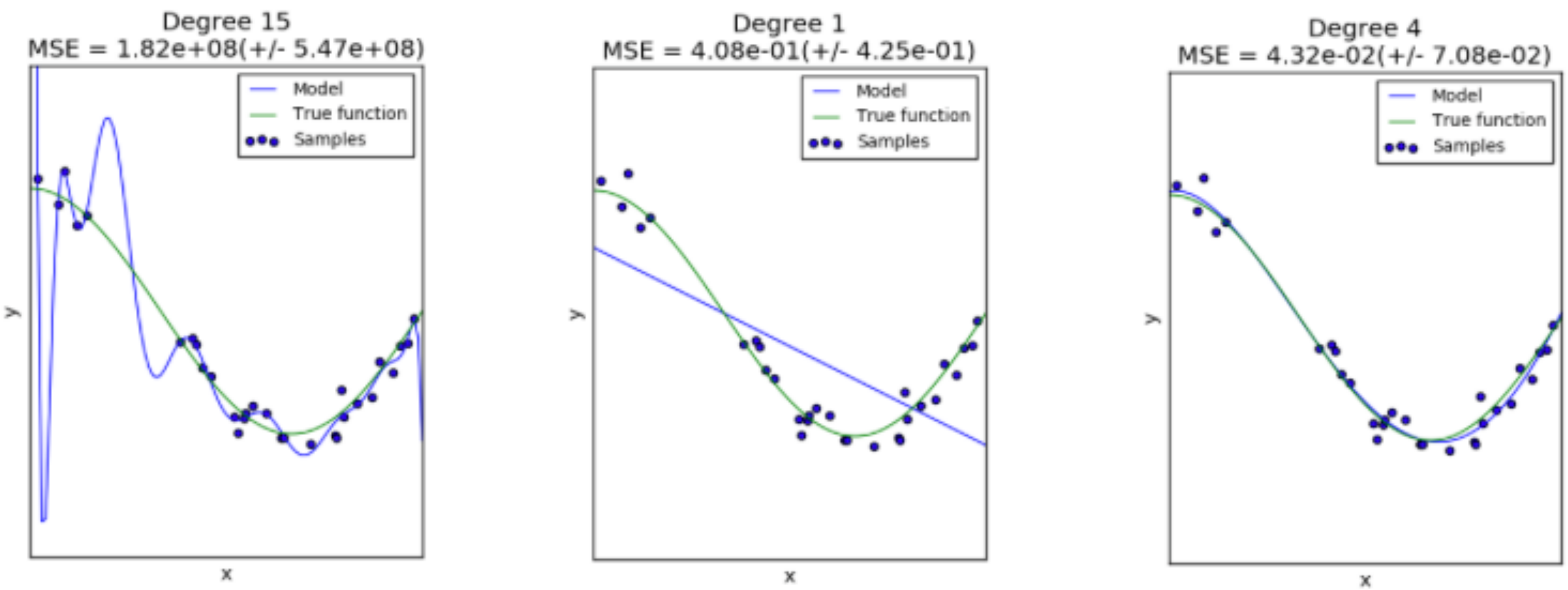
$\mathbf{argmax\ p(c \mid f_1,f_2,f_3)}$

$$p(yes \mid f_1,f_2,f_3) = \frac{p(f_1,f_2,f_3 \mid yes)\ p(yes)}{\cancel{p(f_1,f_2,f_3)}}$$

Bayes Rule

$$p(no \mid f_1,f_2,f_3) = \frac{p(f_1,f_2,f_3 \mid no)\ p(no)}{\cancel{p(f_1,f_2,f_3)}}$$

Underfitting & Overfitting



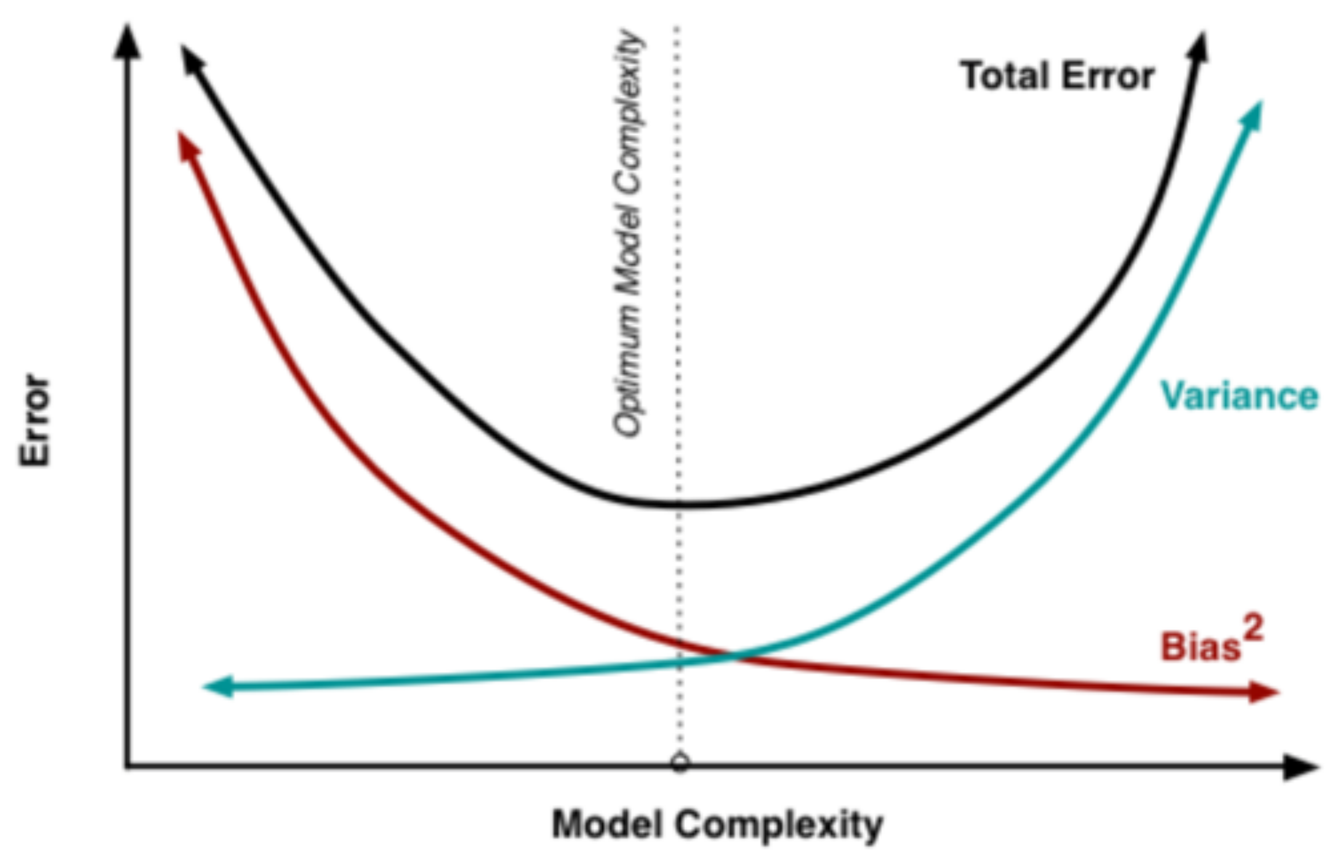
Overfitting	Underfitting	**Just Right**
Model too complex, learns the noise in the data	Model not complex enough to fit training data	

Bias and Variance

Concepts to help us understand the sources of model error

Bias: difference between expected prediction of the model and the value we are trying to predict

Variance: variability of model prediction for a given data point



Project

You will work on a semester-long project

Get you to develop practical activity recognition skills

Push the state-of-the-art in some aspect of sensing or recognition

Explore a problem in the context of a real-world application

Develop your research skills (e.g., reviewing prior work, etc)

Develop your communication skills

Satisfying to complete course with a tangible outcome

Project

Team Formation (Sept 20th)

3 or 4 students per team

Proposal (Sept 27th)

Progress Report (Oct 25th)

Final Report (Nov 29th)

Presentation (Nov 29th)

Project report should be of publishable quality

Use of traditional research methods

Live demos are welcome!

Project Ideas

Swimming style and form detection

Identify different types of cooking gestures

Sensor to detect dog activities

Identify when text and driving is happening

Hand washing detection with wrist sensors

Activity recognition with physiological signals

Detect stress from gestural data

Activity recognition models from media

Upcoming Class(es)

Next class: **Bring a computer to class**

Machine Learning Lab

Python + Scipy/Numpy + Scikit-Learn

Talk to me or TA in case if you have issues

Next week: Project Idea Pitches

Project Team Formation

Possible Guest Lecture

I will be out-of-town but reachable by email