Machine Learning & Artificial Intelligence

Instructor: Ziqi Wang and Tianwei Xing

In This Lecture

- **What is Machine Learning?**
- **Types of Learning**
- **Applications / Real-Life Examples**
- **Overview of Various Approaches**
- **Detailed View of Few Approaches**
- **Project**

In This Lecture

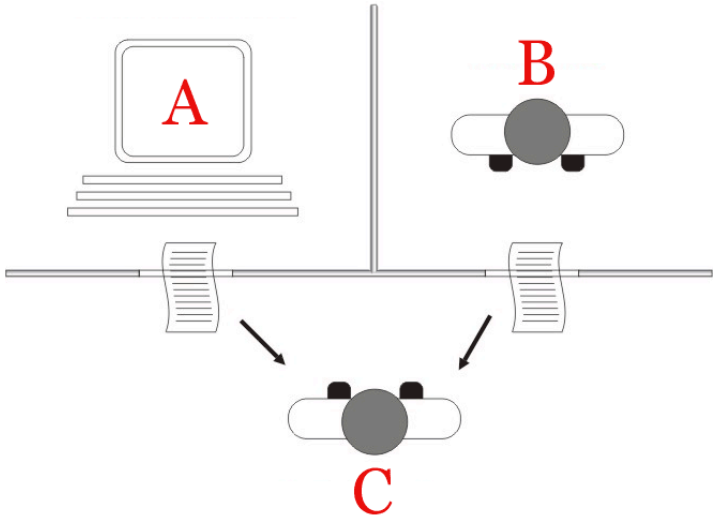
- **What is Machine Learning?**
- Types of Learning
- Applications / Real-Life Examples
- Overview of Various Approaches
- Detailed View of Few Approaches
- Project

Artificial Intelligence and Turing Test

- Turing proposed that a human evaluator **would judge natural language conversations between a human and a machine** designed to generate human-like responses.

- The evaluator would be aware that one of the two partners in conversation is a machine, and all participants would be separated from one another.

- If the evaluator cannot reliably tell the machine from the human, the machine is said to have passed the test.

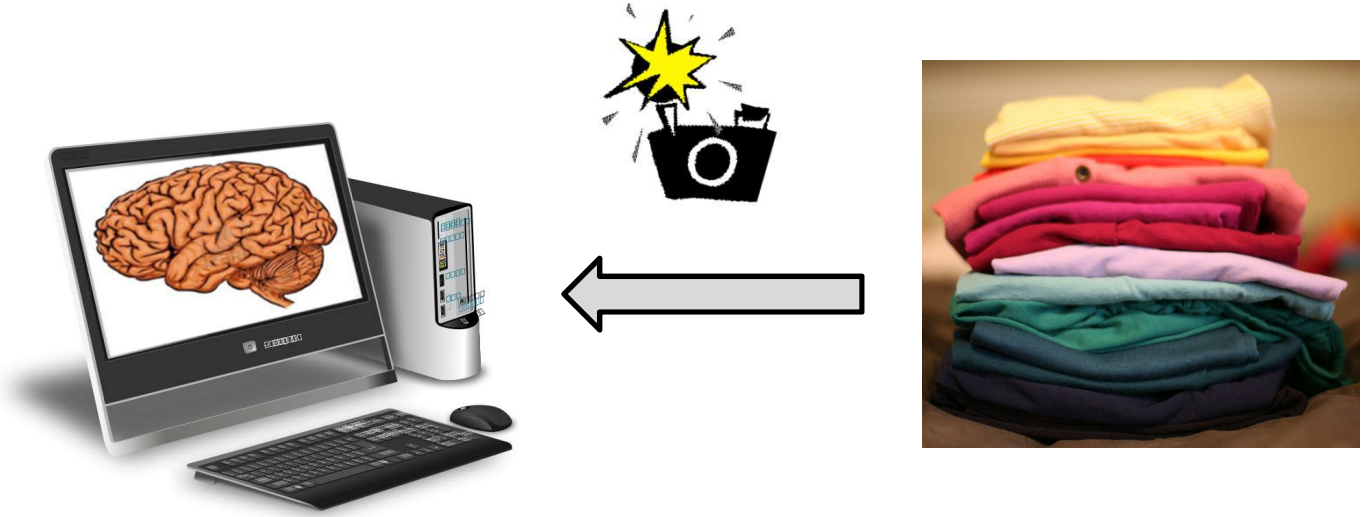


What is Machine Learning: An Example

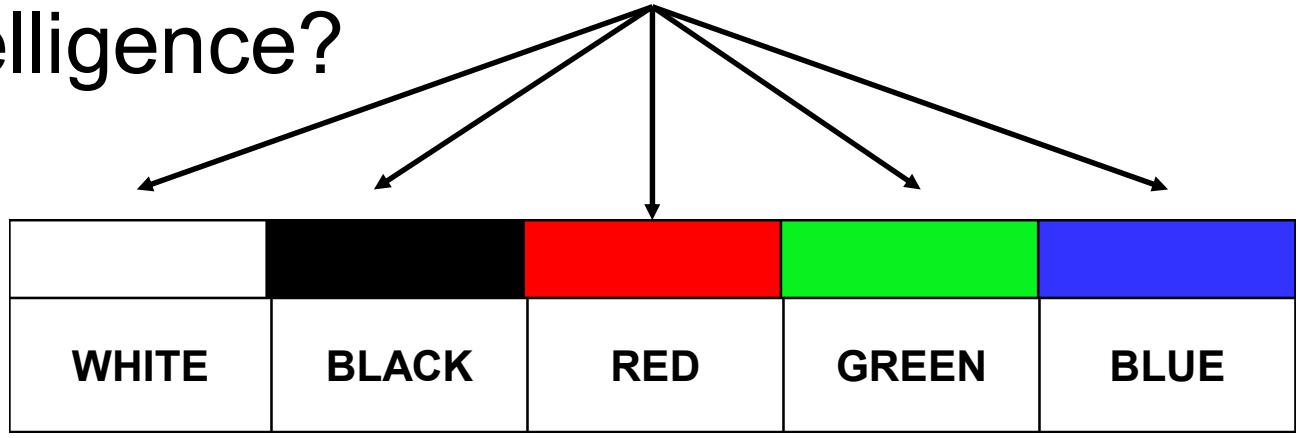
A Laundry company wants to separate washable clothes into five piles of separate color. It puts a **computer** in charge of separating the clothes into different piles.
How does the computer learn how to group the clothes?

WHITE	**BLACK**	**RED**	**GREEN**	**BLUE**

Laundry Sorting System



Intelligence?



Using simple if-then rules

- Can we just **hard-code** some simple rules in the computer**?**
- We use our own understanding to explicitly tell the computer what goes in the red pile, what goes in the blue pile and so on… ?
- If there is a **compactly representable** relationship between the input and the output, then we do not need learning ☺ .
- What if we have lot of rules, many unspoken…

1. *Little bit of color->can't go in white.*
2. *Some colors bleed more, some less.*
3. *Low intensity colors can be mixed together not high intensity no.*
4. *Decisions are cloth dependent…Towels can get lightly tainted but shirts no!*

Benefit of Using Machine Learning

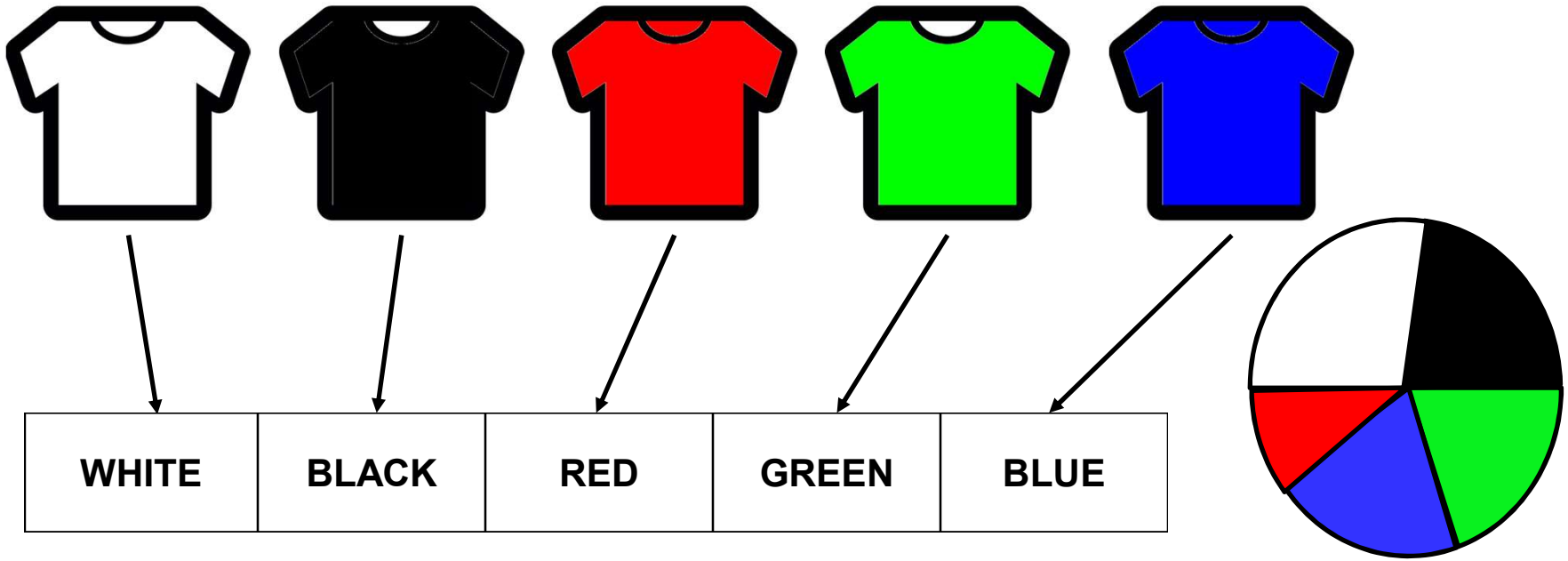
1. In many problems, the relationships are either just too many to write down as **if** *this* **then** *that* OR not even understood properly.

2. Machine learning saves a lot of our time and effort in hand crafting these rules by learning from given examples.

3. With **enough examples** computers can self create **arbitrarily complex** rules or **algorithms** and learn to see, walk, talk, read.

What is Training a Machine with Examples?

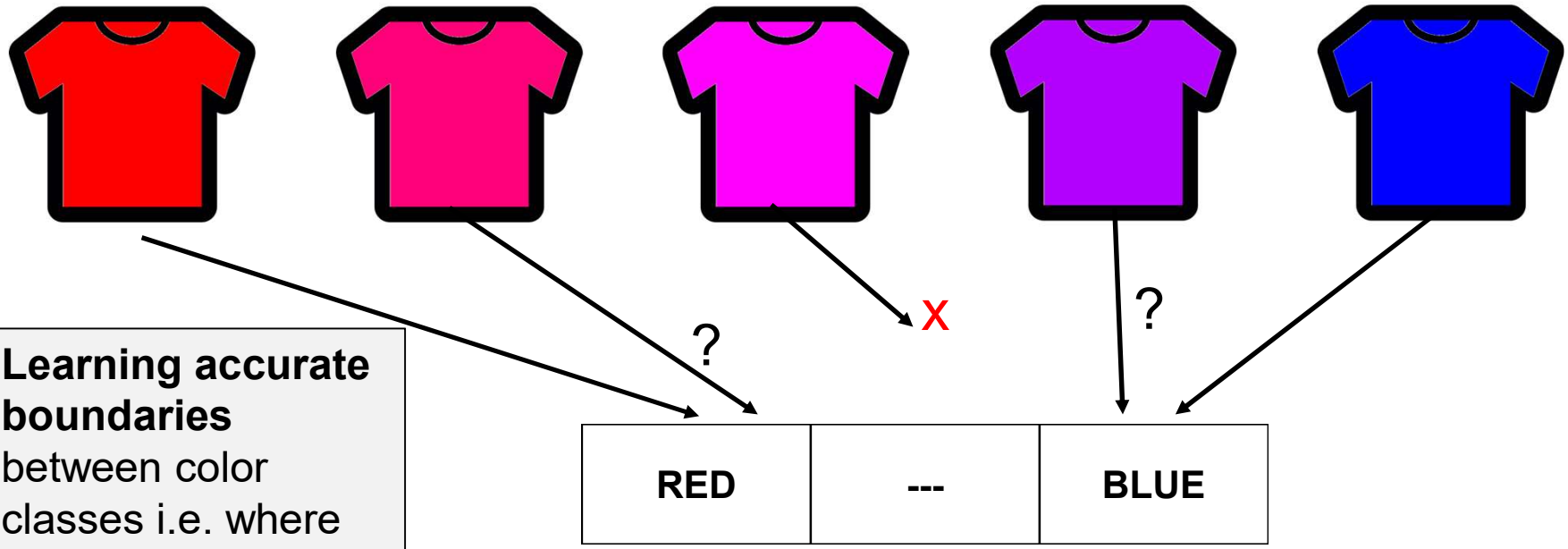
- Represent all the inputs as points in a coordinate (x,y,z,…) system. (Eg ?)
- Define a function representing a score for each class for each input. (Eg.?)
- Classify/assign class to input to the maximum scoring class.



Initial learned boundaries

What does a machine learn

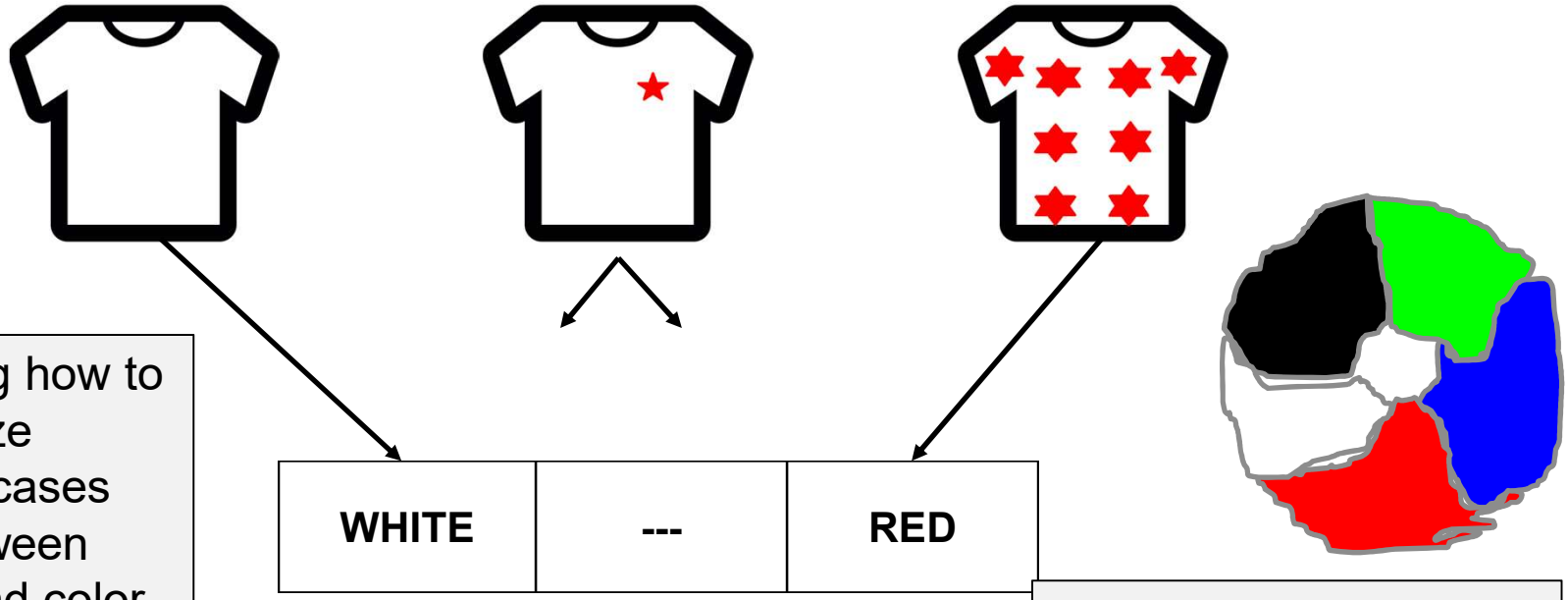
More training = better generalization .
Now the machine has learnt a basic rule (score based model) to classify into 5 classes.
Test if the machine classifies **new** (unseen) clothes correctly.



Learning accurate boundaries between color classes i.e. where red stops and blue starts.

What does a machine learn

The machine learns complex class rules through training.



Learning how to recognize special cases like between white and color.

Boundaries after many Training iterations.

Training and Testing Set

- Usually there is a fixed training set. But for our laundry sorting system, if we can manually indicate the correct responses, then we can increase the size of the training set. This usually helps improve performance.

- But when evaluating, the algorithm is **never** tested on the training set (to avoid bias). It is tested (once at the end) on **new** examples to estimate its performance.

Conclusions from our Laundry Sorting System

Our Laundry Sorting Algorithm:

- Makes a decision or **classification** on a new input**.** *(where to put the cloth)*
- Input data is represented in terms of its **features** *(ratio of each color in the cloth).*
- Wants to **minimize** (or maximize) an **objective** and we can **measure** its learning performance over a random sample of input *(total mismatched color in all washing machines on average).*
- The learning performance **improves** as the algorithm sees **more correctly and incorrectly classified examples** *(operator alerts the computer if he finds a cloth in the wrong pile).*

What is Machine Learning?

http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/

- Arthur Samuel (IBM, Stanford Professor) (pioneer and coined the term Machine Learning in 1959)- "*Field of study that gives computers the ability to learn without being explicitly programmed*"

- Tom Mitchell (CMU Professor)- "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.*"

- **Main idea**: software/algorithms that can improve on a task based on data and previous experience.

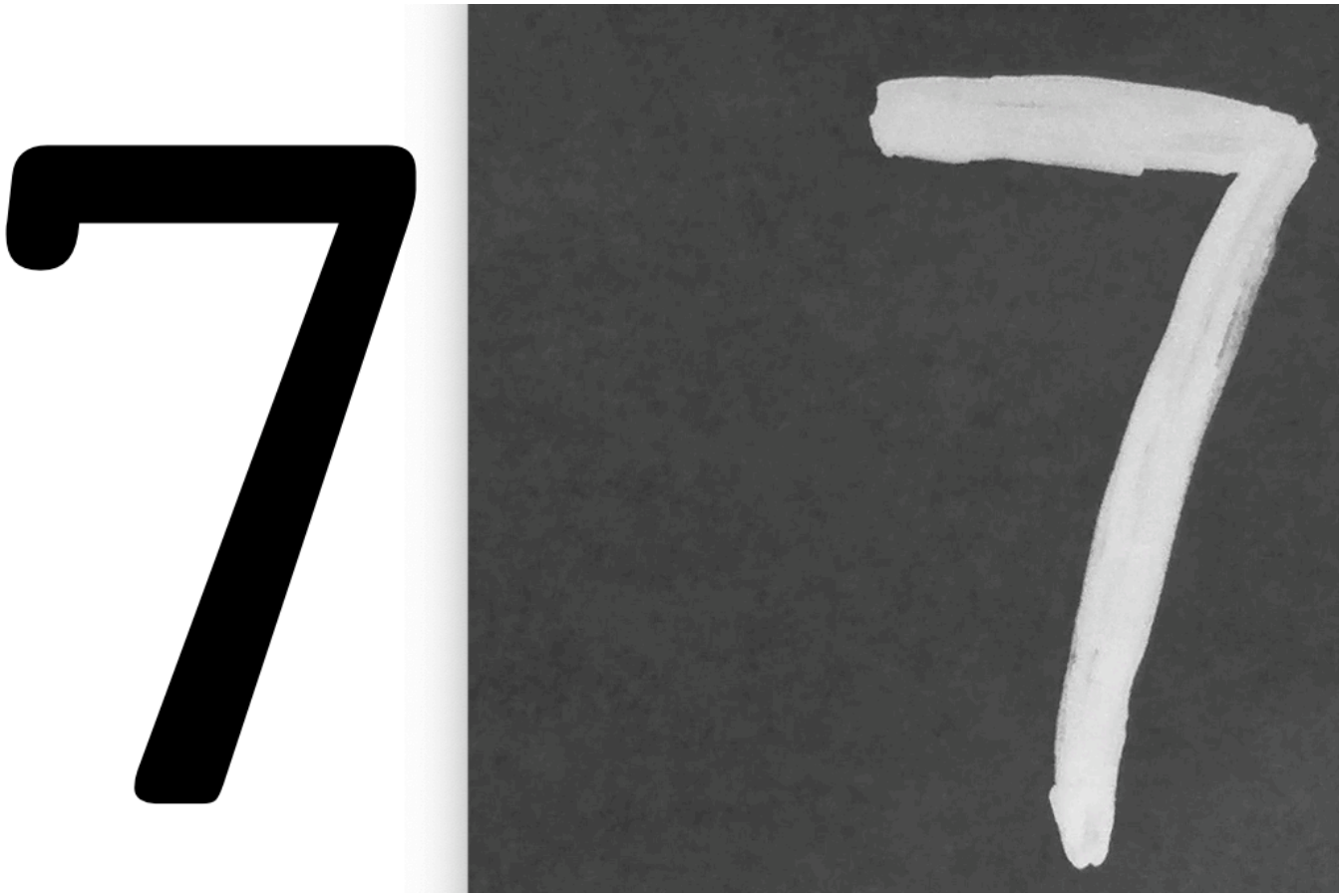
Pop up Question

- Say your Gmail is learning to classify spam better as you mark a new email as Spam. What is Task T, Experience E, and Performance P?

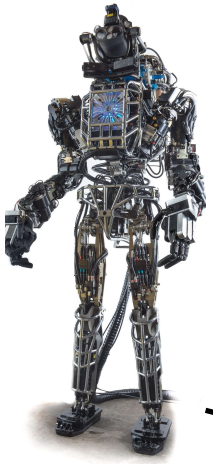
- Classifying a new email as spam or not spam.
- Watching you classify an email as spam.
- The number of emails correctly classified as spam and not spam.
- None of the above- this is not a machine learning problem.

[Ref: Andrew Ng's ML Course: Coursera]

What is Machine Learning: A Live Demo



Applications of Machine Learning



Self learning robots

Optical character recognition

Speech recognition
Semantic analysis

Drone navigation

Facial recognition

amazon

NETFLIX

SOUNDCLOUD

Recommendation engines

Machine Learning Applications

Filtering algorithms/ news feeds

Personal assistants: Google Now, Microsoft Cortana, Apple Siri, etc.

Google Ads

Advertising and business intelligence

Autonomous ("self-driving") vehicles

Past, Present and Future of Machine Learning

- Machine Learning is a sub-branch of Artificial Intelligence that focuses on how machines can learn rather than on end to end intelligent systems.
- Since the invention of computers, scientists wanted computers to be able to imitate humans in skills, such as recognize objects, understand speech, think and give cohesive answers, play games like chess etc.

- In 1952, Arthur Samuel wrote the first computer program that could play checkers and improve with experience.
- In 1957, inspired the by the human brain, the first Artificial Neural Network called the Perceptron was invented by Frank Rosenblatt. It could do binary classification.
- Research stagnated in the 70s because the neural networks were not thought to be powerful enough.
- In 80s it was rediscovered how to train neural networks using backpropagation.
- With better computers and much more data, Machine Learning progress leapfrogged.

Past, Present and Future of Machine Learning

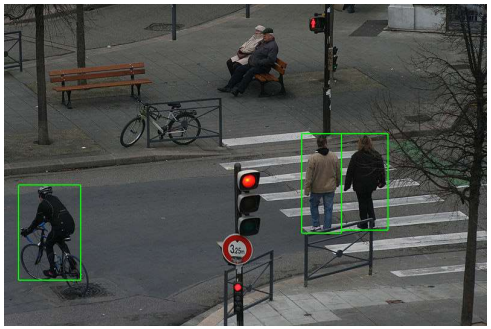
- Numerous learning algorithms have been invented in the last 30 years that are more powerful than the previous in learning complex patterns in data.

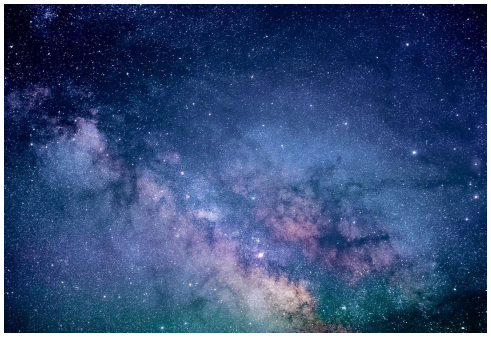
- Some of the recent algorithms that have been invented and are being used are
  - Support Vector Machines
  - Decision Trees and Random Forests
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Deep Neural Networks

Past, Present and Future of Machine Learning

These have allowed these algorithms to do things like

- Detect pedestrians and cyclists from video in real time to avoid crashes.
- Detect distant galaxies and new space objects from telescope images.
- Caption Images and Videos automatically.
- Read out the nearby world through glasses for the blind [link].
- Play and win games like chess, go, Jeopardy against the best players [link].





Future Possibilities of Machine Learning

Can you suggest some practical uses of Machine Learning and Artificial Intelligence?

Healthcare/Biology:
- Helping cure diseases like cancer by massive data analysis and modeling.
- Personalized medicine based on individual genome.
- Understanding how billions of neurons in human brain communicate.

Agriculture:
- Reducing pesticide use by only giving sufficient enough to kill the weeds.

Climate:
- Better weather and storm prediction.

Space:
- Self driving airplanes
- Self driving extra-Solar system rovers

Need of Some Caution

1. Machine Learning skills is more than just programming. It uses knowledge from fields like probability, statistics, linear algebra, calculus, physics, optimization. These skills become more important the deeper you go into the field so please learn them.

2. Use the simplest solution that gets your job done… e.g. Don't use Machine Learning to add two numbers ☺ . Use a calculator.

3. Use your skills wisely and for good. Remain ethical.

4. Keep the privacy of people's data in mind when using it. Ask permission first and remember that the results may affect them (in things like ad targeting, military surveillance, fake news etc..). See 4.

In This Lecture

- What is Machine Learning?
- **Types of Learning**
- Applications / Real-Life Examples
- Overview of Various Approaches
- Detailed View of a Few Approaches
- Project

Classification of Machine Learning

- Supervised Learning
  - Example: Classification – KNN
  - Example: Regression – Linear Regression
- Unsupervised Learning
  - Example: Clustering – K-Means
- Reinforcement Learning

Types of Learning

1. **Supervised Learning** (task driven)
   - Training data (example) set is well-labeled with the desired output class.

Examples

- **Classification**
  - Email spam detector
  - Facial recognition / thumbprint recognition
  - Speech recognition (Siri, Alexa, etc
    - *How Siri learns a new language* https://www.reuters.com/article/us-apple-siri-idUSKBN16G0H3
  - Image classification (MNIST/CIFAR/ImageNet)
- **Regression**
  - Housing price estimate (Zillow)
  - Stock market
- **Medical Diagnosis**

Elements of a ML problem: notation

- We need some notation to discuss ML problems.

- Start with inputs: we refer to inputs by x_i , $i = 1, 2, \ldots, N$
  - x can be just a number, but there's lots of other options:
    - x can be a vector (a list of numbers), OR a matrix OR an image, a graph, a string OR a combination of the above

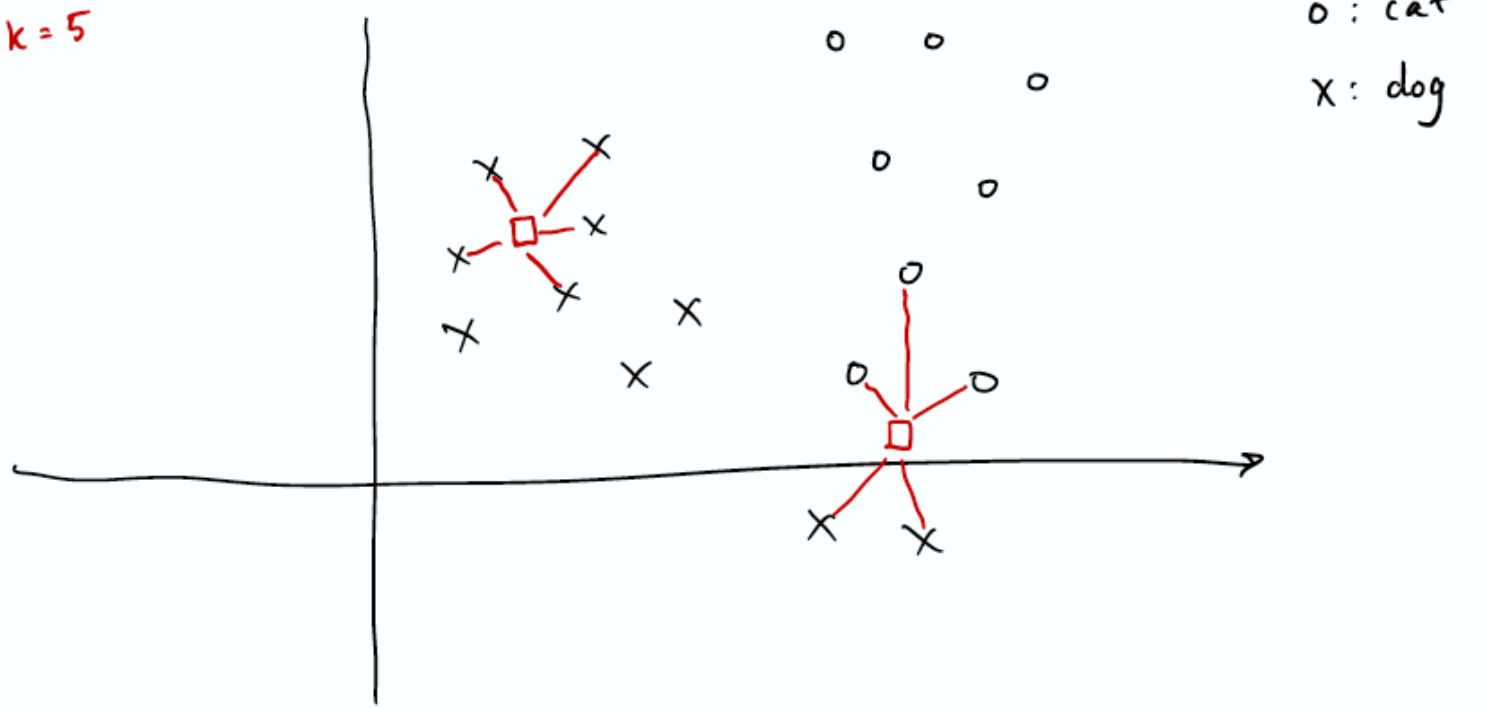
  - We are given n inputs $x_1, x_2, \ldots, x_N$ and the corresponding outputs $y_1, y_2, \ldots, y_N$. This is the **training set**.

  - Our goal is to use the training set to produce a function f(x) that can give an estimate $\hat{y}$ for an unseen x. The hat means we're estimating y.
  - If y is like price of an item (real, continuous) this is a regression or prediction problem. If y is a number (between 0 to K) this is a classification problem. The job of ML is to learn the parameters of the function f(x).

K-Nearest Neighbors(KNN)

k-nearest neighbors

Intuitively, k-nearest neighbors says to find the k closest points (or nearest neighbors) in the training set, according to an appropriate metric. Each of its k nearest neighbors then vote according to what class it is in, and $\mathbf{x}^{\text{new}}$ is assigned to be the class with the most votes.

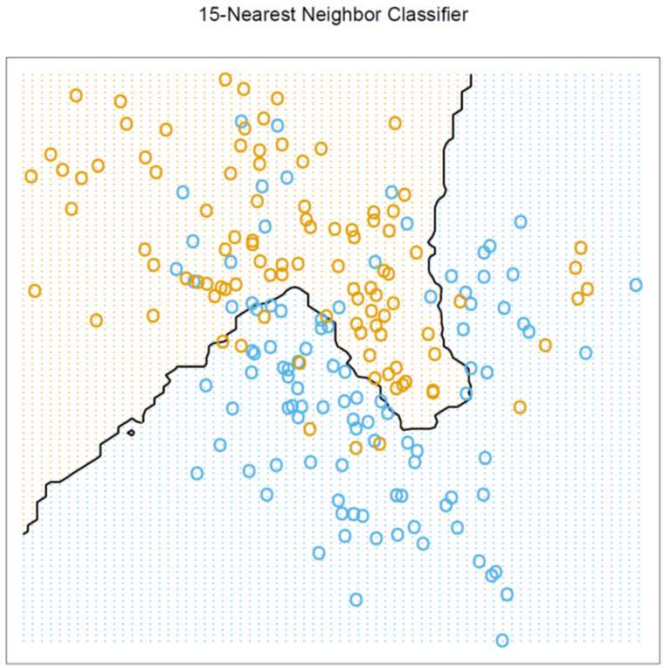


Nearest neighbor methods for Classification

- Let's go back to our orange/blue classification.
- What if we drop the whole "linear" condition for our model?
- We can try to predict class of a point based on the class of points that lie "near" it.
- This technique is called the **nearest neighbors.**
- Let's say for a new point P, we pick the k nearest neighbors, and take the majority vote for the class label.
- If majority neighbors belong to Class 0, then a new point is classified to Class 0, and vice-versa.

K-Nearest Neighbors

- Example: For every bit of area in the square, we evaluate what is the class of majority of my k=15 neighbors?

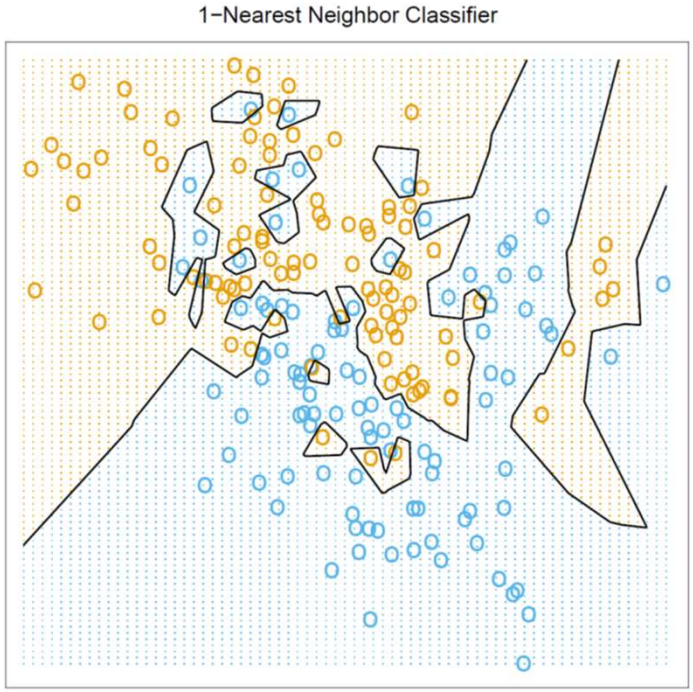


- Decision boundary is not a nice line anymore, it's a squiggly border. Expected?

- How many points from the training set are misclassified compared to linear prediction?

K-Nearest Neighbors

- Let's say we look at only one closest neighbor for decision.
- What does this mean? Well, each point is the same color as the nearest training point.



- How much misclassification of training data is there?
- Will this scheme cause a problem if used?

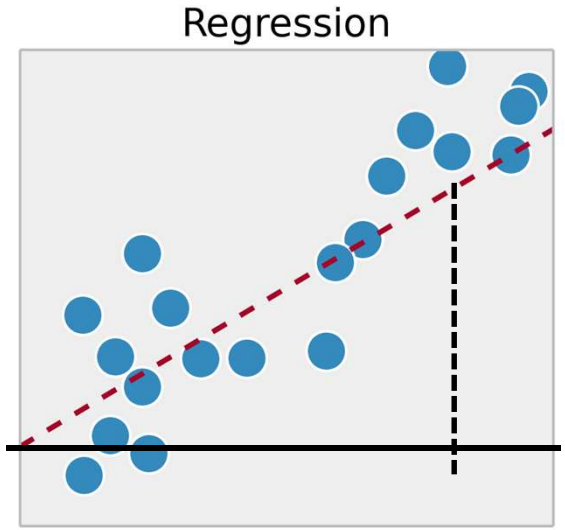
Classification of Machine Learning

- Supervised Learning
  - Example: Classification – KNN
  - Example: Regression – Linear Regression
- Unsupervised Learning
  - Example: Clustering – K-Means
- Reinforcement Learning

Linear Regression

- We are given some data points (x_1, y_1), (x_2, y_2), ..., (x_N, y_N) and given a new point x_{N+1} find what is y_{N+1}?

x	X1	x2	x3	x4
X(1)	1	2	3	0.5
Y	1	2.5	2.5	?

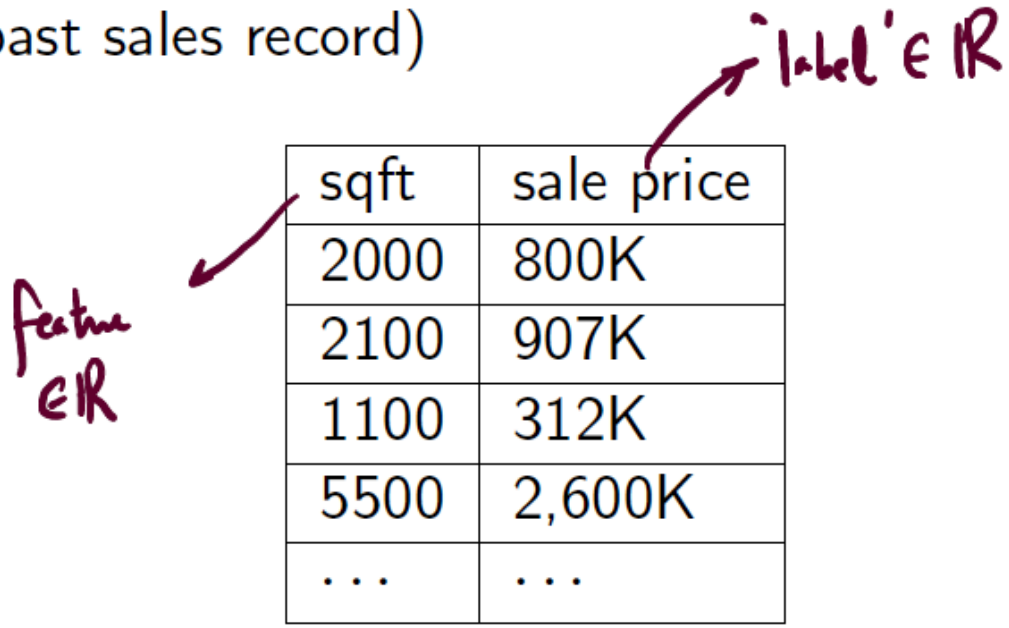


Linear Regression

- For example the x values may be the number of bedrooms in a house and y the price of the home in dollars.
- Such a regression is very helpful in predicting stock prices, predicting drone trajectories, autonomous driving, predicting effect of monsoon on the rice production this year…
- Algorithms like Minimum Residual Sum of Squares where we minimize the sum of square of error from a **linear** prediction.
- Linear prediction: Given features like # of bedrooms, area of house, average price in locality, we predict
- Price of home = **a**(# of bedrooms) + **b**(area) + **c**(avg price in locality)

Linear Regression

training data (past sales record)



sqft	sale price
2000	800K
2100	907K
1100	312K
5500	2,600K
...	...

Linear Regression

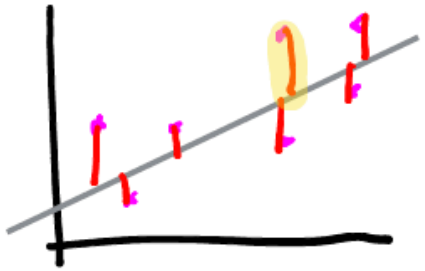
How to measure errors?

0-1 loss

- The classification error (*hit* or *miss*) is not appropriate for continuous outcomes.
- How should we evaluate quality of a prediction?

l_1-norm → *absolute* difference: | prediction - sale price|

l_2-norm → *squared* difference: (prediction - sale price)^2

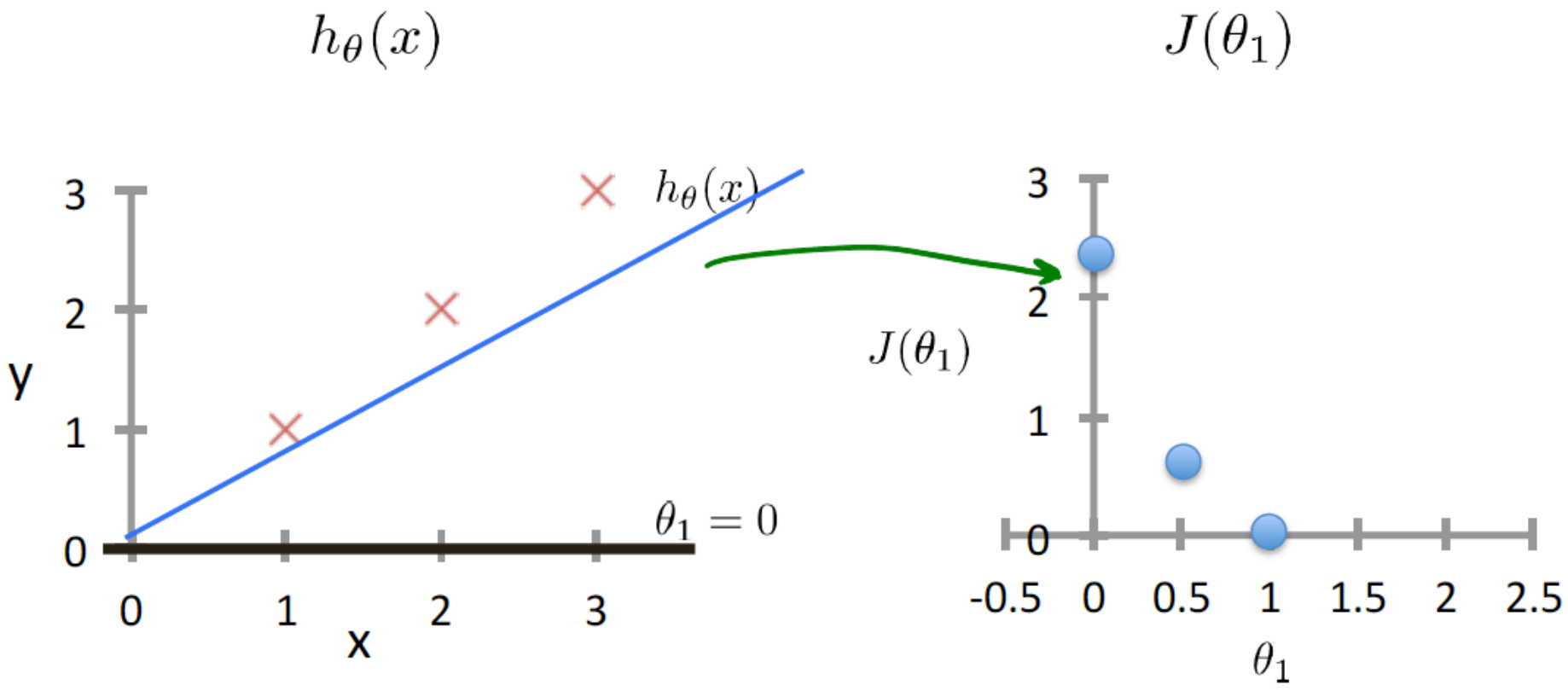


sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	90^2
2100	907K	800K	107K	107^2
1100	312K	350K	-38K	38^2
5500	2,600K	2,600K	0	0
...	...			

Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

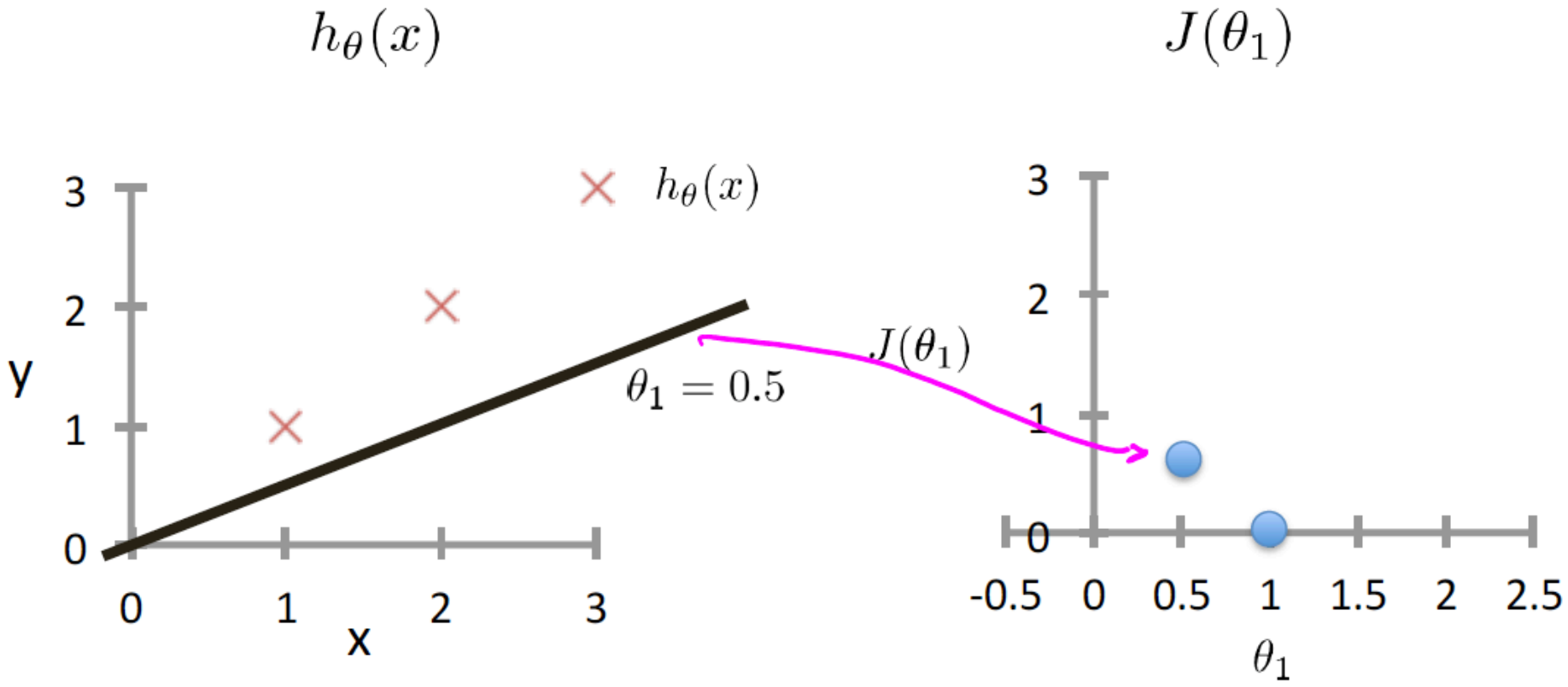
$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$



Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

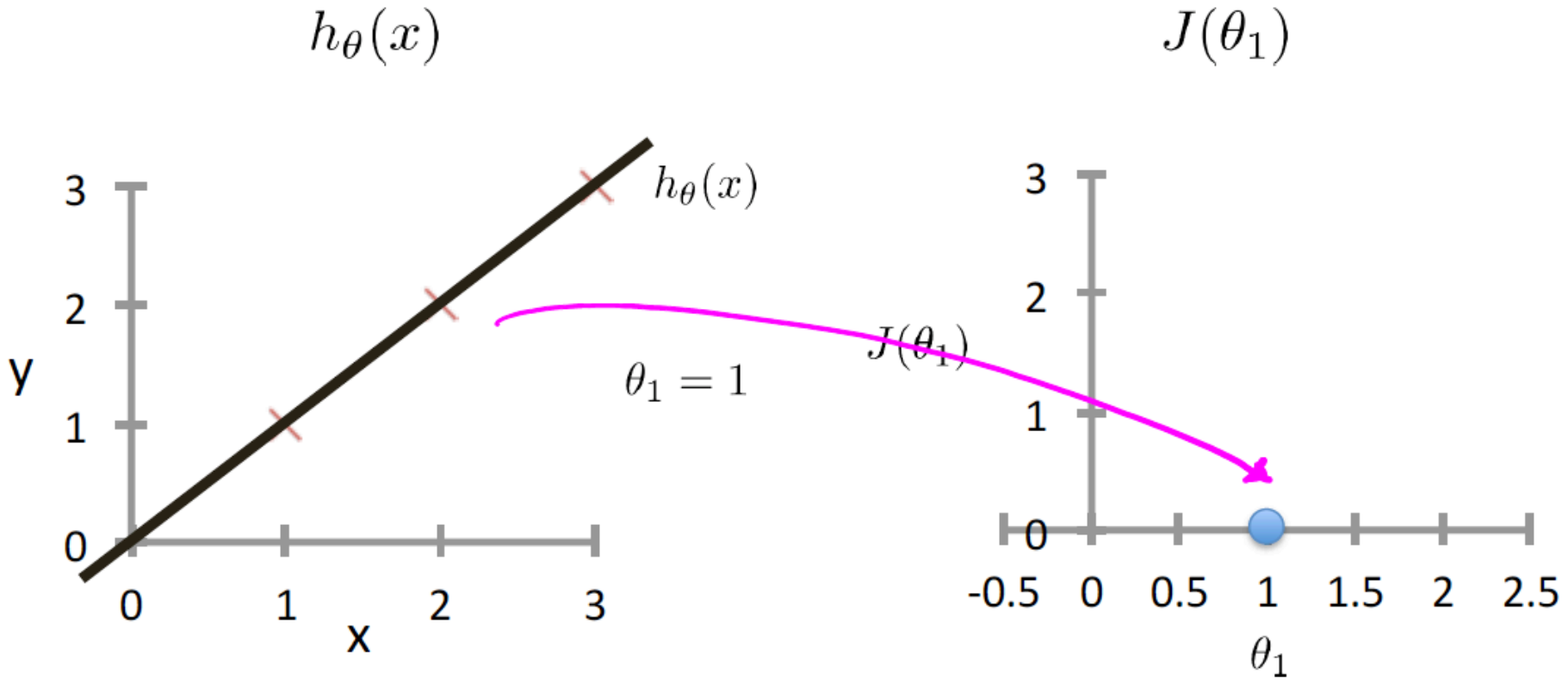
$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$



Intuiton behind cost function (residual sum of squares RSS)

Assume $x \in \mathbb{R}$, $\theta_0 = 0$.

$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \theta_1 x$



Basic Method: Gradient Descent

current position

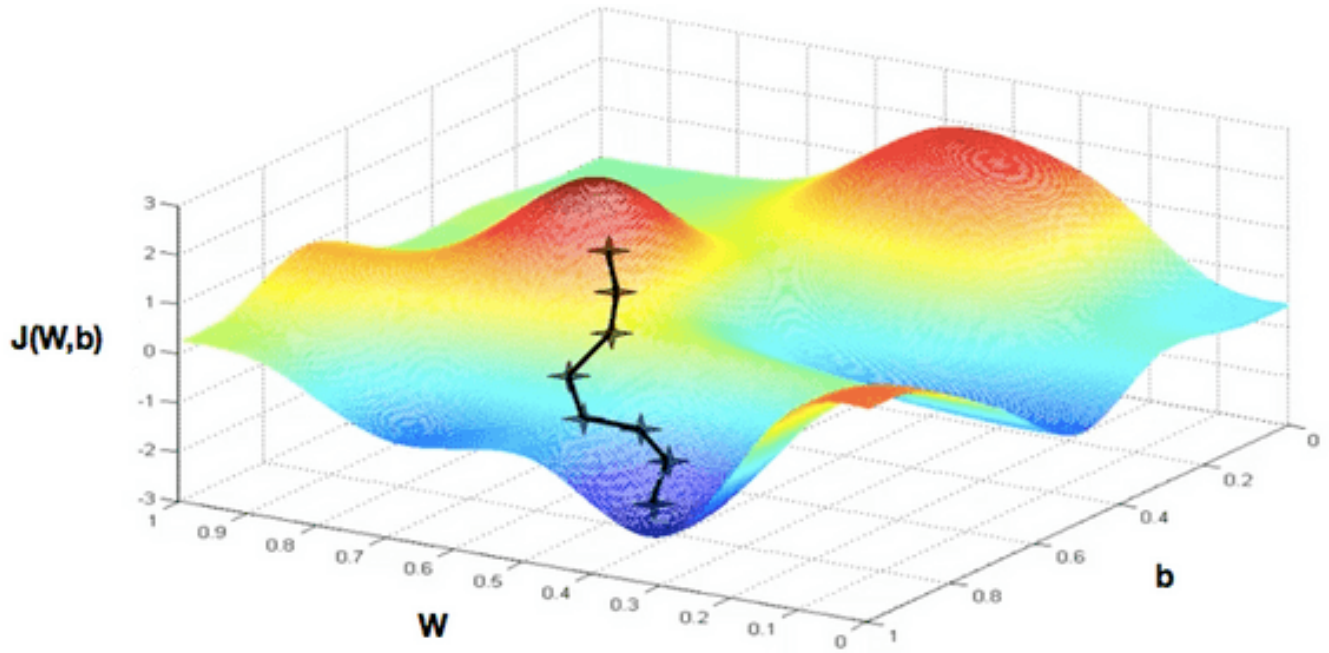
opposite direction

$$\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta) \quad \text{evaluated at } \Theta^0$$

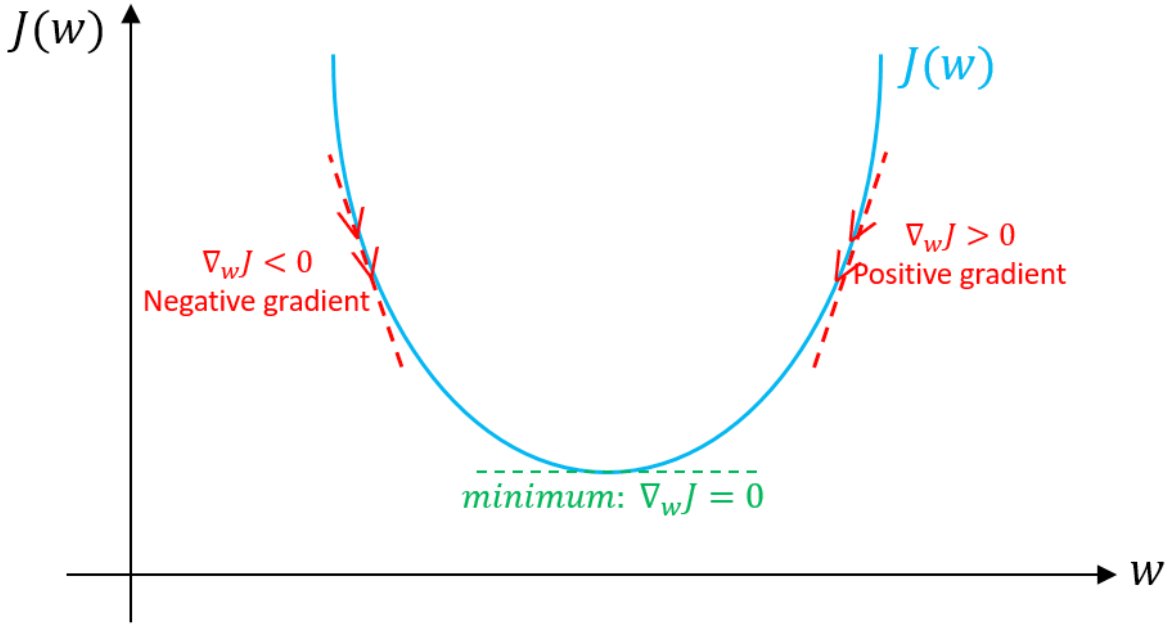
next position

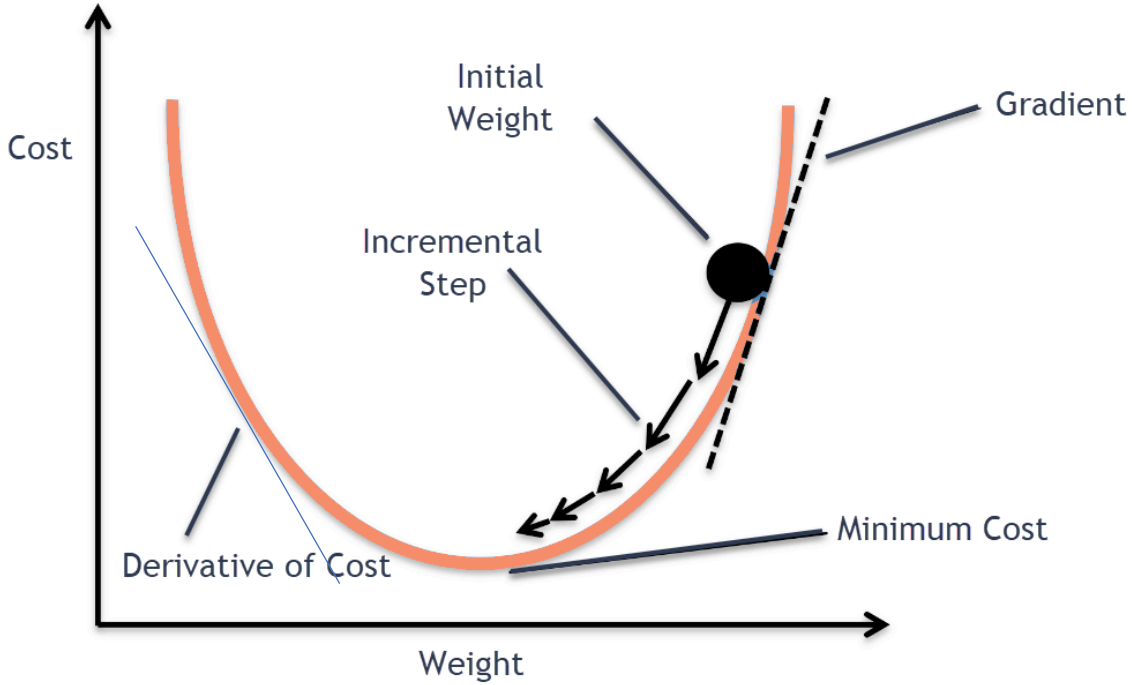
small step

direction of fastest increase



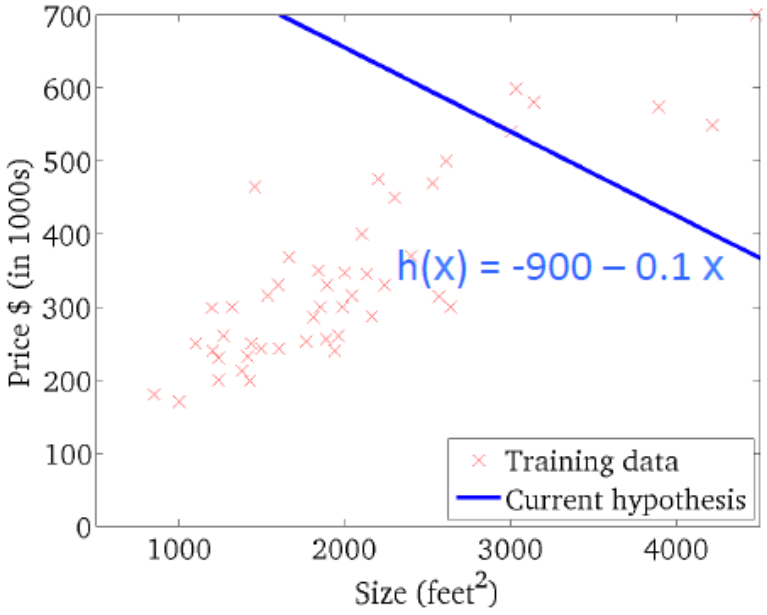
Descent towards the steepest direction with step size α


$J(w)$
$J(w)$
$\nabla_w J < 0$
Negative gradient
$\nabla_w J > 0$
Positive gradient
$minimum$: $\nabla_w J = 0$
w

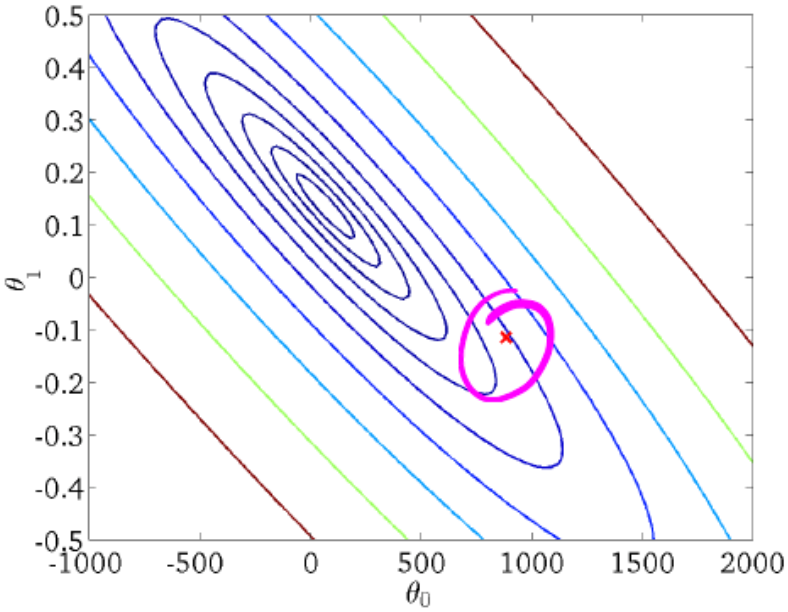

Cost
Initial
Weight
Gradient
Incremental
Step
Minimum Cost
Derivative of Cost
Weight

Gradient descent

$h_\theta(x)$

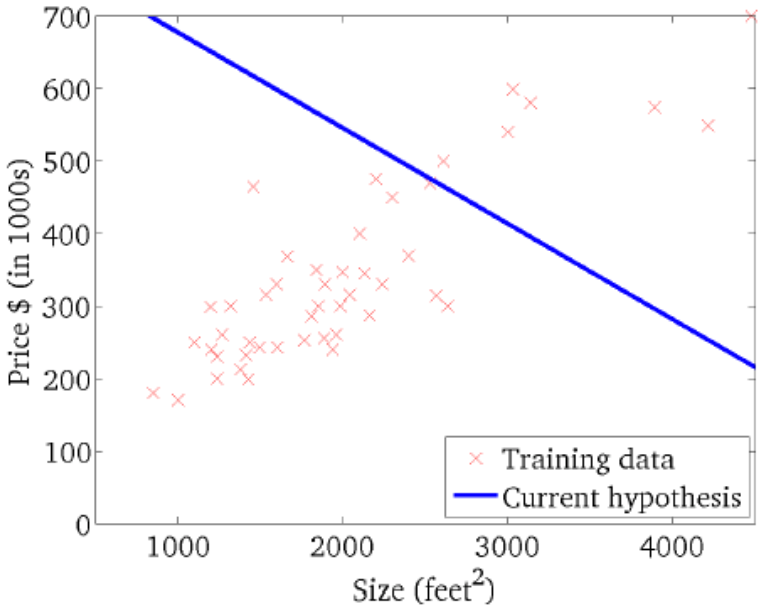


$J(\theta_0, \theta_1)$

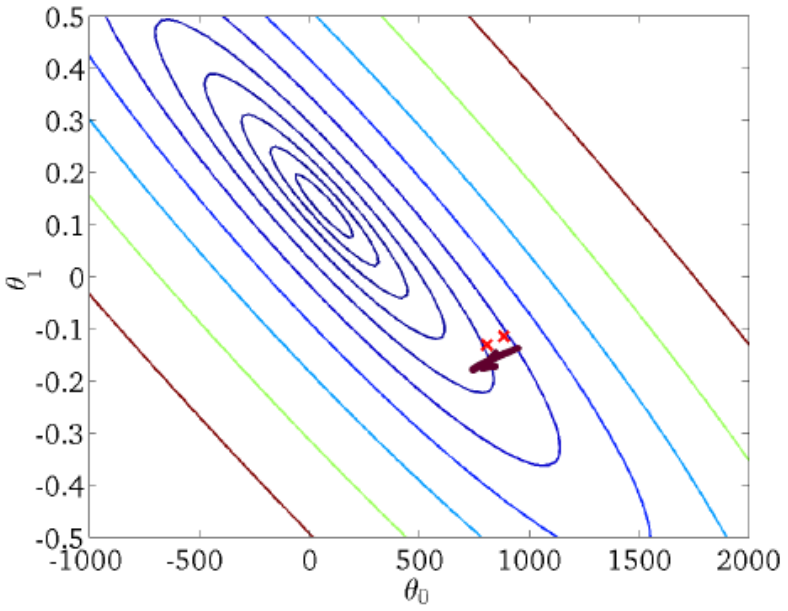


Gradient Descent

$h_\theta(x)$

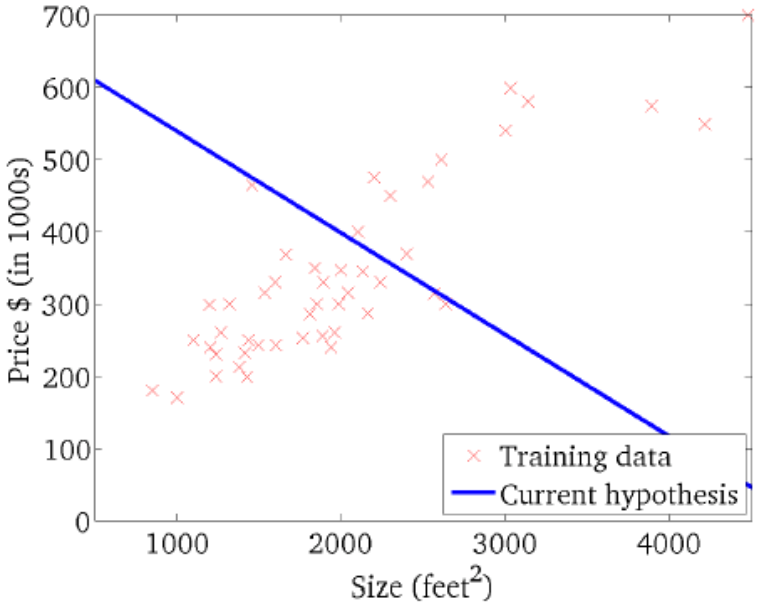


$J(\theta_0, \theta_1)$

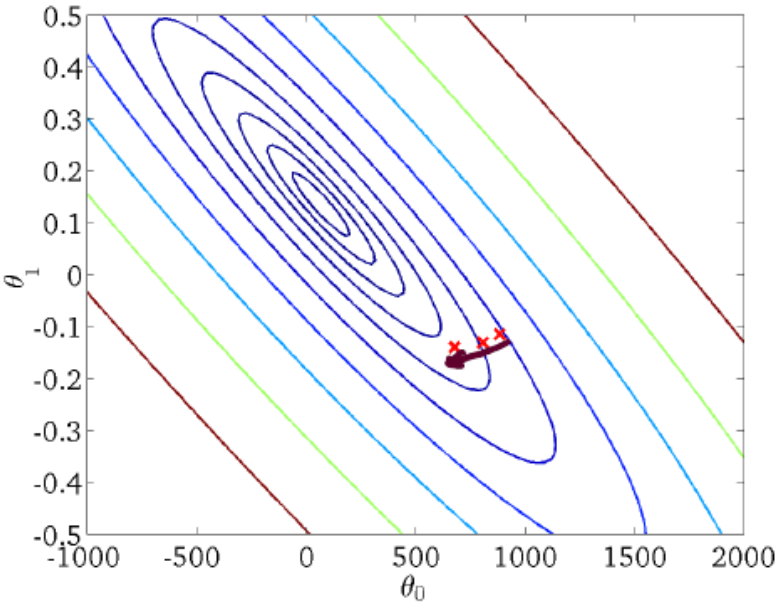


Gradient Descent

$h_\theta(x)$

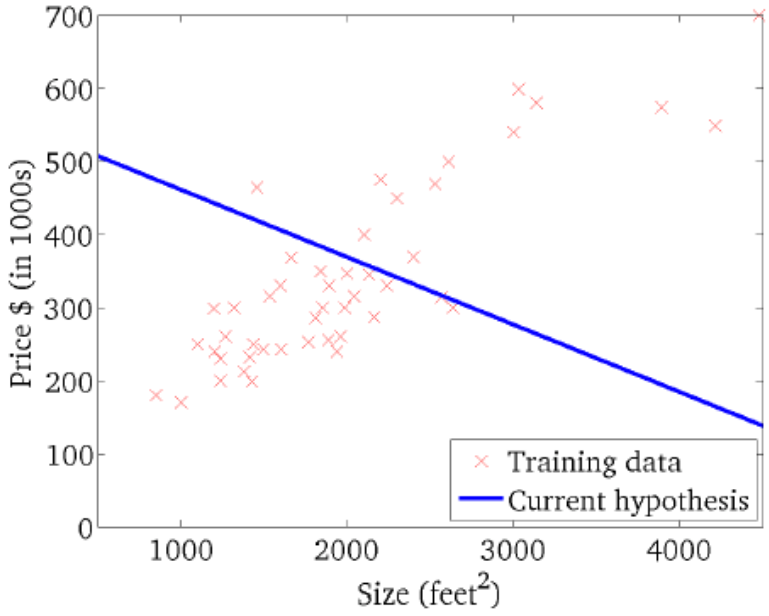


$J(\theta_0, \theta_1)$

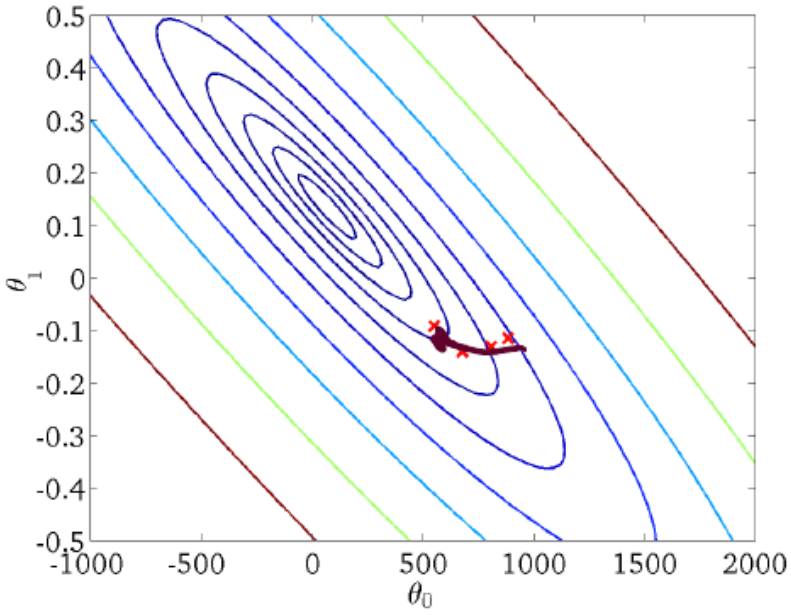


Gradient Descent

$h_\theta(x)$

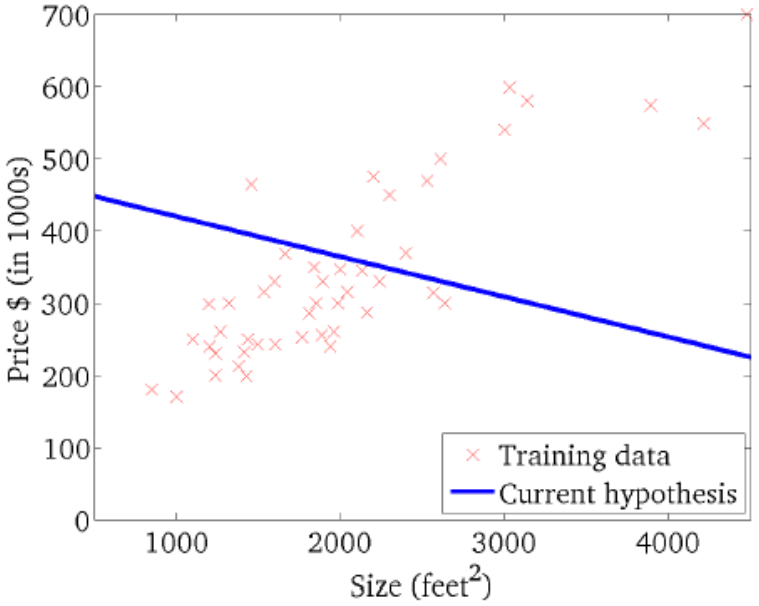


$J(\theta_0, \theta_1)$

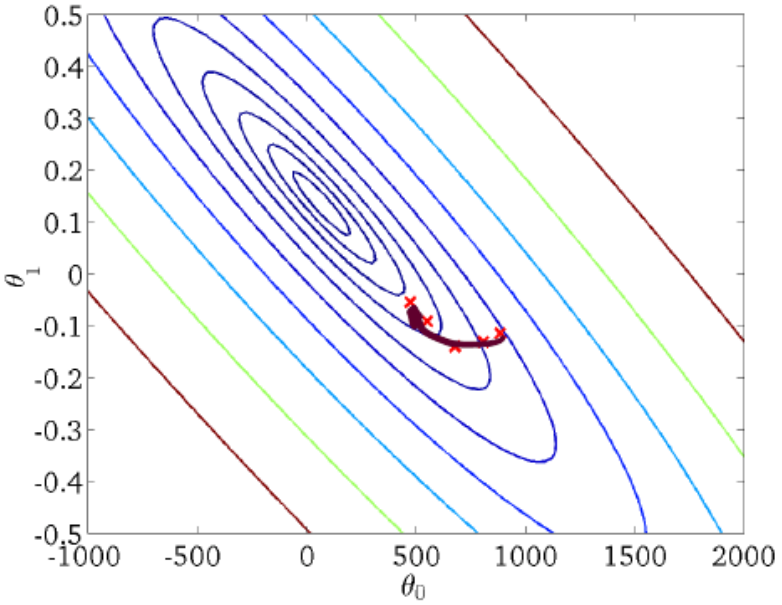


Gradient Descent

$h_\theta(x)$

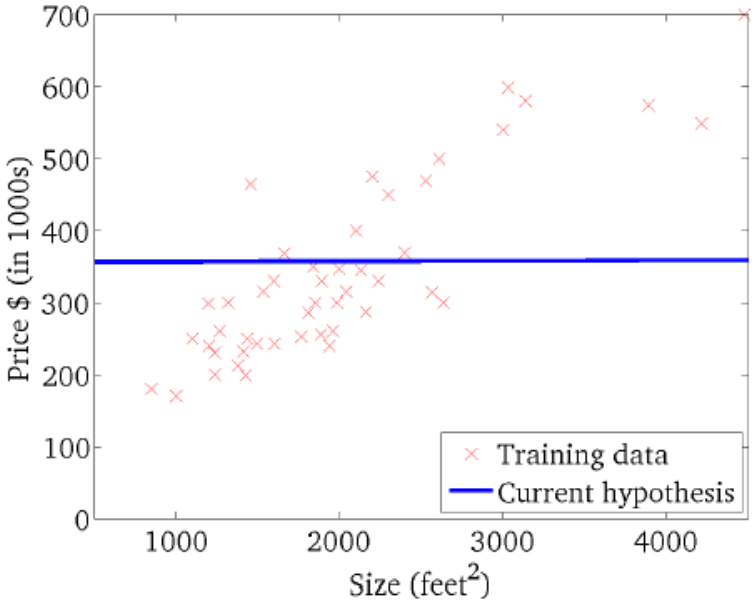


$J(\theta_0, \theta_1)$

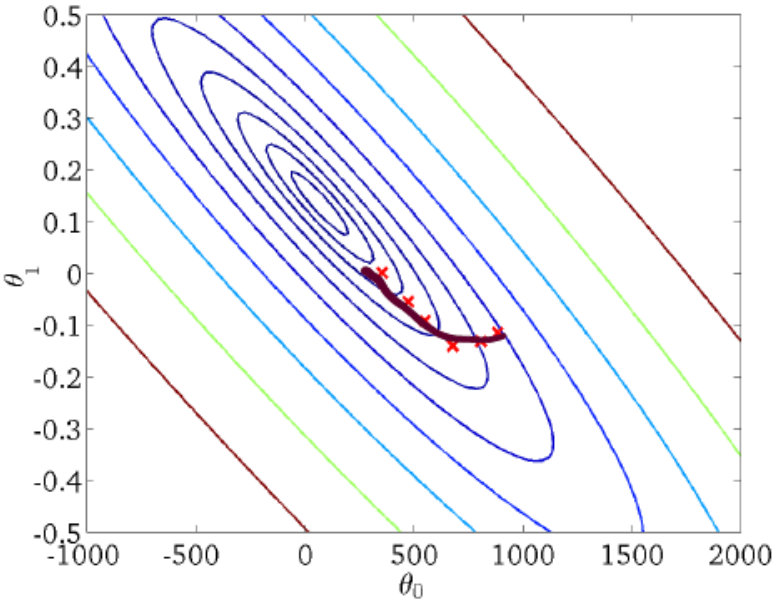


Gradient Descent

$h_\theta(x)$

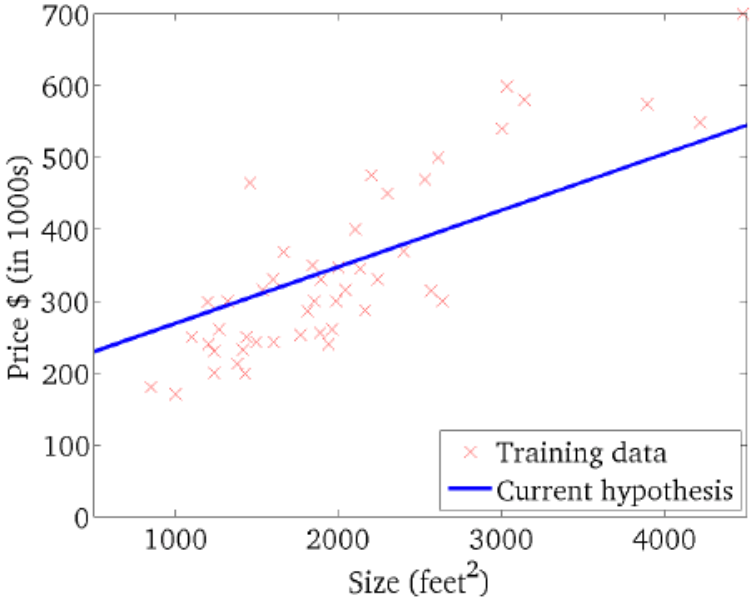


$J(\theta_0, \theta_1)$

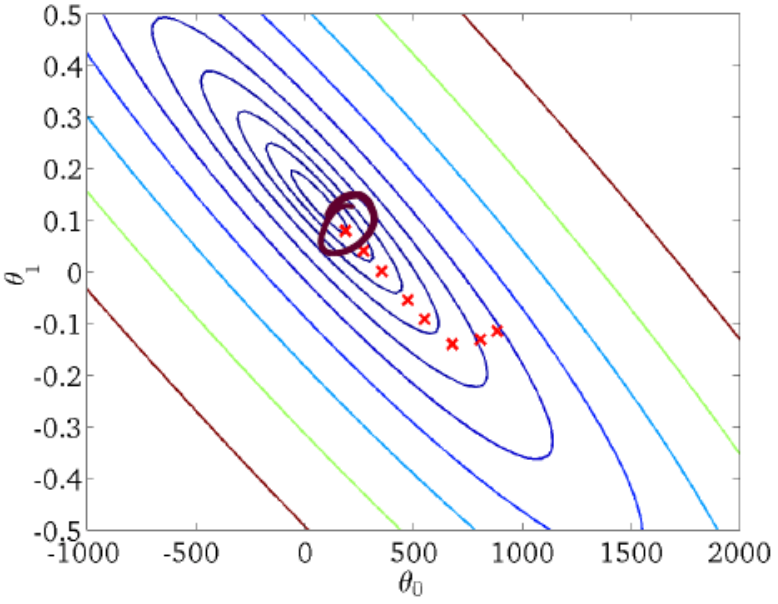


Gradient Descent

$h_\theta(x)$

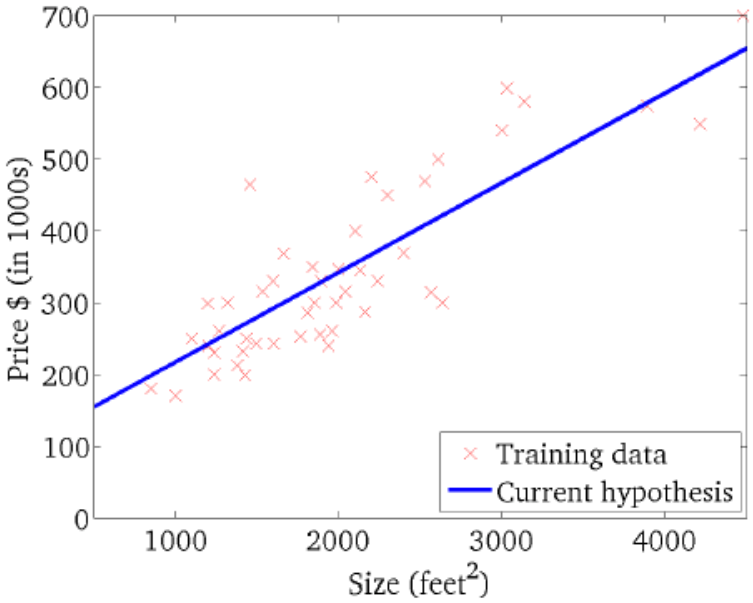


$J(\theta_0, \theta_1)$

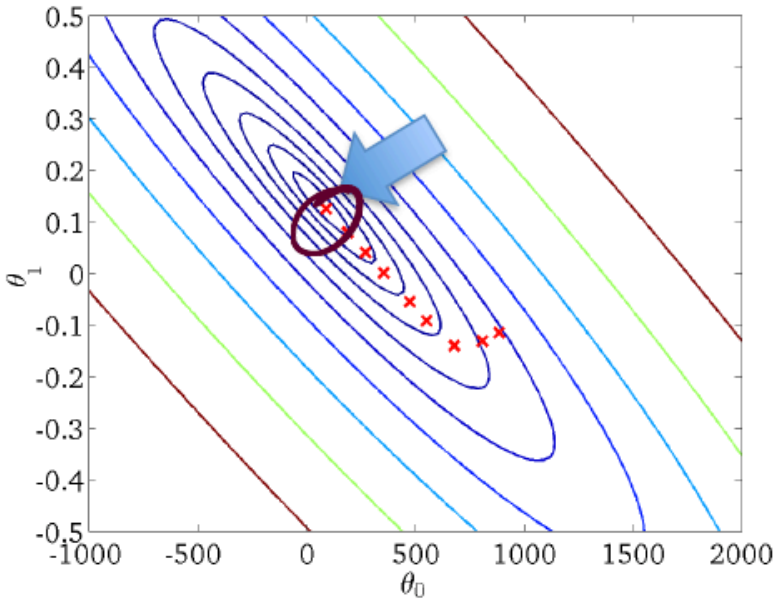


Gradient Descent

$h_\theta(x)$



$J(\theta_0, \theta_1)$



Classification of Machine Learning

- Supervised Learning
  - Example: Classification – KNN
  - Example: Regression – Linear Regression
- Unsupervised Learning
  - Example: Clustering – K-Means
- Reinforcement Learning

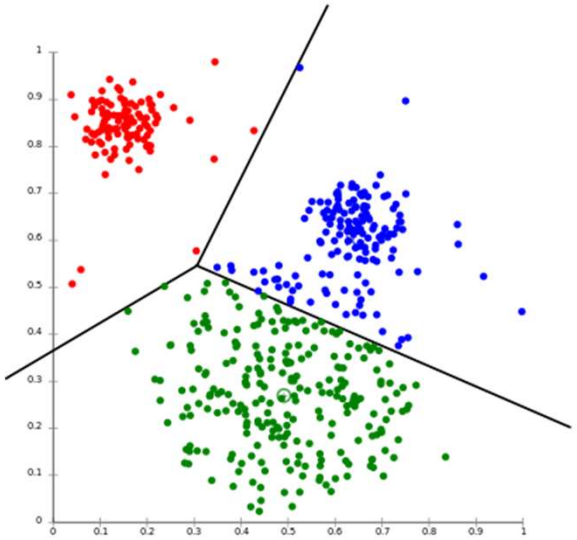
Types of Learning

2. Unsupervised Learning (data driven-exploratory)

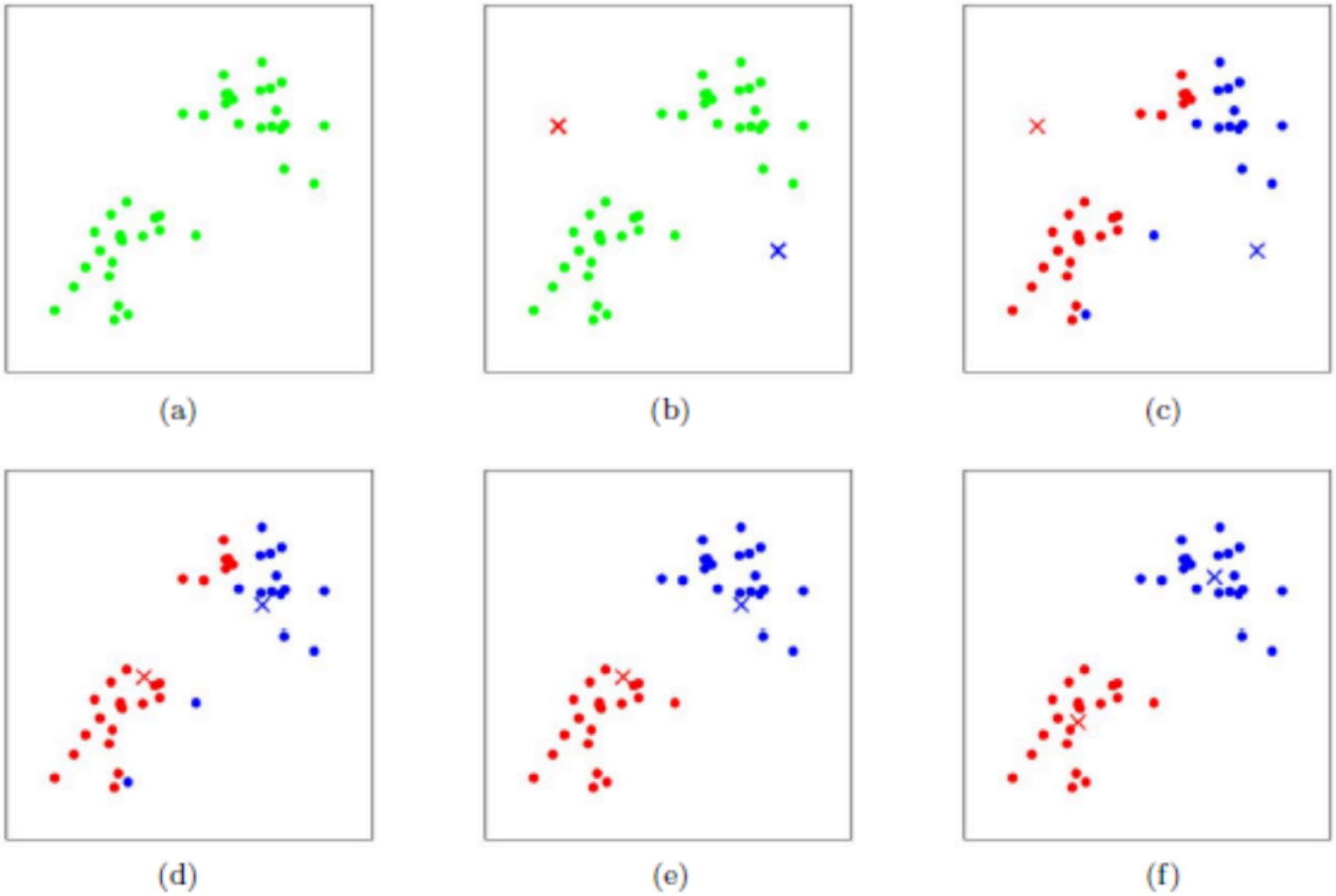
- No desired output label is given to the training data.
- Can be used to find hidden patterns/structure in data.

Examples

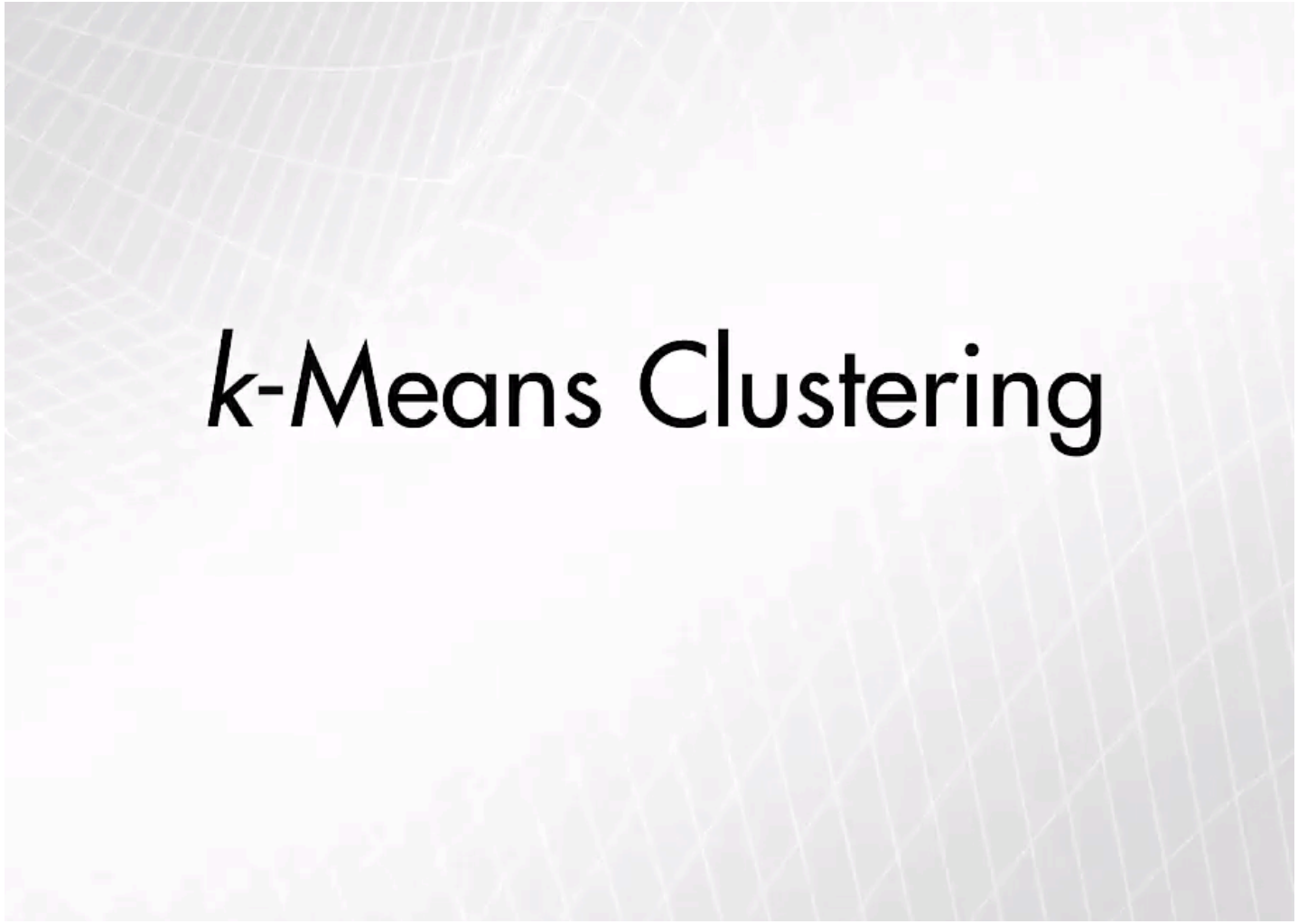
- **Clustering**
  - Shoppers into similar behaviors
  - User modeling for marketing
  - Recommendations (Netflix, Pandora, etc.)
- **Anomaly Detection**
  - Credit card fraud / Banking fraud
  - Network Attack Detection



K-means



K-means(Video)



Classification of Machine Learning

- Supervised Learning
  - Example: Classification – KNN
  - Example: Regression – Linear Regression
- Unsupervised Learning
  - Example: Clustering – K-Means
- Reinforcement Learning

Types of Learning

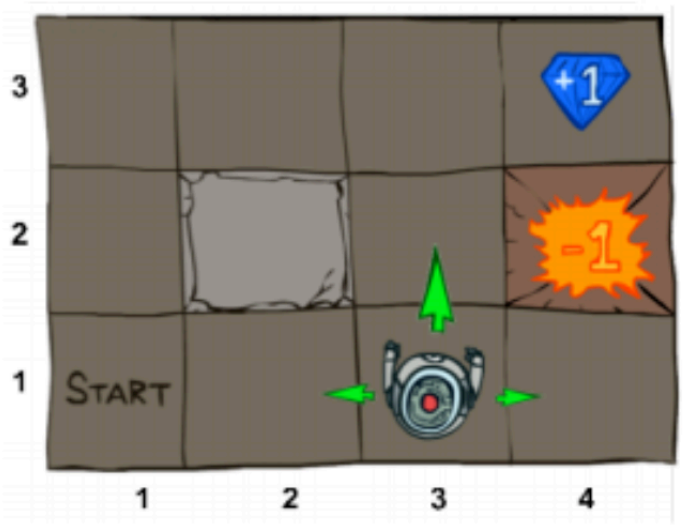
3. Reinforcement Learning (feedback driven)

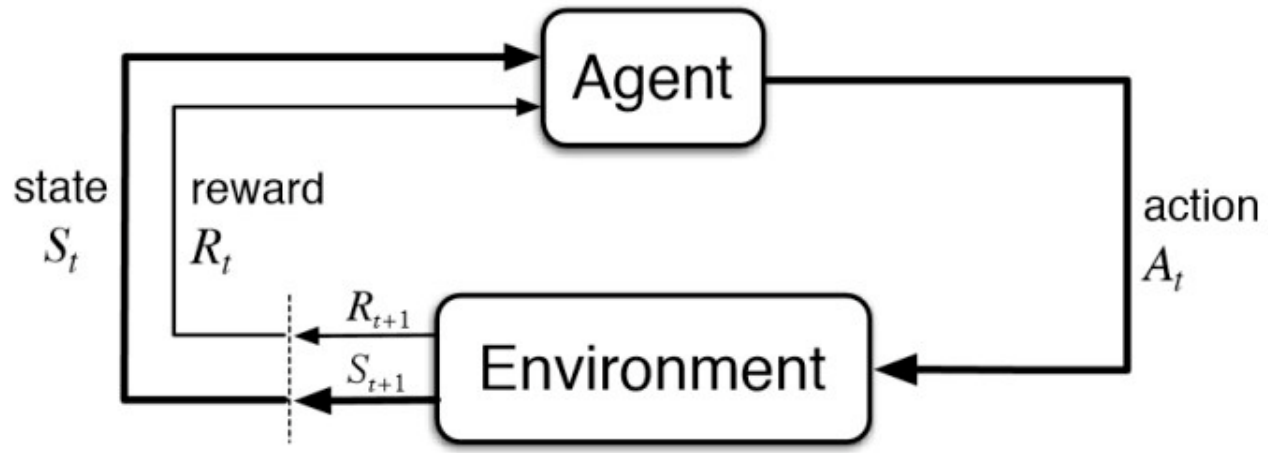
- Interacts with dynamic environment.
- Receives positive and negative feedback.

Examples

- Complex board games (chess, go, etc.)
  - Why not tic-tac-toe?
- Video games
- Self-driving cars / drones
- Robots learning to walk, adapt to environment

Reinforcement Learning





Classification of Machine Learning

- Supervised Learning
  - Example: Classification – KNN
  - Example: Regression – Linear Regression
- Unsupervised Learning
  - Example: Clustering – K-Means
- Reinforcement Learning

What kind of learning is used in these?

- Spotify recommending new songs?
- Google predicting the temperature of tomorrow?
- Face tagging feature on Facebook photos?
- Shazam song detection?
- How will we find our way out of a pitch dark room?
- Detecting fake news on the internet?

In This Lecture

- What is Machine Learning?
- Types of Learning
- **Applications / Real-Life Examples**
- How to Choose a ML Technique
- Detailed View of Few Approaches
- Project

Classification

- One type of application is **classification**.
- A popular example is text classification. Email clients have to decide whether an email is
    - 1. spam
    - 2. not spam
- This is an example of a classification task. What does the client use to make the decision? **Data!**
- The table below shows % of words/characters in an email message. Some words are found a lot in spam and not in non-spam mail:

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

Elements of Statistical Learning, Second Edition

Classification

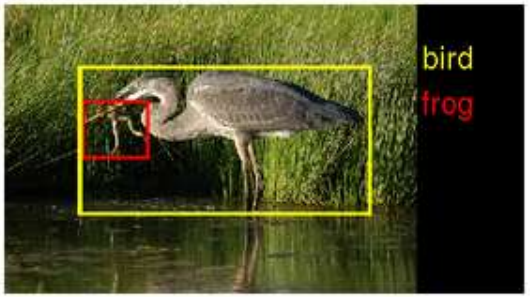
- Also common: **facial recognition**:
  - Think of a security system that has a database of employee photos. Each time a person enters the building, the classifier examines a snapshot.

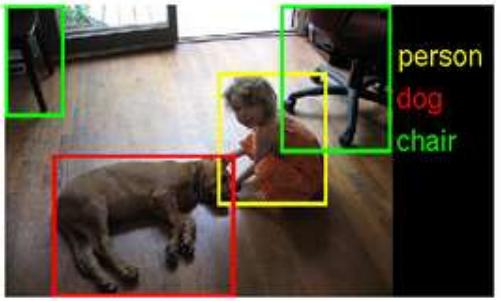
  - Goal: identify the current person as one of the employees, or an unknown person.

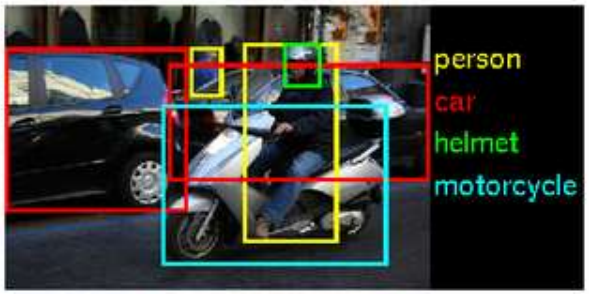
  - Challenge: facial portraits vary a lot.

Classification

- **Image recognition** attempts to identify what is in the picture







- Try this demo: http://www.clarifai.com/demo
- What images did you get tagged correctly?
- What didn't work?

Automatic Translation

- Another task is to translate documents
- We have lots of data to start with: thousands of manually translated documents are available

Soldats de ma vieille Garde,
Je vous fais mes adieux.

Depuis vingt ans, je vous ai trouvés constamment sur le chemin de l'honneur et de la gloire. Dans ces derniers temps, comme dans ceux de notre prospérité, vous n'avez cessé d'être des modèles de bravoure et de fidélité.

Avec des hommes tels que vous, notre cause n'était pas perdue. Mais la guerre était interminable ; c'eut été la guerre civile, et la France n'en serait devenue que plus malheureuse. J'ai donc sacrifié tous nos intérêts à ceux de la patrie ; je pars. Vous, mes amis, continuez de servir la France.

Soldiers of my Old Guard,
I bid you farewell.

For twenty years, I found you constantly on the path of honor and glory. In recent times, as in those of our prosperity, you have ceased to be models of courage and fidelity.

With men like you, our cause was not lost. But the war was endless; it would have been civil war, and France would become more miserable. So I sacrificed all our interests with those of the country; I leave. You, my friends, continue to serve France.

- Try it with translate.google.com
- Does much better than just a word-for-word literal translation

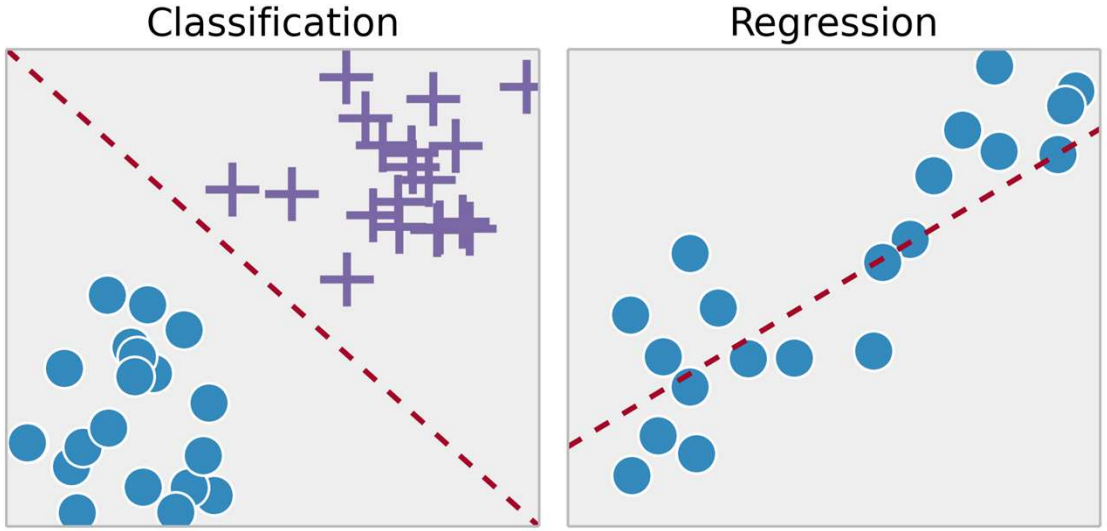
Prediction/ Regression

- The first examples we gave were classification applications
- The task was to classify some data as belonging to a category
- Translation was different: the goal was to produce a different output for each possible input. This is a **prediction** task.
- Prediction is one of the most important and powerful applications for machine learning.
- Example: economic decisions. How much revenue is received per dollar of advertising spent?
- Can you think of other examples?

Classification versus regression

- If the output we're producing from our algorithm is discrete, the problem is **classification**
  - Discrete: takes values in a finite set
  - Binary classification: generate a 0/1 or yes/no output

- If the output we're producing is continuous, we're performing **regression**
  - The output could be anything.
  - Fundamental task is **linear** regression:



Fun Example 1: Picture Stylization (Deep Learning)

- "Supplemental Material for "Visual Attribute Transfer through Deep Image Analogy " – Jing Liao, et al. (Microsoft Research)
- https://liaojing.github.io/html/data/analogy_supplemental.pdf

- Can render image A in the style of image B!

- Good reference: https://handong1587.github.io/deep_learning/2015/10/09/fun-with-deep-learning.html

Fun Example 2: Robotics

- **BRETT: Berkeley Robot for the Elimination of Tedious Tasks**
- https://www.youtube.com/watch?v=JeVppkoloXs
- Not preprogrammed with knowledge how to accomplish specific tasks.

- **Starfish Walking Robot**
- https://www.youtube.com/watch?v=ehno85yl-sA
- Taught itself to walk.

- **Google DeepMind AI taught itself how to walk.**
- https://www.youtube.com/watch?v=gn4nRCC9TwQ
- https://deepmind.com/blog/producing-flexible-behaviours-simulated-environments/#gif-90

- **Boston Dynamics**
- https://www.youtube.com/watch?v=4PaTWufUqqU
- https://www.youtube.com/watch?v=fRj34o4hN4I

Fun Example 3: MarI/O

- Uses a simple Neural Network to play Mario.
- No concept of what any button does, except that going to the right = more points.
- www.youtube.com/watch?v=qv6UVOQ0F44&

- More levels:
- www.youtube.com/watch?v=iakFfOmanJU

- Same setup for Mario Kart
- www.youtube.com/watch?v=iakFfOmanJU

Videos

Object detection and segmentation

https://www.youtube.com/watch?v=OOT3UIXZztE

Autonomous driving:

https://www.youtube.com/watch?v=KS_4xjXNTxg

NVIDIA DRIVE Labs: Predicting the Future with RNNs
https://www.youtube.com/watch?v=OT_MxopvfQ0

Style transfer for videos, as described in the paper "Artistic style transfer for videos"

https://www.youtube.com/watch?v=Khuj4ASldmU

RL playing FPS Games using only raw pixel data.

https://www.youtube.com/watch?v=oo0TraGu6QY

Google Duplex: A.I. Assistant Calls Local Businesses To Make Appointments
https://www.youtube.com/watch?v=D5VN56jQMWM

In This Lecture

- What is Machine Learning?
- Types of Learning
- Applications / Real-Life Examples
- **How to Choose a ML Technique**
- Detailed View of Few Approaches
- Project

Evaluation Matrices

- The 'purity' of positive prediction results

  Precision: $P = \frac{TP}{TP+FP}$
- Percentage of positive samples that are correctly predicted

  Recall: $R = \frac{TP}{TP+FN}$
- Harmonic Mean of Precision and Recall: **F1**

  $\boldsymbol{F_1} = \frac{2}{\frac{1}{P}+\frac{1}{R}}$
- **Accuracy**

  $A = \frac{TP+TN}{TP+TN+FP+FN}$

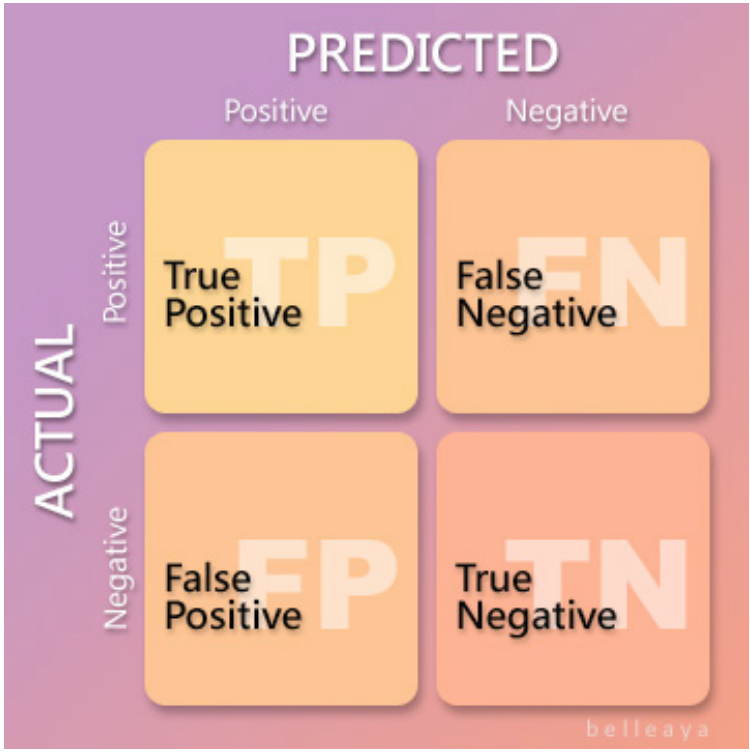
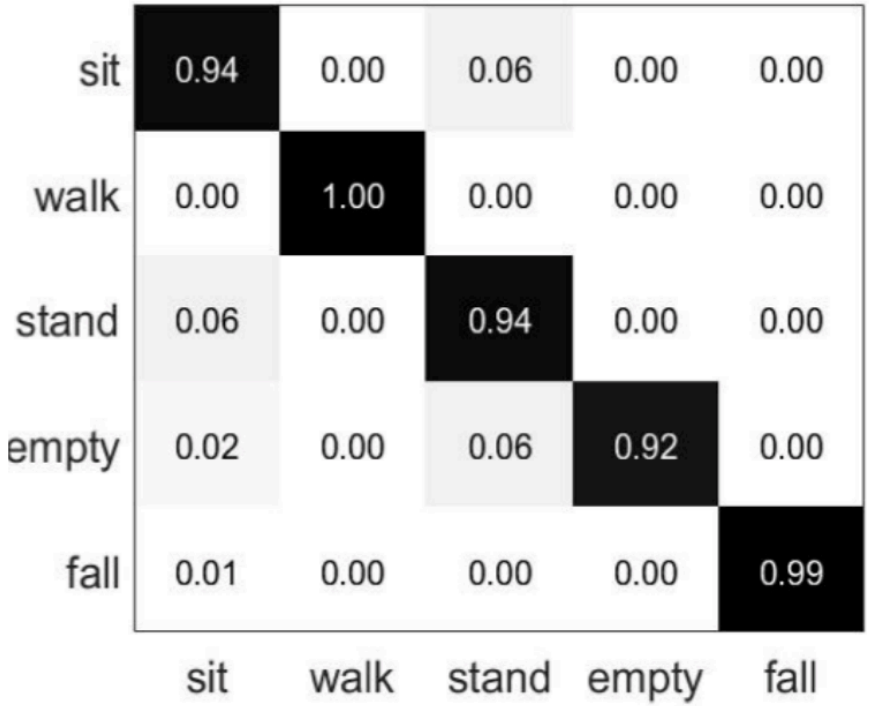


Figure: https://belleaya.pixnet.net/blog/post/43873939-%5B%E5%AD%B8%E7%BF%92%5D-%5B%E5%88%86%E6%9E%90%5D-%E6%B7%B7%E6%B7%86%E7%9F%A9%E9%99%A3%E8%88%87tp%E3%80%81tn%E3%80%81fp%E3%80%81fn

Confusion Matrix

- Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa)

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100



Evaluation Matrices for Regression

RMSE (Root Mean Square Error)

It represents the sample standard deviation of the differences between predicted values and observed values (called residuals)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

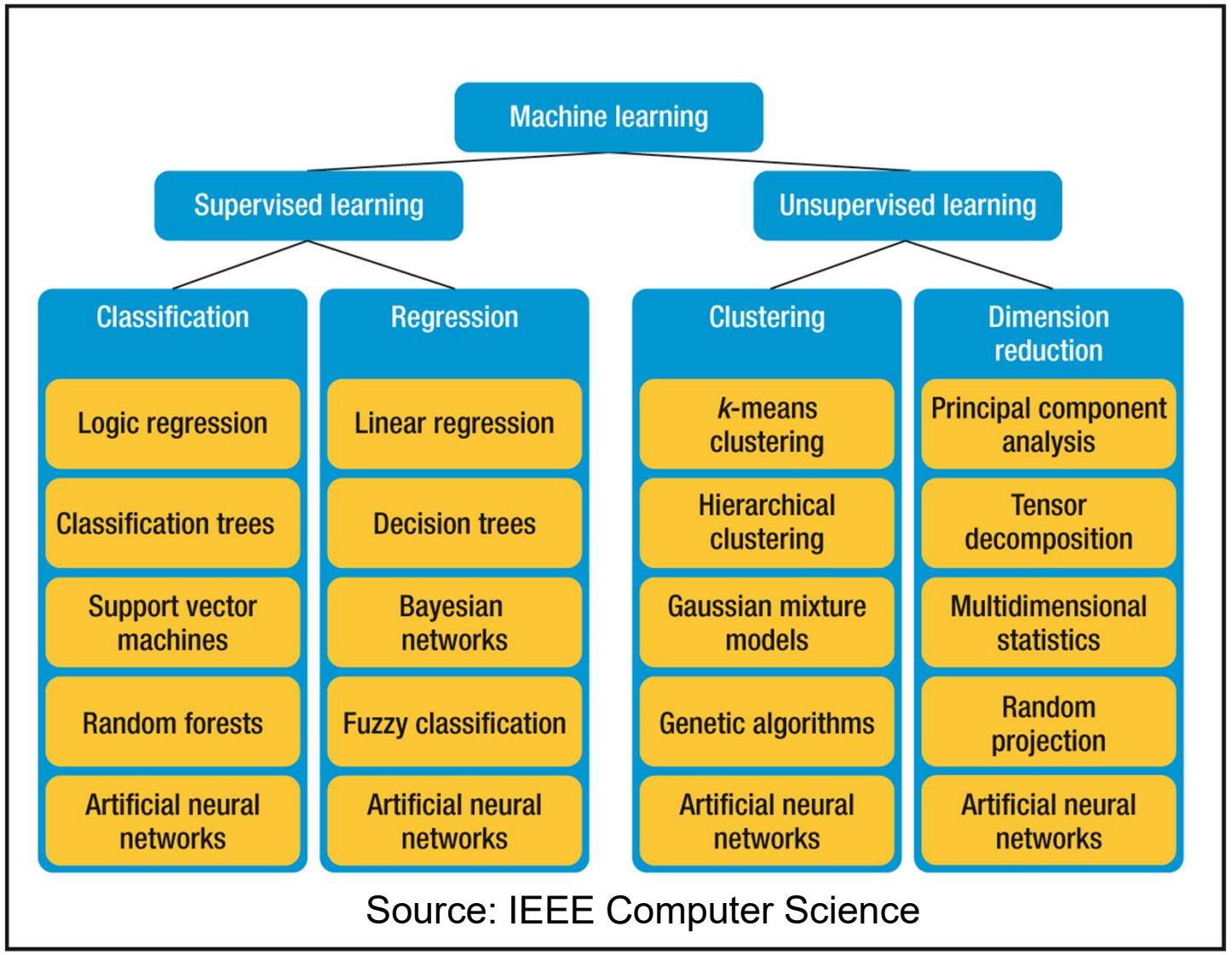
MAE (Mean Average Error)

MAE is the average of the absolute difference between the predicted values and observed value.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Reference: https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

What Machine Learning Techniques Exist?

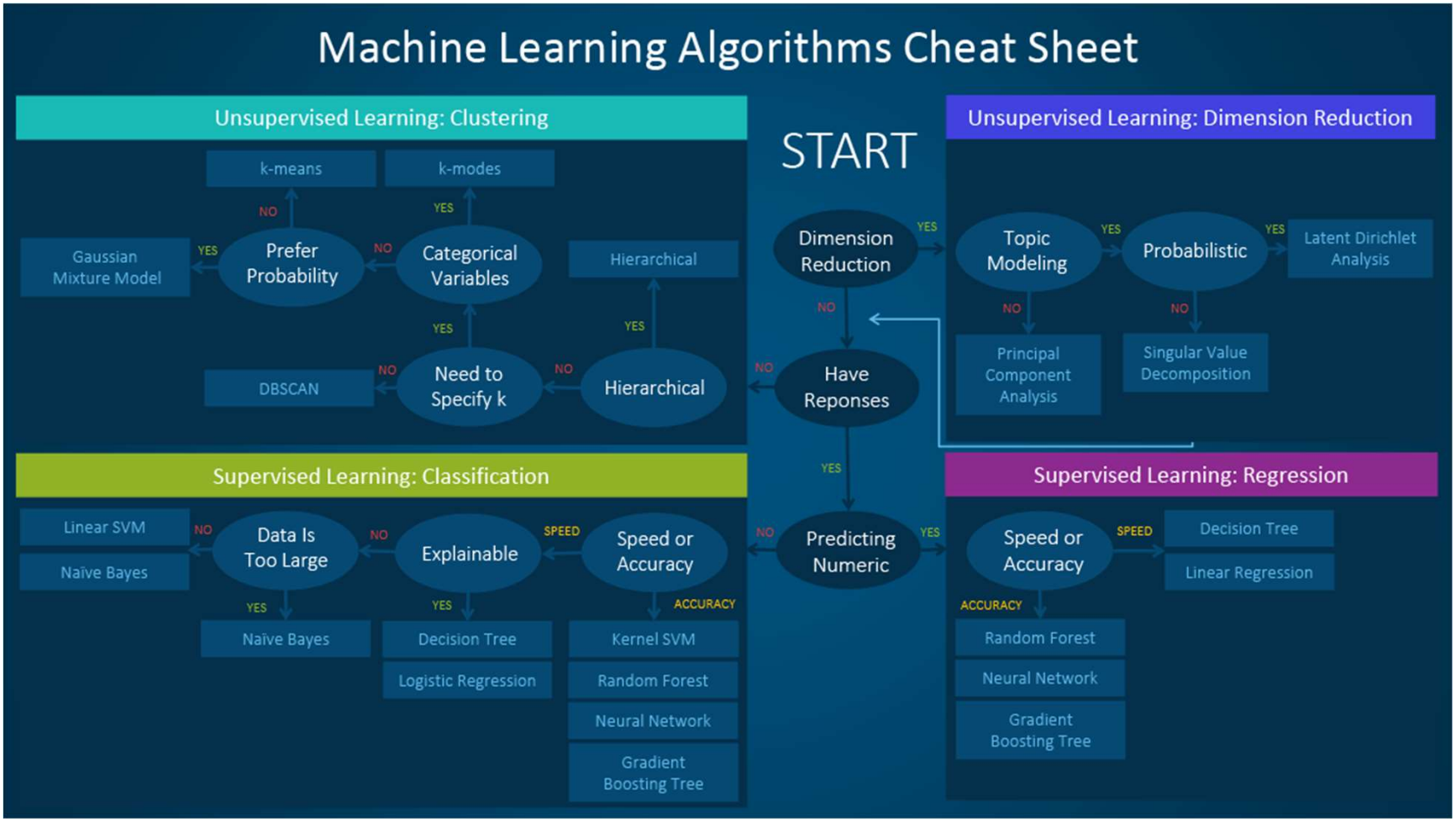


Source: IEEE Computer Science

How to choose the Best Algorithm?

- Very complicated. This is why data scientists are in high demand in the industry.
- Can’t just throw data at the computer.
- It is a lot of trial and error but different communities (like computer vision, natural language processing) have their own best practices.
- We briefly look at a simplified flowchart.
- This is a interesting read https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/#prettyPhoto

How to Choose the Best Algorithm

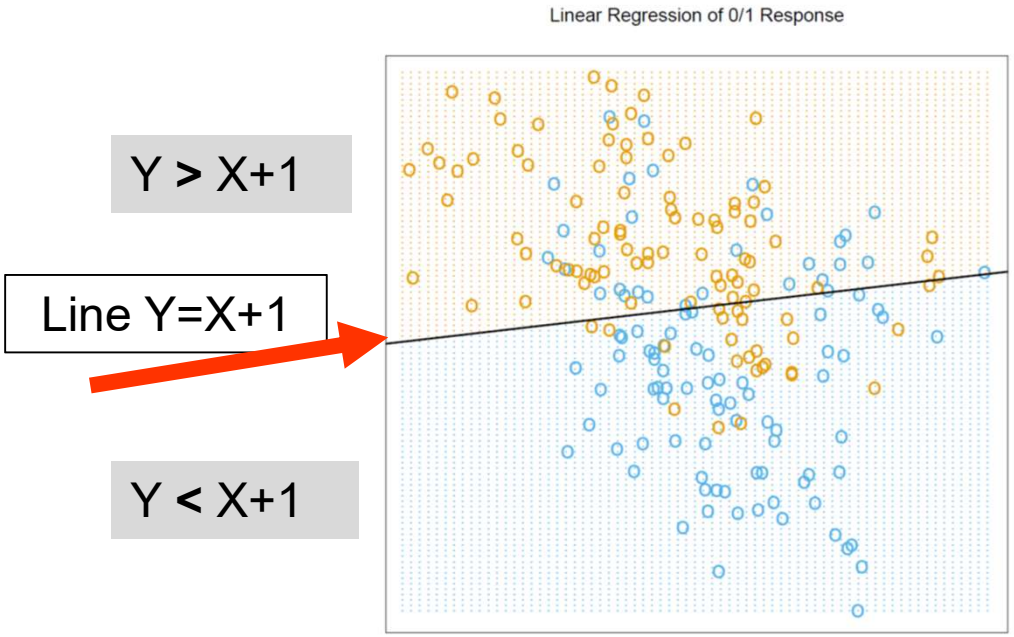


In This Lecture

- What is Machine Learning?
- Types of Learning
- Applications / Real-Life Examples
- How to Choose a ML Technique
- **Detailed View of Few Approaches**
- Project

Linear classification

- We want to define a linear boundary that is our best estimate of the regions where each class belongs.



There are two parts here:

1. The equation of the line separating the two classes is chosen usually to correctly classify **most** points.
2. Misclassified points give some residual training error.

- Our linear separator is just a line in the XY plane; if a point is above it, we classify it as orange. If it's below, we classify it as blue.

Classification in Higher Dimensions

- For a line in 2-D, the equation is $y = \beta_1 x + \beta_0$ with two parameters.
- In 3-D the linear separator would be a plane with equation $y = \beta_1 x_1 + \beta_2 x_2 + \beta_0$ and three parameters.
- In N-D the general linear separator is called a hyperplane.
- For a N-D hyperplane, the equation is $y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{N-1} x_{N-1} + \beta_0$.
- N parameters $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots, \beta_{N-1}]$.

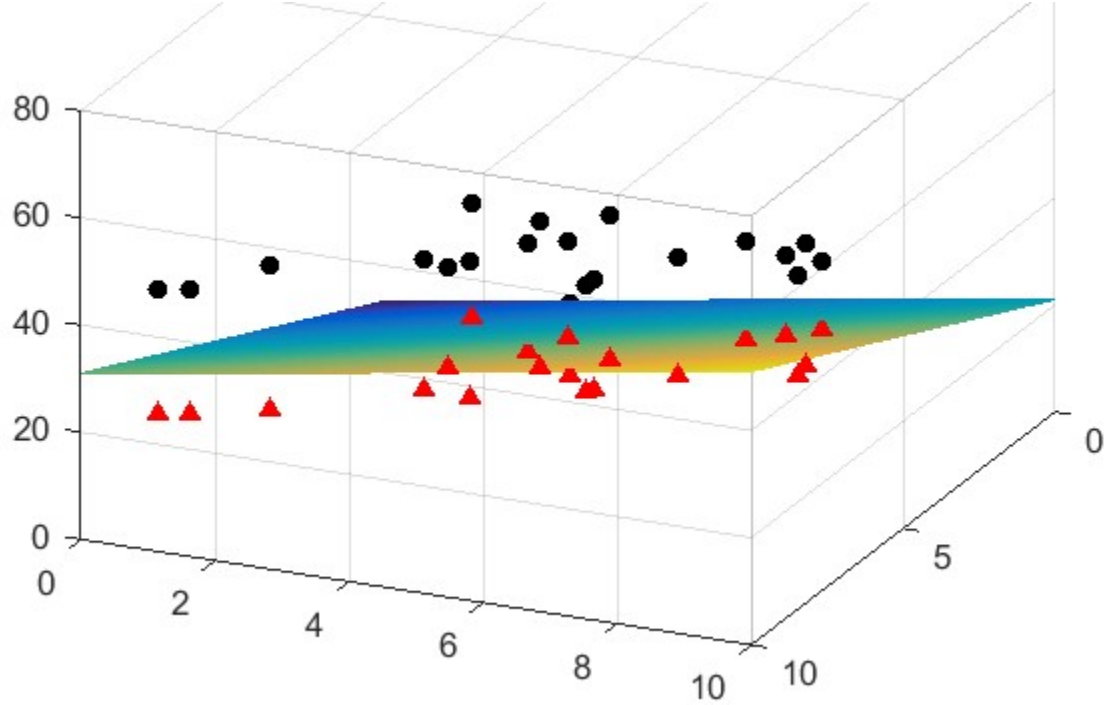


Fig: A linear boundary in three dimensions separating two classes, **red** and **black**.

Decision Tree Classifier

Algorithm 1 DecisionTreeTrain $(data, features)$

1: $guess \leftarrow$ the most frequent label in $data$
2: **if** all labels in $data$ are the same **then**
3: **return** LEAF $(guess)$
4: **else**
5: $f \leftarrow$ the "best" feature $\in features$
6: $NO \leftarrow$ the subset of $data$ on which $f = NO$
7: $YES \leftarrow$ the subset of $data$ on which $f = YES$
8: $left \leftarrow$ DecisionTreeTrain $(NO, features - \{f\})$
9: $right \leftarrow$ DecisionTreeTrain $(YES, features - \{f\})$
10: **return** NODE$(f, left, right)$
11: **end if**

we will specify how to pick best.

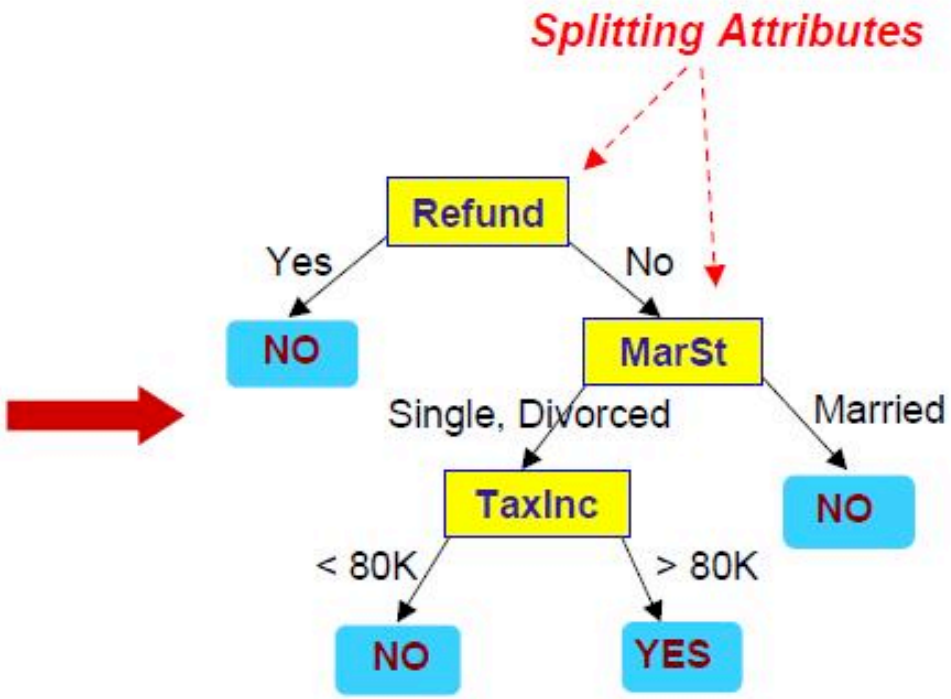
recurse

Decision Tree Classifier

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

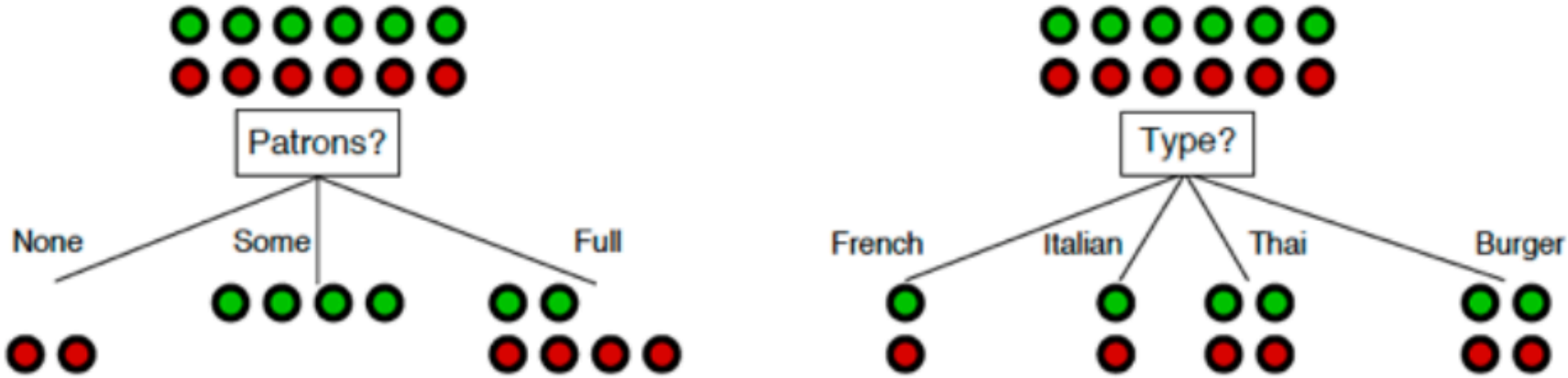
categorical categorical continuous class

Training Data



Model: Decision Tree

Which attribute to split?



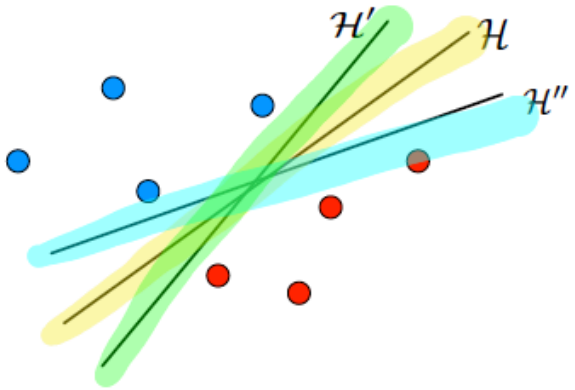
Patrons? is a better choice—gives **information** about the classification

Idea: use information gain to choose which attribute to split

SVM - Intuition

Consider the following *separable* training dataset, i.e., we assume there exists a decision boundary that separates the two classes perfectly. There are an *infinite* number of decision boundaries $\mathcal{H} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$!

Question: Which of these hyperplanes is 'better'?



Intuition: We want training pts to be as far away from hyperplane as possible

⇒ more 'robust'

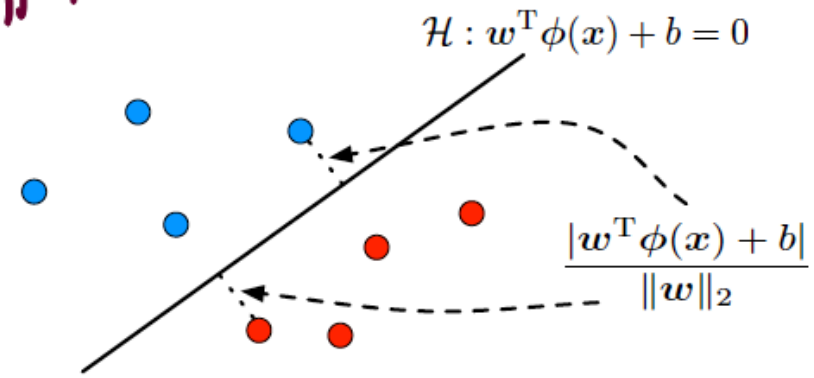
Which one should we pick?

Optimizing the Margin

Margin Smallest distance between the hyperplane and all training points

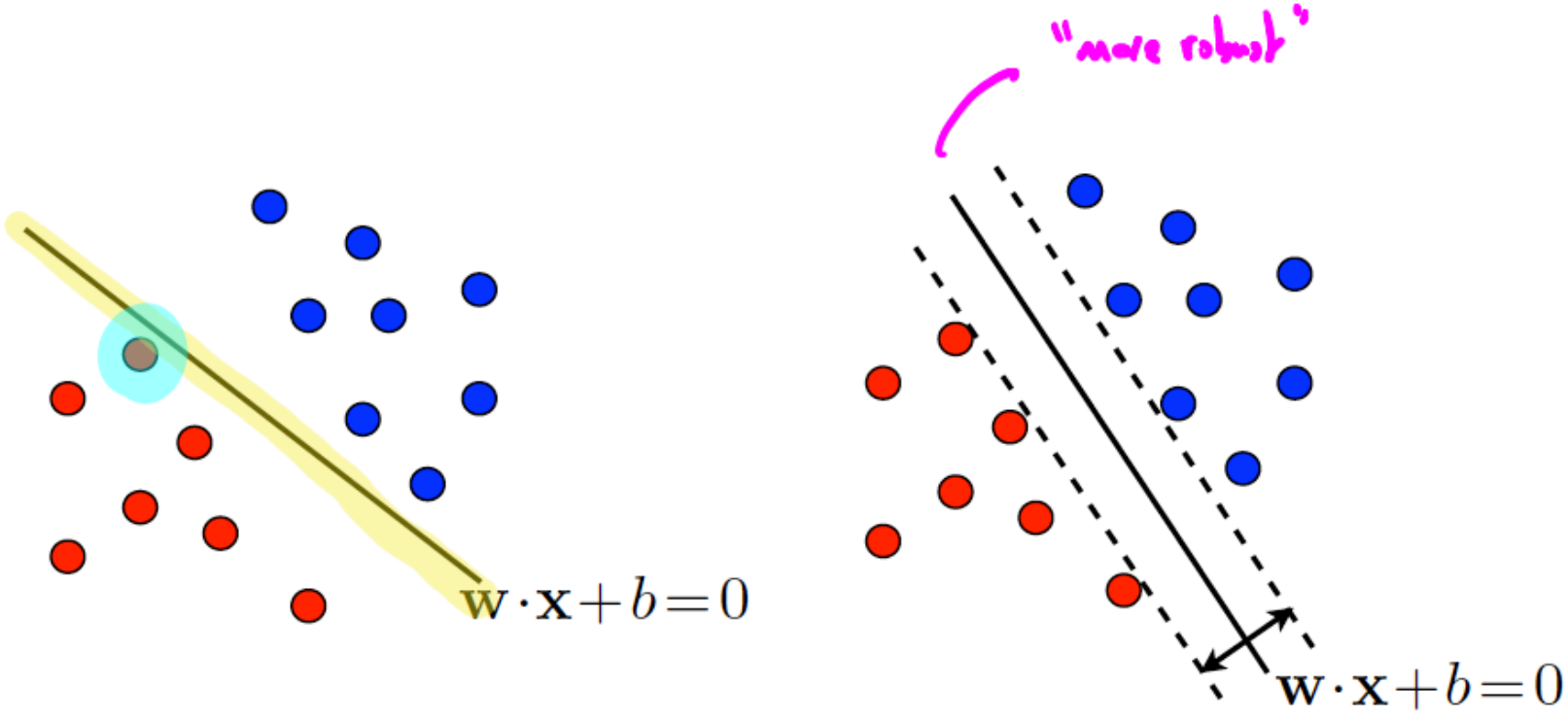
$$\text{MARGIN}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]}{\|\boldsymbol{w}\|_2}$$

distance of $\mathcal{D}$ to hyperplane $(\underline{w}, b)$

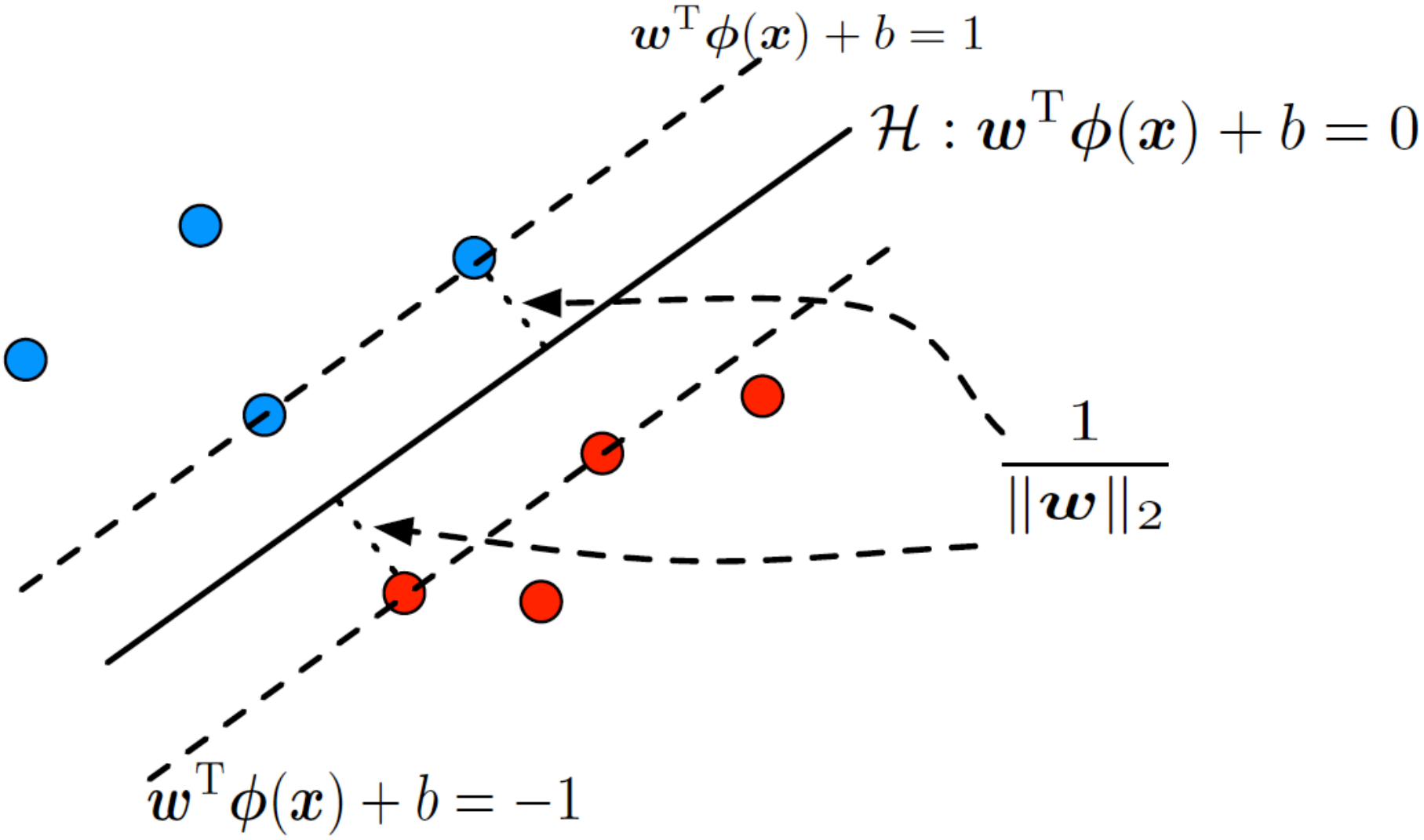


- The goal of SVM is to find the best hyperplane that separates the training data

SVM



- Definition of ‘best’ : The hyperplane that maximize the margin


$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 1$
$\mathcal{H} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$
$\frac{1}{\|\boldsymbol{w}\|_2}$
$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = -1$

SVM for separable data

Constrained optimization problem

Max margin classifier

$$\min_{\boldsymbol{w},b} \quad \underbrace{\frac{1}{2}\|\boldsymbol{w}\|_2^2}_{\text{Objective function}}$$

$$\text{s.t.} \quad \underbrace{y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b] \geq 1}_{\text{Constraint}}, \quad n \in \{1,\ldots,N\}$$

- Variables $(\boldsymbol{w}, b)$
- Objective function
- Constraints

If not linearly separable then constraint is infeasible.

SVM – Hinge Loss

The hinge loss function

The hinge loss is standardly defined for a binary output $y^{(i)} \in \{-1, 1\}$. If $y^{(i)} = 1$, then we would like $\mathbf{w}^T\mathbf{x}^{(i)} + b$ to be large and positive. If $y^{(i)} = -1$, then we would like $\mathbf{w}^T\mathbf{x}^{(i)} + b$ to be large and negative. The larger $a_i(\mathbf{x}^{(j)})$ is in the right direction, the larger the margin.

In this scenario, the hinge loss is for $x^{(i)}$ being in class $y^{(i)}$ is:

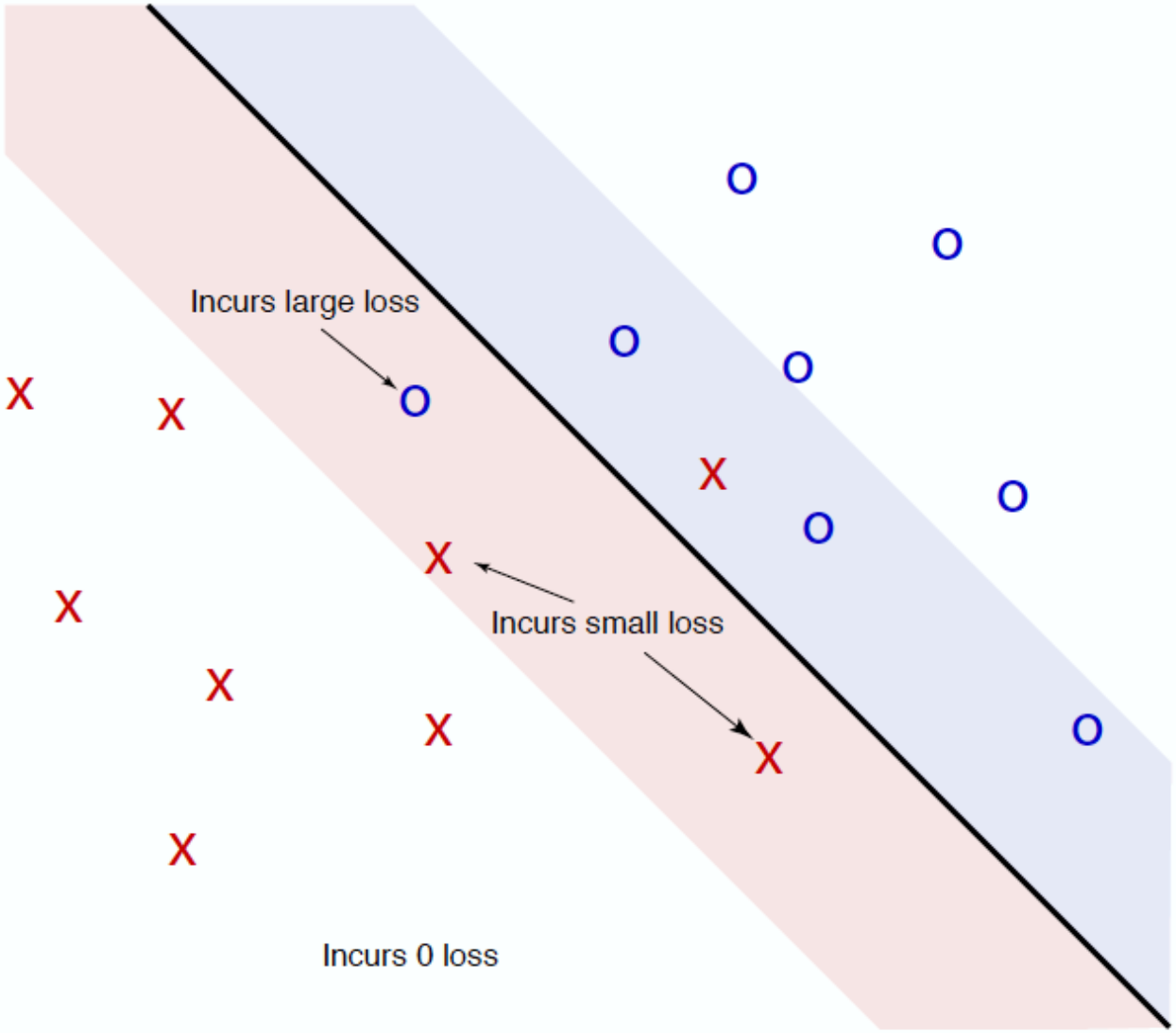
$$\text{hinge}_{y^{(i)}}(\mathbf{x}^{(i)}) = \max(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b))$$

$y^{(i)} = 1$ $\quad w^T x^{(i)} + b >> 0 \Rightarrow$ hinge loss $= \max(0, 1 - \text{large pos. \#}) = 0$

$y^{(i)} = 1$ $\quad w^T x^{(i)} + b = 0.3 \Rightarrow$ hinge loss $= \max(0, 1 - 0.3) = 0.7$

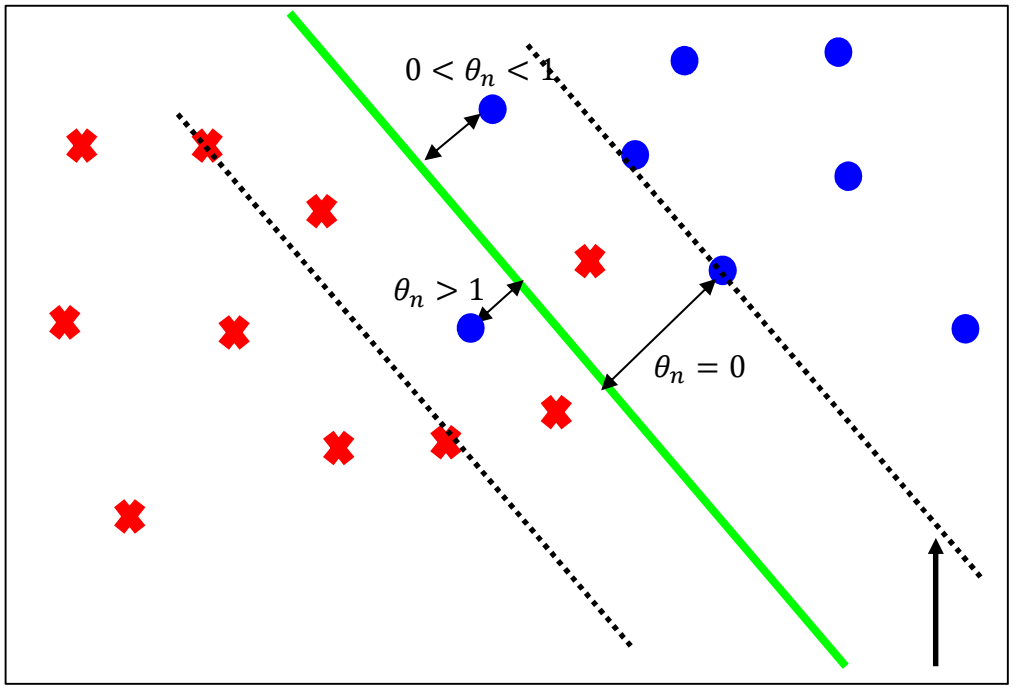
$y^{(i)} = 1$ $\quad w^T x^{(i)} + b = -1 \Rightarrow$ hinge loss $= \max(0, 1 - (-1)) = 2$

This is the picture we should have in mind:



Extending SVM to non-separable data

- What if the classes are not linearly separable?
- Then there are points e.g.
  - With class $y_n = 1$ and
    - $w^T x_n + b < 0$,
- So the constraint $y_n(w^T x_n + b) \geq 1$ can NOT be satisfied!
- We add a slack $\theta_n \geq 0$ to the constraint.
- Minimize θ_n.
- $y_n(w^T x_n + b) \geq 1 - \theta_n$,
- $\theta_n \geq 0$



Margin line

SVM for non-separable data

Constraints in separable setting

$$y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad n \in \{1, \ldots, N\}$$

Constraints in non-separable setting

Idea: modify our constraints to account for non-separability! Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad n \in \{1, \ldots, N\}$$

- For "hard" training points, we can increase ξ_n until the above inequalities are met
- What does it mean when ξ_n is very large?

Soft-margin SVM formulation

We do not want ξ_n to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad n \in \{1,\ldots,N\} \\ & \xi_n \geq 0, \quad n \in \{1,\ldots,N\} \end{aligned}$$

What is the role of C?

- User-defined hyperparameter
- Trades off between the two terms in our objective
- Same idea as the regularization term in ridge regression, i.e., $C = \frac{1}{\lambda}$

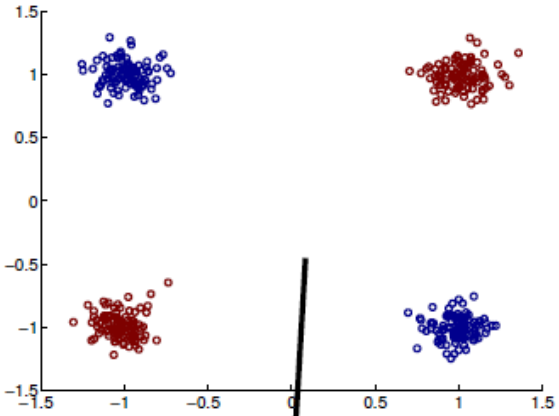
Nonlinear basis for classification

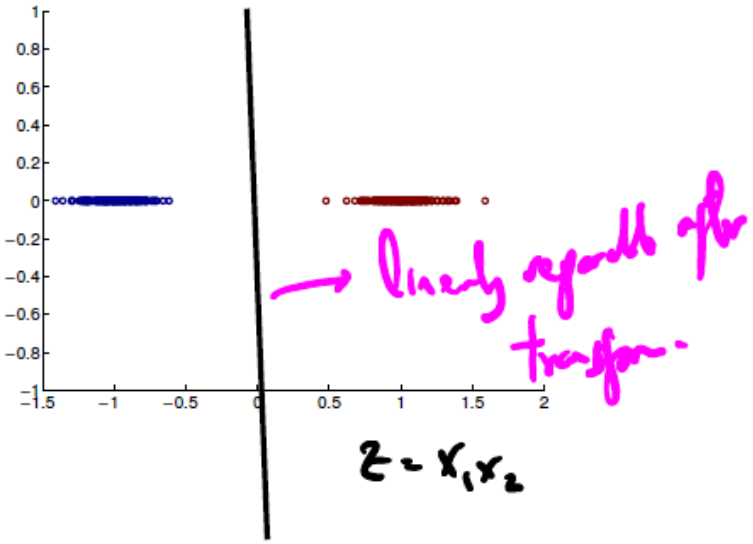
$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad z = x_1 x_2$$

Transform the input/feature

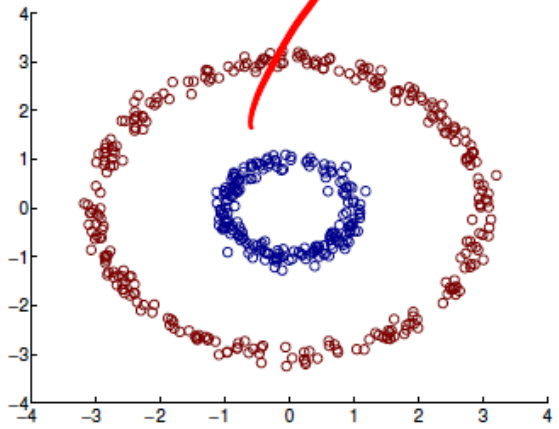
$$\phi(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^2 \rightarrow z = x_1 \cdot x_2$$

Transformed training data: linearly separable!





Another example

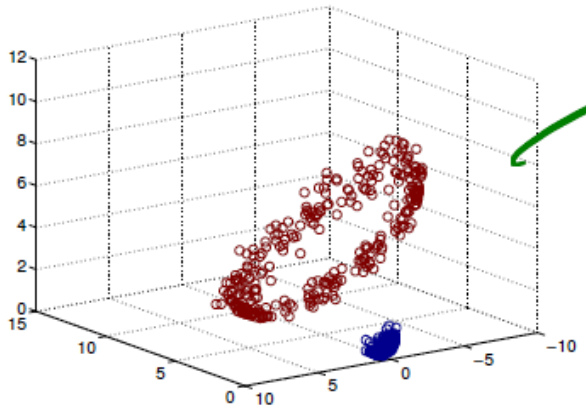


How to transform the input/feature?

$$\phi(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^2 \rightarrow \boldsymbol{z} = \begin{bmatrix} x_1^2 \\ x_1 \cdot x_2 \\ x_2^2 \end{bmatrix} \in \mathbb{R}^3$$

$\phi: \mathbb{R}^D \rightarrow \mathbb{R}^M$

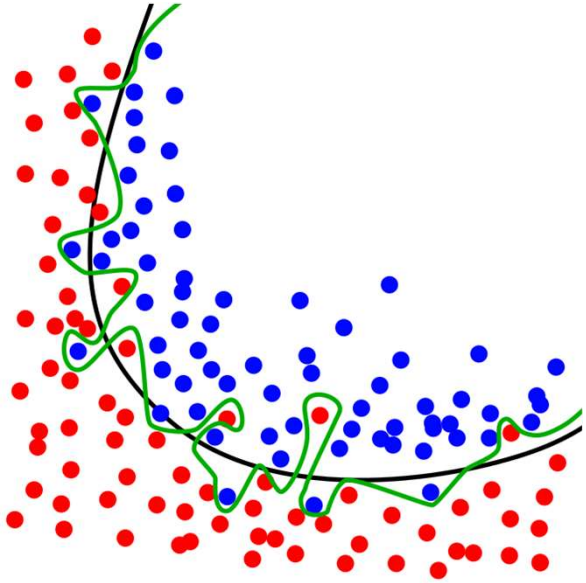
Transformed training data: linearly separable

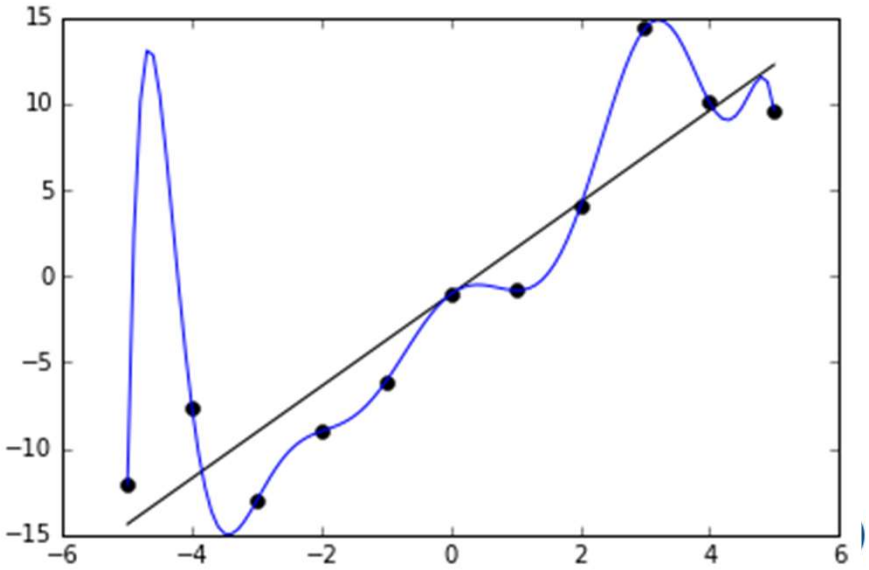


Linearly separable after transform

Machine Learning Pitfall: Overfitting/ Underfitting.

- If your model has too many parameters but not enough training data, you risk overfitting or bad generalization performance.
- Overfitting means your training error will be very low but error on new data will be high.
- On the other hand, your model may be too simplistic and you may have high training and test error. With practice you learn to recognize both and find the right complexity of model.
- Splitting examples into test set and training set is **very important**.





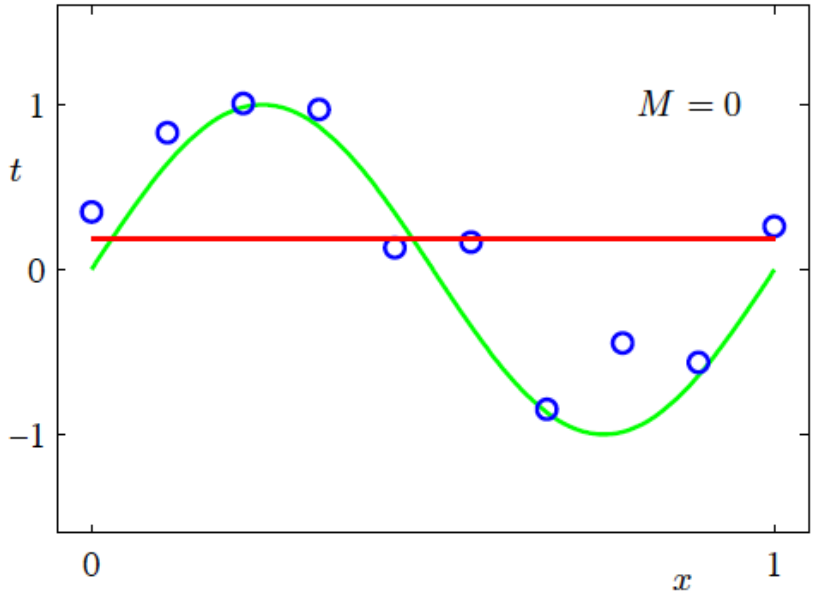
Example with regression

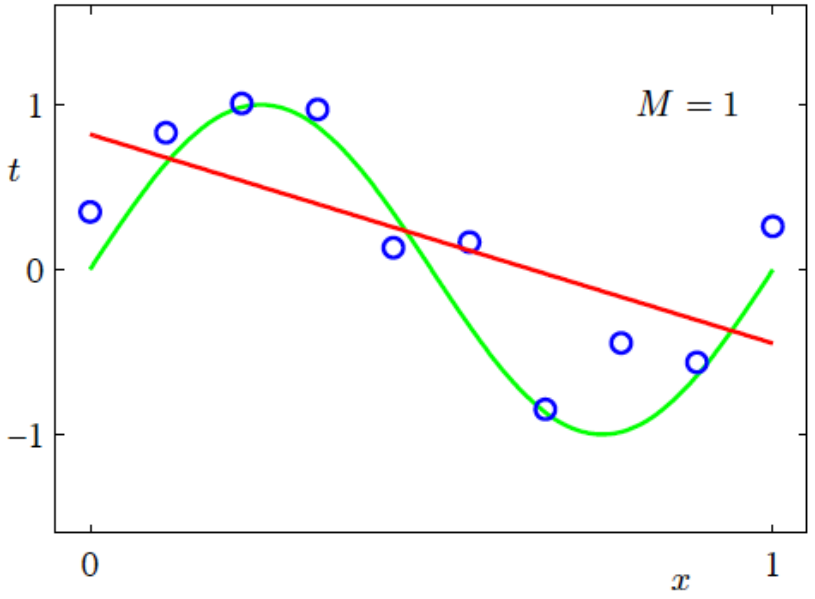
Polynomial basis functions

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^M \end{bmatrix} \Rightarrow h(x) = \theta_0 + \sum_{m=1}^{M} \theta_m x^m$$

hyperparameter

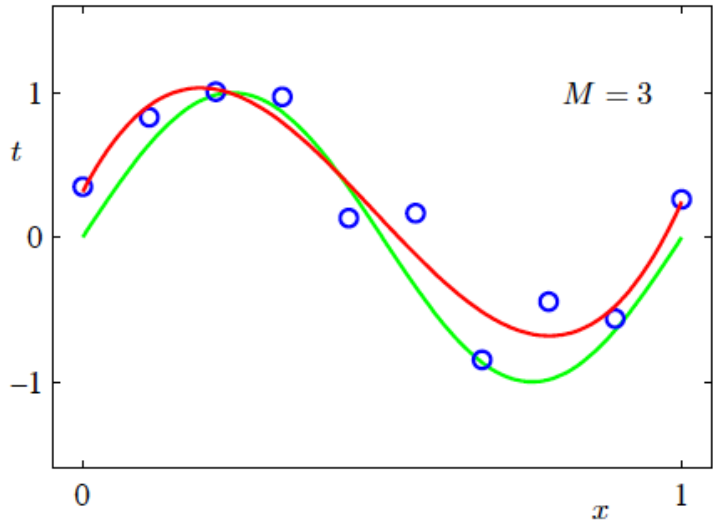
Fitting samples from a sine function: *underrfitting* as $h(x)$ is too simple



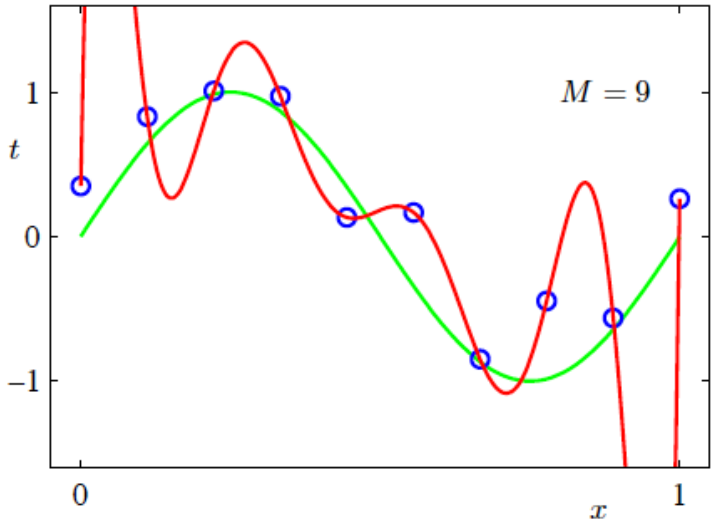


Adding high-order terms

M=3

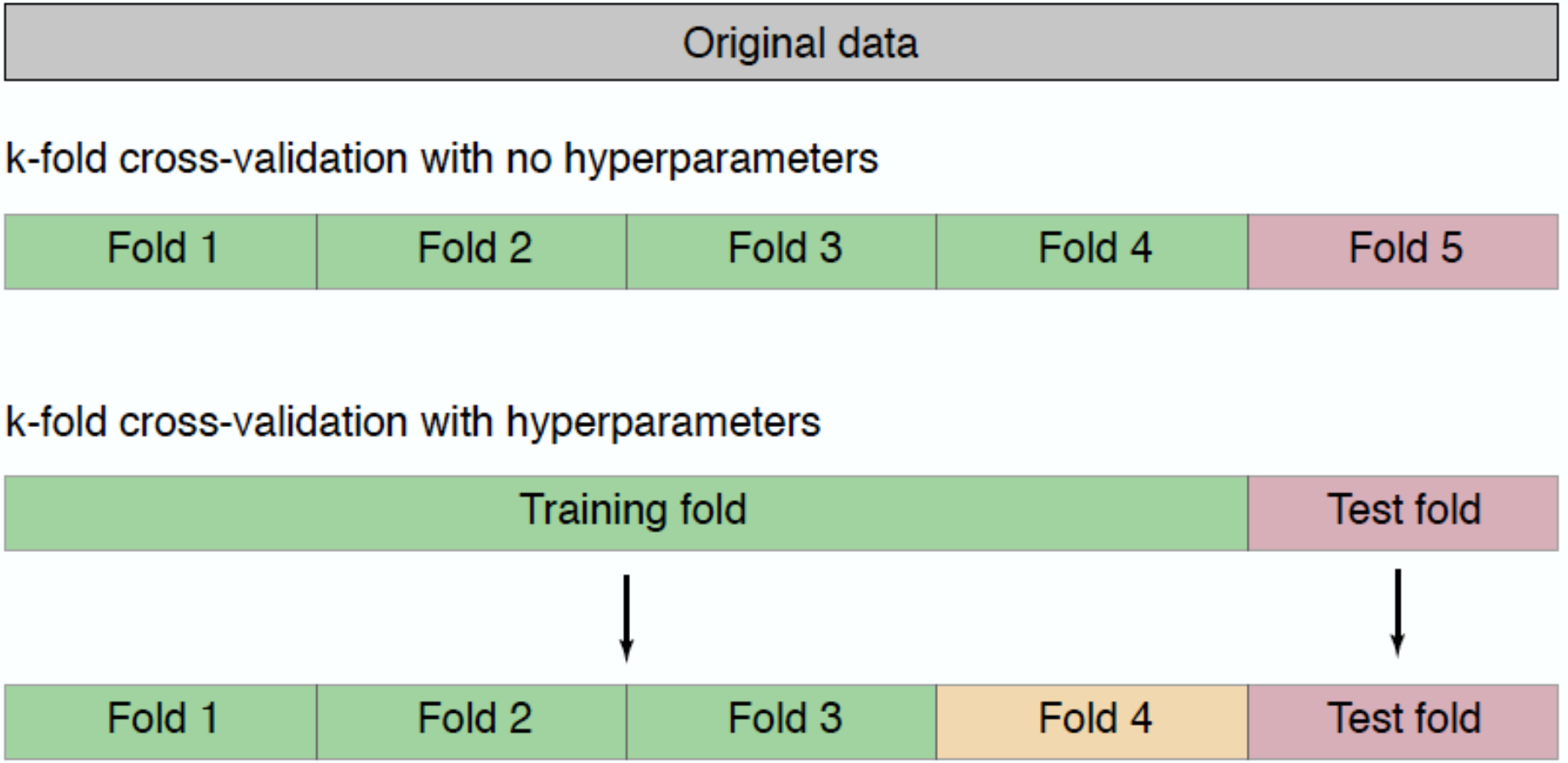


M=9: *overfitting*



More complex features lead to better results on the training data, but potentially worse results on new data, e.g., test data!

K-fold Validation for Hyperparameter Tuning



In This Lecture

- What is Machine Learning?
- Types of Learning
- Applications / Real-Life Examples
- How to Choose a ML Technique
- Detailed View of Few Approaches
- **Project**

Project 1 & 2

Part 1: K-means clustering

We will implement the K-means clustering algorithm to perform unsupervised clustering.

- 2-D data X and the label y_true is generated using sklearn
- Matplotlib is used to visualize the data and clusters
- KMeans from sklearn is called as a comparison
- You are required to write the Kmeans function:
  1. **Find_clusters**(X, n_clusters, max_iter, rseed=2)
  2. Get_labels(X, centers)
  3. Find_centers(X, centers, n_clusters, labels)
  4. Should stop(centers, new_centers, iteration, max_iter)

Project 1 & 2

Part 1: K-means clustering

We will implement the K-means clustering algorithm to perform unsupervised clustering.

- You are required to write the Kmeans function:
  1. **Find_clusters**(X, n_clusters, max_iter, rseed=2)
     a. Given data X, number of clusters, max iteration numbers, it should return the centers and data labels
  2. Get_labels(X, centers)
     a. Assign labels to data with current centers
  3. Find_centers(X, centers, n_clusters, labels)
     a. Find the new centers with current centers and labels
  4. Should stop(centers, new_centers, iteration, max_iter)
     a. Tell if the Kmeans algorithm should stop

Project 1 & 2

Part 2: KNN classification

We will implement a KNN classifier and test it on generated data. (*Extra question: classify MNIST digits with KNN)

- Read the codes of KNN class: train, compute_distances, compute_L2_distances_vectorized, predict_labels
- Write down the missing codes:
  - Calculate distance between data samples
  - Calculate L2 distance in an efficient way (*extra question)
  - Predict labels using calculated distance
- Test KNN on previously generated data
- Test KNN on other dataset (MNIST), change the value of K and observe KNN performance.

Project 3: HTRU2 Data Set

- Data Set Information:
- HTRU2 is a data set which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey (South).
- Pulsars are a rare type of Neutron star that produce radio emission detectable here on Earth. They are of considerable scientific interest as probes of space-time, the inter-stellar medium, and states of matter.
- As pulsars rotate, their emission beam sweeps across the sky, and when this crosses our line of sight, produces a detectable pattern of broadband radio emission. As pulsars rotate rapidly, this pattern repeats periodically. Thus pulsar search involves looking for periodic radio signals with large radio telescopes.

Project 3

- Each pulsar produces a slightly different emission pattern, which varies slightly with each rotation. Thus a potential signal detection known as a 'candidate', is averaged over many rotations of the pulsar, as determined by the length of an observation. In the absence of additional info, each candidate could potentially describe a real pulsar. However in practice almost all detections are caused by radio frequency interference (RFI) and noise, making legitimate signals hard to find.
- Machine learning tools are now being used to automatically label pulsar candidates to facilitate rapid analysis. Classification systems in particular are being widely adopted, which treat the candidate data sets as binary classification problems. Here the legitimate pulsar examples are a minority positive class, and spurious examples the majority negative class. At present multi-class labels are unavailable, given the costs associated with data annotation.

Project 3

- The data set shared here contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators.
- Candidates are stored in both files in separate rows. Each row lists the variables first, and the class label is the final entry. The class labels used are 0 (negative) and 1 (positive).
- Please note that the data contains no positional information or other astronomical details. It is simply feature data extracted from candidate files using the PulsarFeatureLab tool.

Project 3

- Goal
- Split the dataset into training and test subsets
- Balance the data with positive and negative samples
- Train a KNN classifier and evaluate your model using test set
- Train a SVM classifier and evaluate it
- This project will emphasize on the usage of sklearn packages

Acknowledgement

- Theses slides are adapted from LACC material last year and are used exclusively for Los Angeles Computing Circle. No unauthorized distribution or commercial use is allowed.
- The instructors wish to thank Hemant Saggar for the 2018 version slides, as well as the material authors that are mentioned when their materials are used.

Acknowledgements

Part of these slides comes from the Lecture Notes of Prof. Suhas Diggavi's "ECE239AS - Foundations of Statistical Machine Learning" and Prof. Jonathan Kao's "ECE239AS – Deep Learning and Neural Network". Great thanks to their efforts.

Further Learning

"Machine Learning" - Coursera online course by Prof. Andrew Ng.
https://www.coursera.org/learn/machine-learning