Machine Learning & Artificial Intelligence

Instructor: Ziqi Wang and Tianwei Xing

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.

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

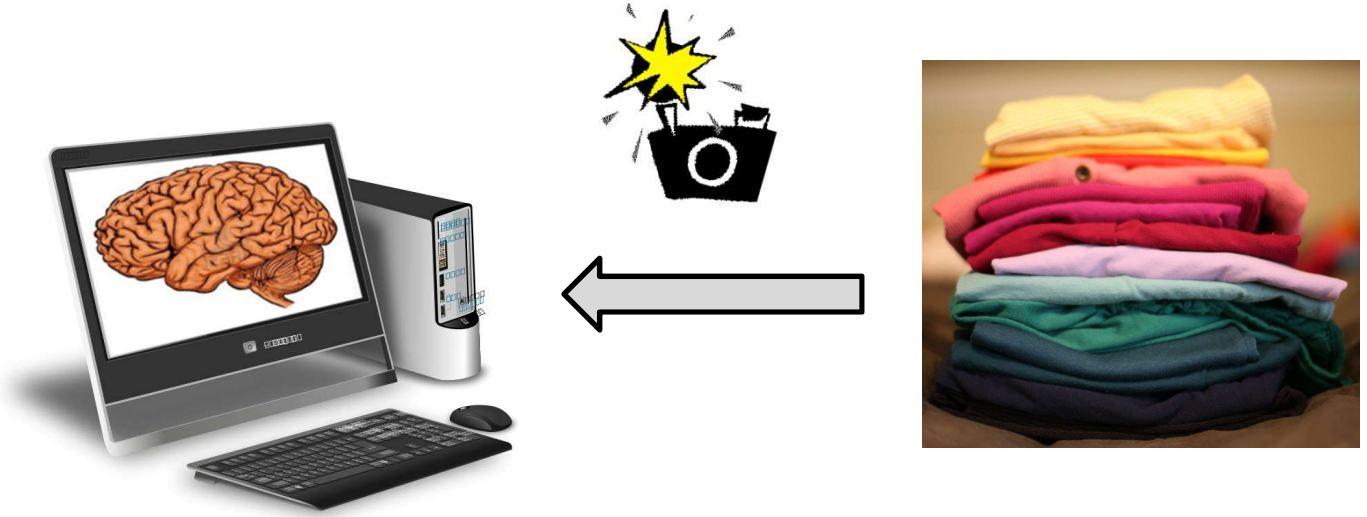
- **What is Machine Learning?**
- Types of Learning
- Applications / Real-Life Examples
- Overview of Various Approaches
- Detailed View of Few Approaches
- Project

What is Machine Learning: An Example

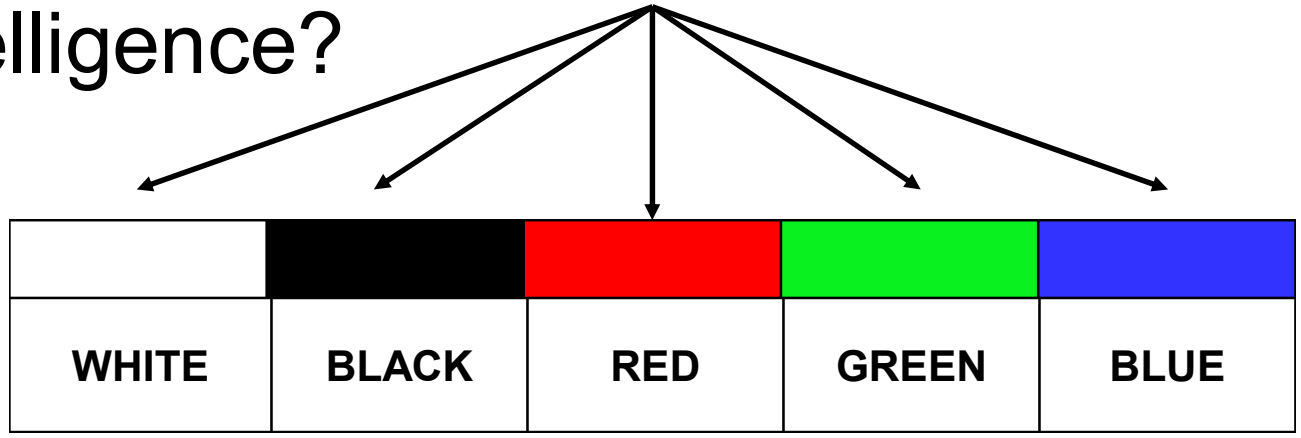
A Laundry company wants to separate washable clothes into five piles of separate color. It puts a **computer** in charge of separating the clothes into different piles.
How does the computer learn how to group the clothes?

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

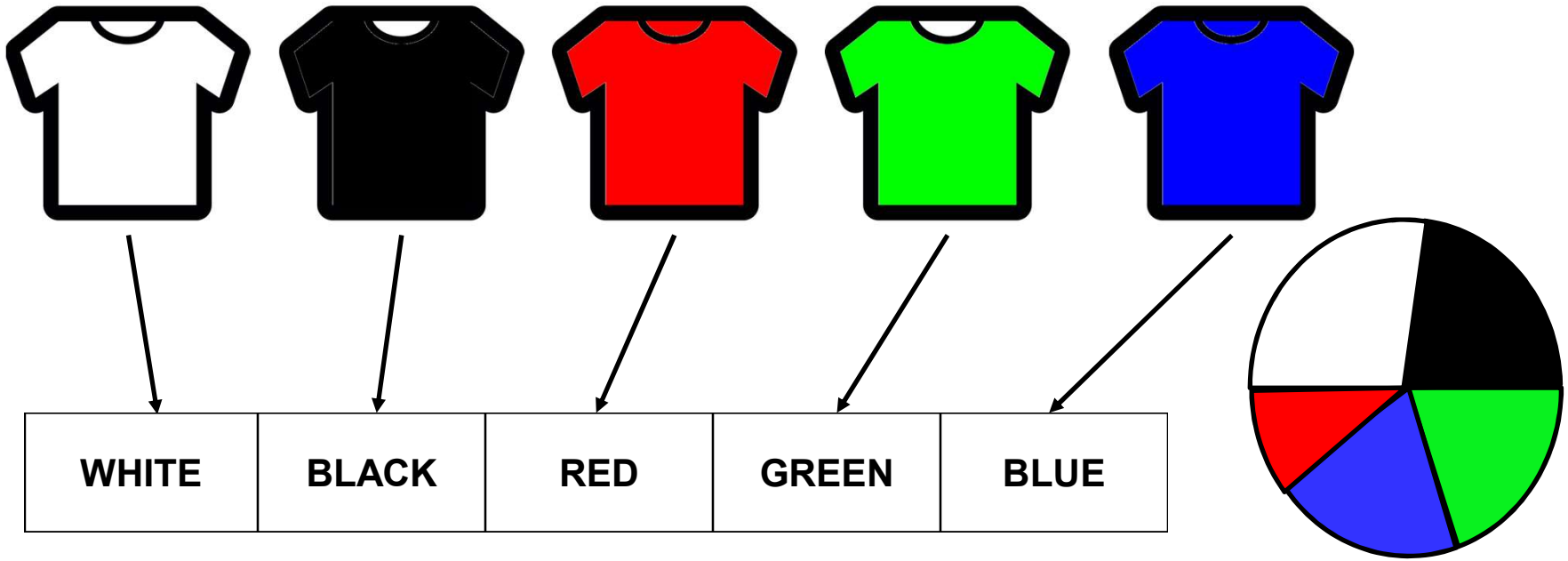
1. In many problems, the relationships are either just too many to write down as **if** *this* **then** *that* OR not even understood properly.

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?

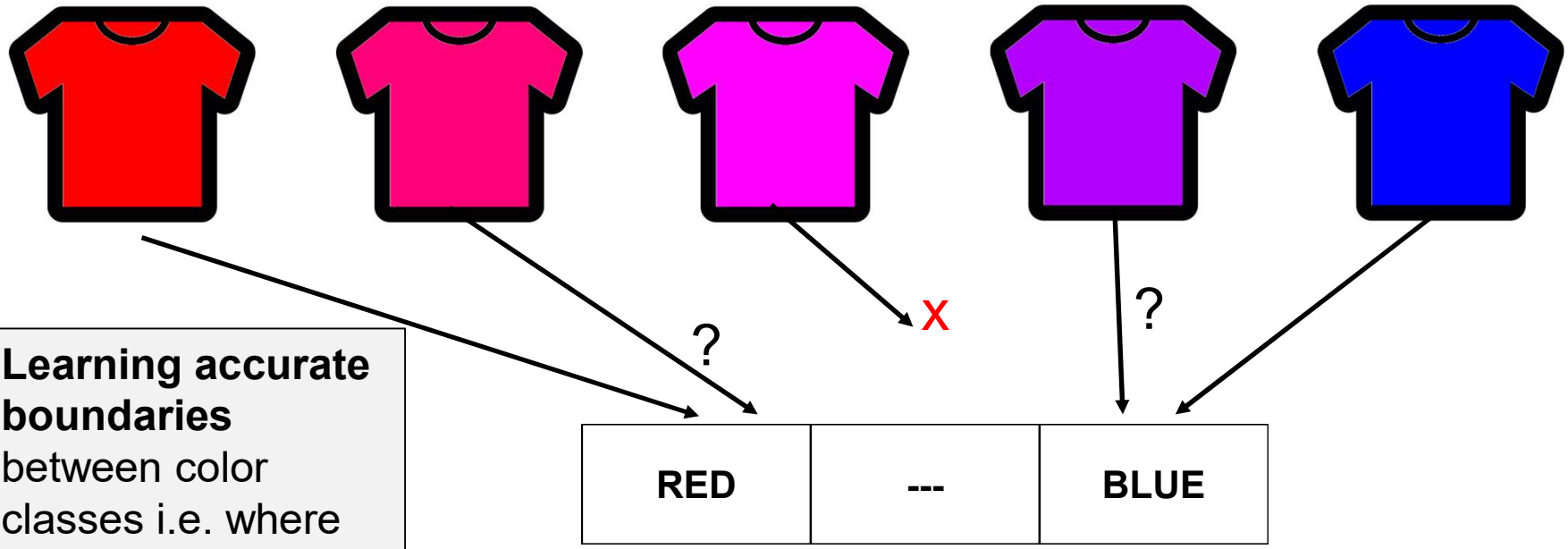
- Represent all the inputs as points in a coordinate (x,y,z,…) system. (Eg ?)
- Define a function representing a score for each class for each input. (Eg.?)
- Classify/assign class to input to the maximum scoring class.



Initial learned boundaries

What does a machine learn

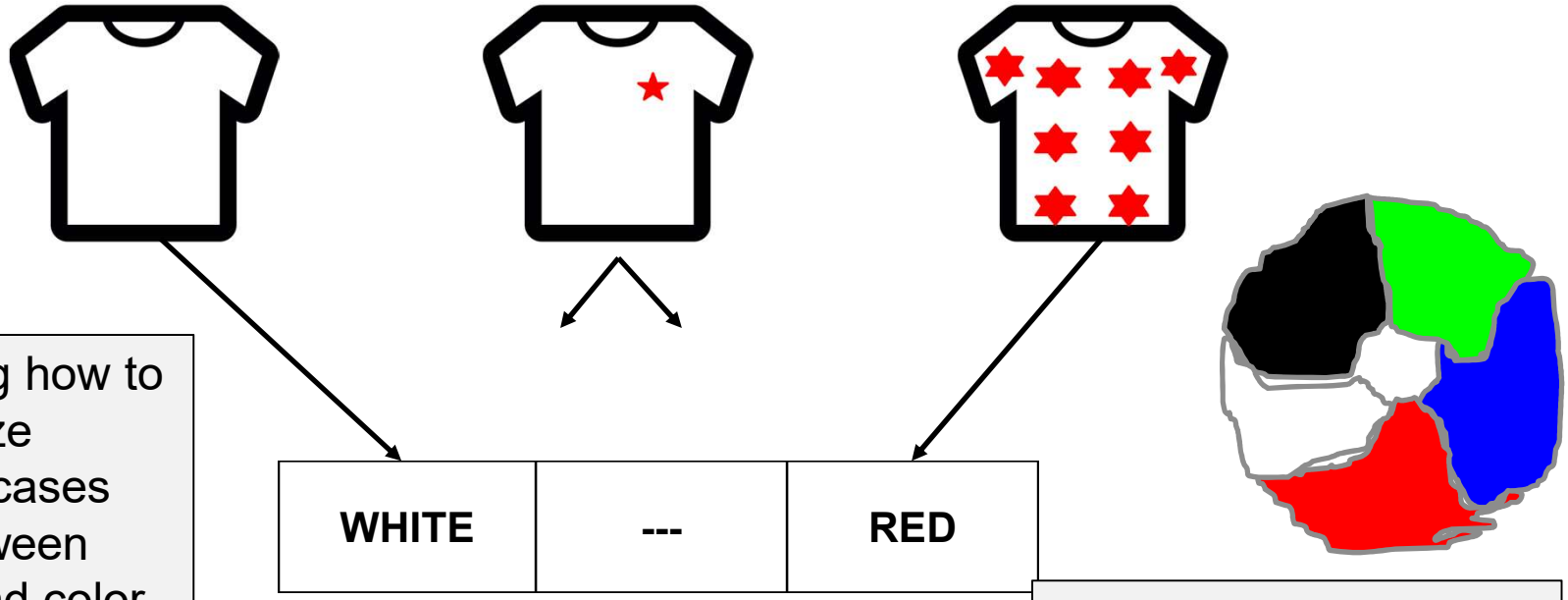
More training = better generalization .
Now the machine has learnt a basic rule (score based model) to classify into 5 classes.
Test if the machine classifies **new** (unseen) clothes correctly.



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.

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).*

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.

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.

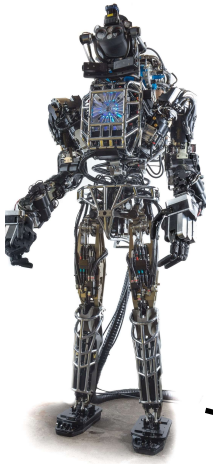
Pop up Question

- 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]

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

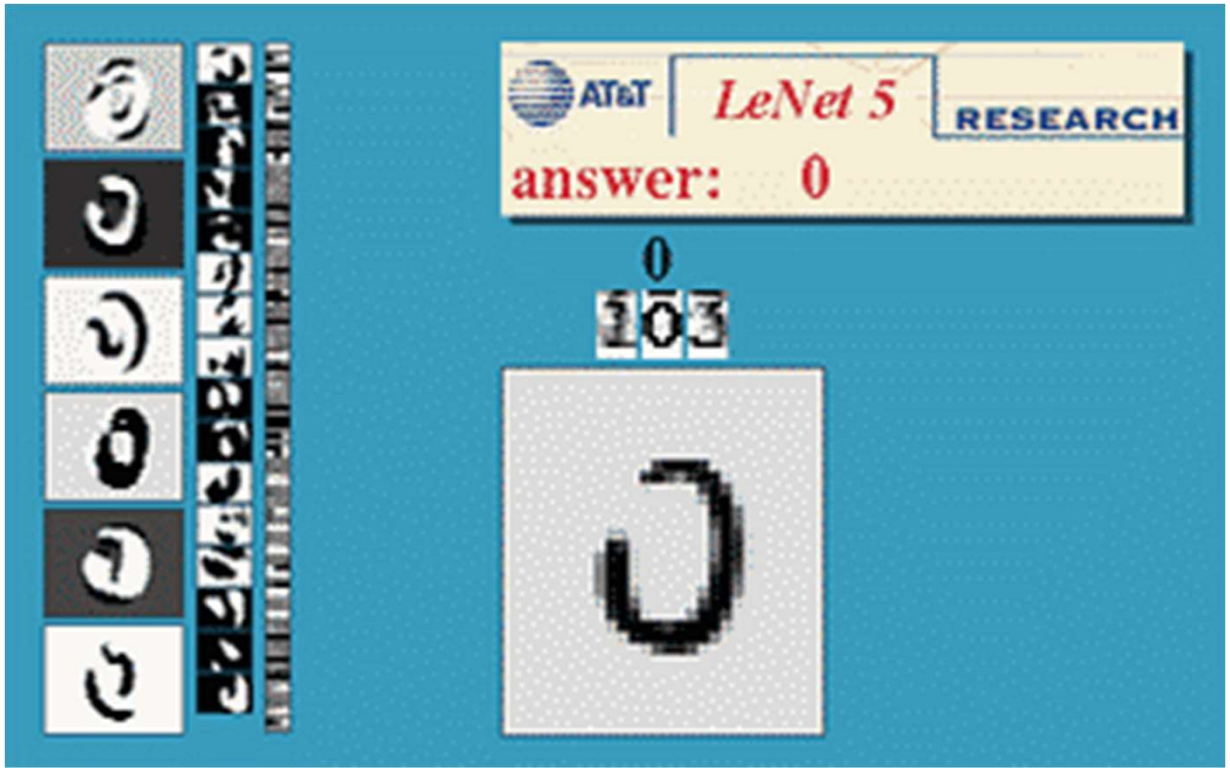
Personal assistants: Google Now, Microsoft Cortana, Apple Siri, etc.

Google Ads

Advertising and business intelligence

Autonomous ("self-driving") vehicles

What is going on? Circa 1989



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

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

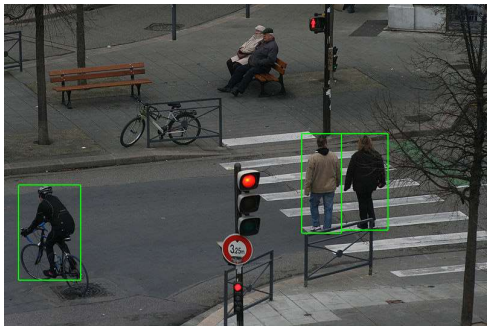
- Numerous learning algorithms have been invented in the last 30 years that are more powerful than the previous in learning complex patterns in data.

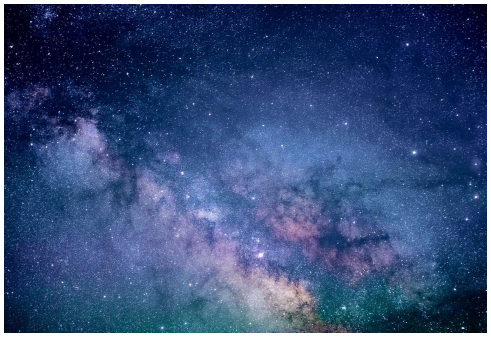
- Some of the recent algorithms that have been invented and are being used are
  - Support Vector Machines
  - Decision Trees and Random Forests
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Deep Neural Networks

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

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)
- **Prediction**
  - Housing price estimate (Zillow)
  - Stock market
- **Medical Diagnosis**

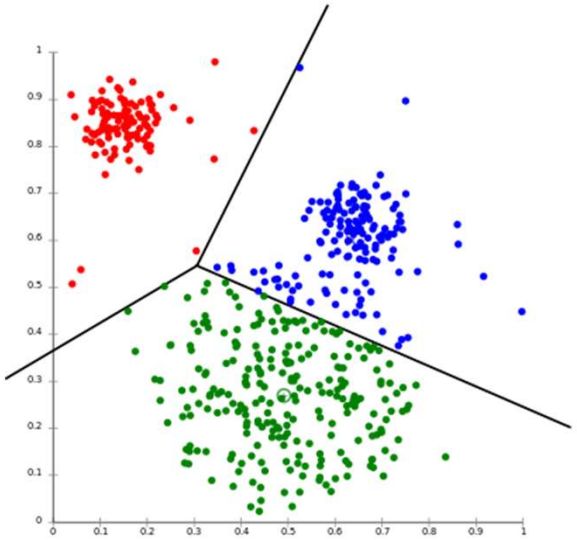
Types of Learning

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



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

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

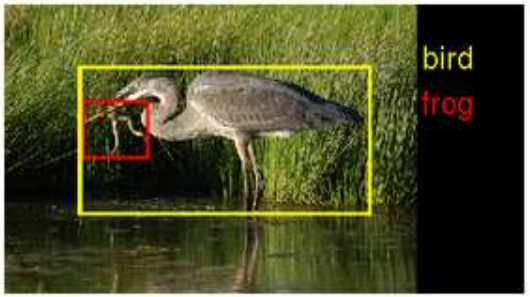
- Also common: **facial recognition**:
  - Think of a security system that has a database of employee photos. Each time a person enters the building, the classifier examines a snapshot.

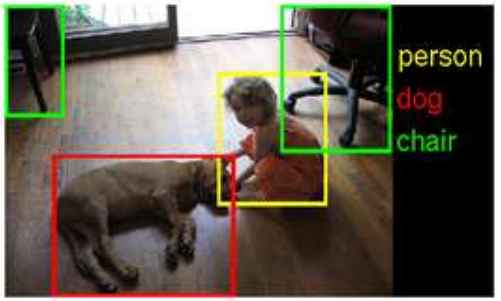
  - Goal: identify the current person as one of the employees, or an unknown person.

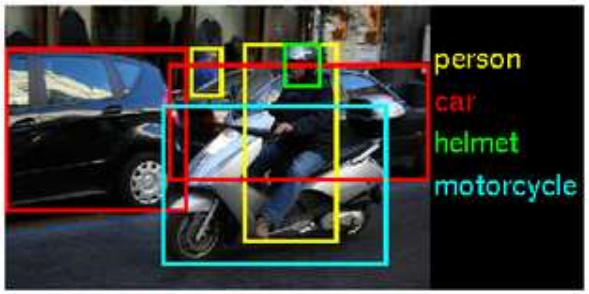
  - Challenge: facial portraits vary a lot.

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

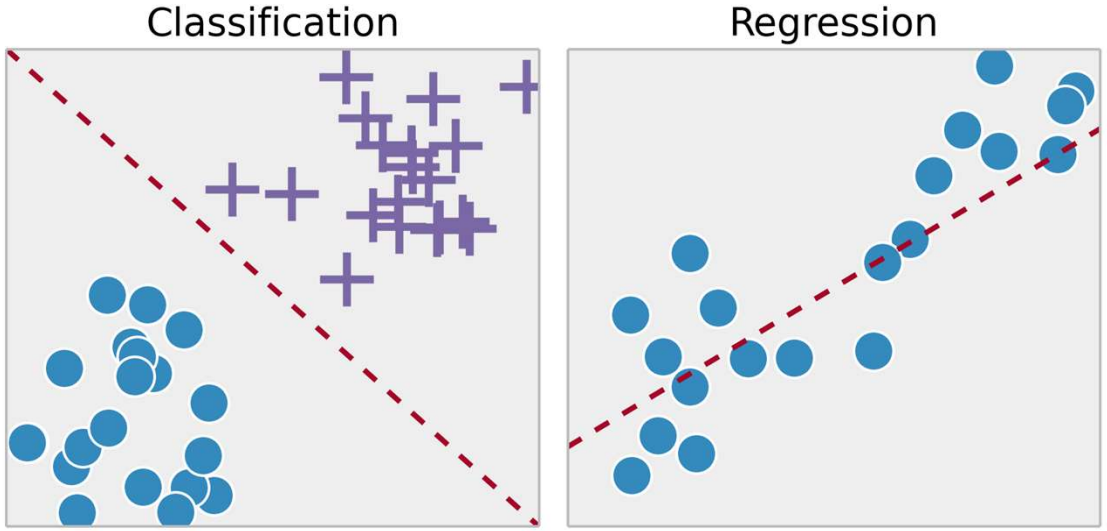
Prediction/ Regression

- 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&

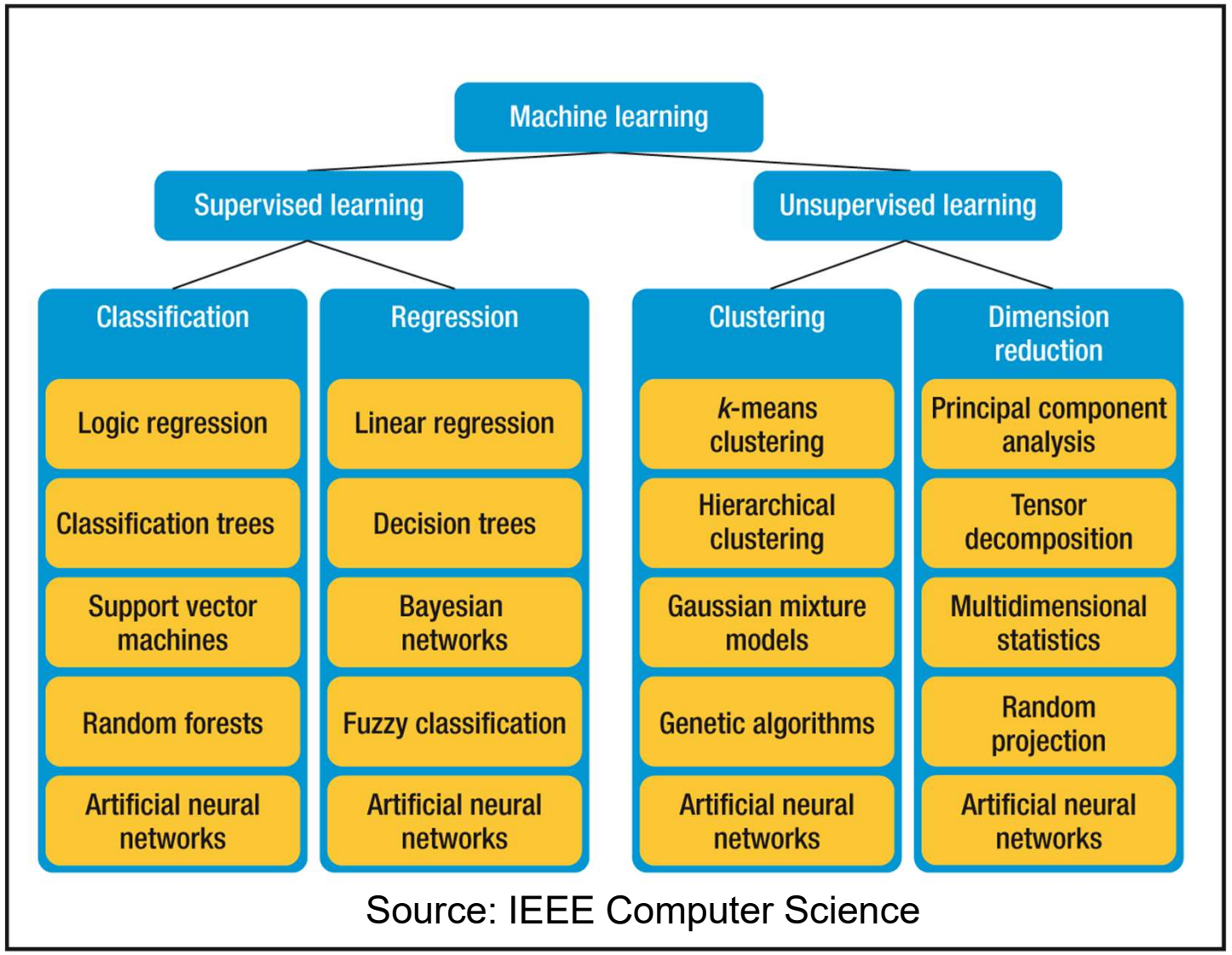
- More levels:
- www.youtube.com/watch?v=iakFfOmanJU

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

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

What Machine Learning Techniques Exist?

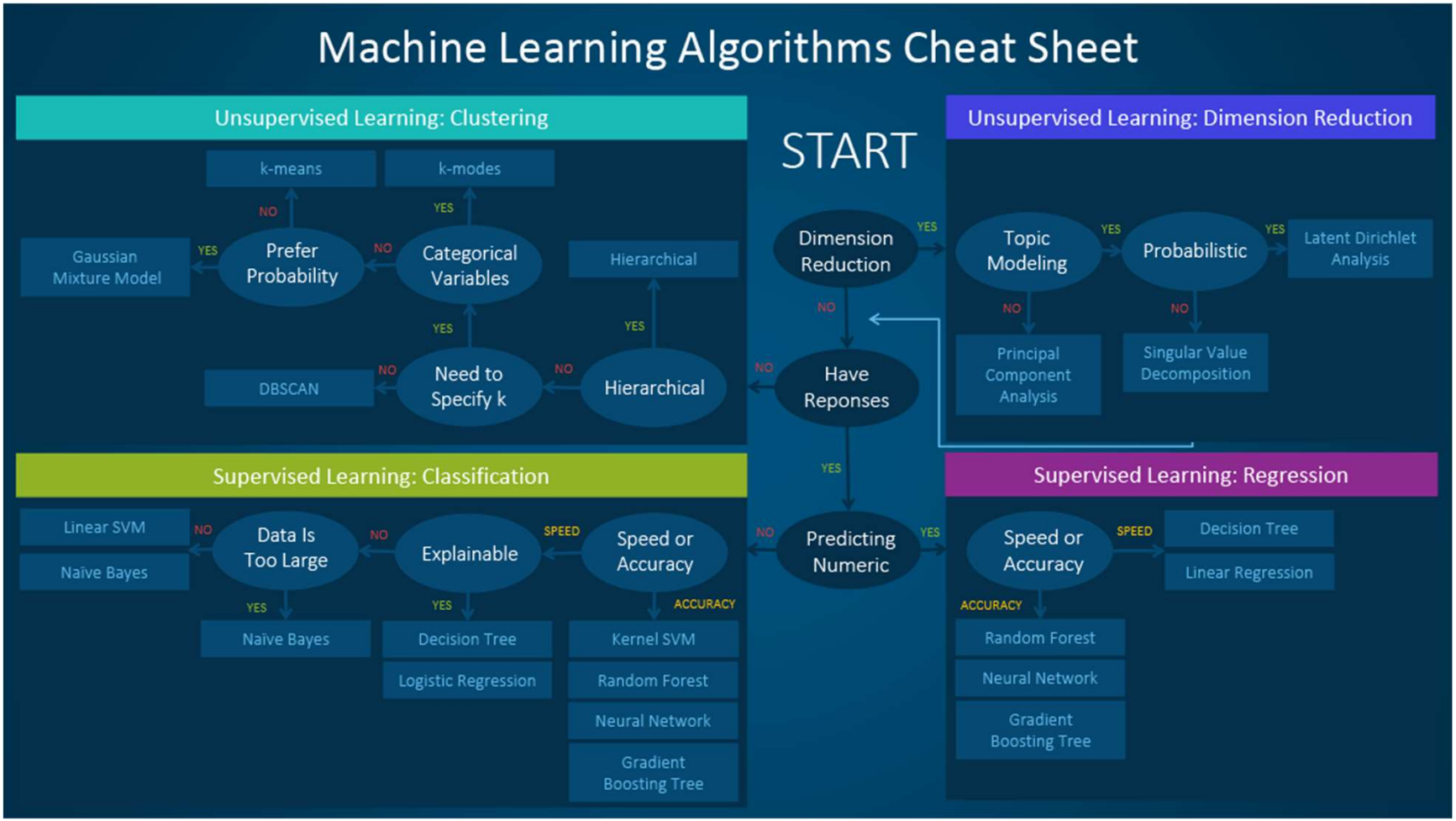


Source: IEEE Computer Science

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



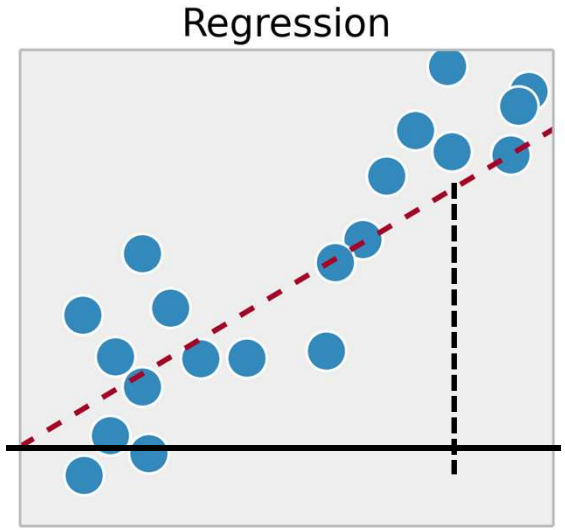
In This Lecture

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

Linear 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	?

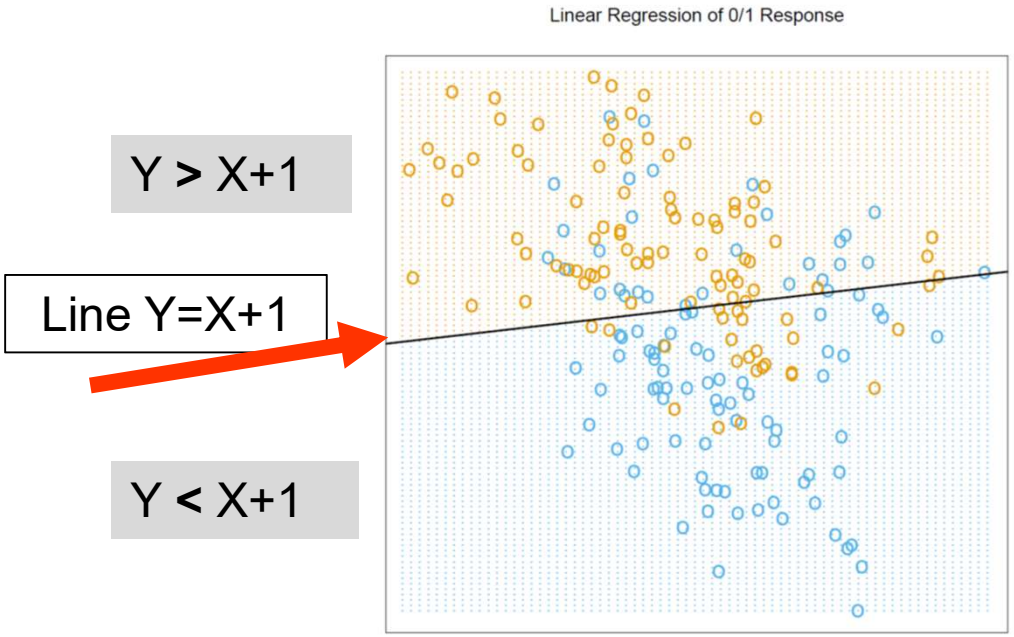


Linear Regression

- 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 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 $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots, \beta_{N-1}]$.

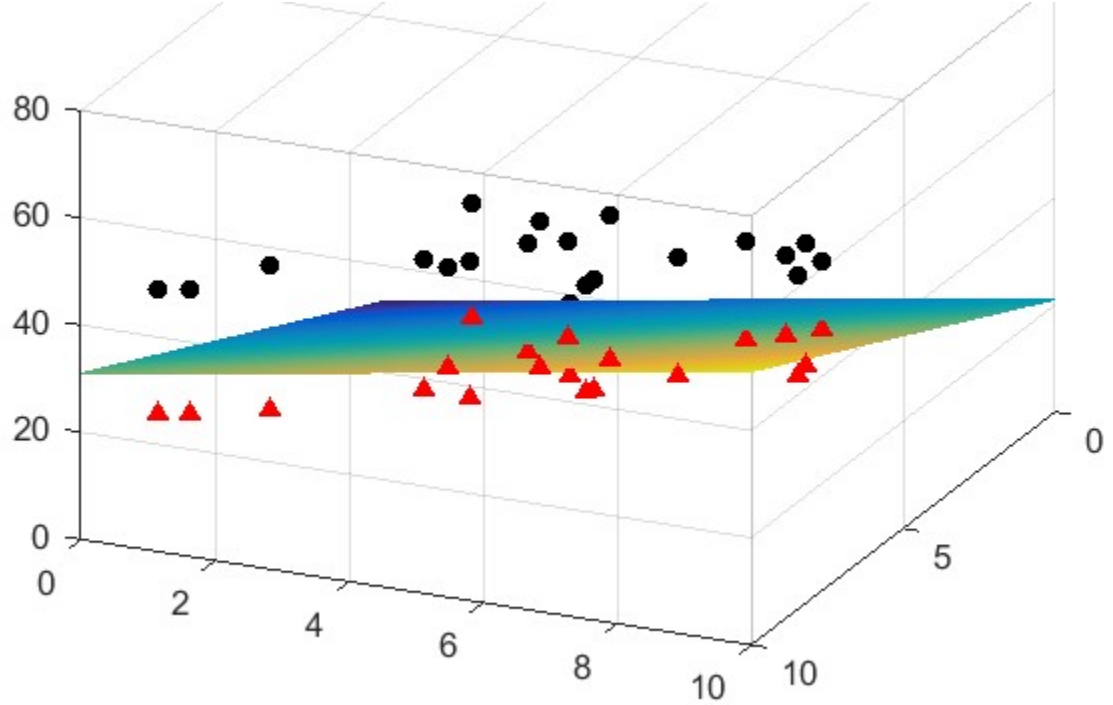
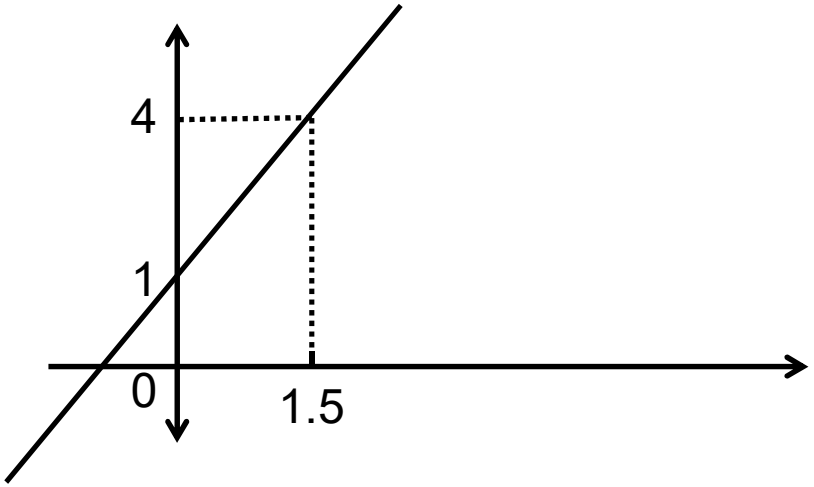


Fig: A linear boundary in three dimensions separating two classes, **red** and **black**.

Question



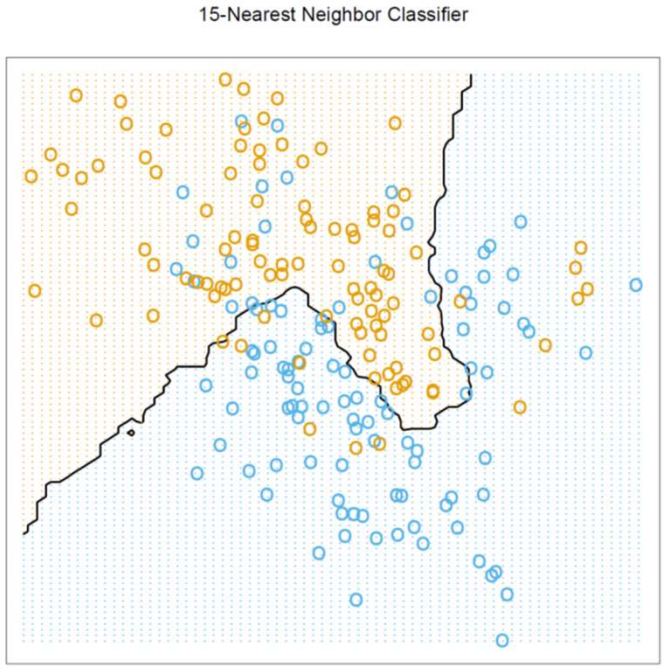
- What is the equation of this line as in ax+by = c?
- Do points (5, 12) and (1.5, 3) lie on the same or opposite side of the line?

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?

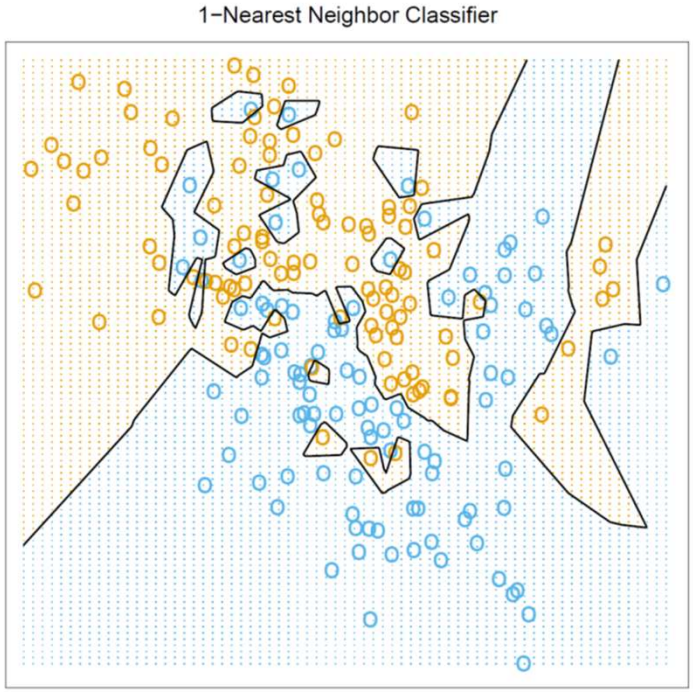


- Decision boundary is not a nice line anymore, it's a squiggly border. Expected?

- 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?

Linear versus nearest neighbors

- So what's the difference between these two approaches?
  - Linear makes many assumptions about problem structure (linearity!)
  - But it's stable – remember our simple decision boundary
  - K-nearest neighbors doesn't have strict assumptions about the problem
  - Very unstable predictions though!
- In general, we have to figure out something about the problem before we know which ML algorithm is appropriate.
- These two algorithms are very fundamental, but there are tons of others also.

Bayesian Classifiers

- Before we talk about this type of classifier, we need some background from probability
- Notice the big connection between machine learning and ideas from **statistics** and **probability**!
- We need a concept called **conditional probability**. If A and B are events, the conditional probability of A given B is called P(A|B) and is given by

$$P(A|B) = P(A \text{ and } B)/P(B)$$

  (this is called Bayes' rule)
- Let's try an example. If we're rolling a normal die, let's let A be the event we roll a 2 and B be the event we roll an even number.
- What are P(A), P(B), P(A and B), and P(A|B)?

Bayesian Classifiers

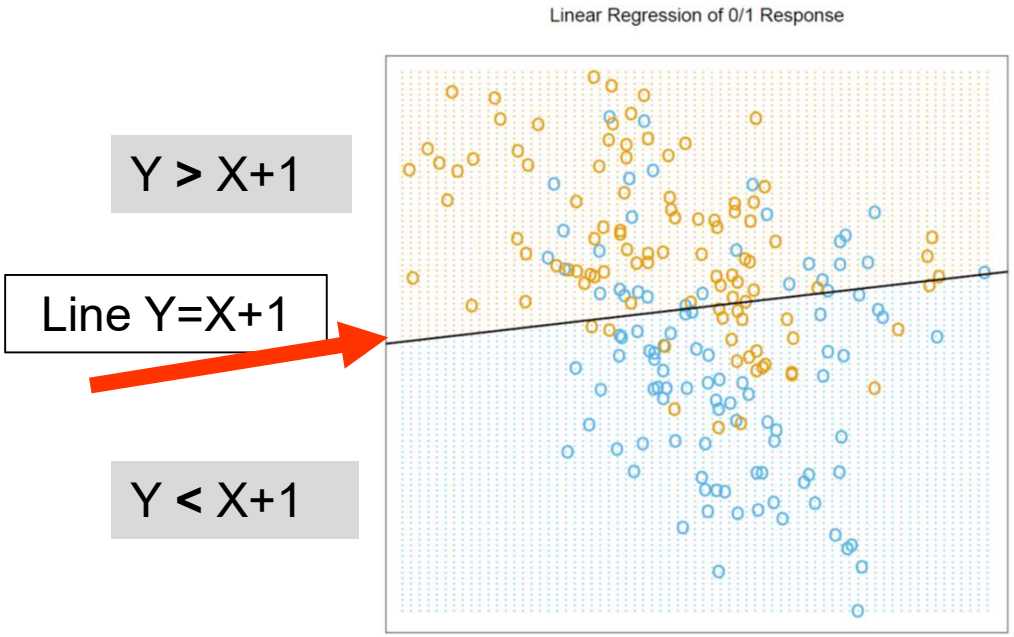
- Back to ML problems. Let's say we're trying to classify input $X=(X_1,\ldots,X_p)$ as being in one of the classes G_1, G_2, …, G_k.
- We should consider what's the **most likely** G that X can belong to.
- If we know that G_k is the correct output, what's the probability we get X?
- Answer: $P(X|G_k)$
- This is hard to compute, since $X=(X_1,\ldots,X_p)$. Let's make the assumption these are all (conditionally) independent! (Not necessarily a smart assumption -> **naïve Bayes classifier.**)
- Then, $P(X|G_k) = P(X_1|G_k)P(X_2|G_k) \ldots P(X_p|G_k)$.
- So, to classify X, let's look at $P(G_1)*P(X|G_1)$, $P(G_2)*P(X|G_2)$, …, and pick the **largest** one!

Naïve Bayes Classifier

- So, to classify X, let's look at $P(G_1)*P(X|G_1)$, $P(G_2)*P(X|G_2)$, …, and pick the **largest** one!
- Basically, we're picking the **most likely** class to produce X.
- Simple, but we still need to compute $P(X_1|G_k)$ and so on.
- This is the theme of our mini-projects.

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 $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots, \beta_{N-1}]$.

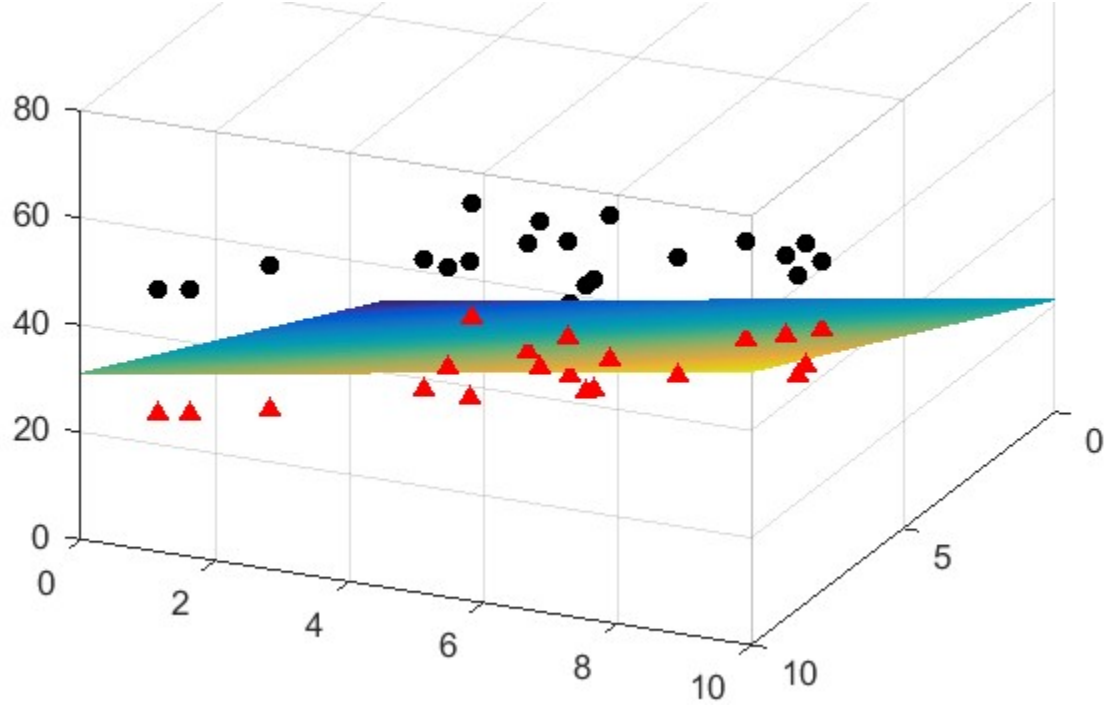


Fig: A linear boundary in three dimensions separating two classes, **red** and **black**.

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).

Linear Prediction

- One really important class of models are **linear** models i.e. f(x) is linear in x.
- Linear just means the output can be produced from the input with multiples and constant terms. Which of these functions are linear?

$$f_1 = x_1 + x\_2$$

$$f_2 = 2(x_1)^2$$

$$f_3 = 13+2x_1+8x_2+9x_3$$

$$f_4 = \log_2(x_1)+3x_3$$

- Let's say the input variables x are vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$
- Then, we predict the output with the linear model:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

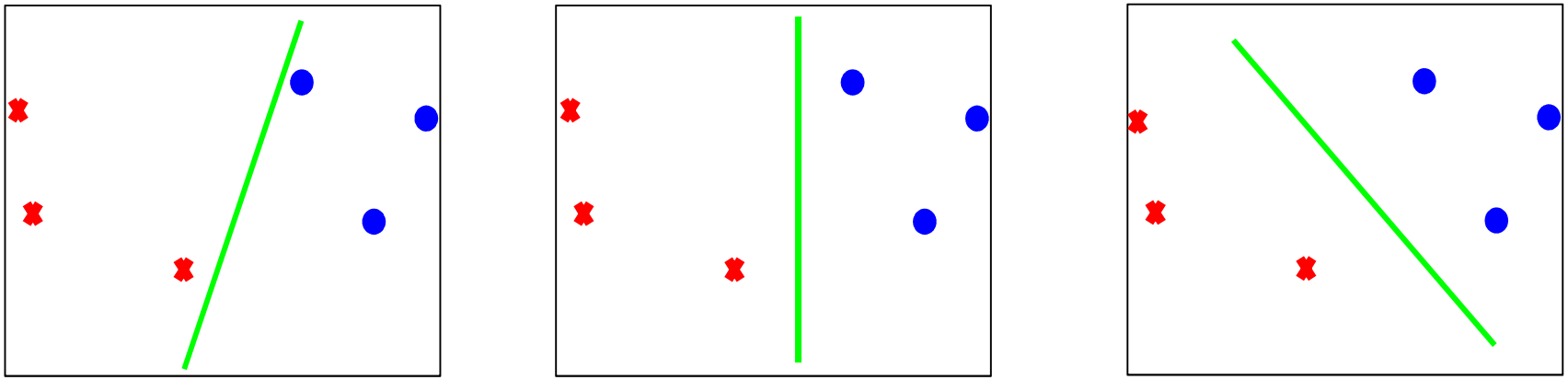
Vector and Matrix Refresher

- A vector is a collection of numbers written like $\boldsymbol{y} = (y_1, y_2, y_3, \ldots, y_N)$.
- This is a row vector.
- We can also transpose this to get a column vector $\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_N \end{pmatrix}$.
- For example latitude,longitude can be written as a vector $\begin{pmatrix} 34.0689 \\ 118.4452 \end{pmatrix}$.
- Vectors can be added and subtracted. Eg: $\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$.
- Two vectors can be multiplied in a special way called inner product. Eg: $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \mathrm{y} = \begin{pmatrix} -1 \\ 4 \end{pmatrix}$. $Inner\ prod. = \boldsymbol{x}^T\boldsymbol{y} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}^T \begin{pmatrix} -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} -1 \\ 4 \end{pmatrix} = -1 + 8 = 7.$
- Norm of a vector = $||y|| = \sqrt{y_1^2 + y_2^2 + y_3^2 + \cdots + y_N^2}$

Matrix

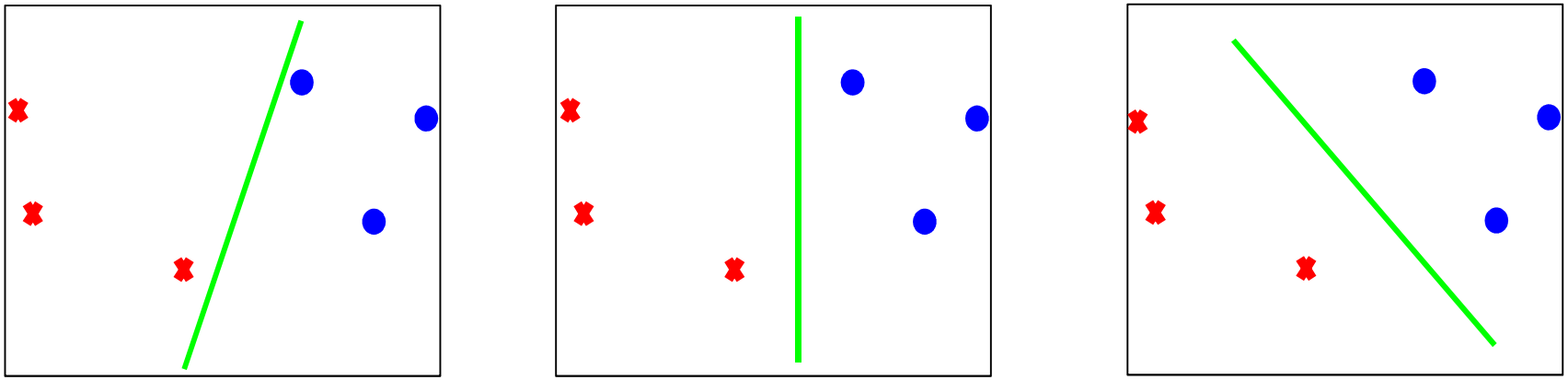
- A Matrix is just many row vectors of the same length stacked one below the other, OR, many column vectors of the same length stacked side by side.
- Eg: $\mathrm{A} = \begin{pmatrix} 1 & 5 \\ 2 & 7 \end{pmatrix}$ is a matrix. A matrix is usually denoted by a capital letter.
- Matrices can be added and subtracted like vectors.
- Matrices can be post or pre-multiplied with vectors. Eg: $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \mathrm{y} = \begin{pmatrix} -1 \\ 4 \end{pmatrix}$, $\mathrm{A} = \begin{pmatrix} 1 & 5 \\ 2 & 7 \end{pmatrix}$.
- $Ax = \begin{pmatrix} 1 & 5 \\ 2 & 7 \end{pmatrix}$

Support Vector Machines



Which separating hyperplane is better?

Support Vector Machines

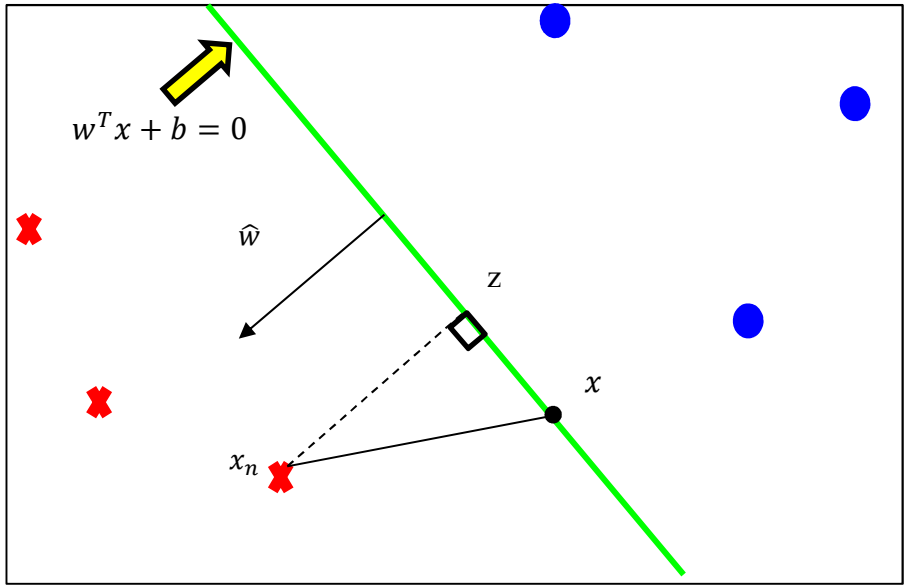


- Sometimes there are multiple linear classifiers.
- The most robust classifier will be one that maximally separates the data.
- We want a classifier that has the maximum margin.
- Margin = perpendicular distance of a point to the hyperplane.

Support Vector Machines

- Consider a line (hyperplane) $w^T x + b = 0$.
- w vector is perpendicular to the plane. How?
  - $w^T x + b = 0$
  - $w^T z + b = 0$
  - Subtract .. $w^T (x - z) = 0$
- Distance from x_n to the line is the shadow of $x_n - x$ on $\hat{w}$ i.e.

$$\hat{w}^T (x_n - x) = \frac{w^T (x_n - x)}{||w||}$$

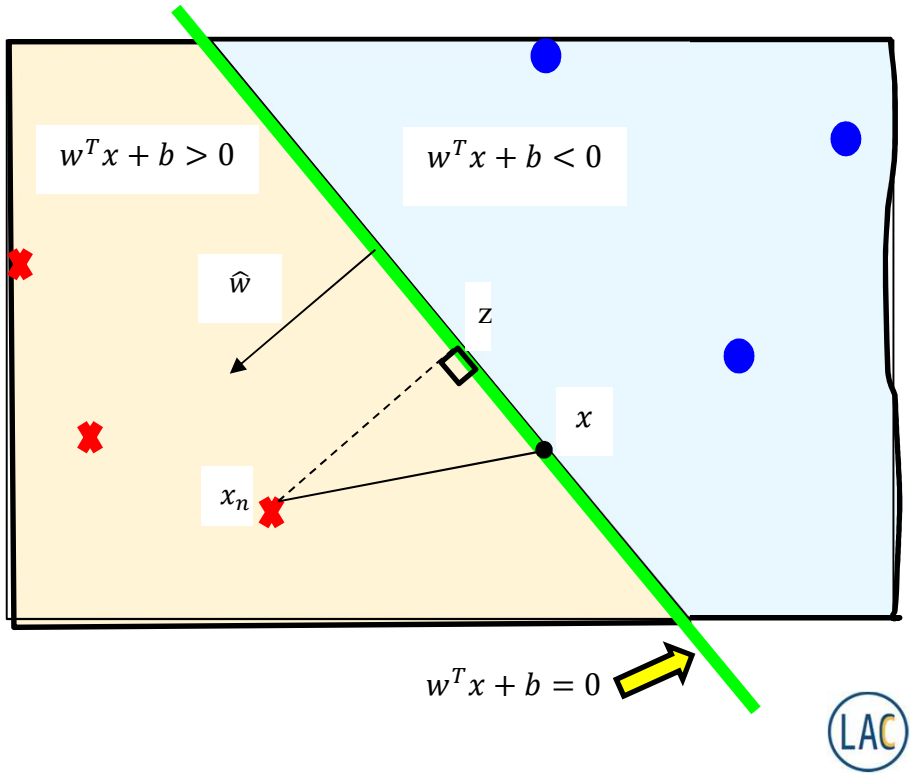


$$\frac{w^T (x_n - x)}{||w||} = \frac{w^T x_n - w^T x}{||w||} = \frac{w^T x_n + b - (w^T x + b)}{||w||} = \frac{w^T x_n + b}{||w||}.$$

- The margin does not change if I scale w and b by a constant K! So I am free to choose an arbitrary constant K. I choose K such that $w^T x_n + b = 1$.

Support Vector Machines

- Consider two **linearly separable** datasets.
- That is a hyperplane $w^T x + b = 0$ exists such that
  - For points with class $y_n = 1$
    - $w^T x_n + b > 0$, and
  - For points in class $y_n = -1$,
    - $w^T x_n + b < 0$.
- Thus $|w^T x_n + b| = y_n(w^T x_n + b)$.

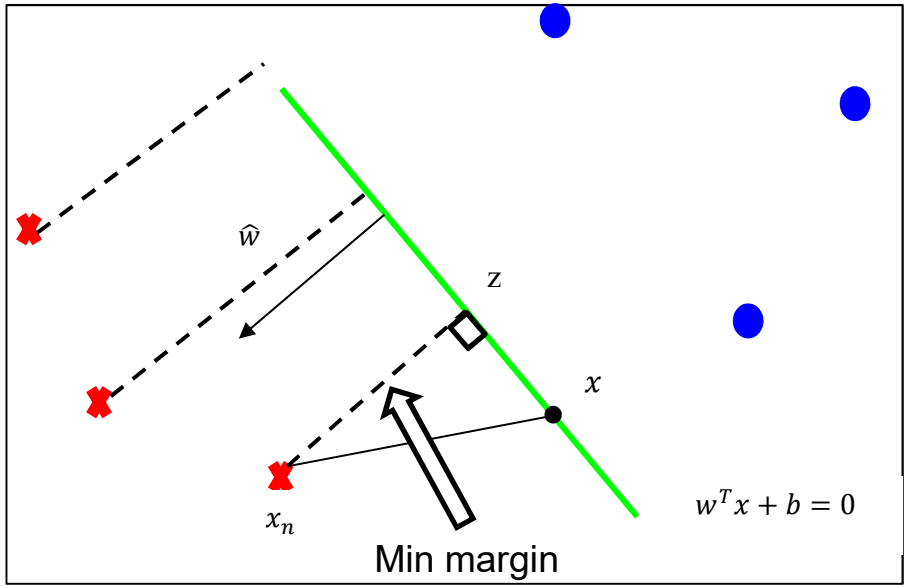


Support Vector Machines: Margin

- The abs perp. distance of x_n from the hyperplane is $\frac{y_n(w^T x_n + b)}{||w||}$.

- This is called the margin of point x_n. Why?

- I wish to pick the point with the minimum margin and have that margin as large as possible.

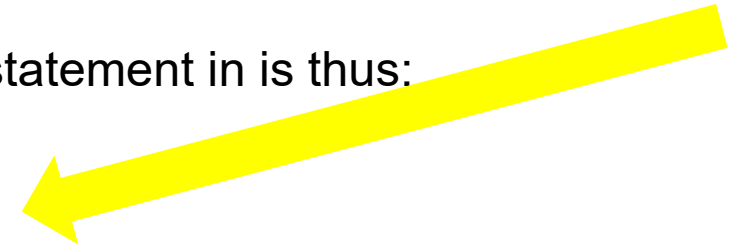


- If a solution w_0, b_0 gives margin M, then what is the margin of the solution $2w_0$, $2b_0$? $0.5\ w_0$, $0.5\ b_0$?

- I have flexibility in choice of w and b up to a constant…

SVM Problem Statement

- I choose w and b such that for the minimum margin point $\mathrm{y_n}(w^T x_n + b) = 1$.
- Hence the minimum margin = $\frac{y_n(w^T x_n + b)}{||w||} = \frac{1}{||w||}$
- The SVM problem statement in is thus:



Maximize $\frac{1}{||w||}$

w.r.t. w, b

Subject to $y_n(w^T x_n + b) \geq 1$, for all data points n (because minimum is 1)

OR equivalent problem:

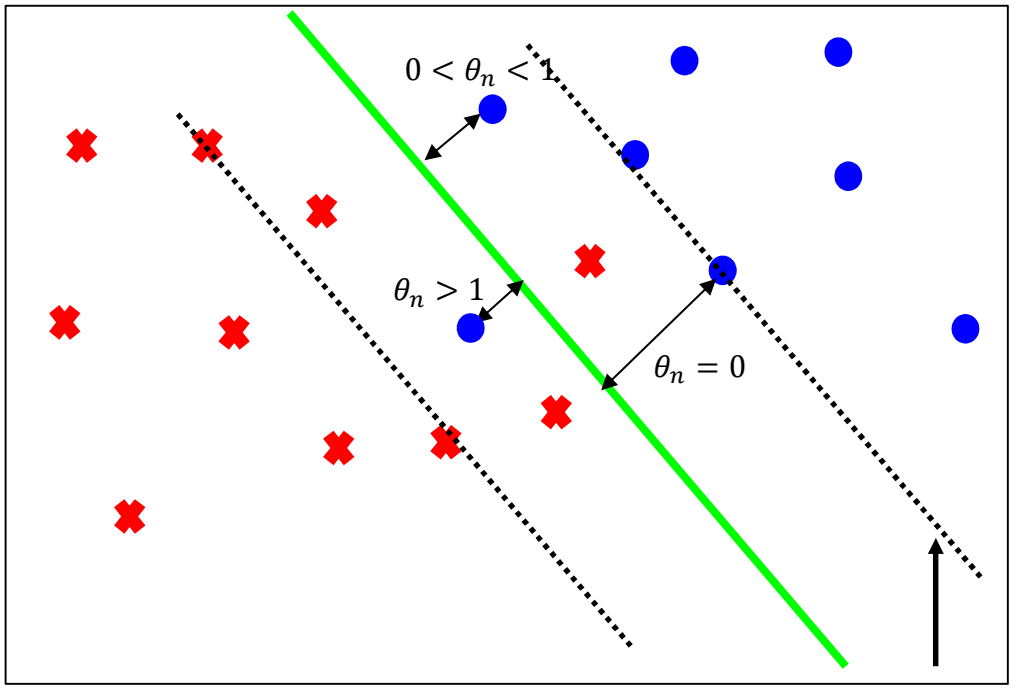
Minimize $||w||^2$

w.r.t. w, b

Subject to $y_n(w^T x_n + b) \geq 1$, for all data points n

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 Problem Statement

- If $1 - y_n(w^T x_n + b) \leq 0$ i.e. point achieves or exceeds the margin the $\theta_n = 0,$
- Else $\theta_n = 1 - y_n(w^T x_n + b)$.
- Thus $\theta_n = \max\big(0, 1 - y_n(w^T x_n + b)\big)$.
- We still want to minimize θ_n

SVM Formulation (Binary Classification)

Minimize $||w||^2 + C \sum_{n=1}^{N} \max\big(0, 1 - y_n(w^T x_n + b)\big)$
w.r.t. w, b, θ_n

C = user chosen parameter

Extending SVM from 2 to K classes

- For 2 classes, if $w^T x_n + b > 0$ we said point is in class +1 and if $w^T x_n + b < 0$ point is in class -1.
- For multiple classes this won't work…We define K hyperplanes $\{w_1, b_1\}, .., \{w_K, b_K\}$.
- Now think of $w_i^T x_n + b_i$ as being the score of point x_n for class i, not just +-.
- Point x_n belongs to class $\mathrm{y_n} = i$ iff for x_n -> score of class i is greater than the score of all other classes by at least a margin 1.

SVM Formulation (Multiple Class Classification)

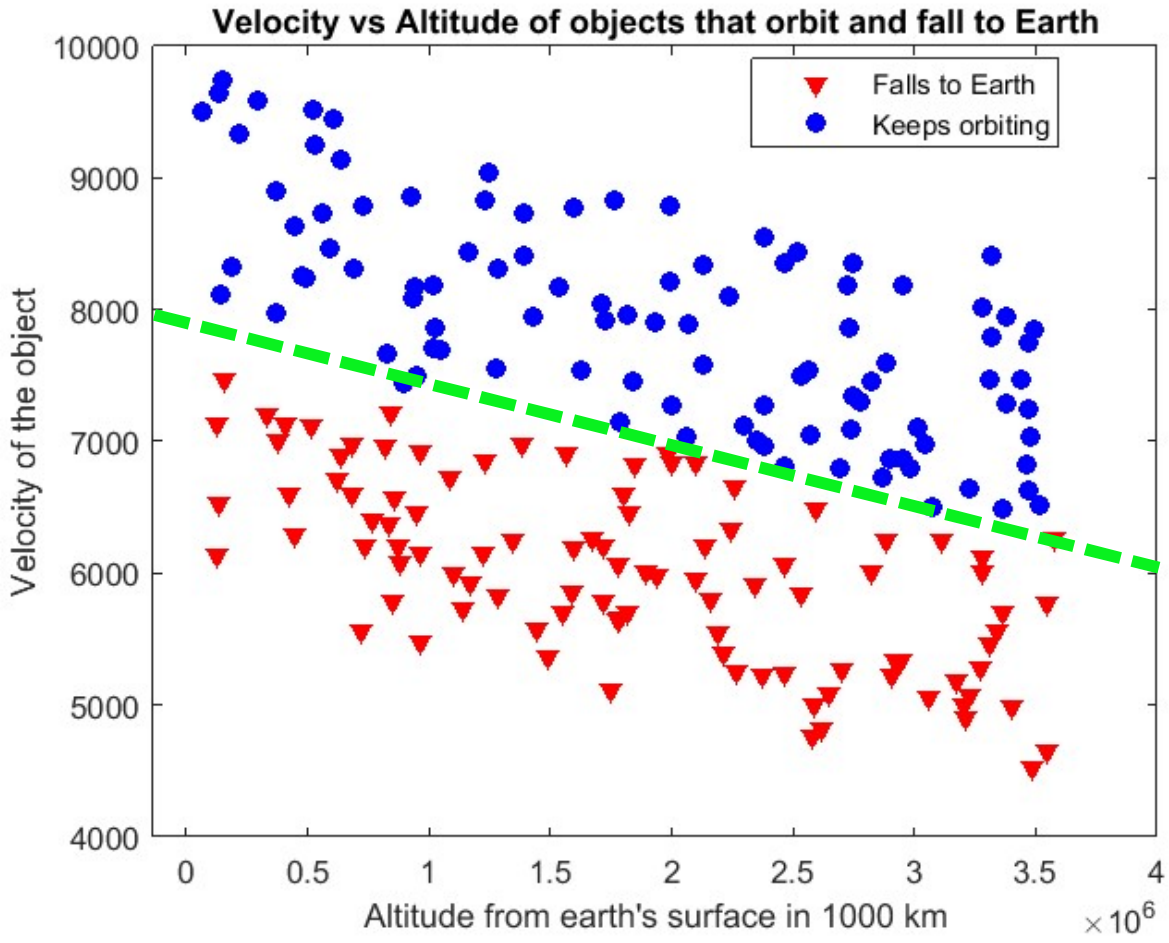
Minimize $||w||^2 + C \sum_{n=1}^{N} \sum_{\substack{j=1 \\ j \neq i \\ i = y_n}}^{K} \max\left(0,\ \underbrace{1}_{\text{Margin}} + \underbrace{w_j^T x_n + b_j}_{\text{Class j score}} - \underbrace{\left(w_i^T x_n + b_i\right)}_{\text{Class i score}}\right)$

w.r.t. w, b, θ_n

C = user chosen parameter

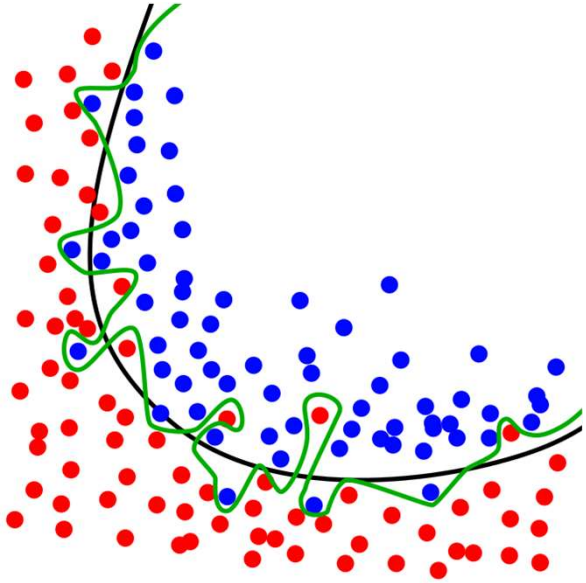
LAC

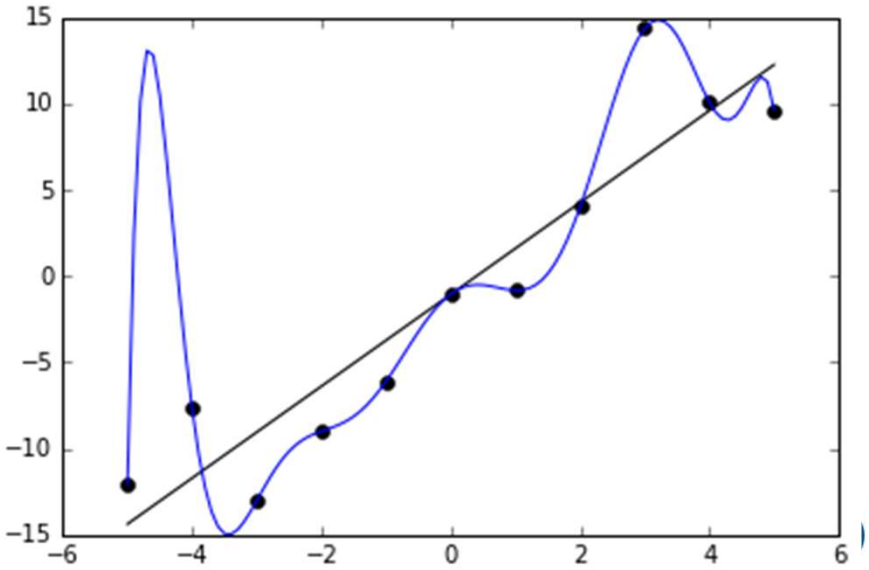
SVM Example with Code



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**.





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**

Mini-Project 1: language classification

- We will implement a text classifier that predicts whether text is in Spanish or French.
  - 1. You are given two sets of training data: french.txt and spanish.txt.
  - 2. You must implement a python function that reads the training files and splits them up into separate words:

    def train(training_set)

  - 3. You should return a dictionary, which contains **keys** and **values**. The keys are words and the values will be their frequency in the training set. So, if “beaucoup” appears in the French training set 4 times, the dictionary should have the pair (beaucoup, 4). Also store the total # of words.

Mini-Project 1: language classification

- We will implement a text classifier that predicts whether text is in Spanish or French.
  - 4. You must implement the following classification function:

    def classify(input_phrase)

  - 5. It should return either 0 (for French) or 1 (for Spanish). The input is a phrase. To test your classifier, try the following: “Faut-il aller au restaurant aujourd'hui?" Also, try the single word « vive ».

  - 6. Once your project is working with these tiny training sets, download a large number of French and Spanish phrases and generate a big training set (try to get 10,000 phrases at least).

  - 7. Test your classifier with another large set of phrases (that you know are either French or Spanish; you can use online news, for example.) What is your probability of misclassification?

Mini-Project 1: language classification

- Hints:
  - You will have to compute probabilities such as $P(X_1|G_k)$ (what are X_1 and G_k here?) To compute the probability of a word conditioned on the language being French, you can use the following rule:

    P(word|French) = (# of copies of word in French training set + 1) / (# of words in training set + 1)

  - Question: why did we add 1 in the numerator and denominator?s
  - To find the overall probability of a phrase, multiply the word probabilities. (They aren't really independent, but we'll assume that for this project.) Remember that logs can turn multiplications into additions.

Mini-Project 2: handwritten digit classification

- Use naïve Bayes classifier again (just like we did for the last project).
- You are provided with labeled training images. These are in the files "trainingimages" and "traininglabels". Credit: cs188 at Berkeley.
- Each image is a 28x28 grid of pixels. Dark (written to) pixels are labeled with "+" or "#". ("+" is gray and "#" is black, but we'll ignore this.)
- Your training function should read the training files and produce 10 matrices (one for each of the digits 0,1,…,9). Each matrix should be 25x25. Each position is the # of times that pixel was dark in the corresponding training data.
- Ex: if there are 400 "9"s in the training data, and the pixel at position (12,19) is dark in 373 of them, your matrix corresponding to "9" should have 373 at that position. Also store the value of 400.

Mini-Project 2: handwritten digit classification

- To classify a new image X, realize that Pixel i,j of X can be either dark or light. Let us say the value of the Pixel is X(i,j). We compute the following probabilities, similar to our previous project:

$$\Pr(G_i)*\Pr(X|G_i) =$$
$$\Pr(G_i)*\Pr(\text{Pixel } 1,1 = X(1,1) |G_i)*\Pr(\text{Pixel } 1,2 = X(1,2) |G_i) * \ldots * \Pr(\text{Pixel } 25,25 = X(25,25) |G_i)$$

- Here X(i,j) = dark or light and G_i for i=0,…,9 is the digit being written to.
- To get Pr(Gi), divide the # of digits representing i in the training set by total # of digits.

- Pr(Pixel j,k = dark | G_i) = (# of times pixel j,k dark in training images labeled i + 1)/ (total # of training images labeled i +1)
- Pr(Pixel j,k = light | Gi) = 1- Pr(Pixel j,k = dark).

- Finally, for whichever i gives the largest $\Pr(G_i)*\Pr(X|G_i)$, classify the image as digit i.

Mini Project 2: Handwritten Digit Classification

- **Use the SVM formulation to find a linear hyperplane that best separates the characters.**
- Solve the SVM problem to find the corresponding hyperplane w, b .
- Report your training accuracy i.e. number of characters correctly classified in training.
- Classify the test data report your classification accuracy i.e. number of characters correctly classified in test dataset.
- Display the 10x10 confusion matrix.
- Plot the characters that are wrongly classified as 8 in the test dataset.

Links to learn further

- https://teachablemachine.withgoogle.com/
- A good tutorial on Movie genre classification using Machine Learning and Deep Learning. https://spandan-madan.github.io/DeepLearningProject/
- https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12

References

• Textbook: Elements of Statistical Learning, 2nd Edition. Available: http://statweb.stanford.edu/~tibs/ElemStatLearn/

• Textbook: Introduction to Machine Learning by Alex Smola. Available: alex.smola.org/drafts/thebook.pdf

• Naive Bayes Classifiers: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

• Text classification with Naïve Bayes: https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

• Image classification: http://www.clarifai.com/#demo