Intro to Machine Learning

EE379K - Architectures for Big Data Sciences
Dr. Sriram Vishwanath
Department of Electrical and Computer Engineering
The University of Texas at Austin
Spring 2016

What is Machine Learning?

- Extremely broad field that is expanding very fast
- Many different definitions, but in general:
- “[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.”
    - Arthur Samuel (1959)

What is Machine Learning?

- Another definition:
- “A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**”
    - Tom Mitchell

Example Tasks

- Predict traffic patterns at an intersection
- Predict stock prices
- Determine if a tumor is cancerous or benign
- Speech and object recognition
- Robot navigation

Machine Learning and Big Data

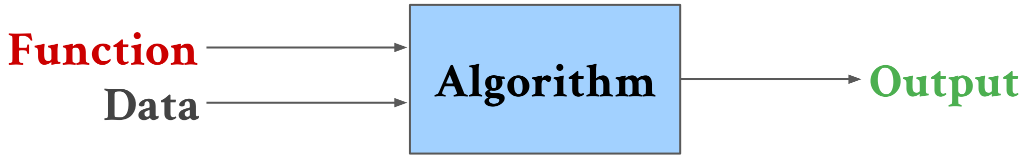
- Many machine learning algorithms are data-driven
- Data provides the “experience” that ML algorithms use to train
- More data generally leads to better outcomes
  - Not always true
- ML and big data go hand in hand

Three Primary Categories

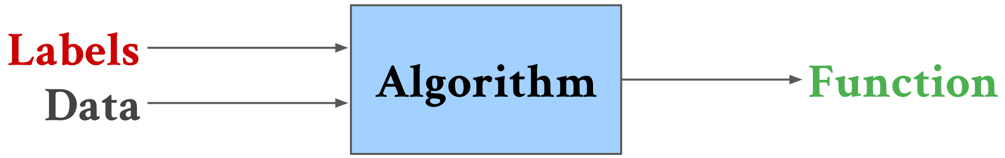
- Supervised Learning
  - Provide sample inputs and the desired outputs (a.k.a. **labels**)
  - Algorithm learns function □ that maps inputs to outputs
- Unsupervised Learning
  - Provide only data (no labels)
  - Algorithm figures out underlying structure
- Reinforcement Learning
  - Computer is placed in a dynamic environment
  - Learns how to act in order to maximize **reward** or minimize **penalty**

Machine Learning vs Traditional Programming

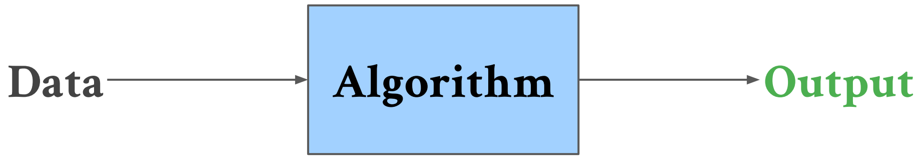
- Traditional programming:



- Supervised machine learning:



- Unsupervised machine learning:



Supervised Learning

Supervised vs Unsupervised Learning

- Unsupervised learning → no notion of "right answer"
  - Many possible answers
- Supervised learning → there is a right answer!
  - Algorithms are trained ("supervised") with training data
  - Data often divided into training and test data sets
    - Supervised learning on training set
    - Model verification on test set

Supervised Learning Goal

- The ultimate goal is to develop a “model”, P(x)
  - “x” describes the **features** of the input
- We use training data to develop P(x)
- Then, we can use P(x) to make “predictions” on new data

Supervised Learning Goal

- Example: train a predictor P(x) that predicts house price based on:
  - Size (sq. ft.)
  - GPS coordinates
  - Number of bedrooms
  - Year built
- In this case, “x” = {size, # bedrooms, year}
- In reality, the number of features could be millions!
  - Feature selection is a major challenge in many cases

Types of Supervised Learning

- There are two major categories of supervised learning:
- Regression
    - For systems where the values are “continuous”
    - “Given a size and GPS coordinates, model how much this house costs”
- Classification
    - For systems where the output labels are discrete
    - “Is this cell benign or cancerous?” “male or female”

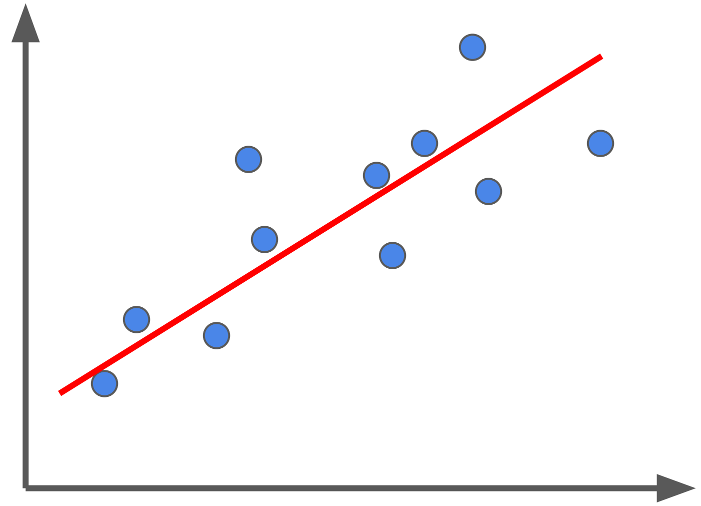
Regression

Regression Learning

- Suited for systems where at least some of the parameters are continuous
- Examples: modeling house prices, financial modeling
- In this case, P(x) may be a (piecewise) continuous function

Fitting a Line

- Simplest form of regression learning
- Find a line that best fits the data

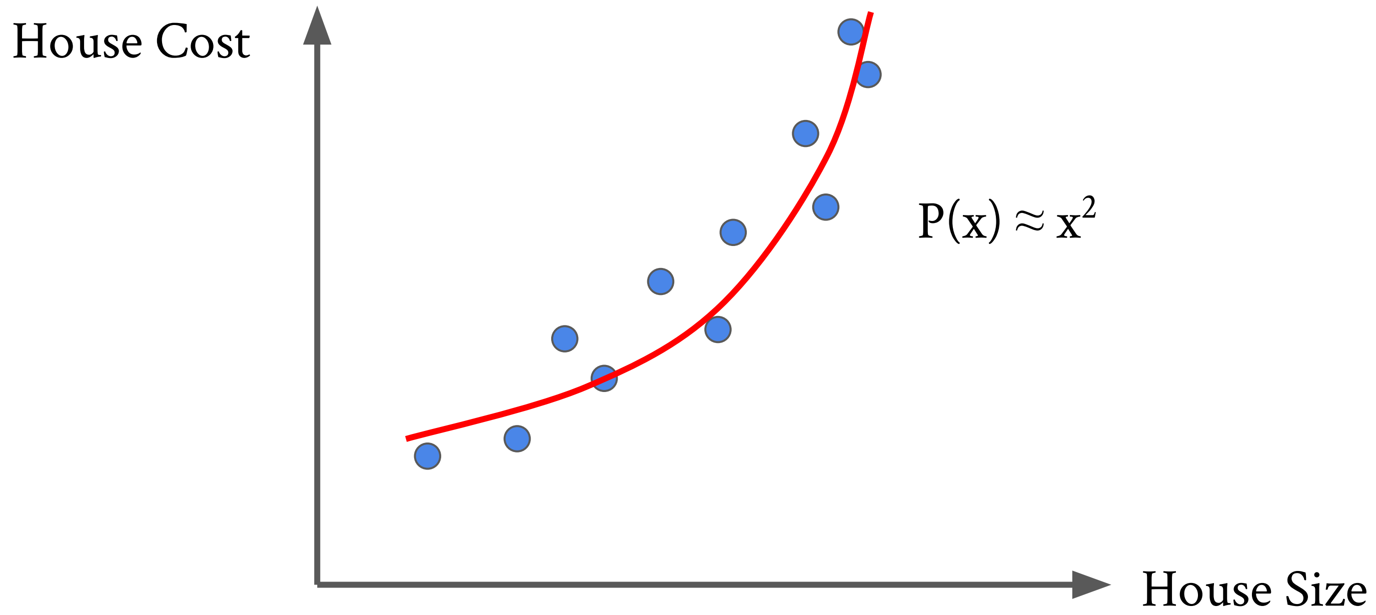


Linear Regression Learning Goal

- Predictor P(x) has the form of a line:
- $P(x) = \theta_0 + \theta_1 x$
- Goal: Find θ_0 and θ_1 that best match the data

Polynomial Relationship

- Sometimes the data is not strictly linear



EE351K: Estimation

Remember point estimation we did in EE351K

1. Finding least squares estimator (LSE)?
2. Regression & Estimation related
    a. Find error function (squared, example)
    b. Find curve that minimizes error function
    c. Depending on error function, different curve (potentially)

Classification

Classification Learning

- Suited for discrete labeling.
- “Is this cell cancerous?”: Binary labeling
- In this case, P(x) can be a likelihood of true/false (probabilistic classifier)
    - **P(x) = 0** $\Rightarrow$ Cell is definitely not cancerous
    - **P(x) = 1** $\Rightarrow$ Cell is definitely cancerous
    - **P(x) = 0.7** $\Rightarrow$ Cell is probably cancerous
    - **P(x) = 0.5** $\Rightarrow$ I don’t know

EE351K: Hypothesis Testing

Remember hypothesis testing from 351K?

Multiple hypotheses +Training data

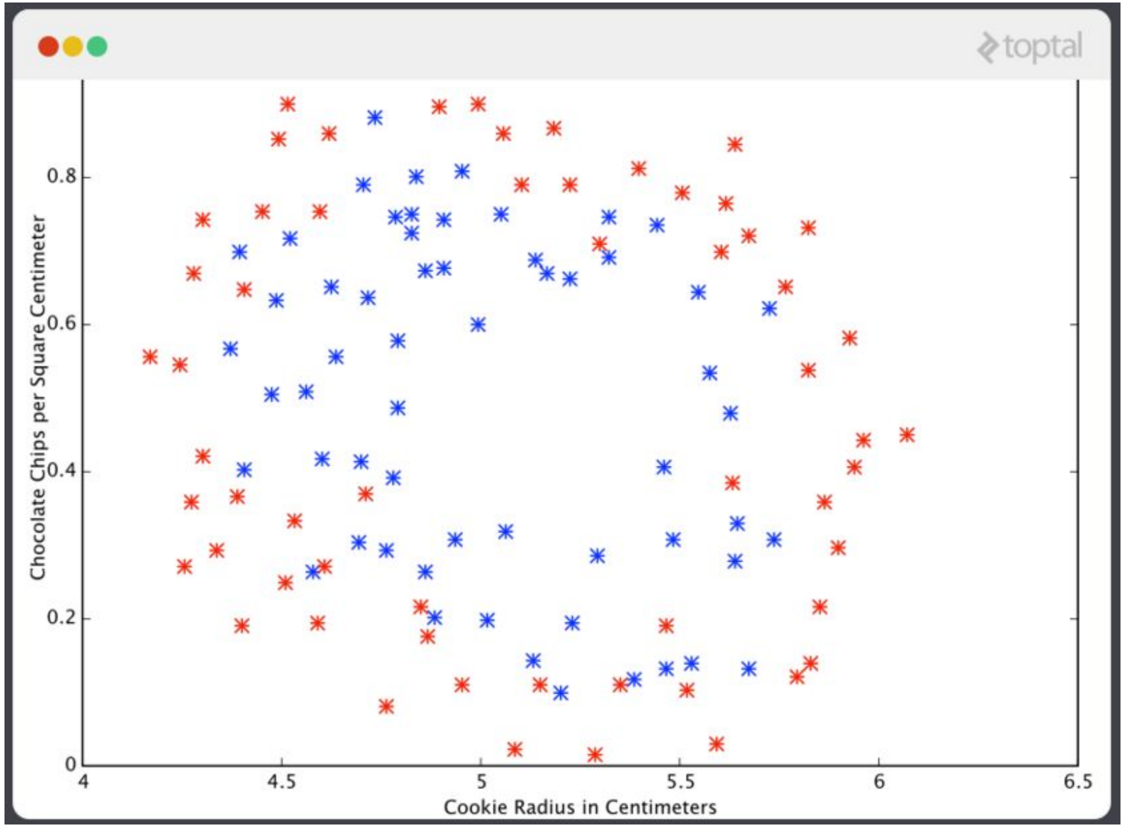
Probabilistic classification = find most likely hypothesis

Classification has many other techniques (non probabilistic)

Classification Learning: Cookie Example

Red - Bad cookie

Blue - Good cookie



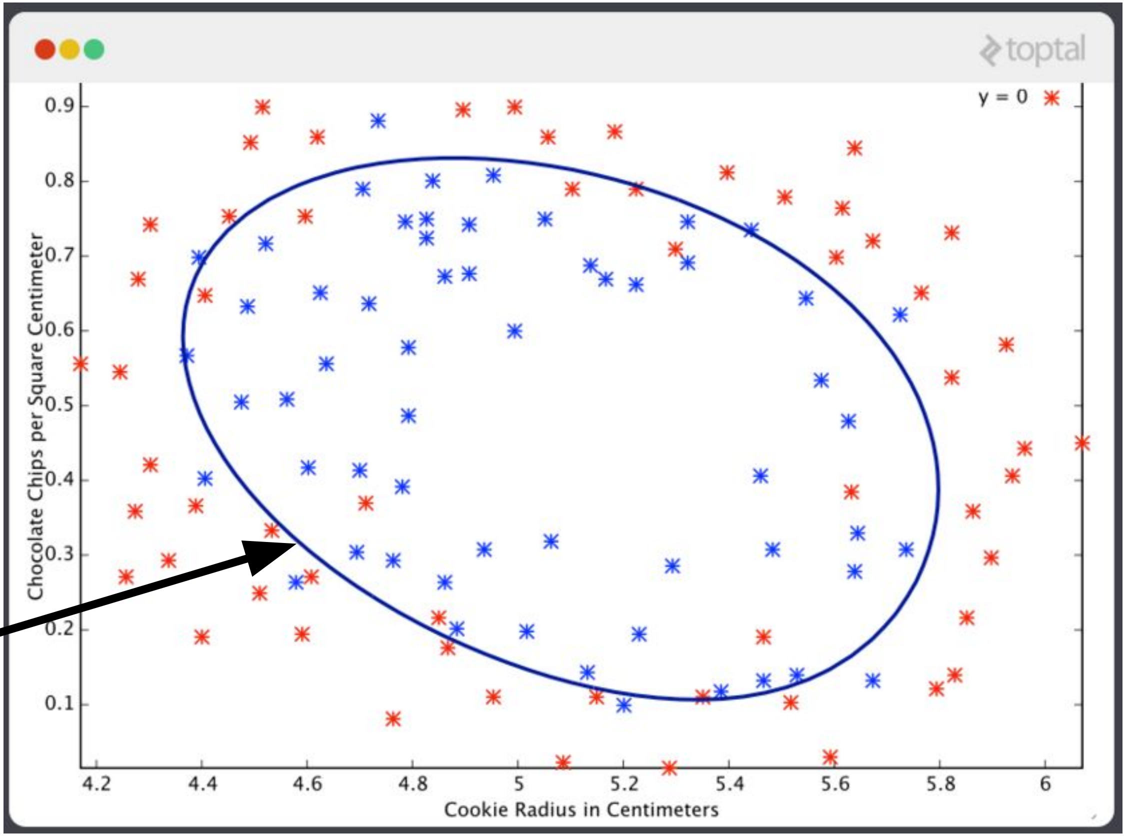
Example from: http://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-

Classification Learning: Cookie Example

Red - Bad cookie

Blue - Good cookie

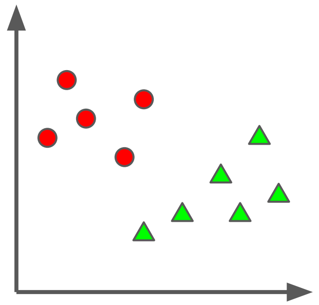
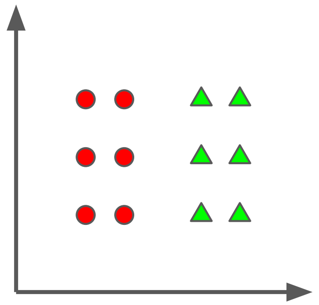
Decision surface



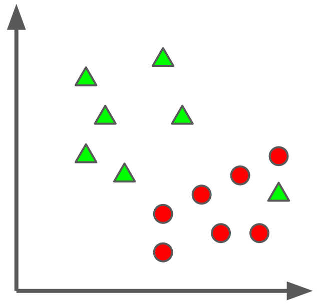
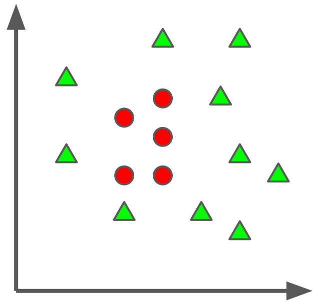
Example from: http://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-

Classification Learning: Linear Separability

- Linearly separable:

- Not linearly separable:

Classification Learning: The Perceptron

- A perceptron is a **binary** classifier
- Classifies input ***x*** into a single binary value

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

- ***w*** is a vector of weights
- ***b*** is the y-intercept or **bias**

Classification Learning: The Perceptron

- Training the perceptron involves determining the weight vector and the bias.
- The perceptron can only learn linearly separable patterns
- This is a big shortcoming, which we will address later

Unsupervised Learning

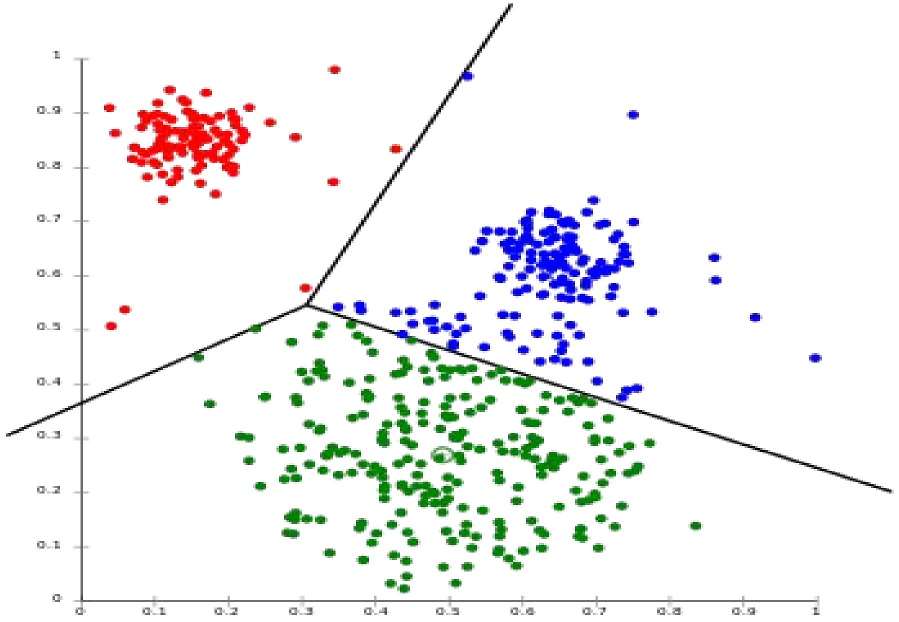
Unsupervised Learning Approaches

- Remember:
  - No “right answer”
  - Algorithm must discover underlying structure
- Approaches:
  - Clustering
  - Method of moments
  - Blind signal separation
    - PCA/ICA
    - Other matrix decomposition type techniques

Clustering

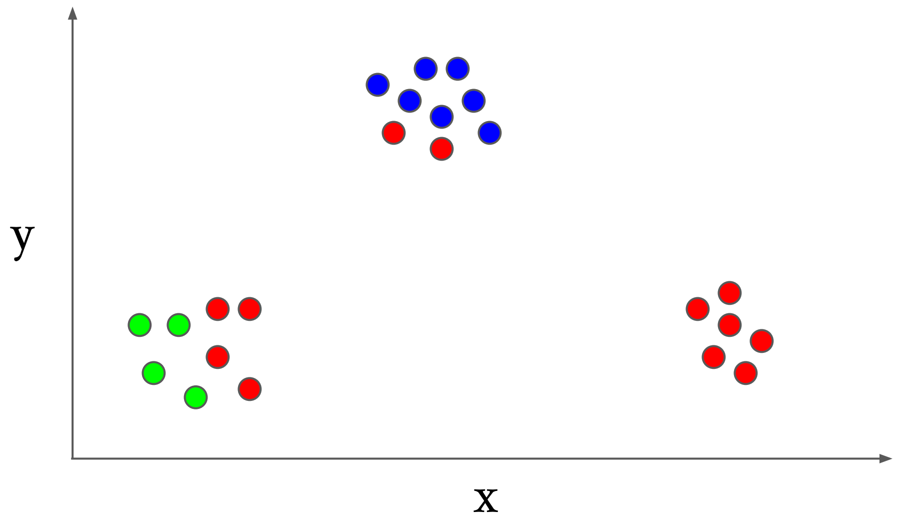
Data points broken into "clusters"

Separated here by affine "dividers"



Clustering

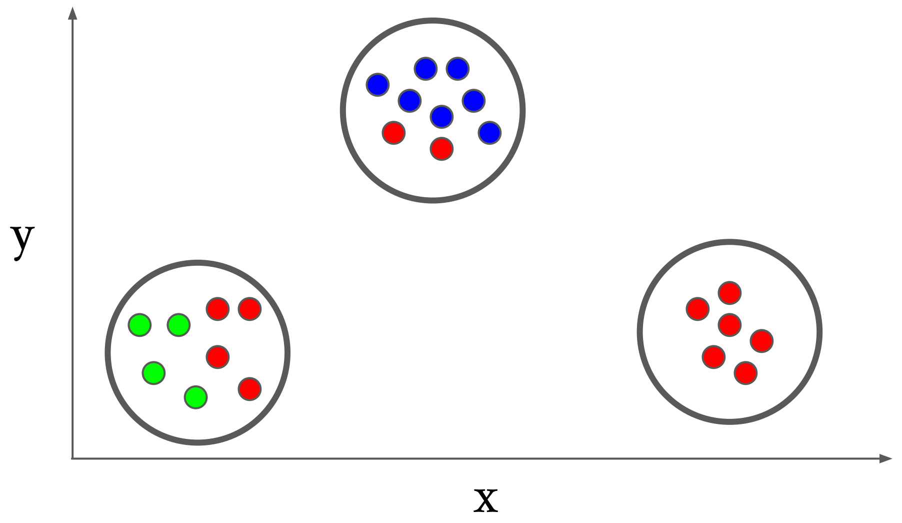
- Group a set of objects into clusters
- “Similar” objects should be in the same cluster



How many clusterings?

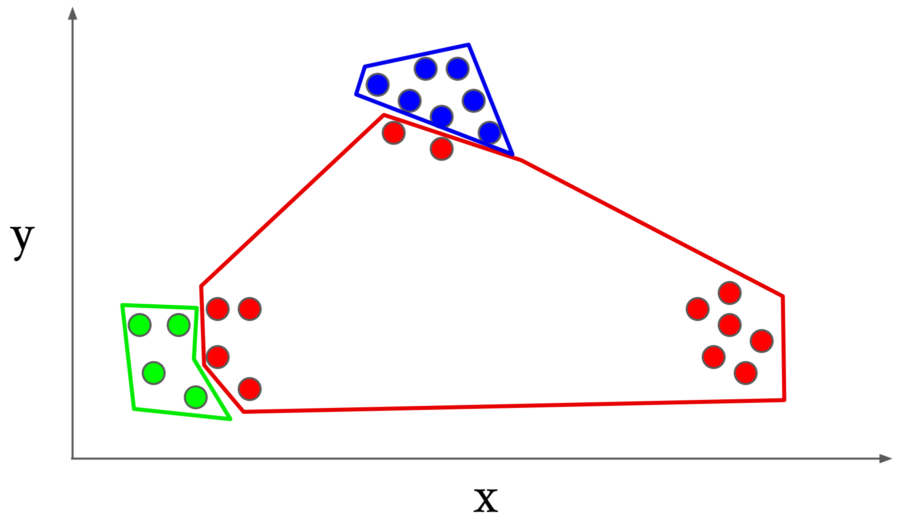
Clustering by Position

- Total of three clusters



Clustering by Color

- Three clusters again, but different points!
- Note the complexity of the fit



Clustering Applications

- Marketing
  - Find distinct groups in your customer base (men, women, children, etc)
  - Develop products suited for each group

- Cell tower deployment
  - Cluster customers by location
  - Place towers at cluster centers to maximize QoS for customers

- Many more

Method of Moments

- E[x^k] = kth moment of x
- Moments tell you a lot about the underlying distribution
- If X is a vector, its kth moment is a kth-order tensor
- Each value tells you interdependencies (or lack thereof) between components of X.

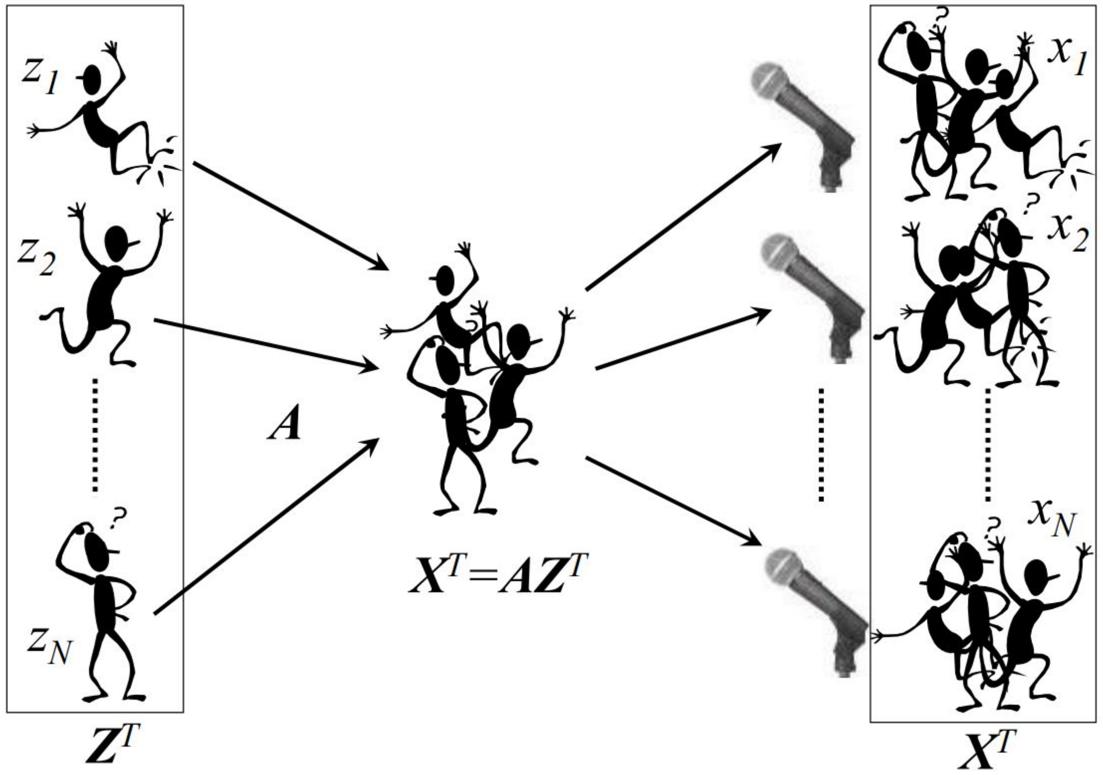
Blind Signal Separation

- $\underline{N}$ unknown sources s_j
- One unknown operator $\mathbf{A}$
- $\underline{P}$ observed signals x_i with the relation: $x = \mathbf{A}(s)$
- Goal: Estimate s

Blind Signal Separation: Cocktail Party Problem

- There are multiple people talking at a cocktail party
- You have recordings of the overall sound signature from multiple microphones
  - At least as many microphones as people
- Can you recover the individual voices of the people?

Blind Signal Separation: Find Z



Blind Signal Separation: Algorithms

- Need a way of separating **independent** sources
- Find a statistical representation of the data
- Project data onto a new set of axes that fulfill some statistical criteria
- Principal Component Analysis (PCA)
  - use an orthogonal transformation to generate linearly uncorrelated values
- Independent Component Analysis (ICA)
  - Not necessarily uncorrelated (for non-Gaussian sources)

Reinforcement Learning

Reinforcement Learning Approaches

- Reinforcement learning deals with how agents should act to maximize reward
  - Extremely general concept that applies to many fields
  - Also called Artificial Intelligence (AI)
- (Some) Approaches and Algorithms:
  - Artificial Neural networks
  - Simultaneous Localization and Mapping (SLAM)

Neural Networks

Neural Networks

- Family of modeling techniques inspired by biology
- Used to approximate functions with a large number of inputs
- Many different types:
  - Single perceptron
  - Feed-forward
  - Auto-encoders

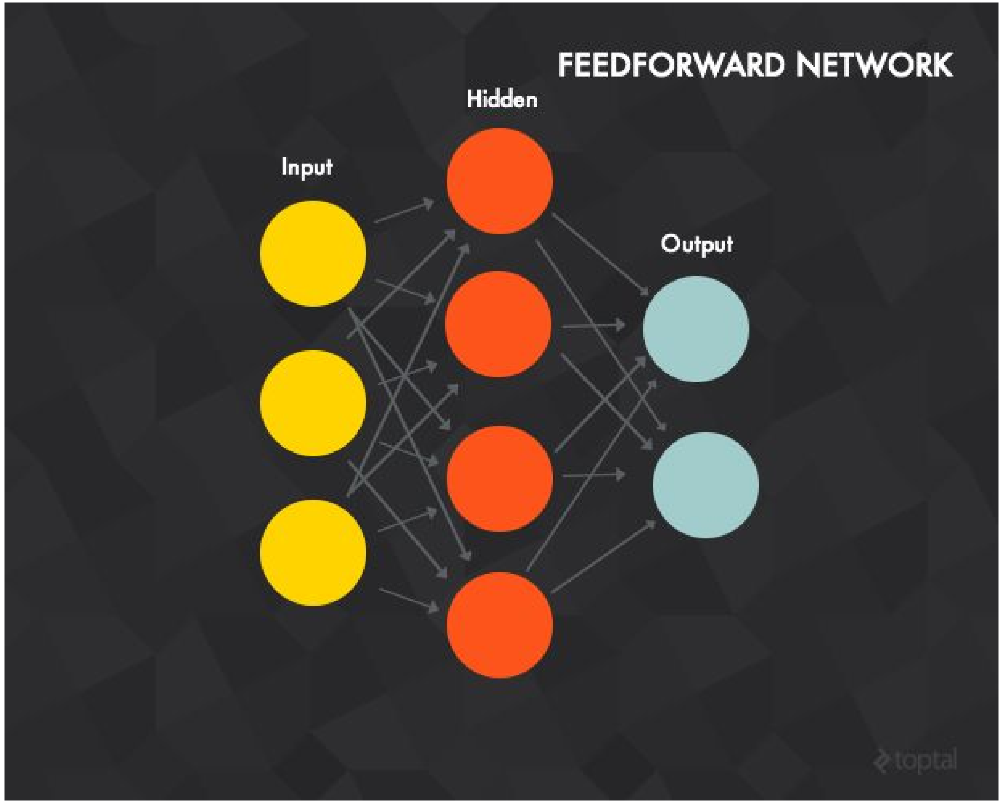
Neural Networks: Building Blocks

- Perceptrons are the building blocks of most neural nets
- A single layer of perceptrons can only learn linearly separable data
- But: Multiple layers of perceptrons are capable of learning more general patterns
    - Piecewise linear approximations

Feed-Forward Networks

Example from: http://www.toptal.com/machine-learning/an-introduction-to-deep-learning-from-perceptrons-to-deep-networks

- Each node is a perceptron
- Output of previous layer is fed as input to the next layer



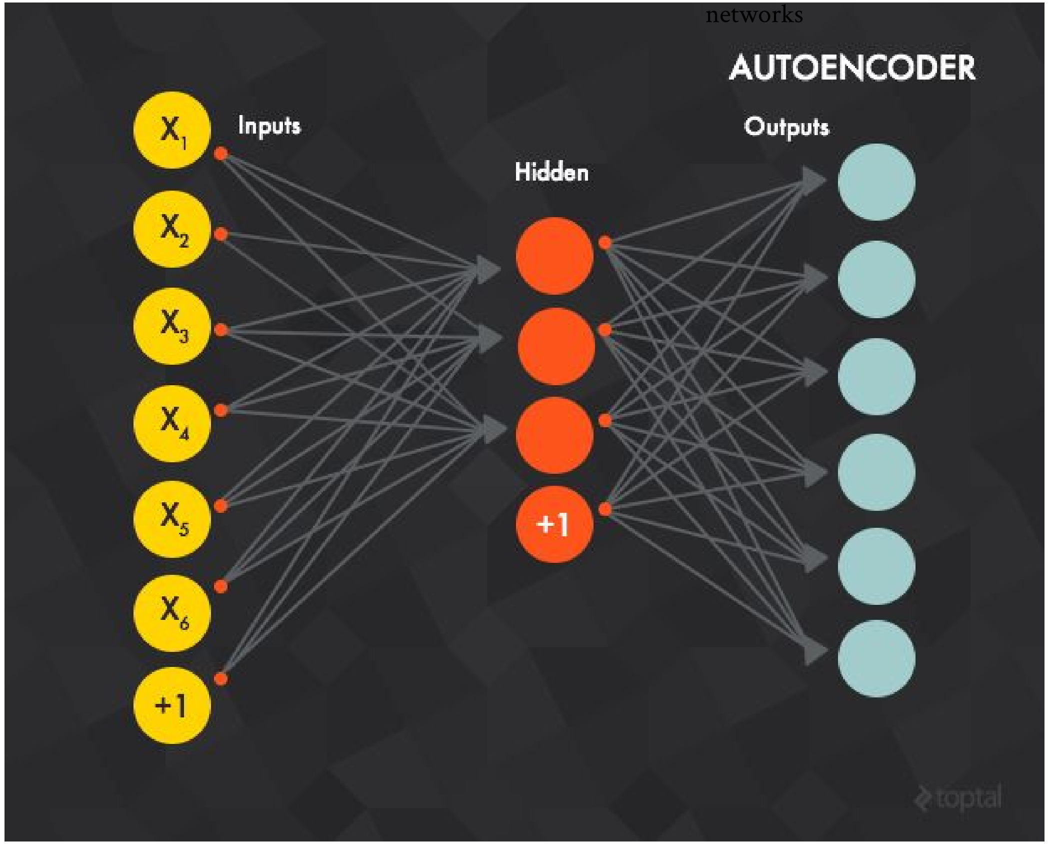
Auto-Encoders

- A class of feed-forward neural nets
- Auto-encoders learn a compressed, distributed **encoding** (representation) of a dataset
- These networks recreate their input in a "compressed" form

Example from: http://www.toptal.com/machine-learning/an-introduction-to-deep-learning-from-perceptrons-to-deep-networks

Auto-Encoders

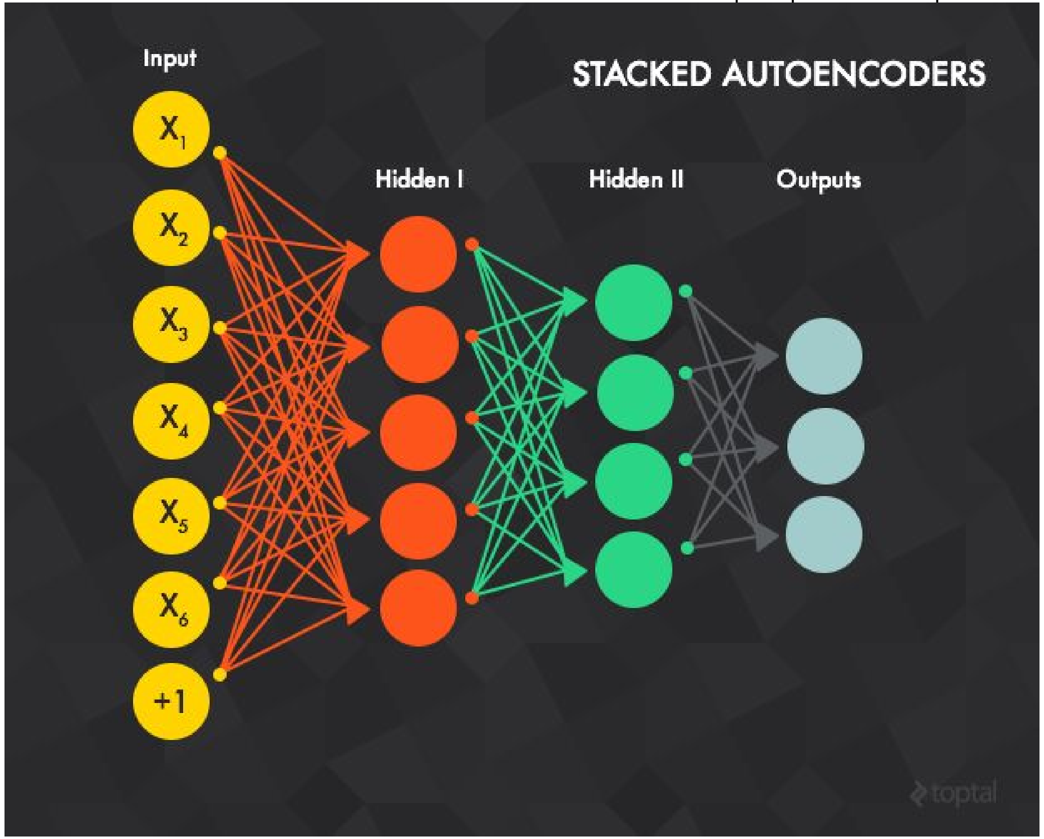
- Fewer hidden nodes than inputs and outputs
- This forces the net to find a compressed representation of the input



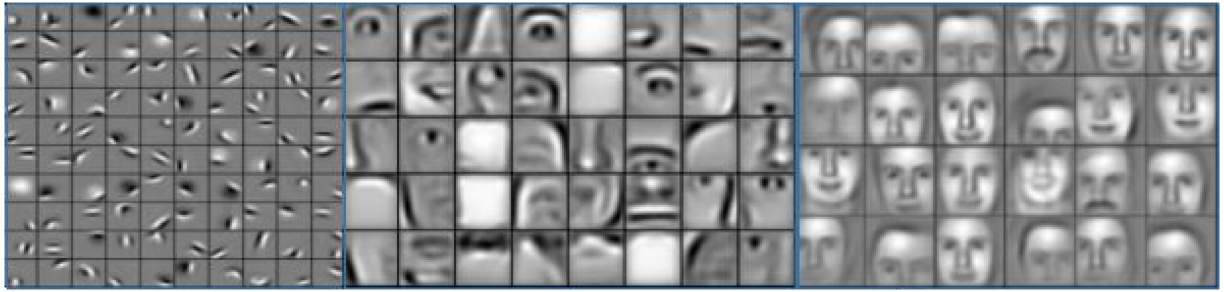
Example from:
http://www.toptal.
com/machine-learning/an-
introduction-to-deep-learning-
from-perceptrons-to-deep-

Deep Learning

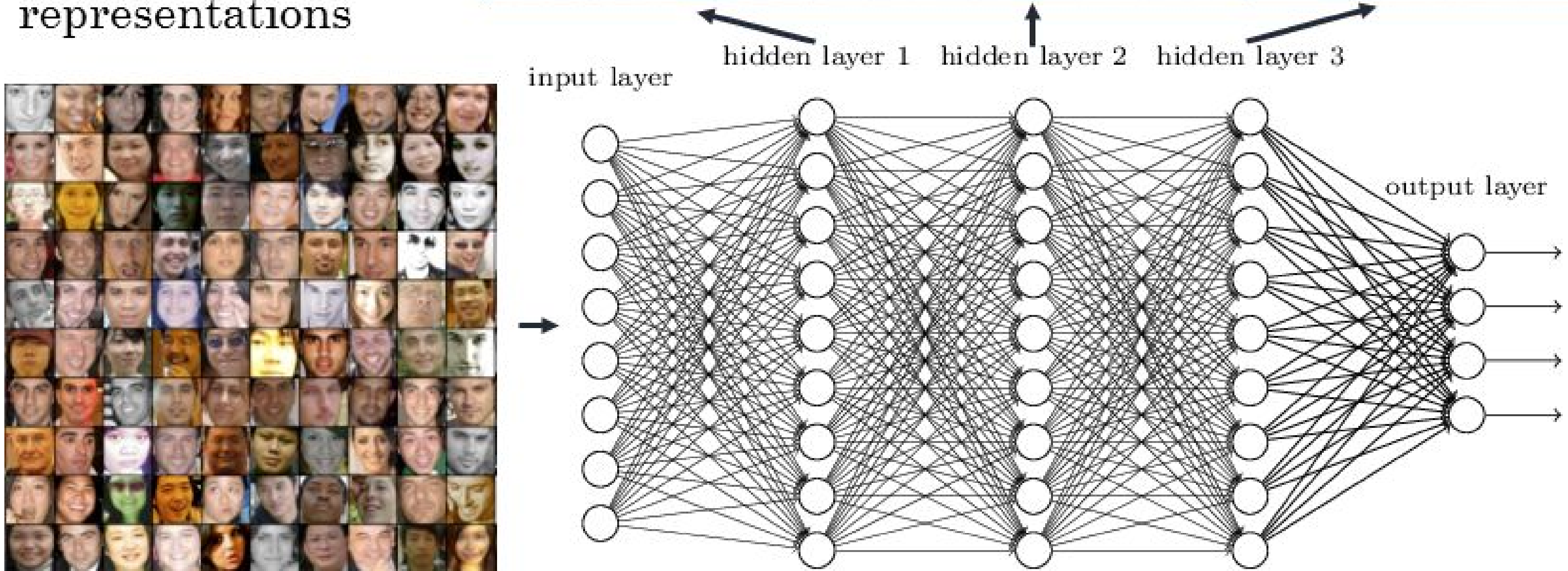
- Auto-encoders can be viewed as "feature" detectors!
- What happens if we stack them?
- We get a net that progressively builds higher level features



Deep Learning: Hierarchical Feature Representation



Deep neural networks learn hierarchical feature representations



Simultaneous Localization and Mapping (SLAM)

SLAM

- Imagine a robot moving in an unknown environment
- Goal: we want to do two things (at the same time):
  - Figure out where we are in the environment
  - Generate a map of the environment
- But:
  - We don't know our **position** at any given time
  - We don't know our **velocity** at any given time
  - Our sensors are imperfect

SLAM

- Chicken and egg problem:
  - To build a map, we need to know our position
  - To know our position, we need a map

- Solution: Find both simultaneously
  - Localization: my position
  - Mapping: My environment

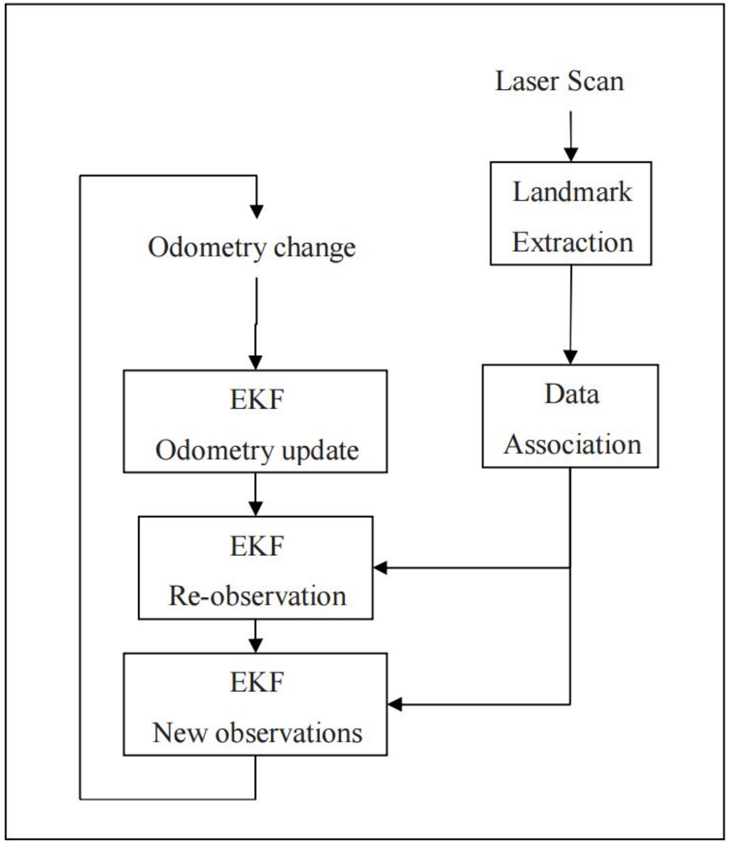
SLAM: Required Components

- A robot
- Odometry for position and velocity data
  - An encoder on your motors
  - An airspeed sensor for flying craft
  - GPS
- Range measurement device (usually a laser/IR scanner)
- Other sensors: Acoustic, Visual (video) etc

SLAM: Algorithm Flow

- **EKF** - Extended Kalman Filter
  - EKF helps the robot determine it's current position, attitude, and velocity

- **Landmark** - distinguishable features in the environment
  - These help the robot understand it's location and orientation in the environment



SLAM

- Very complex problem
- Strong applications in robotics:
  - Self-driving vehicles
  - Path finding
  - Etc.
- Aerial SLAM

Conclusion

- Machine learning is a very large field with many subcategories
  - Regression, classification, localization, etc
- There are three primary classes of machine learning algorithms
  - Supervised - learn from training data
  - Unsupervised - find underlying structures in data
  - Reinforcement - deal specifically with minimizing penalties
- Machine learning is a large part of big data analytics

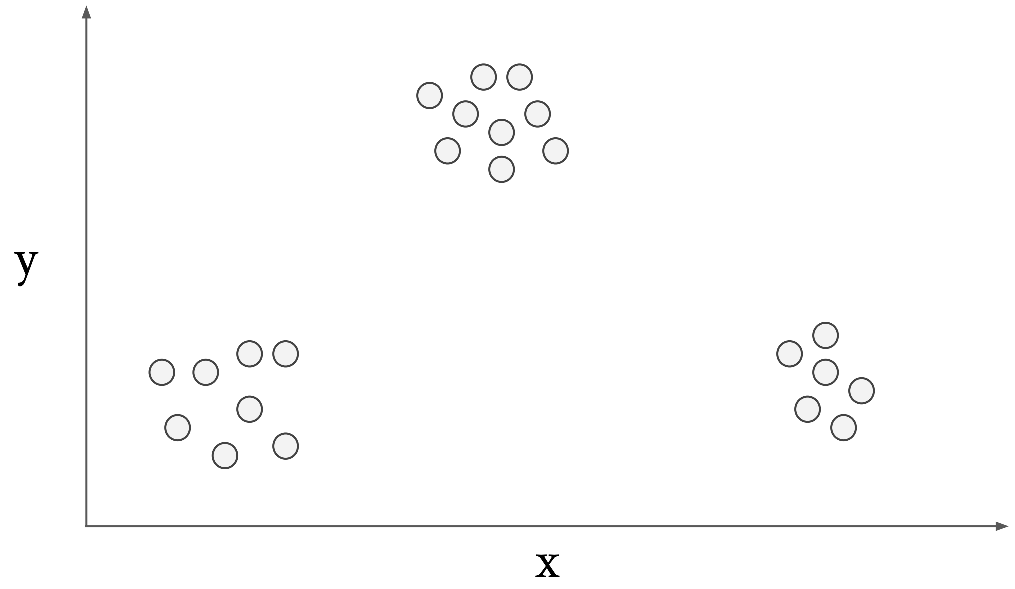
Introduction to K-means Clustering

K-Means Clustering Algorithm

- Partitions **N** input points into **K** clusters
- Clusters defined by their **centroids** ("centers")
- Each input point $\mathbf{n_i}$ belongs to some cluster **k** (with centroid $\mathbf{c_k}$)
- Minimize distance from each point to the centroid of its cluster

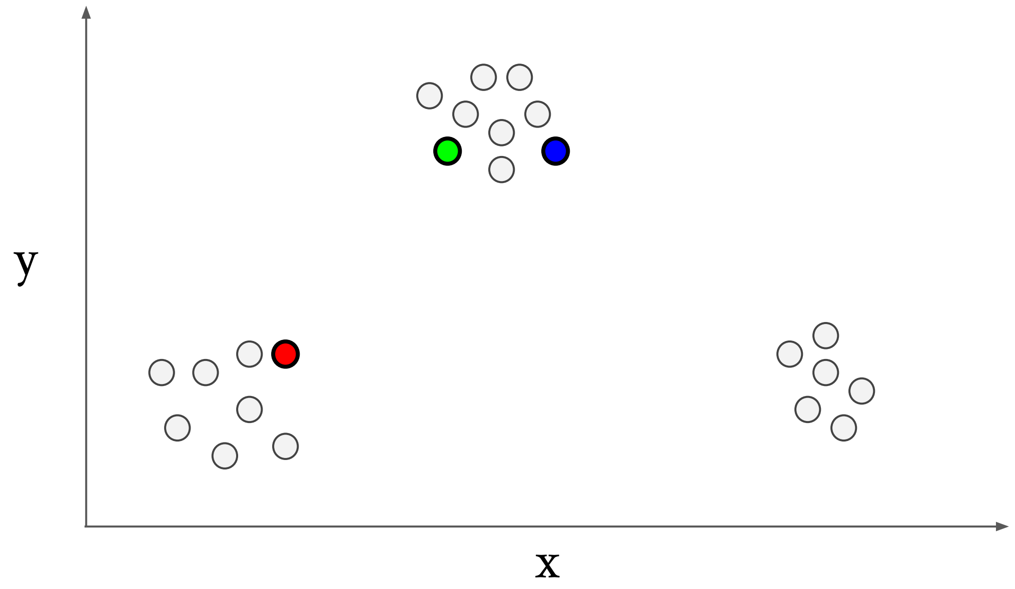
K-Means Example

- Segment the following points into three clusters (by position)



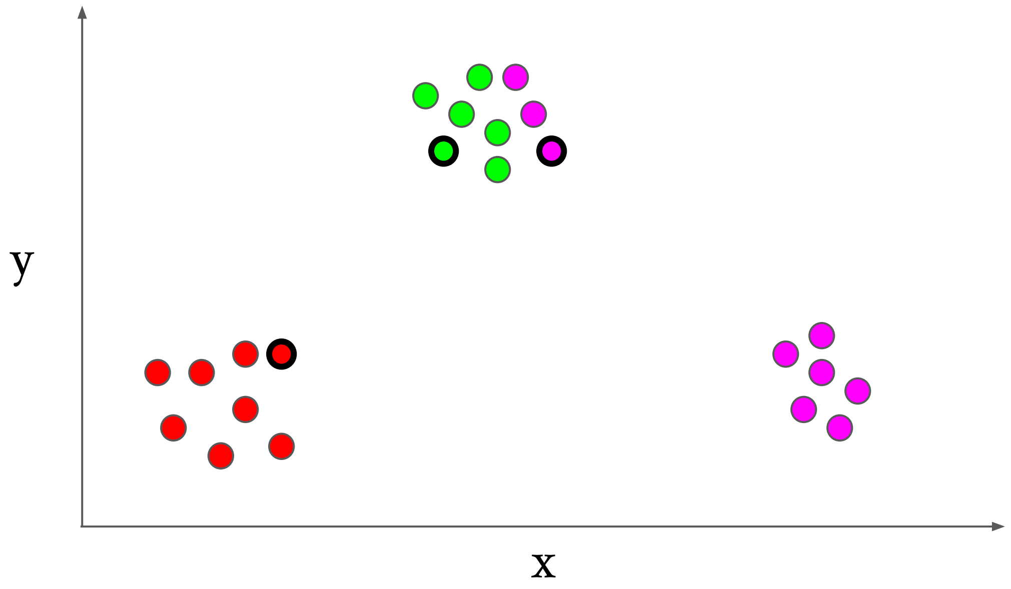
K-Means Example

1. K = 3. Pick three random data points as initial centroids



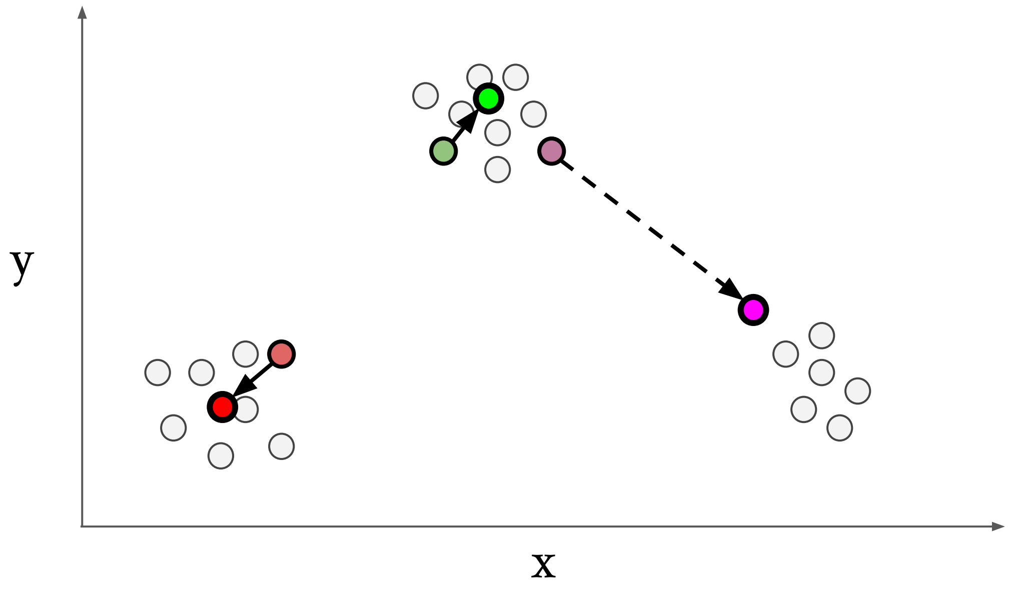
K-Means Example

2. Assign each point to its nearest centroid



K-Means Example

3. Recompute each centroids location as the average location of its children



K-Means Example

4. Repeat assignment and update steps until no assignments change