Machine Learning Exercises for High School Students

Joshua B. Gordon

July 7th, 2011

CS@CU

Outline

- Recommendation systems
- Intuition for algorithms that find patterns in data
- Clustering using Euclidian distance
- Classroom exercises

amazon.com Prime

Hello, Joshua B. Gordon. We have recommendations for you. (Not Joshua?)

New: Kindle with Special Offers - $114

Joshua's Amazon.com | Today's Deals | Gifts & Wish Lists | Gift Cards

Your Digital Items | Your Account | Help

Shop All Departments | Search | All Departments | GO | Cart | Wish List

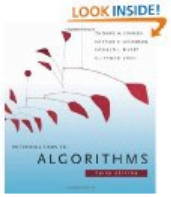
Your Amazon.com | Your Browsing History | Recommended For You | Rate These Items | Improve Your Recommendations | Your Profile | Learn More

Joshua, Welcome to Your Amazon.com (If you're not Joshua B. Gordon, click here.)

Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to **see all recommendations**.

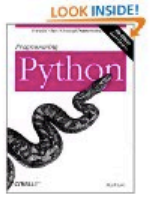
Page 1 of 35



Introduction to Algorithms (Hardcover) by Thomas H. Cormen

(27) $53.81

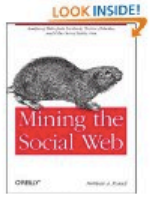
Fix this recommendation



Programming Python (Paperback) by Mark Lutz

(9) $39.23

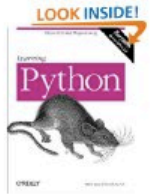
Fix this recommendation



Mining the Social Web: Anal... (Paperback) by Matthew A...

(13) $26.12

Fix this recommendation



Learning Python: Powerful Obje... (Paperback) by Mark Lutz

(27) $33.86

Fix this recommendation

Coming Soon for You

Madden NFL 12

$59.99

Fix this recommendation

› See more recommended future releases

Tap into Your Friends

BETA

Connect to Facebook to get Amazon recommendations for you and discover your friends' Favorites and Likes

Learn more and Connect

Amazon

- Amazon doesn't know what it's like to read a book, or what you feel like when you read a particular book
- Amazon *does* know that people who bought a certain book also bought other books
- Patterns in the data can used to make recommendations
- If you've built up a long purchase history you'll often see pretty sophisticated recommendations

Netflix prize

- Netflix is an online DVD rental company that recommends movies to subscribers
- 2006: Netflix announce **$1 million** to the first person who can improve the accuracy of its recommendation algorithm by 10%
- How can an algorithm recommend movies?
- By leveraging patterns in data (and lots of it)

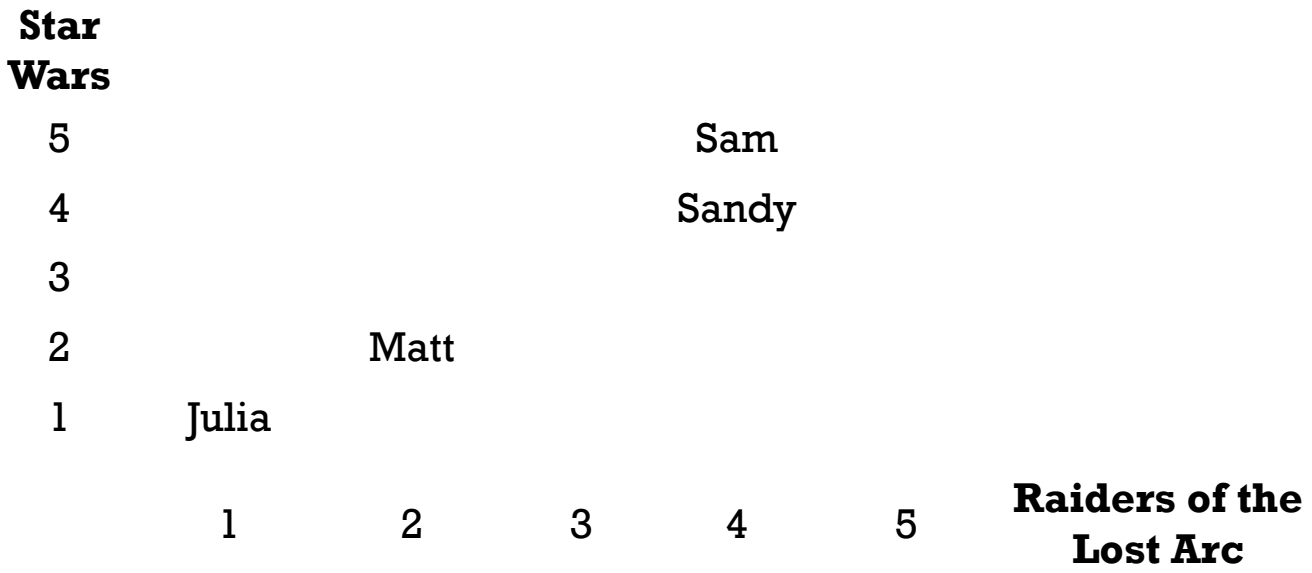
Dataset: movie critics

Critic	Star Wars	Raiders of the Lost Arc	Casablanca	Singin' in the Rain
Sam	****	****	*	**
Sandy	*****	****	**	*
Matt	**	**	****	***
Julia	**	*	***	****
Sarah	*****	?	?	**

- How could an algorithm use this data to recommend movies?
- How would you do it?

Critics with similar tastes

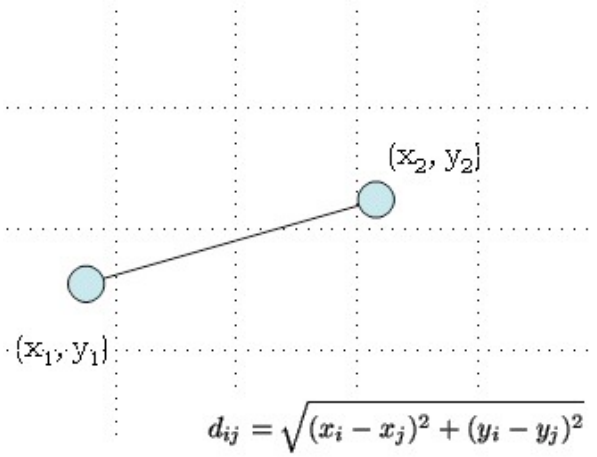
- Preference space



Measuring distance

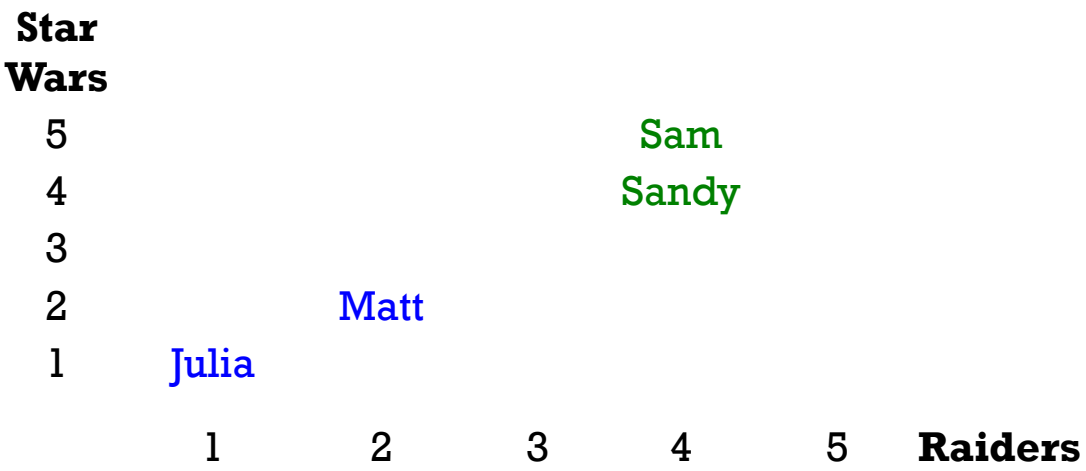
- Measure similarity with Euclidian distance

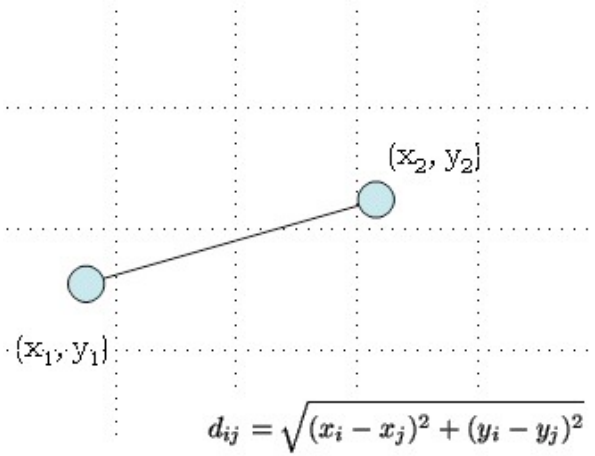
Star Wars						
5				Sam		
4				Sandy		
3						
2		Matt				
1	Julia					
	1	2	3	4	5	**Raiders**



Finding critics with similar taste

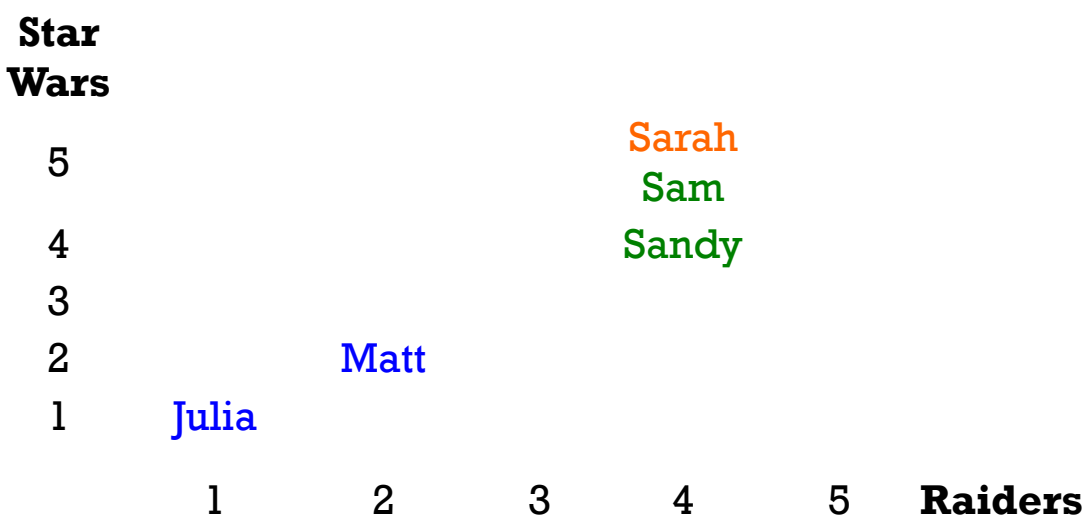
- People who liked Star Wars are close in preference space to people who liked Raiders of the Lost Arc

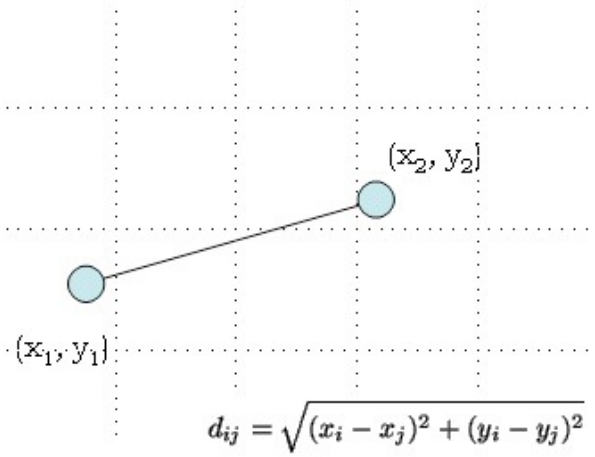




Making a recommendation

- Sarah hasn't seen Raiders, but gave Star Wars five stars
- It's a good bet she'll like Raiders too



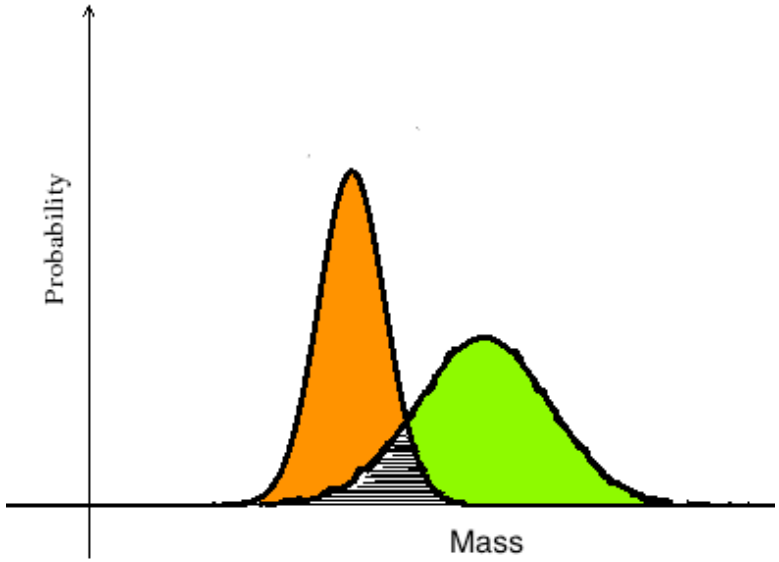


Features

- We used features to compare critics
- Feature: a data attribute used to make a comparison
- Quantify attributes of an object (size, weight, color, shape, density) in a way a computer can understand
- Quality is important

Apples vs. oranges

- A good feature discriminates between classes
- Think: how well does a feature help us tell two things apart?
- Is mass a good feature? By itself?
- What about in conjunction with another feature like color?



Features to compare movies

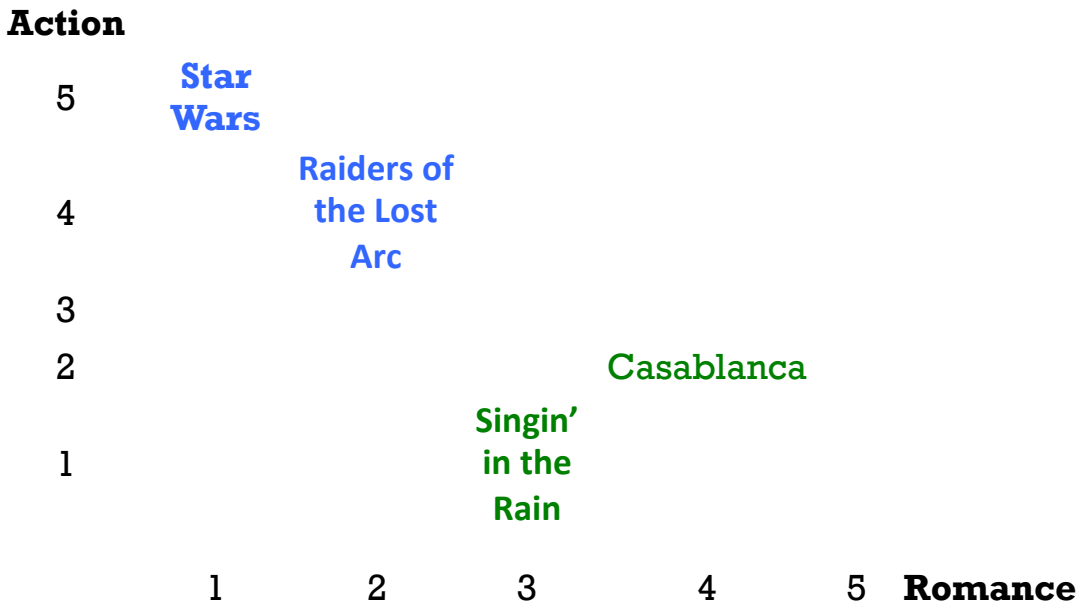
Feature	Star Wars	Raiders of the Lost Arc	Casablanca	Singin' in the Rain
…				
…				
…				
…				
…				

Features to compare movies

Feature	Star Wars	Raiders of the Lost Arc	Casablanca	Singin' in the Rain
Action (1 to 5)	5	4	2	1
Romance (1 to 5)	1	2	4	3
Length (min)	121	115	102	103
Harrison Ford	Y	Y	N	N
Year	1977	1981	1942	1952

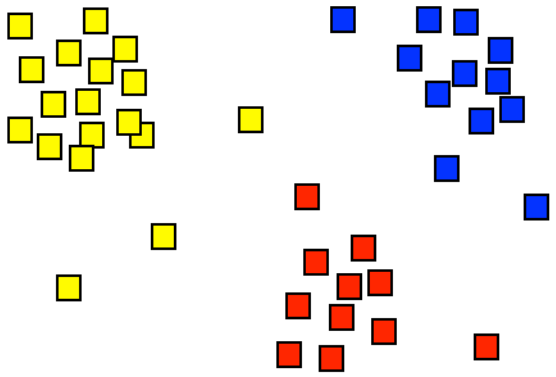
Feature space

- We can compare the similarity of movies in feature space using the same technique we used to compare movie critics.
- So we can compare items and people in the same way!



Clustering

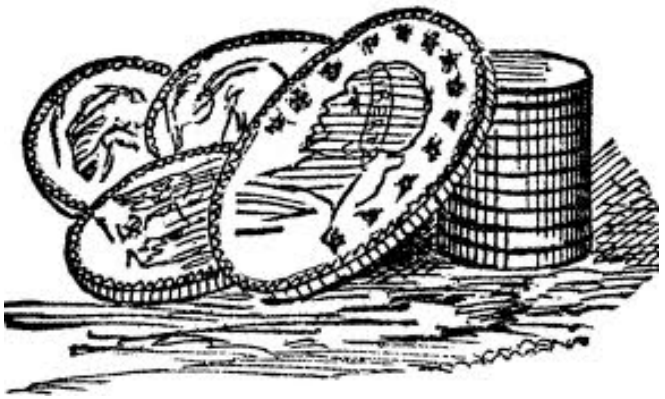
- Cluster: group of related objects
- We did OK at eyeballing clusters, but what if we had lots of data? Or wanted to use more than two dimensions?
- Today we'll learn a Machine learning method called **K-means** that finds clusters **automatically**
- **Machine learning** is a field of computer science that studies algorithms that learn from patterns in data.
- 3 class exercises



Sorting coins without machine learning

- Suppose you wish to separate quarters, nickels, dimes
- What information would the computer need to distinguish between these three types of coins?
- Think about how you would do the task yourself

Exercises from Steve Essinger and Gail Rosen's excellent article: "An Introduction to Machine Learning for Students in Secondary Education"



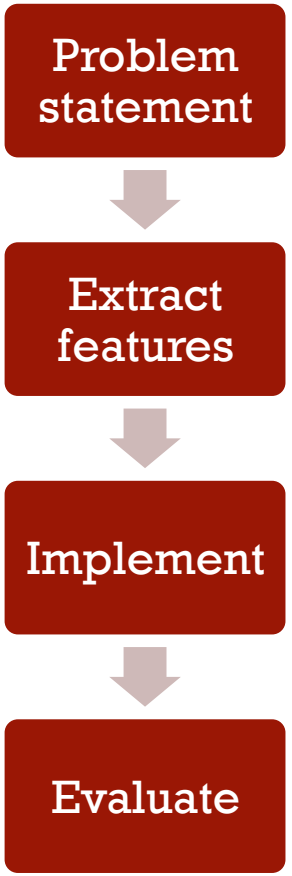
Class exercise: sorting ancient coins

- An archeological expedition has uncovered a massive cache of roman coins!
- We want the computer to sort the coins automatically
- We know there are 9 types of coins, but many are worn down and hard to distinguish
- Algorithm must sort the coins without help from us
- 1,000,000 coins, so manual sorting is undesirable

Exercises from Steve Essinger and Gail Rosen's excellent article: "An Introduction to Machine Learning for Students in Secondary Education"

Problem solving

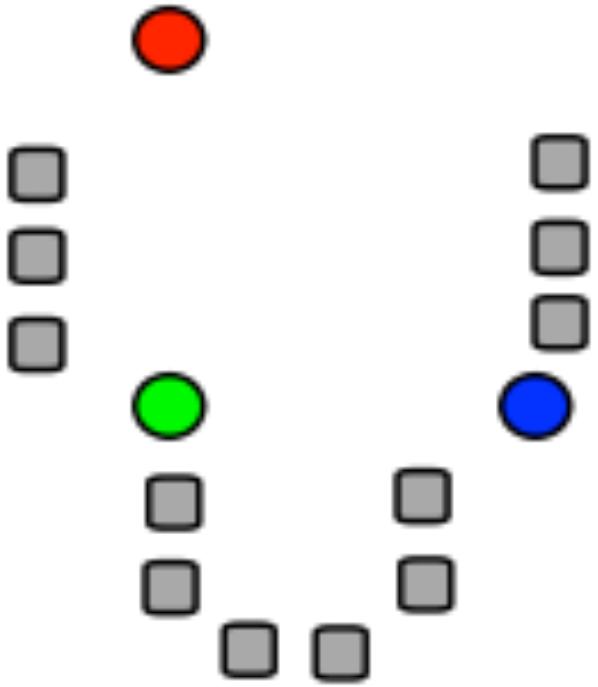
- Problem statement: automatically sort a large bag of ancient coins.
- The K-means algorithm will be used to find clusters
- First it needs features to compare the similarity of data points
- If poor features are chosen, the algorithm will be unable to solve the task.



The K-means algorithm

- To start, K-means needs to know *k,* the number of types of coins in advance.

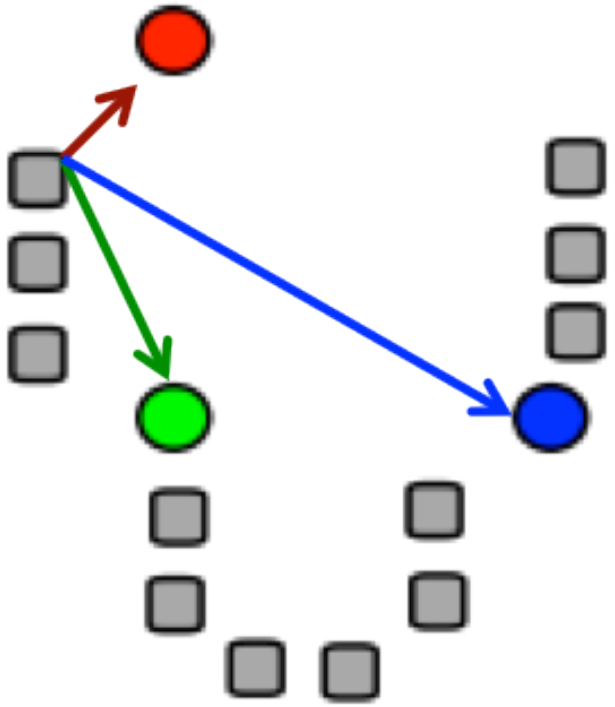
1. Choose *k* starting points randomly. These are called centroids.



The K-means algorithm

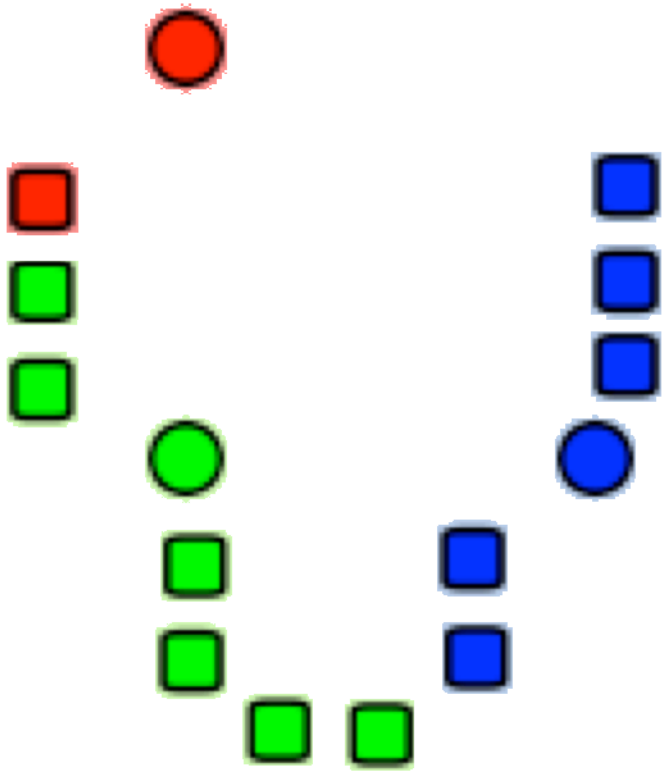
1. Choose *k* starting points randomly. These are called centroids.

2. Calculate the Euclidian distance between each point and each of the centroids.



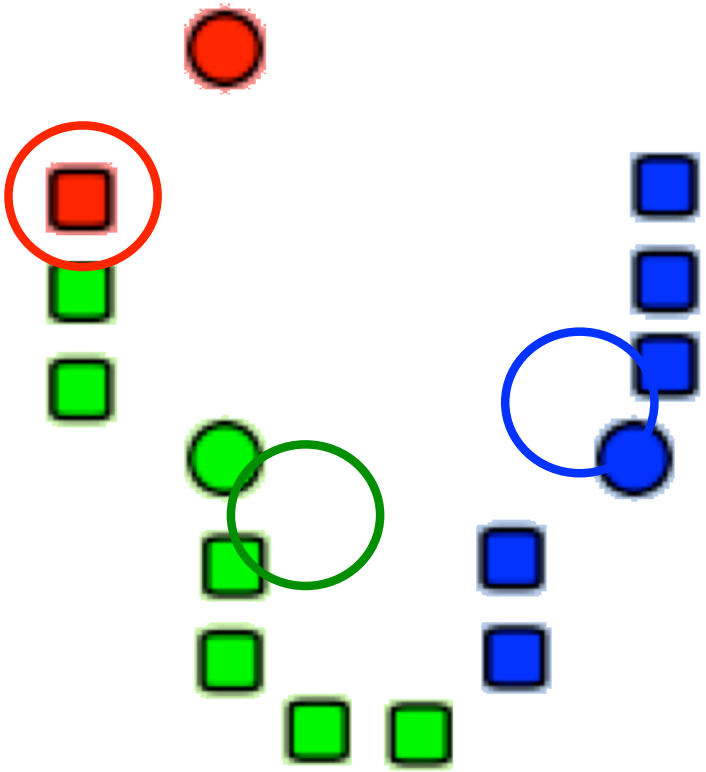
The K-means algorithm

1. Choose *k* starting points randomly. These are called the centroids.
2. Calculate the Euclidian distance between each point and the the centroids.
3. Color each point according to the nearest centroid.



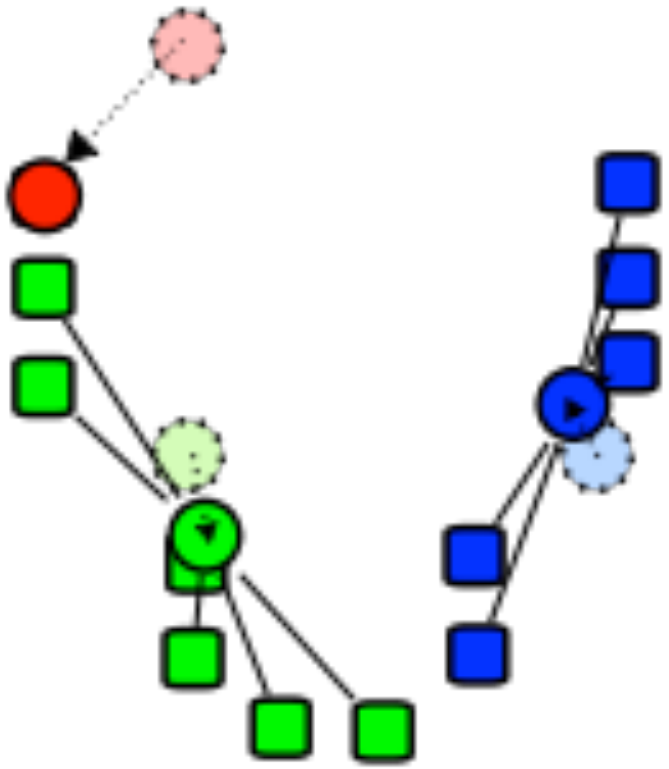
The K-means algorithm

1. Choose *k* starting points randomly. These are called the centroids.
2. Calculate the Euclidian distance between each point and the the centroids.
3. Color each point according to the nearest centroid.
4. Recalculate the mean of each centroid as the mean of the points of the same color.



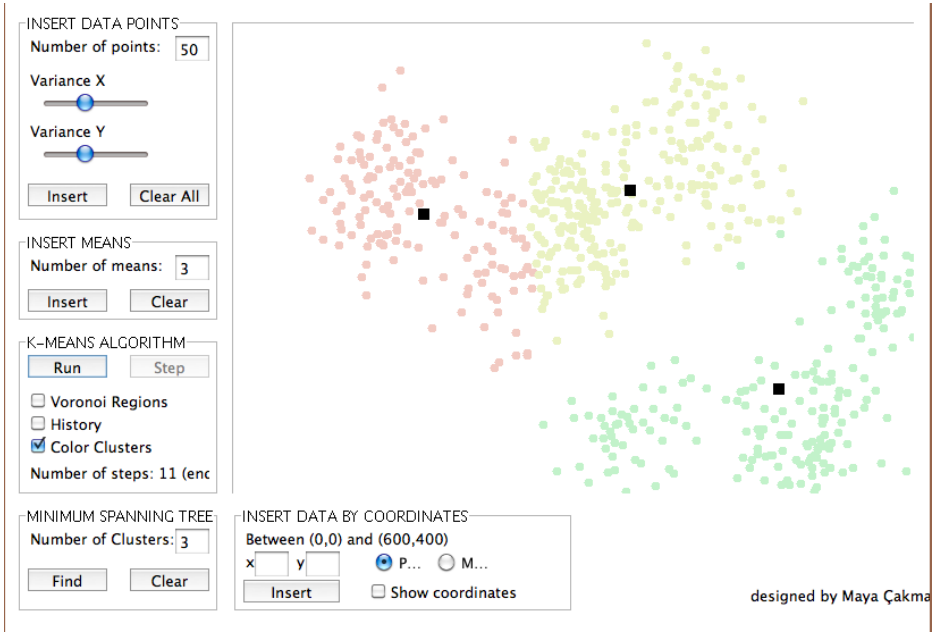
The K-means algorithm

1. Choose *k* starting points randomly. These are called the centroids.
2. Calculate the Euclidian distance between each point and the the centroids.
3. Color each point according to the nearest centroid.
4. Recalculate the mean of each centroid as the mean of the points of the same color.
5. Move the centroid to the new location.



The K-means algorithm

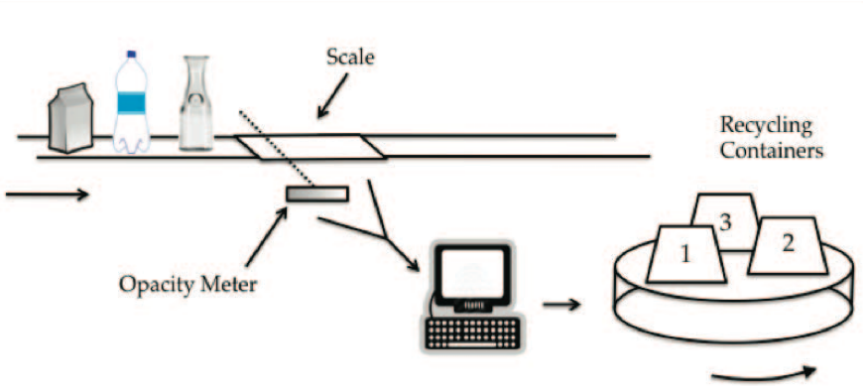
1. Choose *k* starting points randomly. These are called the centroids
2. Calculate the Euclidian distance between each point and the the centroids.
3. Color each point according to the nearest centroid.
4. Recalculate the mean of each centroid as the mean of the points of the same color.
5. Move the centroid to the mean location.
6. Repeat steps 1-5 until the centroids no longer move.



http://www.kovan.ceng.metu.edu.tr/~maya/kmeans/

Class exercise: Recycling

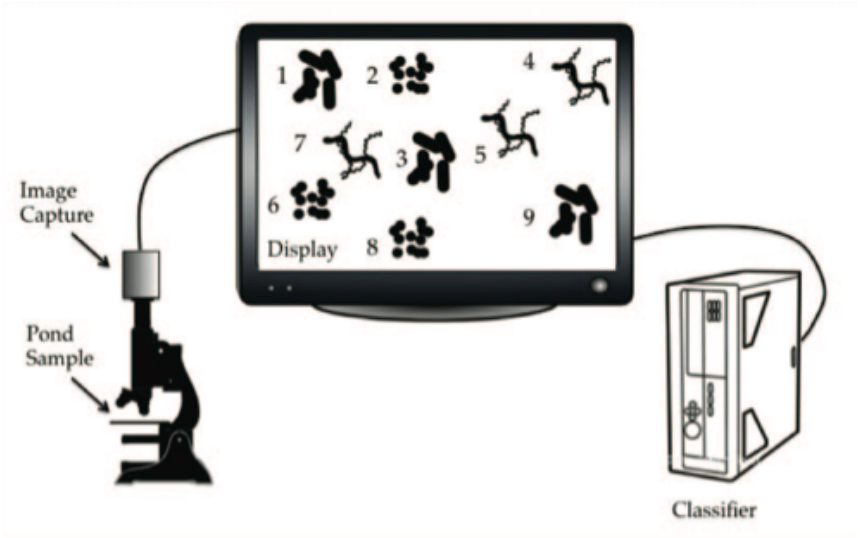
- Students hired to design a container sorting system at a recycling center.
- Design the machine learning portion of the system using K-means.
- Potential solution
- A conveyor belt moves the items toward spinning recycle bins.
- Weight and opacity is measured by a scale and a sensor.
- K-means clusters the items into three types
- The computer rotates the bins



Exercises from Steve Essinger and Gail Rosen's excellent article: "An Introduction to Machine Learning for Students in Secondary Education"

Class exercise: Cells

- Several water samples have been collected from a local pond
- Students learn about identifying and separating animal cells from plant cells, and developed an appreciation for the labor involved.
- They design an ML system to separate the cells.
- For complexity, we introduce a third type of cell, *Euglena*, which has both plant and animal features.



Exercises from Steve Essinger and Gail Rosen's excellent article: "An Introduction to Machine Learning for Students in Secondary Education"

Class discussion

- How do algorithms make recommendations from data?
- Why are features important?
- Would K-means work the same with more than 2 features?
- Could we visualize more than 2 features? More than 3?
- Think of how Euclidian distance is calculated. Do all the features need to be on the same scale?
- All the containers have to be analyzed before they can be placed in their appropriate bin. How does this affect the design of the conveyor belt?
- What challenges do *Euglena* cells present to the algorithm?

Summary

- Machine learning is the study of algorithms that learn from data
- Recommendation systems make decisions based on patterns in large datasets
- K-means can be used to automatically find clusters of similar data
- K-means uses Euclidian distance and features to determine similarity

Resources

- Articles
  - Steve Essinger and Gail Rosen. "*An Introduction to Machine Learning for Students in Secondary Education*," IEEE Signal Processing in Education Workshop, January, 2011.
- Textbooks
  - *"Programming Collective Intelligence"* by Toby Segaran. 2007. ISBN 978-0-596-52932-1.
- Thanks!