Data Science

Deriving Knowledge from Data at Scale

Wee Hyong Tok

July 14, 2016

Data Science

Deriving Knowledge from Data at Scale

*Feature extraction and selection are the **most important** but underrated step of machine learning.*

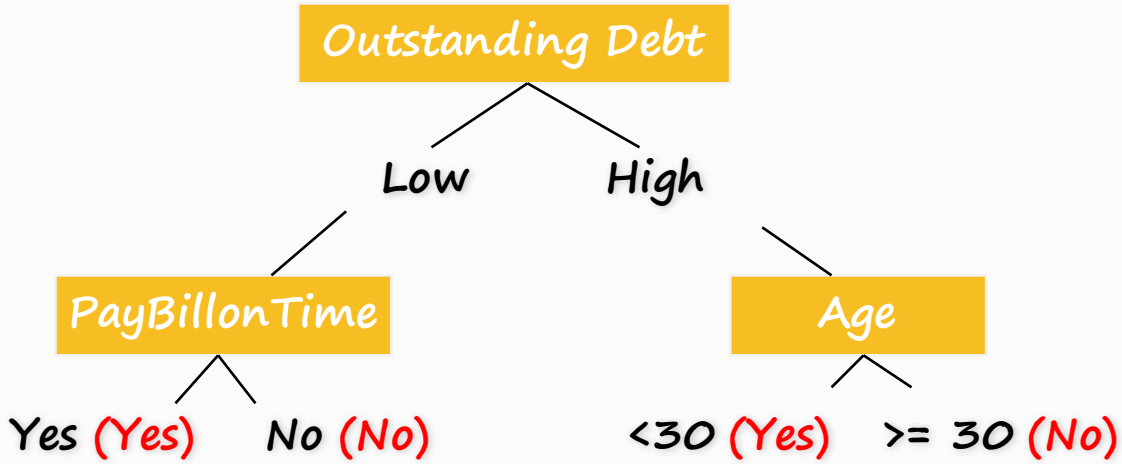
Better features are better than better algorithms…

Lecture Outline

- Model Evaluation
- Class Imbalance
- Clustering

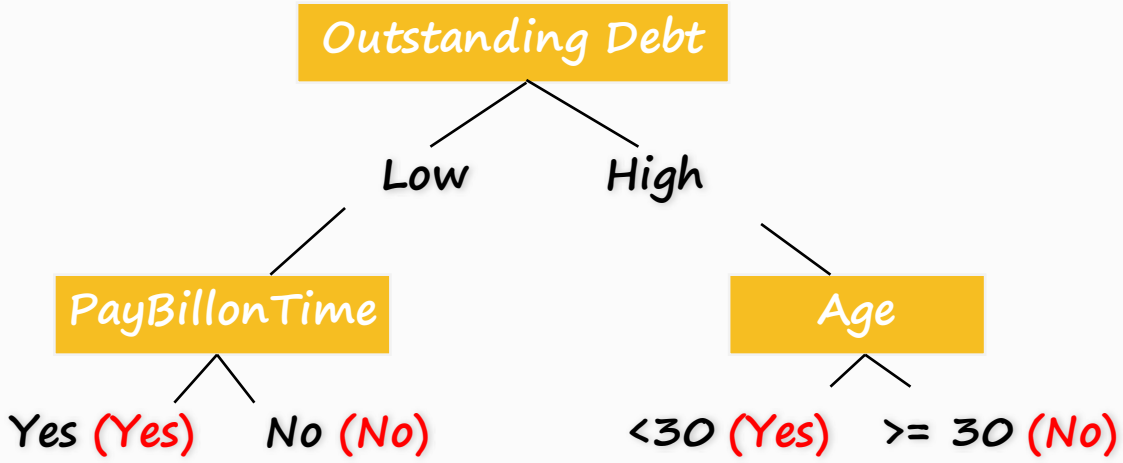
Let’s build our first decision tree

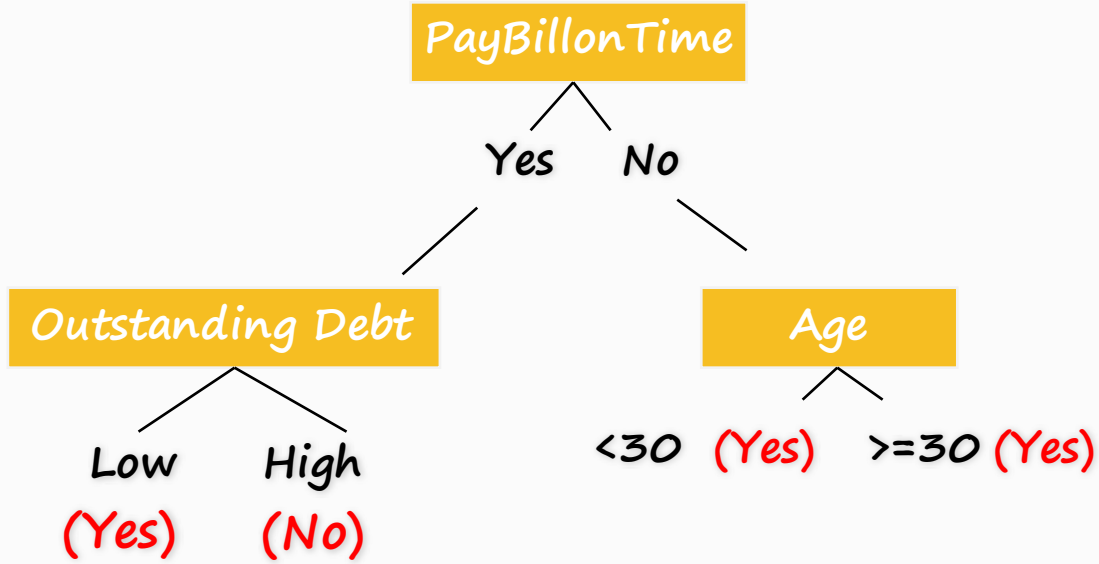
Decision trees classify instances by sorting them from the root of the tree to some leaf nodes (which provides the classification of the instance)



How do we split?

- Which attribute to be used first?
  - Outstanding Debt?
  - Pay bill on time
  - Age





Determining the Split Attribute

- Each attribute is assigned a score
- An attribute that maximizes the score during each iteration is chosen as the Split attribute
- Different methods to compute the score
  Example: Information Gain

Figuring out the Entropy

- Entropy – Measures the impurity/purity of an arbitraty collection of examples

- Binary Classification - Given a collection S, with positive (+) and negative (-) example of a target concept

$$\text{Entropy}(S) = -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$$

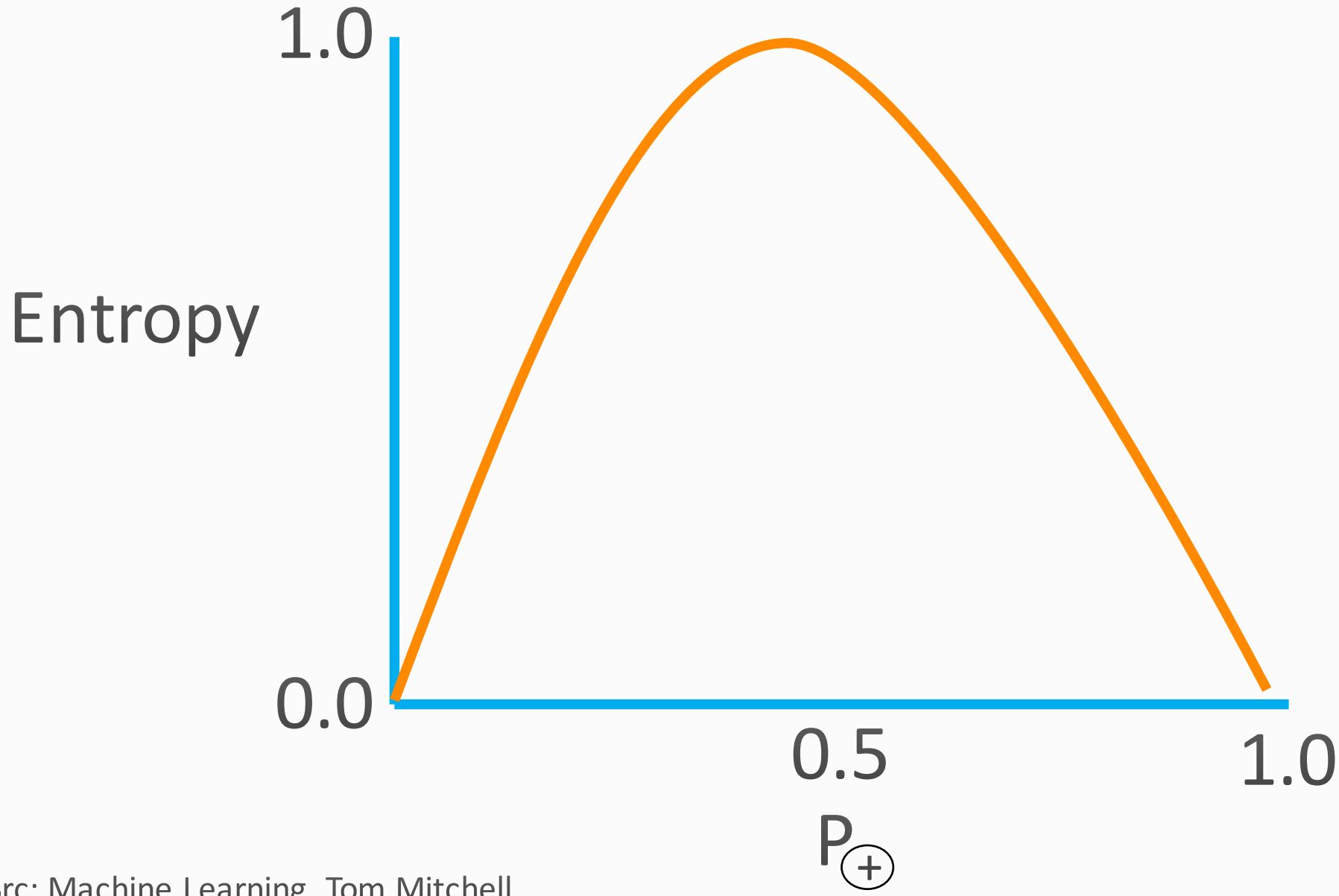
Example – Computing the Entropy

- S is a collection of 14 examples
- 9 are positive examples, 5 are negative examples
  Notation: [9+, 5-]
- Entropy(S)
  $= -(9/14)\log_2 (9/14) -(5/14)\log_2(5/14)$
  $= 0.940$

Question:
When will entropy be 1?
When will entropy be 0?

Src: Machine Learning, Tom Mitchell

Entropy vs P+



Src: Machine Learning, Tom Mitchell

Information Gain

- Information Gain – Expected reduction in entropy if attribute was chosen as the splitting attribute

$$\text{Gain (S, A)} = \text{Entropy(S)} - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Example – Computing the Entropy

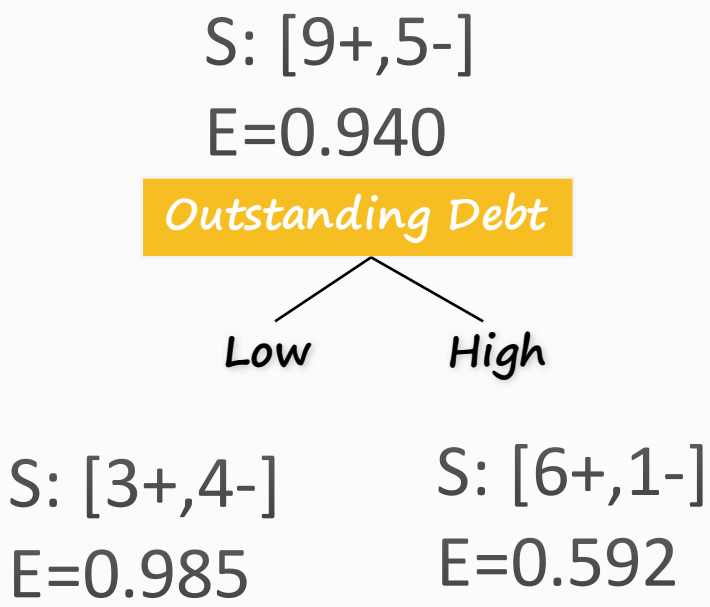
- S is a collection of 14 examples
- 9 are positive examples, 5 are negative examples
  Notation: [9+, 5-]
- PayBillOnTime = True : 6+ and 2-
  PayBillOnTime = False: 3+ and 3-

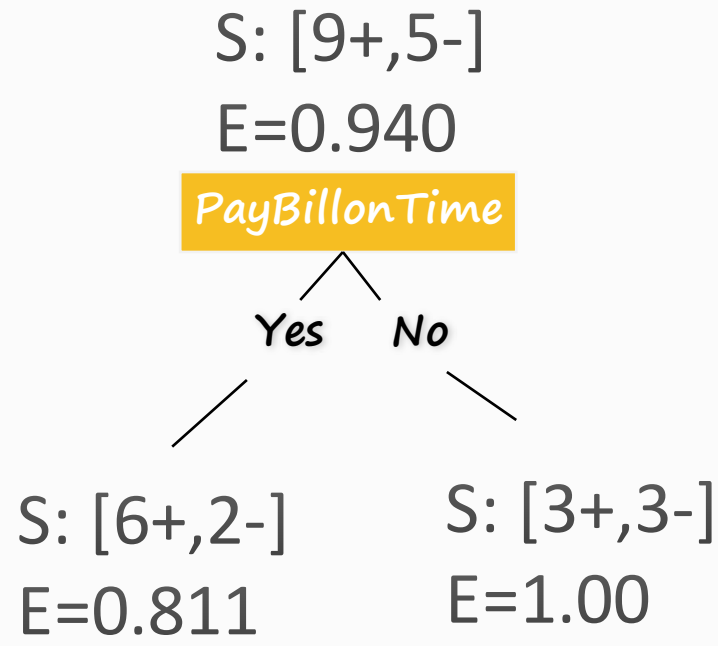
- Values(PayBillOnTime) = True, False

- Gain(S, LoanApproved) = 0.940
  – (8/14)Entropy(PayBillOnTime)
  – 6/14Entropy(Does not pay Bill on Time)
- = 0.940 – (8/14) 0.811 – (6/14) 1.00
  = 0.048

Src: Machine Learning, Tom Mitchell

How do we split?





Gain(S, OutstandingDebt) = 0.151

Gain(S, PayBillOnTime) = 0.048

Which provides more gain?

- Class Quiz – Which is better?

Cross Validation

- In cross validation, you break your training data up into K equally-sized partitions. You train a learning algorithm on K-1 of them and test it on the remaining 1. You do this K times, each time holding out a different partition as the "development" part. You can then average your performance over K runs.
- Suppose that you've presented a machine learning solution to your boss that achieves 7% error on cross validation. Your nemesis, Gabe, gives a solution to your boss that achieves 6.9% error on cross validation. How impressed should your boss be?
- **It depends**. If this 0.1% improvement was measured over 1000 examples, perhaps not too impressed. It would mean that Gabe got exactly *one more example right than you did*. (In fact, he probably got 15 more right and 14 more wrong.) If this 0.1% improvement was measured over 1,000,000 examples, perhaps this is more impressive.

How can you evaluate models?

Type I and Type II errors

- Different types of error losing different types of money…
- What is the cost (opportunity and actual) of a false positive?
- What is the cost of a false negative?

Gain Curves

Don’t forget the user

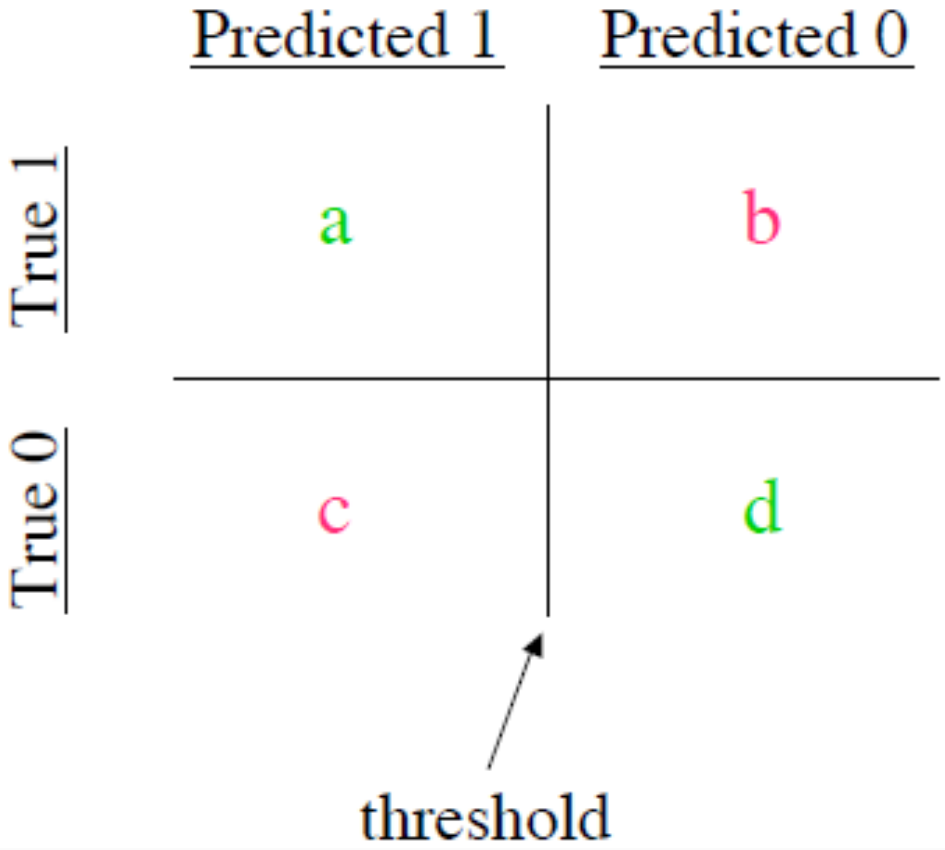
- They may need to be able to explain (or justify) their decision
- Decision trees fight back

Cost Considerations

- Cost / Profit Matrix
- Example: decide whether to offer Personal Equity Plan (PEP)
  - True Positive: Payoff is $1000/customer
  - True Negative: $0
  - False Positive: -$10/customer
  - False negative: -$1000/customer (opp. Cost)

Precision/Recall Curves

Precision and Recall



$$PRECISION = a/(a+c)$$

$$RECALL = a/(a+b)$$

Precision/Recall Curves

Once you can compute precision and recall, you are often able to produce precision/recall curves.

Suppose that you are attempting to identify spam.

You run a learning algorithm to make predictions on a test set. But instead of just taking a “yes/no” answer, you allow your algorithm to produce its confidence.

For instance, using a perceptron, you might use the distance from the hyperplane as a confidence measure. You can then sort all of your test emails according to this ranking. You may put the most spam-like emails at the top and the least spam-like emails at the bottom

Precision/Recall Curves

Once you can compute precision and recall, you are often able to produce precision/recall curves. Suppose that you are attempting to identify spam. You run a learning algorithm to make predictions on a test set. But instead of just taking a "yes/no" answer, you allow your algorithm to produce its confidence. For instance, using a perceptron, you might use the distance from the hyperplane as a confidence measure. You can then sort all of your test emails according to this ranking. You may put the most spam-like emails at the top and the least spam-like emails at the bottom

Once you have this sorted list, you can choose how aggressively you want your spam filter to be by setting a threshold anywhere on this list. One would hope that if you set the threshold very high, you are likely to have high precision (but low recall). If you set the threshold very low, you'll have high recall (but low precision). By considering every possible place you could put this threshold, you can trace out a curve of precision/recall values. This allows us to ask the question: for some fixed precision, what sort of
recall can I get…

F Score

Sometimes we want a single number that informs us of the quality of the solution. A popular way to combine precision and recall into a single number is by taking their harmonic mean. This is known as the balanced f-measure:

$$F = \frac{2 \times P \times R}{P + R}$$

The reason to use a harmonic mean rather than an arithmetic mean is that it favors systems that achieve roughly equal precision and recall. In the extreme case where P = R, then F = P = R. But in the imbalanced case, for instance P = 0.1 and R = 0.9, the overall f-measure is a modest 0.18.

	0.0	0.2	0.4	0.6	0.8	1.0
0.0	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.20	0.26	0.30	0.32	0.33
0.4	0.00	0.26	0.40	0.48	0.53	0.57
0.6	0.00	0.30	0.48	0.60	0.68	0.74
0.8	0.00	0.32	0.53	0.68	0.80	0.88
1.0	0.00	0.33	0.57	0.74	0.88	1.00

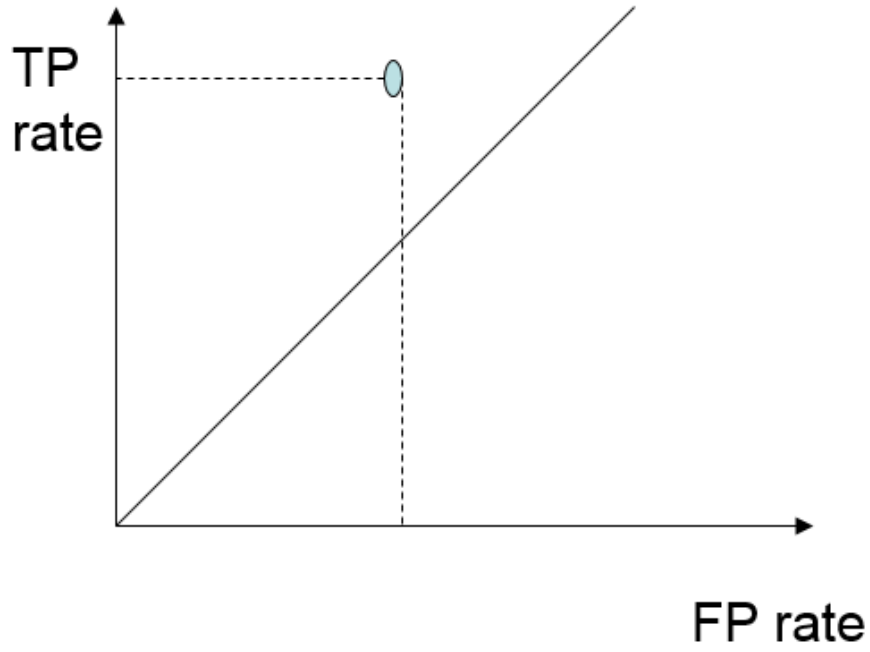
Note…

One thing to keep in mind is that precision and recall (and hence f-measure) ***depend crucially on which class is considered*** the thing you wish to find. In particular, if you take a binary data set and flip what it means to be a positive or negative example, you will end up with completely different precision and recall values.

It is ***not the case that precision on the flipped task is equal to recall on the original task*** (nor vice versa). Consequently, f-measure is also not the same. For some tasks where you are less sure about what you want, report two sets of precision/recall/f-measure numbers, which vary based on which class is considered the thing to spot.

Point in ROC Space

		TRUE CLASS	
		YES	NO
PREDICTED CLASS	YES	TP	FP
	NO	FN	TN
Total:		P	N



FP rate: FP/N TP rate: TP/P (recall)

FN rate: FN/N TN rate: TN/P

Classifier accuracy: (TP+TN)/(P+N)

Shows how good is classifier in discriminating positive instances from negative ones

ROC Curve of a Probabilistic Classifier

Naïve Bayes, for example, outputs the probability of an instance in a testing set to be classified as YES

Outlook	Temp	Windy	**P(Y|E)**	Real class
overcast	mild	yes	**0.95**	YES
rainy	mild	no	**0.80**	YES
rainy	cool	yes	**0.60**	NO
sunny	mild	no	**0.45**	YES
sunny	cool	no	**0.40**	NO
sunny	hot	no	**0.35**	NO
sunny	hot	yes	**0.25**	NO

ROC Curve of a Probabilistic Classifier

In a general case, we classify an instance as YES if the probability is more than 50%

Classified as YES

Classified as NO

Outlook	Temp	Windy	P(Y|E)	Real class
overcast	mild	yes	**0.95**	YES
rainy	mild	no	**0.80**	YES
rainy	cool	yes	**0.60**	NO
sunny	mild	no	**0.45**	YES
sunny	cool	no	**0.40**	NO
sunny	hot	no	**0.35**	NO
sunny	hot	yes	**0.25**	NO

Operating threshold

ROC Curve of a Probabilistic Classifier

We compute the confusion matrix

		TRUE CLASS	
		YES	NO
PREDICTED CLASS	YES	2 (TP)	1 (FP)
	NO	1 (FN)	3 (TN)
Total:		3 (P)	4 (N)

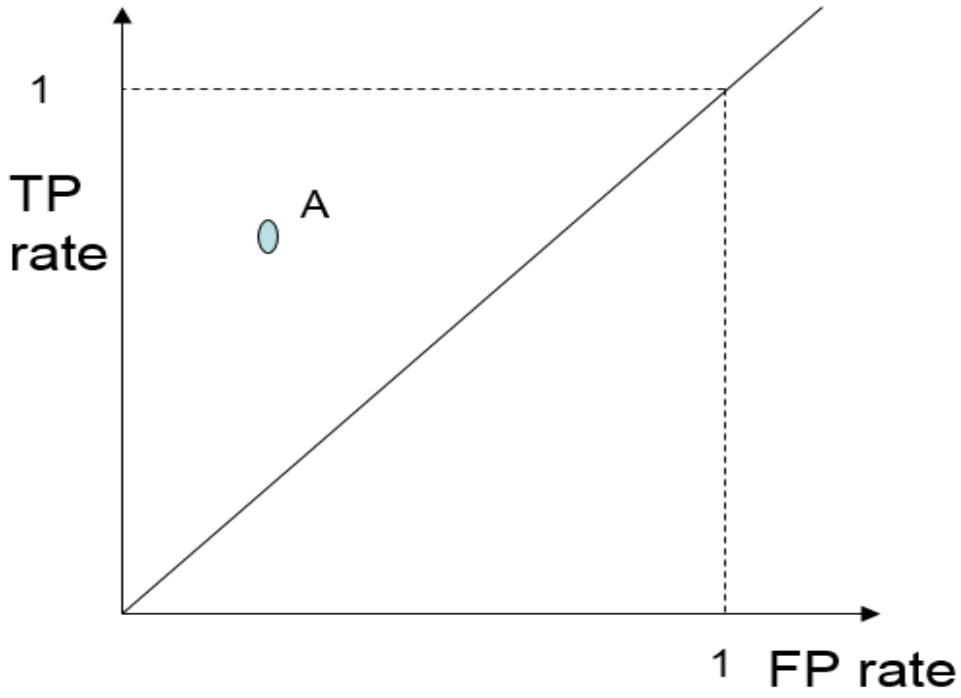
Outlook	Temp	Windy	P(Y|E)	Predicted class	Real class
overcast	mild	yes	0.95	YES	YES
rainy	mild	no	0.80	YES	YES
rainy	cool	yes	0.60	YES	NO
sunny	mild	no	0.45	NO	YES
sunny	cool	no	0.40	NO	NO
sunny	hot	no	0.35	NO	NO
sunny	hot	yes	0.25	NO	NO

And the TP and FP rates:

TP rate: TP/P=2/3≈0.7

FP rate: FP/N=1/4=0.25

ROC Curve of a Probabilistic Classifier



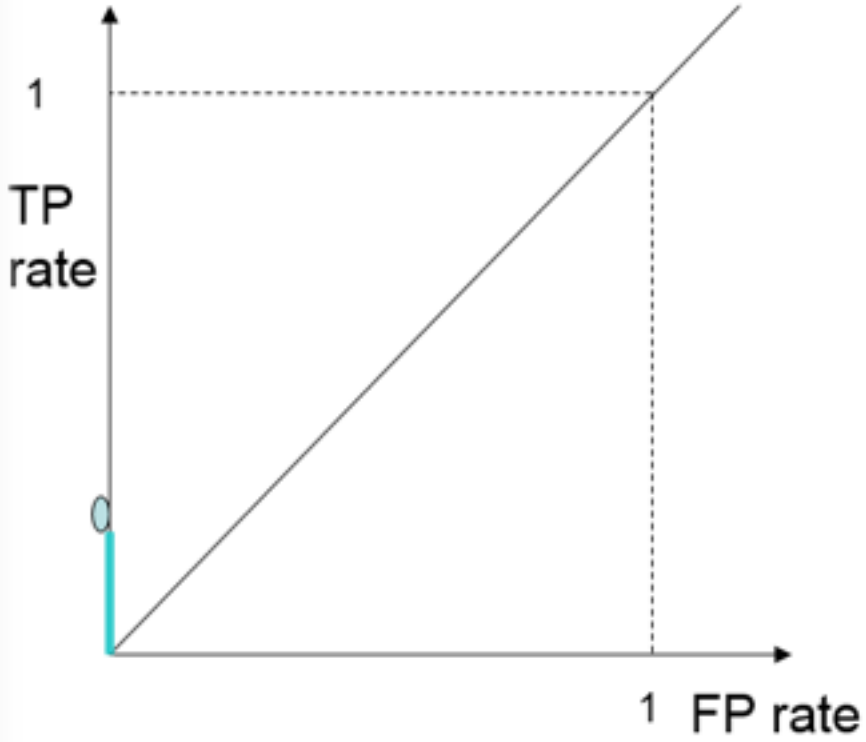
This corresponds to point A in a ROC space

FP rate: FP/N=1/4=0.25

TP rate: TP/P=2/3≈0.7

Outlook	Temp	Windy	P(Y|E)	Predicted class	Real class
overcast	mild	yes	0.95	YES	YES
rainy	mild	no	0.80	YES	YES
rainy	cool	yes	0.60	YES	NO
sunny	mild	no	0.45	NO	YES
sunny	cool	no	0.40	NO	NO
sunny	hot	no	0.35	NO	NO
sunny	hot	yes	0.25	NO	NO

ROC Curve of a Probabilistic Classifier



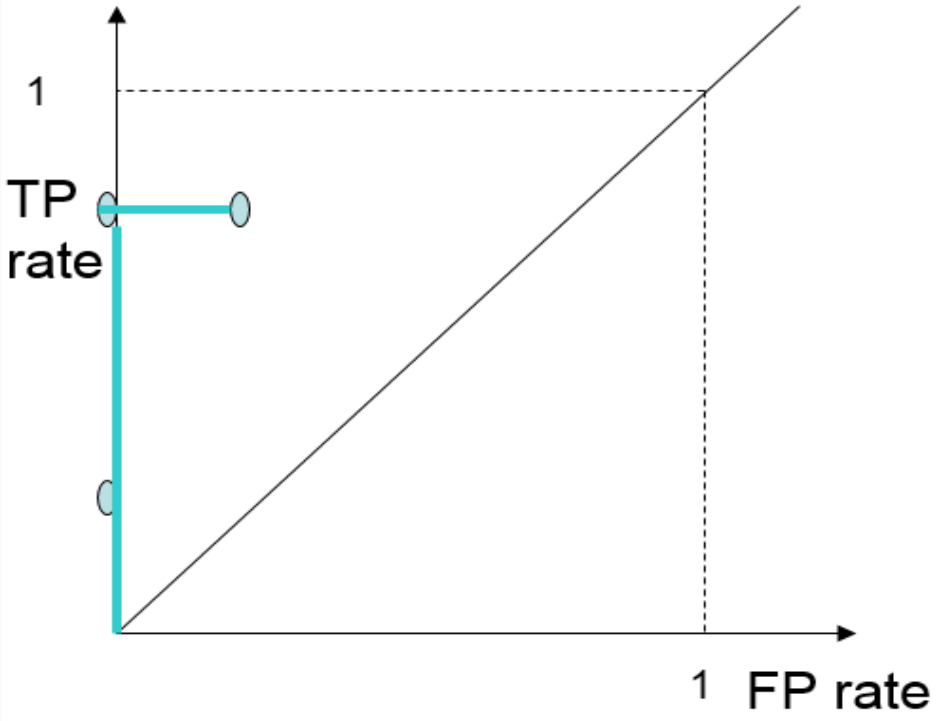
For different threshold values we get different points in a ROC space

Outlook	Temp	Windy	P(Y|E)	Predicted class	Real class
overcast	mild	yes	0.95	YES	YES
rainy	mild	no	0.80	YES	YES
rainy	cool	yes	0.60	YES	NO
sunny	mild	no	0.45	NO	YES
sunny	cool	no	0.40	NO	NO
sunny	hot	no	0.35	NO	NO
sunny	hot	yes	0.25	NO	NO

FP rate: FP/N=0/4=0

TP rate: TP/P=1/3≈0.3

ROC Curve of a Probabilistic Classifier



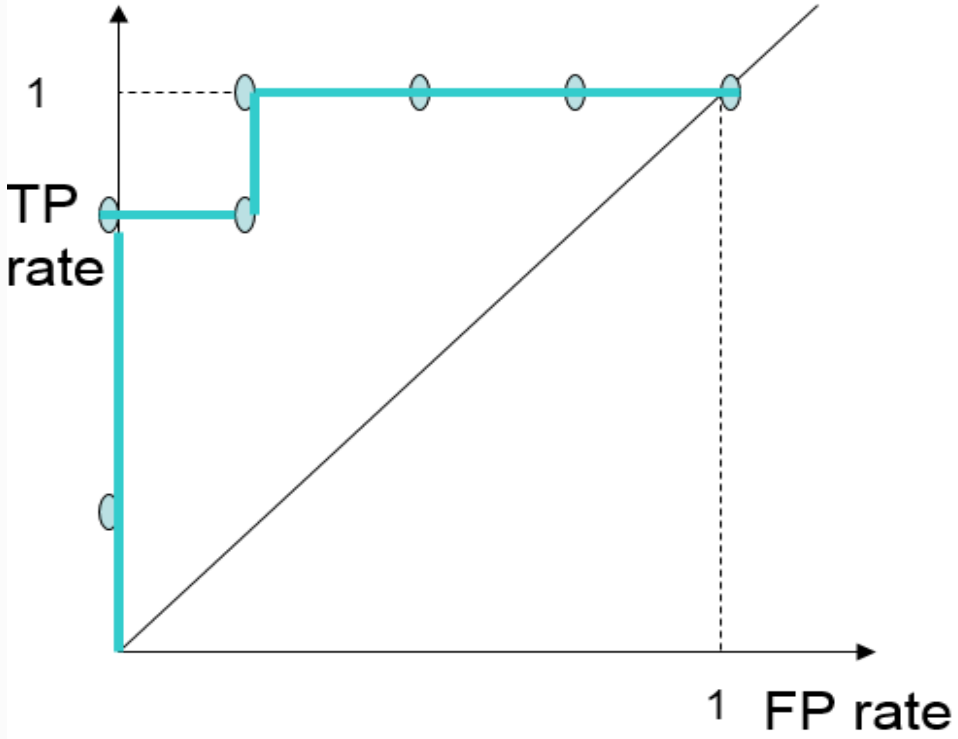
For different threshold values we get different points in a ROC space

Outlook	Temp	Windy	P(Y|E)	Predicted class	Real class
overcast	mild	yes	0.95	YES	YES
rainy	mild	no	0.80	YES	YES
rainy	cool	yes	0.60	YES	NO
sunny	mild	no	0.45	NO	YES
sunny	cool	no	0.40	NO	NO
sunny	hot	no	0.35	NO	NO
sunny	hot	yes	0.25	NO	NO

FP rate: FP/N=1/4=0.25

TP rate: TP/P=2/3≈0.7

ROC Curve of a Probabilistic Classifier



For different threshold values we get different points in a ROC space

Outlook	Temp	Windy	P(Y|E)	Predicted class	Real class
overcast	mild	yes	0.95	YES	YES
rainy	mild	no	0.80	YES	YES
rainy	cool	yes	0.60	YES	NO
sunny	mild	no	0.45	YES	YES
sunny	cool	no	0.40	NO	NO
sunny	hot	no	0.35	NO	NO
sunny	hot	yes	0.25	NO	NO

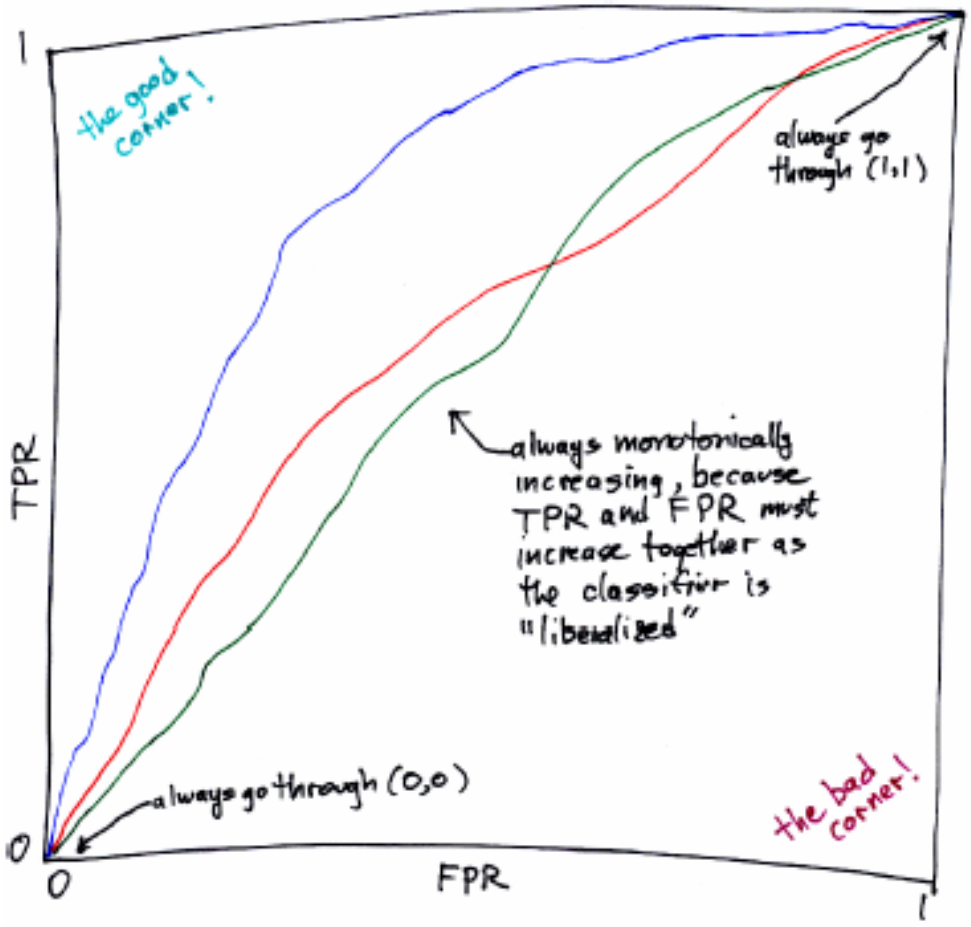
FP rate: FP/N=1/4=0.25

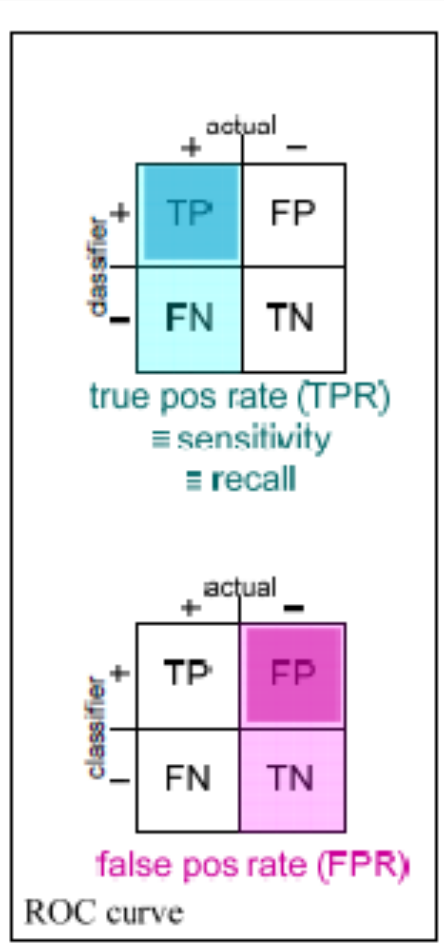
TP rate: TP/P=3/3=1.0, etc…

Performance Metrics

Receiver operating characteristic (ROC) curve

- True positive rate vs. False positive rate for various operating points, as the classifier goes from “conservative” to “liberal”

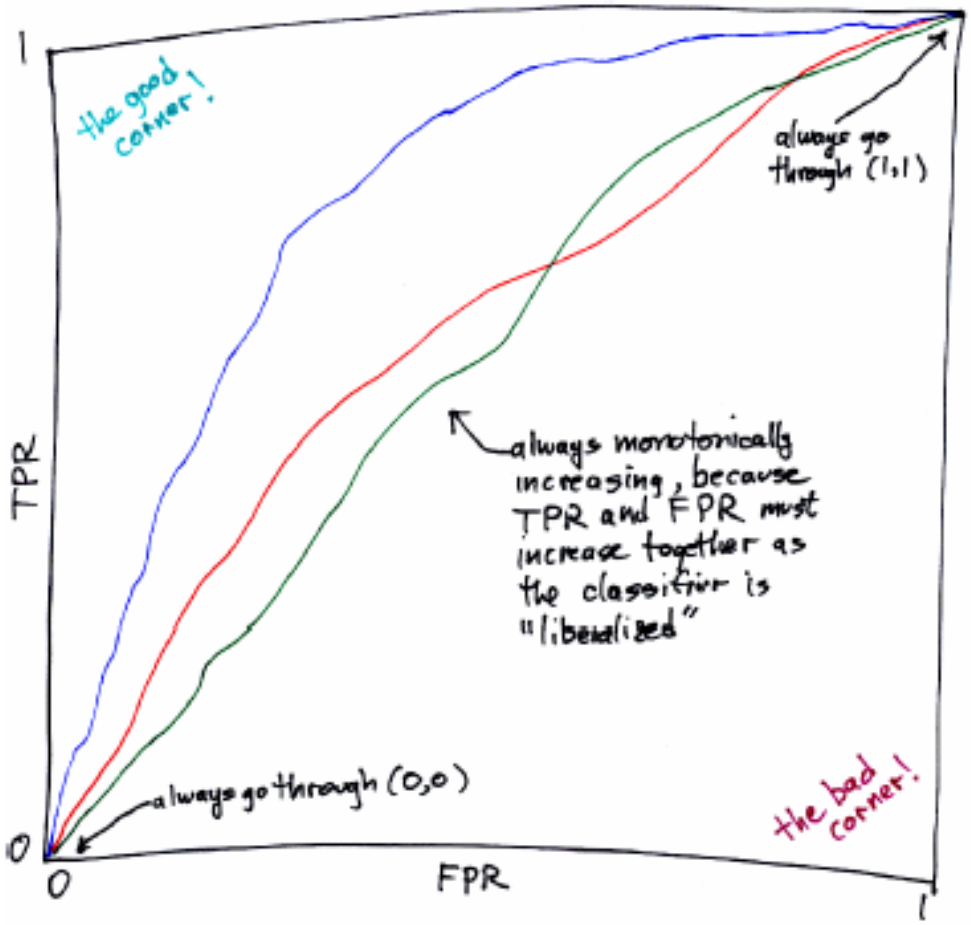




Performance Metrics

Receiver operating characteristic (ROC) curve

- True positive rate vs. False positive rate for various operating points, as the classifier goes from "conservative" to "liberal"



blue dominates red and green

neither red nor green dominate the other

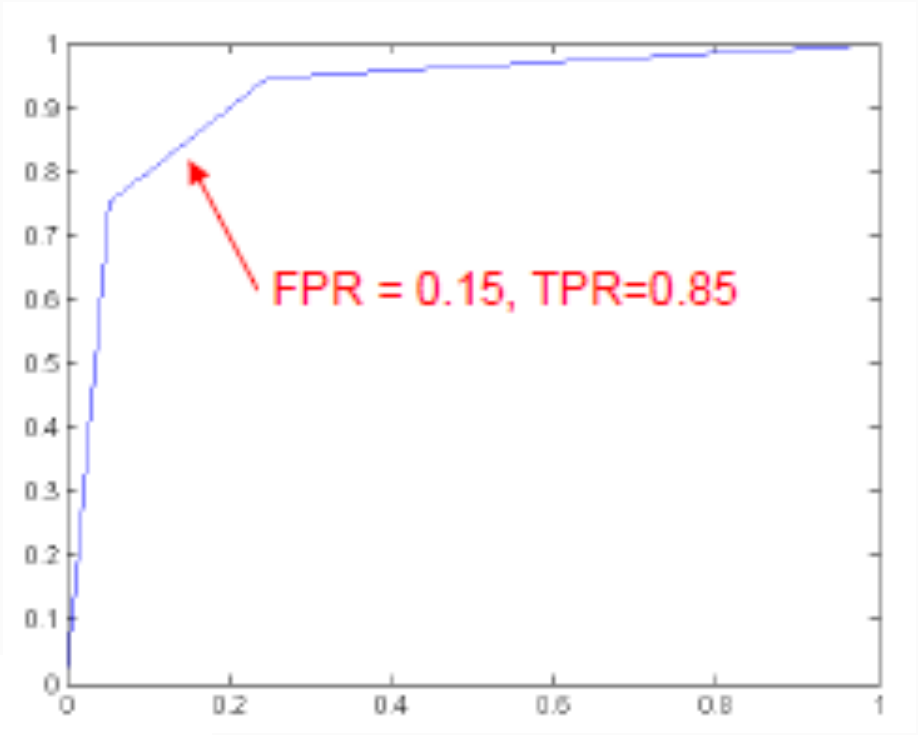
You could get the best of the red and green curves by making a hybrid classifier that switches between strategies at the cross-over points.

Performance Metrics

Since ROC curves don't explicitly show any dependence on the constant P/N (ratio of actual + to – in the sample) they can be misleading if you care about FP versus TP.

Suppose you have a test for Alzheimer's whose false positive rate can be varied from 5% to 25% as the false negative rate varies from 25% to 5% (suppose linear dependences on both):

You try the test on a population of 10,000 people, 1% of whom actually are Alzheimer's positive:



classifier \ actual	+	–
+	85	1485
–	15	8415

FP swamps TP by ~17:1. You'll be telling 17 people that they might have Alzheimer's for every one who actually does. It is unlikely that your test will be used.

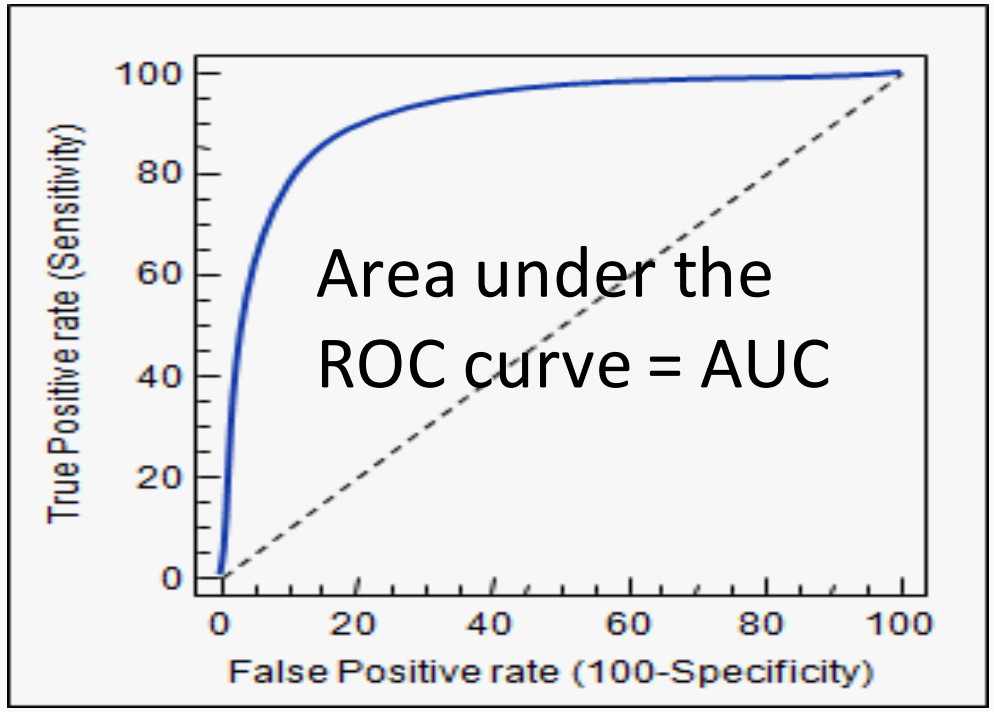
In a case like this, ROC, while correct, somewhat misses the point.

ROC Curve

- Area under the ROC curve (AUC) is a measure of the model performance

$$0.5\ (random\ model) < AUC < 1\ (perfect\ model)$$

- Larger the AUC, better is the model



Performance Metrics

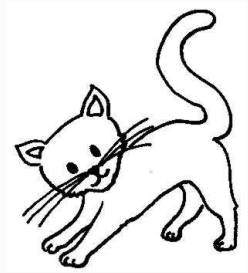
Receiver operating characteristic (ROC) curve, AUC metric…

- 1.0: perfect prediction
- 0.9: excellent prediction
- 0.8: good prediction
- 0.7: mediocre prediction
- 0.6: poor prediction
- 0.5: random prediction
- <0.5: something wrong!

Dealing with Class Imbalance

Distributions Matter…

Because the Internet is all about cute kittens



Resulting in highly skewed distribution in training set…

The Class Imbalance Problem I

- Data sets are said to be balanced if there are, approximately, as many positive examples of the concept as there are negative ones.
- Many domains that do not have a balanced data set.

Examples:

- Helicopter Gearbox Fault Monitoring
- Discrimination between Earthquakes and Nuclear Explosions
- Document Filtering
- Detection of Oil Spills
- Detection of Fraudulent Telephone Calls

The Class Imbalance Problem II

- The problem with class imbalances is that standard learners are often biased towards the majority class.
- That is because these classifiers attempt to reduce global quantities such as the error rate, not taking the data distribution into consideration.
- As a result examples from the overwhelming class are well-classified whereas examples from the minority class tend to be misclassified.

Some Generalities

- Evaluating performance of a model on a class imbalance problem is not done appropriately with standard accuracy/error rate.
  - ROC Analysis is typically used, instead.

- There are three main ways to deal with class imbalances: re-sampling, re-weighing, and one-class learning (cover in SVMs)

- Re-sampling provides a simple way of biasing generalization process.
- It can do so by:
  - Generating synthetic samples accordingly biased
  - Controlling the amount and placement of the new samples

SMOTE: A State-of-the-Art Resampling Approach

- SMOTE stands for Synthetic Minority Oversampling Technique.
  - Technique designed by Chawla, Hall, & Kegelmeyer in 2002.

- It combines Informed **oversampling** of the **minority class** with **random undersampling** of the **majority class**.

- SMOTE currently yields the best results as far as re-sampling and modifying the probabilistic estimate techniques go (Chawla, 2003).

SMOTE's Informed Oversampling Procedure II

For each minority Sample

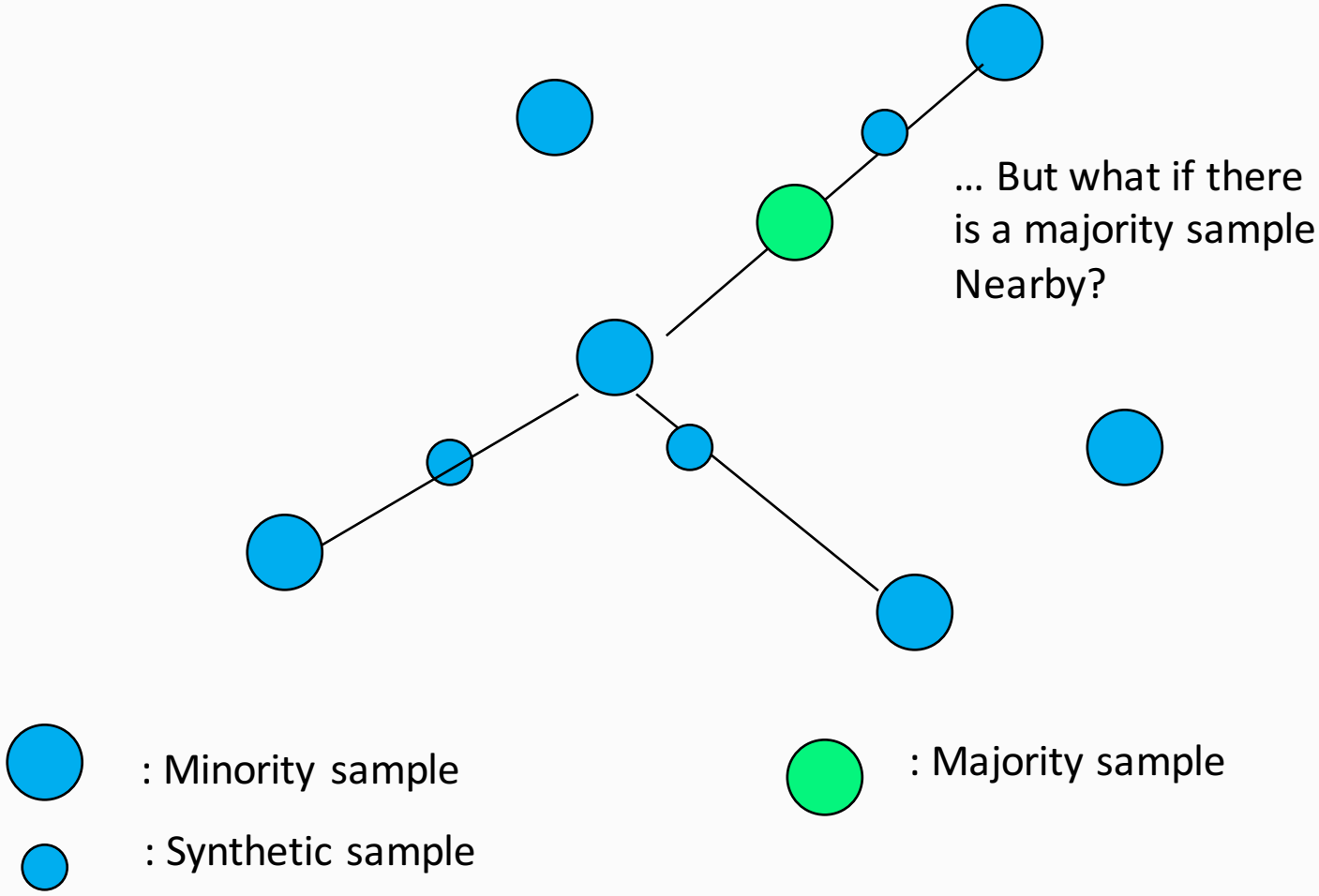
- Find its k-nearest minority neighbors
- Randomly select j of these neighbors
- Randomly generate synthetic samples along the lines joining the minority sample and its j selected neighbors

  (j depends on the amount of oversampling desired)

SMOTE's Informed vs. Random Oversampling

- Random Oversampling (with replacement) of the minority class has the effect of making the decision region for the minority class very specific.
- In a decision tree, it would cause a new split and often leads to overfitting.
- SMOTE's informed oversampling generalizes the decision region for the minority class.
- As a result, larger and less specific regions are learned, thus, paying attention to minority class samples without causing overfitting.

SMOTE's Informed Oversampling Procedure I



SMOTE's Shortcomings

- Overgeneralization
  - SMOTE's procedure can be dangerous since it blindly generalizes the minority area without regard to the majority class.
  - This strategy is problematic in the case of highly skewed class distributions since, in such cases, the minority class is very sparse with respect to the majority class, thus resulting in a greater chance of class mixture.

- Lack of Flexibility
  - The number of synthetic samples generated by SMOTE is fixed in advance, thus not allowing for any flexibility in the re-balancing rate.

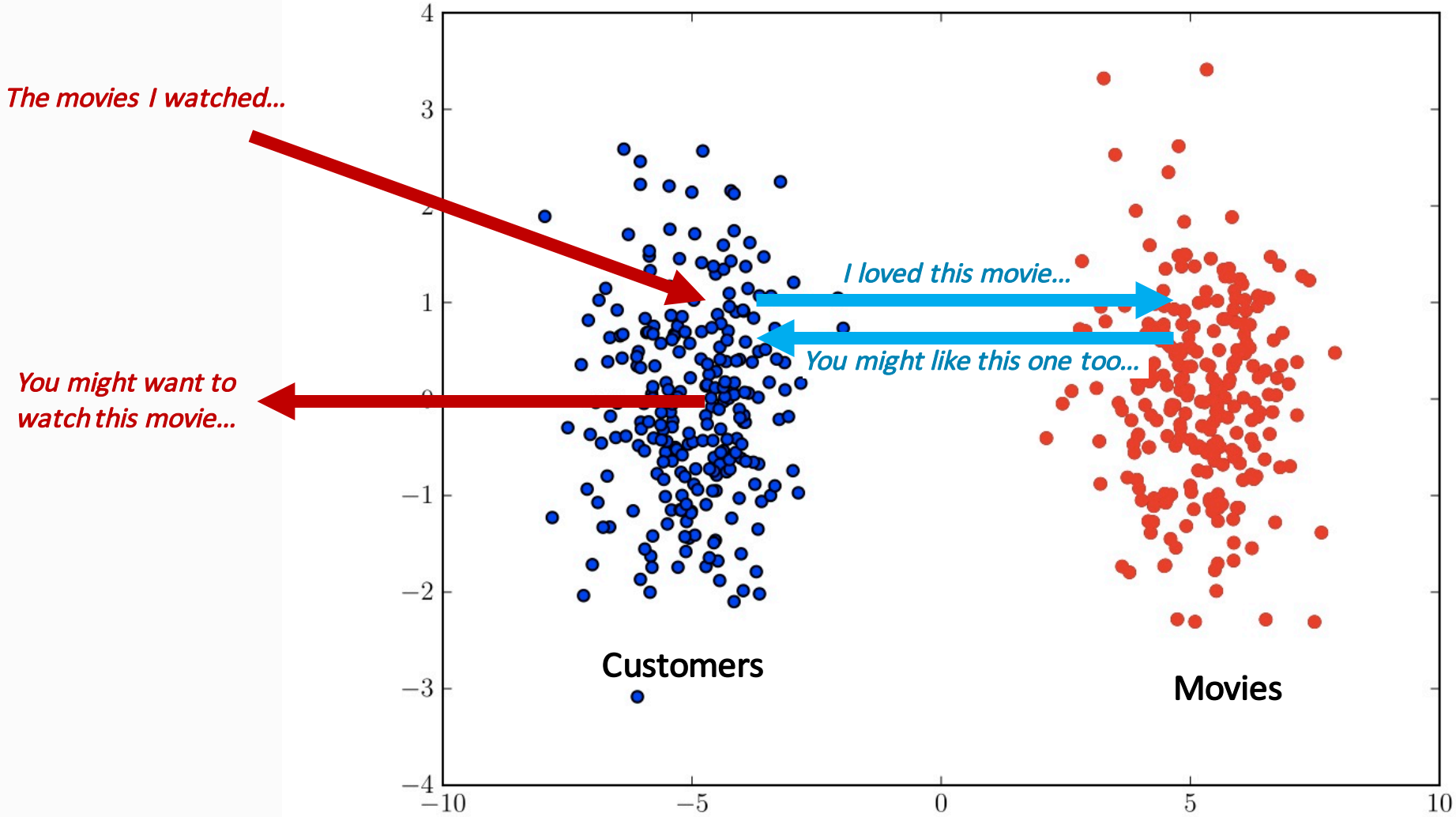
Clustering

Clustering

Fundamental Concepts: Calculating similarity of objects described by data; Using similarity for prediction; Clustering as similarity-based segmentation.

Exemplary Techniques: Searching for similar entities; Nearest neighbor methods; Clustering methods; Distance metrics for calculating similarity.

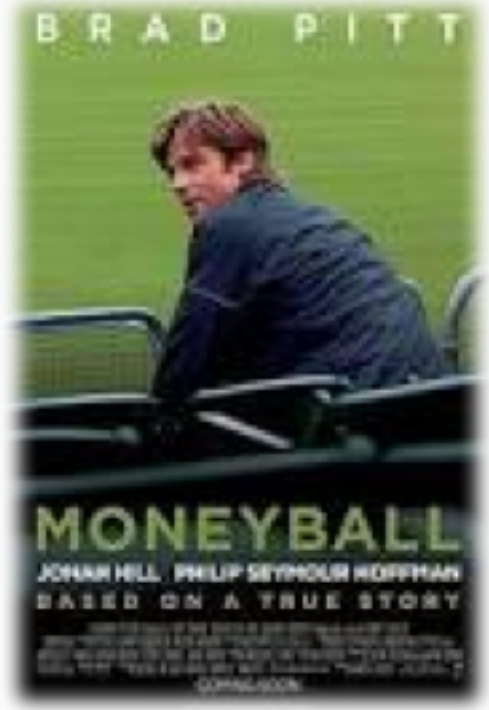
Clustering movies and movie watchers…



Moneyball, *for real…*

Clustering Pitchers

Hitters' Questions	Model Data
What does he throw?	• Top 2 Pitches • Pitch Repertoire/Variety • Horizontal Pitch Location • Vertical Pitch Location
How hard does he throw?	• FB Velocity
What kind of movement?	• Horizontal Movement • Vertical Movement
Where does he come from?	• Release Point
How does he like to pitch?	• Swinging Strike % • Zone % • Edge % • Top 2-pitch Sequence



Clustering

- Cluster images based on key features most useful for identification
- Use Cluster ID as index over images for exact match & retrieval

Clustering

We may want to ***retrieve*** similar things directly. For example, IBM wants to find companies that are *similar to their best business customers*, in order to have sales staff look at them as prospects. Hewlett-Packard maintains many high performance servers for clients; this maintenance is aided by a tool that, given a server configuration, *retrieves information on other similarly configured servers*.

We may want to group similar items together into clusters, for example to see whether our *customer base contains groups of similar customers* and what these groups have in common.

Reasoning from similar cases of course extends beyond business applications; it is natural to fields such as medicine and law. A doctor may *reason about a new difficult case by recalling a similar case and its diagnosis*. A lawyer often argues cases by citing legal precedents, which are *similar historical cases whose dispositions were previously judged and entered into the legal casebook*.

What is Clustering?

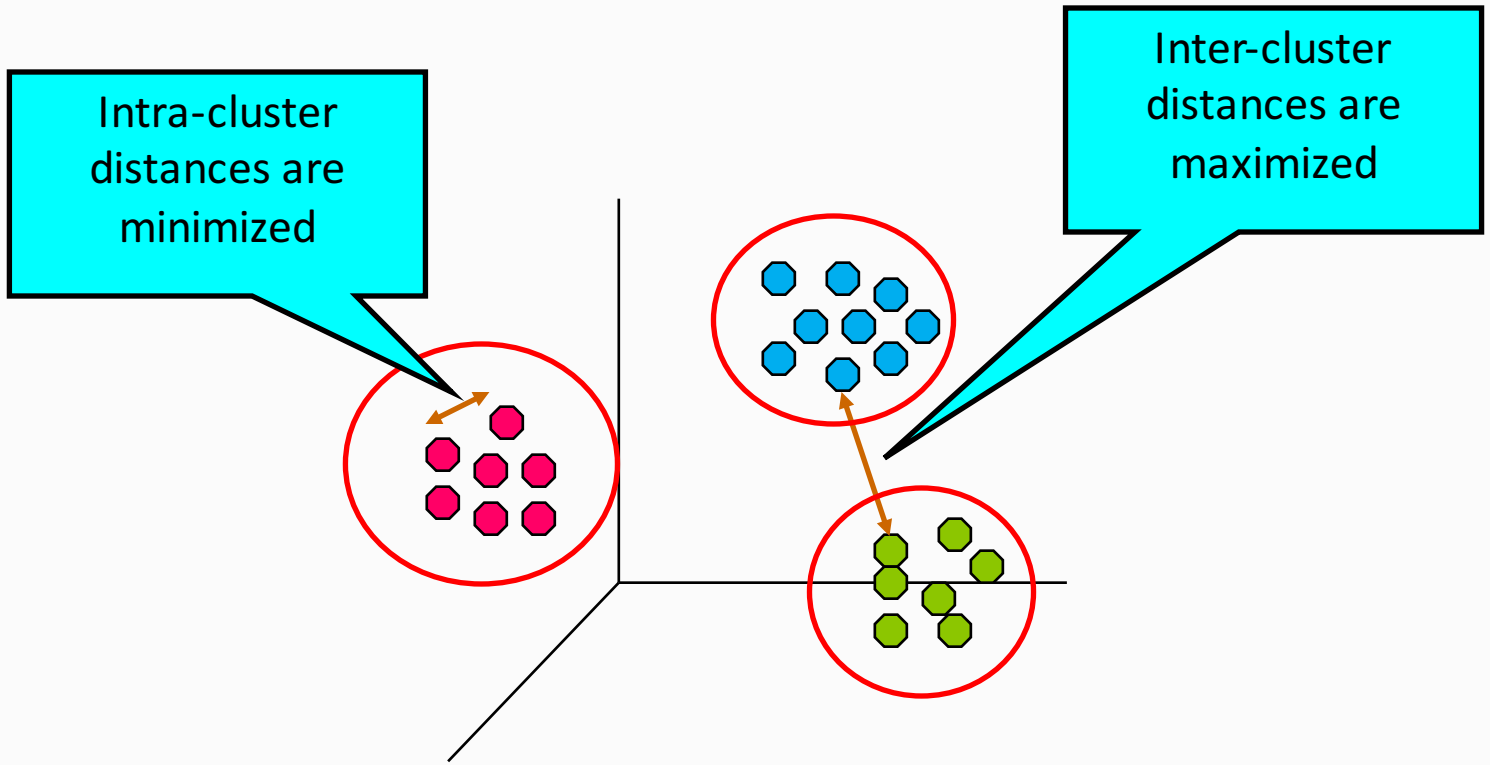
A way of grouping together data samples that are ***similar*** in some way - according to some criteria that you pick

A form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together. *The most important unsupervised learning problem, imho…*

So, it's a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest

What is clustering?

A **grouping** of data objects such that the objects **within a group are similar** (or related) to one another **and different from (or unrelated to) the objects in other groups**

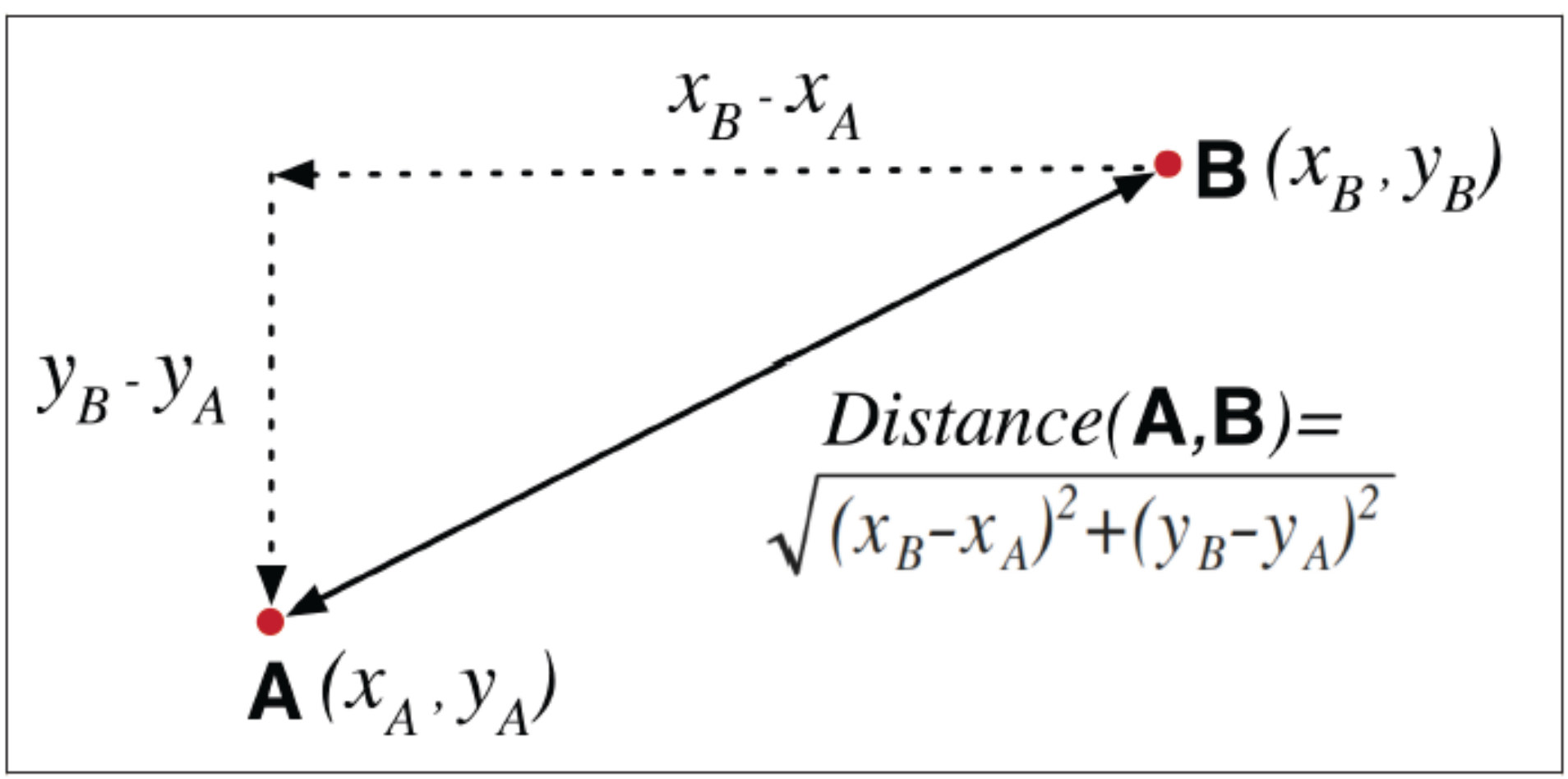


How do we define “similarity”?

- Goal is to group together “similar” data – but what does this mean?
- No single answer, it depends on what we want to find or emphasize in the data; this is one reason why clustering is an “art”
- The similarity measure is often more important than the clustering algorithm used – don’t overlook this choice!
- There is no golden standard, depends on goal: data reduction, “natural clusters”, “useful” clusters, outlier detection, etc.

Similarity Measures

- Euclidean Distance



Similarity Measures

Manhattan Distance

$$d_{\text{Manhattan}}(\mathbf{X},\mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_1 = | x_1 - y_1 | + | x_2 - y_2 | + \cdots$$

Jaccard Distance

$$d_{\text{Jaccard}}(X, Y) = 1 - \frac{| X \cap Y |}{| X \cup Y |}$$

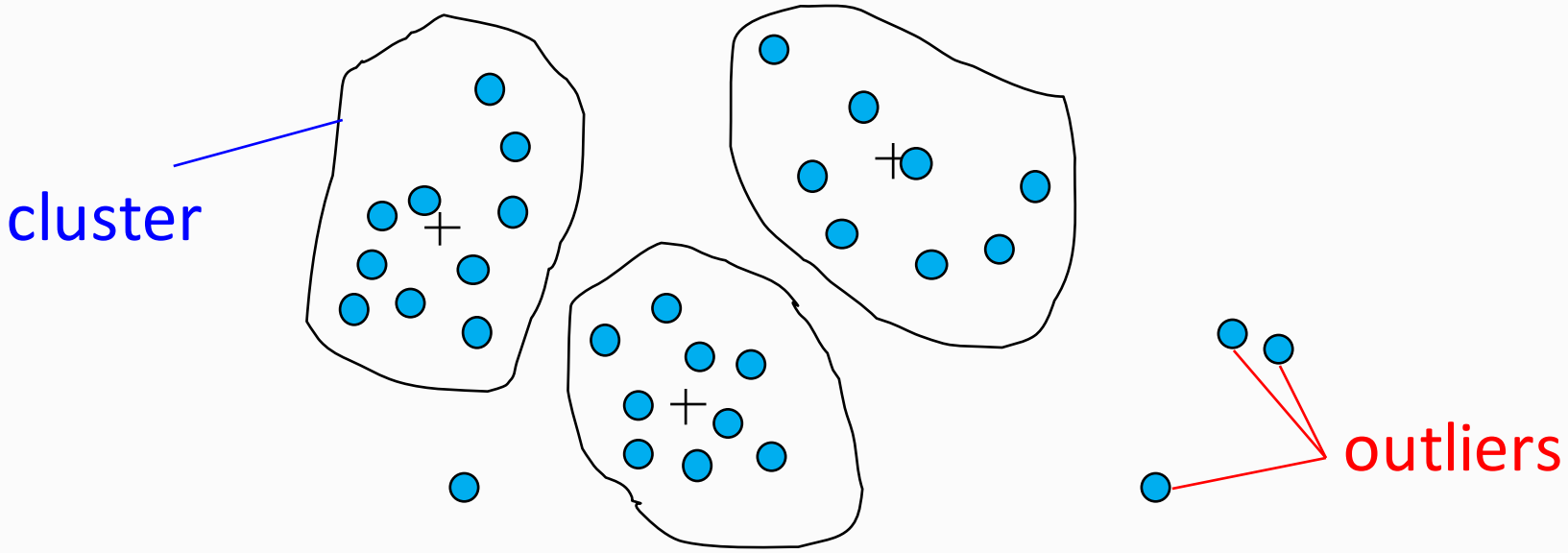
Edit Distance

1. 1113 Bleaker St.
2. 113 Bleecker St.

1. Delete a 1,
2. Insert a c, and
3. Replace an a with an e.

Outliers

- **Outliers** are **objects that do not belong to any cluster** or form clusters of very small cardinality



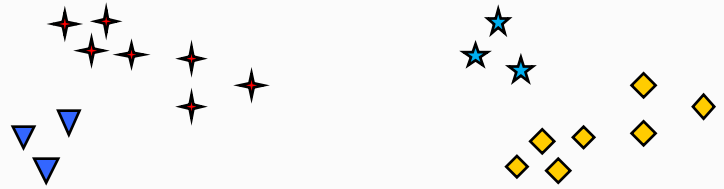
In some applications we are interested in discovering outliers, not clusters (**outlier analysis**)

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

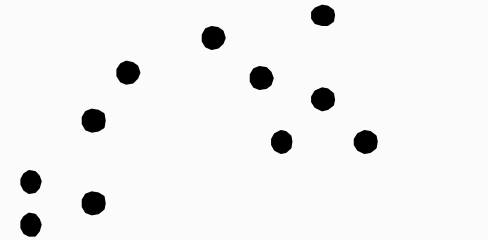


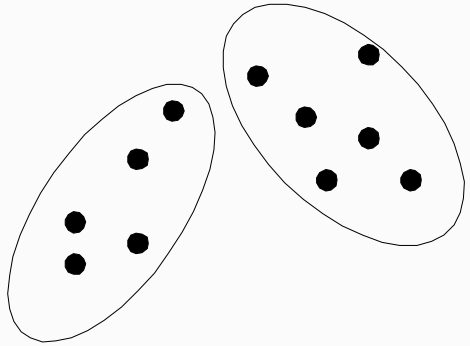
Four Clusters

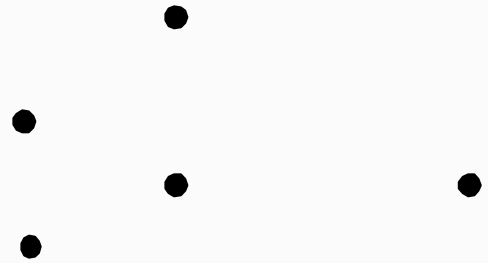
Types of Clusterings

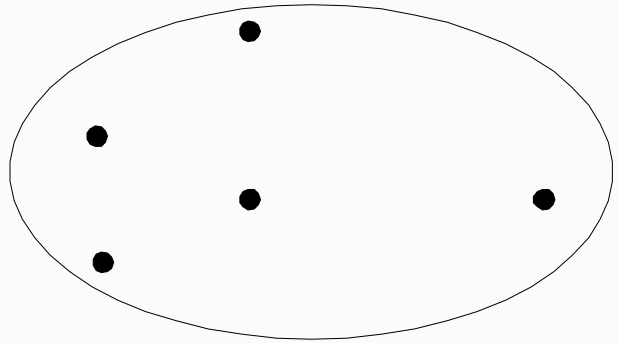
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering





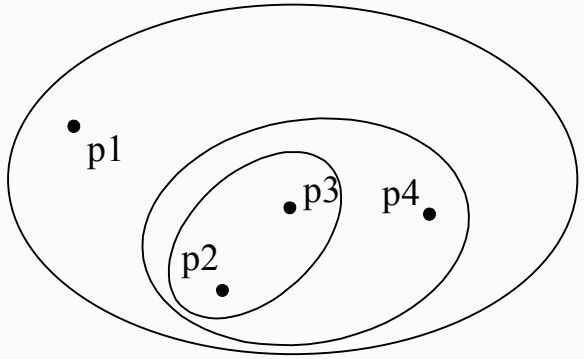




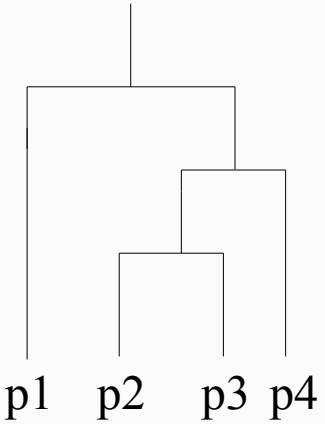
Original Points

A Partitional Clustering

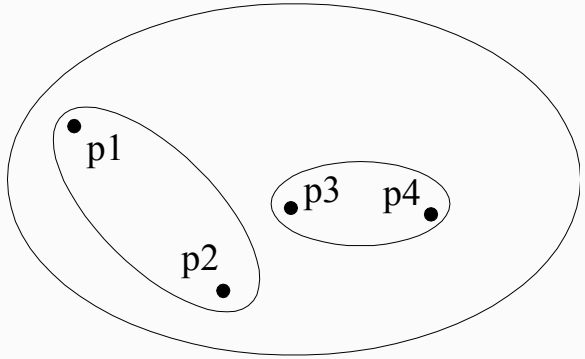
Hierarchical Clustering



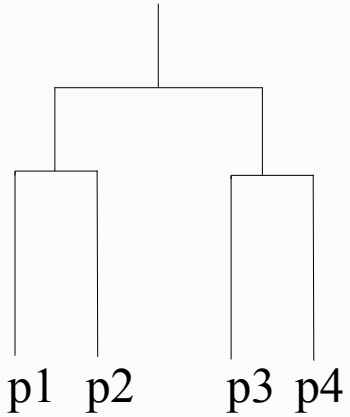
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



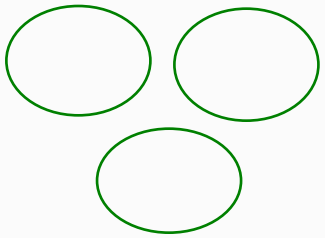
Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

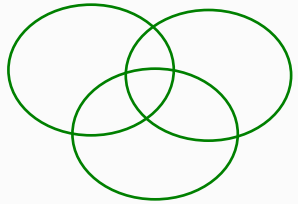
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Clustering Design Space

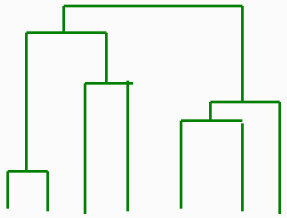
Non overlapping



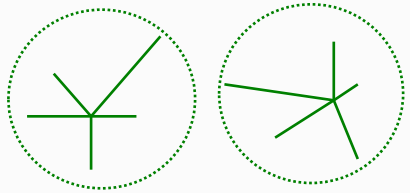
Overlapping



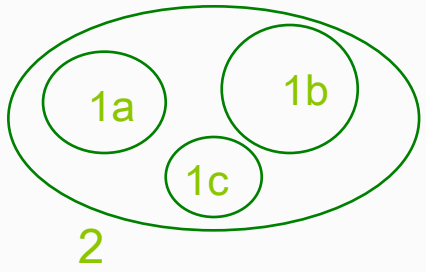
Hierarchical



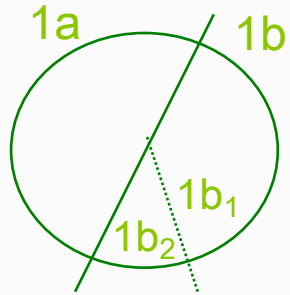
Non-hierarchical

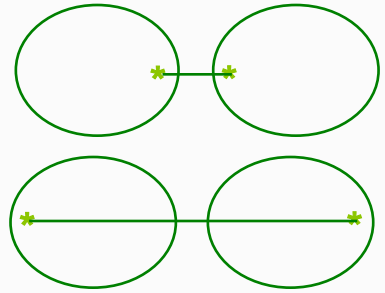


Agglomerative



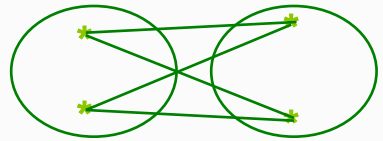
Divisive



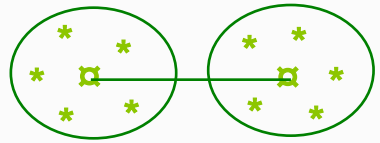


Single Linkage:
Minimum distance

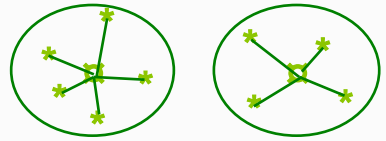
Complete Linkage:
Maximum distance



Average Linkage:
Average distance



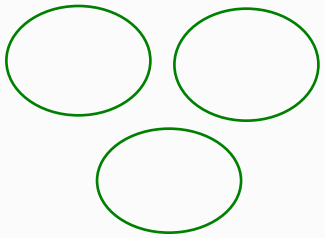
Centroid method:
Distance between centres



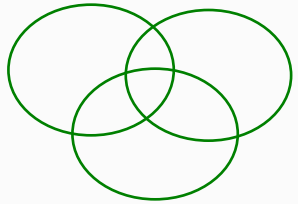
Wards method:
Minimization of within-cluster variance

Clustering Design Space

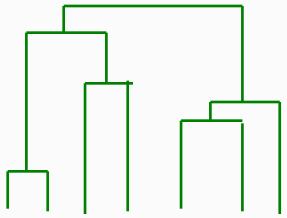
Non overlapping



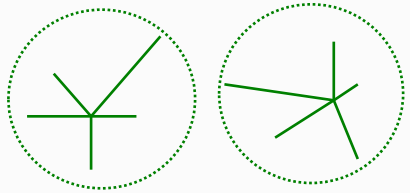
Overlapping



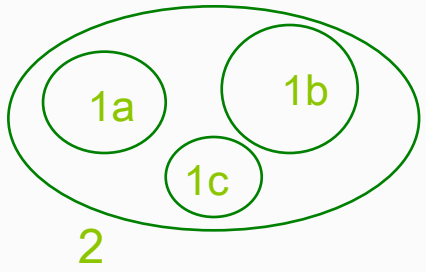
Hierarchical



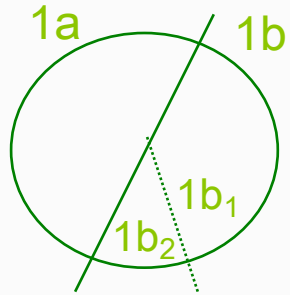
Non-hierarchical

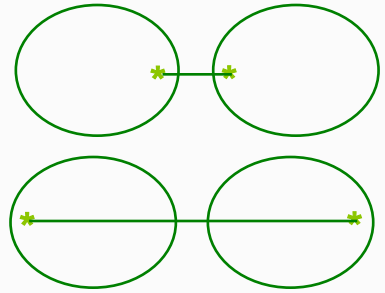


Agglomerative



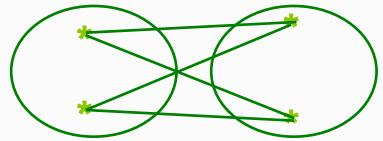
Divisive



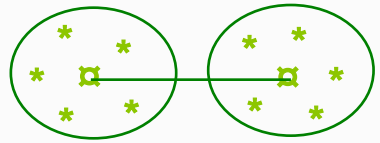


Single Linkage:
Minimum distance

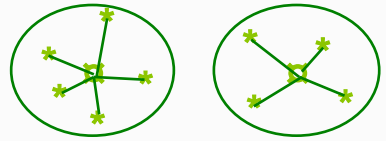
Complete Linkage:
Maximum distance



Average Linkage:
Average distance

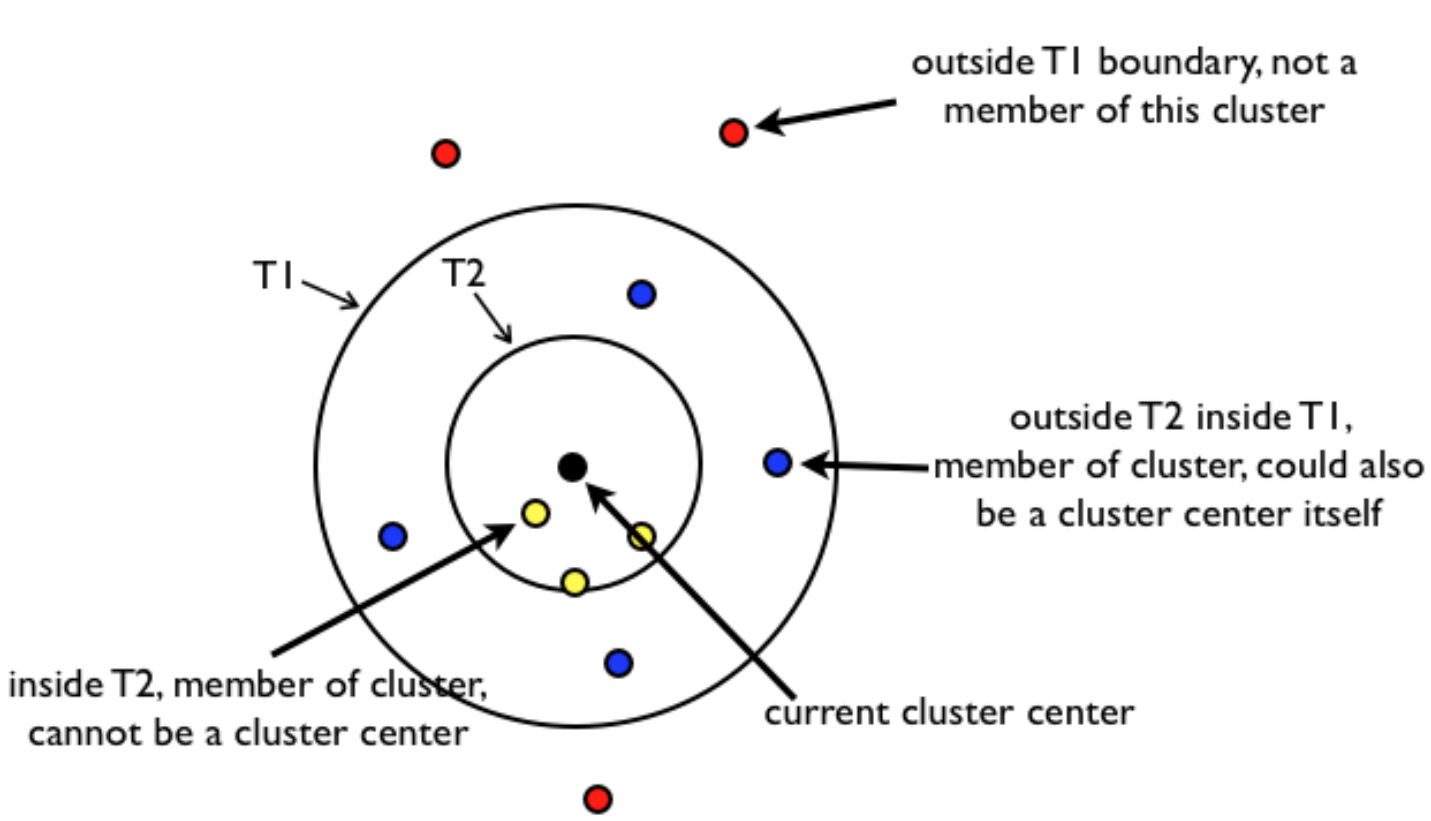


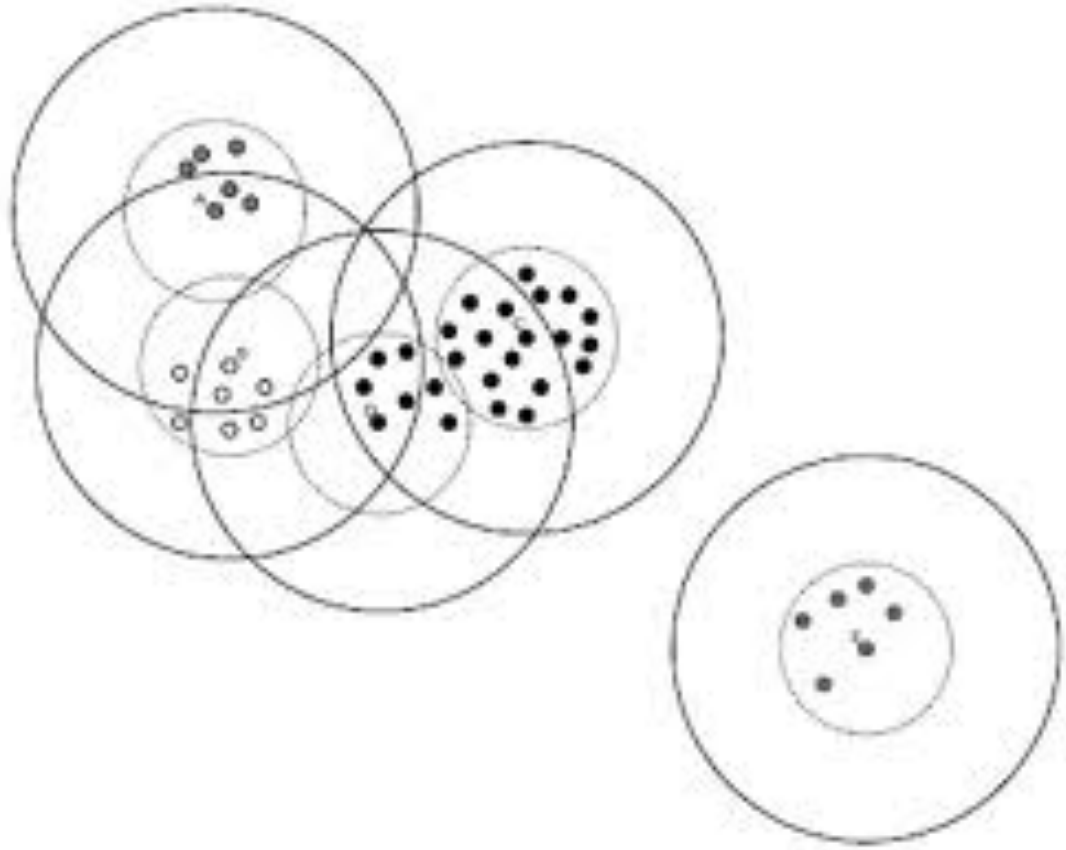
Centroid method:
Distance between centres



Wards method:
Minimization of within-cluster variance

Canopy Clustering



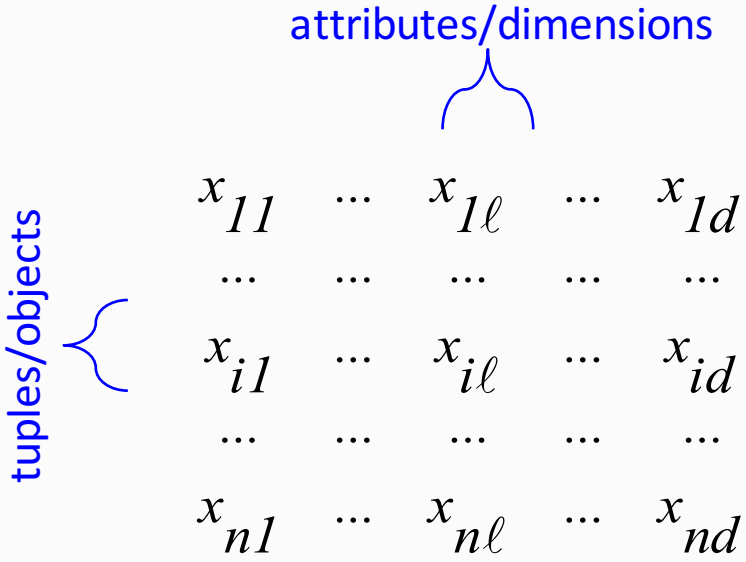


Distance Functions

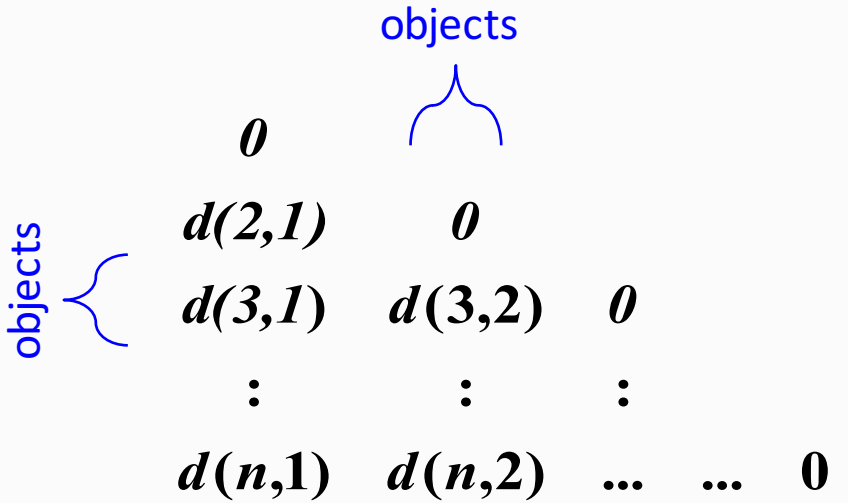
- The distance **d(x, y)** between two objects **x** and **y** is a metric if
  - **d(i, j) ≥ 0** (non-negativity)
  - **d(i, i) = 0** (isolation)
  - **d(i, j) = d(j, i)** (symmetry)
  - **d(i, j) ≤ d(i, h)+d(h, j)** (triangular inequality)
- The definitions of distance functions are usually different for real, boolean, categorical, and ordinal variables.
- Weights may be associated with different variables based on applications and data semantics.

Data Structures

- *data* matrix



- *Distance* matrix



Distance functions for real-valued vectors

- If **p = 2, L_2** is the **Euclidean distance**:

$$d(x,y)=\sqrt{(|x_1 \quad y_1|^2+|x_2 \quad y_2|^2+\ldots+|x_d \quad y_d|^2)}$$

- Also one can use **weighted distance**:

$$d(x,y)=\sqrt{(w_1|x_1 \quad x_1|^2+w_2|x_2 \quad x_2|^2+\ldots+w_d|x_d \quad y_d|^2)}$$

$$d(x,y)=w_1|x_1 \quad y_1|+w_2|x_2 \quad y_2|+\ldots+w_d|x_d \quad y_d|$$

Distance functions for binary vectors

- **Jaccard similarity** between binary vectors **X** and **Y**

$$JSim(X,Y) = \frac{X \quad Y}{X \quad Y}$$

- **Jaccard distance** between binary vectors **X** and **Y**

  Jdist(X,Y) = 1- JSim(X,Y)

- Example:
  - JSim = 1/6
  - Jdist = 5/6

	Q1	Q2	Q3	Q4	Q5	Q6
X	1	0	0	1	1	1
Y	0	1	1	0	1	0

Distance functions for real-valued vectors

- **L_p** norms or *Minkowski distance*:

$$L_p(x,y) = |x_1 \quad y_1|^p + |x_2 \quad y_2|^p + \ldots + |x_d \quad x_d|^p \;^{1/p} = \sum_{i=1}^{d} (x_i \quad y_i)^{1/p}$$

where ***p*** is a positive integer

- If **p = 1, L_1** is the *Manhattan (or city block)* distance:

$$L_1(x,y) = |x_1 \quad y_1| + |x_2 \quad y_2| + \ldots + |x_d \quad y_d| = \sum_{i=1}^{d} |x_i \quad y_i|$$

K-means Clustering

K-means Clustering

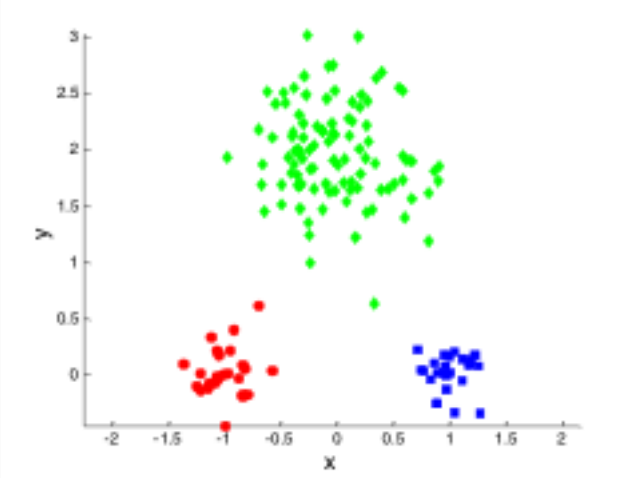
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

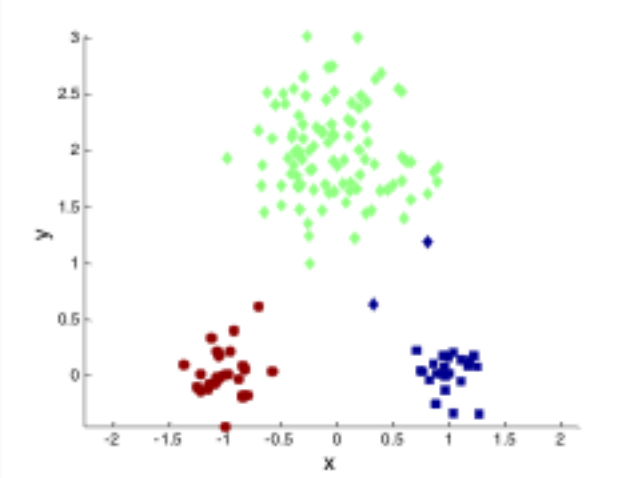
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

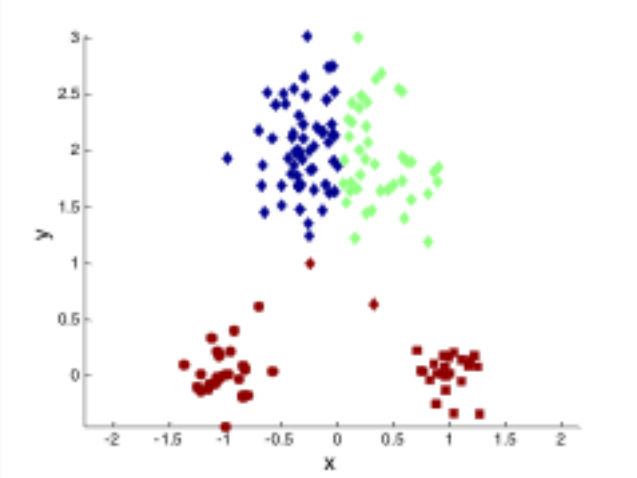
Two different K-means Clusterings



Original Points

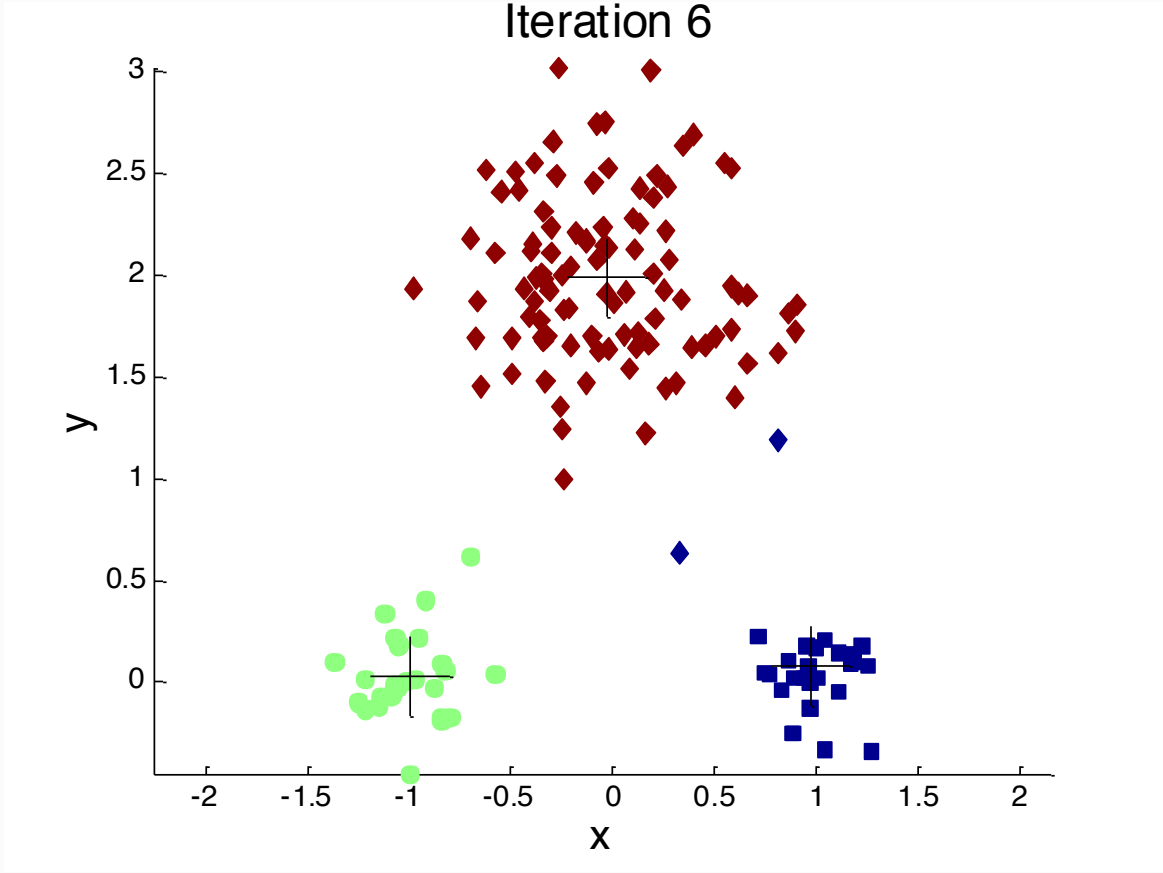


Optimal Clustering

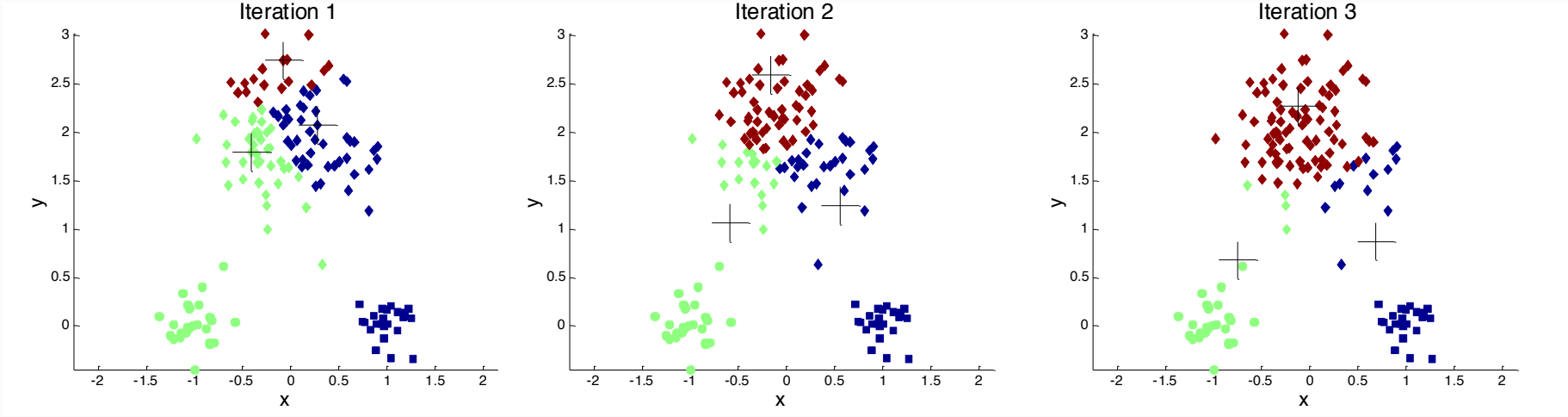


Sub-optimal Clustering

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



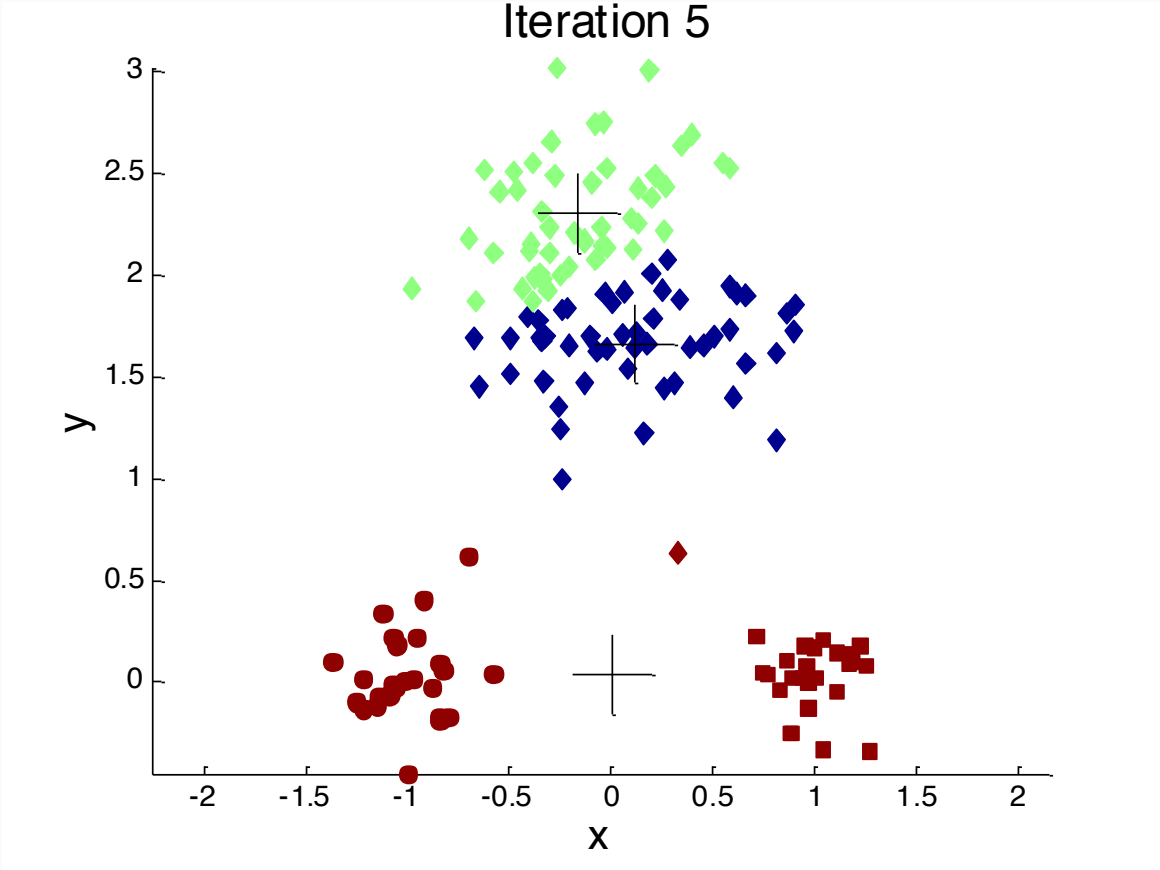
Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

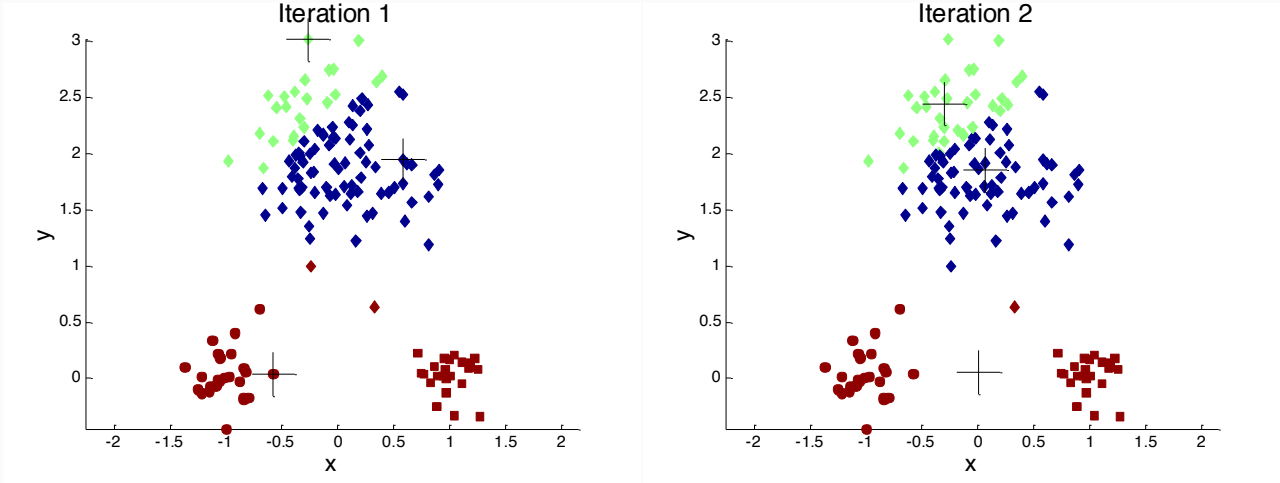
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

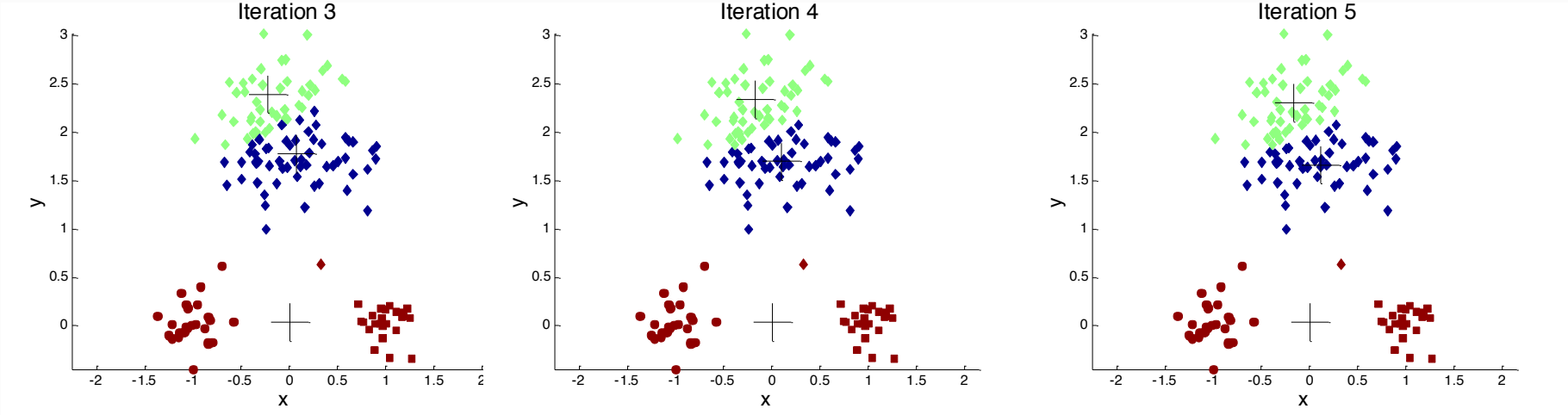
  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …





Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing

K-Means Clustering Discussion

Non-numeric data

Feature values are not always numbers

- Example
  - Boolean Values: Yes or no, presence or absence of an attribute
  - Categories: Colors, educational attainment, gender

How do these values factor into the computation of distance?

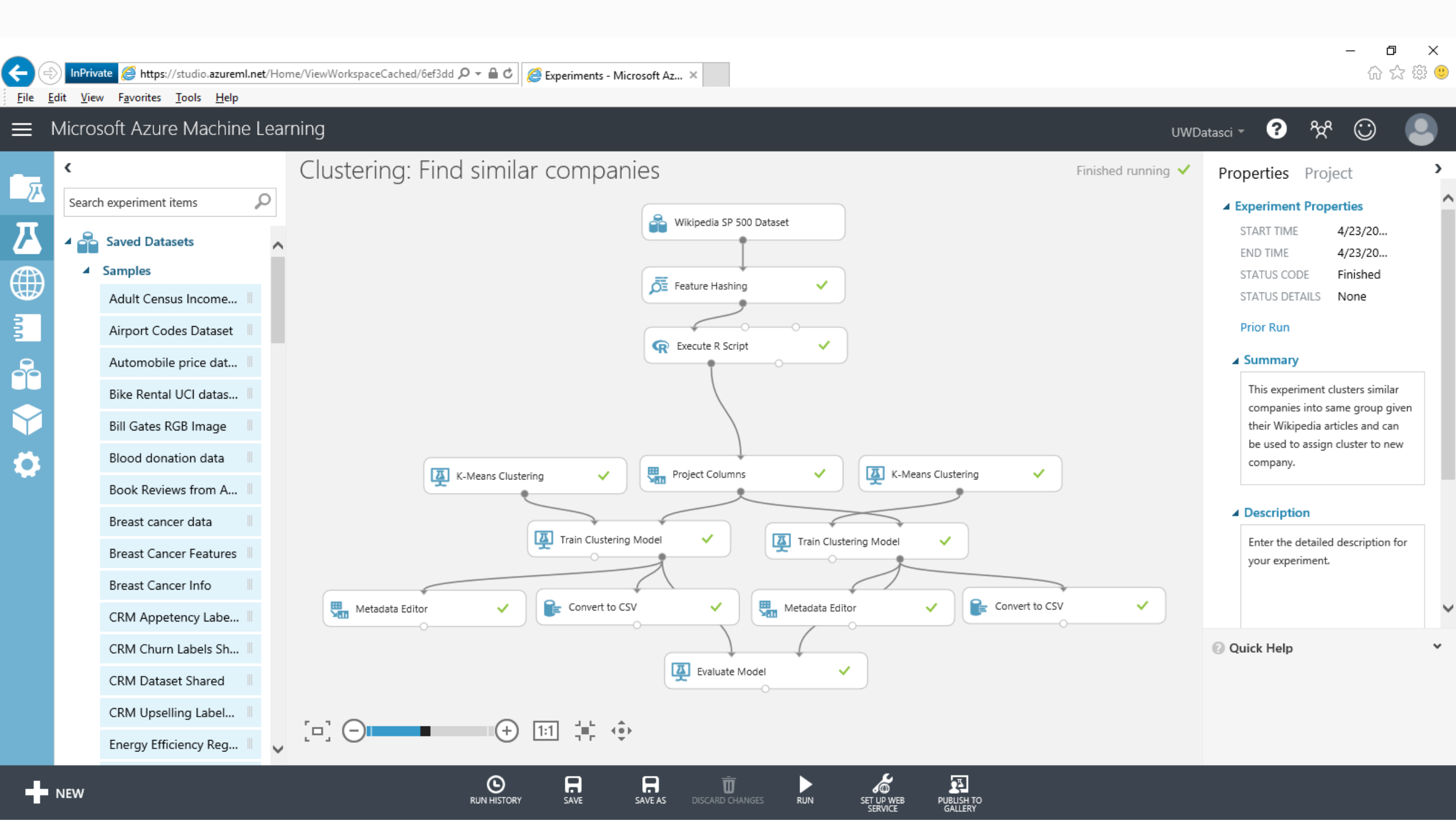
K-Means Clustering Discussion

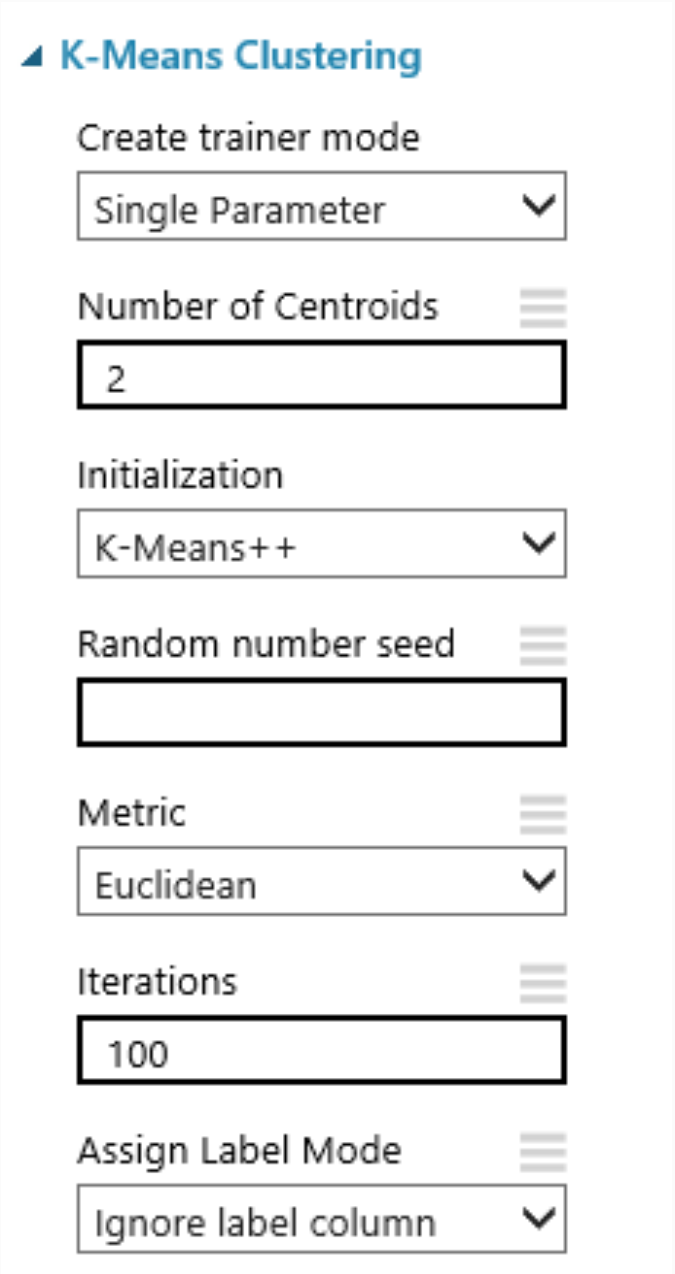
Dealing with non-numeric data

- Boolean values => convert to 0 or 1
  - Applies to yes-no/presence-absence attributes
- Non-binary characterizations
  - Use natural progression when applicable; e.g., educational attainment: GS, HS, College, MS, PHD => 1,2,3,4,5
  - Assign arbitrary numbers but be careful about distances; e.g., color: red, yellow, blue => 1,2,3
- How about unavailable data?
  (0 value not always the answer)

K-Means Clustering Discussion
Preprocessing your dataset

- Dataset may need to be preprocessed to ensure more reliable data mining results
- Conversion of non-numeric data to numeric data
- Calibration of numeric data to reduce effects of disparate ranges
  - Particularly when using the Euclidean distance metric


InPrivate
https://studio.azureml.net/Home/ViewWorkspaceCached/6ef3dd
Experiments - Microsoft Az...
File Edit View Favorites Tools Help
Microsoft Azure Machine Learning
UWDatasci
Clustering: Find similar companies
Finished running
Search experiment items
Saved Datasets
Samples
Adult Census Income...
Airport Codes Dataset
Automobile price dat...
Bike Rental UCI datas...
Bill Gates RGB Image
Blood donation data
Book Reviews from A...
Breast cancer data
Breast Cancer Features
Breast Cancer Info
CRM Appetency Labe...
CRM Churn Labels Sh...
CRM Dataset Shared
CRM Upselling Label...
Energy Efficiency Reg...
Wikipedia SP 500 Dataset
Feature Hashing
Execute R Script
K-Means Clustering
Project Columns
K-Means Clustering
Train Clustering Model
Train Clustering Model
Metadata Editor
Convert to CSV
Metadata Editor
Convert to CSV
Evaluate Model
1:1
Properties
Project
Experiment Properties
START TIME 4/23/20...
END TIME 4/23/20...
STATUS CODE Finished
STATUS DETAILS None
Prior Run
Summary
This experiment clusters similar companies into same group given their Wikipedia articles and can be used to assign cluster to new company.
Description
Enter the detailed description for your experiment.
Quick Help
NEW
RUN HISTORY
SAVE
SAVE AS
DISCARD CHANGES
RUN
SET UP WEB SERVICE
PUBLISH TO GALLERY



Initialization and ***Initialization for sweep***

Choose the algorithm that is used to define the initial cluster configuration. If you leave this parameter blank, the module will generate points using the ***K-Means++*** method.

- ***First N***. Some initial number of data points are chosen from the data set and used as the initial means.

  Also called the Forgy method.

- **Random**. The algorithm randomly places a data point in a cluster and then computes the initial mean to be the centroid of the cluster's randomly assigned points.

  Also called the *random partition* method.

- ***K-Means++***. K-means++ improves upon K-means by using a better method for choosing the initial cluster centers

  The K-means ++ algorithm was proposed in 2007 by David Arthur and Sergei Vassilvitskii to avoid poor clustering by the standard k-means algorithm.

- ***K-Means++Fast***. A variant of the K-means++ algorithm optimized for faster clustering.

- ***Evenly***. Centroids are located equidistant from each other in the D-Dimensional space of N data points.

- ***Use label column***. The values in the label column are used to guide the selection of centroids.

Clustering: Find similar companies

Finished running

Properties Project

Clustering: Find similar companies › Wikipedia SP 500 Dataset › dataset

rows 466

columns 3

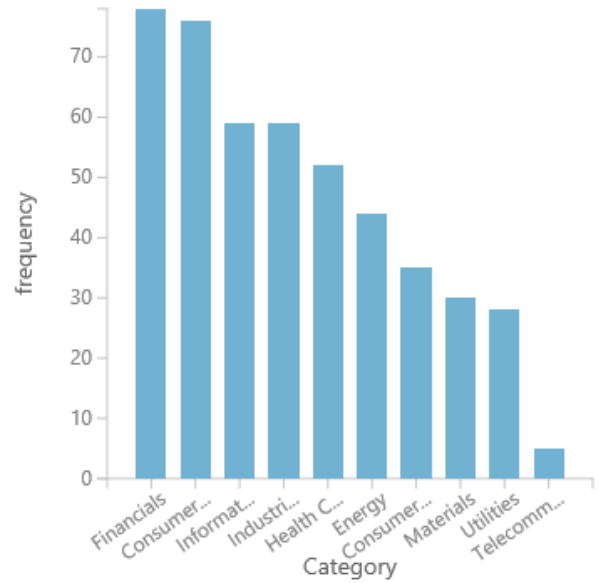
Title	Category	Text
Apple Inc.	Information Technology	nasdaq 100 component s p 500 component foundation founder location city apple campus 1 infinite loop street infinite loop cupertino california cupertino california location country united states u s locations 406 retail stores may 2013 area served worldwide key people ref tim cook ceo steve jobs fou...
		br nasdaq 100 nasdaq 100 component br s p 500 s p 500 component industry computer software foundation br founder charles geschke br john

view as

Category

Histogram

compare to None



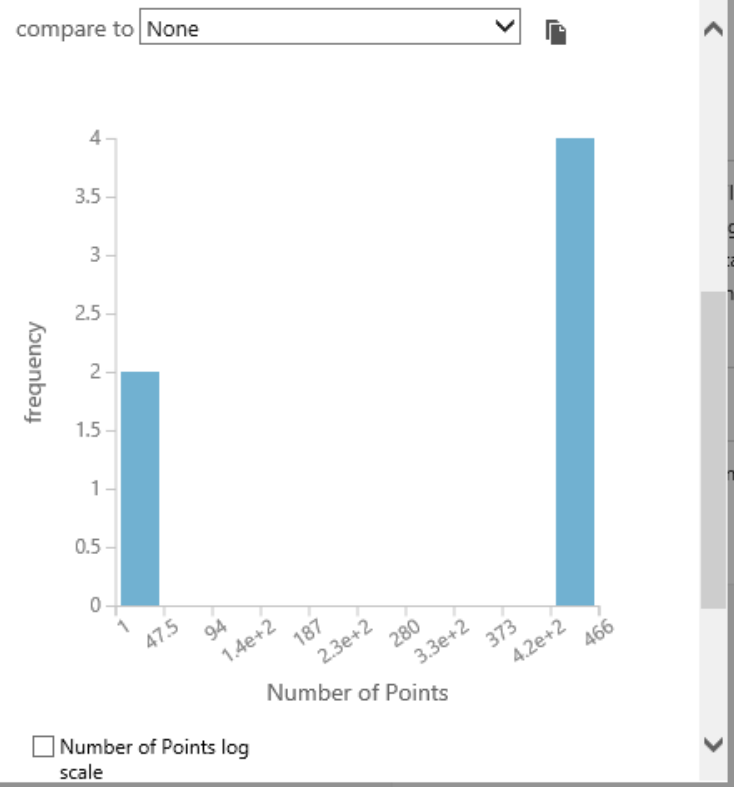
"Closeness of the cluster"
How close the points in a cluster is to the centroid of the point's cluster

"Closeness of the cluster"
How close the points in a cluster is to the centroid of other clusters

Sum the distances between each point and the centroid of that point's cluster.

rows 6 columns 5

Result Description	Average Distance to Cluster Center	Average Distance to Other Center	Number of Points	Maximal Distance To Cluster Center
Combined Evaluation	29.228741	558.095045	466	421.440316
Evaluation For Cluster No.0	29.291598	558.101318	465	421.440316
Evaluation For Cluster No.1	0	555.178265	1	0
Combined Evaluation	29.228741	558.095045	466	421.440316
Evaluation For Cluster No.0	29.291598	558.101318	465	421.440316
Evaluation For Cluster No.1	0	555.178265	1	0



Data Science

Deriving Knowledge from Data at Scale

That's all for tonight....