Constrained K-means Clustering with Background Knowledge

Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schroedl

Presenter : Mingjun Wang

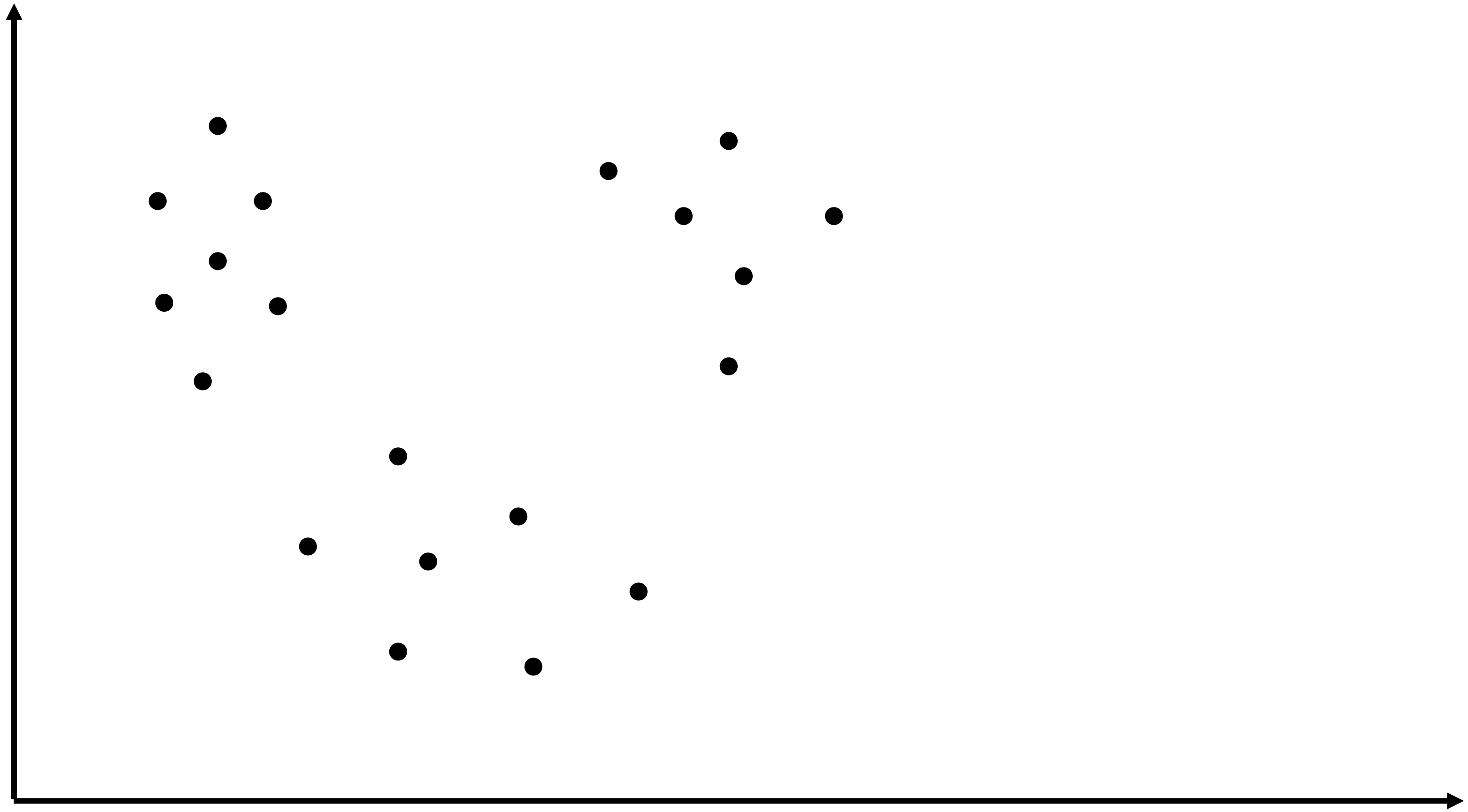
Content

- Introduction
- K-means Clustering
- Constrained K-means Clustering
- Evaluation Method
- Experiments
  - Using Artificial Constraints
  - Using GPS Lane Finding
- Conclusion

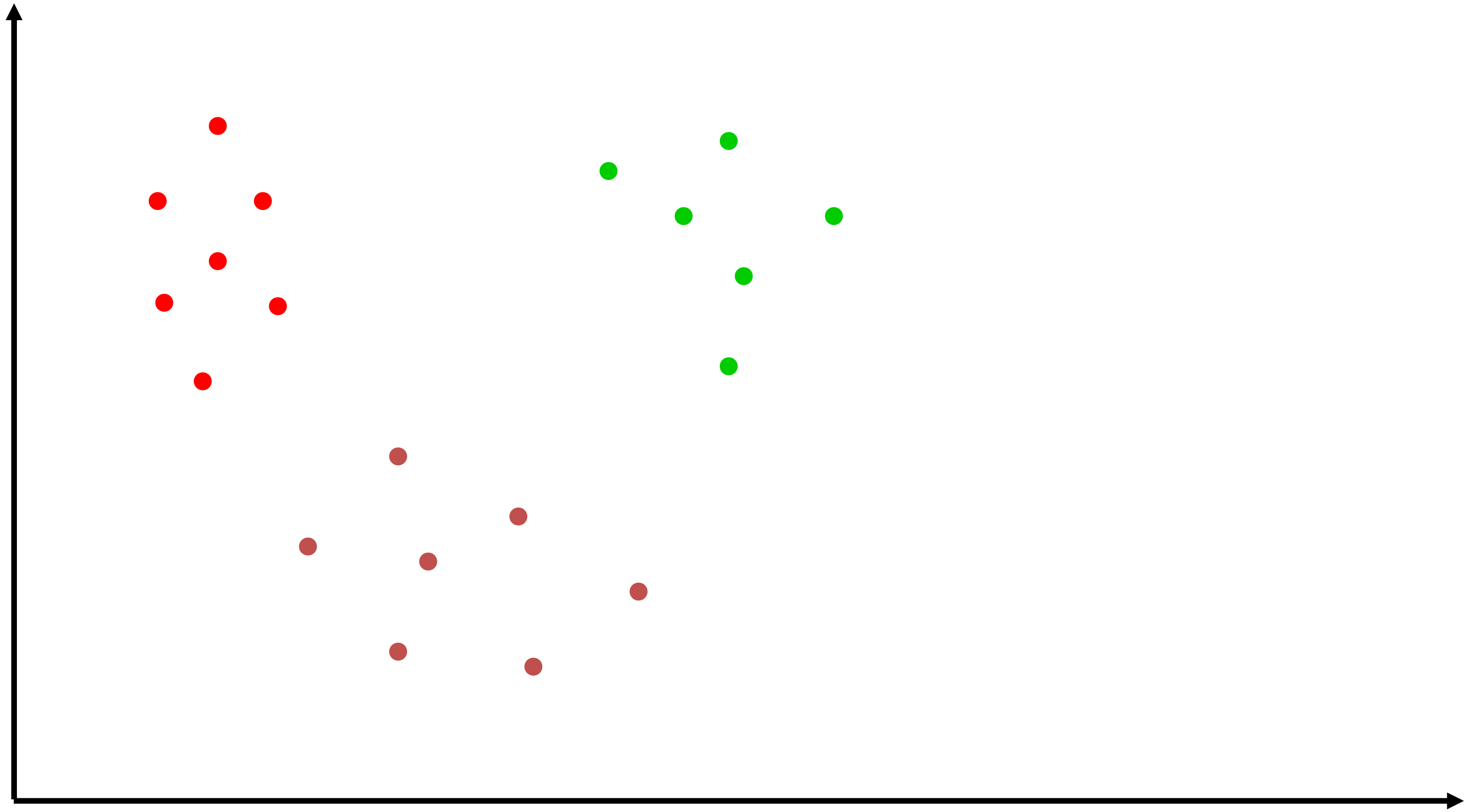
Introduction

- Clustering Algorithms – Unsupervised learning (generally speaking)
  - Do not take advantage of any background knowledge even when this information exists

Unsupervised Clustering Example



Unsupervised Clustering Example



Introduction

- Contribution
  - Developed a k-means variant that can incorporate background knowledge and demonstrate its performance in multiple data sets
  - Applied this method to a significant real world problem
- Highlights
  - Modify k-means algorithm to be not limited to a single clustering methods
  - Incorporate background knowledge
  - Semi-supervised clustering

Clustering: Problem Definition

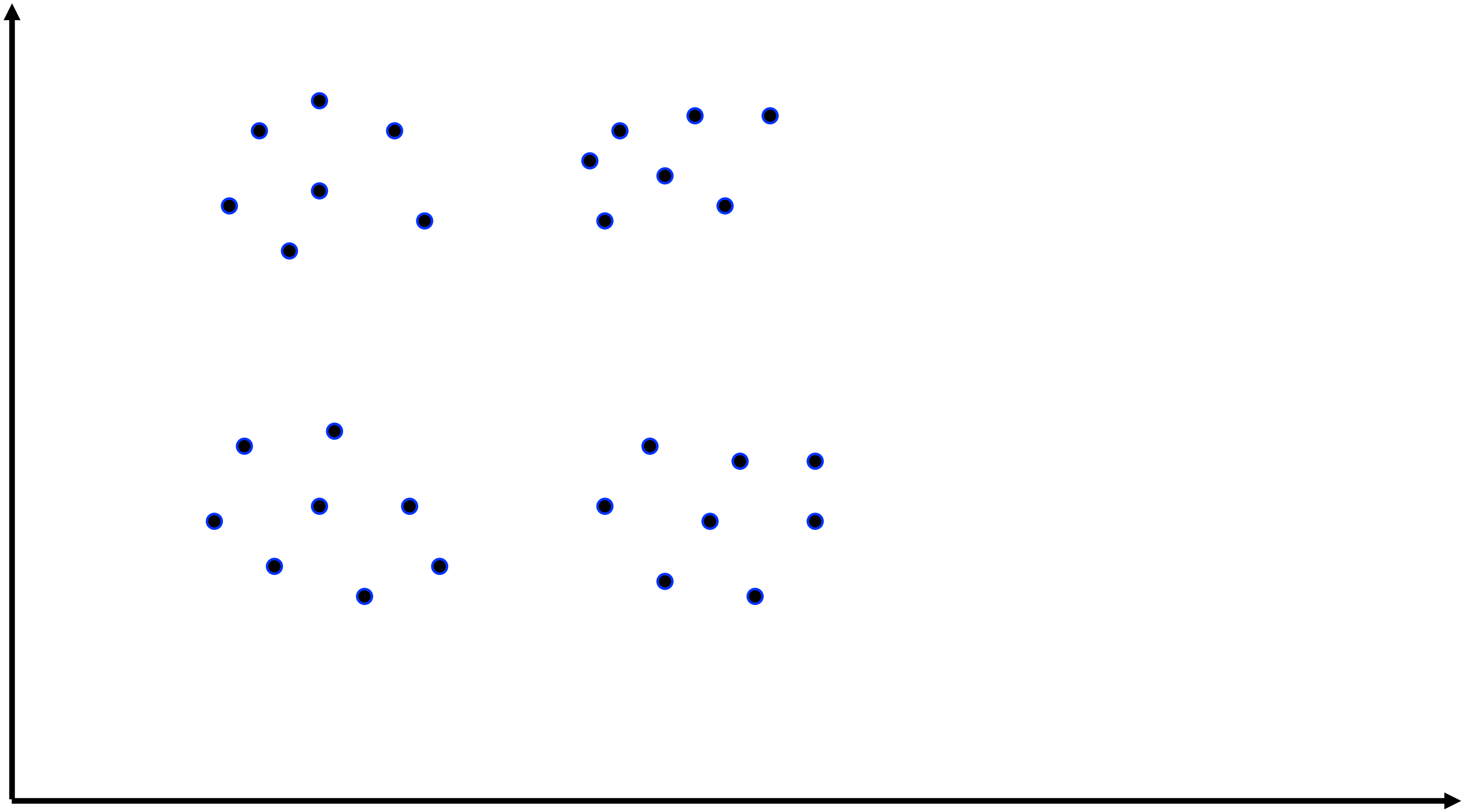
- Input:
  - A set of unlabeled objects
- Output
  - A partitioning of the objects into k clusters
- Object
  - Maximum intra-cluster similarity
  - Minimum inter-cluster similarity

K-means

- Initiate K cluster centers randomly

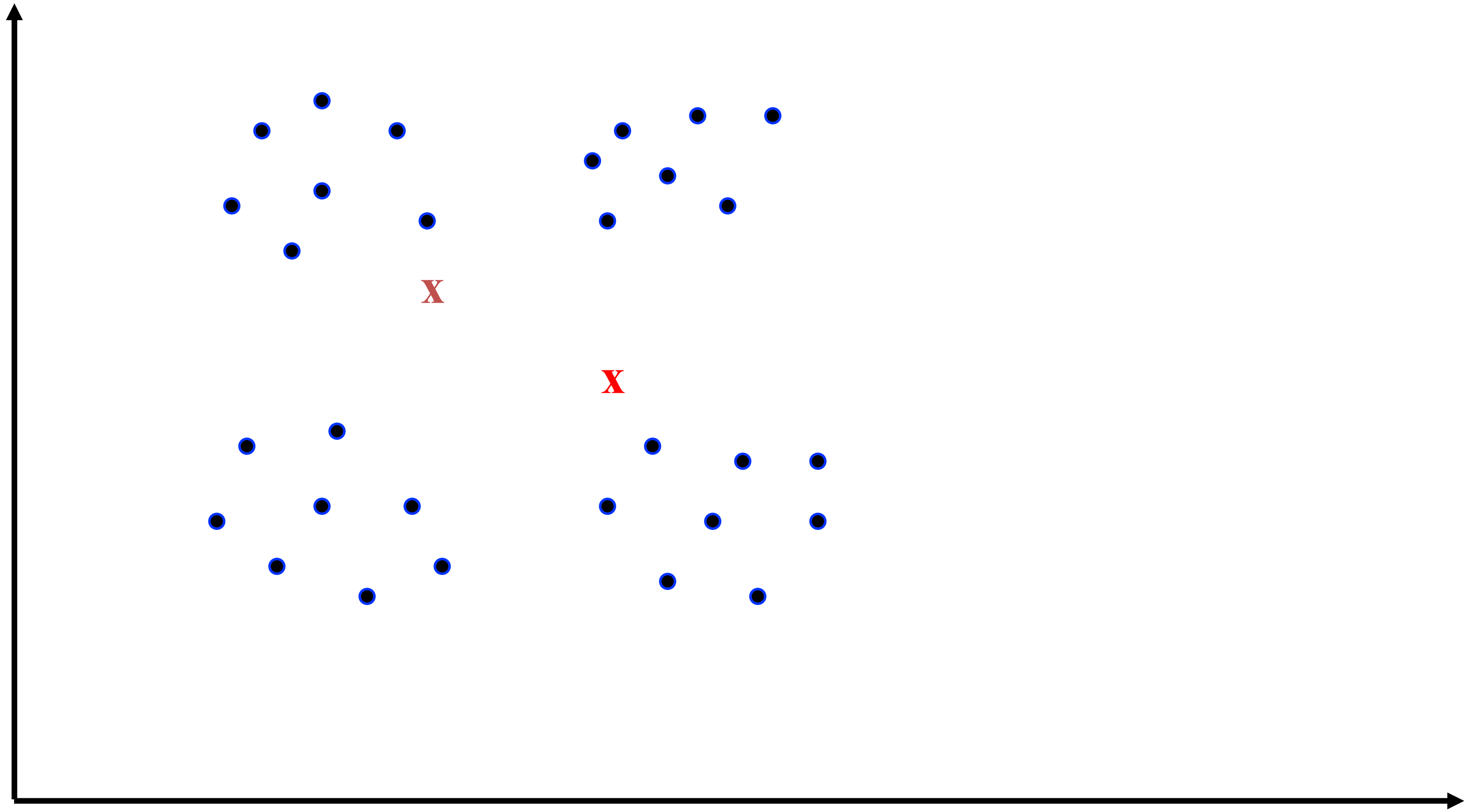
- Repeat following steps until convergence:
  - Cluster Assignment Step: Each instance is assigned to its closest center
  - Center Re-estimation Step: Each cluster is updated to be the mean of its constituent instances

K Means Example



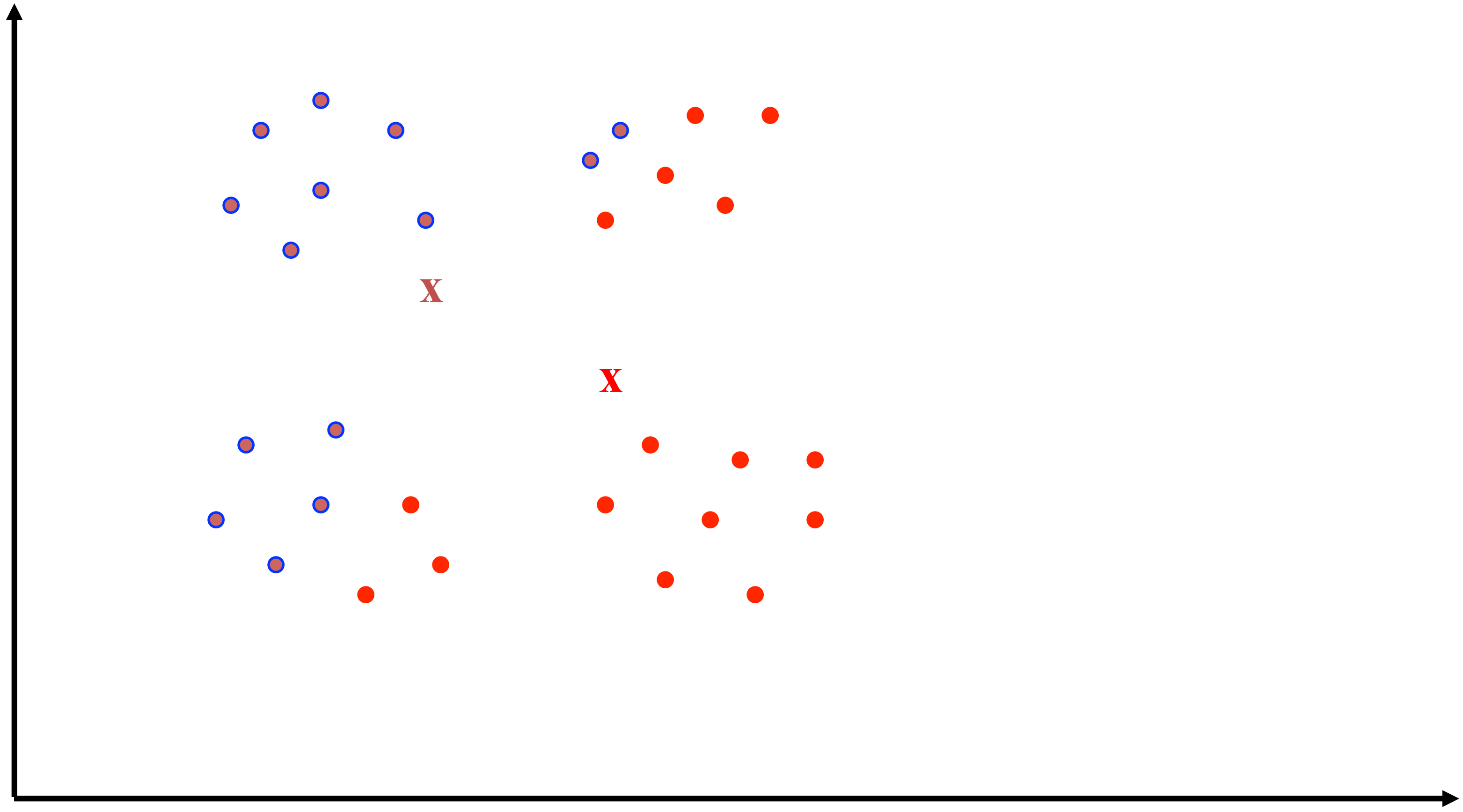
K Means Example

Randomly Initialize Means



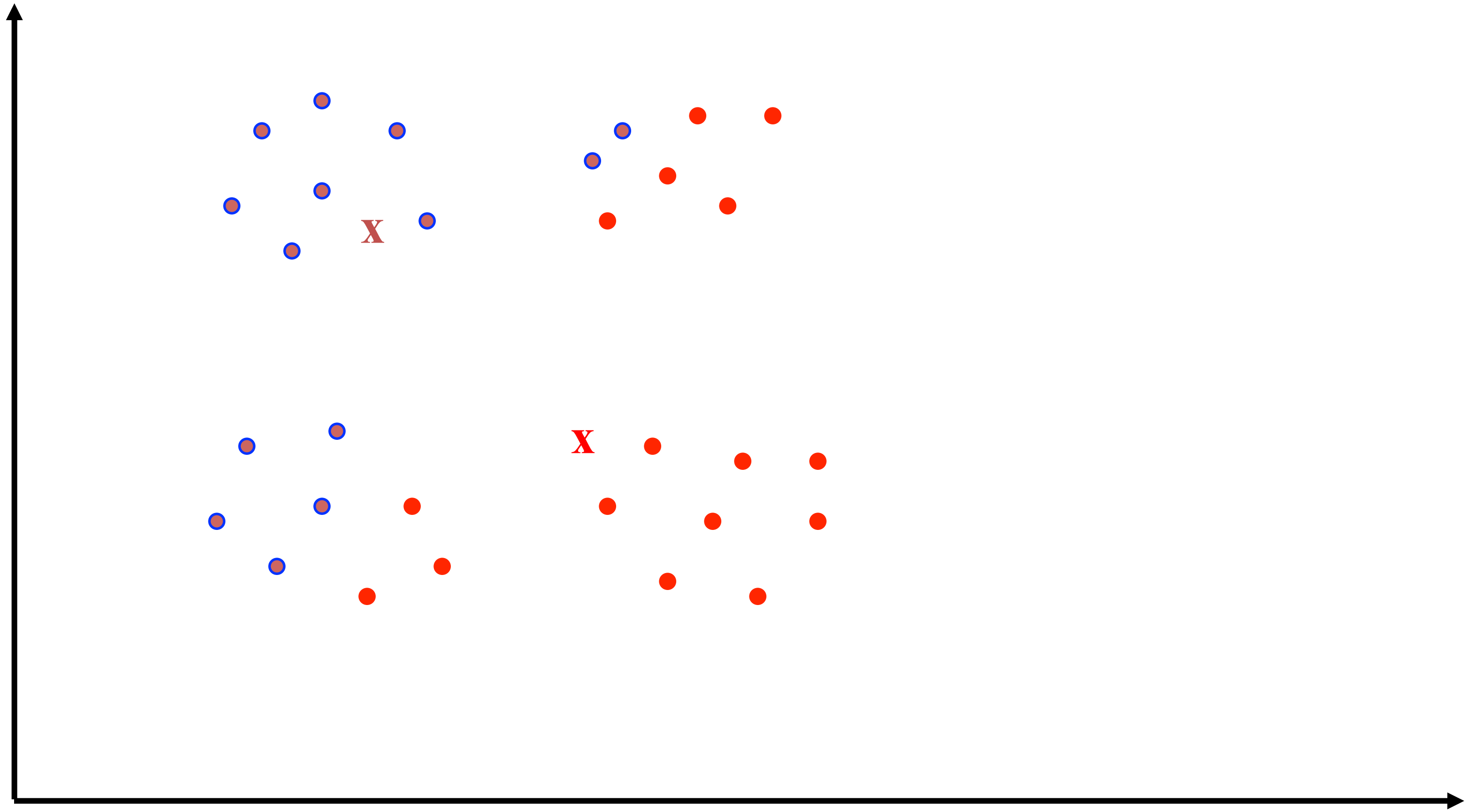
K Means Example

Assign Points to Clusters



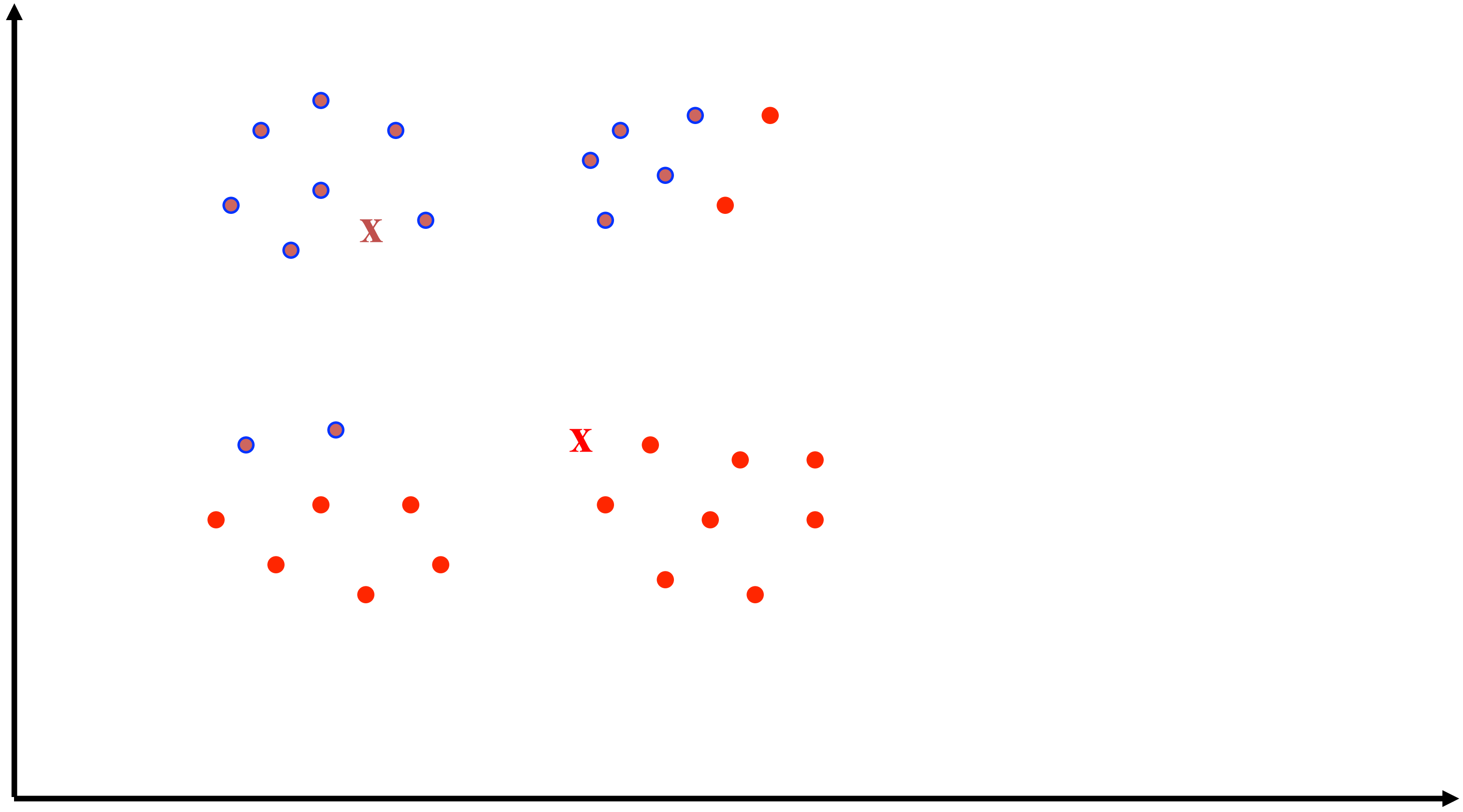
K Means Example

Re-estimate Means



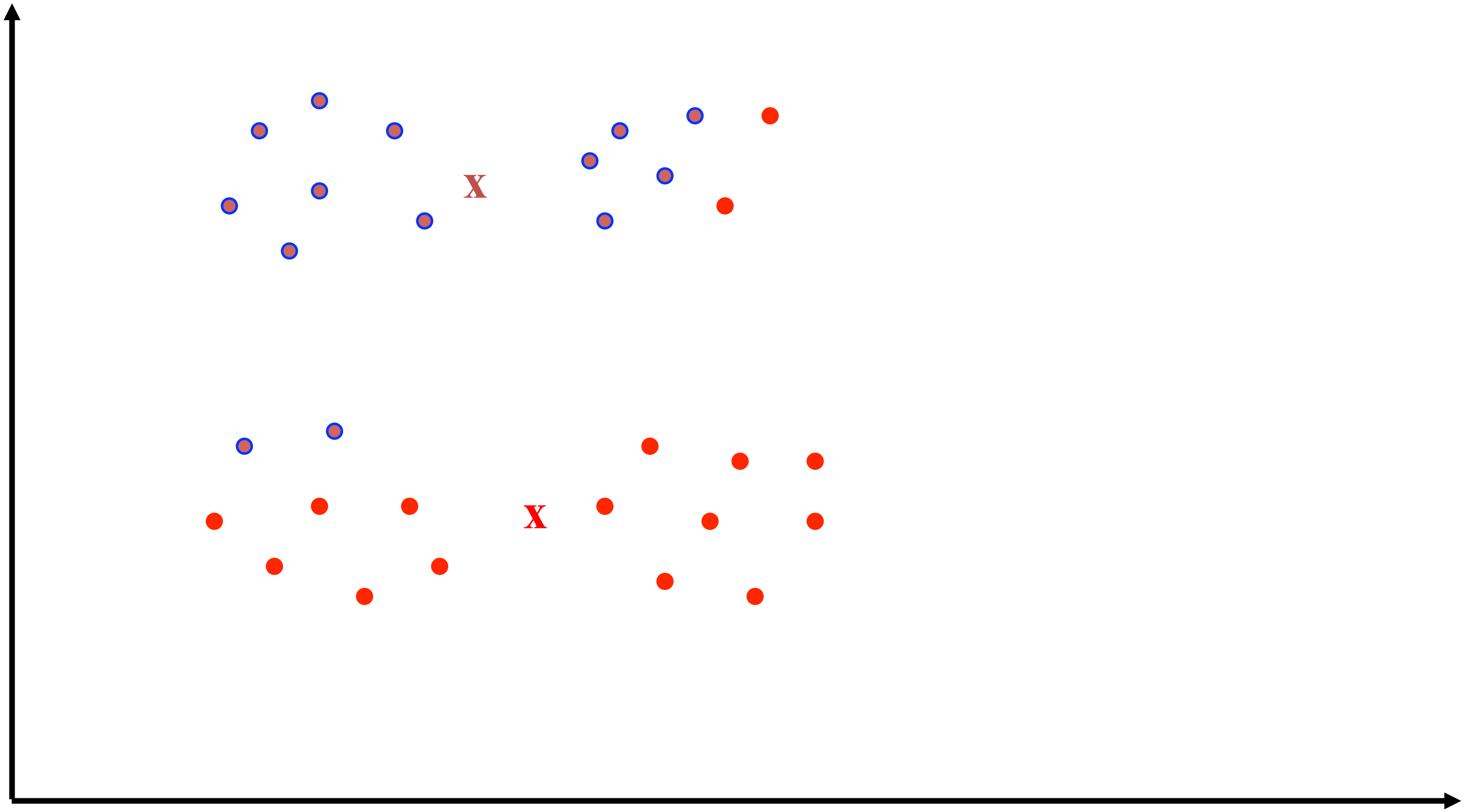
K Means Example

Re-assign Points to Clusters



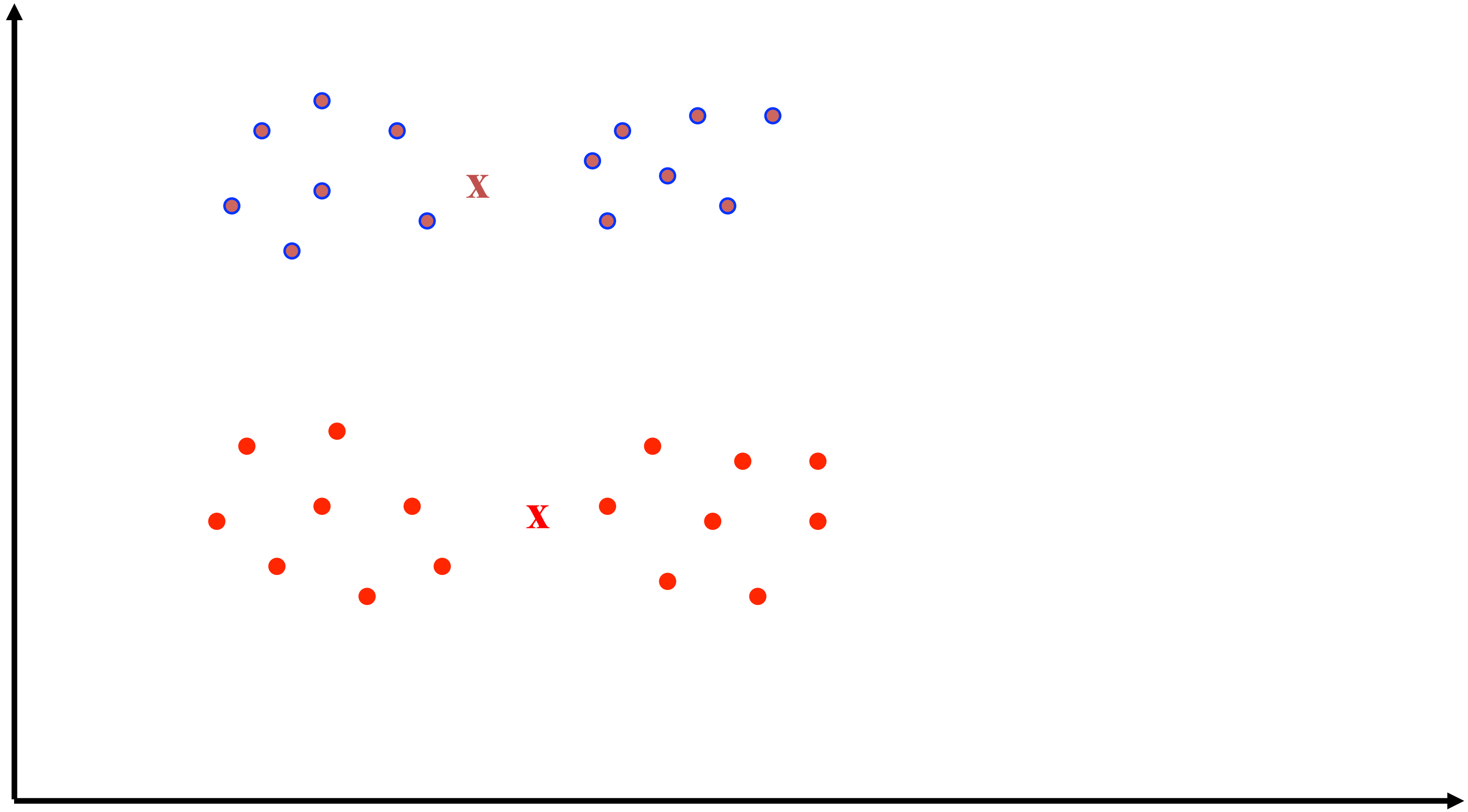
K Means Example

Re-estimate Means



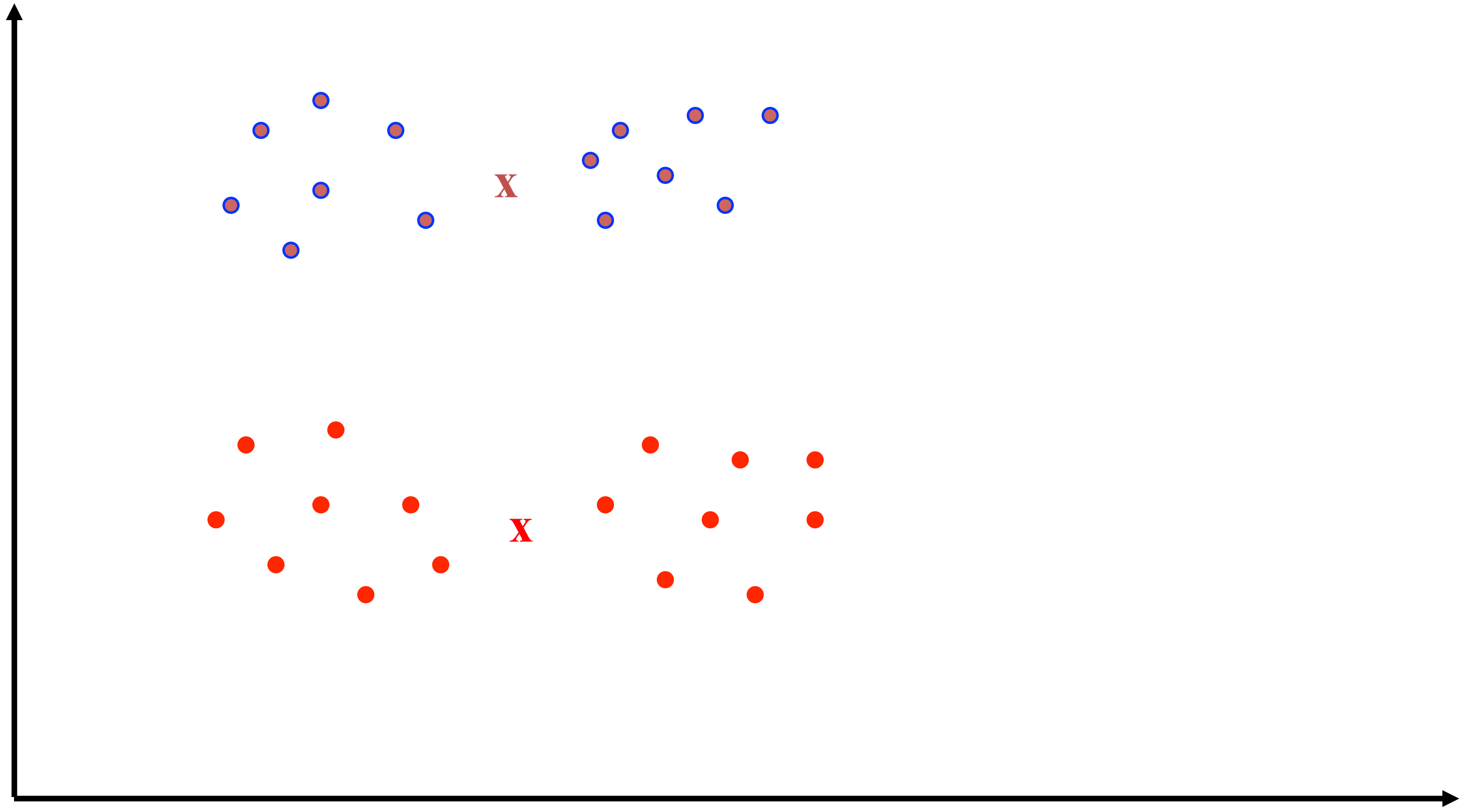
K Means Example

Re-assign Points to Clusters



K Means Example

Re-estimate Means and Converge



Semi-supervised clustering: Problem Definition

- Input:
  - A set of unlabeled objects
  - **A small set of domain knowledge – Constraints**
- Output
  - A partitioning of the objects into k clusters
- Object
  - **High consistency between the partitioning and the domain knowledge**
  - Maximum intra-cluster similarity
  - Minimum inter-cluster similarity

Constrained K-means Clustering with Background Knowledge

- K-Means with must-link and cannot-link **constraints** on data points.
  - must-link : must be in same cluster
  - cannot-link :   cannot be in same cluster

- Constraints: background knowledge about the domain or data set; Partially labeled data

Constrained K-means Algorithm

Table 1. Constrained K-means Algorithm

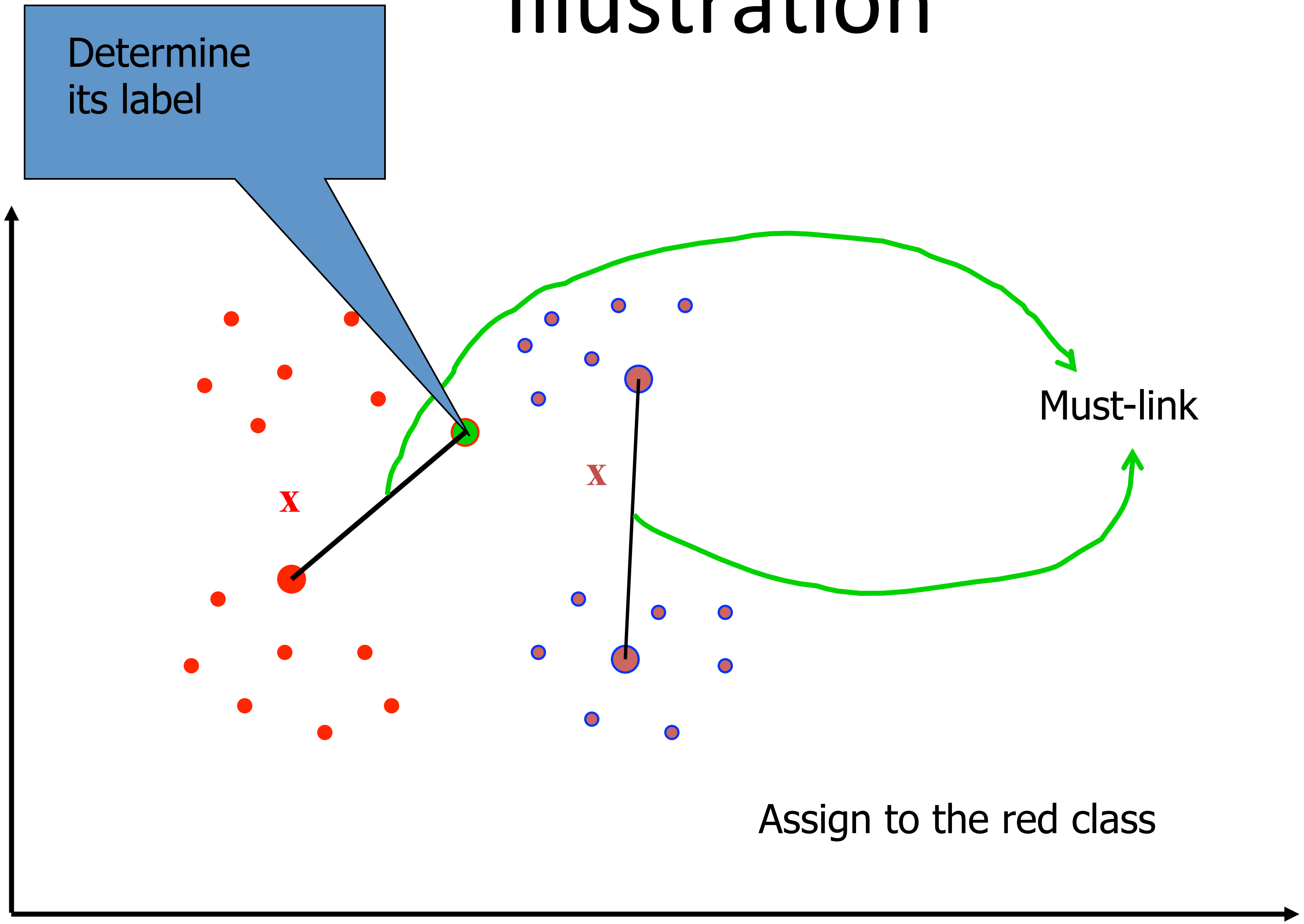
COP-KMEANS(data set D, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. Let $C_1 \ldots C_k$ be the initial cluster centers.
2. For each point d_i in D, assign it to the closest cluster C_j **such that** VIOLATE-CONSTRAINTS(d_i, C_j, $Con_=$, $Con_{\neq}$) **is false. If no such cluster exists, fail (return** $\{\}$).
3. For each cluster C_i, update its center by averaging all of the points d_j that have been assigned to it.
4. Iterate between (2) and (3) until convergence.
5. Return $\{C_1 \ldots C_k\}$.

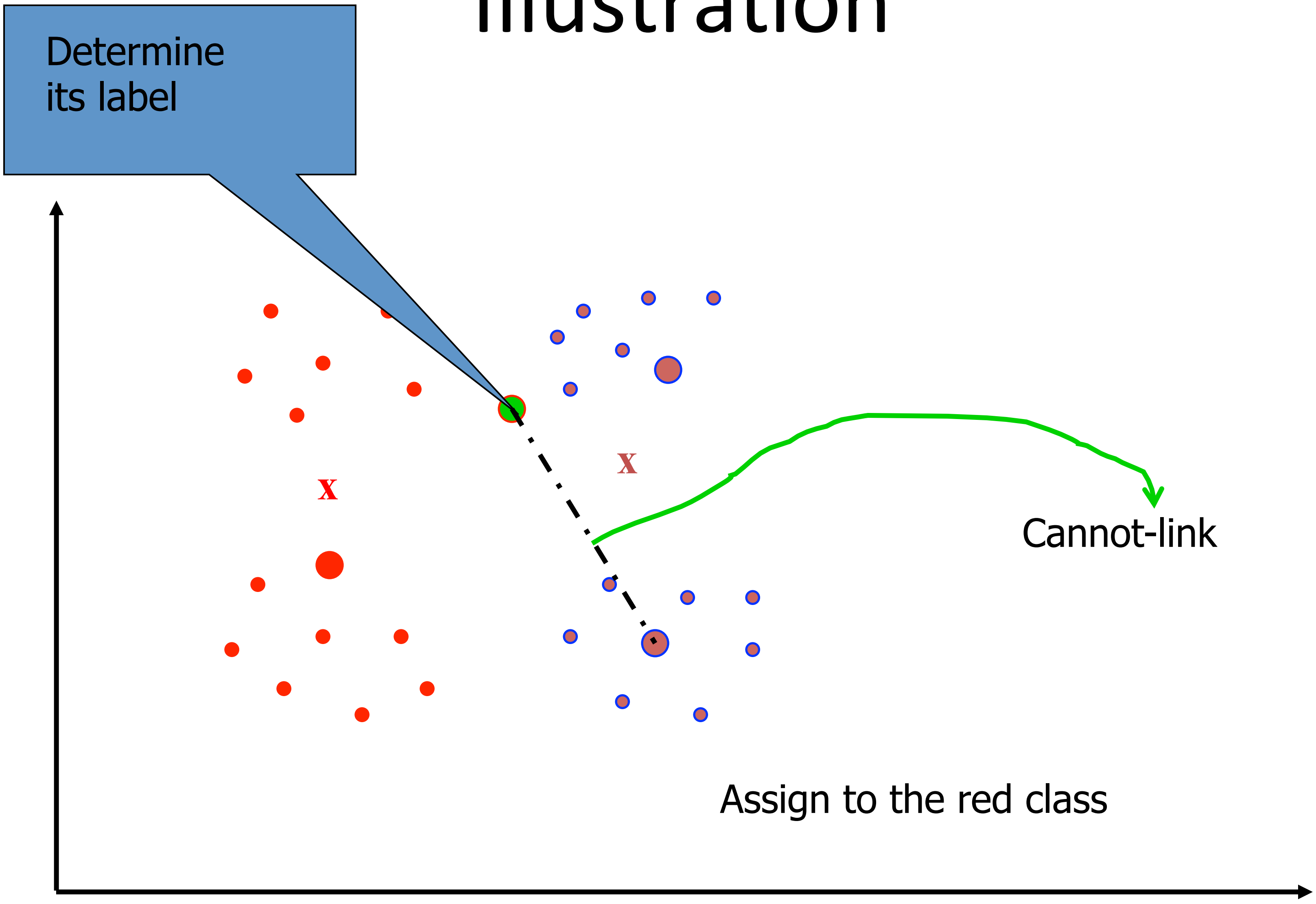
VIOLATE-CONSTRAINTS(data point d, cluster C, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. For each $(d, d_=) \in Con_=$: If $d_= \notin C$, return true.
2. For each $(d, d_{\neq}) \in Con_{\neq}$: If $d_{\neq} \in C$, return true.
3. Otherwise, return false.

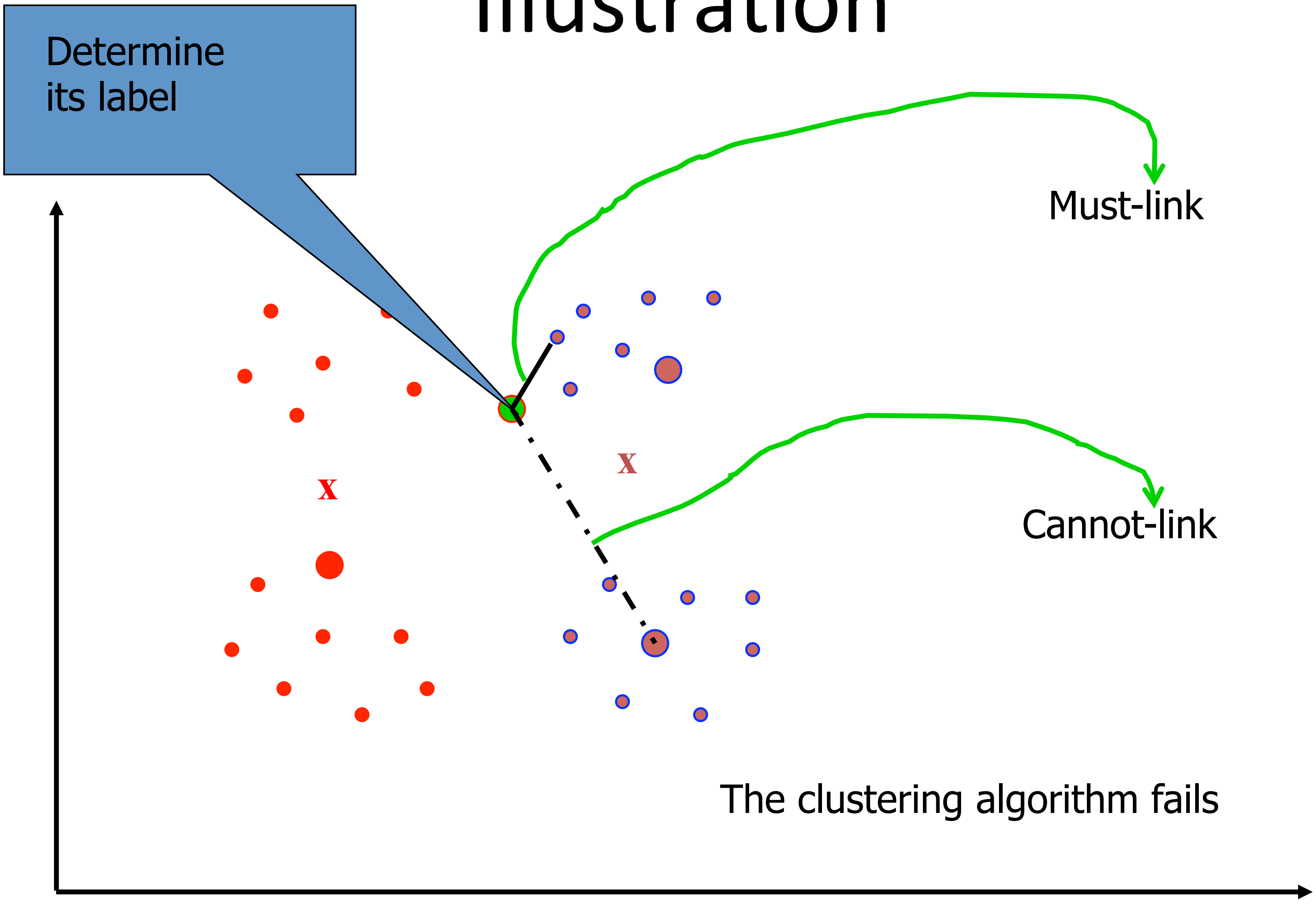
Illustration



Illustration



Illustration



Evaluation Method

- Rand Index – Measure agreement between results with correct labels

$$Rand(P_1, P_2) = \frac{a+b}{n*(n-1)/2}.$$

Given a set of n elements $S = \{o_1, \ldots, o_n\}$ and two partitions of S to compare, $X = \{X_1, \ldots, X_r\}$, a partition of *S* into *r* subsets, and $Y = \{Y_1, \ldots, Y_s\}$, a partition of *S* into *s* subsets, define the following:

- a, the number of pairs of elements in S that are in the same set in X and in the same set in Y
- b, the number of pairs of elements in S that are in different sets in X and in different sets in Y

- Overall accuracy in entire data set and a held out test set

Experiments – Using Artificial Constraints

- Value of k is known as a input for the algorithm

- Constrained:
  - Randomly pick two instances from the data set and check their labels to generate a must-link constraints or cannot – link constraint

- Data:
  - Soybean, mushroom, part of speech tag, tic-tac-toc, iris, wine data sets from UCI Repository

- Result on soybean
  - 100 constraints 48%

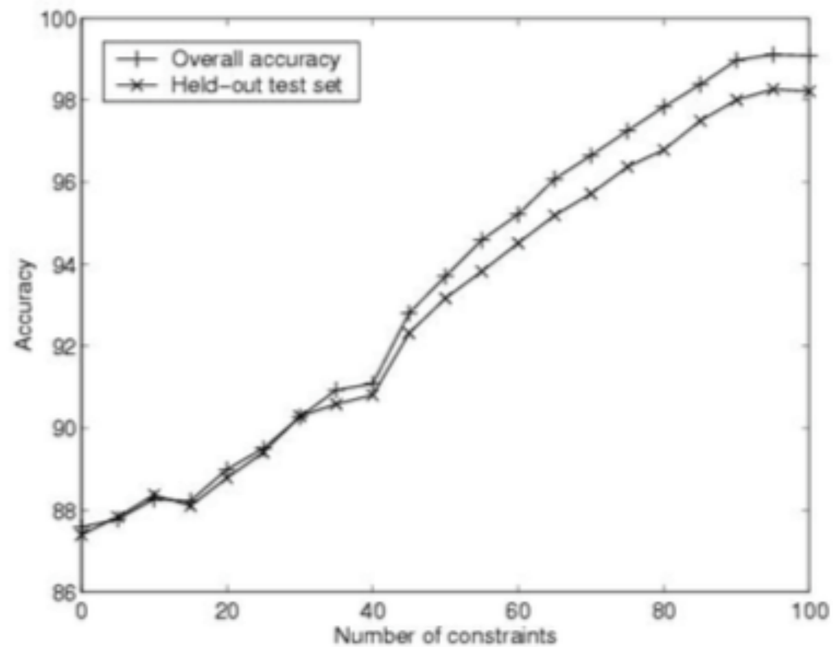


Figure 1. COP-KMEANS results on soybean

- Result on mushroom
  - 100 constraints 73%

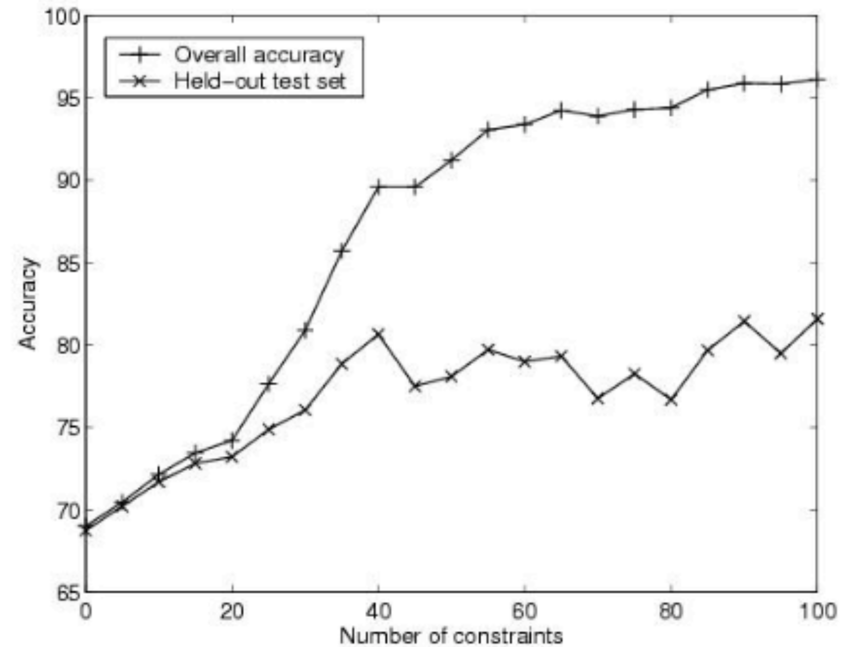


Figure 2. COP-KMEANS results on mushroom

- Part of Speech Tag Data set

Algorithms	Accuracy
K-means	58%
COP-K-means	87%
Held–out accuracy	70%
100 random constraints	56%

- Result on tic-tac-toc
  - 100 constraints 80%

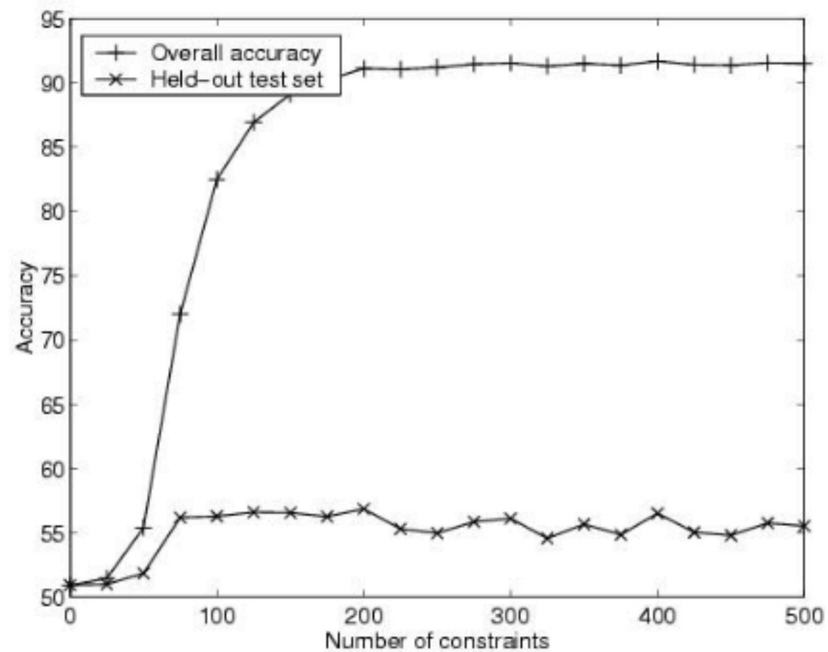


Figure 3. COP-KMEANS results on tic-tac-toe

Experiments – Using Artificial Constraints

- The constrained K-means method could also use with continuous numeric metric and get relatively good result in iris and wine data both from UCI data sets.

- Conclusion:
  - Randomly generated constraints can improve clustering accuracy
  - Improvements can be observed on unconstrained instances
  - Depends on the data set under consideration

Experiments on GPS Lane Finding

- Hypothesis: It would be possible to collect data about the location of cars as they drive along a given road and then cluster that data to automatically determine where the individual lanes are located.

- Data: GPS receivers affixed to the top of the vehicle being driven.
  - distance along the road segment
  - perpendicular offset from the road centerline
- Ask drivers to indicate which lane to use and lane changes for evaluation

Experiments on GPS Lane Finding

- Constraints
  - Trace contiguity
  - Maximum separation

- Represent each lane cluster with a line segment parallel to the centerline instead of average all of its constituent points

Experiment: Compare with K means

- Challenge:
  - Algorithms scaling ability
  - How to select K considering noise in GPS data

- Each algorithm performed 30 randomly-initialized trials with each value of k (from 1 to 5).
  - COP-KMEANS selected the correct value for k for all but one road segment, but k-means never chose the correct value for k

Experiment: Compare with K means

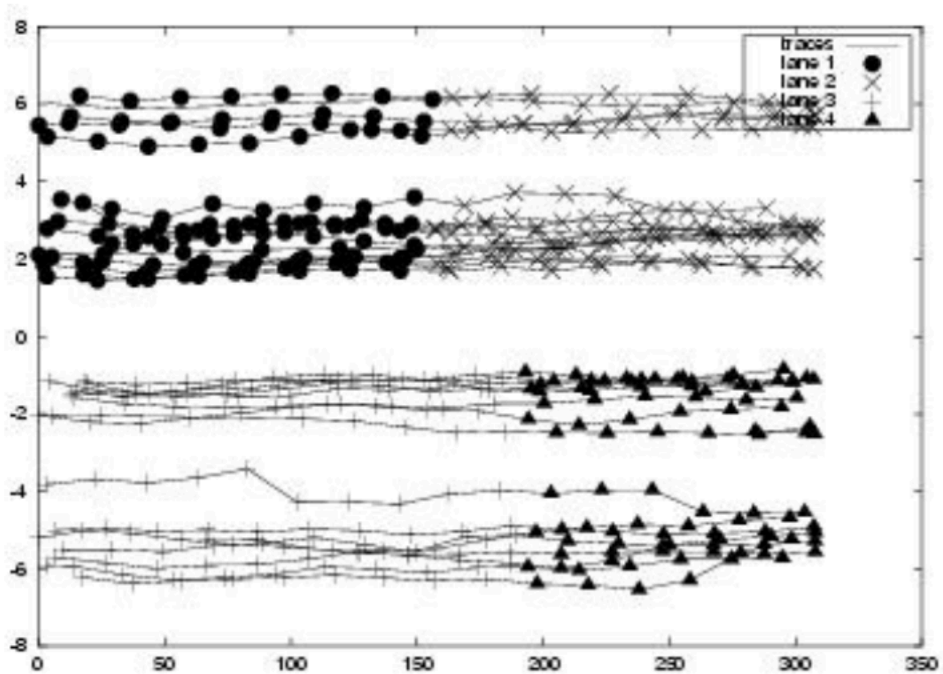


Figure 4. K-means output for data set 6, k=4

Experiment: Compare with K means

Table 2. Lane Finding Performance (Rand Index)

Segment (size)	K-means	COP-KMEANS	Constraints alone
1 (699)	49.8	100	36.8
2 (116)	47.2	100	31.5
3 (521)	56.5	100	44.2
4 (526)	49.4	100	47.1
5 (426)	50.2	100	29.6
6 (503)	75.0	100	56.3
7 (623)	73.5	100	57.8
8 (149)	74.7	100	53.6
9 (496)	58.6	100	46.8
10 (634)	50.2	100	63.4
11 (1160)	56.5	100	72.3
12 (427)	48.8	96.6	59.2
13 (587)	69.0	100	51.5
14 (678)	65.9	100	59.9
15 (400)	58.8	100	39.7
16 (115)	64.0	76.6	52.4
17 (383)	60.8	98.9	51.4
18 (786)	50.2	100	73.7
19 (880)	50.4	100	42.1
20 (570)	50.1	100	38.3
Average	**58.0**	**98.6**	**50.4**

Discussion on this experiment

- Need to compare the algorithms in a larger variety of roads
- Impressive gain in accuracy

Conclusion

- Develop general form to incorporate background knowledge
- Experiments with random constraints on different data sets and show significant improvement
- Demonstrate how background information can be utilized in a real world domain

Thanks!