

Computational Linguistics

Clustering

Stefan Thater & Dietrich Klakow

FR 4.7 Allgemeine Linguistik (Computerlinguistik)

Universität des Saarlandes

Summer 2014

Cluster Analysis

Goal:

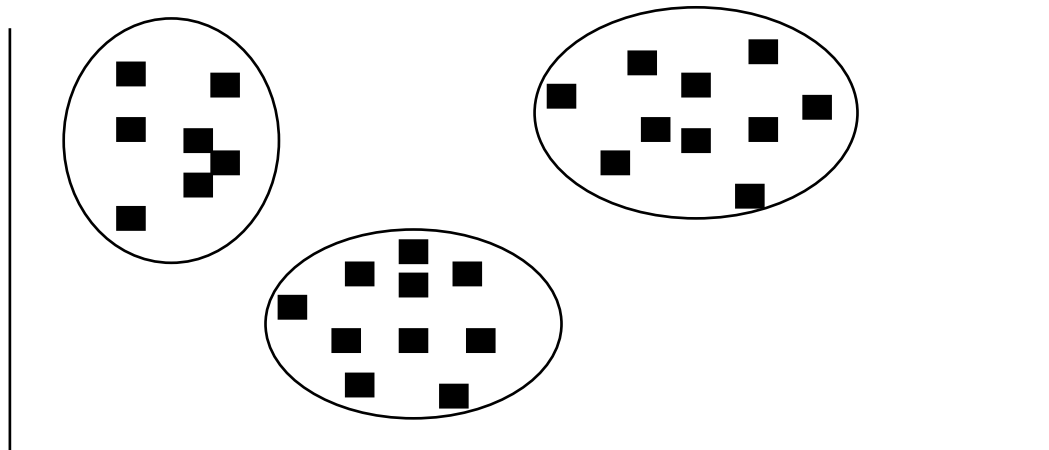
group similar items together in a group

Steps:

define similarity between sample

define a loss function

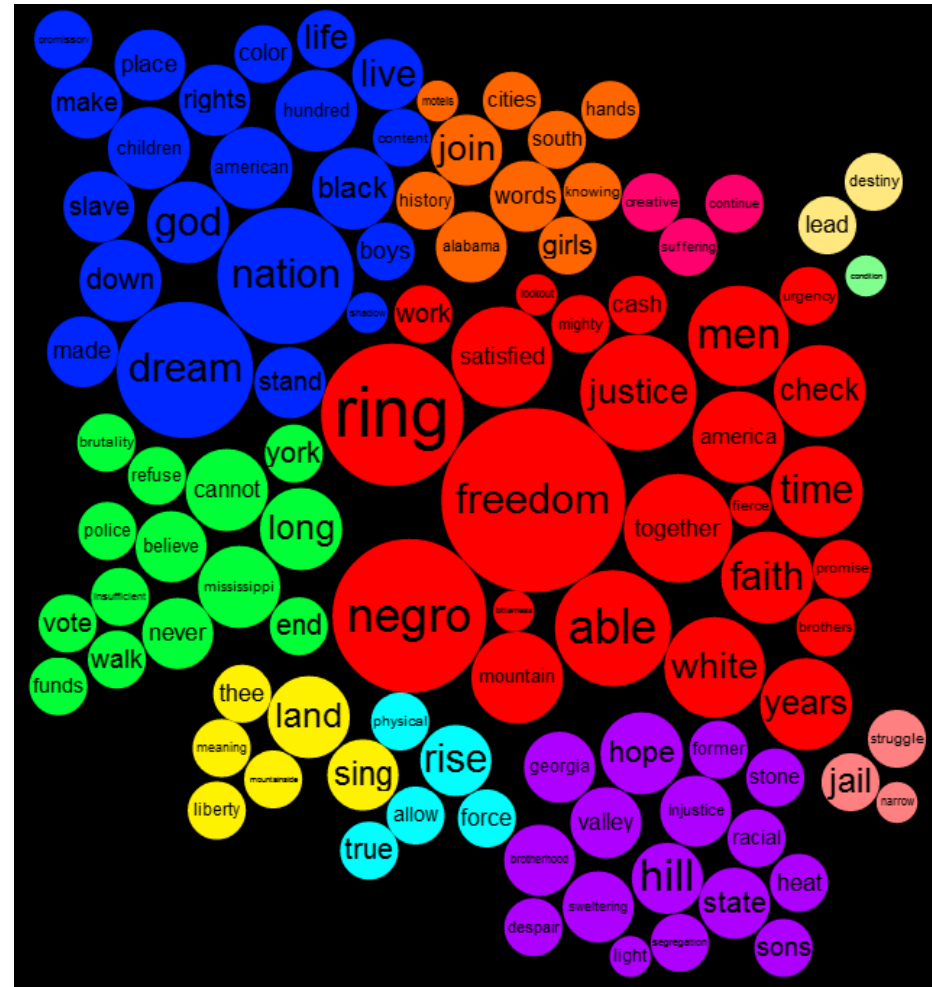
find an algorithm that minimizes this loss function



Examples

Cluster Word

(speech “I have a dream”)



From http://neoformix.com/2011/wcd_KingIHaveADream.png

Cluster Image Regions: Image Segmentation



<http://people.cs.uchicago.edu/~pff/segment/>

Cluster Image Regions

$K = 2$



$K = 3$



$K = 10$

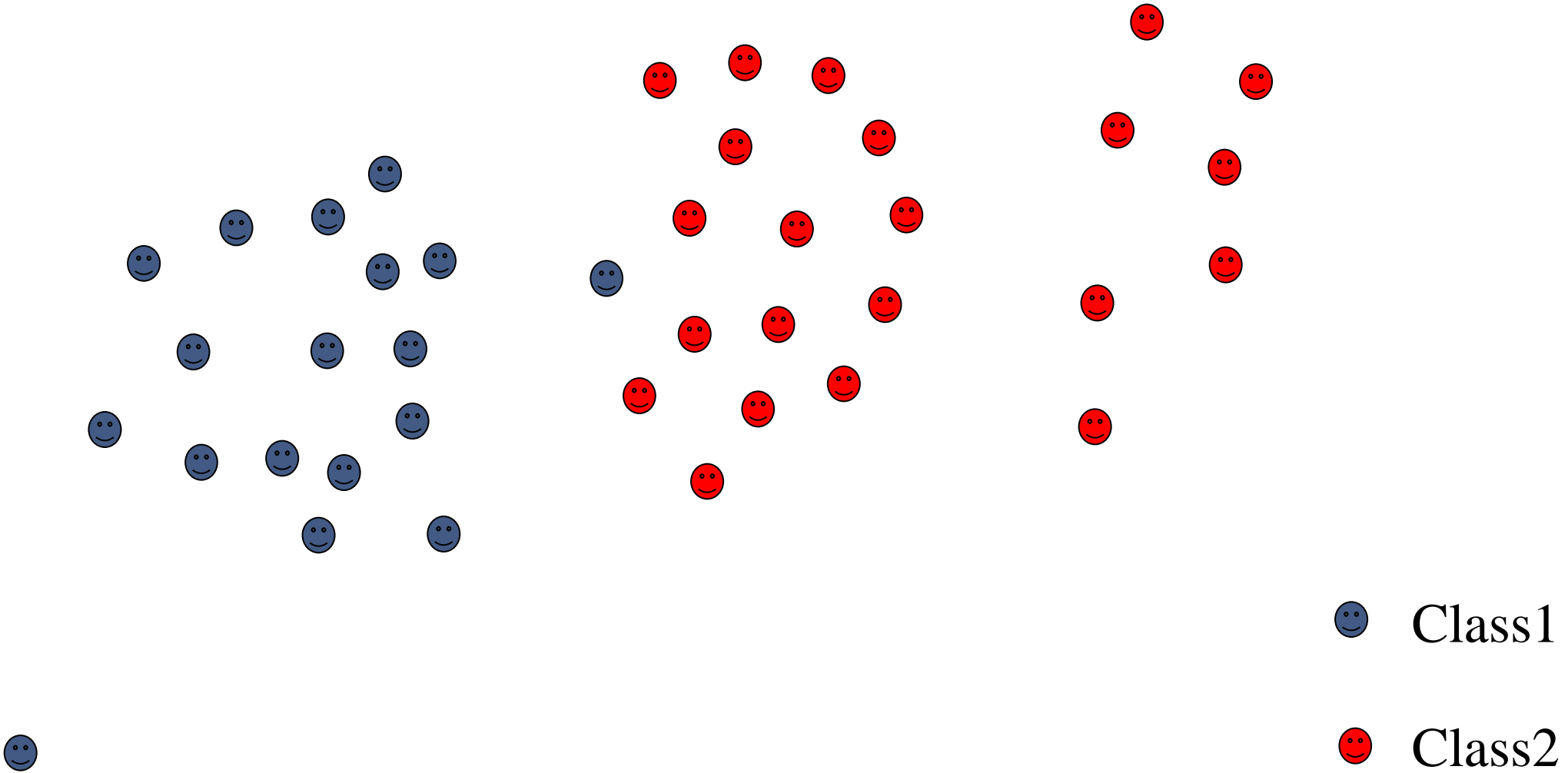


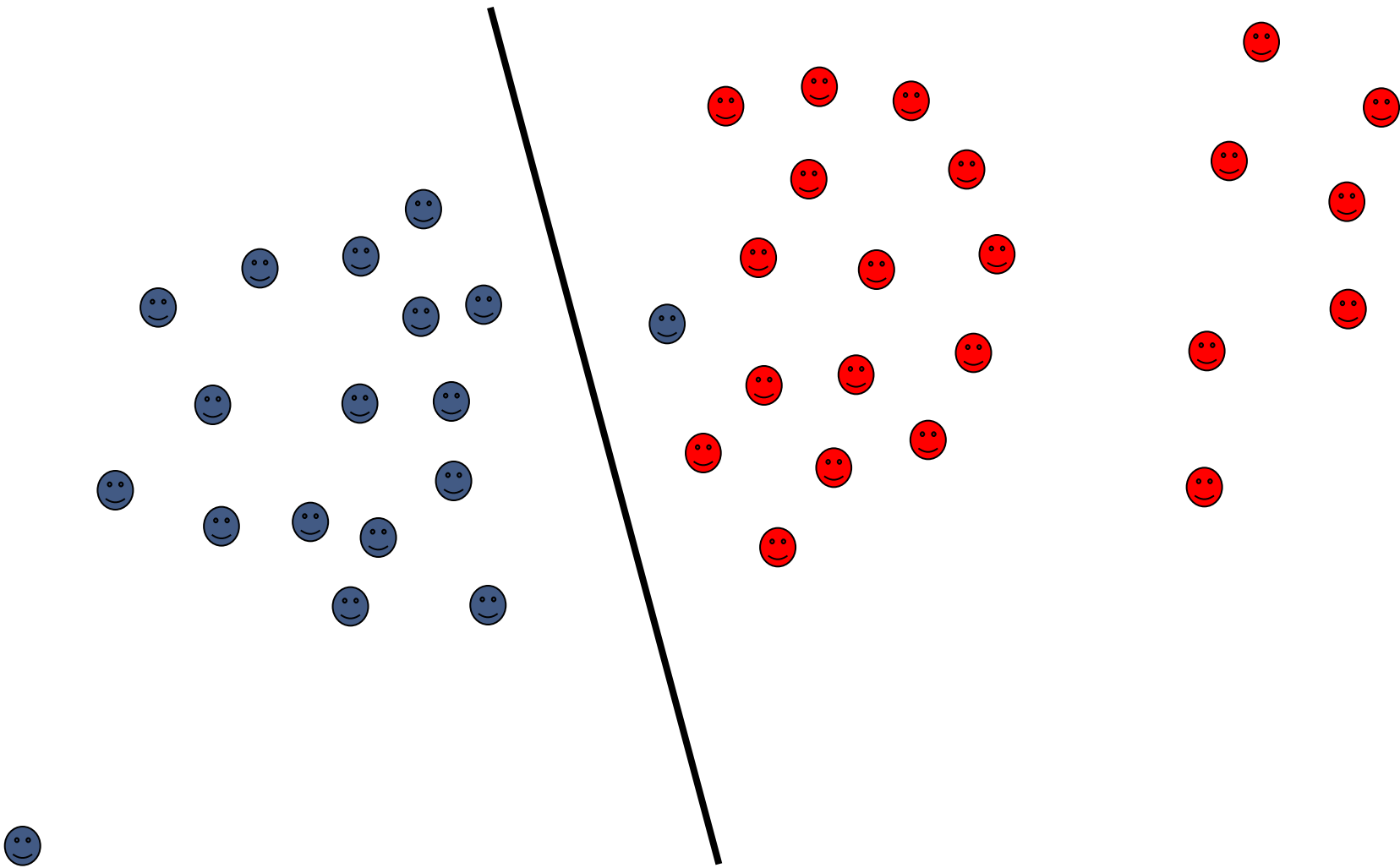
Original image



Unsupervised learning

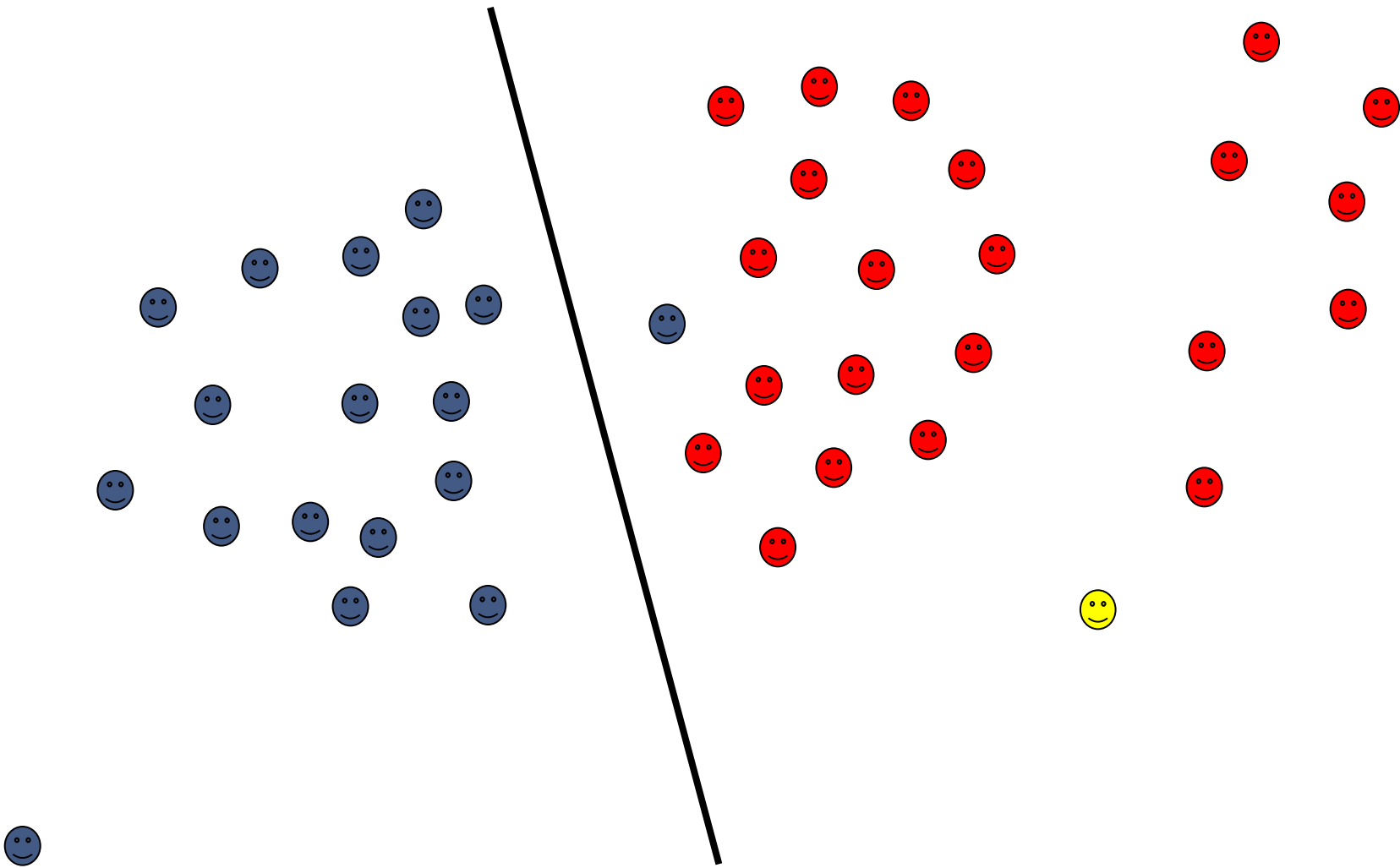
Supervised Classification: Labels known

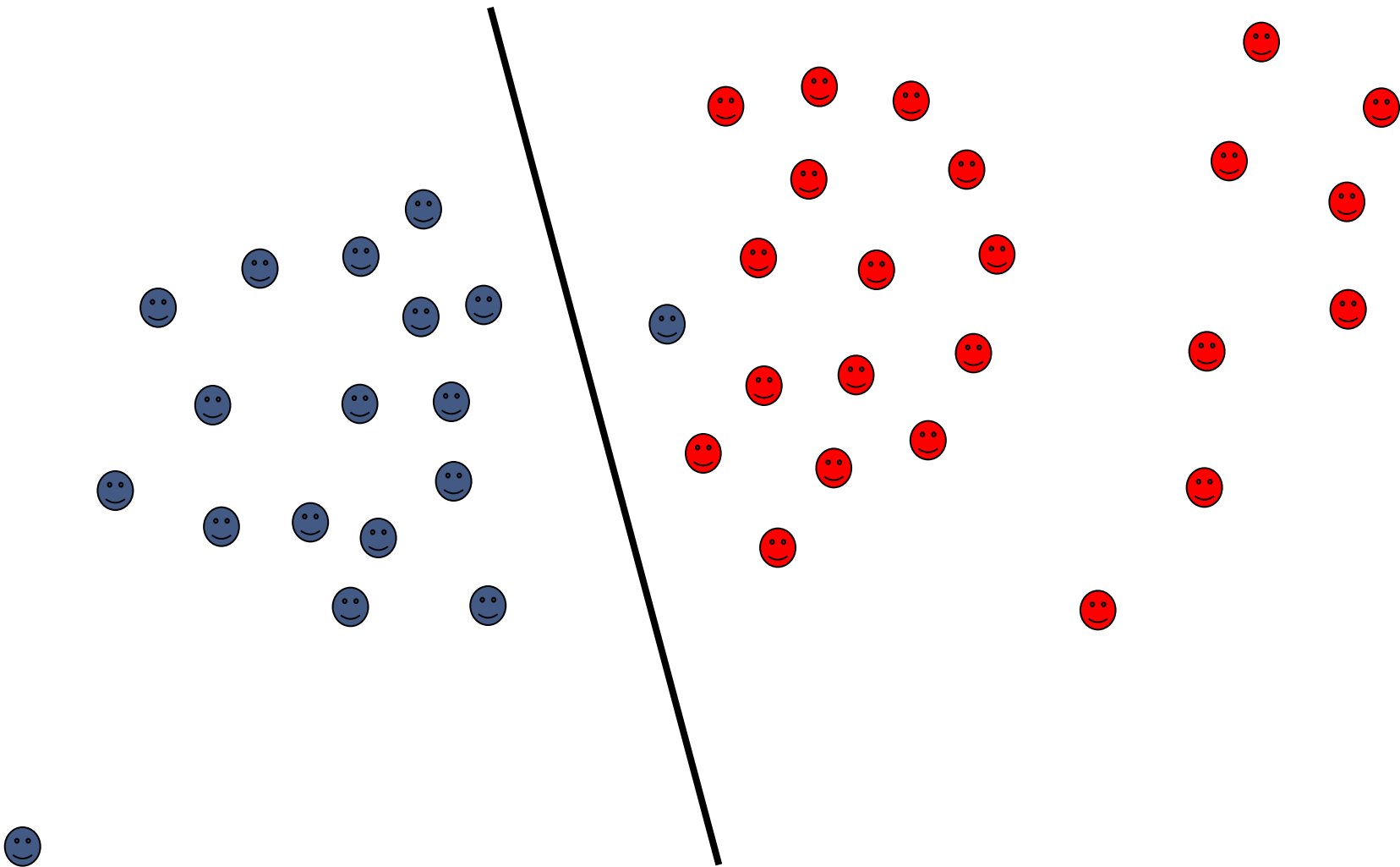




Class 1

Class 2

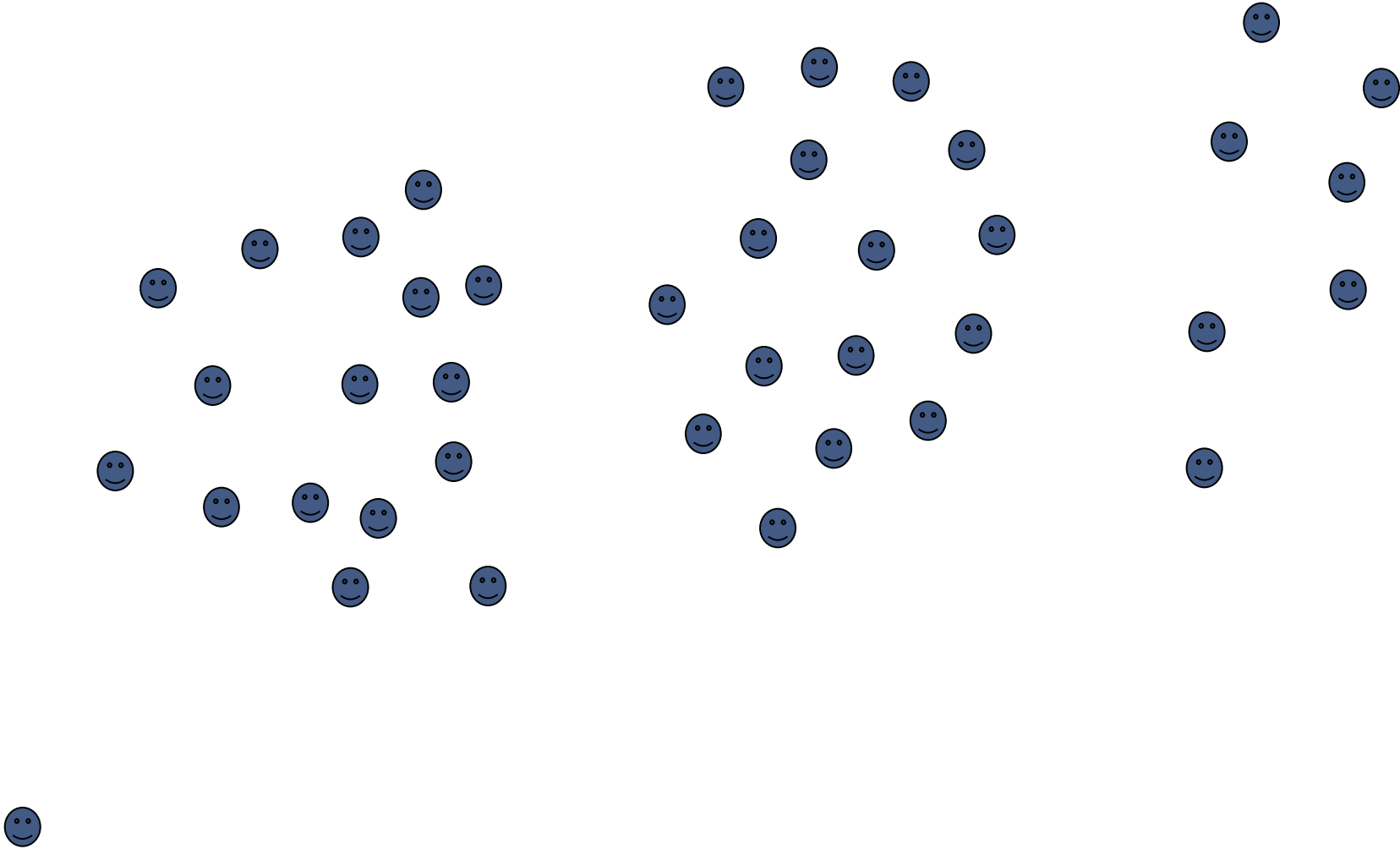




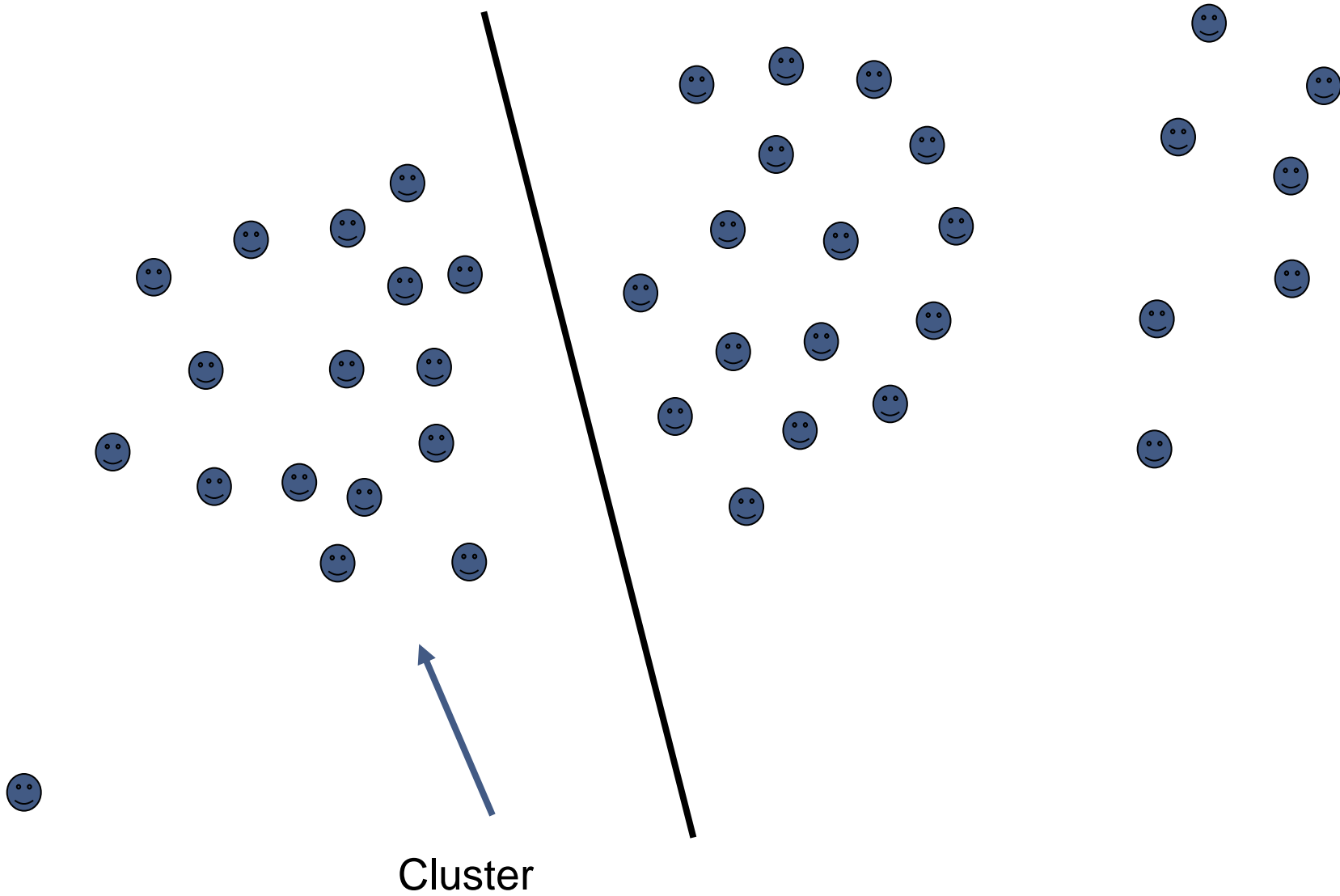
☺ Class1

☹ Class2

Clustering: No labels!

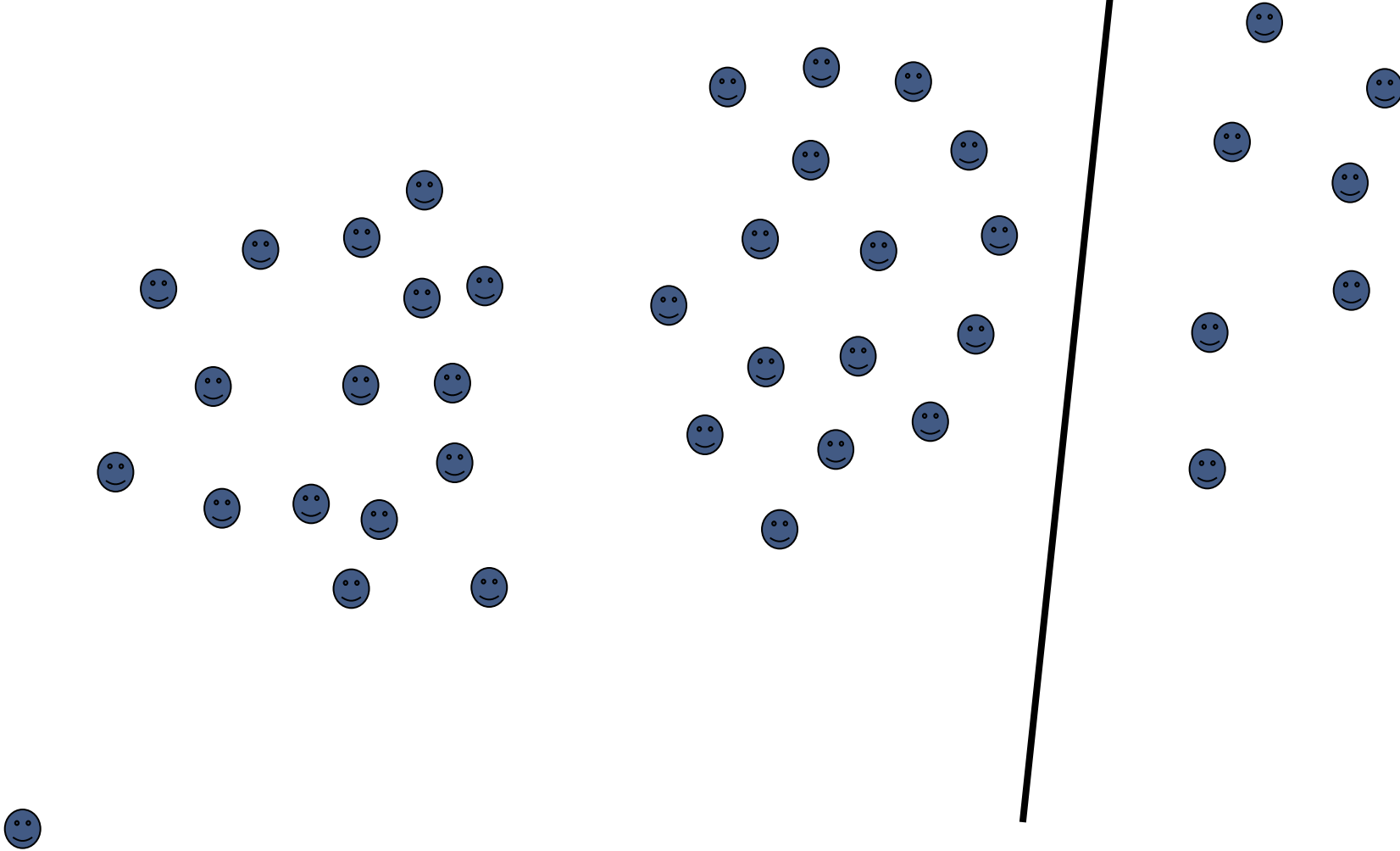


Clustering: No labels!



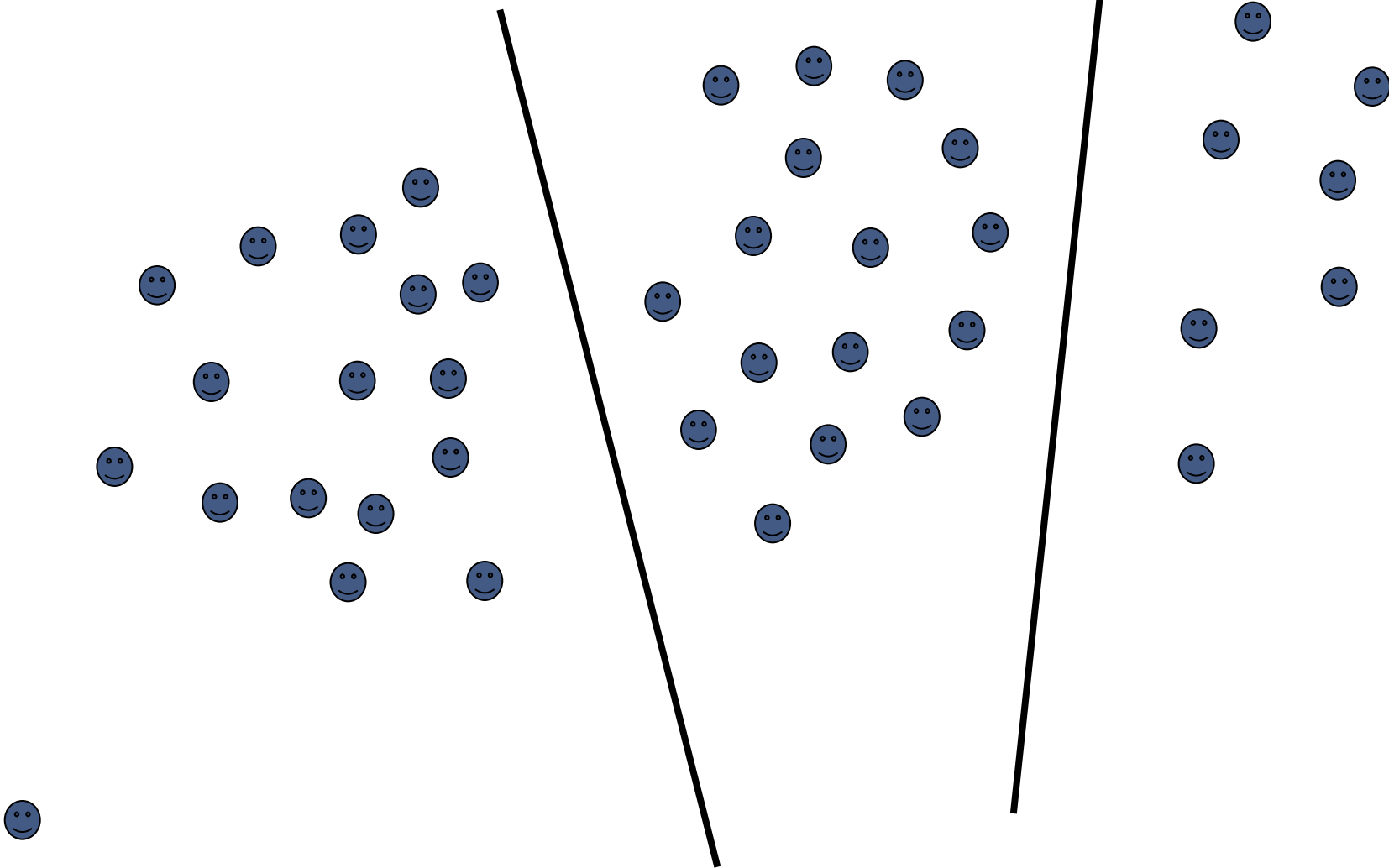
Clustering:

No labels!



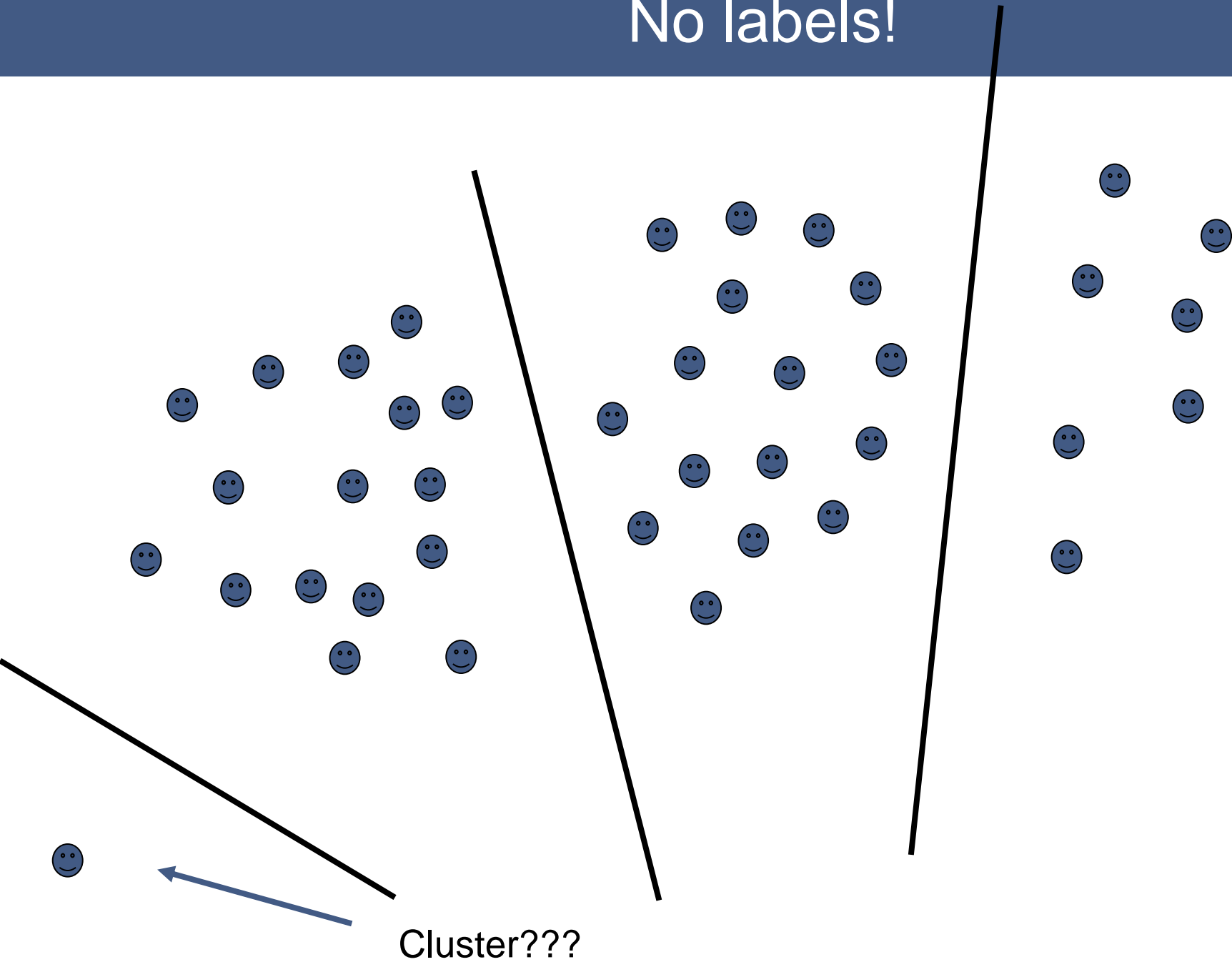
Clustering:

No labels!



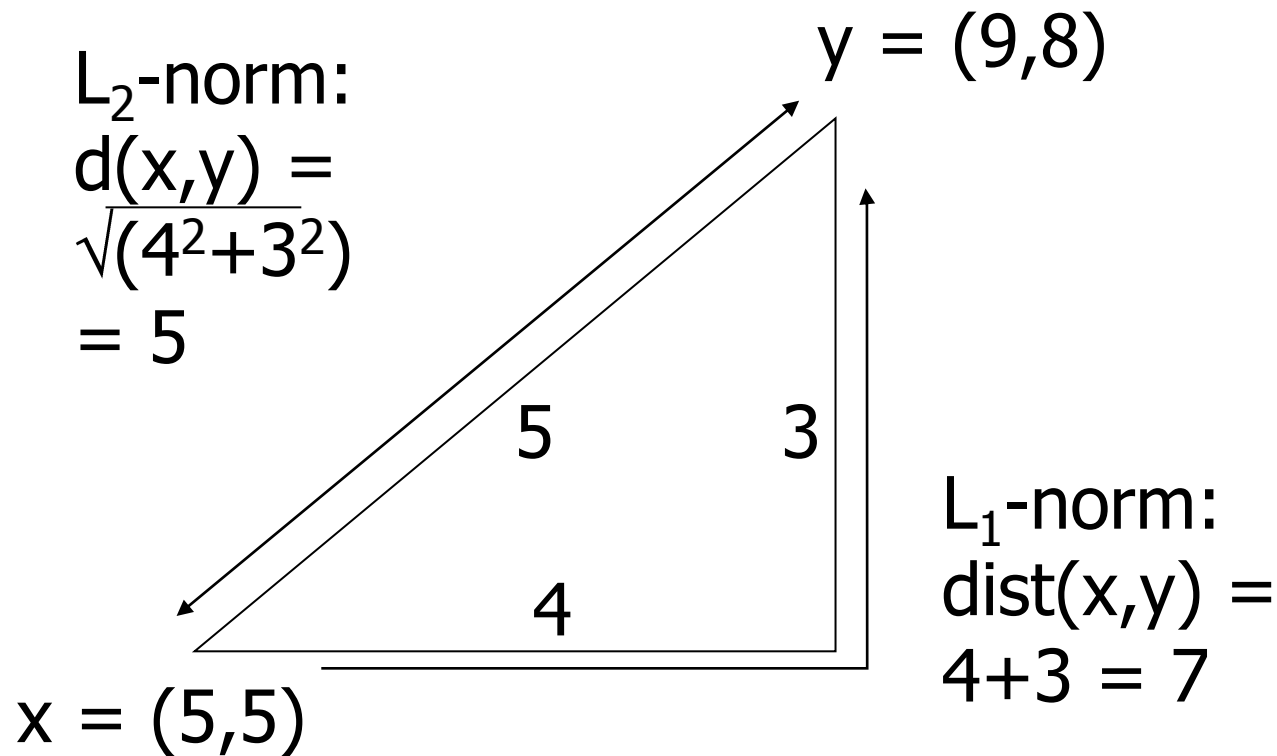
Clustering:

No labels!



Similarity Measures

Euclidean Distances



Axioms of a Distance Measure

d is a *distance measure* if it is a function from pairs of points to reals such that:

$$d(x,y) \geq 0.$$

$$d(x,y) = 0 \text{ iff } x = y.$$

$$d(x,y) = d(y,x).$$

$$d(x,y) \leq d(x,z) + d(z,y) \text{ (} \textit{triangle inequality} \text{)}.$$

Distances measures

L_1 distance (Manhattan distance)

$$d_1(\vec{x}, \vec{y}) = \sum_{k=1}^K |x_k - y_k|$$

L_2 distance (Euclidian distance)

$$d_2(\vec{x}, \vec{y}) = \sqrt{\sum_{k=1}^K |x_k - y_k|^2}$$

L_∞ distance (maximum distance)

$$d_\infty(\vec{x}, \vec{y}) = \max_k (|x_k - y_k|)$$

Example

- Calculate the distance of

$$\vec{x} = \begin{pmatrix} 3 \\ -1 \\ 0 \\ 3 \end{pmatrix} \quad \vec{y} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}$$

- Use all three distance measures introduced on the previous slide

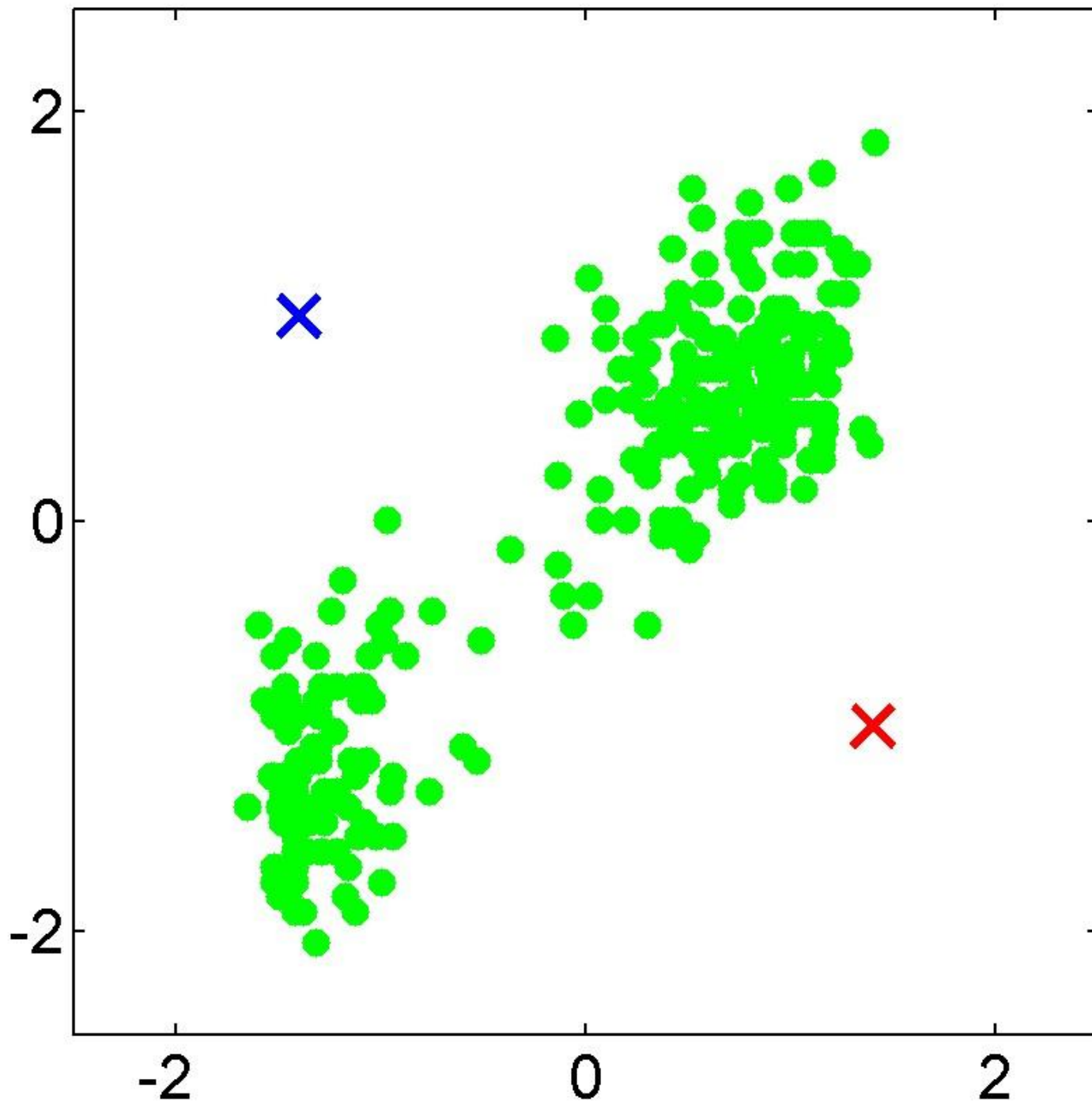
Other distance measures

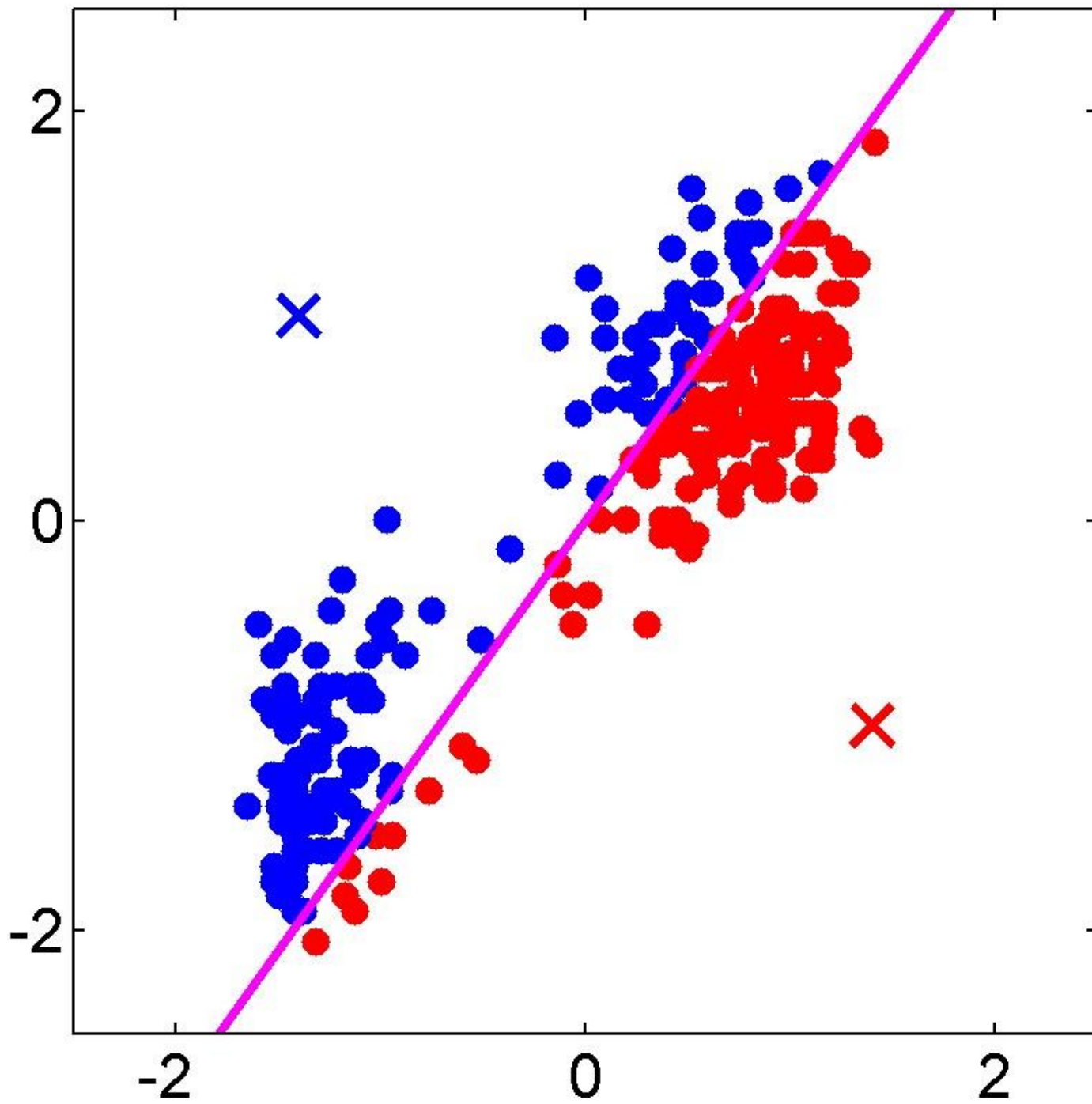
- Cosine
- Edit distance
- Jaccard
- Kernels

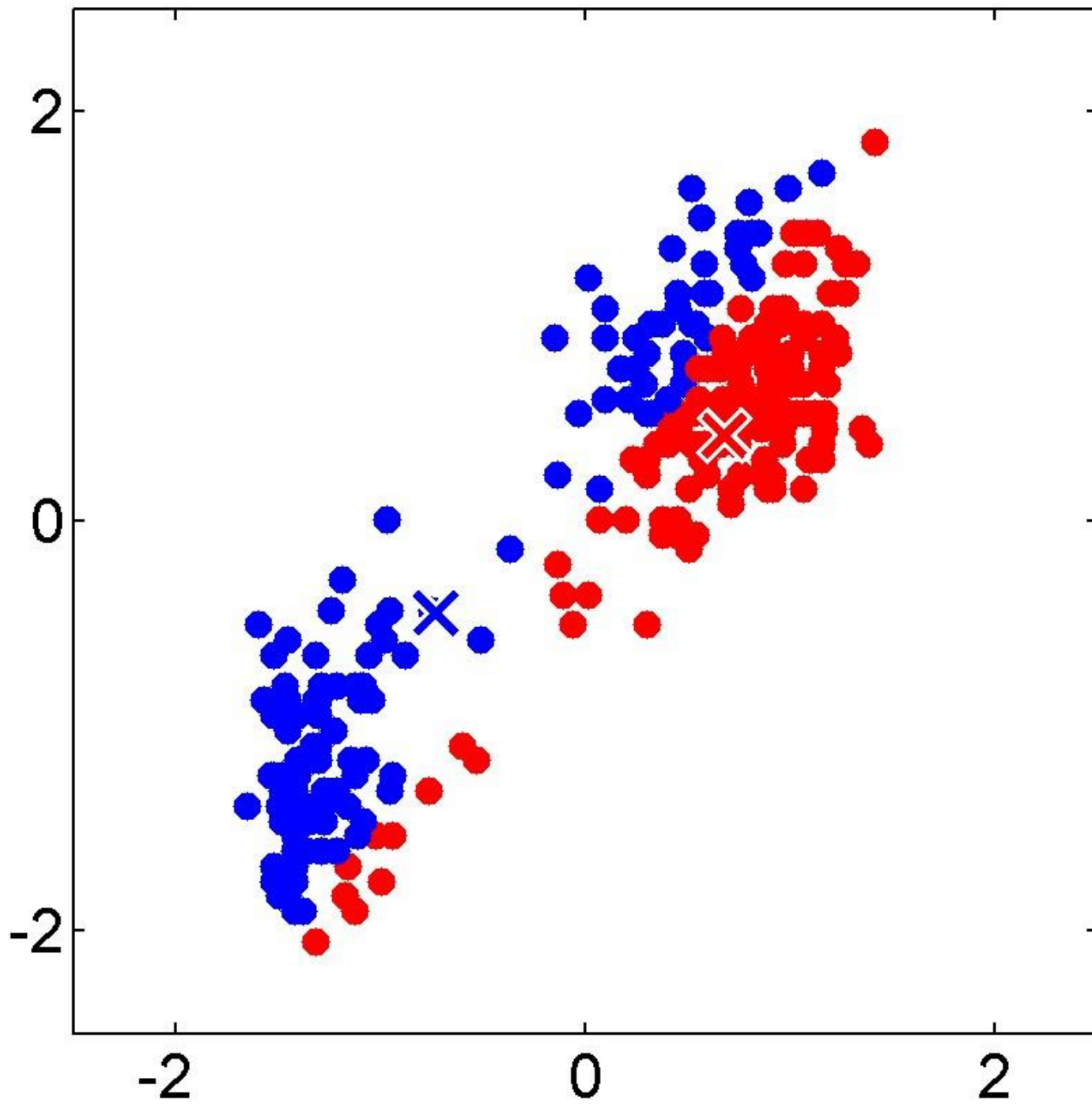
K-Means Clustering

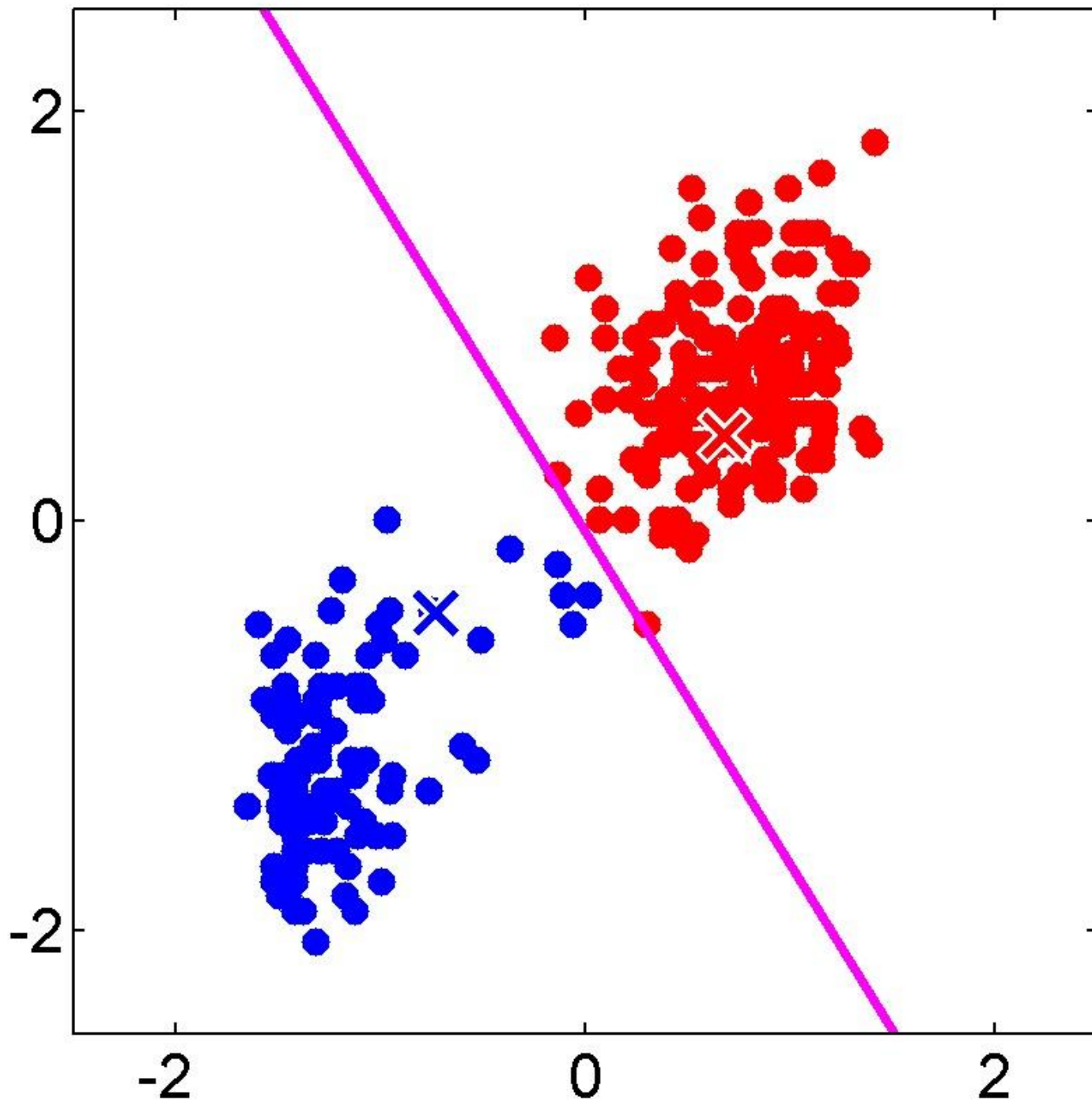
Recipe

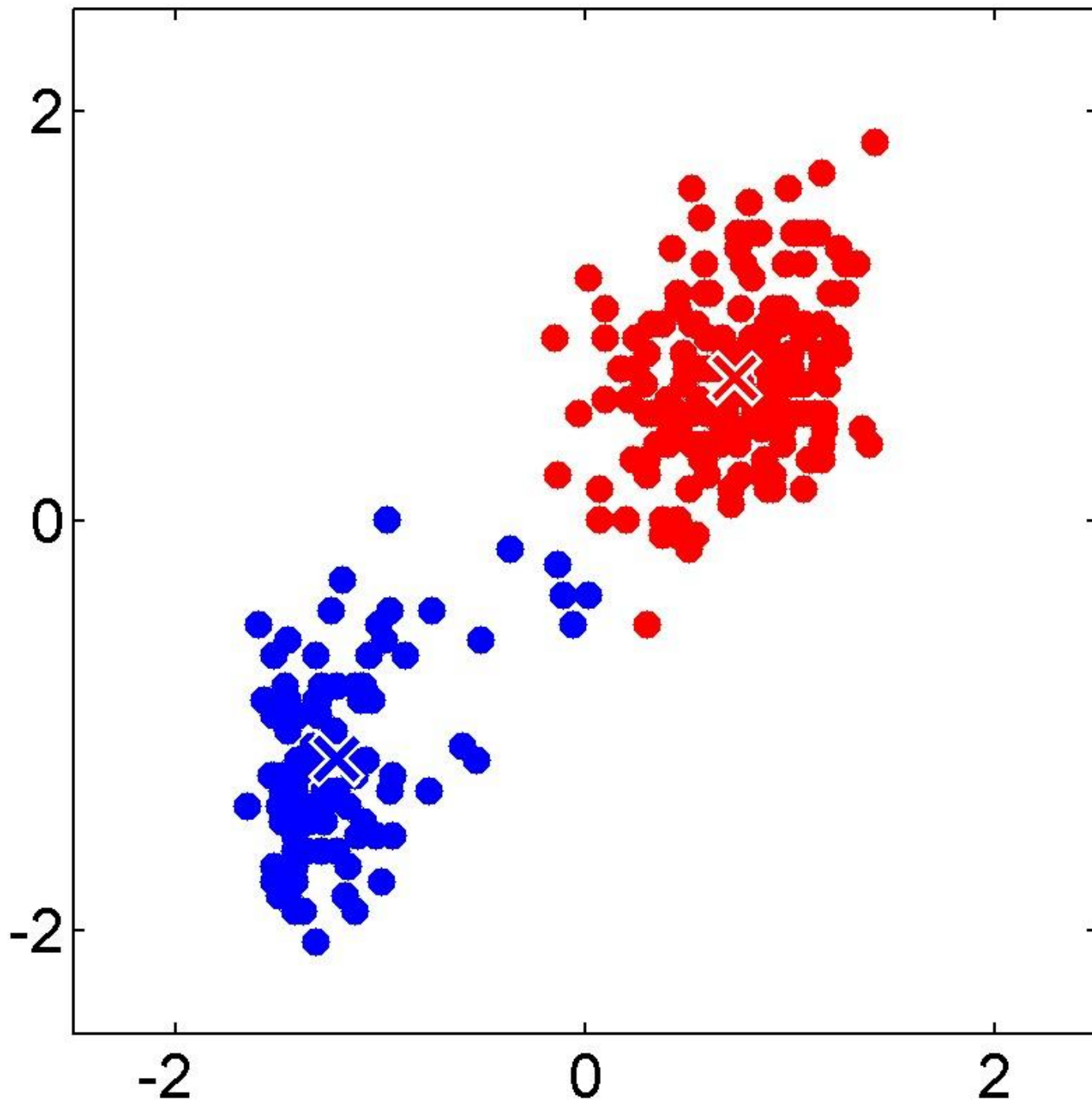
1. For each cluster, decide on a mean
2. Assign each data point to the nearest mean
3. Recalculate means according to assignment
4. If mean changed go back to 1

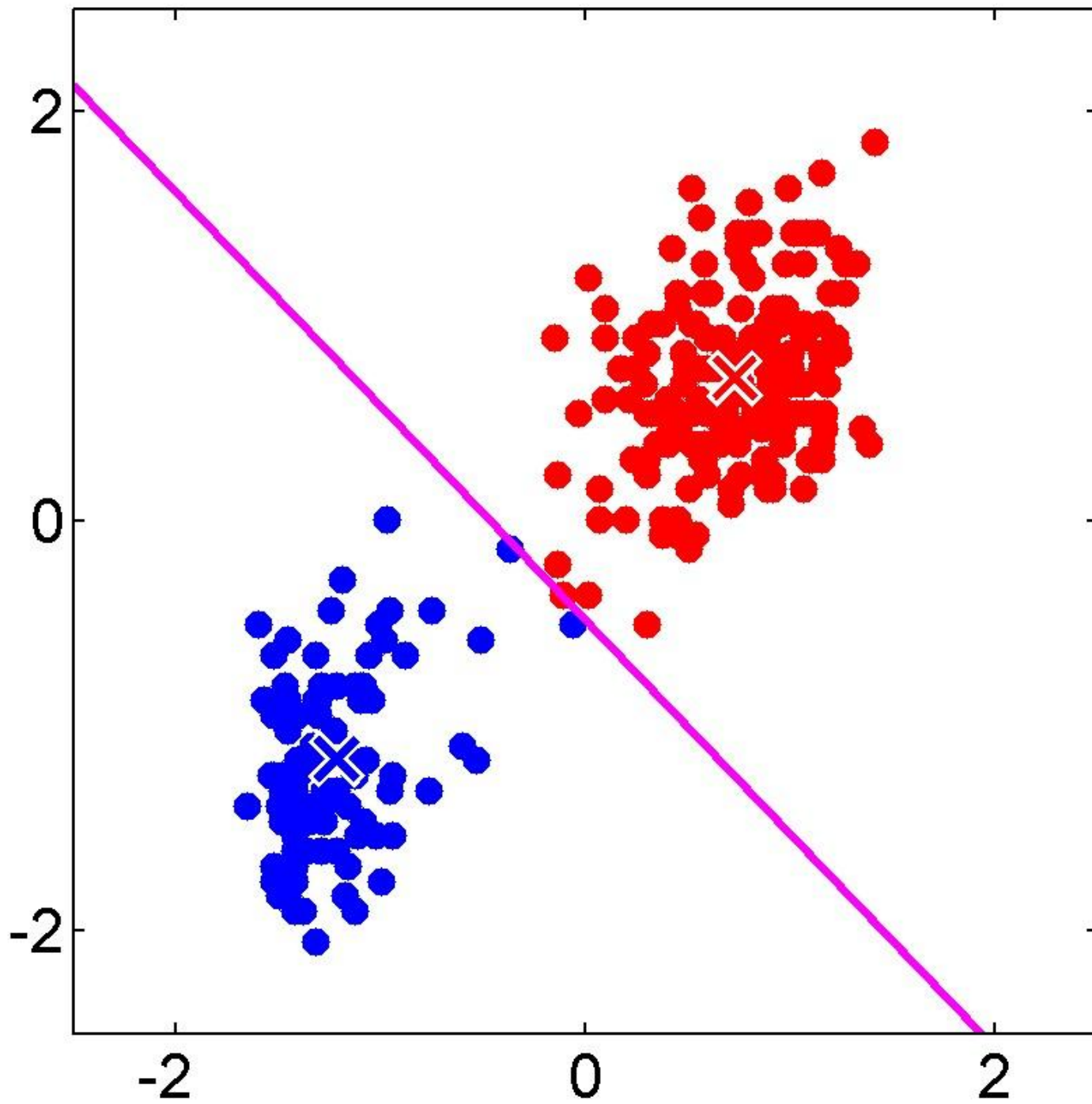


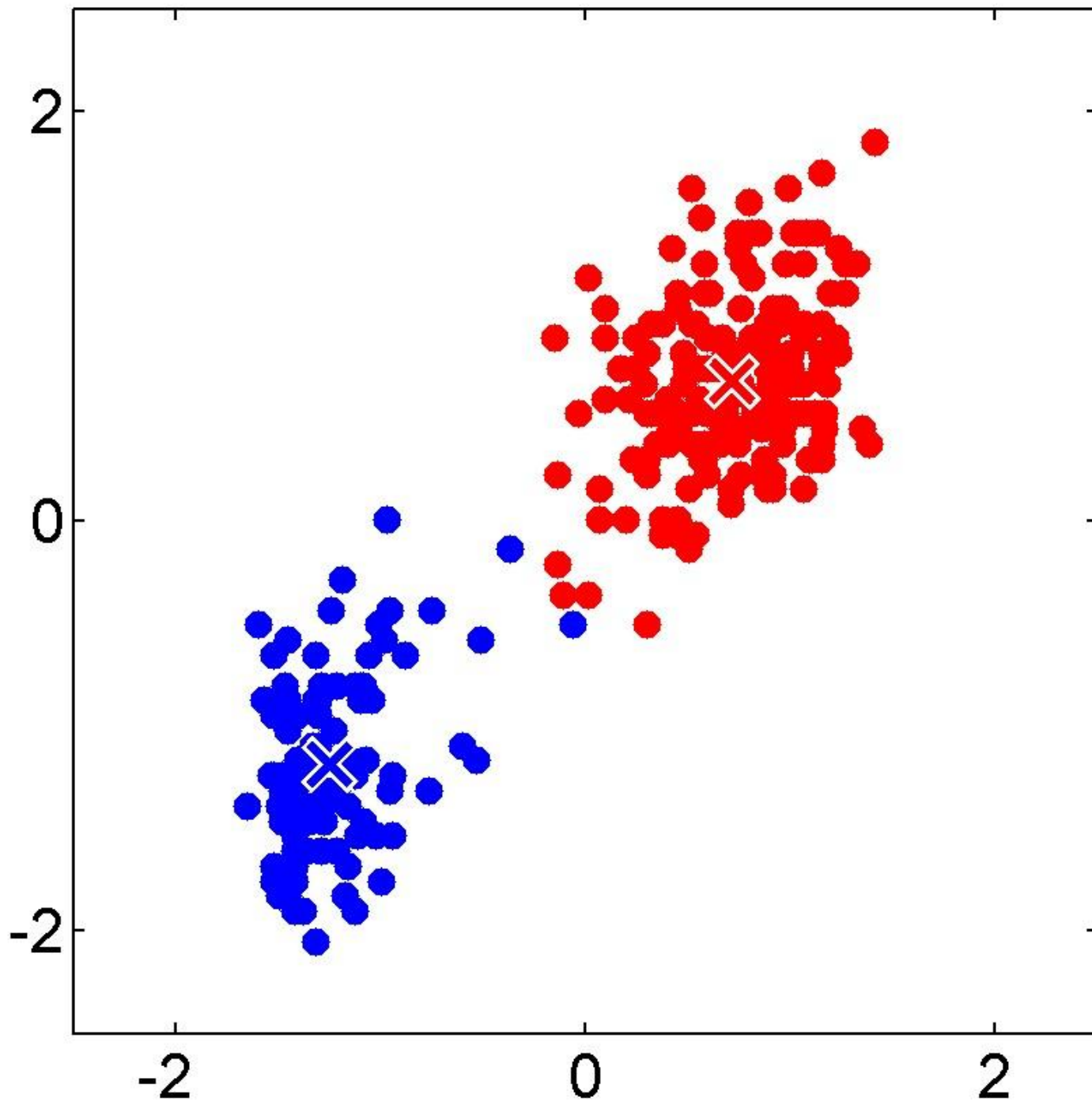


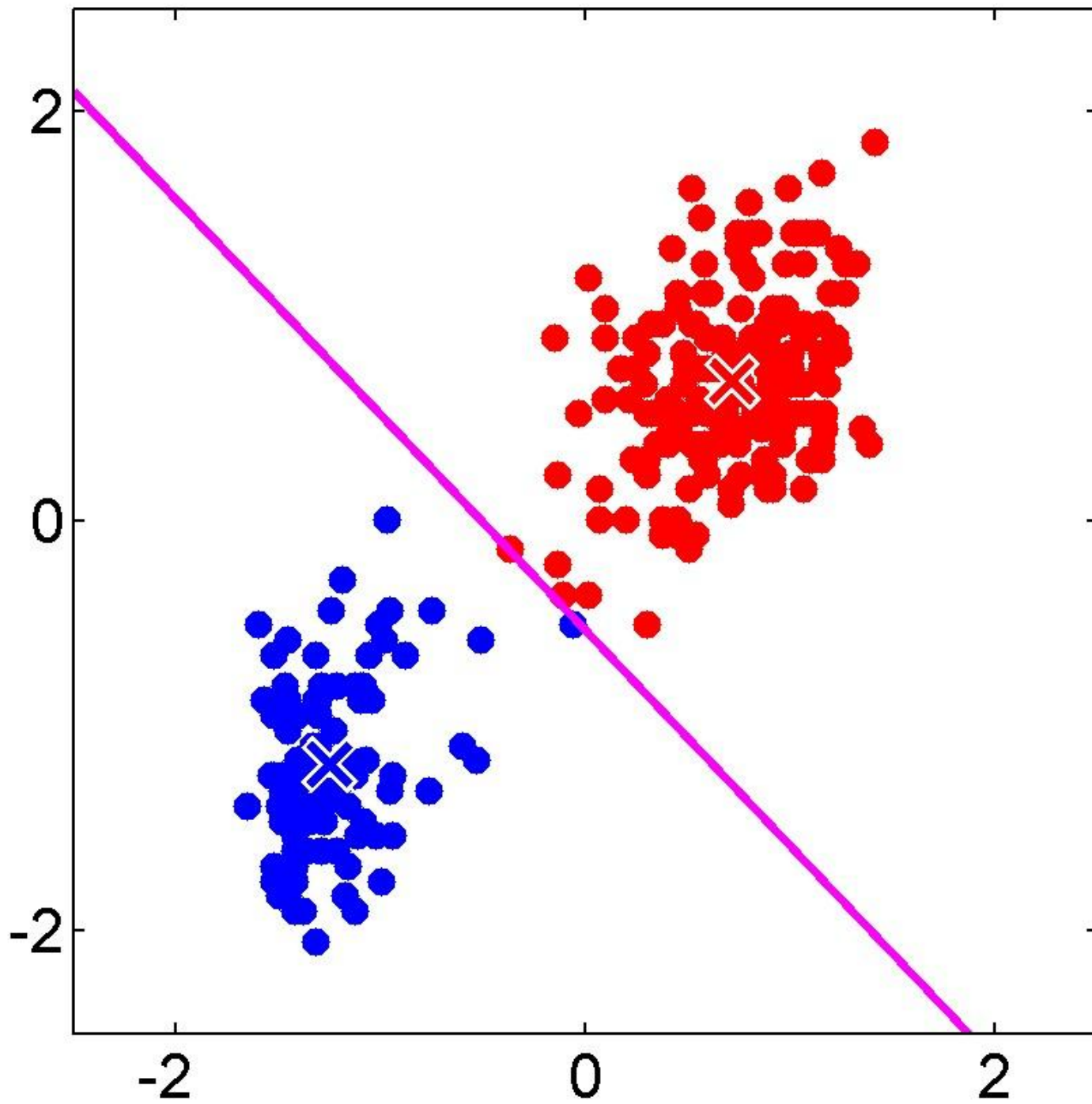


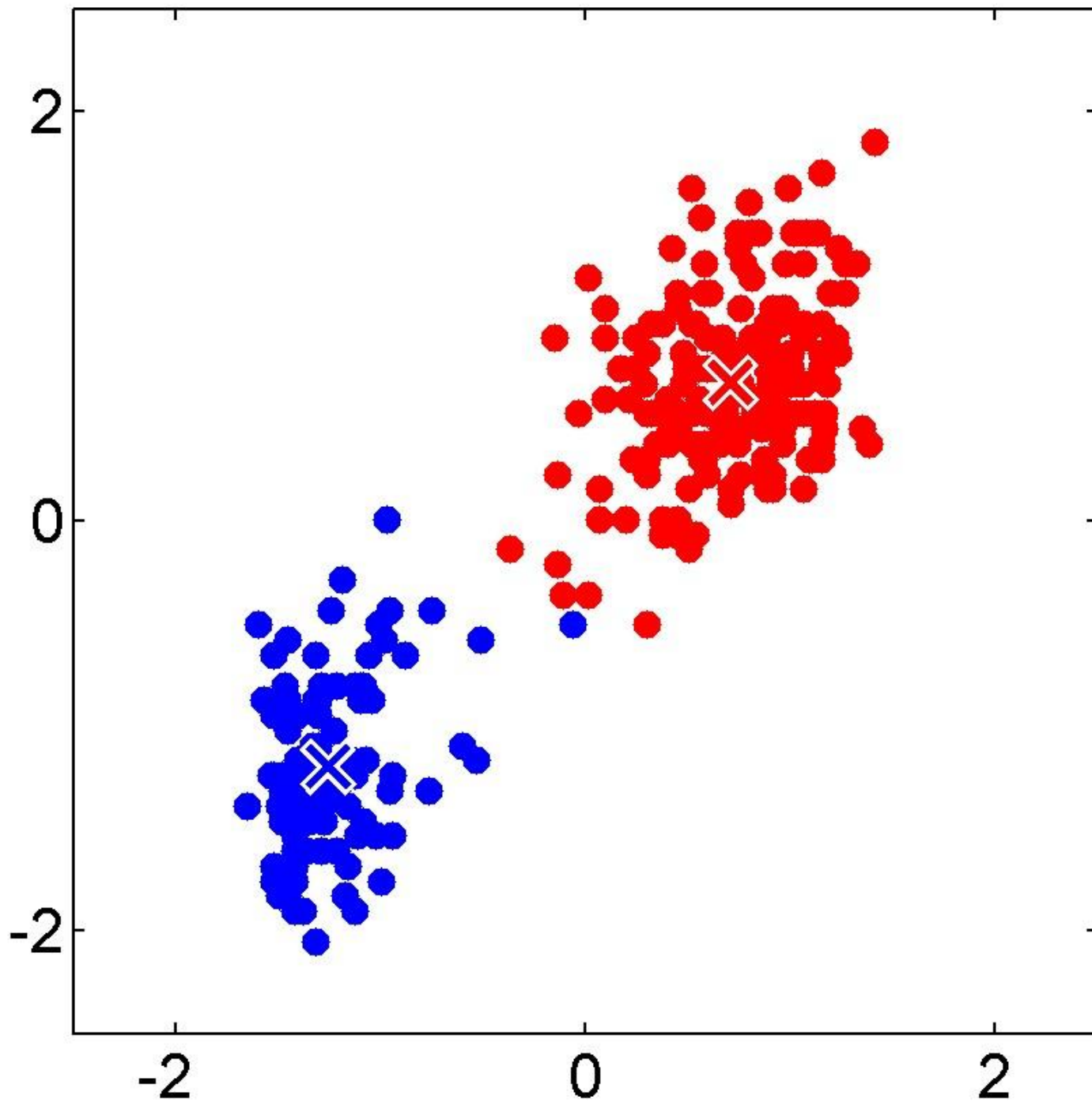












Assignment

$$r_{n,k} = \begin{cases} 1 & \text{if } k = \arg \min_j d(\vec{x}_n, \vec{\mu}_j) \\ 0 & \text{otherwise} \end{cases}$$

\vec{x}_n : n - th training sample (vector)

$\vec{\mu}_j$: mean of the j - th cluster

$d(\vec{x}_n, \vec{\mu}_j)$: distance (your choice, e.g. L_2)

Example

$$r_{n,k} = \begin{cases} 1 & \text{if } k = \arg \min_j d(\mathbf{x}_n, \vec{\mu}_j) \\ 0 & \text{otherwise} \end{cases}$$

See white board

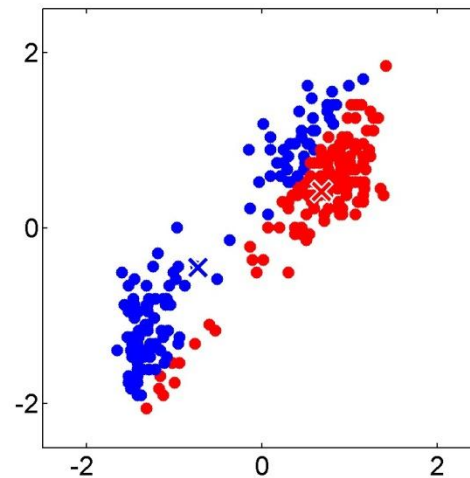
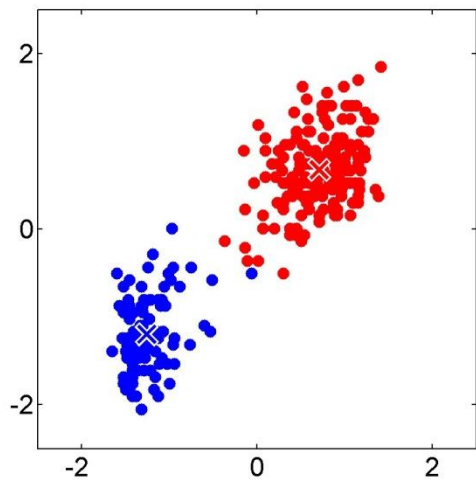
Update mean

$$\vec{\mu}_k = \frac{\sum_{n=1}^N r_{n,k} \vec{x}_n}{\sum_{n=1}^N r_{n,k}}$$

Interpret the denominator

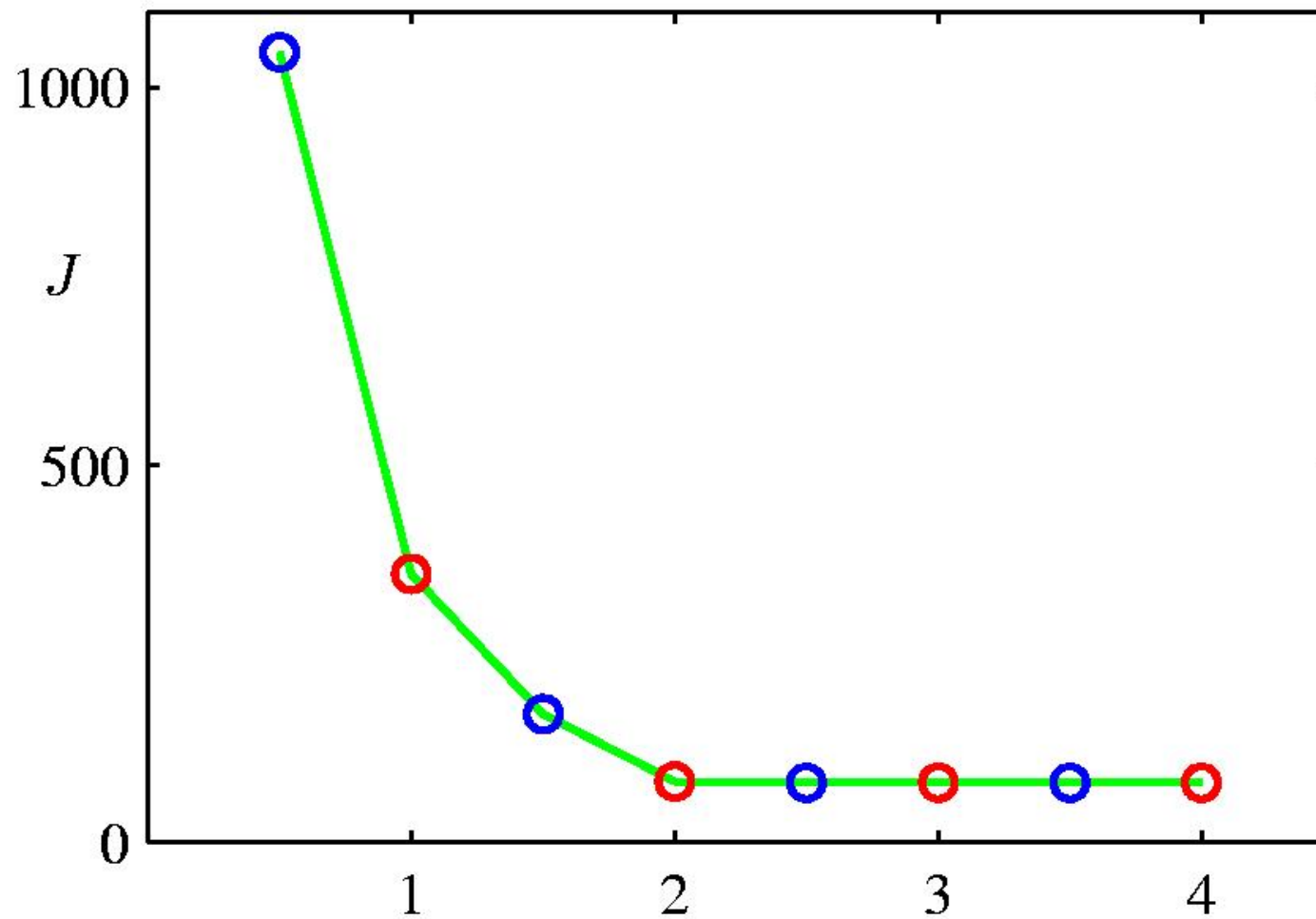
Loss Function: Distortion Measure

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{n,k} d(x_n, \mu_k)$$



Which of the two
has the smaller J?

Distortion Function after each iteration



How to initialize K-Means

- Converges to local optimum
- Outcome of clustering depends on initialization
- Heuristic:
 - pick k vectors from training data
(being furthest apart)

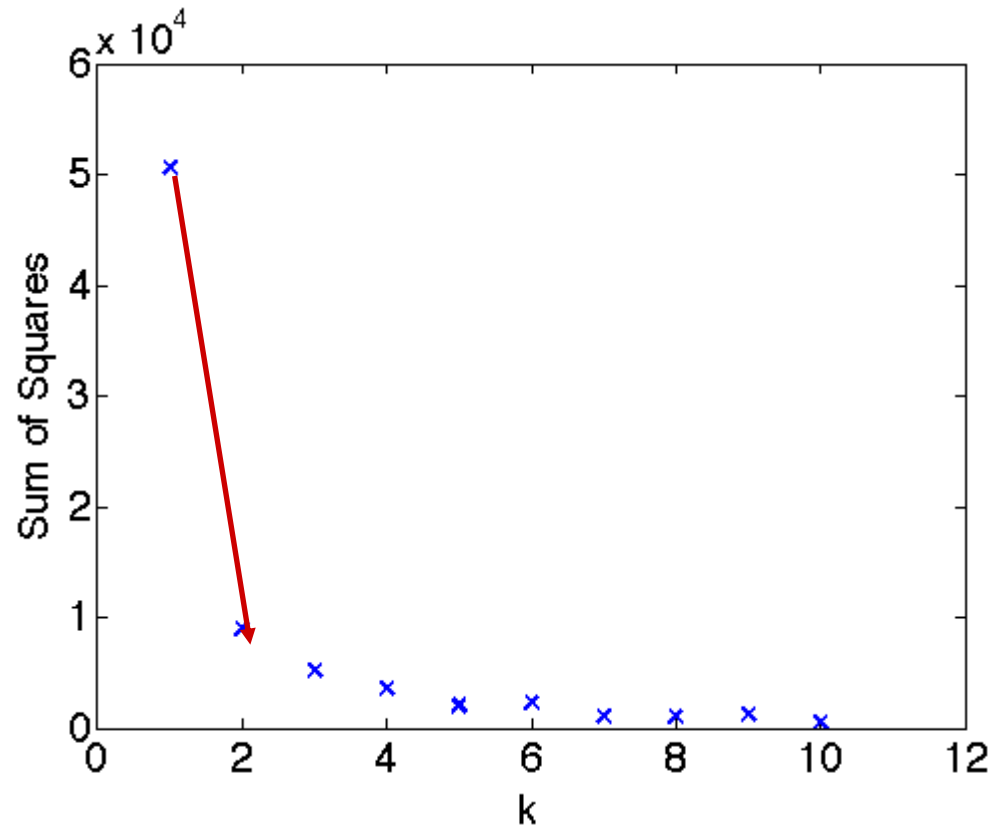
How to determine k

What about
picking k such J
becomes as small
as possible? ?

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{n,k} d(x_n, \mu_k)$$

How to determine K

- For $K=N$ the distortion $J=0$
- Solution: find large jump

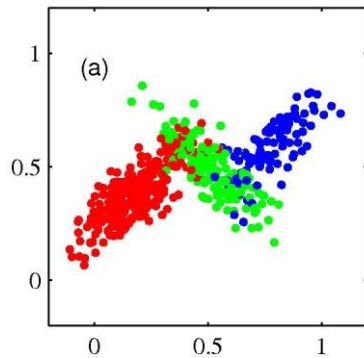


Other aspects of clustering

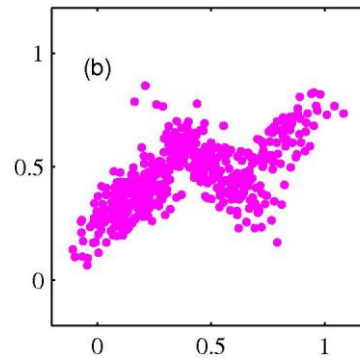
Soft clustering

No strict assignment to a cluster

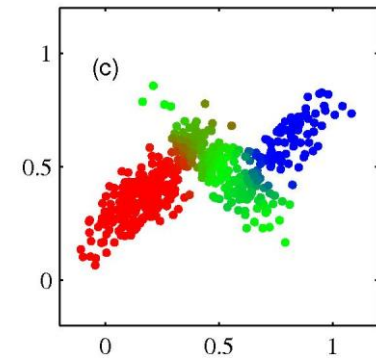
Just probabilities



Original data
Overlapping class regions



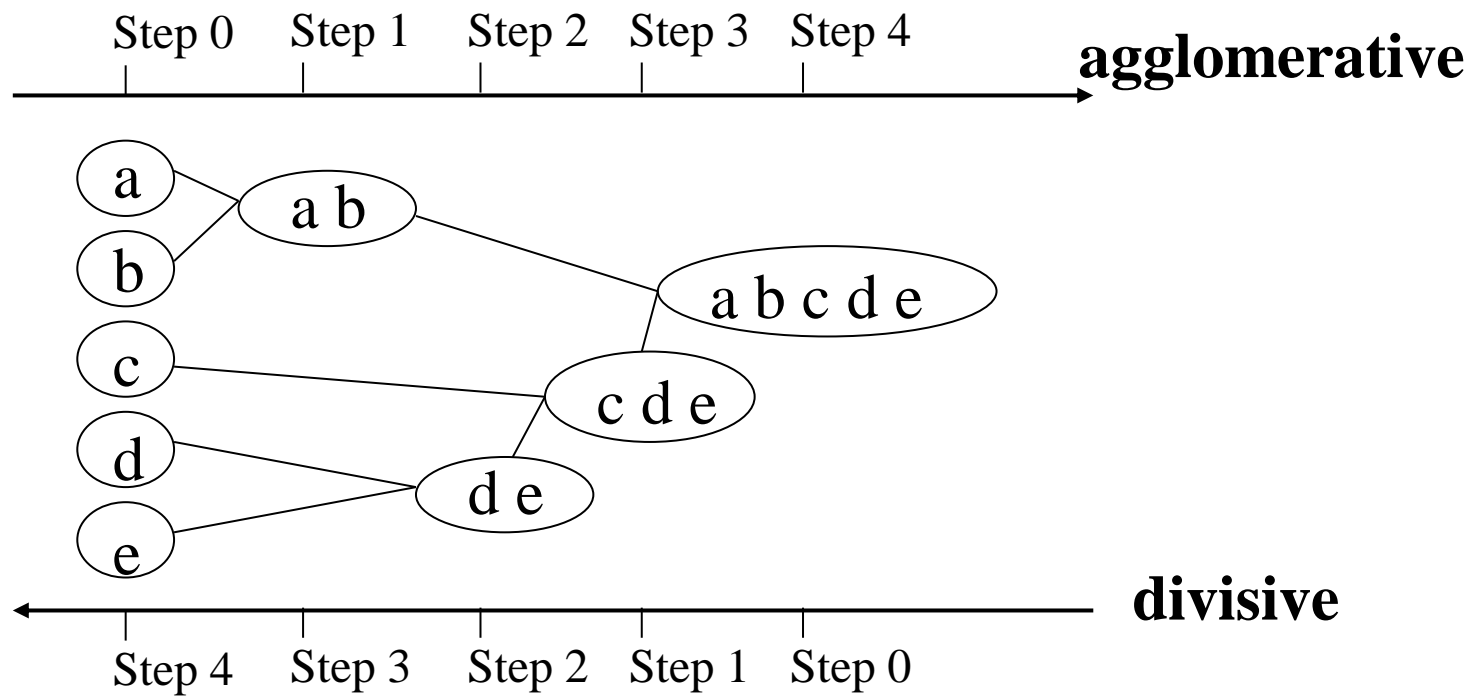
No class
information



Soft clustering

Hierarchical Clustering

Organize cluster in a hierarchy



Text clustering

Derive features from documents

- Frequency of words
- TF-IDF of words
- Stop wording?
- Stemming?

Practical Example

Exercise

At

<http://research.microsoft.com/enus/um/people/cmbishop/prml/webdatasets/faithful.txt>

You find some two dimensional data.

Implement a k-means algorithm for two clusters using just the first (!) column

Homework

1. Apply your method only to the second column
2. Generalize your algorithm to vector valued data and an arbitrary number of clusters. Apply it to the full data set with both columns
3. Suppose the first column has value $c_1(i)$ and the second $c_2(i)$. Is there a new $c(i) = a * c_1(i) + b * c_2(i)$ with suitable well picked a and b such that clustering based on $c(i)$ is better than the one done on $c_1(i)$ or $c_2(i)$ alone.

Word Clustering using the Brown Algorithm

Idea

Cluster words together that have similar neighbours

Minimize perplexity on training test

The Brown Algorithm

start with some initial mapping $w \rightarrow g_w$
for each word w of the vocabulary do
for each class k do
tentatively exchange word w from class g_w to class k and update counts
compute perplexity for this tentative exchange
exchange word w from class g_w to class k with minimum perplexity
do until stopping criterion is met

g_w : class of word w

Example clustering

Cluster	Example members
1	Groß, Rau, Müller, Zimmermann, Frei, Becker, Möllemann, Schmidt
2	Düsseldorf, Berlin, München, Köln, Stuttgart, Hannover, Hamburg
3	nahmen, macht, zeigt, gleichen, bringt, biete, machte, sorgt, enthält

Application in Named Entity Tagging

Training

Word	Class label	Tag
Düsseldorf	C2	City
is	X	O
the	X	O
capital	X	O
of	X	O
NRW	X	O

Application in Named Entity Tagging

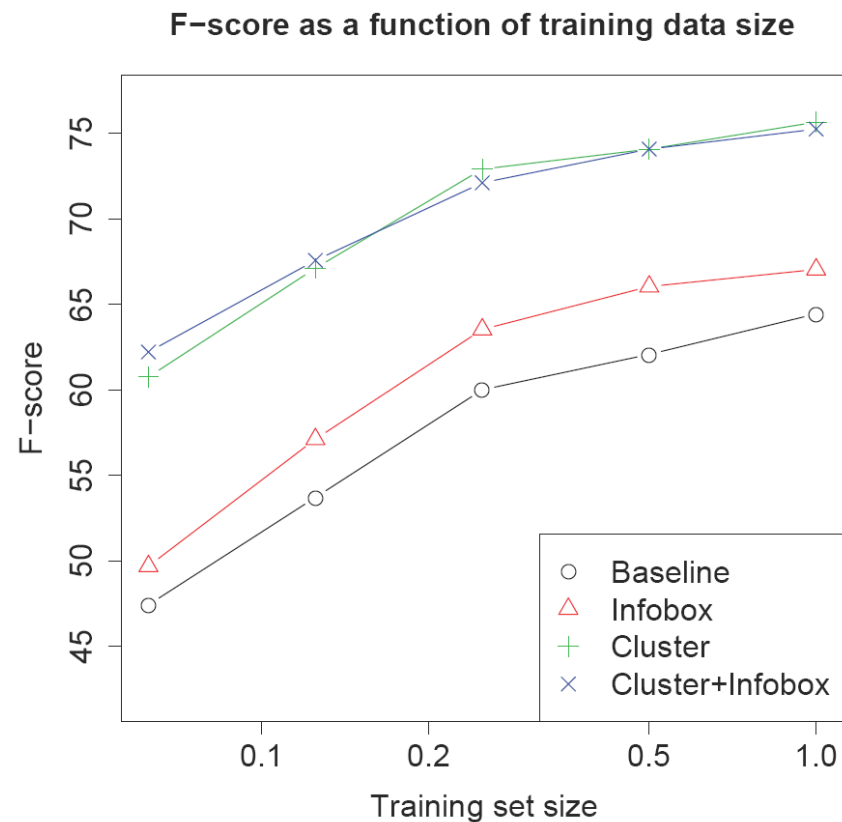
Testing

Word	Class label	Tag
The	X	O
Hofbräuhaus	X	O
is	X	O
in	X	O
Munich	C2	???

How to tag if Munich is not in the training data?

Application

Use class labels as features in named entity tagging



Summary

- Clustering: finding similar items
- Distance metrics
- K-Means
- Brown Algorithm