PROBLEMS WITH BASIC ALGORITHM

1. Overfitting – The decision tree works well on the data from which it was constructed but not on other data from the same population.

2. How can we get the most accurate tree? Examine all possible trees?

3. Which attribute to choose for splitting?

4. Should we split at all?

SPLITTING

How to choose the attribute for splitting?

disorder = entropy = information

Switch to C4.5/J48 terminology

1. Maximize information gain

2. Maximize information ratio

- For 8 equally probable messages the information in any one is

$$-log_2(1/8) = 3\ bits$$

- A message that a case in a set S is in class C_j has probability

$$\frac{freq(C_j, S)}{|S|}$$

- information in the message is

$$-log_2\left(\frac{freq(C_j, S)}{|S|}\right)$$

INFORMATION GAIN

- Information in a (training) data set T containing k classes is (Non equally probable messages need to be weighted by probability (frequency).)

$$info(T) = -\sum_{j=1}^{k} \frac{freq(C_j, T)}{|T|} \times log_2 \left(\frac{freq(C_j, T)}{|T|} \right) \quad bits$$

- Assume we have example set T and are making a decision to split.

- Current information is $info(T)$

- After splitting on attribute X with k values the information will be

$$info_X(T) = \sum_{i=1}^{k} \frac{|T_i|}{|T|} \times info(T_i)$$

$$gain(X) = info(T) - info_X(T)$$

- gain criterion – select attribute split to maximize information gain (mutual information)

- Used by ID3

INFORMATION RATIO

- Gain criterion is biased toward attributes with many values
  - If each example has a unique record number $info_X(T)$ will be 0 so $gain(X)$ will be maximum
  - This split is useless
- Put in a normalizing factor for this bias

$$split\ info(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times log_2 \left(\frac{|T_i|}{|T|} \right)$$

$$gain\ ratio = \frac{gain(X)}{split\ info(X)}$$

- Choose attribute split that maximizes this ratio

PREVENTING OVERFITTING

- How to get most accurate tree (on WITHHELD data)?

- Prevent the tree from becoming too complex

- Pre-pruning or Stopping

  - Only split if the information gain will be significant. ID3 used (χ^2) to test for significance

  - Not as good in practice as post pruning

- Post pruning

  - Build complete tree first

  - Make the tree less complex and hopefully more general by

  - replacing subtrees with leaves

  - replacing subtrees with most common branch

PRUNING OF DECISION TREES

- If a branch is replaced by a leaf, the leaf will correspond to several classes. Its label will be the most common class.

- Error on training set will increase

- Outline of the approach

  1. Assume that one can (magically) predict error rate of (sub) tree and leaf
  2. Start from bottom, examine each non leaf subtree
  3. if predicted error would be lower if this sub tree was a leaf then replace with a leaf
  4. if predicted error would be lower if this sub tree replaced with most common branch, then replace

PREDICTING ERROR RATE OF A SUBTREE

- Can't use error rate on training set directly

- Approach 1. Apply tree to a separate set of cases, validation set [Weka Reduced Error Pruning]
  - Break the original training set randomly into
  - Actual training set
  - Validation set

- Approach 2. Use some smart statistical theory
  1. We know that training set error will always be lower than test set error
  2. Build a statistical model of this difference
  3. Use the model to adjust (raise) the error on training set for pruning.
  4. C4.5/J48 Pruning confidence

CONTINUOUS ATTRIBUTES

- Much more straight forward than it might appear
- Sort on values of attribute A, get $v_1, v_2 ... v_m$
- Can select threshold:

$$\frac{v_i + v_{i+1}}{2}$$

- $m - 1$ possible splits to examine
- C4.5 selects largest value of A that does not exceed midpoint
- Decision trees and rules have numbers actually in the data

RUNNING J48

- Important Parameters
  - Minimum number of objects. Do not perform a split if the node has fewer than this number of examples.
  - Pruning Confidence. Smaller values cause more heavy pruning. Adjust if error rate on pruned trees for test cases is much higher than estimated error rate.
  - Reduced error pruning
- Output
  - Visualise the tree
  - Visualize the errors

OTHER INDUCTION PROGRAMS

- There are dozens of other programs for generating decision trees from data based on different algorithms.

- There are also many programs for generating classification rules directly from data.

- Decision trees or rules are often preferred for data mining over nearest neighbour and neural network methods because they can:
  - Give insight into the data
  - Be used to explain how a decision (eg fraud/not fraud) has been made
  - What do you conclude if
    - If the accuracy of OneR and J48 are very similar, eg 94% vs 98%?
    - If the accuracy of OneR and J48 are far apart, eg 64% vs 90%?

KEY INSIGHT ABOUT CLASSIFICATION

1. Prepare the data, generate arff file

2. Choose classifier

   - Experiment with parameter settings
   - Want high test/cross validation accuracy
   - Want no overfitting

3. Evaluate with error rate or mean absolute error

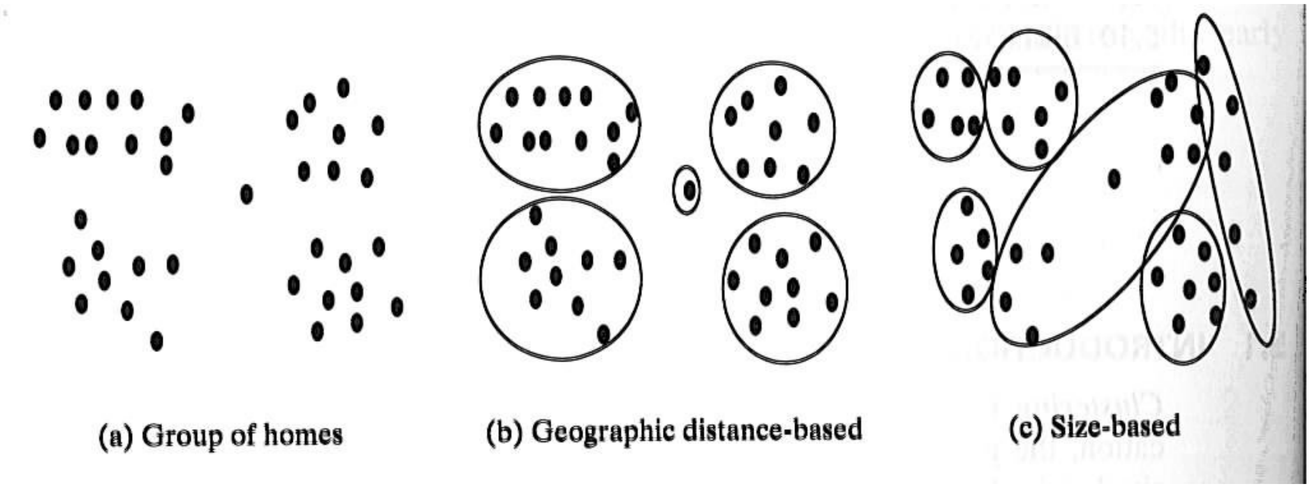
- Red Tasks that need to be done no matter what the classifier

- At step 2 you can find a good classifier without understanding all the details of all the parameters

- After a classifier is chosen for deployment you need to get a good understanding of the algorithm and parameters

CLUSTERING

- A cluster is a set of examples that are similar to each other or ‘close together’.
  - People who have similar purchasing behaviour
  - People who have similar web browsing behaviour
  - People who have similar phone habits
- A clustering algorithm partitions the examples into mutually exclusive clusters [For the purposes of this course].
- Each example is assigned to one cluster.
- Big Issue: How to you measure ‘similar‘ or ‘close‘
- Ideally we want
  - Examples in a cluster to be close together
  - Clusters to be far apart



(a) Group of homes

(b) Geographic distance-based

(c) Size-based

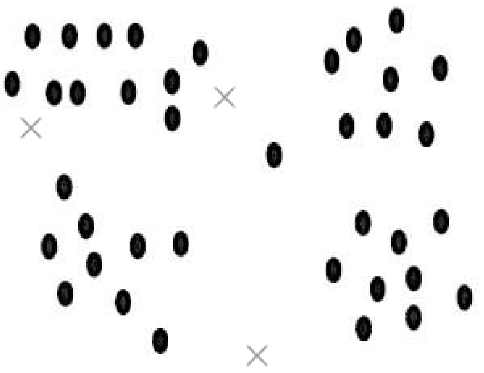
- Alternative clusterings
  - Size of house
  - Size of yard
  - Number of rooms
  - Valuation
- Meaning of “Close”

KMEANS ALGORITHM

- Usually numeric data, but can extend to symbolic

```
User supplies K, number of clusters
Randomly pick K  cluster centres Ci

Repeat until no points change clusters
  For each training example
    Determine distance to each centre
    Assign to closest Ci
  Compute cluster centres
```

- In practice
  - Try several times to see if results are consistent
  - es of K
  - me of the clusters will be very
    e manually combined



CLUSTERING EXAMPLE 1, K-MEANS K=2

- Consider the following loans data from a bank

```
@relation loans
@attribute salaryx1000 real
@attribute loanamtx1000 real
@data
24.4,5.6
24.5,4.2
.
.
```

- How many clusters are in the data?

- What are they?

- For K-Means we need to guess the number of clusters

CLUSTERING EXAMPLE 1, K-MEANS K=2

```
=== Run information ===
Scheme:       weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R fir
Relation:     loans
Instances:    100
Attributes:   2
              salaryx1000
              loanamtx1000
Test mode:    evaluate on training data

=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 1.617422991272118
Missing values globally replaced with mean/mode

Cluster centroids:
                            Cluster
Attribute       Full Data          0          1
                    (100)       (58)       (42)
===============================================
salaryx1000        35.363     24.881    49.8381
loanamtx1000        6.191     4.9828     7.8595


Clustered Instances

0       58 ( 58%)
1       42 ( 42%)
```

CLUSTERING EXAMPLE 2

/KDrive/........../DataMining/data/arff/video-small.arff

- Consider the following data from Netflix

```
@relation video-rental
@attribute Sex {f,m}
@attribute Student {n,y}
@attribute MovieType {action,horror,romance}
@data
f,n,romance
f,y,romance
m,y,horror
m,n,action
```

- Can we identify groups of people with the same viewing pattern?

- Run through Weka EM clusterer and analyse output

```
java weka.clusterers.EM -t video-small.arff
```

CLUSTERING EXAMPLE 2

```
=== Run information ===
Scheme:weka.clusterers.EM -I 100 -N 2 -M 1.0E-6 -S 100
Relation:     cluster1.csv
Instances:200
Attributes:3
              Sex
              Student
              MovieType
Test mode:evaluate on training data

=== Model and evaluation on training set ===
EM
==
Number of clusters: 2

             Cluster
Attribute          0        1
              (0.52)   (0.48)
==============================
Sex
  f             3.7556  91.2444
  m           101.4831   7.5169
  [total]     105.2387  98.7613
Student
  n            51.3565  47.6435
  y            53.8822  51.1178
  [total]     105.2387  98.7613
MovieType
  action       84.8056   3.1944
  horror       13.8633   8.1367
  romance       7.5699  88.4301
  [total]     106.2387  99.7613
Clustered Instances

0      111 ( 56%)
1       89 ( 45%)

Log likelihood: -1.96854
```

CLUSTERING EXAMPLE 2

- EM found 2 clusters

- 56% of the cases are assigned to cluster 0

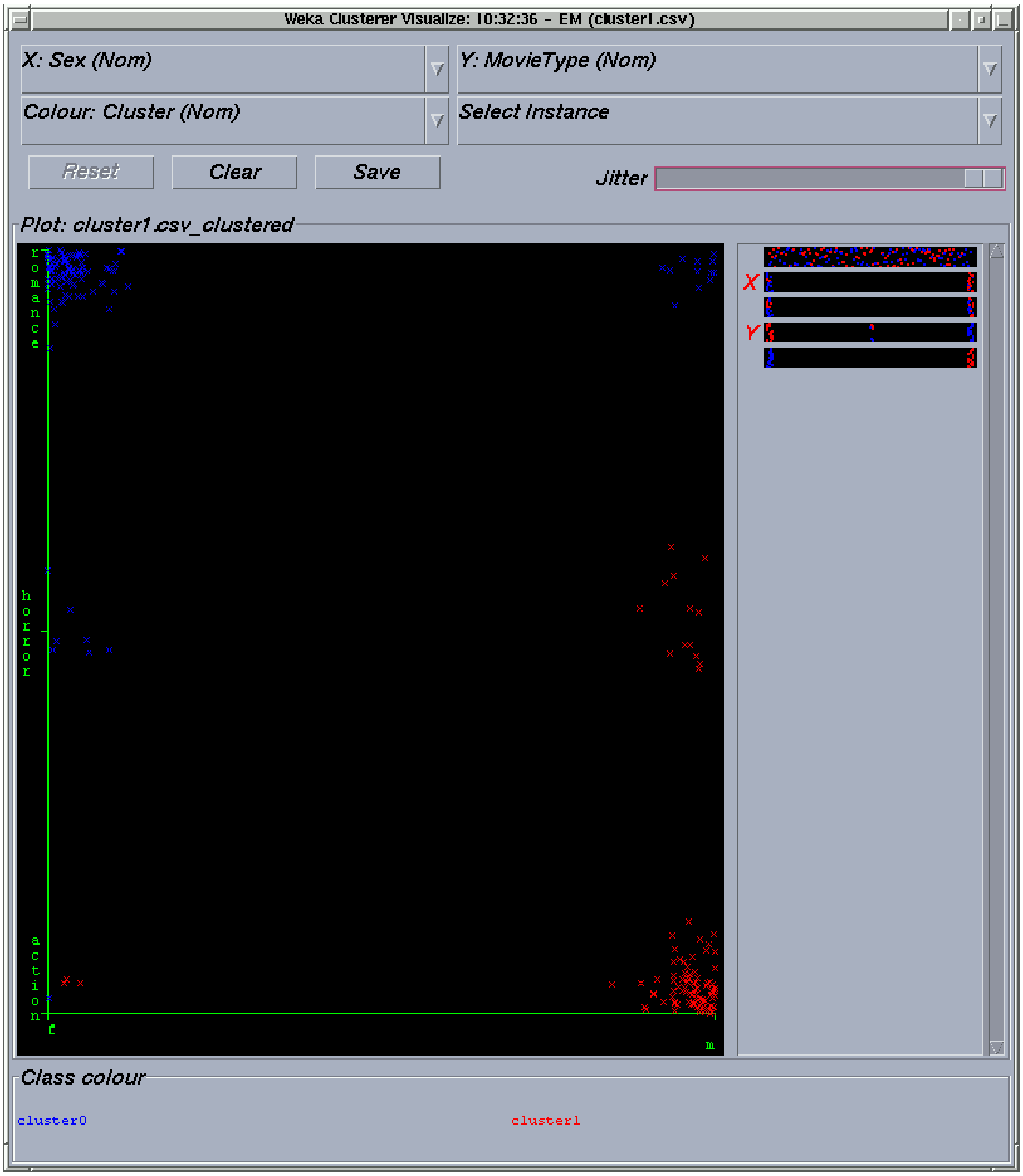
- Cluster 0 consists mainly of males who like action movies
  Whether or not they are students is not relevant.

- Cluster 1 consists mainly of females who rent romances.
  Whether or not they are students is not relevant.

- When a new romance movie is released the company can save money by sending advertising material only to cluster 1.

VISUALIZING CLUSTERS

Large jitter means that points that would be on top of one another are spread out



Weka Clusterer Visualize: 10:32:36 - EM (cluster1.csv)

X: Sex (Nom)

Y: MovieType (Nom)

Colour: Cluster (Nom)

CLUSTERING OF NUMERIC DATA

/KDrive/............/DataMining/data/arff/loans.arff

- Consider the following loans data from a bank

```
@relation loans
@attribute salaryx1000 real
@attribute loanamtx1000 real
@data
24.4,5.6
24.5,4.2
.
.
```

- How many clusters are in the data?

- What are they?

CLUSTERING OF NUMERIC DATA

- The EM program gives the output:

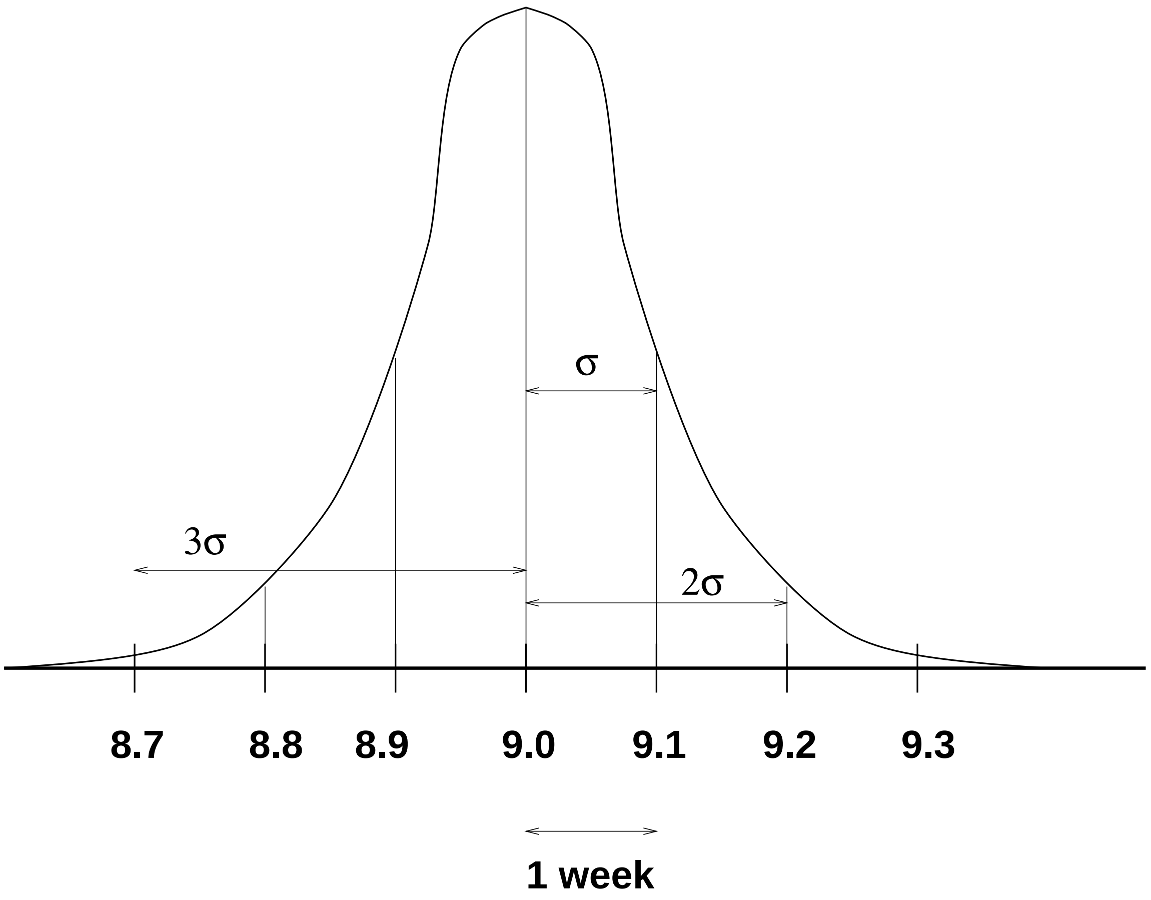
```
=== Run information ===

Scheme:weka.clusterers.EM -I 100 -N 2 -M 1.0E-6 -S 100
Relation:     loans
Instances:2000
Attributes:2
              salaryx1000
              loanamtx1000
Test mode:evaluate on training data

=== Model and evaluation on training set ===

EM
==
Number of clusters: 2

               Cluster
Attribute            0       1
                 (0.5)   (0.5)
===============================
salaryx1000
  mean         49.9507 24.9507
  std. dev.     1.0794  1.0794

loanamtx1000
  mean            7.97  4.9719
  std. dev.      2.295  1.0834

Clustered Instances

0      1000 ( 50%)
1      1000 ( 50%)

Log likelihood: -4.06283
```

- To interpret the output we need to know some properties of the normal distribution

THE NORMAL (GAUSSIAN) DISTRIBUTION

Most naturally occuring data, if enough of it is collected, fits the normal distribution.

Example: Length of a human pregnancy



Parameter	Symbol	Example	Estimator
mean	μ	9 months	average
variance	σ	1 week	standard deviation

THE NORMAL DISTRIBUTION

1. 68% of all observations fall within 1 standard deviation of the mean

2. 95% of all observations fall within 2 standard deviations of the mean

3. 99% of all observations fall within 3 standard deviation of the mean

4. 68% of all babies are born ± 1 week of 9 months

5. 99% of all babies are born ± 3 weeks of 9 months

Equation of the ‘bell’ curve:

$$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$$

AVERAGE AND STANDARD DEVIATION

- The average gives the central point of the data

  $Average = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

- The standard deviation is a measure of the spread.

  $StandardDeviation(SD) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$

- A small standard deviation means that most of the data points are close to the mean

  This is highly desirable for data mining.

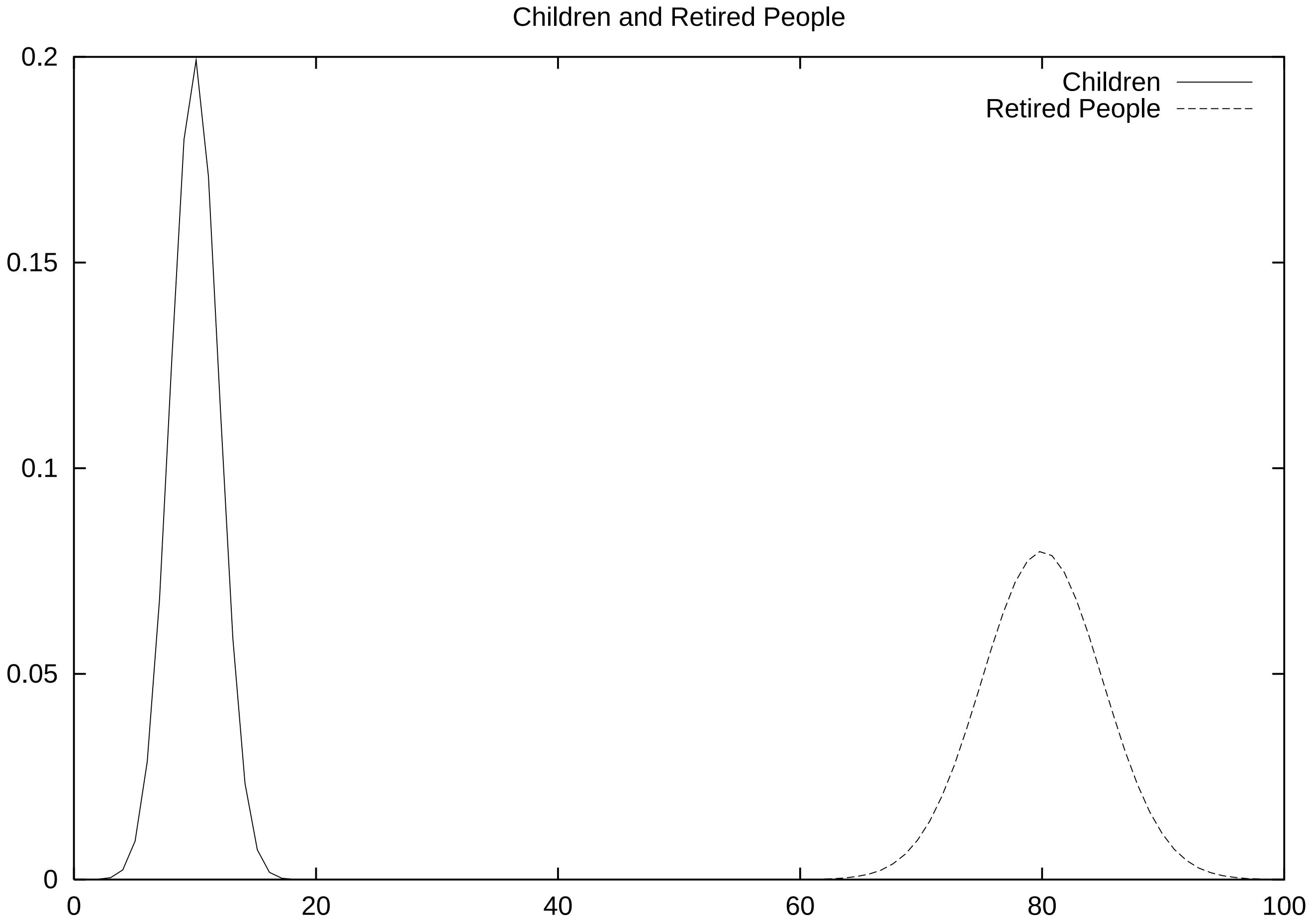
- A large standard deviation means that there are many data points a long way away from the mean

CLUSTERS

- Suppose we have a group of children whose average age is 10 and standard deviation is 2.

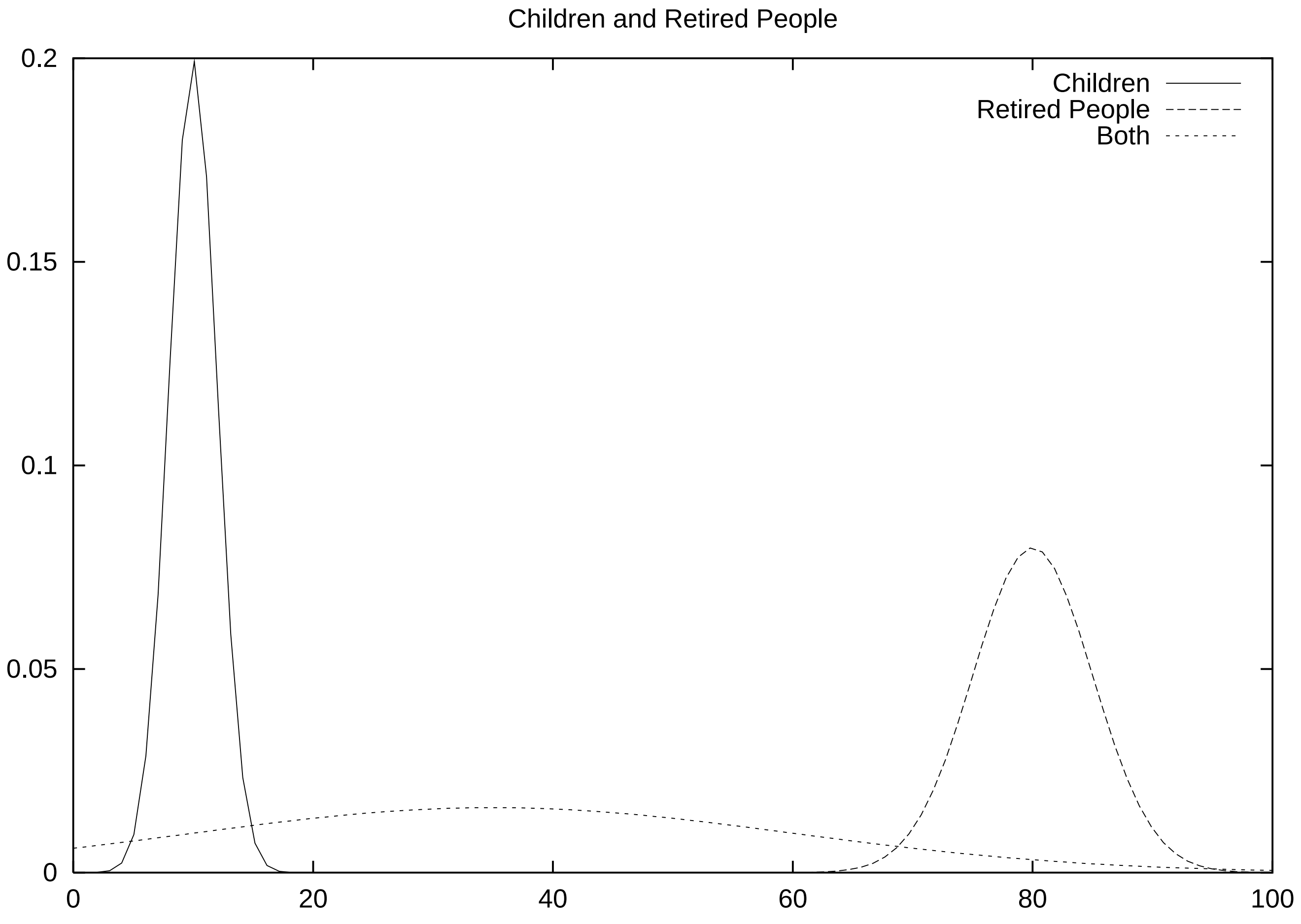
- Suppose we have a group of retirees whose average age is 80 and standard deviation is 5.

- Plotting the distributions on the same axes gives:



CLUSTERS

- Now suppose that we put both children and retirees into the same file and get the mean and standard deviation.

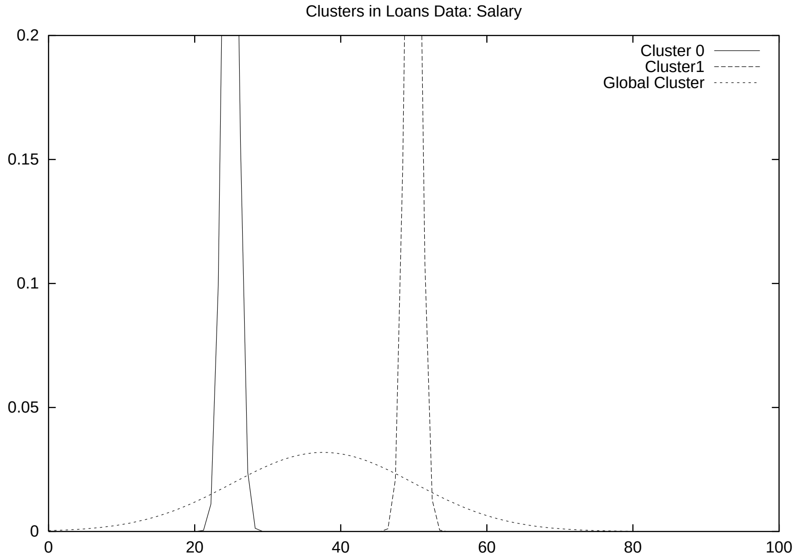


- The distribution for both groups will be wide and flat

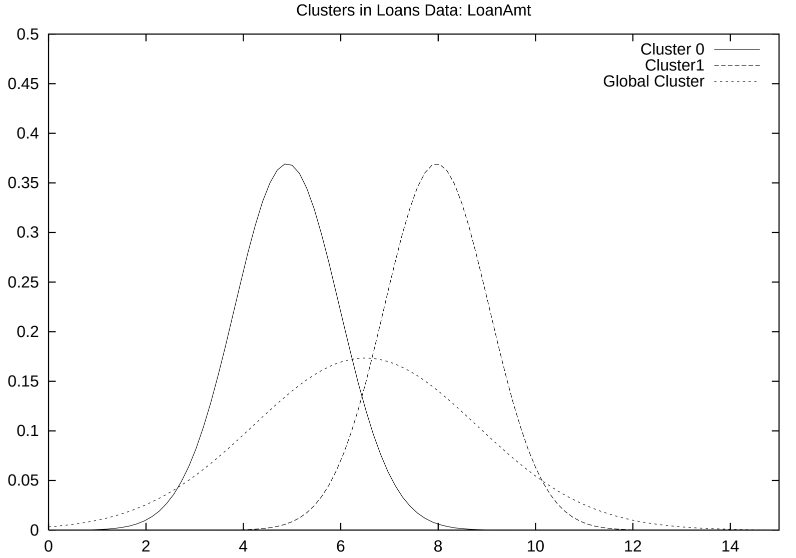
- To find good clusters we need to find groups that are well separated from the global cluster

CLUSTERS IN LOANS DATA

- Plotting salary in loans data gives:



- Plotting Loan Amount gives:



CLUSTERS IN LOANS DATA

- The two clusters are very strongly separated based on salary (No Overlap)

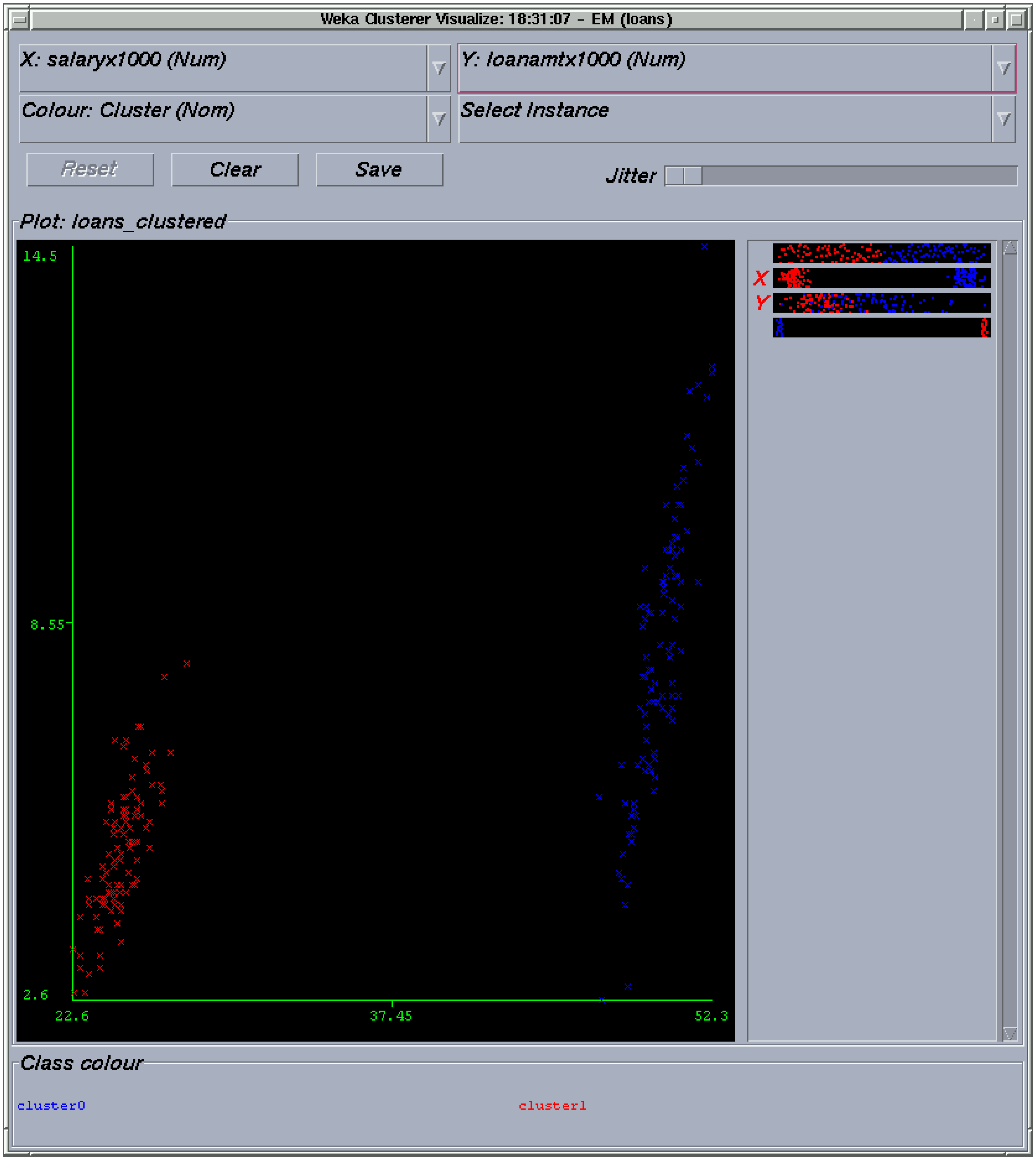
- The two clusters are reasonably well separated on loan amount (A small amount of overlap)

- To a bank, cluster 0 could be medium earners with a medium loan

- To a bank, cluster 1 could be low earners with a small loan

- It's up to the bank what to do with this information

VISUALIZATION OF CLUSTERS



KMEANS ALGORITHM

- Usually numeric attributes

```
User supplies K, number of clusters

Randomly pick K  cluster centres Ci

Repeat until no points change clusters
  For each training example
    Determine distance to each centre
    Assign to closest Ci
  Compute cluster centres
```

- In practice
  - Try several times to see if results are consistent
  - Try different values of K
  - If K is too big some of the clusters will be very similar and can be manually combined