# Associative Classifiers
# Evaluation Measures
# Classification with Rare Classes

Luiza Antonie
PhD Candidate

## Outline

- What are associative classifiers?
- How do you use them?
- How can you evaluate and compare multiple classification systems?
- What measures are better?
- How do classifiers perform when there are rare classes?



## Association Rules (typical)

- Association rule mining aims at discovering associations between items in a transactional database.
- Given D={T1…Tn} a set of transactions and I={i1…in} a set of items such that any Ti in D is a set of items in I.
- An association rule is an implication A➔B where A and B are subsets of Ti given some support and confidence thresholds.
- The **support** of the rule is the probability that A and B hold together among all the possible presented cases.
- The **confidence** of the rule is the conditional probability that the consequent B is true under the condition of the antecedent A.



## Association Rules for Classification

- Association Rules (typical)
  - $F_\alpha \wedge F_\beta \wedge F_\gamma \wedge \ldots \wedge F_\delta$ ➔ **class**
- Negative Association Rules
  - $F_\alpha \wedge$ not $F_\beta \wedge$ not $F_\gamma \wedge \ldots \wedge F_\delta$ ➔ **class**
  - $F_\alpha \wedge F_\beta \wedge$ not $F_\gamma \wedge \ldots \wedge F_\delta$ ➔ not **class**
- Weighted Association Rules
  - $aF_\alpha \wedge bF_\beta \wedge cF_\gamma \wedge \ldots \wedge dF_\delta$ ➔ **class**

## Associative Classifier

- Rule discovery
  - Using an ARM algorithm
- Pruning
  - Discarding those rules that are redundant or not interesting
- Classification
  - Based on a scoring scheme, use the set of rules to classify new, unseen instances



## Rule Discovery (1st stage)

- Given a training set a set of association rules is discovered using an ARM algorithm:
  - Apriori, FP-tree, etc.
  - Modify the algorithms to mine the form of rules that you want;
  - Mine all the association rules and filter them afterwards;
- CBA, ARC-AC and ARC-BC
  - Apriori based;
- CMAR
  - FP-tree;



## Pruning (2nd stage)

Large number of rules ⇨ Noisy information

Large number of rules ⇨ Long classification time

- Removing low ranked specialized rules;

$R_1 : F_1 \Rightarrow C$ Confidence 90%

$R_2 : F_1 \wedge F_2 \Rightarrow C$ Confidence 80%

$\Rightarrow R_1$

- Eliminate conflicting rules;

$F_1 \Rightarrow C_1 \wedge F_1 \Rightarrow C_2$

- Database coverage;



## Pruning (2nd stage)

- Database Coverage
  - select a small set of high quality rules
  - A set of rules (SR)
    - The rules are ordered by confidence and support
  - for each rule (R)
    - if R classifies correctly at least one example, keep R
    - remove the examples covered by R
  - stop when there are no more examples or all the rules have been checked

# Classification (3rd stage)

- A set of rules (SR)
  - The rules are ordered by confidence and support
- A new instance to be classified
- From SR a subset of rules SR' matches the new instance
  - Divide SR' in subsets based on the class label
  - SR'$C_1$, SR'$C_2$, …. SR'$C_n$



# Classification (3rd stage)

- Different strategies
  - CBA
    - Choose the first matching rule (highest confidence)
  - CMAR
    - For each SR'$C_1$, SR'$C_2$, …. SR'$C_n$ set
    - Computes a weighted chi-square
    - Chooses the class with the best score (best chi-square)
  - ARC-AC and ARC-BC
    - For each SR'$C_1$, SR'$C_2$, …. SR'$C_n$ set
    - Computes the average of the confidences
    - Chooses the class with the best score (best average confidence)



# Classification Stage for ARC

Let S be the classification system

A new object O <f1; f3; f4; f7; f9 >

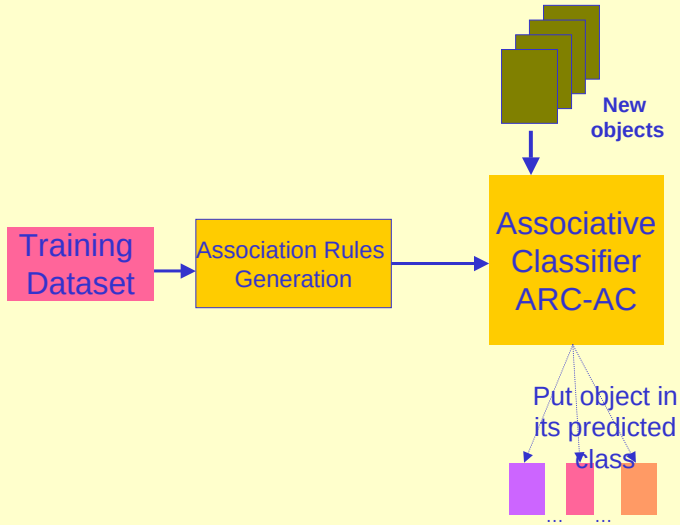
f1 => C1 confidence 0.9
f3 & f4 => C2 confidence 0.85
f4 => C2 confidence 0.8
f7 => C1 confidence 0.6
f9 => C3 confidence 0.5

**C2** 0.825
**C1** 0.75
**C3** 0.5

Using the *dominance factor* we chose the winning categories. If δ=100% C2 is winning. If δ=80% O is predicted to fall in C2 and C1.



# Association Rules Classification with All Categories



# ARC-AC

| Trans ID | Class Label | Attributes |
|---|---|---|
| 1 | C1 | A B |
| 2 | C1 | A C |
| 3 | C2 | B D E |
| 4 | C2 | B C E |
| 5 | C2 | B C D E |
| 6 | C2 | B D |
| 7 | C2 | C D E |
| 8 | C3 | C F |

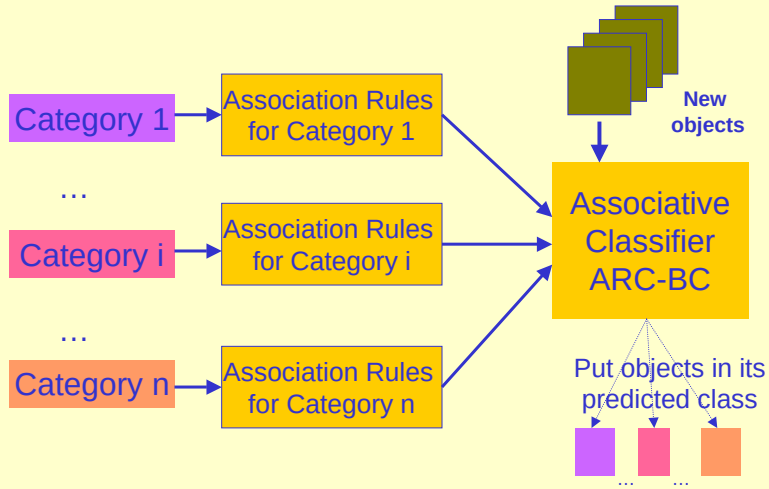
| 1-itemset | support | possible correlations between the 1-itemset and a class label |
|---|---|---|
| A | 2 | $A \Rightarrow C1$ |
| B | 5 | $B \Rightarrow C1$ |
| | | $B \Rightarrow C2$ |
| C | 5 | $C \Rightarrow C1$ |
| | | $C \Rightarrow C2$ |
| | | $C \Rightarrow C3$ |
| D | 4 | $D \Rightarrow C2$ |
| E | 4 | $E \Rightarrow C2$ |
| F | 1 | $F \Rightarrow C3$ |

supp=1/8
conf=100%

supp=1/8
conf=20%



# Association Rules Classification by Category






# ARC-BC

| Trans ID | Class Label | Attributes |
|---|---|---|
| 1 | C1 | A B |
| 2 | C1 | A C |

| Trans ID | Class Label | Attributes |
|---|---|---|
| 3 | C2 | B D E |
| 4 | C2 | B C E |
| 5 | C2 | B C D E |
| 6 | C2 | B D |
| 7 | C2 | C D E |

| Trans ID | Class Label | Attributes |
|---|---|---|
| 8 | C3 | C F |

A => C1
B => C1
C => C1

B => C2
C => C2
D => C2
E => C2

C => C3
F => C3

Supp 50%

A => C1
B => C2
C => C2
D => C2
E => C2
C => C3
F => C3

supp=1/1
conf=100%



# Evaluation

- Why do we do it?
  - To study the performance of the classification systems and to compare with other algorithms
- How do we do it?
  - Accuracy/error
  - Precision, Recall, F-measure
  - Graphical methods
    - ROC
    - Lift curves, PN curves, cost curves

# 2-class Confusion Matrix

| True class | Predicted class | |
|---|---|---|
| | positive | negative |
| positive (P) | TP | FN=P - TP |
| negative (N) | FP | TN=N - FP |

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$error = \frac{FP + FN}{TP + FP + TN + FN}$$



# 2-class Confusion Matrix

| True class | Predicted class | |
|---|---|---|
| | positive | negative |
| positive (P) | TP | FN=P - TP |
| negative (N) | FP | TN=N - FP |

$$precision = \frac{TP}{TP + FN}$$

$$F - measure = \frac{(1 + \beta^2) * P * R}{P + \beta^2 * R}$$

$$recall = \frac{TP}{TP + FP}$$

$$F_1 - measure = \frac{2 * P * R}{P + R}$$



# ROC curve

| True class | Predicted class | |
|---|---|---|
| | positive | negative |
| positive (P) | TP | P - TP |
| negative (N) | FP | N - FP |

- Reduce the 4 numbers to two rates
  - true positive rate = TPR = TP/P
  - false positive rate = FPR = FP/N
- Rates are independent of class ratio

Source: Rob Holte



# Example: 3 classifiers

| True | Predicted | |
|---|---|---|
| | pos | neg |
| pos | 40 | 60 |
| neg | 70 | 30 |

| True | Predicted | |
|---|---|---|
| | pos | neg |
| pos | 70 | 30 |
| neg | 50 | 50 |

| True | Predicted | |
|---|---|---|
| | pos | neg |
| pos | 60 | 40 |
| neg | 80 | 20 |

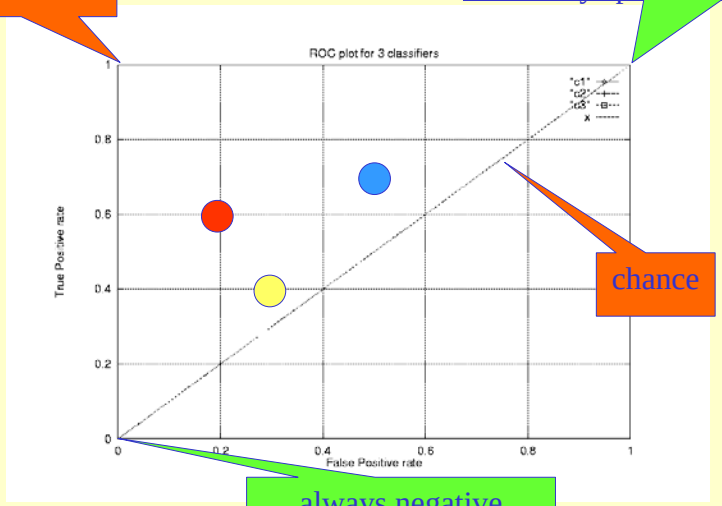
Classifier 1
TPR = 0.4
FPR = 0.3

Classifier 2
TPR = 0.7
FPR = 0.5

Classifier 3
TPR = 0.6
FPR = 0.2

Source: Rob Holte

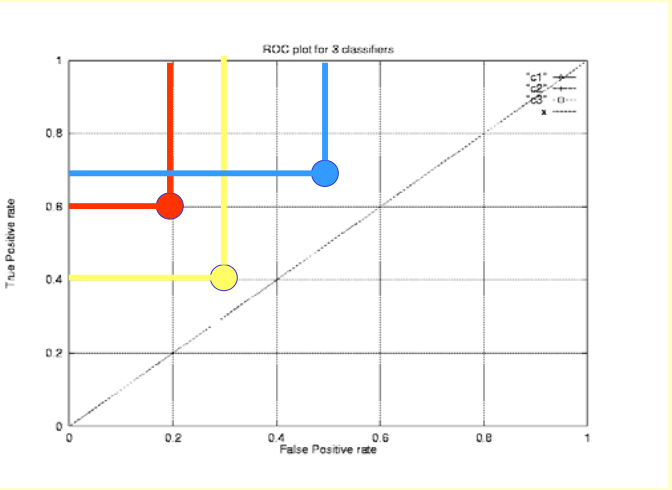
# Example: 3 Classifiers




Source: Rob Holte



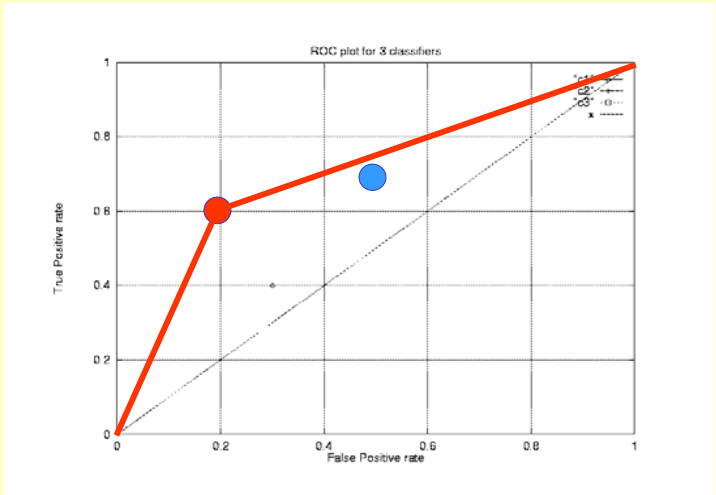
# Dominance




Source: Rob Holte



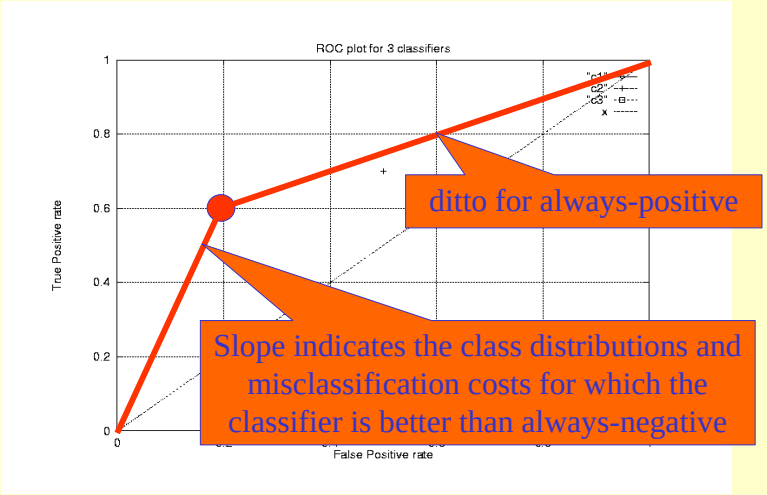
# Linear Interpolation




Source: Rob Holte

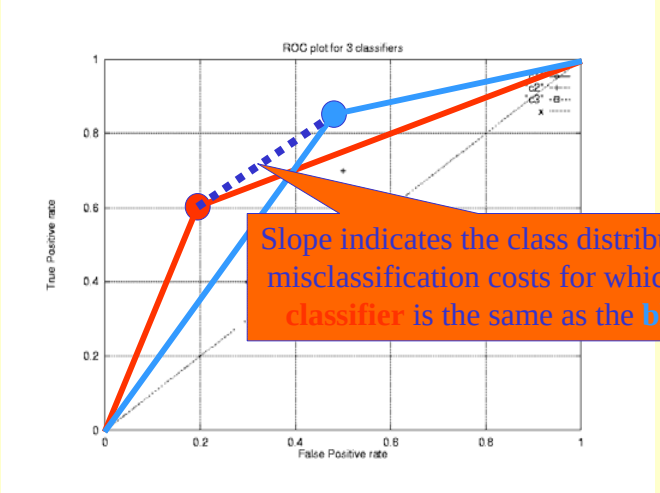


# Operating Range




Source: Rob Holte

# Combining Classifiers – Convex Hull




Source: Rob Holte



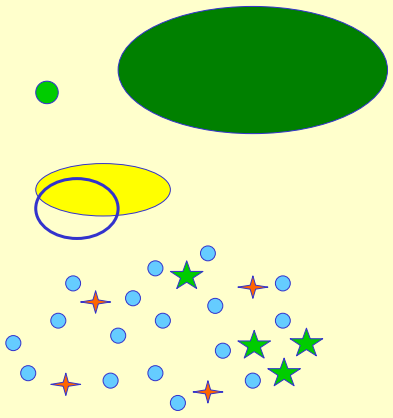
# Rare Classes

- Class imbalance - occurs when some classes have many examples, while others are represented by just a few;
- Small classes are difficult to classify for existing classification algorithms;
- Applications - medical data; text data; biological data; detection of intrusions;



# Why is it difficult?

- Class imbalance
- Class overlapping
- Noisy data
- Small disjuncts





# Methods and Solutions

- Data manipulation
  - balancing data by sampling
  - data segmentations
  - learning only the minority class
- Cost-sensitive classifiers
  - MetaCost [Domingos '99]
  - AdaCost [Fan et al. '99]

# Methods and Solutions

- Creating new algorithms
  - PN-rule [Joshi et al. '01]
    - P-rules (rules that predict the presence of the target class)
    - N-rules (rules that imply the absence of the target class)
    - two phases
      - 1st focuses on recall; completeness – retrieving the results
      - 2nd improves precision; quality – retaining only the desired examples



# Methods and Solutions

- Creating new algorithms
  - Boosting
    - AdaBoost [Schapire '99]
    - starts with a weak classifier and boosts is performance
    - iterative processes
    - At each iteration the weights that are attached to a training example are refined
    - misclassified - weight increase
    - correctly classified – weight decrease
  - ARC-BC



# Methods and Solutions

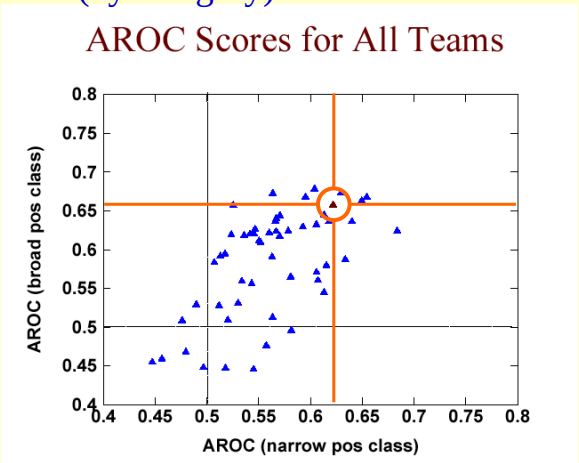
- Associative Classifier (by category)

**Yeast Gene Regulation Prediction**

•A list of test set genes (many missing values) (Instances: 127 positive; 4380 negative)

•Protein / protein interaction

•MEDLIN text abstracts describing genes of interest.






# Sampling

- Under-sampling
  - random – eliminate at random examples of the majority class
  - Tomek links [Tomek '76]
  - Condensed Nearest Neighbour Rule (CNN) [Hart '68]
  - One-sided Selection (OSS) [Kubat and Matwin '97]
  - Neighbourhood Cleaning Rule (NCL) [Laurikkala '01]
  - drawbacks – it can discard good examples

# Sampling

- Over-sampling
  - random – duplicate at random some examples belonging to the minority class;
  - Smote [Chawla '02]– create artificial examples for the minority class by interpolating between existing examples;
  - drawbacks
    - overfitting
    - execution time increase
- Under-sampling + Over-sampling



# Under-Sampling vs. Over-Sampling

- Under-sampling
  - [Drummond and Holte '03]
  - [Domingos '99]
- Over-sampling
  - [Japkowicz et al. '02] – artificial datasets
  - [Batista et al. '04]
- Is it dataset dependant?



# Discussion

- classification errors occur near class boundaries
- difficult to find good boundaries when classes are overlapping
- take into account the overlapping of classes
- application dependant



# Discussion

- develop new algorithms to deal with rare classes
- incorporate in the existing algorithms some knowledge about the classification with rare classes
- use some graphical evaluation measures to improve the classifier

# Thank You!