Multi-instance Learning

Amelia Zafra Gómez

Department of Computer Sciences and Numerical Analysis. University of Cordoba

Overview

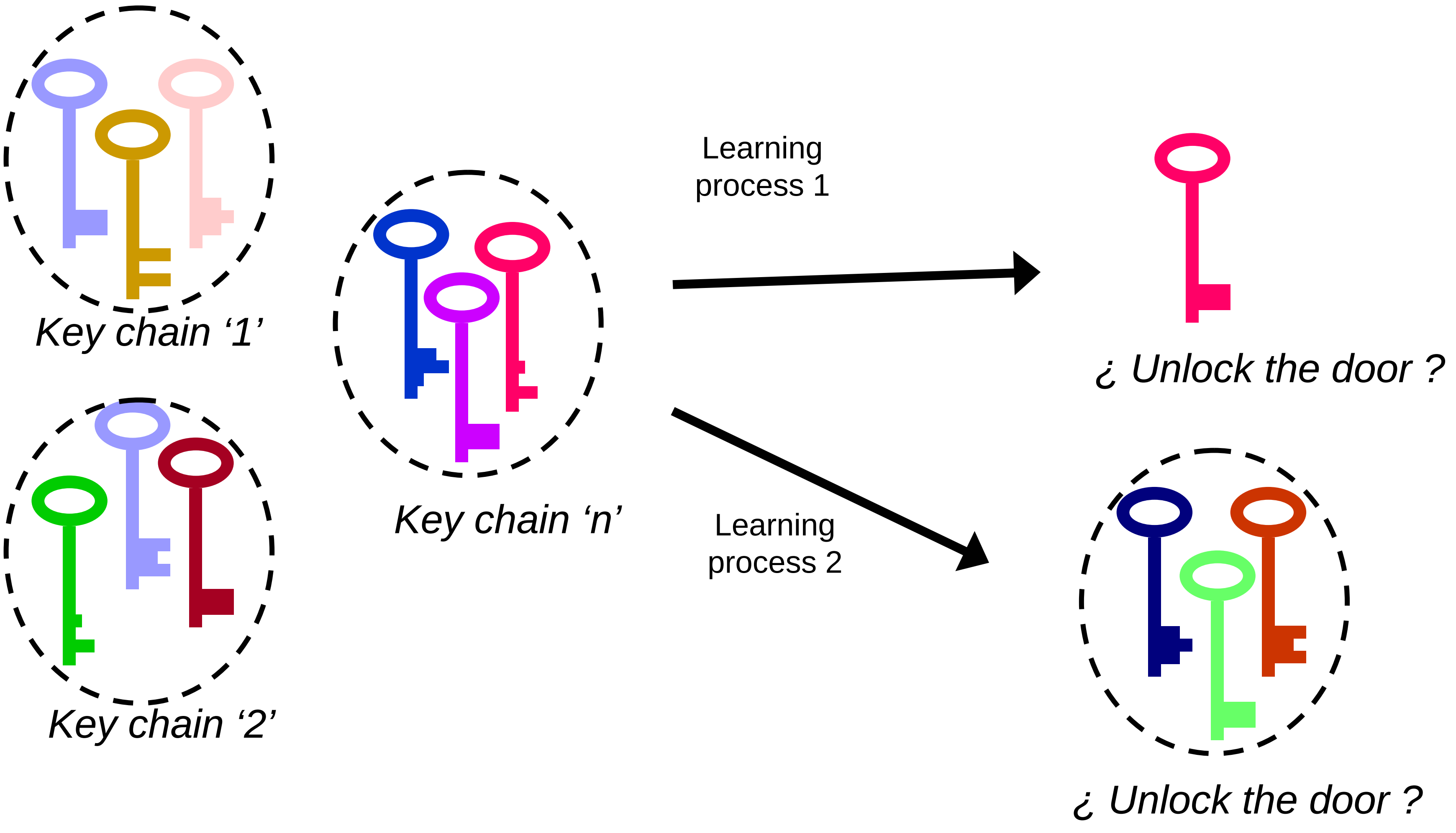
- Introduction
  - The Multiple Instance Learning (MIL) Problem
  - Supervised vs. Multi-Instance Learning
  - Definition and Notation
- Applications of Multiple Instance Learning
  - Drug activity prediction
  - Content-based image retrieval and classification
  - Text Categorization
  - Web Index Recommendation
- Review of Multiple Instance Algorithms
  - Learning Axis-Parallel Concepts (APR)
  - Diverse Density (DD)
  - Expectation-Maximization Diverse Density (EM-DD)
  - K- Nearest Neighbor Algorithms (k-NN)
  - Decision Tree
  - Rule Based Systems (SBR)
  - Support Vector Machine (SVM)
  - Neural Networks (NN)
  - Inductive Logical Programming (ILP)

Overview

- Genetic Programming for Multi-Instance Learning
  - Motivations
  - Individual Representation
  - Function Fitness
  - G3P-MI
  - MOG3P-MI
- Experiment and Results
  - Drug Activity Prediction
  - Web Index Page
- References

Introduction

The Multiple Instance Problem



The Multiple Instance Problem

- This learning appears in complex applications of machine learning where the learner has partial or incomplete knowledge about each training example.
- Each training example can be represented by means of a bag composed of one or several feature vectors.
- The learner only knows that each example can be represented by one of a set of potential feature vectors instead of knowing which particular instance or set of them represent the concept which we want to learn.

Supervised learning vs. multi-instance learning

Supervised Learning

Object ≅ Instance

I know the class of the object

Training Set

$V(v_1, v_2, \ldots v_n)$ **-**

The instance is negative

Multi-Instance Learning

Object ≅ Several Instances

I know the class of the object, but I do not know the class of each instance

Training Set

$V_1(v_{11}, v_{12}, \ldots v_{1n})$ **-**
$V_2(v_{21}, v_{22}, \ldots v_{2n})$ **-**
…
$V_p(v_{p1}, v_{p2}, \ldots v_{pn})$ **-**

The bag is negative, I know that all instances are negatives

Definition and Notation

- *M* is the set of all objects m_i *uses in the learning process*.
- The goal consist of learning a function $f(m_i)$ *which return* 0 for the negative case and 1 for positive case.
- For multi-instance learning, an object or example m_i can have several instances ν_i, that is, $m_{i,1}, m_{i,2}, \ldots, m_{i,\nu_i}$.
- Each instance is represented by means of feature vector $V(m_{i,j})$. *Thus, a training example is represented by:*

$$\langle \{V(m_{i,1}), V(m_{i,2}), \ldots, V(m_{i,\nu_i})\}, f(m_i) \rangle$$

- We can model the problem by introducing a second function g($V(m_{i,j})$) that takes a single instance and produces a result:

$$f(m_i) = \begin{cases} 1, & \text{si } \exists j \;/\; g(V(m_{i,j})) = 1 \\ 0, & \text{otherwise} \end{cases}$$

- This definition is known as Dietterich hypothesis.

Generalized Multiple Instance Learning

- Scott et al. generalized the MIL model in 2003:
  - A bag is positive if and only if it contains a *collection* of instances are predicted to be positive according to function g(V($m_{i,j}$)).

$$f(m_i) = \begin{cases} 1, & \text{si } \exists \{\Phi \text{ of } j\} \,/\, g(V(m_{i,j})) = 1 \\ 0, & \text{otherwise} \end{cases}$$

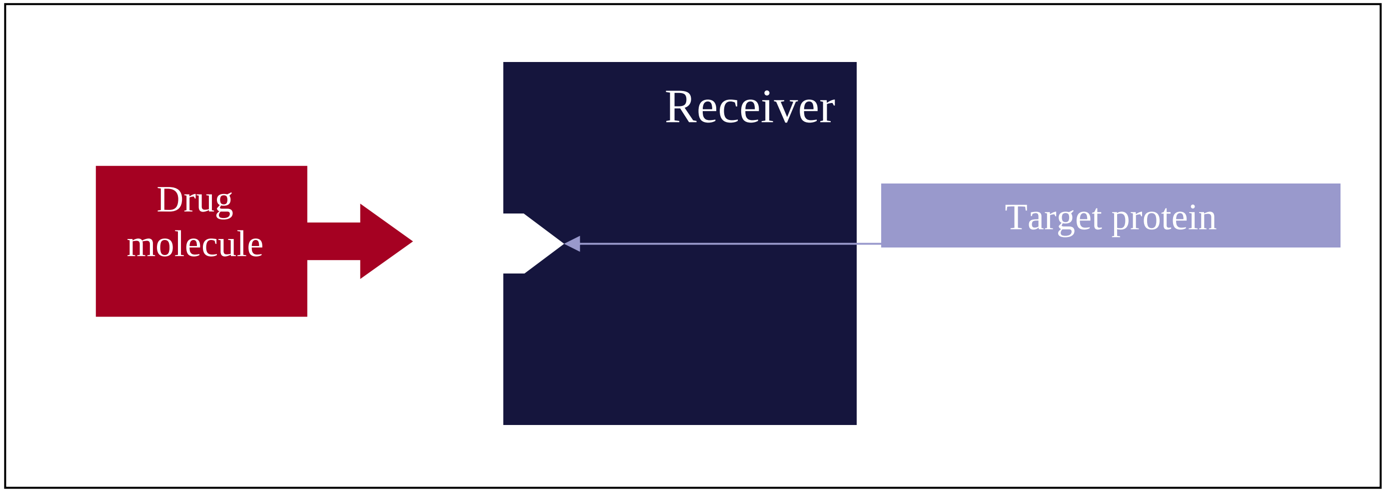
- This generalized model is much more expressive than the conventional multiple instance model and show significant advantages over the conventional MIL model on certain application areas.

S. Scott and J. Zhang and J. Brown. On Generalized Multiple-Instance Learning. International Journal of Computational Intelligence and Applications, 5 pp.21-35. 2005*. (In 2003, they publish a Technical Report)*

Applications of Multiple Instance Learning

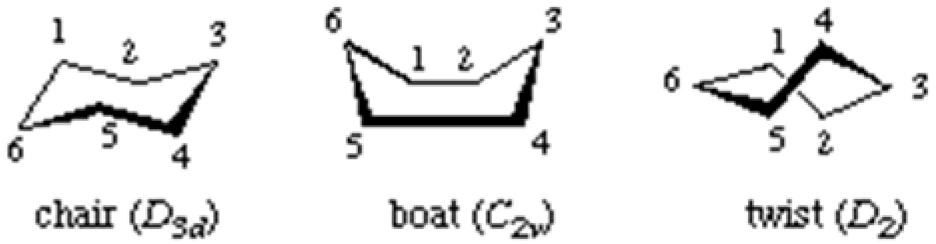
Drug activity prediction

- The first work in MIL was motivated by the problem of determining whether a drug molecule will bind strongly to a target protein.
- Some drug molecules bind well (they are positive examples) and some do not bind well (they are negative ones).
- This problem and his datasets have been extensively used as benchmark in evaluating and comparing MIL methods.



Drug activity prediction

- A molecule may adopt a wide range of shapes or conformation.



  - A positive molecule has at least one shape that can bind well -- but we do not know which one.
  - A negative molecule means none of its shapes can make the molecule bind well.

- This problem fits in the MIL setting as a very natural way
  - Each molecule as a bag.
  - The shapes it can adopt as the instances in that bag. The features of an instance (shape) are the distances from an origin to different positions on the molecule surface at the corresponding shape.

Content-based image retrieval and classification

- The key to the success of image retrieval and image classification is the ability of identifying the intended target object(s) in images.

- This is made more complicated by the fact that an image may contain multiple and possibly heterogeneous objects. Thus, the global description of a *whole* image is too coarse to achieve good classification and retrieval accuracy → it is a hard problem in the supervised learning setting.

- This problem can fit in the MIL setting well:
  - Each image itself is considered as a bag.
  - A region or segment in an image is considered as an instance.

- A bag can have two kinds of labels - *positive* and *negative.*

- Given many bags (images) of instances (regions) with labels on each of them, our goal is to classify new bag.

Text Categorization

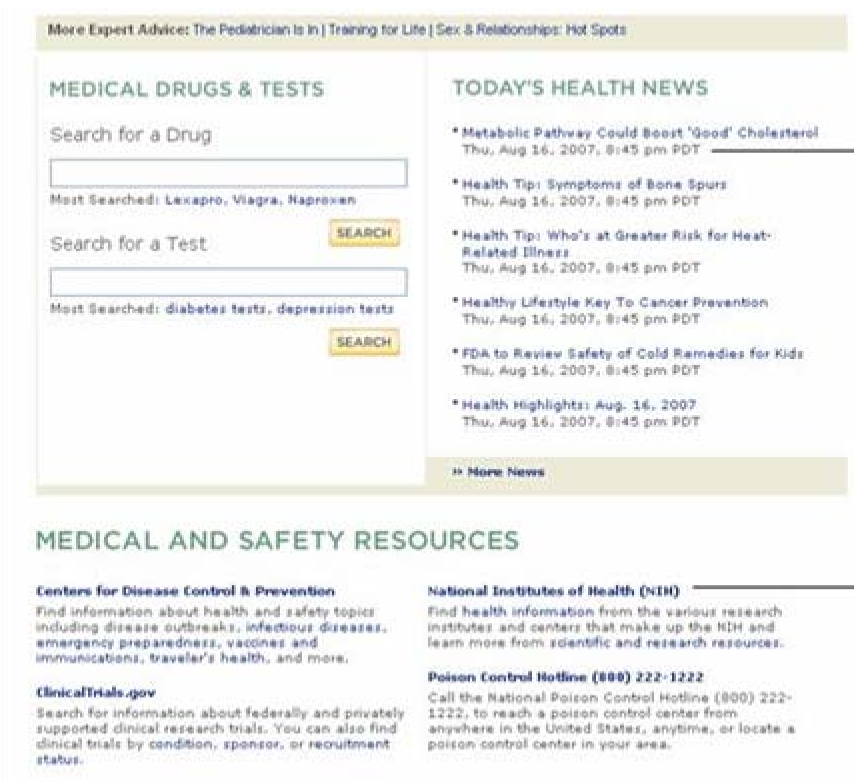
- Similar to the argument made on images, a text document can consist of multiple passages that are of different topics, and thus descriptions at the document level might be too rough → it is a hard problem in the supervised learning setting.

- This problem can fit in the MIL setting well:
  - Each document is represented as a bag.
  - Each passages (with a fixed number of words) of each document is represented as instances of a bag.

- A dataset used to evaluate the algorithm has been TREC9 document categorization sets, unfortunately without comparisons with traditional text categorization methods.

Web Index Recommendation

- *Web index pages* are pages that provide titles or brief summaries and leaving the detailed presentation to their linked pages. For example, the entrance of health at Yahoo!

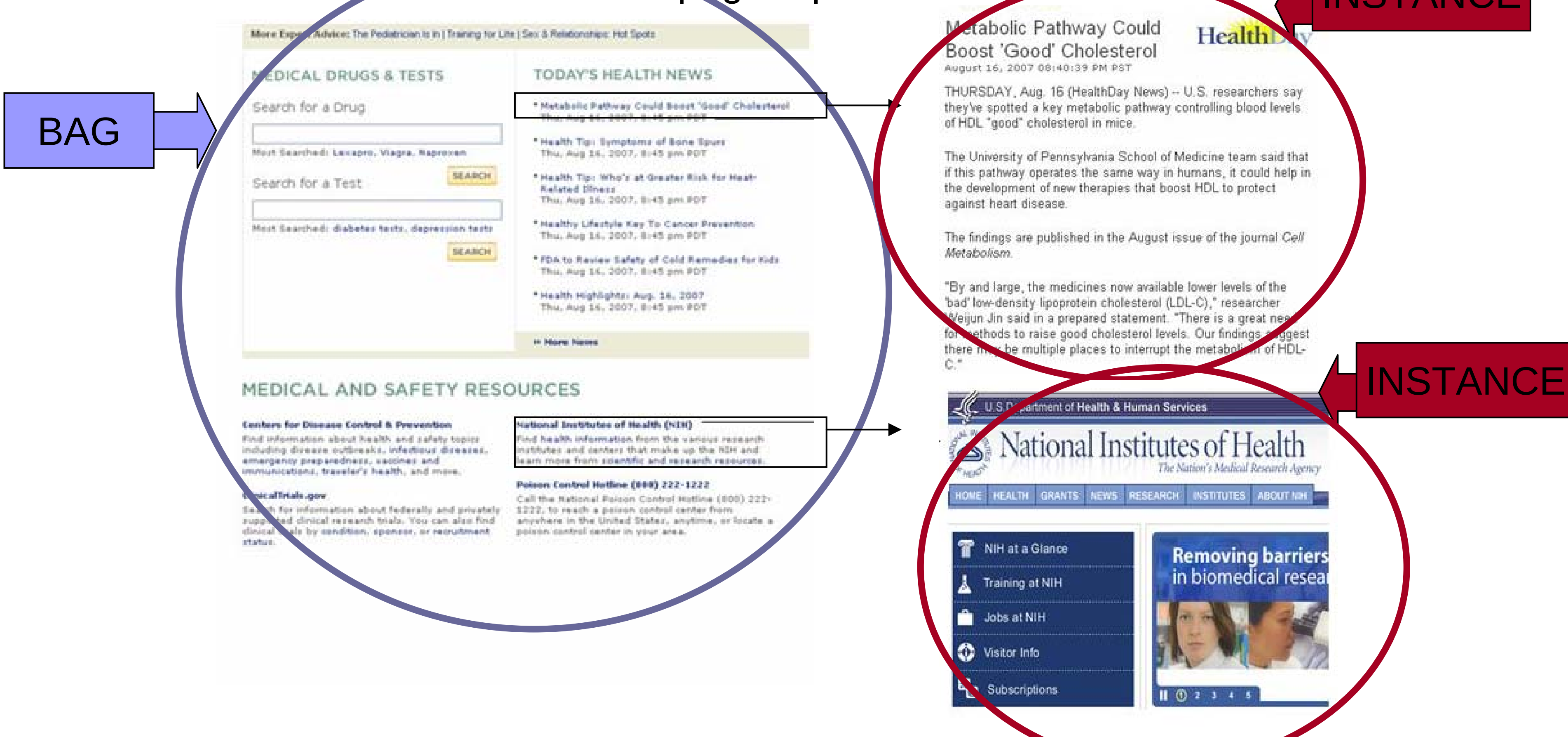


Web Index Recommendation

- This problem consists of labelling unseen web index pages as positive or negative.
  - A positive web index page is such a page that the user is interested in at least one of its linked pages.
  - A negative web index page is such a page that none of its linked pages interested the user.
- The difficulty lies in that the user only specifies whether he or she is interested in an index page, instead of specifying the concrete links that he or she is really interested in.

Web Index Recommendation

- This problem can fit in the MIL setting well:
  - Each web index page represents a bag.
  - Each link in the web index page represents a instance of the ba



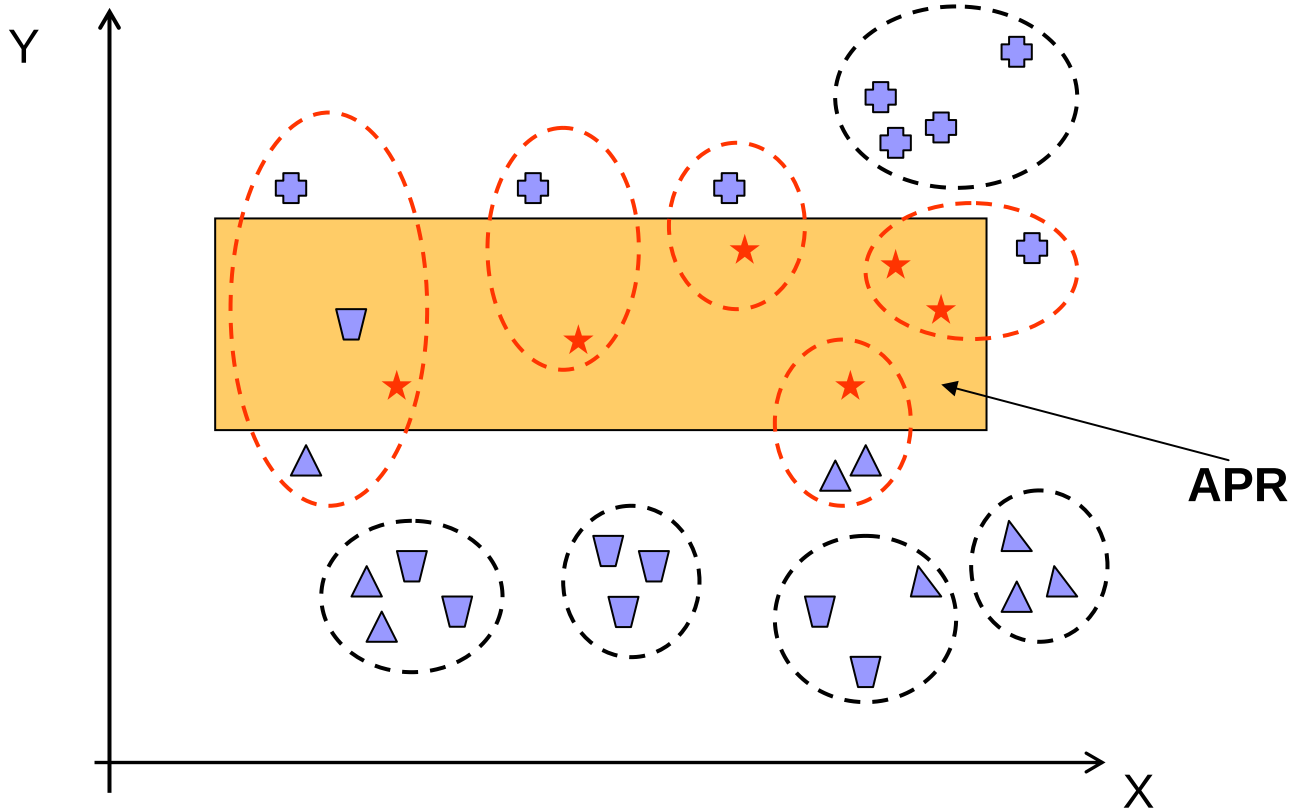
Review of Multiple Instance Algorithms

Learning Axis-Parallel Concepts

T. G. Dietterich, R.H Lathrop & T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89:1-2 (1997), pp 31-71.

- It is the first class of algorithms that were proposed to solve MIL problems.
- The idea is to find an axis-parallel hyper-rectangle (APR) in the feature space to represent the target concept. Intuitively, this APR should contain at least one instance from each positive example and meanwhile exclude all the instances from negative examples.
- They consider three general designs for APR learning algorithms:
  - **A noise-tolerant "standard" algorithm**. The naive APR algorithm just forms the smallest APR that bounds the positive examples.
  - **An "outside-in" algorithm**. This algorithm is a variation on the "standard" algorithm. It constructs the smallest APR that bounds all of the positive examples and then shrinks this APR to exclude false positives.
  - **An "inside-out" algorithm**. This algorithm starts with a seed point in feature space and "grows" a rectangle with the goal of finding the smallest rectangle that covers at least one instance of each positive example and no instances of any negative example.

Learning Axis-Parallel Concepts

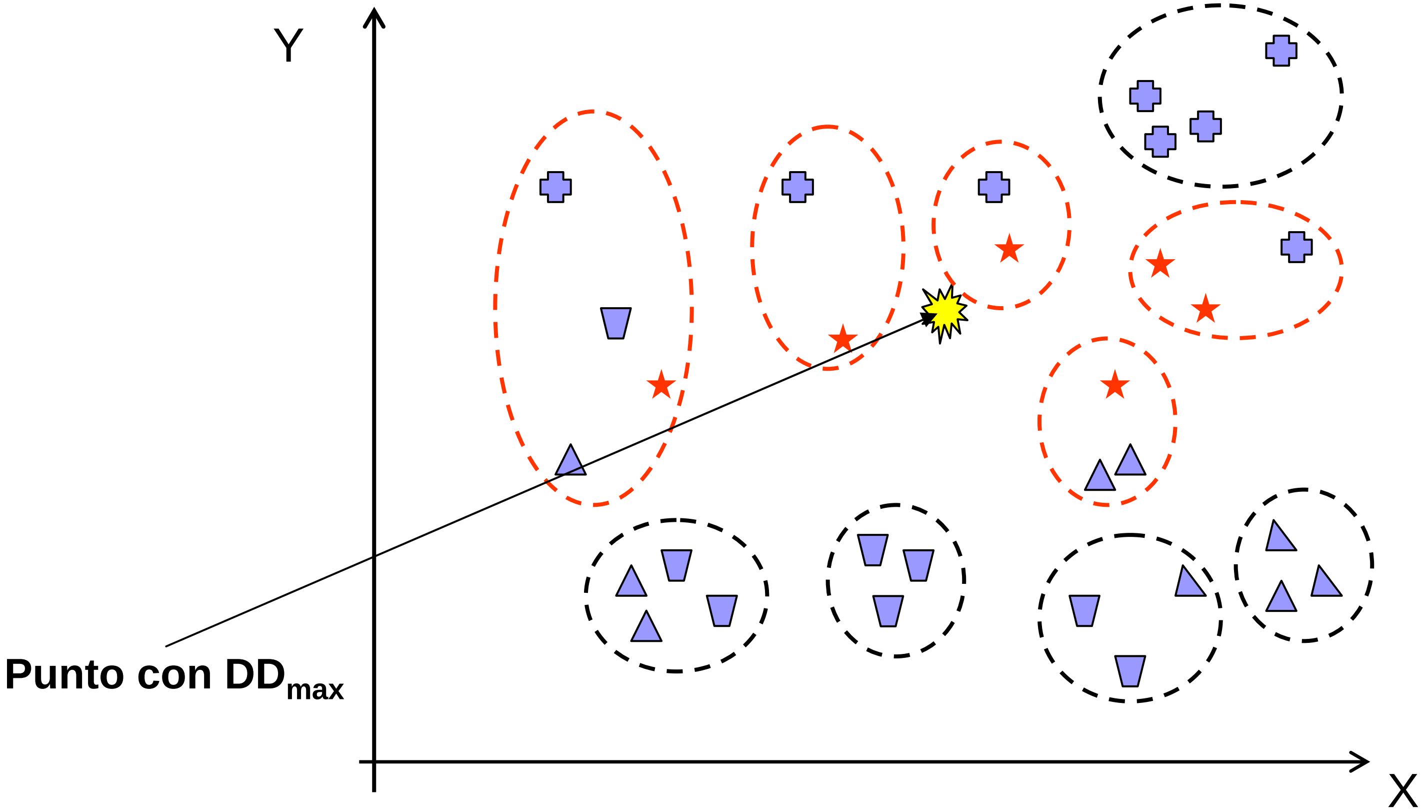


Diverse Density (DD)

O. Maron & T. Lozano Pérez. A Framework for Multiple-Instance Learning In Proc. of the 1997 Conference on Advances in Neural Information Processing Systems (1998) pp 570-576.

- Diverse Density (DD) was proposed by Maron and Lozano-Perez in 1998 as a general framework for solving multi-instance learning problems.
- The main idea of DD approach is to find a concept point in the feature space that are close to at least one instance from every positive example and meanwhile far away from instances in negative examples.
- The optimal concept point is defined as the one with the maximum diversity density, which is a measure of how many different positive bags have instances near the point, and how far the negative instances are away from that point.

Diverse Density (DD)



Expectation Maximization Diverse Density (EM-DD)

Q. Zhang & S. Goldman. EM-DD: An improved multiple-instance learning technique. *In Proc. of Neural Information Processing System* 14 (2001) pp 1073-1080.

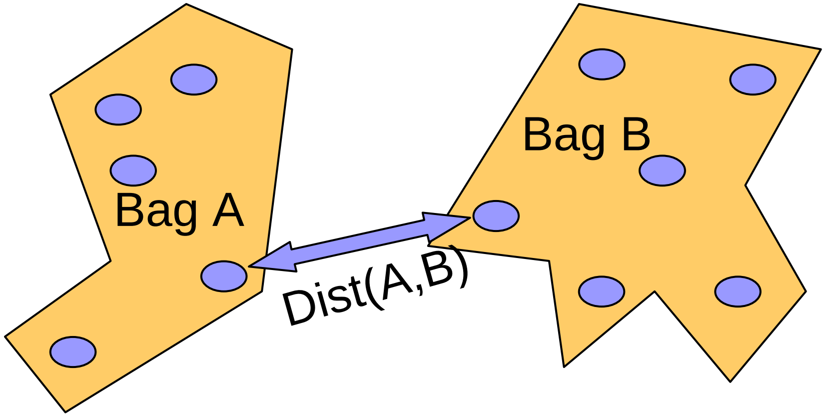
- It is a extended version of Diversity Density, this algorithm combine DD algorithm and Expectation Maximization approach.
- In the MIL setting, the label of a bag is determined by the "most positive" instance in the bag, i.e., the one with the highest probability of being positive among all the instances in that bag. The difficulty of MIL comes from the ambiguity of not knowing which instance is the most likely one.
- In this algorithm the knowledge of which instance determines the label of the bag is modeled using a set of *hidden variables*, which are estimated using the Expectation Maximization style approach.

K-Nearest Neighbor Algorithm

J. Wang & J.-D. Zucker Solving the multiple-instance problem: a lazy learning approach. In Proc of 17th International Conference on Machine Learning (2000), pp 1119-1125.

- The popular k Nearest Neighbor (k-NN) approach was adapted for MIL problems by Wang and Zucker in 2000.
- In the context of the multiple-instance problem, an example is a bag that contains multiple instances → therefore it is necessary to define a distance between two objects composed of two sets of instances.
  - They proposed the use of *minimum Hausdorff distance* to measure the proximity of objects.
- This distance was used as the bag-level distance metric, defined as the shortest distance between any two instances from each bag.

K-Nearest Neighbor Algorithm



$$\text{Dist}(A,B) = \min_{\substack{1 \le i \le n \\ 1 \le j \le m}} (\text{Dist}(a_i, b_j)) = \min_{a \in A} \min_{b \in B} \|a-b\|$$

Bayesian k-NN Algorithm

J. Wang & J.-D. Zucker Solving the multiple-instance problem: a lazy learning approach. In Proc of 17th International Conference on Machine Learning (2000), pp 1119-1125.

- The classic k-NN approach uses the majority vote to determine the class.

$$\underset{c \in \{\text{positive, negative}\}}{\arg\max} \sum_{i=1}^{k} \delta(c, c_i)$$

- Bayesian method provides a probabilistic approach that calculates explicit probabilities for hypothesis.

$$\underset{c \in \{\text{positive, negative}\}}{\arg\max} \; p(c \mid \{ c_1, c_2, \ldots, c_k \}) =$$

$$\underset{c \in \{\text{positive, negative}\}}{\arg\max} \frac{p(\{ c_1, c_2, \ldots, c_k \} \mid c) \; p(c)}{p(\{ c_1, c_2, \ldots, c_k \})} =$$

$$\underset{c \in \{\text{positive, negative}\}}{\arg\max} \; p(\{ c_1, c_2, \ldots, c_k \} \mid c) \, p(c)$$

Citation k-NN Algorithm

J. Wang & J.-D. Zucker Solving the multiple-instance problem: a lazy learning approach. In Proc of 17th International Conference on Machine Learning (2000), pp 1119-1125.

- Another way to adapt k-NN to the multiple-instance problem was inspired to us by the notion of *citation* from library and information science.
  - One well-known method is based on *references* and *citers*
    - If a research paper does cite another previously published paper (as known as its *reference*), the paper is said to be *related* to the reference.
    - Similarly, if a paper is cited by a subsequent article (as known as its *citer*), the paper is also said to be related to its citer.
    - Both citers and references are considered to be candidate documents related to a given paper.
- This algorithm uses the R-nearest references and the C-nearest citers of an unseen bag *b* to derive its class.

Decision Tree Algorithm

- Zucker and Chevaleyre (2001) presented an extension of the ID3 and C4.5 decision trees, named *multi-decision tree* (ID3-MI, C45-MI), to solve MIL problems.

  Y. Chevaleyre & J.-D. Zucker. Solving Multiple-Instance and Multiple-Part Learning Problems with Decision Trees and Rule Sets. Application to the Mutagenesis Problem. In E. Stroulia & S. Matwin (Eds): *Proc. of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence* 2001, LNAI 2056, pp 204-214, 2001.

- The growing of a decision tree is based on the *information gain* of a feature to *set* of instances, which is related to the *entropy* of the instances. They extended the concept of information gain and entropy to bags of instances in the MIL framework.

$$\text{Entropy}_{\text{multi}}(S) = -\frac{u(S)}{u(S)+v(S)}\log_2\frac{u(S)}{u(S)+v(S)} - \frac{v(S)}{u(S)+v(S)}\log_2\frac{v(S)}{u(S)+v(S)}$$

$$\text{InfoGain}_{\text{multi}}(S,F) = \text{Entropy}_{\text{multi}}(S) - \sum_{v\,\in\,\text{Values}(F)}\frac{u(S_v)+v(S_v)}{u(S)+v(S)}\,\text{Entropy}_{\text{multi}}(S_v)$$

Rule Based System Algorithm

- Zucker and Chevaleyre (2001) presented an extension on the rule learning algorithm RIPPER, named RIPPERMI, to solve MIL problems.

  Y. Chevaleyre & J.-D. Zucker. Solving Multiple-Instance and Multiple-Part Learning Problems with Decision Trees and Rule Sets. Application to the Mutagenesis Problem. In E. Stroulia & S. Matwin (Eds): *Proc. of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence* 2001, LNAI 2056, pp 204-214, 2001.

- The classic concept of rule coverage it is not enough when the goal is to discriminate bags not instances. It is necessary to give a formal definition of the multi-instance coverage:

$$Cover_{multi}\left(R, bag\right) = \exists\left(instance \in bag\right) Cover\left(R, instance\right)$$

$$Coverage_{multi}\left(R\right) = \left|\left\{bag_i \middle| Cover_{multi}\left(R, bag_i\right)\right\}\right|$$

Support Vector Machine

- There are several proposals which extend Support Vector Machine (SVM) to solve MIL problems.
  - MI-SVM and mi-SVM

    S. Andrews, T. Hofmann & I. Tsochantaridis. Multiple instance learning with generalized support vector machines. *In 18th National Conference on Artificial Intelligence (AAAI'02)* 2002, pp 943-944.

  - DD-SVM

    Y. Chen & J.Z. Wang. Image Categorization by Learning and Reasoning with Regions. Journal of Machine Learning Research, 5 2004, pp 913-939.

  - MIL-based SVMs

    X. Qi & Y. Han. Incorporating Multiple SVMs for Automatic Image Annotation. Pattern Recognition, 40:2 2007, pp 728-741.

Neural Network

- Neural Networks also have been adapted to solve MIL problems.
  - BP-MIP, is a feedforward neural network algorithm. The key of BP-MIP lies in the formal definition of the *multi-instance error function*

    Z.H. Zhou & M.L. Zhang. Neural Networks for Multiple Instance Learning. In *Proc. Of the 19th International Conference on Machine Learning*, July 2002

  - RBF-MIP, is based on popular methods of radial based function. The key of RBF-MIP lies int the use the a measure that consider the distance between bags.

    M.L. Zhang & Z.H. Zhou. Adapting RBF Neural Networks to Multiple Instance Learning. *Neural Processing Letters* 23:1-26, 2006

Ensembles

- Ensemble algorithms have emerged as an effective strategy in bioinformatics for improving the prediction accuracy by exploiting the synergetic prediction capability of multiple algorithms. Its adaptation to MIL perspective as was to be expected.
  - Ensembles-MI

    Z.H. Zhou & M.L. Zhang. Ensembles of Multiple Instance Learners. In *N., Lavrac and D., Gamberger and H., Blockeel and L., Todorovski (Eds.) ECML 2003*, Lecture Notes in Artificial Intelligence, 2837, pp 492-502.
  - MILogisticRegression y MIBoosting

    X. Xu & E. Frank. Logistic Regression and Boosting for Labeled Bags of Instances. In Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2003, pp 272-281.
  - CCE

    Z.H. Zhou & M.L. Zhang. Solving Multi-instance Problems With Classifier Ensemble Based on Constructive Clustering. Knowledge and Information Systemas, 11:2 2007, pp 155-170.

Inductive Logic Programming (ILP)

- ILP becomes interesting when the expressive power of first-order representation provides comprehensibility to learning result and capability to handle more complex data consisting of their relations.

  We can find proposals with use this approach:
  - TILDE-MI

    H. Bolckeel, L. De Raedt, N. Jacobs & B. Demoen. Scaling up inductive Logic Programming by Learning from Interpretations. *Data Mining Knowledge Discovery*, 3:1 1999, pp 59-93.
  - FOIL-MI

    J.R. Quinlan. Learning Logical Definitions From Relations. *Machine Learning*, 5 1990, pp-239-266
  - PROGOL-MI

    A. Srinivasan & S.H. Muggleton. Comparing the Use of Background Knowledge by Inductive Logic Programming Systems. *In Proc. Of International Workshop on Inductive Logic Programming* 1995, pp-60-67.

Programming Genetic for Multiple Instance Learning

Motivations

- Almost all popular Machine Learning algorithms have been applied to solve the multiple instance problem.
- There was no work about adapting Evolutionary Algorithm to this scenario.
- Evolutionary algorithms have shown its suitability solving learning problems in the supervised framework when the classic techniques do not work well.
- Our initial motivation was analyzing the suitability of Evolutionary Learning algorithms for solving problems in the Multiple Instance framework.

Genetic Programming

- Genetic Programming (GP), introduced by Koza, is becoming a popular paradigm in diverse tasks: both for obtaining classification rules and for prediction task, such as feature selection.
- Their results show that they can achieve of efficient way low rate of error.
- Their main characteristics are
  - A priori knowledge is not needed about the statistical distribution of the data.
  - Can operate directly on the data in their original form.
  - Can detect unknown relationship that exists among data.
  - Can discover the most important discriminative features of a class.

Grammar Guided Genetic Programming

- G3P (Grammar Guided Genetic Programming) is a variant of GP where:
  - Individuals are syntactic trees.
  - A grammar is used to enforce syntactic constraints and satisfy the closure property.
    - A grammar is composed of a set of non-terminal symbols, a set of terminal symbols, a starting symbol and a list of production rules.
    - An individual is created by the complete derivation of the starting symbol, applying the production rules and forcing all individuals to evolve satisfying the grammar.
- We proposed a extension of G3P algorithm to evolve rule sets to handle multiple instance learning:
  - We learn rules to classify instances, not bags.
  - We use the definition of Dietterich to classify bags.
  - The fitness measure the capacity of rules to classify correctly bags.

Individual Representation

- An individual represents a rule which can contains multiple antecedents attached by conjunction or disjunction which predict the same class.

antecedent := comparison |
“**OR**” comparison antecedent |
“**AND**” comparison antecedent

comparison:= numericalComparator valuesToCompare |
categoricalComparator <attribute>

numericalComparator := “**<**” | “**>=**”

categoricalComparator := “**CONTAIN**” | “**NOT_CONTAIN**”

valuesToCompare := <attribute> <value>

Numerical

antecedent := comparison |
“**OR**” comparison antecedent |
“**AND**” comparison antecedent

comparison:= numericalComparator valuesToCompare

numericalComparator := “**<**” | “**>=**”

valuesToCompare := <attribute> <value>

Categorical

antecedent := comparison |
“**OR**” comparison antecedent |
“**AND**” comparison antecedent

comparison:= categoricalComparator valuesToCompare

categoricalComparator := “**CONTAIN**” | **NOT_CONTAIN**”

valuesToCompare := <attribute> <value>

Individual Representation

- The individual genotype represents rules which are applied on instances

$\text{Rule}_{\text{INS}}\,(\text{instance}_j) \Rightarrow$ **IF** (antecedent **(**instance_j**)**)
THEN The instance_j is positive
ELSE The instance_j is negative

- The individual phenotype represents a complete classifier applied on objects according to the classical MIL hypothesis

$\text{Rule}_{\text{BAG}}\,(\text{bag}_i) \Rightarrow$ **IF** ($\exists\ \text{instance}_j\ \varepsilon\ \text{bag}_i$: $\text{Rule}_{\text{INS}}(\text{instance}_j)$)
THEN The bag_i is positive
ELSE The bag_i is negative

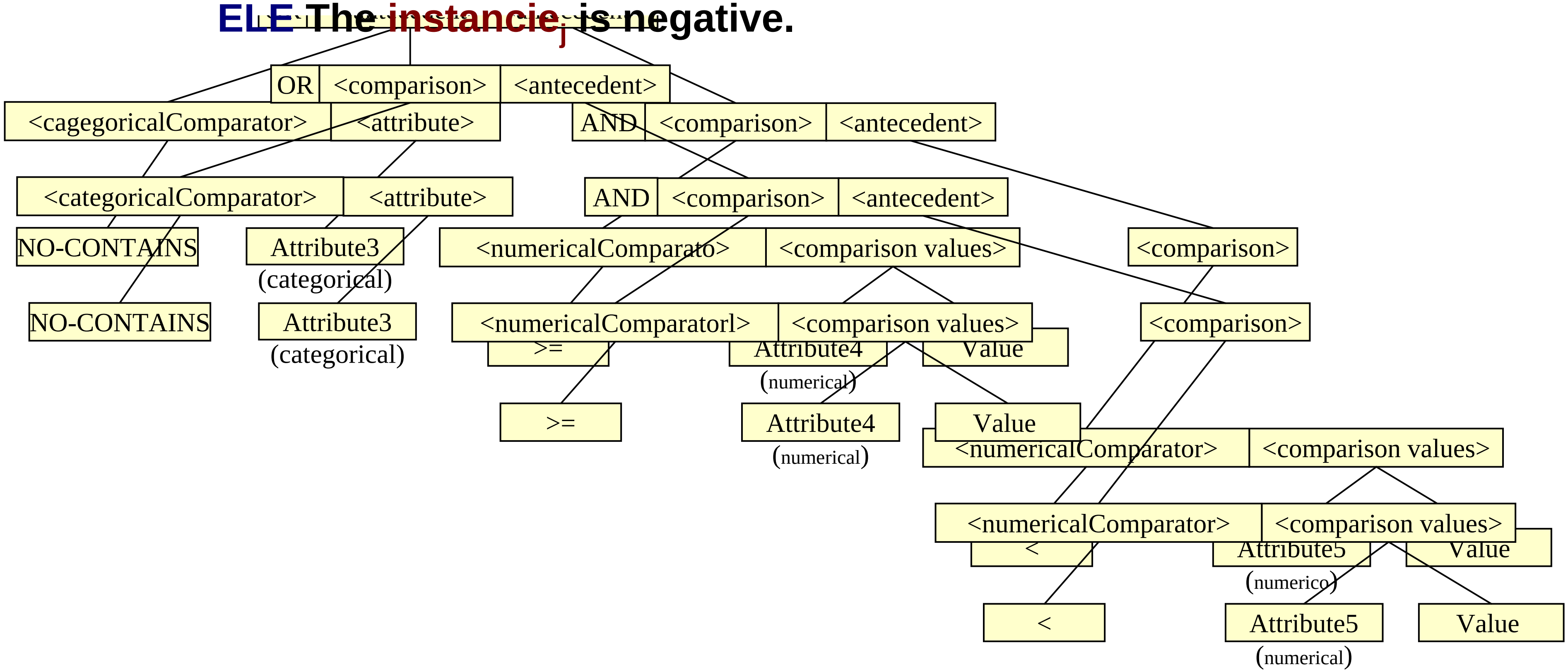
Genetic Operator

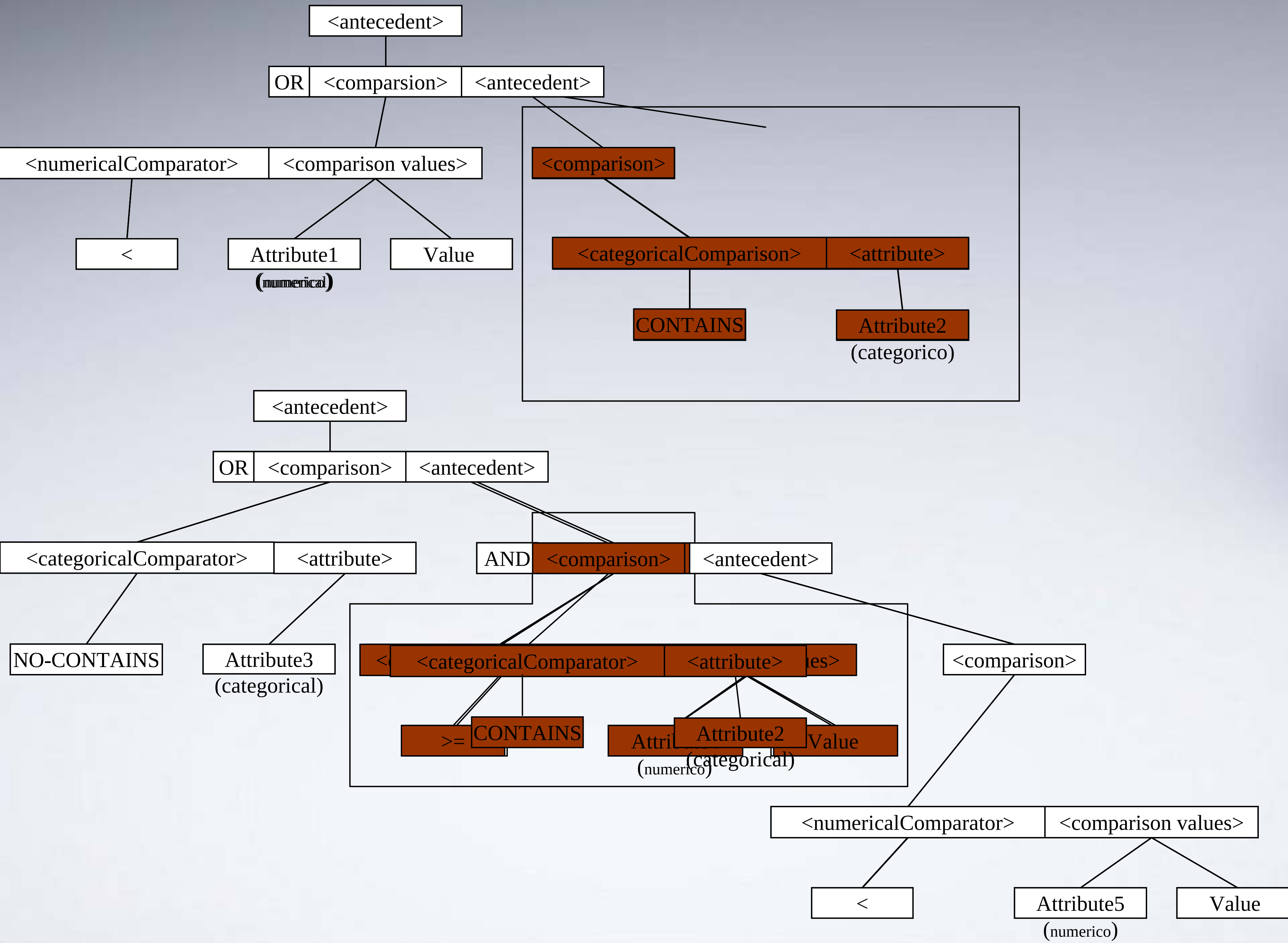
Rule$_{INS}$ (instance$_j$) ⇒

IF ((NOT CONTAIN attribute3) OR ((attribute4>=Value) AND (attribute5<Value)))

THEN The instance$_j$ is positive.

ELE The instancie$_j$ is negative.




<antecedent>
OR
<comparsion>
<antecedent>
<numericalComparator>
<comparison values>
<
Attribute1
(numerical)
Value
<comparison>
<categoricalComparison>
<attribute>
CONTAINS
Attribute2
(categorico)
<antecedent>
OR
<comparison>
<antecedent>
<categoricalComparator>
<attribute>
NO-CONTAINS
Attribute3
(categorical)
AND
<comparison>
<antecedent>
<categoricalComparator>
<attribute>
>=
CONTAINS
Attribute2
(categorical)
(numerico)
Value
<comparison>
<numericalComparator>
<comparison values>
<
Attribute5
(numerico)
Value

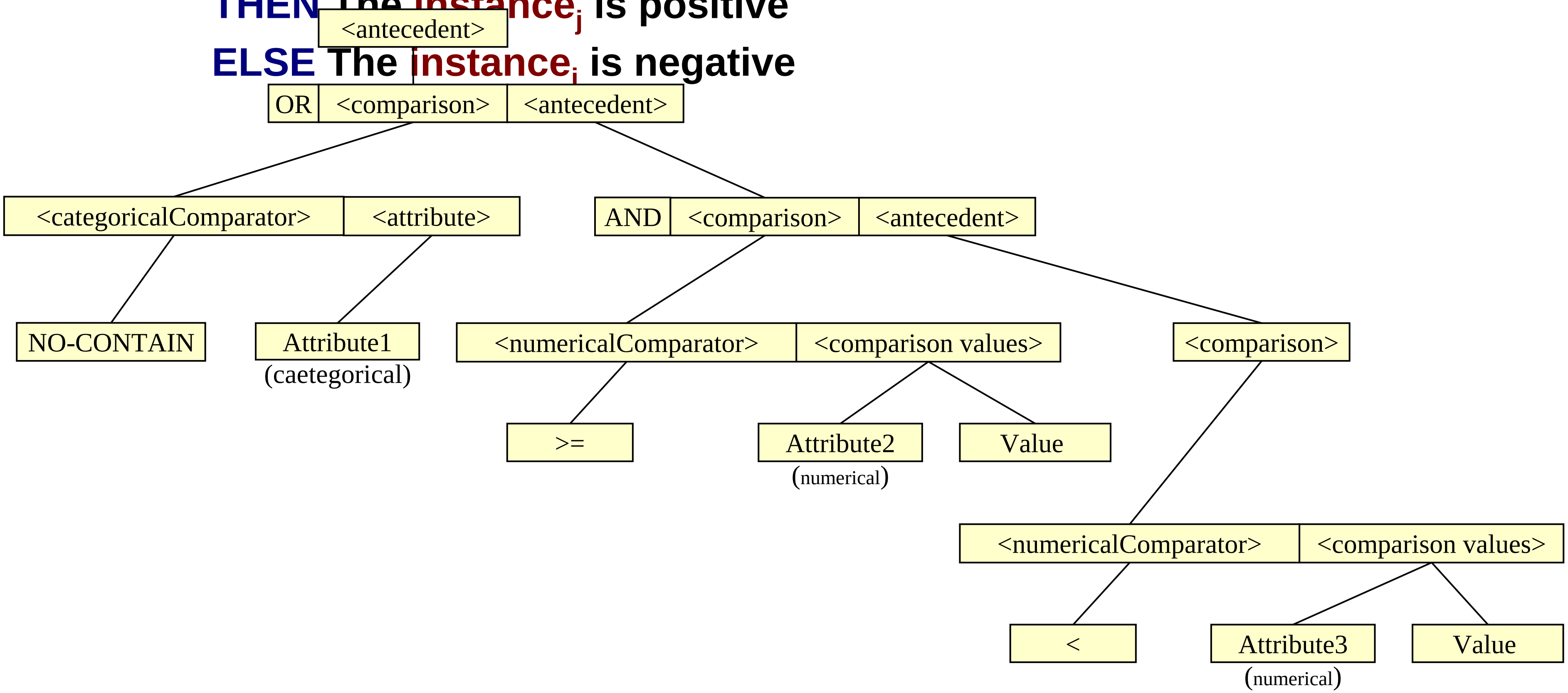
Genetic Operator

Rule$_{INS}$ (instancia$_j$) ⇒

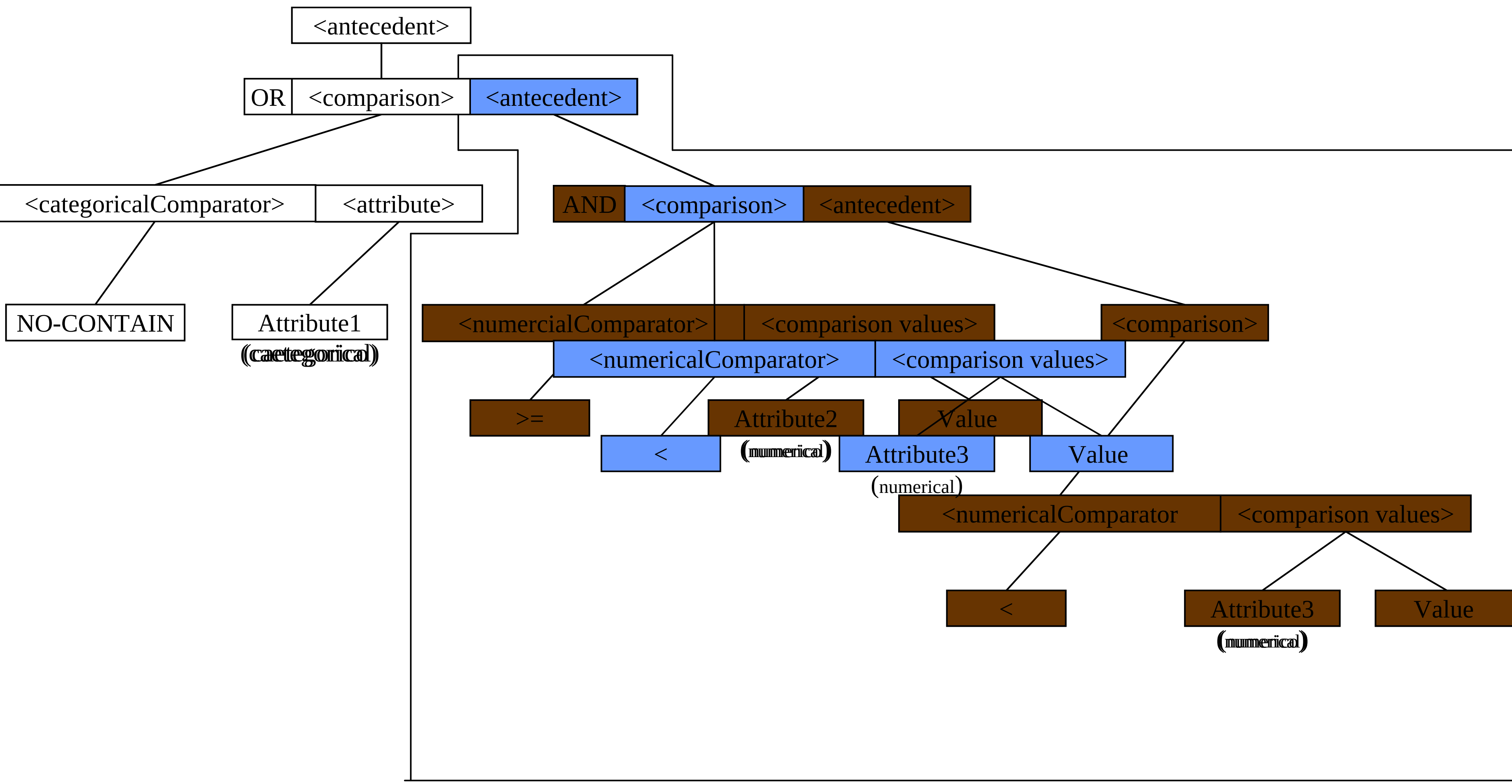
IF ((NOT CONTAIN attribute1) OR ((attribute2>=Value) AND (attribute3<Value)))

THEN The instance$_j$ is positive

ELSE The instance$_j$ is negative



Genetic Operator



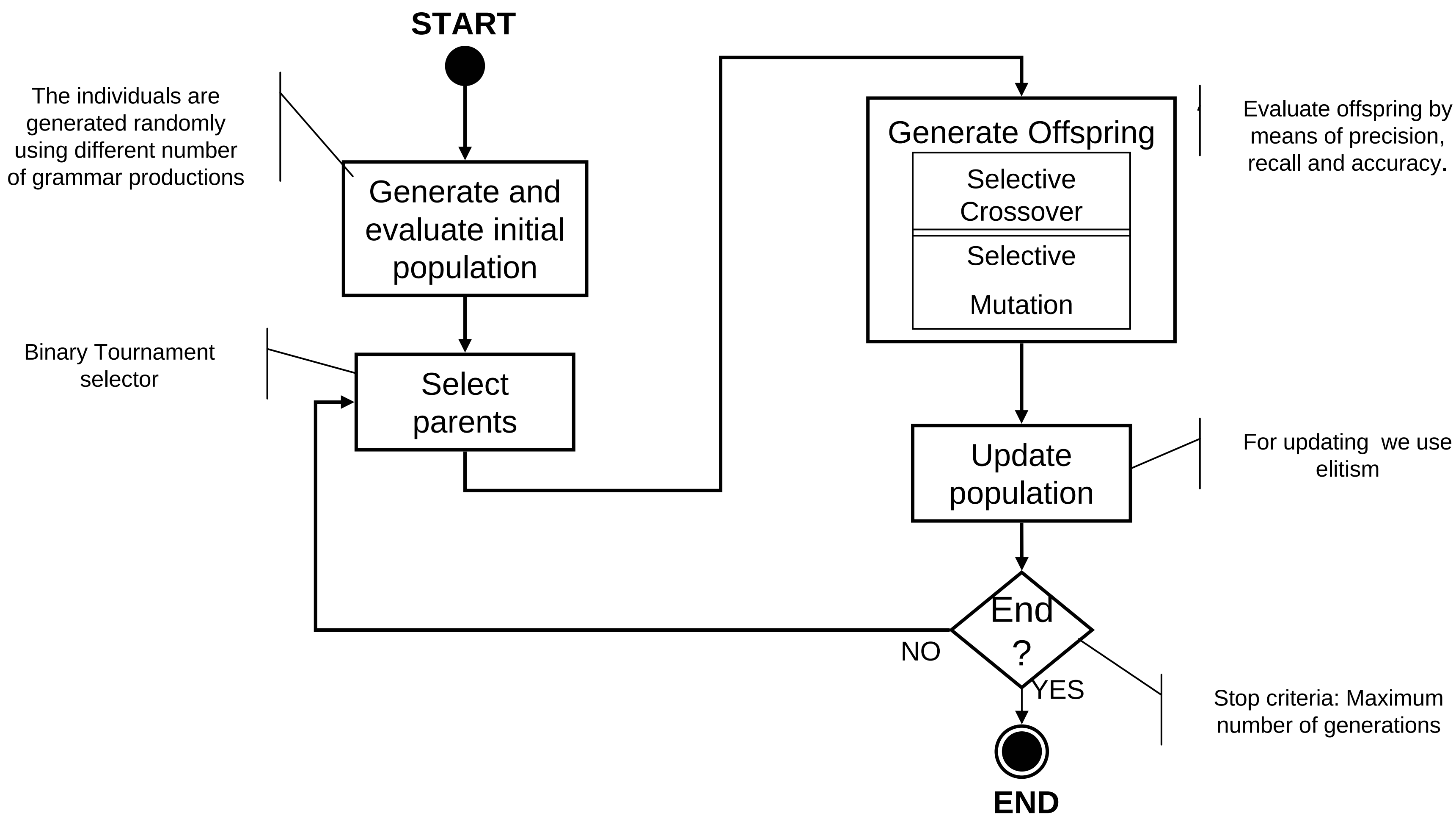
Fitness Function

- The rule evaluation uses basic criteria of supervised learning adapted to Multi-instance Learning:
  - ***Accuracy***, $= \frac{t_p + t_n}{P + N}$ is the proportion of cases correctly identified according to all cases.
  - ***Sensitivity or Recall***, $= \frac{t_p}{t_p + f_n}$ is the proportion of positive cases correctly identified.
  - ***Specificity***, $= \frac{t_n}{t_n + f_p}$ is the proportion of negative cases correctly identified.
  - ***Precision***, $= \frac{t_p}{t_p + f_p}$ is the proportion of positive cases correctly identified according to all cases identified as positives cases.

t_p – number of positive basg correctly identified.
t_n – number of negative basg correctly identified.
f_p – number of positive bags not correctly identified
f_n – number of negative bags not correctly identified.
P y N – positive and negarive bags

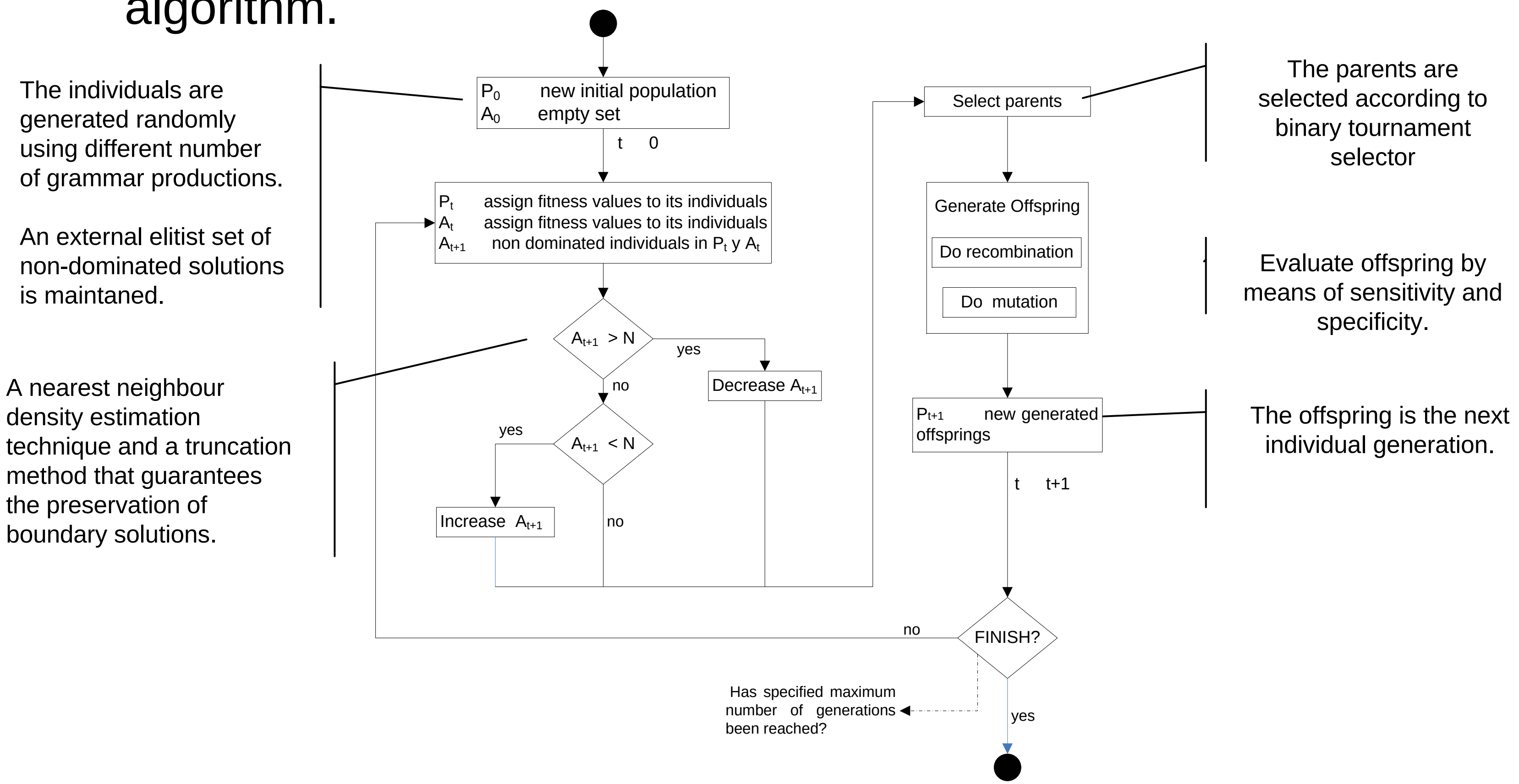
G3P-MI Algorithm

- According to previous specifications, G3P-MI would be



MOG3P-MI Algorithm

- It is a first proposal to extend multi-objective grammar guided genetic programming (MOG3P) to handle Multi-instance Learning. It is derived from traditional G3P method and SPEA2 multi-objetive algorithm.



Experiments and Results

Datasets used for testing G3P and MOG3P algorithms

- In our experiments we used data from two real-world application domains:
  - Drug activity prediction
    - Musk, the aim is to predict whether a new molecule is of “musk" or “non-musk" type. The Musk problem has two versions, Musk1 and Musk2
    - Mutagenesis, the aim is to predict whether a new molecule has “mutagensis” property or does not have this property. The Mutagenesis problem has two versions, Mutagenesis-hard, Mutagenesis-easy.
  - Web index recommendation
    - The aim is to predict whether a web index page is interesting for the user or not. This problem considerers 9 dataset, each dataset is a user that classify each web index page as interesting or not interesting.

Drug Activity Prediction

- The next table list the key properties of datasets used in this application.

	Musk		Mutagenesis	
Datasets	Musk1	Musk2	Easy	Hard
Number of Bags	92	102	188	42
Number of positive bags	47	39	125	13
Number of negative bags	45	63	63	29
Number of instances	476	6598	10486	2132
Minimum bag size	2	1	28	26
Maximum bag size	40	1044	88	86

Drug Activity Prediction

- Parameters used by our algorithms

Algorithms	G3P-MI	MOG3P-MI
Size of Extern Population	-	50
Size of Population	1000	1000
Number of generations	100	100
Crossover probability	95 %	95 %
Mutation probability	10 %	10%
Parent selector	Roulette	Tournament
Generate initial population	Random	
Maximum tree depth	50	50

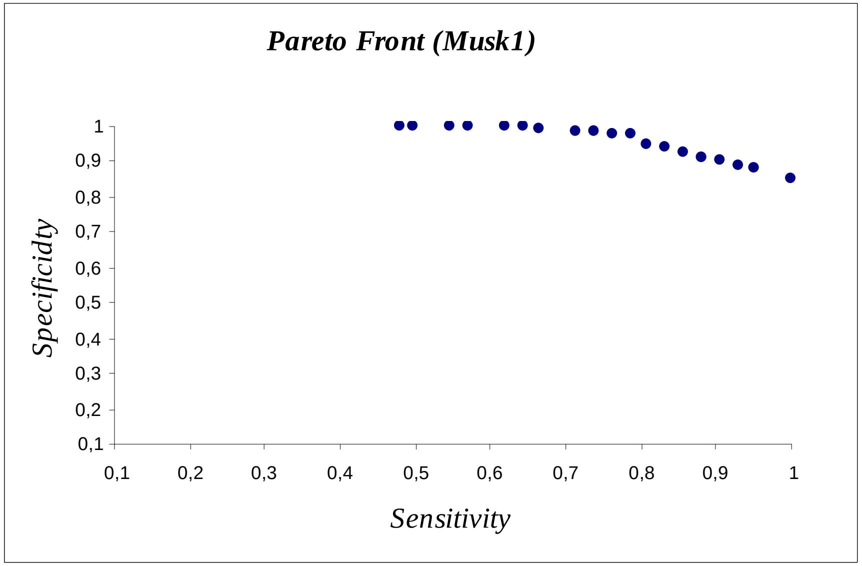
Drug Activity Prediction

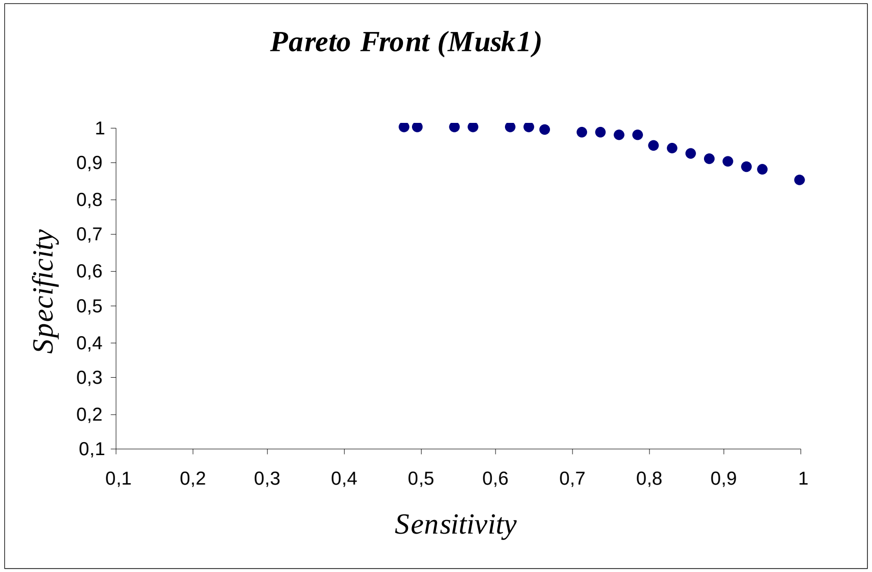
	Musk1	Musk2
Algorithm	**Acc**	**Acc**
EM-DD	96.8	96.0
MOG3P-MI	93.3	97.0
CCE	92.4	87.3
Iterated-discrim APR	92.4	89.2
Citation-kNN	92.4	86.3
MI Kernel	92.4	92.2
GFS elim-kde APR	91.3	80.4
GFS elim-count APR	90.2	75.5
Bayesian-kNN	90.2	82.4
Diverse Density	88.9	82.5
TLC without AS	88.7	83.1
RIPPER-MI	88.0	77.0
NAIVE-RIPPERMI	88.0	77.0

	Musk1	Musk2
Algoritmo	**Acc**	**Acc**
BP-MIP-PCA	88.0	83.3
MI-BOOST	87.9	84.0
BP-MIP-DD	85.9	80.4
RELIC	83.7	80.4
BP-MIP	83.7	80.4
G3P-MI	81.1	82.0
MI-SVM	77.9	84.3
MULTINST	76.7	84.0
BP	75.0	67.7
C4.5	68.9	58.8
TILDE	87.0	79.0

Drug Activity Prediction

- Pareto Front obtained for Musk data sets





Drug Activity Prediction

- The obtained classifiers are simple and understandable. An example of rule obtained for this dataset is the following

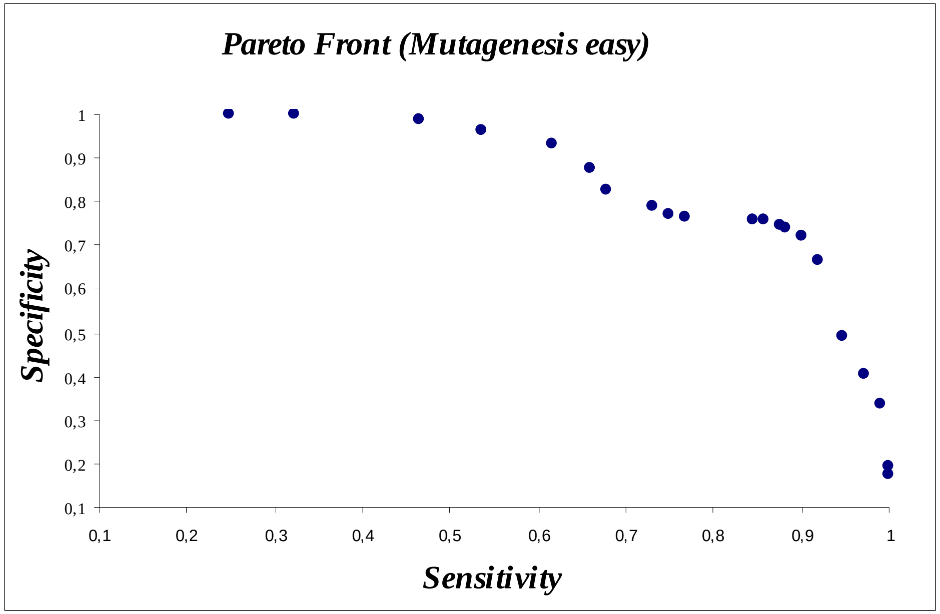
IF ((f10>-220.381) $\wedge$ (f7>35.268) $\wedge$ (f163$\leq$ 199.682)
$\wedge$ (f55>-84.999) $\wedge$ (f134>-216.046) $\wedge$ (f34 >216.149)
$\wedge$ ((f128$\leq$18.115) $\vee$ (f140>13.682) $\vee$ (f136>78.262))
THEN The molecule is musky.
ELSE The molecule is non musky.

Drug Activity Prediction

	Mutagenesis hard	Mutagenesis easy
Algorithm	**Acc**	**Acc**
MOG3P-MI	**100.0 %**	**84.0 %**
G3P-MI	**100.0 %**	**72.0 %**
RIPPER-MI	**91.0 %**	**82.0 %**
TILDE	**86.0 %**	**770 %**
PROGOL	**86.0 %**	**76.0 %**
FOIL	**83.0 %**	**61.0 %**

Drug Activity Prediction

- Pareto Front obtained for Mutagenesis easy



Drug Activity Prediction

- The obtained classifiers are simple and comprehensivility. An example of rule obtained for this dataset is the following

IF ((element2 = 7.0) $\wedge$ (charge1 > -0.486) $\wedge$ (charge2 $\leq$ -0.5673) $\wedge$ ((quanta1 $\neq$ 9.0) $\vee$ (charge2 > -0.371))

THEN The molecule is active.

ELSE The molecule is inactive.

Web Index Recommendation

- Several experiments are performed to evaluate the performance of our G3P-MI algorithm on web index recommendation problem
  - With respect to datasets
    - Data sets that consider all words.
    - Data sets that considerer words that appear in more of 3 bags and less of 40 bags.
  - With respect to document representation
    - **Binary representation**. Each document is a binary feature vector, only we know if the word appears in the document or not.
    - **Numerical representation**. Each document is an integer feature vector, we know the frequency of each word in the document.

Web Index Recommendation

- The next table list the key properties of datasets used in this application.

		Dataset								
		V1	V2	V3	V4	V5	V6	V7	V8	V9
Training	Positive	17	18	14	56	62	60	39	35	37
	Negative	58	57	61	19	13	15	36	40	38
Test	Positive	4	3	7	33	27	29	16	20	18
	Negative	34	35	31	5	11	9	22	18	20

Web Index Recommendation

- Parameters used by our algorithms.

Algorithms	G3P-MI	MOG3P-MI
Size of Extern Population	-	50
Size of Population	1000	1000
Number of generations	100	100
Crossover probability	95 %	95 %
Mutation probability	10 %	30%
Parent selector	Roulette	Tournament
Generate initial population	Random	
Maximum tree depth	50	50

Web Index Recommendation

	Algorithm	Acc	Se	Sp
Binary	Fretcit-kNN	0.8103	0.7007	0.7803
	Txt-KNN	0.7233	0.7380	0.4847
	Citation-KNN	0.7577	0.6073	0.7407
	G3P-MI	0.7810	0.7723	0.7297
	□MOG3P-MI	0.8480	0.7793	0.7567
Numerical	□Fretcit-kNN	0.8043	0.7117	0.7420
	Citation-KNN	0.7357	0.7020	0.5283
	Txt-KNN	0.7630	0.6130	0.7207
	G3P-MI	0.7313	0.9403	0.4013
	MOG3P-MI	0.8420	0.8727	0.7037

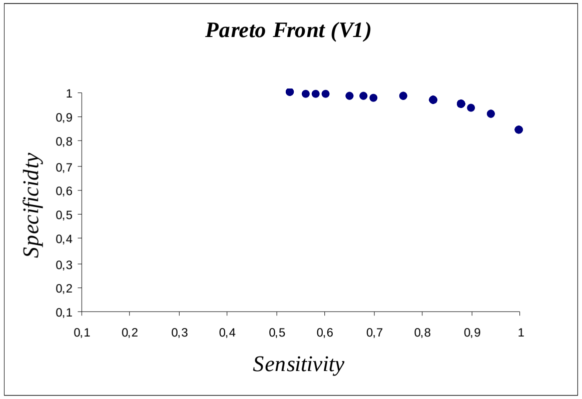
Web Index Recommendation

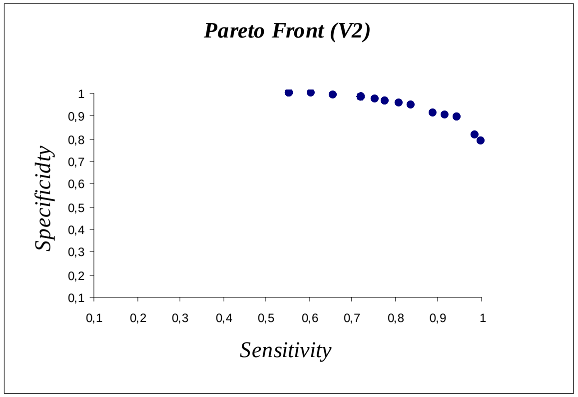
- Results obtained

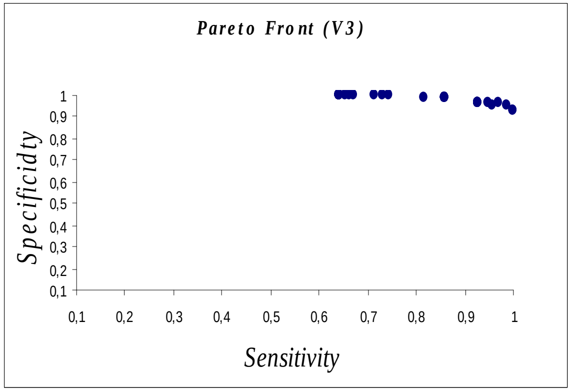
	Algorithm	Majority negative sets Acc	Majority negative sets Se	Majority negative sets Sp	Majority positive sets Acc	Majority positive sets Se	Majority positive sets Sp	Balanced sets Acc	Balanced sets Se	Balanced sets Sp
Binary	**Txt-KNN**	**0.795**	**0.636**	**0.822**	**0.805**	**0.863**	**0.194**	**0.570**	**0.715**	**0.438**
	Citation-KNN	**0.803**	**0.397**	**0.868**	**0.796**	**0.863**	**0.577**	**0.674**	**0.562**	**0.777**
	Fretcit-kNN	**0.879**	**0.579**	**0.919**	**0.854**	**0.924**	**0.634**	**0.698**	**0.599**	**0.788**
	G3P-MI	**0.807**	**0.690**	**0.919**	**0.825**	**0.877**	**0.628**	**0.711**	**0.750**	**0.642**
	MOG3P-MI	**0.904**	**0.579**	**0.950**	**0.868**	**0.975**	**0.557**	**0.772**	**0.784**	**0.763**
Numerical	**Txt-KNN1**	**0.795**	**0.519**	**0.843**	**0.812**	**0.851**	**0.264**	**0.600**	**0.736**	**0.478**
	Citation-KNN1	**0.833**	**0.402**	**0.907**	**0.782**	**0.851**	**0.498**	**0.674**	**0.586**	**0.757**
	Fretcit-kNN1	**0.870**	**0.615**	**0.904**	**0.811**	**0.916**	**0.470**	**0.732**	**0.604**	**0.852**
	G3P-MI1	**0.845**	**0.821**	**0.904**	**0.823**	**1.000**	**0.201**	**0.526**	**1.000**	**0.099**
	MOG3P-MI1	**0.895**	**0.774**	**0.919**	**0.860**	**1.000**	**0.466**	**0.771**	**0.844**	**0.726**

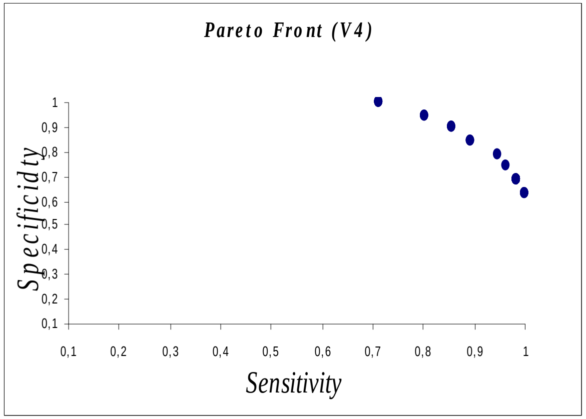
Web Index Recommendation

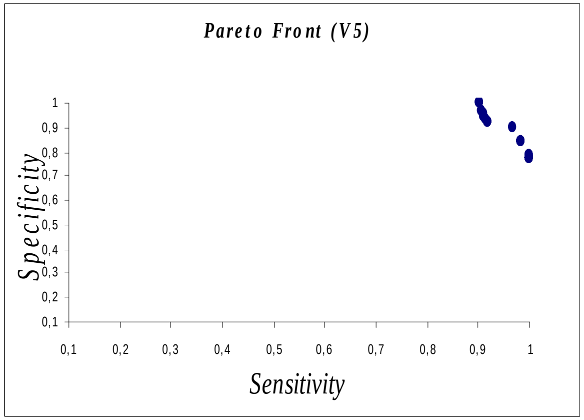
- Pareto Front obtained for each data set

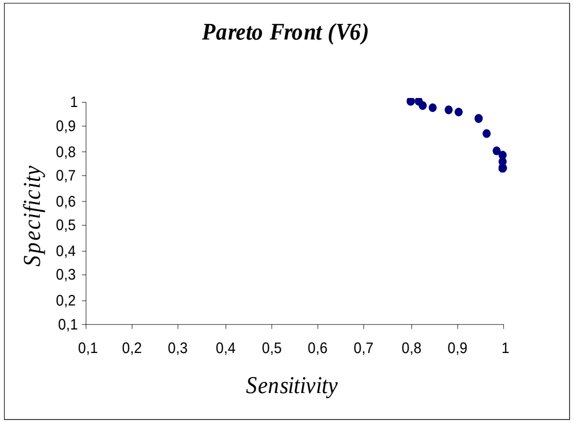












Web Index Recommendation

- A rule obtained for the first user/dataset using binary representation.

IF (((**CONTAINS** “planet”) **AND** (**CONTAINS** “forecast”)) **OR** (**CONTAINS** “atmospheric“) OR (**NOT-CONTAINS** “football“))

THEN *Recommend page to V1 user.*

ELSE *Not recommend page to V1 user.*

- A rule obtained for the first user/datset using numerical representacion.

IF ((“french” > 16) **OR** (“house” > 11) **OR** (“science” > 2) **OR** ((“aol” > 7) **AND** (“online” > 6)))

THEN *Recomend page to V1 user.*

ELSE *Not recommend page to V1 user.*

References

Bibliography - Software

- Bibliography about MIL in KEEL web project http://www.keel.es/
- Software about MIL:
  - **MILK** – A Multi-Instance Learning Kit in Java. It is based on Weka. It can be download in: http://www.cs.waikato.ac.nz/ml/milk/
  - **G3P-MI** – The two versions of algorithm mono-objective and multi-objective are been implemented in a JCLEC framework (http://jclec.sf.net).

Datasets

- http://www.cs.waikato.ac.nz/ml/milk/
  - It is available the datasest *Musk*1 and Musk2 and *Mutagénesis*-easy and Mutagenesis-hard in weka format.
- http://www.cs.columbia.edu/~andrews/mil/datasets.html
  - It is available the datasets of text categorization and image classification. This web is mantained by Andrews.
- http://www.cs.wustl.edu/~sg/multi-inst-data/
  - It is available datasets of content-based imagen retrieve and drug activity prediction. This web is mantained by Sally Goldman, a researcher that work in MIL.
- http://www.cs.wisc.edu/~sray/MIPage.html
  - It is availabe bibliography about MIL. This web is mantainede by S. Ray.