INTELLIGENT AND HYBRID DATA MINING METHODS IN INDUSTRIAL APPLICATIONS.

The 1st International Scientific Conference Intelligent Information Technologies for Industry

Pavel Krömer[1]

[1]Dept. of Computer Science,
VŠB - Technical University of Ostrava,
Ostrava, Czech Republic
pavel.kromer@vsb.cz

OUTLINE

Introduction

Intelligent methods (metaheuristics) form a class of algorithms that try to apply various unconventional approaches to efficiently solve problems.

Nature–inspired metaheuristics mimick successful optimization strategies from nature. The strategies are applied to iteratively develop single or multiple problem solutions and aim at adaptability (to each domain) and reusability (for every domain).

They include methods of swarm intelligence (Ant Colony Optimization, Particle Swarm Optimization), evolutionary computation (Genetic Algorithms, Genetic Programming) and e.g. neural computation (Artificial Neural Networks) and are considered a part of the wide family of Computational Intelligence methods.

Introduction (cont.)

Hybrid intelligent methods is an umbrella term referring to algorithms combining two or more intelligent approaches into a coherent framework adressing some theoretical or practical problem.

This talk provides an (utterly incomplete) overview of selected intelligent methods, describes the anatomy of one hybrid algorithm, and points out some of its applications to industrial problems.

Nature-inspired intelligent methods

SWARM INTELLIGENCE

Swarm intelligence is a collection of methods to solve complex, real-world problems using the paradigm of collective behaviour of distributed agents. This paradigm has been inspired by the intelligent behaviour of systems composed of many simple individuals, such as ants, bees, bats, etc. Similarly, an artificial swarm system consists of many unsophisticated agents that cooperate in order to achieve desired behaviour. This approach is concerned with exploiting global behavioural patterns emerging from local interactions, rather than with the design of sophisticated central controllers.

ANT COLONY OPTIMIZATION

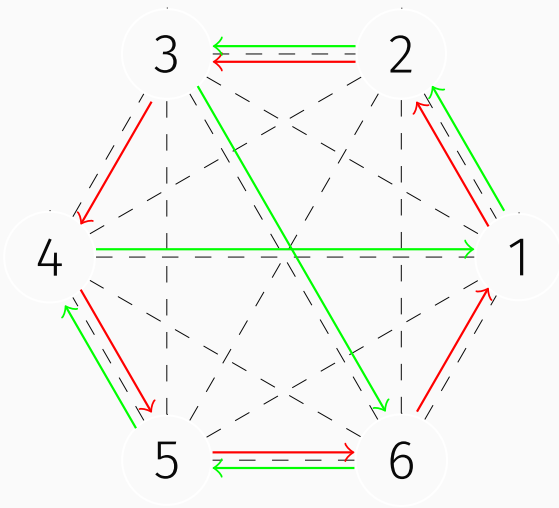
Ant Colony Optimization (ACO) is a meta-heuristic approach based on certain behavioural patterns of foraging ants. Ants have shown the ability to find optimal paths between their nests and sources of food.

This intelligent path-finding activity is based on stigmergy – an indirect communication through modification of the environment.

Emulation of ants' behaviour can be used as a probabilistic computational technique for solving complex problems that can be cast as finding optimal paths.

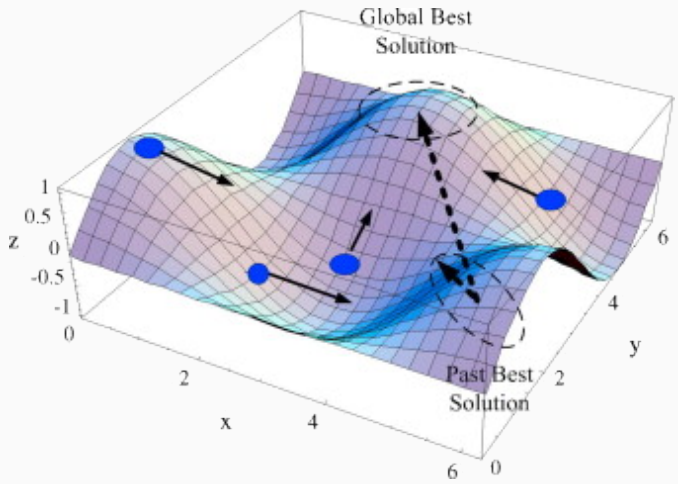
ANT COLONY OPTIMIZATION (CONT.)

Generate initial pheromone matrix P with respect to graph topology;
$0 \rightarrow$ *generation*;
while *Termination criteria not met* **do**
    Place n ants randomly on graph vertices. Set the amount of collected food for each ant to 0;
    foreach *ant* **do**
        Move forward on the graph; follow the probabilistic rule;
        Compute the amount of collected food corresponding to ants trail;
    end
    Find ant with the largest amount of collected food; let the ant lay pheromones in P on its trail;
    Evaporate pheromones in P;
    generation $+ 1 \rightarrow$ *generation*;
end



Particle Swarm Optimization (PSO) is an algorithm based on simulation of swarming behaviour of bird flocks, fish schools and even human social groups.

Create population of M particles with random position and velocity;
Evaluate an objective function f ranking the particles in the population;
while *Termination criteria not satisfied* **do**
  for $i \in \{1, \ldots, M\}$ **do**
    Set personal and global best position:
    if $f(x_i) < f(y_i)$ **then**
      $y_i = x_i$
    end
    if $f(x_i) < f(\bar{y})$ **then**
      $\bar{y} = x_i$
    end
    Update velocity of particle i and its position;
  end
end



EVOLUTIONARY COMPUTATION

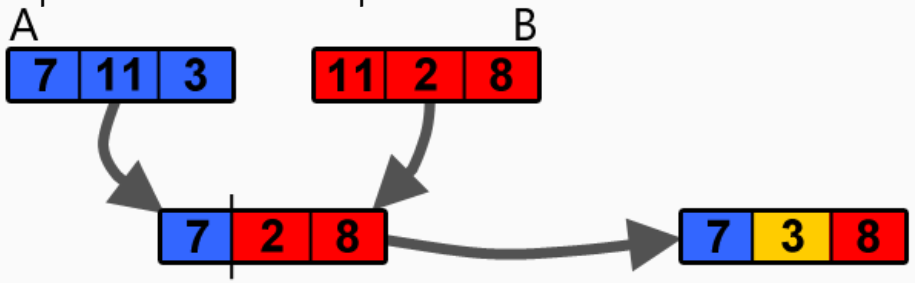
Evolutionary computation is a group of iterative stochastic search and optimization methods based on the programmatical emulation of successful optimization strategies observed in nature. Evolutionary algorithms use Darwinian evolution and Mendelian inheritance to model the survival of the fittest using the processes of selection and heredity.

GENETIC ALGORITHMS

The Genetic Algorithm (GA) is a population-based, meta-heuristic, soft optimization method. GAs can solve complex optimization problems by evolving a population of encoded candidate solutions. The solutions are ranked using a problem specific fitness function. Artificial evolution, implemented by iterative application of genetic and selection operators, leads to the discovery of solutions with above-average fitness.

Encoding

Problem encoding is an important part of GA. It translates candidate solutions from the problem domain (phenotype) to the encoded search space (genotype) of the algorithm. The representation specifies the chromosome data structure and the decoding function.



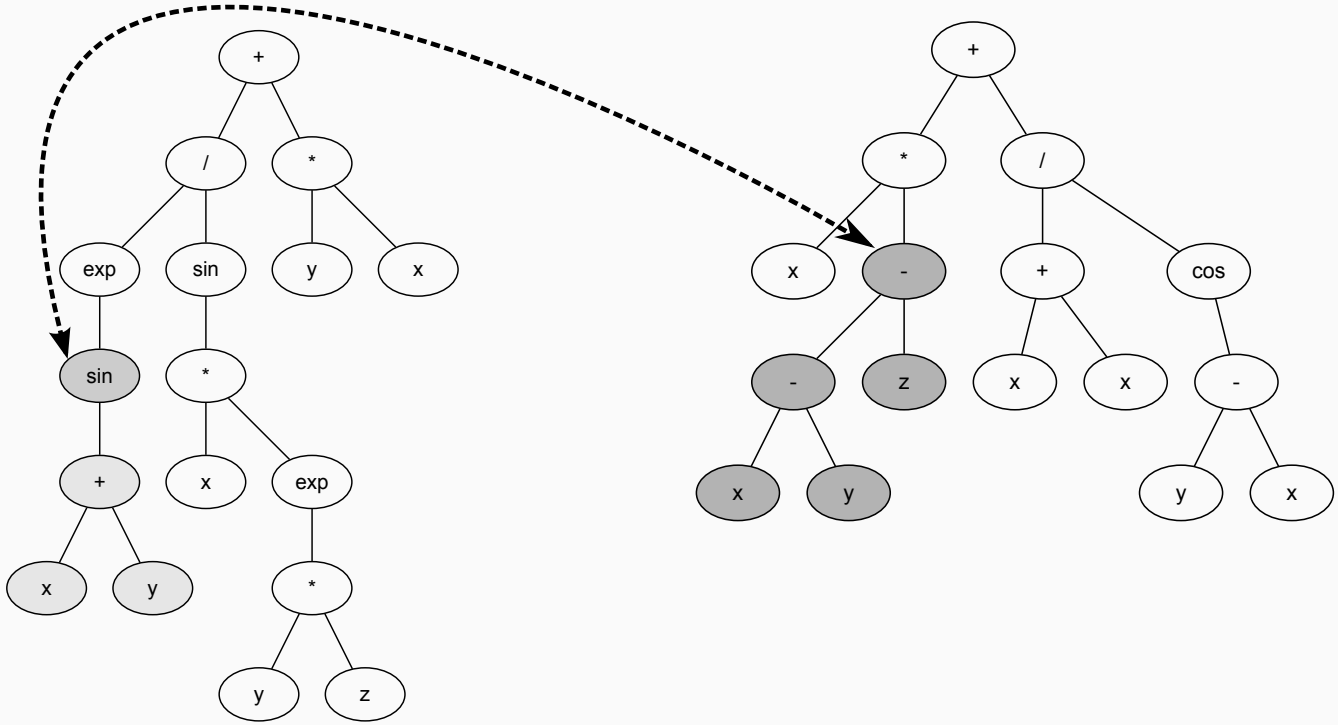
Genetic operators

Crossover recombines two or more chromosomes. It propagates so called building blocks (solution patterns with above average fitness) from one generation to another, and creates new, better performing, building blocks.

In contrast, mutation is expected to insert new material into the population by random perturbation of chromosome structure. This way, new building blocks can be created or old disrupted.

Genetic programming (GP) is an extension to GA, allowing work with hierarchical, often tree-like, chromosomes with an unlimited length

Genetic operators are redesigned to work with trees



Differential evolution (DE) is a versatile stochastic evolutionary optimization algorithm for real-valued problems. It uses differential mutation

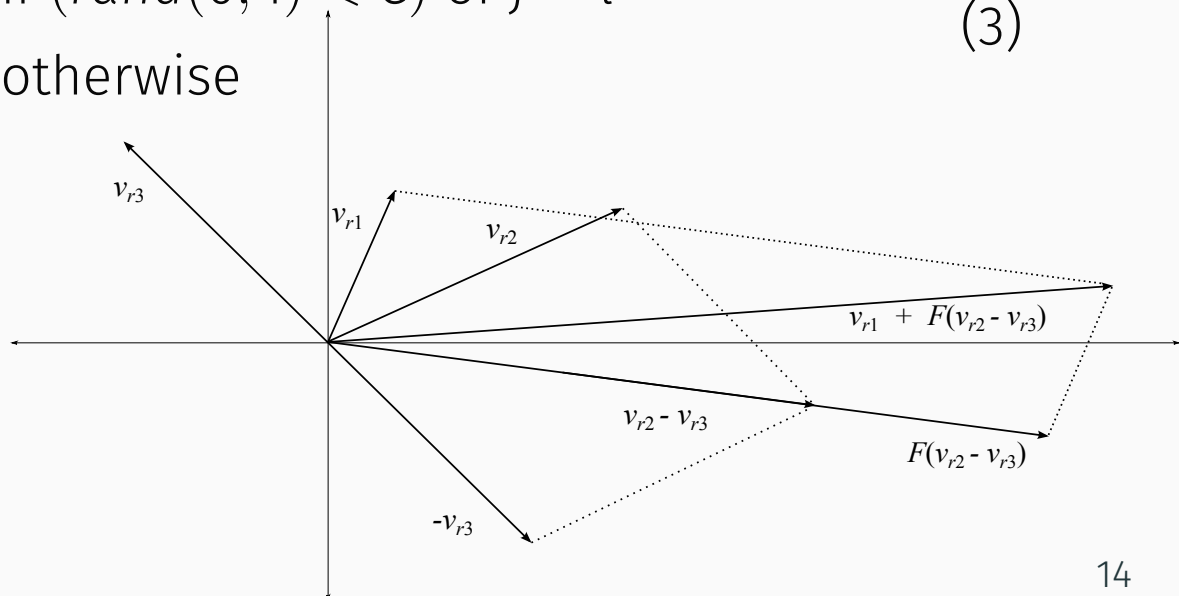
$$\vec{v}^{i} = \vec{v}^{r1} + F\left(\vec{v}^{r2} - \vec{v}^{r3}\right), \tag{1}$$

and crossover operator

$$l = rand(1, N), \tag{2}$$

$$v_j^i = \begin{cases} v_j^i, & \text{if } (rand(0,1) < C) \text{ or } j = l \\ x_j^i, & \text{otherwise} \end{cases} \tag{3}$$

to evolve a population of parameter vectors.



OTHER METHODS

There is a plethora of different intelligent methods based on various computing paradigms, including:

- fuzzy set theory, fuzzy logic, fuzzy systems
- artificial neural networks
- kernel based methods
- search based methods
- network intelligence
- etc

Hybrid Intelligent Methods

HYBRID INTELLIGENT METHODS

Hybrid intelligent methods leverage the power of intelligent algorithms by an efficient combination of the underlying (theoretical, computing) paradigms. The combination addresses the imperfections of used algorithms per–se or in context of a particular application domain.

There are many well–known examples of such multi–paradigm approaches:

- neuroevolution – is a family of methods that apply evolutionary computation to artificial neural networks.
- genetic fuzzy systems – exploit artificial evolution to optimize or synthesize fuzzy systems (e.g. fuzzy controlers).

EVOLUTIONARY FUZZY RULES

Evolutionary Fuzzy Rules (FR) is a multi–paradigm classification and regression method based on the merger of *fuzzy information retrieval* and *genetic programming*.

Evolutionary Fuzzy Rules were conceived as an advanced soft-computing method especially designed to:

- exploit the principles of fuzzy set theory and information retrieval for data processing
- provide adaptive models that can be learned from data
- achieve accurate and inexpensive soft classification and prediction
- deliver comprehensible models with clear syntax and semantics

EVOLUTIONARY FUZZY RULES (CONT.)

A FR is a standalone soft-computing model for data processing.

Fuzzy rule structure

The FR is a symbolic expression that can be parsed into a tree. The tree consists of nodes and leafs (terminals). Three types of terminals are used:

1. *feature*, i.e a requirement on a data attribute
2. *past feature* i.e. a requirement on an attribute in a *previous record*
3. *past output* which puts a requirement on a *previous model output*

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

EVOLUTIONARY FUZZY RULES (CONT.)

A FR is a standalone soft-computing model for data processing.

Fuzzy rule structure

The FR is a symbolic expression that can be parsed into a tree. The tree consists of nodes and leafs (terminals). Three types of terminals are used:

1. *feature*, i.e a requirement on a data attribute
2. *past feature* i.e. a requirement on an attribute in a *previous record*
3. *past output* which puts a requirement on a *previous model output*

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

EVOLUTIONARY FUZZY RULES (CONT.)

A FR is a standalone soft-computing model for data processing.

Fuzzy rule structure

The FR is a symbolic expression that can be parsed into a tree. The tree consists of nodes and leafs (terminals). Three types of terminals are used:

1. *feature*, i.e a requirement on a data attribute
2. *past feature* i.e. a requirement on an attribute in a *previous record*
3. *past output* which puts a requirement on a *previous model output*

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

EVOLUTIONARY FUZZY RULES (CONT.)

FR operators are *and*, *or*, *not*, *prod*, and *sum*.

FRs use the standard t-norm and t-conorm for *and* and *or* operators and fuzzy complement for the *not* operator. Product t-norm is used for the *prod* operator and product t-conorm for the *sum* operator.

$$t(x, y) = \min(x, y) \quad (4)$$

$$s(x, y) = \max(x, y) \quad (5)$$

$$c(x) = 1 - x \quad (6)$$

$$t_{prod}(x, y) = xy \quad (7)$$

$$s_{prod}(x, y) = a + b - ab \quad (8)$$

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

Fuzzy rule evaluation

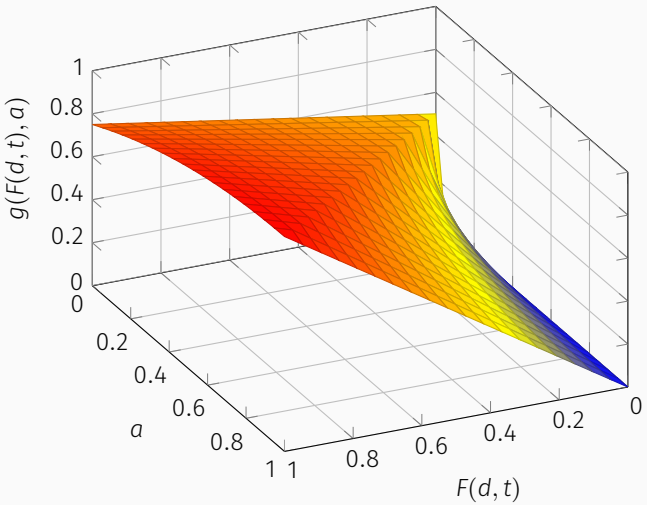
The weight of each node is in a FR evaluated using the Retrieval Status Value concept from fuzzy information retrieval.

In a FR , feature, f, with weight, a, is for a record, d, evaluated using a function $g : [0,1] \times [0,1] \rightarrow [0,1]$. Here, the threshold interpretation of a is used:

$$g(F(d,f),a) = \begin{cases} P(a)\frac{F(d,f)}{a}, & \text{for } F(d,f) < a \\ P(a) + Q(a)\frac{F(d,f)-a}{1-a}, & \text{for } F(d,f) \geq a \end{cases}. \quad (9)$$

In (9), $F(d,f)$, is weight of feature f in record d and $P(a) = \frac{1+a}{2}$ and $Q(a) = \frac{1-a^2}{4}$ are coefficients used to shape the threshold curve.

feature1:0.5 and:0.4 ...



Fuzzy rule evolution

The evolution of FRs is a straightforward application of *genetic programming* with IR measure F_β

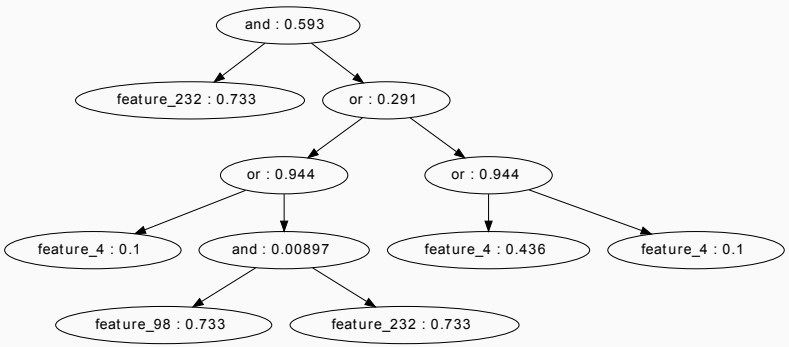
$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \qquad \beta \leq 0 \tag{10}$$

as fitness function.

It facilitates *supervised learning* of FRs from domain specific data and enables different flavours of data processing.

FR APPLICATIONS: PREDICTION OF STEEL PRODUCT QUALITY

FRs have been employed to discriminate between high and low-quality products of a steel processing plant. The production was described by four data sets with 508 features[1].



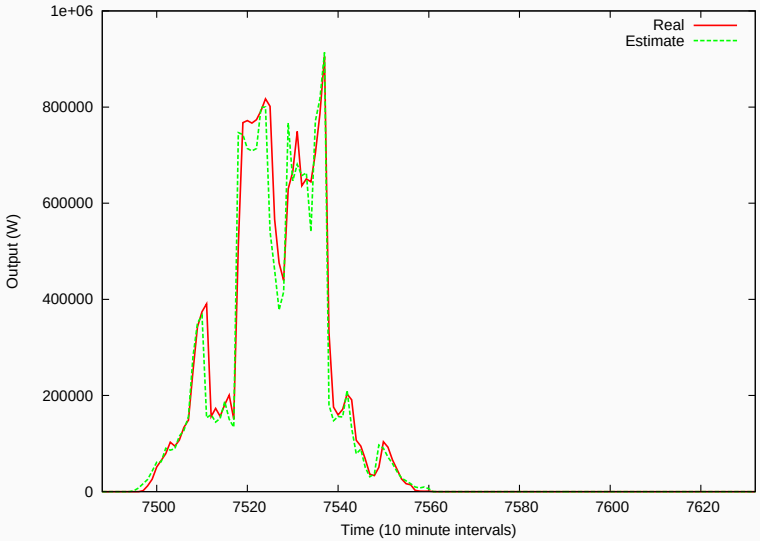
	Data set	
	D73226	D844, D1134, D3034
Correct [%]	96.99	...
False pos. [%]	0.43	...
False neg. [%]	2.58	...

In all cases, the evolved classifier reached good accuracy of 96.99 percent and above.

[1] **P. Krömer**, S. Owais, J. Platoš, and V. Snášel, "Towards new directions of data mining by evolutionary fuzzy rules and symbolic regression," Computers & Mathematics with Applications, vol. 66, no. 2, pp. 190-200, **2013**

FR APPLICATIONS (CONT.): PV POWER AVAILABILITY PREDICTION

FRs have been used to estimate the output of a photovoltaic power plant from environmental data (e.g. solar radiation) and compared with a range of traditional machine learning methods (multi-layer perceptrons, support vector machines)[2].



Data	Average prediction error (W)		
	FR	MLP1	MLP2, SVR RBF, SVR POLY
Full	13589.4	**13552.4**	...
Train	10001.4	**10000**	...
Test	**18313.2**	22033.2	...

[2] L. Prokop, S. Mišák, V. Snášel, J. Platoš, **P. Krömer**, "Supervised Learning of Photovoltaic Power Plant Output Prediction Models," Neural Network World, vol. 23, no. 4, pp. 321-338, **2013**

Summary

SUMMARY

The application of intelligent methods is one of the key trends addressing the increasing complexity and volume of problems in all domains. Nature–inspired approaches have have shown a great potential for dealing with hard tasks in an adaptive, metaheuristic way.

To-date applications provide strong empirical evidence that hybrid intelligent algorithms are useful and can leverage the efficiency of individual intelligent methods by a combination of their strong sides.

Evolutionary Fuzzy Rules are one example of a multi–paradigm method, built on the foundations of several well–established intelligent algorithms, with a good application record in industrial problems.

REFERENCES

1. P. Musílek, P. Krömer, J. Rodway, M. Prauzek: Pressure-based forecasting of next-day solar energy availability using evolutionary fuzzy rules. FUZZ-IEEE 2015: 1-8
2. L. Prokop, S. Mišák, V. Snášel, J. Platoš, P. Krömer, "Supervised Learning of Photovoltaic Power Plant Output Prediction Models," Neural Network World, vol. 23, no. 4, pp. 321-338, 2013, IF: 0.362
3. P. Krömer, S. Owais, J. Platoš, V. Snášel, "Towards new directions of data mining by evolutionary fuzzy rules and symbolic regression," Computers & Mathematics with Applications, Volume 66, Issue 2, August 2013, Pages 190-200, ISSN 0898-1221, 2013. Elsevier. IF: 2.069
4. P. Krömer, T. Novosad, Vá. Snásel, V. Vera, B. Hernando, L. García-Hernandez, Hé. Quintian-Pardo, E. Corchado, R. Redondo, J. Sedano and A. E. Garcia, " Prediction of Dental Milling Time-Error by Flexible Neural Trees and Fuzzy Rules " in H. Yin et al. (Eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2012 - 13th International Conference, Natal, Brazil, August 29-31, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7435, pp. 842-849, Springer, 2012