A Survey on Evolutionary Instance Selection and Generation

Joaquín Derrac, University of Granada, Spain

Salvador García, University of Jaén, Spain

Francisco Herrera, University of Granada, Spain

ABSTRACT

The use of Evolutionary Algorithms to perform data reduction tasks has become an effective approach to improve the performance of data mining algorithms. Many proposals in the literature have shown that Evolutionary Algorithms obtain excellent results in their application as Instance Selection and Instance Generation procedures. The purpose of this article is to present a survey on the application of Evolutionary Algorithms to Instance Selection and Generation process. It will cover approaches applied to the enhancement of the nearest neighbor rule, as well as other approaches focused on the improvement of the models extracted by some well-known data mining algorithms. Furthermore, some proposals developed to tackle two emerging problems in data mining, Scaling Up and Imbalance Data Sets, also are reviewed.

Keywords: *Data Reduction, Evolutionary Algorithms, Imbalanced Data, Instance Generation, Instance Selection, Prototype Generation, Prototype Selection, Scaling Up, Training Set Selection*

1. INTRODUCTION

Data reduction (Pyle, 1999) is one of the data preprocessing tasks which can be applied in a data mining process. The main objective in data reduction is to reduce the original data by selecting its most representative information. This way, it is possible to avoid excessive storage and time complexity, improving the results obtained by any data mining application, ranging from predictive processes (classification, regression) to descriptive processes (clustering, extraction of association rules, subgroup discovery).

Data reduction processes can be performed in many ways, some of the more remarkable being:

- Selecting features (Liu & Motoda, 2007), reducing the number of columns in a data set. This process is known as Feature Selection.
- Making the feature values discrete (Liu et al., 2002), reducing the number of possible values of features. This process is known as attribute Discretization.

DOI: 10.4018/jamc.2010102604

- Generating new features (Guyon et al., 2006) which describe the data in a more suitable way. This process is known as Feature Extraction.
- Selecting instances (Liu & Motoda, 2001; Liu & Motoda, 2002), reducing the number of rows in a data set. This process is known as Instance Selection (IS)
- Generating new instances (Bezdek & Kuncheva, 2001; Lozano et al., 2006), which describes the initial data set by generating artificial examples. This process is known as Instance Generation (IG).

This article discusses a wide number of IS and IG proposals. They can be divided into two types of techniques depending on the goal followed by the reduction. If the set of selected or replaced instances will be used as the reference data to instance-based classification, then we refer to Prototype Selection (PS) and Prototype Generation (PG). On the other hand, if the set of instances obtained will be used as input or training set of any data mining algorithm for building a model, then we refer to Training Set Selection (TSS).

In spite of the differences between PS and PG (the first one finds suitable prototypes, while the second one generates them), both have been mainly employed to improve the same classifier, the Nearest Neighbor rule (Cover & Hart, 1967; see also Papadopoulos & Manolopoulos, 2004; Shakhnarovich et al., 2006). This predicts the class of a new prototype by computing a similarity measure (Cunningham, in press) between it and all prototypes from the training set. In the k-Nearest Neighbors classifier, k nearest prototypes vote to decide the class of the new instance to classify. This algorithm is the baseline of the instance based learning field (Aha et al., 1991).

On the other hand, TSS consists of the selection of reduced training sets to improve the efficiency and the results obtained by any data mining algorithm. It has been mainly applied to improve the performance of decision trees, neural networks and subgroup discovery techniques. Although there exists a wide number of TSS approaches, no IG work on TSS has been reported yet, until our knowledge.

In recent years, the data mining community has identified some challenging problems in the area (Yang & Wu, 2006). Two of these are the *Scaling Up Problem* and the *Imbalance Data Sets Problem*. They are closely related to the data reduction field.

The *Scaling Up Problem* (Provost & Kolluri, 1999; Domingo et al., 2002) appears when an overwhelming amount of data must be processed, overcoming the capabilities of the traditional data mining algorithms. The *Imbalance Data Sets Problem* (Chawla et al., 2004; Batista et al., 2004) appears when the distribution of the class in the training data is not balanced, thus the number of instances of some classes is too low. This distribution can cause several problems in the classification of examples which belong to the minority classes.

Evolutionary Algorithms (Eiben & Smith, 2003) are general-purpose search algorithms that use principles inspired by natural genetic populations to evolve solutions to problems. The basic idea is to maintain a population of chromosomes which represent plausible solutions to the problem and evolve over time through a process of competition and controlled variation.

Evolutionary Algorithms have been successfully used in different data mining (Freitas, 2002; Ghosh & Jain, 2005; Abraham et al., 2006) and data reduction (Cano et al., 2003; Oh et al., 2004) problems. Given that the IS problem can be defined as a combinatorial problem, Evolutionary Algorithms have been used to solve it with promising results (Ho et al., 2002; García et al., 2008); these applications of Evolutionary Algorithms to tackle IS problems are usually called EIS (Evolutionary Instance Selection) methods. Furthermore, Evolutionary Algorithms have shown interesting behavior in their application to IG due to it can be defined as a parameter optimization problem (Fernández & Isasi, 2004; Nanni & Lumini, 2008).

The aim of this article is to present a review on the use of Evolutionary Algorithms for PS, PG and TSS algorithms, called EIS-PS, EPG and EIS-TSS, respectively, giving their description and main characteristics. Several evolutionary proposals developed to tackle the *Scaling Up Problem* and the *Imbalance Data Sets Problem* will be included in the review.

This article is organized as follows: Section 2 presents the definitions of the techniques and problems which will be reviewed in the rest of the article. Section 3 presents an overall analysis of several EIS-PS methods. Section 4 reviews the EPG contributions presented in recent years. Section 5 deals with EIS-TSS methods and their application to tackle different data mining problems. Section 6, concludes the survey.

2. PRELIMINARIES

This section provides some preliminary concepts and definitions about the techniques and problems shown in the rest of this article. Firstly, we describe the main characteristics of IS and IG. Secondly, we present the *Scaling Up Problem* and the *Imbalance Data Sets Problem* for classification, with reference to their relevance to data reduction. Finally, we briefly review some common features of the Evolutionary Algorithms approaches applied to tackle these problems.

Instance Selection and Generation

IS is one of the main data reduction techniques. In IS, the aim is to isolate the smallest set of instances which enable a data mining algorithm to predict the class of a query instance with the same quality as the initial data set (Liu & Motoda, 2001). By minimizing the data set size, it is possible to reduce the space complexity and decrease the computational cost of the data mining algorithms that will be applied later, improving their generalization capabilities through the elimination of noise.

More specifically, IS can be defined as follows: Let X_p be an instance where $X_p = (X_{p1,} X_{p2,}, ..., X_{pm,} X_{pc})$, with X_p belonging to a class *c* given by X_{pc} and a m-dimensional space in which X_{pi} is the value of the i-th feature of the p-th sample. Then, let us assume that there is a training set *TR* which consists of *N* instances X_p and a test set *TS* composed by *t* instances $X_{p.}$ Let $S \subset TR$ be the subset of selected samples that resulted from the execution of a IS algorithm, then we classify a new pattern from *TS* by a data mining algorithm acting over *S*. The whole data set is noted as *D* and it is composed of the union of *TR* and *TS*.

IS methods can be classified in two categories: PS methods and TSS methods. PS methods (Liu & Motoda, 2002) are IS methods which expect to find training sets offering best classification accuracy and reduction rates by using instance based classifiers which consider a certain similarity or distance measure. Recently, PS methods have increased in popularity within the data reduction field. Various approaches to PS algorithms have been proposed in the literature (see (Wilson & Martinez, 2000; Grochowski & Jankowski, 2004) for review). Figure 1 shows the basic steps of the PS process.

TSS methods are defined in a similar way. They are known as the application of IS methods over the training set used to build any predictive model (e.g. decision trees, neural networks …) Thus, TSS can be employed as a way to improve the behavior of predictive models, precision and interpretability (Riquelme et al., 2003). Figure 2 shows the basic steps of processing a decision tree (C4.5) on the TSS.

IG is another important technique in data reduction. It has been mainly applied to instance-based classifiers, thus we can focus on describing PG in depth. PG can be defined as the application of instance construction algorithms (Liu & Motoda, 2001) over a data set to improve the classification accuracy of a nearest neighbor classifier.

More specifically, PG can be defined as follows: Let X_p be an instance where $X_p = (X_{p1,} X_{p2,}, ..., X_{pm,} X_{pc})$, with X_p belonging to a class *c* given by X_{pc} and a m-dimensional space in which X_{pi} is the value of the i-th feature of the p-th sample. Then, let us assume that there is

Figure 1. Prototype selection process

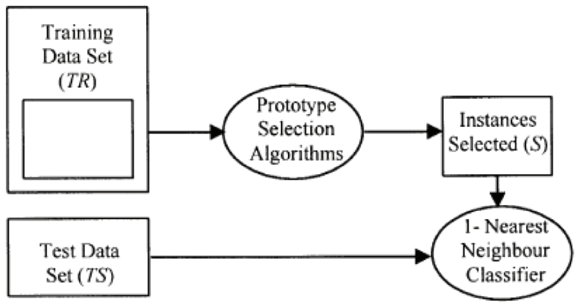
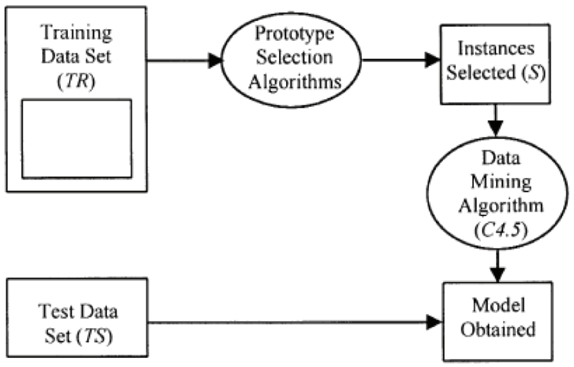


Figure 2. Training set selection process



a training set *TR* which consists of *N* instances X_p and a test set *TS* composed by *t* instances X_p. The purpose of PG is to obtain a prototype generate set, which consists of *r; r <n,* prototypes, which are either selected or generated from the examples of X_p. The prototypes of the generated set are determined to represent efficiently the distributions of the classes and to discriminate well when used to classify the training objects. Their cardinality should be sufficiently small to reduce both the storage and evaluation time spent by a nearest neighbor classifier.

A wide number of PG methods have been designed in the specialized literature, ranging from traditional ones (Chang, 1974; Kohonen, 1990), to more modern approaches (Lozano et al., 2006). It is important to point out the fact that the research on the EPG field has started recently, being (Fernandez & Isasi, 2004) the first proposal in applying Evolutionary Algorithms to perform a PG task, in contrast to the research in evolutionary IS, where the first proposal was made in (Kuncheva, 1995).

The Scaling Up and the Imbalance Data Sets Problems

IS methods are considered a useful tool to perform data reduction tasks, obtaining interesting results. They have also been employed successfully to tackle two emergent challenges in data mining, the *Scaling Up Problem* and the *Imbalance Data SetsProblem* (Yang & Wu, 2006).

The *Scaling Up Problem* appears when the number of training samples increases beyond the capacity of the traditional data mining algorithms, harming their effectiveness and efficiency. Due to large size data sets, it produces excessive storage requirement, increases times complexity and affects to generalization accuracy. Usually, when the input data set size affects the execution of the algorithms, it is possible to face this situation with two different strategies:

- **Scaling up the algorithm:** Proposing faster and lower consumption algorithms that can face large size data sets. (Provost & Kolluri, 1999)
- **Scaling down the data set:** In this case, the attention is directed toward the data set. The idea consists of modifying the data set by means of reductions to make it adequate to the original algorithm (Liu & Motoda, 2002).

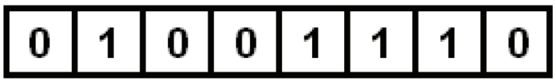
This problem has been recently addressed by many authors. An interesting example can be found in (Haro-García & García-Pedrajas, 2009), where a divide-and-conquer recursive approach to the problem is applied to very large problems, being able to match in accuracy and even improve on storage reduction the results of well-known standard IS algorithms with a very significant reduction in execution time. Another proposal, mainly adapted for use on evolutionary algorithms, is presented in (Cano et al., 2005), it will be described in the EIS-PS Section of this survey.

The *Imbalance Data Sets Problem* appears when the data contains many more examples of one class than the other and the less representative class represents the most interesting concept from the point of view of learning (Chawla et al., 2004). Imbalance in class distribution is pervasive in a variety of real-world applications, including but not limited to telecommunications (Tajbakhsh et al., 2009), web services, finance, ecology (Kubat et al., 1998), biology and medicine (Freitas et al., 2007).

Usually, in imbalanced classification problems, the instances are grouped into two types of classes: the majority or negative class, and the minority or positive class. The minority or positive class has more interest and it is also accompanied with a higher cost of misclassification. A standard classifier might ignore the importance of the minority class because its representation inside the data set is not strong enough. As a classic example, if the ratio of imbalance presented in the data is 1:100 (that is, there is one positive instance versus one hundred negatives), the error of ignoring this class is only 1%, so many classifiers could ignore it or not make any effort to learn an effective model for it.

Many approaches have been proposed to deal with the *Imbalance Data Sets Problem*. They can be divided into algorithmic approaches and data approaches. The first ones assume modifications in the operation of the algorithms, making them cost sensitive towards the minority class (Grzymala-Busse et al., 2005; Tajbakhsh et al., 2009). The data approaches modify the data distribution, conditioned on an evaluation function. Re-sampling of data could be done by means of under-sampling, by removing instances from the data (a process similar to IS) (Kubat & Matwin, 1997; Batista et al., 2004; Estabrooks et al., 2004), and over-sampling, by replicating or generating new minority examples (Chawla et al., 2002, Fernández et al., 2008).

Figure 3. A typical individual of an evolutionary IS algorithm



Basic Ideas on Evolutionary Algorithms for Instance Selection and Generation

A wide number of proposals presented on the use of Evolutionary Algorithms in IS and IG share some common characteristics regarding the two key concepts of any evolutionary algorithm: The representation of the population and the fitness function employed.

Representation: In most of the evolutionary IS algorithms, every member of the population encodes information about all the instances which are currently selected at each step of the search process.

This scheme is often used both in EIS-PS and EIS-TSS proposals. A binary representation is employed. Typically, every individual is defined as a binary string of length N, where each bit represents the current state of each instance of the training set (marked as '1' if the corresponding instance is currently selected, or '0' if not). Figure 3 shows a typical individual of an evolutionary IS algorithm.

On the other hand, in evolutionary IG algorithms every member of the population encodes information describing a new instance (or a new set of instances) to be generated. The concrete representation employed will vary depending on the codification of the problem data, although real coding is mostly preferred.

Fitness function: The majority of the evolutionary IS algorithms define a fitness function where two quality measures are employed: The accuracy of the results obtained by the subsequent data mining algorithm (e.g., a classifier), and the reduction rate achieved between the selected instances and the whole data set. Depending on the concrete method, this reduction rate can be computed by counting the number of instances selected, or employing more sophisticated methods, e.g. valuating the reduction as a measure of the interpretability of the tree obtained, when the evolutionary IS method is applied to improve the results of a decision tree algorithm.

In contrast, in evolutionary IG algorithms the only quality measure employed to define the fitness function is the accuracy obtained by employing the actual set of instances generated (typically, the accuracy obtained with a k-Nearest Neighbors classifier). The reduction rate is not usually employed because the number of instances generated is always small (usually it is fixed to a concrete value), thus every solution is expected to achieve a high reduction rate, and the search process can be focused only on the goal of increasing the accuracy of the classification process.

3. EVOLUTIONARY PROTOTYPE SELECTION

In this section, we will present the main contributions of EIS-PS appeared in the literature in recent years. Firstly, we give a snapshot on the *state of the art* in EIS-PS. Secondly, we describe in depth the characteristics of the most representative EIS-PS methods appeared. Thirdly, we show some EIS-PS proposals dealing with the *Scaling Up* and the *Imbalance Data Sets* problems. Finally, we conclude this section presenting some EIS-PS mixed approaches.

A Snapshot on Evolutionary Prototype Selection

The necessity of the Evolutionary Algorithms in PS is discussed in (Cano et al., 2003) where the authors differentiate between the selection based in heuristics (which appears in classic non-evolutionary PS algorithms, like for example CNN, IB3 or DROP described in (Wilson & Martinez, 2000)) and the selection developed by EIS-PS algorithms. EIS-PS presents a strategy that combines inner and boundary points. It does not tend to select instances depending on their a priori position in the search space (inner class or limit ones). EIS-PS selects the instances that increase the accuracy rates independently of their a priori position.

We will review the main contributions that have included or proposed an EIS-PS model in recent years. The first appearance of the application of an evolutionary algorithm to the PS problem can be found in (Kuncheva, 1995). Kuncheva applied a genetic algorithm to select a reference set for the k- Nearest Neighbors rule. Her genetic algorithm maps the training set onto a chromosome structure composed by genes, each one with two possible states (binary representation). The computed fitness function measures the error rate by application of the k-Nearest Neighbors rule. This genetic algorithm was improved in (Kuncheva & Bezdek, 1998; Ishibuchi & Nakashima, 1999).

At this point, all EIS-PS algorithms considered above adapt a classical genetic algorithm model to the PS problem. Later, a development of EIS-PS algorithms more conditioned to the problem is made. The first example of this can be found in (Sierra et al., 2001). In this article, an Estimation of Distribution Algorithm is used. Another example can be found in (Ho et al., 2002), where a genetic algorithm design for obtaining an optimal nearest neighbor classifier based on orthogonal arrays is proposed.

The technical term EIS-PS has been adopted by Cano et al. (2003), in which they analyze the behavior of different evolutionary algorithms, generational genetic algorithms (GGAs), steady-state genetic algorithms (SS-GAs), the CHC model (Eshelman, 1990) and Population Based Incremental Learning (PBIL) (Baluja, 1994) (which can be considered one of the basic Estimation of Distribution Algorithms). The fitness function used in these models combines two values: classification rate *clasRat* by using a 1-NN classifier and percentage reduction of prototypes of *S* with regards to *TR percRed*:

$$Fitness(S) = \alpha \cdot clasRat + (1-\alpha) \cdot perc\,\mathrm{Re}\,d$$

Where α is a weighting factor usually set to 0.5, and *perc_red* is defined as:

$$perc\,\mathrm{Re}\,d = 100 \cdot (|TR| - |S| / |TR|)$$

One of the newest approaches to EIS-PS employs memetic algorithms (Ong et al., 2007). These are heuristic searches in optimization problems that combine a population-based algorithm with a local search. The memetic algorithm employed by (García et al., 2008) incorporates an *ad hoc* local search specifically designed for optimizing the search in prototype selection problem with the aim of tackling the scaling up problem. Another recent proposal (Gil-Pita & Yao, 2008), is focused in the enhancing of the fitness function and the mutation and crossover operators when applied to PS problems.

Later research has gone further, focusing its interest on other topics apart from improving the design of EIS-PS algorithms. Several efforts have been focused on developing methods which will be able to tackle new challenges in data mining. A representative example can be found in (Cano et al., 2005), where an evolutionary method to tackle the *Scaling Up Problem* on the PS process is proposed. Another challenging problem addressed recently by using EIS-PS algorithms is the *Imbalanced Data Sets Problem* (García & Herrera, 2009).

A brief review will be done regarding mixed evolutionary approaches to IS with another data preprocessing technique, e.g. Feature Selection and Feature Weighting. Two different proposals will be reviewed, (Ros et al. 2008) being the first one. In this article, a hybrid genetic algorithm is applied to perform PS and Feature Selection simultaneously, trying to achieve three objectives simultaneously: Increasing the accuracy of the subsequent classification process, minimizing the dimensionality of the data (the number of features selected), and maximizing the number of instances selected.

The second proposal is an approach of data preprocessing with genetic algorithms in Case-Based Reasoning, which is described by the authors as an instance-based learning procedure (they also employ the k- Nearest Neighbors classifier). A basic genetic algorithm is conducted over a population of chromosomes which performs PS and Feature Weighting (which can be seen as a generalization of Feature Selection, where the weights are real valued between 0 and 1) simultaneously. This approach has been applied to two different scenarios: Customer classification and bankruptcy prediction modeling (Ahn et al., 2006; Ahn et al. 2009).

Overview on the EIS-PS Algorithms

In this subsection, we will describe in depth the characteristics of the most representative EIS-PS methods.

Generational Genetic Algorithm

Its basic idea is to maintain a population of chromosomes, which represent plausible solutions to the particular problem that evolves over successive iterations (generations) through a process of competition and controlled variation. Each chromosome in the population has an associated fitness to determine which chromosomes are to be used to form new ones in the competition process. The new chromosomes are created using genetic operators such as crossover and mutation.

In GGA, the selection mechanism produces a new population *P(t)* with copies of chromosomes in the old population *P(t-1)*. The number of copies received for each chromosome depends on its fitness; chromosomes with higher fitness have a greater chance of contributing copies to *P(t)*. Then, the crossover and mutation operators are applied to *P(t)*.

Algorithm 1 (See Figure 4) shows a basic pseudocode of GGA:

The GGA was the first scheme developed to perform EIS-PS processes. Although its results have been overcome by most of the subsequent proposals, it is still an important milestone in the field of PS.

Figure 4. Algorithm 1: Pseudocode of the GGA

```
Algorithm 1: GGA basic structure.
Input: A population.
Output: An optimized population.
1: Initialize population;
2: While Termination Criterion not satisfied do
3:     Evaluation of individual fitness;
4:     Formation of a gene pool (intermediate population) through selection
       mechanism;
5:     Recombination through crossover;
6:     Mutation operator;
7:     Replacement of the population;
8: end
```

Steady-State Genetic Algorithm

SSGA was firstly employed as an EIS-PS method in (Cano et al., 2003) In the SSGA usually one or two offspring are produced in each generation. Parents are selected to produce offspring and then a decision is made as to which individuals in the population will be selected for deletion in order to make room for the new offspring.

Algorithm 2 (See Figure 5) shows a basic pseudocode of SSGA:

In the construction of the SSGA, it is possible to select the replacement strategy (e.g., replacement of the worst, the oldest, or a randomly chosen individual) and the replacement condition (e.g., replacement if the new individual is better or unconditional replacement). A widely used combination is to replace the worst individual only if the new individual is better. Moreover, in (Goldberg & Deb, 1991), it is suggested that the deletion of the worst individuals can induce a high selective pressure, even when the parents are selected randomly.

This high selective pressure can help SSGA to improve the results of GGA in the performance of PS process. However, SSGA has been beaten also by most of the new proposals presented in recent years.

Population-Based Incremental Learning

PBIL (Baluja, 1994) is a specific Estimation of Distributions algorithm designed for binary search spaces. It attempts to explicitly maintain statistics about the search space to decide where to sample next.

The objective of the algorithm is to create a real valued probability vector V_p, which, when sampled, reveals high quality solution vectors with high probability. Initially, the values of V_p are set at 0.5. Sampling from this vector yields random solution vectors because the probability of generating a 1 or 0 for each gene is equal. As the search progresses, the values of V_p gradually shift to represent better solution vectors through the search process.

Algorithm 3 (See Figure 6) shows a basic pseudocode of PBIL algorithm.

The two basic search moves are performed in steps 6 and 8. In step 6, V_p is pushed toward S_{Best}. *LR* is the learning rate, which specifies how close the movement to S_{Best} is. In step 8, V_p is pushed far away from S_{Worse} . LR_{neg} is the negative learning rate, which specifies how far away the steps are from the worst solution.

PBIL can be seen as one of the most representative evolutionary proposals for performing PS process, because it is almost the only example of EIS-PS method which its search process is

Figure 5. Algorithm 2: Pseudocode of the SSGA

```
Algorithm 2: SSGA basic structure.
Input: A population.
Output: An optimized population.
1: Initialize population;
2: While Termination Criterion not satisfied do
3:     Select two parents from the population;
4:     Create one/two offspring using crossover and mutation;
5:     Evaluate the offspring with the Fitness function;
6:     Select one/two individuals in the population, which may be replaced by the
       offspring;
7:     Decide if this/these individuals will be replaced;
8: end
```

Figure 6. Algorithm 3: Pseudocode of the PBIL algorithm

```
Algorithm 3: PBIL algorithm basic structure.
Input: $V_p$.
Output: Optimized $V_p$.
1:  Initialize $V_p$;
2:  While Termination Criterion not satisfied do
3:      Generate a fixed number of samples based upon the probabilities specified
        in $V_p$;
4:      Select the best sample, $S_{Best}$, and the worst sample, $S_{Worst}$;
5:      Foreach I in $V_p$ do
6:          $V_p[i] = V_p[i] * (1-LR) + S_{Best}[i] * LR$;
7:          If $S_{Best}[i] != S_{Worst}[i]$ then
8:              $V_p[i] = V_p[i] * (1-LR_{neg}) + S_{Best}[i] * LR_{neg}$;
9:          end
10:     end
11: end
```

not based in a genetic algorithm. In general, it obtained good results when compared with genetic-based methods in the study carried in (Cano et al., 2003).

The Evolutionary Model CHC

CHC algorithm (Eshelman, 1990) is a binary-coded genetic algorithm which involves the combination of a selection strategy with a very high selective pressure, and several components inducing a strong diversity. CHC is a robust evolutionary algorithm, which should often offer promising results in several search problems.

The four main components of the algorithm are:

- **An elitist selection:** To compose a new generation, the best individuals among parents and offspring are selected.
- **A highly disruptive crossover:** HUX, which crosses over exactly half of the non-matching alleles.
- **An incest prevention mechanism**, which only allows to cross over those pairs of individuals which has a Hamming distance higher than a difference threshold. This threshold is decreased, as time goes by, to help the population to converge.
- **A restart process**, which is applied when the population has converged (when the threshold has dropped to zero). It generates a new population by randomly flipping a percentage (usually a 35%) of the bits of the old population individuals.

Algorithm 4 (See Figure 7) shows a basic pseudocode of CHC algorithm.

In the study carried out by (Cano et al., 2003), the CHC algorithm was selected as the best EIS-PS strategy, being able to outperform all the remaining methods of the study (the evolutionary and the non-evolutionary ones).

An interesting conclusion derived from that study was that the key feature of the CHC algorithm is its ability to select the most representative instances independently of their position in the search space, satisfying both the objectives of high accuracy and reduction rates. Due to this fact, CHC has been widely used as a baseline method to perform many evolutionary IS tasks. Some examples are shown in the next sections of this survey.

Figure 7. Algorithm 4: Pseudocode of the CHC algorithm

```
Algorithm 4: CHC algorithm basic structure.
Input: A population.
Output: An optimized population.
1:  Initialize population;
2:  While Termination Criterion not satisfied do
3:      Select candidates from the population;
4:      Generate offspring by crossing parents;
5:      Evaluate the offspring with the fitness function;
6:      Select the individuals of the new population
7:      If population not changed then
8:          Decrease threshold;
9:      end
10:     If threshold<0 then
11:         Restart population and reinitialize threshold;
12:     end
13: end
```

Intelligent Genetic Algorithm

Ho et al. (2002) propose the Intelligent Genetic Algorithm (IGA) based on Orthogonal experimental design used for PS and Feature Selection. Besides its initial definition, it can also be applied as a PS method only, without changing its initial objectives of increasing accuracy and reduction rates on the training data.

IGA is a GGA that incorporates an Intelligent Crossover operator. It builds an orthogonal array from two parents of chromosomes and searches within the array for the two best individuals according to the fitness function. It takes about $2^{\log_2 (\gamma-1)}$ fitness evaluations to perform an Intelligent Crossover operation, where γ is the number of bits that differ between both parents. Note that the application of Intelligent Crossover operator to large-size chromosomes (resulting chromosomes from large size data sets) could consume a high number of evaluations.

Algorithm 5 (See Figure 8) shows a basic pseudocode of IGA.

As their authors concluded, the employment of the Intelligent Crossover operator allows IGA to be superior to conventional genetic algorithms when applied to problems where the solution space is large and complex, e.g. when it is composed of high dimensional overlapping patterns. Thus, it is a good EIS-PS method to apply when facing medium and large sized data sets.

Steady-State Memetic Algorithm

The steady-state memetic algorithm (SSMA) was proposed in (García et al., 2008) to cover a drawback of the conventional EIS-PS methods that had appeared before: their lack of convergence when facing large problems.

SSMA makes use of a local search or meme specifically developed for this prototype selection problem. This interweaving of the global and local search phases allows the two to influence each other; i.e. SSGA chooses good starting points, and local search provides an accurate representation of that region of the domain. This local search scheme assigns a probability value to each chromosome generated by crossover and mutation, C_{new}:

$$P_{LS} = \begin{cases} 1 & if\ Fitness(C_{new})\ is\ better\ than\ Fitness(C_{worst}) \\ 0.625 & otherwise \end{cases}$$

Figure 8. Algorithm 5. Pseudocode of the IGA

```
Algorithm 5: IGA basic structure.
Input: A population.
Output: An optimized population.
1: Initialize population;
2: While Termination Criterion not satisfied do
3:     Evaluate all individuals using the fitness function;
4:     Use rank selection to select individuals to make a new population;
5:     Randomly cross some individuals by employing IC crossover;
6:     Select one/two individuals in the population, which may be replaced by the
       offspring;
7:     Apply the conventional bit inverse mutation operator to the population;
8: end
```

Figure 9. Algorithm 6: Pseudocode of the SSMA

```
Algorithm 6: SSMA basic structure.
Input: A population.
Output: An optimized population.
1: Initialize population;
2: While Termination Criterion not satisfied do
3:     Use binary tournament to select two parents;
4:     Apply crossover operator to create offspring ($Off_1$, $Off_2$);
5:     Evaluate $Off_1$ and $Off_2$;
6:     Foreach $Off_i$ do
7:         Invoke Adaptive-PLS-mechanism to obtain $PLS_i$
8:         If u(0,1) < $PLS_i$ then
9:             Perform meme optimization for $Off_i$;
10:        end
11:    end
12:    Employ standard replacement for $Off_1$ and $Off_2$;
13: end
```

Algorithm 6 (See Figure 9) shows a basic pseudocode of the SSMA.

Where u(0,1) is a value in a uniform distribution u[0,1] and the standard replacement means that a the worst individual is replaced only if the new individual is better.

The *Adaptive-PLS-mechanism* is an adaptive fitness-based method. A description of the *Adaptive-PLS-mechanism* and the meme specifically developed for the prototype selection task can be found in (García et al., 2008).

The meme optimization mechanism is a local search specifically designed for the PS problem. It tries to improve the initial chromosome by generating its neighbors by unselecting one of its current selected prototypes. These neighbors are evaluated by a special fitness function which is able to consume only partial evaluations, saving computational resources for the whole evolutionary process.

The objective of the meme optimization mechanism is adjusted dynamically in the execution of the SSMA. Every time a certain number of evaluations have been spent, the accuracy and reduction rates achieved by the best chromosome of the population are registered. If the classification accuracy has not increased, then the meme optimization starts an *improving accuracy* stage, where only better results in accuracy are accepted through the local search. On the other hand, if the reduction rate has not increased, then the meme optimization starts an *Avoiding premature convergence* stage, where the local search accepts worse solutions in order to improve the diversity of the population.

As their authors concluded, the SSMA presents a good reduction rate and computational time. In fact, it is able to outperform the classical PS algorithms, when the accuracy and reduction rates are considered. When compared to other EIS-PS methods, SSMA is able to outperform or equal them, being particularly useful as the size of the databases increases.

Genetic Algorithm Based on Mean Square Error, Clustered Crossover and Fast Smart Mutation

A new genetic algorithm (concretely a GGA model) was proposed in (Gil-Pita & Yao, 2008) as an EIS-PS model. This algorithm (no name was provided by their authors) included the definition of a novel mean square error based fitness function, a novel clustered crossover technique, and a fast smart mutation scheme.

The fitness function employed was based on a mean square error measure:

$$F = \frac{1}{NC}\sum_{n=1}^{N}\sum_{i=1}^{C}\left(\frac{K_n[i]}{K} - d_n[i]\right)^2$$

Where N is the number of training patterns, C is the number of classes, K is the number of nearest neighbors employed, K_n is the number of K nearest neighbors belonging to class *i*, and d_n is 1 when the desired output for the instance n is the class I, and 0 when not. As their authors stated, the error surface defined by this function is smoother than those obtained using counting estimator based functions, making easier the obtaining of its local minimum.

The clustering crossover operator firstly performs a k-means clustering process over the chromosomes, extracting the centroids of all the clusters found (the number of clusters is established randomly). Then, the centroids are employed with a classical random point cross operator to generate the individuals of the new generation.

The fast smart mutation procedure computes the effect of the change of one bit of the chromosome over its fitness value, testing all the possibilities. The change which produces the better fitness value is accepted as the result of the mutation operator. It is applied to every individual of the population. At the end of its application, the fitness value of the individuals is actualized, starting then a new generation of the evolutionary process.

Algorithm 7 (See Figure 10) shows a basic pseudocode of the algorithm.

The results obtained by the authors in the experimental study which was carried out suggested that the joint use of the three proposed methods could be quite interesting in the case of not very large training sets.

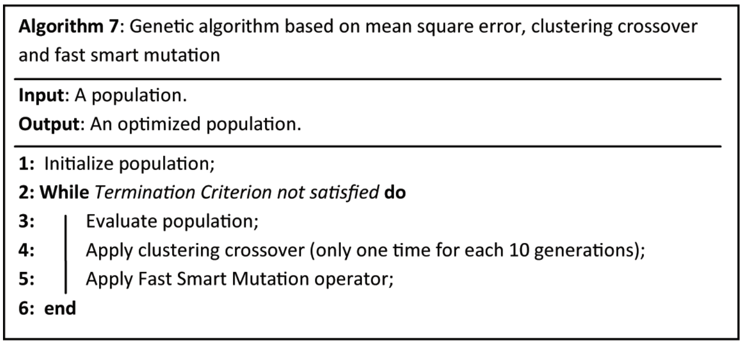
EIS-PS Proposals for the Scaling Up and Imbalanced Data Problems

In this subsection, we will analyze some EIS-PS proposals dealing with the *Scaling Up* and the *Imbalance Data Sets* problems.

Stratification of EIS-PS to Tackle the Scaling Up Problem

Cano et al. (2005) proposed a method to tackle the *Scaling Up Problem* in EIS-PS. The method presented consists of a stratified strategy which divides the initial data set into disjoint strata with equal class distribution. The number of strata chosen will determine their size, depending on the size of the data set. Using the proper number of strata the stratified method is able to significantly reduce the training set, avoiding the drawbacks of the *Scaling Up Problem*.

Figure 10. Algorithm 7: Pseudocode of the genetic algorithm based on mean square error, clustering crossover and fast smart mutation



Algorithm 7: Genetic algorithm based on mean square error, clustering crossover and fast smart mutation

Input: A population.
Output: An optimized population.

```
1: Initialize population;
2: While Termination Criterion not satisfied do
3:     Evaluate population;
4:     Apply clustering crossover (only one time for each 10 generations);
5:     Apply Fast Smart Mutation operator;
6: end
```

Figure 11 shows the basic steps of the process.

Following the stratified strategy, the initial data set *D* is divided into *t* disjoint sets D_j, strata of equal size, D_1, D_2, …, D_t maintaining class distribution within each subset. Then, PS algorithms will be applied to each D_j obtaining a selected subset DS_j. In this way, the subsets *TR* and *TS* will be obtained as follows:

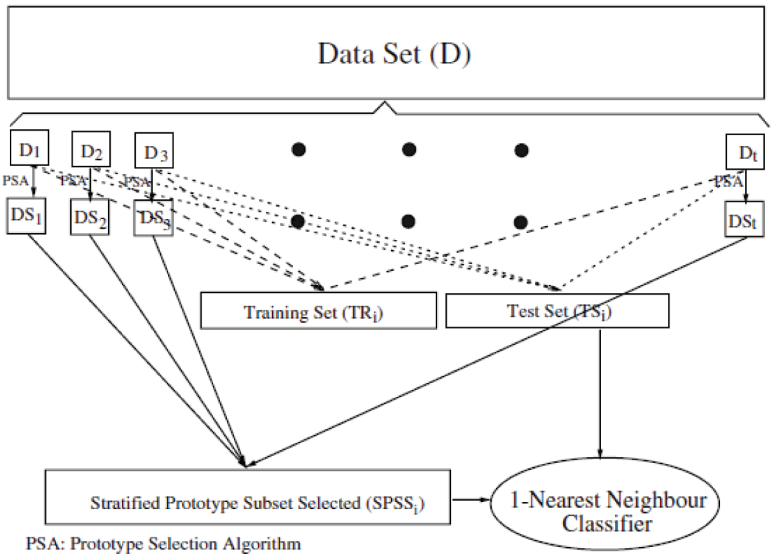
$$TR = \bigcup_{j \in J} D_j, J \subset \{1,2,..,t\} \quad TS = D - TR$$

And the Stratified Prototype Subset Selected (SPSS) is defined as:

$$SPSS = \bigcup_{j \in J} DS_j, J \subset \{1,2,..,t\}$$

The nearest neighbor classifier is then evaluated using as training data the *SPSS* set, and the *TS* set as test data. Thus the classifica-

Figure 11. Structure of the stratification process in EIS-PS


Data Set (D)
D1
D2
D3
Dt
PSA
PSA
PSA
PSA
DS1
DS2
DS3
DSt
Training Set (TRi)
Test Set (TSi)
Stratified Prototype Subset Selected (SPSSi)
1-Nearest Neighbour Classifier
PSA: Prototype Selection Algorithm

tion process can be performed on higher size data sets, avoiding the usual drawbacks and still achieving acceptable results.

The concluding remarks of the study were that a proper choice in the number of strata makes it possible to decrease significantly execution time and resources consumption, maintaining the EIS-PS algorithm's behavior in accuracy and reduction rates. Also, the CHC was selected as the best EIS-PS algorithm when employed within the evolutionary stratified PS process.

EIS-PS Algorithms on Imbalanced Data Problems

In (García & Herrera, in press), a complete study of EUS (Evolutionary Under-Sampling) algorithms is carried out. A set of EUS methods is proposed, which take into consideration the nature of the problem and use different fitness functions to obtain a good trade-off between balance of distribution of classes and performance.

Eight different algorithms compose the set of methods proposed in the study. Furthermore every method shares the same basic structure, which is developed by using the CHC algorithms as an evolutionary model. There are three characteristics that differentiate them:

- The objective that they pursue.
    - Aiming for an optimal balancing of data without loss of effectiveness in classification accuracy. EUS models that follow this tendency will be called Evolutionary Balancing Under-Sampling.
    - Aiming for an optimal power of classification without taking into account the balancing of data, considering the latter as a sub-objective that may be an implicit process. EUS models that follow this tendency will be called Evolutionary Under-Sampling guided by Classification Measures.
- The way that they do the selection of instances.
    - If the selection scheme proceeds over any kind of instance, then it is called Global Selection. That is, the chromosome contains the state of all instances belonging to the training data set and removals of minority class instances (those belonging to positive class) are allowed.
    - If the selection scheme only proceeds over majority class instances then it is called Majority Selection. In this case, the chromosome saves the state of instances that belong to the negative class and a removal of a positive or minority class instance is not allowed.
- The accuracy measure used for its fitness function.
    - Geometric Mean methods, if the geometric mean is used as accuracy measure. Geometric mean was first employed in (Barandela et al., 2003} and is defined as:

$$GM = \sqrt{a^{+} \cdot a^{-}}$$

Where a^{+} denotes accuracy in examples belonging to the minority class and a^{-} denotes accuracy in examples belonging to the majority class.

- Area Under the ROC Curve methods, if the Area Under the Curve measure is used instead the geometric mean. Area Under the ROC Curve (Bradley, 1997) can measure the efficacy of various classifiers simultaneously, employing the True Positive and False Positive rates of a classification process.

The eight methods were tested with several imbalanced datasets, and the results obtained were contrasted by using non-parametric statistical procedures. The main conclusions drawn from the study were:

- PS algorithms must not be used for handling imbalanced problems. They are prone to gain global performance by eliminating examples belonging to the minority class considered as noisy examples.
- During the evolutionary under-sampling process, the employment of a majority selection mechanism helps to obtain more accurate subsets of instances than the use of global selection. However, the latter mechanism is necessary to achieve the highest reduction rates.
- Data sets with a low imbalance ratio should be faced with Evolutionary Under-Sampling guided by Classification Measures models, and should use in particular the model with a global mechanism of selection and evaluation through the geometric mean measure.
- Data sets with a high imbalance ratio should be faced with Evolutionary Balancing Under-Sampling models, and should use in particular the model with a majority selection mechanism and evaluation through the geometric mean measure.

Mixed EIS-PS Approaches

In recent years, some proposals have appeared performing not only the PS process with evolutionary algorithms, but also performing another data preparation process simultaneously. This subsection will review two of the most remarkable approaches.

Ros et al. (2008) proposed a hybrid genetic algorithm (HGA) which performs IS and Feature Selection simultaneously. Its objectives are to increase the accuracy of the k- Nearest Neighbors over the reference set, to minimize the number of features selected (reducing the reference data) and to maximize the number of instances selected (to retain the most information possible without harming the classification accuracy).

The HGA is divided into three phases:

- A genetic algorithm is applied in the first phase. It includes a sophisticated selection scheme and some mechanisms to manage diversity and elitism (including an archive population and a dynamic analysis of the diversity of the population).
- By employing a histogram of the frequency with which each feature has been selected, a feature selection process is carried out, in order to simplify the problem.
- The genetic algorithm is applied again over the population. Also, some of the children generated by each generation are tuned by using local search procedures.

Despite the contradictory objectives in the number of instances and features selected, HGA is able to perform a dual IS and FS process with success, being a suitable evolutionary method to perform data reduction tasks.

A second mixed approach, GOCBR (Global Optimization of feature weighting and instance selection using genetic algorithms for Case Based Reasoning) was proposed in (Ahn et al., 2006; Ahn et al., 2009). This proposal performs a simultaneous IS and Feature Weighting process in the framework of a Case Based Reasoning system.

The search process of GOCBR consists of the application of a genetic algorithm with the common genetic operators (selection, crossover and mutation). Their individuals employ a binary representation, encoding the weights of the features by employing 14 bits to each one, and encoding the IS information in the second part of the chromosome by employing the usual binary scheme. Its fitness function measures only the accuracy obtained by employing the reference set defined by the chromosome to classify the train data in a k- Nearest Neighbors classifier.

GOCBR system has been applied successfully by the authors to various problems, such as customer classification or bankruptcy prediction modeling. Also, it is remarkable that is the only evolutionary method known of which performs a simultaneous IS and Feature Weighting process, until our knowledge.

4. EVOLUTIONARY PROTOTYPE GENERATION

In this section, we will present the main contributions of EPG appeared in the literature in recent years. In the first subsection, we give a snapshot on the *state of the art* in EPG. In the last subsection, we describe in depth the characteristics of the most representative EPG methods appeared.

A Snapshot on Evolutionary Prototype Generation

In recent years, the research efforts in the design of new PG techniques based on Evolutionary Algorithms have started to offer some interesting approaches. All of them still employ the NN rule as a reference classifier to measure the classification accuracy of the prototypes generated.

Usually, the prototypes are encoding as members of the population of the evolutionary algorithm employed to carry out the evolutionary process. A real codification scheme is used to represent them; each of its components has a concrete value for a concrete feature of the problem.

In this section, we will review the main contributions that have proposed an EPG model. The first contribution is (Fernandez & Isasi, 2004), where an evolutionary algorithm based on a two-dimensional grid is proposed, named Evolutionary Nearest Prototype Classifier. It defines a traditional evolutionary process to prepare the prototypes for its use on a 1-NN classifier, by employing a wide number of evolutionary operators.

The next two proposals are based on the Particle Swarm Optimization (PSO) Scheme (Kennedy et al., 2001). This technique is based on a set of potential solutions (particles) which evolves to find the global optimum of a real-valued function (fitness function) defined in a given space (search space). Particles represent the complete solution to the problem and move in the search space using both local information (the particle memory) and neighbor information (the knowledge of neighbor particles).

In (Nanni & Lumini, 2008) a PSO based PG method is proposed (no name is provided for the algorithm). It can be seen as a Pittsburgh-based model, because all the components of the solution (in this case, the prototypes generated) are encoded in a single particle. The method performs a PSO search process where a reduced set of prototypes is generated to finally perform a 1-NN classification process.

In (Cervantes et al., 2007; Cervantes et al, in press), inspired by the results of (Cervantes et al., 2005), an Adaptive Michigan PSO model for PG is proposed. This model is described as a Michigan approach because every particle contains only a component of the solution, thus the complete solution is built by joining the selected particles of the swarm. This approach also improves the traditional PSO scheme, because the online generation and destruction of particles is allowed in the search process.

Finally, the last proposal which will be reviewed, (Garain, 2008), is based on the Clonar Selection Algorithm (Castro & Zuben, 2002), a representation of an Artificial Immune System model (Dasgupta, 1998). Clonar Selection Algorithms are inspired by the behavior of the immune system when performing an immune response to an antigenic stimulus. It advocates the idea that only those cells that recognize the antigens proliferate, thus being selected against those which do not. This idea is employed to develop a PG system which is able to generate suitable prototypes to perform a 1-NN classification process.

Overview on the EIG Algorithms

In this subsection, we will describe in depth the characteristics of the most representative EIG methods.

Evolutionary Nearest Prototype Classifier

In (Fernandez & Isasi, 2004), an EPG method is proposed. It employs as a basic structure a two-dimensional matrix, where each row is associated with a prototype of the whole classifier, and each column is associated with a class to define regions where the prototypes are mapped.

On its initialization, the algorithm only defines one prototype. Then a whole evolutionary process starts: It carries out sequentially a set of evolutionary operations with the aim of generating a robust set of prototypes which will be able to correctly generalize the instances of the train set. The evolutionary operators defined are:

- **Mutation:** This operator is used to label each prototype with the most heavily populated class in each of its own regions.
- **Reproduction:** The reproduction operator function is to introduce new prototypes into the classifier, splitting the instances assigned to a prototype into two sets, where the second set is assigned to a new prototype.
- **Fight:** This operator allows prototypes to exchange their assigned instances. The fight can be performed in a cooperative or a competitive way, and it is ruled by the quality of the prototypes involved.
- **Movement:** The movement operator reallocates a prototype on the centroid of its assigned instances.
- **Die:** The current prototypes have a chance of being erased from the matrix, which is inversely proportional to its quality.

In the whole process, the quality of each prototype is defined by the number of prototypes which it has currently assigned and its classification accuracy. When the evolutionary process is finished, the generated set of prototypes is employed to classify the tests set, by means of the 1-NN classifier.

Algorithm 8 (See Figure 12) shows a basic pseudocode of Evolutionary Nearest Prototype Classifier algorithm.

The Evolutionary Nearest Prototype Classifier algorithm has shown good overall results, when compared against well-known classical methods in PG. Moreover, the results obtained when employing the prototypes generated with a 1-NN classifier were very competitive when compared with many classical instance-based classifiers as C4.5, Naïve Bayes or PART.

Particle Swarm Optimization for Prototype Generation

In (Nanni & Lumini, 2008) a PG method based on PSO is presented. This method defines the particles of the swarm as sets of a fixed number of prototypes, which are modified as the particle is moved in the search space.

Figure 12. Algorithm 8: Pseudocode of the Evolutionary Nearest Prototype Classifier algorithm

```
Algorithm 8: Evolutionary Nearest Prototype Classifier algorithm.
Input: Training set.
Output: A set of prototypes.
1: Initialization;
2: While Termination Criterion not satisfied do
3:     Collect information describing the prototypes;
4:     Mutation phase;
5:     Reproduction phase;
6:     Fight phase;
7:     Movement phase;
8:     Die phase;
9: end
```

The usual operators of PSO are employed by this proposal. The representation of each particle consists of a vector of length S= K · *M* given by the concatenation of *K* prototypes (dealing with *M*-dimensional data). The fitness function employed is defined as the classification error of the set of *K* prototypes over the training data.

Several runs (*N*) of the PSO process are carried out before finishing the PG stage (the authors recommend N=5). Each execution gives as result a reference set of *K* prototypes, being the result of the PG stage a collection of *N* reference sets. To classify a test instance, it is evaluated by each of the *N* reference sets, obtaining the final output as the result of a majority vote above all the reference sets. Thus, the proposed model can be seen as an ensemble of PSO-based classifiers.

Algorithm 9 (See Figure 13) shows a basic pseudocode of PSO-based PG algorithm.

The employment of the ensemble structure allows this proposal to obtain high accuracy rates. Furthermore, the authors suggested some ways to improve the model (like the employment of feature weighting methods). This proposal confirms that PSO is a very suitable model to perform PG processes.

An Adaptive Michigan Approach PSO for Nearest Prototype Classification

In (Cervantes et al., 2007; Cervantes et al., in press), an Adaptive Michigan Approach PSO is proposed. Its particles encode a prototype, each one being the generated train data represented as the whole particle swarm. This method does not have a fixed number of particles. On the contrary, some new operations are defined to allow the PSO search procedure to increase or decrease dynamically the number of particles.

The algorithm employs two different fitness functions: The global fitness function, defined by the standard classification accuracy on a 1-NN classifier, which is used to find the best swarm over the whole PSO procedure; and a local fitness function valued in each particle, defined by using the number of prototypes correctly classified and misclassified by itself. This secondary fitness function is used to evaluate the quality of each particle, in order to judge if it must be erased from the swarm, or if it can be employed as a parent of a new particle.

When the whole PSO process has finished, a cleaning process is carried out on the best swarm found. This swarm is the final output of the algorithm.

Algorithm 10 (See Figure 14) shows a basic pseudocode of the Adaptive Michigan PSO algorithm:

Figure 13. Algorithm 9: Pseudocode of the PSO-Based PG algorithm

```
Algorithm 9: PSO-based PG algorithm.
Input: Training set and test set.
Output: Classification output for the test set.
1: Initialization;
2: For j=1:N do
3:     PG(j)= PSO-PG(Train set_j);
4: end
5: Foreach x_i in test set do
6:     For j=1:N do
7:         PartialOutput(j)= Classify(x_i, PG(j))
8:     end
9:     Output(x_i)= VoteRule(PartialOutput);
10: end
```

This second PSO-based approach is focused in obtaining very high accuracy rates. The employment of a Michigan representation allows the definition of a local fitness function, which helps the method to find quickly suitable solutions in the search space.

Prototype Reduction Using an Artificial Immune Model

In (Garain, 2008), a PG based on a Clonar Selection algorithm is proposed. This model is composed of an immune memory which stores in its cells the best antigens found in the search process.

The Clonar Selection algorithm is initialized by representing the training instances as antigens, and choosing one antigen from each class to fill the immune memory. Then the search process starts. The first stage consists of a Hyper-mutation process. For each antigen on the training set, the most stimulating antigen in the immune memory is selected. The measure of stimulation is based on how close both antigens are (by means of Hamming or Euclidean Distance). The selected antigen of the memory is used as a parent for the Hyper-mutation process, which generates its offspring. Then, a Resource Allocation procedure is called, which balances the total number of clones present in the system by giving half of the resources to the clones of the class of the current antigen. The other half is equally divided among clones of other classes.

While the classification accuracy is improved by the generation of clones, further Mutation processes (and Resource Allocation procedures) are carried out on the surviving clones. This Mutation produces a lower number of clones which depends on the stimulation value of each parent clone. When no improvement is achieved, the best clone found is inserted into the immune memory, performing a replacement with the worst antigen present. Then a new generation starts.

Finally, when the algorithm meets a global termination criterion, the antigens contained in the immune memory are employed as the training set to classify the test instances, by using the 1-NN classifier.

Algorithm 11 (See Figure 15) shows a basic pseudocode of PG-Clonar Selection algorithm:

Figure 14. Algorithm 10: Pseudocode of the AMPSO algorithm

```
Algorithm 10: Adaptive Michigan PSO algorithm.
Input: Training set.
Output: A set of prototypes (best swarm).
1: Initialize swarm. Dimension of particles equals number of attributes;
2: Insert N particles of each class into the training patterns;
3: While iterations < MAX and accuracyRate < 100% do
4:      Check for particle reproduction and deletion;
5:      Foreach particle do
6:           Calculate local fitness;
7:           Calculate its next position;
8:      End
9:      Move the particles;
10:     Assign classes to the training patterns using the nearest particle;
11:     Evaluate the swarm classification accuracy;
12: End
13: Delete, from the best swarm found so far, the particles that can be removed
without a reduction in the classification accuracy.
```

The Clonar Selection Algorithm is a new method for PG based on a field of the Evolutionary Computation which has started to grown recently: The immune systems. Although it has high storage requirements to allocate the clones generated, it is a first example of a new technique which can obtain promising results with further research.

5. EVOLUTIONARY TRAINING SET SELECTION

In this section, we will present the main contributions of EIS-TSS appeared in the literature in recent years. In the first subsection, we give a snapshot on the *state of the art* in EIS-TSS. The next subsections will analyze the main approaches of EIS-TSS in decision trees, neural networks and subgroup discovery, respectively.

A Snapshot on Evolutionary Training Set Selection

The advances in EIS-TSS in recent years have been directed towards improving the results of some well-known data mining algorithms, being principally focused on the enhancement of the performance of decision trees, neural networks and subgroup discovery.

A wide range of these proposals have been inspired by the good results obtained by EIS-PS in the task of improving the performance of instance-based classifiers. Thus, some of the EIS-TSS methods which will be presented in this section will share some components with EIS-PS proposed before, adapting its principles to tackle the IS problems over other data mining algorithms.

Firstly, we will review some approaches of EIS-TSS applied to the construction of decision trees with the well-known C4.5 algorithm (Quinlan, 1993). A first application of IS to improve the construction of decision trees can be found in (Cano et al., 2003), where the results of the IS conducted by four evolutionary proposals are applied to extract reduced training sets in the construction of decision trees. Another proposal (Wu & Olaffson, 2006) also presented a genetic algorithm based IS process to improve the construction of decision trees, but they focused its effort on improving, not only the accuracy of the model and the reduction of the number of instances in the training set, but also in the interpretability and size of the trees obtained.

Later, Cano et al. (2007) presented two proposals of stratification to further improve the trees extracted by the C4.5 algorithm. The aim of both proposals was to improve the accuracy and interpretability of the trees extracted

Figure 15. Algorithm 11: Pseudocode of the PG-Clonar Selection algorithm

```
Algorithm 11: PG- Clonar Selection algorithm.
------------------------------------------------
Input: Training set.
Output: A set of prototypes (immune memory).
------------------------------------------------
1:  Initialization;
2: While Termination Criterion not satisfied do
3:      Proliferation I (Hyper-mutation);
4:      Resource Allocation;
5:      While resources left do
6:          Proliferation II (Mutation);
7:          Resource Allocation;
8:      end
9:      Insert best antigen into immune memory;
10: end
```

by means of stratification of the training data, trying to maintain a good trade-off between both quality measures.

As a last application of EIS-TSS in decision trees, we will review a proposal to improve the performance of C4.5 over imbalanced data sets (i.e. dealing with the *Imbalance Data Sets Problem* in the construction of decision trees) (García & Herrera, 2008). In this contribution it is shown how an evolutionary undersampling method is able to increase the accuracy of the decision tress in the classification of both majority and minority classes, employing the geometric mean accuracy measure.

Two approaches of EIS-TSS applied to neural networks will be analyzed. The first approach is (Ishibuchi et al., 2001), where a basic genetic algorithm is applied to perform both IS and Feature Selection processes to improve the results of standard three layered neural networks in classification.

A second approach to perform EIS-TSS on neural networks is presented in (Kim, 2006). This approach proposes the use of a standard genetic algorithm to perform a weight adjustment on the connections between the layers of a feed-forward neural network and an IS process over the instances which are employed to train the net.

Finally, the review of two proposals of EIS-TSS applied to subgroup discovery will be covered to close this section. The first proposal of this subsection (Cano, Herrera, Lozano & García, 2008) is an enhancement of the CN2-SD algorithm to increase its efficiency over large size datasets, by employing TSS techniques.

The second proposal (Cano et al., 2008) presents two stratification strategies to increase the presence of examples from minority classes in large size data sets with imbalanced data. The benefits shown in this study from the application of stratification includes the enhancement of Apriori-SD in its application to large size problems, and the improvement in the quality of the groups discovered over the minority classes of the problem analyzed.

EIS-TSS in Decision Trees

Cano et al. (2003) performed a complete study of the use of Evolutionary Algorithms to perform IS tasks. Although this work has been analyzed above, due to the number of EIS-PS which were presented in the study, it is important to note that a second experimental study was carried out employing the same IS algorithms as EIS-TSS methods to improve the trees built by C4.5.

The results obtained in the TSS part of the study were similar to the ones reached in the PS part: Evolutionary Algorithms based IS method were able to equal or outperform non-evolutionary methods, maintaining or increasing the accuracy of the trees obtained and increasing the reduction rates obtained by measuring the number of instances selected.

Wu & Olaffson (2006) performed a wide analysis of the application of IS to the induction of decision tress. They proposed a genetic algorithm to conduct the IS process, employing an integer codification in its chromosomes. Each individual is composed of a set of integer values, where each one represents one instance of the training set. Thus the selected instances of each individual are those whose integer identifier forms part of the chromosome.

The genetic algorithm search process employs a set of usual genetic operators: roulette wheel selection, crossover operator (which interchanges members of each set of instances defined by the parents) and mutation operator (which randomly replaces an instance by another one not selected).

The fitness function is a measure of the accuracy of the tree which can be induced by the chromosome, *S*, and its size. It is defined as follows:

$$Fitness(S) = -\log(e(\psi(S))) - a \cdot \log(\frac{size(\psi(S))}{K}),\ a > 1$$

Where e is an estimator of the error rate of the decision tree $\psi(S)$, K is an upper bound on the size of the tree, and a is a weighting factor.

The final tree is obtained by merging the instances selected at least one time with a fixed number of the best chromosomes in the population. Then the tree can be employed to classify new test instances.

In addition, a study of some relevant parameters is presented along with the results of the algorithm. A discussion of two additional measures, the Average Leaf Ratio (a measure of the number of instances in each leaf node of the tree), and the Instance Entropy of the training data, show some interesting conclusions, the most remarkable being:

- Genetic algorithm based IS is able to reduce the tree sizes with minimal loss in prediction accuracy
- The best results are obtained when the Average Leaf Ratio of the final model is higher
- Genetic algorithm based IS works better on low entropy data sets. Higher values of entropy make the construction of more complex decision trees necessary
- IS can be employed as a replacement for the traditional tree pruning techniques because it is able to outperform them when they are compared in terms of accuracy and tree size

A third remarkable proposal of EIS-TSS can be found in (Cano et al., 2007), where the use of stratification is proposed to tackle the *Scaling Up* problem on the induction of decision trees. The aim of the study was to perform the extraction of classification rules from large size data by keeping a good tradeoff between the precision and the interpretability of the model generated. To accomplish its objective, the authors present a stratified strategy (similar as the employed in (Cano et al., 2005).

To conduct the EIS-TSS process, the CHC algorithm is used. Moreover, two different fitness functions are employed; they are based on the usual fitness function used for EIS-PS:

$$Fitness(TSS) = \alpha \cdot clasPer + (1-\alpha) \cdot redPer$$

Where *clasPer* denotes the percentage of correctly classified objects from TR using only TSS to find the nearest neighbor or to extract a C4.5 model, depending on the concrete fitness function used. *redPer* denotes the reduction rate between TR and TSS.

- **Reduction-precision fitness function:** This fitness function uses the 1-Nearest Neighbor classifier for measuring the classification rate
- **Interpretability-precision fitness function:** This fitness function extracts a model with C4.5 to compute the classification performance of TSS

The two fitness definitions were tested with CHC and the stratification strategy. Several TSS methods were included in the experimental framework, to compare the performance of the proposed models against them. Finally, the results obtained were contrasted by using non-parametric statistical procedures.

The main conclusions reached in the study were as follows:

- The evolutionary stratified IS offers the best model size, maintaining an acceptable accuracy. It produces the smallest set of rules, with the minimal number of rules and the smallest number of antecedents per rule.
- The stratified CHC model with Interpretability-Precision fitness function allows us to obtain models with high test accuracy rates, similar to C4.5, but with the advantage that the size of the models are reduced considerably.
- The predictive model extraction by means of evolutionary stratified training set selection (with the CHC model and any of the fitness function presented) presents a good tradeoff between accuracy and interpretability. Thus, a very good scaling up behavior is observed, which allows us to obtain good results when the size of data set grows.

The last EIS-TSS proposal to improve the construction of decision trees which will be reviewed in this survey deals with the *Imbalance Data Sets* problem.

In (García & Herrera, in press) a new method is presented to deal with imbalanced data by performing the TSS process. The aim of the method is to improve the classification accuracy obtained by C4.5 when it is used on imbalanced data sets.

The proposed approach uses the same representation as the basic EIS-TSS methods. C4.5 is used to extract a model in order to compute the accuracy of the training set selected. The accuracy rates obtained by employing this model to classify the examples of the majority and minority classes are used to compute the geometric mean metric, which is used as fitness function.

Figure 16 shows the basic stages of the EUS process in EIS-TSS.

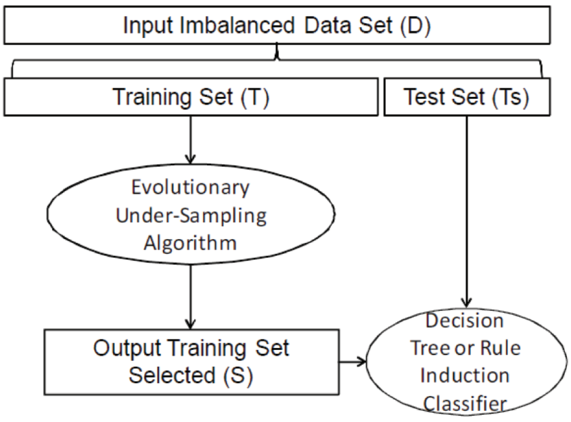
Although C4.5, in its standard definition, incorporates a pruning mechanism to avoid overfitting, the inclusion of the induction tree process within an evolutionary cycle can direct the resulting tree to an optimal model for training data, losing the generalization ability. To avoid this drawback, a simple and effective mechanism is incorporated. It consists of providing a higher weight for the classification costs of the instances that are not included in TSS than to the instances that are. Therefore, the reduction ability of the selected subset is encouraged, allowing the proposed approach to avoid overfitting in the construction of the models.

To test the performance of the proposal, it was compared to a wide number of well-known re-sampling algorithms, including OSS (Kubat & Matwin, 1997), NCL (Laurikkala, 2001) and SMOTE (Chawla et al., 2002), among others. The results obtained were contrasted by using non-parametric statistical procedures, finally showing that the proposed approach is able to outperform, or, at least, to behave similarly to every method in the comparison with respect to the accuracy of the models obtained, it obtains very accurate trees with a low number of rules or leafs. Thus, the proposed approach is confirmed as a very accurate method, which is able to increase the interpretability of the models obtained.

EIS-TSS in Neural Networks

A first application of the use of EIS-TSS on neural networks can be found in (Ishibuchi et al., 2001). The aim of this proposal was to find an optimal subset of instances and features by employing a GGA.

Figure 16. The EUS process in EIS-TSS


Input Imbalanced Data Set (D)
Training Set (T)
Test Set (Ts)
Evolutionary Under-Sampling Algorithm
Output Training Set Selected (S)
Decision Tree or Rule Induction Classifier

The GGA designed to carry out the TSS task employed the usual representation of the solutions (binary coded, employing the first part of the chromosome to code the instances which are currently selected, and the second part to code the features), and a standard set of genetic operators: Random selection of parents, uniform crossover and bit flipping mutation (biased to decrease the number of instances selected). The fitness function defined by this proposal was:

$$Fitness = W_{Perf} \cdot Perf - W_f \left|F\right| - W_P \cdot \left|P\right|$$

Where *Perf* is a measure of the accuracy of a NN classifier when applied to the training data by using as a reference set the current subset selected by the chromosome, *F* and *P* are respectively the number of features and instances selected, and W_{Perf}, W_F and W_P are user-defined weights.

When the evolutionary process was finished, the final subset selected was employed to train a standard three layered neural network, being finally validated with a test data set to test its generalization capability.

Although the concrete EIS-TSS method employed has been outperformed by other EIS-TSS and EIS-PS proposals, this approach can still be considered an important milestone in the field because it was the first application of EIS-TSS to improve the performance of neural networks.

A more modern approach is proposed in (Kim, 2006). In this proposal, a GGA is also employed to optimize the performance of a three layered neural network, in the framework of an application for financial forecasting. The individuals of the genetic algorithm encode information about the instances selected and about the adjustment of the weights of the neural network, employing binary coding in both cases. Although an EIS-TSS process is performed, the genetic algorithm does not have as its objective the maintenance of a good reduction rate. Instead, the fitness function is defined as the classification accuracy of the neural network defined by the chromosome, instead of the usual application of the neural network as a baseline classifier.

The experimental study carried out concluded that, in the context of the financial forecasting problem selected, the application of IS to improve the performance of neural networks outperformed the classical proposals of weights adjustment with genetic algorithms, highlighting the benefits of using EIS-TSS to improve the quality of the training process of neural networks.

EIS-TSS in Subgroup Discovery

In (Cano, Herrera, Lozano & García, 2008), a proposal to improve the performance of the CN2-SD algorithm for subgroup discovery in the evaluation of large size data sets is presented. Although CN2-SD is based on a divide and conquer strategy, it has to face the *Scaling Up* problem. To avoid it, the use of TSS algorithms is proposed for scaling down the data sets before the subgroup discovery task.

The study of the application of TSS algorithms, and the experiments that were carried out, was divided into two parts:

- In the first part, the effect of TSS on the subgroups discovered with CN2-SD in small data sets is studied. The objective is to analyze if the TSS process affects the descriptive qualitative measures of the subgroups (coverage, support, confidence, significance, unusualness, completeness, size and number of antecedents).

The basic IS methods applied to the TSS process were CNN (Hart, 1968), IB2, IB3 (Kibbler & Aha, 1987), DROP3 (Wilson & Martinez, 1997) ICF (Brightom & Mellish, 2002) and EIS-CHC. These methods were applied by using the stratification proposed in (Cano et al., 2005).

The results of this first part of the study were contrasted to a complete set of parametrical and non-parametrical statistical tests. Their application revealed that the use of TSS did not negatively affect the quality indexes of the subgroup discovered. Also the measures on size and number of antecedents were improved, showing that TSS algorithms were able to discover smaller and more interpretable sets of subgroups.

- In the second part, a TSS process is combined with CN2-SD to test its behavior in large size data sets. As basic IS methods, IB2 and EIS-CHC were selected because they were the IS algorithms with the smallest subsets selected in their application to the large size data sets.

The main conclusion of this part was that the combination of the highest reduction rates of IB2 and EIS-CHC with CN2-SD makes it possible to perform a SD task on large size data sets. In particular, EIS-CHC is recommended because it shows very good results in most of the qualitative measures tested, when employed in combination with CN2-SD.

As a final conclusion of the study, the authors stated that, thanks to the application of TSS methods, CN2-SD can be executed on large data set sizes pre-processed, maintaining and improving the quality of the subgroups discovered.

A second application of EIS-TSS for subgroup discovery can be found in (Cano et al., 2008). There, a different application of the stratified strategy presented in (Cano et al., 2005) was proposed: The employment of two modified strategies of stratification to increase the presence of minority classes. The aim of the proposal was to allow a subgroup discovery algorithm, Apriori-SD (Kavsek & Lavrac, 2006), to avoid the *Scalability problem* and to face a large data set without harming its accuracy due to a poor treatment of imbalanced data.

The data set used on the experiment was the KDD Cup'99. Firstly, it is shown that the Apriori-SD could not handle the KDD Cup'99 problem because of its expensive computational cost in time. Then the TSS methods are applied to tackle the problem. ENN (Hart, 1968), IB3 (Kibbler & Aha, 1987), and EIS-CHC (Cano et al., 2005) were proposed as baseline IS methods to be used.

To preserve the number of instances of the minority classes, two different strategies of stratification were proposed:

- **Instance selection in all classes:** The instances of the majority classes are assigned randomly over the strata created. Then the whole minority classes are added to each strata. After the IS process is carried out, the subsets selected are reunited, removing duplicities.

The employment of this strategy showed a severe drawback: Its application to every TSS subset obtained after the reunion of the instances selected from the strata decreased significantly the number of instances present from the minority classes. Thus, the minority classes were not sufficiently represented for a proper subgroup discovery task, due to use of the stratified IS.

- **Instance selection in majority classes:** The selection process is applied without the minority ones, just to the majority classes. The instances which belong to the minority classes were added to the TSS subset after the reunion of the subsets selected, and then the subgroup discovery tasks were carried out.

In this case, the instances which appear in the TSS selected were the most representative of the majority classes and all the instances belonging to the minority ones. Thus the IS process was able to reduce the initial data set without affecting the presence of instances from the minority classes, making the subgroup discovery process possible in those classes.

Both strategies were tested in combination with the selected IS methods, using confidence and support as qualitative measures. The conclusions of the experiment were that the combination of the TSS algorithms with stratification allows us to extract subgroups for most of the classes, including most of the minority ones, with high levels of confidence and support measures. Furthermore, the use of the *instance selection in majority classes* strategy of stratification was recommended to perform this task.

6. CONCLUSION

This article presents a review of PS, TSS and PG techniques performed by evolutionary algorithms. A wide number of algorithms and proposals of the *state-of-the-art* have been discussed, showing that the research in these fields have produced numerous advances in recent years to improve the quality of Instance Selection and Generation techniques in Data Mining.

Furthermore, this survey has considered the use of Evolutionary Algorithms to tackle two important issues in Data Mining: *The Scaling up Problem* and the *Imbalance Data Sets Problem*. These proposals have provided a way to improve the results obtained over large sized and imbalanced data in such fields as supervised classification and subgroup discovery, being a clear example of how Evolutionary Algorithms can be a useful tool in order to overcome these challenging problems.

ACKNOWLEDGMENT

This work was supported by TIN2008-06681-C06-01.

REFERENCES

Abraham, A., Grosan, C., & Ramos, V. (Eds.). (2006). *Swarm intelligence in data mining*. Berlin, Germany: Springer-Verlag.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37–66.

Ahn, H., & Kim, K. (2009). Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. *Applied Soft Computing*, *9*, 599–607. doi:10.1016/j.asoc.2008.08.002

Ahn, H., Kim, K., & Han, I. (2006). Hybrid genetic algorithms and case-based reasoning systems for customer classification. *Expert Systems: International Journal of Knowledge Engineering and Neural Networks*, *23*(3), 127–144. doi:10.1111/j.1468-0394.2006.00329.x

Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* (Tech. Rep. CMU-CS-94-163). Pittsburgh, PA: Carnegie Mellon University.

Barandela, R., Sánchez, J. S., García, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, *36*(3), 849–851. doi:10.1016/S0031-3203(02)00257-1

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Special Interest Group on Knowledge Discovery and Data Mining . SIGKDD Explorations*, *6*(1), 20–29. doi:10.1145/1007730.1007735

Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems*, *16*(12), 1445–1473. doi:10.1002/int.1068

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. doi:10.1016/S0031-3203(96)00142-2

Brightom, H., & Mellish, C. (2002). Advances in instance selection for instance based learning algorithms. *Data Mining and Knowledge Discovery*, *6*, 153–172. doi:10.1023/A:1014043630878

Cano, J. R., García, S., & Herrera, F. (2008). Subgroup discovery in large size data sets preprocessed using stratified instance selection for increasing the presence of minority classes. *Pattern Recognition Letters*, *29*, 2156–2164. doi:10.1016/j.patrec.2008.08.001

Cano, J. R., Herrera, F., & Lozano, M. (2003). Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transactions on Evolutionary Computation*, *7*, 561–575. doi:10.1109/TEVC.2003.819265

Cano, J. R., Herrera, F., & Lozano, M. (2005). Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters*, *26*, 953–963. doi:10.1016/j.patrec.2004.09.043

Cano, J. R., Herrera, F., & Lozano, M. (2007). Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability. *Data & Knowledge Engineering*, *60*, 90–100. doi:10.1016/j.datak.2006.01.008

Cano, J. R., Herrera, F., Lozano, M., & García, S. (2008). Making CN2-SD subgroup discovery algorithm scalable to large size data sets using instance selection. *Expert Systems with Applications*, *35*, 1949–1965. doi:10.1016/j.eswa.2007.08.083

Castro, L. N., & Zuben, F. V. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, *6*, 239–251. doi:10.1109/TEVC.2002.1011539

Cervantes, A., Galván, I., & Isasi, P. (2007). An adaptive michigan approach PSO for nearest prototype classification. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering* (LNCS 4528, pp. 287-296).

Cervantes, A., Galván, I., & Isasi, P. (in press). AMPSO: A new particle swarm method for nearest neighborhood classification. *IEEE Transactions on Systems, Man and Cybernetics, part B*.

Cervantes, A., Isasi, P., & Galván, I. (2005). A comparison between the pittsburgh and michigan approaches for the binary pso algorithm. In *Proceedings of the 2005 IEEE Congress on Evolucionary Computation,* Munchen, Germany (pp. 290-297).

Chang, C.-L. (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, *23*(11), 1179–1184. doi:10.1109/T-C.1974.223827

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM Special Interest Group on Knowledge Discovery and Data Mining . SIGKDD Explorations*, *6*(1), 1–6. doi:10.1145/1007730.1007733

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27. doi:10.1109/TIT.1967.1053964

Cunningham, P. (in press). A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*.

Dasgupta, D. (Ed.). (1998). *Artificial immune systems and their applications*. Berlin, Germany: Springer Verlag.

Domingo, C., Gavalda, R., & Watanabe, O. (2002). Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery*, *6*(2), 131–152. doi:10.1023/A:1014091514039

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Berlin, Germany: Springer Verlag.

Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. Rawlins (Ed.), *Foundations of genetic algorithms and classifier systems* (pp. 265-283). San Mateo, CA: Morgan Kaufmann.

Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, *20*(1), 18–36. doi:10.1111/j.0824-7935.2004.t01-1-00228.x

Fernández, A., García, S., del Jesus, M. J., & Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, *159*(18), 2378–2398. doi:10.1016/j.fss.2007.12.023

Fernández, F., & Isasi, P. (2004). Evolutionary design of nearest prototype classifiers. *Journal of Heuristics*, *10*, 431–454. doi:10.1023/B:HEUR.0000034715.70386.5b

Freitas, A. A. (2002). *Data mining and knowledge discovery with evolutionary algorithms*. New York: Springer-Verlag.

Freitas, A. A., da Costa Pereira, A., & Brazdil, P. (2007). Cost-sensitive decision trees applied to medical data. In *Data Warehousing and Knowledge Discovery* (LNCS 4654, pp. 303-312).

Garain, U. (2008). Prototype reduction using an artificial immune model. *Pattern Analysis & Applications*, *11*, 353–363. doi:10.1007/s10044-008-0106-1

García, S., Cano, J. R., & Herrera, F. (2008). A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition*, *41*(8), 2693–2709. doi:10.1016/j.patcog.2008.02.006

García, S., & Herrera, F. (in press). Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy. *Evolutionary Computation*.

Ghosh, A., & Jain, L. C. (Eds.). (2005). *Evolutionary computation in data mining*. Berlin, Germany: SpringerVerlag.

Ghosh, A., & Jain, L. C. (Eds.). (2005). *Evolutionary computation in data mining*. Berlin, Germany: Springer Verlag.

Gil-Pita, R., & Yao, X. (2008). Evolving edited k-nearest neighbor classifiers. *International Journal of Neural Systems*, *18*(6), 1–9. doi:10.1142/S0129065708001725

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. Rawlins (Ed.), *Foundations of genetic algorithms and classifier systems* (pp. 69-93). San Mateo, CA: Morgan Kaufmann.

Gómez-Ballester, E., Micó, L., & Oncina, J. (2006). Some approaches to improve tree-based nearest neighbour search algorithms. *Pattern Recognition*, *39*(2), 171–179. doi:10.1016/j.patcog.2005.06.007

Grochowski, M., & Jankowski, N. (2004). Comparison of instance selection algorithms II. Results and comments. In *Proceedings of Artificial Intelligence and Soft Computing - ICAISC 2004* (LNCS 3070, pp. 580-585).

Grzymala-Busse, J. W., Stefanowski, J., & Wilk, S. (2005). A comparison of two approaches to data mining from imbalanced data. *Journal of Intelligent Manufacturing*, *16*, 565–573. doi:10.1007/s10845-005-4362-2

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. (Eds.). (2006). *Feature extraction*. Heidelberg, Germany: Springer.

Haro-García, A., & García-Pedrajas, N. (2009). A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Mining and Knowledge Discovery*, *18*, 392–418. doi:10.1007/s10618-008-0121-2

Hart, P. E. (1968). The condesed nearest neighbour rule. *IEEE Transactions on Information Theory*, *18*(3), 431–433.

Ho, S.-Y., Liu, C. C., & Liu, S. (2002). Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Pattern Recognition Letters*, *23*(13), 1495–1503. doi:10.1016/S0167-8655(02)00109-5

Ishibuchi, H., & Nakashima, T. (1999). Evolution of reference sets in nearest neighbor classification. In *Selected papers from the Second Asia-Pacific Conference on Simulated Evolution and Learning on Simulated Evolution and Learning* (LNCS 1585, pp. 82-89).

Ishibuchi, H., Nakashima, T., & Nii, M. (2001). Learning of neural networks with GA-based instance selection. In *Proceedings of the 20th North American Fuzzy Information Processing Society International Conference,* Vancouver, Canada (Vol. 4, pp. 2102-2107).

Kavsek, B., & Lavrac, N. (2006). APRIORI-SD: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, *20*(7), 543–583. doi:10.1080/08839510600779688

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers.

Kibbler, D., & Aha, D. W. (1987). Learning representative exemplars of concepts: An initial case of study. In *Proceedings of the 4th International Workshop on Machine Learning,* Irvine, CA (pp. 24-30).

Kim, K. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*, *30*, 519–526. doi:10.1016/j.eswa.2005.10.007

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*, 1464–1480. doi:10.1109/5.58325

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, *30*(2-3), 195–215. doi:10.1023/A:1007452223027

Kubat, M., & Matwin, S. (1997). Addressing the course of imbalanced training sets: One-sided selection. In *Proceedings of the 14th International Conference on Machine Learning,* Nashville, TN (pp. 179-186).

Kuncheva, L. I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, *16*, 809–814. doi:10.1016/0167-8655(95)00047-K

Kuncheva, L. I., & Bezdek, J. C. (1998). Nearest prototype classification: Clustering, genetic algorithms, or random search? *IEEE Transactions on Systems, Man, and Cybernetics*, *28*(1), 160–164. doi:10.1109/5326.661099

Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe,* Cascais, Portugal (pp. 63-66).

Lavrac, N., Kavsek, B., Flach, P., & Todorovski, L. (2004). Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, *5*, 153–188.

Liu, H., Hussain, F., Lim, C., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, *6*(4), 393–423. doi:10.1023/A:1016304305535

Liu, H., & Motoda, H. (Eds.). (2001). *Instance selection and construction for data mining*. New York: Springer.

Liu, H., & Motoda, H. (2002). On issues of instance selection. *Data Mining and Knowledge Discovery*, *6*(2), 115–130. doi:10.1023/A:1014056429969

Liu, H., & Motoda, H. (Eds.). (2007). *Computational methods of feature selection*. New York: Chapman & Hall.

Lozano, M., Sotoca, J. M., Sánchez, J. S., Pla, F., Pekalska, E., & Duin, R. P. W. (2006). Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, *39*(10), 1827–1838. doi:10.1016/j.patcog.2006.04.005

Nanni, L., & Lumini, A. (2008). Particle swarm optimization for prototype reduction. *Neurocomputing*, *72*, 1092–1097. doi:10.1016/j.neucom.2008.03.008

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases*. Irvine, CA: University of California, Irvine, Department of Information and Computer Sciences. Retrieved from http://www.ics.uci.edu/~mlearn/MLRepository.html

Oh, I., Lee, J., & Moon, B. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1424–1437. doi:10.1109/TPAMI.2004.105

Ong, Y. S., Krasnogor, N., & Ishibuchi, H. (2007). Special issue on memetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics . Part B*, *37*(1), 2–5.

Papadopoulos, A. N., & Manolopoulos, Y. (2004). *Nearest neighbor search: A database perspective*. Berlin, Germany: Springer-Verlag.

Paredes, R., & Vidal, E. (2006). Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization. *Pattern Recognition*, *39*(2), 180–188. doi:10.1016/j.patcog.2005.06.001

Provost, F. J., & Kolluri, V. (1999). A survey of methods for scaling up inductive learning algorithms. *Data Mining and Knowledge Discovery*, *2*, 131–169. doi:10.1023/A:1009876119989

Pyle, D. (1999). *Data preparation for data mining*. San Francisco: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco: Morgan Kaufmann.

Riquelme, J. C., Aguilar, J. S., & Toro, M. (2003). Finding representative patterns with ordered projections. *Pattern Recognition*, *36*(4), 1009–1018. doi:10.1016/S0031-3203(02)00119-X

Ros, F., Guillaume, S., Pintore, M., & Chretien, J. R. (2008). Hybrid genetic algorithm for dual selection. *Pattern Analysis & Applications*, *11*, 179–198. doi:10.1007/s10044-007-0089-3

Sanchez, J. S., Barandela, R., Marques, A. I., Alejo, R., & Badenas, J. (2003). Analysis of new techniques to obtain quaylity training sets. *Pattern Recognition Letters*, *24*, 1015–1022. doi:10.1016/S0167-8655(02)00225-8

Sebban, M., Nock, R., Chauchat, J. H., & Rakotomalala, R. (2000). Impact of learning set quality and size on decision tree performances. *International Journal of Computers . Systems and Signals*, *1*(1), 85–105.

Shakhnarovich, G., Darrel, T., & Indyk, P. (Eds.). (2006). *Nearest-neighbor methods in learning and vision: Theory and practice*. Cambridge, MA: MIT Press.

Sierra, B., Lazkano, E., Inza, I., Merino, M., Larrañaga, P., & Quiroga, J. (2001). Prototype selection and feature subset selection by estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS. *Artificial Intelligence in Medicine* (LNAI 2101, pp. 20-29).

Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. *Applied Soft Computing*, *9*(2), 462–469. doi:10.1016/j.asoc.2008.06.001

Wilson, D. R., & Martinez, T. R. (1997). Instance pruning techniques. In *Proceedings of the 14th International Conference on Machine Learning,* Nashville, TN (pp. 403-411).

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, *38*, 257–286. doi:10.1023/A:1007626913721

Wu, S., & Olafsson, S. (2006). *Optimal instance selection for improved decision tree induction*. Paper presented at the 2006 IIE Annual Conference and Exhibition, Orlando, FL.

Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, *5*(4), 597–604. doi:10.1142/S0219622006002258

APPENDIX: ACRONYMS TABLE

In this appendix, a table with all the acronyms employed on the text is provided (Table 1). For each acronym, it is shown its meaning and the page where it was defined:

Table 1.

Acronym	Meaning	Page
EIS	Evolutionary Instance Selection	4
EIS-PS	Evolutionary Instance Selection – Prototype Selection	4
EIS-TSS	Evolutionary Instance Selection – Training Set Selection	4
EUS	Evolutionary Under Sampling	23
EPG	Evolutionary Prototype Generation	4
GGA	Generational Genetic Algorithm	10
GOCBR	Global Optimization of feature weighting and instance election using genetic algorithms for Case Based Reasoning	26
HGA	Hybrid Genetic Algorithm	26
IG	Instance Generation	3
IGA	Intelligent Genetic Algorithm	18
IS	Instance Selection	3
PBIL	Population Based Incremental Learning	10
PG	Prototype Generation	3
PS	Prototype Selection	3
PSO	Particle Swarm Optimization	28
SSGA	Steady State Genetic Algorithm	10
SSMA	Steady State Memetic Algorithm	18
TSS	Training Set Selection	3

Joaquín Derrac received the MSc degree in computer science from the University of Granada, Granada, Spain, in 2008. He is currently a PhD student in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. His research interests include data mining, lazy learning and evolutionary algorithms.

Salvador García received the MSc and PhD degrees in computer science from the University of Granada, Granada, Spain, in 2004 and 2008, respectively. He is currently an associate professor in the Department of Computer Science, University of Jaén, Jaén, Spain. His research interests include data mining, data reduction, data complexity, imbalanced learning, statistical inference and evolutionary algorithms.

Francisco Herrera received the MSc degree in mathematics in 1988 and the PhD degree in mathematics in 1991, both from the University of Granada, Spain. He is currently a professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has published more than 150 papers in international journals. He is coauthor of the book Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases *(World Scientific, 2001). As edited activities, he has co-edited five international books and co-edited twenty special issues in international journals on different soft computing topics. He acts as associated editor of the journals:* IEEE Transactions on Fuzzy Systesms, Mathware and Soft Computing, Advances in Fuzzy Systems, Advances in Computational Sciences and Technology, *and* International Journal of Applied Metaheuristic Computing. *He currently serves as area editor of the* Journal Soft Computing *(area of genetic algorithms and genetic fuzzy systems), and he serves as member of the editorial board of the journals:* Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, International Journal of Computational Intelligence Research, The Open Cybernetics and Systemics Journal, Recent Patents on Computer Science, Journal of Advanced Research in Fuzzy and Uncertain Systems, International Journal of Information Technology and Intelligent and Computing, *and* Journal of Artificial Intelligence and Soft Computing Research. *His current research interests include computing with words and decision making, data mining, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.*