A Preliminary Study on the Selection of Generalized Instances for Imbalanced Classification

Salvador García[1], Joaquín Derrac[2], Isaac Triguero[2],
Cristóbal Carmona[1], and Francisco Herrera[2]

[1] University of Jaén, Department of Computer Science, 23071 Jaén, Spain
sglopez@ujaen.es, ccarmona@ujaen.es
[2] University of Granada, Department of Computer Science and Artificial Intelligence,
18071 Granada, Spain
jderrac@decsai.ugr.es, isaaktriguero@gmail.com, herrera@decsai.ugr.es

Abstract. Learning in imbalanced domains is one of the recent challenges in machine learning and data mining. In imbalanced classification, data sets present many examples from one class and few from the other class, and the latter class is the one which receives more interest from the point of view of learning. One of the most used techniques to deal with this problem consists in preprocessing the data previously to the learning process.

This contribution proposes a method belonging to the family of the nested generalized exemplar that accomplishes learning by storing objects in Euclidean n-space. Classification of new data is performed by computing their distance to the nearest generalized exemplar. The method is optimized by the selection of the most suitable generalized exemplars based on evolutionary algorithms. The proposal is compared with the most representative nested generalized exemplar learning approaches and the results obtained show that our evolutionary proposal outperforms them in accuracy and requires to store a lower number of generalized examples.

1 Introduction

In the last years, the class imbalance problem is one of the emergent challenges in data mining [20]. The problem appears when the data presents a class imbalance, which consists in containing many more examples of one class than the other one, being the less representative class the most interesting one [4]. Imbalance in class distribution is pervasive in a variety of real-world applications, including but not limited to telecommunications, WWW, finance, biology and medicine.

Usually, the instances are grouped into two type of classes: the majority or negative class, and the minority or positive class. The minority or positive class is often of interest and also accompanied with a higher cost of making errors. A standard classifier might ignore the importance of the minority class because its representation inside the data set is not strong enough. As a classical example, if

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 601–610, 2010.

the ratio of imbalance presented in the data is 1:100 (that is, there is one positive instance in one hundred instances), the error of ignoring this class is only 1%.

A main process in data mining is the one known as data reduction [16]. In classification, it aims to reduce the size of the training set mainly to increase the efficiency of the training phase (by removing redundant data) and even to reduce the classification error rate (by removing noisy data). Instance Selection (IS) is one of the most known data reduction techniques in data mining.

The Nested Generalized Exemplar (NGE) theory was introduced in [17] and makes several significant modifications to the exemplar-based learning model. The most important one is that it retains the notion of storing verbatim examples in memory but, it also allows examples to be generalized. In NGE theory, generalizations take the form of hyperrectangles or rules in a Euclidean n-space. The generalized examples may be nested one inside another and inner generalized examples serve as exceptions to surroundings generalizations.

Several works argue the benefits of using generalized instances together with instances to form the classification rule [19,6,15]. With respect to instance-based classification [1], the use of generalizations increases the comprehension of the data stored to perform classification of unseen data and the achievement of a substantial compression of the data, reducing the storage requirements. Considering rule induction [10], the ability of modeling decision surfaces by hybridizations between distance-based methods (Voronoi diagrams) and parallel axis separators could improve the performance of the models in domains with clusters of exemplars or exemplars strung out along a curve. In addition, NGE learning allows capture generalizations with exceptions.

Evolutionary Algorithms (EAs) [7] are general purpose search algorithms that use principles inspired by nature to evolve solutions to problems. EAs have been successfully used in data mining problems [9,14]. Their capacity of tackling IS as a combinatorial problem is especially useful [3].

In this contribution, we propose the use of EAs for generalized instances selection in imbalanced classification tasks. Our objective is to increase the accuracy of this type of representation by means of selecting the best suitable set of generalized examples to improve the classification performance for imbalanced domains. We compare our approach with the most representative models of NGE learning: BNGE [19], RISE [6] and INNER [15]. The empirical study has been contrasted via non-parametrical statistical testing [5,11,12], and the results show an improvement of accuracy whereas the number of generalized examples stored in the final subset is much lower.

The rest of this contribution is organized as follow: Section 2 reviews the preliminary theoretical study. Section 3 explains the evolutionary selection of generalized examples. Section 4 describes the experimental framework used and presents the analysis of results. Finally, in Section 5, we point out the conclusions achieved.

2 Background and Related Work

This section shows the main topics of the background in which our contribution is based. Section 2.1 describes the evaluation framework of imbalanced classification.

Section 2.2 highlights the main characteristics of NGE theory and finally, Section 2.3 shows the EAs in which our model is based.

2.1 Evaluation in Imbalanced Classification

The measures of the quality of classification are built from a confusion matrix (shown in Table 1) which records correctly and incorrectly recognized examples for each class.

Table 1. Confusion matrix for a two-class problem

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

The most used empirical measure, accuracy, does not distinguish between the number of correct labels of different classes, which in the ambit of imbalanced problems may lead to erroneous conclusions. Because of this, more correct metrics are considered in imbalanced learning. Specifically, from Table 1 it is possible to obtain four metrics of performance that measure the classification quality for the positive and negative classes independently:

- **True positive rate** $TP_{rate} = \frac{TP}{TP+FN}$ is the percentage of positive cases correctly classified as belonging to the positive class.
- **True negative rate** $TN_{rate} = \frac{TN}{FP+TN}$ is the percentage of negative cases correctly classified as belonging to the negative class.
- **False positive rate** $FP_{rate} = \frac{FP}{FP+TN}$ is the percentage of negative cases misclassified as belonging to the positive class.
- **False negative rate** $FN_{rate} = \frac{FN}{TP+FN}$ is the percentage of positive cases misclassified as belonging to the negative class.

One appropriate metric that could be used to measure the performance of classification over imbalanced data sets is the Receiver Operating Characteristic (ROC) graphics [2]. In these graphics, the tradeoff between the benefits (TP_{rate}) and costs (FP_{rate}) can be visualized, and acknowledges the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary for the performance of learning algorithms.

The way to build the ROC space is to plot on a two-dimensional chart the true positive rate (Y axis) against the false positive rate (X axis) as shown in Figure 1. The points (0, 0) and (1,1) are trivial classifiers in which the output class is always predicted as negative and positive respectively, while the point (0, 1) represents perfect classification. To compute the AUC we just need to obtain the area of the graphic as:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{1}$$

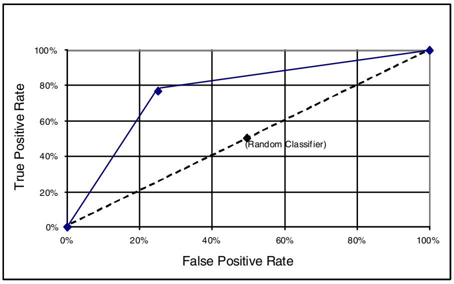

Fig. 1. Example of an ROC plot. Two classifiers are represented: the solid line is a good performing classifier whereas the dashed line represents a random classifier.

2.2 Nested Generalized Exemplar Theory

This subsection provides an overview on learning with generalized instances. First, we explain the needed concepts to understand the classification rule followed by this type of methods. After this, the three main proposals of NGE learning will be briefly described.

Matching and Classification. The matching process in one of the central features in NGE learning and it allows some customization, if desired. This process computes the distance between a new example and an generalized exemplar memory object. For remainder of this contribution, we will refer to the example to be classified as E and the generalized example stored as G, independently of G is formed by an unique instance or it has some volume.

The model computes a match score between E and G by measuring the Euclidean distance between two objects. The Euclidean distance is well-known when G is a single point. In case contrary, the distance is computed as follows (numeric attributes):

$$D_{EG} = \sqrt{\sum_{i=1}^{M} \left(\frac{dif_i}{max_i - min_i} \right)^2}$$

where

$$dif_i = \begin{cases} E_{f_i} - G_{upper} \text{ when } E_{f_i} > G_{upper} \\ G_{lower} - E_{f_i} \text{ when } E_{f_i} < G_{lower} \\ 0 \text{ otherwise} \end{cases}$$

M is the number of attributes of the data, E_{f_i} is the value of the ith feature of the example, G_{upper} and G_{lower} are the upper and lower values of G for a specific attribute and max_i and min_i are the maximum and minimum values for ith feature in training data, respectively.

The distance measured by this formula is equivalent to the length of a line dropped perpendicularly from the point E_{f_i} to the nearest surface, edge or corner of G. Note that points internal to a generalized instance have distance 0 to it. In the case of overlapping generalized examples, a point falling in the area of overlap belongs to the smaller instance, in terms of volume. The size of a hyperrectangle is defined in terms of volume. In nominal attributes, if the features are equal, the distance is zero, else it is one.

BNGE: Batch Nested Generalized Exemplar. BNGE is a batch version of the first model of NGE (also known as EACH [17]) and it is proposed to alleviate some drawbacks presented in it [19]. The generalization of examples is done by expanding its frontiers just to cover the desired example and it only merges generalized instances if the new generalized example dos not cover (or overlap with) any other stored example from any other classes. It does not permit overlapping or nesting.

RISE: Unifying Instance-Based and Rule-Based Induction. RISE [6] is an approach proposed to overcome some of the limitations of instance-based learning and rule induction by unifying the two. It follows similar guidelines explained above, but it furthermore introduces some improvements regarding distance computations and selection of the best rule using the Laplace correction used by many existing rule-induction techniques [10]).

INNER: Inflating Examples to Obtain Rules. INNER [15] starts by selecting a small random subset of examples, which are iteratively inflated in order to cover the surroundings with examples of the same class. Then, it applies a set of elastic transformations over the rules, to finally obtain a concise and accurate rule set to classify.

2.3 CHC Algorithm

We have studied its main characteristics to select it as the baseline EA which will guide the search process of our model. During each generation, the CHC algorithm [8] develops the following steps:

1. It uses a parent population of size R to generate an intermediate population of R individuals, which are randomly paired and used to generate R potential offspring.
2. Then, a survival competition is held where the best R chromosomes from the parent and offspring populations are selected to form the next generation.

CHC also implements HUX recombination operator. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. It also employs a method of incest prevention: Before applying HUX to two parents, the Hamming distance between them is measured. Only those parents who differ from each other by some number of bits (mating threshold) are mated. If no offspring is inserted into the new population then the threshold is reduced.

No mutation is applied during the recombination phase. Instead, when the search stops making progress the population is reinitialized to introduce new diversity. The chromosome representing the best solution found is used as a template to re-seed the population, randomly changing 35% of the bits in the template chromosome to form each of the other chromosomes in the population.

We have selected CHC because it has been widely studied, being now a well-known algorithm on evolutionary computation. Furthermore, previous studies like [3,13] support the fact that it can perform well on data reduction problems.

3 Selection of Generalized Examples Using the Evolutionary Model CHC

The approach proposed in this contribution, named Evolutionary Generalized Instance Selection by CHC (EGIS-CHC), is explained in this section. The specific issues regarding representation and fitness function complete the description of the proposal.

Let us assume that there is a training set TR with P instances and each one of them has M input attributes. Let us also assume that there is a set of generalized instances GS with N generalized instances and each one of the N generalized instances has M conditions which can be numeric conditions, expressed in terms of minimum and maximum values in interval $[0, 1]$; or they can be categorical conditions, assuming that there are v different values for each attribute. Let $S \subseteq GS$ be the subset of selected generalized instances resulted in the run of a generalized instances selection algorithm.

Generalized instance selection can be considered as a search problem in which EAs can be applied. We take into account two important issues: the specification of the representation of the solutions and the definition of the fitness function.

- *Representation:* The search space associated is constituted by all the subsets of GS. This is accomplished by using a binary representation. A chromosome consists of N genes (one for each sample in GS) with two possible states: 0 and 1. If the gene is 1, its associated generalized example is included in the subset of GS represented by the chromosome. If it is 0, this does not occur.
- *Fitness Function:* Let S be a subset of samples of GS and be coded by a chromosome. We define a fitness function based on AUC evaluated over TR through the rule described in Section 2.2.

  $$Fitness(S) = \alpha \cdot AUC + (1 - \alpha) \cdot red\_rate.$$

  AUC denotes the computation of the AUC measure from TR using S. red_rate denotes the ratio of generalized examples selected.

  The objective of the EAs is to maximize the fitness function defined. We preserve the value of $\alpha = 0.5$ used in previous works related to instance selection [3].

  The same mechanisms to perform a classification of a unseen example exposed in [17] are used in our approach. In short, they are:

- If no rule covers the example, the class of the nearest generalized instance defines the prediction.
- If various rules cover the example, the one with lowest volume is the chosen to predict the class, allowing exceptions within generalizations. The volume is computed following the indications given in [19].

– There is a detail not specified yet. It refers to the building of the initial set of generalized instances. In this first approach, we have used a heuristic which is fast and obtain acceptable results. The heuristic yields a generalization from each example in the training set. For each one, it finds the $K-1$ nearest neighbours being the Kth neighbour an example of different class. Then each generalization is built getting the minimal an maximal values (in case of numerical attributes) to represent the interval in such attribute or getting all the different categorical values (in case of nominal attributes) of all the examples belonging to its set of $K-1$ neighbours. Once all the generalizations are obtained, the duplicated ones are removed (keeping one representant in each case), hence $|GS| \leq |TR|$.

4 Experimental Framework and Results

This section describes the methodology followed in the experimental study of the generalized examples based learning approaches. We will explain the configuration of the experiment: used imbalanced data sets and parameters for the algorithms.

4.1 Experimental Framework

Performance of the algorithms is analyzed by using 18 data sets taken from the UCI Machine Learning Database Repository [18]. Multi-class data sets are modified to obtain two-class non-balanced problems, defining one class as positive and one or more classes as negative.

The data sets are sorted by their Imbalance Ratio (IR) values in an incremental way. IR is defined as the ratio between number of instances of the negative class divided by the number of instances of the positive class. Data sets considered have an IR lower than 9. The main characteristics of these data sets are summarized in Table 2. For each data set, it shows the number of examples (#Examples), number of attributes (#Attributes) and class name (minority and majority).

The data sets considered are partitioned using the *ten fold cross-validation (10-fcv)* procedure. The parameters of the used algorithms are presented in Table 3.

4.2 Results and Analysis

Table 4 shows the results in test data obtained by the algorithms compared by means of the *AUC* evaluation measure. It also depicts the number of generalized

Table 2. Summary Description for Imbalanced Data-Sets

Data-set	#Ex.	#Atts.	Class (min., maj.)	%Class(min.; maj.)	IR
Glass1	214	9	(build-win-non_float-proc; remainder)	(35.51, 64.49)	1.82
Ecoli0vs1	220	7	(im; cp)	(35.00, 65.00)	1.86
Wisconsin	683	9	(malignant; benign)	(35.00, 65.00)	1.86
Pima	768	8	(tested-positive; tested-negative)	(34.84, 66.16)	1.90
Glass0	214	9	(build-win-float-proc; remainder)	(32.71, 67.29)	2.06
Yeast1	1484	8	(nuc; remainder)	(28.91, 71.09)	2.46
Vehicle1	846	18	(Saab; remainder)	(28.37, 71.63)	2.52
Vehicle2	846	18	(Bus; remainder)	(28.37, 71.63)	2.52
Vehicle3	846	18	(Opel; remainder)	(28.37, 71.63)	2.52
Haberman	306	3	(Die; Survive)	(27.42, 73.58)	2.68
Glass0123vs456	214	9	(non-window glass; remainder)	(23.83, 76.17)	3.19
Vehicle0	846	18	(Van; remainder)	(23.64, 76.36)	3.23
Ecoli1	336	7	(im; remainder)	(22.92, 77.08)	3.36
New-thyroid2	215	5	(hypo; remainder)	(16.89, 83.11)	4.92
New-thyroid1	215	5	(hyper; remainder)	(16.28, 83.72)	5.14
Ecoli2	336	7	(pp; remainder)	(15.48, 84.52)	5.46
Glass6	214	9	(headlamps; remainder)	(13.55, 86.45)	6.38
Yeast3	1484	8	(me3; remainder)	(10.98, 89.02)	8.11

Table 3. Parameters considered for the algorithms

Algorithm	Parameters
BNGE	It has not parameters to be fixed
RISE	$Q = 1,\ S = 2$
EGIS-CHC	$Pop = 50,\ Eval = 10000,\ \alpha = 0.5$
INNER	Initial Instances= 10, MaxCycles= 5, Min Coverage= 0.95, Min Presentations= 3000, Iterations to Regularize= 50, Select Threshold= -50.0

instances maintained for each approach across all the data sets. The best case in each data set is remarked in bold.

Observing Table 4, we can make the following analysis:

- EGIS-CHC proposal obtains the best average result in *AUC* measure. It clearly outperforms the other techniques use in learning from generalized examples and 1NN.
- The number of generalized instances needed by EGIS-CHC to achieve such AUC rates is much lower than the needed by BNGE and RISE. In average, it also needs less generalized instances than INNER.

We have included a second type of table accomplishing a statistical comparison of methods over multiple data sets. Specifically, we have used the Wilcoxon Signed-Ranks test [5,11,12]. Table 5 collects results of applying Wilcoxon's test between our proposed methods and the rest of generalized instance learning algorithms studied in this paper over the 18 data sets considered. This table is divided into two parts: In the first part, the measure of performance used is the accuracy classification in test set through *AUC*. In the second part, we accomplish Wilcoxon's test by using as performance measure the number of generalized instances resulted for each approach. Each part of this table contains one column, representing our proposed methods, and N_a rows where N_a is the number of algorithms considered in this study. In each one of the cells can appear three symbols: +, = or -. They represent that the proposal outperforms (+), is similar (=) or is worse (-) in performance than the algorithm which appears in the row (Table 5). The value in brackets is the p-value obtained in the comparison and the level of significance considered is $\alpha = 0.10$.

Table 4. *AUC* in test data and number of generalized instances resulted from the run of the approaches used in this study

	AUC					number of generalized instances			
dataset	**1NN**	**BNGE**	**INNER**	**RISE**	**EGIS-CHC**	**BNGE**	**RISE**	**INNER**	**EGIS-CHC**
glass1	0.7873	0.6420	0.6659	0.6808	**0.7870**	74.40	63.80	17.30	**9.40**
ecoli0vs1	0.9630	0.9663	**0.9764**	0.9283	0.9708	10.40	53.60	5.00	**3.00**
wisconsin	0.9550	**0.9705**	0.9176	0.9351	0.9668	61.90	153.10	7.00	**2.90**
pima	0.6627	0.7099	0.6329	0.6499	**0.7223**	329.80	436.90	15.90	**14.80**
glass0	**0.8345**	0.7698	0.6579	0.7752	0.7579	67.50	69.80	34.00	**9.50**
yeast1	0.6262	0.6303	0.6467	0.6187	**0.7054**	817.00	780.30	**14.90**	16.00
vehicle1	0.6513	0.6143	0.5275	0.6499	**0.6872**	319.50	293.70	20.00	**17.20**
vehicle2	**0.9521**	0.8503	0.5293	0.9132	0.9122	155.30	133.90	26.40	**16.80**
vehicle3	0.6591	0.5559	0.5127	0.6414	**0.7119**	311.40	291.30	22.60	**17.10**
haberman	0.5618	0.5762	**0.5962**	0.5222	0.5924	211.30	132.40	17.20	**7.80**
glass0123vs456	0.9235	0.9175	0.8358	0.9199	**0.9410**	21.80	23.00	6.20	**4.60**
vehicle0	**0.9214**	0.6819	0.5342	0.8298	0.8962	214.90	166.70	37.70	**12.40**
ecoli1	0.7951	0.7983	0.8366	0.8440	**0.8759**	71.10	109.00	11.60	**5.10**
new-thyroid2	0.9819	0.9750	0.8861	0.9500	**0.9917**	12.50	26.60	3.50	**2.20**
new-thyroid1	**0.9778**	0.9208	0.9278	0.9583	**0.9778**	12.40	23.00	5.00	**2.20**
ecoli2	**0.9023**	0.8681	0.8528	0.8461	0.8821	66.60	100.20	7.00	**5.90**
glass6	0.9113	0.8613	0.7835	0.9086	**0.9534**	17.30	30.20	7.70	**3.30**
yeast3	0.8201	0.7613	0.8510	0.7588	**0.8725**	280.80	663.80	14.60	**12.10**
Average	0.8270	0.7817	0.7317	0.7961	**0.8447**	169.77	197.29	15.20	**9.02**

Table 5. Wilcoxon's test results over *AUC* and number of generalized instances resulted

algorithm	EGIS-CHC *AUC*	EGIS-CHC num. gen. instances
1NN	+ (.064)	
BNGE	+ (.000)	+ (.000)
RISE	+ (.000)	+ (.000)
INNER	+ (.000)	+ (.000)

We can see that the Wilcoxon test confirms the analysis carried out above.

5 Concluding Remarks

The purpose of this contribution is to present an evolutionary model developed to tackle data reduction tasks to improve imbalanced classification based on the nested generalized example learning. The proposal performs an optimized selection of previously defined generalized examples.

The results show that the use of generalized exemplar selection based on evolutionary algorithms can obtain promising results to optimize the performance in imbalanced domains.

Acknowledgement. This work was supported by TIN2008-06681-C06-01 and TIN2008-06681-C06-02.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Bradley, A.P.: The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition 30(7), 1145–1159 (1997)

3. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transactions on Evolutionary Computation 7, 561–575 (2003)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
6. Domingos, P.: Unifying instance-based and rule-based induction. Machine Learning 24, 141–168 (1996)
7. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
8. Eshelman, L.J.: The CHC adaptative search algorithm: How to safe search when engaging in nontraditional genetic recombination. Foundations of Genetic Algorithms, 265–283 (1991)
9. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, New York (2002)
10. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)
11. García, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694 (2008)
12. García, S., Fernandez, A., Luengo, J., Herrera, F.: A Study of Statistical Techniques and Performance Measures for Genetics-Based Machine Learning: Accuracy and Interpretability. Soft Computing 13(10), 959–977 (2009)
13. García, S., Herrera, F.: Evolutionary Under-Sampling for Classification with Imbalanced Data Sets: Proposals and Taxonomy. Evolutionary Computation 17(3), 275–306 (2009)
14. Ghosh, A., Jain, L.C.: Evolutionary Computation in Data Mining. Springer, Berlin (2005)
15. Luaces, O., Bahamonde, A.: Inflating examples to obtain rules. International Journal of Intelligent Systems 18(11), 1113–1143 (2003)
16. Pyle, D.: Data Preparation for Data Mining. The Kaufmann Series in DMS (1999)
17. Salzberg, S.: A nearest hyperrectangle learning method. Machine Learning 6, 151–276 (1991)
18. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
19. Wettschereck, D., Dietterich, T.G.: An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. Machine Learning 19, 5–27 (1995)
20. Yang, A., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making 5(4), 597–604 (2006)