

Derivation of Relational Fuzzy Classification Rules Using Evolutionary Computation

Vahab Akbarzadeh

Alireza Sadeghian

Marcus V. dos Santos

Abstract—An evolutionary system for derivation of fuzzy classification rules is presented. This system uses two populations: one of fuzzy classification rules, and one of membership function definitions. A constrained-syntax genetic programming evolves the first population and a mutation-based evolutionary algorithm evolves the second population. These two populations co-evolve to better classify the underlying dataset. Unlike other approaches that use fuzzification of continuous attributes of the dataset for discovering fuzzy classification rules, the system presented here fuzzifies the relational operators “greater than” and “less than” using evolutionary methods. For testing our system, the system is applied to the Iris dataset. Our experimental results show that our system outperforms previous evolutionary and non-evolutionary systems on accuracy of classification and derivation of interrelation between the attributes of the Iris dataset. The resulting fuzzy rules of the system can be directly used in knowledge-based systems.

I. INTRODUCTION

In the context of machine learning, classification is the task of predicting the value of one attribute (goal attribute) based on the value of other attributes (predicting attribute). Most common classification algorithms work as follows; the dataset to be mined is divided into two mutually exclusive sets, namely the training and the test set. Then the classification algorithm is applied to the training set, which includes the value of the predicting and the goal attributes. When the classification process is finished, prediction accuracy of the classification algorithm is tested with the test set [1]. Different algorithms have been proposed for classification tasks, some of which includes: Multi Layer Perceptron Artificial Neural Network, Bayesian Networks, and Decision Trees.

IF-THEN rules are one of the most popular ways of expressing explicit classification knowledge in data mining applications. Rules can express implicit knowledge of datasets in a form that is comprehensible for humans. High level comprehensible rules can help an expert through the decision making process. This process can be disease diagnosis, credit evaluation or any other classification process. The expert can combine the system rules with his own knowledge to make well-informed decisions. Classification methods based on IF-THEN rules are more comprehensible for the end-user than the black box methods, e.g., Artificial Neural Networks.

Real-world datasets are filled with inconsistent and vague information. By integrating fuzzy logic with a rule derivation system, the system will manage the vagueness and

uncertainty of the underlying information with more flexibility. Fuzzy systems include linguistic identifiers, i.e., “low”, “high”, “very”, that humans use for reasoning. Derivation of fuzzy rules with linguistic identifiers as terminals, improves the comprehensibility of rules for humans.

Genetic programming (GP) [2] is a powerful search method, which is applicable when a direct search is impossible. Classification of a dataset based on its attribute values is an example of a large search space. Holland [3] and Smith [4] initiated the application of evolutionary algorithms in classification problem. Their rule representation methods are referred to as Michigan and Pittsburgh approaches, respectively. In the Michigan approach, each individual includes only one rule, while Pittsburgh approach encodes a set of rules in one individual. It is now understood that the Pittsburgh approach has higher accuracy for complicated problems, while the Michigan approach converges faster. The following hybrid approaches are also worth mentioning: COGIN [5], REGAL [6], and G-Net [7]. More information regarding application of evolutionary algorithms for classification task can be found in [1].

Some authors have integrated fuzzy logic with evolutionary based classifiers [8], [9], [10], [11]. Chien et al. [9] proposed a classification algorithm based on discrimination functions and fuzzification of numerical attributes. The main problem with their discrimination function approach is that the end result is less understandable for the end-user compared to methods based on classification rules. Dounias et al. [12] designed a system to derive fuzzy classification rules from medical data. Kishore et al. [13] proposed a binary classification algorithm for multiclassification problem. In that approach, each run of the system is dedicated to classify one of the goal attribute classes, so the rest of the goal attributes are considered as one undesired class. At the end, all the classifiers compete with each other to classify the test set. Mendes et al. [14] proposed a co-evolutionary fuzzy classification system. Their system fuzzified each continuous attribute of the system using a membership function and specification of all the membership functions co-evolve with GP-based classification rule induction system.

In this paper, we propose a classification rule induction evolutionary system that derives relational fuzzy classification rules from the defined dataset. Our system is based on two populations. One population encodes the antecedent of inductive classification fuzzy rules, and the other one keeps a population of membership functions. The membership function population is used for the fuzzification of continuous attributes of the dataset and fuzzification of

Vahab Akbarzadeh, Alireza Sadeghian, and Marcus V. dos Santos are with the Department of Computer Science, Ryerson University, Toronto, Ontario, Canada (email: {vahab.akbarzadeh, asadeghi, m3santos}@ryerson.ca).

relational operators between the attributes of the dataset that has the same type. The first population is evolved using a GP algorithm whereas the second population is evolved based on a mutation-based evolutionary algorithm. These two populations co-evolve to better adapt to each other.

The main characteristic of our system that makes it different from previous works is that it derives relational fuzzy rules from dataset. We will fuzzify the relational operators of “greater than” and “less than” for two variable valued operands, i.e., dataset attributes in the dataset. As will be shown, this improves the classification performance of our algorithm and helps us to derive more information about the interrelation between the attributes of the mined dataset.

II. RELATIONAL FUZZY CLASSIFICATION RULE DERIVATION (RFCRD) SYSTEM

Our system derives fuzzy classification rules from any defined dataset. This system is based on the evolutionary computation methods. A genetic programming algorithm evolves a set of fuzzy classification rules, and a simple mutation-based evolutionary algorithm calculates two sets of membership functions to fuzzify continuous attributes and relational operators between the attributes of the dataset, as will be explained below. These two sets of rules and membership function definitions will co-evolve to derive a set of rules and function definitions that fit together on predicting the class of each example.

A. Fuzzy Classification Rule Population

Classification is the task of assigning a class label C_i from the set of class labels $\{C_1 \dots C_n\}$ to each example of the dataset. Each example of the dataset can be represented as a point in an N dimensional space, where N represents the number of predicting attributes of each example. The main task of the classifier system is to find a mapping function S such that:

$$S : R^N \rightarrow C_i \quad (1)$$

Classification rules are one of the common methods to represent the mapping function of a classifier system. The classification rules have the form IF (antecedent)-THEN (consequent). The antecedent of the classification rule includes a logical expression among the predicting attributes of the dataset, and the consequent has the form $Class = C_i$. If the predicting attribute values of an example in the dataset satisfy the antecedent of a rule, then the class in the consequent is assigned to that example.

Fuzzy logic can be integrated with classification rules. The resulting system’s rules are more comprehensible for human and also can better manage the uncertainty of the underlying dataset. In the fuzzy classification rule-based system the antecedent of each rule is a fuzzy logic expression that contains a set of fuzzy expressions connected by fuzzy operators. So far, only logical operators of $\{AND, OR, NOT\}$ have been used as fuzzy operators. We fuzzified the relational operators $\{>, <\}$ for two variable valued operands. More precisely, the fuzzy expression $Att_i > Att_j$ return a value between $[0,1]$. This value shows the amount that Att_i is bigger than Att_j .

1) *Rule representation:* We followed the Pittsburgh approach for rule representation [4]. In this population the function set contains the fuzzy logic operators $\{AND, OR, NOT, >, <, =\}$. We used the standard fuzzy operators for $\{AND, OR, NOT\}$ [15]. More precisely, if $\mu_A(x)$ represents the membership degree of element x in fuzzy set A , then the fuzzy operators are defined as follows:

$$\mu_A(x) AND \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (2)$$

$$\mu_A(x) OR \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \quad (3)$$

$$NOT \mu_A(x) = 1 - \mu_A(x) \quad (4)$$

Disjunctive Normal Form (DNF) is a logical form which is the disjunction of conjunction clauses. In our system the antecedent of the classification rules are represented in DNF. In order to show the antecedent of the formulas in DNF, we propose some restrictions on the structure of the individuals. These restrictions are: (a) The root of each tree is an *OR* node. Each *OR* node can have one or more than one *AND* node (rule) as its child(ren). (b) The second level of the tree includes *AND* nodes (rules). Each *AND* node can have one or more than one terminal node as its child(ren). (c) The third level is the *Terminal* level. Terminals of the system are defined using the following rules:

$\langle terminal \rangle ::= NOT \langle literal \rangle$
 $\langle terminal \rangle ::= \langle literal \rangle$
 $\langle literal \rangle ::= \langle op \rangle \langle relational operator \rangle \langle op \rangle$
 $\langle literal \rangle ::= \langle op \rangle \langle linguistic value \rangle$
 $\langle relational operator \rangle ::= \{>, <\}$
 $\langle linguistic value \rangle ::= \{Low, Med, High\}$
 $\langle op \rangle ::= \{Att_1 \dots Att_N\}$

Note that in the case of relational operator, two operands of the operator should have the same type, so the values of them can be compared with each other. (d) No more than one child of an *AND* node can refer to a terminal with the same attribute as the first operand and the = as the operator. For example, two terminals of $Temp = Low$ and $Temp = High$ cannot be the children of the same *AND* node. A sample rule set of the system is shown in figure 1. This rule set is equal to: $IF((Att_1=Low) AND (Att_2=High)) OR ((Att_3 < Att_1) AND (NOT Att_1 > Att_2) AND (Att_3=Med))$

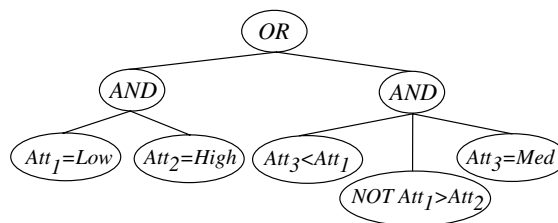


Fig. 1. Sample rule set of the system

2) *Fitness function:* We used binary classification in our system, so at each run of the system, all the rules of the system are trying to predict only one class of the goal attribute. We used tournament selection with tournament size

2. Fitness of each individual is calculated using the following formula [16]:

$$fitness = \left(\frac{TP}{TP + FP}\right) \times \left(\frac{TN}{TN + FN}\right) \quad (5)$$

where,

- TP (true positive) is the number of examples that satisfied the antecedent of one of the rules of the individual and the examples have the same class as the consequent of the rule.
- FP (false positive) is the number of examples that satisfied the antecedent of one of the rules of the individual but the examples do not have the same class as the consequent of the rule.
- TN (true negative) is the number of examples that did not satisfy the antecedent of any of the rules of the individual and the examples do not have the same class as the consequent of the rule.
- FN (false negative) is the number of examples that did not satisfy the antecedent of any of the rules of the individual but the examples have the same class as the consequent of the rule.

It is generally assumed that the above values are crisp numbers, but since we used fuzzy classification rules in our system, antecedents of each classifying rule return a fuzzy value when trying to classify each example of the dataset. In other words, each example satisfies the antecedent of each rule to a certain degree between [0, 1]. We could either defuzzify the returning values of the rules or fuzzify value of TP, TN, FP and FN. We fuzzified value of TP, TN, FP and FN with the same approach proposed in [14].

3) *Mutation*: Regarding mutation, a *Terminal* node is randomly selected in the individual and its value is swapped with a randomly selected terminal value.

4) *Crossover*: We used a modified version of cut and splice method [17] for crossover. The method works as follows:

- A node is randomly selected in parent one.
 - If the selected node is an *AND* node, one other *AND* node is randomly selected in parent two, and all the *AND* nodes beyond the two selected *AND* nodes are swapped between the two parents.
 - If the selected node is a *Terminal* node, one other *Terminal* node is randomly selected in parent two, and all the *Terminal* nodes beyond the two selected *Terminal* nodes are swapped between the rules in two parents.

B. Membership Function Population

The second population keeps the encoding of the membership functions that the first population uses for its rule interpretation. Each individual has a predefined static structure that encodes the membership function for all continuous variables and relational operators between each two attributes of the dataset that has the same type. Each continuous variable is divided into three linguistic variables namely,

{*Low, Med, High*}. This process is shown in figure 2. Each individual of the population keeps the values for the parameters of *mc1, mc2, mc3* and *mc4*. These values evolve to better adapt to their corresponding attributes. The system ensures that after each evolution $mc1 < mc2 < mc3 < mc4$.

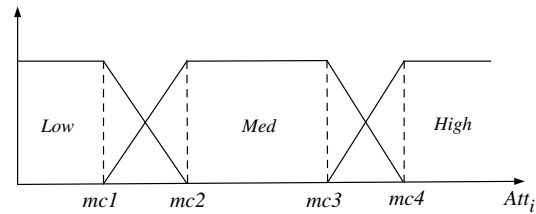


Fig. 2. Membership function for continuous attributes

As mentioned before, the main characteristic of our system is fuzzification of relational operators. For this purpose, the relation between each two variables of the system that have the same type is fuzzified through the definition of membership function for “greater than” operator. This process is shown in figure 3. In this figure, the difference between the values of two attributes (i.e. $Att_i - Att_j$) is the main parameter that determines the degree that Att_i is bigger than Att_j . Note that in our system the expression $Att_i > Att_j$ represents the degree that Att_i is bigger than Att_j , so the value of $NOT Att_i > Att_j$ shows how much Att_i is “not much bigger” or “less bigger” than Att_j . Also note that the “less than” operator is only a different representation of “greater than” operator. More precisely, the value that Att_i is bigger than Att_j is equal to the amount that Att_j is less than Att_i , so there is no need to explicitly fuzzify the less than operator in the system. Like the membership function of continuous variables, the system evolves the value of *mr1* and *mr2* and ensures that $mr1 < mr2$.

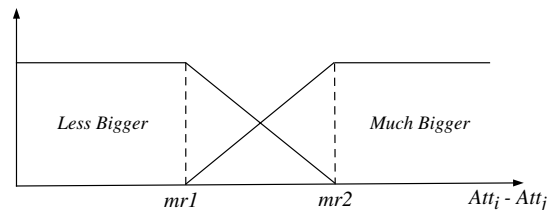


Fig. 3. Membership function for “greater than” operator

We used a mutation based evolutionary algorithm for the evolution of the membership functions [14]. More specifically, we used the $(\mu + \lambda)$ -evolutionary algorithm where $\mu = 1$ and $\lambda = 5$. In this algorithm, each individual of the second population is cloned 5 times, and then a relatively high degree of mutation is applied to each clone. The mutation function increases or decreases the value of each parameter of the membership functions with a small value. Once all 5 clones have undergone mutation process,

the fitness function of each clone is calculated in the process discussed below.

As we mentioned before the fitness value of each membership function individual depends on the predictive accuracy of the fuzzy rule individuals, so, in order to calculate the fitness value of each membership function individual, the individual is considered as the membership function definition for a group of the best fuzzy rule individuals (best 5 in this project). At the end, the fitness value of the membership individual is equal to the sum of the fitness values of the fuzzy rule individuals.

III. RESULTS AND DISCUSSION

Recall that our system should be run c times for a dataset that has c classes. As a result, we would have c fuzzy classification rule sets and each rule set would have its own membership function definition. Each example of the test set will be classified with all resulting classification rule sets and the example will be assigned to the rule set that has the highest degree of membership value. The classification accuracy of the system is calculated as the number of examples that was correctly classified divided by the total number of examples in the test set.

We compared our Relational Fuzzy Classification Rule Derivation (RFCRD) system with other evolutionary and machine learning methods over the Iris dataset from UCI machine learning data set repository [18]. The Iris dataset is one of the most well-known datasets in machine learning literature. The dataset contains three classes of Iris plant namely: Setosa, Versicolour and Virginica. There are 50 instances of each plant in the dataset. The algorithm should classify these examples based on four attributes of sepal length, sepal width, petal length and petal width. The Setosa class is linearly separable from the two others, while the remaining two classes are not linearly separable. We evaluated our system using 10-fold cross-validation method. In this method, the original dataset is divided into 10 parts with equal size, and each time one of the 10 parts is selected as the test set and the rest 9 sets form the training set. The overall accuracy of the algorithm is calculated as the average classification accuracy of the algorithm over 10 different training and test sets. In order to compare the classification accuracy of our system, we will also mention two other evolutionary methods and one non-evolutionary method found in the literature. The evolutionary methods are: CEFR-MINER [14], ESIA [19]. CEFR-MINER finds propositional fuzzy classification rules while ESIA builds crisp decision trees. The non-evolutionary method is C4.5 [20] which is a well-known decision tree builder method. We used the results reported in [21] for C4.5 algorithm. All of these systems used 10-fold cross-validation for system evaluation. The parameters that we used in our system are summarized in Table 1 and Table 2. Table 1 shows the parameters of fuzzy rule population, and Table 2 shows the parameters of membership function evolution population.

As it can be seen in Table 3, our system got better results compared to other three methods. Recall that our system

TABLE I
FUZZY CLASSIFICATION RULES POPULATION PARAMETERS

Parameter	Value
Population Size	250
Maximum Generations	100
Initial Number of rules per individual	3
Initial Number of terminals per rule	3
Maximum Number of rules per individual	20
Maximum Number of terminals per rule	8
Crossover Rate	0.9
Mutation Rate	0.05
Reproduction Rate	0.1

TABLE II
MEMBERSHIP FUNCTION POPULATION PARAMETERS

Parameter	Value
Population Size	50
Maximum Generations	100
Mutation Rate	0.33
Parent Selection	Truncation Selection
Selection Parameter	$\mu = 1, \lambda = 5$

derives fuzzy rules, so the end rules are more comprehensible for the end user than other crisp rule derivation systems. Also, as we derive relational rules we got more understanding of the underlying relation between the attributes of the dataset. A sample final rule set of the system is shown in figure 4.

TABLE III
COMPARISON OF RESULTS

Algorithm	Classification Accuracy	S.D.
Our System	98.0%	4.7
CEFR-MINER	95.3%	7.1
ESIA	95.3%	0.0
C4.5	95.3%	5.0

We ran our system on a dual-processor 2.13 GHz machine with 2.0 GB of memory. Each run of the system for classification of one of the goal attributes took a processing time of 90 seconds.

IF
((NOT Petal-length>Sepal-width) AND (Sepal-length=High) AND (Sepal-width=Med) AND (NOT Sepal-length>Petal-length))
OR
((NOT Petal-width=Med) AND (NOT Petal-length=Low) AND (Sepal-length=High) AND (NOT Petal-length>Sepal-width))
THEN
Class= Iris-Versicolour

Fig. 4. Sample rule set of the system derived from Iris dataset

IV. CONCLUSION AND FUTURE RESEARCH

We proposed a Relational Fuzzy Classification Rule Derivation (RFCRD) system in this paper. Our system uses two populations for discovering classification rules. The first population encodes the fuzzy classification rules, and the second one keeps the membership functions definition for fuzzification of continuous attributes and the relational operators between the attributes that have the same type. The first population uses the second population for defining the rule antecedents evolved by the first population. The GP algorithm evolves the population of fuzzy rules, and a simple mutation based evolutionary algorithm evolves the second population. These two populations co-evolve with each other to better adapt to the underlying dataset. The main advantage of our system that makes it different from previous work is using fuzzy relational operators. We fuzzified the relational operator of “greater than” and “less than” between to attributes that have the same type. More precisely, we fuzzified the degree that one attribute is bigger or less than the other attribute, and used that in the antecedent of our rules. We have evaluated our system over the well-known Iris dataset and compared the end result with three other evolutionary and non-evolutionary classification methods.

- Our system outperformed the other systems with respect to the classification accuracy.
- Our system produces relational fuzzy rules which give the user more understanding of the interaction between the attributes of the dataset.
- Our system produces fuzzy classification rules which are more comprehensible for the user.

There are several directions for the future search. For example, in this paper we only fuzzified the “greater than” and “less than” operator. Fuzzification of other relational operators can be an interesting approach. For example, one can fuzzify how much two attributes are equal to each other. All of these approaches will give the user more understanding of the underlying dataset, and also produces the rules that are comprehensible for the human and can be directly used in knowledge-based systems.

Another important extension of the research presented here is the replacement of the GP algorithm with the Gene Expression Programming [22] (GEP) algorithm. The latter allows for the evolution of multi-genic chromosomes, each gene coding for a different sub-expression. Recall that, in the GP system proposed here, if the problem includes n classes, then we have to run the system n times to obtain n classification rule sets. On the other hand, by using a multi-genic GEP system (i.e., a GEP system that evolves chromosomes with n genes, each gene coding for a specific classification rule set), any given run would simultaneously evolve n classification rule sets.

REFERENCES

- [1] A. Freitas, “A survey of evolutionary algorithms for data mining and knowledge discovery,” *Advances in Evolutionary Computation*, pp. 819–845, 2002.

- [2] J. Koza, *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.
- [3] J. Holland, “Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems,” *Computation & intelligence: collected readings table of contents*, pp. 275–304, 1995.
- [4] S. Smith, “Flexible learning of problem solving heuristics through adaptive search,” *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 422–425, 1983.
- [5] D. Greene and S. Smith, “Competition-based induction of decision models from examples,” *Machine Learning*, vol. 13, no. 2, pp. 229–257, 1993.
- [6] A. Giordana and F. Neri, “Search-Intensive Concept Induction,” *Evolutionary Computation*, vol. 3, no. 4, pp. 375–419, 1995.
- [7] C. Anglano and M. Botta, “NOW G-Net: learning classification programs on networks of workstations,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 463–480, 2002.
- [8] O. Cerdón, F. Herrera, and M. del Jesus, “Evolutionary approaches to the learning of fuzzy rule-based classification systems,” *The Crc Press International Series On Computational Intelligence*, pp. 107–160, 1999.
- [9] B. Chien, J. Lin, and T. Hong, “Learning discriminant functions with fuzzy attributes for classification using genetic programming,” *Expert Systems With Applications*, vol. 23, no. 1, pp. 31–37, 2002.
- [10] H. Ishibuchi, T. Nakashima, and T. Kuroda, “A hybrid fuzzy GBML algorithm for designing compact fuzzy rule-based classification systems,” *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, vol. 2, 2000.
- [11] D. Walter and C. Mohan, “ClaDia: a fuzzy classifier system for disease diagnosis,” *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 2, 2000.
- [12] G. Dounias, A. Tsakonas, J. Jantzen, H. Axer, B. Bjerregaard, and D. Keyserlingk, “Genetic Programming for the Generation of Crisp and Fuzzy Rule Bases in Classification and Diagnosis of Medical Data,” *Proc. 1st Int. NAISO Congr. Neuro Fuzzy Technologies*, 2002.
- [13] J. Kishore, L. Patnaik, V. Mani, and V. Agrawal, “Application of genetic programming for multicategory pattern classification,” *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 242–258, 2000.
- [14] R. Mendes, F. Voznika, A. Freitas, and J. Nievola, “Discovering Fuzzy Classification Rules with Genetic Programming and Co-evolution,” *Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Freiburg, Germany, September 3-5, 2001: Proceedings*, 2001.
- [15] L. Zadeh, “Knowledge representation in fuzzy logic,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, no. 1, pp. 89–100, 1989.
- [16] D. Hand, *Construction and assessment of classification rules*. Wiley New York, 1997.
- [17] F. Hoffmann and G. Pfister, “Evolutionary design of a fuzzy knowledge base for a mobile robot,” *International Journal of Approximate Reasoning*, vol. 17, no. 4, pp. 447–469, 1997.
- [18] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [19] J. Juan Liu and J. Tin-Yau Kwok, “An extended genetic rule induction algorithm,” *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, pp. 458–463, 2000.
- [20] J. Quinlan, *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [21] J. Gama, “Oblique Linear Tree,” *Advances in Intelligent Data Analysis: Reasoning About Data: Second International Symposium, Ida-97, London, Uk, August 4-6, 1997: Proceedings*, 1997.
- [22] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.