C-FOCUS: A continuous extension of FOCUS

*Antonio Arauzo Azofra   Jose Manuel Benítez Sánchez
Juan Luis Castro Peña
Department of Computer Science and Artificial Intelligence
University de Granada (Spain)
email: arauzo@decsai.ugr.es

8th September 2002

Abstract

This paper deals with the problem of feature selection. Almuallim and Dieterich [1] developed the FOCUS algorithm which performs optimal feature selection on boolean domains. In this paper an extension of FOCUS is developed to deal with discrete and continuous features. The extension, C-FOCUS, is verified on an artificial geometric figure classification problem and a real world classification problem.

1 Introduction

Feature selection help us to focus the attention of an induction algorithm in those features that are the best to predict a target concept. Although one might think that the more information available to an induction algorithm the better it works, this has revealed to be false for the following two main reasons. First, a large number of features in the input of induction algorithms may turn them very inefficient as memory and time consumers. And second, irrelevant data may confuse algorithms making them to reach false conclusions.

In feature selection we are interested in finding the minimal set of features which allows us to induce the target concept. John, Kohavi and Pfleger[4] classify the features in three relevance classes: irrelevant, weakly relevant and strongly relevant. FOCUS algorithm[1] is successful identifying the set with all strongly relevant and the minimal number of weakly relevant features to the target concept. As result of this, FOCUS is an ideal algorithm to use when a minimal set of features is required and noisy free samples are available.

FOCUS always finds the optimal set through a complete search on the features subsets space in quasi-polynomial time. It has achieved very good results in comparisons, where it has also been proved to work quite well on datasets with some noise[2].

However FOCUS is limited to boolean domains, while many real problems have discrete and continuous attributes. In order to see if FOCUS good behavior could be exported to other problem domains we have extended FOCUS-2[1] (the optimized version of FOCUS) to select features with different data types:

*This research has been supported by project CICYT-TIC2000-1362-C02-01

nominal, discrete and continuous. The extension to continuous values has been done by defining a concept of what is considered to be distinct in a continuous domain, while the extension to nominal and discrete values is direct since this concept is clear on these domains.

In section 2 we describe FOCUS algorithm and its extension C-FOCUS. In section 3 we create a geometric figure classification problem, which is adequate to apply original FOCUS algorithm but with a mix of continuous and discrete features. Then the results of C-FOCUS application to this problem and a real world problem are shown. And we end in section 4 with some conclusions and future work.

2 Description of the Algorithm

The main idea of original FOCUS algorithm is to identify all pairs of examples with a different boolean result. Each of these pairs is called a conflict, and FOCUS goal is to select the minimal set of features that solves all conflicts. A feature is considered to solve a conflict when its value is different between both examples. That is when the feature allow us to distinct between the two examples.

It is clear when two values are different in a boolean or discrete domain, so it is clear when a conflict is solved by a boolean or discrete variable. But we need to define when two continuous values will be considered different. To this purpose our extension utilizes the absolute difference between the two values in the following simple way. All values in samples of a given feature are normalized to $[0, 1]$. If the difference is greater than a given threshold U the two values will be considered distinct.

FOCUS searches through the space of feature subsets to find the one with a minimal number of features that solves all conflicts.

This search can be done trying sequentially with all sets of $1, 2, 3, \dots N$ variables until one set that solves all conflicts is found. But if one conflict is solved only by a feature X_i, we know that X_i should belong to the set of features selected. With this idea Almuallim and Dieterich[1] developed an optimized version of FOCUS: FOCUS-2.

Algorithm FOCUS-2(*Sample*)

1. If all the examples in *Sample* have the same class, return $\emptyset$.
2. Let G be the set of all conflicts generated from *Sample*.
3. $Queue = \{M_{\emptyset,\emptyset}\}$.
4. Repeat

    4.1 $M_{A,B}$ = Pop the first element in *Queue*.

    4.2 $OUT = B$.

    4.3 Let a be the conflict in G not covered by any of the features in A, such that $|Z_a - B|$ is minimized, where Z_a is the set of features covering a.

    4.4 For each $x \in Z_a - B$

    4.4.1 If $Sufficient(A \cup \{x\}, Sample)$, return $A \cup \{x\}$.

    4.4.2 Insert $M_{A\cup\{x\},OUT}$ at the tail of *Queue*.

4.4.3 $OUT = OUT \cup \{x\}$.

end.

$M_{A,B}$ denotes the space of all feature subset that include all of the features in the set A and none of the features in the set B.

As the sufficiency test of step 4.4.1, $Sufficient(Features, Sample)$, we have used a simple search through $Sample$ of two examples, with values not considered different in selected $Features$, that belong to a different class. If there are no such two examples the $Features$ set is sufficient, not being sufficient otherwise.

3 Empirical Study

3.1 Geometric Figures Problem

3.1.1 Problem Description.

To test C-FOCUS we have created a simple geometric figure classification problem.

We get some examples from the following figures:

- Equilateral triangle
- Isosceles triangle
- Square
- Rectangle

With its values for the following features:

- Number of sides (NSides)
- Longest side length (LS)
- Shortest side length (SS)
- Perimeter
- Area
- Shortest side length / longest side length (SS/LS)

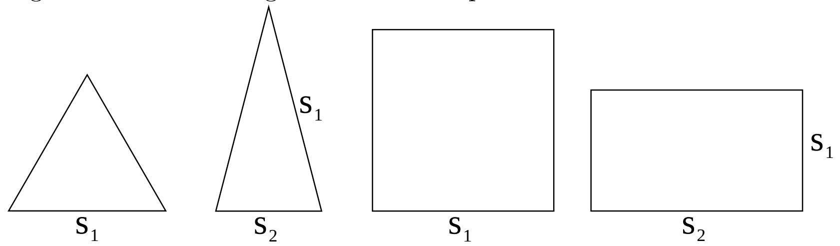
The formulas and constant values of this features for the figures considered are shown in Figure 1.

The process used to generate the samples has been the following:

Repeat N times (where N is the number of examples to generate)

- Choose a figure class (Uniform random generator in $\{0, 1, 2, 3\}$)
- Repeat until values satisfy restrictions
  - Generate sides length (Uniform random generator in $[0, 1]$)

Figure 1: Geometric figures and its sample features



Nsides	3	3	4	4
LS	s_1	$\max(s_1, s_2)$	s_1	$\max(s_1, s_2)$
SS	s_1	$\min(s_1, s_2)$	s_1	$\min(s_1, s_2)$
Perimeter	$3 * s_1$	$2 * s_1 + s_2$	$4 * s_1$	$2 * s_1 + 2 * s_2$
Area	$\sqrt{\frac{3}{4} s_1^2}$	$\frac{s_2 \sqrt{4{s_1}^2 - {s_2}^2}}{4}$	s_1^2	$s_1 * s_2$
SS/LS	1	$\frac{SS}{LS}$	1	$\frac{SS}{LS}$

The restrictions named above are: In isosceles triangles, the sum of the two equal sides should be greater than the other side. And the difference between sides s_1 and s_2 in rectangles and isosceles triangles should be greater than 5%, to avoid them to be almost squares and equilateral triangles respectively.

All of the above features are related to the classification problem. To test if our extension is able to reject all the irrelevant features we have introduced other features with random values.

The goal is to select the minimal number of features that allow to classify each example as one of the 4 figure types.

Based on our previous knowledge of the problem, we know that, among the available features, the minimal set that allows to classify the 4 figure types correctly is {Number of sides, Longest side / Shortest side}. While other feature sets like {Longest side, Shortest side, Area} are also good for classification.

3.1.2 Results.

The tests have been made with different sample sets in number of irrelevant features included and size. We have created three types of samples with 1, 10 and 25 irrelevant features added. In order to see if the behavior of the algorithm is affected from the number of irrelevant features present on the data. We have used samples of 50, 100, 250 and 500 examples, for every of these sample types, to show that from a small number of examples C-FOCUS can achieve good results.

Running with the same datasets C-FOCUS threshold parameter had been varied in the following values: 0.025, 0.05, 0.1 and 0.2. The results are shown in the tables: 1, 2, 3 and 4 respectively. Some of the feature names are abbreviated as indicated in the feature list at problem description. Irrelevant variables are referred as “IrrN” where N is the position of the variable.

C-FOCUS has found a sufficient set of features that allows to classify correctly in 41 cases. It informs that at given threshold level the problem can not be solved in 5 cases. And finally only in 2 cases, which are from the smallest ones (50 examples datasets), returns a not sufficient set of feature sets.

Table 1: Selected features on each dataset with U=0.025

Examples	Number of irrelevant features		
	1	10	25
50	NSides, SS/LS	NSides, SS/LS	SS, Perimeter
100	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS
250	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS
500	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS

Table 2: Selected features on each dataset with U=0.05

Examples	Number of irrelevant features		
	1	10	25
50	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS
100	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS
250	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS
500	NSides, SS/LS	NSides, SS/LS	NSides, SS/LS

Table 3: Selected features on each dataset with U=0.1

Examples	Number of irrelevant features		
	1	10	25
50	NSides, SS/LS	NSides, SS/LS, Irr0, Irr8	NSides, SS/LS, Irr19
100	NSides, SS/LS, Irr0, Area	NSides, SS/LS, Irr0, Irr2, Irr8	NSides, SS/LS, Irr4
250	NSides, SS/LS	NSides, SS/LS, Irr0, Irr3, Irr4	NSides, SS/LS, Irr0, Irr1, Irr21
500	(Not solved)	(Not solved)	NSides, SS/LS, Irr2, Irr10, Irr12

Table 4: Selected features on each dataset with U=0.2

Examples	Number of irrelevant features		
	1	10	25
50	NSides, SS/LS, Area, Irr0	NSides, SS, Irr0, Irr1, Irr4	NSides, SS/LS, Irr0, Irr2, Irr23
100	(Not solved)	NSides, SS/LS, SS, Irr0, Irr1, Irr6	NSides, SS/LS, SS, Irr2, Irr6, Irr12
250	(Not solved)	NSides, SS/LS, SS, Irr0, Irr1, Irr2, Irr3, Irr4	NSides, SS/LS, Irr0, Irr4, Irr6, Irr7, Irr18
500	(Not solved)	NSides, SS/LS	NSides, SS/LS, SS, Area, Perimeter, Irr2, Irr4, Irr12, Irr17

Table 5: Forest problem results without feature selection

Topology	Test set				Max	Mean
	1	2	3	4		
54-4-7	50.4	55.6	57.8	68.4	68.4	58.05
54-5-7	56.8	52.9	54.4	72	72	59.025
54-6-7	43.4	52.1	56.8	70.8	70.8	55.775
54-7-7	48.9	51.5	51.5	70.4	70.4	55.575
54-8-7	41.7	54.8	57.5	69.5	69.5	55.875
54-9-7	44.5	52.7	57.8	68.1	68.1	55.775
54-10-7	53	50.3	52	70.7	70.7	56.5
Max	56.8	55.6	57.8	72	72	60.55
Mean	48.385	52.843	55.400	69.986	69.986	56.654

Threshold parameter has been very important to the results, as higher values make C-FOCUS to introduce more features than necessary and sometimes irrelevant.

3.2 Forest CoverType problem

This problem deals with getting the forest cover type for a 30x30 meter cell from a given set of 54 boolean and quantitative features. The dataset for this problem is available at the UCI KDD Archive[3].

We chose randomly 2000 examples from the dataset. C-FOCUS was run on them with different threshold levels, starting with 0.2 and dividing by 2 on each step. The first threshold that gave a feature selection was 0.0125 (previous ones found that the conflict set was unsolvable at that threshold level).

In order to test if the features selected by C-FOCUS are good to this classification problem we have used neural networks as classifier. We have compared the results obtained with CFOCUS + NN, NN without using feature selection and Relief-E[6] + NN.

Relief-E has been chosen because it is a very well known algorithm, compared with many others[2]. Also a similar version of Relief was chosen as representative of filter feature selection methods to present the wrapper approach[5].

We used four training sets with 4000 examples and respectively four disjoint 1000 examples test sets. Neural networks were initialized with uniform random weights and back-propagation with a learning rate of 0.05 was used as training method.

All the results shown are the percentage of correct classification. Those obtained directly with neural networks without feature selection are in table 5.

The features selected by C-FOCUS were: Elevation, Aspect, Slope, Horizontal-Distance-To-Hidrology, Vertical-Distance-To-Hydrology, and Horizontal-Distance-To-Roadways. Table 6 shows the results.

Given that Relief-E only assign a valuation to each feature but does not give the number of features that should be used, we have taken the same number of features as C-FOCUS most valued. In this way the features selected have been: Aspect, Horizontal-Distance-To-Roadways, Horizontal-Distance-To-Fire-Points, Horizontal-Distance-To-Hydrology, Slope, and Hillshade-3pm. Table 7 shows the results.

Table 6: Forest problem results using C-FOCUS

Topology	Test set				Max	Mean
	1	2	3	4		
6-4-7	55.6	57.1	64	58.7	64	58.85
6-5-7	61.8	50	64.5	58.2	64.5	58.625
6-6-7	59.5	52.3	69.2	62.2	69.2	60.8
6-7-7	58.7	53	60.3	66.2	66.2	59.55
6-8-7	59.8	52.4	71.7	63.9	71.7	61.95
6-9-7	63.8	54.7	67.7	66.3	67.7	63.125
6-10-7	59.5	52.7	60.8	64.5	64.5	59.375
Max	63.8	57.1	71.7	66.3	71.7	64.725
Mean	59.814	53.171	65.457	62.857	65.457	60.325

Table 7: Forest problem results using RELIEF-E

Topology	Test set				Max	Mean
	1	2	3	4		
6–4-7	21.5	25.5	40.8	41.1	41.1	32.225
6–5-7	25.3	26.6	45	41.3	45	34.55
6–6-7	25.3	28.9	41.4	45.4	45.4	35.25
6–7-7	23.8	25.5	42.5	48.6	48.6	35.1
6–8-7	32	23.8	42	42.8	42.8	35.15
6–9-7	28.5	22.8	43.4	46	46	35.175
6–10-7	25.3	22.3	40.3	47.7	47.7	33.9
Max	32	28.9	45	48.6	48.6	38.625
Mean	25.957	25.057	42.200	44.700	44.700	34.479

4 Summary and Conclusions

We have developed C-FOCUS algorithm as an extension of FOCUS[1] algorithm to discrete and continuous domains. In this way it can be used in a wider set of problems.

This algorithm is recommended in classification problems in which we have noise free samples and the main goal is to reduce the number of features. We have created such a problem and found another appropriate real world problem. We have tested C-FOCUS algorithm on them having good results on both.

Choosing an appropriate threshold parameter is very important as it has been shown in the experiments. We have used an approach decrementing threshold until it solves all conflicts in the forest cover problem. In our artificial problem we have tried some different values getting different results. On this results it can be seen (as we know the preferred features) that when best results are achieved (U=0.025 and U=0.05) the features selected are identical or pretty similar on the different example sets. Having this on mind we suggest that the right threshold can be chosen running C-FOCUS on some training subsets and choosing the threshold that gives most similar results on the different training sets. We leave this as an open problem that can be more deeply studied.

More future work can be done to fine-tuning the way continuous features are treated. The approach presented here may has problems with features that affect the target concept in a non-continuous way. For example, in the geometric figures problem if we have an rectangle with very similar sides length it will be hard for C-FOCUS to distinct this rectangle from a square.

References

[1] Hussein Almuallim and Thomas G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994.

[2] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997.

[3] S. Hettich and S. D. Bay. The uci kdd archive. http://kdd.ics.uci.edu/, 1999.

[4] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994. Journal version in AIJ, available at http://citeseer.nj.nec.com/13663.html.

[5] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[6] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.