Important Features Detection in Continuous Data

Piotr Fulmański, Alicja Miniak-Górecka
Faculty of Mathematics and Computer Science, Department of Mathematical Analysis and Control Theory
University of Łódź
Łódź, Poland

***Abstract*—In this paper, a method for calculating the importance factor of continuous features from a given set of patterns is presented. A real problem in many practical cases, like medical data, is to find which parts of patterns are crucial for correct classification. This leads to the need of preprocessing all data, which has influence on both time and accuracy of applied methods (when unimportant data hide those which are important). There are some methods that allow selection of important features for binary and sometimes discrete data or, after some preprocessing, continuous data. Very often however, such conversion is burdened with the risk of losing important data, which is a result of lack of knowledge of optimal discretization consequence. Proposed method allows to avoid that problem, because it is based on original, non-transformed continuous data. Two factors - concentration and diversity - are defined and are used to calculate the importance factor for each feature and pattern. Based on those factors e.g. unimportant features can be identified to decrease dimension of input data or ''bad'' patterns can be detected to improve classification. An example how proposed method can be used to improve decision tree is given as well.**

Keywords-important features extraction; continuous data analysis; decision tree.

I. INTRODUCTION

In this paper, the following problem of the data processing and analysing is presented. Let L be a given learning set defined as:

$$L = \{l_1 = (p_1 = (c_1^1, \dots, c_m^1), t_1), \dots, l_n = (p_n = (c_1^n, \dots, c_m^n), t_n)\} \quad (1)$$

L is a set of pairs $l_1,\dots,l_n$, where the first element (called: input signal) is an m-components vector of features $(p_i, i=1,\dots n)$, while the second is a value which belongs to a given, finite set T. Notation c_j^i denotes j-th feature from i-th pattern. T is a set of correct (expected) output signals (also: responses, targets or classes). It can consist of numbers, but also of logic values: *yes*, *no*, *unknown* or linguistic: *brake*, *move slowly*, *move*, *accelerate*, *stop*. Features $c_1^i,\dots,c_m^i$, $i=1,\dots,n$ are independent of each other (i.e. set of values for feature c_p^i does not depend on set of values for feature c_q^i, $p \neq q$), can be both discrete and continuous.

Presented problem is solved when for each $t \in T$ there is known such a set of features, which is sufficient to unambiguous identification (classification) of all of the learning data for which t is an expected class. As an example, consider set L defined in table I. All patterns are divided into five different classes: A, B,…, E. Features which, according to our assumption, should characterize each class are embolden. Assumptions for set L were as follow.

- Class A should be recognized based on fact that feature 1 takes values from interval 10-30, whereas the rest of features should not have any regularity.
- Class B should be recognized based on fact that feature 1 takes values from interval 10-30 and features 2 and 3 take values from interval 50-65, whereas the rest of features should not have any regularity.
- Class C should be recognized based on fact that feature 2 takes values from interval 90-110, feature 3 takes values from interval 60-75, feature 4 takes values from interval 25-55, whereas the rest of features should not have any regularity.
- Class D should be recognized based on fact that feature 4 takes values from interval 0-25, whereas the rest of features should not have any regularity.
- Class E should be recognized based on fact that feature 1 takes values from interval 50-70, whereas the rest of features should not have any regularity.

According to the above assumptions a few randomly generated sets were created – the set L is one of them. In all cases results were similar.

TABLE I. EXAMPLE OF LEARNING SET *L*

Pattern	Feature					Class
	1	*2*	*3*	*4*	*5*	
p_1	**10**	65	50	50	50	A
p_2	**20**	70	60	25	70	A
p_3	**25**	80	100	95	130	A
p_4	**29**	100	90	100	105	A
p_5	**15**	110	**50**	**50**	80	B
p_6	**25**	90	**55**	**75**	55	B
p_7	**29**	60	**60**	**60**	60	B
p_8	**31**	75	**63**	**65**	150	B
p_9	5	**90**	**60**	**25**	110	C
p_{10}	30	**105**	**70**	**30**	145	C

Pattern	Feature					Class
	1	*2*	*3*	*4*	*5*	
p_{11}	15	**100**	**65**	**50**	60	C
p_{12}	58	**95**	**57**	**52**	45	C
p_{13}	95	**87**	**72**	**48**	50	C
p_{14}	100	**110**	**60**	**27**	90	C
p_{15}	40	70	60	**0**	60	D
p_{16}	70	80	55	**10**	70	D
p_{17}	80	100	95	**25**	120	D
p_{18}	**50**	110	95	10	70	E
p_{19}	**60**	60	80	60	80	E
p_{20}	**70**	75	50	110	110	E

II. Decision Tree and Continuous Data

From the previous section it can be seen, that the goal is to create a model that predicts the value of a target variable based on several input variables (features). As a predictive model a decision tree which maps observations about an item to conclude the target value of an item can be used. An interior node corresponds to one of the input variables; each of these nodes has a number of children nodes equal to the number of the possible values of that input variable. Each leaf node represents a possible outcome depending on the values of the input variables represented by the path from the root node to the leaf node. It is essential that a tree can be "learned" by splitting the source set into subsets based on an **attribute value test**. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion ends when the subset at a node has the same value of the target variable, or when further splitting no longer adds a value to the predictions.

In pseudocode, the general algorithm for building decision trees is [1]:

1. Check for base cases.
2. For each attribute *a* find the normalized information gain from splitting on *a*.
3. Let a_{best} be the attribute with the highest normalized information gain.
4. Create a decision node that splits on a_{best}.
5. Recur on the sublists obtained by splitting on a_{best}, and add those nodes as children of node.

In presented algorithm the most important are steps 2 and 3: selection a_{best} attribute. Selection of that attribute should be based on some factor describing its importance regarding data that are not classified yet. Term *importance* in this case is understood as an ability to create (based on that attribute) correct pattern classification -- the more patterns are classified correctly, the better (the more important) the attribute is. While for discrete data methods for attribute importance factor calculating were developed (see for example [2], where method for binary patterns recognition is described or C4.5 algorithm), the lack of such methods can be observed for continuous data. As an example of this problem consider one of the widely used free data mining tool i.e. C4.5 algorithm developed by Ross Quinlan [3] used to generate a decision tree and implemented in SIPINA Data Mining Software [4].

C4.5 is an extension of Quinlan's earlier ID3 algorithm and is followed in turn by See5/C5.0[1][5]. C4.5 made a number of improvements to ID3 -- one is important from our point of view: the ability to handle both continuous and discrete attributes. Unfortunately in order to handle continuous attributes, C4.5 creates a threshold and then splits the list into two: those which attribute value is above the threshold and those that are less than or equal to it [6]. As a result, continuous data are subject to some kind of discretization. This process can be performed before the main algorithm or as a one of auxiliary sub-steps of it. Anyway, continuous data are de facto treated as discrete. In many cases, discretization results in loss of information. In this paper, method for calculating importance factor of continuous features from given patterns set, without discretization necessity, is presented.

III. Measure of Importance of Features

While searching for important features that distinguish a given class among other classes, for each feature the following factors should be determined:

- if a feature is a distinctive feature within a given class (so-called importance factor for all patterns within a given class) -- for example, for all patterns this feature has the same value;
- if a feature is a distinctive feature for a given class within all classes (so-called importance factor for a given class within all classes) -- for example, for all patterns which are not from a given class this feature takes value from interval 0-10, while for patterns from a given class this feature takes value 15.

In a given examplary set of patterns *L* (table I) one can notice that feature 4 is the most important (the most distinctive) feature for class D within this class (the smallest diversity can be observed for it). Feature 4 is an example of second factor as the most important (the most distinctive) feature for class D within all classes, because for none of the other classes values of this feature belong to interval 0-25[2].

A. Importance factor for all patterns within a given class

For each feature, the smaller the changeability of its values within a given class is, the more important this feature is. In other words, concentration of this feature is higher. **Concentration factor** of feature *a* in class *b* is defined as:

[1] C5.0/See5 is a commercial and closed-source product. C5.0 offers a number of improvements on C4.5 like speed (C5.0 is several orders of magnitude faster than C4.5), more memory usage efficient or smaller decision trees (C5.0 gets similar results to C4.5 with considerably smaller decision trees).

[2] Values from this interval that can be observed for feature 4 in other classes e.g. pattern 2 (class A) with value 25 or pattern 18 (class E) with value 10 simulate anomalies in the data and were added intentionally.

$$cf_a^b = \int_{-\infty}^{+\infty} exp\left(\frac{-(x-\mu_a^b)^2}{2\sigma_a^{b\,2}}\right) dx, \quad (2)$$

where μ_a^b is a mean (expected value) and σ_a^b is a standard deviation of all values for feature *a* in class *b*. The smaller the concentration factor is, the closer the values of a considered feature within a given class are. It can be interpreted in the following way: if all values of a considered feature within a given class are (almost) identical, it can be stated that this feature (its values) is being characteristic for all patterns within a given class.

For example a characteristic feature of all tanks is to have tracks (but not all tracked vehicle are tanks). Examining concentration factors for patterns from set *L* (see table II), one can notice that for each class the smallest value of this factor is located in one of the features, which were assumed to be characteristic. It is worth highlighting that the set *L* is not "perfect" -- as some patterns are not necessarily fulfilling all assumptions for a corresponding class to which these patterns belong in a way that a wrong classification would be excluded. For example, pattern 16 (from class D) could be assigned to class E.

B. Importance factor for a given class within all classes

A feature is considered to be the more diversified, the greater changeability of its values within all classes is observed. **Diversity factor** of feature *a* within all classes is defined as:

$$cf_a = \int_{-\infty}^{+\infty} exp\left(\frac{-(x-\mu_a)^2}{2\sigma_a{}^2}\right) dx, \quad (3)$$

where μ_a is a mean (expected value) and σ_a is a standard deviation of all values for feature *a* within all classes. Diversity factor is a little bit more difficult to describe than concentration factor. It has much more sense when considered jointly with the concentration factor (see next subsection). For now, we can say that a small value of this factor means that many patterns from different classes take similar values.

TABLE II. CONCENTRATION FACTORS FOR PATTERNS FROM SET *L*. THE SMALLEST VALUE FOR EACH CLASS IS UNDERLINED.

Feature					Class
1	*2*	*3*	*4*	*5*	
17.81	33.60	51.67	78.51	77.44	A
15.45	46.36	**12.40**	**22.59**	95.19	B
92.87	**20.23**	**13.80**	**28.78**	89.69	C
42.60	31.26	44.60	**25.75**	65.79	D
20.46	52.51	46.89	102.33	42.60	E

TABLE III. DIVERSITY FACTORS FOR PATTERNS FROM SET *L*.

Feature				
1	*2*	*3*	*4*	*5*
69.40	41.19	39.26	74.42	79.84

C. Discriminants

A discriminant describes how important a given feature of the considered pattern is for its correct classification. Discriminants are calculated for all features of all patterns with the following formula:

$$d_a^{b,c} = \frac{df_a}{cf_a^b} exp\left(\frac{-(x-\mu_a^b)^2}{2\sigma_a^{b\,2}}\right), \quad (4)$$

where *x* is a value of feature *a* from pattern *c* and class *b*. In formula (4) two component can be distinguished.

- The first component is a quotient which is calculated for each feature as a diversity factor for a given class (and feature) within all classes over concentration factor for all patterns within a given class (and feature). Value of this quotient close to 1 means that the feature which is being under consideration cannot be treated as a characteristic feature (discriminant) for the class. The most desirable is a "big" value of this component, which is obtained when values of a given feature in a selected class compared to values of this feature in other classes are evidently concentrated, that is when a feature is perfect to act as a characteristic (discriminant) of the class. This component is being calculated for every feature in all classes (see table IV).
- The second component, *exp()*, serves to eliminate data which are (very) different from the average value for a given class, that is data which could be an effect of measuring errors or some kind of an anomaly which should be considered individually. A value of this component close to 0 means that the feature in a considered pattern is greatly deviated from the average value for an appropriate class. On the other hand, when the value of this component is close to 1 it means that the feature in a considered pattern has a typical value for an appropriate class. In other words, **second component describes the grade of membership of a feature in a given pattern to the usual values of this feature in patterns from an appropriate class.** Averaging all grades of membership of features of a pattern, the *grade of membership of a pattern to a class* is obtained, which is denoted as $\mu^{b,c}$, where *c* - patterns, *b* -- class. Knowledge of the grades of membership of patterns is useful for "bad" patterns identification. Values of this component and the grades of membership are given in table V.

Taking into consideration the total effect of described elements, one can state that values calculated with formula (4) lower or equal to 1, shows features which should not be considered.

If this value is greater than 1 (the more, the better) then the considered feature is important. Final values of the discriminants for set *L* are presented in table VI.

The greatest value for each pattern is underlined. It can be noticed, that in all cases discriminants reach the greatest value for a feature which, according to initial assumptions, should be characteristic for a given class.

TABLE IV. VALUE OF QUOTIENT $\frac{df_a}{cf_b^a}$ FOR DATA FROM TABLE II AND III.

Feature					Class
1	*2*	*3*	*4*	*5*	
3.89	1.12	0.75	0.94	1.03	A
4.49	0.88	**3.16**	**3.29**	0.83	B
0.74	**2.03**	**2.84**	**2.58**	0.89	C
1.62	1.31	0.88	**2.89**	1.21	D
3.39	0.78	0.83	0.72	1.87	E

TABLE V. THE GRADES OF MEMBERSHIP OF FEATURES AND PATTERNS.

Feature					$\mu^{b,c}$	Class
1	*2*	*3*	*4*	*5*		
0.3	0.59	0.47	0.85	0.45	0.532	A
0.99	0.8	0.76	0.39	0.83	0.754	A
0.85	0.99	0.47	0.68	0.41	0.68	A
0.53	0.28	0.76	0.58	0.87	0.604	A
0.26	0.36	0.36	0.38	0.71	0.79	B
1.0	0.94	0.92	0.38	0.71	0.79	B
0.81	0.43	0.83	0.96	0.78	0.762	B
0.62	0.89	0.47	0.96	0.24	0.636	B
0.47	0.62	0.76	0.49	0.75	0.618	C
0.85	0.67	0.55	0.75	0.22	0.608	C
0.63	0.96	0.98	0.61	0.8	0.796	C
0.97	0.94	0.44	0.5	0.56	0.682	C
0.48	0.4	0.34	0.71	0.64	0.514	C
0.4	0.32	0.76	0.59	0.98	0.61	C
0.38	0.56	0.85	0.52	0.67	0.596	D
0.92	0.96	0.7	0.98	0.87	0.886	D
0.61	0.4	0.37	0.43	0.37	0.436	D
0.47	0.4	0.56	0.47	0.61	0.502	E
1.0	0.58	0.96	1.0	0.92	0.892	E
0.47	0.95	0.4	0.47	0.38	0.534	E

In case of classes A, D and E one feature was selected explicitly: first, fourth and first respectively. Explicitness is not observed in case of class B and C. For class B first feature (once) and fourth feature(twice) was detected as the most characteristic. For class C: third (three times), fourth (twice) and second (once). Those inconsistencies signal the need for usage of more features in case of some classes. In table VII the second highest discriminants relative to the values of discriminant for class B and C are shown (these values are underlined; for clarity, the highest value for each pattern is removed).

TABLE VI. DISCRIMINANTS FOR PATTERNS FROM SET *L*. THE HIGHEST VALUE FOR EACH PATTERN IS UNDERLINED.

Pattern	Feature					Class
	1	*2*	*3*	*4*	*5*	
p_1	**1.17**	0.72	0.36	0.81	0.46	A
p_2	**3.85**	0.99	0.58	0.37	0.85	A
p_3	**3.32**	1.22	0.36	0.64	0.42	A
p_4	**2.06**	0.34	0.58	0.55	0.89	A
p_5	**1.2**	0.32	**1.16**	**1.25**	0.82	B
p_6	**4.49**	0.83	**2.91**	**1.25**	0.59	B
p_7	**3.63**	0.38	**2.63**	**3.16**	0.66	B
p_8	**2.79**	0.79	**1.51**	**3.16**	0.2	B
p_9	0.35	**1.27**	**2.18**	**1.27**	0.67	C
p_{10}	0.64	**1.37**	**1.57**	**1.94**	0.2	C
p_{11}	0.47	**1.96**	**2.79**	**1.58**	0.71	C
p_{12}	0.73	**1.91**	**1.26**	**1.31**	0.5	C
p_{13}	0.36	**0.82**	**0.99**	**1.85**	0.57	C
p_{14}	0.3	**0.65**	**2.18**	**1.54**	0.87	C
p_{15}	0.63	0.74	0.75	**1.51**	0.81	D
p_{16}	1.5	1.27	0.61	**2.85**	1.06	D
p_{17}	1.0	0.53	0.32	**1.24**	0.45	D
p_{18}	**1.6**	0.31	0.47	0.34	1.15	E
p_{19}	**3.39**	0.45	0.8	0.72	1.73	E
p_{20}	**1.6**	0.74	0.34	0.34	0.73	E

TABLE VII. THE SECOND HIGHEST DISCRIMINANTS (UNDERLINED) FOR PATTERNS FROM SET *L*. FOR CLARITY, THE HIGHEST VALUE FOR EACH PATTERN IS REMOVED.

Pattern	Feature					Class
	1	*2*	*3*	*4*	*5*	
p_5	**1.2**	0.32	**1.16**		0.82	B
p_6		0.83	**2.91**	**1.25**	0.59	B
p_7		0.38	**2.63**	**3.16**	0.66	B
p_8	**2.79**	0.79	**1.51**		0.2	B
p_9	0.35	**1.27**		**1.27**	0.67	C
p_{10}	0.64	**1.37**	**1.57**		0.2	C
p_{11}	0.47	**1.96**		**1.58**	0.71	C
p_{12}	0.73		**1.26**	**1.31**	0.5	C
p_{13}	0.36	**0.82**	**0.99**		0.57	C
p_{14}	0.3	**0.65**		**1.54**	0.87	C

Taking into consideration those two features (one and four), the correct classification for class B should be possible.

Class C can be still a source of problems, because different pairs of features were selected: feature two and three (twice), feature three and four (three times) and finally feature two and four (once). Nothing prevents the next feature (the third highest discriminant) from being considered.

As a result, for class C features two, three and four will be selected[3]. Features one, three and four are selected if for class B three features are also considered.

Notice that based on values from table IV importance of features can also be estimated. However, information about sets of features, as it was described above, cannot be determined. Therefore data from table IV can be treated as a rough selection of important features, while richer information is contained in discriminants calculated with formula (4) (see table VI).

IV. Usage Example

In this section an example how the proposed method can be used to improve a decision tree is given. A decision tree generated in SIPINA [4] tool (C4.5 algorithm was selected) for learning set *L* is presented on Fig. 1.

It can be noticed, that feature five was not considered in any nodes, which could be predicted by analyzing table VI. Feature one is the first feature that splits the data set. Afterwards, feature three and four are considered. This is also reflected in table VI.

Knowledge of the grade of membership $\mu^{b,c}$ of pattern *c* to class *b* (how representative the selected pattern is for that class) allows one to modify learning set in such a way that smaller classification error will be achieved. For the considered learning set *L*, the smallest grades of membership are for patterns p_5 (0.468), p_{17} (0.436) and p_{18} (0.502).

One can notice that the decision tree from Fig. 1 does not make correct classification for all data. Data which are classified ambiguously are: (p_2, p_9), (p_1, p_5) and (p_{15}, p_{16}, p_{17}, p_{18}).

However, there is a correct classification for every case for reduced learning set *L* (patterns p_5, p_{17} and p_{18} were removed; see Fig. 2). Of course reduction of the data learning set may not have a permanent and strict character - it can be treated as a selection of potentially problematic patterns which should be treated separately.

V. Conclusions and Plans

All sets which were used during the tests (presented learning set *L* is one of them) are characterized by

- randomly generated set of features according to some assumptions which was described in section 1;
- existence of contradictory data - pattern 16 could just as well belong to class E and pattern 18 to class D.

In all cases the presented method for important features detection in continuous data works well. All features which, according to our assumptions, should be important were identified as such. The grade of membership usage allows more effective utilization of a data learning set through isolation of potentially problematic patterns (which could e.g. have negative influence during classification process). Notice, that global knowledge of important features gives new abilities. Instead of splitting data based on one feature (like in decision tree), a set of them (the most important) can be used to improve the decision process.

We want to stress, that in this paper an answer for a question: *which features are essential for correct pattern classification of a given class* is given. Proposed method is not a complete tool for data classification - it can be considered as an element of such system. This will be our next research problem - how to use information about important features to build classification system for a really problematic data, like medical data, which in many cases are incomplete or contradictory. Additionally, a new problem that we want to investigate arose: how to treat incomplete patterns.

References

[1] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica, vol. 31, 2007, pp. 249-268.

[2] A. Horzyk, "Fast Automatic Configuration of Artificial Neural Networks Used for Binary Patterns Recognition", Biocybernetics and Biomedical Engineering, vol. 21, 2001.

[3] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993.

[4] R. Rakotomalala, SIPINA, http://eric.univ-lyon2.fr/~ricco/sipina, 2010.

[5] RULEQUEST RESEARCH, "Is See5/C5.0 Better Than C4.5?, http://www.rulequest.com/see5-comparison.html, 2009.

[6] J. R. Quinlan, "Improved use of continuous attributes in c4.5.", Journal of Artificial Intelligence Research, 1996, pp.77-90.

[3] Feature five selected by pattern p_{14} is omitted - we treat it as an anomaly.

)

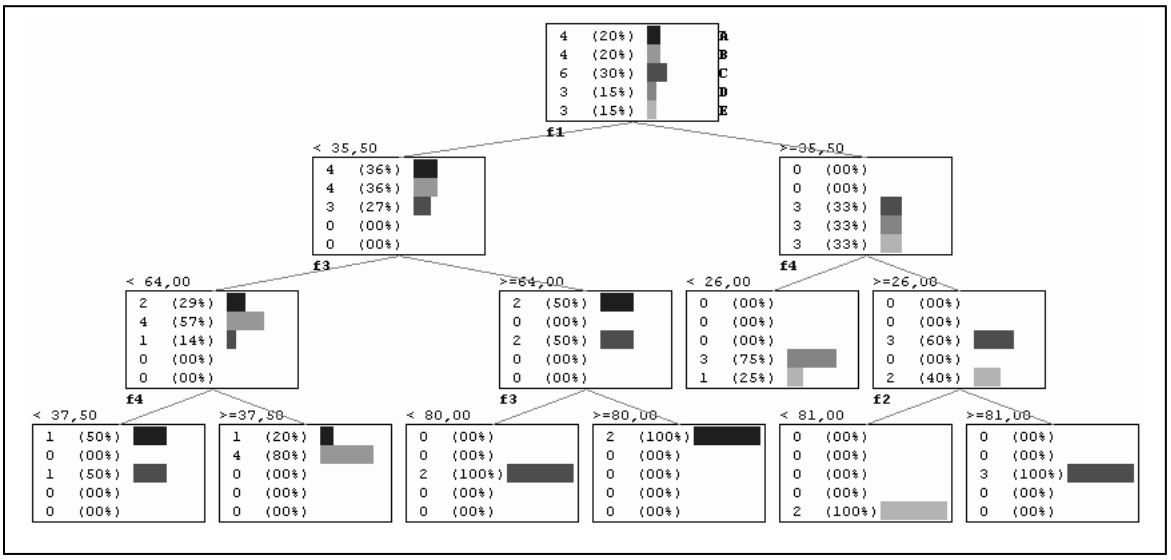
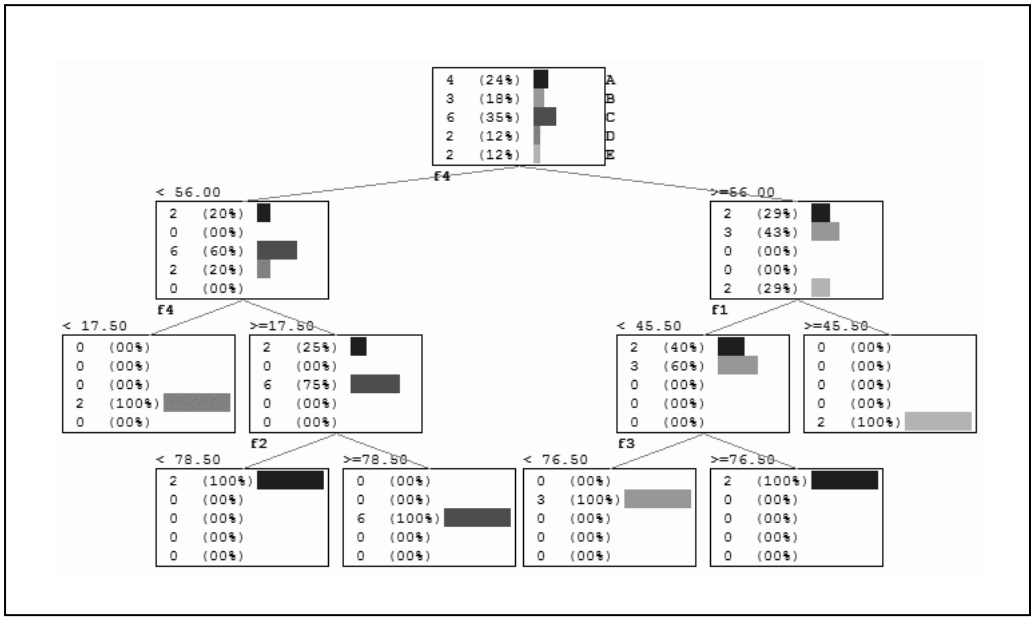


Figure 1. Decision tree generated in SIPINA tool (with C4.5 algorithm) for the learning set *L*.



)

Figure 2. Decision tree generated in SIPINA tool (with C4.5 algorithm) for a reduced learning set *L*.