# Comprehensive Review of Classification Algorithms for Medical Information System

Anna Kasperczuk and Agnieszka Dardzinska(✉)

Department of Mechanical Engineering, Division of Biocybernetics and Biomedical Engineering, Bialystok University of Technology, ul. Wiejska 45c, 15-351 Bialystok, Poland
{a.kasperczukuk,a.dardzinska}@pb.edu.pl

**Abstract.** Nowadays, the Internet and information systems become an integral part of everyday life. The trend of using advanced recommendation systems is still growing in various areas, also in medicine. Two of the diseases where diagnosis is a big problem for specialists are colon disease and Crohn's disease. The course of the disease strongly resembles other diseases in the large intestine, so it became extremely important to help doctors and find symptoms that would clearly indicate the colon disease, excluding others. In order to find rules that distinguish these two diseases, together data mining and statistical methods were mixed and used.

**Keywords:** Classification · Decision tree · Decision system
Information system

## 1 Introduction

Machine Learning algorithms have been widely used to solve various kinds of data classification problems also in medicine. Ulcerative colitis is a disease that causes long-term inflammation of the colon, which creates irritation or ulcers. This can lead to debilitating abdominal pain and potentially life-threatening complications. It affects only the colon or rectum and destroys the innermost part of the mucosa, not passing through the mouth. Ulcerative colitis causes inflammation and ulcers in the large intestine, which can cause a frequent feeling of need for bowel movement. Exact causes of the disease are not known, therefore their search is extremely important.

## 2 Main Assumptions

We work on data presented in form of a decision table $S = (X, A, V)$, where:

- $X$ is a nonempty, finite set of objects,
- $A$ is a nonempty, finite set of attributes,
- $V = \{V_a : a \in A\}$ is a set of all attributes values.

Additionally, $a : X \rightarrow V_a$ is a function for any $a \in A$, that returns the value of the attribute of a given object [4]. The attributes are divided into different categories: set of


T. K. Dang et al. (Eds.): FDSE 2018, LNCS 11251, pp. 299–309, 2018.
https://doi.org/10.1007/978-3-030-03192-3_23

stable attributes $A_{St}$ (e.g. date of birth, place of birth, color of skin), set of flexible attributes $A_{Fl}$ (blood pressure, weight, sugar level) and set of decision attributes $D$ (e.g. method of treatment, class of illness) such that $A = A_{St} \cup A_{Fl} \cup D$. In this paper we analyze information systems with only one decision attribute $d$. The example of an information system $S$ is represented as Table 1 [4, 8].

**Table 1.** Information system $S$

| $X$ | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| $x_1$ | $a_1$ | $b_2$ | $c_2$ | $d_1$ |
| $x_2$ | $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| $x_3$ | $a_2$ | $b_1$ | $c_1$ | $d_1$ |
| $x_4$ | $a_2$ | $b_2$ | $c_1$ | $d_2$ |
| $x_5$ | $a_2$ | $b_2$ | $c_2$ | $d_2$ |
| $x_6$ | $a_2$ | $b_1$ | $c_1$ | $d_1$ |
| $x_7$ | $a_2$ | $b_2$ | $c_1$ | $d_2$ |
| $x_8$ | $a_2$ | $b_1$ | $c_2$ | $d_2$ |

Information system is represented by eight objects, one stable attribute $a$ (its value cannot be changed), two flexible attributes $b, c$ (their values can change under some conditions) and one decision attribute $d$.

## 3 Classification

The classifier is an algorithm that implements classification, especially in a concrete implementation. There are many different classifiers and many different types of classification results. Moreover it is difficult, especially working with medical data, to decide which classifier is the most effective one for the given set of data. It is already widely known that some classifiers perform better than others on different datasets. Having medical data and decide which classifier gives better results there are two options. First is to put all the trust in an expert's opinion based on his knowledge and experience. Second is to run through each possible classifier that could work on the dataset, and identify rationally the one which performs the best [2, 3]. We use the classification method, where both data mining techniques and statistical methods divide objects into different decision classes.

Mixture of data mining algorithms [6] with statistical methods [2] is an algorithm that creates a step-by-step guide how to determine the output of a new data instance. It is the process of finding a set of models that differentiate data classes and concepts. We use it to predict group memberships for data instances [7]. In first step we describe a set of predetermined classes on the basis of logical regression. Each tuple is assumed to belong to a predefined class as determined by classification attribute, the set of tuples are used for model construction, called training sets. The model can be represented as classification rules, decision trees or mathematical formulas. It is used then for

prediction of future data trends, or eventually reclassification of objects. It estimates the accuracy of the constructed model by using certain test cases. Test sets are always independent of the training sets [3, 6].

### 3.1 Decision Trees

Among the classification methods, one of the most popular method is decision tree. It is particularly attractive because of the intuitive way of knowledge representation understood by people [10, 11]. Initially decision trees appeared in the 1960s in the areas of research on psychology and sociology. In computer science, for the first time they found their application in the works in the 80's [1, 13].

Compared to other methods of classification, decision trees can be constructed relatively quickly. Their main advantage is the clear representation of knowledge, the possibility of using multidimensional data, and scalability with the use of large data sets. Additionally, the accuracy of this method is comparable to the accuracy of other classification methods. However, the main disadvantage of the discussed method is the high sensitivity to the missing values of attributes, because at their bases there is an explicitly expressed assumption of full availability of information gathered in the database. The disadvantages also include the inability to capture the correlation between attributes [13]. Therefore we can use ERID algorithm first, which help us to reduce some missing values in dataset with high accuracy.

Classification trees are used to determine the affiliation of objects to the quality class of a dependent variable. This is done based on measurements of one or more prediction variables. The classification tree presents the process of dividing the set of objects into homogeneous classes. The division is based on the values of the features of the objects, the leaves correspond to the classes to which the objects belong, while the edges of the tree represent the values of the features on the basis of which the division was made [13].

The process of creating a decision tree is based on the recursive division of the teaching set into subsets, which takes place to achieve their homogeneity due to the belonging the objects to classes. The goal is to create a tree with the fewest number of nodes, and as a consequence, the simplest classification rules [1].

The decision tree creation algorithm can be written as follows [7, 10]:

1. For a given set of objects, using ERID algorithm we find all missing values of attributes, compose the containment relation, and make more complete new information system;
2. For more complete set of attributes values corresponding to the set of objects we check whether they belong to the same class (if they belong - end the process, if they do not belong - consider all possible divisions of a given set into all possible homogeneous subsets);
3. Evaluate the quality of each of these subsets according to the previously accepted criterion and select the best one;
4. Split the set of objects on the basis of step 3;
5. Repeat above steps for each of the subsets.

### 3.2 Support Vector Machine (SVM)

Vector transport machine (SVM), which Vladimir Vapnik and Corinna Cortes [15] made for the first time when removing the cover on the floors and/or in the car. SVM is a version of a binary classifier that gives a set of input data and then classifies one device. The goal is to map the $n$-dimensional entrance space to a higher space. Thanks to the new ticket is classified by constructing a linear class. In SVM, a sample of data is viewed as a $p$-dimensional vector that SVM separates with a hyperplane of sets $(p - 1)$. The SVM algorithm has the advantage that it does not affect the minimum minima [14]. We modified this method, and the constraint is softened. Therefore these hyperplanes are built more independently. The main procedure starts with partitioning all negative objects into dense clusters. The same step is repeated for all positive objects also dividing them into dense clusters. To learn a negative rule, we take all objects in one of this negative clusters jointly with all positive objects. The algorithm [12] constructs a minimal number of hyperplanes needed to build classification part of a rule describing this negative cluster. The same procedure is repeated for all the remaining negative clusters. Rules describing positive clusters are constructed the same way. Taking the medical data with 152 instances affected by ulcerative colitis, as an example, we show that the overall support and confidence of rules, extracted from that database, using our strategy [11, 12] is much higher than the confidence and support of rules obtained using methods described in Fig. 1 shows that there are many possible hyper-planes that can perfectly separate the two classes [15]. However, we need to find the best hyperplane that represents the largest distance between the two classes. SVM maximizes the margin between the hyperplane and two classes.

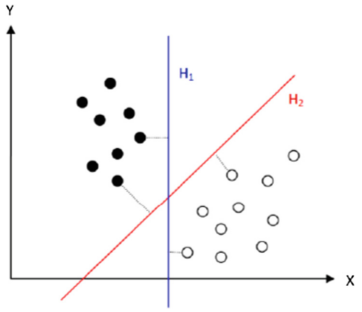


**Fig. 1.** Two separated classes described by the method [15]

In two dimension space, two groups can be separated by a line, using the equation $ax + by \geq c$ for the first group and $ax + by \leq c$ for the second group.

To choose the best possible hyperplane and minimize the risk of overfitting, it is very important to find the one with the maximum margin between the two classes. This is a typical optimization problem that can be solved using the Lagrangian formula. After finding the optimal hyperplane, only the data points closest to the hyperplane will have a positive weight, while others will take zero. Points regarding data in which

distances are closest to the decision surface are called support vectors and are the most critical elements of training data. The position of the hyperplane is shifted when the support vectors are removed.

The distance between the data point $(x_0, y_0)$ and the straight $ax + by + c = 0$ can be measured using the formula below:

$$\frac{|ax_0 + by_0 + c|}{\sqrt{(a^2 + b^2)}}$$

We have $L$ training data, where each instance of $X_i$ has $D$ attributes and two classes: $-1$ and 1. We assume that the training data can be separated in a linear way, therefore we can draw a hyperplane that separates two classes. This hyperplane can be described as $x.w - b = 0$, where $w$ is normal for the hyperplane. $H_1$ is a hyperplane for the first class, and $H_2$ is a hyperplane for the second one. $H_1 : x_i.w - b = 1$ and $H_2 : x_i.w - b = -1$. The perpendicular distance from the hyperplane is $\frac{b}{\|w\|}$. All points that are closest to $H_1$ and $H_2$ are auxiliary vectors.

We define $d_1$ as the distance from $H_1$ to the hyper-plane and $d_2$ as the distance from $H_2$ to the above-mentioned hyperplane. The SVM margin is the distance from $H_1$ to $H_2$ and is expressed as $d_1 + d_2$.

The distance between $H_0$ and $H_1$ expresses the following formula:

$$\frac{|w.x + b|}{\|w\|} = \frac{1}{\|w\|},$$

The distance between $H_1$ and $H_2$ is equal $\frac{2}{\|w\|}$.

The distance between two hyperplanes ($H_1$ and $H_2$) can be maximized by minimizing the value of $\|w\|$. The margin is $\frac{1}{\|w\|}$ and can be maximized using the following formula:

$$\min\|w\| = y_i(x_i.w + b) - 1 \geq 0, \forall_i$$

Minimizing $\|w\|$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$ then using QP optimization (Quadratic Programming). In the next step, find $\frac{1}{2}\|w\|^2$ such that $y_i(x_i.w + b) - 1 \geq 0, \forall_i$. Minimization can be continued through the use of Lagrange multipliers $\alpha$, where $\alpha_i \geq 0, \forall_i$.

$$L = \frac{1}{2}\|w\|^2 - \alpha[y_i(x_i.w + b) - 1 \geq 0 \forall_i]$$

$$L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i[y_i(x_i.w + b) - 1]$$

$$L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i y_i(x_i.w + b) + \sum_{i=1}^{L} \alpha_i$$

For derivatives of 0, we get:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i, \sum_{i=1}^{l} \alpha_i y_i = 0$$

### 3.3 Rating of the Classifier

Each built-in classifier should be evaluated in terms of its quality. For this purpose, two sets of data are necessary. The first, so-called training set is intended for learning the classifier. The second - validation test is used to test the classifier. In both sets, it is necessary to know how the samples belong to the classes. In many cases, the division of data into a teaching and testing set is not given from above. Then, a random division into two disjoint sets can be repeatedly made, usually in such way that the test set is smaller than the teaching one. In such case, we deal with simple validation. Another type of validation is k-fold validation called the k-fold cross-check [9] (called k-fold cross validation). In this method, the input set is divided into $k$ subsets. Then, each of the subsets is a test set, and the classifier is taught on the $k - 1$ of the other subsets. In this way, the validation is repeated $k$ times, and the final result is usually the average of all repetitions [9].

Various metrics are used to evaluate the classifier [5]. In order to present the metrics used in the work, the designations as in Table 1 for different cases of classifier response were adopted depending on the class value for the sample. In the field of machine learning, specifically the problem of statistical classification, the confusion matrix (Table 2), also known as the error matrix [8, 14] is a specific table layout of usually supervised learning (in unprotected learning mode it is usually called matching matrix). Each row in the matrix represents occurrences in the projected class, while each column represents occurrences in the actual class or vice versa [12].

**Table 2.** Confusion matrix if $a$ is taken to be the positive class (e.g. patient has the provided disease)

| | $a$ | $b$ |
|---|---|---|
| Actual $a = 0$ | TP | FN |
| Actual $b = 1$ | FP | TN |

In order to evaluate the quality of a binary classifier, a group of additional metrics should be considered. A true positive (TP) example is the one whose true label is 1 and the classifier has returned such label. The concepts of genuinely negative, false positive and false negative examples (which are denoted as follows: TN, FP, FN) are analogously defined.

Sensitivity – (TPR, hit rate, recall) the probability that the classification will be correct, provided that the case is positive. For a medical case, it may be the probability

that the test performed by a sick patient will show that he has the predicted illness. Sensitivity can be described by the following formula:

$$TPR = \frac{TP}{TP + FN}$$

Specificity – (TNR) the probability that the classification will be correct, provided the case is negative. An example is the probability that a healthy person will not be diagnosed with a test. Specificity is defined by the following formula:

$$TNR = \frac{TN}{TN + FP} = 1 - FPR$$

False positive rate – (FPR) the coefficient of instances falsely classified as a given class, which we write with the following formula:

$$FPR = \frac{FP}{FP + TN} = 1 - TNR$$

False discovery rate – (FDR) error factor type I. The FDR aims to control the expected proportion of "discoveries" that are false (incorrect rejections):

$$FDR = \frac{FP}{FP + TP}$$

Positive predictive value – (PPV, precision) this indicator answers the example question: If the test result is positive, what is the probability that the patient has the illness? We can express the measure using the following formula:

$$PPV = \frac{TP}{TP + FP}$$

Negative predictive value – (NPV) the indicator answers the question: If the test result is negative, what is the probability that the patient is healthy?

$$NPV = \frac{TN}{TN + FN}$$

F1-score – the harmonic mean of precision and sensitivity, and its set of values is the interval [0, 1]. The measure assesses the relationship between sensitivity and precision. However, it does not include true negative results:

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

## 4 Experiments

Our dataset contains clinical data of 152 patients affected by ulcerative colitis. Patients are characterized by 117 attributes and classified into two groups: patients with ulcerative colitis (UC) and patients with Crohn disease (CD). Our goal was to find classification rules. The study group consisted of patients with inflammatory bowel diseases. In the first group, ulcerative colitis was diagnosed (N = 86, women N = 32, men N = 54), and the second group were patients with Crohn disease (N = 66, women N = 32, men N = 34).

Too many variables can negatively impact the performance of the model. As a consequence, the first stages of the study, during which initial data processing is performed, are important. The data can be subjected to selection, transformation, or delete unwanted variables.

After completing the data using ERID and removing variables where the percentage of missing data exceeded 60%, the number of attributes decreased. There are 73 attributes left. Subsequently, all the attributes associated with treatment were excluded from the analysis, since predicates describing the treatment cannot determine the occurrence of the disease. Then, the attributes were selected using significance tests. Finally, a set of attributes was obtained that significantly differed in the two analyzed groups. The next stages of the analysis were carried out using data mining methods. Classification algorithms such as J48, SVM and Random Forest were used. Finally, the best algorithm was selected by analyzing the quality of classification measures (Tables 3, 4, 5, 6 and 7).

**Table 3.** Confusion matrix for J48 algorithm

| Observed effects | Expected effects | |
|---|---|---|
| | UC | CD |
| UC | 81 | 5 |
| CD | 6 | 60 |

After using logical regression model connected with ERID algorithm, the highest values of sensitivity and high specificity were obtained in the case of the Random Forest algorithm. For the aforementioned classifier, the sensitivity value was 100%, which proves the ideal ability to detect patients with CD. The specificity value determining the ability to detect people with UC within 98.48%. After applying the J48 algorithm, sensitivity of 94.19% and specificity of 90.91% were achieved. In the case of SVM, the sensitivity reached 93.02%, and the specificity was 84.85%.

The frequency of false alarms in the case of the J48 algorithm was at the level of 0.09, while the frequency of false discoveries was 0.07. For the SVM and Random Forest algorithms, these values were 0.15 and 0.11 and 0.02 and 0.01 respectively.

In the next step, the predictive properties of the constructed model were determined. The positive precision indicator in the case of the J48 algorithm was at the level

**Table 4.** Confusion matrix for SVM

| Observed effects | Expected effects | |
|---|---|---|
| | UC | CD |
| UC | 78 | 8 |
| CD | 17 | 49 |

**Table 5.** Confusion matrix for Random Forest algorithm

| Observed effects | Expected effects | |
|---|---|---|
| | UC | CD |
| UC | 86 | 0 |
| CD | 1 | 65 |

**Table 6.** The values of the measures

| | FPR | FDR | PPV | NPV | F-score |
|---|---|---|---|---|---|
| J48 | 0.09 | 0.07 | 0.93 | 0.92 | 0.94 |
| SVM | 0.15 | 0.11 | 0.89 | 0.90 | 0.91 |
| Random Forest | 0.02 | 0.01 | 0.99 | 1.00 | 0.99 |

**Table 7.** Sensitivity and specificity

| | Sensitivity | Specificity |
|---|---|---|
| J48 | 94.19% | 90.91% |
| SVM | 93.02% | 84.85% |
| Random Forest | 100.00% | 98.48% |

of 0.93, while the other two methods were respectively: 0, 89 and 0.99. The negative precision value was J48: 0.92, SVM: 0.9 and Random Forest: 1, respectively.

In addition, the value of F1-score, which is a balanced measure, which to a certain extent describes the model as a whole, was calculated. In the first discussed algorithm F1 = 0.92, for the other two F1 = 0.91 for SVM and F1 = 0.99, for Random Forest.

The proposed method was compared with currently used methods. All variables were introduced to the classifier and three algorithms were compared: J48, SVM and Random Forest. The results are shown (Tables 8 and 9).

**Table 8.** The values of the measures

| | FPR | FDR | PPV | NPV | F-score |
|---|---|---|---|---|---|
| J48 | 0.11 | 0.08 | 0.92 | 0.87 | 0.91 |
| SVM | 0.26 | 0.18 | 0.82 | 0.86 | 0.86 |
| Random Forest | 0.06 | 0.05 | 0.95 | 0.97 | 0.97 |

**Table 9.** Sensitivity and specificity

| | Sensitivity | Specificity |
|---|---|---|
| J48 | 89.53% | 89.39% |
| SVM | 90.70% | 74.24% |
| Random Forest | 97.67% | 93.94% |

Sensitivity in the case of the J48 algorithm was 89.53% and reached a value lower by more than 5 percentage points, comparing with the classifier discussed earlier. At the same time, it was the lowest value among the three compared algorithms. For a classifier built using the SVM method, the value discussed was 90.70%, while for Random Forest it was 97.67%. These values, in both cases, were lower compared to the model built on the basis of the developed methodology.

Similar results were obtained for specificity. The measure in question in the case of J48 reached the value of 89.39%, SVM - 74.24%, and for Random Forest - 93.94%. In the case of three algorithms, the level of specificity was lower compared to the classifier discussed earlier.

The instance rate falsely classified as a given class (FPR) has reached the following values for three algorithms respectively: 0.11 (J48), 0.26 (SVM), 0.06 (Random Forest). The type I error rate (FDR) assumed the following levels: 0.08, 0.18, 0.05.

The positive precision value was 0.92 (J48), 0.82 (SVM), 0.95 (Random Forest). The negative pretension for J48 was 0.87, SVM 0.86, Random Tree 0.97.

The harmonic mean of precision and sensitivity, i.e. the measure of F1, achieved high, but less satisfactory values, comparing with the classifier built by using the developed methodology. This value reached the following levels: 0.91 (J48), 0.86 (SVM), 0.97 (Random Forest).

## 5 Conclusion and Future Work

In this work we dealt with the data of patients suffering from ulcerative colitis and Crohn's disease. In order to find rules that distinguish these two diseases, classification methods were used. Three popular methods were compared: methods of decision trees (J48 and Random Forest) and SVM. Patients' data were selected using statistical methods. The proposed method gives better results than the method consisting in the introduction of all attributes to the model. In the future, the obtained classification models will be used to build the rules of action from classification rules to reclassify patients from one class to another (more desirable one).

**Acknowledgements.** This work was supported by MB/WM/8/2016 and financed with use of funds for science of MNiSW. The Bioethical Commission gave the permission for the analysis and publication of our results.

## References


1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont (1984)
2. Cheng, J., Greiner, R.: Learning Bayesian belief network classifiers: algorithms and system. In: Stroulia, E., Matwin, S. (eds.) AI 2001. LNCS (LNAI), vol. 2056, pp. 141–151. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45153-6_14
3. Dardzinska, A.: Action Rules Mining. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35650-6
4. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)
5. Frawley, W., Piatetsky-Shapiro, G., Matheus, C.: Knowledge discovery in databases, an overview. Knowl. Disc. Databases 1–27 (1991)
6. Hand, D., Mannila, H., Smyth, P.: Eksploracja danych. Wydawnictwa Naukowo – Techniczne, Warszawa, 35–61, 91–127, 181–201 (2005)
7. Kasperczuk, A., Dardzinska, A.: Comparative evaluation of the different data mining techniques used for the medical database. Acta Mechanica et Automatica **10**(3), 233–238 (2016)
8. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the International Joint Conference on Artificial Intelligence, vol. 2, pp. 1137–1143 (1995)
9. Powers, D.M.W.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
10. Quinlan, J.R.: Introduction of decision trees. In: Machine Learning, pp. 81–106. Kluwer Academic Publishers (1986)
11. Ras, Z.W., Dardzinska, A., Liu, X.: Rule discovery by axes-driven hyperplanes construction. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. Advances in Soft Computing, vol. 25. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39985-8_62
12. Ras, Z.W., Dardzinska, A., Liu, X.: System ADReD for discovering rules based on hyperplanes, special issue on selected problems in knowledge representation. Int. J. Eng. Appl. Artif. Intell. **17**(4), 401–406 (2004)
13. Raś, Z.W., Dardzińska, A.: Data security and null value imputation in distributed information systems. In: Raś, Z.W., Dardzińska, A. (eds.) Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, vol. 28. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-32370-8_9
14. Stehman, S.V.: Selecting and interpreting measures of thematic classification accuracy. Remote Sens. Environ. **62**(1), 77–89 (1997)
15. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-2440-0