Hybrid Approaches to Solving Classification Problems with Constraints

Oksana Pichugina[a], Olha Matsiy[b]

[a]National Aerospace University "Kharkiv Aviation Institute", 17 Chkalova Street, 61070 Kharkiv, Ukraine
[b]Kharkiv National Automobile and Highway University, Str.Yaroslava Mudrogo 25, 61002 Kharkiv, Ukraine

Abstract

Hybrid approaches (Approach 1 and Approach 2) that combine Supervised Learning with Linear Combinatorial Optimization are offered to solve the problem of minimization of the total delivery expenditure for supplying customers by batches of rocks. Approach 1 applies multi-class classification for customers' assessment delivery expenditures, and Approach 2 solves an additional regression prediction problem. Numeric characteristics utilizing all available information are proposed allowing to evaluate the effectiveness of applying the approaches. The presented way to deal with classification problems with additional constraints can be extended on wide class real-world problems.

Keywords

Machine Learning, Supervised Learning, Combinatorial Optimization, Linear Assignment Problem, classification, regression

1. Introduction

Combinatorial optimization problems (CO problems, COPs) arise in almost every practice area where decisions are made based on the choice of indivisible objects [1]. Since the absolute majority of real COPs are np-hard, difficulties in their solution arise in the development of pseudopolynomial algorithms for their exact solving [2, 3, 4, 5, 6, 7, 8, 9]. On the other hand, the presence of additional constraints does not allow developing effective heuristics. Conditionally, to the class of COPs, it is possible to include the classical problems of Machine Learning, such as classification and clustering. At the same time, the main Supervised and Unsupervised Learning methods heuristic. This is due to the fact that, in these problems, there are no restrictions preventing forming an arbitrarily large sample of feasible solutions to the problem and applying heuristics to them. Indeed, each sample can be assigned to any class in a classification problem and an arbitrary cluster in a clustering problem. However, passing from the classical formulations to a practical problem, we invariably face natural constraints imposed on single elements or their combinations. This can be restrictions on the number of elements in some classes, constraints from above and below, on the distance between elements of the same or different classes, etc. In addition, some additional information is often known, such as misclassification penalties and correct classification rewards. Evidently, it is highly desirable to take them into account in the solution, but this is not always possible when using classical methods of Classification and Cluster Analysis. Meanwhile, having formalized such COPs as discrete or continuous optimization problems, there appears a real opportunity to solve them in a reasonable time exactly or with the accuracy assessment by suitable methods. In this paper, we will consider a practical problem that can be effectively solved by combining Machine Learning methods with classical CO methods.

Consider the following practical problem.

International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2021), September 20-21, 2021, Kharkiv, Ukraine
EMAIL: oksanapichugina1@gmail.com (O. Pichugina); olga.matsiy@gmail.com (O. Matsiy)
ORCID: 0000-0002-7099-8967 (O. Pichugina); 0000-0002-1350-9418 (O. Matsiy)

Problem statement. Rocks are mined in several quarries and supplied to consumers in batches. It is necessary to develop a delivery plan for a certain set of such batches in order to fulfill customers' demand and minimize the total delivery expenditure consisting of delivery costs, expenses for exceeding the price of rocks to the customer requests and penalties for the supply of rock batches of lower quality than provided for by the contract with the consumers.

This problem can be thought of as a classification problem, where the samples are the batches and classes are the consumers. Also, on the one hand, additional constraints are imposed on the delivery volume to particular consumers. On the other hand, a target function appears expressing the total delivery expenditure.

The presence of these two essential components deduce the problem from the standard classification problems, respectively, do not allow its solving by standard methods of classification. Therefore, we need new approaches to solving this problem. This is what this work is about. This paper offers two hybrid approaches (Approach 1 and Approach 2), combining Supervised Machine Learning with classical Combinatorial Optimization methods dealing with a generalized Linear Assignment Problem that directly maps these batches to the set of consumers. Approach 1 solves one more classification problem at the first stage of Machine Learning, which identifies a class of rock batches, and Approach 2 solves this problem, where the cost of a batch of rocks is predicted.

Supposedly, the proposed way will allow, on the one hand, to use historical data, namely, to apply Machine Learning on batches of rocks previously mined in these quarries, and on the other hand, to implement the linear programming method in the second phase of the approaches.

2. Prerequisites

2.1. Supervised Learning elements

Practical Machine Learning (ML) mostly uses Supervised Learning (SL) [10, 11, 12, 13]. In SL, you have an input variable vector (x) and an output variable (y) and you use a SL algorithm to learn the mapping function from the input x to the output $y = f(x)$, where $x \in X$. The goal is to approximate the mapping function f so well that when you have new input data x' that you can predict the output variables y' for that data. We come to regression or classification problems depending on which values, discrete or continuous, takes the function f. Particularly, if $f : X \to \mathbb{R}1$, the problem of its finding is a regression problem (RP) [10]. In contrast, if $f : X \to \mathbb{Y}$, where $|\mathbb{Y}| < \infty$, the problem under consideration is a classification problem (CP) [10].

In other words, a CP [10, 14] is a problem of identifying to which class a new instance belongs based on available class membership for instances in a training set (TS).

Instances to which regression/classification is applied form a test set (TeS).

In classification, a TS-instance is given by a feature tuple x and a class label y. A TeS-instance is given by a feature tuple x', while a class label y' is unknown and needs to be found.

In regression, y - is a real value representing outcome, while the task is to predict the real value y' for the input vector x.

x can be a numeric vector but not necessarily since features presented in x-components characterize certain instance' properties, which can be categorical, ordinal, integer-valued, or real-valued.

If categorical or ordinal features are present, a preprocessing stage is required making mapping x into Euclidean space:

$$x \xrightarrow{\phi} x' \in \mathrm{X} \subset \mathrm{R}^L. \tag{1}$$

Let

$$\mathrm{C} = \{C_0, ..., C_s\}$$

be a set of classes.

To X, we will refer to as an instances' space and to

$$\mathrm{Y} = \{0, ..., s\} \tag{2}$$

as a class label space.

A class $C_{i_0} \in \mathrm{C}$ of cardinality $|C_{i_0}| = \max_{i \in J_s^0} |C_i|$ is called a major class (positive class), the one of size $|C_{i'_0}| = \min_{i \in J_s^0} |C_i|$ is called a minor class (negative class). A classification algorithm (CA) is intended to train a classifier, which is a function mapping an instances' space X into a class label space (2):

$$f : \mathrm{X} \to \mathrm{Y}. \tag{3}$$

Type of function f, linear or not, defines two large groups of classification algorithms – linear CAs and nonlinear CAs (see overview in [15, 11]). In addition, a wide group of highly effective methods called ensemble CAs (ECAs) exists [16, 17, 18].

Popular CAs are:

1. Linear CAs:
   a) Logistic Regression (generalized linear model, LR) [19];
   b) Linear Support Vector Machine Classification (SVM) [20];
   c) Naive Bayes (NB) [21];
   d) Linear Discriminant Analysis (LDA) [22];
2. Nonlinear CAs:
   a) Decision Tree (DT) [23] - C4.5 [24], CART [25];
   b) k-Nearest Neighbors (KNN) [26];
   c) Artificial Neural Networks (Deep Learning, DL) [27];
   d) Kernel SVM (KSVM) [28];
   e) Quadratic Discriminant Analysis (QDA) [22];
3. Ensemble CAs:
   a) Bagged Decision Trees (BDT);
   b) Random Forest (RF) [25];
   c) Extra Trees (ET);
   d) Adaptive Boosting (AdaB) [29];
   e) Gradient Boosting Machine (GBM) [30];
   f) Stochastic Gradient Boosting (SGB) [31];
   g) Extreme Gradient Boosting (eXtreme Gradient Boosting,XGBoost, XGB) [30].

A regression algorithm (RA) is targeted to train a regressor, which is a function mapping an instances' space X into a prediction space $\mathrm{Z} \subset \mathrm{R}^1$:

$$g : \mathrm{X} \to \mathrm{Z}. \tag{4}$$

In SL, most of the algorithms are CAs. Some of them are generalized to regression. Among regression algorithms (RAs) are:

1. Linear Regression (LR) [13];
2. Linear Support Vector Machine Regression (SVR) [32];
3. Decision Tree Regression (DTR) [23];
4. Deep Learning Regression (DLR) [33];
5. Random Forest regression (RFR) [25, 34].

2.2. Linear Assignment Problem

Assignment problems (APs) deal with the question of how to assign n items (jobs) to n machines (workers) in the best possible way. APs consist of two components: the assignment as an underlying combinatorial structure and an objective function describing the "best way" [35, 36, 37].

Mathematically an assignment can be seen a bijective mapping of a finite set $J_n = \{1,...,n\}$ into itself, i.e., an n-permutation of J_n forming a permutation set Π_n. Assignments can be modelled in different ways: every permutation $\pi \in \Pi_n$ corresponds in a unique way to a permutation matrix $X_\pi = (x_{ij})_{i,j}$ with $x_{ij} = 1$ for $j = \pi(i)$ and $x_{ij} = 0$ for $j \neq i$. If the objective function is linear on X-components, the AP is called linear (LAP); if quadratic, it is called a quadratic AP (QAP) etc.

LAP integer programming model is [35]: find $X \in \Pi' n$ such that

$$F(X) = \sum_{i,j=1}^{n} c_{ij} x_{ij} \min; \tag{5}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i \in J_n; \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j \in J_n; \tag{7}$$

$$x_{ij} \in \{0,1\}, i, j \in J_n. \tag{8}$$

Here, $C = \left(c_{ij}\right)_{i,j}$ is the cost matrix of the assignments.

LAP is polynomially solvable [36]. It is generalized differently, such as allowing multiple assignments, when multiple machines can be assigned to one worker [35, 37]. As a result, the classical LAP-model (3)-(6) becomes

$$F(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \to \min; \tag{9}$$

$$\sum_{j=1}^{m} x_{ij} = 1, i \in J_n; \tag{10}$$

$$\sum_{i=1}^{n} x_{ij} = n_j, j \in J_m; \tag{11}$$

$$x_{ij} \in \{0,1\}, i \in J_n, j \in J_m, \tag{12}$$

where $m \leq n$, $n_1 + ... + n_m = n$. The AP (9)-(12) as it is reducible to LAP with columns having multiplicities $n_1,...,n_m$, respectively.

3. Approaches 1,2 description

3.1. Modelling

Let us build a mathematical model of the problem stated in Introduction. Let $C \in \mathrm{C}$ be class of rocks such as granite, marble, limestone, etc., $y = y(C) \in \mathrm{Z}_+$ be its number (a label), while $z = p(y)$ be the known price of the rock batch of class y.

Suppose that several rocks can be mined in a quarry, rock samples are automatically processed. Namely, their size, shape, spectral characteristics are measured, after which the type of rock is automatically determined. Based on this information, the rock class of the entire batch is predicted. Suppose all the features are numeric, thus, $\mathrm{X} = X$.

As a training set TS, we use historical data on rock batches for which their type is known, then $x_i \in X$ is a vector of parameters of the shape, size and spectral characteristics of a training batch

($i \in J_n$), while $x'_i \in X'$ is a vector the same parameters for the batch involving in our problem, i.e. an element of our training set TS ($i \in J_n$). Respectively, $Y = (y_i)_{\in J_n}$ are real labels of TS, $Z = (z_i)_{i \in J_n}$, where $z_i = p(y_i)$ are the real cost of the batch i in a TS.

In order to assess the effectiveness of our approaches numerically, we offer two metrics that use actual labels of the test set. So, let $Y' = (y'_i)_{i \in J_{n'}}$ be real labels of TeS, then $Z' = (z'_i)_{i \in J_{n'}}$, where $z'_i = p(y'_i)$ will be the real cost of the batch i in a TeS ($i \in J_{n'}$).
multiplicities $n_1, ..., n_m$, respectively.

3.1.1. Stage 1

The first stage differs for these two approaches that we offer.

Approach 1. Here, the training set has the form of

$$TS = \left\{ \left\langle x_i, y_i \right\rangle \right\}_{i \in J_n},$$

while the test set is

$$TeS = \left\{ \left\langle x'_i \right\rangle \right\}_{i \in J_{n'}}.$$

Stage 1 consists of solving a CP on TS and applying a found classificator f for predicting labels of a TeS. It results in a multiset

$$\hat{Y}' = \{\hat{y}'_i\}_{i \in J_{n'}}, \text{where} \hat{y}'_i = f(\hat{x}'_i), i \in J_{n'}.$$

$\hat{Y}$ can be used for prediction of the cost of the batches, particularly, the collection of the prediction is

$$\hat{Z}' = \{\hat{z}'_i\}_{i \in J_{n'}}, \text{where} \hat{z}'_i = p(\hat{y}'_i), i \in J_{n'}.$$

Approach 2. At Stage 1, we set a regression problem. In this case, the training set looks like this:

$$TS = \left\{ \left\langle x_i, z_i \right\rangle \right\}_{i \in J_n}.$$

The result of applying a regression algorithm to the set TS will be the regressor g. Applying it to our test set TeS, we obtain a set

$$\tilde{z}'_i = g\left(x'_i\right), i \in J_{n'},$$

where $\tilde{z}'_i$ is a predicted cost of a rock batch i obtained by Approach 2.

Note that, in this step, we can use standard regression metrics to evaluate the quality of this prediction.

Now, let us move to the second Combinatorial Optimization stage.

3.1.2. Stage 2

This stage will be common for our approaches. The only difference is that, in Approach 1, $\hat{Z}'$ is used as a vector of predicted cost while, in Approach 2, it will be $\tilde{Z}$.

Suppose that for the test set TeS, we also know from which quarry k out of a set J_K rocks come. Accordingly, it is known their location, distances between the quarries is the same as the location of the

consumers and the distance between them and to the quarries. This makes it possible to find the cost of delivering a batch from each of the quarries to the consumers.

Let $B_1 \dots B_m$ be the customers, $A = \left(a_{kj}\right)_{k \in J_k,\ j \in J_m}$ be a cost matrix, where a_{kj} is the cost of delivery of the batch from the k-th quarry to the consumer B_j.

Approach 2. With the help of A, let us build another cost matrix

$$\tilde{A}' = \left(\tilde{a}'_{ij}\right)_{i \in J_{n'}}, \tag{13}$$

where $\tilde{a}'_{ij}$ is the predicted (with the help of Approach 2) total delivery expenditure of the batch i to the consumer B_j, $j \in J_m$.

Here, a'_{ij} includes, as one component, the delivery cost $a_{k_i j}$, where k_i is the quarry where the batch i was mined.

The other two components directly depend on how the real cost of the batch relates to the cost of the rocks ordered by the customer and on the penalties that the consumers regulate for violating the delivery quality.

So, let b_j be the cost of the rock ordered by the customer B_j, v_j be the volume of ordered rock batches, d_j be the penalty for delivering a low-quality batch i (per unit).

Then, if $z > b_i$ is satisfied for $z \in \tilde{Z}' = \left\{\tilde{z}'_i\right\}_{i \in J_n}$, then the mining company owners will receive less money for exceeding the quality of rock batches of the ordered level. If $z = b_i$, then there will be no additional costs. Finally if $z < b_i$, then the company will have to pay $\left(b_i - z\right) d_i$ currency units as a penalty for the delivery of poor quality rocks. Thus

$$\tilde{a}'_{ij} = a_{k_i j} + \begin{cases} \tilde{z}'_i - b_i, \text{if } \tilde{z}'_i \geq b_i; \\ \left(b_j - \tilde{z}'_i\right) d_j, \text{if } \tilde{z}'_i < b_i. \end{cases} \tag{14}$$

Having filled the matrix $\tilde{A}'$, we can proceed to the optimization step, in which it is decided where to send which batch in order to minimize the total delivery expenditure. This can be represented as the need to assign the batches to the consumers subject to the constraint on fulfilling the customers' orders.

The mathematical model is: find a set

$$T = \left(t_{ij}\right)_{i \in J_{n'},\ j \in J_m}$$

of unknowns such that

$$t_{ij} = \begin{cases} 1, \text{if the batch } i \text{ goes to the customer } B_j; \\ 0, \text{otherwise.} \end{cases} \tag{15}$$

The objective function is

$$\tilde{F}'(t) = \sum_{i=1}^{n'} \sum_{j=1}^{m} \tilde{a}'_{ij}\ t_{ij} \to \min \tag{16}$$

under constraints:

– the batch is either supplied or not supplied at all, which can be expressed by inequality

$$\sum_{j=1}^{m} t_{ij} \leq 1,\ i \in J_{n'}, \tag{17}$$

– the costumers' orders are obligatory for execution:

$$\sum_{i=i'_2}^{n'} t_{ij} \geq v_j,\ j \in J_m \tag{18}$$

The problem (15)-(18) is a generalized linear assignment problem that can be solved in polynomial time in dimension by methods such as Hungarian. Let $T'^* = \left(t'^*_{ij}\right)_{i \in J_{n'}, j \in J_m}$ be an optimal solution to this AP. For the consistency of the problem, the condition

$$\sum_{j=1}^{n} v_j \geq n \tag{19}$$

must be satisfied, meaning that the number of rock bathes is sufficient to meet the consumer demand.

Approach 1. Here, the difference of this stage in Approach 1 compared to Approach 2 will be outlined.

The formula (13) is replaced by

$$\hat{A}' = \left(\hat{a}'_{ij}\right)_{i \in J_{n'}}, \tag{20}$$

where $\hat{a}'_{ij}$ is the predicted (with the help of Approach 1) total delivery expenditure of the batch i to the consumer B_j, $j \in J_m$.

The formula (14) becomes

$$\hat{a}'_{ij} = a_{k_i j} + \begin{cases} \hat{z}'_i - b_i, \text{if } \hat{z}'_i \geq b_i; \\ \left(b_j - \hat{z}'_i\right) d_j, \text{if } \hat{z}'_i < b_i. \end{cases} \tag{21}$$

Respectively, the objective function (16) becomes

$$\hat{F}'(t) = \sum_{i=1}^{n'} \sum_{j=1}^{m} \hat{a}'_{ij}\ t_{ij} \to \min. \tag{22}$$

Let a solution to the AP (22) with constraints (15), (17)-(19) be denoted $T''^* = \left(t''^*_{ij}\right)_{i \in J_{n'}, j \in J_m}$.

3.2. Comparison metrics

To compare the quality of the solutions T'^*, T''^*, we solve one more AP, where actual costs of training set's items are used when comparing the optimal global solution with these two. This is an AP

$$F'(t) = \sum_{i=1}^{n'} \sum_{j=1}^{m} a'_{ij}\ t_{ij} \to \min, \tag{23}$$

subject to constraints (15), (17), (18), where

$$A' = \left(a'_{ij}\right)_{i \in J_{n'}} \tag{24}$$

is a matrix of the total delivery expenditures, for instance, a'_{ij} is the total delivery expenditure for the batch i delivered to the consumer B_j, $j \in J_m$. A' can be found similarly to (21) as

$$a'_{ij} = a_{k_i j} + \begin{cases} z'_i - b_i, \text{if } z'_i \geq b_i; \\ \left(b_j - z'_i\right) d_j, \text{if } z'_i < b_i. \end{cases} \quad (25)$$

Let T^* be an optimal solution for the AP (15), (17), (18), (23) and $F'^* = F'(T^*)$ be its optimal value. Substituting T'^*, T''^* into the target function, we get not estimated but an actual delivery expenditure of implementing the plans T'^*, T''^* in real life. Let us denote $\tilde{F}'^* = F'(T'^*)$, $\hat{F}'^* = F'(T''^*)$. If $\tilde{F}'^* = \hat{F}'^*$, then Approaches 1, 2 work equally, if $\tilde{F}'^* < \hat{F}'^*$, Approach 1 showed itself better, if $\tilde{F}'^* > \hat{F}'^*$, then it is the worst.

To understand how much we lost applying the predicted cost rather than real, we analyze the increments

$$\tilde{\Delta} = \tilde{F}'^* - \tilde{F}'^*, \hat{\Delta} = \hat{F}'^* - \hat{F}'^* \quad (26)$$

representing absolute error if applying Approach 1 and Approach 2, respectively. The corresponding relative errors can be found using already found values

$$\tilde{\delta} = \tilde{\delta} / \tilde{F}'^*, \hat{\delta} = \hat{\Delta} / \hat{F}'^*. \quad (27)$$

In our opinion, the metrics (26), (27) highlight specifics of the problem under consideration better than any classification metrics applicable to multi-classification problems [14, 15, 38, 39] such as the accuracy, balanced accuracy, recall, precision, F -score, G -mean, etc.

4. Discussion of Results and Future work

The absolute majority of research literary sources relating to Combinatorial Optimization dealing with exact solutions and Machine Learning, these two research fields are exiled. Perhaps, this is because the combinatorial optimization problems are highly complex, including designing qualitative heuristics, while Machine Learning is entirely based on heuristics.

The main contribution is that this paper presents an innovative approach to how these two research domains can be combined. The second one is the mathematical models developed here. Together with the scheme of both approaches, they provide a road map of implementing in programming languages such as Python, where powerful libraries in Machine Learning and Integer Linear Programming are embedded.

Areas of further work are:

– Generalization to a higher class of practical problems.

– Collecting data on the location of Ukrainian open pits and the structure of their rocks, followed by a computational experiment using the presented approaches.

– Generalization to a class of stochastic programming problems, in which the predicted price is involved and predictions of the standard deviations from the average cost.

At the next stage of the research, we plan to apply a generalization of the classical regression model with one output for two outcomes - the predicted cost and standard deviation. In this regard, this stage is supposed to be carried out using neural networks.

5. Conclusion

This paper offers two hybrid approaches to minimize the total delivery expenditure for supplying customers by rock batches. They combine Machine Learning techniques with classical Linear Combinatorial Optimization methods. Approach 1 solves an auxiliary classification problem at the stage of Machine Learning, while Approach 2 solves an additional regression problem instead. Numerical evaluation characteristics for assessing the approaches are presented. Ways to develop the contribution theoretically and experimentally are outlined.

6. References

[1] Y. G. Stoyan, S. V. Yakovlev, Theory and Methods of Euclidian Combinatorial Optimization: Current Status and Prospects, Cybernetics and Systems Analysis 56 (2020) 366-379. doi:10.1007/s10559-020-00253-6.

[2] O. Pichugina, O. Matsyi, Boolean Satisfiability Problem: Discrete and Continuous Reformulations with Applications, in: 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, pp. 623-627. doi:10.1109/TCSET49122.2020.235507.

[3] O. Pichugina, N. Muravyova, Data Batch Processing: Modelling and Applications, in: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 765-770. doi:10.1109/PICST51311.2020.9467928.

[4] O. Pichugina, On Modelling of Computer Cluster Optimization Problem With Applications, in: 2020 IEEE KhPI Week on Advanced Technology (KhPIWeek), 2020, pp. 429-436. doi:10.1109/KhPIWeek51551.2020.9250145.

[5] O. Pichugina, Mathematical Modeling Of Transport Logistics Problems, in: 2020 IEEE 2nd International Conference on System Analysis Intelligent Computing (SAIC), 2020, pp. 1-5. doi:10.1109/SAIC51296.2020.9239155.

[6] O. Pichugina, Diet-Menu Problem Modelling And Applications, in: 2020 IEEE 2nd International Conference on System Analysis Intelligent Computing (SAIC), 2020, pp. 1-5. doi:10.1109/SAIC512 96.2020.9239149.

[7] O. Pichugina, New Approaches to Modelling Covering Problems in Monitoring Optimization, in: 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S T), 2019, pp. 300-304. doi: 10.1109/PICST47496.2019.9061386.

[8] S. Yakovlev, O. Kartashov, O. Pichugina, K. Korobchynskyi, Genetic Algorithms for Solving Combinatorial Mass Balancing Problem, in: 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2019, pp. 1061-1064. doi:10.1109/UKRCON.2019. 8879938.

[9] S. V. Yakovlev, On Some Classes of Spatial Configurations of Geometric Objects and their Formalization, Journal of Automation and Information Sciences 50 (2018) 38-50. doi:10.1615/JAutomatInfScien.v5 0.i9.3 0.

[10] E. Alpaydin, Introduction to machine learning, Adaptive computation and machine learning, 2nd ed., MIT Press, Cambridge, Mass, 2010. OCLC: ocn317698631.

[11] C. M. Bishop, Pattern Recognition and Machine Learning, 1st ed. 2006. corr. 2nd printing 2011 ed., Springer, New York, 2011.

[12] M. Kubat, An Introduction to Machine Learning, 2 ed., Springer International Publishing, 2017. doi:10.1007/978-3-319-63913-0.

[13] D. Forsyth, Applied Machine Learning, Springer International Publishing, 2019. URL: https://www.springer.com/gp/book/9783030181130. doi:10.1007/978-3-030-18114-7.

[14] C. Drummond, Classification, in: C. Sammut, G. I. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, Boston, MA, 2010, pp. 168-171. doi:10.1007/978-0-387-30164-8_111.

[15] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4 ed., Morgan Kaufmann, Amsterdam, 2016.

[16] T. G. Dietterich, Machine-learning research; four current directions, AI Magazine 18 (1997) 97-137.

[17] G. Brown, Ensemble Learning, in: C. Sammut, G. I. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, Boston, MA, 2010, pp. 312-320. doi:10.1007/978-0-387-30164-8\

[18] W. Fan, Cost-sensitive, Scalable and Adaptive Learning Using Ensemble Methods: Theory and Application, VDM Verlag, Saarbrucken, 2009.

[19] P. McCullagh, J. A. Nelder, Generalized Linear Models, 2nd edition ed., Chapman and Hall/CRC, Boca Raton, 1989.

[20] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273-297. doi:10.1007/BF00994018 .

[21] P. Domingos, M. Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Machine Learning 29 (1997) 103-130. doi:10.1023/A:1007413511361.

[22] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, Wiley-Interscience, Hoboken, N.J, 2004.
[23] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and Regression Trees, 1st edition ed., Chapman and Hall/CRC, Boca Raton, Fla., 1984.
[24] J. R. Quinlan, C4.5: Programs for Machine Learning, 1st ed., Morgan Kaufmann, San Mateo, Calif, 1992.
[25] L. Breiman, Classification and Regression Trees, 1st edition ed., Chapman and Hall/CRC, Place of publication not identified, 2017.
[26] N. S. Altman, An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, The American Statistician 46 (1992) 175-185. doi:10.2307/2685209.
[27] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436-444. doi:10.1038/nature14539.
[28] I. W. Tsang, J. T. Kwok, P.-M. Cheung, Core Vector Machines: Fast SVM Training on Very Large Data Sets, Journal of Machine Learning Research 6 (2005) 363-392. URL: http://jmlr.org/papers/v6/tsang05a.html.
[29] Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: G. Goos, J. Hartmanis, J. Leeuwen, J. G. Carbonell, J. Siekmann, P. Vitanyi (Eds.), Computational Learning Theory, volume 904, Springer Berlin Heidelberg, Berlin, Heidelberg, 1995, pp. 23-37. doi:10.1007/3-540-59119-2_166.
[30] J. H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, The Annals of Statistics 29 (2001) 1189-1232. URL: https://www.jstor.org/stable/2699986.
[31] J. H. Friedman, Stochastic Gradient Boosting, Computational Statistics and Data Analysis 38 (1999) 367-378.
[32] An Introduction to Statistical Learning: with Applications in R, 1st ed. 2013, corr. 5th printing 2015 edition ed., Springer, New York, 2013.
[33] S. Skansi, Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence, Undergraduate Topics in Computer Science, Springer International Publishing, 2018. doi: 10.1007/978-3-319-73004-2.
[34] L. Breiman, Random Forests, Machine Learning 45 (2001) 5-32. doi:10.1023/A: 1010933404324.
[35] R. E. Burkard, E. A$ela, Linear Assignment Problems and Extensions, in: D.-Z. Du, P. M. Pardalos (Eds.), Handbook of Combinatorial Optimization, Springer US, 1999, pp. 75-149. doi:10.1007/978-1-4757-3023-4_2.
[36] D. P. Bertsekas, Network Optimization: Continuous and Discrete Models, Athena Scientific, Belmont, Mass, 1998.
[37] Nonlinear Assignment Problems: Algorithms and Applications, softcover reprint of hardcover 1st ed. 2001 edition ed., Springer, Boston, MA, 2010.
[38] Y. Sun, M. S. Kamel, Y. Wang, Boosting for Learning Multiple Classes with Imbalanced Class Distribution, in: Sixth International Conference on Data Mining (ICDM'06), 2006, pp. 592-602. doi:10.1109/ICDM.2 006.2 9, iSSN: 2374-8486.
[39] L. Kirichenko, T. Radivilova, V. Bulakh, Machine Learning in Classification Time Series with Fractal Properties, Data 4 (2019) 5. doi:10.3390/data4010005.