Various Machine Learning Methods Efficiency Comparison in Application to Inorganic Compounds Design

© O.V. Sen'ko[1] © N.N. Kiselyova[2] © V.A. Dudarev[2] © A.A. Dokukin[1] © V.V. Ryazanov[1]
[1] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia
[2]Institution of Russian Academy of Sciences A.A. Baikov Institute of Metallurgy and Materials Science RAS, Moscow, Russia
senkoov@mail.ru kis@imet.ac.ru vic@imet.ac.ru alex_dok@mail.ru rvvccas@mail.ru

Abstract. Various machine learning methods («Recognition» package and «Scikit-learn» package for Python) accuracy comparison was made on example of inorganic chemistry tasks solution. The cross-validation and the ROC-analysis were applied to accuracy estimation.

Keywords: machine learning methods comparison, pattern recognition, «Recognition», «Scikit-learn».

1 Introduction

Machine learning (ML) methods are widely used in the inorganic compounds formation predicting and their properties estimation [1-7]. The paper [5] contains a statistical analysis of popularity of various ML methods that applied to inorganic materials science. However, in spite of these methods success for numerous tasks solution in this subject field, no effort of accuracy comparison of wide variety of methods was made using ROC-analysis.

To solve this task the subject field particularities must be taken into account. In particular, it is obvious that an attribute description has a composite structure: the set of chemical elements parameters (the components of an inorganic substance) is repeated as many times as there are elements included into the compound. Due to periodical dependence of chemical elements properties on their atomic numbers the strong correlation within sets of each component parameters is observed. Relative informativeness of individual element's properties is low. For this reason, the simpler compounds properties (e.g., simple oxides, halogenides, chalcogenides, etc.) as well as the algebraic functions of components' properties are used. Although these parameters are studied very well but there are gaps of properties' values (incomplete data). They are filled in a variety of ways. For example, the periodic dependences of elements' parameters on their atomic numbers and the appropriate interpolation and extrapolation are used. The large asymmetry of training sample sizes for different classes is a peculiarity in inorganic chemistry tasks. Very often the least representative classes (as a rule – newly discovered classes of substances) are the most interesting point to chemists. The experimental errors and discrepancies of inorganic compounds classification in training samples are yet another problem at compound design that decreases a prediction accuracy drastically. Doubtless, that an accuracy depends on attribute description informativeness and training sample representativeness. Therefore, to evaluate various ML methods we have chosen a number of tasks with highly reliable predictions (more than 85 % according to the later experimental verification) [6, 7].

2 Prediction accuracy estimation methods

The cross-validation (CV) on training sample of objects is the most widely used universal and reliable tool for machine learning quality estimation. At that a number of recognition error can be taken into account. However, one of the problems in ML accuracy estimation task is the recognition efficiency determination in the asymmetrical classes case where the number of different classes objects differs significantly. This situation is very common in cases when only a very few new materials were obtained with the important practical properties and a search for analogues of these substances that are not yet synthesized allows an experimental researches time and cost reduction. In the majority of ML methods application cases the standard decision rule minimizes the total number of erroneous predictions. It results in good recognition of compounds from the large class and in bad recognition of substances representing small class. As a result, the overall recognition accuracy gives poor notion of the efficiency of one or another method or one or another attribute description. The Receiver Operating Characteristic (ROC) analysis application is an alternative approach. It allows a recognition accuracy comparison for the targeted and alternative classes at variation of cut-offs which identifies belonging to different classes.

The following prediction accuracy estimation procedures were used in this analysis fulfilling. The available training sample is divided into two nonintersecting stratified subsamples which were later used to train and assess simple and collective methods independently. Further, the ROC-analysis is carried-out and the Area Under Curve (AUC) measure is calculated. As a rule, in collective decision making the methods with AUC more than some fixed threshold value is used in

Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018), Moscow, Russia, October 9-12, 2018

prediction.

3 The test tasks

3.1 Prediction of formation of compounds with the composition A_2BCHal_6 (A and C are various monovalent metals; B are trivalent metals; and Hal is F, Cl, or Br) [7].

2 classes:
4. formation of the compound – 744 examples;
5. nonformation of the compound – 170 examples.

137 attributes including 3 the most informative algebraic functions of the initial attributes.

3.2 Prediction of formation and crystal structure type of compounds with composition A_2BCHal_6 [7].

4 classes:
1. elpasolites – 283 examples;
2. compounds with the Cs_2NaCrF_6 crystal structure type – 19 examples;
3. another crystal structure types – 57 examples;
4. nonformation of the compound – 83 examples.

134 attributes.

3.3 Prediction of formation of compounds with the composition $ABHal_3$ (A are various monovalent metals; B are bivalent metals; Hal is F, Cl, Br, or I) [6].

2 classes:
1. formation of the compound – 237 examples;
2. nonformation of the compound – 107 examples.

88 attributes.

3.4 Prediction of formation and crystal structure type of compounds with composition $ABHal_3$ [6].

6 classes:
1. perovskites – 46 examples;
2. compounds with the $GdFeO_3$ crystal structure type – 20 examples;
3. compounds with the $CsNiCl_3$ crystal structure type – 38 examples;
4. compounds with the NH_4CdCl_3 crystal structure type – 23 examples;
5. another crystal structure types – 39 examples;
6. nonformation of the compound – 111 examples.

88 attributes.

The most important attribute sets were selected using the program based on the method [8].

4 The analysis of obtained results

The Table 1 contains the efficiency estimation results for single machine learning methods. The following algorithms notations were used ("*Recognition" package* [9]):

- ECA – the estimates calculation algorithm (fixed size of support sets = 1), Leave-One-Out CV (LOOCV);
- SBT – the search for the best test (maximal number of ε- thresholds for one attribute = 5; maximal size of sample = 20; number of samples of the same size = 3; percent of tests using in recognition – 10 %; unitary weights), LOOCV;
- TLS – the two-dimensional linear separators method (bias step – 0; right part components – 0.1; number of iterations – 10000; number of start iteration – 100; percentage of removed objects – 1; step – 100; threshold of regularity selection – 80 %), 10-fold CV;
- BDT – the binary decision tree learning (maximal number of nodes (interior nodes) – 15; minimal significant value of entropy reduction – 0.2; minimal number of objects in leaf nodes – 5), LOOCV;
- LDF – the linear Fisher discriminant (confidence threshold for correlation coefficient – 0), LOOCV;
- LM – the linear machine method (bias step – 0; right part components – 0.1; number of iterations – 10000; number of start iteration – 100; percentage of excluding objects – 1; step – 100), LOOCV;
- LoReg – the voting algorithm where estimations for classes are calculated with the help of voting by logical regularities system ("greedy" way; number of intervals - 5; maximal number of iterations – 100000; beginning of removal – 100; percentage of removed inequalities – 1%; removal step– 100; minimal rate of objects – 0.1; number of random permutations – 3), 10-fold CV;
- MNN – the multiplicative neural network algorithm (number of iterations – 1000), LOOCV;
- MP – the multilayer perceptron (neural network configuration: number of hidden layers – 3 (number of neurons in layer - 10); number of training iterations – 3000; activation function – sigmoid; training speed – 0.1; moment of inertia – 0; lack of criterion function increase if there is no increase during last 1000 iterations then the speed is decreased in 2 times), 10-fold CV;
- ANN – the artificial neural network learning using back-propagation (neural network configuration: number of hidden layers – 3 (number of neurons in layer - 10); number of training iterations – 500; activation function – sigmoid; training speed – 0.1; threshold – 0.1; lack of criterion function increase if there is no increase during last 100 iterations then the speed is decreased in 2 times), 10-fold CV;
- KNN – the k-nearest neighbors method (number of nearest neighbors – 1; prior class probabilities are took into account), LOOCV;
- SVM – the support vector machine (penalty coefficient – 5; kernel function type – Gaussian; kernel function parameter – 6; maximal number of iterations – 500, 10-fold CV);
- SWS – the statistical weighted syndromes (rapid mode; number of partition borders – 1; optimized criteria threshold – 4.5; representativeness threshold – 0.5; instability threshold - 0.2; denial zone – 0.1), 10-fold CV;
- DTA – the deadlock test algorithm (test searching algorithm – effective; divisor of ε- thresholds = 2;

maximal size of sample = 20; the number of subsamples of the same size = 3), LOOCV.

"Scikit-learn package for Python" [10] - 10-fold CV:

- LIR – linear_model.LinearRegression;
- R – linear_model.Ridge;
- L – linear_model.Lasso;
- EN – linear_model.ElasticNet;
- LL – linear_model.LassoLars;
- OMP – linear_model.OrthogonalMatchingPursuit;
- BR – linear_model. BayesianRidge;
- HR –linear_model. HuberRegressor;
- KR – KernelRidge;
- PLS – PLSRegression;
- SGDC – linear_model.SGDClassifier;
- P –linear_model.Perceptron;
- PACH – the passive aggressive classifier (loss='hinge');
- PACS – the passive aggressive classifier (loss='squared_hinge');
- LSVC – linear SVC;
- NSVC1 – nuSVC (nu=0.1);
- NSVC3 – nuSVC (nu=0.3);
- LR – linear_model.LogisticRegression;
- GPC – Gaussian process classifier;
- GNB – Gaussian naive Bayes;
- DTC – tree.DecisionTreeClassifier;
- KNN – KNeighborsClassifier (n_neighbors=5);
- MP – neural_network.MLPClassifier;
- BC – ensemble.BaggingClassifier;
- RFC – ensemble.RandomForestClassifier;
- ETC – ensemble.ExtraTreesClassifier;
- ABC – ensemble.AdaBoostClassifier;
- GBC – ensemble.GradientBoostingClassifier.

Table 1 The accuracy estimation of various single machine learning methods

Algorithm	CV accuracy, %	AUC
System "Recognition" – *Task 1*		
SVM	89.8	0.916
LM	90.7	0.884
ANN	89.1	0.880
SWS	*82.3*	0.872
LoReg	87.8	0.877
TLS	84.7	0.863
DTA	84.0	0.861
MNN	87.1	0.827
MP	84.5	0.816
KNN	87.6	0.805
ECA	*83.6*	0.799
LDF	86.0	0.754
SBT	85.6	0.745
BDT	*81.4*	0
System "Recognition" – *Task 2*		
DTA	*61.5*	0.864
SVM	71.8	0.842
SWS	*58.2*	0.780
LoReg	68.1	0.776
ANN	67.1	0.766
KNN	70.0	0.751
LM	65.3	0.734
MNN	66.7	0.694
TLS	64.8	0.675
LDF	71.4	0.671
MP	66.7	0.666
SBT	*60.6*	0.657
ECA	70.4	0.653
BDT	71.8	0.251
System "Recognition" – *Task 3*		
SVM	77.2	0.845
TLS	75.0	0.822
ECA	81.1	0.816
LM	77.8	0.804
DTA	78.9	0.801
LoReg	77.2	0.799
SWS	73.3	0.788
MNN	73.3	0.772
ANN	75.0	0.767
SBT	78.3	0.737
MP	72.8	0.733
KNN	71.7	0.700
LDF	71.7	0.675
BDT	78.9	0.607
System "Recognition" – *Task 4*		
DTA	59.4	0.865
LM	62.9	0.857
ANN	71.3	0.850
SWS	*47.6*	0.847
LoReg	64.3	0.843
SBT	56.6	0.836
SVM	67.1	0.832
BDT	59.4	0.803
ECA	60.1	0.780
LDF	50.3	0.756
KNN	62.9	0.742
MP	*49.7*	0.725
MNN	*48.3*	0.684
Scikit-learn in Python [9] – *Task 1*		
GBC	93.3	0.959
BC	92.2	0.951
ETC	92.0	0.948
RFC	92.2	0.945
MP	92.0	0.935
NSVC1	93.3	0.930
ABC	91.6	0.927
NSVC3	89.6	0.911
LIR	89.8	0.907
R	89.6	0.905
KR	77.1	0.905
LSVC	89.2	0.902
GPC	90.7	0.900
BR	88.3	0.895
LR	89.0	0.895
OMP	88.7	0.886
KNN	89.8	0.880

Algorithm	CV accuracy, %	AUC
HR	82.5	0.850
PACH	83.3	0.834
PACS	83.3	0.834
SGDC	82.5	0.828
PLS	81.8	0.815
P	83.3	0.812
DTC	88.1	0.806
GNB	78.1	0.796
EN	81.4	0.5
LL	81.4	0.5
L	81.4	0.5
Scikit-learn in Python [9] – *Task 3*		
RFC	85.4	0.935
MP	89.0	0.935
GBC	85.4	0.931
NSVC3	85.4	0.925
HR	85.4	0.923
P	89.6	0.917
PACH	85.4	0.917
PACS	85.4	0.917
BC	86.6	0.916
LIR	86.6	0.916
NSVC1	84.1	0.916
R	87.8	0.913
KR	81.7	0.913
LR	87.8	0.913
ETC	85.4	0.912
BR	87.8	0.911
SGDC	86.0	0.910
PLS	86.0	0.905
LSVC	86.6	0.905
OMP	88.4	0.899
KNN	82.3	0.881
GPC	82.9	0.860
ABC	83.5	0.856
GNB	76.2	0.831
DTC	84.1	0.813
L	69.5	0.5
EN	69.5	0.5
LL	69.5	0.5

The Table 2 includes the results of efficiency of algorithms ensembles methods estimation. The following notations of algorithms were used (*"Recognition" package* [8]):

- AC – the algebraic corrector (quadratic merit functional; minimal mean deviation = 0);
- CS – the convex stabilizer (function type – Gaussian);
- WD – the Woods dynamic method (number of objects in vicinity = 10);
- CCA – the complex committee method–averaging;
- CCM – the complex committee method–majority voting;
- BM – the Bayes method;
- CAS – the clustering and selection (number of clusters = 3);
- LC – the logic corrector;
- GPC – the generalized polynomial corrector (minimal mean deviation = 0);
- DC – the domains of competence (number of the domains of competence =3);
- DT – the decision templates.

"Scikit-learn package for Python" [9]:

- VCS – ensemble.VotingClassifier (voting='soft');
- VCH – ensemble.VotingClassifier (voting='hard');

Table 2 The accuracy estimation of various collective methods

Algorithm	CV accuracy, %	AUC
System "Recognition" – *Task 1*		
CCA	91.8	0.920
LC	90.3	0.918
GPC	88.3	0.896
CCM	87.4	0.893
BM	86.8	0.885
DC	92.9	0.847
DT	92.0	0.796
AC	91.6	0.770
WD	82.3	0.719
System "Recognition" – *Task 2*		
CCA	81.2	0.906
GPC	80.8	0.904
LC	72.1	0.893
DC	79.5	0.874
CCM	75.5	0.864
BM	79.0	0.812
WD	62.0	0.742
DT	80.8	0.727
AC	55.0	0.711
System "Recognition" – *Task 3*		
CCA	87.2	0.906
GPC	87.2	0.904
LC	86.0	0.893
DC	87.2	0.874
CCM	87.2	0.864
BM	86.6	0.812
WD	81.1	0.742
DT	85.4	0.727
AC	82.3	0.711
System "Recognition" – *Task 4*		
LC	50.7	0.847
BM	55.2	0.840
WD	55.2	0.827
CCA	61.2	0.815
CCM	63.4	0.787
DT	59.0	0.745
DC	60.4	0.651
GPC	59.7	0.646
AC	52.2	0.646
Scikit-learn in Python [9] – *Task 1*		
VCS	94.4	0.889
VCH	93.7	0.867
Scikit-learn in Python [9] – *Task 3*		
VCS	87.2	0.852

Algorithm	CV accuracy, %	AUC
VCH	86.6	0.836

The collective decision-making methods use algorithms for which AUC-values were marked by boldfaced types (see Table 1). We used «default option»-mode for choosing of algorithms parameter values.

It should be noted that in most cases the choice of the most accurate single ML methods using the cross-validation and the ROC-analysis coincides. The best algorithms (according to AUC-value) (see Table 1) are methods based on the support vector machine (SVM), the deadlock test (DTA), the artificial neural network learning (ANN), as well as the linear machine (LM), the statistical weighted syndromes (SWS), and the two-dimensional linear separators (TLS). The Gradient Boosting (GBC) crowds the top of the list in Scikit-learn package. The worst algorithms are the binary decision tree learning (BDT), the search for the best test (SBT), the linear Fisher discriminant (LDF), the Elastic Net (EN), and the Lasso (L and LL).

The most efficient algorithms ensembles (see Table 2) are the complex committee method–averaging (CCA), the logic corrector (LC), the generalized polynomial corrector (GPC), and the voting (VC). In most cases the algorithms ensembles application allows a prediction accuracy increase.

5 Conclusions

The problem of the most accurate algorithms selection belongs to the most important tasks of ML. To solve this task the subject field peculiarities must be taken into account. In this research the ML-software from «Recognition» and «Scikit-learn» packages were tested in inorganic compounds prediction tasks. As a rule, small sizes of training samples in these tasks do not allow a selection of representative objects subset for examinational recognition. In that context the cross-validation using training sample is the most acceptable procedure for ML algorithms accuracy estimation. The substantial difference in numbers of different classes of objects is a peculiarity of inorganic chemical tasks. Therefore, the ROC-analysis is the most acceptable method for these algorithms accuracy evaluation.

Acknowledgments. This work was partially supported by the Russian Foundation for Basic Research (project nos. 17-07-01362 and 18-07-00080) and State assignments No. 007-00129-18-00 and 0063-2020-0003.

References

[1] N.N. Kiselyova. Komp'yuternoe konstruirovanie neorganicheskikh soedinenii. Ispol'zovanie baz dannykh i metodov iskusstvennogo intellekta (Computer Design of Inorganic Compounds: Use of Databases and Artificial Intelligence Methods). Moscow: Nauka. 2005.

[2] N.Y. Chen, W.C. Lu, J. Yang, G.Z. Li, Support vector machine in chemistry. Singapore: World Scientific Publishing Co. Pte. Ltd. 2004.

[3] N.N. Kiselyova. Computer design of materials with artificial intelligence methods. In Intermetallic Compounds. Principles and Practice, Vol.3, Westbrook, J.H. & Fleischer, R.L. eds., p. 811-839, Chichester, UK: John Wiley&Sons, Ltd. 2002.

[4] T. Mueller, A.G. Kusne, R. Ramprasad. Machine Learning in Materials Science. Recent Progress and Emerging Applications. Reviews in Computational Chemistry, 29, p. 186–273, 2016.

[5] N.N. Kiselyova, A.V. Stolyarenko, V.A. Dudarev. Machine Learning Methods Application to Search for Regularities in Chemical Data. Selected Papers of the XIX International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2017). Moscow, Russia, October 9-13, 2017. CEUR Workshop Proceedings, v.2022, p. 375-380, 2017. http://ceur-ws.org/Vol-2022/paper57.pdf.

[6] N.N. Kiseleva. Prediction of the new compounds in the systems of halogenides of the univalent and bivalent metals. Russian Journal of Inorganic Chemistry, 59(5), p. 496–502, 2014.

[7] N.N. Kiselyova, A.V. Stolyarenko, V.V. Ryazanov, O.V. Sen'ko, A.A. Dokukin. Prediction of New Halo-Elpasolites. Russian Journal of Inorganic Chemistry. 61(5), p. 604-609, 2016.

[8] O.V. Senko. An Optimal Ensemble of Predictors in Convex Correcting Procedures. Pattern Recognition and Image Analysis. 19(3), p. 465-468, 2009.

[9] Yu. I. Zhuravlev, V. V. Ryazanov, and O. V. Sen'ko. RECOGNITION. Mathematical methods. Software system. Practical solutions. Moscow: Phasis. 2006.

[10] Pedregosa et al. Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.