UDC 004.855.5(045)
DOI:10.18372/1990-5548.81.18978

[1]V. M. Sineglazov,
[2]K. S. Lesohorskyi

SEMI-SUPERVISED MULTI-VIEW ENSEMBLE LEARNING WITH CONSENSUS

[1]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
[2]Faculty of Informatics and Computer Science, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060,
[2]lesogor.kirill@gmail.com ORCID 0000-0003-2773-7398

Abstract—*This paper is devoted to enchasing existing multi-view semi-supervised ensemble learning algorithms by introducing a cross-view consensus. A detailed overview of three state-of-the-art methods is given, with relevant steps of the training highlighted. A problem statement is formed to introduce both semi-supervised framework and consider the semi-supervised learning in the context of optimization problem. A novel multi-view semi-supervised ensemble learning algorithm called multi-view semi-supervised cross consensus (MSSXC) is introduced. The algorithm is tested against 5 synthetic datasets designed for semi-supervised learning challenges. The results indicate improvement in the average accuracy of up to 10% in comparison to existing methods, especially in low-volume, high density scenarios.*

Index Terms—Machine learning; semi-supervised learning; label propagation; multi-view training; ensemble.

I. INTRODUCTION

With the development of modern technologies and computing capabilities, machine learning methods are becoming more and more powerful and effective. One of the important directions in this field is the methods of training using multi-view learning, which is aimed at improving the accuracy of models by combining information from different models or data representations.

The multi-view boosting method is one of the promising approaches in this field. It is based on the combination of ideas of boosting and multi-view learning, resulting in more accurate and reliable models due to the use of different types of data. Boosting is known to be a powerful ensemble learning technique that combines weak classifiers to create a single strong classifier. The application of this technique in the context of multi-view learning allows to significantly increase the effectiveness of models due to the use of additional information from various sources.

In this paper, the main concepts and approaches of multi-view boosting are considered, existing methods and architectures analyzed, and an improvement to the approaches proposed.

The purpose of this work is to research and analyze the effectiveness of multi-view boosting in comparison with traditional approaches, as well as to determine the conditions under which this method gives the best results. The implementation of experiments and the analysis of the obtained results will highlights the advantages and limitations of multi-view boosting in various classification tasks.

This paper structured in the following way: in the literature review section we provide an overview of existing techniques of semi-supervised learning and provide an overview of the existing methods and brief descriptions of the algorithms used. In the problem statement section we provide a formal statement of the problem and outline the setting of semi-supervised learning that is considered in this work. In the method section a detailed description of proposed method is given. In the result sections details of the experiment setup and hyper parameters is outlined. In the discussion section, the interpretation of results is given and some nuances of the approach and experiment are highlighted. Lastly, in the conclusion section we highlight the pros and cons of the proposed approach and propose the direction of future research.

The results of this work are useful for researchers and practitioners in the field of machine learning, engaged in the development and implementation of new methods and algorithms for processing large volumes of data using multi-view approaches.

II. LITERATURE REVIEW

Semi-supervised learning combines a limited amount of labeled data with a large amount of unlabeled data to achieve better accuracy when compared to supervised learning [1]. Most common approaches to semi-supervised learning include self-learning [2], co-training [3], graph methods [4], and boosting-based methods [5].

Boosting is a general ensemble technique for improving the performance of classification algorithms by combining several weak classifiers to create a single strong one. In particular, algorithms based on boosting, such as Multi-SemiAdaBoost (MSAB) [6], Multi-View Semi-Supervised Boosting (MSSBoost) [7], and Multi-View Semi-Supervised Self-Adaptive Algorithm (MS3A) [8] have demonstrated high performance in multi-class classification tasks, including sentiment analysis in texts.

In article [8], a hybrid semi-supervised boosting algorithm for text sentiment analysis is introduced. This approach involves using a classifier predictions together with similarity information to assign labels to unlabeled examples. The proposed model demonstrates that effective use of unlabeled data can significantly improve classification performance.

One of the key advantages of multi-view boosting is the ability to combine information from different sources (views) to build a more accurate model [9]. This allows the use of different types of data, which can contain additional information that is not taken into account when using only one type of data. However, finding appropriate similarity functions and computing optimal parameters can be complex tasks that require significant computational resources and experimental validation.

Research shows that using semi-supervised boosting algorithms such as MS3A-Ensemble can significantly improve classification performance. For example, in sentiment analysis of tweets, where different similarity functions and hybrid models were used, high accuracy and stability of results were achieved even with a limited amount of labeled data.

In this section we will provide a brief overview of the methods outlined above as they are the basis for the improvements proposed in this work. Specifically, we review MSAB and MS3A-Ensemble as the state-of-the art algorithm.

We will start by introducing the MSAB algorithm. The MSAB algorithm is designed for semi-supervised multi-class classification. Its architecture is based on traditional boosting methods, but includes mechanisms for using unlabeled data. MSAB algorithm consists of the three steps:

1) Initialization. During this step all of the hyper parameters for the baseline learner are defined and each sample (both labeled and unlabeled) receives a weight, initially set to the same constant greater than 0.

2) Boosting Cycle. During this step, each weak classifier is trained on the current weights of the samples. During the training both labeled and unlabeled data is used. After the raining is complete, it's error rate is tested on labeled samples. Based on the error rate, the weights for labeled samples are updated. The weights for correctly labeled samples are decreased, while the weights for incorrect samples are increased.

3) Results from each weak classifier are combined based on the weights derived during the second step. During the inference, the result from each weak classifier weighted, and a consensus-based decision is formed.

4) This algorithm is great, as it has few hyper parameters and is easy to apply, however it's accuracy boost is weaker than multi-view methods.

5) As such, let's take a look at *a multi-view boosting method – MS3A-Ensemble*. This algorithm uses information from different views (representations) of data to improve classification accuracy. The algorithm has 3 primary steps:

6) Initialization. The weights of all training examples (marked and unmarked) are initialized with equal values. The number of views (representations) and the number of iterations of the algorithm (T) is set.

7) Boosting Cycle. Firstly, a weak classifier is trained for each view. At each iteration, a weak classifier is trained for each view of data (V_1, V_2, ..., V_n). Weak classifiers are trained using both labeled and unlabeled data. Errors of classifiers are estimated separately for each view. Sample weights are updated for each view independently, taking into account the errors of the corresponding classifiers.

8) View result combination. After completing the training iterations, the results of all weak classifiers for each view are combined based on the weights. A visual representation of the training method is presented in Fig. 1.

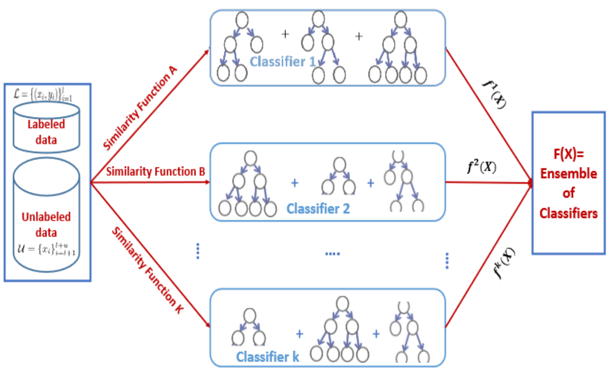


Fig. 1. MS3A-ensemble method

III. Problem Statement

In this paper the problem of semi-supervised learning is considered in the generall setting: given l labeled points $L=\{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled points $U=\{x_{l+1}, ..., x_{l+u}\}$; with l being much smaller than u ($l<<u$). Let $n = l + u$ be the total number of data points. In this paper, the testing will be performed with binary labels, or $y \in \{0, 1\}$.

The goal of semi-supervised learner f_{SSL} is to provide an improvement of the goal metric g over the supervised learner f_S. This can be considered an optimization problem:

$$\max_{f_{\mathrm{SSL}}} g\left(f_{\mathrm{SSL}}(x)\right) - g\left(f_S(x)\right). \quad (1)$$

As such, the aim of this research is to propose a learner f_{SSL} that outperforms supervised learning methods f_S, and provides an improvement over existing methods.

It is important to properly select a goal metric g, as suboptimal selection can lead to skewed results. Selecting model-level metrics such, as loss, is acceptable when comparing two models with the same architecture, however it is not useful when a different definition of loss or different models are used. As such, the common choice of goal metric is batch-level aggregated metrics like average accuracy, recall, and precision. In the setting that is explored in this paper, a two-class dataset is used, with each class being equal. As such, average accuracy is the best metric to optimize:

$$\mathrm{AA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}. \quad (2)$$

where TP are true positives; TN are true negatives; FP are false positives; FN are false negatives.

IV. Method

In the methods outlined in the literature review section, we have outlined two common methods – MS3A and MSBA. One of the aspects that these methods do not explore, however, is cross-view consensus during the training.

In the inference stage, the results from each window are aggregated to generate the final decision, however in the training stage each view (partition) of data is considered independently. Please note that view and partition are considered interchangeable terms in this paper. However, if we are able to introduce the cross-view interaction in the training process it will solve several problems, namely:

- view splitting can be done by using different classifiers or slices of data, no explicit definition of similarity function required;
- weak classifiers will be able to share domain knowledge, reinforcing the learning, improving the convergence and accuracy;
- we could leverage smoothness assumption more during the training process and provide a more rigid framework for the application of semi-supervised methods;
- semi-supervised learning can hurt the accuracy in comparison to supervised learning. This is especially true for multi-view training as each view can introduce bias.

As such, proposed method aims to address all of the issues by applying smoothness-based consensus over the partial results for each view, called "MSSXC" (multi-view semi-supervised cross consensus).

The proposed method is based on applying the smoothness assumption in the optimization problem setting (1) to improve the target goal in comparison to the baseline supervised classifier for each view. Then, a proxy labeling approach is used to select high confidence samples from best-performing classifiers..

The training process consists of three steps – initialization, semi-supervised cross-view weak classifier training, and aggregation weight derivation.

The initialization step is similar to MS3A algorithm – we initialize k weak classifiers and set the hyper parameters of our models. What is different, however, is the choice of view generation methods. Two strategies that can be used are k different classifier types trained on the same data in the democratic co-training scenario and splitting the dataset either into slices with all of the attributes or slices by attribute group. The choice of strategy should be based on the dataset used to train the learner.

After the view generation strategy is selected, the boosting loop begins. The goal of MSSXC is to provide sustainable learning performance that is better or equal to the supervised learning. To achieve this, before the first loop a fully supervised weak learner f_S is trained using all of the labeled data. Additionally, each semi-supervised learner f_{SSL_k} is trained using the labeled data L_k. This setup ensures that all of the views are properly initialized and the baseline for comparisons is set up.

Once the boosting preparation is complete, a training cycle begins. The first step is learner evaluation. To ensure that the training will not degrade target metric, only weak learners that outperform the baseline classifier are considered for the label propagation step. More formally, an evaluation derived from (1) is performed as:

$$s_i = g\left(f_{\mathrm{SSL}_i}\right) - g\left(f_S\right), \quad (3)$$

where $i \in [0, k]$ is a one of the training partitions (views); g is the goal metric based on (1); f_{SSL_i} is the learner for ith partition; f_S is the baseline supervised classifier.

After each weak semi-supervised learner is evaluated and compared, learners that are weaker than the baseline are removed from consideration (as their knowledge capacity is below the supervised learner and they are likely to degrade the result). This is done by filtering out all f_{SSL_i}, for which s_i is less than 0. This concludes the learner evaluation sub step.

The next step is dominant learner label propagation. The idea of this step is to use the strongest weak learner to propagate it's knowledge to other views. This is achieved by first adding all of the weak learners f_{SSL_i} to the priority queue Q_{ls} based on their metric score s_i If no learner is stronger than the base learner f_S, then it is selected as the dominant learner Dl. Otherwise, the learner with the highest goal score s_i is selected from the Q_{ls} and becomes a dominant learner Dl. After the dominant learner is chosen, it is used as a proxy-label for each partition of the data. More formally, this process is represented by formulae (4) and (5).

$$Dl = \max_{S_i} f_{\mathrm{SSL}_i},\ f_{\mathrm{SSL}_i} \in Q_{ls} \tag{4}$$

$$L_i = L_i \bigcup \left\{ \left(x_j, Dl\left(x_j\right) \right) \right\} \forall x_j \in U_i,\ Dl\left(x_j\right)_{\mathrm{conf}} \geq \varepsilon, \tag{5}$$

where L_i are labeled samples for partition I; U_i are unlabeled samples for partition i; x_j are unlabeled samples from partition U_i; $Dl(x_j)_{\mathrm{conf}}$ is the confidence of the dominant learner in the label for the input x_j; and ε is the minimal confidence threshold for labeling decision that is set up as a hyperparameter. After all of the partitions' labels are propagated, the each weak learner f_{SSL_i} is retrained using the updated dataset L_i. Evaluation step is performed again and Q_{ls} is repopulated with updated confidence scores. It is worth noting that the dominant classifier is evicted from Q_{ls} to prevent overfitting. The training loop is repeated either T times, or until g converges. L_i is resetored to it's original state after each iteration to prevent a mistake by a weak classifier from biasing the results of the training.

After the training is conducted, the final step is to construct an ensemble from the weak learners. This is achieved by applying the bagging technique. The evaluation of each weak learner against the metric g is performed, and voting coefficients are defined as:

$$v_{ic} = \frac{g_c\left(f_{\mathrm{SSL}_i}\right)}{\sum_{j \in k}^{j} g_c\left(f_{\mathrm{SSL}_i}\right)}, \tag{6}$$

where v_{ic} is the voting coefficient for the learner i for class c; g_c is the goal metric for class c.

Voting coefficient are saved together with model weights and are utilized in the inference time. After each weak learner creates a prediction vector for a given input, it is multiplied by the associated voting coefficient and summed together. This provides a probability vector, akin to applying softmax:

$$y'_c = \sum_{i \in k}^{i} v_{ic} \times f_{\mathrm{SSL}_{i_c}}(x), \tag{7}$$

where y'_c is the probability of class label belonging to class c; x is the given input; v_{ic} is the voting coefficient for the learner i for class c; $f_{\mathrm{SSL}_{i_c}}$ is the probability of x belonging to class c given by the ith weak learner.

The proposed method is flexible and has few hyper parameters to tune, which makes it ideal for low-dimensional and relatively simple datasets with a limited label count available.

V. RESULTS

To test the effectiveness of the algorithm, an experiment was conducted on five synthetic data sets with different percentages of labeled and unlabeled data.

A. *Datasets*

In this work, five datasets are used to evaluate the performance of the algorithm – three variations of the synthetic data set "Two Moons", the control data set "Circles" and the data set "Banana".

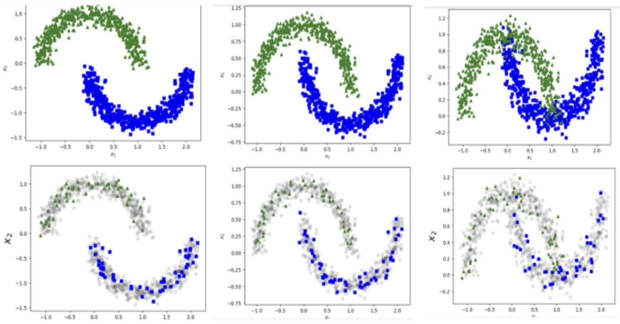
"Two Moons" is a common dataset that is used for evaluating the performance of semi-supervised learning. The main challenge with this data set is that naive label propagation algorithms will capture part of the other crescent depending on the distance between them. There are three data set options – wide, normal and narrow. Each option posses a certain challenge. Wide dataset is easy to label, so it serves as a "control", classic dataset have a minor overlap without any high-density regions, and tight variation introduces a high-density area, blurring the decision boundary.

A visualization of three variants of the "Two Moons" data set is shown in Fig. 2.

The "Banana" data set is more complex because it consists of two variants, one of them can be demarcated with a small distance between classes, the second has several intersections.

One of the classes is located in the middle of the other, but at the same time there is a small distance between them and, with the exception of a few anomalies, they do not intersect (Fig. 3a).

The data set “Circles” acts as a control data set for testing the correctness of the algorithm implementation (Fig. 3b). Supervised and semi-supervised learning should show high accuracy on this dataset.



a) b) c)

Fig. 2. “Two Moons” dataset – wide (a), classic (b), tight (c)

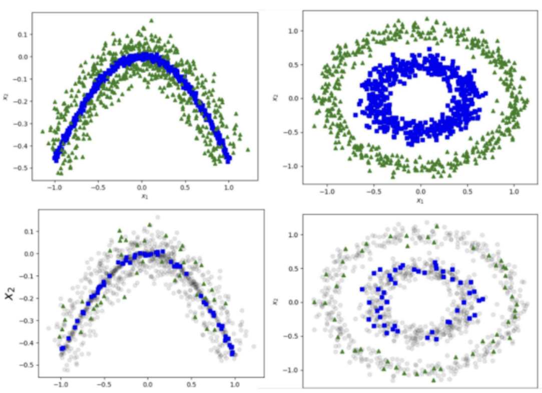


Fig. 3. Data sets “Banana” (a) and “Circles” (b)

While selected datasets are relatively simple, they still poses a certain degree of challenge, especially “Banana” and “Two Moons” tight. Due to the small distance between classes (or even data overlap), it is almost impossible to get a perfect accuracy, especially on the lower percentages of labeled data.

Synthetic datasets also ensure reproducibility of results, and their customization ability allows for a more flexible experiment setup.

B. Methodology

After the fully labeled dataset is generated, it is prepared for a semi-supervised learning. First, we train a baseline for the experiment, using 100% of labeled data. This serves as a “sanity check” of sorts, and enables the detection of issues early.

After the baseline classifier is trained, the data is split into labeled and unlabeled sets. In this experiment we test 1, 10 and 50 percent of labeled data. To prevent data skew, especially on lower labeled data count, we use stratified strategy to drop labels from classes equally.

Our method is learner-agnostic (it can use any learner type), as such we use support vector machine (SVM) [10] ensemble with radial bias function kernel for its simplicity.

In this setting a 5-view partitioning is used. We use democratic co-training splitting function, however as in this experiment only two classes are present, a bias is introduced by mixing two classes in the following proportions for each learner: 0–100, 20–80, 50–50, 80–20, 100–0. This ensures that each view have different representation of the groundtruth data.

We select average accuracy as presented in (2) as our goal metric for optimization g. We set $T = 20$ (training epochs), and classification confidence threshold $\varepsilon = 0.95$ for 10 and 50 percent labeled data scenarios and $\varepsilon = 0.9$ for 1 percent labeled data. The experiment is repeated to achieve a 95% confidence at 5% margin of error.

The results of the experiments are given in Table I.

TABLE I. ACCURACY OF CLASSIFIERS

Algorithm / Percentage of labeled data	**1%**	**10%**	**50%**
Two Moons Wide MSAB	97.2%	97.4%	97.4%
Two Moons Wide MS3A	100%	100%	100%
Two Moons Wide MSSXC	100%	100%	100%
Two Moons Wide Supervised	89.4%	98.5%	100%
Two Moons Classic MSAB	98.2%	98.9%	99.9%
Two Moons Classic MS3A	100%	100%	100%
Two Moons Classic MSSXC	100%	100%	100%
Two Moons Classic Supervised	97.2%	99.0%	100%
Two Moons Tight MSAB	79.6%	92.1%	95.3%
Two Moons Tight MS3A	81.2%	93.4%	94.9%
Two Moons Tight MSSXC	87.2%	94.1%	95.8%
Two Moons Tight Supervised	79.5%	80.3%	92.1%
Banana MSAB	49.3%	50.8%	54.2%
Banana MS3A	50.2%	55.9%	75.7%
Banana MSSXC	53.4%	65.8%	79.7%
Banana Supervised	50.1%	52.8%	60.1%
Circles MSAB	80.3%	100%	100%
Circles MS3A	96.3%	100%	100%
Circles MSSXC	97.0%	100%	100%
Circles Supervised	80.1%	100%	100%

VI. DISCUSSION

As can be seen from the experiment results in Table 1, the proposed algorithm has an advantage in low-volume scenarios with a data overlap. In other scenarios, it’s performance is comparable to MS3A, and always either meets or exceeds it. It is also worth noting that MSSXC never looses to the supervised learner, as by design it never selects a learner that would perform worse than a supervised learner.

Two datasets worthy of paying more detailed attention two are two moons tight and banana. In other scenarios, algorithms performed as expected and did not face any challenges.

In two moons tight there are slight areas of overlap, near the point where two crests connect to each other, which introduces a high-density area. In this scenario, on low data volume (1%), MSSXC shows better (by 6%) classification accuracy.

Same observation can be found when analyzing the results for banana dataset. MSSXC outperforms MS3A in all scenarios, with 1% and 10% of data being the most prominent. The biggest difference is at 10% of labeled data.

The explanation for this phenomena can be two-fold. Firstly, by design, the algorithm is better at identifying a decision boundry in high-density spaces as it uses the strongest learner and is also capable of sharing the domain knowledge cross-view by switching dominant learner on each iteration. Secondly, the experiment setup is favorable for this type of learning, as two dominant learners come from two most saturated views (100–0 and 0–100 partitions respectively). Re-labeling of unlabeled data on each iteration ensures that even if a mistake was made previously, a dominant learner has an opportunity to correct it, creating a balanced environment which provides better decision boundaries for each class.

VII. Conclusion

In this work, a novel algorithm for multi-view semi-supervised cross consistency learning was introduced. The algorithm expands on existing multi-view semi-supervised methods by introducing a consistency to classifiers by using highest confidence learner in label propagation stage and introducing a dominant learner rotation mechanism that prevents (or at least minimizes the impact of) over fitting. Additionally, the algorithm has mechanisms that prevent it's performance from being worse than just a supervised classifier ensemble.

Experimental results indicate that proposed algorithm works well in high-density regions with low label count, outperforming both supervised classifiers and existing algorithms. This is achieved by consistency mechanism providing better decision boundary placement in high-density regions.

The primary limitation of this research is that it was tested on fairly simplistic synthetic datasets. While such datasets are great for testing, evaluation and creating extreme scenarios, they might not represent all aspects of real-world challenges.

Future work involves testing the dataset on more real-world datasets and expanding learners to support high-dimensional data. This can be achieved by using more complex learners (e.g. neural networks), instead of support vector machines that were used in this paper.

References

[1] Y. C. A. P. Reddy, P. Viswanath, and B. Eswara Reddy, "Semi-supervised learning: A brief review," *Int. J. Eng. Technol*, 7.1.8, 2018: 81. https://doi.org/10.14419/ijet.v7i1.8.9977

[2] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki, "SSL-lanes: Self-supervised Learning for Motion Forecasting in Autonomous Driving," *Conference on Robot Learning. PMLR*, pp. 1–12, June 2022. arXiv:2206.14116v1 [cs.CV] 28 Jun 2022

[3] Islam Nassar, et al., "All labels are not created equal: Enhancing semi-supervision via label grouping and co-training," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. https://doi.org/10.1109/CVPR46437.2021.00716

[4] Yixin Liu, et al. "Graph self-supervised learning: A survey," *IEEE transactions on knowledge and data engineering*, 35.6, 2022, 5879–5900.

[5] Yuhao Chen, et al., "Boosting semi-supervised learning by exploiting all unlabeled data," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. https://doi.org/10.1109/CVPR52729.2023.00729

[6] Jafar Tanha, Maarten van Someren, and Hamideh Afsarmanesh, "An adaboost algorithm for multiclass semi-supervised learning," *2012 IEEE 12th International Conference on Data Mining. IEEE*, 2012. https://doi.org/10.1109/ICDM.2012.119

[7] Ion Muslea, Steven Minton, and Craig A. Knoblock, "Active+semi-supervised learning=robust multi-view learning," *ICML*, vol. 2, 2002.

[8] Dan Shi, et al., "Flexible multiview spectral clustering with self-adaptation," *IEEE Transactions on Cybernetics*, 53.4, 2021, 2586–2599. https://doi.org/10.1109/TCYB.2021.3131749

[9] Jing Zhao, et al., "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, 38, 2017, 43–54. https://doi.org/10.1016/j.inffus.2017.02.007

[10] Hearst, Marti A., et al., "Support vector machines," *IEEE Intelligent Systems and their applications*, 13.4 1998, 18–28. https://doi.org/10.1016/j.inffus.2017.02.007

Received August 12, 2024

Sineglazov Victor. ORCID 0000-0002-3297-9060. Doctor of Engineering Science. Professor.
Head of the Department of Aviation Computer-Integrated Complexes.
Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant, artificial intelligence.
Publications: more than 700 papers.
E-mail: svm@nau.edu.ua

Lesohorskyi Kyrylo. ORCID 0000-0002-3297-9060. PhD Student.
Department of Information Systems, Faculty of Informatics and Computer Science, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine, (2022).
Research interests: artificial neural networks, artificial intelligence, distributed computing.
Publications: 8.
E-mail: lesogor.kirill@gmail.com

В. М. Синєглазов, К. С. Лесогорський. Напівкероване багатовидове навчання з ансамблями на основі консенсусу

Статтю присвячено вдосконаленню існуючих алгоритмів напівкерованого ансамблевого багатовидового навчання шляхом введення консенсусу між видами. Подано детальний огляд трьох найсучасніших методів із виділенням відповідних етапів навчання. Формується постановка задачі, щоб представити як напівкеровану структуру, так і розглянути напівкероване навчання в контексті проблеми оптимізації. Представлено новий багатовидовий напівкерований ансамблевий алгоритм навчання під назвою багатовидовий напівкерований перехресний консенсус (MSSXC). Алгоритм перевірено на п'яти синтетичних наборах даних, призначених для напівкерованого навчання. Результати вказують на підвищення середньої точності до 10% порівняно з існуючими методами, особливо в сценаріях з малим обсягом і високою щільністю.

Ключові слова: машинне навчання; напівкероване навчання; поширення мітки; багатовидове навчання; ансамблі.

Синєглазов Віктор Михайлович. ORCID 0000-0002-3297-9060. Доктор технічних наук. Професор.
Завідувач кафедри авіаційних комп'ютерно-інтегрованих комплексів. Факультет аеронавігації, електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки, штучний інтелект.
Кількість публікацій: більше 700 наукових робіт.
E-mail: svm@nau.edu.ua

Лесогорський Кирило Сергійович. ORCID 0000-0003-2773-7398. Аспірант.
Кафедра штучного інтелекту, Факультет інформатики та обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: Національний технічний університет України «Київський Політехнічний інститут імені Ігоря Сікорського», (2022).
Напрям наукової діяльності: штучні нейронні мережі, штучний інтелект, розподіленні обчислення.
Кількість публікацій: 8.
E-mail: lesogor.kirill@gmail.com