COMPUTER SCIENCES AND INFORMATION TECHNOLOGIES

UDC 004.855.5(045)
DOI:10.18372/1990-5548.77.18001

[1]V. M. Sineglazov,
[2]S. S. Yarovyy

SEMI-SUPERVISED LEARNING BASED ON GRAPH STOCHASTIC CO-TRAINING

[1]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
[2]Institute for Applied System Analysis, 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]yarovyy.s@gmail.com

***Abstract*—*This article is devoted to the development of a new approach in semi-supervised machine learning. The goal of this article is to analyze the accuracy of the single-view co-training system, based on the use of a modified graph-based stochastic label propagation algorithm for a multiclass classification problem. Graph transformation of data is preceded by feature decomposition, with three algorithms being compared: Singular Value Decomposition, Truncated Singular Value Decomposition, Iterative Primary Component Analysis, Kernel Primary Component Analysis. To improve the accuracy of the proposed method, additional parameter was included in the label propagation algorithm, allowing for the usage of the algorithm in co-training systems. Further performance increases are achieved via optimization of data modification, which is achieved by applying feature decomposition methods and parallelizing the calculation-heavy processes. As examples of practical use were considered solutions to the problem of multiclass classification for standard datasets of the library sklearn and for the real dataset Traffic Signs Preprocessed. Analyses of the results of the implementation of the proposed approach showed improvements in accuracy and of performance solving the multiclass classification problem.*

Index Terms—Multiclass classification; semi-supervised learning; single-view co-training; stochastic label propagation.

I. INTRODUCTION

In the past decade, machine learning has become one of the main tools in the arsenal of information processing. Often, unsupervised learning algorithms are not sufficiently controllable to process complex information structures, and supervised learning requires significant costs to create training data.

Semi-supervised learning algorithms, in particular co-training, can achieve a level of accuracy and controllability that is only marginally inferior to the supervised learning models, at a much lower cost of training the system, similar to that of an unsupervised one. This is achieved by clustering a large amount of unlabeled or fuzzily labeled data based on their similarity to a small number of labeled samples, which minimizes the need for skilled agents.

Key feature that allows usage of semi-supervised learning algorithm lays within three assumptions about the data structure – Smoothness (Data points that are close to each other are more likely to have the same label), Cluster(samples form clusters with a higher probability of having common labels) and Manifold (data can be roughly projected onto a manifold of much lower dimensionality).

In this work we primarily employ two assumptions – Manifold, to justify usage of the feature decomposition operation on the data set, simplifying and speeding up the subsequent steps of the algorithm, as it resolves the so-called curse of dimensionality [1], a phenomenon first described in a work which considers the exponential growth of the required experimental data in accordance with the dimensionality of space when solving classification and comparable problems, and Smoothness to conform to the sufficient and necessary condition for semi-supervised learning on ε-good graphs [4], which is assured by the inclusion of Laplaccian transformation of the predicted label vector into the target function.

The traditional approach to supervised learning requires information structured so that it can be split into two views to train two classifiers independently. By using newly developed approaches to splitting monolithic data into pseudo-views, it is possible to

train semi-supervised learning systems on unstructured information much more efficiently.

Major challenge of semi-supervised learning that was rarely explored in previous years is the issue of polyquadratic operational complexity, that is $O(v \times d \times n^2)$ time costs of solving a linear system, with v being the amount of views and d - amount of neighbors for each point of the graph, inherent in the nature of graph-based label propagation, which caused it to be less efficient, especially combined with $O(n^2)$ costs connected to the construction of the graph matrix.

II. Problem Statement

This paper examines the problem of co-training, a subtype of machine learning in which a group of classifiers uses a common set of labeled and unlabeled data for supervised learning.

We represent the dataset as $L \bigcup U$ where $L = \left((x_1^1, \ldots, x_1^m), y_1\right), \ldots, \left((x_l^1, \ldots, x_l^m), y_l\right) \subset X \times Y$ denotes the labeled samples subspace, $U = \left(x_{l+1}^1, \ldots, x_{l+1}^m\right), \ldots, \left(x_u^1, \ldots, x_u^m\right) \subset X$ – unlabeled samples subspace, m – number of data features, l – amount of labeled samples, u – total samples count, $l \ll u$, $x_i^k \left(i \in [1, u], k \in [1, m]\right)$ – kth feature of ith element, $y_i \left(i \in [1, l]\right)$ – ith element's label, the labels assume {0, 1, ..., n} for multiclass problems (n – class number) or {–1, +1} for binary problems, X – example space, Y – label space.

Traditionally, co-training problems only consider example spaces, which can be divided into two different views X^1, X^2, where $X = X^1 \times X^2$. Initial requirements, laid out in the cornerstone work by Bloom and Mitchell [2] assume that each of the presented ones is sufficient for the correct classification. We assume that all examples with nonzero probability labels correspond to objective functions $f_1 \in C_1, f_2 \in C_2$, thus, for every element $x = (x_1, x_2)$ with a label y, $f(x) = f_1(x_1) = f_2(x_2) = y$, where f denotes the shared target function over the entire example space, $f_a(x^a)$ – target function of the ath classifier for the corresponding representation.

For the graph representation of co-training, we represent the example space X via a graph $G = (V, E)$, where $V(G)$ denotes the vertices set, $E(G)$ – edges set, combined with affinity matrix W to store the weights which reflect the similarities between examples. Each cell W_{ij} represents the similarity between the examples x_i and x_j, defined via the weight function, usually chosen as $W_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right)$, with σ as the hyper-parameter [3].

To match the graph representation, we represent the n class classification problem as a set of n binary one-versus-rest classification problems, which is more time- and computationally-efficient than the one-versus-one representation. For each of the binary problems "one against the rest" we apply a modified stochastic label propagation algorithm with two views.

The goal is to create a set of labeling predictions that are both consistent with the original labeling and close to the real value function $f(x) = y$. The former is achieved by the fact that the resulting set must comply with the assumption of smoothness, mathematically represented as

$$\sum_{i,j=1}^{n} W_{ij} \left(f_i - f_j\right)^2 = \varphi^{\mathrm{T}} \left(D - W\right) \varphi = \varphi^{\mathrm{T}} L \varphi,$$

where $L \in \mathbb{R}^{n \times n}$ denotes the laplassian matrix, $L = \mathrm{diag}\left(W_{ij}, i = \overline{1, n}\right) - W$, $\varphi_l = \left(f_1, \ldots, f_l\right)$ – set of predicted values for labeled examples, $\varphi = \left[\varphi_l, \varphi_u\right]$ – vector of predicted values for the entire data set, $W \in \mathbb{R}^{n \times n}$ – adjacency matrix, where each value W_{ij} denotes the weight of the edge between nodes i and j. Of the metrics for predicted value congruence, we use Mean Squared Error weight function for the training dataset: $\|f_l - y_l\|^2$, as well as Accuracy, Precision & Recall for the test dataset, to evaluate the optimal performance of different classifiers on pseudo-labelled data. Both test data metrics can be aggregated into a single quadratic cost minimisation function

$$C(\varphi) = \varphi^{\mathrm{T}} L \varphi + \mu \|\varphi_l, \ldots, y_l\|^2 + \varepsilon \|\varphi\|^2. \qquad [4]$$

This algorithm meets the necessary and mandatory conditions for co-training on ε-good graphs.

III. Existing Works

In the course of this work we attempt to tackle several key issues impeding the efficiency of the co-training algorithms. Some of them are generic, characteristic for the whole field of machine learning, while others are specific to the semi-supervised learning systems.

The key work in the field was written in 1998 by Blum and Mitchell [2], first forming the framework of co-training, formally formulating it as a PAC-

style learning. It has proven that given a conditional independence assumption on the distribution *D*, if the target class is learnable from random classification noise in the standard PAC model then any initial weak predictor can be boosted to arbitrarily high accuracy using unlabeled examples only by co-training.

While the original work posed some rather strict requirements for the nature of data used, further studies allowed for significant relaxation of requirements, such as proving that the co-training with random feature set splitting outperforms classical supervised learning algorithm. [4], [5].

The key to sidestepping the curse of dimensionality lies within various methods of feature decomposition, such as pseudo-multiview decomposition [6], subspace feature splitting [7], singular value decomposition and primary component analysis.

Key element of most semi-supervised learning method is some variety of label propagation algorithm. The benefits of the original form of the algorithm include it not requiring prior knowledge of the dataset, the number of communities that the network should be partitioned into, and need not define functions determining when to stop iteration. [8] At the beginning, we assign a unique label, indicating their attributive community, to each node in the network, and then each node updates its label according to the label with maximum number in its neighbours. As the labels propagate, the tightly connected individual in the network can quickly reach a stable state with a unique label, and the nodes with the same label are considered to belong to the same community structure.

Further advancements mostly were made in solving community detection problems, integrating various additions into the basis, such as adding a multistep greedy agglomerative algorithm (MSG) [9], avoiding the tendency to fall into local optimum resulting in a modularity optimization and hierarchical label propagation algorithm LPAm+.

Other developments include discovering that after five iterations, ninety-five percent of the nodes can be correctly clustered by LPA [10]; creating semi synchronous [11]; community belonging degree-based [12] (unfortunately catastrophically less efficient than baseline, due to the $O(n^2)$ time complexity); NIBLPA, that either considers both the k-shell value and the degree of node itself as well as its neighbors' k-shell values to calculate the node improtance of every node [13] or, alternatively, using Bayesian network [14], downside being uncertainty to the community detection results due to the use of additional parameters that require adjustment;

Another approach is an integration of label propagation with modularity and node importance (LPA-MNI) in order to solve the instability problem of LPA. [15] The algorithm first assigns different communities to each node, joining the communities which provide maximal gain in Newman-Girvan modularity function, which means that when the number of more than one label reaches maximum, the importance of neighbour nodes is calculated, and the most important node's label are assigned to the current node. The time complexity of initializing each node as an independent community is equal to $O(n)$. For the process of discovering rough communities, the complexity is denoted by $O(n \times k)$, in which k represents the average degree of the network. The time complexity of computing importance of all nodes is $O(l)$, and that of the process of ranking nodes according to degree centrality can be expressed as $O(n \log n)$. In the worst case, the time complexity of updating the labels for the remaining nodes is $O(n \times k)$. Consequently, the time complexity of the proposed algorithm is $O(n) + 2 \times O(n \times k) + O(l) + O(n \log n) \approx O(l + n \log n)$.

Other improved LPA algorithms, such as COPRA [16] and SLPA [17], have also been put forward for community detection in complex network.

Other important field of research considered in this work is the graph-based representation of semi-supervised learning [18] [3], with a work dedicated to research of a stochastic label propagation algorithm for self-learning [19] being especially important in the development of this work.

All those algorithms have key issues – namely they are designed primarily for community detections, thus struggling with classification tasks. Additionally, two step algorithms are very time and memory inefficient due to either compound logarithmic or quadratic costs, which leads to their impracticability on large datasets.

IV. Proposed Method

Among the works that were considered when creating this one, the most optimal for the problem under consideration is the study of a stochastic label propagation algorithm [19].

1) At the initial stage, the vector φ_l – the set of predicted values for labelled examples, is set equal to the values of the corresponding labels (–1 or 1), while the vector φ_u – the set of predicted values for unlabelled examples, is set equal to 0.

2) Next, the neighbourhood for each point is determined, by using either the k-nearest points method, or an edge weight limit [19].

3) Before initiating the main body of the algorithm, a preliminary propagation of labels by one step is performed – the so-called "warm start" procedure [19]. Updated pseudo-labels are calculated as follows:

$$f_{\text{neib}} = f_{\text{neib}} - \alpha_0 \left(f_{\text{neib}} - f_{\text{marked}}\right) W_{(\text{marked, neib})} = f_{\text{marked}} \alpha_0 W_{(\text{marked, neib})},$$

where α_0 is the algorithm's initial step size; $W_{i,j}$ is the weight of the edge between two elements; f_i is the element of the predicted values vector for the node i.

This operation is performed for the neighbourhood of each marked element, thereby improving the pseudo-label propagation rate at early epochs of the algorithm.

4) At each epoch t, a random element r is randomly selected. After that, we pass through all elements in the sequence [r,..., u, 1, ..., r – 1] and perform the following operation for all elements j in the neighbourhood of the current element i:

$$f_j = f_j - \alpha_t \left(f_j - f_i\right) W_{ij}.$$

5) After passing through all the elements of the neighbourhood of the element i, the value of the element i itself is recalculated as

$$f_i = f_i - \alpha_t \left(\mu\left(f_i - y_i\right) + \varepsilon f_1\right),$$

where μ and ε are hyperparameters, if i lies in the labeled elements set, or $f_i = f_i - \alpha_t \varepsilon f_i$, otherwise.

6) At the end of each epoch, a new value for the algorithm step size is set as

$$\alpha_t = \frac{\alpha(t-1)}{\sqrt{t}}.$$

This algorithm is single-stage, which gives it an advantage in epoch processing speed compared to the more traditional approach of using two-stage label propagation algorithms. The disadvantages of this algorithm are that this algorithm is adapted exclusively to self-learning systems, which significantly limits the accuracy of work and the task space, as well as the lack of a system for pre-processing and data graphing, which significantly limits the volume of data formats with which it can interact. In the version of the algorithm presented in this paper, the quadratic cost function is

$$C(\varphi) = \varphi^{\mathrm{T}} L \varphi + \mu \left\|\varphi_l - y_l\right\|^2 + \varepsilon \left\|\varphi\right\|^2,$$

where L is the Laplassian matrix of the adjacency table; φ_l is the set of predicted values for labelled examples; $\varphi = \left[\varphi_l, \varphi_u\right]$ is the vector of predicted values for the whole data set; μ, ε are hyperparameters; which is optimised iteratively using approximation

$$C(\varphi) = \sum_{i,j}^{n} W_{ij} \left(f_i - f_j\right)^2 + \mu \sum_{i=1}^{n} W_{ij} \left(f_i - y_i\right)^2 + \varepsilon \sum_{i=1}^{n} f_i^2,$$

which gives a significant performance advantage over matrix inversion or gradient computation.

It is proved that by using stochastic gradient descent it is possible to obtain a solution for the φ_u: $\varphi^{t+1} = \varphi^t - \eta \nabla C_i(\varphi)$, using the gradient

$$C_i(\varphi) = \left(\varphi - f_i\right) \circ W_i^{\mathrm{T}} + \left(\mathrm{I}_{[l]}(i) \mu \left(f_i - y_i\right) + \varepsilon f_i\right) e_i.$$

However, this gradient can still be improved by first caching the neighbouring vertex indices for all vertices in the graph and computing deg(i) of non-zero elements, which reduces the computational cost per iteration to $O(u \times \deg(i))$ – polylinear value, compared to the usual polyquadratic cost.

Despite the significant performance gains, the original work only considered use within self-training algorithms, which limits accuracy on high-dimensional data. Adapting the algorithm to co-learning principles should allow us to process more complex data structures with increased accuracy.

A. Co-training

Co-training methods require the following sequence of steps for data processing: The label propagation algorithm uses an initial set of labelled data to create pseudo labels, using the resulting dataset to train a pair of classifiers which, in some variations of the method, are used to improve the accuracy of label propagation. Due to the conceptual simplicity of this method, it can handle a large number of data formats, but has an important weakness – extremely low speed. By isolating the influence of classifiers on the data propagation process, and optimising performance through the use of feature decomposition and a modified stochastic label propagation algorithm, performance far superior to previously discussed implementations can be achieved.

B. Feature decomposition

To solve those problems, a method is proposed based on a one-stage algorithm for stochastic label propagation, adapted for co-training systems.

The method consists of a data pre-processing stage that carries out feature decomposition, reducing the dimension of the training data to improve the performance of further robotic steps and increase data clustering.

During this stage we compare the effectiveness of several feature decomposition algorithms – SVD, truncated SVD, iterative PCA & kernel PCA.

The next stage is splitting the data obtained as a result of decomposition into two views (there is room for further improvement - finding the optimal splitting method) and bringing each of the data views into a graph form using the method described in the second section. Next, we move on to the central algorithm of the method – co-training with stochastic label propagation:

C. Algorithm description

The necessary adaptation requires adding a third summand that represents the mutual influence of the jointly trained classifiers. We denote it as $\zeta\|\ddot{\varphi}\|^2$.

Thus, the modified algorithm minimises the cost function in the following form:

$$C(\varphi) = \varphi^{\mathrm{T}} L\varphi + \mu\|\varphi_l - y_l\|^2 + \varepsilon\|\varphi\|^2 + \zeta\|\ddot{\varphi}\|^2,$$

where $\ddot{\varphi}$ is the value of the predicted labels vector for the second data view.

This minimisation is applied to each of the n classes of the dataset. Thus n coupled instances of the algorithm are used, each computing the label propagation for two views. Before starting label propagation, we select a neighbourhood of k nearest neighbours (points with the highest edge weight) for each point. When working with binary classification, it was determined that the optimal $k = \{4, 8\}$.

The epoch of the algorithm proceeds as follows: At the beginning of each epoch, a new value of the algorithm step is set as $\alpha_t = \dfrac{\alpha(t-1)}{\sqrt{t}}$.

As with the original algorithm, the "warm start" procedure is used to optimize the work of the algorithm at the initial stages. In each epoch, a random element r is chosen at random (using the standard Python random number generator). After that, we go through all elements in the sequence [r, ..., n, 1, ..., r – 1] and perform the following operation for all elements j in the neighbourhood of the current element i for the first and second classifier respectively:

$$f_j^{(1)} = f_j^{(1)} - \alpha_t\left(f_j^{(1)} - f_i^{(1)}\right)W_{ij},$$
$$f_j^{(2)} = f_j^{(2)} - \alpha_t\left(f_j^{(2)} - f_i^{(2)}\right)W_{ij}.$$

After passing through all the elements around element i, the value of element i itself is modified as

$$f_i^{(1)} = f_j^{(1)} - \alpha_t\left(\mu\left(f_i^{(1)} - y_i\right) + \varepsilon f_i^{(1)} - \zeta f_i^{(2)}\right),$$
$$f_i^{(2)} = f_j^{(2)} - \alpha_t\left(\mu\left(f_i^{(2)} - y_i\right) + \varepsilon f_i^{(2)} - \zeta f_i^{(1)}\right),$$

if element i is labeled, or

$$f_i^{(1)} = f_j^{(1)} - \alpha_t\varepsilon f_i^{(1)},$$
$$f_i^{(2)} = f_j^{(2)} - \alpha_t\varepsilon f_i^{(2)},$$

otherwise.

After T epochs we obtain $2n$ vectors φ on which we train $2n$ binary classifiers (testing effectiveness of Perceptron, Gaussian Process Classifier, Logistic Regression, Gradient Boosting Classifier, Linear SVC).

On test data, the scores obtained for one class are summed up, the highest is selected, and returned as the predicted class for the test data element.

V. EXPRIMENTS

First step of the algorithm lies within decomposition of training data features and construction of proximity matrices based on generated pseudo-views. Due to the nature of transformations employed in feature decomposition algorithms, they are required to achieve desired rates of productivity, while preserving details required for further usage of label propagation, as stipulated by the manifold assumption. (Tables I and III)

This process is greatly improved time-wise with usage of parallel computations using python numba library.

TABLE I. USED DATASETS

Dataset	Number of attributes	Number of instances	Number of classes
digits	64	1797	10
wine	13	178	3
iris	4	150	3

The effect of parallelization of calculations on the binarization of labels, used randomly generated arrays of labels. The percentage of missing labels (simulation of a partially labeled data array) is 80%.(Table II, Fig. 1).

TABLE II. LABEL BINARIZATION

Labels	Parallelized	Classic
(1000, 10)	0.0012	0.0073
(10000, 10)	0.0046	0.1244
(100000, 10)	0.0308	0.8999
(1000000, 10)	0.3164	8.5341
(10000000, 10)	1.6954	47.4049

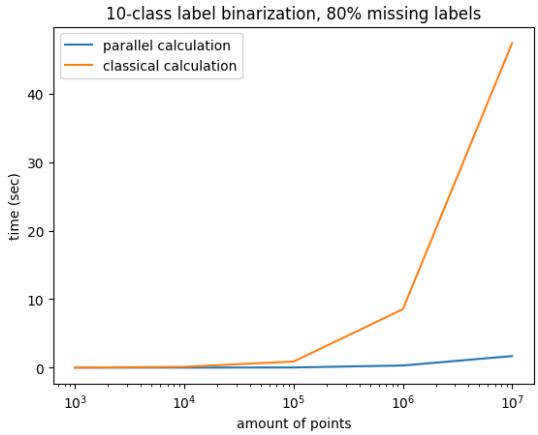

Fig. 1. Comparison of parallel and classical label binarization

TABLE III. FEATURE DECOMPOSITION COMPARISON

		SVD		
		prep	par	clc
1d	digits	0.3663	0.073	16.834
	wine	0.0018	0.003	0.4872
	iris	0.0001	0.161	0.2194
2d	digits	0.3916	0.0993	31.9704
	wine	0.3916	0.0016	0.3871
	iris	0.002	0.0011	0.4189
		t_SVD		
		prep	par	clc
1d	digits	0.027	0.0716	21.1054
	wine	0.003148	0.0013	0.3126
	iris	0.004305	0.001	0.2572
2d	digits	0.0479	0.0762	24.5238
	wine	0.0130	0.0014	0.3112
	iris	0.0044	0.0025	0.2778
		i_PCA		
		prep	par	clc
1d	digits	0.0649	0.0701	16.6874
	wine	0.0061	0.0012	0.3763
	iris	0.005	0.001	0.3225
2d	digits	0.1346	0.1026	27.7802
	wine	0.0059	0.0016	0.38
	iris	0.008	0.0013	0.2224
		k_PCA		
		prep	par	clc
1d	digits	0.3015	0.1824	35.5147
	wine	0.006	0.0014	0.3157
	iris	0.0213	0.0027	0.2753
2d	digits	0.3168	0.0882	18.0312
	wine	0.0067	0.0023	0.3932
	iris	0.0554	0.0035	0.5305

VI. CONCLUSIONS

It is shown that the existing methods of co-training in semi-supervised problems aren't satisfying the current demands of performance. A case proposed a new approach of SSL problems under use co-training, based on modified graph-based stochastic label propagation algorithm, paired with view data feature decomposition. The received results had shown an improvement within chosen metrics: Mean Square Error, Accuracy, Precision, Recall, Performance Speed.

REFERENCES

[1] R. E. Bellman, *Dynamic programming. Princeton*: Princeton University Press, 1957. p. ix ISBN 978-0-691-07951-6.

[2] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, July 1998, pp. 92–100, Madison, Wisconsin, United States, 24–26 July 1998, New York, New York, USA, https://doi.org/10.1145/279943.279962

[3] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, "Semi-supervised learning," *MIT Press*, 2006, pp. 193–205, ISBN:978-0-262-03358-9.

[4] J. Chan, I. Koprinska and J. Poon, "Co-training with a Single Natural Feature Set Applied to Email Classification," *In proceeding Conference on Web Intelligence*, Beijing, China, 2004.

[5] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training," *In Proceeding of the 9th, International Conference on Information and Knowledge Management*, McLean, Virginia, USA, 2000. https://doi.org/10.1145/354756.354805

[6] Minmin Chen & Kilian Weinberger, "Automatic Feature Decomposition for Single View Co-training," *Proceedings of the 28th International Conference on Machine Learning*, ICML 2011. 953–960.

[7] W. Zhang and Q. Zheng, "TSFS: A Novel Algorithm for Single View Co-training," *2009 International Joint Conference on Computational Sciences and Optimization*, Sanya, China, 2009, pp. 492–496, https://doi: 10.1109/CSO.2009.251.

[8] U. N. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys. Rev.*, E76, 036106, 2007. https://doi.org/10.1103/PhysRevE.76.036106

[9] X. Liu, T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Phys. A: Stat. Mech. and Appl.*, vol. 389, pp. 1493–1500, 2012. https://doi.org/10.1016/j.physa.2009.12.019

[10] J. Xie and B. K. Szymanski, "Community Detection Using a Neighborhood Strength Driven Label Propagation Algorithm," *In Proceedings of the 2011 IEEE Network Science Workshop*, IEEE Computer Society, West Point, NY, USA, 22–24 June 2011, pp.

188–195.
https://doi.org/10.1109/NSW.2011.6004645

[11] G. Cordasco and L. Gargano, “Community detection via semi-synchronous label propagation algorithms,” *In Proceedings of the IEEE International Workshop on Business Applications of Social Network Analysis*, Bangalore, India, 15 December 2011, pp. 1–8. https://doi.org/10.1109/BASNA.2010.5730298

[12] Chun Gui, Ruisheng Zhang, Zhili Zhao, Jiaxuan Wei, and Rongjing Hu, “LPA-CBD An Improved Label Propagation Algorithm Based on Community Belonging Degree for Community Detection,” *Int. J. Mod. Phys. C*, vol. 29, no. 02, 1850011, 2018. https://doi.org/10.1142/S0129183118500110

[13] Yan Xing, Fanrong Meng, Yong Zhou, Mu Zhu, Mengyu Shi, and Guibin Sun, "A Node Influence Based Label Propagation Algorithm for Community Detection in Networks", *The Scientific World Journal*, vol. 2014, Article ID 627581, 13 p., 2014. https://doi.org/10.1155/2014/627581

[14] X. K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, “Label propagation algorithm for community detection based on node importance and label influence,” *Phys. Lett. A*, vol. 381, Issue 33, pp. 2691–2698, 2017, https://doi.org/10.1016/j.physleta.2017.06.018

[15] Huan Li, Ruisheng Zhang, Zhili Zhao, and Xin Liu, “LPA-MNI: An Improved Label Propagation Algorithm Based on Modularity and Node Importance for Community Detection,” *Entropy*, 23(5), 497. https://doi.org/10.3390/e23050497.

[16] S. Gregory, “Finding overlapping communities in networks by label propagation,” *New J. Phys.*, vol. 12, pp. 2011–2024, 2010, https://doi.org/10.1088/1367-2630/12/10/103018

[17] J. Xie, B. K. Szymanski, and X. Liu, “SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process,” *In Proceedings of the IEEE International Conference on Data Mining Workshops*, Vancouver, BC, Canada, 11 December 2012, pp. 344–349. https://doi.org/10.1109/ICDMW.2011.154

[18] Z. Song, X. Yang, Z. Xu and I. King, "Graph-Based Semi-Supervised Learning: A Comprehensive Review," *in IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8174–8194, Nov. 2023, https://doi.org/10.1109/TNNLS.2022.3155478.

[19] De-Ming Liang & Yu-Feng Li, “Lightweight Label Propagation for Large-Scale Network Data,” *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Main track*, 2018, pp. 3421–3427. https://doi.org/10.24963/ijcai.2018/475

Received August 25, 2023

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

Yarovyi Serhii. Masters Student.
Department of Artificial Intelligence, Institute for Applied System Analysis, 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.
E-mail: yarovyy.s@gmail.com

В. М. Синєглазов, С. С. Яровий. Спільне навчання на основі стохастичного поширення міток на графі
Статтю присвячено розробленню нового підходу в машинному навчанні з частковим залученням учителя. Мета статті – аналіз точності системи спільного навчання з частковим залученням учителя, що ґрунтується на використанні модифікованого графового стохастичного алгоритму поширення міток для задачі багатокласової класифікації. Графовому перетворенню даних передує декомпозиція ознак, при цьому порівнюються чотири алгоритми: декомпозиція сингулярних значень, декомпозиція усічених сингулярних значень, ітеративний аналіз первинних компонент і ядерний аналіз первинних компонент. Для підвищення точності запропонованого методу в алгоритм поширення міток було включено додатковий параметр, що дає змогу використовувати алгоритм у системах спільного навчання. Подальше збільшення продуктивності досягається за рахунок оптимізації модифікації даних, що досягається застосуванням методів декомпозиції ознак і розпаралелюванням обчислювально-витратних процесів. Як приклади практичного використання було розглянуто розв’язання задачі багатокласової класифікації для стандартних наборів даних бібліотеки sklearn і для реального набору

даних Traffic Signs Preprocessed. Аналіз результатів реалізації запропонованого підходу показав підвищення точності та продуктивності під час розв'язання задачі багатокласової класифікації.
Ключові слова: машинне навчання; напівкероване навчання; стохастичне поширення міток; спільне навчання з одним уявленням.

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

Яровий Сергій Сергійович. Магістр.
Кафедра штучного інтелекту, Інститут прикладного системного аналізу, Національний технічний університет Україні «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: Національний технічний університет Україні «Київський Політехнічний Інститут імені Ігоря Сікорського», (2022).
Напрям наукової діяльності: штучні нейронні мережі, штучний інтелект, розподілені обчислення.
E-mail: yarovyy.s@gmail.com