UDC 004.855.5(045)
DOI:10.18372/1990-5548.76.17664

[1]V. M. Sineglazov,
[2]O. I. Chumachenko,
[3]K. S. Lesohorskyi

MODIFICATION OF SEMI-SUPERVISED ALGORITHM BASED ON GAUSSIAN RANDOM FIELDS AND HARMONIC FUNCTIONS

[1]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics
and Telecommunications, National Aviation University, Kyiv, Ukraine
[2,3]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]eliranvik@gmail.com ORCID 0000-0001-7803-9181,
[3]lesogor.kirill@gmail.com ORCID 0000-0003-2773-7398

Abstract—*In this paper we propose an improvement for a semi-supervised learning algorithm based on Gaussian random fields and harmonic functions. Semi-supervised learning based on Gaussian random fields and harmonic functions is a graph-based semi-supervised learning method that uses data point similarity to connect unlabeled data points with labeled data points, thus achieving label propagation. The proposed improvement concerns the way of determining similarity between two points by using a hybrid RBF-kNN kernel. This improvement makes the algorithm more resilient to noise and makes label propagation more locality-aware. The proposed improvement was tested on five synthetic datasets. Results indicate that there is no improvement for datasets with big margin between classes, however in datasets with low margin proposed approach with hybrid kernel outperforms existing algorithms with a simple kernel.*

Index Terms—Machine learning; semi-supervised learning; label propagation; Gaussian random fields; k nearest neighbors; harmonic functions.

I. INTRODUCTION

By definition, semi-supervised learning (SSL) is a branch of machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training to significantly improve training accuracy. It is between supervised and unsupervised. The concept of SSL avoids the cost of manually labeling the training data and makes use of the large amount of unlabeled data that is available in abundance.

Based on the principle of model learning, semi-supervised learning can be divided into inductive learning and transductive learning.

Inductive learning is learning on specific (training) examples in an attempt to generalize the condition for the entire input space. This means that in inductive learning, a generalized function is learned using the existing training data set. This generalized function may be logically true, but may or may not be realistically true for every data point in the sample space.

Transductive learning, on the other hand, generates rules based on specific training examples and then applies them to test examples. This approach is entirely domain-based and does not work for other input sample cases. Transductive learning does not solve a more general problem as an in-between space, but rather gets the specific answer that we really need. For example, forming a graph with connections between similar data points through which information is distributed. It does not require a training and testing step, and it does not need to train the classifier for the entire input space.

II. ANALYZING REQUIREMENTS FOR THE DATASET

The selection of data sampling criteria for semi-supervised learning depends on the specific problem and available data, but some general criteria for data sampling for semi-supervised learning can be identified:

- the availability of unlabeled data – one of the main requirements for semi-supervised learning is the availability of a large amount of unlabeled data, so the availability of unlabeled data is a key criterion for selecting data for semi-supervised learning, the more unlabeled data available, the more opportunities to improve model performance;
- data diversity – the data selected must be diverse enough to represent the distribution of all data; if the data selected is biased or unrepresentative of the entire data set, this may adversely affect model performance;

- label sparsity – the sampled data should have a high degree of label sparsity, i.e. only a small percentage of the data points should be labeled; this is because semi-supervised learning aims to use unlabeled data to improve the model's performance on labeled data;
- label quality – the quality of the labeled data is crucial for semi-supervised learning; if the labeled data is noisy or contains errors, it may cause the model to perform poorly, so it is very important to carefully select the quality of the labeled data;
- label distribution – the distribution of labeled data must be representative of the general data distribution; if the labeled data is biased or does not reflect the true distribution, the model may not generalize well to new, unobserved data;
- similarity – another data selection criterion for semi-supervised learning is the similarity between labeled and unlabeled data; if the labeled and unlabeled data are too different, it may be difficult for the model to extract useful information from the unlabeled data.

Hence, data selection criteria for semi-supervised learning include availability of unlabeled data, diversity of data, sparsity of labels, quality of labels, distribution of labels, and similarity between labeled and unlabeled data.

III. Label Propagation Method Based on Gaussian Random Fields

The problem statement of semi-supervised learning in general is: given l labeled points $L=\{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled points $U=\{x_{l+1}, ..., x_{l+u}\}$; usually $l<<u$. Let $n = l + u$ be the total number of data points. We will consider the example of a binary classification problem, so the labels will be binary: $y \in \{0,\ 1\}$.

In semi-supervised learning based on gaussian random fields and harmonic functions we consider a connected graph $G = (V, E)$ with vertices W corresponding to data points n, with vertices L[1, ..., 1] corresponding to labeled points with labels $y_1, \ldots, y_l$, and vertices $U[l+1, ..., l + u]$ corresponding to unlabeled points thus achieving label propagation as introduced in [1].

Our goal is to assign labels to the vertices of W. Let a symmetric weight matrix W of size $n\times n$ be given on the edges of the graph G. In the case when $x \in \mathbb{R}^m$, the weight matrix can be

$$W_{ij} = \exp\left(-\sum_{d=1}^{m}\frac{\left(x_{id}-x_{jd}\right)^2}{\sigma_d^2}\right), \tag{1}$$

where x_{id} is the dth component of instance x_i, represented as a vector $x_i \in \mathbb{R}^m$, and $\sigma_1, ..., \sigma_m$ are hyper-parameters of the length scale for each dimension [1]. Thus, neighboring points in the Euclidean space are assigned a large edge weight. Of course, other weights are possible, which may be more appropriate when x is a discrete or symbolic quantity. For our purposes, the matrix completely defines the structure of the data set.

Our strategy is to first compute a real function $f : V \to \mathbb{R}$ on G with some properties that satisfy our learning problem and then assign labels based on f. The function f is restricted to take the value $f(i) = f_l(i) \equiv y_i$ on labeled data for $i = 1, ..., l$. It is intuitive that unlabeled points that are close to each other on the graph should have similar labels. This motivates the choice of a quadratic energy function [1]:

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}\left(f(i) - f(j)\right)^2. \tag{2}$$

To specify probability distributions for functions f, a Gaussian field is formed $p_\beta(f) = e^{-\beta E(f)}/Z_\beta$ component, which is normalized on all functions bounded by f_1 on labeled data [2].

It is not difficult to show that the minimum energy function f = argmin $f_{|L=f1}E(f)$ is harmonious, namely, it satisfies $f = 0$ on unlabeled data points W and is equal to f_1 on labeled data points L. Here Δ is the combinatorial Laplacian given in matrix form as $\Delta=D - W$, and $D = \mathrm{diag}(d_i)$ is a diagonal matrix with elements $d_i = \sum_j W_{ij}$ and $W=[w_{ij}]$ is the weight matrix.

The harmonic property means that the value of f at each unlabeled data point is the average value of f at neighboring points:

$$f(j) = \frac{1}{d_j}\sum_{i \neq j} w_{ij} f(i) \tag{3}$$

for $j = l + 1, ..., l + u$ which is consistent with our previous notion of smoothness f relative to the graph. Calculated as $f = P_f$, and $P = D_{-1}W$ is a matrix of transitions on the graph. According to the principle of maximum HF [3], f is unique and is either a constant or satisfies the condition $0<f(j)<1$ for $j \in U$.

To calculate the harmonic solution explicitly in terms of matrix operations, the weight matrix W (and similarly D, P) is divided into 4 blocks after lth row and column [2]:

$$W = \begin{pmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{pmatrix} \tag{4}$$

Let $f = [f_l\ f_{in}]^{\mathrm{T}}$, and f_{in} denotes the value of the unlabeled data points, the harmonic solution $f = 0$ provided $f_{|L=fl}$ looks like

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l = (I - P_{uu})^{-1} P_{ul} f_l. \tag{5}$$

For the transition from f to labels, an obvious rule is to assign label 1 to vertex i if $f(i)>½$, and labels 0 in the opposite case [2]. In terms of the interpretation of the random walk, if $f(i)>½$, then, starting from i, a random walk is more likely to reach a positively labeled point than a negatively labeled one, as shown in [4].

data, is often poorly evaluated in practice and does not reflect the purpose of classification. Class priorities are a valuable source of additional information. Suppose that the desired proportions for classes 1 and 0 are q and $1 - q$, respectively, where these values are either given by an "oracle" or estimated from labeled data. In such a case, a simple procedure called class mass normalization (CMS) is used to bring the class distributions into line with the priority. The mass of class 1 is defined as $\sum_i f_u(i)$, and the mass of class 0 as $\sum_i (1 - f_u(i))$. Then these masses are scaled in such a way that the unlabeled point i belongs to class 1, if for f the following condition is met:

$$q\frac{f_u(i)}{\sum\limits_i f_u(i)} > (1-q)\frac{1-f_u(i)}{\sum\limits_i (1-f_u(i))}. \tag{6}$$

This method naturally extends to the general case with several labels. Optimization will be considered p_d both on labeled and unlabeled data.

A common parameter optimization criterion is the maximization of the likelihood of labeled data. However, the probability criterion is not suitable in this case, since the value f for labeled data is fixed during training, and furthermore the likelihood is meaningless for unlabeled data because there is no generative model. Instead, the average label entropy is used as a heuristic criterion for parameter optimization [5]. Average label entropy $H(f)$ fields f is defined as

$$H(f) = \frac{1}{u}\sum_{i=l+1}^{l+u} H_i\big(f(i)\big), \tag{7}$$

as well $H_i\big(f(i)\big) = -f(i)\log f(i) - \big(1 - f(i)\big) \cdot \log\big(1 - f(i)\big)$ is the entropy of the field at a single unlabeled data point i. A random walk interpretation is used here f, relying on the principle of the maximum of harmonic functions, which guarantees that $0 < f(i) < 1$ for $i \ge l+1$. Low entropy means that $f(i)$ is close to 0 or 1; it corresponds to intuition, which is good W (equivalently, a good set of hyperparameters $\{p_d\}$) should result in confident labeling. Of course, there are many arbitrary data labels that have low entropy, which may indicate that this criterion will not work. However, a limitation is imposed f on labeled data – most of these low-entropy arbitrary labels do not conform to this constraint. In fact, the space of low-entropy labels achieved by harmonic energy minimization is small and lends itself well to tuning the σ_d parameters.

However, there is a complication, which is that H has a minimum at 0 since $p_d \to 0$. As the length scale approaches zero, the tail of the weight function (1) becomes increasingly sensitive to distance. Ultimately, the label predicted for the unlabeled example dominates the label of its nearest neighbor, leading to the following equivalent labeling procedure:

1) starting from a labeled data set, find an unlabeled point x_u, which is closest to some labeled point x_l;

2) denote x_u with a label x_l, put x_{in} in the labeled set and repeat.

Since these are hard labels, the entropy is zero. This solution is desirable only when the classes are very well separated, and can be expected to be otherwise worse

This complication can be avoided by smoothing the transition matrix: P is replaced by a smoothed matrix $\tilde{P} = \varepsilon U + (1 - \varepsilon)P$, and W is a uniform matrix with elements $W_{ij} = 1/(l+u)$.

Gradient descent is used to find hyperparameters σ_d that minimizes H. The gradient is calculated as

$$\frac{\partial H}{\partial \sigma_d} = \frac{1}{u}\sum\nolimits_{i=l+1}^{l+u} \log\left(\frac{1 - f(i)}{f(i)}\right)\frac{\partial f(i)}{\partial \sigma_d}, \tag{8}$$

where the value of $\partial f(i)/\partial\sigma_d$ can be calculated from the vector $\partial f_u/\partial\sigma_d$, which is given by the formula:

$$\frac{\partial f_u}{\partial \sigma_d} = (I - \tilde{P}_{uu})^{-1}\left(\frac{\partial \tilde{P}_{uu}}{\partial \sigma_d} f_u + \frac{\partial \tilde{P}_{ul}}{\partial \sigma_d} f_l\right). \tag{9}$$

Both $\partial\tilde{P}_{he}/\partial\sigma_d$, and $\partial\tilde{P}_{ul}/\partial\sigma_d$ are submatrices of $\partial\tilde{P}/\partial\sigma_d = (1-\varepsilon)\partial P/\partial\sigma_d$. Since the original matrix of transitions P is obtained by normalizing the weight matrix W,

$$\frac{\partial p_{ij}}{\partial\sigma_d} = \frac{\dfrac{\partial w_{ij}}{\partial\sigma_d} - p_{ij}\sum_{n=1}^{l+u}\dfrac{\partial w_u}{\partial\sigma_d}}{\sum_{n=1}^{l+u} w_u}, \qquad (10)$$

finally

$$\frac{\partial w_{ij}}{\partial\sigma_d} = \frac{2w_{ij}(x_{di}-x_{dj})^2}{\sigma_d^3}. \qquad (11)$$

In the above equation, f_u is used directly as label probabilities, i.e. $p(\text{class}(x_i) = 1) = f_u(i)$. Since the previous information about the class is taken into account, it is necessary to minimize the entropy on the combined probabilities, therefore, the probability has the form:

$$\bar{f}(i) = \frac{q\left(u-\sum f_u\right)f_u(i)}{q\left(u-\sum f_u\right)f_u(i)+(1-q)\sum f_u\left(1-f_u(i)\right)}. \qquad (12)$$

IV. Modification of the Algorithm

Construction of a complete weighted connected graph based on the radial basis function (hereinafter RBF) kernel, is the main method in [2] and [5]. However, other graph construction algorithms are also proposed, for example, using k nearest neighbors (kNN), or tanh-weighted graph methods [5].

To modify the original algorithm, the approach of combining two graph construction methods, namely the method based on the RBF kernel and kNN, will be used. This approach leaves the same basis of constructing a graph based on the similarity of features of vertices, but allows to reduce its density and the mutual influence of distant points and strengthens the connection of neighboring points due to the reduction of the number of edges in the graph thanks to kNN.

However, since the weight matrix W must be non-negative and symmetric, it is necessary to modify the algorithm for constructing the kNN graph.

Let A be the adjacency matrix obtained using the kNN method, the elements of which are equal to 1 if the vertices of the graph are connected by an edge, and 0 otherwise. The following operation is used to convert the adjacency matrix into a symmetric one

$$\text{mutual\_}k\text{NN} = \max(A, A^{\mathrm{T}}). \qquad (13)$$

Having obtained a symmetric adjacency matrix, which represents the connections between the vertices of the graph, but does not reflect their similarity. After obtaining the modified adjacency matrix, it is combined with the RBF kernel to assign the edges of the weight graph to display the similarity of the vertices according to the formula:

$$W_{ij} = A_{ij}K_{\sigma}\left(x_i, x_j\right). \qquad (14)$$

In addition, the calculation of the initial value of the hyperparameter σ is introduced. When using the kNN method to build the weight matrix, the formula for calculating the initial value of σ using the formula proposed in [6] can be applied:

$$\sigma = \frac{\sum_{i=1}^{n}\psi(x_i, x_{ik})}{3n}, \text{...... ....}(15)$$

where x_I is the kth nearest neighbor of the point x_i, and n is the number of graph vertices.

The further algorithm remains without significant changes, only the use of the kNN method when optimizing the hyperparameter by gradient descent is adjusted.

V. Experiment Setup

To test the effectiveness of the algorithm, an experiment was conducted on 5 data sets with different percentages of labeled and unlabeled data.

A. Datasets

Five data sets are considered in the work – three variations of the synthetic data set "Two Moons", the control data set "Circles" and the data set "Banana".

Two echoes are a typical data set 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.

A visualization of three variants of the "Two Moons" data set is shown in Fig. 1.

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. 2a).

The data set "Circles" acts as a control data set for testing the correctness of the algorithm implementation (Fig. 2b). Supervised and semi-supervised learning should show high accuracy on this dataset.

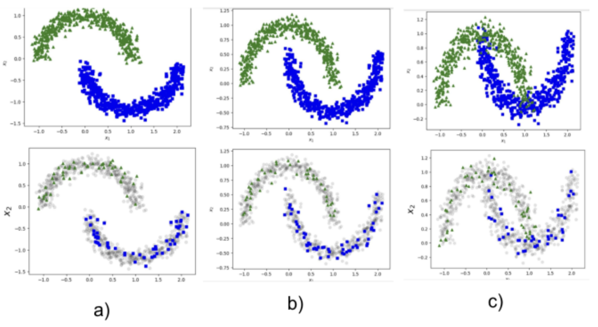

Fig. 1. “Two Moons” dataset – wide (a), classic (b), tight (c)

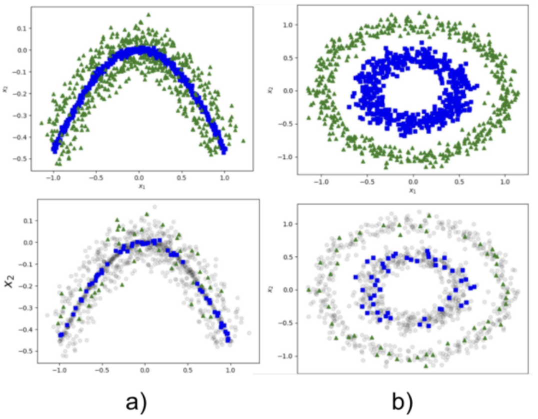

Fig. 2. Data sets “Banana” (a) and “Circles” (b)

B. Methodology

Data sets with 1, 2.5, 5, 7.5, 10, 15, and 50 percent labeled data are used to test the performance of the algorithm.

Given a similarity function based on the Euclidean distance between points, $\Psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. This similarity function is chosen, with its help you can build a graph with connections based on the similarity or similarity of the characteristics of the vertices.

The training array of labeled and unlabeled points X and the array of labels of a certain part of points L are also given.

With the help of these data, a connected weighted graph is constructed based on the RBF kernel, which depends on a given similarity function P_s, as well as from the hyper parameter p (1). The initial value of the hyper parameter in this case is set empirically. As a result, we get a graph of the presented weight matrix W, which is the edges of the graph and the array X, which are its vertices.

In parallel, the process of optimizing the hyper parameter σ using the gradient descent algorithm and formulae (8) – (12) takes place. The algorithm iteratively builds a new weighted graph, taking into account changes in the hyperparameter σ, by computing the entropy value of the Gaussian random field obtained from the main algorithm described below.

After calculating the weight matrix W based on the optimized hyperparameter σ, the calculation of the main elements of the method based on Gaussian random fields and harmonic functions begins. First, the diagonal matrix D of the ranks of the vertices of the constructed graph is calculated. Then, with its help, the matrix of transitions on the graph is calculated $P = D^{-1}W$.

As mentioned above, difficulties may arise when the classes are not very well separated, and in order to improve the performance of the algorithm, it is suggested to smooth the transition matrix using the formula $\tilde{P} = \varepsilon U + (1 - \varepsilon)P$, and W is a uniform matrix with elements $W_{ij} = 1/(n)$, n is the size of the training sample together with the unlabeled data.

At this stage, the main process of this algorithm takes place: spreading the probabilities of points belonging to a certain class due to the solution of the Laplace equation for the harmonic function and finding the harmonic solution.

Finally, the process of distribution of labels based on the obtained probabilities is carried out with the help of normalization of the mass of classes for a more accurate result.

After this process, we have a pseudo-labeled dataset that can be used to train supervised learning models and to test these models on a test dataset to evaluate the performance of the underlying algorithm. Intuitively, if the algorithm propagates pseudo-labels incorrectly, supervised learning models will perform worse on test samples after training on an invalid set with pseudo-labels.

In order to carry out a more qualitative study, the performance of the semi-supervised learning algorithm based on graphs and Gaussian fields will be tested with constant hyper parameters on various models of supervised learning, such as:

- Gaussian Process Classifier;
- *k*-nearest neighbors classifier (*K*-Neighbors Classifier).

The main metric for evaluating the quality of an algorithm is its accuracy on the test data set.

VI. Results

The results of the experiments are given in Table I, and the used hyper parameters are given in Table II.

As can be seen from the research results in Table 1, the original and modified algorithms do not differ much in terms of accuracy and show a very good result. The reason for this is the clear separation of the classes of the first two data sets, which allows even the unmodified algorithm to propagate pseudo-labels to unlabeled data without error. However, when examining the last data set, which contains a small cross-section of classes, the results of the methods worsened, but the modified algorithm showed a more accurate result compared to all supervised learning models.

TABLE I. ACCURACY OF CLASSIFIERS

Algorithm / Percentage of labeled data	**1%**	**10%**	**50%**
Two Moons Wide GP	97.2%	97.4%	97.4%
Two Moons Wide (GP Modified)	97.4%	97.4%	97.4%
Two Moons Wide (KN)	100%	100%	100%
Two Moons Wide (KN Modified)	100%	100%	100%
Two Moons Classic GP	99.9%	99.9%	99.9%
Two Moons Classic (GP Modified)	99.9%	99.9%	99.9%
Two Moons Classic (KN)	100%	100%	100%
Two Moons Classic (KN Modified)	100%	100%	100%
Two Moons Tight GP	78.6%	90.5%	90.9%
Two Moons Tight (GP Modified)	82.1%	92.4%	90.9%
Two Moons Tight (KN)	86.5%	92.9%	95.8%
Two Moons Tight (KN Modified)	88.0%	94.0%	95.6%
Banana GP	50.1%	49.7%	50.7%
Banana (GP Modified)	50.5%	49.9%	51.7%
Banana (KN)	50.5%	50.1%	70.0%
Banana (KN Modified)	50.8%	73.4%	94.5%
Circles GP	78.5%	100%	100%
Circles (GP Modified)	96.3%	100%	100%
Circles (KN)	74.1%	99.9%	100%
Circles (KN Modified)	91.5%	99.9%	100%

TABLE II. HYPERPARAMETERS OF CLASSIFIERS

Algorithm	σ	**k**
Two Moons Wide	[0.3, 0.3]	20
Two Moons Classic	[0.17, 0.13]	13
Two Moons Tight	[0.05, 0.05]	25
Banana	[0.08, 0.04]	15
Circles	[0.17, 0.13]	20

Let's consider the results of algorithm forecasts using the Gaussian Process model as an example, because it has the largest difference in forecast accuracy at 1% of labeled data. Figure 3a shows the forecast results of the model that used the data of the original algorithm for training, in the Fig. 3b – data of the modified algorithm.

It can be seen in the figures that the modified algorithm stopped the flow of one class to another and thus improved the result.

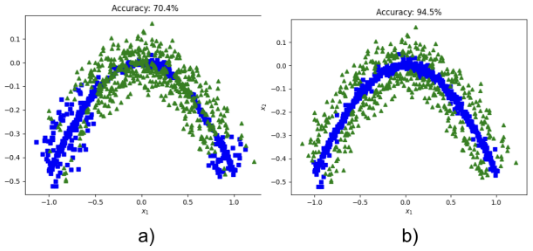

Fig. 3. Banana dataset propagated by (a) basic version of GP (b) modified version of GP

It can be assumed that the modified algorithm copes better with data containing class intersections due to reducing the density of the graph to weaken the influence of distant vertices on each other. This assumption will be tested on the following datasets.

VII. CONCLUSION

According to the results of experiments on synthetic data sets, the strengths and weaknesses of both the original algorithm and the developed modification were revealed.

The modification showed itself best when performing tasks on data that have clearly identified classes that may have intersections or noisy data that lead to the distribution of labels to the wrong areas by the original algorithm. The modification solves this problem by reducing the density of the graph and the number of connections between its vertices.

However, as noted in the algorithm performance review, the modification does not perform better on datasets with clearly separated classes and datasets with complex group structure.

REFERENCES

[1] Zhu Xiaojin, and Zoubin Ghahramani, "Learning from labeled and unlabeled data with label propagation," *ProQuest Number: INFORMATION TO ALL USERS*, 2002.

[2] Zhu, Xiaojin, Zoubin Ghahramani, and John D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," *In Proceedings of the 20th International conference on Machine learning* (ICML-03), 2003, pp. 912–919.

[3] Peter G. Doyle, and J. Laurie Snell, "Random walks and electric networks," vol. 22, *American Mathematical Soc.,* 1984. https://doi.org/10.5948/UPO9781614440222

[4] Wu, Xiao-Ming, Zhenguo Li, Anthony So, John Wright, and Shih-Fu Chang, "Learning with partially absorbing random walks," *Advances in neural information processing systems*, 25, 2012.

[5] Zhu, Xiaojin, John Lafferty, and Ronald Rosenfeld, "Semi-supervised learning with graphs (Ph. D. thesis)," Pittsburgh, PA, USA, 2005.

[6] Jebara, Tony, Jun Wang, and Shih-Fu Chang, "Graph construction and b-matching for semi-supervised learning," *In Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 441–448.

Received March 18, 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

Chumachenko Olena. ORCID 0000-0003-3006-7460. Doctor of Engineering Science. Professor. Head of the Department Artificial Intelligence, Faculty of Informatics and Computer Science, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Research area: artificial neural networks, artificial intelligence
Publications: more than 120 papers.
E-mail: eliranvik@gmail.com

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: 3.
E-mail: lesogor.kirill@gmail.com

В. М. Синєглазов, О. І. Чумаченко, К. С. Лесогорський. Модифікація алгоритму напівкерованого навчання на основі випадкових гаусівських полів та гармонічних функцій

У статті запропоновано вдосконалення алгоритму напівкерованого навчання, заснованого на гауссових випадкових полях і гармонічних функціях. Напівкероване навчання на основі гаусових випадкових полів і гармонійних функцій – це метод напівкерованого навчання на основі графів, який використовує подібність точок даних для з'єднання немаркованих точок даних із позначеними точками даних, таким чином досягаючи розповсюдження міток. Запропоноване вдосконалення стосується способу визначення подібності між двома точками за допомогою гібридного ядра RBF-kNN. Це вдосконалення робить алгоритм більш стійким до шуму та покращує розповсюдження міток з урахуванням локальності. Запропоноване вдосконалення перевірено на п'яти синтетичних наборах даних. Результати вказують на відсутність покращень для наборів даних із великим запасом між класами, однак у наборах даних із низьким запасом запропонований підхід із гібридним ядром перевершує існуючі алгоритми з простим ядром.

Ключові слова: машинне навчання; напівкероване навчання; поширення мітки; Гауссові випадкові поля; *k* найближчих сусідів; гармонічні функції.

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 670 наукових робіт.
E-mail: svm@nau.edu.ua

Чумаченко Олена Іллівна. ORCID 0000-0001-7803-9181. Доктор технічних наук. Професор. Завідувачка кафедри штучного інтелекту, Факультет інформатики та обчислювальної техніки, Національний технічний університет Україні «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.
Кількість публікацій: більше 120 наукових робіт.
Напрям наукової діяльності: штучні нейронні мережі, штучний інтелект.
E-mail: eliranvik@gmail.com

Лесогорський Кирило Сергійович. ORCID 0000-0003-2773-7398. Аспірант. 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, розподіленні обчислення.
Кількість публікацій: 3.
E-mail: lesogor.kirill@gmail.com