Modified Poisson label propagation graph-based method with regularization for task of classification intersection dataset

Michael Z. Zgurovsky[1,†], Victor Sineglazov[2,3,*,†] and Alexander Ruchkin[1,†]

[1]*National Technical University of Ukraine Igor Sikorsky Kyiv Polytechnic Institute, Peremohy ave., 37, Kyiv, 03056, Ukraine*
[2]*National Aviation University, Liubomyra Huzara ave., 1, Kyiv, 03058, Ukraine*
[3]*Institute of Cybernetics of the National Academy of Sciences of Ukraine, Pr. Akademika Glushkova, 40, Kyiv, 03187, Ukraine*

Abstract

In this paper we are devoted to the development of a new hybrid graph-based method of semi-supervised neural network learning for solving the classification problem. We analyze existing GSSL methods based on Laplace and Poisson Learning, identifies their advantages and disadvantages. We are proposed new generalizing error function for Poisson learning and some modifications. A new semi-supervised learning method is based on the generalized error function with an additional Tikhonov regularizer and ADAM optimizer. The proposed method allows obtaining better results compared to other existing GSSL methods on samples where there is an intersection of classes. Experimental verification of the proposed approach has shown an improvement in accuracy.

Keywords

Graph-Based Semi-Supervised Learning, Label propagation, Laplace equation, Poisson equation, Tikhonov regularization, ADAM method optimization 1

1. Introduction

Modern methods of Semi-Supervised Learning (SSL) have been successfully used to solve data classification problem problems in the presence of large amounts of unlabeled data and a small amount of labeled data. In existing state-of-the-art theoretical and applied problems, the process of data separation (the task of dividing data into classes) encourages the use of unlabeled data. There are many SSL methods that are used to solve such problems, for example: generative models, contrast models, cluster learning models, etc. [1]. When considering the classification task, data sampling plays an important role, since further tuning and results of the algorithm will depend on the definition of a set of its properties. Some semi-supervised learning methods use graph structures to represent the data set and distribute labels according to the conditions imposed by the graph. This category is called semi-supervised graph-based learning. The main features that distinguish it are scalability for huge datasets, and the use of a graph to find and represent label dependencies and use the information to predict values at unknown nodes. Semi-supervised graph-based learning is a powerful approach among other SSL learning models with significant advantages such as significantly improved performance on samples with very small amounts of unlabeled data and capturing complex relationships, making it more effective for solving the classification problem in class overlap [2]. Semi-supervised learning method based on the graph approach has practical applications in various elds of data analysis, including, for example, medicine, finance, meteorology, archeology, etc.

CIAW-2024: Computational Intelligence Application Workshop, October 10-12, 2024, Lviv, Ukraine

[*] Corresponding author.
[†] These authors contributed equally.

mzz@kpi.ua (Michael Z. Zgurovsky); svm@nau.edu.ua (V. Sineglazov); alex3005r@gmail.com (A. Ruchkin)
0000-0001-5896-7466 (Michael Z. Zgurovsky); 0000-0002-3297-9060 (V. Sineglazov)

The general method of Graph Based Semi-Supervised Learning (GSSL) can be represented as an algorithm and has the following steps [33].

1. Building a graph and initializing initial labels.
2. Building a penalty function. Adding regularization elements.
3. Finding the minimum of the penalty function. Label propagation algorithm.

Note that in this algorithm, the label values are calculated only for the vertices of the constructed graph, not for all the data, which significantly reduces the number of algorithm operations.

Currently, semi-supervised learning based on graphs underlies the construction of various Hybrid Neural Networks that are used in medical diagnostics [28, 29, 30].

The method of label propagation based on the Laplace equations was proposed [1, 2]. The method facilitates the propagation of known weights across the graph, allowing unlabeled nodes to be inferred based on their distance and connections to labeled counterparts. This approach not only increases the value of small amounts of available labeled data, but also effectively utilizes the huge reserves of unlabeled data. The development of this area is considered very promising and effective for solving many applied problems.

A modern and more efficient method of semi-supervised learning is Poisson Learning [9]. This method is also found in other works [11], [12]. From the GSSL point of view, Poisson learning is similar to Laplace learning, but the label propagation is performed using the Poisson equation rather than the Laplace equation. In the following, we will use MODIFIED Poisson learning. Let's consider the existing methods in the GSSL field.

2. Review of the current state of GSSL methods

2.1. Overview of existing methods of GSSL

Currently, there are many different approaches for implementing semi-learning, which can be viewed in the reviews [1, 2, 4, 5, 7, 9, 11]. The effectiveness of the graph structure approach lies in its ability to encapsulate complex relationships in data. By treating data as a structured set, one can take advantage of the hidden information embedded in the topological structure of the data.

Current research in GSSL is focused on the following methods: Label Propagation, Gaussian Random Fields, Harmonic Functions and etc. Among the works that study these methods are the following: [1], [2], [3], [4], [5], [6], [9] and others. Some of them provide a general taxonomy of GSSL methods.

One of the first studies in the eld of GSSL is the dissertation of [1]. This work provides a comprehensive analysis of semi-supervised graph-based learning using the Laplace equation and harmonic functions. Based on the results of the research, a more detailed review was later published by [2], which covers the work related to the application of this approach and semi-supervised learning using traditional classifiers. This work also includes an overview of the fundamental concepts of the methods: semi-supervised learning, generative models, collaborative learning, and semi-supervised learning based on graphs. The paper discusses the results of applying various elements of regularization. The paper also examines the use of these methods in practical tasks such as visual object recognition, word mapping for teenagers, and others. After that, several fundamental review papers on semi-supervised learning were published, in which the authors presented a classification and taxonomy of existing mathematical approaches in GSSL [1- 7]. In the review [3], the authors primarily focus on the separation of semi-supervised learning methods into inductive and transinductive methods. Inductive methods, which typically extend supervised algorithms to include unlabeled data, are further differentiated in the taxonomy based on how they apply unlabeled data: either in the preprocessing stage; directly in the objective function, or through a pseudo-labeling step. In all cases, transinductive methods are based on graphs and are grouped based on the choices made at different stages of the learning process. A seminal survey [4] focuses on the scalability of GSSL methods for large datasets, i.e., large graphs. In order to improve the scalability of semi-supervised graph-based methods, it is proposed to use a granularity mechanism. In work [5], the following semi-supervised learning methods are considered and compared: Laplace Label Propagation, Directed Regularization, Manifold Regularization, Deformed Graph Laplacian, Poisson Learning, Factorization Based Methods, Lazy Random Walk, and others methods. Work [6] focuses on an overview of deep learning models of neural networks, especially the use of generative models and their various types. We also note several interesting methods studied in the literature: Mutli-class MBO [34], Entered Kernel Method [35], Sparse Label Propagation [36], Weighted Nonlocal Laplacian (WNLL) [37].

Despite its effectiveness, there is no single universal method to determine a priori which learning method is best suited for any particular problem. Moreover, it is impossible to guarantee that the input of unlabeled data will not lead to performance degradation. Such degradation has been observed in practice, and its prevalence is likely underestimated due to publication bias [2]. The problem of performance degradation has been identified in other studies, [2], [38], [39].

This issue is especially relevant in examples where high performance can be achieved with purely supervised classifiers. In such cases, a significant decrease in performance is possible, outweighing the potential gain.

Several work [11], [16] have independently evaluated the performance of different semi-supervised learning methods on different datasets. [38] empirically com-pared eleven different semi-supervised learning algorithms using semi-supervised support vector and nearest neighbor methods, label propagation, and diversity regularization methods, applying hyperparameter optimization to each algorithm. By comparing the performance of the algorithms on eight different datasets, the authors show that no single algorithm uniformly outperforms the others. Some datasets showed significant performance improvements over the baseline, while others showed a decrease in performance. The relative performance also varied with the amount of unlabeled data. [39] compared several semi-supervised neural networks, including the average teacher model, virtual adversarial learning, and a wrapper method called the pseudo-label method, on two image classification problems. They reported significant performance improvements.

In recent years, the variational approach, which uses partial differential equations, in particular the Laplace equation, has become a popular semi-supervised learning method [1, 2, 9, 10]. Laplace learning means that there is a set of initial labels in the data set that propagate to all nodes using the harmonic function given by the Laplace equation. This method was used by [1, 2], [40], [41], [42], [43]. Some works used more complex differential equations.

For example, in [8], the adaptation of two partial differential equations, the p-Laplacian equation and the Eikonal equation, was considered. These equations are discrete analogs of the well-known partial differential equations widely used in image processing. The methods were illustrated for a semi-supervised classification task. The results showed that these methods work well with modern technologies and are applicable to semi-supervised classification.

In [9], a new framework called Poisson Learning is proposed and used for semi-supervised learning based on graphs with a very fast method speed. Pois-son learning solves the degeneracy problem of Laplacian semi-supervised learning. This method introduces a non-homogeneous function (the so-called "sources" and "receivers" based on the initial values of the labeled vertices), and solves the resulting Poisson equation on the graph. The results obtained are more stable and informative compared to Laplace learning. Poisson learning is efficient and easy to apply, and numerous experiments demonstrate its superiority over other state-of-the-art semi-supervised learning approaches on datasets such as MNIST, FashionMNIST, and Cifar-10. In addition, an advanced modification of Poisson learning, called Poisson MBO, is proposed, which provides higher accuracy and allows for prior knowledge of class sizes. Thus, differential equations, including the Laplace equation, the p-Laplace equation, the Eikonal equation, and the Poisson equation, play a crucial role in the variational approach to semi-supervised learning based on graphs. Poisson's equations are especially useful when the amount of labeled data is very small.

Regularization and multiple regularization. Semi-supervised methods based onlabel propagation using discrete partial differential equations are a powerful mathematical tool for solving classification problems. However, even when using these approaches, there are problems when they do not provide the required training results. Such cases include data noise, data sparsity, class overlap, etc. The use of regularization approaches is considered in [1-7, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26] and allows to partially solve these problems.

The issue of regularization is discussed in detail in [1], [5]. In fact, all classical GSSL methods can be simplified as a search for an error function on a graph that must satisfy two criteria simultaneously: it must be as close as possible to the given labels and it must be smooth over the entire graph. These two conditions can be further expressed in a general regularization framework, in which the loss function can be decomposed into two main parts. The first term is the controllable loss constraint according to the first criterion, and the second term is the graph regularization loss constraint according to the second criterion.

One of the little-studied semi-supervised learning methods is the multiple regularization method, which is successfully used to solve practical problems in which parameter tuning is a rather complex and non-trivial task [19-23].

For example, in [19] , the authors consider a problem in which signal and noise are additively combined. To solve it, they use the method of adaptive parameter selection for multiple regularization. With the proposed algorithm, the regularization parameters are selected based on the degree of sparsity of the data area. In [20], the multiple regularization method is used to nd the optimal neural network training rate. In [21], a new regularization scheme is considered to recover the solution of a linear uncorrected operator equation given noisy data in Hilbert space. Paper [22] discusses multiple penalty regularization, which has been successfully used to solve uncertain sparse regression problems of the non-mixing type. The advantage of methods with multiple penalties is that any prior information can be incorporated into additional penalties. For example, in [23], to solve the extrapolation problem, forecasting points are included as a priori information when constructing an extrapolation estimate. Other works in this area, such as [24 -26], should also be noted. However, more complex regularization methods with multiple penalties require a more thorough study.

2.2. Disadvantages of existing GSSL methods

Semi-supervised graph-based learning (GSSL) faces several key challenges at different stages. We list below some of the drawbacks and methods to compensate or eliminate them.

Graph construction stage. Building a suitable graph that represents the main relationships between data points is crucial. Various methods have been proposed, such as k-K-nearest neighbors, ϵ-nearest neighbors, Laplacian on a graph, and other approaches. These methods aim to identify local or global data structures and relationships. Some of them are listed in [5].

Label propagation stage. The main goal of GSSL is to propagate label values in the face of class overlap (in a dataset with a very small number of labeled nodes) [1]. However, the information on the basis of which the labels get their values may be incorrect, which in turn leads to incorrect label values, which worsens the classification results. To solve this problem, various algorithms have been developed: label diffusion, etc. [5]. These algorithms use the structure of the graph to generate labels, while taking into account the coincidence between data points.

Scaling phase. Scaling GSSL methods for data sets whose size is constantly changing may require more time, financial and human resources. Approximation methods and parallel computing are used to optimize these costs while scaling. Some approaches, such as graph clustering and graph shredding, aim to reduce the size and complexity of a graph without additional information.

The stage of processing a limited amount of labeled data: GSSL seeks to utilize both labeled and unlabeled data. However, when labeled data is not enough, methods such as self-learning, collaborative learning, and active learning can be used [5]. These methods iteratively select informative samples for labeling or use multiple representations of the data to improve the learning process.

Resistance to noise. Noise and outliers in the data can significantly affect the performance of GSSL methods. Robust convergence measures and outlier detection algorithms are used to mitigate the impact of noisy data points on classification results [11].

In general, the following conclusions can be drawn: first, semi-supervised learning methods are powerful enough to solve many classification problems with different initial conditions; second, the most productive semi-supervised learning methods are label propagation methods based on iterative solution of partial differential equations (Laplace and Poisson); third, regularization and multiple regularization methods are used for complex data cases.

In this paper, we will study neural network training for solving a classification problem based on the Poisson equation algorithm with L2 regularization. Next, we will formulate the problem statement taking into account the solution of the existing shortcomings of other methods.

3. Problem statement

We consider the problem of classifying dataset presented by a graph with a small number of labeled and big amount vertices are given at the beginning. The dataset can have difficult cases: torch or intersection classes data.

Let there be given an undirected weighted graph $G(X, V, W)$ with n vertices. X $= \{x_1, x_2, \ldots, x_n\}$ - the set of vertices of the graph, V- the set of edges, $W = \left(w_{ij}\right)_{i,\,j=1}^{n}$- is the weight matrix of the graph G. Let us assume $w_{ij} = 1$ if the vertices $x_i,\ x_j$ are similar, and $w_{ij} = 0$ if the vertices $x_i,\ x_j$ are different. The degree of the node x_i is determined by the formula $d_i = \sum_{j=1}^{n} w_{ij}$. According to Poisson learning [10], label propagation occurs by solving the Poisson equation, which has the form:

$$L(u_i) = \sum_{j=1}^{m} \left(y_j - \bar{y}\right)\delta_{ij}\,, \qquad 1 \le i \le m \tag{1}$$
$$L(u_i) = 0, \qquad m + 1 \le i \le n$$

where L - is the non-normalized Laplace operator, x_i - are the vertices of the undirected weighted graph, $y_i = y(x_i)$ - are the initial labels of the graph vertices, $u_i = u(x_i)$ - is the function of graph vertex labels G, w_{ij}- edge weight $\left(x_i, x_j\right)$, n - is the total number of graph vertices, the first m of which are considered labeled. $\bar{y} = \frac{1}{m}\sum_{j=1}^{m} y_j$. $\delta_{i,j}$- is the Kronecker symbol. $\sum_{j=1}^{n} d_i u(x_i) = 0$.

In [10], an iterative process for solving this system based on the solution of the diffusion equation was proposed - the Poisson Label Propagation method. We note some advantages of Poisson Label Propagation over the classical Laplace Label Propagation.

First of all, it means higher speed and shorter execution time due to the use of exact differential methods rather than stochastic methods (Random Walk, Diffuse, Broun Move).

Secondly, it is overcoming degeneracy at "zero" - that is, when the number of given labeled vertices is very small in relation to all vertices - or the ratio of labeled vertices to all vertices converges to zero.

Third, it is overcoming the problem of "forgetting" for a large amount of data. That is, in Laplace learning, with a small number of initial labeled vertices and a large number of unlabeled vertices, it is possible to re-label their initial values.

Fourth, there is still the problem of the speed of algorithm convergence for big data. There are many optimization algorithms to improve the solution convergence problem. One of the best optimization methods is the ADAM method, which will be used in the following. [29].

In addition to the general problems, the Poisson label propagation method has other drawbacks that arise in the case of class intersection. To solve the problem of semi-supervised graph-based learning for the case of class intersection, various regularization additives are used. [21], [23].

Thus, in this paper, we will consider a semi-supervised learning method using the munificent Poisson equation with regularization elements for more efficient application to samples with class crossover.

4. Modified Poisson Label Propagation with additional regularization

4.1. Rationale for applying the Poisson equation

Nowadays, in the field of semi-supervised learning, the number of examples of label propagation models based on the variational principle and using second-order partial differential equations, such as, for example, Laplace's equation, Poisson's equation, etc., is increasing. This fact demonstrates the relevance and prospects of this approach. One of the well-known approaches in this direction is to use the Laplace equation, which is a homogeneous second-order partial differential equation. It is also known as Laplace learning. The use of the Poisson equation in semi-supervised learning is relatively recent and limited to a few studies [10]. However, the use of the Poisson equation in semi-supervised learning has broader prospects for the future. Laplace learning and Poisson learning have a lot in common, but there is also a significant difference. Let's take a closer look at them.

It is known that the Laplace equation applies to various physical processes, such as heat distribution (heat conduction equation), electrostatic potential distribution, and others. However, Laplace's equation can be applied only in certain cases where there are homogeneous physical fields, such as a homogeneous electrostatic field or a homogeneous thermal field. If we need to consider a non-homogeneous problem and a non-homogeneous field, then a generalization of the Laplace equation to a non-homogeneous field, known as the Poisson equation, is used. From a mathematical point of view, the Poisson equation is a second-order inhomogeneous partial differential equation. Its left-hand side is the Laplace operator, and the right-hand side is the output function. The solution of a second-order nonhomogeneous partial differential equation can be found as the sum of the general solution of a homogeneous equation and the partial solution of a nonhomogeneous equation. The general solution is a solution to the Laplace equation. The particular solution of a nonhomogeneous equation depends on the initial term and must reflect the structure of the non-homogeneity in at least one particular case. Thus, the Poisson's equation generalizes the solution of the Laplace equation to inhomogeneous physical fields or, in our case, to inhomogeneous information fields.

In information and data theory, the existence of relationships between data means that the data is represented in a heterogeneous space when examining the data field and its distribution in the data space. It is also necessary to assume that the structure of the heterogeneity is correct and can be described by a similar iterative algorithm. Thus, using the Poisson equation to identify data classes and dependencies within a class in a heterogeneous data space may be more promising than using Laplace equations because the heterogeneity function is used in the solution construction. This is significant when we have a small number of initial data points (labeled data). With a small number of labeled data and a large number of total data, Poisson learning has a significant advantage over Laplace learning in terms of computation time.

Thus, the study and use of Poisson learning, in particular semi-supervised graph-based learning using the Poisson method, is a more promising area that requires further research for various application tasks.

4.2. Poisson Label Propagation

Let's consider the problem of semi-supervised learning using Poisson equations more thoroughly. Poisson learning was proposed in [10] and has the form (2)

$$\begin{gathered} L(u_i) = \sum_{j=1}^{n} w_{i,j}\left[u(x_i) - u(x_j)\right] = \sum_{j=1}^{m} (y_j - \bar{y})\delta_{ij}\,, 1 \le i \le m \\ L(u_i) = 0, \qquad m+1 \le i \le n \end{gathered} \tag{2}$$

where L - is the non-normalized Laplace operator, x_i - are the vertices of the undirected weighted graph, $y_i = y(x_i)$ - are the initial labels of the graph vertices, $u(x_i)$ - is a function of graph vertex labels, w_{ij}- edge weight (x_i, x_j), n - is the total number of vertices in the graph, the first m of which are considered labeled. $\bar{y} = \frac{1}{m}\sum_{j=1}^{m} y_j$. $\delta_{i,j}$- is the Kronecker symbol. $\sum_{j=1}^{n} d_i u(x_i) = 0$.

The weight of the graph edges can be calculated using the following formulas [9]:

1) $w_{ij} = \eta\left(\frac{|x_i - x_j|}{\varepsilon}\right)$ - geometric weight, (3)

2) $w_{ij} = \eta\left(\frac{|x_i - x_j|}{\varepsilon_k(x_i)}\right)$ - KNN weight, (4)

3) $w_{ij} = exp\left(\frac{-|x_i - x_j|^2}{2\sigma^2}\right)$ - Gaussian weight. (5)

Where X $= \{x_1, x_2, \dots, x_n\}$ - is the set of vertices in the graph, η - some function, ε - neighborhood, σ is a parameter that controls the variance of neighbors.

Poisson learning also has a variational interpretation, which reduces semi-supervised learning to minimizing the Dirichlet energy. Dirichlet energy is often used as a penalty function. The Dirichlet energy formula can be represented as

$$E(u) = \sum_{i,j=1}^{n} w_{ij}\left|u(x_i) - u(x_j)\right|^2 - \sum_{j=1}^{m}(y_j - \bar{y})u(x_j) \tag{6}$$

The iterative solution is as follows

$$u^{\{k+1\}}(x_i) = u^{\{k\}}(x_i) + \frac{1}{d_i}\left(\sum_{j=1}^{m}(y_j - \bar{y})\delta_{ij} - \sum_{j=1}^{n} w_{ij}\left|u(x_i) - u(x_j)\right|\right) \tag{7}$$

The label selection rule can be written as:

$$u_l(x_i) = argmax_{j\in 1,2} s_j u_j(x) \tag{8}$$

where. $s_j = \left(\frac{b_j}{\underline{y}}\right)$, b_j - is the share of data belonging to the class $j \in 1,2$.

These formulas can be called **Poisson label propagation.**

In the semi-supervised learning algorithm, to determine the value of the function $u$that has labels for the unlabeled data, while maintaining the correspondence to the labels of the labeled data, an iterative solution is required. In general, the following four steps can be distinguished.

1. Initialization. The algorithm starts with an initial assumption that can be random or based on observed data.

2. Optimization. The function u is iteratively updated to minimize the function $E(u)$which includes both smoothing u by similar data points and fitting the labeled data.

3. Convergence. The process continues until u converges, which means that further iterations do not significantly change the function u.

4. Prediction. After learning its functions, you can use it to predict methods for unlabeled data points.

In the next paragraph, we will **consider the step, i.e., the optimization method,** which will use the ADAM algorithm as one of the most efficient among analogs for this problem [29].

In the next subsection, we will consider the application of a regularization application to the L2 penalty function for the proposed neural network training method.

4.3. Application of regularization elements

L2 regularization will help prevent overlearning by adding a penalty term to the loss function that is proportional to the square of the weights. The regulatory term has the following form:

$$P(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2}\sum_{i}^{m} w_i^2 \tag{9}$$

The objective function using Tikhonov regularization will have a modified entry:

$$E(u) = \sum_{i,j=1}^{n} w_{ij}\left|u(x_i) - u(x_j)\right|^2 - \sum_{j=1}^{m}(y_j - \bar{y})u(x_j) + \frac{\alpha_1}{2}\|w\|^2 \tag{10}$$

A qualitative difference in the behavior of $L2$ regularization behavior is the sparsity of the solution obtained by Lasso regression, i.e., the optimal value of some parameters is zero.

Above, we have outlined ways to improve the proposed approach for solving the classification problem when the amount of labeled data is too small and the classes overlap.

Further, within the framework of the chosen GSSL approach, a new method will be proposed with its further improvement in the form of regularization terms and an optimizer.

4.4. ADAM optimization

Let's consider the optimization stage and apply the ADAM algorithm. The ADAM algorithm is often used to optimize the search for the minimum of error functions because of its stable performance. The structure of the algorithm is shown below.

ADAM (an extension of Adaptive Moment Estimation) is one of the best optimization algorithms commonly used in deep learning to tune model parameters during training. In practice, it has been found to perform well and often outperforms other optimization algorithms in many tasks. ADAM tracks the exponentially increasing average of previous gradients. This helps speed up convergence by taking previous gradients into account during the calculation.

For the hybrid ADAM algorithm, the optimization step looks like this

$$g_k(x_i) = \frac{1}{d_i}\left(\sum_{j=1}^{m}(y_j - \bar{y})\delta_{ij} - \sum_{j=1}^{n} w_{ij}\left(u(x_i) - u(x_j)\right)\right) + \frac{\alpha}{2}\|u_k(x_i)\| \tag{11}$$

$$\begin{aligned} m_k &= \beta_1 m_{k-1} + (1-\beta_1)g_k, \quad \overline{m}_k = \frac{m_k}{1-\beta_1^k} \\ v_k &= \beta_2 v_{k-1} + (1-\beta_2)g_k^2, \quad \bar{v}_k = \frac{v_k}{1-\beta_2^k} \end{aligned} \tag{12}$$

$$u_{k+1}(x_i) = u_k(x_i) - \frac{\beta}{\sqrt{(\bar{v}_k + \epsilon)}}(\overline{m}_k) \tag{13}$$

Where $\overline{m}_k$ - this is the moment of impulse (within the middle gradient).

g_k- is the gradient at any time step k.

β_1- is the rate of exponential tightening of the momentum of the member

ADAM rescales the gradient using an exponential damped average of the previous gradient squares.

v_k - is the velocity term (ковзне the mean square of the gradient).

β_2- is the rate of exponential decrease of the term velocity.

Because of the initialization m_k and v_k at the beginning (initialized with zeros), they can be shifted towards zero. To counteract this, ADAM includes a change correction step. Using the corrected momentum and velocity terms, the parameters are updated:

α is the learning rate. It determines the step size in the space parameters.

ϵ - is a small constant that prevents division by zero.

This method will be presented in the form of the following algorithm.

4.5. Algorithm modified Poisson Label Propagation Graph-Based Method Learning with and additional regularizer and optimizer ADAM

Let's consider a modified graph-based semi-supervised learning method based on Poisson learning with an additional regularizer and ADAM optimizer. So, this method will consist of the following steps.

1. **Building a weighted graph.**

To build a weighted graph, we usually use the formulas of geometric distance (weight), Gaussian distance (weight), KNN distance (weight) according to formulas (3)-(5). The use of the weight function depends on the data layout. For classes that have intersections, it is better to use KNN weights.

2. **Initialize the initial labels.**

Initialization of the initial labels is done by selecting some labeled data from each class. It is preferable to use samples with the same number of labeled data in each class.

3. **Building a penalty function. Adding regularization.**

When constructing the penalty function, the Dirichlet energy function on the graph is used with additional terms that characterize the regularization of the data or the boundary conditions of the Dirichlet function. L2, L1, etc. regularization can be used as regularization additives.

4. **Finding the minimum of the penalty function. Label propagation algorithm.**

If the full energy function is used as the energy function on the graph, then finding the minimum of the function is reduced to solving the Poisson equations on the graph. If the incomplete energy function is used, the algorithm is reduced to solving the Laplace equations. The numerical solution of these equations is the basis of the label propagation algorithm. In the proposed algorithm, we will use the ADAM method.

Thus, the algorithm can be rearranged as follows.

Step 1. Initially, we assume that an array of data and its labels is given.

1.1 Set m - the number of labels that will be selected from each class The balanced approach assumes that the number of labels from each class will be selected the same, the unbalanced approach assumes that the total number of labels for both classes will be selected randomly. But there will be at least one label in each class.

1.2 We take m points from the given data set $x_1, x_2, \dots, x_m$ and their labels $y_1, y_2, \dots, y_m \in \{0,1\}$ The selection can be done randomly or by some other method.

Step 2.

2.1 Set (or calculate) $W = \left(w_{ij}\right)_{i,j=1}^{n}$ an input symmetric weight matrix of dimension n. (The matrix of weights of edges (vertex degrees) is calculated using the KNN method, assuming that the vertex degree should not be higher than a given number, for example, 10)

2.2 We form. $F^{2\times m} = (f_{2m})_{i,j}^{n}$ - a matrix of classes of dimension 2 $m$assuming that we have 2 classes. m_j- j-points from the initial labeled sample m.

2.3 We define $b \in R^2$a vector on the set of classes, in which, i^{th} whose element b_i is the fraction of data points belonging to the class i. If this information is not available, we assume $b == \frac{1}{2}1$.

Step 3. In the third step, the degree matrix of the graph D by the following formula:

$$D = W * \mathbf{1} = \left(w_{ij}\right)_{i,j=1}^{n} * \mathbf{1} \tag{14}$$

Here, using the unit matrix **1**, the matrix D is transformed to a diagonal form.

Step 4. The fourth step calculates the non-normalized Laplace matrix of the graph L by the formula

$$L = D - W = D_{ij} - \left(w_{ij}\right)_{i,j=1}^{n} \tag{15}$$

Step 5. In the fifth step, the average label vector is calculated$\bar{y}$:

$$\bar{y} = \frac{1}{m} * F * \mathbf{1} \tag{16}$$

On the labeled set x_1, ..., x_m the computation of the solution $u(x_i)$ will be shifted by an approximately constant value $\bar{y}$for each labeled node:

$$Lu(x_i) = \sum_{j=1}^{n} w_{ij} \left(u(x_i) - u(x_j)\right) \tag{17}$$

Since the function $u(x_i)$ corresponds to some label y_i , while the neighboring points $u(x_j)$ will be constant and have the value $\bar{y}$ equation (18) can be written as:

$$Lu(x_i) \ = \sum_{j=1}^{n} w_{ij}(y_i - \bar{y}) = \ d_i(y_i - \bar{y}) \tag{18}$$

From equation (28), we can conclude that the Laplacian of a labeled vertex is equal to the product of the degree d_i of the node by the centered label vector.

Step 6. In the sixth step, the solution of equation (19) is given in the form $\bar{y}$and write an array of zeros Z of dimension $2(n - m)$:

$$B_{ij} = [F - \ \bar{y}, Z(2, n - m)] \text{ або } B_{ij} = \sum_{j=1}^{m} (y_j - \bar{y})\delta_{ij} \tag{19}$$

Thus, the matrix B will be equal to the matrix F for j from n-m to n. For j from 1 to m, calculate by the formula

$$F_{2m} - \ \bar{y} = \ (f_{2m})_{i,j}^{n} - \frac{1}{m}\sum_{j=1}^{m} y_j \tag{20}$$

An array of zeros Z is created in dimension $2(n - m)$:

$$Z_{ij} = \ \left(z_{2,(n-m)}\right)_{i,j}^{n} \tag{21}$$

Step 7. The seventh step is to create an array of functions $u(x_i)$ which we define as an array of zeros of dimension $2n$ (the starting point):

$$U_{ij} = Z(n, 2) = \ (z_{n2})_{i,j}^{n} \tag{22}$$

Step 8. In the eighth step, the main cycle of the calculation of the function $u(x_i)$ for the number of steps T (11)-(13). Using these equations, we fill in the matrix of solutions $U_{ij} \in R^{2n}$

Step 9. In the eleventh step, the labels are weighted. Feature weighting $u(x_i)$ takes into account the label selection rule and is calculated using the following formula:

$$u(x_i) = u(x_i) * \left(\frac{b_j}{\bar{y}}\right), \tag{23}$$

b_j - vector, the share of data belonging to class j.

Step 10. In the twelfth step, the label selection rule being tested can be written as:

$$u_l(x_i) = argmax_{j \in 1,2} \left(\frac{b_j}{\bar{y}}\right) u_j(x) \tag{24}$$

If formula (24) is true, the algorithm stops, otherwise the previous steps are repeated.

We will consider the settings of the hyperparameters α_1 and β. When considering the regularization problem, the hyperparameter plays an important role. In general, there are several algorithms for its adjustment **[5]**:

Grid search: in this method, a set of possible values for $\lambda$and then iteratively selects the optimal values. Although this can be computationally intensive, especially when multiple hyperparameters are being tuned simultaneously, it ensures a thorough search of the parameter space.

Random search: This approach first sets possible bounds for the value of $\lambda$and then selects a hyperparameter randomly from these bounds. Compared to grid search, random search is less thorough, but can be more efficient in cases of high-dimensional data.

Cross-Validation: Usually used in combination with grid or random search. The idea is to split the data into training and validation samples. The model is trained on the training set with a certain choice of λ and then evaluated on the validation set. The value of λ is chosen that gives the best results on the training set.

Bayesian optimization: This is a more sophisticated method that models the objective function (e.g., validation error as a function of λ) using a Gaussian process and then selects hyperparameters for a principled attempt based on this model. This is especially useful when grid search and random search are time-consuming.

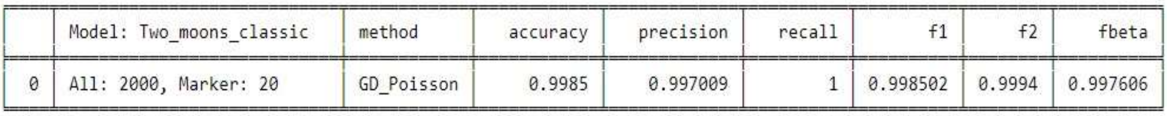
Next, we will use **grid search** and **random search**. To do this, for the hyperparameters α_1 and β a range of values was selected for the hyperparameters and values were selected in increments.

5. Experiments and results

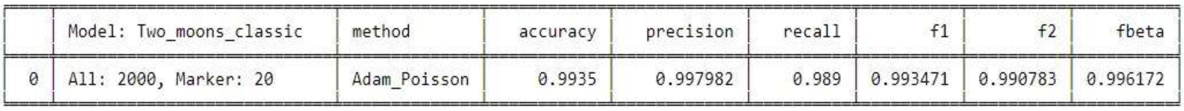
5.1. Results of the calculation of method metrics

Examples for synthetic data. There is the task classification of two moons without intersection and with intersection data. We consider four cases of two months (with, without class intersection, torch and embedded). A total vertex of graph - 2000 points and labeled vertex - 20 by 10 in each class (less than 1%). Here are the results of calculating the following metrics for the proposed method: accuracy, precision, recall, f1, f2, fbeta, which are summarized in the table. Figure 1,2,3 shows the given classes, initial labels, graph construction using the KNN-10 method, classification results, and the confusion matrix.

This result is shown that we can are achieved accuracy more 80%, so cases – two moons with intersection and more 75% for difficult cases two moons embedded. We have that accuracy for modified Poisson with ADAM more than accuracy Poisson with Gradient Descent.



	Model: Two_moons_classic	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	GD_Poisson	0.9985	0.997009	1	0.998502	0.9994	0.997606



	Model: Two_moons_classic	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	Adam_Poisson	0.9935	0.997982	0.989	0.993471	0.990783	0.996172

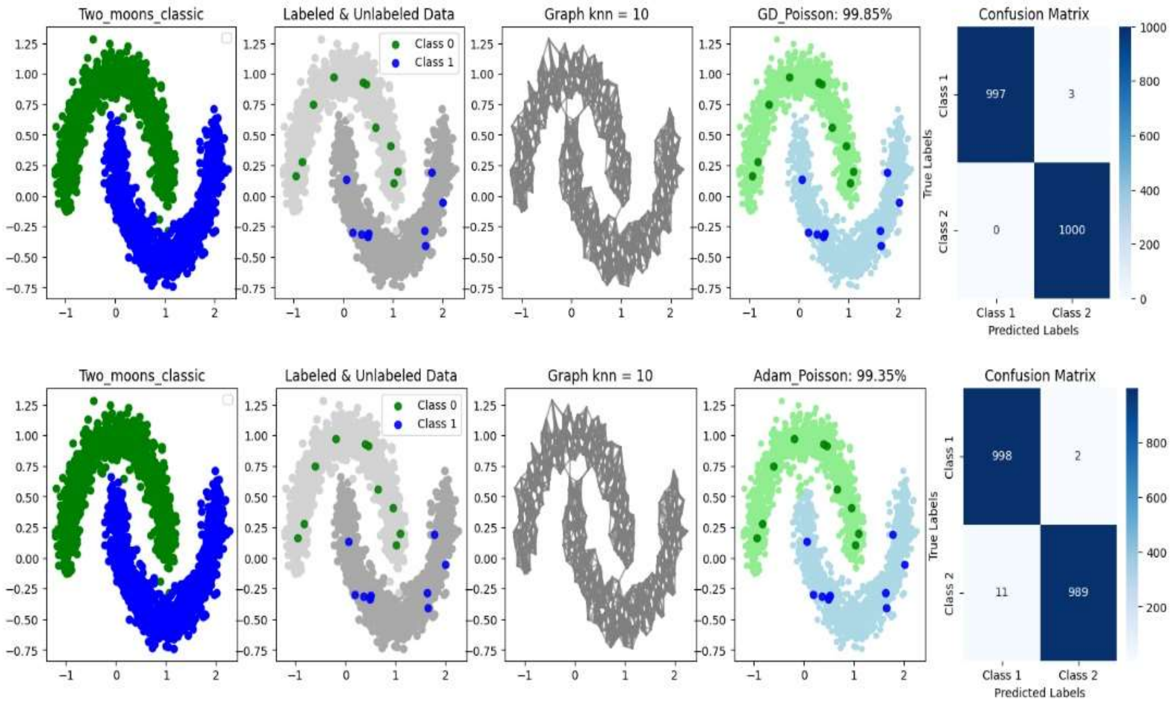
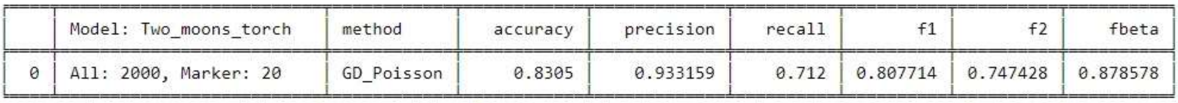
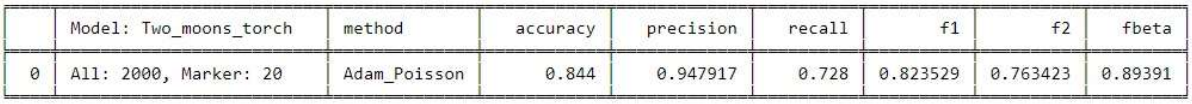


Figure 1: Metrics characteristic of methods in cases Two moons classic without intersection classes



	Model: Two_moons_torch	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	GD_Poisson	0.8305	0.933159	0.712	0.807714	0.747428	0.878578



	Model: Two_moons_torch	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	Adam_Poisson	0.844	0.947917	0.728	0.823529	0.763423	0.89391

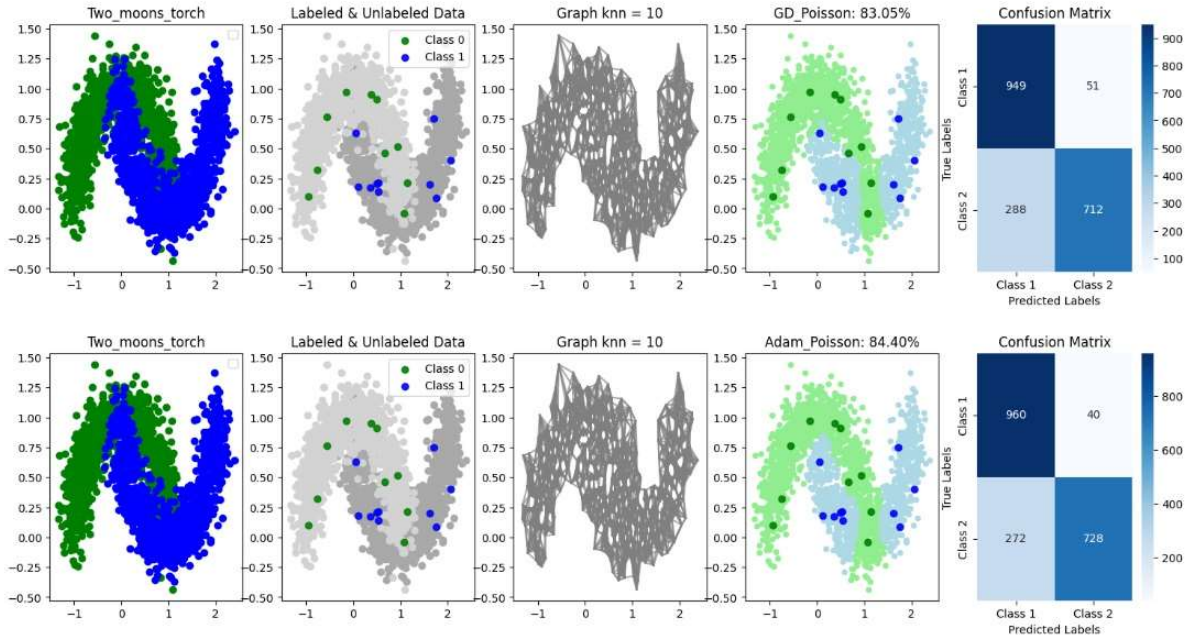
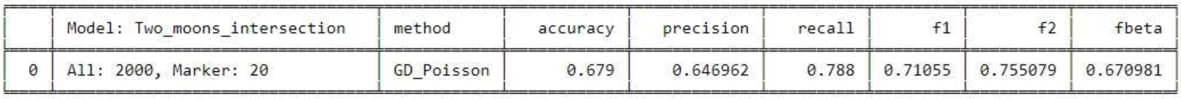
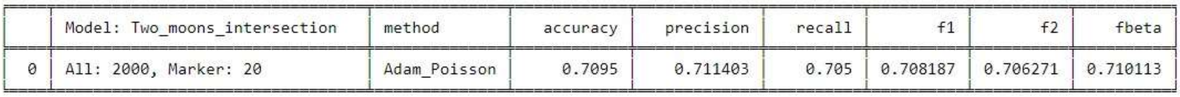


Figure 2: Metrics characteristic of methods in cases Two moons classic with torch classes



	Model: Two_moons_intersection	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	GD_Poisson	0.679	0.646962	0.788	0.71055	0.755079	0.670981



	Model: Two_moons_intersection	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	Adam_Poisson	0.7095	0.711403	0.705	0.708187	0.706271	0.710113

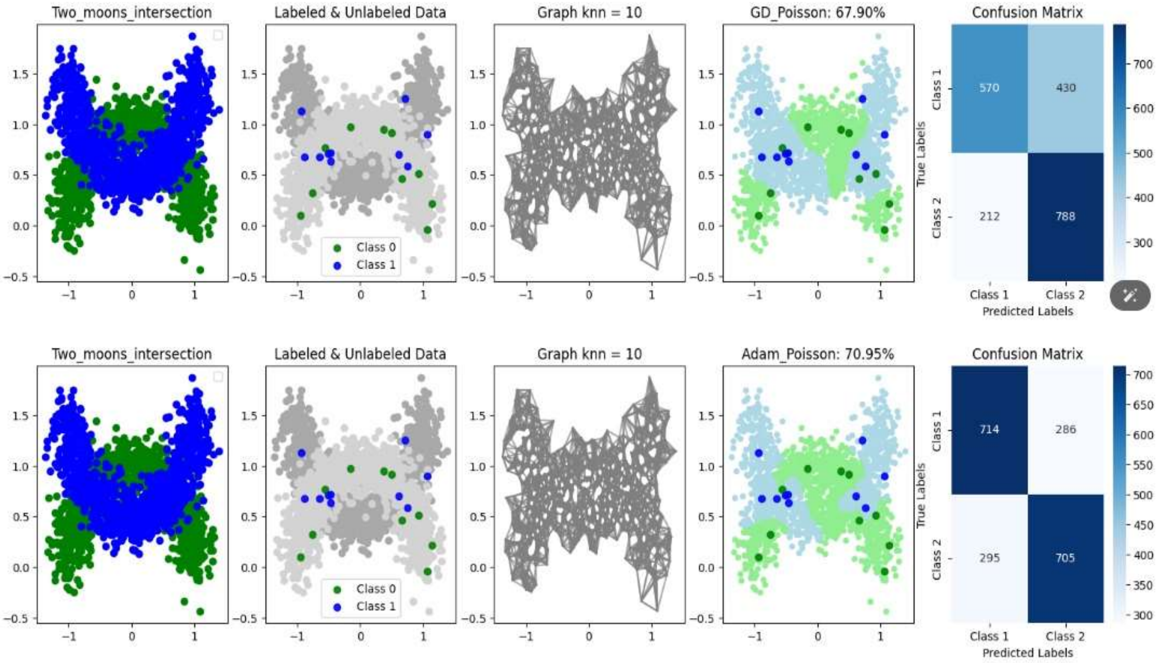
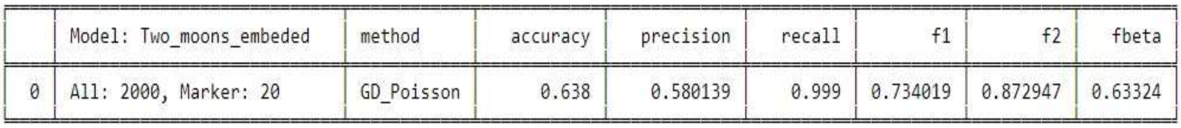
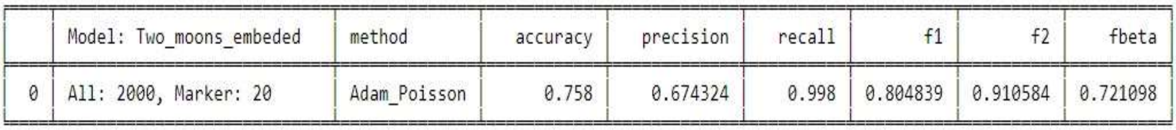


Figure 3: Metrics characteristic of methods in cases Two moons classic with intersection classes



	Model: Two_moons_embeded	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	GD_Poisson	0.638	0.580139	0.999	0.734019	0.872947	0.63324



	Model: Two_moons_embeded	method	accuracy	precision	recall	f1	f2	fbeta
0	All: 2000, Marker: 20	Adam_Poisson	0.758	0.674324	0.998	0.804839	0.910584	0.721098

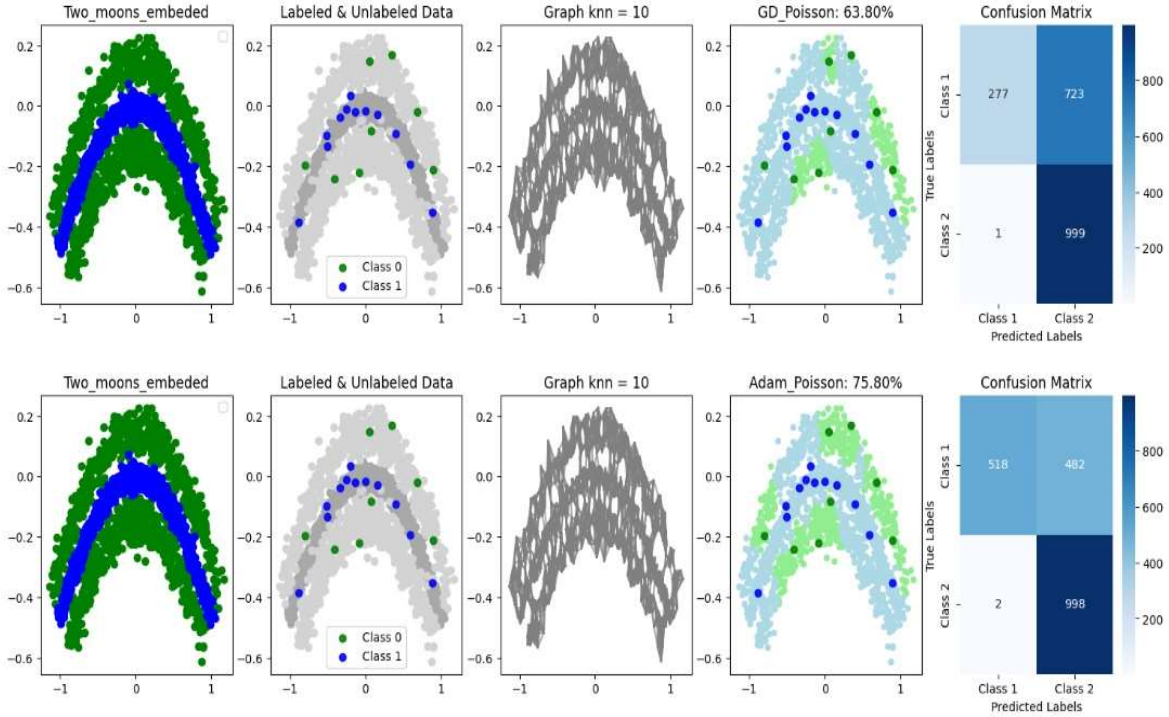


Figure 4: Metrics characteristic of methods in cases Two moons classic with embedded classes

5.2. Comparing with another method

We are considering case number two (two moons tight (torch)) and comparing with next methods: KNN, Laplace Propagation, Laplace Propagation WNLL, WNLL Poisson, Spectral Method, Conjugate gradient, Poisson Descent, Poisson Nesterov, Poisson Nesterov L1, Poisson Nesterov L2, Poison ADAM, Poisson ADAM L2, Poisson ADAGRAD, Poisson heavyball. All number of data 10000, label data {1,2,3,4}.

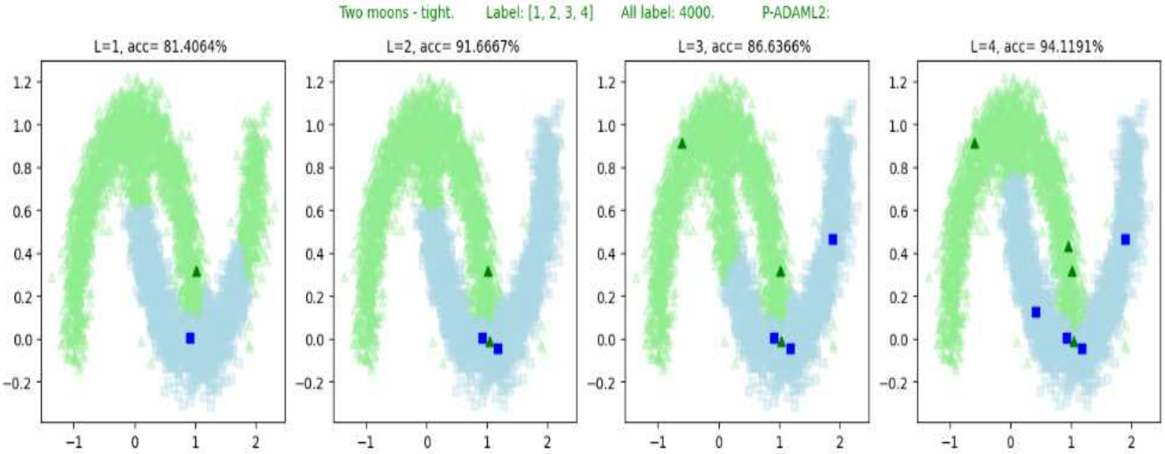
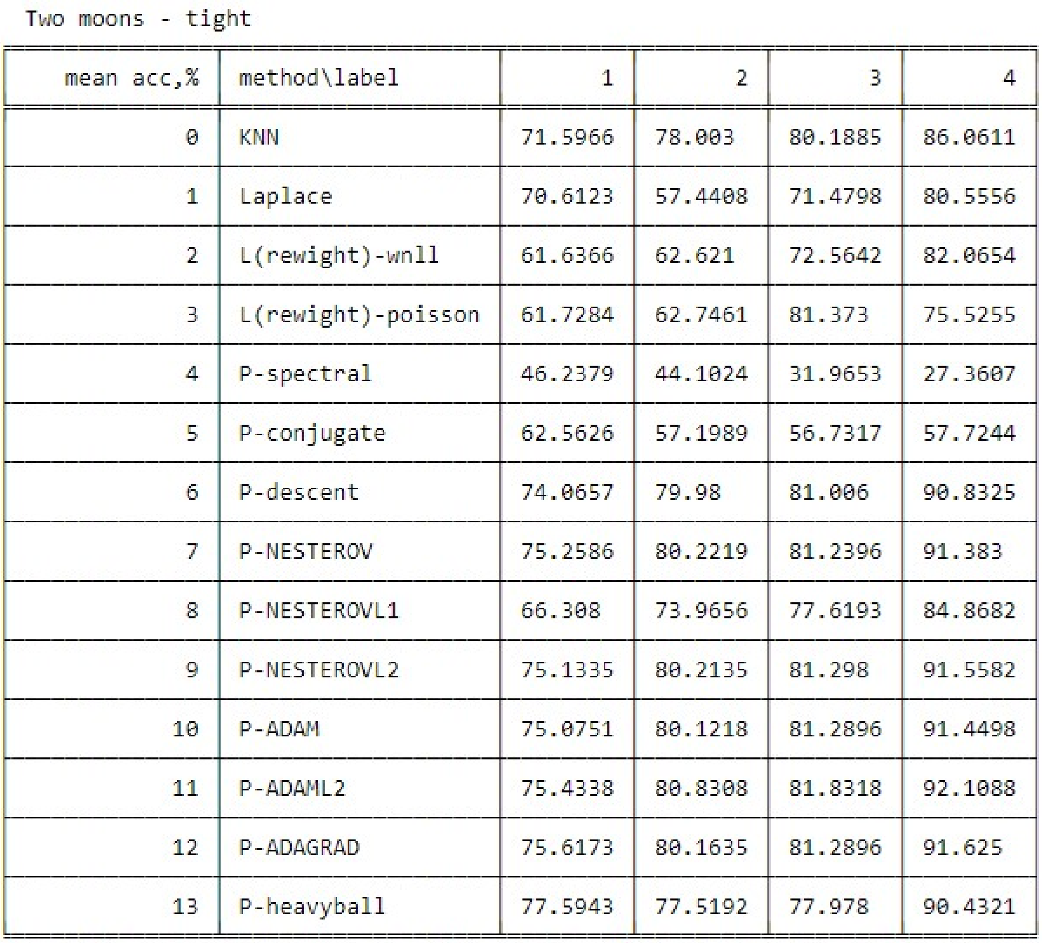


Figure 5: Case of two moons torch for small number label data



Two moons - tight

mean acc,%	method\label	1	2	3	4
0	KNN	71.5966	78.003	80.1885	86.0611
1	Laplace	70.6123	57.4408	71.4798	80.5556
2	L(rewight)-wnll	61.6366	62.621	72.5642	82.0654
3	L(rewight)-poisson	61.7284	62.7461	81.373	75.5255
4	P-spectral	46.2379	44.1024	31.9653	27.3607
5	P-conjugate	62.5626	57.1989	56.7317	57.7244
6	P-descent	74.0657	79.98	81.006	90.8325
7	P-NESTEROV	75.2586	80.2219	81.2396	91.383
8	P-NESTEROVL1	66.308	73.9656	77.6193	84.8682
9	P-NESTEROVL2	75.1335	80.2135	81.298	91.5582
10	P-ADAM	75.0751	80.1218	81.2896	91.4498
11	P-ADAML2	75.4338	80.8308	81.8318	92.1088
12	P-ADAGRAD	75.6173	80.1635	81.2896	91.625
13	P-heavyball	77.5943	77.5192	77.978	90.4321

Figure 6: Accuracy of cases two moons torch for different methods.

Modified method Poisson Label Propagation with regularize is shown the best accuracy for compare other methods so for number label data very small amount {1,2,3,4}.

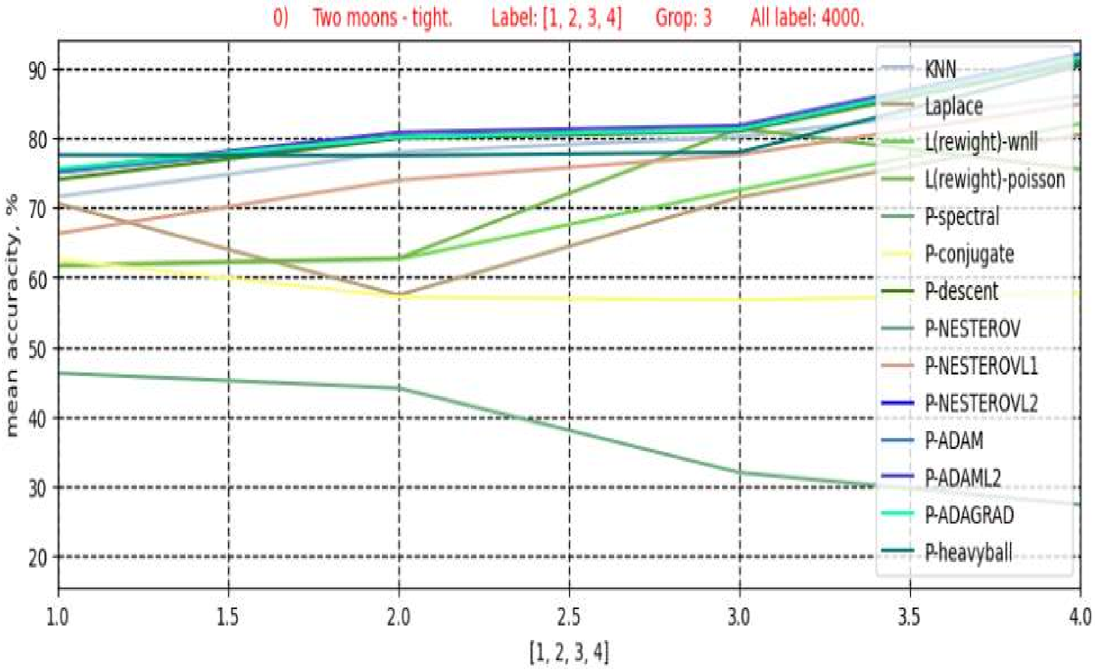


Figure 7: Comparing of accuracy for different method in cases two moons torch

Conclusion

It is proposed a new approach for solving the classification task by using Poisson equations for SSL, which makes it possible to use a small labeled sample and a large amount of unlabeled data. The accuracy is improved by solving the Poisson equation using modern optimization methods such as ADAM. The validity of the obtained results is verified on different variants of the "two moons" sample. As a result, high classification accuracy was achieved (75-85%) - for the case of overlapping classes, (99%) - for the case of 1% of labeled data.

References

[1] Zhu, Xiaojin (Jerry). "Semi-Supervised Learning Literature Survey". Computer Sciences TR 1530 University of Wisconsin - Madison Last modied on Sep 15, 2005. https://minds.wisconsin.edu/handle/1793/60444

[2] Zhu, Xiaojin (Jerry). "Semi-Supervised Learning Literature Survey". Computer Sciences TR 1530 University of Wisconsin - Madison Last modied on July 19, 2008. https://www.researchgate.net/publication/200688680_Semi-Supervised_Learning_Literature_Survey

[3] Jesper E. Van Engelen, J.E., Hoos, H.H. "A survey on semi-supervisedlearning". MachLearn 109, 373-440, (2020). https://doi.org/10.1007/s10994-019-05855-6

[4] Yunsheng Song, Jing Zhang, Chao Zhang, "A survey of large-scale graph-based semi-supervised classication algorithms". International Journal of Cognitive Computing in Engineering, Volume 3, 2022, Pages 188-198, ISSN 2666-3074. https://doi.org/10.1016/j.ijcce.2022.10.002

[5] Z. Song, X. Yang, Z. Xu, 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. doi:10.1109/TNNLS.2022.3155478

[6] X. Yang, Z. Song, I. King, Z. Xu, "A Survey on Deep Semi-Supervised Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 9, pp. 8934-8954, 1 Sept. 2023. doi: 10.1109/TKDE.2022.3220219

[7] Aromal M. A., Akhtar Rasool. "Semi-Supervised Learning Using Graph Data Structure - A Review". 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mo-bile Networks (ICICV) (2021): 894-899. https://api.semanticscholar.org/CorpusID:233138169

[8] Matthieu Toutain, Abderrahim Elmoataz, Olivier LØzoray. "Geometric PDEs on weighted graphs for semi-supervised classication". International Conference on Machine Learning and Applications, IEEE, 2012, Detroit, United States. [2014] hal-01108860> https://www.researchgate.net/publication/278798624_Geometric_PDEs\protect\relax$\@@underline{\hbox{}}\mathsurround\z@$\relaxon_Weighted_Graphs_for_Semi-supervised_Classification

[9] J. Calder. Brendan Cook. Matthew Thorpe. Dejan Slepcev. "Poisson Learning: Graph-based Semi-Supervised Learning At Very Low Label Rates". Proceedings of the 37th International Conference on Machine Learning. C Proceedings of Machine Learning Research. 2020. Hal DaumØ III Aarti Singh. Fpmlr-v119-calder20a. PMLR. P 1306-1316. V 119. https://proceedings.mlr.press/v119/calder20a.html

[10] Calder, J.: "The game theoretic p-Laplacian and semi-supervised learning with few labels". Nonlinearity 32(1), 301330 (2018). DOI:10.1088/1361-6544/aae949

[11] Y. Chong, Y. Ding, Q. Yan, and S. Pan, "Graph-based semi-supervised learning: A review". Neurocomputing, vol. 408, pp. 216-230,2020. https://doi.org/10.1016/j.neucom.2019.12.130

[12] Konstantin Avrachenkov, Inria Sophia-Antipolis, Maximilien Dreveton, "Chapter 5. Graph-based semi-supervised learning". France. 2021. http://dx.doi.org/10.1561/9781638280514.ch5

[13] C. Chenetal. "Interactive Graph Construction for Graph-Based Semi-Supervised Learning", in IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 9, pp. 3701-3716, Sept. 1, 2021.https://doi.org/10.1109/TVCG.2021.3084694

[14] Zhang, Chen Guang, Yan Zhang, Xia Huan Zhang. "Graph Based Semi-Supervised Learning Method for Imbalanced Dataset". Applied Mechanics and Materials. Trans Tech Publications, Ltd., May 2014. https://doi.org/10.4028/www.scientific.net/amm.556-562.4040

[15] Zhuang L, Zhou Z, Gao S, Yin J, Lin Z, Ma Y. "Label Information Guided Graph Construction for Semi-Supervised Learning". IEEE Trans Image Process. 2017 Sep; 26(9): 4182-4192. https://doi.org/10.1109/TIP.2017.2703120

[16] Liang, J., Cui, J., Wang, J. et al. "Graph-based semi-supervised learning for improving the quality of the graph dynamically". MachLearn 110, 1345-1388 (2021). https://doi.org/10.1007/ s10994-021-05975-y

[17] Mehdi Sajjadi, Mehran Javanmardi, Tolga Tasdizen. "Regularization With Stochastic Transfor-mations and Perturbations for Deep". NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems. December. 2016. Pages. 11711179. https://proceedings. neurips.cc/paper_files/paper/2016/file/30ef30b64204a3088a26bc2e6ecf7602-Paper.pdf

[18] Ke Chen, Shihai Wang. "Regularized Boost for Semi-Supervised Learning". Part of Advances in Neural Information Processing Systems 20 (NIPS 2007) https://papers.nips.cc/paper_files/ paper/2007/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf

[19] S. Sivananthan, "Multi-penalty regularization in learning theory", Journal of Complexity, Volume 36, 2016, Pages 141-165, ISSN 0885-064X, https://doi.org/10.1016/j.jco.2016.05.003

[20] Zheng-Chu Guo, Shao-Bo Lin, Lei Shi, "Distributed learning with multi-penalty regularization", Applied and Computational Harmonic Analysis, Volume 46, Issue 3, 2019, Pages 478-499, ISSN 1063-5203, https://doi.org/10.1016/j.acha.2017.06.001

[21] Naumova, V. Pereverzyev, S. V. "Multi-penalty regularization with a component-wise penalization". Inverse Problems, 2013. IOP Publishing - 075002 - 7 VL - 29 SN - 0266-5611. https: //dx.doi.org/10.1088/0266-5611/29/7/075002

[22] Markus Grasmair, Timo Klock, Valeriya Naumova, "Adaptive multi-penalty regularization based on a generalized Lasso path", Applied and Computational Harmonic Analysis, Volume 49, Issue 1, 2020, Pages 30-55, ISSN 1063-5203, https://doi.org/10.1016/j.acha.2018.11.001

[23] Mikhail Belkin, Partha Niyogi, Vikas Sindhwani. "Manifold Regularization: A Geometric Frame-work for Learning from Labeled and Unlabeled Examples"; Journal of Machine Learning Research, 7,(85):2399-2434, 2006. http://jmlr.org/papers/v7/belkin06a.html

[24] M. Fornasier, V. Naumova, S.V. Pereverzyev. "Parameter choice strategies for multi penalty regularization". SIAM J. Numer. Anal., 52 (4) (2014), pp. 1770-1794. https://www.ricam.oeaw.ac.at/files/reports/13/rep13-10.pdf

[25] K. Ito, B. Jin, T. Takeuchi. "Multi-parameter Tikhonov regularization - an augmented approach". Chin. Ann. Math. Ser. B, 35 (3) (2014), pp. 383-398. https://www.researchgate.net/publication/ 241689312_Multi-Parameter_Tikhonov_Regularization_--_An_Augmented_Approach

[26] S. Lu, S.V. Pereverzev. "Multi-parameter regularization and its numerical realization". Numer. Math., 118 (1) (2011), pp. 1-31. https://www.researchgate.net/publication/225666729_ Multi-parameter_regularization_and_its_numerical_realization

[27] R. Llugsi, S. E. Yacoubi, A. Fontaine and P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andeancity of Quito", 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 2021, pp.1-6, https://doi.org/10.1109/ETCM53643.2021.9590681

[28] V. Sineglazov, K. Riazanovskiy, A. Klanovets, E. Chumachenko, and N. Linnik, "Intelligent tuberculosis activity assessment system based on an ensemble of neural networks", Comput. Biol. Med., vol. 147, Aug. 2022, Art. no. 105800. https://www.sciencedirect.com/science/article/ pii/S0010482522005613

[29] Sineglazov, V., Kot, A. "Design of hybrid neural networks of the ensemble structure". Eastern-European Journal of Enterprise Technologies, 2021. 1/4 (109), 3145. https://doi.org/10.15587/ 1729-4061.2021.225301

[30] Zgurovsky, M., Sineglazov, V., Chumachenko, E. "Formation of Hybrid Articial Neu-ral Networks Topologies. In: Articial Intelligence Systems Based on Hybrid Neural Networks". Studies in Computational Intelligence, 2021. vol 904. Springer, Cham. https://doi.org/10.1007/ 978-3-030-48453-8-3

[31] Zgurovsky, M., Sineglazov, V., Chumachenko, E. "Classication and Analysis of Multi-criteria Optimization Methods". In: Articial Intelligence Systems Based on Hybrid Neural Networks. Studies in Computational Intelligence, vol 904. Springer, Cham. 2021. https://doi.org/10.1007/ 978-3-030-48453-8-2

[32] Zgurovsky, M., Sineglazov, V., Chumachenko, E. "Classication and Analysis Topologies Known Articial Neurons and Neural Networks". In: Articial Intelligence Systems Based on Hybrid Neural Networks. Studies in Computational Intelligence, 2021. vol 904. Springer, Cham. doi.org/10.1007/ 978-3-030-48453-8-1

[33] Smola, Alexander J.; Kondor, Risi. "Kernels and regularization on graphs". (English) Zbl 1274.68351 Schölkopf, Bernhard (ed.) et al., Learning theory and kernel machines. 16th annual conference on learning theory and 7th kernel workshop, COLT/Kernel 2003, Washington, DC, USA, August 2427, 2003. Proceedings. Berlin: Springer (ISBN 3-540-40720-0/pbk). Lect. Notes Comput. Sci. 2777, 144-158 (2003). https://people.cs.uchicago.edu/~risi/papers/SmolaKondor.pdf

[34] Garcia-Cardona, Cristina; Merkurjev, Ekaterina; Bertozzi, Andrea; Flenner, Arjuna; Percus, Allon. "Multiclass Data Segmentation Using Diuse Interface Methods on Graphs". IEEE Transactions on Pattern Analysis and Machine Intelligence. 36. (2013). doi:10.1109/TPAMI.2014.2300478. https://ieeexplore.ieee.org/document/6714564

[35] R. Couillet, Z. Liao and X. Mai, "Classication Asymptotics in the Random Matrix Regime", 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 2018, pp. 1875-1879, doi: 10.23919/EUSIPCO.2018.8553034. https://ieeexplore.ieee.org/document/8553034

[36] Jung, Alexander. "Sparse Label Propagation". (2016). https://www.researchgate.net/publication/311425813_Sparse_Label_Propagation

[37] Shi, Zuoqiang; Osher, Stanley; Zhu, Wei. (2017). "Weighted Nonlocal Laplacian on Interpolation from Sparse Data". Journal of Scientic Computing. 73. doi:10.1007/s10915-017-0421-z

[38] O. Chapelle, B. Scholkopf and A. Zien, Eds., "Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]," in IEEE Transactions on Neural Networks, vol. 20, no. 3, pp. 542-542, March 2009, doi: 10.1109/TNN.2009.2015974 https://ieeexplore.ieee.org/document/4787647

[39] A. Oliver, A. Odena, C.A. Rael, E.D. Cubuk, I. Goodfellow. Realistic evalua-tion of deep semi-supervised learning algorithms - Advances in neural information processing systems, 2018 https://scholar.google.com /citations?view_op=view_citation&hl=en&user=Suu45K8AAAAJ&sortby=pubdate&citation_for_view=Suu45K8AAAAJ:d1gkVwhDpl0C

[40] Rie Kubota Ando, Tong Zhang. "Learning on graph with Laplacian regularization". In Proceed-ings of the 19th International Conference on Neural Information Processing Systems (NIPS'06). MIT Press, Cambridge, MA, USA, 2532. 2006 https://dl.acm.org/doi/10.5555/2976456.2976460

[41] Xu, Shuo; An, Xin; Qiao, Xiao-Dong; Zhu, Li-Jun; Li, Lin. "Semi-Supervised Least Squares Support Vector Regression Machines". Journal of Information and Computational Science. 8. 885-892. (2011). https://www.researchgate.net/publication/215913159_Semi-Supervised_Least_Squares_Support_Vector_Regression_Machines

[42] M. Toutain, A. Elmoataz and O. LØzoray, "Geometric PDEs on Weighted Graphs for Semi-supervised Classication", 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 2014, pp. 231-236, 2014 doi:10.1109/ICMLA.2014.43

[43] A. Tsitsulin, J. Palowitch, B. Perozzi, E. Müller, "Graph clustering with graph neural networks". Journal of Machine Learning Research 24, 1-21, 2023 https://jmlr.org/papers/volume24/20-998/ 20-998.pdf