UDC 004.67.2
DOI https://doi.org/10.24144/2616-7700.2021.39(2).125-144

N. Boyko

Lviv Polytechnic National University,
Ph.D., Associated Professor at the Department of Artificial Intelligence,
nataliya.i.boyko@lpnu.ua
ORCID: https://orcid.org/0000-0002-6962-9363

SEMI-SUPERVISED LEARNING PARADIGM ANALYSIS FOR CLASSIFICATION OF MULTIMODAL DATA

The paper considers machine learning algorithms. The focus is on semi-controlled learning, which seems to be the balance between teaching accuracy with a teacher and the cost of teaching methods without a teacher. Examples of careful processing of labelled data sets for which supervised learning can be very effective are considered. The semi-supervised and supervised approaches are compared, and the effectiveness of each is analyzed. The paper considers S3VM and TSVM approach. The work aimed to investigate whether semi-controlled methods can compete with controlled or even surpass them. Applying these approaches to the proposed dataset to determine a more accurate classification of data, namely at the reference limit, is described.

Keywords: semi-supervised learning, support vector machine, transudative support vector machines, semi-supervised support vector machines, application programming interface.

1. Introduction. The work is performed by studying semi-supervised learning, which is learning with a teacher. Supervised learning is a learning paradigm related to studying how computers and natural systems, such as humans, remember in the presence of both labelled and unlabeled data. Traditionally, learning is studied either in an uncontrolled paradigm (e.g., clustering, external shape detection), where all data is unlabeled, or in a controlled paradigm (e.g., classification, regression), where all information is labelled [1].

The purpose of semi-controlled learning is to understand how a combination of labelled and unlabeled data can change learning behaviour and develop algorithms that use such a combination [2, 6].

Supervised learning is of great interest in machine learning and data exchange, as it can use readily available unlabeled data to improve tasks when labelled data is scarce or expensive. Supervised learning also shows potential as a quantitative tool for understanding human categories of knowledge where much of the contribution is not apparent [3, 8].

The object of study of this work is semi-supervised learning is an extension of supervised and supervised learning. SSL algorithms, as a rule, provide a way to learn about the structure of data from unlabeled examples Reducing the need for labels. Most problems in the real world have a lot of data, and labelling them is a cumbersome or even impossible task. Supervised learning is one of the approaches to overcoming these types of problems. For training, he uses only a small set, marked by substantial unlabeled data. In semi-controlled training, it is essential what data are labelled and, depending on the position of the data, its effectiveness changes [4, 10].

The semi-supervised learning paradigm attracts much attention in many different areas, from bioinformatics to web mining. It is easier to get unlabeled than labelled data because it requires less effort, experience and time [5, 12]. In this context, traditional supervised learning is limited to using labelled data to build a model.

However, SSL is a learning paradigm for designing models with both labelled and unlabeled data.In essence, SSL methods use unspecified samples to change or rethink a hypothesis derived only from labelled samples.

2. Review of methods. Before defining the approaches and benefits of SSL and supervised learning algorithms, you should first learn what supervised learning is in everyday use:

– Classification of web pages;
– Detection of fraud;
– Face recognition;
– Language recognition;
– Genetic sequencing.

There is no doubt about the potential and growth of machine learning, and semi-controlled education often seems to be a balance between the accuracy of teaching with a teacher and the cost of teaching methods without a teacher [9, 13, 16].

Due to the careful processing of labelled data sets, supervised learning can be very effective.

Imagine the following situation (See Figure 1):

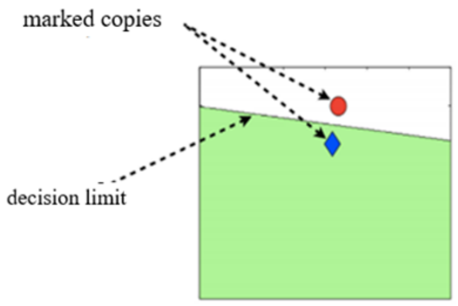

Figure 1. Finding a solution.

In Figure 1, you have only two data points that fall into two different categories, and the drawn line is the limit of any controlled model [16, 17].

Now, let's say we add some undefined data to this data, as shown in the image below:

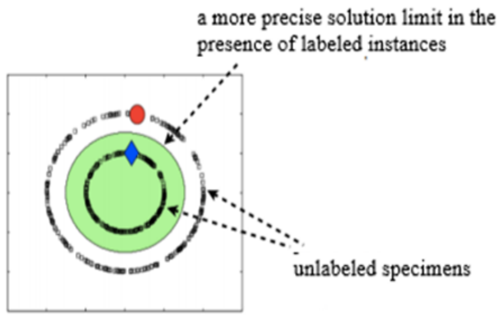
a more precise solution limit in the presence of labeled instances
unlabeled specimens

Figure 2. Adding unlabeled instances.

In Figure 2 you can see the difference between the two images listed above, the same can be said that after adding data without labeling, the decision limit of our model has become more accurate.

Thus, among the advantages of using unlabeled data are:

– Labeled data is expensive and difficult to obtain, while unlabeled data is plentiful and cheap.
– This improves the stability of the model through a more precise decision boundary.

Typically, the purpose of categorizing images is to classify whether the image belongs to the category or not. In this work, not only images are used for modelling, but keywords related to labelled and unlabeled images and unlabeled images are used to improve the classifier through semi-control training.

Semi-supervised SVM is an SVM with variable ξi. This model is based on the assumption that y_i can be either -1 or 1.

Variables ξi are variables, one for each sample, introduced to reduce the strength imposed by the initial condition $(\min \|w\|)$, which is based on a hard stock that incorrectly classifies all samples that are on the wrong side. They are determined by the loss of the hinge as follows [12]:

$$\max\left(0.1 - \ y_i(w^T x_i + b)\right) \tag{1}$$

With these variables, we allow some points to cross the boundary without classifying them if they remain at a distance controlled by the corresponding weak variable (which is also minimized during the training phase to avoid uncontrolled growth). The following diagram shows a schematic representation of this process:

The last elements of each high-density region are reference vectors. Between them is a low-density region (it can also be zero density), where lies our separate hyperplane [13].

Theoretically, each function, which is always bounded by two hyperplanes containing reference vectors, is a good classifier, but we need to minimize the empirical risk (and, yes, the expected risk). Therefore, we are looking for the maximum margin between areas of high density. This model can separate two dense regions with irregular boundaries. By adopting the kernel function, it can also work in nonlinear scenarios. The main issue at the moment is the question of the best strategy for the integration of labelled and unlabeled samples when we need to solve a similar problem in a semi-supervised scenario [6, 9].

The first element to consider is the relationship. If we have a low percentage of marked scores, the problem is mainly controlled, and the generalization skills learned through the training kit should be sufficient to correctly classify all unmarked scores. On the other hand, if the number of unlabeled samples is much larger, we return to the almost pure clustering scenario (as discussed in the section on generative Gaussian mixtures). This means that to use the power of semi-control methods in low-density separation problems, we must consider situations where the labelled / unlabeled ratio is approximately 1.0 [1, 8].

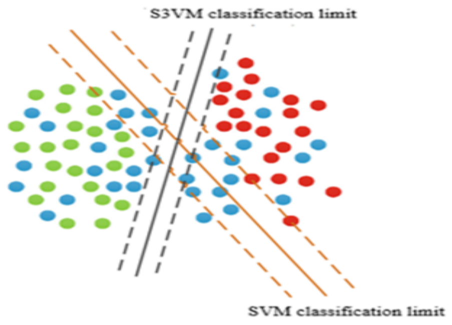

Figure 3. Hyperlines SVM and S3VM.

Figure 3 shows two hyperlines - one SVM method, the other - S3VM. Blue dots - unlabeled data, red - class 1, green - class 2.

We can see that SVM has a large gap from the hyperline, but there is a reasonably high density. Although in S3VM, this interval is smaller, it is more accurate, and we see that the boundaries of the reference vectors are located just on the last elements. Then the gap is empty, i.e. the hyperline correctly classifies the classes [5, 10].

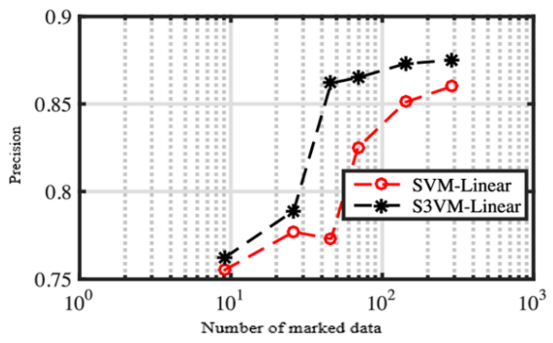

Figure 4. Comparison of the accuracy of linear SVM and S3VM.

Figure 4 shows that the linear S3VM works much better than SVM at a relatively small amount of labeled data $(10^1, 10^2)$. As the amount of labeled data increases, the accuracy of these two methods converges to one value [13].

Transductive Support Vector Machines. Another approach to the same problem is offered by Transductive Support Vector Machines, which are especially suitable when the unlabeled sample is not very noisy. The overall structure of the data set is robust. Everyday use of TSVM is a classification on a data set that contains data points obtained from the same data generation process (e.g., medical photographs collected using the same tool) but only partially labelled for, for example, economic reasons. Because all images can be trusted, the TSVM can use a dataset structure to achieve accuracy more significantly than the controlled classifier.

The idea is to keep the original goal with two sets of variables - the first for labelled samples and the second for unlabeled:

$$\min\left[||w|| + C_L \sum_{i=1}^{N} \eta_i + \ C_U \sum_{i=N+1}^{N+M} \xi_j\right] \tag{2}$$

Formula (2) describes minimization of the objective function of the TSVM method. Because this is a transductive approach, we need to treat unlabeled samples as variable-labelled (depending on the learning process), imposing a constraint similar to controlled items. From a certain point of view, this is equivalent to introducing a preliminary idea of the final classification, firmly based on the cluster and the assumption of smoothness.

In other words, the TSVM can trust the structure of the data set more than the S3VM, and the data scientist has more flexibility in choosing behaviour. Different combinations of CL and CU give results from the complete trust given to the labelled points to the opposite condition. As explained in the introduction, the purpose of transductive learning is only to classify unlabeled samples using both labelled and data set structures. However, contrary to inductive methods, the limitations imposed by labelled samples can be relaxed in favour of a more geometrically consistent solution.

Comparison of modified SVM methods. An alternative to S3VM is TSVM, which tries to minimize the target with a condition based on variable labels. Thus, the problem is divided into two parts: controlled, which is precisely the same as the standard SVM, and partially controlled, which has a similar structure but does not have fixed y labels. This problem is also not convex, and various optimization strategies need to be evaluated to find the best trade-off between accuracy and computational complexity. TSVM's transductive approach relies heavily on the data set structure; this is only a reasonably reasonable choice when both labelled and labelled samples are known to be taken from the same data generation process.

Also, the TSVM can trust the data structure more than the S3VM. However, S3VM is particularly suitable when the design of the unlabeled sample is partially (or even wholly) unknown, and the primary responsibility for labelling should lie with the labelled examples.

Formulation of the problem. It is necessary to solve the classification problem, namely, to find the best strategies for semi-managed classification, which could compete with the approaches of controlled learning. The idea is to minimize costs with unlabeled data, thus proving the need for semi-controlled systems.

You need to select a dataset for this task. Make Classification is often used to compare methods and solve various classification problems because we can manually set many parameters, thus creating our dataset.

This dataset is generated to solve n-class classification problems.

First, clusters of points are created, usually distributed (std = 1, std is a tensor with a standard deviation of the normal distribution of each source element) with vertices of n_informative-sized hypercube with sides of length 2 * class_sep and each class is assigned an equal number of clusters. This introduces an interdependence between these features and adds data of different types of noise.

The main idea is to generate centroids by some randomness for different clusters by their number. How accurately each centroid of the cluster can be controlled is set by the value of the classes' parameter. Because data will be generated in a Gaussian distribution with a deviation of 1, this argument contains how each cluster overlaps with other clusters and adding correlations for informative features by multiplying the matrix of functions by a randomly generated covariance matrix.

For each data point, the function changes its target to some random class (actually, it can be adjusted to an actual type) with a speed of flip_y, a number between 0 and 1. This makes some noise in the data set.

There are two things to control noise (or how different classes of data overlap) class_sep: determines how clusters are divided. A large value will cause the data sets to intersect less.

flip_y: determines how many data points are marked randomly (noise).

We will generate the following dataset with the following parameters:

Number of signs = 2,

Number of points = 1000,

Number of classes = 3.

As a result, we will receive such dataset:

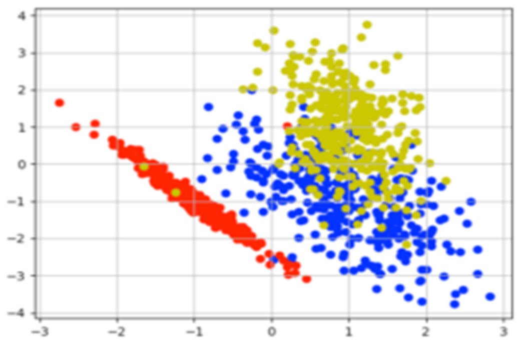

Figure 5. Dataset visualization.

Create the worst data set by setting flip_y = 0.3:

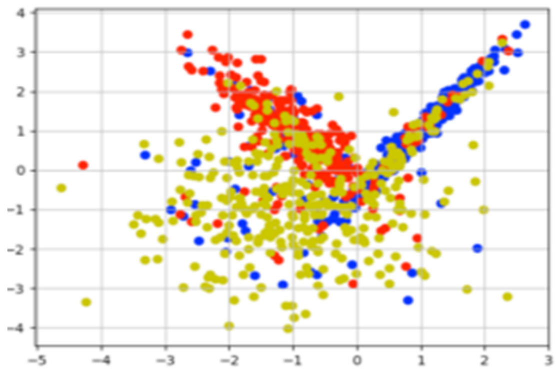

Figure 6. Dataset visualization.

In comparisons I use the following dataset:

Number of signs = 2,

Number of points = 200,

Number of classes = 2,

Randomness score = 1000.

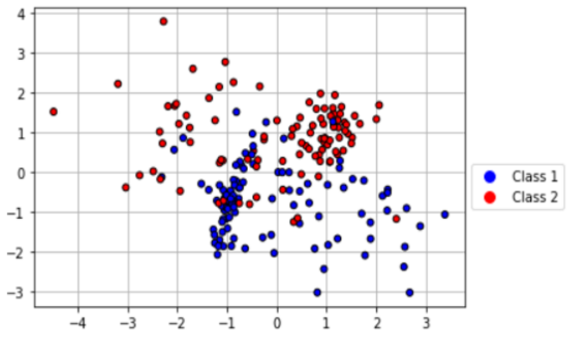

Figure 7. Visualization of the dataset for further work.

Figure 7 shows two classes. Blue dots - class 1, red - class 2. In the future we will divide these dots into the third class - unlabeled data.

The next step is to implement the best strategies for conducting a semi-supervisory classification that could compete with the supervised approach.

To implement the calculations, you must additionally download the following libraries:

– Sklearn;
– Scikit-learn;
– Matplotlib;
– Numpy;
– SVC.

Add a two-dimensional dataset, with labeled and unlabeled data (50% each). There will be 200 marked, 150 unmarked.

Number of marked (Marked) - 200,

Number of unmarked labels (notMarked) - 150.

Classified dataset:

```
[0 0 0 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1
0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 0
...
0 0 0 0 1 0 0 1 1 0 0 0 1 1 0]
```

In this array we contain the notation of two classes - class 1 (notation 0), and class 2 (notation 1).

Add unlabeled data and change the structure of the dataset:

```
[-1 -1 -1 1 1 1 -1 1 1 1 1 1 -1 1 -1 -1 -1 -1 -1 -1 1 -1 1 1
1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 -1 1 -1 -1 -1 -1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
...
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0]
```

Now the marked data is marked as -1 (class 1) and 1 (class 2), the unmarked data is 0.

We also have the meaning of signs:

```
[[-1.05545237e+00 -9.32716065e-01]
[-1.09043812e+00 -1.50493373e+00]
[ 1.22265567e+00 -1.67332726e+00]
[-8.89284086e-01 -6.713955478e-01]
[-3.42811427e-01 2.156535159e+00]
[ 4.19337679e-01 -1.156065519e+00]
[-1.74121223e+00 7.535533374e-01]
[ 1.02333526e+00 -3.215662196e-01]
[ 2.55739074e+00 -2.381244523e+00]
[-1.68630379e+00 2.599753855e+00]
[-9.12360330e-01-1.860154292e+00]
[-1.07341058e+00 -1.866005323e+00]
[-7.41616980e-01 2.6384354742e-01]
[ 1.40819086e-01 8.4433174723e-01]
[-2.186617475e+00 1.65834393e+00]
[ 7.62332115e-01 -1.9242596277e+00]
[-9.30308962e-01 -4.6841778176e-01]
[-8.69596340e-01 -1.9678237778e-01]
[ 1.259574172e+00 1.037730881e-01]
[-4.745967011e-01 6.549572729e-01]
. . . .
[-1.19220109e+00 -2.076067172e+00]]
```

We visualize this data.

Initial data:

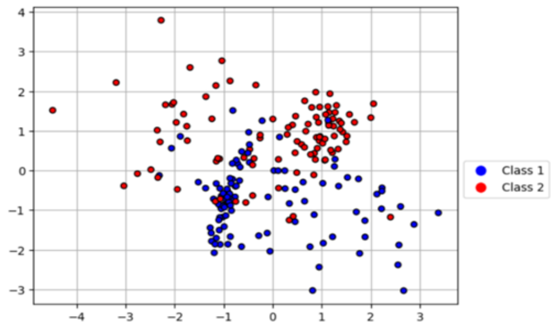

Figure 8. Visualization of the initial dataset.

Let’s mark 150 points as not marked we will receive the following schedule:

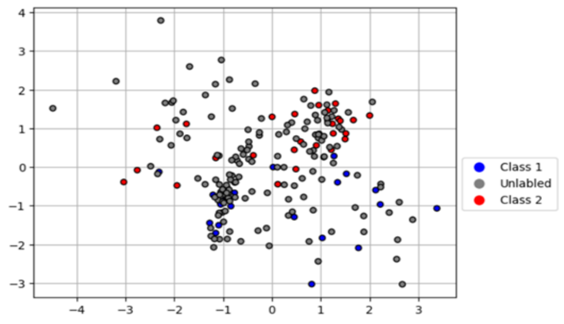

Figure 9. Visualization of the dataset after adding unlabeled data.

Figure 9 shows class 1 - blue dots, class 2 - red dots, and unmarked dots, which are shown in gray.

Initialize our constant C with a value of 0.1.

We also need to specify the initial variables w (normal vector), b (offset along the y axis) and weak variables η_i, ξ_i, z_i.

For the values of w and b the sample will be [-0.1, 0.1], for the rest - [0.0, 0.1].

w - [-0.01975875 0.02131162] size= 2,

$\boldsymbol{\eta}$**i** - [0.04901986 0.02258875 0.01560311 0.04587382 0.0352636 0.00871087 0.08424851 0.0857313 0.0733342 0.09528408 0.05455903 0.06449431 0.05901452 0.04758597 0.01772247 0.07832201 0.07269677 0.09613403 0.02368684 0.06946833] - size = (Marked – notMarked) = 50,

$\boldsymbol{\xi}$**i** - [0.09799687 0.08343227 0.00994421 0.09511786 0.03769766 0.04774885 0.03169544 0.03145913 0.07661224 0.04653426 0.0142706 0.09941006 0.00069853 0.03098653 0.00093538 0.08666854 0.09679364 0.03014508 0.07180447 0.00543835 0.05054611 0.09301137 0.0755189 0.03497578

. . . .

0.0983856 0.09866195 0.08204016 0.09872704 0.06635472 0.07370298]
– size = notMarked = 150,

zi - [0.07221 0.09092026 0.03663539 0.07945663 0.0902111 0.05445606 0.01707353 0.01452738 0.08280087 0.08758461 0.09761866 0.07808664 0.04225707 0.07620617 0.06079858 0.02328625 0.05549894 0.07257086

. . .

0.08668178 0.08250609 0.03545815 0.01125552 0.06446315 0.01788753 0.03401358 0.05677734 0.09136062 0.08023616 0.08339306 0.0975363]
- size = notMarked = 150,

b - [0.0529] – size = 1.

Next we need to minimize the following expression:

$$\min\left[||w|| + C\left(\sum_{i=1}^{N}\eta_i + \sum_{i=N+1}^{N+M}\min(\xi_j, z_j)\right)\right]. \tag{3}$$

Formula 3 describes minimization of the objective function of the S3VM method. The first term of the formula shown in Figure 3.3 imposes the standard SVM condition on the maximum separation distance, and the second block is divided into two parts:

- We need to add N weak variables to ensure a soft gap for the labelled samples.
- At the same time, we need to consider unmarked points classified as +1 or -1. Thus, we have two corresponding sets of variables ξi and zi. However, we want to find the minor variable for each possible pair to ensure that the unlabeled sample is placed in a subspace where maximum accuracy is achieved.

Therefore, we need to impose the following restrictions:

$$\begin{gathered} y_i\left(w^T x_i + b\right) \geq 1 - \eta_i \ \ , \ \eta_i \ \geq 0 \ \forall \ i \in (1, N) \\ \left(w^T x_i - b\right) \geq 1 - \xi_j \ , \xi_j \geq 0 \ \forall \ j \in (N+1, N+M) \\ -\left(w^T x_i - b\right) \geq 1 - \ \mathrm{z}_j \ , \ \mathrm{z}_j \ \geq 0 \ \forall \ j \in (N+1, N+M) \end{gathered} \tag{4}$$

Formula 4 describes conditions for the objective function.

The first constraint in Figure 3.4 is limited to the marked points, it is the same as in the controlled SVM. The next two instead consider the possibility that an unspecified label can be classified as +1 or -1.

We minimize the function with the above restrictions:

Table **1**. *Minimize the target function S3VM.*

Step	$\sum_{\mathrm{i}=0}^{n} \boldsymbol{\eta}\mathbf{i}$	$\sum_{\mathrm{i}=0}^{n} \mathbf{min}(\boldsymbol{\xi}\mathbf{i}, \mathbf{zj})$	**Vector W**	**Target function**
0	2.410	5.182	[-0.022 0.042]	7.594
100	2.410	5.182	[-0.022 0.042]	7.594
200	2.410	5.182	[-0.022 0.042]	7.594
300	2.410	5.182	[-0.022 0.042]	7.594
400	-47.590	-144.818	[-0.000 -0.000]	-192.408
500	-47.590	-144.818	[-0.000 -0.000]	-192.408
600	-47.590	-144.818	[-0.000 -0.000]	-192.408
700	-47.590	-144.818	[-0.000 -0.000]	-192.408
800	-297.601	-894.851	[0.000 -0.000]	-1192.452
900	-297.601	-894.851	[0.000 -0.000]	-1192.452
1000	-297.601	-894.851	[0.000 -0.000]	-1192.452
1100	-1547.657	-4645.019	[-0.000 -0.000]	-6192.676
1200	-1547.657	-4645.019	[-0.000 -0.000]	-6192.676
1300	-1547.657	-4645.019	[-0.000 -0.000]	-6192.676
1400	-1547.657	-4645.019	[-0.000 -0.000]	-6192.676
1500	-7797.937	-23395.859	[0.000 0.000]	-31193.797
1600	-7797.937	-23395.859	[0.000 0.000]	-31193.797
1700	-7797.937	-23395.859	[0.000 0.000]	-31193.797
1800	-39049.337	-117150.060	[0.000 0.000]	-156199.397
1900	-39049.337	-117150.060	[0.000 0.000]	-156199.397
2000	-39049.337	-117150.060	[0.000 0.000]	-156199.397
2100	-39049.337	-117150.060	[0.000 0.000]	-156199.397
2200	-195306.338	-585921.062	[0.002 0.000]	-781227.400
2300	-195306.338	-585921.062	[0.002 0.000]	-781227.400
2400	-195306.338	-585921.062	[0.002 0.000]	-781227.400
2500	-976591.342	-2929776.073	[0.009 0.001]	-3906367.415
2600	-976591.342	-2929776.073	[0.009 0.001]	-3906367.415
2700	-976591.342	-2929776.073	[0.009 0.001]	-3906367.415
2800	-976591.342	-2929776.073	[0.009 0.001]	-3906367.415
2900	-4883016.361	-14649051.131	[0.047 0.007]	-19532067.491
3000	-4883016.361	-14649051.131	[0.047 0.007]	-19532067.491
3100	-4883016.361	-14649051.131	[0.047 0.007]	-19532067.491
3200	-24415141.456	-73245426.416	[0.236 0.033]	-97660567.844
3900	-195321158.304	-585964226.958	[-3.108 0.268]	-781285380.396
4000	-195321164.767	-585964246.348	[4.892 0.268]	-781285399.114
4100	-195321164.767	-585964246.348	[4.892 0.268]	-781285399.114
4200	**-195321164.767**	**-585964246.348**	**[4.892 0.268]**	**-781285399.114**

Table 1 contains 4200 iterations of minimization. Column $\sum_{\mathrm{i}=0}^{\mathrm{n}} \eta \mathrm{i}$ contains the value of the loss on the labeled data, Column $\sum_{\mathrm{i}=0}^{\mathrm{n}} \min(\xi \mathrm{i}, \mathrm{zj})$ contains the value of losses on unlabeled data. The vector W contains the value of the normal vector. This is our maximum distance.

Therefore, our target function has been minimized successfully and contains the following values:

Vector of normal $\boldsymbol{w}$ – [4.892 0.268],

$\boldsymbol{b}$ (offset along the y axis) – 0.0529.

Then we calculate the equation to obtain the classes of our points:

$$\boldsymbol{Y} = X * w^T + b \tag{5}$$

Formula 5 – equation of the line.

Having obtained the value of Y, we impose the following condition, which finalizes our calculations and determines to which class our data belong:

If $x \leq 0$, then class -1,

If $x > 0$, then class 1.

And we get the following result:

```
[ 1. 1. 1. 1. 1. -1. 1. -1. 1. 1. 1. -1. -1. -1. 1. -1. 1. -1.
-1. -1. -1. -1. -1. -1. 1. -1. -1. 1. 1. 1. 1. 1. 1. 1. 1. -1.
1. -1. -1. -1. 1. 1. 1. 1. -1. 1. 1. 1. -1. -1. 1. -1. 1. 1.
-1. 1. -1. 1. 1. -1. 1. -1. -1. 1. -1. 1. 1. 1. -1. 1. -1. 1.
1. -1. 1. -1. 1. 1. -1. -1. 1. 1. 1. -1. 1. -1. -1. 1. -1. -1.
1. 1. 1. 1. -1. -1. 1. -1. 1. 1. 1. -1. 1. 1. -1. -1. -1. 1.
-1. 1. 1. 1. -1. 1. 1. -1. -1. -1. -1. 1. -1. 1. -1. -1. 1. 1.
-1. -1. -1. 1. -1. 1. -1. 1. 1. 1. 1. 1. -1. 1. 1. 1. 1. -1.
-1. -1. 1. -1. 1. 1.]
```

This array contains data after classification. Two classes - class 1 (designation 1) and class 2 (designation -1).

Visualize the result: Graph showing labeled and unlabeled data (See Figure 10):

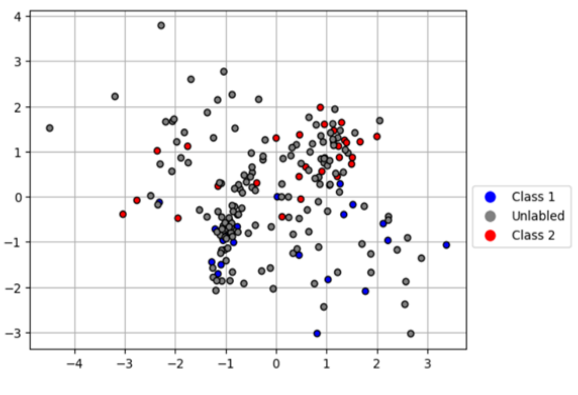

Figure 10. Visualization of the dataset after adding unlabeled data.

Figure 10 shows class 1 in blue, class 2 in yellow, and unlabeled data in gray.

Schedule after classification (See Figure 11):

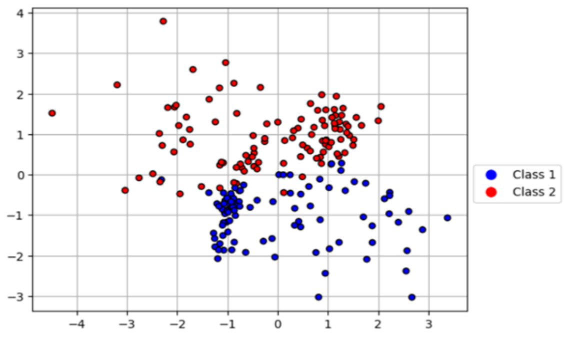

Figure 11. Dataset visualization after classification.

As you can see from Figure 11, the classification was quite successful.

Almost all data are separated, so the hyperline is successful, but contains points that have the wrong class. This applies to areas of low density.

TSVM approach calculations. Visualization of input data and addition of unlabeled data is already described in the previous section, so let's start with the initialization of the initial values of the variables w (normal vector), b (offset along the y-axis), $\boldsymbol{y^u}$(variable for the second constraint condition, this condition must occur as the first, but for unlabeled points) and weak variables $\boldsymbol{\eta i}$, $\boldsymbol{\xi i}$.

For values of w and b the sample will be [-0.1, 0.1], for $\boldsymbol{\eta i}$, $\boldsymbol{\xi i}$ – [0.0 0.1], and for $\boldsymbol{y^u}$ – [-1.0, 0.1].

$\boldsymbol{w}$ – [-0.01975875 0.02131162] size= 2,

$\boldsymbol{\eta i}$ – [0.04901986 0.02258875 0.01560311 0.04587382 0.0352636 0.00871087
0.08424851 0.0857313 0.0733342 0.09528408 0.05455903 0.06449431
0.07807958 0.05834814 0.07014477 0.0219474 0.06762864 0.02508063
0.05901452 0.04758597 0.01772247 0.07832201 0.07269677 0.09613403
0.08828265 0.09579222 0.00191197 0.0985599 0.08494134 0.00936994
...
0.02368684 0.06946833] - size = (Marked – notMarked) = 50,

$\boldsymbol{\xi i}$ – [0.09799687 0.08343227 0.00994421 0.09511786 0.03769766 0.04774885
0.04659362 0.09062985 0.03513581 0.05616333 0.02952544 0.0812517
0.0750658 0.02050575 0.089315 0.06914368 0.01608069 0.00452057
0.069853 0.03098653 0.00093538 0.086660854 0.09679364 0.03014508
...
0.0983856 0.09866195 0.08204016 0.09872704 0.06635472 0.07370298]
– size = notMarked = 150,

$\boldsymbol{y^u}$ –
[0.04063871 0.24102629 0.64432042 -0.71708787 -0.07513165 -0.98990635
-0.2922075 0.00560538 -0.14231635 0.18206645 0.55786814 0.89377577
0.54712419 0.70159914 0.77993996 -0.46725851 -0.84731204 0.37656981
-0.03340037 -0.21165283 0.71084185 -0.40639437 0.44571452 0.01685231
...
0.7710943 0.29409488 -0.08909666 -0.31046288 -0.97264293 0.0147352] - size= notMarked = 150,

$\boldsymbol{b}$ – [0.04816553] – size = 1.

Next you need to define the constants CL and CU, thanks to which we can determine which data should be relied on more, if CL is a large value and SU is a small one, it means that we trust more labeled data, if on the contrary, we look more at the structure of unlabeled data.

So, let's set CL = 2.0 and SU = 0.1, which will mean that we will rely more on the labeled data.

Basically, the idea is to preserve the objective function with two sets of lysis variables (The lysis variable is a variable that is added to the constraint to convert it from inequality to equality) - the first for labeled samples and the second for unlabeled samples:

$$\min\left[||w||+C_L\sum_{i=1}^{N}\eta_i \ + \ C_U\sum_{i=N+1}^{N+M}\xi_j\right]. \tag{6}$$

As in the previous algorithm, we assume that we have N labeled samples and M unlabeled, and therefore the conditions are as follows:

$$\begin{gathered} y_i\left(w^Tx_i+b\right)\geq 1-\eta_i \ , \ \eta_i\geq 0 \ \forall \ i\in(1,N) \\ y_j^{(u)}(w^Tx_j+b)\geq 1-\xi_j \ , \ \xi_j\geq 0 \ \forall \ j\in(N+1,N+M) \\ y_j^{(u)}\in\{-1,1\} \end{gathered} \tag{7}$$

The first limitation is the classic SVM, and it only works on labeled samples.

The second uses a variable y_j^u with corresponding weak variables ξ_j, to impose a similar condition on the labeled samples, while the third is needed to limit the labels to -1 and 1.

We minimize the function with the above restrictions:

Table 2. Minimize the target TSVM function.

Step	**($\sum_i^N \eta i$)*CL**	**($\sum_i^N \xi i$)*CU**	**Vector W**	**Target function**
0	2.501	8.116	[-0.097 -0.092]	5.822
100	2.501	8.116	[-0.097 -0.092]	5.822
200	2.501	8.116	[-0.097 -0.092]	5.822
300	2.501	8.116	[-0.097 -0.092]	5.822
400	-97.499	-6.884	[-0.000 0.000]	-195.687
500	-97.499	-6.884	[-0.000 0.000]	-195.687
600	-97.499	-6.884	[-0.000 0.000]	-195.687
700	-97.499	-6.884	[-0.000 0.000]	-195.687
800	-597.677	-81.911	[0.000 0.000]	-1203.545
900	-597.677	-81.911	[0.000 0.000]	-1203.545
1000	-597.677	-81.911	[0.000 0.000]	-1203.545
1100	-3098.566	-457.050	[0.000 0.000]	-6242.837
1200	-3098.566	-457.050	[0.000 0.000]	-6242.837
1300	-3098.566	-457.050	[0.000 0.000]	-6242.837
1400	-3098.566	-457.050	[0.000 0.000]	-6242.837

Step	**($\sum_i^N \eta i$)*CL**	**($\sum_i^N \xi i$)*CU**	**Vector W**	**Target function**
1500	-15603.040	-2332.772	[0.000 0.000]	-31439.356
1600	-15603.040	-2332.772	[0.000 0.000]	-31439.356
1700	-15603.040	-2332.772	[0.000 0.000]	-31439.356
1800	-78125.688	-11711.473	[-0.000 0.000]	-157422.523
1900	-78125.688	-11711.473	[-0.000 0.000]	-157422.523
2000	-78125.688	-11711.473	[-0.000 0.000]	-157422.523
2100	-78125.688	-11711.473	[-0.000 0.000]	-157422.523
2200	-390772.967	-58611.180	[-0.001 -0.008]	-787407.052
2300	-390772.967	-58611.180	[-0.001 -0.008]	-787407.052
2400	-390772.967	-58611.180	[-0.001 -0.008]	-787407.052
2500	-1953100.666	-292967.544	[-0.005 -0.002]	-3935498.087
2600	-1953100.666	-292967.544	[-0.005 -0.002]	-3935498.087
2700	-1953100.666	-292967.544	[-0.005 -0.002]	-3935498.087
2800	-1953100.666	-292967.544	[-0.005 -0.002]	-3935498.087
2900	-9764739.162	-1464749.366	[-0.024 -0.011]	-19675953.261
3000	-9764739.162	-1464749.366	[-0.024 -0.011]	-19675953.261
3100	-9764739.162	-1464749.366	[-0.024 -0.011]	-19675953.261
3200	-49277493.151	-7391962.039	[-0.120 -0.054]	-99294182.496
3300	-49277493.151	-7391962.039	[-0.120 -0.054]	-99294182.496
...	...	...	...	...
4900	-10834827210.337	-1625327305.35	[-26.31 -11.82]	-21832186735

Table 2 contains 4900 iterations of minimization. Column $(\sum_i^N \eta i) * CL$ contains the value of the loss on the tagged data, taking into account the constant CL, by which we determine how much we will rely on the tagged data. Column $(\sum_i^N \xi i) * CU$ contains the value of losses on unlabeled data, which also takes into account the constant CU, contains the value of losses on unlabeled data, which also takes into account the constant CU, by which we determine how much we will rely more on the structure of the dataset than on the labeled data. The vector W contains the value of the normal vector. This is our maximum distance.

Therefore, our target function has been minimized successfully and contains the following values:

Vector of normal **w** – [-26.317 -11.823],

b (offset along the y axis) – 0.04816553,

Then we calculate the equation to obtain the classes of our points:

$$\boldsymbol{Y} = X * w^T + b \tag{8}$$

Having obtained the value of Y, we impose the following condition, which finalizes our calculations and determines to which class our data belong:

If $x \leq 0$, then class -1,

If $x > 0$, then class 1.

And we get the following result:

[-1. -1. -1. 1. -1. 1. -1. 1. -1. -1. -1. 1. 1. 1. -1. 1. -1. 1.
1. 1. -1. 1. 1. 1. -1. 1. 1. -1. -1. 1. -1. -1. -1. -1. -1. 1.

```
-1. 1. 1. 1. -1. -1. -1. -1. 1. -1. -1. -1. 1. 1. -1. 1. -1. -1.
1. -1. 1. -1. 1. -1. -1. 1. 1. -1. 1. -1. -1. -1. 1. -1. -1. -1.
-1. 1. 1. 1. -1. -1. 1. -1. -1. 1. -1. 1. -1. 1. 1. -1. 1. 1.
-1. -1. -1. -1. 1. 1. -1. 1. -1. -1. -1. 1. 1. -1. 1. 1. 1. -1.
1. -1. -1. -1. 1. -1. -1. 1. 1. 1. 1. -1. 1. -1. 1. 1. -1. -1.
1. 1. 1. -1. 1. -1. 1. -1. -1. -1. -1. -1. 1. -1. -1. -1. -1. 1.
1. -1. -1. 1. -1. -1.]
```

This array contains data after classification. Two classes - class 1 (designation 1) and class 2 (designation -1).

Graph showing labeled and unlabeled data (Figure 12):

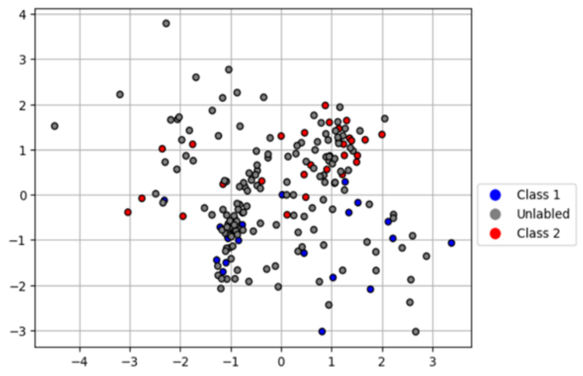

Figure 12. Visualization of the dataset after adding unlabeled data.

Class 1 is shown in blue, class 2 - in yellow, unlabeled data are marked in gray. Schedule after classification:

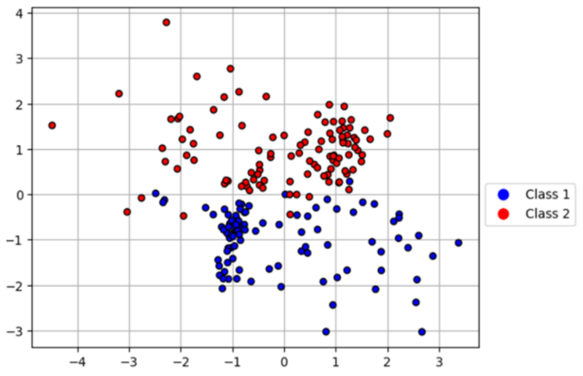

Figure 13. Dataset visualization after classification.

Analyzing Figure 13, we see that TSVM also showed a pretty good result, dividing the data into two classes. Further from the boundary of the section, all points are classified perfectly, but it is on the hyperline in areas of low density that you can see points with erroneous classification. Given the cost, we can assume that this method has coped with the task.

SVM approach calculations. For comparison, use the semi-supervised SVM approach and compare its results (which are quite expensive to obtain) with the results of the semi-supervised approach.

We use the dataset that we used in the two previous methods.

Number of marked labels - 200,

Number of unlabeled labels - 150.

Let us set the constant C = 1.

We use a trained SVM model with a linear function, and get the following results:

```
[-1 -1 1 1 -1 1 1 1 -1 -1 1 -1 1 1 -1 1 1 1 1 1 -1 1 1 -1
-1 -1 1 -1 1 1 1 -1 -1 1 1 -1 -1 1 -1 1 1 1 -1 -1 1 1 1 -1
-1 1 -1 1 -1 -1 1 1 -1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 1 -1 -1
1 -1 1 1 1 1 1 -1 1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 1 1 1
1 -1 1 -1 -1 1 1 1 -1 1 1 -1 1 1 -1 -1 -1 -1 1 1 -1 1 1 1
1 1 -1 -1 -1 -1 1 1 1 -1 -1 1 -1 1 -1 -1 1 -1 -1 1 -1 -1 1 1
-1 -1 -1 1 1 -1]
```

This array contains data after classification. Two classes - class 1 (designation 1) and class 2 (designation -1).

Graph showing labeled and unlabeled data (Figure 14):

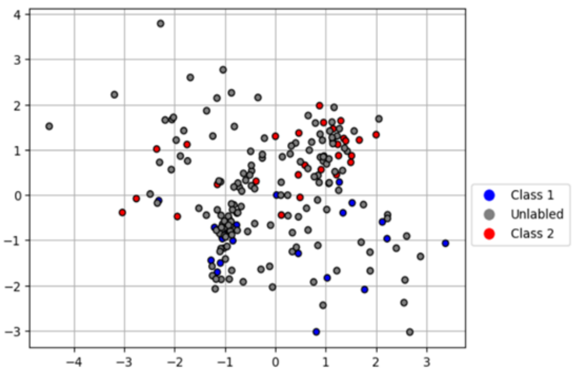

Figure 14. Visualization of the dataset after adding unlabeled data.

Class 1 is shown in blue, class 2 - in yellow, unlabeled data are marked in gray.

Schedule after classification:

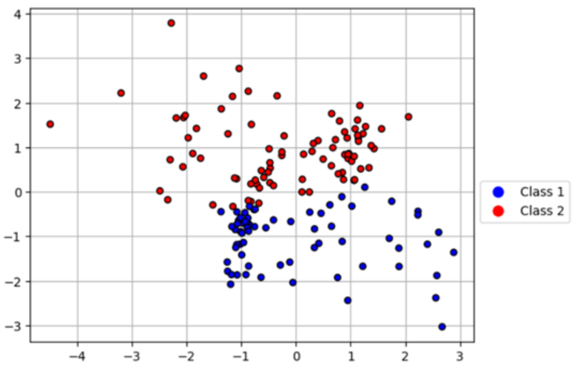

Figure 15. Dataset visualization after classification.

As we can see in Figure 15, the classification was better than in the two previous methods. Almost all points are classified correctly. The hyperline was successful because the data was correctly separated both at the dividing line and in the high and low density zones. The protégé is also important for how costly these results are. Semi-supervised approaches can also compete with controlled, because they show close to the controlled approach.

3. Conclusions and prospects for further research. Modern semi-supervised approaches require serious assumptions and work poorly if assumptions are violated (e.g., clustering assumptions). In some cases, they may perform worse than a controlled classifier trained only on labelled instances. In addition, the vast majority need memory. However, my goal was to investigate whether semi-controlled approaches can compete with controlled ones or outperform them.

Having gathered all the necessary material, I learned about S3VM and TSVM.

Applying these approaches to my dataset, I determined that the TSVM algorithm more accurately classified the data at the reference limit, but the result of S3VM was not too bad.

The next step was to compare these algorithms with a method supervised approach, such as SVM. This algorithm showed the best results, but S3VM and TSVM have very close results to SVM. And as mentioned above, you need to choose between accuracy and cost.

You can also consider the case where the TSVM method will work better than supervised SVM. This is the case when we have a minimal amount of labelled data and a large number of unlabeled; consider this case:

Suppose we have 100 records, 90 of which are not macros.

Figures 16-18 show two graphs each. The first is the data for classification; blue is class 0, orange is class 1, green is unlabeled. The second is the data after classification. The red dots are divided into unlabeled squares of class 0 and labelled circles of class 0. The blue dots are divided into unlabeled triangles (inverted down) of class 1 and labelled triangles (upwards) of class 1.

SVM will work as follows:

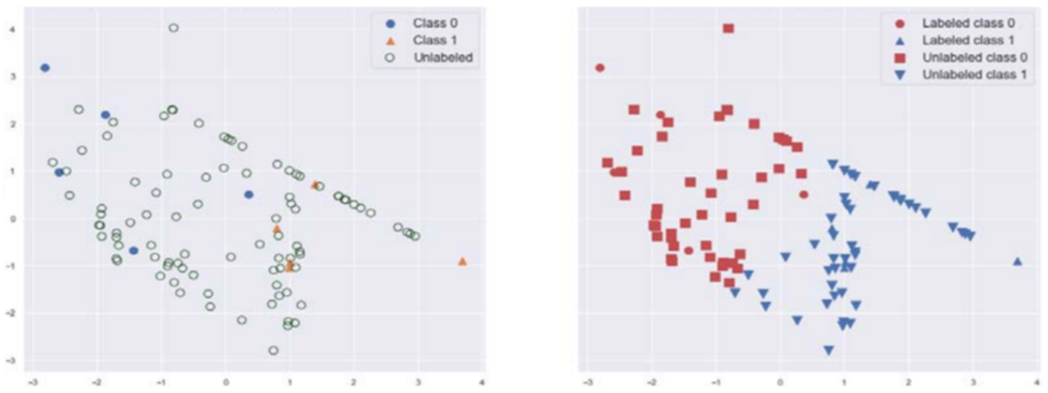

Figure 16. Comparison of data before and after classification (data before category on the left, after - on the right)

In Figure 16, we see a good result, but this line is not entirely stable, so consider how the TSVM will work.

We get a similar result by putting the following parameters in TSVM: CL = 1.0 and SU = 10.0 (the set parameters mean that we rely more on the structure of the dataset than on the labelled data).

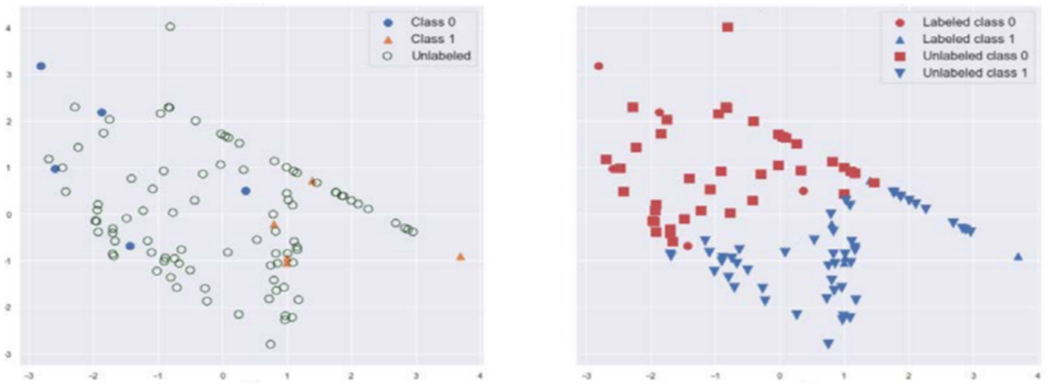

Figure 17. Comparison of data before classification and after (left data before classification, right - after).

However, by inverting the parameters, we get the following result: CL= 10.0 and CU= 0.1 (the parameters set mean that we rely more on the tagged data than on the dataset structure).

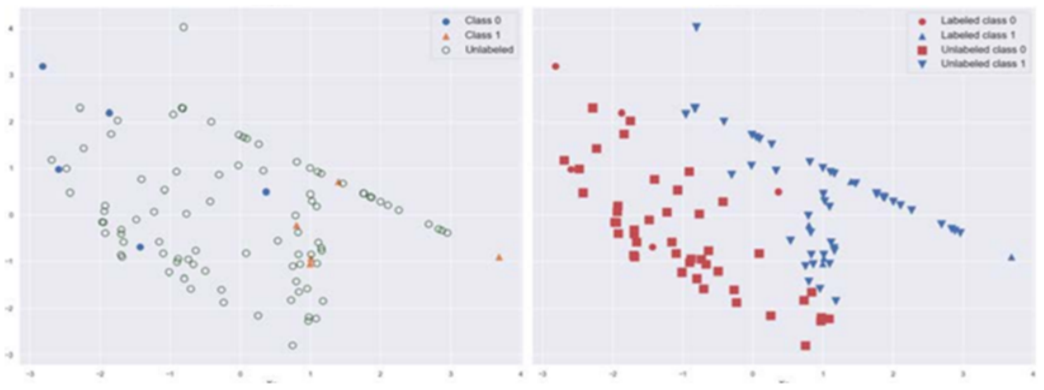

Figure 18. Comparison of data before and after classification (before classification on the left, after - on the right).

In this case, in Figure 18, labelled data can use more flexible boundaries, while unlabeled data is only allowed to be rigid.

We see that the hyperline runs entirely differently and better classifies our data at the dividing line and in the high and low-density zones. Also, this result is less expensive because most of the unlabeled data is used.

As a result, we can assume that semi-supervised approaches solve the classification problem no worse and sometimes better than supervised approaches.

References

1. Estivill-Castro, V., & Lee, I. (2000). Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: *9th Intern. Symp. on spatial data handling,* Beijing, China. 26-41.
2. Boehm, C., Kailing, K., Kriegel, H., & Kroeger, P. (2004). Density connected clustering with local subspace preferences, *IEEE Computer Society, Proc. of the 4th IEEE Intern. conf. on data mining,* Los Alamitos. 27-34.
3. Boyko, N., & Shakhovska, K. (2018). Information system of catering selection by using cluster-

ing analysis, in: *2018 IEEE Ukraine Student, Young Professional and Women in Engineering Congress (UKRSYW).* Kyiv, Ukraine. 7-13.

4. Harel, D., & Koren, Y. (2001). Clustering spatial data using random walks, in: *Proc. of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining.* San Francisco, California. 281-286.
5. Tung, A.K., Hou, J., & Han, J. (2001). Spatial clustering in the presence of obstacles, in: *The 17th Intern. conf. on data engineering (ICDE'01),* Heidelberg. 359-367.
6. Boyko, N., Bronetskyi, A., & Shakhovska, N. (2019). Application of Artificial Intelligence Algorithms for Image Processing, in: CEUR. Workshop Proceedings of the 8th International Conference on "Mathematics. *Information Technologies. Education", MoMLeT&DS-2019, Vol-2386 urn: nbn: de: 0074-2386-1,* Shatsk, Ukraine, June 2-4, 2019, 194-211.
7. Agrawal, R., Gehrke, J., Gunopulos, D., & Raghava, P. (2005). Automatic sub-space clustering of high dimensional data, *Data mining knowledge discovery, 11(1),* 5-33.
8. Ankerst, M., Ester, M., & Kriegel, H. P. (2000). Towards an effective cooperation of the user and the computer for classification, in: *Proc. of the 6th ACM SIGKDD Intern. conf. on knowledge discovery and data mining.* Boston, Massachusetts, USA. 179-188.
9. Zhang, C., & Murayama, Y. (2000). Testing local spatial autocorrelation using. *Intern. J. of Geogr. Inform. Science, 14,* 681-692.
10. Estivill-Castro, V., & Lee, I. (2000). Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: *9th Intern. Symp. on spatial data handling.* Beijing, China. 26-41.
11. Guo, D., Peuquet, D. J., & Gahegan, M. (2003). ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata, *Geoinformatica, 3(7)*, 229-253.
12. Boyko, N., & Basystiuk, O. (2018). Comparison Of Machine Learning Libraries Performance Used For Machine Translation Based On Recurrent Neural Networks, in: *2018 IEEE Ukraine Student, Young Professional and Women in Engineering Congress (UKRSYW).* Kyiv, Ukraine. 78-82.
13. Aggarwal, C., & Yu, P. (2000). Finding generalized projected clusters in high dimensional spaces, in: *ACM SIGMOD Intern. conf. on management of data.* 70-81.
14. Thanki, R., & Borra, S. (2019). Application of Machine Learning Algorithms for Classification and Security of Diagnostic Images, *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging,* 273-292.
15. Peuquet, D. J. (2000). Representations of space and time. N. Y.: Guilford Press.
16. Procopiuc, C. M., Jones, M., Agarwal, P. K., & Murali, T. M. (2002). A Monte Carlo algorithm for fast projective clustering, in: *Intern. conf. on management of data, ACM SIGMOD,* Madison, Wisconsin, USA. 418-427.
17. Chitra, K., & Maheswari, D. (2017). A Comparative Study of Various Clustering Algorithms in Data Mining. *International Journal of Computer Science and Mobile Computing, 6(8)*, 109-115.

Бойко Н. Аналіз парадигми Semi-supervised learning для класифікації мультимодальних даних.

У роботі розглядаються алгоритми машинного навчання. Увага зосереджена на напівконтрольному навчанні, яке здається балансом між точністю навчання з учителем та витратами методів навчання без учителя. Розглядаються приклади ретельного опрацювання мічених наборів даних, для яких навчання під наглядом може бути дуже ефективним. Порівнюються підходи semi-supervised та supervised та проаналізована ефективність кожного. В роботі розглядаються підходи S3VM та TSVM. Метою роботи було дослідити чи можуть напівконтрольовані підходи конкурувати з контрольованими або навіть їх перевершити. Описується застосування даних підходів до запропонованого датасету для визначення більш точної класифікації даних, а саме на опорній межі.

Ключові слова: навчання під наглядом, метод опорних векторів, трандуктивний метод опорних векторів, метод опорних векторів для часткового навчання, прикладний програмний інтерфейс.

Список використаної літератури

1. Estivill-Castro V., Lee I. Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: *9th Intern. Symp. on spatial data handling,* Beijing, China, 2000. P. 26–41.
2. Boehm C., Kailing K., Kriegel H., Kroeger P. Density connected clustering with local subspace preferences, *IEEE Computer Society, Proc. of the 4th IEEE Intern. conf. on data mining,* Los Alamitos, 2004. P. 27–34.
3. Boyko N., Shakhovska K. Information system of catering selection by using clustering analysis, in: *2018 IEEE Ukraine Student, Young Professional and Women in Engineering Congress (UKRSYW).* Kyiv, Ukraine, 2018. P. 7-13.
4. Harel D., Koren Y. Clustering spatial data using random walks, in: *Proc. of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining.* San Francisco, California, 2001. P. 281–286.
5. Tung A.K., Hou J., Han J. Spatial clustering in the presence of obstacles, in: *The 17th Intern. conf. on data engineering (ICDE'01),* Heidelberg, 2001. P. 359–367.
6. Boyko N., Bronetskyi A., Shakhovska N. Application of Artificial Intelligence Algorithms for Image Processing, in: CEUR. Workshop Proceedings of the 8th International Conference on "Mathematics. *Information Technologies. Education",* MoMLeT&DS-2019. Vol. 2386 urn: nbn: de: 0074-2386-1, Shatsk, Ukraine, June 2-4, 2019. P. 194-211.
7. Agrawal R., Gehrke J., Gunopulos D., Raghava P. Automatic sub-space clustering of high dimensional data. *Data mining knowledge discovery,* 2005. Vol. 11, 1. P. 5–33.
8. Ankerst M., Ester M., Kriegel H.-P. Towards an effective cooperation of the user and the computer for classification, in: *Proc. of the 6th ACM SIGKDD Intern. conf. on knowledge discovery and data mining,* Boston, Massachusetts, USA, 2000. P. 179–188.
9. Zhang C., Murayama Y. Testing local spatial autocorrelation using. *Intern. J. of Geogr. Inform. Science.* 2000. Vol. 14. P. 681–692.
10. Estivill-Castro V., Lee I. Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: *9th Intern. Symp. on spatial data handling,* Beijing, China, 2000. P. 26–41.
11. Guo D., Peuquet D. J., Gahegan M. ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata, *Geoinformatica,* 2003. Vol. 3, No. 7. P. 229–253.
12. Boyko N., B̃asystiuk O. Comparison Of Machine Learning Libraries Performance Used For Machine Translation Based On Recurrent Neural Networks, in: *2018 IEEE Ukraine Student, Young Professional and Women in Engineering Congress (UKRSYW),* Kyiv, Ukraine, 2018. P. 78-82.
13. Aggarwal C., Yu P. Finding generalized projected clusters in high dimensional spaces, in: *ACM SIGMOD Intern. conf. on management of data,* 2000. P. 70–81.
14. Thanki R., Borra S. Application of Machine Learning Algorithms for Classification and Security of Diagnostic Images, *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging,* 2019. P. 273-292.
15. Peuquet D. J. Representations of space and time. N. Y.: Guilford Press, 2000.
16. Procopiuc C.M., Jones M., Agarwal P. K., Murali T. M. A Monte Carlo algorithm for fast projective clustering, in: *Intern. conf. on management of data, ACM SIGMOD,* Madison, Wisconsin, USA, 2002. P. 418–427.
17. Chitra K., Maheswari D. A Comparative Study of Various Clustering Algorithms in Data Mining. *International Journal of Computer Science and Mobile Computing,* Vol.6 Issue.8, August 2017. P. 109–115.

Recived 25.10.2021