Margin-Based Semi-supervised Learning Using Apollonius Circle

Mona Emadi, Jafar Tanha

Margin-based Semi-supervised Learning Using Apollonius circle

Mona Emadi and Jafar Tanha

[1] Azad University, Borujerd, Iran
[2] Electrical and Computer Engineering Department,
University of Tabriz, Tabiz, Iran
emadi.mona@pnu.ac.ir
tanha@utabrizu.ac.ir

Abstract. In this paper, we focus on the classification problem to semi-supervised learning. Semi-supervised learning is a learning task from both labeled and unlabeled data examples. We propose a novel semi-supervised learning algorithm using a self-training framework and support vector machine. Self-training is one of the wrapper-based semi-supervised algorithms in which the base classifier assigns labels to unlabeled data at each iteration and the classifier re-train on a larger training set at the next training step. However, the performance of this algorithm strongly depends on the selected newly-labeled examples. In this paper, a novel self-training algorithm is proposed, which improves the learning performance using the idea of the Apollonius circle to find neighborhood examples. The proposed algorithm exploits a geometric structure to optimize the self-training process. The experimental results demonstrate that the proposed algorithm can effectively improve the performance of the constructed classification model.

Keywords: Apollonius circle · Semi-supervised classification · Self-training · Support vector machine.

1 Introduction

Typically, supervised learning methods are useful when there is enough labeled data, but in many real word tasks, unlabeled data is available. Furthermore, in practice, labeling is an expensive and time consuming task, because it needs human experience and efforts [14]. Therefore, finding an approach which can employ both labeled and unlabeled data to construct a proper model is crucial. Such a learning approach is named semi-supervised learning. In semi-supervised learning algorithms, we use labeled data as well as unlabeled data. The main goal of semi-supervised learning is to employ unlabeled instances and combine the information in the unlabeled data with the explicit classification information of labeled data to improve the classification performance. The main challenge in semi-supervised learning is how to extract knowledge from the unlabeled data [4, 21, 30]. Several different algorithms for semi-supervised learning have been

introduced, such as the Expectation Maximization (EM) based algorithms [11, 13, 15], self-training [7, 9, 23], co-training [19, 27], Transduction Support Vector Machine (TSVM) [1, 12, 26], Semi-Supervised SVM (S3VM) [3], graph-based methods [2, 28], and boosting based semi-supervised learning methods [5, 16, 24].

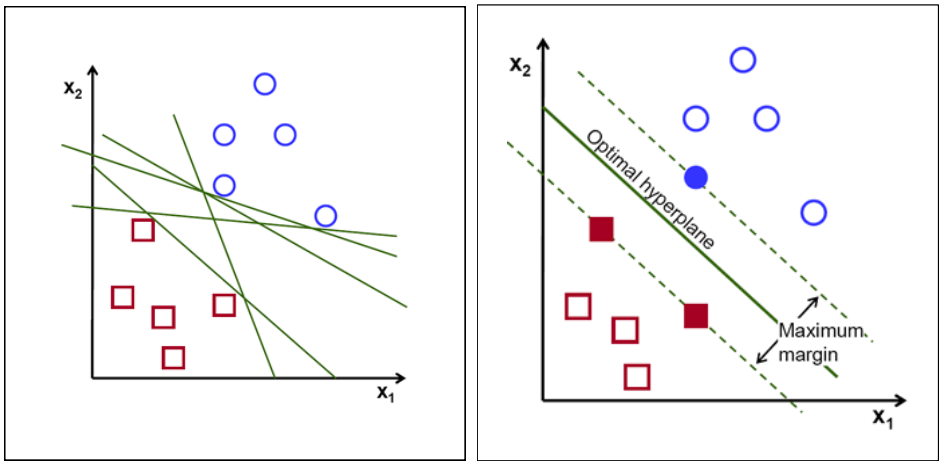
Most of the semi-supervised algorithms follow two main approaches, extension of a specific base learner to learn from labeled and unlabeled examples and using a framework to learn from both labeled and unlabeled data regardless of the used base learner. Examples of the first approach include S3VM, TSVM, and LapSVM. Wrapper-based and Boosting-based methods follow the second approach, like Co-training, SemiBoost [16], MSAB [24], and MSSBoost [22]. In this article, we focus on both approaches and propose a novel semi-supervised approach based on the SVM base learner.

Support vector machine is proposed by Cortes and Vapnik [6] and is one of the promising base learners in many practical domains, such as object detection, document and web-page categorization. It is a supervised learning method based on margin as well as statistical learning theory [8]. The purpose of the support vector machine algorithm is to find an optimal hyperplane in an N-dimensional space in order to classify the data points. As shown in Fig 1, there are many possible hyperplanes that could be selected. The optimal classification hyperplane of SVM needs not only segregating the data points correctly, but also maximizing the margin [8]. Maximizing the margin leads to a strong classification model. The standard form of SVM is only used for supervised learning tasks. This base learner can not directly handle the semi-supervised classification tasks. There are several extensions to SVM, like S3VM and TSVM. These methods use the unlabeled data to regularize the decision boundary. These methods mainly extend the SVM base classifier to semi-supervised learning, which are computationally expensive [16]. Therefore, these approaches are suitable only for small datasets.

More recently, a semi-supervised self-training has been used to handle the semi-supervised classification tasks [7, 23, 29]. Self-training is a wrapper-based algorithm that repeatedly uses a supervised learning method. It starts to train on labeled data only. At each step, a set of unlabeled points is labeled according to the current decision function; then the supervised method is retrained using its own predictions as additional labeled points [23]. However, the performance of this method depends on correctly predicting the labels of unlabeled data. This is important, because the selection of incorrect predictions will propagate to produce further classification errors. In general, there are two main challenges. The first is to select the appropriate candidate set of unlabeled examples to label at each iteration of the training procedure. The second is to correctly predict labels to unlabeled examples. To handle these issues, the recent studies tend to find a set of high-confidence predictions at each iteration [7,23,29]. These selected examples are typically far away from the decision boundary. Hence, this type of algorithm cannot effectively exploit information from the unlabeled data and the final decision boundary will be very close to that of the initial classifier [22].

In [29], subsets of unlabeled data are selected which are far away from the current decision boundary. Although these data have a high confident rate, they

are not informative. Indeed they have little effect on the position of the hyperplane. Furthermore, adding all the unlabeled points is time consuming and may not change the decision boundary. In this paper, we propose a novel approach which tends to find a set of informative unlabeled examples at each iteration of the training procedure.



(a) possible hyperplane (b) optimal hyperplane

Fig. 1: Support vector machine

In our proposed approach, we select a set of unlabeled data that is potentially close to the hyperplane. In order to assign correct pseudo-label, we employ the Apollonius circle idea regarding the neighborhood examples. A Neighborhood Construction algorithm based on Apollonius Circle is used to find the neighborhood of the data points [18]. The main contributions of this paper include:

1. By selecting unlabeled data close to the decision boundary, we improve the margin.
2. We define a new method to measure the similarity between the labeled and unlabeled data points based on a geometrical structure.

We use the UCI benchmark datasets [10] to evaluate the proposed algorithm. Our experiments on a number of UCI datasets show that the proposed algorithm outperforms the state-of-the-art wrapper-based methods to semi-supervised learning.

The rest of the paper is organized as follows: Section 2 reviews the literature related to the proposed algorithm. Section 3, the proposed algorithm and framework are discussed. Section 4, presents the experiment results on the datasets and compare to the-state-of-the-art algorithms. Section 5, we conclude this paper.

2 Review of concepts related to the density peaks and the Apollonius circle

In this section, the concepts of the Apollonius circle are discussed. We propose the neighborhood structure of the Apollonius circle to label the unlabeled data.

Apollonius circle is one of the most famous problems in geometry [18]. The goal is to find accurate neighborhoods. In a set of points, there is no information about the relationship between the points and some databases may not even contain topological information. It is more accurate than the Gabriel graph [25] and neighborhood graph in the neighborhood construction. It evaluates important neighborhood entirely. In this method, the first step is to find high density points and the second step is to build neighborhood groups with the Apollonius circle. The third step is analyzing points outside the radius of the Apollonian circles or points within the region.

2.1 Finding high density points

Rodriguez and Laio [20] presented the algorithm to find high density points (DPC). The high density points are found by this method and then stored in an array.

The points are shown by the vector $\mathbf{M} = (M_{11}, M_{22}, ..., M_{mn})$ where m and n are number of attributes and number of points respectively, also N_{mi} shows k nearest neighbors of M_i. $d(M_i, M_J)$ is the Euclidean distance between M_i and M_j. The percent of neighborhood is shown by p. The number of neighbors is obtained by the formula ($r = p \times n$). The local density ρ_i is defined as:

$$\rho_i = \exp\left(-\left(\frac{1}{r}\sum_{M_j \in N(M_i)} d(M_i, M_J)^2\right)\right). \tag{1}$$

$$d(M_i, M_J) = \|M_i, M_J\|. \tag{2}$$

where δ_i is the minimum distance between M_i and any other sample with higher density than p_i, which is define as below:

$$\delta_i = \begin{cases} \min_{\rho_i < \rho_j} \{d(M_i, M_j)\}, & \text{if} \quad \exists j \quad \rho_i < \rho_j \\ \max_j \{d(M_i, M_j)\}, & otherwise \end{cases} \tag{3}$$

Peaks (high density points) are obtained using the score function. The points that have the highest score are considered as peak point.

$$score(M_i) = \delta_i \times \rho_i \tag{4}$$

In this article, number of peaks are selected based on the number of classes. Peaks are selected from the labeled set. We assign the label of peaks to the unlabeled data based on neighborhood radius of peaks. Neighboring groups are found by the Apollonius circle.

2.2 Neighborhood groups with the Apollonius circle

The Apollonius circle is the geometric location of the points on the Euclidean plane which have a specified ratio of distances to two fixed points A and B, this ratio is called K [17]. Apollonius circle can be seen in Fig. 2.

$$K = {}^{d_1}/_{d_2}. \tag{5}$$

The Apollonius circle based on A and B is defiend as:

$$C_{AB} = \begin{cases} C_A & \text{if} \quad K < 1 \\ C_B & \text{if} \quad K > 1 \\ C_{inf} & \text{if} \quad K \equiv 1 \end{cases} \tag{6}$$

After finding the high density points, we sort these points. The peak points are indicated by $P = (P_1, P_2, ..., P_m)$ which are arranged in pairs (P_t, P_{t+1}), $t \in \{1, 2, ..., m-1\}$, the data points are denoted by $M = \{M_i | i \in \{1, 2, ..., n-m\}, M_i \notin P\}$. In the next step, data points far from the peak points are determined by the formula 7. Finally the distance of the furthest point from the peak points is calculated by the formula 8.

$$\begin{aligned} Fd_t = \max \{ & d(P_t, M_i) \, | M_i \in M \; and \; d(P_t, M_i) < d(P_t, P_{t+1}) \\ & and \; \min_{l=1}^{m} d(P_t, M_i) \;\; s.t. \; t \neq 1 \Big\}. \end{aligned} \tag{7}$$

$$FP_t = \{ M_i | d(P_t, M_i) = Fd_t \}. \tag{8}$$

The points between the peak point and the far point are inside the Apollonius circle, circle [18].

The above concepts are used to label the unlabeled samples confidently. In our proposed algorithm, label of the peak points are assigned to the unlabeled example which are inside the Apollonius circle. The steps are shown in Fig. 3.

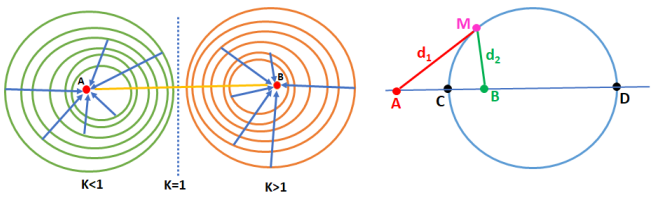


Fig. 2: Apollonius circles C_{AB} and C_B

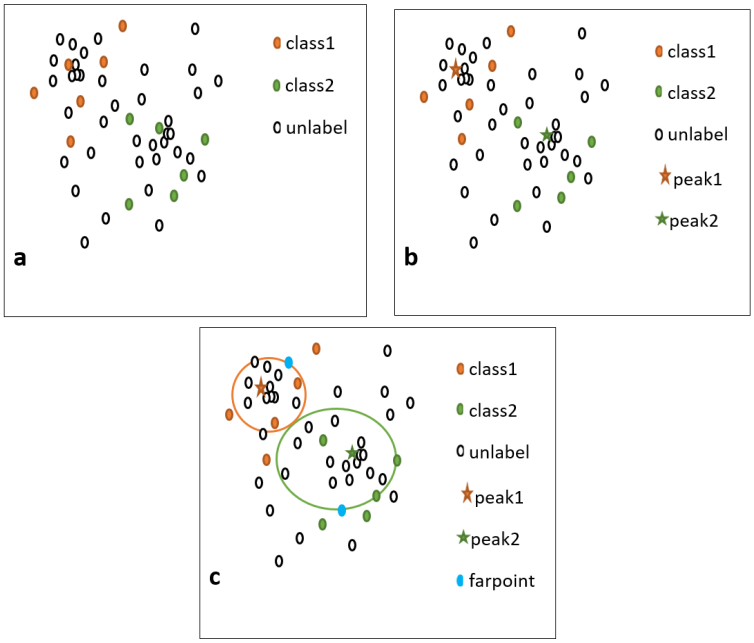


Fig. 3: steps of making neighborhood groups with the Apollonius circle:(a) display the data distribution, (b) detecting the peak that selected from the labeled data and located in the high density region,(c) finding the far points and draw the apollonius circle

3 Our proposed algorithm

The proposed framework and algorithm is described in this section.The classifier works better with more labeled data, unlabeled data is much more than labeled data in semi supervised algorithms. If unlabeled data is labeled correctly and added to the labeled set, performance of the classifier improves.

Our proposed algorithm is a semi-supervised self-training algorithm. Fig. 4 shows the steps of the proposed framework. In the first step, we find δ and ρ for all training data set (labeled and unlabeled) and then the high density peaks are found in the labeled data set. The number of peaks is the number of classes. Then, for each peak we find corresponding far point. Step 2 consists of two parts. One section is about selecting a set of unlabeled data which is the candidate for labeling. The distance of all unlabeled data are calculated from decision boundary. Next unlabeled data are selected for labeling that are closer to the decision boundary. Threshold (distance of decision boundary) is considered the mean distance of all unlabeled data from decision boundary.

Another part of this step is how to label unlabeled data. Our proposed method predicts the label of unlabeled data by high density peak and Apollonius circle concepts. The label of $peak_i$ is assigned to the unlabeled data that

are inside the Apollonius circle corresponding $peak_i$ and in parallel the same unlabeled subset is given to SVM to label them. The agreement based on the classifier predictions and Apollonius circle is used to select a newly-labeled set The labeled set is updated and is used to retrain the classifier. The process is repeated until it reaches the stopping condition. Fig. 4 represents the overview of proposed algorithm.

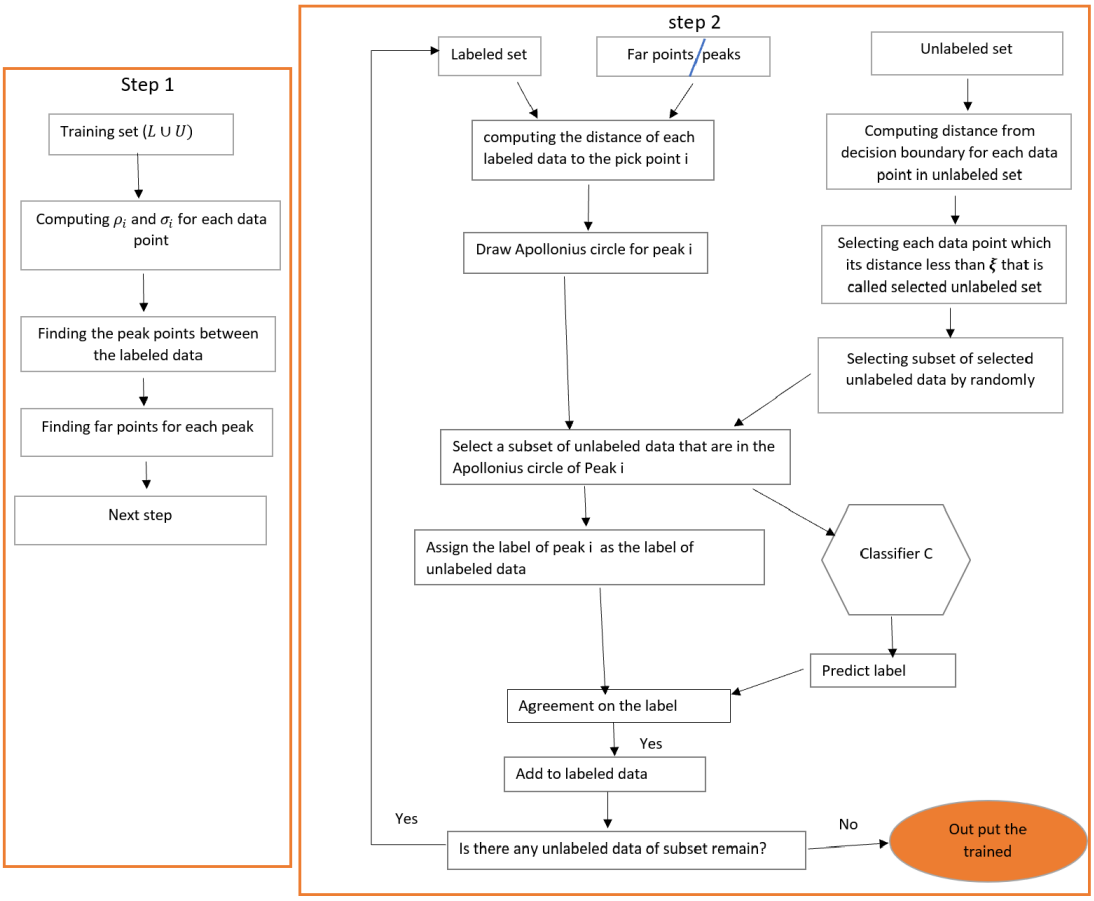


Fig. 4: overview of proposed algorithm

4 Experiments

In this section, we perform several experiments to compare the classification performance of our proposed algorithm to the state-of-the-art semi-supervised

methods using several different datasets. We also setup several experiments to show, the impact of selecting data close the decision boundary for improving the classification performance.

4.1 UCI dataset

In the experiment, some UCI datasets are used. Table 1 summarizes the specification of 8 benchmark datasets from the UCI data repository which are used in our experiments.

4.2 Experimental Setup

For each dataset, 30% of data are kept as test set randomly and the rest are used for training set. The training set is divided into two sets of labeled data and unlabeled data. Classes are selected at a proportions equal to the original dataset for all sets. Initially, we have assigned 5% to the labeled data and 95% to the unlabeled data. We have repeated each experiment ten times with different subsets of training and testing data. We report the mean accuracy rate (MAR) of 10 times repeating the experiments.

Table 1: Summarize the properties of all the used datasets

Name	#Example	#Attributed(D)	#Class
iris	150	4	3
wine	178	13	3
seeds	210	7	3
thyroid	215	5	3
Glass	214	9	6
banknote	1372	4	2
liver	345	6	2
Blood	748	4	2

4.3 Results

Table 2 shows the comparison of our algorithm with some other algorithms when labeled data is 10%. The second and third columns in this table give respectively the performance of the supervised SVM and selftraining SVM. The fourth column is the performance of state-of-the-art algorithm that is called STC-DPC algorithm [7]. The last column is the performance of our algorithm. Base learner for all of algorithms is SVM. Cut of distance parameter (dc) for our algorithm and STC-DPC is 0.05. From Table 2, we observe that Our algorithm works well for datasets that have a separate data density such as Iris, Seeds, Wine. Our proposed algorithm doesn't work very well if dataset is very mixed, such as banknote Fig 5 and Fig 6. We also investigate the behavior of the algorithms based

on increasing ratio of labeled data. Fig 7 is a comparison of the three algorithms with the increase ratio of label data from 5% to 50%.

Table 2: Experimental results of comparisons accuracy of the algorithms with 10% labeled data

dataset	supervised SVM	Self training	STC-DPC algorithm	Our algorithm
iris	92.50	87	91	95.76
wine	88.30	90.81	86.96	91.40
seeds	84.16	74.40	81.19	92.35
thyroid	88.95	87.21	89.65	91.72
Glass	47.44	51.15	51.15	51.93
banknote	98.39	98.77	98.12	96.62
liver	58.04	57.31	55.29	61.90
Blood	72.42	72.58	72.01	74.98

4.4 Impact of selecting data close the decision boundary

In most datasets, labeling all unlabeled data can not improve the performance but also reduces the accuracy. In addition to decreasing accuracy, runtime is also increased. Although the unlabeled data that are far from the decision boundary are more reliable, they are not informative. They play little role in improving decision boundary. That's why we haven't added all the unlabeled data to the training set, rather, we add those that are closer to the decision boundary than a certain value.

We show the results on a number of datasets in Table 3. The second column is the accuracy of our algorithm when we have added all the unlabeled data and the third column is the accuracy of our algorithm when we add only the data point closes to the decision boundary. As can be seen from Table 3, the accuracy of the algorithm increases when we only add unlabeled data closer to the decision boundary instead of all the points.

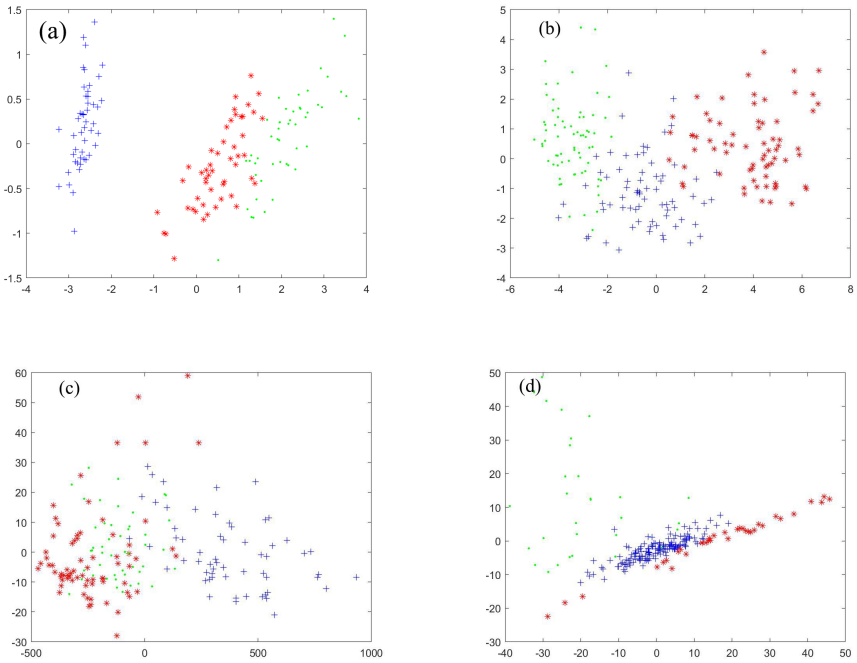


Fig. 5: Well-separated datasets : (a) iris , (b) seeds , (c) wine , (d) thyroid

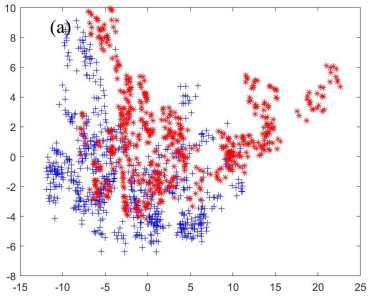


Fig. 6: dataset with mix classes : (a) banknote

Table 3: accuracy rate of our algorithm with all unlabeled data and near decision boundary unlabeled data

dataset	All unlabeled data	Close unlabeled
iris	95.59	95.76
wine	89.18	91.26
seeds	91.75	92.35
thyroid	92.31	92.20
banknote	96.58	96.62
liver	59.85	61.90

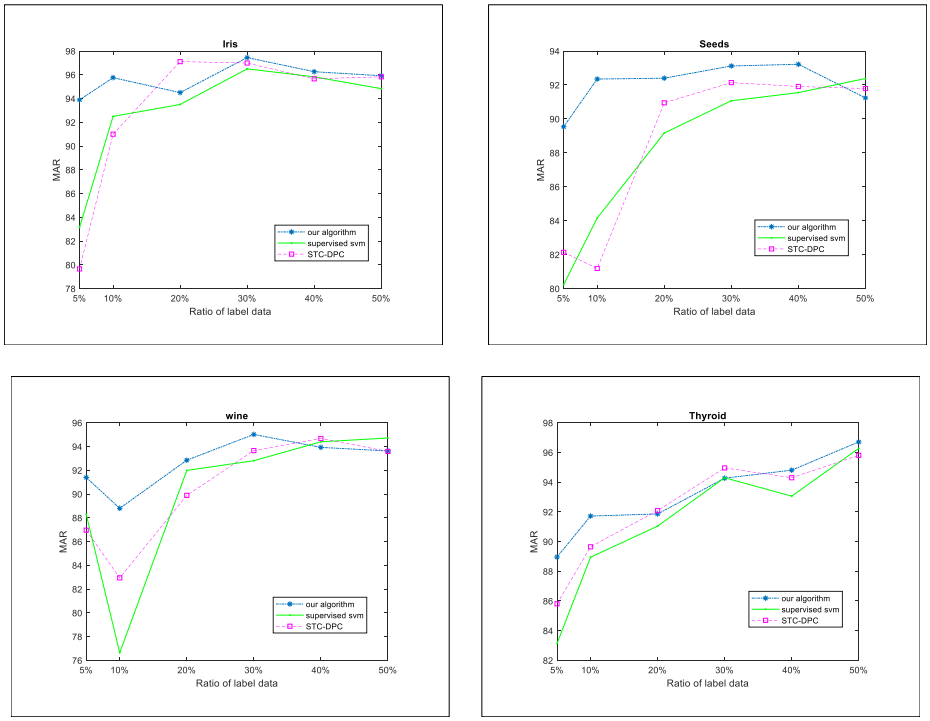


Fig. 7: Test MAR of our algorithm and supervised SVM and STC-DPC with respect to the ratio of labeled data on different datasets

5 Conclusion

In this paper, we proposed a semi-supervised self-training method based on Apollonius, named SSApolo. First candidate data are selected from among the unlabeled data to be labeled in the self training process, then, using the density peak clustering,the peak points are found and by making an Apollonius circle of

each peak points,their neighbors are found and labeled. Support vector machine is used for classification. A series of experiments was performed on some datasets and the performance of the proposed algorithm was evaluated. According to the experimental results, we conclude that our algorithm performs better than STC-DPC algorithm and supervised SVM and self-training SVM, especially when classes of dataset are not very mixed. In addition, the impact of selecting data are close to decision boundary was investigated. We find that selecting data are close to decision boundary can improves the performance.

References

1. Revisiting transductive support vector machines with margin distribution embedding. Knowledge-Based Systems **152**, 200 – 214 (2018). https://doi.org/https://doi.org/10.1016/j.knosys.2018.04.017, `http://www.sciencedirect.com/science/article/pii/S095070511830176X`
2. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. The Journal of Machine Learning Research **7**, 2399–2434 (2006)
3. Bennett, K., Demiriz, A.: Semi-supervised support vector machines. NIPS pp. 368–374 (1999)
4. Chapelle, O., Schlkopf, B., Zien, A.: Semi-Supervised Learning. The MIT Press, 1st edn. (2010)
5. Chen, K., Wang, S.: Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. Pattern Analysis and Machine Intelligence **33**(1), 129–143 (2011)
6. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
7. Di Wu, Mingsheng Shang, X.L.J.X.H.Y.W.D., Wang, G.: Self-training semi-supervised classification based on density peaks of data. Neurocomputing **275, No. C**, 180–191 (January 2018)
8. Ding, S., Zhu, Z., Zhang, X.: An overview on semi-supervised support vector machine. Neural Comput. Appl. **28**(5), 969978 (May 2017). https://doi.org/10.1007/s00521-015-2113-7, `https://doi.org/10.1007/s00521-015-2113-7`
9. Fazakis, N., Karlos, S., Kotsiantis, Sotiris, S.K.: Self-trained lmt for semisupervised learning. Hindawi **2016** (Dec 2016). https://doi.org/10.1155/2016/3057481
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
11. Habib, R., Mariooryad, S., Shannon, M., Battenberg, E., Skerry-Ryan, R.J., Stanton, D., Kao, D., Bagby, T.: Semi-supervised generative modeling for controllable speech synthesis. ArXiv **abs/1910.01709** (2019)
12. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML. pp. 200–209 (1999)
13. Langevin, M., Mehlman, E., Regier, J., Lopez, R., Jordan, M.I., Yosef, N.: A deep generative model for semi-supervised classification with noisy labels. CoRR **abs/1809.05957** (2018), `http://arxiv.org/abs/1809.05957`
14. Li, Y., Guan, C., Li, H., Chin, Z.: A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer

interface speller system. Pattern Recognition Letters **29**(9), 1285 – 1294 (2008). https://doi.org/https://doi.org/10.1016/j.patrec.2008.01.030, http://www.sciencedirect.com/science/article/pii/S016786550800055X
15. Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O.: Improving semi-supervised learning with auxiliary deep generative models. In: NIPS Workshop on Advances in Approximate Bayesian Inference (2015)
16. Mallapragada, P., Jin, R., Jain, A., Liu, Y.: Semiboost: Boosting for semi-supervised learning. Pattern Analysis and Machine Intelligence **31**(11), 2000–2014 (2009)
17. Partensky, M.B.: The circle of apollonius and its applications in introductory physics. The Physics Teacher **46**(2), 104–108 (2008). https://doi.org/10.1119/1.2834533, https://doi.org/10.1119/1.2834533
18. Pourbahrami, S., Khanli, L.M., Azimpour, S.: A novel and efficient data point neighborhood construction algorithm based on apollonius circle. Expert Systems with Applications **115**, 57 – 67 (2019). https://doi.org/https://doi.org/10.1016/j.eswa.2018.07.066, http://www.sciencedirect.com/science/article/pii/S095741741830486X
19. Qiao, S., Shen, W., Zhang, Z., Wang, B., Yuille, A.L.: Deep co-training for semi-supervised image recognition. ArXiv **abs/1803.05984** (2018)
20. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. science **344, Issue 6191**, 1492–1496 (27 Jun 2014). https://doi.org/10.1126/science.1242072
21. Seeger, M.: Learning with labeled and unlabeled data (technical report). Edinburgh University (2000)
22. Tanha, J.: A multiclass boosting algorithm to labeled and unlabeled data. International Journal of Machine Learning and Cybernetics (May 2019)
23. Tanha, J., van Someren, Maarten, A.H.: Semi-supervised self-training for decision tree classifiers. International Journal of Machine Learning and Cybernetics **8**, 355370 (2017). https://doi.org/10.1007/s13042-015-0328-7
24. Tanha, J., van Someren, M., Afsarmanesh, H.: Boosting for multiclass semi-supervised learning. Pattern Recognition Letters **37**, 63–77 (2014)
25. Vehlow, C., Beck, F., Weiskopf, D.: The state of the art in visualizing group structures in graphs. In: EuroVis (STARs). pp. 21–40 (2015)
26. Wang, X., Wen, J., Alam, S., Jiang, Z., Wu, Y.: Semi-supervised learning combining transductive support vector machine with active learning. Neurocomput. **173**(P3), 12881298 (jan 2016). https://doi.org/10.1016/j.neucom.2015.08.087, https://doi.org/10.1016/j.neucom.2015.08.087
27. Zhang, Y., Wen, J., Wang, X., Jiang, Z.: Semi-supervised learning combining co-training with active learning. Expert Systems with Applications **41**(5), 2372 – 2378 (2014). https://doi.org/https://doi.org/10.1016/j.eswa.2013.09.035, http://www.sciencedirect.com/science/article/pii/S0957417413007896
28. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. NIPS **16**, 321–328 (2004)
29. Zhou, Y., Kantarcioglu, M., Thuraisingham, B.: Self-training with selection-by-rejection. p. 795803. ICDM '12: Proceedings of the 2012 IEEE 12th International Conference on Data Mining, https://doi.org/10.1109/ICDM.2012.56 (December 2012)
30. Zhu, X.: Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison (2005), http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf