Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Self-training algorithm combining density peak and cut edge weight

Yang Liu[a,*]

[a]Mathematics and Statistics, China, and Chongqing University, Chongqing

ARTICLE INFO

Article History:
Submitted 3.23.2020
Revised 6.19.2020
Second Revision 7.07.2020
Accepted 7.08.2020

Keywords:
self-training
Density
cut edge weight
hypothesis testing

ABSTRACT

In view of the influence of mislabeled samples on the performance of self-training algorithm in the process of iteration, a self-training algorithm based on density peak and cut edge weight is proposed. Firstly, the representative unlabeled samples are selected for labels prediction by space structure, which is discovered by clustering method based on density of data. Secondly, cut edge weight is used as statistics to make hypothesis testing. This technique is for identifying whether samples are labeled correctly. And then the set of labeled data is gradually enlarged until all unlabeled samples are labeled. The proposed method not only makes full use of space structure information, but also solves the problem that some data may be classified incorrectly. Thus, the classification accuracy of algorithm is improved in a great measure. Extensive experiments on real datasets clearly illustrate the effectiveness of proposed method.

1. Introduction

Data classification is a very active research direction in the field of machine learning. In order to train an effective classifier, traditional supervised classification methods often require a large number of labeled samples. However, in practical applications, the acquisition of labeled samples requires a large price and is not easy to obtain, and the acquisition of unlabeled samples is relatively easy. Therefore, when the number of labeled samples is small, supervised classification methods are difficult to train an effective classifier[5,9].In this case, the semi-supervised classification method, which requires only a small number of labeled samples and makes full use of a large number of unlabeled samples, has attracted more and more attention. [4,7] Self-training is one of the commonly used methods in semi-supervised classification. First, an initial classifier is trained with a small number of labeled samples, and the unlabeled samples are classified. Then, select unlabeled samples with higher confidence and their predicted labels, expand the labeled sample set, and update the classifier. These two processes continue to iterate until the algorithm converges.

[1,3,7]Self-training methods do not require any specific assumptions, are simple and effective, and have been widely used in many fields such as text classification, face recognition, biomedicine, and so on. But self-training classification algorithms also have some drawbacks, such as the classification performance is limited by the size of the initial labeled data set and their distribution across the entire data set. Aiming at the shortcomings of the self-training method, [11] Considering the spatial distribution of the data set, a semi-supervised fuzzy c-means clustering method is proposed to optimize the self-training algorithm (ST-FCM). This method integrates the semi-supervised clustering technology as an auxiliary strategy into the self-training process. The semi-supervised clustering technology can effectively mine the internal data spatial structure information contained in the unlabeled samples and better train the classifier. However, the fuzzy c-means clustering method cannot find the spatial structure of non-Gaussian distributed data sets well. [2,5,8] proposed Self-training based on density peak of data (ST-DP). In the ST-DP algorithm, the spatial structure of the data is found using density peak clustering. Although the method based on density peak clustering can make effective use of the spatial structure of various data distributions, the ST-DP classification of some datasets with more overlapping samples after visualization Ineffective. Subsequently, [11,14] used Differential evolution (DE) to improve

DOI reference number: 10.18293/JVLC2020-N1-004

the self-training algorithm, and proposed a self-training algorithm based on differential evolution (ST-DE). [15] This method uses DE algorithm to optimize the newly added labeled samples during self-training. Although the ST-DE algorithm solves the problem of overlapping samples, the optimization algorithm brings too many complicated operations to a certain extent. This method does not fundamentally solve the shortcomings of the ST-DP algorithm. The main reason is that in the self-training labeling process, those overlapping samples after visualization are extremely easy to be labeled. The ST-DP algorithm uses these mislabeled samples directly for subsequent iterative labeling, which ultimately reduces the performance of the trained classifier.

Based on the ST-DP algorithm, this paper proposes a Self-training method based on density peak and cut edge weight (ST-DP-CEW). This method not only selects unlabeled samples, uses the density clustering-based method to discover the underlying spatial structure of the data set, and selects representative samples for label prediction. Further, the correctness of the predicted labels can be identified by using the statistical method of cutting edge weights. Cutting edge weights and density peak clustering make full use of the sample spatial structure and unlabeled sample information, solve the problem of some samples being labeled incorrectly, reduce the accumulation of errors during iteration, and can effectively improve the performance of the classifier.

.

2. Algorithm construction

In this paper, we improve the classification accuracy of the self-trained semi-supervised classification algorithm by starting with the wrongly labeled samples during the self-training process. Based on ST-DP, the ST-DP-CEW algorithm is proposed. First, the spatial structure of the data set is discovered by density clustering method, and representative samples can be preferentially selected for label prediction during each iteration. Then, we use the statistical method of cutting edge weights to judge whether the samples are correctly labeled, and update the labeled set with the correctly labeled samples. The above process is iterated until all unlabeled samples are completely labeled.

1.Spatial structure of data

Clustering is a typical unsupervised learning method. The process of clustering can discover the spatial structure of data. The method based on density clustering can find the spatial structure of non-Gaussian distributed data sets and can automatically determine the number of clusters.

In this paper, let $L=\{(x_i,y_i)\}$ be the labeled sample set, where x_i is the training sample, and y_i is its label. $y_{i1}\in\{\omega_1,\omega_2,\cdots,\omega_s\}$, $i=1,2,\cdots,m$. S is the number of categories. $U=\{x_{m+1},x_{m+2},\cdots,x_n)$ is the unlabeled sample set. The local density of sample x_i is defined as follows:

$$\rho_i=\sum\chi\left(d_{ij}-d_c\right)$$

Among them:

$$\chi(x)=\begin{cases}1, & x<0\\ 0, & x\geq 0\end{cases}$$

d_{ij} is the Euclidean distance between samples x_i and x_i , and d_c is called the truncation distance. It is a constant that has no fixed value and is related to the data set itself(Wang & Xu, 2017). After calculating the ρ_i value of each sample x_i , find the sample x_{j} that is closest to sample x_i and has a greater local density, point x_i to x_{j} , and find the spatial structure of the data set.

2. Statistical method of cutting edge weights

[7]Trim weighting is a method to identify and process mislabeled samples. First, in order to illustrate the similarity of the samples, a relative adjacency graph is established on the data set. The two samples x_i and x_{j} are connected side by side, if the following conditions are met: $d(x_i,x_j)\leq\max\left(d(x_i,x_m),d(x_j,x_m)\right),\forall m\neq i,j$,Where $d(x_i,x_j)$ is the distance between samples x_i and x_{j} . In an adjacency graph, if two samples with edges connected by different labels, this edge is called a cut edge. In an adjacency graph, if two samples with edges connected by different labels, this edge is called a cut edge. If x_i has many cut edges, that is, most of the samples in the neighborhood have labels that are different from those of x_i , it is considered that it may be labeled incorrectly. Therefore, cut edges play an important role in identifying mislabeled samples. For different samples, they may have the same number of cutting edges, but the importance of each cutting edge is different, so each edge in the adjacent graph is given a weight. Let w_{ij} be the weight of the edges connecting samples x_i and x_{j} .

. Finally, the hypothesis test was used to identify whether sample x_i was labeled incorrectly. The sum of the trimming weights J_i of sample x_i is defined as follows:

$$J_i=\sum_{j=1}^{n_i}w_{ij}I_i(j)$$

Among them,

$$I_i(j)=\begin{cases}1, & y_i\neq y_j\\ 0, & y_i=y_j\end{cases}$$

$\mathbf{n}_i$ is the number of samples with edges connected to sample x_i , and y_i is the label of sample x_i . If the J_i value of the sample x_i to be tested is large, it is considered that the sample may be labeled incorrectly. For hypothesis testing, the null hypothesis is defined as follows:

H_0 : All samples in the adjacent graph are labeled independently of each other according to the same probability distribution pro_y . pro_y represents the probability that the sample label is y .

In order to do a bilateral test, you must first analyze the distribution of J_i under H_0 . Under the null hypothesis, $I_i(j)$ is an independent identically distributed random variable subject to a Boolean parameter of $1 - pro_{y_i}$. So the expected μ_0 and variance σ^2 of J_i under H_0 are:

$$\mu_0 = \left(1 - \mathrm{pro}_{y_i}\right) \sum_{j=1}^{n_i} w_{ij}$$

$$\sigma^2 = \mathrm{pro}_{y_i} \left(1 - \mathrm{pro}_{y_i}\right) \sum_{j=1}^{n_i} w_{ij}^2$$

J_i follows the normal distribution $J_i \sim N\left(\mu_0, \sigma^2\right)$ under the original hypothesis H_0 , so the selected test statistic is

$$u = \frac{J_i - \mu_0}{\sigma}$$

Given a significance level of α , the rejection domain is:

$$W = \left\{ | u | \geq u_{1-\alpha/2} \right\}$$

The rejection domain that gets the sum of the trimming weights is

$$W = \left[-\infty, \mu_0 - \sigma \cdot u_{1-\alpha/2} \right] \cup \left[\mu_0 + \sigma \cdot u_{1-\alpha/2}, +\infty \right]$$

For sample x_i to be tested, if the value of J_i is significantly lower than the expected value under H_0 , that is, the rejection field on the left, the sample is marked correctly, otherwise it may be marked incorrectly. The main steps of the algorithm for identifying wrongly labeled samples using the edge-cut weights statistical method are as follows:

Step1. Create a relative adjacency graph for the sample set, and initialize the correctly labeled sample set $T = \{\varnothing\}$ and the incorrectly labeled sample set $T' = \{\varnothing\}$.

Step2. To assign weights to each edge in the adjacent graph, calculate the cut-edge weights of each sample and the expected and variance under the original hypothesis.

Step3. Given the significance level, calculate the rejection domain.

Step4. If the value of J_i is in the rejection field on the left, the label is correct, and the correct label set is updated; if it is not in the rejection field on the left, look at its neighbor samples. If the neighbor samples are all within T, then relabel with most label markers Otherwise, x_i mark errors, update the error mark set.

Step5. Repeat the above steps until all samples are tested.

3. Weight selection

The weight of each edge plays an important role in the statistical method of the edge weight. In this paper, the weight is first used to normalize the other nearest neighbor distances in the neighborhood by using the maximum nearest neighbor distance of each sample. Then calculate the probability that the sample has the same label as each neighboring sample, which is the weight of the edge.

Let sample set $\left\{ x|_{i,1}, x_{i,2}, \cdots, x_{i,k} \right\}$ be the k adjacent samples of sample $\left(x_i, y_i\right)$, that is, they are connected to x_i with edges. x_i is the training sample, y_i is the label of x_i , and the distance between each adjacent sample and x_i satisfies the condition: $d\left(x_{i,1}, x_i\right) \leq d\left(x_{i,2}, x_i\right) \leq \cdots \leq d\left(x_{i,k}, x_i\right)$. Use the k -th nearest neighbor sample distance of x_i to normalize the distance from the first $k-1$ adjacent samples to x_i , then the normalized distance is:

$$D\left(x_{i,j}, x_i\right) = \frac{d\left(x_{i,j}, x_i\right)}{d\left(x_{i,k}, x_i\right)}, \quad j = 1, 2, \cdots, k$$

The weight of each edge in the adjacency graph is:

$$w_{ij} = P\left(x_{i,j} \mid x_i\right) = \frac{1}{\sqrt{2\pi}} \exp\left(- \frac{D\left(x_{i,j}, x_i\right)}{2} \right)$$

4. Self-training algorithm based on density and trimming weights

Since the self-training algorithm tends to mark unlabeled samples at each iteration, these errors will be involved in the next iteration, which will affect the training of the classifier and reduce the performance of the algorithm. Therefore, in the process of self-training, identifying wrong labeled samples plays an important role in the performance of the algorithm. There are many methods for identifying sample labels, the common ones are filtering methods based on classifiers and data editing techniques based on nearest neighbor rules.

The classifier-based filtering method mainly divides the existing labeled sample set into n subsets during each iteration training, and uses the same learning algorithm such as C4.5 to train n in all possible n-1 subsets to get n Different classifiers. Then use n classifiers to classify the unlabeled samples, and select the labels of the samples according to the principle of consensus or majority voting. The data editing technique based on the nearest neighbor rule mainly relies on distance, and judges whether the label of the sample to be predicted is correct according to the labels of k nearest neighbor samples.

Classifier-based methods have extremely high requirements for the partitioning of sample sets and the selection of learning algorithms. The selection of distance metrics and values based on the nearest neighbor method need to be set in advance. If it is not selected properly in advance, it will cause a judgment error and affect the final classification effect. In addition, neither of these two methods uses a lot of valuable information carried by unlabeled samples in the recognition process, which reduces the accuracy of recognition. The method of cutting edge weight statistics to identify wrongly labeled samples does not need to set any parameters in advance, and it can also make full use of the information of unlabeled samples. Therefore, in order to improve the classification accuracy of the self-training algorithm, this paper incorporates the method of cutting edge weights to statistically identify the wrong label samples into the

ST-DP algorithm, and proposes the ST-DP-CEW algorithm. The algorithm first uses the density clustering method to discover the spatial structure of the data set, and uses the spatial result information to preferentially select representative unlabeled samples for label prediction during the iteration process, which improves the accuracy of predicting labels. Then use the method of cutting edge weight statistics to judge whether the prediction label is correct. Use the correctly labeled samples for the next training. The specific steps of the algorithm are described as follows:

Step1. Use the density clustering method to find the true space structure of the entire data set.

Step2. (a) Use KNN or SVM as the base classifier, and train an initial classifier with the initial labeled sample set;

(b) label prediction on the "next" unlabeled sample of all samples in;

(c) identify whether the "next" sample is correctly labeled by using the method of trimming edge weights to obtain a correctly labeled sample;

(d) Repeat (a) through (c) until all "next" samples of have been marked.

Step3. (a) Perform label prediction on the "previous" unlabeled samples of all the updated samples;

(b) Identify the "previous" sample using the edge-cut weighting statistical method to obtain the correct labeled sample, and then update the classifier;

(c) Repeat (a) and (b) until all "previous" samples of have been marked.

Obviously, Step3 is similar to Step2, except that the "next" in Step2 is replaced with "previous".

3. Experimental results and analysis

In order to illustrate the effectiveness of the algorithm, the proposed algorithm is compared with existing self-training algorithms on 8 real data sets. The datasets are derived from the KEEL database[6]. Samples with missing values are deleted from the Cleveland and Dermatology datasets, and the rest of the datasets are not processed. Related information is shown in Table 1.

Table 1 Experimental data set

data set	size	dimension	category
Bupa	345	6	2
Cleveland	297	13	5
Dermatology	358	33	6
Glass	214	9	7
Haberman	306	3	2
Ionosphere	351	34	2
pima	768	8	2
yeast	1484	8	10

The comparison algorithms used are: traditional self-training algorithms using KNN and SVM as classifiers, self-training classification algorithms based on fuzzy c-means clustering (ST-FCM), density-based self-training classification algorithms (ST-DP), and Self-training classification algorithm (ST-DE) based on differential evolution. The specific parameter settings are shown in Table 2.

Table 2 Parameter settings of related algorithms in the experiment

algorithm	parameter
KNN	K=3
SVM	Same settings as Literature(Chih-Chung & Chih-Jen, 2011)
ST-FCM	$\varepsilon_1 = 1$
ST-DP	$P_a = 2$
ST-DE	$P_a = 2$; $DE - POAC\left(L^{'}, L\right)$ Same settings as Literature(Chih-Chung & Chih-Jen, 2011)
ST-DP-CEW	$P_a = 2$;Significance level: $\alpha = 0.05$

1. Implementation of the experiment

A ten-fold cross-validation strategy was used to perform experiments on the dataset using KNN and SVM as base classifiers. Take one fold as the test set and the remaining nine fold as the training set. In each experiment, 10% of the samples in the training set are randomly selected as the initial labeled sample set, and the rest are unlabeled sets. In order to ensure the accuracy of the experiment, the ten-fold cross-validation experiment was repeated ten times, and the average value of the ten experiments was finally selected as the final experimental result. Accuracy rate (AR), Mean accuracy rate (MAR), and Standard deviation (SD-AR) are used as comparison criteria for the classification performance of the algorithm. Calculated as follows:

$$AR = \frac{1}{N_{T_s}} \sum_{i=1}^{N_{TS}} \psi\left(\omega, f\left(x_i\right)\right)$$

$$MAR = \frac{1}{n} \sum_{k=1}^{n} AR_k$$

$$SD - AR = \sqrt{\frac{1}{n} \sum_{k=1}^{n} \left(AR_k - MAR\right)^2}$$

$f\left(x_i\right)$ is the predicted label of the sample, N_{T_s} is the size of the test set, n is the number of times the experiment is repeated, MAR represents the classification performance of the algorithm, and SD-AR represents the robustness of the algorithm. MAR ± SD-AR is selected as the basis for judging the performance of the algorithm.

Tables 3 and 4 show the experimental results of the data set with KNN and SVM as the base classifier, respectively. The bold data indicates that the algorithm performs better in classification. As shown in Tables 1 and 2, when the initial labeled sample is 10%, the average classification accuracy of ST-DP-CEW on multiple data sets is significantly better than other comparison algorithms. However, when the algorithm is based on the SVM classifier, the classification accuracy of ST-DP-CEW on the dataset Cleveland has basically not improved. This is mainly because the values of most attributes in the dataset are close to 0. For the same attribute, The differences between the samples are small, resulting in a small difference between the samples as a whole, and the discrimination of each category is reduced, which affects the final classification effect.

Table 3 Experimental results when the base classifier is KNN (MAR ± SD-AR, %)

data set	Classifier: KNN				
	KNN only	ST-FCM	ST-DP	ST-DE	ST-DP-CEW
Bupa	54.48±7.99	56.91±9.34	58.88±8.79	59.13±8.43	**62.27±6.21**
Cleveland	46.79±6.70	46.47±7.46	48.16±8.65	49.15±8.54	**52.17±7.84**
Dermatology	53.60±8.10	56.18±7.58	70.94±8.18	73.98±7.21	**78.19±6.64**
Glass	50.54±7.59	5L58±7.67	55.26 M.84	57.40±8.35	**61.65±6.83**
Haberman	67.59±9.28	67.92±9.52	69.31±6.91	68.91±8.29	**72.19±7.11**
Ionosphere	74.35±8.00	72.35±8.33	80.61±4.05	81.20±5.44	**83.45±7.78**
pi ma	67.72±5.32	64.98±4.56	66.40±2.54	66.93±4.57	**70.05±2.70**
yeast	45.96±5.83	48.32±3.22	49.19±3.28	50.74±4.71	**53.10±3.62**

Table 4 Experimental results when the base classifier is SVM (MAR ± SD-AR, %)

data set	Classifier: SVM				
	KNN only	ST-FCM	ST-DP	ST-DE	ST-DP-CEW
Bupa	60.86±7.33	62.57±7.70	65.50±7.56	65.80±6.30	**67.01±8.20**
Cleveland	**53.84±8.33**	**53.84±4.29**	53.82±8.76	53.82±7.39	**53.84±9.32**
Dermatology	56.41±9.64	57.28±9.65	68.14±6.54	72.36±9.72	**78.25±9.28**
Glass	44.81±9.87	46.34±8.07	49.46±9.10	51.36±7.99	**54.72±7.75**
Haberman	70.59±7.06	71.61±4.10	71.85±5.56	72.24±7.62	**74.62±5.71**
Ionosphere	78.33±4.16	79.75±8.16	80.92±6.10	82.34±5.22	**84.92±6.82**
pi ma	71.75±6.13	72.53±6.37	75.12±4.72	75.78±2.40	**77.23±3.16**
yeast	31.54±2.29	30.76±3.68	31.21±3.34	32.43±4.25	**35.81±2.63**

4. Conclusion

In this paper, based on the ST-DP algorithm, a self-training algorithm based on density peaks and edge trimming weights is proposed based on the samples that may be mislabeled during the self-training iteration process. That is, the method of statistically identifying cut-off weights to identify incorrectly labeled samples is integrated into the ST-DP algorithm. It not only considers the spatial structure of the data set, but also solves the problem that the samples are incorrectly labeled. In addition, the calculation of the weights in the adjacency graph also makes better use of the spatial structure of the data set and the information carried by the unlabeled samples. The effectiveness of the ST-DP-CEW algorithm is fully analyzed on the real data set. Especially when the proportion of initially labeled samples is low, the proposed algorithm has greatly improved performance compared to existing algorithms. In the subsequent work, we will discuss how to better construct the adjacency graph, and introduce a function that measures the probability of label error in the recognition process to make label recognition more accurate.

References

[1]Alcala-Fdez, J., Fernández, A., Luengo, J., Derrac,J., & García, S. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Multiple-Valued Logic and Soft Computing*, *17*, 255–287.

[2]Chih-Chung, C., & Chih-Jen, L. (2011). Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol., Vol. 2(3)*, 1–27.

[3]Gan, H., Sang, N., Huang, R., Tong, X., & Dan, Z.(2013). Using clustering analysis to improve semi-supervised classification. *Neurocomputing*, *101*, 290–298. https://doi.org/https://doi.org/10.1016/j.neucom.2012.08.020

[4] Liu, J., Gong, M., & He, H. (2019). Deep associative neural network for associative memory based on unsupervised representation learning. *Neural Networks : The Official Journal of the International Neural Network Society*, *113*, 41–53. https://doi.org/10.1016/j.neunet.2019.01.004

[5]Pavlinek, M., & Podgorelec, V. (2017). Text classification method based on self-training and LDA topic models. *Expert Systems with Applications*, *80*, 83–93. https://doi.org/https://doi.org/10.1016/j.eswa.2017.03.020

[6]fast search and find of density peaks. *Science*, *344*(6191), 1492–1496. https://doi.org/10.1126/science.1242072

[7]Triguero, I., Sáez, J. A., Luengo, J., García, S., & Herrera, F. (2014). On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, *132*, 30–41. https://doi.org/https://doi.org/10.1016/j.neucom.2013.05.055

[8]Vijayan, A., Kareem, S., & Kizhakkethottam, J. J. (2016). Face Recognition Across Gender Transformation Using SVM Classifier. *Procedia Technology*, *24*, 1366–1373. https://doi.org/https://doi.org/10.1016/j.protcy.2016.05.150

[9]Wang, X.-F., & Xu, Y. (2017). Fast clustering using adaptive density peak detection. *Statistical Methods in Medical Research*, *26*(6), 2800–2811. https://doi.org/10.1177/0962280215609948

[10]Wu, D, Luo, X., Wang, G., Shang, M., Yuan, Y., & Yan, H. (2018). A Highly Accurate Framework for Self-Labeled Semisupervised Classification in Industrial Applications. *IEEE Transactions on Industrial Informatics*, *14*(3), 909–920. https://doi.org/10.1109/TII.2017.2737827

[11]Wu, Di, Shang, M., Luo, X., Xu, J., Yan, H., Deng, W., & Wang, G. (2018). Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, *275*, 180–191. https://doi.org/https://doi.org/10.1016/j.neucom.2017.05.072

[12]Wu, Di, Shang, M. S., Wang, G., & Li, L. (2018). *A self-training semi-supervised classification algorithm based on density peaks of data and differential evolution*. 1–6. https://doi.org/10.1109/ICNSC.2018.8361359

[13]Xu, G., Zhang, M., Zhu, H., & Xu, J. (2017). A 15-gene signature for prediction of colon cancer recurrence and prognosis based on SVM. *Gene*, *604*, 33–40. https://doi.org/https://doi.org/10.1016/j.gene.2016.12.016

[14]Alcala-Fdez, J., Fernández, A., Luengo, J., Derrac, J., & García, S. (2011). KEEL Data-Mining Software Tool: Data Set

Repository, Integration of Algorithms and Experimental Analysis Framework. Multiple-Valued Logic and Soft Computing, 17, 255–287.

[15]Chih-Chung, C., & Chih-Jen, L. (2011). Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol., Vol. 2(3), 1–27.

[16]Liu, J., Gong, M., & He, H. (2019). Deep associative neural network for associative memory based on unsupervised representation learning. Neural Networks : The Official Journal of the International Neural Network Society, 113, 41–53. https://doi.org/10.1016/j.neunet.2019.01.004.