PU-LP: A NOVEL APPROACH FOR POSITIVE AND UNLABELED LEARNING BY LABEL PROPAGATION

Shuangxun Ma, Ruisheng Zhang
School of Information Science & Engineering,
Lanzhou University, Lanzhou, P.R China, 730000
zhangrs@lzu.edu.cn

ABSTRACT

For the positive and unlabeled learning algorithms, when there is only small amount of labeled positive examples available, the algorithms can hardly extract reliable negative examples from the unlabeled examples in step one, which makes it hard to build the classifier with good performance in step two. Based on the *same label* assumption from graph based semi-supervised learning, we propose a novel graph-based PU learning algorithm, PU-LP, which takes Katz index to measure the similarities between vertices. After enlarging labeled positive set and extracting reliable negative examples, PU-LP build the classifier by label propagation algorithm. Experiments on UCI datasets shows that PU-LP has excellent performance when there is only small amount of labeled positive examples available, and it outperforms than PNB algorithm.

Keywords—PU Learning, label propagation, Katz index

1. INTRODUCTION

PU learning (Positive and Unlabeled Learning) is a special form of semi-supervised learning. It is different from the conventional semi-supervised methods, as the classifier is built only using labeled positive and unlabeled examples without negative examples [1]. Currently, there is a typical kind of approach for PU learning problem by building the classifier in two-step strategy. In step one, a set of reliable negative examples are identified from the given unlabeled set, and considered as negative set. In step two, an existing supervised or semi-supervised method is applied to train the classifier. The key of two-step strategy techniques is to get reliable negative examples. If there are too many noisy examples in the extracted negative examples, the performance of the classifier built in step two will be damaged. However, most existing methods for step one can't extract reliable negative examples when only small amount of labeled positive examples available.

Graph-based semi-supervised learning has attracted great attention in recent years. It starts by constructing a graph from the training set, and the vertices are the labeled and unlabeled examples. An edge between two vertices reflects the similarity between them. The graph-based semi-supervised learning is made possible by the *same label* assumption [2], which assumes that vertices connected by a large-weight edge tend to have the same class label, and labels can propagate throughout the graph until a global stable state is reached. Based on the *same label* assumption, a further assumption can be made: unlabeled vertices with small-weight edges connected to labeled positive vertices tend to be negative; unlabeled vertices with large-weight edges connected to the labeled positive vertices tend to be positive [3]. These unlabeled vertices can be considered as reliable positive and negative examples in step one of two-step strategy PU learning methods.

In this paper, we propose a novel graph-based algorithm for transductive PU learning: PU-LP. In PU-LP, reliable positive examples are extracted from unlabeled set iteratively to enlarge the labeled positive set by a novel graph-based approach. Based on the enlarged labeled positive set, relatively reliable negative examples can be extracted. At last, label propagation algorithm, which is a graph-based semi-supervised learning algorithm, is applied to propagate the labels to the rest of unlabeled examples. The experiments on UCI datasets show that PU-LP outperforms PNB algorithm when there is only a small amount of labeled positive examples.

The organization of this paper is as follows. In Section 2, we discuss related work. Section 3 presents the problem description. In Section 4, we present our algorithm in detail. In Section 5, experimental results and evaluation are given. Section 6 concludes this paper.

2. RELATED WORK

In this section, we will introduce related work in three aspects: PU learning algorithms, common ways to construct a graph in graph-based semi-supervised learning, and similarity measurements for vertices used in graphs.

PU learning: The two-step strategy is an important approach for PU learning, such as S-EM [4] and PEBL [5].

978-1-5386-0560-8/17/$31.00 ©2017 IEEE

Firstly, in step one, reliable negative examples are extracted from unlabeled examples. Reliable positive examples can also be extracted to enlarge the given labeled positive set before the reliable negative examples are extracted. Many techniques can be used in this step, such as the Naive Bayesian technique [6]. In step two, a supervised or semi-supervised learning algorithm is applied to train the classifier, such as SVM [5] and EM [4]. By now, only a few works [3][7] were dedicated to graph-based PU learning based on two-step strategy. In step one of GB-TPU [3], the similarity for each pair of vertices is measured by graph laplacian, and then reliable negative examples are extracted by dissimilarity model. In step two of GB-TPU, labels are propagated from labeled vertices to unlabeled vertices by a graph-based approach. The graph-based PU algorithms do not always take graph based classifier as the final classification model. For example, in PE-PUC [7], the final classifier is built by Naïve Bayesian technique.

Common ways to construct a graph: According to [2][8], graph-based semi-supervised learning starts by constructing a graph from the training set. Both labeled and unlabeled examples are represented by vertices in the graph. An edge between two vertices reflects the similarity of them, which is usually undirected. The ways to construct a graph can be divided into two categories: fully connected graph and sparse graph. In a fully connected graph, each pair of vertices is connected by a weighted edge. It is proposed in [2] that the fully connected graph is easy to construct, but it is not the best choice for graph-based semi-supervised learning. Sparse graph can be classified as kNN graph and εNN graph, where each vertex only connects to only a few vertices. In the process of constructing a kNN graph, each vertex connects to its k nearest neighbor vertices in Euclidean distance. In a εNN graph, if the Euclidean distance between vertices v_i and v_j satisfies $d(v_i, v_j) \leq \epsilon$, then $v_i$and v_j are connected by an edge. The edges in a sparse graph can be either unweighted or weighted. kNN graph can automatically adapts to the density of examples in feature space. εNN graph does not have this property, which makes kNN graph is superior to εNN graph. Empirically conclusion came from [2][8] shows that kNN graph with small k tend to perform well.

The similarity measurements: graph laplacian is a popular weight function in fully connected graphs [3][7][9]. In sparse graphs, if two vertices are connected, the edge weight is either the constant 1 for unweighted graph, or a function for weighted graph, such as graph laplacian. If two vertices are not connected, the weight is always set to zero. [2] argues that based on the *same label* assumption, the similarity between two vertices, which are not directly connected in a sparse graph, can be measured by a sequence of vertices where each pair of neighbor vertices are connected. The similarity index based on paths will be discussed in Section 4 particularly.

3. PROBLEM DEFINITION

Given a set of examples $\Pi = \{v_1, \dots, v_l, v_{l+1}, \dots, v_{l+u}\} \subset \mathbb{R}^m$ and a set of class labels $C = \{-1,1\}$, 1 represents positive and -1 represents negative. The first l elements of Π are labeled positive examples with class label 1, forming the labeled positive set P. The remaining u elements of Π are unlabeled examples, forming the unlabeled set U. $u \gg l$ and $\Pi = \mathrm{P} \cup \mathrm{U}$. In this paper, we cope with the problem of transductive PU learning, and the goal of transductive PU learning is to predict the class label of the examples in U.

4. PU LEARNING BY LABEL PROPAGATION

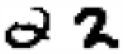
We focus on the problem to build PU learning classifiers when the size of labeled positive dataset is small. The key issue is how to extract reliable negative examples from unlabeled set. When the size of labeled positive set is small, the labeled positive examples can hardly reflect the true feature distribution of the positive class, so there will be many noisy examples in extracted negative examples. As a consequence, the performance of classifier built in step two will be damaged. Based on graph mining, we present PU-LP, a novel PU learning algorithm by label propagation following two-step strategy. The main steps of PU-LP are shown below.

Step 1: Construct a graph and calculate the similarity matrix. Based on the similarity matrix, a set of reliable positive examples RP is extracted from U to enlarge P. Based on $\mathrm{P} \cup \mathrm{RP}$, a set of reliable negative examples RN is extracted from $\mathrm{U} - \mathrm{RP}$.

Step 2: Propagate the labels from $\mathrm{P} \cup \mathrm{RP}$ and RN to $\mathrm{U} - \mathrm{RP} - \mathrm{RN}$ by using label propagation algorithm.

The detail of the steps will be introduced in the following.

4.1 Constructing the kNN graph and calculating the similarity matrix



(a) Two examples of '2' with large Euclidean distance

(b) A sequence of examples '2' between examples in **Fig.1** (a)

Fig. 1. ORC Examples

PU-LP algorithm starts by constructing a graph and calculating the similarity matrix. For most of existing graph-based PU learning algorithms and graph-based semi-supervised learning algorithms, the fully connected graph is selected, and then the similarity matrix is calculated

by graph laplacian. Graph laplacian decreases as the Euclidean distance increases. If there is a large Euclidean distance between two vertices, the similarity measured by graph laplacian will be small, that is, the two vertices are dissimilar. However, this does not always hold in real-life applications. Here, an example appeared in optical character recognition (ORC) for handwritten digits is presented in Fig. 1.

Due to the large distance, the similarity is small when measured by graph laplacian. So the examples in Fig.1(a) can be hardly classified into the same class. In Fig.1(b), for each pair of neighbor examples, there is only small Euclidean distance (large similarity) between them. Based on the *same label* assumption, each pair of neighbor examples tends to have same label. By a path that consists of a sequence of "2", examples in Fig.1(a) can be connected as shown in Fig.1(b) and classified into the same class. There are more examples mentioned other references. From these research work, it can be concluded as following: (1) While the Euclidean distance works excellently as a *local* similarity measurement, its performance as global similarity measurement is not excellent enough [9]. (2) Label can propagate along paths, and it is feasible to measure similarity by considering the impact of the paths between the vertices in the whole graph.

kNN method is used to construct the graph in PU-LP. Considering that Euclidean distance is a good local similarity measure, we can expect that two vertices to have same class label if they are directly connected (with small Euclidean distance). For pairs of vertices which are not directly connected (with large Euclidean distance), we need a similarity measurement which is based on the paths.

The similarity measurement which takes consideration of local paths evolved from Common Neighbors (CN) [10][11]. The basic idea of CN is that if two vertices have many common neighbor vertices, these two vertices are supposed to have the same class label. In CN, a common neighbor of v_i and v_j can be also considered as a path with length 2 between these two vertices. Vertices are more likely to have the same class label if they are connected by many paths with length 2. CN only requires information on the nearest neighbors. However the information usually seems insu□cient and the paths with length longer than 2 should be considered. Based on CN, Zhou et al. considered the paths with length 3 and proposed Local Path (LP) index [10][11]. LP index is defined as:

$$s_{ij} = (A^2)_{ij} + \alpha \cdot (A^3)_{ij} \tag{1}$$

Here, α is a free parameter which controls the weights of the path with length 3. A denotes the adjacent matrix of the graph, $(A^3)_{ij}$ is equal to the number of different paths with length 3 connecting v_i and v_j. LP index can be extended to higher-order forms easily. When the order approaches infinity, LP index is equal to Katz index [10][11], which considers all paths in the graph. The Katz index is defined as:

$$s_{ij} = \sum_{h=1}^{\infty} \alpha^h \cdot |path_{i,j}^{<h>}| = \alpha A_{ij} + \alpha^2 (A^2)_{ij} + \alpha^3 (A^3)_{ij} + \cdots \tag{2}$$

Under the control of the parameter α, the path with longer length makes less contribution to the similarity measurement. It is easy to calculate the similarity matrix by Katz index. When $\alpha < 1/\varepsilon$ (ε is the largest eigenvalue of the adjacency matrix A), formula (2) converges, and the similarity matrix can be calculated following:

$$W = (I - \alpha \cdot A)^{-1} - I \tag{3}$$

Here, I denotes the identity matrix. According to [10], the whole time cost of calculating W almost equals to the time cost of matrix inversion.

In PU-LP, the adjacent matrix A of the graph is a $(l+u) \times (l+u)$ matrix, where $A_{ij} = 1$ if vertices v_i and v_j are directly connected, and $A_{ij} = 0$ otherwise. Based on the kNN graph, formula (3) is used to calculate the similarity matrix W, $((l+u) \times (l+u))$, where $W_{ij} \in \mathbb{R}$ denotes the similarity between vertices v_i and v_j.

4.2 Extracting reliable positive and negative examples

Most published two-step strategy techniques can hardly extract reliable negative examples in step one with small amount of labeled positive examples available. In PU-LP, a small set of reliable positive examples RP are extracted from U to enlarge P at first. As only small amount of labeled positive examples are provided, it is hard for the algorithm to extract too many reliable positive examples at a time. To solve this problem, an iterative method is applied in PU-LP. The task of enlarging P is divided into m steps. We assume that the total number of reliable positive examples that need to be extracted is $\lambda \times |P|$, where λ controls the size. In each iterative step, examples in U are ranked according to their mean similarities to the all examples in P based W. The top $(\lambda/m) \cdot |P|$ examples are extracted from the ranked list, so as to form the set of RP′. After each iteration ends, $P = P \cup RP'$ and $RP = RP \cup RP'$. The enlarged P will be used as the new labeled positive set for next iterative step. After m iterations, RP is formed by examples from RP′ in each iterative step and $|RP| = \lambda \times |P|$.

After enlarging P iteratively, the reliable negative examples are extracted from $U - RP$ based on $P \cup RP$. Based on W, examples in $U - RP$ are ranked according to their mean similarities to the all examples in $P \cup RP$, the algorithm extracts the bottom $|P \cup RP|$ examples from the ranked list, so as to form the set of RN.

Algorithm 1. Extracting reliable positive and negative examples

PU-LP: Extracting reliable positive and negative examples $(\mathrm{P}, \mathrm{U}, m, \lambda, W)$

Input: a set of labeled positive examples, P; a set of unlabeled examples, U; the number of iteration, m; the parameter which controls the size of RP($|\mathrm{RP}| = \lambda \times |\mathrm{P}|$), λ, $\lambda \in (0,1)$; the similarity matrix for $\mathrm{P} \cup \mathrm{U}$ which calculated by Katz index, W, $W_{ij} \in \mathbb{R}$ denotes the similarity between vertices v_i and v_j;

Output: a set of extracted reliable positive examples, RP; a set of extracted reliable negative examples, RN;

1: $\mathrm{RP} \leftarrow \emptyset$, $\mathrm{P}' \leftarrow \mathrm{P}$, $\mathrm{U}' \leftarrow \mathrm{U}$; //$\mathrm{P}'$ and U' are the copies of P and U, which will be used in iterations followed.

2: **for** k=1:m **do**

3: based on W, calculate $\overline{S_{v_i}}$, $v_i \in \mathrm{U}'$, $\overline{S_{v_i}} = \frac{\sum_{j=1}^{|\mathrm{P}'|} W_{ij}}{|\mathrm{P}'|}$; //$\overline{S_{v_i}}$ is the mean similarity between an unlabeled examples v_i and all labeled positive examples;

4: rank each example v_i according to the $\overline{S_{v_i}}$, $v_i \in \mathrm{U}'$;

5: $\mathrm{RP}' \leftarrow$ the top $\frac{\lambda}{m} \times |\mathrm{P}|$ ranked examples in U';

6: $\mathrm{P}' \leftarrow \mathrm{P}' \cup \mathrm{RP}'$, $\mathrm{U}' \longleftarrow \mathrm{U}' - \mathrm{RP}'$, $\mathrm{RP} \leftarrow \mathrm{RP} \cup \mathrm{RP}'$;

7: **end for**

8: based on W, calculate $\overline{S_{v_i}}$, $v_i \in \mathrm{U} - \mathrm{RP}$, $\overline{S_{v_i}} = \frac{\sum_{j=1}^{|\mathrm{P} \cup \mathrm{RP}|} W_{ij}}{|\mathrm{P} \cup \mathrm{RP}|}$;

9: rank each example v_i according to the $\overline{S_{v_i}}$, $v_i \in \mathrm{U} - \mathrm{RP}$;

10:$\mathrm{RN} \leftarrow$ the bottom $|\mathrm{P} \cup \mathrm{RP}|$ ranked examples in $\mathrm{U} - \mathrm{RP}$;

11:**return** RP, RN

4.3 Build the final classifier

After the process of enlarging P and getting RN, the $\mathrm{P} \cup \mathrm{RP}$, RN and $\mathrm{U} - \mathrm{RP} - \mathrm{RN}$ are used to train the final classifier by label propagation algorithm [12]. The label propagation algorithm is a graph-based semi-supervised learning method, where labels from the labeled vertices (which are fixed) gradually through the edges to all the unlabeled vertices.

5. EXPERIMENTS RESULT

5.1 Experimental setting

In order to validate the performance of the proposed PU-LP algorithm, we made experiments on the UCI[1] machine learning datasets. Ten datasets were selected as benchmark datasets for our experiments and the detail of the selected datasets is given in Table 1. For each dataset, examples from one certain class are regarded as positive examples, and the rest examples are regarded as negative examples.

We follow the method in [1] for constructing the training set. Examples were selected randomly from positive examples with probability $d\%$ to form the labeled positive set P, the remaining positive examples and all negative examples were used to form the unlabeled set U. As in this paper, we are coping with transductive PU learning scenario, the test set in our experiments is U. We set $d\% = 10\%, 20\%$ in our experiment to create dataset with small amount of labeled positive examples.

Table 1. Experimental data sets from UCI

Dataset	Size	#Attributes	#Class	Positive Class	Pos/All
iris	150	4	3	versicolor	33.3%
seeds	210	7	3	1	33.3%
ecoli	336	7	8	im	22.91%
yeast	1484	8	10	MIT	16.44%
wine	178	13	3	1	33.14%
abalone	4177	8	3	M	36.58%
balance-scale	625	5	3	R	46.08%
cmc	1473	9	3	1	42.7%
harberman	306	3	2	2	26.5%
transfusion	748	4	2	1	23.8%

We compare the classification performance of the proposed PU-LP with PNB [13] (Positive Naive Bayes). PNB is a PU learning algorithm which applies Naive Bayesian (NB) classification algorithm to the PU learning scenario. The experiment result in [13] shows that the performance of PNB is similar to that of NB when there are sufficient training examples and labeled positive examples. However, the performance of PNB when there is only small amount of labeled positive examples available is not given. The prior probability for PNB in our experiments is estimated from positive and unlabeled examples according to [1].

The graph for PU-LP in our experiments is constructed by kNN method, with $\mathrm{k} = 4$. Then, the similarity matrix is calculated by Katz index following formula (3). Other parameters used in the experiments will be discussed later.

Both PU-LP and PNB algorithms in our experiments were implemented in MATLAB (Version 2014a), and were executed on a PC with Intel 2.40GHz processor, 8GB memory. The performance of PU-LP and PNB algorithms were measured by F1-score, which are widely used by the research community of PU learning [1][3][6][13]. In each group of experiment, fifty trails of experiment are conducted, and the averaged results are reported.

5.2 Classifier Performance Evaluation

[1] http://archive.ics.uci.edu/ml/index.html

In order to compare the classification performance of PU-LP and PNB algorithms, the experiment is conducted on the same training and test set. The experiment result is shown in Table 2.

Table 2. Experiment result of PU-LP and PNB

Dataset	PU-LP		PNB	
	d=10%	*d*=20%	*d*=10%	*d*=20%
iris	**91.66%**	**88.43%**	68.40%	85.05%
seeds	**76.72%**	**76.92%**	72.32%	43.46%
ecoli	**72.53%**	**71.14%**	NA	NA
yeast	**42.31%**	**35.00%**	NA	NA
wine	**84.54%**	84.54%	69.05%	**86.47%**
abalone	**50.12%**	**54.35%**	11.69%	27.64%
balance-scale	**83.45%**	**84.56%**	NA	3.05%
cmc	**46.47%**	**52.18%**	15.30%	19.87%
harberman	**41.09%**	**39.16%**	NA	NA
transfusion	**19.54%**	**24.53%**	NA	NA

In Table 2, we can see that, overall speaking, the F1-score of PU-LP for most of datasets is significantly better than that of PNB. For some datasets, the F1-score of PNB can't be measured, which is denoted as *NA*. The reason is that all examples (unlabeled examples) in the test set are classified as negative by PNB classifier, which causes both the precision and recall of the classifier being zero. In these datasets, when $d\% = 10\%$ and 20%, there is only small amount of positive examples in the labeled dataset P. Following the approach proposed in [13], the estimated priori probability for PNB of these datasets is a very small value, which makes the classifier classify all the examples into the negative class. This experiment result shows that PU-LP has excellent classification performance with only small amount of positive training examples, while PNB performs badly under this scenario.

5.3 Experiment with parameters

Here, we report our experiment result with parameter α, λ and m. Following this experiment, we set the parameter values for PU-LP algorithm. For lacking of space, only the experiment result on part of datasets is given here. The similar result can be got for the rest of the datasets.

Experiment with α: Here, we set α=0.10, 0.08, 0.06, 0.04, 0.02, and give precision of the extracted reliable positive examples. Due to lacking of space, experiment result for only three datasets (*Iris*, *ecoli* and *balance-scale*) is given here.

It is shown in Table 3 that with the decreasing of α, the precision of the extracted positive examples increases slightly. Hence, we set $\alpha = 0.02$ in the rest of our experiment.

Table 3. Experimental result with parameter α

Dataset	#Labeled examples /#Extracted examples	α	Precision of extracted examples
iris	5/10	0.10	98.00%
		0.08	98.50%
		0.06	98.50%
		0.04	99.00%
		0.02	**99.00%**
ecoli	7/14	0.10	76.36%
		0.08	78.00%
		0.06	80.79%
		0.04	**82.71%**
		0.02	82.36%
balance-scale	29/58	0.10	91.72%
		0.08	92.72%
		0.06	91.69%
		0.04	91.72%
		0.02	**93.26%**

Experiment with λ: In PU-LP, reliable positive examples are extracted from the unlabeled set to enlarge labeled positive set. Here, we study the impact of the size of RP on the performance of the final classifier. We have $|RP| = \lambda \times |P|$. Due to lacking of space, experiment result for only four datasets (*cmc*, *iris*, *ecoli* and *balance-scale*) is given here, and the result is measured by F1 index.

Table 4. Experiment result with parameter λ

Dataset	λ	F1	Dataset	λ	F1
cmc	0.2	54.51%	iris	0.2	92.49%
	0.6	54.67%		0.6	91.78%
	0.8	54.48%		0.8	92.63%
	1.0	54.35%		1.0	92.31%
	1.2	54.39%		1.2	91.88%
	1.4	**54.76%**		1.4	**92.65%**
	2.0	54.71%		2.0	92.14%
ecoli	0.2	72.93%	balance-scale	0.2	**86.06%**
	0.6	72.39%		0.6	85.13%
	0.8	**73.55%**		0.8	84.36%
	1.0	72.59%		1.0	83.38%
	1.2	72.63%		1.2	83.45%
	1.4	72.74%		1.4	82.89%
	2.0	72.97%		2.0	81.32%

In Table 4, with the increasing of parameter λ from 0.2 to 2.0, the F1 index of the classifier grows up slowly at first, and then falls down slowly. The reason is that with the increasing of |RP|, the noisy examples in RP also increase, which decrease the performance of the classifier. Hence, in this experiment, we set $\lambda = 0.8$.

Experiment with m: In PU-LP, the precision of extracted positive examples is improved by iteratively extracting

reliable positive examples from the unlabeled set. Table 5 shows the precision of the extracted positive examples with respect to the number of iterations. In this experiment, we set $d\% = 10\%$, Due to lacking of space, experiment result for only two datasets (*balance-scale* and *ecoli*) is given here. In Table 5, parameter m represents the number of iterations. There is no iteration when $m = 1$.

Table 5. Experimental result with parameter m

Dataset	λ	Precision (m=1)	Precision (m=2)	Precision (m=3)
balance-scale	0.6	95.12%	97.39%	95.50%
	0.8	94.09%	95.12%	96.50%
	1.0	92.41%	95.57%	95.27%
	1.2	92.34%	94.97%	95.06%
ecoli	0.6	82.80%	83.50%	87.17%
	0.8	87.75%	84.08%	85.00%
	1.0	81.69%	84.75%	85.06%
	1.2	83.20%	83.45%	83.67%

It is shown in Table 5 that more iterations will results in a better performance. However, the more iterations the algorithm takes, the more time it consumes. Considering when $m = 2$, the precision of extracted positive examples of PU-LP algorithm is relatively high. Therefore, in all our experiments, we set $m = 2$.

6. CONCLUSION AND FUTURE WORK

Existing PU learning algorithms can hardly train classifiers with good performance when only small amount of labeled positive examples available. In this paper, we propose PU-LP, a novel graph-based transductive PU learning algorithm. PU-LP is based on *same label* assumption, taking Katz index to measuring similarities between examples. The positive set is enlarged by extracting reliable positive examples from unlabeled set, and then the reliable negative example set is extracted. At last, label propagation algorithm is applied to train the final classifier. The experiment result on UCI datasets shows that PU-LP has better performance than PNB when there is only small amount of labeled positive examples available.

In the future, we will focus on the parameter selection, so as to build a system which can determine the parameter settings automatically.

7. REFERENCES

[1] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," *14th International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pp. 213-220, 2008.

[2] X. Zhu and Andrew B. Goldberg, "Introduction to Semi-Supervised Learning," Morgan & Claypool, pp. 43-55, 2009.

[3] T. Ke, J. Tan, B. Yang, L. Song, Y. Li and L. Jing, "A novel graph-based approach for transductive positive and unlabeled learning," *Journal of Computational Information Systems*, vol. 10, no. 4, pp. 1439-1447, 2014.

[4] B. Liu, WS. Lee, PS. Yu and X. Li "Partially Supervised Classification of Text Documents," *9th International Conference on Machine Learning (ICML 2002)*, pp. 387-394, 2002.

[5] H. Yu, J. Han and CC. Chang, "PEBL: positive example based learning for Web page classification using SVM," *8th International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, pp. 239-248, 2002.

[6] B. Calvo, P. Larrañaga and JA. Lozano, "Learning Bayesian classifiers from positive and unlabeled examples," *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2375-2384, 2007.

[7] S. Yu and C. Li, "PE-PUC: A Graph Based PU-Learning Approach for Text Classification," *5th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2007)*, pp. 574-584, 2007.

[8] L. Berlin and ALA. De, "Graph construction based on labeled instances for semi-supervised learning," *22nd International Conference on Pattern Recognition (ICPR 2014)*, pp. 2477-2482, 2014.

[9] D. Zhou, O. Bousquet, TN. Lal, J. Weston and BS. Olkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems*, vol. 16, no. 4, pp. 321-328, 2004.

[10] T. Zhou, L. Lü and YC. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623-630, 2009.

[11] L. Lü, CH. Jin and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Physical Review E*, vol. 80, no. 4, 046122, 2009.

[12] X. Zhu, Z. Ghahramani and JD. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *20th International Conference on Machine Learning (ICML 2003)*, pp. 912-919, 2003.

[13] J. He, Y. Zhang, X. Li and Y. Wang, "Bayesian Classifiers for Positive Unlabeled Learning," *12th International Conference on Web-Age Information Management (WAIM 2011)*, pp. 81-93, 2011.