Similarity-Based Approach for Positive and Unlabelled Learning

Yanshan Xiao[1,2], **Bo Liu**[3,4,2] *, **Jie Yin**[5], **Longbing Cao**[2], **Chengqi Zhang**[2], **Zhifeng Hao**[1]

[1]School of Computer, Guangdong University of Technology, Guangzhou, China

[2]Faculty of Engineering and IT, University of Technology, Sydney, NSW, Australia

[3]College of Automation Science and Engineering, South China University of Technology, Guangzhou, China

[4]School of Automation, Guangdong University of Technology, Guangzhou, China

[5]Information Engineering Laboratory, CSIRO ICT Centre, Australia

{xiaoyanshan; csbliu}@gmail.com; jie.yin@csiro.au; {lbcao; chengqi}@it.uts.edu.au; mazfhao@scut.edu.cn

Abstract

Positive and unlabelled learning (PU learning) has been investigated to deal with the situation where only the positive examples and the unlabelled examples are available. Most of the previous works focus on identifying some negative examples from the unlabelled data, so that the supervised learning methods can be applied to build a classifier. However, for the remaining unlabelled data, which can not be explicitly identified as positive or negative (we call them *ambiguous examples*), they either exclude them from the training phase or simply enforce them to either class. Consequently, their performance may be constrained. This paper proposes a novel approach, called similarity-based PU learning (SPUL) method, by associating the ambiguous examples with two similarity weights, which indicate the similarity of an ambiguous example towards the positive class and the negative class, respectively. The local similarity-based and global similarity-based mechanisms are proposed to generate the similarity weights. The ambiguous examples and their similarity-weights are thereafter incorporated into an SVM-based learning phase to build a more accurate classifier. Extensive experiments on real-world datasets have shown that SPUL outperforms state-of-the-art PU learning methods.

1 Introduction

Traditional supervised learning methods require that both the positive and negative examples are available for training. However, in many real-world applications [Fung *et al.*, 2006], it is not easy to obtain the negative examples. For example, in Web page classification, the users may mark their favorite Web pages, but they are unwilling to mark the boring pages that they show no preference. Therefore, the positive and unlabelled learning (PU learning) is studied to deal with the situation where only the positive examples and the unlabelled examples are available in the training phase [Liu *et al.*, 2002].

In the PU learning, since the negative examples are unavailable, most of the existing works [Liu *et al.*, 2002; Yu *et al.*, 2004; Li and Liu, 2003; Liu *et al.*, 2003; Li *et al.*, 2009] focus on identifying some reliable negative examples from the unlabelled examples, so that the supervised learning methods can be applied to build the classifier. However, there may exist a number of unlabelled examples which can not be explicitly identified as positive or reliable negative (we call them *ambiguous examples* here). Compared to the examples which can be clearly classified to be positive or negative, the ambiguous examples are more likely to lie near the decision boundary and play a critical role in building the classifier. Therefore, it is essential to appropriately deal with the ambiguous examples in order to learn an accurate classifier.

Considering the existing PU learning works, different strategies have been proposed to deal with the ambiguous examples. Since the labels of ambiguous examples are difficult to be determined, one group of works [Liu *et al.*, 2002; Yu *et al.*, 2004; Li and Liu, 2003; Liu *et al.*, 2003] excludes the ambiguous examples from the learning phase, and the classifier is trained by using only the positive and some negative examples. For example, Spy-EM (Spy Expectation Maximization) [Liu *et al.*, 2002] uses a spy technique to identify some reliable negative examples from the unlabelled examples, and then EM is run to build the classifier by using the positive examples and the extracted negative examples. However, the classification ability of these methods may be limited, since the ambiguous examples, which can contribute to the classifier construction, are excluded from the learning process.

Another group of works includes the ambiguous examples in learning the classifier by straightforwardly assigning them to the positive class or the negative class. For example, in LELC [Li *et al.*, 2009], the ambiguous examples are clustered into micro-clusters. For each micro-cluster, the distances from its examples to the positive prototypes and the identified negative prototypes are calculated. Based on a voting strategy, the micro-cluster (including all its examples) is assigned to the class which the micro-cluster is closer to. By considering the ambiguous examples, LELC performs better than other PU learning methods [Li *et al.*, 2009]. However, in LELC, there may exist some micro-clusters, in which some examples are biased towards the positive class, while the others are closer to the negative class. In such case, if we simply

*Bo Liu is the corresponding author.

enforce the whole micro-cluster of examples to any of the two classes, misclassification may be incurred.

In this paper, we propose a novel approach, called similarity-based PU learning (SPUL), by utilizing the ambiguous examples as an effective way to improve the PU learning classification accuracy. Instead of eliminating the ambiguous examples from the learning phase or enforcing the ambiguous examples directly to one class, our proposed approach explicitly deals with the ambiguous examples by considering their similarity towards both of the positive class and the negative class. Specifically, our proposed approach works in three steps. In the first step, we extract reliable negative examples from the unlabelled data and build the representative positive and negative prototypes. In the second step, we cluster the remaining unlabelled examples into micro-clusters and assign each example with two similarity weights, which indicate the similarity of an ambiguous example towards the positive class and the negative class, respectively. To do this, the local similarity-based and global similarity-based mechanisms are proposed to generate the similarity weights. In the third step, we extend the standard support vector machine (SVM) to incorporate the ambiguous examples with their similarity weights into a learning phase, such that the ambiguous examples can contribute differently on the classifier construction based on their similarity weights. Extensive experiments have been conducted to investigate the performance of SPUL and the statistical results show that SPUL outperforms state-of-the-art PU learning methods.

2 Related Work

2.1 Positive and Unlabelled Learning

In recent years, the PU learning has found various applications in text mining due to the fact that collecting a large set of negative documents is always expensive and challenging [Li *et al.*, 2007; Liu *et al.*, 2002; Yu *et al.*, 2004; Lee and Liu, 2003; Li and Liu, 2003; Liu *et al.*, 2003; Zhou *et al.*, 2010; Scholkopf *et al.*, 2001]. We briefly review the existing works on PU learning in the following.

The first group of works [Liu *et al.*, 2002; Yu *et al.*, 2004; Li and Liu, 2003; Liu *et al.*, 2003] adopts an iterative framework to extract the negative examples from the unlabelled examples, and train the classifier alternatively. For example, Spy-EM (Spy Expectation Maximization) [Liu *et al.*, 2002] uses a Spy technique to extract the negative examples, and EM algorithm is used to train the classifier iteratively. Roc-SVM (Rocchio-Support Vector Machine) [Li and Liu, 2003] extracts the reliable negative examples by using the information retrieval technique Rocchio [Rocchio, 1971]. In this category, except for positive examples and the extracted examples, the rest ambiguous examples are excluded from the training process. Therefore, the performance may be limited.

The second group of work does not include the iterative framework. For example, one-class classification method [Scholkopf *et al.*, 2001] is proposed to build an one-class classifier by using only the positive examples. Since the unlabelled data information is not considered, the one-class classifier is always inferior to the binary classification-based methods [Li and Liu, 2003]. Another example is LELC [Li *et al.*, 2009]. LELC clusters the ambiguous examples into micro-clusters, and then assigns a whole micro-cluster of examples to the class which the micro-cluster is closer to. However, there may be some micro-clusters in which some examples are biased to the positive class and the other examples are closer to the negative class. In such case, enforcing the micro-cluster to any of the two classes may lead to misclassification.

In this paper, we propose a similarity-based PU learning method. Compared to the works in the first group, our proposed method explicitly incorporates the ambiguous examples to improve the classification accuracy of PU learning. Furthermore, rather than enforcing the ambiguous examples to either class, as some of the second group's works do, we incorporate the ambiguous examples in the training by measuring their similarity to the positive class and the negative class, such that the classification boundary can be refined based on the similarity information.

2.2 Support Vector Machine

SVM [Vapnik, 1998] has been proven to be a powerful classification tool. We briefly review SVM as follows.

Let $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{|S|}, y_{|S|})\}$ be a training set, where $\mathbf{x}_i \in R^d$ and $y_i \in \{+1, -1\}$. SVM aims at seeking an optimal separating hyperplane $\mathbf{w} \cdot \phi(\mathbf{x}) + b = 0$, where $\phi(\mathbf{x})$ is the image of example $\mathbf{x}$ in the feature space. The optimal separating hyperplane can be obtained by solving the following optimization function:

$$\begin{aligned} \min \quad & F(\mathbf{w}, b, \xi_i) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\frac{1}{2}\sum_{i=1}^{|S|} \xi_i \\ st. \quad & y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i > 0, \;\; i = 1, \ldots, |S|. \end{aligned} \tag{1}$$

where ξ_i are variables to relax the margin constraints, and C is a parameter to balance the classification errors. By introducing the Lagrange function [Vapnik, 1998], the decision classifier can be obtained. For a test example $\mathbf{x}$, if $\mathbf{w} \cdot \phi(\mathbf{x}) + b > 0$, it is classified into the positive class; otherwise, it is negative.

In the following, we will extend SVM to incorporate the examples with similarity weights into a learning phase, such that the ambiguous examples can contribute differently to the classifier construction.

3 Preliminary

Let S be a training set of a PU learning problem. Assume that PS and US store the positive examples and the unlabelled examples, respectively. Hence, we have $S = PS \cup US$.

For the ambiguous examples, since we do not know which class it should belong to, we represent an ambiguous example $\mathbf{x}$ using a similarity-based data model:

$$\{\mathbf{x}, (m^+(\mathbf{x}), m^-(\mathbf{x}))\}, \tag{2}$$

where $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ are similarity weights which represent the similarity of $\mathbf{x}$ towards the positive class and the negative class, respectively. We have $0 \leq m^+(\mathbf{x}) \leq 1$ and $0 \leq m^-(\mathbf{x}) \leq 1$. $\{\mathbf{x}, (1, 0)\}$ means that $\mathbf{x}$ is positive, while $\{\mathbf{x}, (0, 1)\}$ indicates that $\mathbf{x}$ is identified to be negative. For $\{\mathbf{x},\ (m^+(\mathbf{x}),\ m^-(\mathbf{x}))\}$, where $0 < m^+(\mathbf{x}) < 1$ and

$0 < m^-(\mathbf{x}) < 1$, it implies that the similarity of $\mathbf{x}$ towards the positive class and the negative class is both considered.

By using the similarity-based data model, we can generate similarity weights for the ambiguous examples based on the positive and extracted negative examples. These ambiguous examples and their similarity weights are thereafter incorporated into an SVM-based learning model.

4 Similarity-Based PU Learning Approach

In this section, we will introduce the proposed approach in details. The PU learning aims at constructing a classifier by using the positive examples and the unlabelled examples. It has been found various applications in text mining area. Based on the similarity-based data model introduced in Section 3, our proposed SPUL approach works in the following three steps.

1. In the first step, we extract the reliable negative examples and build the representative positive and negative prototypes.
2. In the second step, we cluster the remaining unlabelled data (ambiguous examples) into micro-clusters and assign similarity weights to the ambiguous examples. The local similarity-based and global similarity-based mechanisms are proposed to generate the similarity weights.
3. In the third step, we extend the standard SVM to incorporate the ambiguous examples and their similarity weights into the learning phase to build a more accurate classifier.

In the following, we present the detailed information of the above three steps.

4.1 Step 1: Negative Example Extraction

In the first step, we extract the reliable negative examples and put them in subset NS. Together with the positive examples, those negative examples are used to set up the representative positive prototypes and negative prototypes.

First of all, we extract the reliable negative examples from the unlabelled data. As LELC, we integrate the Spy technique [Liu *et al.*, 2002] and the Rocchio technique [Li and Liu, 2003] to extract the most reliable negative examples. Let subsets S_1 and S_2 contain the corresponding reliable negative examples extracted by the Spy technique and the Rocchio technique. Examples are classified as reliable negative only if both techniques agree that they are negative. That is, $NS = S_1 \cap S_2$, where subset NS contains the reliable negative examples. After the reliable negative examples are determined, we get rid of them from the unlabelled data subset, i.e., $US = US - NS$.

Furthermore, the representative positive prototypes and the representative negative prototypes are then set up by clustering the reliable negative examples into micro-clusters. Specifically, K-mean clustering is used to cluster the examples in NS into m micro-clusters, denoted as $NS_1, NS_2, \ldots, NS_m$ where $m = t * |NS|/(|US| + |NS|)$ and t is set to be 30 in the experiments, as recommended in [Buckley *et al.*, 1994; Li *et al.*, 2009]. Then, the k^{th} representative positive prototype, denoted as $\mathbf{p}_k$, and the k^{th} representative negative prototype, denoted as $\mathbf{n}_k$, are built as follows:

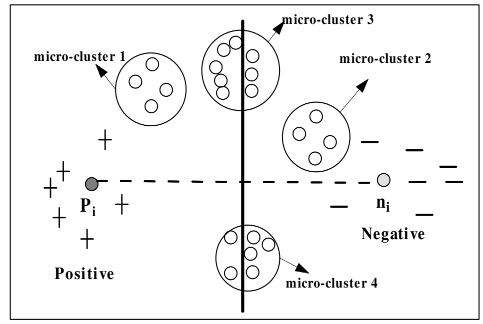

Figure 1: Illustration of similarity weight assignment to the micro-clusters in the local generation scheme. "+" represents the positive example. "-" denotes the reliable negative example. "o" stands for the ambiguous example.)

$$\mathbf{p}_k = \alpha \frac{1}{|PS|} \sum_{\mathbf{x} \in PS} \frac{\mathbf{x}}{\|\mathbf{x}\|} - \beta \frac{1}{|NS_k|} \sum_{\mathbf{x} \in NS_k} \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad (3)$$

$$\mathbf{n}_k = \alpha \frac{1}{|NS_k|} \sum_{\mathbf{x} \in NS_k} \frac{\mathbf{x}}{\|\mathbf{x}\|} - \beta \frac{1}{|PS|} \sum_{\mathbf{x} \in PS} \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad (4)$$

$$k = 1, \ldots, m.$$

where $\| \mathbf{x} \|$ represents the norm of example $\mathbf{x}$; parameters α and β are set to be 16 and 4, respectively, as recommended in [Buckley *et al.*, 1994; Li *et al.*, 2009].

After Step 1, we obtain the reliable negative examples in NS, m representative positive prototypes $\mathbf{p}_k$ and m representative negative prototypes $\mathbf{n}_k$.

4.2 Step 2: Similarity Weight Generation

In this step, we aim at creating two similarity weights $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ for the examples in subset US, such that the ambiguous examples can be incorporated into the training phase by considering their similarity to the positive class and the negative class. Specifically, we first cluster the examples in subset US into r micro-clusters, i.e., $US_1, US_2, \ldots, US_r$, where r is set as $r = t * |US|/(|US| + |NS|)$ and again we set t to be 30 in the experiments. Then, the similarity weights $m^+(\mathbf{x}_i)$ and $m^-(\mathbf{x}_i)$ are generated for each example in subsets US_i $(i = 1, \ldots, r)$. To generate the similarity weights, we put forward the local similarity-based and global similarity-based schemes in the following.

Local Similarity Weight Generation Scheme

In this scheme, we generate the similarity weights by capturing the local data information. For each micro-cluster US_j $(j = 1, 2, \ldots, r)$, we assume that there are l_p^j examples similar to the closest positive prototype $\mathbf{p}_k$, and l_n^j examples similar to the closest negative prototype $\mathbf{n}_k$. That is, for the l_p^j examples, we have

$$\max_{k=1}^{m} Sim(\mathbf{x}, \mathbf{p}_k) > \max_{k=1}^{m} Sim(\mathbf{x}, \mathbf{n}_k) \quad (5)$$

where $Sim(.,.)$ is calculated as $Sim(\mathbf{x}, \mathbf{p}_k) = \frac{\mathbf{x} \cdot \mathbf{p}_k}{\|\mathbf{x}\| \cdot \|\mathbf{p}_k\|}$. Similarly, for the l_n^j examples, we have

$$\max_{k=1}^{m} Sim(\mathbf{x}, \mathbf{p}_k) < \max_{k=1}^{m} Sim(\mathbf{x}, \mathbf{n}_k). \quad (6)$$

Based on the above functions, the similarity weights for ambiguous data in US_j are calculated as

$$m^+(\mathbf{x}_i) = \frac{l_p^j}{l_p^j + l_n^j}, \mathbf{x}_i \in US_j \quad (7)$$

$$m^-(\mathbf{x}_i) = \frac{l_n^j}{l_p^j + l_n^j}, \mathbf{x}_i \in US_j \quad (8)$$

Figure 1 presents an example of assigning the similarity weights to ambiguous data. Based on Equations (7) and (8), the examples in micro-clusters 1, 2, 3 and 4 are assigned with weights $(1, 0)$, $(0, 1)$, $(\frac{5}{8}, \frac{3}{8})$ and $(\frac{1}{3}, \frac{2}{3})$, respectively. Distinguished from LELC, which directly assigns a whole micro-cluster of examples to one class, SPUL allows the ambiguous examples having different weights associated with the positive class and the negative class, such that the similarity of ambiguous examples towards the two classes can be considered. The advantage of the local generation scheme is that it is simple to implement. However, it can not distinguish the difference of examples in the same micro-cluster. The examples from the same micro-cluster have exactly the same weights towards the two classes. In fact, the similarity weights of examples from the same micro-cluster can be different, since they are located physically different.

Global Similarity Weight Generation Scheme

To consider the location of ambiguous examples, we further propose a global generation scheme to assign weights to ambiguous examples.

For the ambiguous example $\mathbf{x}_i$ in subset US, we first calculate its similarity to each of the representative positive and negative prototypes. That is,

$$Sim(\mathbf{x}_i, \mathbf{p}_k) = \frac{\mathbf{x}_i \cdot \mathbf{p}_k}{\|\mathbf{x}_i\| \cdot \|\mathbf{p}_k\|}, \quad k = 1, 2, \ldots, m \quad (9)$$

$$Sim(\mathbf{x}_i, \mathbf{n}_k) = \frac{\mathbf{x}_i \cdot \mathbf{n}_k}{\|\mathbf{x}_i\| \cdot \|\mathbf{n}_k\|}, \quad k = 1, 2, \ldots, m. \quad (10)$$

For $\mathbf{x}_i \in US$, the corresponding weights towards the positive class and the negative class are computed as follows:

$$m^+(\mathbf{x}_i) = \frac{\sum_{k=1}^m Sim(\mathbf{x}_i, \mathbf{p}_k)}{\sum_{k=1}^m (Sim(\mathbf{x}_i, \mathbf{p}_k) + Sim(\mathbf{x}_i, \mathbf{n}_k))}, \quad (11)$$

$$m^-(\mathbf{x}_i) = \frac{\sum_{k=1}^m Sim(\mathbf{x}_i, \mathbf{p}_k)}{\sum_{k=1}^m (Sim(\mathbf{x}_i, \mathbf{p}_k) + Sim(\mathbf{x}_i, \mathbf{n}_k))}. \quad (12)$$

The global generation scheme treats each ambiguous example in subset US differently and the weights are calculated based on the locations of examples towards the representative positive and negative prototypes, respectively. As shown in the experiments, the global generation scheme outperforms the local generation scheme.

4.3 Step 3: SVM-Based Classifier Construction

After performing the above two steps, each ambiguous example is assigned two similarity weights: $m^+(\mathbf{x}_i)$ and $m^-(\mathbf{x}_i)$. In the following, we will give a novel formulation of SVM by incorporating the data in positive set PS, negative set NS, ambiguous example set US and the similarity weights into an SVM-based learning model.

Primal Formulation

Since the similarity weights $m^+(\mathbf{x}_i)$ and $m^-(\mathbf{x}_i)$ indicate the different degrees of similarity for an ambiguous example towards the positive class and the negative class, respectively, the optimization function can be formulated as follows:

$$\begin{aligned}
\min \quad & F(\mathbf{w}, b, \xi) \\
& = \tfrac{1}{2}\mathbf{w} \cdot \mathbf{w} + C_1 \textstyle\sum_{PS} \xi_i + C_2 \sum_{US} m^+(\mathbf{x}_j)\xi_j \\
& \quad + C_3 \textstyle\sum_{US} m^-(\mathbf{x}_k)\xi_k + C_4 \sum_{NS} \xi_g \\
s.t. \quad & \mathbf{w} \cdot \phi(\mathbf{x}_i) + b \geq 1 - \xi_i, \qquad \mathbf{x}_i \in PS \\
& \mathbf{w} \cdot \phi(\mathbf{x}_j) + b \geq 1 - \xi_j, \qquad \mathbf{x}_j \in US \\
& \mathbf{w} \cdot \phi(\mathbf{x}_k) + b \leq -1 + \xi_k, \quad \mathbf{x}_k \in US \\
& \mathbf{w} \cdot \phi(\mathbf{x}_g) + b \leq -1 + \xi_g, \quad \mathbf{x}_g \in NS \\
& \xi_i \geq 0,\ \xi_j \geq 0,\ \xi_k \geq 0,\ \xi_g \geq 0,
\end{aligned} \quad (13)$$

where C_1, C_2, C_3 and C_4 are penalty factors controlling the tradeoff between the hyperplane margin and the errors. ξ_i, ξ_j, ξ_k and ξ_g are the error terms. $m^+(\mathbf{x}_j)\xi_j$ and $m^-(\mathbf{x}_k)\xi_k$ can be considered as errors with different weights. Note that, a smaller value of $m^+(\mathbf{x}_i)$ can reduce the effect of parameter ξ_i, so that the corresponding example $\mathbf{x}_i$ becomes less significant towards the positive class.

Dual Problem

Assume that α_i, α_j, α_k and α_g are Lagrange multipliers. To simplify the presentation, we redefine some notations in the following:

$$\alpha_i^+ = \begin{cases} \alpha_i, & \mathbf{x}_i \in PS \\ \alpha_j, & \mathbf{x}_j \in US \end{cases} \quad C_i^+ = \begin{cases} C_1, & \mathbf{x}_i \in PS \\ C_2 m^+(\mathbf{x}_j), & \mathbf{x}_j \in US \end{cases}$$

$$\alpha_j^- = \begin{cases} \alpha_k, & \mathbf{x}_k \in US \\ \alpha_g, & \mathbf{x}_g \in NS \end{cases} \quad C_j^- = \begin{cases} C_2 m^-(\mathbf{x}_k), & \mathbf{x}_k \in US \\ C_3, & \mathbf{x}_g \in NS \end{cases}$$

Based on the above definitions, we let $S_+ = PS \cup US$, $S_- = US \cup NS$ and $S_* = S_+ \cup S_-$. The Wolfe dual of (13) can be obtained as follows:

$$\begin{aligned}
\max \ & F(\alpha) = \textstyle\sum_{\mathbf{x}_i \in S_*} \alpha_i - \frac{1}{2} \sum_{\mathbf{x}_i, \mathbf{x}_j \in S_*} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\
s.t. \quad & 0 \leq \alpha_i \leq C_i^+, \quad \mathbf{x}_i \in S_+ \\
& 0 \leq \alpha_j \leq C_j^-, \quad \mathbf{x}_j \in S_- \\
& \textstyle\sum_{\mathbf{x}_i \in S_+} \alpha_i - \sum_{\mathbf{x}_j \in S_-} \alpha_j = 0,
\end{aligned} \quad (14)$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the inner product of $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$.

After solving the problem in (14), $\mathbf{w}$ can be obtained in the following:

$$\mathbf{w} = \textstyle\sum_{\mathbf{x}_i \in S_+} \alpha_i^+ \phi(\mathbf{x}_i) - \sum_{\mathbf{x}_j \in S_-} \alpha_j^- \phi(\mathbf{x}_j). \quad (15)$$

By using Karush-Kuhn-Tucker conditions (KKT) [Vapnik, 1998], b can be obtained. For a test example, if $f(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b > 0$ holds true, it belongs to the positive class. Otherwise, it is negative.

5 Experiment

All the experiments are performed on a laptop with a 2.8 GHz processor and 3GB DRAM.

5.1 Baselines and Metrics

We implement two variants of our proposed method, i.e., local similarity-based PU learning (Local SPUL) and global similarity-based PU learning (Global SPUL). For comparison, another three methods are used as baselines. The first one is Spy-EM [Liu *et al.*, 2002], which uses Spy technique

Table 1: Average F-measure values on the nine sub-datasets.

Data Subset	Global SPUL	Local SPUL	LELC	Spy-EM	Roc-SVM
Reuter-interest	0.597	0.556	0.543	0.335	0.496
Reuter-trade	0.595	0.574	0.552	0.348	0.514
Reuter-ship	0.632	0.607	0.596	0.537	0.588
WebKB-faculty	0.473	0.446	0.417	0.314	0.302
WebKB-course	0.449	0.426	0.407	0.376	0.366
WebKB-project	0.353	0.342	0.325	0.305	0.299
Newsgroups-mac.hardware-crypt	0.535	0.512	0.494	0.483	0.487
Newsgroups-graphic-space	0.672	0.658	0.626	0.614	0.605
Newsgroups-os-med	0.594	0.572	0.557	0.527	0.516

to extract negative examples and utilizes EM to construct the classifier. The second one is Roc-SVM [Li and Liu, 2003], which employs Rocchio method to extract the negative examples and builds an SVM classifier. Both methods exclude the ambiguous examples from the training. The third one is LELC [Li *et al.*, 2009], which clusters the ambiguous examples into micro-clusters and assigns each micro-cluster to either the positive class or the negative class. The third baseline is used to demonstrate the capability of our method in coping with the ambiguous examples.

The performance of text classification is typically evaluated based on F-measure [Liu *et al.*, 2002]. F-measure trades off the precision p and the recall r: $F = \frac{2pr}{r+p}$. Only when both are large will F-measure exhibit a large value. A desirable algorithm should have a F-measure value closer to one.

5.2 Datasets and Settings

To evaluate the properties of our approaches, we conduct experiments on three real-world datasets:

- Reuters-21578 [1]: This dataset contains 21578 documents. Since it is highly skewed, we follow the same operations in [Fung *et al.*, 2006] to select the top 10 largest categories, i.e., "acq", "corn", "crude", "earn", "grain", "interest", "money", "ship", "trade" and "wheat". In all, we have 9981 documents for the experiments.
- 20 Newsgroups [2]: There are 20 sub-categories and each sub-category has 1000 messages. For a fair comparison, we have removed all the UseNet headers, which contain the information of subject lines.
- WebKB [3]: It has 8282 Web pages and 7 categories. The dataset is slightly skewed. The number of Web pages in different categories ranges from 3764 to 137.

For each of the above datasets, we conduct the following operations to obtain sub-datasets for PU learning. We choose a category (a) from a dataset (A), and randomly select g percentage of examples from this category (a) to form a positive example set. The remaining examples in category (a) and the examples from other categories are used to form an unlabelled dataset.

By considering each category as the positive class, we obtain 10 sub-datasets from Reuters-21578, 7 sub-datasets from WebKB and 20 sub-datasets from 20 Newsgroups. In addition, since the sizes of some categories are small, e.g., "corn" category in Reuters-21578 only containing 238 examples, we first set $g = 15\%$, and then investigate the performance of each method when g increases.

In the experiments, the linear kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ is used, since it generally performs well for text classification [Sebastiani, 2002]. In our Local SPUL and Global SPUL methods, we let C_1, C_2, C_3 and C_4 range from 2^{-5} to 2^5. Moreover, t is set to be 30, as recommended in [Li *et al.*, 2009]. For the parameters contained in Spy-EM [Liu *et al.*, 2002], Roc-SVM [Li and Liu, 2003] and LELC [Li *et al.*, 2009], we adopt the settings in their own works.

5.3 Experimental Results

For each generated sub-dataset, we randomly select sixty percent of data to form a training set, and the remaining data forms a testing set. 10-fold cross validation is conducted on the test set. To avoid sampling bias, we repeat the above process for 10 times, and calculate the average F-measure values for each sub-dataset. Since there are thirty two sub-datasets, due to limited space, we only show the average F-measure values on nine sub-datasets, as reported in Table 1. In Table 1, the listed sub-datasets are denoted as "A-a" format, where "A" denotes the original dataset's name, and "a" represents the category which is used as the positive class. Note that, the reported results are all consistent on the other sub-datasets.

As shown in Table 1, both of Global SPUL and Local SPUL outperform the other baselines on the nine sub-datasets. This is because our SPUL methods assigns similarity weights to the ambiguous examples, so that the similarity of ambiguous examples to the positive and negative classes can be evaluated and contribute to the classifier construction. In terms of our two variants Global SPUL and Local SPUL, Global SPUL generally performs better than Local SPUL on all the sub-datasets, since the former one generates similarity weights based on the locations of ambiguous samples towards the positive and negative prototypes, but the later one just considers the votes of a micro-cluster towards the positive and negative prototypes.

We further discover that, Spy-EM and Roc-SVM always obtain lower accuracy, since they discard the ambiguous samples from training. Consequently, they are inferior to the other methods. This is consistent with the findings in [Li *et al.*, 2009]. Furthermore, LELC method performs better than Spy-EM and ROC-SVM, but worse than our methods. This

[1] http://www.daviddlewis.com/resources/testcollections/

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

[3] www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data

Table 2: Overall F-measure values on the three real-world datasets.

Baselines	Reuter	WebKB	Newsgroups
Global SPUL	0.616	0.449	0.557
Local SPUL	0.591	0.418	0.534
LELC	0.562	0.401	0.505
Spy-EM	0.443	0.365	0.478
Roc-SVM	0.527	0.326	0.493

is because, though LELC takes the ambiguous samples into learning the classifier, it simply enforces the micro-cluster into either class, and misclassification may be incurred if some of its examples are more biased towards the positive class and the other examples are closer to the negative class.

In the above, we report the detailed results on the nine sub-datasets. In the following, the average F-measure values of the sub-datasets from the same dataset are also reported, as shown in Table 2. Here, we can have similar findings to the nine sub-datasets.

Moreover, Figure 2 illustrates the performance of each method on the Reuter-interest sub-dataset when the percentage g of positive examples increases. As shown in Figure 2, when g increases, the F-measure value of each method increases. At the same time, it is observed that our Local SPUL and Global SPUL methods consistently outperform the other baselines.

6 Conclusions and Future Work

This paper proposes a similarity-based PU learning (SPUL) method to handle the ambiguous examples by associating them with two weights which imply the similarity of the ambiguous examples towards the positive class and the negative class, respectively. The local similarity-based and global similarity-based methods are presented to generate the similarity weights for ambiguous examples. We then extend the standard SVM to incorporate the ambiguous examples and their similarity weights into an SVM-based learning phase. Extensive experiments have shown the good performance of our method.

In the future, we plan to exploit an online process to learn the SPUL classifier in the streaming data environment.

Acknowledgments

This work is supported by QCIS Center and Advanced Analytics Institute of UTS, Australian Research Council (DP1096218, DP0988016, LP100200774 and LP0989721), Natural Science Foundation of China (61070033), Natural Science Foundation of Guangdong province (9251009001000005), Science and Technology Planning Project of Guangdong Province (2010B050400011,2010B080701070,2008B080701005), Eleventh Five-year Plan Program of the Philosophy and Social Science of Guangdong Province (08O-01), Opening Project of the State Key Laboratory of Information Security (04-01), China Post-Doctoral First-Class Fund (L1100010) and South China University of Technology Youth Fund (x2zdE5090780).

References

[Buckley *et al.*, 1994] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 292–300, 1994.

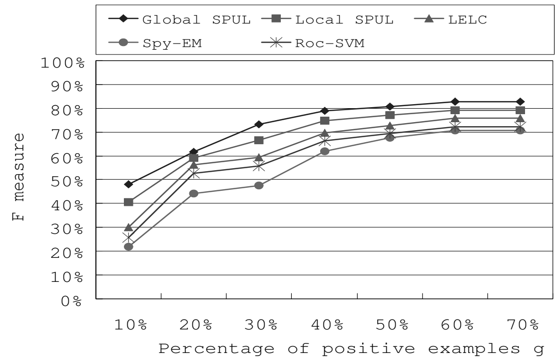

Figure 2: The performance on the Reuter-interest sub-dataset when the percentage of positive examples (g) increases.

[Fung *et al.*, 2006] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu. Text classification without negative examples revisit. *TKDE*, 18:6–20, 2006.

[Lee and Liu, 2003] W. S. Lee and B. Liu. Learning with positive and unlabeled examples using weighted logistic regression. *ICML*, 2003.

[Li and Liu, 2003] X. Li and B. Liu. Learning to classify texts using positive and unlabeled data. *IJCAI*, pages 587–592, 2003.

[Li *et al.*, 2007] X. Li, B. Liu, and S. K. NG. Learning to classify documents with only a small positive training set. In *ECML*, 2007.

[Li *et al.*, 2009] X. L. Li, P. S. Yu, B. Liu, and S. K. NG. Positive unlabeled learning for data stream classification. *SDM*, pages 257–268, 2009.

[Liu *et al.*, 2002] B. Liu, W. S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. *ICML*, 2002.

[Liu *et al.*, 2003] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. *ICDM*, pages 179–186, 2003.

[Rocchio, 1971] J. Rocchio. Relevance feedback in information retrieval. *In G. Salton (ed), The SMART retrieval system: Experiments in automatic document processing. Prentice Hall, Englewood Cliffs, NJ*, 1971.

[Scholkopf *et al.*, 2001] B. Scholkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[Sebastiani, 2002] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Vapnik, 1998] V. Vapnik. Statistical learning theory. *Springer*, 1998.

[Yu *et al.*, 2004] H. Yu, J. Han, and K. C. C. Chang. Pebl: web page classification without negative examples. *TKDE*, 16(1):70–81, 2004.

[Zhou *et al.*, 2010] K. Zhou, G. Xue, and Q. Yang. Learning with positive and unlabeled exmples using topic-sensitive plsa. *TKDE*, 22(1):28–29, 2010.