Computer Science and Application 计算机科学与应用, 2018, 8(5), 788-797
Published Online May 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.85088

Hans汉斯

PU Text Classification Method Based on Similarity under Data Distribution Estimation

Xuegang Hu, Lu Zhang, Peipei Li

School of Computer and Information, Hefei University of Technology, Hefei Anhui
Email: zhangl541@mail.hfut.edu.cn

Received: May 7th, 2018; accepted: May 22nd, 2018; published: May 29th, 2018

Abstract

In actual applications, due to various reasons, it is usually impossible to obtain the marked negative data, which causes the traditional classification algorithm to fail. Based on positive data and unlabeled data learning, it is called the PU classification problem. The key of the PU problem lies in the extraction of negative data and the construction of effective classifiers. The algorithm proposed in this paper firstly evaluates the data distribution in the sample, adopts the integration mechanism to extract positive and negative example data from the unlabeled sample with reasonable proportion, and then uses similarity to extract the representative positive micro-clusters and negative micro-clusters. After sufficient samples of positive and negative samples were obtained, the PU problem was converted to a binary classification problem. Numerical experiments showed the effectiveness of the method.

Keywords

PU Learning, Text Classification, Multiple Kernel Learning

数据分布估计下基于相似度的PU文本分类方法

胡学钢，张　路，李培培

合肥工业大学计算机与信息学院，安徽　合肥
Email: zhangl541@mail.hfut.edu.cn

收稿日期：2018年5月7日；录用日期：2018年5月22日；发布日期：2018年5月29日

文章引用：胡学钢, 张路, 李培培. 数据分布估计下基于相似度的 PU 文本分类方法[J]. 计算机科学与应用, 2018, 8(5): 788-797. DOI: 10.12677/csa.2018.85088

摘　要

在实际的应用中，由于各种原因通常无法获取已标注的反例数据，这使得传统分类算法失灵，这一类基于正例数据与未标注数据的学习称为PU分类问题。PU问题的关键在于反例样本提取与有效分类器的构建。本文提出算法首先通过评估样本中数据分布情况，采用集成机制从未标注样本中抽取出合适比例可信的正反例数据，其次利用相似度抽取有代表性的正例微簇和反例微簇，在获取足量的正反例样本后，将PU问题转换为二元分类问题，数值实验表明方法的有效性。

关键词

PU学习，文本分类，多核学习

1. 引言

文本分类问题是数据挖掘最重要的任务之一。现有的文本分类任务通常遵循一个通用的框架：从训练样本学习一个模型，然后利用该模型预测新的实例。现有的框架依赖以下假设：训练数据标签必须全部进行标注，并且训练标签集合必须覆盖被预测的实例类别。但是在实际应用中，我们经常会收集到这一类型的数据：只有有标注的正例样本，而其余剩下的大量样本数据均为未标注数据。这类问题被称为PU 分类[1]问题。目前已经有许多国内外学者对 PU 分类问题进行了研究，提出一系列新的分类方法。主要可概括为两类：1) 仅使用训练集中的正例，完全忽略未标注数据集。这类方法又称之为单类分类方法[2] [3]，这类方法的核心思想是构造一个近似覆盖训练集的最小区域，而位于区域之外的实例都属于反类。2) 使用正例与未标注数据集中的部分样例来构建最终的分类器[4] [5] [6] [7]。这类方法的核心思想是利用正例样本识别未标注样本中的可信度较高的反例样本，然后基于正例样本与反例样本迭代使用 EM 算法或者 SVM 算法建立最终的分类器。

Hu 等人[8]提出一种可以评估未标注数据中数据分布的算法 Auto-KL，其核心思想是先利用当前分类任务所提供的数据样本生成不同数据分布比例的模拟数据，然后根据模拟数据训练数据分布评估分类模型，最后根据当前数据进行数据分布估计。从而保证从未标注数据中抽取的足够多的有用信息，同时也尽量避免抽取到错误信息。但是剩余的未标注数据中仍有大量数据未被利用，从而使训练出来的分类器泛化能力不强。

针对上述问题，本文提出了在 Auto-KL 算法的基础上改进提出了一种数据分布估计下基于相似度的 PU 文本分类算法 Auto-KLBS (Auto-KL Based Similarity)。首先通过评估样本中数据分布情况，采用集成机制从未标注样本中抽取出合适比例可靠反例数据，其次利用相似度抽取有代表性的正例微簇和反例微簇，在获取足量的正反例样本后，将 PU 问题转换为二元分类问题。数值实验表明方法的有效性。

2. 研究现状

近年来，国内外学者针对 PU 分类问题取得了一系列的成果，其中根据对未标注数据集的使用情况可以概括以下 2 类方法。

1) 忽略未标注数据集，例如 Tax 等人[2]提出的 SVDD 方法，Mabevtz 等人[3]提出了一种单类 SVM [6]方法用于文本分类中。这类方法由于完全忽略了未标注数据集，从而丢失了未标注数据集中隐藏的有用信息，例如未标注数据集中存在可靠的反例样本时，忽略这类信息极易出现过拟合的现象。

2) 利用未标注数据集中有效信息增强训练模型。针对第 1 类方法的不足，显而易见的方法是考虑将无标签数据加入到训练集中，利用已有的正例样本和加入无标签数据中的知识可以训练获得到更有效的分类器。考虑从无标注数据中提取反例数据，结合已有的正例样本可以训练一个标准的二元分类器。Yu 等人[4]提出了 PEBL 算法来解决 PU 分类问题，它首先利用 1-DNF [4]技术来识别未标注数据中的反例数据，然后利用 SVM 算法训练分类模型。Liu 等人[1]提出 S-EM 算法，利用 Spy [1]技术识别未标注数据中的可信度较高的反例数据，然后使用 EM 算法来进行训练模型。Li 等人[5]将传统的 PU 问题应用到流式数据环境下，提出了一种基于聚类的 PU 学习算法。Xiao 等人[6]提出了一种基于相似度的 PU 学习算法，首先利用正例样本提取未标注样本数据集中可信度较高的反例样本，然后基于正例样本与提取的反例样本，计算剩余未标注样本分属正例与反例的概率，基于以上数据建立带有概率权重的 SVM 分类器。Ren 等人[7]提出了一种基于相似度 PU 学习算法应用于虚假评论识别领域。但是由于未标注样本正反例样本分布未知，最终分类器效果也会受到抽取反例参数的影响。

3. 数据分布估计下基于相似度的 PU 文本分方法

3.1. 问题定义及符号标注

给定一个训练集 P，P 中只含有正例样本，不含有任何反例样本，训练集 P 中标签集合为 $C=\{c_1,\cdots,c_n\}$；未标注数据集 U 中则同时含有正例样本和反例样本，即存在样 $d_i \notin c_k$，$\forall c_k \in C$；我们的任务就是在训练集 P∪U 上构建一个分类器 φ；$\varphi(\mathrm{P},\mathrm{U}) \Rightarrow U_n$，其中 P 和 U 是分类器 φ 的输入，$U_n=\{d_i \in U, \text{但} d_i \notin c_k, \forall c_k \in C\}$ 是分类器 φ 的输出。算法主要分为四个步骤：1) 对仅含有正例和未标注数据抽取可信正反例样本；2) 计算代表性样例；3) 确定 U 中剩余样例(称为间谍样本)的类别标签；4) 建立最终分类器。

3.2. 抽取可信正反例

由于数据集中只包含正例和未标注数据，使得传统算法失效。因此，算法的首先要解决的问题是从未标注数据中提取一些可信的反例，Yu 等人[4]提出 PEBL 的方法中采用 1-DNF 技术去识别未标注数据中的反例数据，由于 1-DNF 技术基于正例样本特征来识别反例数据，因此在正例数据较少的情况下，识别反例数据的效果较差。Liu 等人[5]提出了 Spy 技术的来识别反例数据，这方法对初始阈值十分敏感，所提取的反例数据可信度不高。Sha 等人[10]提出基于最大熵的方法来识别反例数据。许等人[11]提出了基于 KL 距离的反例识别方法用于不平衡的 PU 分类问题中。Hu 等人[12]通过大量实验证明了在未标注数据分布未知情况下，采用不同的方法来抽取未标注数据中的反例数据，以及从未标注数据中抽取反例数据的数量，都会影响到最终的分类模型性能。本文选取了 Auto-KL 算法[9]来提取反例数据，该算法通过评估未标注数据分布，自适应抽取反例数据，有效降低反例提取数目这一参数的敏感性。在得到未标注中反例数据比例后，将 Spy 和 Rocchio 两种识别反例的方法集成，从未标注数据中自适应抽取合适数量的正负例数据。记正例数据集合为 RP 中，负例数据集合记为 RN 中。抽取完正负例数据后，未标注数据集 U 中剩下的样例，称为间谍样例，记为数据集合 US。

3.3. 计算代表性正负例原型

第一步中获得可信反例集合 RN 与可信正例集合 RP，加上训练集中原本已有的少量正例数据集合 P

合并成新的训练样本，在此基础上即可初步训练一个分类器，但是，该分类器的泛化性能不高，主要原因是未标注数据集 U 中仍有大量有用的数据样例未被充分利用，即间谍样例集合 US 仍然包含大量有用信息，这些样本信息可进一步提升分类器的性能，为了进一步将间谍样例加入到训练数据中，首先需要确定间谍样例的类别标签，这里我们首先需要计算出代表性正负例原型，这里使用经典的 K-means 算法对可信反例 RN 进行聚类，即 RN 聚类为 RN_1，RN_2，…，RN_m，其中 $m=|RN|/|U|*k$，根据文献[5] [6]，m 设置为 10，最后使用 Rocchio 分类器[8]分别为正例和反例计算出 10 个代表性样例，如算法 1 所示：

算法 1：计算代表性样例原型算法。

1) 输入：P 和 RN

2) 输出：p_k 和 n_k，$k=1,\cdots,10$。

3) 将 RN 聚类成 10 个子类：$RN_1, RN_2, \cdots, RN_{10}$；

4) FOR $k=1,\cdots,10$ DO

5) $p_k=\alpha\frac{1}{|P+RP|}\sum_{e\in p}\frac{e}{\|e\|}-\beta\frac{1}{|RN_k|}\sum_{e\in RN_k}\frac{e}{\|e\|}$;

6) $n_k=\alpha\frac{1}{|RN_k|}\sum_{e\in p}\frac{e}{\|e\|}-\beta\frac{1}{|P|}$;

7) END FOR

算法 1 中：步骤 2)中每个样例使用 TFIDF[13]这一特征权值的计算方法建立向量空间模型 $v=(q_1,q_2,\ldots q_n)$，p_k 和 n_k 分别代表正例和负例的代表性样例原型，根据文献[5] [6]，α 和 β 分别设置为 16 和 4。

3.4. 确定间谍样例的类别标签

为了进一步扩充训练样本，我们必须正确计算方法剩余未标注数据 US(即间谍样例)的类别标签。将 US 中类别标签为正例的样例数据记为 LP，将 US 中类别标签为负例的样例数据记为 LN。从而整个间谍样本集合 $US=LP\cup LN$，这里我们利用算法 1 分别为正例和反例建立 10 个代表性样例原型，来估计 US 中每个间谍样例的类别标签。US 中标注为正例的间谍样例记为 LP，US 中反例的间谍样例记为 LN。由于算法 2 中采用的 Rocchio 算法已经可以初步的分离出正例数据和负例数据。但是正例数据与负例数据的决策边界不一定是线性的，Rocchio 作为一个线性分类器会出现分类错误，从而导致间谍样例发生错误标注，进而影响最终的 PU 分类模型的性能。因此简单计算每个间谍样例同代表性样例的相似度来确定其类别标签将导致一定的错误。本文提使用 K-means 对间谍样例聚类，即 US 聚类为 NS_1，NS_2，…，NS_m，其中 $n=t\times|US|/(|US|+|NS|)$，根据文献[5] [6]，$n$ 设置为 30，然后使用了基于样例局部相似度和样例全局相似度这两种方法来评估间谍样例类别，减低标注误差。

3.4.1. 样例局部相似性

样例的局部相似性的基本思想是相同微簇中的样例应有很高类别相同，算法 2 展示了样例局部相似性，对于 US 的每个微簇，首先对微簇中的每个样例与算法 1 中产生的正负代表性样例做正弦相似度计算，将最相似代表性样例的类别标签作为每个样例的临时类别标签，最后通过投票机制决定整个微簇的类别标签。微簇的类别标签作为微簇中每个样例的最终类别标签。

算法 2：样例局部相似性的计算算法(Auto-KLBSL)。

输入：$US_i, i=1,2,\cdots,m$；

输出：LP_i 和 LN_i。

1) $LP_i = \varnothing, LN_i = \varnothing, pos\_vote = 0, neg\_vote = 0$；

2) FOR 每个样例 $e \in US_i$ DO

3) IF $\max_{i=1}^{10} sim(e, p_i) > \max_{i=1}^{10} sim(e, n_i)$

4) THEN pos_vote++;

5) ELSE neg_vote++;

6) END IF

7) END FOR

8) IF pos_vote > neg_vote

9) THE $LP_i = LP_i \cup US_i$；

10) ELSE $LN_i = LN_i \cup US_i$；

11) END IF

算法 2 中，

$$sim(x, y) = x \cdot y / \|x\| \cdot \|y\|. \tag{1}$$

算法 2 通过确定微簇的类别标签来确定其内部每个样例的类别标签，该算法考虑微簇中每个样例同正反例代表性样例的相似度，基于投票机制确定微簇的类别标签。但当微簇中正反例样本比例接近时，基于样例局部相似性通过投票机制判别微簇的类别标签，会发生错误标注，从而会导致训练出来的分类模型性能较差。图 1 所示，间谍样本通过 K-means 算法的部分聚类结果。

根据算法 2，可以发现微簇 Micro-C1 与微簇 Micro-C2 很容易确定其子类标签，分别为正例和反例。但是对于微簇 Micro-C4，由于其内部正反样例的数目十分接近，若基于算法 3 来确定子类类别标签会产生一定错误。

3.4.2. 样例全局相似性

样例全局相似性基本思想是：充分考虑 US 中每个样例同全体正反例代表性样例间的相似度，并忽略样例所在的微簇间的关系。对于每个间谍样例同全体的代表性样例进行相似度计算，求出该样例属于正例和反例的类别概率，如公式 2、3 所示。选择概率最大的类别作为样例的类别标签。

算法 3：样例全局相似性的计算算法(Auto-KLBSG)。

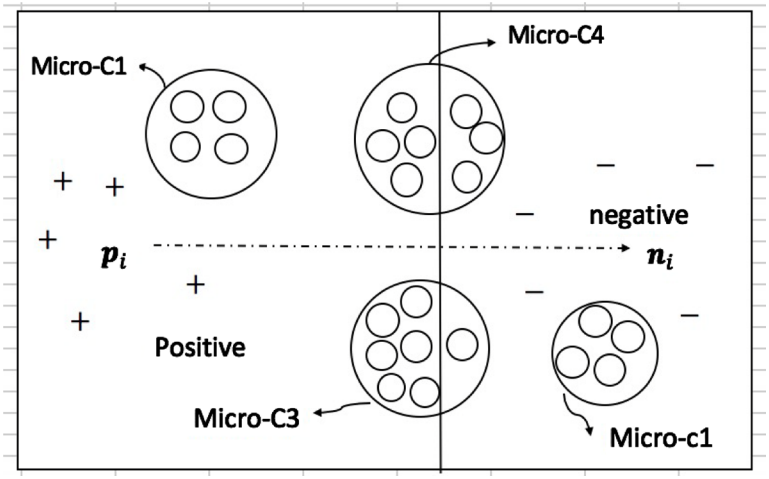

Figure 1. Local similarity of the sample
图 1. 样例的局部相似性

输入： $\mathrm{US}_i, i=1,2,\cdots,m$；

输出： LP_i 和 LN_i。

1) $\mathrm{LP}_i=\varnothing, \mathrm{LN}_i=\varnothing$；

2) FOR 每个样例 $e\in US_i$ DO

3) IF $proba\_positive(e) > proba\_negative(e)$

4) THE $\mathrm{LP}_i=\mathrm{LP}_i\bigcup\{e\}$；

5) ELSE $\mathrm{LN}_i=\mathrm{US}_i\bigcup\{e\}$；

6) END IF

7) END FOR

算法 3 中，

$$Proba\_positive(e)=\sum_{i=1}^{10} sim(e,p_i)\Big/\sum_{i=1}^{10} sim(e,p_i)+sim(e,n_i); \tag{2}$$

$$Proba\_negative(e)=\sum_{i=1}^{10} sim(e,n_i)\Big/\sum_{i=1}^{10} sim(e,p_i)+sim(e,n_i); \tag{3}$$

算法 3 通过计算间谍样例与全体代表性样例的相似度，直接判别样例的类别标签，忽略间谍样例所在微簇的类别标签，在微簇中正反例比例接近的情况下，可有效避免样例标签被错误标注。

3.5. 建立最终的分类器

将已有的正例数据集合 P 和算法第二步提取的可信正例数据集合 RP 与可信反例数据集合 RN，以及算法第三步提取的间谍样本中的正例数据集合 LP，反例数据集合 LN，组成最终的训练数据。即可训练最终的 SVM 分类器。但是考虑到传统的 SVM 算法采用一个核函数，不足以解决样本数据含有异构信息，或者数据高维特征空间分布不平坦时情况。本文使用多核 SVM 算法来进行训练最终的分类器，进一步提升分类器的性能。

多核学习[14] [15] [16]利用多个核函数将输入空间变换为高维特征空间，转化为凸优化问题。文献论述[14] [15] [16]已经证明了多核学习能获得比单核模型更好的性能。在本文提出的 PU 框架下，分别使用多核学习算法 SimpleMKL[16]进行实验，建立最终分类器算法如下：

算法 4：建立最终分类器

输入：P，RN，LP 和 LN；

输出：分类器 $\mathrm{F}_{\mathrm{SimpleMKL}}$

1) $\mathrm{P}=\mathrm{P}\bigcup\mathrm{LP}$；

2) $\mathrm{N}=\mathrm{RN}\bigcup\mathrm{LN}$；

3) 使用 SimpleMKL 在 P+N，训练最终分类器 $\mathrm{F}_{\mathrm{SimpleMKL}}$。

4. 数据分布估计下基于相似度的 PU 文本分类方法

4.1. 实验与分析

4.1.1. 实验设置

针对文本实验数据，Auto-KLBS 主要跟性能比较好的 Spy-SVM [1]，Roc-EM [8]，LELC [5]，AutoKL [12]共 4 种方法相比较，全部实验运行在 2.50 GHz 的处理器和 4 GB 的内存台式机上。

为了能更好地比较不同分类方法的优劣，根据文献[6]，使用评估分类算法性能的 F-score 作为分类器的评价指标，其定义如下： $\text{F-score}=2\times p\times\dfrac{r}{p+r}$，其中 p 和 r 分别为正确率(precision)和召回率(recall)。

F-score 同时考虑了查准率和查全率。只有当正确率与召回率越大，F-score 值也越大。F-score 越接近于 1，这证明该算法分类效果越好。

在实验中，使用两个真实文本数据集 20-NewsGroup 和 Reuters corpus (表 1)。

对于 20Newsgroups 数据，首先，根据文献[11]从中选择了 4 个计算机相关主题以及 2 个科学相关主题，比如：{comp.graphics, comp.ibm.hardware, comp.mac.hardware, comp.windows.x} × {camsci.crypt, sci.space}，共 $C_4^1 \times C_2^1 = 8$ 组数据。从中选出一组数据作为正例数据，我们把剩余数据的所有数据都作为反例加入到无标注集中。在 Reuters corpus 数据上进行同样的处理，选取的主题组合为{acq-crude, acq-earn, crude-interest, crude-eam，earn-interest, interest-acq}。以 graphics × sci.crpypt 为例，从中随机选主题相同的文本作为相应的正例样本，其余部分作为未标注数据中的正例数据。然后从剩余 18 个主题中随机抽取作为反例数据放进未标注数据 U 中，其数量为 $\alpha \times |U|$，α 为比例参数，首先设置 $\alpha = 0.1$，然后在 α 递增的条件下，$\alpha = \{0.1, 0.2, 0.4.0.6, 0.8\}$ 依次实验，验证算法的有效性。

4.1.2. 特征提取

在构造完 PU 文本数据后，我们对文本数据建立向量空间模型。特征选择是文本分类任务中极为重要的一个环节，它能有效剔除冗余特征和无效特征，从而达到特征降维的效果，从而可以提升整个分类模型的分类效果。本文中使用 TF-IDF [13]对数据集建立的向量空间模型。在剔除停用词后，选取每个主题下 TF-IDF 值最高的 150 个单词作为该主题的特征代表词。对多个主题合并为一个正例的时候，对主题特征空间进行并集操作。

4.1.3. 实验结果与分析

对于生成的实验数据，本文采取无放回抽样随机选取 60%的实例数据作为训练数据，剩余的则作为测试数据，进行 10 折交叉验证。为了避免采样误差，对上述过程重复 10 次，然后计算平均的 F-score，实验中 α 是未标注数据中反例数据所占比例。例如 $\alpha = 0.1$ 表示未标注数据中含有 10%的反例。表 2 显示了 6 组子数据下实验的 F-score。

Table 1. Data sets
表 1. 实验数据集

数据集	类数	属性数	实例数	类型	URL
20-Newsgroup	20	>100	20,000	文本	URL1
Reuters corpus	10	>100	9981	文本	URL2

注：URL1：http://www.daviddlewis.com/resources/testcollections/；URL2：http://people.csail.mit.edu/jrennie/20Newsgroups/。

Table 2. 6 sets of sub-datasets on each algorithm F-score
表 2. 6 组子数据集上各算法 F-score

Data Subset	Spy-SVM	ROC-EM	Auto-KL	Auto-KLBSL	Auto-KLBSG
graphics-crypt	0.235	0.241	0.274	0.268	0.342
graphics-space	0.305	0.276	0.287	0.312	0.310
ibm.hard-crypt	0.257	0.262	0.284	0.278	0.280
acq-crude	0.248	0.256	0.258	0.262	0.267
acq-carn	0.234	0.249	0.245	0.258	0.265
crude-interest	0.326	0.301	0.357	0.368	0.409

鉴于本文篇幅限制，如表 2 所示，选取了本文提出的 Auto-KLBSL (样例局部相似性)和 Auto-KLBSG (样例全局相似性)在 6 个子数据集上有 5 个子数据集上效果优其他对比算法。这是因为我们的 Auto-KLBS 方法首先利用 Auto-KL 算法评估了未标注数据中的正反例数据分布比例，提取合适的可信正反例，降低了分类器对反例提取数目的参数敏感性，同时利用相似度方法对剩余的大量未标注样例进行标签评估，进一步扩充了训练数据集，相对于单核 SVM，使用多核学习构建了更高性能的分类器。对比 PU-BSDS 两种不同采取两种不同的样例相似度计算策略，Auto-KLBSG (样例全局相似性)通常在子数据集上的分类效果要比 Auto-KLBSL (样例局部相似性)的分类效果好，这是因为前者基于样例所在子类的局部关系，但当子类中正反例样本比例接近时，基于投票机制会出现类标签错误标注，后者计算样例与全体代表性样例的相似度，直接判别样例的类别标签，忽略间谍样例所在子类的类别标签，在子类中正反例比例接近的情况下，可有效避免样例标签被错误标注。

图 1，图 2 分别展示了随着 α 增大，在两组 20-NewsGroup 和 Reuters corpus 数据上各算法的实验结果。

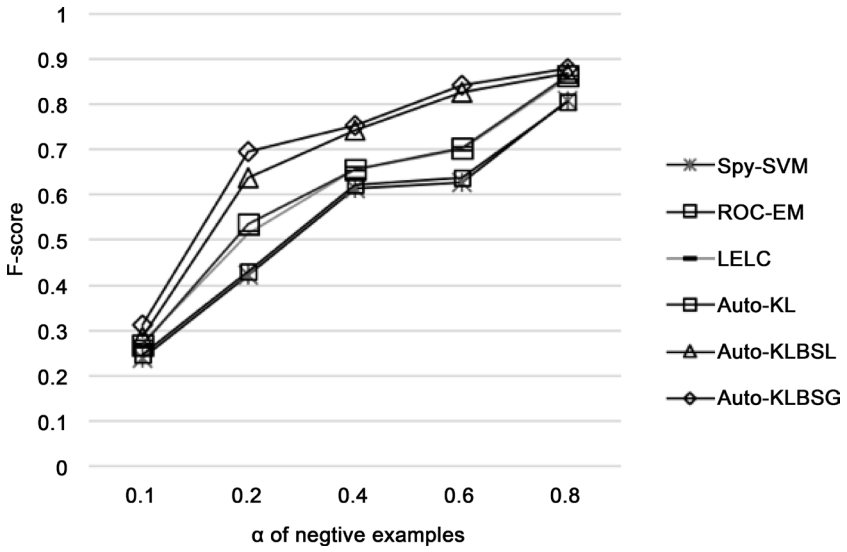

Figure 2. Classification performance comparison on 20-Newsgroup
图 2. 20-Newsgroup 上分类性能比较

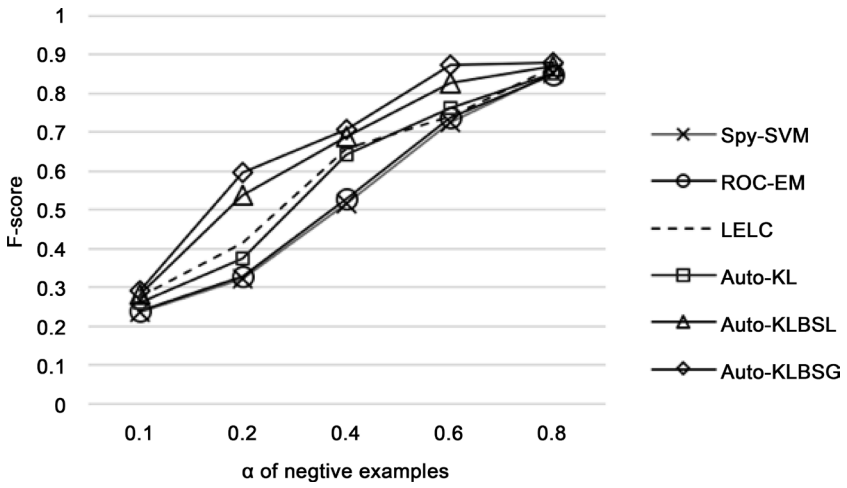

Figure 3. Classification performance comparison on Reuters corpus
图 3. Reuters corpus 上分类性能比较

如图 2，图 3 所示随着 α 增大，即未标注数据中反例样本比例增加，各个算法 F-score 均有所提高，即未标注数据中反例样本比例增加，各个算法的 F-score 均有所提高，这是因为随着未标注数据中反例样本增加，通过反例提取算法从未标注数据中收集的反例数据也随之增加，最终构建的训练集的有用信息也更加丰富，所以从训练集学习的模型性能也越强，从而相应的分类效果越好。同时可以明显观察到在同一□值下，本文提出的两种算法均优于其他几种对比算法。

5. 数据分布估计下基于相似度的 PU 文本分类方法

本文提出了一种数据分布估计下基于相似度的 PU 学习方法，首先利用数据分布评估算法估计未标注数据中正反例数据的分布比例，并利用集成机制从未标注数据提取可靠负例样本，对于剩余仍存在的间谍样本利用两种相似度策略，提取正负微簇，从而极大的丰富了训练集，最后利用多核学习训练分类器。数值实验证明了本文所提方法的有效性。未来考虑将本文方法拓展到流式数据环境中。

基金项目

国家自然科学基金：(61503112)；国家重点基础研究发展计划(973)项目(2016YFC0801406)。

参考文献

[1] Liu, B., Lee, W.S., Yu, P.S., *et al.* (2002) Partially Supervised Classification of Text Documents. *Nineteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., Sydney, July 2002, 387-394.

[2] Tax, D.M.J. (1999) Data Domain Description Using Support Vectors. *European Symposium on Artificial Neural Networks*'99, Brugge, 21-23 April 1999, 251-256.

[3] Manevitz, L.M. and Yousef, M. (2002) One-Class Svms for Document Classification. *Journal of Machine Learning Research*, **2**, 139-154.

[4] Yu, H., Han, J. and Chang, C.C. (2002) PEBL: Positive Example Based Learning for Web Page Classification Using SVM. *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, 12-16 August 2012, 239-248.

[5] Li, X.L., Yu, P.S., Liu, B., *et al.* (2009) Positive Unlabeled Learning for Data Stream Classification. *Siam International Conference on Data Mining*, *SDM* 2009, Sparks, Nevada, 30 April-2 May 2009, 257-268.

[6] Xiao, Y., Liu, B., Yin, J., *et al.* (2011) Similarity-Based Approach for Positive and Unlabeled Learning. *IJCAI* 2011, *Proceedings of the, International Joint Conference on Artificial Intelligence*, Barcelona, July 2011, 1577-1582.

[7] Ren, Y., Ji, D. and Zhang, H. (2014) Positive Unlabeled Learning for Deceptive Reviews Detection. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing*, *EMNLP*, Doha, 25-29 October 2014, 488-498. https://doi.org/10.3115/v1/D14-1055

[8] Li, X. and Liu, B. (2003) Learning to Classify Texts Using Positive and Unlabeled Data. *International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., Acapulco, 9-15 August 2003, 587-592.

[9] Hu, H., Sha, C., Wang, X., *et al.* (2012) Estimate Unlabeled-Data-Distribution for Semi-Supervised PU Learning. *Asia-Pacific Web Conference*, Springer, Berlin, Heidelberg, Kunming, 11-13 April 2012, 22-33.

[10] Sha, C., Xu, Z., Wang, X., *et al.* (2009) Directly Identify Unexpected Instances in the Test Set by Entropy Maximization. *Journal of Clinical Oncology*, **31**, 659-664. https://doi.org/10.1007/978-3-642-00672-2_67

[11] 许震, 沙朝锋, 王晓玲, 等. LiPU: 一种基于 KL 距离的主动分类算法[C]//中国数据库学术会议. 北京, 2009.

[12] Hu, H., Sha, C., Wang, X., *et al.* (2014) A Unified Framework for Semi-Supervised PU Learning. *World Wide Web-Internet & Web Information Systems*, **17**, 493-510. https://doi.org/10.1007/s11280-013-0215-7

[13] Shaw Jr., W.M. (1986) On the Foundation of Evaluation. *Journal of the Association for Information Science & Technology*, **37**, 346-348. https://doi.org/10.1002/(SICI)1097-4571(198609)37:5%3C346::AID-ASI10%3E3.0.CO;2-5

[14] 汪洪桥, 孙富春, 蔡艳宁, 等. 多核学习方法[J]. 自动化学报, 2010, 36(8): 1037-1050.

[15] Sun, T., Jiao, L., Liu, F., *et al.* (2013) Selective Multiple Kernel Learning for Classification with Ensemble Strategy. *Pattern Recognition*, **46**, 3081-3090. https://doi.org/10.1016/j.patcog.2013.04.003

[16] Li, J. and Sun, S. (2010) Nonlinear Combination of Multiple Kernels for Support Vector Machines. *International Con-*

ference on Pattern Recognition. *IEEE Computer Society*, Istanbul, 23-26 August 2010, 2889-2892.

[17] Rakotomamonjy, A., Bach, F.R., Canu, S., *et al.* (2008) Simplemkl. *Journal of Machine Learning Research*, **9**, 2491-2521.