通过评估示例中概念的重要性来解决多示例学习问题

甘 睿 印 鉴
(中山大学信息科学与技术学院 广州 510006)

摘 要 在多示例学习问题中,训练数据集里面的每一个带标记的样本都是由多个示例组成的包,其最终目的是利用这一数据集去训练一个分类器,使得可以利用该分类器去预测还没有被标记的包。在以往的关于多示例学习问题的研究中,有的是通过修改现有的单示例学习算法来迎合多示例的需要,有的则是通过提出新的方法来挖掘示例与包之间的关系并利用挖掘的结果来解决问题。以改变包的表现形式为出发点,提出了一个解决多示例学习问题的算法——概念评估算法。该算法首先利用聚类算法将所有示例聚成 d 簇,每一个簇可以看作是包含在示例中的概念;然后利用原本用于文本检索的 TF-IDF(Term Frequency-Inverse Document Frequency)算法来评估出每一个概念在每个包中的重要性;最后将包表示成一个 d 维向量——概念评估向量,其第 i 个位置表示第 i 个簇所代表的概念在某个包中的重要程度。经重新表示后,原有的多示例数据集已不再是"多示例",以至于一些现有的单示例学习算法能够用来高效地解决多示例学习问题。

关键词 多示例学习,重新表示,单示例学习,概念评估
中图法分类号 TP181 **文献标识码** A

Solving Multi-instance Learning Problem with Evaluating the Importance of Concept in Instances

GAN Rui YIN Jian
(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China)

Abstract In multi-instance learning, the training set is composed of labeled bags, each of which consists of many unlabeled instances, and the goal is to learn some classifier from the training set for correctly labeling unseen bags. In the past, some researches about multi-instance learning aim at improving single-instance learning algorithms to meet the multi-instance representation, and others try to propose some new methods to find the relationship between instances and bags and use the result to solve the problem. This paper started from adapting the representation of the bag and proposed a new algorithm——concept evaluating algorithm. First, this algorithm uses a cluster algorithm to cluster all instances into d group, here each group can be treated as a concept in the instances. Then, it uses the TF-IDF (term frequency-inverse document frequency) algorithm to get the importance of each concept in the bag. Finally, each bag is re-represented as a d dimensional vector——concept evaluating vector, the ith value in this vector is the importance of the ith group in the bag. Because after re-representing the data set is not "multi" again, some propositional single-instance learning algorithms can be used to solve multi-instance learning problem effetely.

Keywords Multi-instance learning, Re-represent, Single-instance learning, Concept evaluating

多示例学习(multi-instance learning)[1]这一概念,是由 Dietterich 等人在 1997 研究药物活性预测问题时提出的,其目的是为了让学习系统通过分析已经被标记为适合或不适合制药的分子来预测还没有被标记的新分子。其难点在于,每一个分子里面都包含了很多种低能形状(low-energy shape),专家们能够知道的是哪个分子适合制药,至于该分子里面哪种低能形状起决定性作用,则一无所知。

一开始,Dietterich 等人尝试用监督学习的方法来解决问题,并把所有在适合制药的分子里面的低能形状当作是正例,不适合的分子里面的低能形状当作反例。但是,他们很快就发现这样做是行不通的,因为在一个被标记为适合的分子里面包含着很多种低能形状,而在这些低能形状集合中可能就只有一种起到决定性作用,其他的根本起不了作用。Dietterich 等人的做法是把不起作用的也当作正例,从而增加数据的噪音,影响了学习系统的学习效果。于是,Dietterich 等人把分子定义为包,分子里面的低能形状当作是包中的示例。并假设,如果一个包被标记为正,那么该包至少包含一个正示例;如果一个包被标记为反,那么该包里面的所有示例都是反例。而在训练集中只给出了包的标记,并没有给出示例的标记。由此,一种新的机器学习问题诞生了。

1 相关工作

在 Dietteric 等人的研究之后,很多关于这一新的机器学习问题的研究陆续展开。

到稿日期:2011-08-30 返修日期:2011-12-31
甘 睿(1979-),男,博士生,主要研究方向为数据挖掘、机器学习,E-mail:469370205@qq.com。

在这些研究中，有的是以示例为出发点，通过挖掘示例与包之间的关系，并利用挖掘的结果来解决多示例学习问题。例如，多样性密度(Diverse Density，DD)[2]算法把每包表示成由示例构成的集合，其任务就是在由示例形成的属性空间中找到具有最大多样性密度的那个点。在这里，多样性密度是一种度量，如果一个点附近出现的正包数越多，而反包示例出现得越远，那么该点的多样性密度越大。找到该点后，就可以把这个点作为参照点来标记新的包。但多样性密度算法有个很大的缺点，就是效率低。要在由示例形成的属性空间中找到目标点，是一件很耗时的事情；而且在寻找的过程中，由于采用梯度下降法使得算法并不能确保找到全局最优解，寻找结果的好坏直接影响到最终的分类结果。即使是后来提出的期望最大多样性密度(Expectation Maximization Diverse Density，EM-DD)算法[3]，结合了 EM 算法的思想来针对多样性密度算法效率低的缺点进行了改进，但仍然需要耗费一定的时间才能获取到最终结果。

随着研究的深入，有些研究者发现通过对现有的基于单示例的监督学习算法进行改进，可以使这些算法能够用于解决多示例学习问题。例如 Citation-kNN[4]算法就是对 k-近邻分类算法的一种改进，该算法认为一个包的标记不但由这个包的近邻来决定，还应该由把这个包当作近邻的那些包来决定。在寻找包的近邻时，该算法使用最小化豪斯多夫距离(minimum Hausdorff distance)来衡量包与包之间的距离，而不是使用传统的欧氏距离(euclidean distance)。直到现在，Citation-kNN 算法依旧是众多经典的多示例算法中分类效果相对较好的算法之一，但是其效率不高，这也是它唯一缺点。虽然与多样性密度算法相比，该算法在效率方面有了很大的提高，但是要在一个数据集中同时查找两类近邻，也需要耗费一定的时间来计算包与包之间的距离，尤其是在数据集相对较大的情况下更为明显。

本文从改变包的表现形式入手，提出了概念评估算法。首先通过聚类算法挖掘出包含在所有示例中的概念；再运用文本检索中的 TF-IDF(Term Frequency-Inverse Document Frequency)算法评估出每个概念在每个包的重要性；最后把包表示成概念重要性向量形式，不但考虑了概念在包中出现的次数，还把概念在整个数据集中的重要性也考虑进去。其具体做法如下。

2 通过评估示例中概念的重要性解决多示例学习问题

在多示例分类中，我们获取到的是一个由包构成的训练集合$\{(B_1,y_1),\cdots,(B_m,y_m)\}$，其中每个包又由多个示例构成，而$B_i=\{x_{i,1},x_{i,2},\cdots,x_{i,m_i}\}$，这里的$m_i$表示构成$B_i$包的示例集合的大小，$y_i\in\{\pm1\}$是包的标记集合。

从词袋模型入手，尝试找出解决多示例问题的算法。

2.1 词袋模型(Bag-of-Words Model)的主要思想

词袋模型主要运用于自然语言处理和文本信息检索方面，大大简化了文本的表现形式。它把文本表示成一堆单词的集合，在表示过程中，每个词的出现都是独立的，并不需要考虑单词出现的先后顺序，也不用考虑文本的语法和句法。由于在文本挖掘方面取得成功，词袋模型已经开始被运用到其他领域，例如图像中的对象分类。其主要思想[5]是把图像里面的每个特征点量化到几个有代表性的关键点上面，然后利用量化的结果重新表示原图像。从这里可以看出，要想利用词袋模型解决其他领域的问题，首先要做以下两步工作：

(1)发现关键点，一般采用聚类方法；

(2)把所有的特征点量化到关键点上。

因此我们做的第一件事情是利用聚类算法把数据集中所有示例划分为d簇$G=\{G_1\cdots G_d\}$(这里并没有考虑各示例所在的包的标记)；然后把每个包中的示例量化到这d簇上。

表 1 给出了一个量化结果的例子。在这里，假设在聚类时设定了簇的数量为 5。每一个格子里面的数字表示在一个包中有多少个示例属于某个簇。

表 1 包示例量化到 5 个簇上的结果

包	G_1	G_2	G_3	G_4	G_5
B_1	2	1	0	0	0
B_2	0	0	2	1	0
B_3	0	0	3	1	0
B_4	1	1	0	0	1
B_5	2	1	0	0	1

每个簇又可以被看作是一个包含在所有示例中的概念。如果一个包包含n个示例属于簇d，那么可以把n看作是概念d在这个包出现的次数。

2.2 利用 TF-IDF 方法评估概念在包中的重要性

TF-IDF[6](Term Frequency-Inverse Document Frequency)是一种用于资讯检索与资讯探勘的常用加权技术，用以评估某个词语对于一个文件集中的一份文件的重要程度。其主要思想是：如果某个词语在一篇文章中出现的频率高，并且在其他文章中很少出现，则认为此词具有很好的类别区分能力，适合用来分类。

TF-IDF 实际上是 TF * IDF。TF 表示词频(Term Frequency)，指的是某一个给定的词语t_i在某个文件d_j中出现的次数，其计算方法如下：

$$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

式中，分子$n_{i,j}$表示t_i在文件中出现的次数，分母则是在文件d_j中所有的词出现的次数之和。

IDF 则表示逆向文件频率(Inverse Document Frequency)，是某一个给定的词语t_i普遍重要性的度量，其计算方法如下：

$$Idf_i=\log\frac{|D|}{|\{j:t_i\in d_j\}|} \tag{2}$$

式中，分子$|D|$表示文件集的文件总数，分母$|\{j:t_i\in d_j\}|$表示包含词语t_i的总文件数。

如果把表 1 中的每个包均看成是一篇文章，每个簇均看成是一个词，那么就可以运用 TF-IDF 方法来评估每个簇所代表的那个概念在每个包中的重要性。

2.3 概念评估算法

结合词袋模型和 TF-IDF 方法，提出了一个解决多示例学习问题的算法——概念评估算法。其主要思想是，要把原来多示例数据集中的每一个包表示成如下向量。在这里，把这种表现形式命名为概念评估向量。

$$Bag_i=[\text{TF-IDF}(i,1),\text{TF-IDF}(i,2),\cdots,\text{TF-IDF}(i,d)]$$

式中，TF-IDF(i,j)表示通过 TF-IDF 方法获得的第j个簇在Bag_i中的重要性。

经过重新表示后，原来的多示例数据集变成了普通的单示例数据集，一些基于单示例的监督学习方法可以用来解决原来的多示例学习问题。整个概念评估算法的伪代码见表 2。

表 2　概念评估算法伪代码

```
概念评估算法(concept evaluating algorithm)
Concept_ Evaluating (B_Train,B_Test,d,Cluster,Classfier)
  Input:B_Train:A train set of m bags {(B1,y1),…,(Bm,ym)}
        B_Test:A test set of n bags {B1,…,Bn}
        d:The number of cluster
        Cluster:Clustering algorithm
        Classfier:Classifier training algorithm
  Z←Ø
  for Bi∈B_Train do
    for x∈Bi do
      Z←Z ∪ {x}
    end of for
  end of for
/*把所有示例聚成 d 个簇*/
  {Group1…Groupd}=Cluster(Z,d)
/*把训练集中的包量化到 d 个簇 */
  InitialTrain←Ø
  for Bi∈B_Train do
    for k∈{1…d} do
      y^i_k←Overlap(Bi,Groupk) /*y^i_k 表示包 Bi 中属于簇 k 的示例数 */
    end of for
    InitialTrain←InitialTrain∪{⟨y^i_1…y^i_d⟩}
  end of for
/*利用 TF-IDF 方法计算 d 个簇在训练集中每个包的重要性*/
/*并把每个包表示成带标记概念重要性向量形式*/
  InitialTrainTfidf←Ø
    for i∈{1…m} do
      for k∈{1…d} do
        tf(i,k)=CountTF(InitialTrain) /*计算 Groupk 的 TF 值*/
        idfk= CountIDF(InitialTrain) /*计算 Groupk 的 IDF 值*/
        TF-IDF(i,k)=tf(i,k) * idfk
      end of for
        InitialTrainVector←InitialTrainVector ∪{⟨TF-IDF(i,1)…TF-IDF
(i,d),yi⟩}
    end of for
/*利用改变后的训练集训练分类器*/
  Classifier. Train(InitialTrainVector)
Z←Ø
for Bi∈B_Test do
  for x∈Bi do
    Z←Z ∪ {x}
  end of for
end of for
/*把测试集中的包量化到 d 个簇*/
  InitialTest←Ø
  for Bi∈B_Test do
    for k∈{1…d} do
    y^i_k←Overlap(Bi,Groupk) /*y^i_k 表示包 Bi 中属于簇 k 的示例数*/
  end of for
  InitialTest←InitialTest ∪ {⟨y^i_1…y^i_d⟩}
end of for
/*利用 TF-IDF 方法计算 d 个簇在测试集中每个包的重要性*/
/*并把每个包表示成不带标记的概念重要性向量形式*/
  InitialTestVector←Ø
  for i∈{1…n} do
    for k∈{1…d} do
      tf(i,k)=CountTF(InitialTest) /*计算 Groupk 的 TF 值*/
      idf_k= CountIDF(InitialTest) /*计算 Groupk 的 IDF 值*/
      TF-IDF(i,k)=tf(i,k) * idf_k
    end of for
      InitialTestVector←InitialTestVector ∪{⟨ TF-IDF(i,1)…TF-IDF(i,d)⟩}
  end of for
/*利用训练好分类器标记经过转变的测试集*/
   Output:Label← Classifier. classify (InitialTestVector)
```

3　实验结果与比较

3.1　多示例数据集

Musk 数据集是由 T. G. Dietterich 等人提供的专门用于测试多示例学习算法的公共数据集。它包括两个独立数据集，即 Musk1 和 Musk2。Musk1 包含 47 个正包和 45 个反包，每个包所含的示例个数从 2 到 40 不等；Musk2 则包含 39 个正包和 63 个反包，每个包所含的示例个数从 1 到 1044 不等。关于这两个数据集的详细信息见表 3。

表 3　Musk 数据集详细信息

数据集	维度	包			示例总数
		包总数	正包数	反包数	
Musk1	166	92	47	45	476
Musk2	166	102	39	63	6,598

3.2　单示例学习算法与多示例学习算法比较

为了找出哪个单示例学习算法最适合充当概念挖掘算法的分类器，我们选择了 5 个由 Weka 所支持的单示例学习算法分别作为概念挖掘算法的分类算法；然后选择 5 个经典多示例学习算法，分别比较这两类算法在 Musk1 和 Musk2 数据集上的分类效果。关于这 5 个单示例学习算法的详细信息见表 4。

表 4　5 个算法的详细信息

Weka 内部名称	描述
SMO	利用序列最小最优化(Sequential Minimal Optimization[7])法实现的支持向量机(SVM)
MultilayerPerceptron	采用反向传播的神经网络
Ibk	K-近邻分类算法
J48	C4.5 决策树
NaiveBayes	贝叶斯(Naive Bayes)分类器

对于 5 个单示例学习算法，首先利用概念挖掘算法，把数据集中的每一个包都转变成概念向量形式，然后将其交给这 5 个算法来进行训练。实验重复了 10 次 10 折交叉验证，在转变包的过程中采用了 K-means 算法对所有示例进行聚类，聚簇数设为 40。表 5 给出了这 5 个单示例学习算法在 Musk 数据集上实验结果。在这里，除了 Ibk 设定近邻数为 3 之外，其他所有单示例学习算法都是以 Weka 设定的默认值来运行。表 6 给出了 5 个经典的多示例学习算法在 Musk 数据集上的最好分类正确率。

表 5　5 个单示例算法在 Musk 数据集上 10 次 10 折交叉验证的结果

算法	Musk1 上的分类正确率(%)	Musk2 上的分类正确率(%)
SMO	94.5	83.5
MultilayerPerceptron	**90.1**	**87.2**
Ibk	87.0	82.5
J48	92.3	81.5
NaiveBayes	90.2	65.9

表 6　5 多个示例学习算法在 Musk 数据集上的最好分类结果

算法	Musk1 上的分类正确率(%)	Musk2 上的分类正确率(%)
MI-SVM	77.9[8]	84.3[8]
Diverse Density	88.9[2]	82.5[2]
EM-DD	**96.8**[3]	**96.0**[3]
Citation-KNN	92.4[4]	86.3[4]
MiBoost	87.9[9]	84.0[9]

从表 5 和表 6 中可以看到，对于 Musk1 数据集，当选择 MultilayerPerceptron(神经网络)充当分类算法时，概念评估算法的分类正确率为 90.1%，仅次于 Citation-KNN 和 EM-DD 算法，但比其他 3 个多示例学习算法的分类正确率要高。对于 Musk2 数据集，同样选择 MultilayerPerceptron 充当分类算法时，概念评估算法的分类正确率为 87.2%，仅次于 EM-DD 算法，但比其他 4 个多示例学习算法的分类正确率都要高。实验结果表明，使用我们提出的概念评估算法来改变包的表现形式后，一些单示例学习算法能够被用来解决多示例学习问题，其效果比一些经典的多示例学习算法还要好。

3.3 簇的数量对正确率的影响

从第 2 节提出的方法可以看出，在聚类过程中，设定不同的簇数量，会生成不同维度的概念向量，因此会产生不同版本的数据集。如下实验主要研究分类正确率与不同簇数量之间的关系。这里设定簇数量的变化范围为 2 到 80。图 1 是 5 个单示例算法在经过转变后的 Musk1 数据集上 10 次 10 折交叉验证的结果与簇数量之间的关系，图 2 是在经过转变后的 Musk2 数据集上的结果。

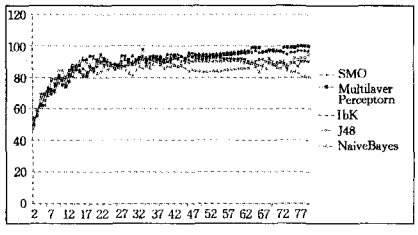

图 1 Musk1 数据集上分类正确率与簇数量之间的关系

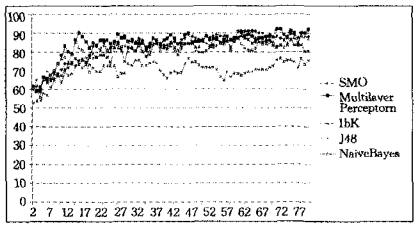

图 2 Musk2 数据集上分类正确率与簇数量之间的关系

从上两图中可以看出，无论使用哪个数据集，随着簇数量的改变，5 个单示例学习算法的分类准确率先递增后在一定的范围内上下波动。簇数量的递增，可以看作是对数据集的理解从肤浅到深入的过程。簇数量较少时，生成的概念数也相对较少，对数据集理解相对肤浅，因此分类准确率不高；随着簇数量越来越大，生成的概念数也就越来越多，对数据集的理解越来越深入，分类准确率也逐步提高；当到达一定幅度后，分类准确率相对稳定，然后在一定的范围里上下波动。这是因为随着簇数量的进一步增加，一些起决定性作用的概念被分成更加小的概念，从而使分类准确率有所下降；同时，一些新的起决定性作用的概念又被分割出来，从而导致分类准确率有所回升。表 7 对上面的实验进行了总结。

表 7 簇数量与算法分类正确率之间的关系总结

算法	Musk1		Musk2	
	最高正确率(%)	最好簇数量	最高正确率(%)	最好簇数量
SMO	96.8	68～72,77～80	91.2	60～63
MultilayerPerceptron	100	77～80	92.1	80
Ibk	93.4	21	88.1	78,79
J48	93.4	33,37	88.3	69
NaiveBayes	94.4	80	76.5	46

从表 7 可以看出，对于不同的多示例数据集，使用不同的单示例学习算法来充当分类器的概念评估算法，在获取到最好的分类正确率时，对簇数量的要求各不相同。如何根据不同的数据集以及不同的单示例学习算法来自动找出最好的簇数量，是我们下一步研究工作的重点。

结束语 本文从改变包的表现出发，结合词袋模型和 TF-IDF 统计方法，提出概念评估算法。该算法把包转变成概念评估向量的形式，经过转变后，原来的多示例数据集变成了普通的单示例数据集，则可以利用现有的基于单示例的监督学习算法来解决多示例学习问题。实验证明，数据集的表现形式虽然发生了改变，但分类的效果并没有受到影响，分类准确率比一些经典的多示例学习算法还要好。

参 考 文 献

[1] Dietterich T G, Lathrop R H, Lozano-Pérez T. Solving the multiple-instance problem with axis-parallel rectangles[J]. Artificial Intelligence, 1997, 89(1/2): 31-71

[2] Maron O, Lozano-Pérez T. A framework for multiple-instance learning[M]. Neural Information Processing Systems 10, Cambridge, MA: MIT Press, 1998: 570-576

[3] Zhang Qi, Goldman S A. EM-DD: An Improved Multiple-instance Learning Technique[M]. Neural Information Processing Systems, 2001

[4] Wang Jun, Jean-Daniel Z. Solving Multiple-instance Problem: A Lazy Learning Approach[C]//17th International Conference on Machine Learning. 2000: 1119-1125

[5] Zhang Y, Jin R, Zhou Z-H. Understanding bag-of-words model: A statistical framework[J]. International Journal of Machine Learning and Cybernetics, 2010, 1(1): 43-52

[6] Wikipedia. tf-idf[EB/OL]. http://en.wikipedia.org/wiki/Tf%E2%80%93idf

[7] Platt J. Machines Using Sequential Minimal Optimization[M]// Schoelkopf B, Burges C, Smola A, eds. Kernel Methods-Support Vector Learning. 1998

[8] Andrews S, Tsochantaridis I, Hofmann T. Support Vector Machines for Multiple-instance Learning[M]. Neural Information Processing Systems 15, 2003: 561-568

[9] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C] // Thirteenth International Conference on Machine Learning. San Francisco, 1996: 148-156