计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(7):127−132 [doi: 10.15888/j.cnki.csa.006392]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 Doc2Vec 与 SVM 的聊天内容过滤①

岳文应

(浙江理工大学 信息学院, 杭州 310018)
通讯作者: 岳文应, E-mail: 783360782@qq.com

摘　要: 直播系统中用户聊天内容的实时拦截具有非常重大的意义, 为了提高分类的准确率和效率, 提出了一种基于 Doc2Vec 与 SVM 结合的文本分类模型对聊天内容分类, 判断聊天内容是否应该被拦截. 首先使用 Doc2Vec 模型将聊天内容表示成密集数值向量的形式, 第二部分使用 SVM 分类器进行分类. 通过实验表明, 该模型有效地减少了文本表示的维度, 提高了训练效率, 而且具有的 97% 的准确率和 89.82% 召回率, 性能优于朴素贝叶斯和基于 Doc2Vec 的 Logistic 模型.

关键词: 文本分类; 自然语言处理; Doc2Vec 模型; 支持向量机

引用格式: 岳文应.基于 Doc2Vec 与 SVM 的聊天内容过滤.计算机系统应用,2018,27(7):127–132. http://www.c-s-a.org.cn/1003-3254/6392.html

Chat Content Filtering Based on Doc2Vec and SVM

YUE Wen-Ying

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: The real-time interception of user chat content in live broadcast system is of great significance. In order to improve the accuracy and efficiency of the classification, a text classification model based on the combination of Doc2Vec and SVM is proposed to classify the chat content and judge whether the chat content should be intercepted. The First part uses the Doc2Vec model to represent the chat content as a dense numeric vector, and then an SVM classifier is used to classify. The experimental results show that the model greatly reduces the dimension of text representation with high efficiency, and it has excellent accuracy rate (97%) and recall rate (89.82%), which are superior to Naive Bayes and the logistic based on Doc2Vec.

Key words: text classification; Natural Language Processing (NLP); Doc2Vec model; Support Vector Machine (SVM)

1　引言

在高度网络化的社会中, 对直播系统中的聊天内容进行过滤拦截有着十分重要的意义, 因为垃圾聊天及广告内容一方面污染网络环境, 另一方面对主播及观众会造成很大的困扰, 极大的影响客户的体验. 甚至一些垃圾聊天内容会传播色情、反动等各式各样的有害信息, 给社会带来危害.

聊天内容过滤问题可以看为文本的二分类问题, 即将用户的聊天内容分类为 0 或 1(其中 1 表示需要拦截的垃圾或广告聊天内容, 0 表示正常的不需要拦截的内容). 类似于垃圾邮件的过滤问题. 目前常用的垃圾信息过滤方法主要包括三类:

(1) 黑白名单过滤法: 比如基于 IP 地址的垃圾信息过滤, 即只拦截黑名单中用户发的信息. 但是这种方法有两大缺陷: 一这种方法需要手工维护黑名单, 而且在实际中发送垃圾信息的用户可以采用动态变化的地址; 更严重的是, 这种方法具有一次拒绝性, 即一旦用户发过被拦截信息, 则该用户就会被加入拦截黑名单

① 收稿时间: 2017-10-16; 修改时间: 2017-11-03; 采用时间: 2017-11-08; csa 在线出版时间: 2018-06-27

中, 以后的信息都会被拦截. 所以, 在实际中会造成极高的误拦率.

(2) 基于关键词规则的过滤[1]: 根据历史或专家经验, 定义一些能反映需拦截聊天内容的关键词或短语, 比如: “免费”, “加 QQ”, “进微信群”等. 当聊天内容匹配到若干条关键词或短语时就会被判定为垃圾邮件. 但是这种方法具有很大的局限性, 只能拦截固定的形式内容的垃圾聊天内容, 不具备灵活性, 不能智能判断.

(3) 基于机器学习的方法, 首先从训练样本中学习得到分类模型, 然后再利用分类模型对未知样本进行分类,目前主要的模型有朴素贝叶斯分类器[2–7]、SVM 分类器、随机森林等. 朴素贝叶斯算法是一种简单而高效的基于概率统计的分类算法, 能够适用于垃圾信息分类. 文献[3]提出使用朴素贝叶斯算法对邮件进行分类过滤, 文献[4]在基于贝叶斯算法的基础上建立一个用于邮件过滤的机器学习应用系统 ifle, 这种方法分类的速度很快, 并且可以对其进行动态调整, 但是错误率较高. 文献[5]提出了基于最小风险的贝叶斯过滤方法, 它做决策时不仅考虑了后验概率的大小, 也把是否损失最小作为决策依据来考虑, 虽然对分类的查全率有所提高, 但是召回率却有所下降. 但是朴素贝叶斯贝叶斯方法对各个属性的独立性要求很高, 而在实际中很难满足. 此外朴素贝叶斯的维数一般很高, 对内存的需求很大.

从以上方法看出, 目前聊天内容的过滤方法中基于机器学习的方法精确度比其他模型高, 且灵活度很强, 但是现有的机器学习方法的效率不高, 准确率需要提升, 而且大多数机器学习方法在文本分类任务上对文本的表示都是使用基于词频或者 one-hot 分布式表示方法, 这种方法的方法忽略了文本的具体内容及文本中各个词之间的上下文关系, 导致每个文本的表示维度非常大, 处理效率低且误判率很高. 为了客服以上问题, 在基于 SVM 模型的基础上, 本文使用了一种 Doc2Vec 与 SVM 结合的方法对聊天内容进行分类, 即首先使用 Doc2Vec 模型将聊天内容转化成密集数值向量表示, 然后使用 SVM 对转化后的向量进行训练分类, 最后对测试数据进行预测分类. 实验表明, 这种方法能有效提高分类结果且具有很好的泛化性能.

本文的结构如下: 第 2 节介绍了本文模型的整体框架流程, 第 3 节介绍了 doc2vec 模型和 SVM 模型, 第 4 节介绍了实验过程并对实验结果进行了分析比较, 第 5 节对整篇文章做了总结.

2 模型框架介绍

本文的模型的框架图 1 所示.

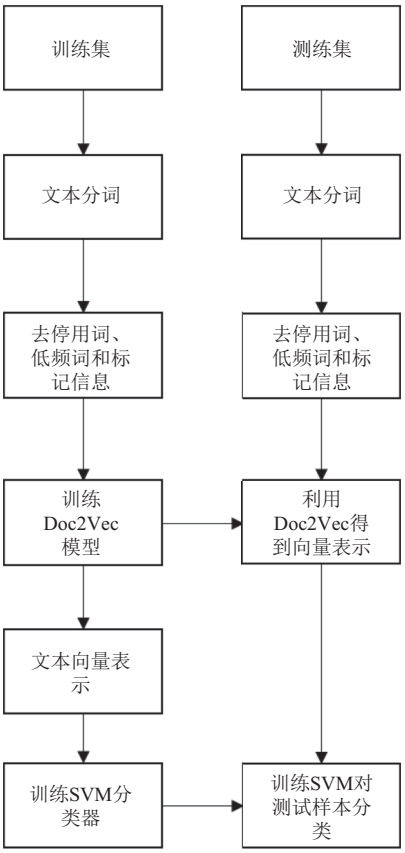

图 1 模型的总体框架

即首先对于数据集进行划分得到训练集和测试集, 训练集用来训练模型, 测试集用来评估模型, 然后分别对训练集和测试集进行中文分词, 去掉停用词、低频词将聊天内容转换成词组的形式, 然后应用 Doc2Vec 模型将聊天内容的词组形式转换成密集向量表示. 最后使用训练集的数据训练 SVM 分类模型, 并使用 SVM 模型对测试样本进行分类, 评估模型效果.

3 模型介绍

3.1 Doc2Vec 模型

Doc2Vec 也称 Word Embeddings, 是 Quoc Le 和 Tomas Mikolov[8]提出的一种处理可变长度文本的总结性方法, 是一种将句子向量转换成密集向量表示的无监督学习方法[9]. 不同于传统的稀疏式表示方法, 该模

型是一种高效的算法模型, 它能将文本或句子表征成密集实值向量, 即通过深度学习的方法将每个文本或句子映射成 K 维向量 (K 为模型的超参数, 可以通过交叉验证调整), 因此文本或句子之间的运算就可以转化为 K 维向量空间中的向量运算, 而向量空间上的相似度可以用来表示文本语义上的相似.

Doc2Vec 试图在给定上下文和段落向量的情况下预测单词的概率. 为了训练出句子的向量表示, 在句子的训练过程中, 句子的 id 保持不变, 共享同一个句子向量. Doc2Vec 模型可以在对语言模型进行建模的同时获得单词和句子在向量空间上的表示. 这种方法与基于 one-hot 的表示方法相比, 充分地利用词的上下文内容, 句子的语义信息更加丰富, 能更有效地提高分类精度.

其中主要的思想就是利用一个包含输入层-隐层-输出层三层的分类神经网络对句子进行建模, 即利用句子中词的上下文信息去预测该词, 常用的方法为 Continuous Bag of Words, 简称为 CBOW.

CBOW 的目标为根据上下文来预测当前词语的概率, 并利用神经网络作为分类算法, 初始时, 为每一个单词和句子随机生成一个 N 维向量, 经过训练之后可以获得每个词及句子的最优向量.

首先取一个合适的语境窗口, 输入层读入窗口内的词, 将它们的向量加和在一起, 作为隐藏层的输出传给输出层. 输出层是一个巨大的二叉树, 其中叶节点代表语料库中所有的词. 而这整棵二叉树构建的算法就是 Huffman 树.

具体模型如图 2 所示, 该模型的语境窗口为 5, 表示对于句子中的任意一个词, 使用该词周围的 4 个词及该词所在句子的 id 去预测该词. 其中 $W(t-2)$, $W(t-1)$, $W(t+1)$, $W(t+2)$ 分别表示当前词的前面第二个词、前面第一个词、后面的第一个词和后面的第二个词. 相应的 V 表示对应词的向量表示. SV 表示当前句子的向量表示, 其维数与词向量的维数一致. 隐层是一个累加层, 其节点数与词向量的维度相同, 将输入层的向量累加起来作为隐藏层的输出, 最后一层是一个 softmax 层, 在一个句子的训练过程中, 句子的向量 SV 保持不变, 相当于每次在预测单词的概率时, 都利用了整个句子的语义.

在预测阶段, 给待预测的每个句子分配一个不同的 id, 词向量和输出层的 softmax 的参数保持训练阶段得到的参数不变, 重新利用梯度下降训练待预测的句子. 训练完之后即可以得到带预测句子的句子向量.

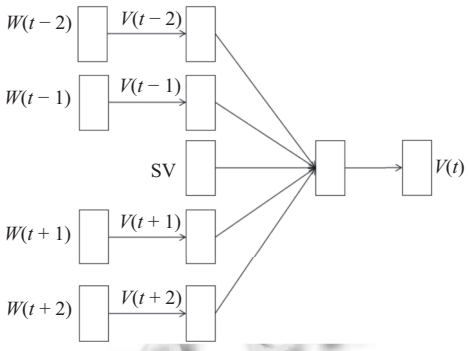

图 2 CBOW 模型

在本文中, 我们选择的语境窗口为 3, 即选择前后相邻的词及句子向量来预测该词, 词和句子的维度选择为 200, 然后将这三个的向量 (初始为随机的 200 维向量) 加和在一起, 隐藏层有 200 个节点, 与输入层全连接, 输出层为一个巨大的二叉树 (Huffman 树), 叶子节点代表语料中所有的词, 根据 Huffman 树可以得到每个词的二进制编码. 隐层的每一个节点都会跟二叉树的内节点有连接边. CBOW 的输出值就是预测二进制编号的每一位, 目标函数是使得预测的二进制编码概率最大. 在训练的过程中, 与一般的神经网络的训练不同, 该网络中输入层的三个向量也为需要更新的参数, 在模型的训练过程中一起更新直到收敛, 因此在训练完成后, 即可以得到每个句子和每个词的密集向量表示.

3.2 SVM 分类器

支持向量机[9–11](Support Vector Machine, SVM) 是在 20 世纪 90 年代以来发展起来的一种统计学习方法, 在解决小样本学习、非线性及高维模式识别问题中表现较好. SVM 主要采用结构风险最小化原则来训练学习并使用 VC 维理论来度量结构风险.

给定训练样本集:

$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$

其中 x_i 表示第 i 个样本的输入变量, 在本文中是指第 i 条聊天内容通过 Doc2Vec 模型训练后得到的句子向量, 表示该条聊天内容对应的标签, y_i 取值为 0 或 1, 其中 1 表示该条聊天内容是需要拦截的内容, 0 表示正常内容. 在样本线性可分的情况下, SVM 的原理是预找到具有“最大间隔”的划分超平面, 即该分类器不但能将两类样本分开, 而且还要使两类的间隔最大, 即优化

目标函数为:

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \quad \text{s.t.} \quad y_i(\omega^{\mathrm{T}}x_i+b)\geq 1, i=1,2,\cdots,m \tag{1}$$

但是在实际问题中, 原始的样本空间可能是非线性的. 所以需要引进非线性映射函数$\varphi(x)$将输入空间中的样本映射到高维特征空间中, 使得映射后的高维特征空间中的样本线性可分, 然后再在高维特征空间构造线性最优超平面. 由于高维空间中的内积运算极为耗时, SVM 引入了核函数$\kappa(x_i,x_j)$来代替高维空间中的内积运算. 本文选择高斯函数作为 SVM 的核函数, 其高斯核函数形式为:

$$\kappa(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|_2^2}{2\sigma^2}\right) \tag{2}$$

其中, σ 为高斯核的带宽. Drucker 和 Androutsopoulos 等人在垃圾邮件过滤中使用支持向量机的方法[8,12].

由于 SVM 具有良好的泛化性能, 且通过核函数可以处理低维线性不可分的情况, 避开高维空间的复杂性,直接用内积函数, 很适合对聊天文本进行分类, 所以本文选择 SVM 作为后续的聊天内容分类器. 由 CBOW 模型可以得到每个句子的密集向量表示, 维度为 200, 将句子向量与其相应的标签相结合得到 SVM 的训练数据, 然后训练 SVM 模型.

4 实验过程及结果分析

4.1 初始数据的处理

本文的所有的数据都来自于真实的聊天记录, 在数据预处理阶段, 由于数字对聊天内容具有很强的干扰性, 但是去掉所有数字对结果可能造成影响, 因此本文采用折中的办法, 使用数字替换的方法, 将聊天内容中连续出现的数字统一都用其字符长度代替, 比如将“电话 128324”替换为“电话 6”, 一方面可以有效减少词表中词的数量, 另一方面又能充分利用数字信息. 然后去掉聊天内容中的一些特殊符号, 作为数据的初步处理.

初步处理之后, 本文使用 python 中的 jieba 分词对处理后的聊天内容进行分词处理, 得到每个聊天内容的词组表示, 接下来使用停用词表去掉词组中的停用词, 本文使用的停用词表是由哈工大停用词表、四川大学机器智能实验室停用词库、百度停用词表等组合而成的, 共包含 2792 个停用词. 对于处理完之后为空的聊天内容, 则将该条内容删除, 将该条内容视为正常聊天内容处理. 对于新的预测样本, 如果经过同样的处理之后其内容为空, 则直接将该样本分类为正常的聊天内容而不需要进行后续的模型预测.

4.2 模型的训练及参数设置

对于数据预处理后的数据, 使用 Doc2Vec 模型进行训练, 可以得到每条聊天内容的数值向量表示及其标签. 由于聊天系统中的语句一般都较短, 所以本文使用的 Doc2Vec 的语境窗口为 3, 选择的词向量及句子向量的空间维度为 200, 即对于第 i 条聊天内容, 通过 Doc2Vec 模型之后可以得到一个 201 维的数值向量, 其中前 200 维表示的是该条聊天内容的数值向量表示记为x_i, 最后一维表示的是该条句子对应的标签y_i, 其中y_i的取值为 0 或 1, 表示该条聊天内容是否应该被拦截. 令 X 表示所有样本的向量集合, 即 X= $\{x_1,x_2,\cdots,x_m\}$, 其中 m 表示数据集样本的总数量, Y= $\{y_1,y_2,\cdots,y_m\}$表示每个样本对应的标签.

接下来对聊天内容进行分类, 即对数据 X,Y 进行训练得到分类模型. 本文使用基于高斯核的 SVM 模型进行分类, 为了测试模型的效果, 本文将数据集划分为训练集和验证集, 训练集用来训练 SVM 模型, 验证集用来测评模型的效果. 为了充分利用训练样本的信息, 本文使用 70% 的数据作为训练集, 30% 的数据作为验证集, 由于本文数据样本不平衡, 正负样本的比例接近 7:3, 所以本文采用分层抽样的方法生成训练集和测试集, 即在正样本中随机抽样 70% 的样本, 在负样本中随机抽样 30% 的样本作为训练样本集, 剩下的样本作为测试样本集.

整个数据集共有 218 356 条数据, 其中正反样例的分布如表 1 所示.

表 1 数据集中正负样本分布

总样本数	反例样本数	正例样本数
218 356	31 236	187 120

本文中反例表示需要拦截的聊天信息, 正例表示不需要拦截的正常信息. 最终选择的训练样本集和测试样本集中数据的分布如表 2 所示.

表 2 训练集及测试集样本分布

	正样本数	负样本数
训练集	130 984	21 865

测试集	56 136	9371

4.3 评估函数的选择

为了评价模型的好坏, 需要使用评估函数[13–16].评估函数是用来评价模型性能的函数, 表示模型预测结果与真实数据结果之间的差别, 差别越小表示模型的性能越好, 所以评估函数可以用来比较多个模型之间的性能. 在介绍测评函数之前先介绍混淆矩阵, 对于分类模型, 其分类结果的混淆矩阵如表 3 所示.

表 3 混淆矩阵

真实情况	预测结果	
	正例	反例
正例	*TP*(真正例)	*FN*(假反例)
反例	*FP*(假正例)	*TN*(真反例)

即 *TP* 表示真实样例为正例模型预测结果为正例的样例数, *FN* 表示真实样例为正例模型预测为反例的样例数, *FP* 表示真实样例为反例模型预测结果为正例的样例数, *TN* 表示真实样例为反例模型预测结果为反例的样例数.

由于本文样本分布不平衡, 且误分类带来的损失也不一样, 因为在聊天内容拦截中, 将正常的聊天内容误分类为需拦截的内容造成的损失比将需要拦截的内容误判为正常的内容大. 所以本文选择 *F*1 指标作为度量模型精确度的标准, *F*1 度量的计算公式如下:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

由于本文主要关注的是反例, 即需要被拦截的信息, 所以用 *P* 表示反例的查准率即被模型预测为反例的所有样例中真实反例的比例, *R* 表示查全率, 即反例被成功预测为反例的比例, 具体 *P* 和 *R* 的计算公式如下:

$$P = \frac{TN}{TN + FN} \tag{4}$$

$$R = \frac{TN}{TN + FP} \tag{5}$$

4.4 实验结果及分析

本文得模型得到的结果的混淆矩阵如表 4 所示.

表 4 模型预测结果混淆矩阵

真实样例	预测结果	
	正例	反例
正例	55 124	1012
反例	954	8417

由表 4 可以得出表 5 所示的各种评价指标, 由表 5 可以得到该模型的预测准确率为 97%, 查准率为 89.27% 表示模型预测为反例的样本中真正反例所占的比例为 89.27%, 查全率为 89.82% 表示真是的反例中被模型预测出来的比例为 89.82%. 为了比较模型的结果, 可对于相同的数据集我们分别使用了朴素贝叶斯方法及基于 Doc2Vec 的 logistic 模型, 其中朴素贝叶斯方法是基于文本的 one-hot 表示, 即将每个聊天记录表示一个维度为词典长度的向量, 向量的取值为 0 或 1, 其中 0 表示该聊天记录包含了相应的词, 1 表示该聊天记录没有包含相应的词, 然后通过 TF-IDF 方法进行降维, 选取具 TF-IDF 值高的词作为最终的变量, 最终选择的变量有 5321 个. 三个模型的最终结果如表 6 所示.

表 5 模型评价指标 (单位: %)

准确率	*P*	*R*	*F*1
97	89.27	89.82	89.54

表 6 各模型的实验结果对比

模型	维度	准确率 (%)	*P*(%)	*R*(%)	*F*1(%)
Doc2Vec SVM	201	97	89.27	89.82	89.54
Doc2Vec Logistic	201	94.39	85.57	73.09	78.8
朴素贝叶斯	5322	96.27	92.56	80.35	86.03

由表 6 可以看出, 本文的 Doc2vec SVM 模型与 Doc2Vec Logistic、朴素贝叶斯相比具有更高的准确率、查准率、查全率及 *F*1 值.而且维度只有 201 维, 远远小于朴素贝叶斯的 5322 维. 虽然朴素贝叶斯的准确率为 96.17%, 查准率为 92.56%, 但是其查全率只有 80.35%, 即朴素贝叶斯倾向于将样本预测为正样本. Logistic 由于模型过于简单, 导致欠拟合, 所以各项指标都较低. 因此, 可以得出结论本文的 Doc2Vec SVM 模型在提高了分类精度的同时有效的减小了数据维度, 提高了分类的效率, 此外还可以通过训练得到每个词的向量表示.

5 结论

本篇文章使用了基于 Doc2Vec 与 SVM 的方法, 对用户的聊内容进行分类, 首先使用 Doc2Vec 对聊天内容进行处理, 将聊天内容表示为数值向量, 然后再使用 SVM 进行分类. 实验表明, 这种方法能有效的提高分类精度, 能有效的识别垃圾聊天内容或广告, 从而大

大提高了聊天内容拦截过滤的效率.

参考文献

1 Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 2002, 34(1): 1–47. [doi: 10.1145/505282.505283]

2 Androutsopoulos I, Paliouras G, Karkaletsis V, *et al*. Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach. Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases. Lyon, France. 2000. 1–13.

3 Sahami M, Dumais S, Heckerman D, *et al*. A Bayesian approach to filtering junk e-mail. Madison, WI, USA: AIAA, 1998.

4 Rennie JDM. Ifile: An application of machine learning to e-mail filtering. Proceedings of KDD Workshop on Text Mining. Boston, MA, USA. 2000.

5 石霞军, 林亚平, 陈治平. 基于最小风险的贝叶斯邮件过滤算法. 计算机科学, 2002, 29(8): 50–51, 46.

6 McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization. Menlo Park, CA, USA. 1998. 41–48.

7 Leopold E, Kindermann J. Text categorization with support vector machines. How to represent texts in input space? Machine Learning, 2002, 46(1-3): 423–444.

8 Le Q, Mikolov T. Distributed representations of sentences and documents. Proceedings of the 31st International Conference on Machine Learning. Beijing, China. 2014. II-1188–II-1196.

9 Kiros R, Zemel RS, Salakhutdinov R. A multiplicative model for learning distributed text-based attribute representations. Proceedings of Advances in Neural Information Processing Systems. Montreal, Quebec, Canada. 2014. 2348–2356.

10 Cherkassky V. The nature of statistical learning theory. IEEE Transactions on Neural Networks, 1997, 8(6): 1564. [doi: 10.1109/TNN.1997.641482]

11 Joachims T. Text categorization with support vector machines: Learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning. London, UK. 1998. 137–142.

12 Lan M, Tan CL, Low HB, *et al*. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. Proceedings of the Special Interest Tracks and Posters of the 14th International Conference on World Wide Web. Chiba, Japan. 2005. 1032–1033.

13 Drucker H, Wu DH, Vapnik VN. Support vector machines for Spam categorization. IEEE Transactions on Neural Networks, 1999, 10(5): 1048–1054. [doi: 10.1109/72.788645]

14 Drummond C, Holte RC. Cost curves: An improved method for visualizing classifier performance. Machine Learning, 2006, 65(1): 95–130. [doi: 10.1007/s10994-006-8199-5]

15 Zhou B, Yao YY, Luo JG. Cost-sensitive three-way email spam filtering. Journal of Intelligent Information Systems, 2014, 42(1): 19–45. [doi: 10.1007/s10844-013-0254-7]

16 Provost F, Fawcett T. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. Newport Beach, CA, USA. 1997. 43–48.