

Contents

基于内容挖掘的垃圾短信分类算法研究 2

 前言 2

 相关工作 2

 文本特征提取 3

 文本分类算法 5

 分类结果评价 9

 实验 10

 分类结果对比与分析 17

 在线检测系统 18

 小组成员分工 19

基于内容挖掘的垃圾短信分类算法研究与实现

前言

随着近几年我国移动通信技术以及基础设施建设的快速发展，智能手机用户数量飞快增加。根据工信部《2014 年通信营业同级公报》显示，截止到 2014 年，我国移动电话已达到 12.8 亿部，2014 年我国手机短信发送总量已达到 7630.5 亿条。人们在享受手机短信带来的便利的同时，也受到了垃圾短信的骚扰，有的甚至造成经济损失。据《2015 上半年中国移动互联网安全报告》显示，中国手机用户半年接收到的垃圾短信总数近 200 亿条。不法分子为了追求经济利益，发送大量中奖诈骗、冒充银行扣款诈骗、网络购物诈骗、非法金融活动等垃圾短信，不仅耗费网络资源，影响在正常短信的传输，微信手机数据的安全，更是让广大手机用户遭受经济损失。根据中国互联网协会公布的《中国网民权益用户调查报告（2015）》显示，近一年来，全国网民经济财产损失高达 805 亿元。垃圾短信的自动识别和过滤愈发必要。

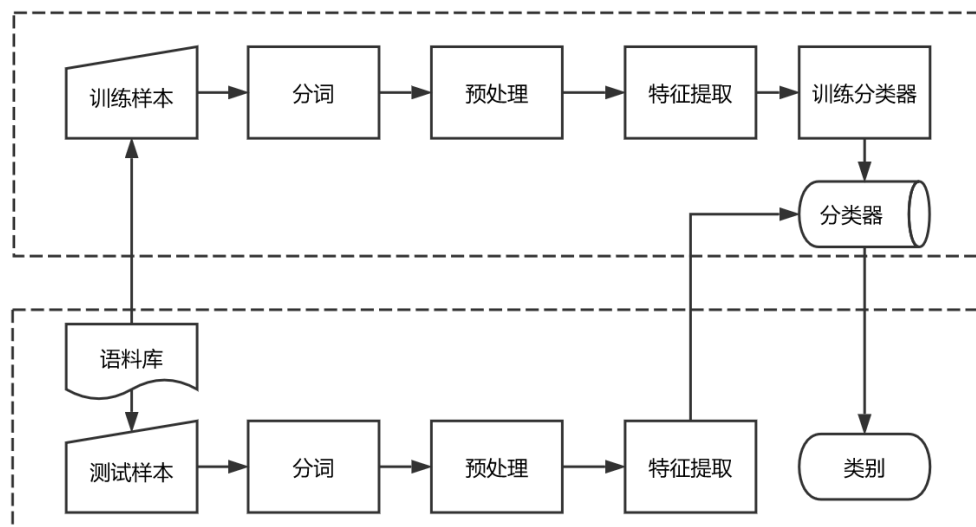
垃圾短信分类技术^[1] 主要黑白名单分类技术、基于规则的分类技术、基于关键字的分类技术和基于内容挖掘的分类技术等，其中基于内容挖掘的分类技术不仅分类效果好，而且拥有自学习的能力。早期对基于文本挖掘垃圾短信分类的研究包括：2004 年文献^[2] 中提出了支持向量机适用于垃圾短信分类问题、2005 年文献^[3] 考虑使用 K-NN 分类器，但都未作出具体的评估，直到 2006 年文献^[4] 在两个垃圾短信数据集上进行测试，证实了其有效性，并且支持向量机的效果更好。2009 年文献^[5] 提出将 K-NN 作为多过滤分类器中的一个环节，首先短信经过黑白名单的过滤之后再经过一个粗糙集分类，如果粗糙集分类将此短信视为垃圾短信，则被丢进 K-NN 做最后的确认。在一个包括 550 条垃圾短信和 200 条非垃圾短信的数据集上，当 $K=12$ 时，这个复合分类器比单个 K-NN 分类器更加快和准确。2010 年文献^[6] 提出了一个计算短信“垃圾系数”的简单索引模型。2011 年，文献对 13 种有监督学习方发做出了比较，包括朴素贝叶斯、线性 SVM、最小描述长度分类器、K-NN、决策树 C4.5、PART 等。结果发现线性 SVM 精确度可以达到 97.64%，垃圾短信上的召回率为 83.1%。改进朴素贝叶斯和改进 C4.5 以及 PART 精确度达到 97.5%。

本报告对基于内容挖掘的垃圾短信分类常用算法进行探讨、实现、对比和分析。

相关工作

基于内容挖掘的短信分类技术流程如下图所示：

样本被分为训练样本和测试样本两大部分并基于语料库分别提取特征，构造一个分类器并利用训练样本



的特征训练分类器，最后将测试样本的特征输入分类器完成分类。其中最为关键的两大重要步骤：提取文本特征、构建文本分类器。

文本特征提取

文本特征提取包括分词、预处理、文本表示、特征提取等几个关键环节。

分词

文本分类首先要对文本进行分词处理。英文有天然的空格分隔符，而中文的词与词之间并没有分个标记，因此要先将短信文本进行分词处理，讲短信风格成若干词语的组合。分词算法主要有基于统计、基于词典、基于语法等。目前，很多中文分词的平台和库效果足以达到可以正常使用。

预处理

文本预处理主要是对文本进行停用词、标点符号、非法字符过滤，是分词后对文本做的一次优化。在中文中虚词、连接词、感叹词、介词、副词等出现频率高，但却没有实质意义，反而会影响分类的准确性，因此要去掉。

文本表示

将文本进行分词和预处理之后，就要将文本表示成对应的特征项，表示成计算机所能识别的模型。常用的特征表示法主要有向量空间模型表示法、布尔逻辑型表示法、LDA 生成模型等。

向量空间模型表示法是将文本视为多个词语的集合，特征项在文本中出现的频率决定了短信文本所属类别，而这些词的先后顺序不被关注，每个特征词对应文本特征向量中的一个元素，从而将短信文本表示成一个由特征项的权重所构成的向量，权重一般与频率相关。该模型简单高效，在实际应用中使用较多。

布尔逻辑型表示法则是采用布尔值来表示一段文本中是否出现了某个值。若文本 d_i 中出现了特征词 w_j ，则对应的关联矩阵中的元素 $r_{ij} = 0$ ，否则 $r_{ij} = 1$ 。此方法简单便捷，但不能体现某词在某文本中的使用频率。

LDA (Latent Dirichlet Allocation) 是一种文档主题生成模型，也成为一个三层贝叶斯概率模型。认为“以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语”。

特征提取

通过文本表示得到的向量往往是高维的，这会导致需要分类器花更多的时间和样本继续学习，而且中文文本具有无结构化的特点，且大部分词对分类的贡献很小，有的甚至严重影响分类结果。特征提取是指通过函数映射（或变换）的方法，选择代表意义较强、分类性较好的特征项进行文本表示，组合成新的向量。常用的特征选择方法^[1]有：特征频度 (TF) []、文档频率 (DF)、特征熵 (TE)、互信息 (MI)、信息增益 (IG)^[7]、 χ^2 统计量 (CHI)^[8]，相关系数 (CC)、特征权 (TS)、期望交叉熵 (ECE)^[9]、文本证据权 (WET) 和几率比 (OR) 等 11 种。

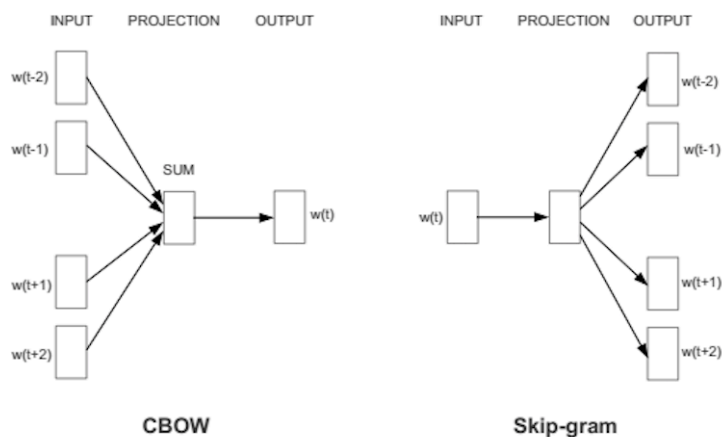
其中，CHI 的主要思想是认为词条与类别之间符合 χ^2 分布，词条的 χ^2 统计量表示词条对某个类别的贡献大小。 χ^2 统计量越高，词条和类别之间的独立性越小、相关性越强，即词条对此类别的贡献越大。词条 i 对应类别 j 的 χ^2 统计量的计算公式如下：

$$\chi_{ij}^2 = \frac{n \times (n_{11} \times n_{22} - n_{12} \times n_{21})^2}{(n_{11} + n_{12}) \times (n_{21} + n_{22}) \times (n_{11} + n_{21}) \times (n_{12} + n_{22})}$$

其中 n_{11} 、 n_{12} 、 n_{21} 和 n_{22} 分别表示词条 i 在类别 j 中出现的频数、词条 i 在类别 j 外的其他类别中出现的频数、除词条 i 外的其他词条在类别 j 中出现的频数、除词条 i 外的其他词条在除类别 j 外的其他类别中出现的频数， n 为所有词条的频数总和。 χ^2 统计量的计算方法说明了词条对类别贡献程度。

Word2vec 是 Google 在 2013 年年中开源的一款将词表征为实数值向量的高效工具，其利用深度学习的思想，通过训练，把文本表示为一个 K 维向量。Word2vec 是 Mikolov 等所提出模型的一个实现，可以用

来快速有效地训练词向量。Word2vec 包含了两种训练模型，分别是 CBOW 和 Skip-gram，如下图所示



从上图中可以看出, CBOW 和 Skip-gram 模型均包含输入层, 投影层和输出层。其中, CBOW 模型通过上下文来预测当前词, Skip-gram 模型则通过当前词来预测其上下文。Word2vec 考虑了词的上下文, 语义信息更加地丰富。然后利用词向量进而构造语句特征。

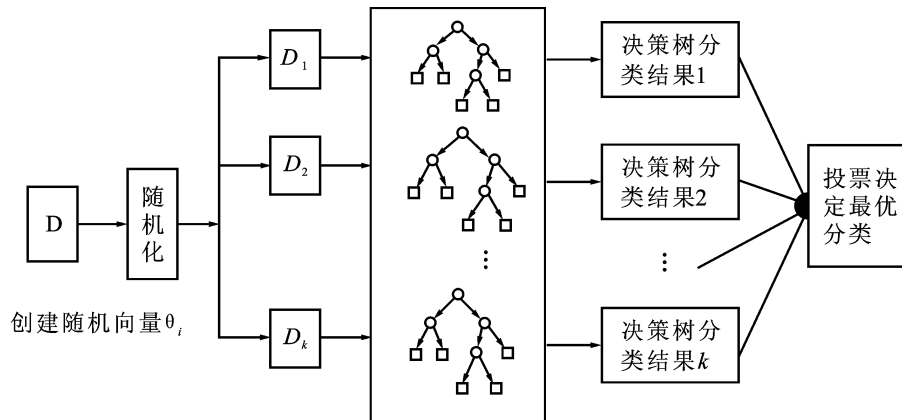
文本分类算法

随机森林

随机森林分类 (RFC)^[10] 是由很多决策树分类模型 $h(X, \Theta_k), k = 1, \dots$ 组成的组合分类模型, 且参数集 Θ_k 是独立同分布的随机向量, 在给定自变量 X 下, 每个决策树分类模型都由一票投票权来选择最优的分类结果。RFC 的基本思想: 首先, 利用 bootstrap 抽样从原始训练集抽取 k 个样本, 且每个样本的样本容量都与原始训练集一样; 其次, 对 k 个样本分别建立 k 个决策树模型, 得到 k 种分类结果; 最后, 根据 k 种分类结果对每个记录进行投票表决, 决定其最终分类, 如下图:

RF 通过构造不同的训练集增加分类模型间的差异, 从而提高组合分类模型的外推预测能力。通过 k 轮训练, 得到一个分类模型序列 $h_1(X), h_2(X), \dots, h_k(X)$, 再用它们构成一个多分类模型系统, 该系统的最终分类结果采用简单多数投票法, 最终的分类决策为:

$$H(x) = \arg \max_Y \sum_{i=1}^k I(h_i(x) = Y)$$



其中, $H(x)$ 表示组合分类模型, h_i 是单个决策树分类模型, Y 表示输出变量 (或称目标变量), $I(*)$ 为示性函数, 上式说明了随机森林是如何通过多数投票决策的方式来确定最终分类结果的。

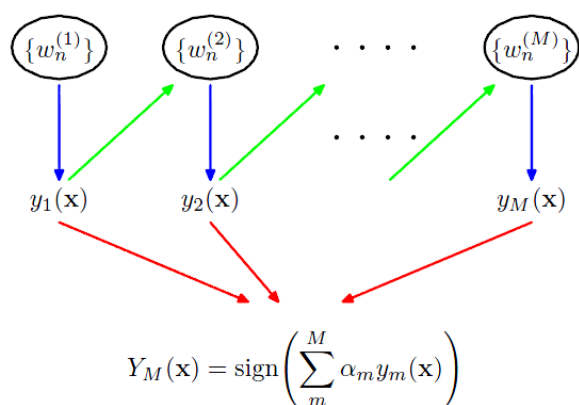
大量的理论和实证研究都证明了 RF 具有很高的预测准确率, 对异常值和噪声具有很好的鲁棒性, 且不容易出现过拟合, RF 是一种自然的非线性建模工具, 是最常用的数据挖掘方法之一。

Gradient Boosting

Boosting 是一种可以提高任意给定学习算法准确率的算法。它起源于 Valiant 提出的近似正确 (Probably Approximately Correct, PAC) 学习模型。Kearns 和 Valiant^[11] 证明当训练样本足够多的情况下, 一系列效果仅仅比随机猜测性能强的弱分类器可以组合成任意性能很好的组合分类器。Boosting 算法的思想十分简单, 如下图所示, 即对一份数据建立 M 个模型, 一般来说这些模型均比较简单, 也就是我们经常说的弱分类器。每次分类时, 都将上一次分错的数据的权重提高一些再次进行分类。这样, 最终得到的分类器组合不论在训练数据还是测试数据上都可以得到比较好的效果。

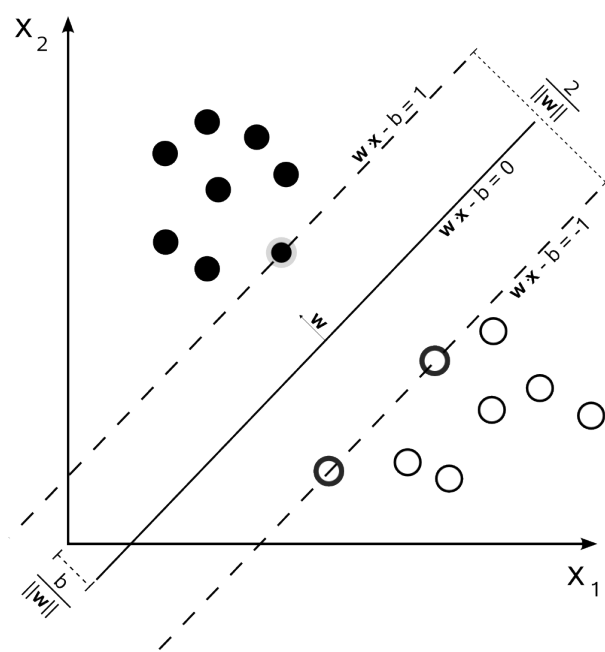
Gradient Boosting 是 Boosting 方法的一种, 其主要思想是, 每一个模型都是建立在上一次所建立模型的损失函数的梯度下降方向上的。其中, 损失函数 (Loss Function) 表示的是模型的不靠谱程度, 损失函数越大说明模型越容易出错。为了不停地改进模型, 需要持续令损失函数下降, 而使其下降最快捷的方式就是让损失函数沿着自己的梯度下降。因此 Gradient Boosting 与传统 Boosting 相比在分错样本的权重更新上有较大的区别。

支持向量机



支持向量机以统计学习理论为基础，可以很好地处理回归问题、分类问题和判别分析等诸多问题。并在预测和综合评价等问题中也表现出很好的效果。本文主要将支持向量机用于分类问题。

支持向量机的原理在于寻找一个最优分类超平面能够在满足分类要求的同时最大化超平面两侧的空白区域。如下图



线性 SVM 原理示意图

已两类线性分类为例，给定训练数据集 $(x_i, y_i), i = 1, 2, \dots, l, x \in R^n, y \in \{1, -1\}$ ，将超平面记做 $(\omega \cdot x_i) + b = 0$ ，其中 ω 是一个 n 维向量， b 是一个常量。为使分类将所有样本分类正确并有分类间隔，

需要满足约束： $y_i(w \cdot x_i + b) \geq 1 \quad i = 1, 2, \dots, n$

可以计算出分类间隔为 $2/|\omega|$ ，因此求解最优化超平面可以转化成如下约束式进行求解：

$$\min \Phi(\omega) = \frac{1}{2}|\omega|^2 = \frac{1}{2}\omega^T \omega$$

为了解决这个问题，引入拉格朗日函数： $L(\omega, b, a) = \frac{1}{2}|\omega|^2 - a(y((\omega \cdot x) + b) - 1)$ 其中， $a_i > 0$ 为拉格朗日乘数。最优解由拉格朗日函数的鞍点决定，最优化解应在鞍点处 ω 和 b 的偏导为 0，将该问题转换成相应的对偶问题即：

$$\begin{aligned} \max Q(a) = & \sum_{j=1}^l a_j - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l a_i a_j y_i y_j (a_i \cdot x_j) \\ \text{s.t.} \quad & \sum_{j=1}^l a_j y_j = 0 \quad j = 1, 2, \dots, l, a_j \geq 0, j = 1, 2, \dots, l \end{aligned}$$

计算最优解为 $a^* = (a_1)$ 最优权值向量和最优偏移量，分别为：

$$\begin{aligned} \omega^* &= \sum_{j=1}^l a_j^* y_j x_j \\ b^* &= y_i - \sum_{j=1}^l y_j a_j^* (x_j \cdot x_i) \end{aligned}$$

其中，下标 $j \in j | a_j^* > 0$ 。得到最优分类超平面 $(\omega^* \cdot x) + b^*$ ，最优分类函数为：

$$\begin{aligned} f(x) &= \text{sign}\{(\omega^* \cdot x) + b^*\} = \\ &\text{sign}\{(\sum_{j=1}^l a_j^* y_j (x_j \cdot x_i)) + b^*\}, x \in R^n \end{aligned}$$

对于非线性问题，支持向量机的主要思想是先将输入数据映射到一个高维空间中，使数据在高维空间中线性可分。设从 x 做从输入空间到 R^n 到高维特征空间 H 的变换为 Φ ，得：

$$x \rightarrow \Phi(x) = (\Phi_1(x), \Phi_2(x), \dots, \Phi_l(x))^T$$

以特征向量 $\Phi(x)$ 代替输入向量 x ，可以得到非线性最有分类函数为：

$$\begin{aligned} f(x) &= \text{sign}\{(\omega^* \cdot \Phi(x)) + b^*\} = \\ &\text{sign}\{(\sum_{j=1}^l a_j^* y_j (\Phi(x_j) \cdot \Phi(x_i)) + b^*\}, x \in R^n \end{aligned}$$

而寻找合适的映射函数 Φ 是非常复杂, 不容易实现。仔细观察公式, 可以发现最优分类超平面只与内积 $\langle x_i, x_j \rangle$ 有关, 因此支持向量机引入核函数来完成从线性到非线性的变换。常用的核函数有:

1. 多项式核函数: $K(x_i, K_j) = (x_1^T x_1)^d$
2. Gauss 径向基核函数: $K(x_i, K_j) = \exp(-q|x_1 - x_2|^2)$
3. 其他一些核函数有 B-样条函数, Fourier 核函数, 双曲正切函数等。

Logistic 回归

logistic 回归是一种广义线性回归 (generalized linear model), 因此与多重线性回归分析有很多相同之处。它们的模型形式基本上相同, 都具有 $w'x + b$, 其中 w 和 b 是待求参数, 其区别在于他们的因变量不同, 多重线性回归直接将 $w'x + b$ 作为因变量, 即 $y = w'x + b$, 而 logistic 回归则通过函数 L 将 $w'x + b$ 对应一个隐状态 p , $p = L(w'x + b)$, 然后根据 p 与 $1-p$ 的大小决定因变量的值。如果 L 是 logistic 函数, 就是 logistic 回归, 如果 L 是多项式函数就是多项式回归。

Logistic 回归二值分类问题模型为:

$$P(Y = 1|x) = \frac{\exp(w \cdot x + b)}{1 + \exp(w \cdot x + b)} P(Y = 0|x) = \frac{1}{1 + \exp(w \cdot x + b)}$$

分类结果评价

样本的分类结果分为四种, 描述和表示如下表:

	预测为正样本	预测为负样本
标注为正样本	TP(true positive)	FN(false negative)
标注为负样本	FP(false positive)	TN(true negative)

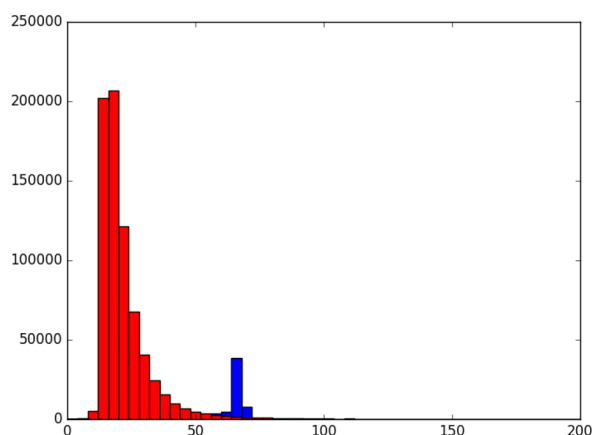
分类的评价准则主要有以下四种:

1. 正确率 (Accuracy) = $\frac{TP+TN}{TP+FN+FP+TN}$ 。
2. 精确率 (Precision) = $\frac{TP}{TP+FP}$ 。
3. 召回率 (Recall) = $\frac{TP}{TP+FN}$ 。
4. $F1=2 \times \frac{Precision \times Recall}{Precision+Recall}$ 。

实验

长度统计分类

首先，通过对数据的观察我们可以很明显的看出垃圾短信和非垃圾短信的长度差异较大。我们作出垃圾短信和非垃圾短信长度的分布图如下（红色为非垃圾短信，蓝色为垃圾短信）



可以看出绝大多数非垃圾短信的长度都在 50 一下，垃圾短信长度基本在 50 以上。猜想通过设置一个合理的长度阈值就可以作出较为准确的预测。下图为仅以长度做一维特征随机抓取 1/5 的数据进行测试。可以看出能得到很好的结果。

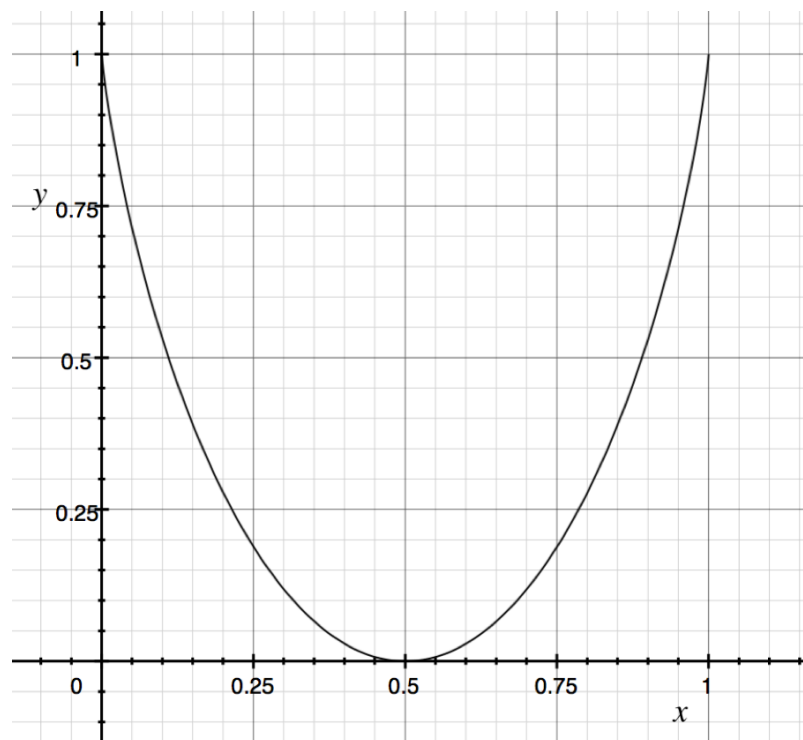
	precision	recall	f1-score	support
0.0	0.996999	0.999646	0.998320	143891
1.0	0.996757	0.973121	0.984797	16109
avg / total	0.996974	0.996975	0.996959	160000

信息熵 + 随机森林

• 特征选择

我们通常在“文档-单词”类型的推荐或是分类中会很自然的在分词完成后将标点符号和停用词去掉。但是对于短信而言，由于其长度过短（非垃圾短信的平均长度仅有 21），那么一些标点符号和停用词包含的信息比例就有很大的提高，从而不能被忽略。

接着进行词袋的压缩，从 34 万的单词压缩至一万左右。显然，我们是希望这一万个词能够尽可能的有更好的分类效果。因此，我们需要这些词能够尽可能的有更高的“区分度”，同时我们希望这些词具有更高的频数。所谓“区分度”即如果一个词出现，我们希望要么它全出现在“0”类（非垃圾短信）中，要么全出现在“1”类（垃圾短信）中。根据这种想法我们用出现条件下的熵值来刻画。构造关于词在垃圾短信中出现的频率 p 的函数 $f = 1 + p\log(p) + (1 - p)\log(1 - p)$ 。 f 的函数图像为下图，函数值越大表明该词的区分度越高。

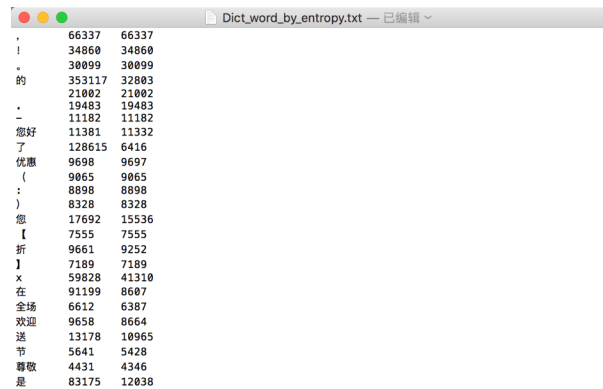


如果 $p > 0.5$ 则分类为“1”，否则为“0”。那么我们的用“ $t = 1$ ”来作为频数的代表。最终我们用 $t * f$ 对单词进行排序。结果如下（数据依次为词、出现频数、在垃圾短信中出现频数）。

一个有趣的事情是，前三个标点符号（逗号、感叹号和句号）具有很强的分类型。尤其是逗号，出现在了 6.6 万（总数为 8 万）的垃圾短信中，且说明只要有逗号，一定是垃圾短信。所以说通过该方法，选出的词都是具有很强的“分类性”的特征。

• 分类

采用随机森林方法进行分类。



,	66337	66337
!	34860	34860
.	30099	30099
的	353117	32803
	21002	21002
.	19483	19483
-	11182	11182
您好	11381	11332
了	128615	6416
优惠	9698	9697
(9065	9065
:	8898	8898
)	8328	8328
您	17692	15536
【	7555	7555
折	9661	9252
】	7189	7189
x	59828	41310
在	91199	8607
全球	6612	6387
欢迎	9658	8664
送	13178	10965
节	5641	5428
尊敬	4431	4346
是	83175	12038

• 实验结果

用以上滤词的方法在随机森林上的实验结果过如下,

特征数 = 1001, ntree = 10

	Precision	Recall	F1
负样本	0.998715	0.999423	0.999069
正样本	0.994815	0.998516	0.991655
平均	0.998323	0.998325	0.998323

特征数 = 1001, ntree = 100

	Precision	Recall	F1
负样本	0.998826	0.999472	0.999149
正样本	0.995255	0.989509	0.992374
平均	0.998467	0.998469	0.998467

特征数 = 101, ntree = 10

	Precision	Recall	F1
负样本	0.996964	0.999666	0.998314
正样本	0.996962	0.972810	0.984730
平均	0.996962	0.996962	0.996946

特征数 = 11, ntree = 10

	Precision	Recall	F1
负样本	0.997061	0.999527	0.998292
正样本	0.995683	0.973679	0.984558
平均	0.996922	0.996925	0.996910

通过利用随机森林的分类，可以看出只要使用强“分类性”的特征时（如短信长度，上述的重要单词），分类效果都会很好。

χ^2 统计量 + Gradient Boosting

- 特征选择

本方法具体采用的为 χ^2 统计量方法，使用 χ^2 统计度量 term 和类别独立性的缺乏程度， χ^2 越大，独立性越小。本实验采用 χ^2 统计量方法选取相关性最高的 5000 个特征。

- 分类

本实验分类采用 Gradient Boosting 方法，借助 xgboost 进行实现。xgboost 的全称是 eXtreme Gradient Boosting。它是 Gradient Boosting Machine 的一个 c++ 实现。创建之由为受制于现有库的计算速度和精度，xgboost 最大的特点在于，它能够自动利用 CPU 的多线程进行并行，同时在算法上加以改进提高了精度。

- 实验结果采用 χ^2 统计量方法选取相关性最高的 5000 个特征，使用 xgboost 进行 5-fold 交叉验证结果如下：

序号	F1
1	0.987383925712
2	0.987745868158
3	0.986261345269
4	0.987070464388
5	0.987418438388
avg	0.987176008383

其中参数设置为：

Parameter	Setting
objective	binary logistic
max_depth	50
num_boost_round	50
learning_rates	0.5

Word2vec+ 支持向量机

• 特征选择

使用 Word2vec 对短信数据进行处理。首先把所有的短信分词，然后使用 Word2vec 计算词向量。建立 word-特征的词典，用于构造文本特征。把文本分词，每个词的 word2vec 计算的词向量叠加求平均值作为文档的特征。这种处理方法简单快速，经试验，只需要 30-300 维特征就可以有比较好的效果，特征维度低，训练分类器较快。

• 分类方法

对垃圾短信分类问题可以使用 SVM 算法对短信进行分类。使用库 sklearn 进行实现。另外实现了一个 C++ 的 SVM 分类器，使用 SMO 进行求解。

• 分类结果

sklearn svm 5 折交叉验证结果，训练数据 64 万条，验证数据 16 万条。5 次实验结果如下：

	precision	recall	F1
0	1.00	1.00	1.00
1	1.00	0.98	0.99
<i>avg/total</i>	1.00	1.00	1.00

	precision	recall	F1
0	1.00	1.00	1.00
1	1.00	0.98	0.99
<i>avg/total</i>	1.00	1.00	1.00

	precision	recall	F1
0	1.00	1.00	1.00
1	1.00	0.98	0.99
<i>avg/total</i>	1.00	1.00	1.00

	precision	recall	F1
0	1.00	1.00	1.00
1	1.00	0.98	0.99
<i>avg/total</i>	1.00	1.00	1.00

	precision	recall	F1
0	1.00	1.00	1.00
1	1.00	0.98	0.99
<i>avg/total</i>	1.00	1.00	1.00

实验次	precision	recall	F1
1	0.98360346	0.9966771	0.9900971
2	0.98315658	0.99696356	0.9900119
3	0.98281485	0.99656576	0.9896425
4	0.98261791	0.98261791	0.9893053
5	0.98293515	0.9973555	0.9900928
<i>avg</i>	0.9830256	0.9967293	0.98982995

svm(C++)，用 C++ 实现了 SVM 算法，用 200 条数据训练，100 条数据测试，word2vec 处理出来 50 维特征，实验结果如下：

precision	recall	F1
0.970000	0.928571	0.948833

稀疏表示 + Logistic/SVM

• 特征选择

如果使用简单的使用分词后统计词的数量、熵特性，如 DF (Document Frequency)、TF-IDF (term frequency - inverse document frequency)、信息增益 (Information Gain, IG) 等方法作为特征，势必会产生维度灾难。

本方法一开始使用 TF-IDF 作为特征，而后经过数据过滤，删除停用词等操作后，将实际有用的词数量降到了 32k 个。由于一条短信所包含的词的数量十分有限，对于一般的短信来说删选后的词的数量不会超过 10 个。这样我们的训练矩阵就会很稀疏（具体表现为一条短信的特征维度为 32k，但是其中非零维度一般不超过 10 个）。为此我们使用矩阵的稀疏表示来完成特征的提取。

• 分类使用 Logistic 回归和 SVM 作为二值分类器。

• 实验结果

这里选取前 10 万条短信上作为训练集，采用 5-fold 交叉验证，结果如下：

Logistic

序号	F1
1	0.9915
2	0.9914
3	0.9908
4	0.9919
5	0.9910
avg	0.9913

未使用核函数，直接使用线性 SVM

序号	F1
1	0.9921
2	0.9917
3	0.9930
4	0.9915
5	0.9922
avg	0.9921

分类结果对比与分析

通过多种方法对垃圾短信分类，如果将所有的样本进行训练，分类结果都比较好，F1 都达到 0.987 以上，SVM 甚至达到 1。各方法达到的最好 F1 如下表：

方法	平均 F1
长度统计分类	0.997
信息熵 + 随机森林	0.998
χ^2 统计量 + Gradient Boosting	0.987
Word2vec+ 支持向量机	0.990
稀疏表示 + Logistic	0.991
稀疏表示 + 支持向量机	0.992

经实验，发现垃圾短信本身的长度普遍较长、高频词在正常短信中使用极少等特征非常明显，因此比较容易区分，能达到较好的效果。

在线检测系统

基于对垃圾分类算法的研究，我们构建了在线垃圾短信检测系统，地址为<http://nd-fe.zale.site>。由于 χ^2 统计量 +Gradient Boosting 模型收敛快，占用空间小，为最终线上使用的模型。检测场景如下：

在线垃圾短信检测

中科院计算所软件所学生联手打造
使用机器学习相关技术
垃圾短信检测的准确度高达99%

共查扣违法三轮车304辆

检测

检测结果为非垃圾短信，耗时
0.0454981327057s

在线垃圾短信检测

中科院计算所软件所学生联手打造
使用机器学习相关技术
垃圾短信检测的准确度高达99%

《依林美容》三.八.女

检测

检测结果为垃圾短信，耗时
0.0242760181427s

小组成员分工

成员	工作
张乐	用 TF-IDF 提取特征并对提取的特征进行初步降维；使用 SVM 和 LR 进行分类，对比 SVM 和 LR 的性能；部分实验文档编写；在线系统前端实现。
陈敬伍	研究垃圾短信与非垃圾短信长度分布的差异；设计“区分度”公式并对分词数据用筛词法选出应该用来做训练的词集；用随机森林模型做训练，取得 0.998 的 F1 值；书写部分实验文档
胡耀康	主要基于 χ^2 统计量 (Chi-Square) 对特征进行筛选；使用 xgboost 训练模型，模型 F1 值为 98.8%；负责在线系统后端实现；编写相关文档。
张卫民	使用 word2vec 处理短信数据，构造特征 100 维特征；使用 SVM 对短信进行分类；用 C++ 实现 SVM 算法；书写部分文档。
李贝贝	查找文献了解、总结垃圾短信分类研究现状；介绍实验所用核心算法；对实验结果总结、对比和分析；负责文档撰写和整合。

[1] 钟延辉. 基于文本挖掘的垃圾短信过滤方法 [D][D]. 电子科技大学, 2009.

[2] XIANG Y, CHOWDHURY M, ALI S. Filtering mobile spam by support vector machine[C]//CSITeA' 04: Third International Conference on Computer Sciences, Software Engineering, Information Technology, E-Business and Applications. International Society for Computers; Their Applications (ISCA), 2004:

1 - 4.

[3] HEALY M, DELANY S J, ZAMOLOTSKIKH A. An assessment of case base reasoning for short text message classification[C]//Conference papers. 2004: 42.

[4] GÓMEZ HIDALGO J M, BRINGAS G C, SÁNZ E P, 等. Content based SMS spam filtering[C]//Proceedings of the 2006 ACM symposium on Document engineering. ACM, 2006: 107 - 114.

[5] LONGZHEN D, AN L, LONGJUN H. A new spam short message classification[C]//2009 First International Workshop on Education Technology and Computer Science. 2009.

[6] LIU W, WANG T. Index-based online text classification for sms spam filtering[J]. Journal of Computers, 2010, 5(6): 844 - 851.

[7] YANG Y, PEDERSEN J O. A comparative study on feature selection in text categorization[C]//ICML. 1997, 97: 412 - 420.

[8] RONGLU L, JIANHUI W, XIAOYUN C, 等. Using maximum entropy model for Chinese text categorization [J][J]. Journal of Computer Research and Development, 2005, 1: 22 - 29.

[9] SCHÜTZE H, HULL D A, PEDERSEN J O. A comparison of classifiers and document representations for the routing problem[C]//Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1995: 229 - 237.

[10] 李欣海. 随机森林模型在分类与回归分析中的应用 [J]. 应用昆虫学报, 2013, 50(4): 1190 - 1197.

[11] KEARNS M, VALIANT L. Cryptographic limitations on learning Boolean formulae and finite automata[J]. Journal of the ACM (JACM), ACM, 1994, 41(1): 67 - 95.