Gesture Recognition and its Application

Ziyang Wang
ShanghaiTech University
wangzy4@shanghaitech.edu.cn

Kefei Wu
ShanghaiTech University
wukf@shanghaitech.edu.cn

Donghao Zhao
ShanghaiTech University
zhdh@shanghaitech.edu.cn

Yiduo Gu
ShanghaiTech University
guyd@shanghaitech.edu.cn

Abstract

In this project we focus on how to successfully classify four kinds of SMSs including spams, notifications, verifications and normal messages with different features including sender and message information, and finally classify with five methods including Naive Bayes Classifier, Logistic Regression, Support Vector Machine, Random Forests and Gradient Boosting Decision Tree, to reach the goal to teach the program to classify automatically.

1. Introduction

We receive many different kinds of text messages in our daily life, one of the simplest way to classify them of which is to divide them into spam and non-spam messages. We hope to further divide non-spam messages into more detailed categories. One of the widely used natural language processing task in different business problems is "Message Classification". The goal of Message classification is to automatically classify the text message into one or more defined categories.

In this project, the message classification is implemented by python. This project classifies messages based on python's machine learning library scikit-learn and the complete Chinese word segmentation tool jieba.

2. Motivation

As described above, the importance of filtering the useless messages and garbage information from the strangers, and the classifier that iOS system can only classify the messages according to the address book, which is naïve enough for classifying that may not work well to our expecting result. What we're going to do here is to do the real 'classify' that the application should understand more than what is garbage or not, but also the kind of it such as verifying code or relatives' communication, by analyzing the information given in the text and number, which will largely improve our using experience.

3. Table of Contents

Text Classification is an example of supervised machine learning task since a labelled dataset containing text documents and their labels is used for train a classifier. Our message classification project is composed of three main components:

1. Dataset preparation: The first step is the dataset preparation step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then split into training and testing sets.

2. Feature engineering: The next step is the feature engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data.

3. Model training: The final step is the Model Building step in which a machine learning model is trained on a labelled dataset.

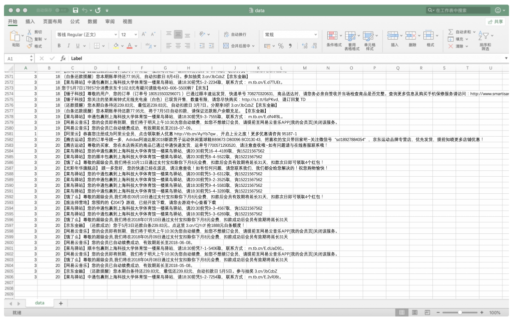
4. Testing result analysis: the result of different features along with different models. We evaluate the performance and analyze the reason of it.

5. Possible improvement: In this article, we will also look at the different ways to improve the performance of message classifiers.

4. Dataset preparation

We've collected more than 2500 messages from our own mobile phones and manually labeled them with 4 categories: normal message, spam, verification code, and notification.

Our dataset reflects the real distribution of each categories and their features. features from our dataset.



	A	B	C
2571	3	18	（白条还款提醒）您本期账单待还77.95元，自动扣款日 8月4日。参加抽奖 3.cn/JbCdbZ【京东金融】
2572	3	18	【菜鸟驿站】中通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭5-2-2234取，联系方式： m.tb.cn/E.d7TUEt。
2573	3	18	您于5月7日17时57分消费京东卡102.8元有疑问请致电400-606-5500转7【京东】
2574	3	18	【锤子科技】尊敬的用户，您的订单（订单号 180515503296071）已通过顺丰速运发货，快递单号 708270320631，商品送达时，请您务必亲自签收并当场检查商品是否完整。查询更多信息及购买手机保修服务请访问：http://www.smartisar
2575	3	18	【锤子科技】您关注的坚果闹钟式无线充电座（白色）已现货开售，数量有限，请您尽快购买：http://s.t.tt/6zPKvd，退订回复 TD
2576	3	18	（还款提醒）您本期白条待还239.83元，最低还239.83元，自动扣款日 3月7日。分期享8折 3.cn/JbCdbZ【京东金融】
2577	3	19	（白条还款提醒）您本期账单待还款77.95元，将于7月5日自动扣款，请保证还款账户余额充足。【京东金融】
2578	3	19	【菜鸟驿站】申通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭9-3-7555取，联系方式： m.tb.cn/E.dN4f8L。
2579	3	19	【网易云音乐】您的会员即将到期，我们将于明天上午10:30为您自动续费，如您不想续订会员，请提前至网易云音乐APP[我的会员页]关闭该服务。
2580	3	19	【网易云音乐】您的会员已自动续费成功，有效期延长至2018-07-09。
2581	1	19	【阿里云】恭喜您注册成为阿里云会员，点击领取新人优惠 http://tb.cn/AyYb7qw，开启上云之旅！更多优惠请咨询 95187-1
2582	3	20	【腾吉运动】您的订单号顾一多，Adidas阿迪达斯2018新款男子运动休闲篮球鞋BB9673 DB0096 BC0130 43，把喜欢的宝贝带回家吧~关注微信号 "sd18927884054"，京东运动品牌专营店，优先发货，提前知晓更多店铺优惠！
2583	3	20	【腾吉运动】尊敬的买家，您在本店购买的商品已通过申通快递发货，运单号7700571293520，请注意查收哦~如有问题请与在线客服联系哦！
2584	3	20	【菜鸟驿站】您的申通包裹到上海科技大学体育馆一楼菜鸟驿站，请20:30前凭16-4-4189取，询15221567562
2585	3	20	【菜鸟驿站】您的顺丰包裹到上海科技大学体育馆一楼菜鸟驿站，请20:30前凭8-4-5522取，询15221567562
2586	3	20	【饿了么】尊敬的超级会员,我们将在10月11日通过支付宝扣除你下月8元会费，扣款后会员有效期将延长31天，扣款次日即可领取4个红包！
2587	3	20	【光彩年华旗舰店】顾一多您好，您的快递已经在派送，请注意查收！如有任何问题，请您联系我们，我们都会给您解决的！祝您购物愉快！
2588	3	20	【菜鸟驿站】您的中通包裹到上海科技大学体育馆一楼菜鸟驿站，请20:00前凭5-3-6312取，询15221567562
2589	3	20	【菜鸟驿站】您的申通包裹到上海科技大学体育馆一楼菜鸟驿站，请20:00前凭9-2-3525取，询15221567562
2590	3	20	【菜鸟驿站】您的申通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭9-4-5583取，询15221567562
2591	3	20	【菜鸟驿站】您的中通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭5-4-3289取，询15221567562
2592	3	20	【饿了么】尊敬的超级会员,我们将在09月10日通过支付宝扣除你下月8元会费，扣款后会员有效期将延长31天，扣款次日即可领取4个红包！
2593	3	20	【旅法师营地】您预约的《2047》游戏，已经开放下载，请您去游戏中心查看下载
2594	3	20	【菜鸟驿站】您的申通包裹到上海科技大学体育馆一楼菜鸟驿站，请20:30前凭9-3-4567取，询15221567562
2595	3	20	【菜鸟驿站】您的中通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭5-3-6269取，询15221567562
2596	3	20	【饿了么】尊敬的超级会员,我们将在2018年07月10日通过支付宝扣除你下月8元会费，扣款成功后会员有效期将延长31天
2597	3	20	【京东金融】（还款成功）您于5月3日还款白条239.83元。点这里 3.cn/CjYrJf 抢1888元白条额度！
2598	3	20	【网易云音乐】您的会员即将到期，我们将于明天上午10:30为您自动续费，如您不想续订会员，请提前至网易云音乐APP[我的会员页]关闭该服务。
2599	3	20	【饿了么】尊敬的超级会员,我们将在2018年05月09日通过支付宝扣除你下月8元会费，扣款成功后会员有效期将延长31天
2600	3	20	【网易云音乐】您的会员已自动续费成功，有效期延长至2018-06-08。
2601	3	20	【菜鸟驿站】顺丰包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭7-1-5406取，联系方式： m.tb.cn/E.dUaD91。
2602	3	20	【网易云音乐】您的会员即将到期，我们将于明天上午10:30为您自动续费，如您不想续订会员，请提前至网易云音乐APP[我的会员页]关闭该服务。
2603	3	20	【饿了么】尊敬的超级会员,我们将在2018年04月08日通过支付宝扣除你下月8元会费，扣款成功后会员有效期将延长31天
2604	3	20	【网易云音乐】您的会员已自动续费成功，有效期延长至2018-05-08。
2605	3	20	【京东金融】（还款提醒）您本期白条待还239.83元，最低还239.83元，自动扣款日 5月5日。参与抽奖 3.cn/JbCdbZ
2606	3	20	【菜鸟驿站】中通包裹到上海科技大学体育馆一楼菜鸟驿站，请18:30前凭5-2-7254取，联系方式： m.tb.cn/E.2vRJ6t。

Our dataset is stored in csv format.

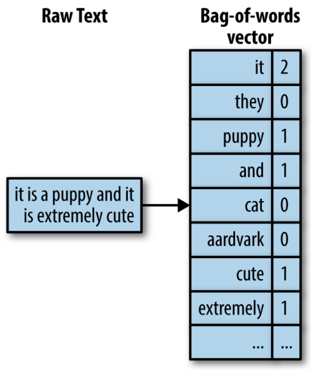
5. Feature engineering

The next step is feature engineering. In this step, raw text data will be transformed into feature vectors and new features will be created using the existing dataset. We will implement the following different ideas in order to obtain relevant

5.1. Bag of Words Representation

First, we split each message into combination of words, and use each words that more than 2 characters as a local feature. We put local features of all message into a "vocabulary bag" as a dictionary.

For each message, we extract the local features in dictionary and count the number of the occurrences of each local feature to form a histogram. By $CountVectorizer$ of $sklearn$, we could store each message's histogram into a sparse matrix.



An example of bag of words representation.

5.2. TF-IDF vectors as features

Term Frequency(TF) is the number of times a given word appears in a message.Inverse Document Frequency(IDF) diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

$$TF_t = \frac{t\ occurrences\ in\ a\ message}{Total\ number\ of\ words\ in\ a\ message} \tag{1}$$

$$IDF_t = log(\frac{Number\ of\ messages}{Messages\ include\ word\ t\ +1}) \tag{2}$$

$$TF - IDF = TF * IDF \tag{3}$$

TF-IDF score represents the relative importance of a term in the document and the entire corpus.TF-IDF vectors tends to filter out common words and retain important words.

TF-IDF vectors can be generated at different levels of input tokens (words, characters, n-grams) by using $TfidfTransformer$ of $sklearn$ on bag of words representation matrix.

5.2.1 Word Level TF-IDF

Matrix representing tf-idf scores of every term in different documents

5.2.2 N-gram Level TF-IDF

N-grams are the combination of N terms together. This Matrix representing tf-idf scores of N-grams

5.2.3 Character Level TF-IDF

Matrix representing tf-idf scores of character level n-grams in the corpus

5.3. The length of the sender's number as features

In real life, normal messages are always sent by humans, whose phone numbers are usually 11 digits (or 13 digits when the country number is included in front), while notifications from banks or airplane companies are always from a 5-digit number.

Therefore, we use the length of sender's number as a feature outside the text.

5.4. Length Encoding

Inspired by label encoder, we use label 1, 2, 0 to label 5-digit number, 11(13)-digit number and other numbers.

```
def lengthencode(x):
    if x == 5:
        return 1
    elif x == 11:
        return 2
    elif x == 13:
        return 2
    return 0
```

6. Model building

The final step in the text classification framework is to train a classifier using the features created in the previous step. There are many different choices of machine learning models which can be used to train a final model. We will implement following different classifiers for this purpose:

6.1. Naive Bayes Classifier

Naive Bayes is a classification technique based on Bayes' Theorem, which is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

We've implemented a Naive Bayes classifier by ourselves, based on this fundamental principle.

The table below shows the relationship between arguments in Laplace smoothing equation and final results.

Weight	Assumed Probability	Accuracy
1	0.5	88.12%
1	0.25	88.51%
0.5	0.25	89.27%
0.5	0.5	88.51%
2	0.5	87.35%
2	0.25	87.36%

Bayes' theorem:

$$P(A|B) = \frac{P(A) \times P(B|A)}{P(B)} \tag{4}$$

6.2. Logistic Regression

As the assumption of Naive Bayes classifier, Logistic regression also assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Given the sigmoid function:

$$g(z) = \frac{1}{1+e^{-z}} \tag{5}$$

The probability of a message belong to a class given it's feature x and a parameter θ is:

$$\begin{aligned} P(Y=1|x;\theta) &= g(\theta^T x) \\ &= \frac{1}{1+e^{-\theta^T x}} \\ &= (g(\theta^T x_i))_i^y * (1-g(\theta^T x_i))^{1-y_i} \end{aligned} \tag{6}$$

Joint distribution of all n samples is :

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} P(y_i|x_i,\theta) \\ &= \prod_{i=1}^{n} (g(\theta^T x_i))_i^y * (1-g(\theta^T x_i))^{1-y_i} \end{aligned} \tag{7}$$

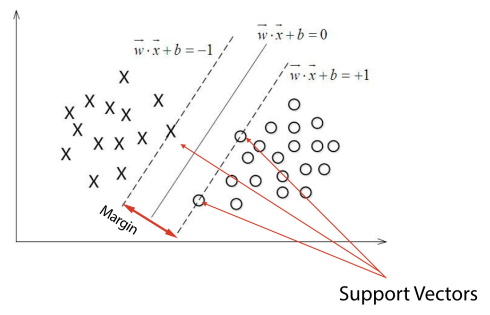
Take a log-likelihood function:

$$\begin{aligned} l(\theta) &= log(L(\theta)) \\ &= \sum_{i=1}^{n} y_i log(g(\theta^T x_i)) + (1-y_i) log(1-g(\theta^T x_i)) \end{aligned} \tag{8}$$

The maximum likelihood estimation is the θ required to take $l(\theta)$ to the maximum value, which can be solved by the gradient ascending method.

6.3. Support Vector Machine

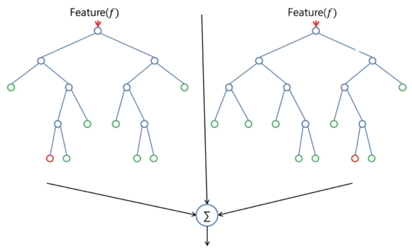
Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. The model extracts a best possible hyper-plane / line that maximizes the margin between the positive and negative examples.


$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=0$
$\vec{w}\cdot\vec{x}+b=+1$
Margin
Support Vectors

Positive: $\vec{w} \cdot \vec{x} + b \geq 1$
Negative: $\vec{w} \cdot \vec{x} + b \leq -1$
Support vector: $\vec{w} \cdot \vec{x} + b = \pm 1$

6.4. Random Forests

Random forests construct a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees to get a more accurate and stable prediction.

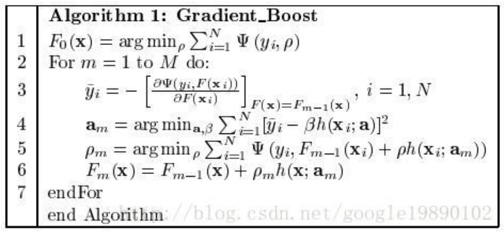


Random Forest Algorithm

6.5. Gradient Boosting Decision Tree

Boosting algorithm iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier.

In gradient boosting, comparing to boosting algorithm, each new model is established to reduce the residual of the previous model to the gradient direction instead of adding with previous models.



```
Algorithm 1: Gradient_Boost
$F_0(\mathbf{x}) = \arg\min_\rho \sum_{i=1}^{N} \Psi(y_i, \rho)$
For $m = 1$ to $M$ do:
    $\tilde{y}_i = -\left[\frac{\partial\Psi(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \ i = 1, N$
    $\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
    $\rho_m = \arg\min_\rho \sum_{i=1}^{N} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$
    $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$
endFor
end Algorithm
```

Gradient Boosting Decision Tree

7. Result analysis

Feature	Classifier	Accuracy
Bag of Words	Naive Bayes	90.80%
Bag of Words	Logistic Regression	92.72%
Bag of Words	SVM	51.72%
Bag of Words	Random Forest	89.27%
Bag of Words	Gradient Boosting Decision Tree	90.04%

Feature	Classifier	Accuracy
TF-IDF	Naive Bayes	89.27%
TF-IDF	Logistic Regression	90.42%
TF-IDF	SVM	51.72%
TF-IDF	Random Forest	91.18%
TF-IDF	Gradient Boosting Decision Tree	90.80%

From the result above, we can know that TF-IDF and Bag of Words basically has no difference, and TF-IDF' s results is worse than Bag of Words feature.

Feature	Classifier	Accuracy
Bag of Words+Sender	Naive Bayes	91.18%
Bag of Words+Sender	Logistic Regression	93.10%
Bag of Words+Sender	SVM	51.72%
Bag of Words+Sender	Random Forest	88.89%
Bag of Words+Sender	Gradient Boosting Decision Tree	90.80%

Feature	Classifier	Accuracy
Bag of Words+Sender with encoding	Naive Bayes	90.80%
Bag of Words+Sender with encoding	Logistic Regression	93.10%
Bag of Words+Sender with encoding	SVM	51.72%
Bag of Words+Sender with encoding	Random Forest	89.66%
Bag of Words+Sender with encoding	Gradient Boosting Decision Tree	90.42%

We found that use sender's number as feature will help to improve training results, but encode sender's number doesn't help at all. We guess maybe the following reasons: 1. Our dataset is lack of normal message (which meets real condition). 2. Banks sends too much spam messages.

Feature	Classifier	Dataset	Accuracy
Bag of Words	Naive Bayes	1600	87.89%
Bag of Words	Logistic Regression	1600	92.99%
Bag of Words	Naive Bayes	2600	90.80%
Bag of Words	Logistic Regression	2600	92.72%

After training with a smaller dataset, we found that even if logistic regression performs better in these two dataset, Naive Bayes performs better with the dataset increasing. So we think Naive Bayes might have better performance than logistic regression when the dataset is larger.

We also realize our own Naive Bayes function and Logistic Regression classifier by ourselves, due to some reasons relating to the smoothing function arguments, but which mostly has reached our expectations.

Feature	Classifier	Dataset	Accuracy
Bag of Words	Naive Bayes (by ourselves)	1600	85.35%
Bag of Words	Naive Bayes (by ourselves)	1600	88.12%
Bag of Words	Logistic Regression (by ourselves)	1600	33.14%

8. Possible improvement

While the above framework can be applied to message classification problems, but to achieve a good accuracy some improvements can be done in the overall framework.

8.1. Text Cleaning

text cleaning can help to reduce the noise present in text data in the form of stopwords, punctuations marks, suffix variations etc.

8.2. NLP features with text feature vectors

In the feature engineering section, we generated a number of different feature vectors, combining them together can help to improve the accuracy of the classifier.

8.3. Ensemble Models

Stacking different models and blending their outputs can help to further improve the results.

We have envisioned it, but it still needs further verification.

References

1.Unsupervised learning of finite mixture models Figueiredo, MAT (Figueiredo, MAT); Jain, AK (Jain, AK) IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 24 3 381-396

2.Short Text Classification in Twitter to Improve Information Filtering Sriram, B (Sriram, Bharath)[1] ; Fuhry, D (Fuhry, David)[1] ; Demir, E (Demir, Engin)[1] ; Ferhatosmanoglu, H (Ferhatosmanoglu, Hakan)[1] ; Demirbas, M (Demirbas, Murat) SIGIR 2010: PROCEEDINGS OF THE 33RD ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH DEVELOPMENT IN INFORMATION RETRIEVAL 841-842

3.Sentiment Analysis of Short Informal Text Kiritchenko, S (Kiritchenko, Svetlana)[1] ; Zhu, XD (Zhu, Xiaodan)[1] ; Mohammad, SM (Mohammad, Saif M.)[1] JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH 50 723-762

4.Rough set-aided keyword reduction for text categorization Chouchoulas, A (Chouchoulas, A); Shen, Q (Shen, Q) APPLIED ARTIFICIAL INTELLIGENCE 15 9 843-873

5.Gradient Boost Decision Tree OR Treelink

6.Baidu Grammar analysis 7. Jie Ba analysis