Research on Classification Algorithm for Blackberry Lily Data

Wenjun Yu, Yu Huang*
College of informatics, Huazhong Agricultural University, Wuhan, Hubei, China
*Corresponding author’s e-mail: yhuang@mail.hzau.edu.cn

Abstract. Blackberry Lily is the dried rhizome of the Iris plant, which has the functions of clearing heat and detoxifying, eliminating phlegm and pharyngeal. The actual structure of its various categories is very different, and this difference is widely used by botanists to build the interspecific relationship of various varieties of Blackberry Lily. Therefore, the classification of Blackberry Lily is of great significance to the research of its evolution. In this paper, we propose several algorithms based on Naive Bayes to classify Blackberry Lily data, and some algorithms are improved.

1. Introduction

Blackberry Lily (Belamcanda chinensis) is a perennial herb whose rhizomes are irregularly massive, obliquely extended, yellow or yellowish brown, with many fibrous roots, and the stem is upright, with a height of 1~1.5 meters and solid. Blackberry Lily’s taste is bitter and cold, which has the functions of clearing away heat and detoxifying, eliminating phlegm and pharyngeal, it is commonly used for heat and poison phlegm and heat, knot, sore throat, phlegm and saliva, coughing and wheezing[1-3]. The actual structures of various Blackberry Lily in nature are very different, these differences are of great significance for the evolutionary research of the iris family and are widely used by botanists to build the interspecies relationship of each iris family[4-8]. For example, the two subclasses of angiosperms can be distinguished by the number of petals: dicotyledonous plants usually have 4 or 5 (or multiples of 4 or 5) petals, while monocotyledonous plants are usually 3 or multiples of 3.

Based on the above, we use several algorithms based on Naive Bayes to classify the simulated data of Blackberry Lily. In all machine learning classification algorithms, Naive Bayes is different from most other classification algorithms[9-12]. For most classification algorithms, such as decision trees, KNN, logistic regression, and support vector machines, they are all discriminant methods, that is, directly learning the relationship between the feature output Y and the feature X, or the decision function Y=f(X), or the conditional distribution P(Y|X). However, Naive Bayes is a generation method, that is, to directly find the joint distribution P(X,Y) of the feature output Y and the feature X, and then use P(Y|X)=P(X,Y)/P(X )inferred. Naive Bayes is very intuitive, the amount of calculation is not large, and it is widely used in many fields.

2. Methods

2.1. Datasets and Preprocessing

The shape of the flowers of Blackberry Lily and irises are most similar. The Iris data is a classic dataset and is often used as an example in the field of statistical learning and machine learning. The dataset contains 150 records in 3 categories, 50 data in each category, and each record has 4 characteristics:
sepal length, sepal width, petal length, petal width, these four characteristics can be used to predict which iris flower belongs to (iris-setosa, iris-versicolour, iris-virginica). Based on the iris dataset, we generated the Blackberry Lily dataset, a total of 150,000 data, each record has similar characteristics, divided into Bc-1, Bc-2, Bc-3 --- a total of 3 categories. As shown in Figure 1 (partial data):

| sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | species |
|-------------------|------------------|-------------------|------------------|---------|
| 6.3               | 3.3              | 6.0               | 2.5              | 2       |
| 7.2               | 3.6              | 6.1               | 2.5              | 2       |
| 5.8               | 2.8              | 5.1               | 2.4              | 2       |
| 6.4               | 3.2              | 5.3               | 2.3              | 2       |
| 7.7               | 2.6              | 6.9               | 2.3              | 2       |
| 6.9               | 3.2              | 5.7               | 2.3              | 2       |
| 7.7               | 3.0              | 6.1               | 2.3              | 2       |
| 6.3               | 3.4              | 5.6               | 2.4              | 2       |
| 6.7               | 3.1              | 5.6               | 2.4              | 2       |
| 6.9               | 3.1              | 5.1               | 2.3              | 2       |
| 6.8               | 3.2              | 5.9               | 2.3              | 2       |

Fig1. The simulated data set of Blackberry Lily.

We divide the simulation data into 70% training data and 30% test data. The python code is as follows:
from sklearn.model_selection import train_test_split
X = df.iloc[:,:4]
y = df.iloc[:,4]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3)

2.2. Algorithms
Naive Bayes is a relatively simple algorithm. Compared with decision tree, KNN and other algorithms, Naive Bayes needs to pay attention to fewer parameters, which is also easier to master[13]. In scikit-learn, there are 3 naive Bayes classification algorithms that are GaussianNB, MultinomialNB and BernoulliNB. Among them, GaussianNB is naive Bayes with a priori Gaussian distribution, MultinomialNB is naive Bayes with a priori polynomial distribution, and BernoulliNB is naive Bayes with a priori Bernoulli distribution.

2.2.1. Gaussian Naïve Bayes. It is used to process continuous variables, requires data to conform to a Gaussian distribution, and has a certain ability to deal with sample imbalance problems. The basic python code is as follows:
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(X_train, y_train)
print(clf.score(X_test, y_test))

Consider that if the amount of data in the training set is very large, it cannot be loaded into memory at one time. At this time, we improve the algorithm to divide the training set into several equal parts, and repeatedly call partial_fit to learn the training set step by step[14]. The final accuracy score is 0.97, and the score before the improvement is 0.91.

2.2.2. Bernoulli’s Naïve Bayes. It can only be used to handle binomial distribution data and is good at processing data sets with short text, text mining. After the data is normalized, the prediction effect becomes lower, and a threshold value needs to be set, which is insensitive to sample imbalance (dumb
variable/binomialization/data characteristics that need to conform to the binomial distribution), but the
data requirements are very high. The basic code is as follows:

```python
from sklearn.naive_bayes import BernoulliNB
clf = BernoulliNB()
clf.fit(X_train, y_train)
clf.score(X_test, y_test)
```

The accuracy score obtained is 0.28. The problem here is the adjustment of the parameter binarize.
This parameter is mainly used to help BernoulliNB to deal with the binomial distribution, which can be
numeric or not entered. If you do not enter it, BernoulliNB considers that each data feature is already
binary. Otherwise, those smaller than binarize will be classified into one category, and those larger than
binarize will be classified into another category[15]. Here our input is 3, and the final score is: 0.45.

### 2.2.3. Polynomial Naïve Bayes
It is used to dealing with discrete variables. The characteristics involved
are often times, frequencies, and counts[16]. The polynomial Naïve Bayes in sklearn does not accept
negative values and is greatly affected by the data structure. The basic code is as follows:

```python
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(X_train, y_train)
print(clf.score(X_test, y_test))
```

MultinomialNB has more parameters than GaussianNB, but there are only 3 in total. Among them,
the parameter alpha is the above constant λ, if you have no special needs, you can use the default of 1.
If you find that the fitting is not good and you need to tune, you can choose a number slightly greater
than 1 or slightly less than 1. Therefore, the final accuracy score after improvement is 0.85, while the
score before improvement is 0.71.

### 3. Results
The classification scenarios applicable to the above three algorithms are different, and the model
selection is mainly based on the data type. In general, if the distribution of sample features is mostly
continuous, it is better to use GaussianNB. If most of the sample features are multivariate discrete values,
MultinomialNB is more appropriate. If the sample features are binary discrete values or very sparse
multivariate discrete values, BernoulliNB should be used. As shown in Table 1:

|                     | GaussianNB | BernoulliNB | MultinomialNB |
|---------------------|------------|-------------|---------------|
| Accuracy score      | 0.91       | 0.28        | 0.71          |
| score(improved)     | 0.97       | 0.45        | 0.85          |

### 4. Discussion
In this article, we use three naive Bayes classification algorithms for the Blackberry Lily simulation data
classification, and make improvements, and get the following conclusions[17-18]:

- The Naive Bayesian model originates from classical mathematical theory and has a stable
classification efficiency.
- It performs well on small-scale data. It can handle multi-classification tasks and is suitable for
incremental training. Especially when the amount of data exceeds the memory, we can do
incremental training in batches.

The main disadvantages of Naive Bayes are:

- It is necessary to know the prior probability that often depends on the hypothesis, there can be
many kinds of hypothetical models, so in some cases, the prediction effect is not good due to
the hypothetical prior model.
Since we determine the posterior probability through priors and data to determine the classification, there is a certain error rate in the classification decision.

It is more sensitive to the expression of input data.

Acknowledgments
This work was financially supported by the Horizontal Subject of Huazhong Agricultural University (Title: Network clinical medical data analysis system).

References
[1] Ni G, Li J Y, Yu D Q. Belamchinenin A, an unprecedented tricyclic-fused triterpenoid with cytotoxicity from Belamcanda chinensis[J]. Organic & biomolecular chemistry, 2018, 16(20): 3754-3759.
[2] Szandruk M, Merwid-Łąd A, Szeląg A. The impact of mangiferin from Belamcanda chinensis on experimental colitis in rats[J]. Inflammopharmacology, 2018, 26(2): 571-581.
[3] Ni G, Shi G R, Li J Y, et al. The unprecedented iridal lactone and adducts of spiroiridal and isoflavonoid from Belamcanda chinensis[J]. RSC advances, 2017, 7(33): 20160-20166.
[4] Song Y Y, Miao J H, Qin F Y, et al. Belachinanes A–D from Belamcanda chinensis: Triterpenoids with an Unprecedented Carbon Skeleton and Their Activity against Age-Related Renal Fibrosis[J]. Organic letters, 2018, 20(17): 5506-5509.
[5] Li J, Ni G, Li L, et al. New iridal-type triterpenoid derivatives with cytotoxic activities from Belamcanda chinensis[J]. Bioorganic chemistry, 2019, 83: 20-28.
[6] Nowak B, Matuszewska A, Szandruk M, et al. Effect of long-term administration of mangiferin from Belamcanda chinensis on bone metabolism in ovariectomized rats[J]. Journal of Functional Foods, 2018, 46: 12-18.
[7] Silva G H, Baldini L S, Tomaz C A, et al. Germination temperatures and treatments to overcome dormancy in Belamcanda chinensis seeds[J]. Pesquisa Agropecuária Tropical, 2017, 47(4): 377-383.
[8] Zhang Y, Yao G, Huang X, et al. Advances in research on the main chemical constituents and pharmacological effects of Belamcanda chinensis (L.) DC[J]. Asian Journal of Traditional Medicines, 2017, 12(5).
[9] Rouder J N, Morey R D, Verhagen J, et al. Bayesian analysis of factorial designs[J]. Psychological Methods, 2017, 22(2): 304.
[10] Benavoli A, Corani G, Demšar J, et al. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis[J]. The Journal of Machine Learning Research, 2017, 18(1): 2653-2688.
[11] Bierkens J, Fearnhead P, Roberts G. The zig-zag process and super-efficient sampling for Bayesian analysis of big data[J]. The Annals of Statistics, 2019, 47(3): 1288-1320.
[12] Fairfield T, Charman A. Explicit Bayesian analysis for process tracing: Guidelines, opportunities, and caveats[J]. Political Analysis, 2017, 25(3): 363-380.
[13] van Doorn J, van den Bergh D, Bohm U, et al. The JASP guidelines for conducting and reporting a Bayesian analysis[J]. 2019.
[14] Caldwell A, Merle A, Schulz O, et al. Global Bayesian analysis of neutrino mass data[J]. Physical Review D, 2017, 96(7): 073001.
[15] Nicholl M, Guillochon J, Berger E. The Magnetar Model for Type I Superluminous Supernovae. I. Bayesian Analysis of the Full Multicolor Light-curve Sample with MOSFiT[J]. The Astrophysical Journal, 2017, 850(1): 55.
[16] Cortes A, Dendrou C A, Motyer A, et al. Bayesian analysis of genetic association across tree-structured routine healthcare data in the UK Biobank[J]. Nature genetics, 2017, 49(9): 1311.
[17] Xiaohua Y, Shengmao G, Chunjuan Y, et al. Diurnal variation of chlorophyll fluorescence parameters in Belamcanda chinensis in summer[J]. South China Forestry Science, 2017 (3): 4.

[18] Li J, Ni G, Liu Y, et al. Iridal-Type Triterpenoids with a Cyclopentane Unit from the Rhizomes of Belamcanda chinensis[J]. Journal of natural products, 2019, 82(7): 1759-1767.