Naïve Bayes Classifier

Seoul AI Meetup

Martin Kersner, 2017/06/10

Introduction

- Classifier.
- Handles multiple classes.
- Nominal values (in case of Multinomial Naïve Bayes Classifier)
- Does not need much of training data.
- Decent classifier, bad estimator.
- Types
  - **Gaussian** Naïve Bayes Classifier
  - **Multinomial** Naïve Bayes Classifier
  - **Bernoulli** Naïve Bayes Classifier
- Example usage
  - Document classification
  - Spam Filtering

How does it work?

$$P(y_1|X) >= P(y_2|X)$$

$$P(y_1|X) < P(y_2|X)$$

binary classifier

$$\mathrm{argmax}_n\ P(y_n|X)$$

multi-class classifier

X represents feature vector.
y and **c** symbols in following slides represent the same thing; **class**.

From (Con)joint probability to Bayes Rule

$$P(A \text{ and } B) = P(A)P(B|A)$$

$$P(B \text{ and } A) = P(B)P(A|B)$$

$$P(A \text{ and } B) = P(B \text{ and } A)$$

$$P(A)P(B|A) = P(B)P(A|B)$$

$$P(B|A) = (P(B)P(A|B)) / P(A)$$

Bayes Rule

Single feature

Likelihood

Class Prior Probability

$$\boxed{P(c|x)} = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

Do not need to compute!

Bayes Rule

Multiple features

$$P(y \mid x_1, \ldots, x_n) = \frac{\overbrace{P(y)P(x_1, \ldots x_n \mid y)}^{\text{joint probability}}}{\cancel{P(x_1, \ldots, x_n)}}$$

P (x$_1$, …, x$_n$, y)

Chain Rule of Conditional Probability

$$
\begin{aligned}
p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\
&= p(x_1 \mid x_2, \dots, x_n, C_k) p(x_2, \dots, x_n, C_k) \\
&= p(x_1 \mid x_2, \dots, x_n, C_k) p(x_2 \mid x_3, \dots, x_n, C_k) p(x_3, \dots, x_n, C_k) \\
&= \dots \\
&= p(x_1 \mid x_2, \dots, x_n, C_k) p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k) p(x_n \mid C_k) p(C_k)
\end{aligned}
$$

Naïvity of Naïve Bayes Classifier

$$
\begin{aligned}
p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\
&\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k)\ \cdots \\
&\propto \boxed{p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,.}
\end{aligned}
$$

$\propto$ represents relation called “is proportional to”.

Multinomial Naïve Bayes Classifier

1. Create vocabulary and convert features accordingly.
2. Compute $p(C_k)$.
   a. Sum number of C_k instances and divide by total number of training examples (scikit-learn *fit_priors=True* parameter).
   b. **OR** set manually based on prior knowledge about class distribution (scikit-learn *class_prior=[...]* parameter).
3. Compute for every x_i, $p(x_i|C_k)$, where x_i is i-th feature.
   a. What is the probability of feature occurrence x_i in class C_k?
   b. Element-wise sum of all feature vectors in from each class.
   c. Normalize. Divide each element by a sum of all feature occurrences from one class.
4. For each class we will end up with vector of probabilities for each feature from vocabulary.

Multinomial Naïve Bayes Classifier

1. Convert features according to vocabulary ⇒ $\mathbf{x_t}$
2. Multiply all relevant $\mathbf{p(x_i|C_k)}$ and $\mathbf{p(C_k)}$ probabilities.
    a. Perform for each class.
    b. Compare.
    c. Select class with the highest **output**.

- Potential issues
    - If any of $p(x|C_k)$ probabilities is zero, then **probability of whole document is zero as well**!
    - Multiplication of small values ⇒ **underflow** ⇒ `ln(a*b) = ln(a) + ln(b)`

scikit-learn MultinomialNB

- For multinomial data distribution.
- Default parameters
  - alpha [= 1.0]
  - fit_prior [= True]
  - class_prior [= None]

```
>>> import numpy as np
>>> X = np.random.randint(5, size=(6, 100))
>>> y = np.array([1, 2, 3, 4, 5, 6])
>>> from sklearn.naive_bayes import MultinomialNB
>>> clf = MultinomialNB()
>>> clf.fit(X, y)
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
>>> print(clf.predict(X[2:3]))
[3]
```

cl-ml [1] naive-bayes-classifier

- Inspired by 4th chapter of Machine Learning in Action [2].
- Multinomial Naive Bayes **Binary** Classifier.

```
(setf nbc (make-instance 'naive-bayes-classifier))
(multiple-value-setq (X y) (get-default-dataset nbc))
(fit nbc X y)
(setf X-test '((my dog has steak)      ; okay sentence
               (your dog is stupid))) ; abusive sentence
(print (predict nbc X-test))
#S(MATRIX :ROWS 1 :COLS 2 :DATA ((0 1)))
```


[1] https://github.com/martinkersner/cl-ml
[2] https://www.manning.com/books/machine-learning-in-action

References

- http://scikit-learn.org/stable/modules/naive_bayes.html
- https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/
- http://greenteapress.com/wp/think-bayes/
- https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- https://en.wikipedia.org/wiki/Joint_probability_distribution