Tsinghua-Berkeley Shenzhen Institute
LEARNING FROM DATA
Fall 2019

Programming Assignment 2

Issued: Friday 11th October, 2019 **Due:** Friday 25th October, 2019

2.1. (4 points) *MLE.* Recall that the naive Bayes model tries to maximize the likelihood of the joint distribution $P(X, Y)$ as:

$$L(\phi_y, \phi_j(x|y)) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$$

where the $\phi_j(x|y) := p(X_j = x|Y = y)$ and $\phi_y := p(Y = y)$ are the parameters required to be estimated for the model. Here, the $x_j, \forall j \in \{1, .., d\}$ is assumed be binary-valued: $x \in \mathcal{X} := \{0, 1\}$, and the label $y \in \mathcal{Y} := \{1, ..., K\}$. Please derive the **maximum likelihood estimates** of the $\phi_j(x|y)$ and ϕ_y for the NB model.

2.2. (6 points) *Naive Bayes.* We have prepared a binary classification dataset drawn from the UCI Adult which contains 1,605 data with 123 features in total. The task here is to build a **Bernoulli naive Bayes** classifier that does inference as:

$$\begin{aligned} p(y = k|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|y = k)p(y = k)}{p(\boldsymbol{x})} \\ &= \frac{p(y = k)\prod_{i=1}^{n} p(x_i|y = k)}{p(\boldsymbol{x})} \end{aligned}$$

The Bernoulli naive Bayes implements the naive Bayes training and classification algorithms for data that is distributed according to **multivariate Bernoulli distributions**; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors. See details in the **naive_bayes.py**.

Notice:

1. Again, use matrix operations other than loops for efficiency. If the running time exceeds 5 minutes, you will get point deductions.
2. You are ought to acquire at least 75% test accuracy, and the correctness of your implementation will also be checked.
3. **Submit your solution of question 2.1 in pdf and the naive_bayes.py together!**

Tips:

1. Do not forget the Laplace smoothing.
2. Use $\log \prod(\cdot) = \sum \log(\cdot)$ to avoid operating on products.
3. Note that the $P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i)$ is used in Bernoulli naive Bayes's decision rule. You can compare your results with the BernoulliNB in **scikit-learn**, if your implementation is right, the results test accuracy shall be the same (or very close).