Some notes about classifier based on comonotonicity

James Cheng Peng (HKU, Email: [u3532960@connect.hku.hk](mailto:u3532960@connect.hku.hk), Repository: <https://github.com/jamespengcheng/Improvement-to-Naive-Bayes-Classifier>)

**Background information:** Naïve Bayes classifier is popular for its comparatively good performance on various datasets and easiness for implementation. However, the largest problem for Naïve Bayes is the “naive” assumption which treats the input variables as mutually independent. Thus, it performs bad when applied to high dimensional datasets in which dependence structure exists among variables.

**Objective:** To surpass Naïve Bayes method.

**Inspiration:** Comonotonicity is used to tackle the problem of mutual dependence among variables. However, we should not assume all variables are comonotonic. Otherwise, we are going to another extreme.

**Methodology:**

1. In general, we still use the idea in Naïve Bayes to compute: Pr(X\_1 ~ X\_n | class = c) \* Pr(class = c). The crucial part is to estimate Pr(X\_1 ~ X\_n | class = c). We denote it as A\_i. For each observation in the test set, we initialize A\_i with 1 and multiply values later.
2. There are three types of variables: continuous, discrete but rankable, discrete but unrankable. For the continuous and discrete rankable variables, we need to do clustering in order to find highly correlated variables. We compute the correlation coefficient matrix and use 1 minus the absolute value of it. This results in a distance matrix. Then we use the Agglomerative Nesting algorithm to do the clustering task. Note that we should specify a minimum correlation min\_corr. The clustering will stop if and only if the shortest distance among all existing clusters is larger than 1 – min\_corr.
3. For the unrankable variables, we simply multiply P(X\_j | class = c) with A\_i, which is the same as that in Naïve Bayes.
4. Now that we have used the original data to do clustering, we can discretize the continuous variables in the training set and store the bins for each continuous variable and each discrete value. Then when doing prediction, we use the bins to do discretization for the data in the test set. Basically, the way of discretization is of vital importance. Here we discretize each continuous variable into 8 categories. The bins are: < mean – 3\*std, mean – 3\*std ~ mean – 2\*std, mean – 2\*std ~ mean – std, mean – std ~ mean, mean ~ mean + std, mean + std ~ mean + 2\*std, mean + 2\*std ~ mean + 3\*std, > mean + 3\*std. Then for each rankable variable X\_j, we store the value of Pr(X\_j = k | class = c) for any possible value k. Note that we should use Laplacian correction to avoid zero probability.
5. Now the variables are all discrete, there are two types of variables: rankable and unrankable. We have dealt with unrankable variables in step 3. For the rankable variables, we compute the probability mass of each cluster. Within the cluster, say, we have variables X\_1 ~ X\_5, we assume that they are comonotonic. We treat X\_1 as the basic variable and for every other variable X\_i, if corr(X\_1, X\_i) < 0, we take the negative of X\_i. Otherwise, keep the X\_i unchanged. Suppose in an observation in the test set, X\_i = x\_i for I between 1 and 5, we take the intersection of five intervals. The infimum is summation from Pr(X\_i = 0 | class = c) to Pr(X\_i = x\_i – 1 | class = c) and the supremum is summation from Pr(X\_i = 0 | class = c) to Pr(X\_i = x\_i | class = c). The length of intersection is the probability mass of this cluster conditional to a certain class c.
6. Conducting the method in step 5 for every observation in the test set, we can get the probability distribution of each test data over all possible classes. Then we choose the largest one to be the predicted class.

**Dataset:** MNIST dataset created by Prof. Yann LeCun in Courant Institute, NYU. More information can be found here: <http://yann.lecun.com/exdb/mnist/>

**Result:** For Gaussian naïve Bayes, the accuracy on MNIST is 55.58%. For comonotonic classifier, the accuracy is 84.57% if min\_corr = 0.9.