Lund University
Faculty of Engineering (LTH)

EDAN95

Applied Machine Learning

Project 6

Author:
Marcel Attar, 941127-2173

The report was handed in on: December 19, 2019

Write a brief (1-2 pages) report that discusses the following issues / answers the following questions. Give proper references in case you consult any material other than your implementation.

Question 1:

In lab 5 you were asked to implement the "statistics based" NBC using counts over the encountered values for the attributes in each class. Why was that an oversimplification and how does the resulting problem relate to your first implementation of the decision tree in lab 2 for the digits data?

The "statistics based" NBC using counts over the encountered values for the attributes in each class is heavily reliant on having a lot of data to produce good results. You especially need a lot of data if the pixel values have a large range. Essentially what the Naive Bayesian Classifier is doing is counting the number of times a certain pixel has a certain value and belong to a specific class, thus if a pixel hasn't had a certain value in the training set the probability of it belonging to any class is zero. This of course isn't optimal. It is the following part in eq. 1 that gets the value of 0, $P\left(\text{Pixel}_i = X_i | \text{Number} = k\right)$ [3].

$$\begin{aligned} P(X = &\text{Number}) = \arg\max_k \left[P\left(\text{Number} = k, \text{Pixel}_0, \ldots, \text{Pixel}_{n-1}\right)\right] \\ &= \arg\max_k \left[P(\text{Number} = k) \prod_i P\left(\text{Pixel}_i = X_i | \text{Number} = k\right)\right] \end{aligned} \tag{1}$$

Another problem with the NBC is that it cannot tell if a pixel value is close to another pixel value. If a certain sample in the test dataset has the value 45 in the 25 pixel and that has never happened in the training dataset but it has had the value 46 in the training dataset and then it always belonged to class 8, then our model won't be able to tell that they are close and tell that our test sample probably belongs to class 8.

The decision tree in lab 2 had the exact same problem and was really ineffective for large resolution pixtures since that requires that the we have a large amount of training data, creating a very wide and deep tree.

Question 2:

The issue above was handled by switching to Normal distributions instead of the count based approach. Mitchell (see lecture slides lecture 10) suggests the m-estimate to solve the problem. What does the m-estimate do?

The m-estimator solves the problem of when the number of instances, n_c, with attributes a_i and class v_j is very small or even 0. Then, instead of the of the probability $P\left(a_i | v_j\right)$ being calculated in the "statistic approach" of

$$P\left(a_i | v_j\right) = \frac{n_c}{n} \tag{2}$$

where n is the total number of training instances of class v_j, it is instead calculated by [1]

$$P\left(a_i|v_j\right) = \frac{n_c + mp}{n + m} \tag{3}$$

p is the prior estimate of the probability, which is assumed to have uniform prior $p = \frac{1}{k}$ where k is the number of values the attribute a_i can take. m is the equivalent sample size, a weighting factor for the prior. We now see that $P\left(a_i|v_j\right)$ won't be zero even when n_c is zero.

Question 3:

In lab 5 you implemented a Nearest-Centroid-Classifier (NCC) and a Gaussian NBC. How are those related?

Both models use the euclidean distance from the sample and the different centroids for each class cluster. The NCC just picks the cluster with the smallest distance to the sample and classifies the sample to that class. The Gaussian NBC however also use the variance and normalizes the distance with it.

Question 4:

Further, what is the relationship between the NCC and the result of the k-Means?

The NCC and the K-Means both create centroids that is the used to label the data by measuring the euclidean distance. However, the difference is that the NCC uses supervised data (i.e. labeled data) and K-Means uses unsupervised. Therefore, the NCC calculates centroids with all the training data for each class, i.e. outliers will be included. The K-Means creates clusters of the data and then calculates the centroid for each cluster.

Question 5:

Explain in your own words the difference between k-Means clustering and the basic EM for GMMs as given in the Murphy-book (see above).

The k-Means algorithm is a special case of the EM-algorithm for GMM. In the k-Means algorithm we instead assumes that the covariance is fixed: $\mathbf{\Sigma}_k = \sigma^2 \mathbf{I}_D$ and that the class prior is fixed: $\pi_k = \frac{1}{K}$, therefore only the cluster center needs to be updated iterative [2, p.352].

References

[1] Frank Keller. Naive bayes classifiers. URL: `http://www2.cs.uh.edu/~arjun/courses/nlp/naive_bayes_keller.pdf`.

[2] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[3] Elin A. Topp Nugues. Bayesian classifiers (with a recap of probabilistic representation). URL: `http://fileadmin.cs.lth.se/cs/Education/EDAN95/Lectures/2019/BayesianClassifiers.pdf`.