DSC-540: Week 8 – Ensemble Methods

Seve Martinez

Grand Canyon University

Abstract

Ensemble methods can be a great way to improve the predictive ability of a data science model. There are several methods including bagging (bootstrap aggregating), bagging, and others. In this paper, a custom implementation is proposed using three fusion methods: Weighted Majority Voting, Behavior Knowledge Space, and Naive Bayes Combination. The data set PAMAP is used from the UCI machine learning repository. Using a collection of scikit-learn methods, an accuracy score of 93% was achieved on a sample of the PAMAP dataset (Chowdhary, et al., 2017) using Weighted Majority Voting, which was better than the study's 83% combined average.

Keywords: ensemble methods, ANN, BDT, SVM, weighted majority voting

Week 8 – Ensemble Methods

Ensemble methods seek to take a series of weak learners and combine their results to build a stronger and more robust model. The idea is that the weak model outputs can be aggregated in a variety of ways that could provide a better prediction. Some of the combination methods explored in this paper are the weighted majority voting, Naive Bayes Combination, and the Behavior Knowledge Space.

Methods

Data Set

The data set comes from the PAMAP Physical Activity Monitoring study from UCI Machine learning repository. The data contains 54 columns of various data points from three accelerometers, worn in different locations on the body. Nine subjects were measured while doing various activities of differing exertion levels. Some activities include walking, lying down, sitting, ironing, cycling, and others. A sampling rate of 100Hz was used to capture data on a 3D axis of ± 16g. While three locations were used, only the wrist location was used for the study (Chowdhary et al, 2017).

Additional features were then calculated from the three columns of wrist data which include mean, median, standard deviation, variance, skewness, and kurtosis. Chowdhary et al used a 10-second sliding window on the timestamp value, but this proved to be programatically challenging, so a random sampling of ten records was chosen for each row set of the additional features. This will produce somewhat inaccurate data because dissimilar time stamped values could be for different activities, which will skew the data. However, for the purposes of this paper, this was an acceptable compromise.

The data set used an 80/20 train/test split. The entire data set was about 2.4M rows, and when pruned down per the authors' requirements, about 1.9M records remained. In the interest of time, a subset of 200,000 rows was selected at random.

Models

Binary Decision Tree. A classification and regression tree (CART) is leveraged using the scikit-learn library. The tree decides at each node how to split between two values (Packt_pub, 2018). This is then performed in a recursive manner until the data is as pure as possible. Chowdhary et al used a max_split value of 20, so that is used here.

K-Nearest Neighbors. This classifier uses the Euclidean distance of nearby points to determine the class of the $\mathbf{x}$. This algorithm is very simple but has a surprisingly high success rate. It is best suited for continuous data (Wikipedia, 2020) which is what is used in this study. Per Chowdhary, a k of 7 was used.

Support Vector Machine. A support vector machine seeks to maximize the distance between data using a linear boundary. It does this by using 'support vectors,' or vectors that are nearest to the boundaries of the line. It has the ability to project data into a higher dimension in order to find a linear boundary between the data. Because this data set has more than two classes, a 'one-vs-rest' method will be used (Pedregosa et al., 2011).

Artificial Neural Network. This is a type of forward feed deep learner that is based on the perceptron design of the 1940s (Dormehl, 2019). The difference here is the use of backpropagation and gradient descent to minimize the cost function. ANN's have seen a lot of use in recent times thanks to the improvement of technology and vast amounts of data.

This ANN used the ReLu activation function, 250 epochs, and a hidden layer size of 50. This is consistent with the study.

Fusion Methods

Weighted Majority Voting. This method works by assigning greater power to more competent classifiers in the bunch, given that all classifiers are not the same level of performance (Kuncheva, 2004). In order to calculate the weights, the following equation is used:

$$score(k) = \sum_{l_t = C_k} w_i \qquad \text{(Equation 1)}$$

for all k = 1, 2,…,m. The label chosen is the one with the maximum score.

$$finallabel = arg \max_{k=1}^{m} score(k) \qquad \text{(Equation 2)}$$

Naive Bayes. Assumed independence, for each classifier D_i, an $m \times m$ confusion matrix CM_i is calculated using the training data set. The following formula calculates the posterior probabilities of each class label for all total number of objects T:

$$score(k) = \frac{T_k}{T} \prod_{i=1}^{N} cm^i(k, l_i) \qquad \text{(Equation 3)}$$

Behavior Knowledge Space. This is simply a fancy name for multinomial combination (Kuncheva, 2004). It works by estimating the posterior probabilities $P(\omega_k|\mathbf{s})$ for all k = 1,…,c and every combination of votes $s \in \Omega^L$. Then, given $\mathbf{x} \in \mathbb{R}^n$ Labels are assigned by D and the highest posterior probability label is taken.

Conclusion

Each model individually created a weak classifier with sub-20% accuracy. The accuracy of the individual models are as follows:

```
Ann:  {0.008025}
BDT:  {0.0034}
SVM:  {0.02585}
```

This was to be expected and is consistent with the idea of the weak classifier methodology. It expected that the ANN would have the slowest computational time, but the KNN was orders of magnutiude slower to train.

When combined and pushed through a weighted majority voting ensemble, the accuracy shot up to 93%. The confusion matrix is as follows:

```
array([[3857,   25,    0,    0,    0,    0,    0,    0,    0,    0,    0,
           4],
       [  15, 3814,   69,    0,    0,    0,    0,    0,    0,    0,    0,
           0],
       [   0,   85, 3712,    0,    0,    0,    0,    0,    0,    0,   88,
           0],
       [   0,    0,    0, 4995,    2,   49,   49,    2,    4,    0,    0,
           0],
       [   0,    0,    0,   10, 1765,   83,  111,    0,    0,    0,    0,
          36],
       [   0,    0,    0,   89,   42, 3147,   65,    0,    0,    0,    0,
           8],
```

```
[    0,     0,     0,    59,    88,    48, 3541,     0,     7,     0,     0,
     4],
[    0,     0,     0,   137,     0,     0,     4, 1921,   330,   107,     0,
     0],
[    0,     0,     0,    47,     0,     0,    25,   522, 1654,     0,     0,
     0],
[    0,     0,     0,     0,     0,     0,     0,    74,     0, 3364,    99,
     0],
[    0,     1,    84,     0,     0,     0,     0,     0,     0,    73, 4761,
     0],
[   35,     0,     0,     0,    83,    56,    13,     0,     0,     0,     0,
   737]])
```

And the ground truth label:

```
4.0     5101
17.0    4919
2.0     3898
1.0     3886
3.0     3885
7.0     3747
16.0    3537
6.0     3351
12.0    2499
13.0    2248
5.0     2005
24.0     924
```

Clearly the use of the models together provided a superior solution than any individual model, which was consistent with the prior study. A future method would be to work with the larger dataset and build a more robust model as the example set provided by the authors was only 2,234 records, when the UCI dataset is over 2M. Given the time constraint, it became too computationally difficult to calculate the extra features. A more efficient generator-type python method may provide better results. However, the reduced feature model actually provided a higher result than the triple ensemble method used by Chowdhary et al.

Further time and analysis would allow for the use of the Naive Bayes combination and the BKS fusion method, then combining all three for a final label output. This may yield an even higher accuracy rate, but 93% is quite satisfactory, especially considering the use of considerably fewer features.

References

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Kuncheva, L. (2004). *Combining Pattern Classifiers.* Hoboken, New Jersey. John Wiley & Sons, Inc.

Dormehl, L. (2019). What is an artificial neural network? Here's everything you need to know. Digital Trends. https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Chowdhury, A. (2017). Ensemble Methods for Classification of Physical Activities from Wrist Accelerometry. *American College of Sports Medicine*, 1965-1973.

Packt Pub. 2018). Binary Decision Trees. Medium. https://medium.com/@Packt_Pub/binary-decision-trees-1ec94cfed208

Full title of the article. (Date). Title of the Journal, volume number, page numbers.

Author's last name, first initial. (Year, Month Day). Title of article. *Name of Website.* Retrieved from URL