Marc Llort Maulion - marc.llort

Data mining - Data Mining P3

Learning to categorize images and classification digit newsgroup

Goal

The main objectives of this practice are:

- Spread knowledge of Python package scikit-learn
- Take fluency with the selection, standardization, preprocessing and search attributes
- Knowing the different implementations of the algorithm-based learning with examples IBL
- Working with the concept of optimizing a learner.

1. Use cross-validation to estimate the optimal number of neighbors K

a. How can a search process parameters?

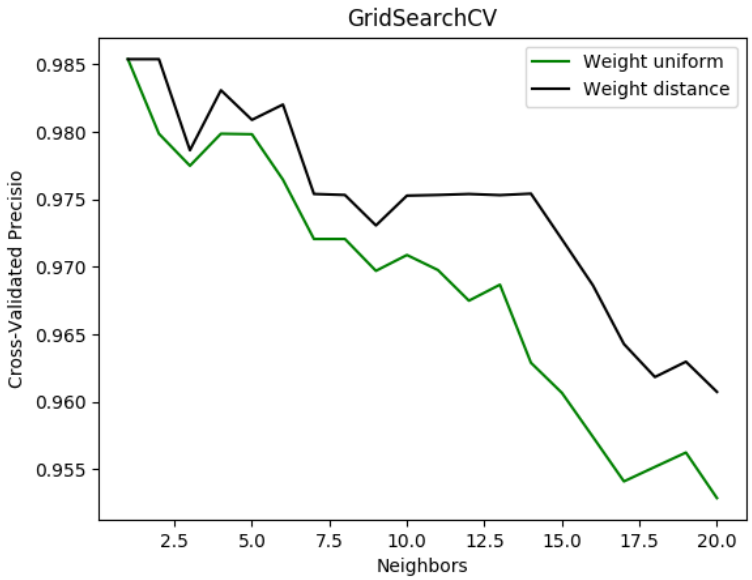
So far we simultaneously perform a search parameter did a couple of loops to find the number of residents within the range.

```
# Proves amb numero de veins variant
for i in range(min_neighbours, max_neighbours):
    success = 0
    total = 0

    for j in range(len(folds)):
        train, test = folds[j]

        x_train = [X[k] for k in train]
        x_test = [X[k] for k in test]
        y_train = [Y[k] for k in train]
        y_test = [Y[k] for k in test]

        # Prediccio en base al classificador
        predicted_y = get_predicted(x_train, y_train, x_test, i, weight)

        for k in range(len(y_test)):
            total = total + 1
            if y_test[k] == predicted_y[k]:
                success = success + 1
```

In this case, use GridSearchCV, which will help us to perform the same functionality itself. This is a function provided by sklearn.

```
results = grid_search (KNeighborsClassifier () , param_grid , x_train , y_train)
```

Once we have the results of the GridSearchCV classified KNN (nearest Neighbors is the best finisher in this case, since it is programmed specifically for KNN), we can begin to display data. In this first year, we have the following result:



2. Use another learner (recommended Ensemble Neural Networks and any method)

a. MPLClassifier (Neural Networks)

We can see how we achieve high accuracy very quickly, proving that a good use case for neural networks.

b. Adaboost (Ensemble)

In the latter case the results are much worse Ensemble.

3. Working with another dataset (Twenty news groups)

In this last year we have worked with twentynewsgrups DataSet.

```
twenty_train = fetch_20newsgroups(subset='train', shuffle=True, random_state=42)
twenty_test = fetch_20newsgroups(subset='test', shuffle=True, random_state=42)

text_clf = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer()),
    ('clf', MultinomialNB()),
])

text_clf.fit(twenty_train.data, twenty_train.target)

docs_test = twenty_test.data
predicted = text_clf.predict(docs_test)
mean = np.mean(predicted == twenty_test.target)
print "Mean: %.2f%%" % mean
```

And the result is accuracy:

```
Mean: 0.77%

Process finished with exit code 0
```

4. Conclusions (extended)

a. How does the Learner efficacy results

Failure to use the learner most appropriate in each situation will allow us to achieve better accuracy while classifying the results.
If making an incorrect or optimal training, once we make the classification, the results are incorrect.

b. As the effectiveness of the learner changes depending on the problem

In this practice we have classified different types of data (digits and news). In the case of digits results have been very good except you do kind of ensemble methods.

In the case of news, once we had the vectoritzades, the results have been less good than in the case of digits, probably because the learner is thought to be used with numbers.

As we have seen this used learner can face better or worse the problem is raised. According to the results we seek, we use fast learners with results less accurate or slower learners with very accurate results.

c. Ratings NN and Ensemble Method?

The NN have achieved superior results to Ensemble as its accuracy was above 95%. In the case of Ensemble us back in all cases lower than the accuracy of neural networks.

With datasets used find that NN has been better.