Koshkareva Julia
MS-DS-2

Advanced Machine Learning

Project I
Report

Classification with k-nearest-neighbors algorithm

Algorithm: k-nearest-neighbors

To classify an instance with the k-NN algorithm it is necessary to calculate the distance between feature vectors. After finding the most common class among the k-closest examples, the unknown image receives label. Every data point among k examples gives a vote and the category with the biggest number of votes wins.

Implementation. Description.

Function name	Description
get_labels (training_set)	returns labels for training example. Just take the last column of the dataset
find_neighbors (distances, k)	cuts first k elements from calculated distances
predict_label (neighbors, classes)	calculates votes of the data points, returns number of votes and label
knn(training_set, test_set, k)	calculates distances between instances from training set and test instances. Next, sorts these distances and cut with *find_neighbors().* Next, counts the votes and predict label with *predict_label()*

Preprocessing. Description.

Function name	Description
img_to_feature_vector (image, size)	resizes image and returns array of pixels using OpenCV library
img_color_histogram (image, bins)	extracts a color histogram from the HSV color space using OpenCV library

load_images (path_to_data)	takes a url of some folder and then collect paths of all images inside, returns array of image paths
img_to_vectors (imagePaths)	takes array of image paths and then convert each image to pixel vector, extracts histogram. Returns arrays of pixels and histograms for each image
train_test_split (dataset, labels, test_size = 0.25)	splits dataset to train and test parts. Parameter test_size gives the part of test size
get_accuracy (predicted_classes, testLabels)	Calculates accuracy (number of right answers/all answers)

Results for Datasets:

- Flowers (https://www.kaggle.com/alxmamaev/flowers-recognition)

This dataset consists of 4323 images of flowers. All flowers can be divided into 5 groups: daisy, dandelion, rose, sunflowers, tulip.

During the preprocessing the image is flatten to pixel vector. Color histogram was also extracted from picture. I try to analyse by both of these features, but experiments show that classifying by pixels gives worse results that classifying by color histogram.

To evaluate how my K-means implementation works I decided to compare it with SKlearn implementation. Accuracy of the both algorithms is demonstrated in table below.

	My K-means implementation	Sklearn implementation
Accuracy	45.164 %	46.077 %

The results are not so great (less than 50%), because the data is really noisy. For example, there are some pictures of flowers' fields or pictures of flower without bud (only leaves and stem).

- Fruits 1 (https://www.kaggle.com/moltean/fruits)

This dataset has 28247 images of 59 classes. My laptop can't handle so much information, that is why I cut the dataset to 5376 images and 6 classes: apple, banana, cherry, cocos, kiwi, lemon, lime, mandarin, mango, pear, pomegranate, strawberry. Processing for this dataset was the same as for flowers dataset.

Accuracy of the my K-means implementation and for SKlearn implementation is demonstrated in table below.

	My K-means implementation	Sklearn implementation
Accuracy	99.99 %	99.99 %

Results are very good, because the dataset just contain a lot of pictures of one fruit. For example, it has 492 images of 1 apple from different sides. So, it is not very hard for the algorithm to classify this dataset. To receive more clear results it was need to enlarge this dataset. So, I just found another dataset with fruits.

- Fruits 2 (http://www.vicos.si/Downloads/FIDS30)

In this dataset we have different images of various types of fruits. And as we can see in the table below results are worse, than in dataset Fruit 1.

	My K-means implementation	Sklearn implementation
Accuracy	59.596 %	56.566 %

- Audio Cats and Dogs (https://www.kaggle.com/mmoreaux/audio-cats-and-dogs)

This dataset consists of 277 wav-files. To process these files I read files as numpy array and as feature calculate mean, median, min, max and std of these arrays. For this task we have only 2 classes (dogs and cats). The results of classification is as follows. To receive better results may be another types of features are need to be used. For example, sound can be presented as a spectrum and then parameters of this spectrum can become features.

	My K-means implementation	Sklearn implementation
Accuracy	69.565 %	69.565 %

Totals.

For some datasets accuracy is not good. Reasons for these results are in disadvantages of k=NN algorithms:

- not robust to noisy data
- slow for real-time prediction if there are a large number of training examples

Accuracy can be improved with the help of hyperparameter tuning (grid or randomized search).