Naive Bayes and KNN for Image Classification

By Jordan Boulanger and Samuel Sandell

ABSTRACT

In this paper we examine the merits of simple image classification methods such as Naive Bayes and KNN. Image classification and detection is an area of increasingly growing interest in the field of Artificial Intelligence and Machine Learning, especially since image classification is a first step to computer vision. There is a lot of buzz around deep learning and methods such as neural networks these days with pretty much everyone using it. With so many convenient platforms available now, one doesn't even need to know how deep learning actually works to use it. All deep learning really is is a tool that relies on the same core principles of much simpler methods. The goal of this paper is to present simpler and easier to understand algorithms for image classification and show they are still able to reach a higher accuracy than one might expect.

INTRODUCTION

Image classification and object classification has been a prominent topic in computer vision research with a myriad of different approaches and methods applied to the problem. With the increasing popularity of deep learning techniques such as neural networks, it is not surprising to find that deep learning techniques are often applied to the problem of image classification. However, deep learning is simply an advanced tool that builds upon the same basic principles of much simpler techniques. Classic algorithms like Naive Bayes, K-Nearest-Neighbor and Naive Bayes Nearest Neighbor (NBNN) can be just as effective but are often overlooked because of the existence of more advanced algorithms. Image classification, like many other problems, is simply extracting a set of relevant features (such as color, texture, faces, eyes) and developing an accurate classifier based on these features. Naive Bayes and KNN are both capable algorithms for performing this task. The advantage of Naive Bayes and KNN is that they can perform the classification task at extremely high speeds when compared to deep learning-based algorithms [1]. Furthermore, Naive Bayes and KNN require much less initial data, because no training time is required, compared to learning-based methods which require a significant amount of time to train and large training sets [1]. With Naive Bayes and KNN, the initial data is only needed to test the accuracy of the classifier, rather than to train it, so much less data is required. These advantages are achievable while still maintaining an acceptable and comparable accuracy. It can be argued that current common practices in image classification like quantization of local image descriptors and usage of 'Image-to-Image' distance over 'Image-to-Class' distance is often a factor when expensive learning-based approaches outperform simplistic methods such as Naive Bayes and K-NN [3]. NBNN uses NN distances between local image descriptors rather than distances between entire images. Thus, NBNN is able to compute 'Image-to-Class' distances without the need for complicated descriptor quantization methods (Boiman, 2008). These common practices used in learning algorithms aren't necessarily more effective. It can be shown that NBNN is just as capable of approximating the 'theoretically optimal' classifier as top-performing deep learning-based methods, all while retaining the performance benefits mentioned above, even on complicated or challenging datasets [3].

RELATED WORK

Boiman and other researchers set out in a paper to defend KNN as an image classifier [3]. They describe an approach to image classification using K-nearest neighbor. Instead of classifying the test image with images from the training set, however, the method expounded on in this paper uses local features of an image for classification. Local features are features drawn from subsets of the image. The paper postulates that this method will improve accuracy and efficiency [3]. It goes on to tell about the simple KNN image classification algorithm with which to compare to the new algorithm. It executes a KNN search among members of the training set, looking to assign a "similarity score" for each class of images. This score denotes how similar the test image is to the class. Next, the local feature classification algorithm is discussed. Instead of assigning classes to full images, it assigns classes to each local feature [3]. The local features for the images in the training set are classified according to the class to which their image belongs. The test image then has its local features extracted, which are compared against the corresponding local features in the training set using KNN. Once all local features have been classified, the image can be classified based on the local feature classifications. The class is chosen by calculating confidence scores for each class based on the results of the local feature classification, and then choosing the class which maximizes the confidence score [3]. Experimental results found the the local feature based method performed better that the basic whole image KNN classification.

Taheri and other researchers discuss a proposed improvement to the naive Bayes classification method [4]. In naive Bayes classification, each feature is assumed to be independent of all the other features. However, this assumption does not always hold true in the real world. The paper attempts to find a way to do naive Bayes classification without this independence assumption. This new algorithm is described as semi-naive Bayes, which locates dependencies between different features with conditional probabilities [4]. After introducing the initial problem, the paper goes on to describe the naive Bayes approach, as well as accuracy implications for the algorithm when the independence assumption falls apart. It tells of extensions of the naive Bayes algorithm with which to remedy this problem. Among these are algorithms which use statistical methods to remove features when doing so improves the accuracy of the classification[4]. In the third section of the paper, the proposed algorithm is described. The algorithm involves using conditional probabilities to compute connections between features for each class. Dependencies are found for feature X by finding feature Y which maximizes the probability that X and Y occur given the class C [4]. These dependencies are considered along with the class when doing the semi-naive Bayes calculations. The paper also talks about discretization, the process of converting continuous values into discrete values. Finally, the paper describes the results of the algorithm as opposed to the original naive Bayes. The accuracy of predictions was found to be better when dependencies between features were taken into account.

[1], [2], [3] and [4] discuss simplistic Naive Bayes or KNN related approaches to classification. Three of these are specific to image classification, but all of the methods can be expanded to different types of problems. While the first two papers focus mostly on a defense or justification of these algorithms (in particular, Naive Bayes and Naive Bayes Nearest Neighbor) in the face of newer and more popular, but more expensive, deep learning methods. The latter two papers discuss improvements upon the Naive Bayes and K-NN. The third tries to deal with one of the problems of Naive Bayes, which is the assumption that features are independent of the each other. It proposes what they call Semi-Naive Bayes which takes into account that features may not be completely independent. Rather than discussing the merits of naive Bayes or kNN, the fourth and final paper focuses on an approach specifically for image classification. It focuses on classifying based on local features, as opposed to the more classic approach of classifying the whole image. This approach is utilized in part by the Naive Bayes Nearest-Neighbor

(NBNN) algorithm discussed in the second paper. These four papers together were sufficiently informative and give us a strong foundation on simple yet effective image classification as well as propose small improvements that will enable us to produce an effective implementation for classifying dog and cat images.

IMPLEMENTATION

The implementation portion of the project is a simple image classification program using the k nearest neighbor and naive bayes methods to classify dog and cat images. It is written in python and compares two variations of the searches: one which classifies based on the whole image and one which classifies based on a local feature extracted from the image. The local feature variation takes the center portion of the images and uses these partial images for the knn and naive bayes classifiers. We are hoping for higher accuracy for the local feature based classification based on the assumption that the subject of the image, the dog or cat, will most likely be at the center of the photo. The implementation makes use of a few libraries for the classification process. The opencv library is a computer vision library and is used for image processing and feature extraction. The scikit-learn library is a python machine learning library and is used for the knn and naive bayes searches, training, and testing the data. The extracted features used for classification in this implementation are color histograms of the images. Color histograms are representations of the colors that compose the image. This is just one of many possible features that can be extracted from images. Color histograms have their advantages and disadvantages. They are simple to understand and extract, but also provide less accuracy than other more sophisticated feature extraction methods. In this implementation, the method of classification is the focus rather than the feature extraction method. The results compare the accuracies of the knn and naive bayes classifiers, as well as the accuracies of the whole image approach to classification versus the local feature based approach. Much of the material we read for this project, including one paper in the related work section, mentioned the local feature approach to image classification. Using simple KNN and naive bayes, we can easily compare the two approaches.

RESULTS

KNN Results

The dataset used has a total of 25,000 images, 12,500 dog images and 12,500 cat images (courtesy of Kaggle). Ninety percent of the images were used for the training set and the remaining ten percent were used for test data. The algorithm was tested for various values of k ranging from 1 to 100. For the local feature based classification, two values for extracting the center of the image were used: the center 80% of the image and the center 90% of the image. The results for both the whole image and local feature approaches are given in figure 2. Accuracy for both methods hover consistently around 60%. This is not great, but has more to do with the downsides of the selected feature, the color histogram. Classifying the images based on color is not the best option when dogs and cats are often similarly colored. A more interesting result is that the center 90% local feature classification consistently outperformed the whole image classification after about $k = 10$. The reason for this could be that as the number of neighbors gets larger, the assumption that the dogs and cats are in the center of the image is more likely to hold. We are more likely to get multiple neighbors with the animals centered in the photo. The 80% local feature classification outperformed the whole image approach even quicker, as well as consistently outperforming the 90% center local feature approach. This can be explained by the 80% center approach eliminating more irrelevant portions of the images. The accuracy gains here are only a couple percentage points due to the simplicity of the feature extraction and local feature selection. However, these results show the potential accuracy gains that are waiting to be made by employing smartly selected local features for image classification.

Naive Bayes Results

Our implementation of the Naive-Bayes algorithm uses the same dataset of 25,000 dog and cat images provided by Kaggle, with half of the photos being of dogs and the other half of cats. Again, 90% of the set was used for training while 10% was used for testing. The particular Bayes equation that was utilized was Multinomial Bayes. Multinomial Bayes is particularly good when the features are discrete. Gaussian Naive Bayes actually does not work for classifying dog and cat images because it requires that the data be normally distributed. Bernoulli Bayes actually works as well for classifying dog and cat images because there are only two categories dog or cat. However, if we expanded our model to have three classifications dog, cat or neither then Bernoulli Bayes would no longer work. Still, we found that Multinomial Bayes was the more accurate algorithm. Please see Figures 1.1 and 1.2 for the equations behind Multinomial Bayes and Bernoulli Bayes. A limitation to the Bayes algorithm is that it assumes all the features are independent. With our main feature being the color of pixels, which is then stored in a histogram this might not be the best assumption. It is easy to understand that if a certain pixel is one color pixels near it will be similar to that color, especially if the pixels make up part of a dog or cat. Still each pixel is indeed independent from each other pixel so from that perspective the independence assumption holds. We found that Multinomial Bayes was slightly more accurate than KNN when classifying dog and cat images, however only by about 1.5%. The accuracy of both Multinomial Bayes and Bernoulli Bayes was increased by about another 1.5 percentage points when only considering the center 70% of each image and ignoring the outer 30% of each image. However, this may be biased towards the particular dataset in that the images are high-quality and most of the dogs and cats tend to be towards the center of each image. The final result when running the Multinomial Bayes algorithm on the dataset was 63.28% accuracy. The final result when running Bernoulli Bayes was 60.5%.

Overall, Bernoulli Bayes outperformed standard KNN, while localized KNN outperformed Bernoulli Bayes. Multinomial Bayes was the best performing of all four of the algorithms that we implemented and tested.

CONCLUSION

Image classification is an area of research that is gaining interest. It is an important part of computer vision, and thus a worthy topic of study. The image classification methods explored in this paper are k nearest neighbor and naive bayes. Numerous approaches and tweaks to these basic algorithms were researched and discussed. The algorithms were implemented to classify 25,000 images of dogs and cats. This implementation was used to evaluate the performance of the algorithms. Results for whole image classification and local feature based classification were both examined. Local selection helped improve performance by a small amount. Naive Bayes using the Multinomial Bayes variant was our best performing algorithm with an accuracy of 63.28%. While this accuracy may seem trivial compared to the extremely high accuracy of modern deep learning techniques, it shows that simple, easy to understand algorithms can be used for image classification with decent effectiveness. There are also extensions that could be added to our implementation such as hyperparameter tuning that could be added to algorithms to achieve an estimated accuracy of around 70%. We did not have the timeframe to implement these extended techniques.

FIGURES

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample of class y in the training set T, and $N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

Figure 1.1: Multinomial Bayes

$$P(x_i \mid y) = P(i \mid y) x_i + (1 - P(i \mid y))(1 - x_i)$$

Figure 1.2 Bernoulli Bayes

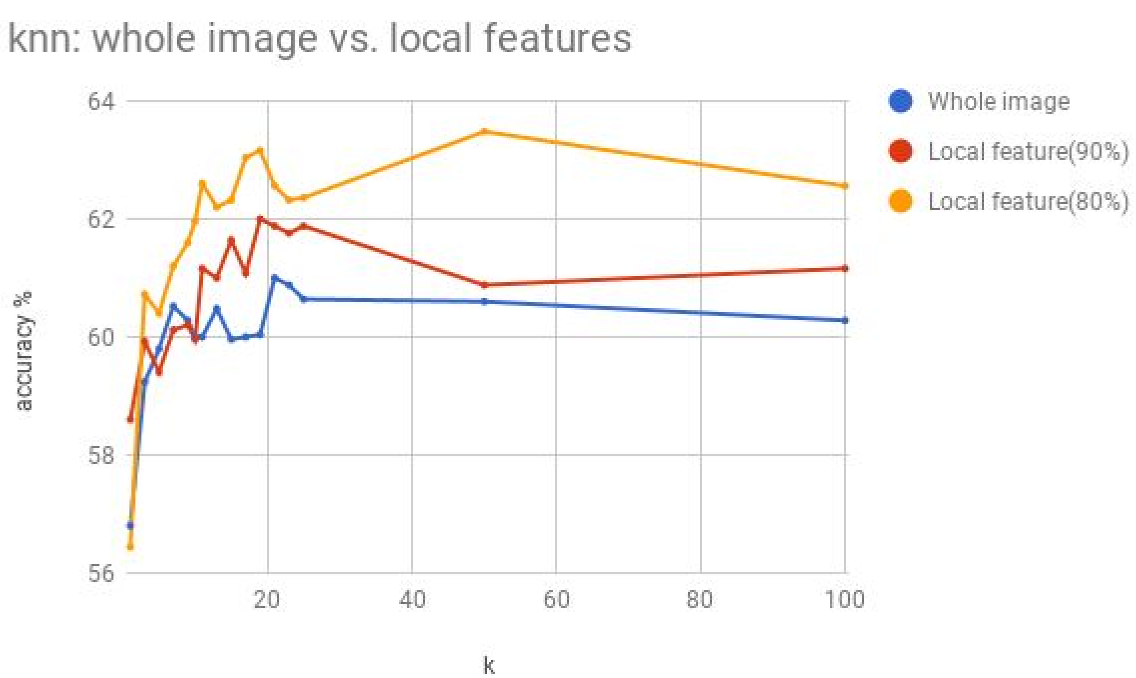

Figure 2: knn accuracies: whole image, local feature(center: 90%), local feature(center: 80%)

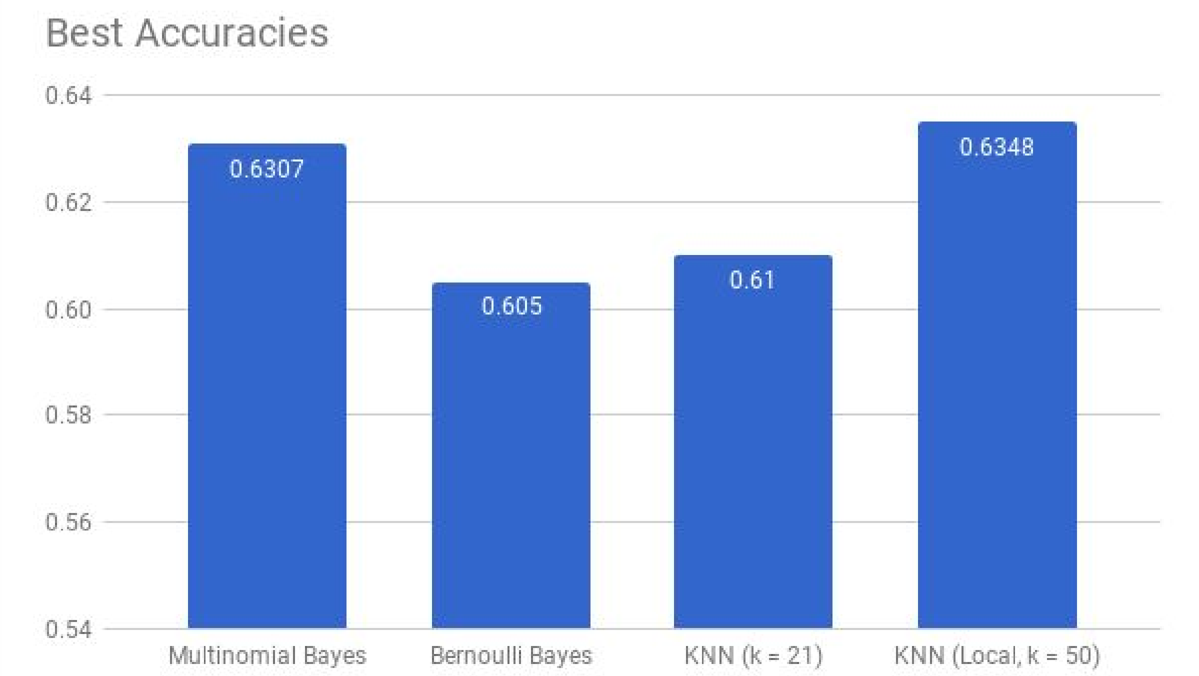

Figure 3 - final results for each algorithm

REFERENCES

[1] D. Park, “Image Classification Using Naïve Bayes Classifier”, *Park International Journal of Computer Science and Electronics Engineering (IJCSE*E), pp 135-139, Vol. 4, No. 3, 2016.

[2] G.Amato and F. Falchi, “kNN based image classification relying on local feature similarity”, *Proceedings of the Third International Conference on SImilarity Search and Applications*, pp 101-108, 2010.

[3] O. Boiman, E. Shechtman, M. Irani, “In defense of Nearest-Neighbor based image classification”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[4] S. Taheri, M. Mammadov, and A. Bagirov, “Improving naive Bayes classifier using conditional probabilities”, *Proceedings of the Ninth Australasian Data Mining Conference*, pp 63-68, 2011.