K-Means vs KNN: A Performance Comparison on Image Classification of Bird Species

Phineas Pham

March 2023

1 Abstract

Bird image classification is one of the most common tasks used to compare the performance of algorithms. We will use this task to compare the effectiveness of K-means and KNN algorithms. Specifically, we will compare the K-means and KNN by their ability to classify images. In this research, we setup the competition with 60 images of Bobolink and 60 images of Yellowthroat, a pair of two very similar bird species. To preprocess the dataset before performing K-means clustering and KNN algorithm, we will use Principal Component Analysis, a dimensionality reduction technique, to create a new dataset with two principal components.

2 Introduction

Classification is a classic task in machine learning field. Although K-means clustering is considered an unsupervised learning algorithm, we can still use K-means to group data points that are close to each other into k clusters, and label all points in the same cluster with the majority ones.

Every image can be presented as a matrix of pixels that form the image. Based on whether an image is in RGB or grayscale format, each pixel of an image can be represented by 3 values (red, green, blue) or 1 value of black or white density (gray scale). When dealing with image classification tasks, a common approach is first to represent each image as a data point, and each point has as many features as the number of pixels it has. Thus, in situation where we want to train a model with limited number of images, we will eventually have a dataset where the number of features bigger than the number of images, imposing a 'curse of dimensionality' problem.

In this paper, we examine the use of K-means and KNN in classifying images on a dataset that has its number of dimensions reduced to two by using PCA. Specifically, we transform the dataset into its first two principal components,

then we apply K-means and KNN algorithms on the transformed datasets to see how well k-means can classify those images with two principal components.

3 Preliminaries

a) K-Means algorithm:

In this section, we formally define K-means and Principal Component Analysis algorithms.

The K-means algorithm first initializes k cluster centers by randomly selecting k training points from the dataset. The algorithm then repeats two phases: the assignment phase and the adjustment phase.

In the assignment phase, the algorithm computes the distance between each training point and the k cluster centers, and then assigns each training point to the cluster whose center is closest. In the adjustment phase, the algorithm computes the mean of all the training points assigned to each cluster, and then moves the cluster centers to the mean of those points.

The algorithm repeats the assignment and adjustment phases until the cluster centers no longer move or a maximum number of iterations is reached.

At the end of the algorithm, each data point will be assigned to one of the k clusters, and the centers of each cluster are placed at the mean of the data points of that cluster.

Algorithm 1 K-Means Algorithm

```
1: Select value for K, the number of centers
2: Initialize centers by selecting K random training points
3: while centers move do
4:    #assignment phase
5:    for each training point s_i do
6:       compute the distance of s_i to each center
7:       assign s_i to the closest center
8:    #adjustment phase
9:    for each center C do
10:      compute the mean of all samples assigned to C
11:      move C to the mean of its assigned points
```

b) K-Nearest Neighbor (KNN):

K-Nearest Neighbor (KNN) is a simple yet powerful algorithm used in the field of machine learning for classification and regression tasks. It is a non-parametric and instance-based learning algorithm that makes predictions based

on the similarity of new instances to the training instances.

KNN is a supervised learning algorithm, which means it requires labeled data to train the model. The algorithm calculates the distance between the new instance and all the instances in the training set, and then selects the k nearest neighbors to the new instance. The value of k is chosen by the user and is typically an odd number to avoid ties. Once the k nearest neighbors are identified, the algorithm classifies the new instance based on the majority class of its k neighbors.

Algorithm 2 K-Nearest Neighbor

1: Choose an odd number of nearest neighbors, k
2: Calculate the distance between the new instance and all instances in the training set
3: Select the k instances with the smallest distance to the new instance
4: Classify the new instance based on the majority class of its k nearest neighbors

c) Principal Component Analysis:

Principal Component Analysis (PCA) is a statistical technique that is used to reduce the dimensionality of a dataset. PCA helps to simplify the dataset by identifying patterns and relationships between the variables, and then projecting the data onto a smaller number of dimensions or components.

The goal of PCA is to identify the most important patterns or relationships in the data and then to represent the data in terms of these patterns. This is achieved by finding the directions or axes of maximum variance in the data, and then projecting the data onto these axes. The resulting components are linear combinations of the original features, with each component capturing as much variance in the data as possible.

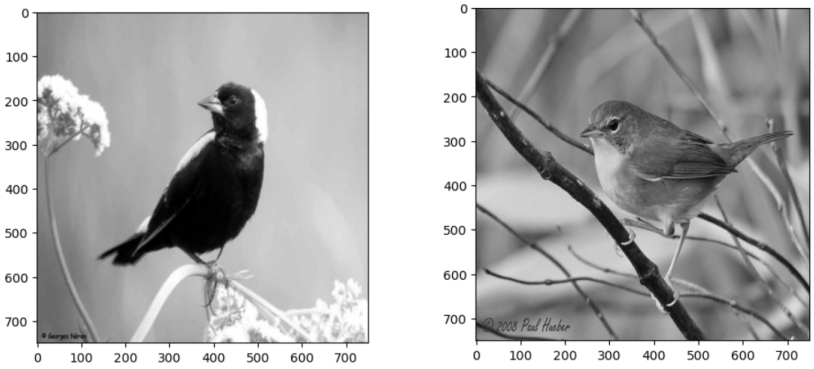
To perform PCA, we first standardize the data to have zero mean and unit variance. We then compute the covariance matrix of the standardized data, and find its eigenvectors and eigenvalues. The eigenvectors represent the principal directions of the data, and the corresponding eigenvalues represent the amount of variance explained by each component. We then can choose the top k eigenvectors with the largest eigenvalues, and use these to project the data onto a k-dimensional subspace.

4 Dataset and Preprocessing

In this research, we will train the model on images to label which image is of the Bobolink or Yellowthroat. These two birds are very similar to each other,

but possible to recognize different features distinguishing the two. Our dataset includes 60 images of each type of birds. We split our dataset into training and testing sets with a 75-25 split. Furthermore, we will use the same training and testing sets for both algorithms, so that we have a fair comparison between the two.

To make the images easier for the model, we turn those images into black and white, and resize them to 750 x 750 pixels. And before moving the dataset to PCA, we standardize the data by subtract the mean from each feature and divide each feature by its standard deviation. This normal standardization, also known as feature scaling, is a data preprocessing technique used to rescale the features or variables of a dataset so that they have zero mean and unit variance.



(a) Bobolink (b) Yellowthroat

Figure 1: Black-and-white images of Bobolink and Yellowthroat

5 Methodology

Here, we introduce the step-by-step approach of the research. Our inputs are bird images in black-and-white format and resized to 750 x 750 pixels. We label 0 for a Bobolink, and 1 for a Yellowthroat. Our dataset then have 120 rows and 562,500 features, where each row is an image, and each image has $750 * 750$ (= 562,500) features. Next, we preprocess with normal standardization StandardScaler() from Scikit-Learn library. We then are able to perform PCA on our training and testing datasets with Scikit-Learn library, transforming the datasets into a 120 rows and 2 features of most principal components. Plotting the dataset with 2 features in a 2D scatterplot we have this graph:

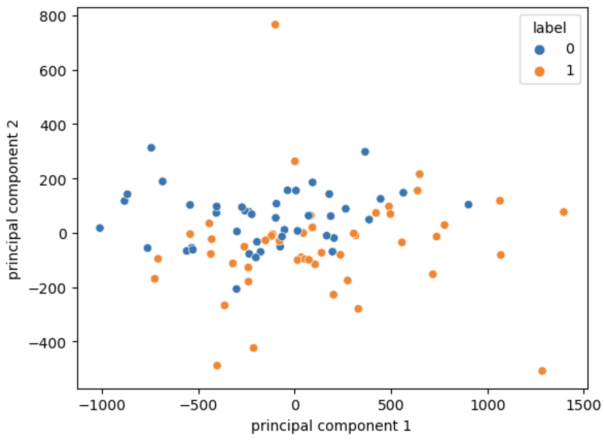


Figure 2: 2D plot of data points after PCA

From first glance, it seems that a big number of images of Bobolink and Yellowthroat lies across to each other in the middle. This would make classification algorithms perform worse especially for points in the center.

6 Metrics

Because we only care about how accurate K-means and other algorithm performs on this task, accuracy is the main metric that we will use to see how many times the algorithm is correct. However, precision, recall, and F-score metrics are useful in pointing out where the model performs well or not.

7 Results

We first perform Elbow Method for K-means algorithm on training dataset to look for the optimal k values.

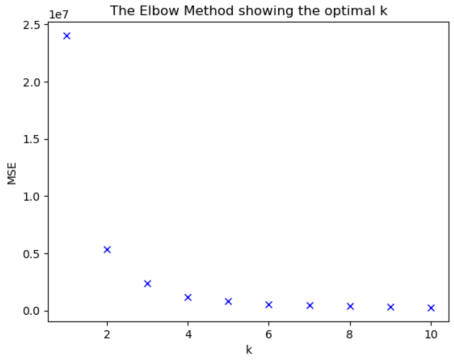


Figure 3: Elbow Method for K-Means

K = 4 has the smallest MSE, thus we will use k = 4 for K-means clustering. Using k = 4 means we will have 4 clusters.

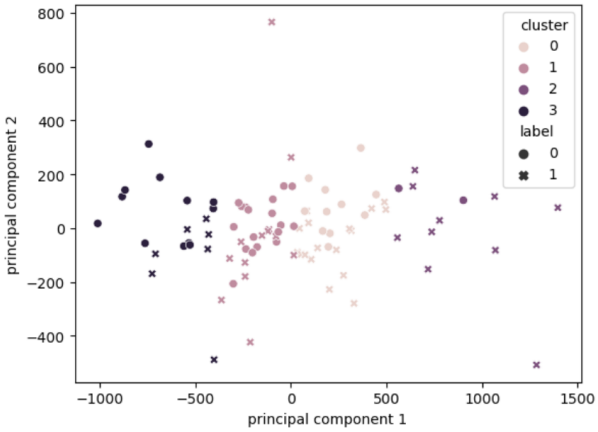


Figure 4: K-Means with k = 4 on training dataset

Based on figure 4, because cluster 1 and cluster 3 has more than half of their points are of class 0, while other clusters have more points of class 1, we label cluster 1 and 3 with label 0, otherwise we label them 1. Applying K-means algorithm to the testing set, we get the classification report below, giving details on how well our K-means algorithm work on this dataset.

Class	Precision	Recall	F-score	Support
0	0.56	0.53	0.55	17
1	0.43	0.46	0.44	13
accuracy			0.50	30
macro avg	0.50	0.50	0.49	30
avg	0.50	0.50	0.50	30

Table 1: K-Means Classification Report

With the overall accuracy of 0.50, and the precision and recall scores for both classes range from 0.40 to 0.60, K-means performs decently on this task.

Next, we train KNN on the training dataset. In KNN, one of the most important parameters is k value of the number of nearest neighbors. We perform sensitivity analysis with KNN on the training dataset and create figure 5. Having a small number of datapoints, we run the train KNN with values of k ranging from 1 to 30 and examine the changes in accuracy.

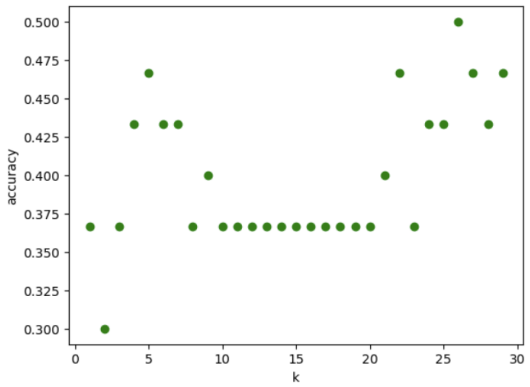


Figure 5: KNN Sensitivity Analysis on training dataset

From figure 5, we pick k = 26 where KNN performs best with highest accuracy. Applying the model on testing set, we get the classification report:

Class	Precision	Recall	F-score	Support
0	0.57	0.47	0.52	17
1	0.44	0.54	0.48	13
accuracy			0.50	30
macro avg	0.50	0.50	0.50	30
avg	0.51	0.50	0.50	30

Table 2: KNN Classification Report

Table 2 shows the performance metrics of KNN on bird classification. It shows an average accuracy score of 0.47, and the precision and recall scores ranging from 0.40 to 0.55.

Comparing the 2 tables of performance scores, we see that both algorithm perform same as good as the other, making true predictions half the time.

8 Conclusion and Future work

Although K-means perform a bit better, we would consider that both algorithms have about the same performance. The classification tasks between two

very similar bird species is a complicated problem, but with PCA and K-means or KNN, we are able to build a tool with 50% accuracy. The final result for the competition between K-means and KNN is draw.

9 Acknowledgement

I want to acknowledge the dataset used for this research is from a subset of dataset Birds-200-2011, created by Caltech Library. Caltech publicized the bird dataset on their site: `https://data.caltech.edu/records/65de6-vp158`

10 Reference

1. Venkat, N. (2018). The Curse of Dimensionality: Inside out. Researchgate. Retrieved February 22, 2023, from `https://www.researchgate.net/publication/327498046_The_Curse_of_Dimensionality_Inside_Out`

2. Curse of Dimensionality - Georgia Tech - Machine Learning, retrieved 2022-06-29

3. Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc., 2nd edition, 2019.