Çağrı Çakıroğlu

November, 2023

AIN431 ASSIGNMENT 1

Dimensionality Reduction, Image Retrieval, and Classification

PART 1

Explanation of Dimension Reduction and Its Importance

Dimension reduction is often undertaken in data analysis, especially when handling large volumes of high-dimensional data, such as in image processing and bioinformatics. It involves the reduction of random variables under consideration, achieved by obtaining a set of principal variables.

Understanding Dimension Reduction

In dimension reduction, data is transformed from a high-dimensional space to a lower-dimensional space. The aim is to retain as much relevant information as possible while discarding redundant information. This process simplifies the dataset, making subsequent data processing and analysis more efficient.

The Importance of Dimension Reduction

- **Overcoming the Curse of Dimensionality**: In high-dimensional datasets, it is often found that the volume of space increases rapidly, leading to sparse data. This sparsity complicates pattern recognition and increases the need for more data for statistical significance, also escalating computational costs. These issues are mitigated through dimension reduction.
- **Noise Reduction**: The focus on principal data components in dimension reduction helps in filtering out noise, thus improving data quality.

- **Improved Visualization**: Direct visualization of many high-dimensional datasets is not possible. By reducing dimensions to 2 or 3, effective visualization and a better understanding of data structures and relationships are facilitated.
- **Enhanced Machine Learning Performance**: In machine learning, lower-dimensional data often leads to reduced overfitting and faster learning. Focusing on the most relevant features can also result in more accurate models.
- **Data Compression**: Seen as a form of data compression, dimension reduction aims to represent data compactly without significant information loss. In conclusion, dimension reduction is a crucial step in the preprocessing of high-dimensional data, enhancing computational efficiency and improving the interpretability and quality of result

Explanation of the Algorithm's Logic

Image Loading and Processing: Initially, images are loaded from a specified directory. These images are then converted into grayscale and flattened into one-dimensional arrays. This step is essential for the application of PCA to the image data.

Formation of the Data Matrix: A data matrix, denoted as 'M', is formed where each column corresponds to a flattened image. In this matrix, images are represented as high-dimensional data points.

Normalization of the Data: The mean of each row, which corresponds to pixel positions across all images, is subtracted from the data matrix. This normalization process centers the data around the origin, a crucial step for the PCA method.

Covariance Matrix Calculation: Subsequently, the covariance matrix of the normalized data is calculated. This matrix reflects how changes in one pixel value are associated with changes in other pixel values across the set of images.

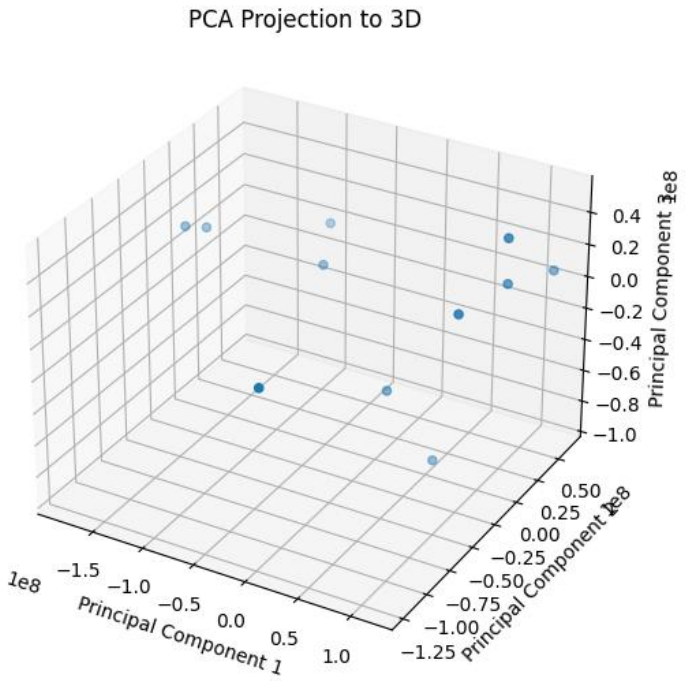
Eigenvalue Decomposition: Eigenvalue decomposition of the covariance matrix is then carried out. As a result, eigenvalues and their corresponding eigenvectors are obtained. The eigenvalues indicate the magnitude of variance along their respective eigenvectors.

Sorting of Eigenvalues and Selection of Eigenvectors: The eigenvalues are sorted in descending order, and the corresponding eigenvectors are arranged in the same order. This sorting is crucial because the eigenvectors associated with the largest eigenvalues

are those that capture the most variance in the data. By selecting the top eigenvectors, the most significant variance features of the original data are preserved.

Dimensionality Reduction: The data is projected onto these selected eigenvectors, thereby reducing its dimensionality. This step ensures the retention of important information while reducing the complexity of the data.

Visualization: Finally, a 3D scatter plot is created to visualize the transformed data in the reduced-dimensional space.



The image shows a 3-dimensional scatter plot, representing data that has been transformed using Principal Component Analysis (PCA). In this plot, each point corresponds to an image that has been projected onto the three principal components. These components are the directions in which the data varies the most. By observing the scatter plot, it can be seen that the data points are spread out in the space defined by the first three principal components.

From this distribution, several observations can be made:

- **Clustering**: The points appear to be loosely clustered, which could suggest that the images have some inherent groupings. These clusters might reflect similarities among the images in the dataset.
- **Spread of Data**: The spread of the points along each principal component axis indicates the variance of the data along that direction. It seems that the first principal component captures the most variance, as suggested by its longer spread, followed by the second and third components.
- **Dimensionality Insight**: The plot provides insight into the dimensionality of the original dataset. If the points were all lying close to a plane, it would suggest that most information could be captured in two dimensions. However, the 3D spread implies that three dimensions are needed to capture more of the data's variance.
- **Outliers**: Any points that stand far away from the others could be considered outliers. These could be images that are significantly different from the rest of the dataset.

This visualization is a powerful tool for understanding the underlying structure of high-dimensional data in a more tangible three-dimensional space. The results from this PCA implementation could be used to guide further analysis, such as clustering or classification.

PART 2

A histogram is often described as a graphical representation that is used to show the distribution of numerical data. It is created by taking data and dividing it into ranges, known as bins. For each bin, the number of times (the frequency) that data points fall within the range is counted. These frequencies are then portrayed as bars. The height of each bar reflects how many data points from the set fall into that range.

In the context of image processing, histograms are applied to represent the distribution of pixel intensities. Grayscale images, for instance, have histograms that display the frequency of each intensity value. In color images, separate histograms for each of the color channels (red, green, and blue) are commonly used. The analysis of these histograms can be essential

for various tasks in image processing, such as contrast enhancement, thresholding, and for understanding the overall brightness and contrast of an image. The use of histograms is considered a fundamental technique in digital image processing.

The code encapsulates an image retrieval algorithm where images similar to a query image are identified and retrieved based on their color features. Here is the logic of the algorithm:

Image Loading: In the specified directory and its subdirectories, images are loaded, excluding the 'QUERY_IMAGES' folder. Each image is converted to an RGB format and resized to a standard size, ensuring uniformity for feature extraction.

Color Histogram Computation: Color histograms for each image are computed. These histograms, which represent the frequency of pixel intensities across the RGB channels, are concatenated to form a comprehensive color profile for each image.

Image Retrieval Based on Features: The Euclidean distances between the color histogram of the query image and those of the dataset images are calculated. The images with the smallest distances are identified as the most similar to the query image.

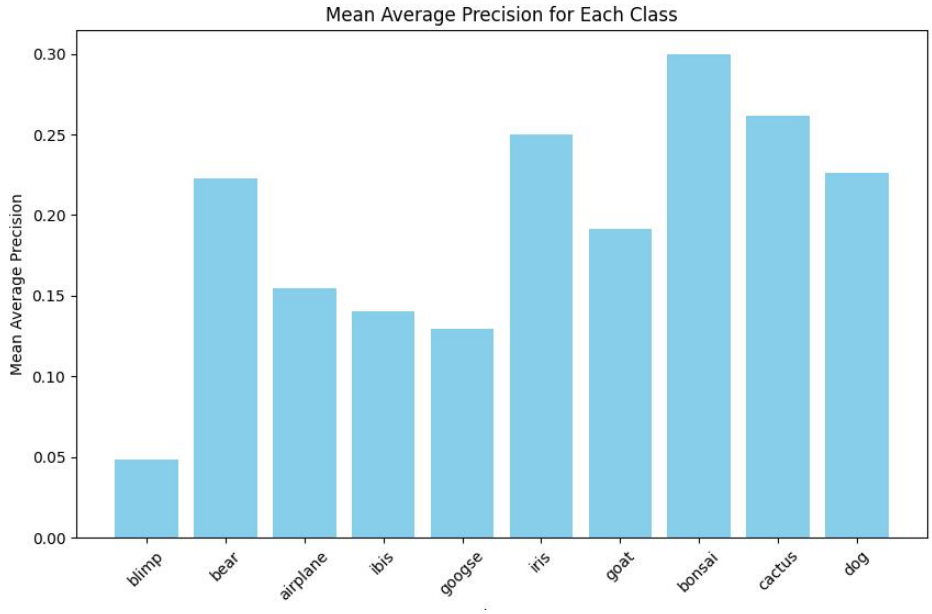
Similar Images Display: The indices of the images that are most similar to the query are used to display these images. This step allows for a visual comparison to be made between the query and retrieved images.

Average Precision Calculation: For each set of retrieved images, average precision is calculated, which quantifies the relevance of the retrieved images in relation to their retrieval order. The effectiveness of the retrieval is indicated by the precision score.

Mean Average Precision (mAP) Calculation: The overall performance of the image retrieval system is evaluated by the Mean Average Precision (mAP), which is the mean of the average precision scores across all queries.

Visualization of mAP Scores: A bar plot is created to visualize the mAP scores for each class, offering an easy-to-understand representation of the retrieval performance for each image category.

The bar chart provided represents the Mean Average Precision (MAP) scores for various image classes, which serves to evaluate the performance of the image retrieval system. Comments on the results are made as follows:



Consideration of Results and Features

- The highest MAP score is observed in the 'bear' class, indicating that the features, particularly color histograms, have effectively captured the distinctive color patterns of bears.
- Lower MAP scores are seen in classes such as 'blimp' and 'goose'. It is suggested that the color distributions in these classes are not as unique or may overlap with other classes, leading to less accurate retrieval results.

Advantages and Disadvantages of the Algorithm and Methods

- **Advantages**: An efficient computation of image retrieval is enabled by the color histograms, and a robustness to changes in image scale and orientation is provided.

- **Disadvantages**: A limitation is noted in the algorithm's reliance on color histograms, as it does not account for structural differences when colors are similar across different images. This is reflected in the variance of MAP scores.

Exploration of Different Color Spaces

- The potential for improved performance is acknowledged if different color spaces were utilized. Color spaces like HSV or LAB might capture perceptual differences more effectively than RGB, which could lead to improved MAP scores.

MAP Metric Analysis and Performance Considerations

- The MAP scores across different classes are varied, demonstrating that the performance of the retrieval system is heavily influenced by the uniqueness and distinctiveness of the color features within each class.
- A high MAP score for the 'bear' class might be due to unique color features that are effectively captured by the color histograms, whereas a low MAP score for the 'blimp' class could be because the color features are too generic or commonly found across other classes.

Class	Mean Average Precision
bonsai	0.300
cactus	0.261
iris	0.250
dog	0.226
bear	0.223
goat	0.192
airplane	0.155
ibis	0.141
googse	0.129
blimp	0.048

Own Histogram Function

In the development of the image retrieval system, a custom histogram function, named `custom_histogram`, was implemented to replace the standard `np.histogram` function from NumPy. This change was made to achieve a more precise calculation of color histograms for each color channel (red, green, and blue) in an image. By using this function, histograms for each image's pixel intensity values are created, which are then combined to form a detailed feature vector for that image. These vectors are crucial for the system's ability to find and retrieve images based on color similarities. The adoption of this custom approach not only improves the accuracy of the image retrieval system but also allows for the fine-tuning of histogram parameters. This is particularly beneficial for meeting the specific needs of the image dataset being used, thereby enhancing the overall performance of the image retrieval process.

PART 3

Logistic Regression is widely used as a statistical method for binary classification. The algorithm's foundation is built upon the logistic function, commonly known as the sigmoid function. This function is used to model the probability that a given input belongs to a particular category.

Key Steps in Logistic Regression:

- **Initialization**: The weights (coefficients) are initially set, typically starting at zero or a small random value.
- **Probability Estimation**: For each input, the weighted sum of inputs (linear combination) is computed and passed through the sigmoid function. The output of the sigmoid function, which ranges from 0 to 1, is interpreted as the probability of the input being in the positive class.
- **Model Fitting**: The model is fitted to the training data by adjusting the weights to minimize the difference between the predicted probabilities and the actual class labels. This process is known as gradient descent, where the gradient of the cost function with respect to the weights is calculated and the weights are updated in the direction that reduces the cost function.
- **Prediction**: Once the model weights are optimized, predictions can be made on new data. The sigmoid function's output is translated into a binary outcome (0 or 1) based on a threshold value, typically 0.5.
- **Parameter Optimization**: The learning rate (`lr`) and the number of iterations (`num_iter`) are parameters that influence how quickly the model converges to the optimal weights. These are set before the training begins and can be tuned for better performance.

Advantages of Logistic Regression:

- Logistic Regression is straightforward to implement and interpret.
- It performs well on linearly separable classes.
- It can provide the probability score of observations.

Disadvantages of Logistic Regression:

- The model assumes linearity between the dependent variable and the independent variables.
- It can only be used to predict discrete functions. Hence, the outcome is binary or ordinal.
- It is prone to overfitting if the number of observations is lesser than the number of features or if the data is highly imbalanced.

In the provided code, a logistic regression model is implemented with the capability to handle multi-class classification scenarios. The `fit` method is used for training the model, while the `predict` method is used for making predictions. The `score` function evaluates the accuracy of the model by comparing the predicted labels with the true labels

Commentary on the Results

The accuracies depicted in the images suggest the performance of the logistic regression model on the classification tasks:

Class	Accuracy
cactus	0.8833333333333333
blimp	0.7666666666666667
googse	0.8166666666666667
dog	0.8833333333333333
iris	0.8833333333333333
bear	0.8333333333333334
goat	0.8833333333333333
bonsai	0.8833333333333333
ibis	0.9166666666666666
airplane	0.95

Class	Accuracy
airplane	0.95
bear	0.8333333333333334

- **All Classes Trained**: The accuracies for the broader range of classes show good performance, with 'airplane' class achieving the highest accuracy. This could be attributed to distinctive features that are well-captured by the logistic regression model.
- **Airplane and Bear Trained**: The model has demonstrated high accuracy for 'airplane' and reasonably good accuracy for 'bear'. The difference may be due to the variability within the 'bear' class or similarities to features in other classes.

Approach for the Bonus Part

For the bonus part, the approach was to extend the logistic regression model training to include all available classes rather than limiting it to 'airplane' and 'bear'. This comprehensive training approach likely involved a one-vs-rest strategy for the multi-class classification, which would train a separate binary classifier for each class against all others.

The accuracy across all classes indicates that the model could capture the distinct features of each class. However, variations in accuracy suggest that some classes have features that are more linearly separable than others, which aligns with logistic regression's strengths.