Dimensionality Reduction Analysis

EEL4930 Special Topics in CISE: Applied Machine Learning – Project 2

Shane Ferrell  
Department of Computer and Information Sciences and Engineering  
University of Florida Wertheim College of EngineeringGainesville, Florida, USA  
shaneferrell@ufl.edu

*Abstract*—This paper discusses approaches to the design process of various machine learning techniques with a characters dataset. Various dimensionality reduction, manifold learning techniques, and visualization practices for classification tasks are explored. (*Abstract*)

Keywords—machine learning, scikit-learn, dimensionality reduction, manifold learning, classification (key words)

# Tools

Machine learning has been a surging field in the past decade. Machine learning practice and knowledge has become much more complex and extensive due to the increasing power of machine learning technology. In this paper, we explore the use of programming language python paired with the use of the free software libraries Scikit-Learn, NumPy, and Matplotlib.

# Dataset

## Dataset Information

The data used to explore these techniques and approaches is the characters dataset. This dataset includes 6720 photos of 10 various characters, each of which is 300x300 pixels in grayscale.

| **Character** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **$** | **#** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

1. A picture containing chart

   Description automatically generatedClass labels
2. Class frequency in the characters dataset

The dataset was collected at the University of Florida. Samples may be in various orientations and of different handwriting and capitalization. All labels are assumed to be correct and accurate.

## Data Preprocessing

To prevent overfitting of our models, we split the data into test and training sets for the supervised learning aspects of our classification task. Though the data is labeled and sufficient in size, it cannot efficiently be immediately used by machine learning algorithms and tools.

The dataset is grayscale images of a character written with black pen on a white background. The pixel values are a measure of intensity, meaning most pixels on the outside edges of the samples take very high intensity values, while the pixels representing the actual characters take very low values. The character should be the values we are interested in measuring, so the pixel intensities must be inverted.

Before any estimators can be used on the data, it must first be scaled to generalize points as well. Scikit-learn has many features that can be used to easily scale data. The sklearn.preprocessing module contains scalers for all types of data and distributions. The StandardScaler function in this package can be used on the numerical attributes of this dataset because standard scaler works well with data that will be used in principal component analysis, a dimensionality reduction technique that will be used on the data.

# Dimensionality Reduction

In terms of machine learning, the characters dataset has a relatively high dimensionality. Each sample contains 90,000 dimensions. This data will be difficult to work with in such a high-dimensional space. The Curse of Dimensionality refers to how high-dimensional feature spaces can inhibit a machine learning algorithm’s performance. High-dimensional data can lead to high computational time during learning, can lead to lack of generalization and often overfitting, and can hamper learning algorithms based on distances as points are pushed to the outside of the feature space.

## Downsampling the Dataset

The first dimensionality reduction step is to downsample the dataset. Some dimensionality reduction techniques can be computationally expensive if dimensionality is too high. The dataset in this analysis was downsampled from images of size 300x300 to 30x30 pixels, each downsampled photo having pixels of intensity value of the average of 10x10 pixels in the original photo. The downsampled images are used for all analysis.

## Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a dimensionality reduction technique that selects features by recursively removing less useful features until the desired number of features remain. RFE uses the data and its labels because it needs a supervised learning estimator to determine features’ usefulness. The sklearn.feature selection module contains dimensionality reduction techniques, including a simple RFE class.

The scaled data was trained on two RFE objects: one with a logistic regression classifier with Lasso regularization (L1 penalty and Library for Large Linear Classification solver) and one with a decision tree classifier with Gini impurity. Both RFE objects select 100 of the 900 features, removing 25 features at a time, fitting the object with the data 32 times. Figure 3 shows the pixels selected by the RFE algorithms.

A picture containing text, crossword puzzle

Description automatically generatedChart, scatter chart

Description automatically generated

1. Black represents the 100 pixels of the 30x30 images selected by RFE

Based on samples from the dataset, it appears that RFE with Decision Tree estimator chooses more useful features as the information of the character is roughly in the center of the images.

## Principal Component Analysis

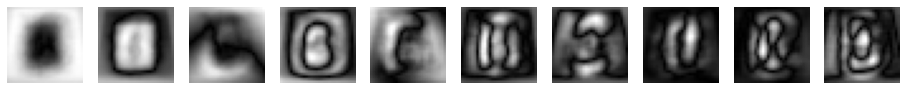
Graphical user interface

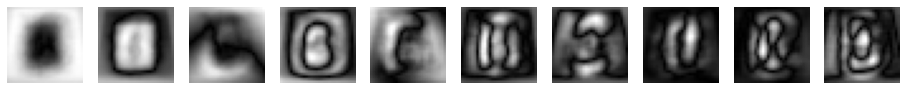
Description automatically generated with medium confidencePrincipal Component Analysis (PCA) is a dimensionality reduction technique that uncorrelates features to maximize variance. PCA transforms unlabeled data into principal components. The sklearn.decomposition module contains matrix decomposition dimensionality reduction techniques, including a simple PCA class.

1. Explained variance vs. number of principal components by PCA

The scaled data was trained on a PCA object. It was discovered that the first 49 principal components are needed to obtain 90% cumulative explained variance in the data. Figure 4 shows the cumulative explained variance for increasing amounts of principal components.

The eigenvectors of the first 10 components are visualized in Figure 5.

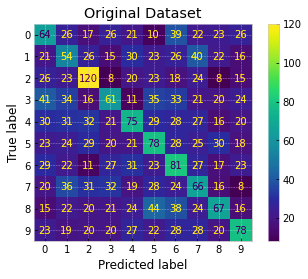




1. Visualization of the first 10 eigenvectors from PCA

The eigenvectors, while very abstract, can provide some information alone on the dataset. The first two eigenvectors appear to represent the area where characters most appear in the sample. There is not much information on the outside of the photos, as each photo is mostly centered on the character. Each consequent eigenvector provides more information on the shape of the characters and areas of higher importance.

Dimensionality reduction for classification tasks can be evaluated by performance. Figures 6-8 show the confusion matrices and performance statistics of the scaled data and the data represented by its first 49 PCA components of the test dataset with a Decision Tree classifier with Gini impurity.

 A screenshot of a game

Description automatically generated with medium confidence

1. Confusion matrices for classification of the test set

**Performance Scores for Original Dataset**

class precision recall support

0 0.22 0.23 274

1 0.19 0.20 273

2 0.37 0.42 285

3 0.24 0.21 296

4 0.27 0.24 309

5 0.25 0.26 296

6 0.24 0.28 291

7 0.22 0.24 280

8 0.28 0.23 291

9 0.32 0.27 285

accuracy 0.26 2880

1. Performace measures for classification of the original test set

**Performance Scores for Reduced Dataset**

class precision recall support

0 0.29 0.30 274

1 0.29 0.29 273

2 0.32 0.32 285

3 0.35 0.37 296

4 0.31 0.27 309

5 0.23 0.21 296

6 0.31 0.31 291

7 0.34 0.36 280

8 0.35 0.37 291

9 0.33 0.32 285

accuracy 0.31 2880

1. Performace measures for classification of the reduced test set

Classification performance is overall better with the reduced dataset. The reduced dataset has an accuracy of 31% compared to the original dataset’s accuracy of 26%. This is likely a result of the prevention of overfitting from dimensionality reduction.

original dataset duration 7.5 sec

reduced dataset duration 0.4 sec

1. Computation time for training classifier

Not only does dimensionality reduction show better classification performance, but it also reduces computation time. The training time for the Decision Tree classifier takes over 18 times longer for the original dataset than the reduced dataset.

Dimensionality reduction strategies are an effective and efficient solution to combat overfitting and reduce training time.

## Manifold Learning Algorithms

Manifold learning algorithms are a dimensionality reduction technique that takes a high-dimensional feature space with data that takes the shape of an unknown function that lies on a low-dimension manifold and transforms it into a low-dimensional feature space. The sklearn.manifold module contains classes for manifold learning algorithms, which can be used as a form of dimensionality reduction.

Three manifold learning algorithms that will be examined are T-distributed Stochastic Neighbor Embedding (t-SNE), ISOMAP Embedding, and Locally Linear Embedding (LLE). The t-SNE algorithm models the probability distribution of neighbors around each point in the high-dimensional space. ISOMAP is a form of multidimensional scaling used to translate information from geodesic distances between pairs of points. The LLE algorithm preserves information from neighborhoods of data compared globally to find the ideal embedding. All three of these algorithms are unsupervised.

The most useful manifold learning algorithm for dimensionality reduction can be identified using cross-validation. The model with the highest accuracy is the model that should be chosen.

Figures 10-11 show the set of hyperparameters tested using grid search cross-validation and its results.

manifold learning algorithm [TSNE(), Isomap(), LLE()]

number of components [2, 4, 8]

1. Tested hyperparameter values

manifold learning algorithm Isomap

number of components 8

best score 0.151785714285

1. Grid search cross-validation best parameters and best score

Figures 12-13 show the confusion matrix and performance statistics of the best estimator and hyperparameters found by grid search cross-validation.

Graphical user interface

Description automatically generated

1. Confusion matrix for classification of the test set with best manifold learning algorithm

**Performance Scores for Reduced Dataset by ISOMAP**

class precision recall support

0 0.15 0.15 274

1 0.14 0.14 273

2 0.17 0.19 285

3 0.21 0.18 296

4 0.15 0.14 309

5 0.12 0.12 296

6 0.16 0.17 291

7 0.15 0.16 280

8 0.16 0.15 291

9 0.19 0.19 285

accuracy 0.16 2880

1. Performance measures for classification of the test set with best manifold learning algorithm and number of components

Classification performance is poor for all tested values and manifold learning algorithms, but ISOMAP with 8 dimensions performs slightly better than others with a 16% accuracy.

The dataset following use of the manifold learning algorithm can be visualized in this lower-dimensional space. Figure 14 shows the visualization of the first two dimensions of the ISOMAP embedding.

A picture containing chart

Description automatically generated

1. Visualization of training data of class 6 with first two dimensions of ISOMAP embedding with 8 components

The dimension along the horizontal axis appears to roughly correspond to character size and thickness. Lower horizontal component values appear to correspond to smaller, thinner characters, while higher values appear to correspond to larger, thicker characters. The dimension along the vertical axis appears to roughly correspond to the darkness and photo contrast. Lower vertical component values appear to correspond to lighter, low contrast samples, while higher values appear to correspond to darker, high contrast samples.

# Conclusions

Dimensionality reduction techniques can be effectively used to avoid many disadvantages of using data with high-dimensional feature space. Various strategies and algorithms have been shown as a solution to combat overfitting and reduce training time. This leads to increased classification performance. Manifold learning algorithms are a possible unsupervised solution for data that can be transformed to project information that lies on a high-dimensional manifold.