Image Classification

Approach

As training dataset consists of 21186 records with 887 dimensions. Hence, primary approach was to reduce dimensions. In order to do so, I used three dimensionality reduction techniques.

I predicted classes of the images considering reduced dimensions and using following three algorithms:

1. PCA: Principal Component Analysis
   This technique uses an orthogonal transformation to convert a set of data points with high dimensions into a set of linearly uncorrelated principal components.

2. SVD: Singular Value Decomposition
   matrix factorization technique that factors a matrix M into the three matrices U, Σ, and V.

3. Random Projections
   Technique in which the points projected into lower dimensional space from higher dimension, approximately preserves the distances between them.

Then, I combined the result of the above algorithms to classify images more precisely. I used KNeighbors classifier for the same.

Methodology

Step 1: Imported varies libraries and modules such as pandas, numpy, metrics, cross validation, decomposition(TruncatedSVD, PCA), random projections, KNeighborsClassifier.

Step 2: Then, I imported the dataset using numpy and store it as a data frame.

Step 3: I declared the classification model. Before fitting the model with data, I reduced its dimensionality using 3 algorithms.

Step 4: First, I used SVD to reduce the dimensions.
After examining performance of a model against various number of

components, I selected 30 (Number of Components), which yields f1 score, precision, recall almost same on the train data.

Algorithm:

a. Initialize the model
b. Fit the model
c. Predict the values

Step 5: In order to determine the various metrics like accuracy, F1 score, precision, recall, etc. I used cross validation, split the data and predict values and compare with known outcomes.

Step 6: Like SVD, I used PCA with 150 components and Random Projections with 80 components.

Step 7: I combined the result of the predictions done by using PCA, Random Projections, SVD by comparing their values in each iteration.

Step 8: The result obtained in step 7 determine the final predicted class of an image.