Kannada Letters Classification using KNN and SVM with PCA for Dimensionality Reduction

Chandra Prakash Jagrat, *1MS17EE017,* Kartik Kuwal, *1MS17EE028*, Shivam Sanju, *1MS17EE054*, Sourabh Saha, *1MS17EE057,*

Abstract— The text classification is designed to extract useful information from a large number of data. The documents can contain unnecessary information that affects the classifier's accuracy. The step of data pre-processing aims at cleaning the texts with unnecessary information removed. The main purpose of this paper is to explain and assess the accuracy of both training model and classifier in complete word pre-processing. The first is the evaluation of data sets and the second is the stop word estimation technique. This is achieved by two methods. In the experiment, the SVM an K-nearest (NKN) were used to establish the instruction.

keywords - Classification, machine learning, K-nearest neighbour, support vector machines

I. INTRODUCTION

Text classification is utilized to give useful information from the large amount of data. It is one of the important research issues in the field of data. Based on the content of the text, categorization is defined as the process of grouping this text into one or more predefined categories based on linguistic features. In general, the terms 'Text categorization' and 'Text Classification' refers to the same meaning. Text categorization is sometimes taken to mean sorting documents by content,
while text classification is used to include any kind of assignment of documents to the specific classes like sorting by author, by publisher, or even by language.
Text classification is the many important research problems in information retrieval (IR), data mining and natural language processing. It is the primary requirement of text retrieval systems, which retrieve texts in response to a user query.
There are many applications of text classifications, such as e-mail filtering, news monitoring, spam filtering, sorting archives, automated indexing of scientific articles in e-libraries, classification of news stories and searching for interesting information on web.
The rest of this paper is organized as follows: give a brief review of researches in the area of Kannada text classification which used the stop words estimation as technique of preparing the data without stemming. Explains the more important point in the Kannada language structure. details, explains the proposed techniques to enhance the accuracy of the classifier. Highlight the structure of the text classification algorithm that used to test the data given in section 5. Finally, section 6 explains the conclusion and future work.

II. OBJECTIVE OF THE PAPER

This project is to classify handwritten Character. The goal is to take an image of a handwritten Character, and determine what that Character is. The Character range 48 character. In this study, we will look into the Support Vector Machines (SVMs) and K-Nearest Neighbor (KNN) techniques to solve the problem. The tasks involved are the following:
1. Prepare Handwritten dataset
2. Preprocess the dataset
a. Apply feature transformation for dimensionality reduction
b. Split the dataset into training and test sets
3. Train a classifier that can categorize the handwritten digits
4. Apply the model on the test set and report its accuracy
In that study, we describe the image dataset, we briefly explain the preprocessing steps of the input data, and describe our implementation of SVM-KNN. We present the results of our algorithm.

III. RELATED WORK

K-nearest neighbor (KNN) algorithm and support vector machines (SVM) algorithm evaluated on a collection of news articles. The authors used the full word features.
They had considered the as the weighting method for feature selection and statistics as a ranking metric. Experiments showed that both methods were of superior performance on the test corpus while SVM has better Micro recall than KNN. The Micro precision values of both classifiers are similar, with KNN a bit better. When the number of features is low, classifiers have similar performance, with advantage given to SVM. But SVM outperforms KNN clearly as the number of features

increases. The prediction time of SVM is better also than KNN.

By used support vector machine (SVM) classifier and k-nearest neighbor (KNN) the author had implemented text classifier for Kannada articles. The results were shown in term of Precision, recall, accuracy measure. The author used an in-house Kannada that consists of 48 folder which classified in different categories and referred as the result of unavailable Kannada corpus. The SVM classifier outperformed, 72%, classifier and k-nearest neighbor (KNN) classifier in the experiment result.

By investigated Support Vector Machine algorithm (SVM) on different Kannada data sets. The authors preprocessed the data by removed characters, punctuation marks, kannada letters, kannada function words and normalization some Kannada letters. The experimental results revealed that SVM algorithm outperforms the other methods with regards to all measures.

IV. DATASET AND ALGORTHIM

Data Exploration

In this study, we are using the handwritten Kannada letters. Some initial data exploration reveals that our training set contains many samples in total and 785 features. Each sample in the dataset represent an image that is 28 pixels in height and 28 pixels in width, hence the total of 785 pixels. Each image is labeled with their corresponding category that is the actual kannada characters. Some examples are shown in Figure 1.

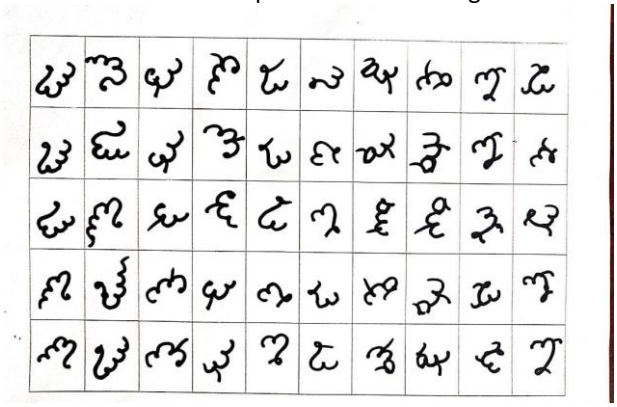

Figure 1

Algorithms and Techniques

It has been shown that Support Vector Machines (SVMs) can be applied to image and hand-written character recognition. SVMs are effective in high dimensional spaces, hence it makes sense to use SVMs for this study given the high dimensionality of our input space, i.e. 785 features. However, SVMs don't perform well in large datasets as the training time becomes cubic in the size of the dataset. This could be an issue as our dataset containing each folder 410 samples which is quite large. To deal with this issue, we will adopt a technique proposed by a study conducted at the University of California, Berkeley, which is to train a support vector machine on the collection of nearest neighbors in a solution they called “SVM-KNN”. Training an SVM on the entire data set is slow and the extension of SVM to multiple classes is not as natural as Nearest Neighbor (NN). However, in the neighborhood of a small number of examples and a small number of classes, SVMs often perform better than other classification methods.

We use KNN as an initial pruning stage and perform SVM on the smaller but more relevant set of examples that require careful discrimination. The process of a quick categorization, followed by successive finer but slower discrimination was the inspiration behind the “SVM-KNN” technique.

Data Preprocessing

Since the original dimension is quite large (785 input features), the dimensionality reduction becomes necessary. First, we extract the principal components from the original data. We do this by fitting a Principle Component Analysis (PCA) on the training set, then transforming the data using the PCA fit. We used the PCA module of the scikit-learn Python library with components set to 60 to transform the dataset. The first 60 principal components can interpret approximately 72% of total information (in terms of total variance retained), which suffice to be representative of the information in the original dataset. We thus choose the first 60 principal components as the extracted features.

We also applied cross validation to split the dataset into training and testing sets, retaining 40% of the data for testing. We use the Stratified Shuffle Split module of the scikit-learn Python library passing in the label values as one of the parameters. Stratified Shuffle Split returns train and test indices to split the data in train and test sets while preserving the percentage of sample of each class.

Implementation

Our simple implementation of SVM-KNN goes as follows: for a query, we compute the Euclidean distances of the query to all training examples and pick the K nearest neighbors. If the K neighbors have all the same labels, the query is labeled and exit.

Else, we compute the pairwise distances between the K neighbors, convert the distance matrix to a kernel matrix and apply multiclass SVM. We finally use the resulting classifier to label the query. The implementation is illustrated.

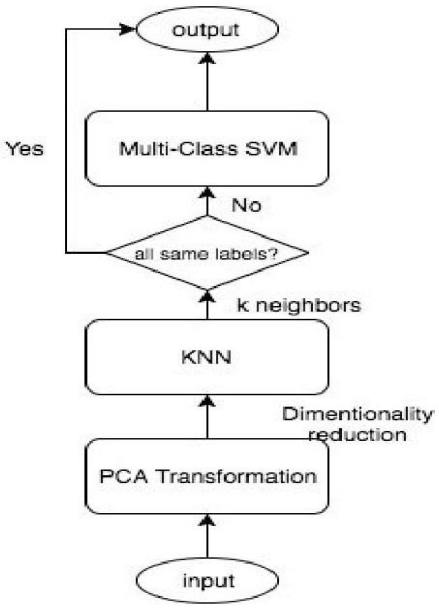

Figure 2

All the algorithms used in our implementation came from the scikit-learn Python library, version 0.17.1. Namely we used PCA for dimensionality reduction, Stratified Shuffle Split for cross-validation, K-Neighbors Classifier to find the k nearest neighbors and SVC to train our multi-class SVM.

V. RESULTS

Model Evaluation and Validation

During development, a validation set was used to evaluate the model. I split the dataset into training and test sets. The final hyperparameters were chosen because they performed the best amongst the tried combinations. A final value of k=3 yielded the best results, as increasing this value resulted to lower accuracy scores but also increased prediction time as it takes more time to find the nearest neighbors. A lower k value makes sense for our model because we are trying to find the few samples where KNN has a hard time establishing a decision boundary and apply SVM to perform a more coarse-grained classification.

To verify the robustness of the final model, I use a cross validation technique (Stratified Shuffle Split) on the dataset to ensure that the model generalizes well by using the entire dataset for both training and testing. The model consistently categorized the handwritten characters with a 72% accuracy.

Justification

Applying the SVM-KNN technique on the handwritten dataset , I got the following result:

The classification accuracy is **72.4%**, which is better than that of the benchmark. Therefore, we can conclude that our model is adequate for solving the problem of classifying handwritten characters in the dataset as it is able to accurately categorize well with an accuracy quite close to humans. However, our model is useful in a limited domain. Some changes would have to be made to solve a bigger problem of recognizing multiple Character in an image, or recognizing arbitrary multi-Character text in unconstrained natural images.

VI. CONCLUSION

In this paper the problem of classifying Kannada letters was discussed. The KNN and SVM techniques were used to handle the classification problem. Datasets were pre-processed by used approaches to improve the accuracy of the classifiers. Firstly, to build the corpus small visual basic tool used to remove the special marks. Normalisation was used to some datasets to decrease the no. of features. The accuracy measures indicated that the SVM algorithm outperformed all the other algorithm in the training stage. The accuracy of the classifier was measures by using precision recall and measures. In the future work the investigation of stemming approaches will be using to compare it with this recent experiment.

VII. REFERENCES

[1] Cheng-Lin Liu, Hiroshi Sako, Hiromichi Fujisawa, “Performance evaluation of pattern classifiers for handwritten character recognition”, International journal on Document Analysis and Recognition, September 2001.

[2] B.V.Dhandra, Gururaj Mukarambi, Mallikarjun Hangarge, “Handwritten Kannada Vowels And English Character Recognition System” International journal of image processing and vision science, Volume 1, Issue 1,2012.

[3] Shashikala Parameshwarppa, B.V.Dhandra, “ Handwritten Kannada Characters Recognition Using Curvelet Transform”, International Journal Of Computer Applications, National Conference On Digital Image And Signal Processing, DISP 2015.

[4] Sandhya Arora, Mita Nasipuri, L.Malik, D.K.Basu,”Performance Comparision Of SVM And ANN For Handwritten Devanagari Character Recognition”, International Journal Of Computer Science Issues, Volume 7, Issues 3, May 2010.

[5] Mahesh Jangid Kartar Singh, Renu Dhir, Rajneesh Rani,"Performance Comparision Of Devanagari Handwritten Numerals Recognition", International Journal Of Computer Applications, Volume 22-No-1, May 2011.

[6] Baheti M.J, Kale K.V,"Gujarati Numeral Recognition: Affine Invariant Moments Approach", International Conference On Recent Trends In Engineering & Technology, March 2012.

[7] Amrita Hirwani, Sandeep Gonnade,"Handwritten Character Recognition System Using Neural Network", International Journal Of Advance Research In Computer Science And Management Studies, Volume 2, Issue 2, Feb 2014.

[8] Rajbala Tokas, Aruna Bhadu," A Comparative Analysis Of Feature Extraction Techniques For Handwritten Character Recognition", International Journal Of Advanced Technology & Engineering Research.

[9] Suruchi G.Dedgaonkar, Anjali A.Chandavale, Ashok M.Sapkal,"Survey Of Methods For Character Recognition", International Journal Of Engineering And Innovative Technology, Volume 1, Issue 5, May 2012.