Empirical Evaluation of Character Classification Schemes

Neeba N.V, C.V Jawahar
Centre for Visual Information Technology, International Institute of Information Technology, Hyderabad, India - 500 032

Abstract

In this paper, we empirically study the performance of a set of pattern classification schemes for character classification problems. We argue that with a rich feature space, this class of problems can be solved with reasonable success using a set of statistical feature extraction schemes. Experimental validation is done on a data set (of more than 5,00,000 characters) collected and annotated from books printed primarily in Malayalam. Scope of this study include (a) applicability of a spectrum of classifiers and features (b) scalability of classifiers (c) sensitivity of features to degradation (d) generalization across fonts and (e) applicability across scripts.

1. Introduction

Large number of pattern classifiers exist in the literature. Performance of these classifiers depend on the problem, features used and other problem parameters[1]. A number of comparative studies on classifiers have been found in the literature. STATLOG [2] was considered to be the most comprehensive empirical comparative study for pattern classifiers 10 years back. A recent study focusing on empirical comparison of many recent approaches has been reported by Caruana [3]. The best performing classifier in their study was problem dependent, even though some of the classifiers always outperformed most others. Lecun *et. al* [4] reported a comparative study of various convolutional neural network architectures as well as other classifiers for the problem of handwritten digit recognition. Most of the previous studies were limited to relatively small number of classes, and often tested on the UCI, NIST or USPS data sets.

This study is primarily focused on character classification issues in Indian scripts, with special emphasis on Malayalam. Commercial OCR systems are available for Roman scripts. However, character recognition problem in Indian scripts is still an active research area [5]. A major challenge in the development of OCRs for Indian scripts comes from the larger character set, which results in a large class classification problem.

Some of the interesting results of our experiments are summarized below. SVM classifiers are found to outperform other classifiers throughout the experiments. The naive Bayes and decision tree classifiers are the poorly performing ones. Statistical features with a rich feature space performed well across the classifiers. A large feature space derived with the statistical feature extraction schemes, and a classifier with high generalization capability are found to be the ideal candidates for solving character classification problems in Indian languages.

2. Problem Parameters

This study consider a spectrum of classifiers and features for the comparison.

Classifiers. One of the most popular classifiers is a nearest neighbour classifier. Its extension to K-nearest neighbor (KNN) is a supervised learning algorithm where a new instance is classified based on majority of the labels of K-nearest neighbors. It has been shown that by computing nearest neighbors approximately, it is possible to achieve significantly smaller computational time (of the order of 10's to 100's) often with a relatively small actual errors. The approximate nearest neighbour algorithm we employ here corresponds to [6].

Another popular classifier, which classifies samples by a series of successive decisions, is a decision tree. The most important feature of a Decision Tree Classifier(DTC) is its capability to break down a complex decision-making process into a collection of simpler decisions, thus providing a solution which is often easier to interpret. We employ a binary decision tree computed using OC1 [7]. We also study the performance of neural network classifiers. Two different architectures are explored: multilayer perceptron (MLP) and convolutional neural network (CNN). Experiments were conducted by changing parameters like the number of hidden units, number of epochs, and the momentum term in MLP. Finally, the best results are reported. Convolutional Neural Networks (CNN) are shown to produce excellent recognition rates for digit recognition problem by Lecun *et. al.* [4]. We use a 5 layers CNN with architecture same as LeNet-5. We also compare with a Naive Bayes (NB) classifier. This classifier is known to be mathematically optimal under restricted settings.

No empirical evaluation is complete without evaluating the Support Vector Machine (SVM) classifier, at this stage. SVMs have received considerable attention in recent years. SVMs are large margin classifiers with high generalization capability [8]. SVMs are basically binary classifiers. In our

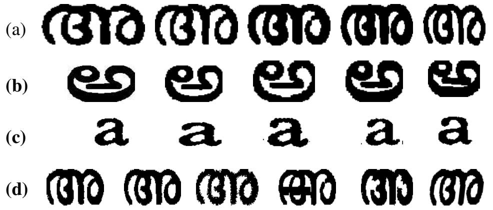

Figure 1. Examples of character images from Malayalam (a), Telugu (b) and English (c) scripts, and degraded characters (d).

study, we consider two possible methods of fusing the results of the pair-wise classifiers. First one computes the majority of all the classifiers. We refer to this as SVM-1. The second SVM classifier (SVM-2) integrates the decisions using a decision directed acyclic architecture (DDAG) [8].

Features. In this study, we employ features which are relatively generic. In the context of character classification, this means that the features are highly script-independent. The first class of features are based on moments. We use Central Moments (CM) and Zernike Moments (ZM). They are popular for 2D shape recognition in image processing literature. Another class of feature extraction strategies, we used, employ orthogonal transforms for converting the input into a different representation and select a subset of dimensions as effective features. We use Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) as the representative from this class of feature extraction schemes. Technical details of many of these feature extraction methods can be found in [9].

A popular class of feature extraction schemes extracts the features by projecting the input (image) into a set of vectors. We consider three candidate algorithms from this class of feature extraction schemes. They are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Random Projections (RP). The PCA and LDA are popular in pattern recognition literature. In both these schemes, the feature extraction scheme is derived out of the data covariance. Random projection is a data independent method for dimensionality reduction. A set of orthogonalized and normalized random vectors are used as basis vectors for this transformation (or feature extraction).

We also compare the performance by treating the image itself (IMG) as the feature vector. This does not result in any dimensionality reduction. Avoiding any explicit feature extraction assumes that the data do not vary significantly in appearance. For the benchmarking, we also consider an image resulting out of distance transform (DT) as yet another feature. This feature is similar to a fringe map.

There could be numerous other possibilities for feature extraction. However, we have limited our attention to a set of popular and promising feature extraction schemes. One could also think of extracting script specific features to exploit the specific characteristics of the script. However, generating a rich high dimensional feature space with such hand-crafted features could be a difficult task.

Datasets. We employ binary character images from documents in multiple languages for the study. All the images are scaled to a common size of 20×20 pixels. The script used for experiments are Malayalam, Telugu and English. We work on an annotated corpus mentioned in [10]. Examples of character images from the the datasets are given in the Figure 1(a)-(c). Around 5% of the datasets are randomly picked for training, and the rest is used for testing. The number of classes in this experiment is 205, 72 and 350 respectively for Malayalam, English and Telugu. Malayalam dataset is more than $5,00,000$ samples, collected from 5 different books.

3. Empirical Evaluation and Discussions

Experiment 1: Comparison of Classifiers and Features. In the first experiment, we compare the performance of different classifiers and features on the Malayalam data and the error rates are presented in the Table 1. The classifiers considered for the study are MLP, KNN, ANN, SVM-1, SVM-2, NB and DTC. We also compared the results with CNN, which resulted in an error rate of 0.93%. Reader may note that feature extraction is embedded in the CNN, and can not be compared as in Table 1.

For MLP, the reported results are with the number of nodes in the hidden layer 60, number of epochs 30 and momentum 0.6. For both KNN and ANN, Euclidean distance is used, and the results are reported with $K = 5$. Here SVM results are reported with linear kernel.

Feat	Dim.	MLP	KNN	ANN	SVM-1	SVM-2	NB	DTC
CM	20	12.04	4.16	5.86	10.04	9.19	11.93	5.57
DFT	16	8.35	8.96	9.35	7.88	7.86	15.33	13.85
DCT	16	5.43	5.11	5.92	5.25	5.24	8.96	7.89
ZM	47	1.30	1.98	2.34	1.24	1.23	3.99	8.04
PCA	350	1.04	1.14	2.39	0.37	0.35	4.83	5.97
LDA	350	0.55	0.52	1.04	0.35	0.34	3.20	4.77
RP	350	0.33	0.50	0.74	0.34	0.34	3.12	8.04
DT	400	1.94	1.27	1.98	1.84	1.84	4.28	2.20
IMG	400	0.32	0.56	0.78	0.32	0.31	1.22	2.45

Table 1. Error rates on Malayalam dataset.

The Malayalam data, described in the previous section, is used for this experiment. A series of feature extraction schemes starting from moments to linear discriminant analysis is used for the study. Please refer to the previous section for the acronyms used in the Table 1. These feature extraction schemes are language/script independent.

It can be seen that SVM classifiers outperformed all other classifiers because of their high generalization capability. KNN also performs moderately well, with a very high classification time. The DTC and NB performed the worst of all. In cases of certain features, KNN had performance comparable to SVM. However, this was obtained with significantly higher computational requirement.

Observation: We observe that SVM classifier outperform other classifiers. A class of feature extraction techniques, *based on the use of raw image and its projection onto an uncorrelated set of vectors resulted in the best performance.*

Experiment 2:Richness in the Feature space. One other important observation from the previous experiment is that, the classification accuracy can be improved using a large number of features. For a set of feature extraction techniques (LDA, PCA, RP, DCT, DFT), we varied the number of features used and conducted the classification experiment on the Malayalam data. Results are presented in Figure 2. It is observed that the error rates rapidly decreases with the increase in number of features initially and then remain more or less constant. When the number of features is small, LDA outperforms PCA. However, with a large number of features PCA, LDA, RP etc. performs more or less similarly.

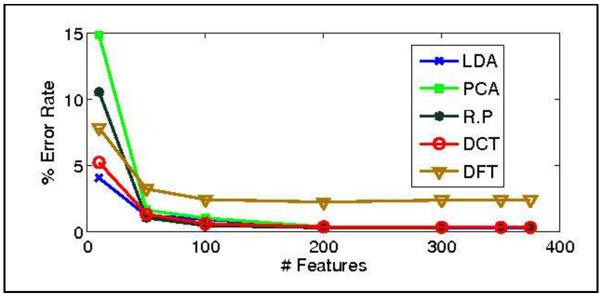

Figure 2. Richness in feature space.

Observation: We observe that for better performance, a rich feature space is required for large class problems. If the feature space is rich, they could also become discriminative for most classifiers. It may be noted, with a large feature vector computed using a statistical technique, character classification problem can be solved with reasonable success.

Experiment 3: Scalability of classifiers. We now look into a relatively un-noticed aspect of pattern classifiers – scalability to the number of classes. We conduct the experiments with increasing the number of classes. The classes are selected randomly and the experiments are conducted multiple times, and finally the average accuracies are reported in Figure 3. It is observed that the performance of all the classifiers goes down as the number of classes increases. Most of the publicly available multiclass datasets (eg. UCI data sets) have total number of classes in few tens. One of the challenges in character recognition in Indian languages is to design classifiers that can scale to hundreds of classes [5]. *Observation: Out of all the classifiers considered, we observe that the SVM classifiers (SVM-1 and SVM-2) degrade gracefully when the number of classes increases. The second best class of classifiers is the Neural network classifiers.*

Experiment 4: Degradation of Characters. Characters in real documents are often degraded. We now investigate the performance of various feature extraction schemes for degradation. We used the degradation models in [11] for the systematic studies. On a similar line, we also modeled ink blobs, cuts and shear to analyze the performance of the features. Some examples of the degraded images from the dataset is shown in Figure 1(d).

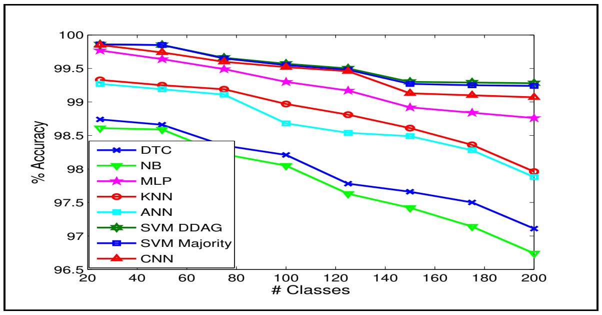

Figure 3. Scalability: Accuracy of different classifiers Vs. no. of classes.

Out of a wide spectrum of features studied, our observation has been that the statistical features are more robust compared to structural features. In presence of excessive degradation, when the characters gets cut into multiple parts, most of the feature extraction schemes had difficulty. This problem will have to be understood as a segmentation problem. Structural features like number of loops, junctions etc. were found to be highly sensitive to degradations.

Statistical features are reasonably insensitive to the small degradations (D-1, D-2 and D-3) as shown in Table 2. These degradations are primarily three different levels of boundary erosion. Features like distance transforms (DT) which works well with the clean images fails drastically in the presence of ink blobs as well as cuts in the symbols. With shear, performance of all the features reduces. But the performance degradation with PCA, LDA, RP and raw image (IMG) are much better than the other features in the study.

Feature	D-1	D-2	D-3	Blob	Cuts	Shear
CM	9.45	9.46	10.97	16.28	12.33	30.07
DFT	7.89	7.93	7.98	26.70	8.73	18.90
DCT	5.71	5.72	6.07	19.80	7.93	16.46
ZM	1.96	1.98	2.10	8.41	4.35	17.75
PCA	0.30	0.31	0.32	2.17	0.64	8.59
LDA	0.39	0.39	0.40	2.01	0.61	7.32
DT	1.75	1.98	2.21	10.33	5.07	12.34
RP	0.48	0.67	1.04	3.61	0.71	6.75
IMG	0.32	0.33	0.33	2.78	0.66	6.84

Table 2. Error rates of degradation experiments on Malayalam Data, with SVM-2.

Observation: Our observation is that statistical features like LDA are better suited to address the degradations in the data set. Shear is a challenging degradation to address. Traditional feature extraction schemes need modifications to obtain acceptable performance on shear.

Experiment 5: Generalization Across Fonts. This study mainly points towards the performance variation of classifier schemes with minor variations in the font. We included 5 popular fonts in Malayalam (MLTTRevathi, MLTTKarthika, MLTTMalavika, MLTTAmbili and MLTTKaumudi)in this study. The experiment is conducted by training the classifier with samples from 4 different fonts and test on the fifth font. We use SVM-2 as the classifier and LDA features. The results are reported in Table 3. The one dataset(S1) is without any degradation, and the second one(S2) is with degradation. It can be observed that better generalization across fonts can be obtained by adding degradation to the training data. Also note that, this observation need not applicable to a completely different and fancy font. This experiment is limited to popular fonts which are often used for pubilshing. *Observation: Generalization across similar type fonts can be achieved by adding some degradation to the training data.*

	Font-1	Font-2	Font-3	Font-4	Font-5
w.o.d	98.15	95.49	92.52	94.27	92.22
w.d	98.97	97.14	95.22	94.59	94.65

Table 3. Error rates on different fonts.(*w.o.d - without degradation, w.d- with degradation.*)

Experiment 6: Applicability across scripts. Now, we demonstrate that the observations of the previous experiments are also extend-able to other scripts. For this, we consider, the Telugu and English data. We use around 50000 real character images for Telugu and English experiments. They are obtained from scanned document images for the experimentation. In all our experiments, SVM-2 classifier had shown the best results and we present the results of this SVM-2 classifier in Table 4.

Feature	Telugu		English	
	20×20	40×40	20×20	40×40
CM	20.78	12.32	7.25	6.48
ZM	8.45	5.48	2.04	1.12
DCT	9.67	2.71	2.14	1.04
DFT	15.71	6.71	5.37	3.31
PCA	4.62	2.93	0.86	0.46
LDA	2.56	1.67	0.29	0.23
RP	2.49	1.66	0.28	0.23
DT	3.48	3.17	0.98	0.87
IMG	3.18	2.84	0.28	0.23

Table 4. Experiments on various scripts, with SVM-2.

We conducted experiments with 2 different image sizes, 20×20 and 40×40 pixels. The images of size 40×40 resulted in better accuracy than the 20×20. This is because the character in Telugu have relatively more complex shapes than English and Malayalam. With increase in image size, the feature space becomes further rich and possibly more discriminative. *Observation: We observe that our conclusions on character classification are highly script/language independent.*

In this paper, we have tried to provide certain level of the low-level details of the experiments and implementation. However, some fine aspects have been avoided due to the constraint in space. The absolute values of error rates may not mean that the OCR problem for these scripts can expect these performances. These rates are obtained on isolated segmented characters. To improve the accuracies beyond whatever we have reported here, one may have to tune the parameters, fuse the features and employ better image processing and segmentation algorithms. That is not the objective of this work.

The cost we need to pay for a richer feature space is the additional computations in pattern classification. An alternate way of achieving this is using kernels as in SVM. Our experience is that, one can obtain very high classification rates and efficient classification on our state of the art desktop computers.

4. Conclusion

In this paper, we present the results of our empirical study on character classification problem focusing on Indian scripts. The dimensions of the study included performance of classifiers using different features, scalability of classifiers, sensitivity of features on degradation, generalization across fonts and applicability across three scripts etc. We have demonstrated that with a rich feature space, the problem is solvable with an acceptable performance using state of the art classifiers like SVMs. In the future work, we would like to provide more detailed/analytic explanations for the empirical evidences reported here.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.

[2] R. King, C. Feng, and A. Shutherland, “Statlog: comparison of classification algorithms on large real-world problems,” *AAI*, vol. 9, pp. 259–287, June 1995.

[3] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *ICML*, 2006.

[4] Y. e. Lecun, “Learning algorithms for classification: A comparison on handwritten digit recognition,” in *Neural Networks: The Statistical Mechanics Perspective*, pp. 261–276, 1995.

[5] U. Pal and B. B. Chaudhuri, “Indian script character recognition: a survey,” *Pattern Recognition*, vol. 37, no. 9, 2004.

[6] S. Arya and H.-Y. A. Fu, “Expected-case complexity of approximate nearest neighbor searching,” in *SODA*, pp. 379–388, 2000.

[7] S. K. Murthy, S. Kasif, S. Salzberg, and R. Beigel, “Oc1: A randomized algorithm for building oblique decision trees,” tech. rep., 1993.

[8] J. Platt, N. Cristianini, and J. Shawe-Taylor, “Large margin dags for multiclass classification,” in *ANIPS*, pp. 547–553, 2000.

[9] O. Trier, A. Jain, and A. Taxt, “Feature extraction methods for character recognition - a survey.,” in *Pattern Recognition 29,*, pp. 641–662, 1996.

[10] C. V. Jawahar and A. Kumar, “Content-level annotation of large collection of printed document images,” in *ICDAR*, pp. 799–803, 2007.

[11] Q. Zheng and T. Kanungo, “Morphological degradation models and their use in document image restoration,” in *Int. Conf. on Image Processing*, pp. 193–196, 2001.