UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

A New Approach for Optimal Selection of Features for Classification based on Rough Sets, Evolution and Neural Networks

Application to Handwritten Digits

Eddy Alejandro Torres Constante

Matemáticas

Trabajo de fin de carrera presentado como requisito para la obtención del título de Matemático

Quito, 29 de Noviembre de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN DE TRABAJO DE TITULACIÓN

A New Approach for Optimal Selection of Features for Classification based on Rough Sets, Evolution and Neural Networks

Application to Handwritten Digits

Eddy Alejandro Torres Constante

Calificación:

Nombre del profesor, Titulo académico: Julio Ibarra, M.Sc

Firma del profesor ....

Quito, 29 de Noviembre de 2021

Derechos de Autor

Firma del estudiante	....
Nombres y apellidos:	Eddy Alejandro Torres Constante
Código:	00326635
Cédula de Identidad:	1104732043
Lugar y fecha:	Quito, Noviembre de 2021

Dedicado a mi familia

Resumen

En el reconocimiento de números, uno de los desafíos es la alta dimensionalidad de los datos que afecta el rendimiento de los algoritmos. El reconocimiento de patrones permite establecer propiedades clave entre conjuntos de objetos. En este contexto, la teoría de conjuntos aproximados juega un papel importante al trabajar con el concepto de superreductos que son de hecho subconjuntos de atributos que preservan la capacidad de todo el conjunto de atributos para distinguir objetos que pertenecen a diferentes clases. Desafortunadamente, encontrar esta reducción para grandes conjuntos de datos tiene una complejidad exponencial debido a la cantidad de objetos por clase y la cantidad de atributos por objeto. Este artículo propuso un nuevo enfoque para tratar estos problemas de complejidad presentes en conjuntos de datos reales para obtener un discriminador lo suficientemente cercano a un mínimo. Aprovecha el trasfondo teórico de la teoría de conjuntos aproximados, considerando especialmente aquellos superreductos de longitud mínima. En la literatura, existe un algoritmo para encontrar estas reducciones de longitud mínima. De hecho, funciona bien para una pequeña muestra de objeto por clase de todo el conjunto de datos. Para ampliar la capacidad de esta lista de superreductos para retener la capacidad de discernir sobre un enorme conjunto de datos, se realiza la evolución, tomando como población inicial un subconjunto de la lista completa de superreductos. El discriminador propuesto se evalúa y compara con algoritmos de última generación y el rendimiento declarado del conjunto de datos para diferentes modelos.

Palabras clave: reconocimiento de patrones, complejidad exponencial, clasificación numérica manuscrita, superreductos, redes neuronales, precisión, estrategia evolutiva, longitud mínima.

Abstract

In number recognition one of the challenges is the high dimensionality of data that affects the performance of algorithms. Pattern recognition allows establishing key properties among sets of objects. In this context, Rough Set Theory plays an important role as working with the concept of super-reducts which are in fact subsets of attributes that preserve the capability of the entire set of attributes to distinguish objects that belong to different classes. Unfortunately, finding this reducts for large data sets has exponential complexity due to the number of object per class and the number of attributes per object. This paper proposed a new approach for dealing with this complexity problems present in real data sets to obtain a close enough to a minimal discriminator. It takes advantage of the theoretical background of Rough Set Theory, specially considering those super-reducts of minimal length. In literature, there is an algorithm for finding this minimal length reducts. In fact, it performs well for an small sampling of object per class of the entire data set. To extend the ability of this super-reduct list to retain the ability of discern over a huge data set evolution is performed, taking as initial population a subset of the entire list of super-reducts. The proposed discriminator is evaluated and compared against state-of-the art algorithms and data set declared performance for different models.

Keywords: pattern recognition, exponential complexity, handwritten number classification, super-reducts, neural networks, accuracy, evolutionary strategy, minimal length.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Handwritten number recognition implies great challenges due to the huge quantity of information that is required for training a classification model [1], [2]. Moreover, the multiple difference of inputs, the distinct ways of interpreting as humans a handwritten text allow us to think that our brain only consider a reduced amount of all the information received by our senses. In this research work, a new approach for optimal selection of features for handwritten number recognition is proposed to find a close enough to a minimal subset of attributes that preserve the capability of the entire set of attributes to distinguish objects that belong to different classes. For this purpose, Rough Set Theory, artificial neural networks and evolution algorithms play the key roles.

For machine learning models, having a high dimensionality of data causes multiple problems in complexity time and performance for accurate object recognition or classification. In fact, train the model with a large number of features becomes difficult as the number of operations raises exponentially [3], [4]. In reducing the dimensionality, is crucial to ensure that essential information is preserved considering every aspect of the classes that are part of the entire set. As the performance of algorithms can be degraded by data sets that contains large number of attributes by reducing the training of models reduces in complexity being specially useful for image recognition, text mining or big data [5], [6], [7], [8].

Rough Set Theory (RST) handle features reduction ensuring that the ability of discern between sets of objects is preserved. Another key point in this theory is that every aspect of this objects is considered [9], [10]. Following this technique it is ensured that every potential useful information is retained. The concept of reduct in RST states it as a subset of attributes that preserve the discernibly capacity of the entire set of attributes [11]. Hence, by its definition it matches as a candidate for using them in feature selection for machine learning models. Several algorithms have been developed during last years such to compute a single reduct as [12], [13], or [14]. There are also those to compute the entire set as [15], [16]. In our case we focus our attention on an algorithm that searches for those reducts of minimal length such as [17]. However, finding this reducts over an entire data set is an NP-hard problem, for this reason we use the power of artificial neural networks and evolution strategy to extend the discern capability of a reduct found over a sample of the entire data set.

In literature, there exist algorithms for dimensionality reduction such as PCA [18], LDC [19] or GDA [20]. Nevertheless, the theoretical background do not ensure to be strong for classification over machine learning models; we do not either ensure to preserve a minimal quantity of features such that prediction can be preformed with accuracy. Thus, in this paper, we propose a new approach to build a close enough to a minimal subset of features to ensure precision over accuracy for Handwritten Digits.

The rest of this paper is organized as follows. In Chapter II we formally describe a reduct for a boolean matrix, artificial neural networks, and evolution strategy used to achieve an equal or closed enough to a minimal subset of features. The data set MNIST is used to perform all the calculations, assesment metrics and present results. Then we detail our method and experimental configuration. Chapter III gives the results and analysis of the study. Finally, some conclusions and future works are presented in Chapter IV.

Chapter 2

MATERIALS AND METHODS

In this chapter, the theoretical background is introduced in order to understand every step of the proposed algorithm. The algorithm as the first three sections of the chapter is divided in three key components: Rough Set Theory, Neural Networks and Evolutionary Strategy. The following sections describes the data set used and finally all the components are combined to explain in detail the proposed method.

2.1 Rough Set Theory

Let U be a finite non-empty collection of objects and A a finite non-empty set of attributes. For every attribute a in A there exist a set V_a called the value set of a and a mapping $\alpha : U \to V_\alpha$. Also the attributes of A are divided into decision attributes D and condition attributes C such that $A = C \cup D$ and $C \cap D \neq \emptyset$.

Let B be a condition subset of attributes of A. The Indiscernibility Relation is defined as:

$$IND(B|D) \quad = \quad \{(x, y) \in U^2 | [\alpha(x) = \alpha(y) \forall \alpha \in B] \vee [\delta(x) = \delta(y)]\}$$

where $\alpha(x)$ is the value attribute defined previously and $\delta(x)$ is the value of the decision attribute. Hence, the set of all pairs of objects that cannot be distinguished between different classes by the attributes of B and the elements of the same class belong to the indecirnibility relation for B.

The concept of a decision reduct is important as it is defined in terms of the previously defined indecirnibility relation. In a decision systems DS this decision reduct allows us to distinguish between objects that belong to different classes.

Definition 1. Let D be the set of decision attributes and C be the set of condition attributes of a decision system DS, the set $B \subseteq C$ is a decision reduct of DS if:

1. $IND(B|D) = IND(C|D)$

2. $\forall b \in B, IND(B - \{b\}|D) \neq IND(C|D)$

For simplicity, decision reducts will be simply called reducts.

A binary table where rows represent comparisons of pairs of objects of different decision classes and columns are condition attributes is called a Binary Discernibility Matrix DM. The discernibility element $dm_{ij} \in \{0, 1\}$. $dm_{ij} = 0$ and $dm_{ij} = 1$ means that the objects of pair denoted by i are similar or different respectively in the attribute j.

Definition 2. Let DM be a discernibility matrix and r_k be a row of DM. r_k is a superfluos row of DM if there exists a row r in DM such that $\exists i|(r[i] < r_k[i]) \wedge \forall i|(r[i] \leq r_k[i])$ where $r[i]$ is the i-th element of the row r.

There is a related concept in Testor Theroy where they call the matrix obtained by removing every superfluous row matrix as Basic Matrix [21]. For simplicity we will call to the Binary Discernibility Matrix as basic matrix. Recall from [22] that the reducts of a decision system can be calculated from this basic matrix, which is an important fact to consider for the development of the algorithm.

A super-reduct is a subset of features that discerns between objects that belong to different classes.

Definition 3. Let BM be a basic matrix and L be an ordered list of condition attributes. L is associated to a super-reduct if and oly if in the sub-matrix of BM considering only the attributes in L, there is no zero row (a row with only zeros).

Proposition 1. Let BM be a basic matrix and L be an ordered list of attributes. If $\exists c_x \in L$ such that $em_L \wedge cm_{cx} = (0, .., 0)$. Then, L is not associated to a reduct.

Where em_L and cm_{cx} are the exclusion mask and the cummulative mask respectively defined in [17]. Proposition 1 ensures that no superfluous attributes are present in the reduct. This proposition is used to evaluate if a super-reduct is a reduct.

The *minReduct* [17] algorithm supports most part of the theoretical background needed. In our work we make use of their algorithm and only declare declare those definitions and propositions that were explicitly used.

2.2 Neural Networks

Feed-forward back propagation neural networks have gained their reputation due to their high use rate among the time [23]. They are present in several fields as prediction image recognition as in [24], [25], medicine problems [26],chemistry problems [27], oil and gas industry [28], water level prediction [29].

Their repercussion an usage made them the best candidate for using their properties for feature selection. The theoretical background consider the concept of neurons. Each of this is the composition of a linear regression, a bias and an activation function. This neurons are ordered by layers and their connections are known as weights. The first layer is the input layer and the last one is known as the output layer. All the layers in between are called hidden layers [30].

The back-propagation algorithm disperses the output error from the output layer through the hidden layers to the input layers so that the connection between the neurons can be recurrently calculated on training looking forward to minimize the loss function in each training iteration, so that with the enough quantity of data and training we are able to classify and predict [31]. The definition of accuracy in fact is the number of correct predictions divided by the total number of predictions and its value lays in the interval $[0, 1]$.

2.3 Evolutionary Strategy

In evolutionary strategy the main idea follows this behavior: from a population of individuals within in an environment with limited resources, a competition for those resources is performed so that the survival of the fittest as natural selection does plays its role. From generation to generation the fitness of the population is increased. Given a metric on how to evaluate the quality of an individual it is treated as a function to be maximised.

For an initial population we can initialize randomly in the domain of the function. After that we apply the metric as an abstract way to measure how fitness an individual is, where a higher value implies better. We must ensure that only some of the better candidates are chosen to seed the next generation. This is performed by applying mutation and/or recombination to them. Mutation is applied to an individual by altering some of their attributes resulting in a new individual. Recombination performed to two or more selected individuals, called parents, producing one or more new individuals, called children. By executing these operations on the parents we will end with a the next generation, called the offspring.

These new generation retains at least all the best from the previous one, being variation operations the way to increase fitness in further generations. This process has an stop criteria so that new generation creation is iterated until an individual that satisfies the metric, in a defined level, is found or computational iteration limit is reached [32], [33].

2.4 MNIST Data set

For a real testing purpose, a widely used data set in machine learning is required. The MNIST is a data set of handwritten digits which has been used for several classification and prediction models and all their values are reported officially. Handwriting recognition is a difficult task as mentioned due to the high number of attributes (pixles in the case of images) and what makes specially difficut to this data set is the huge amount of images for training (60,000) and testing (10,000). These are binary images centered at 28 pixels per 28 pixels. [34].

2.5 Proposed method

In this section we introduce the method proposed for reaching a close enough to a minimal subset of attributes able to distinguish between elements of different classes. For this purpose we dive the algorithm into two stages.

The first stage is to find a subset of reducts. As dealing with the entire 60,000 training images is impossible for memory and time complexity of the NP-hard problem we decide to randomly select a sample of 10 objects of each class. As in the MNIST data set each pixel is in the range of $[0, 255]$ we set the threshold to 100, so that every pixel with a value high to 100 was set to 1 and everything else to 0. This number of elements and threshold was chosen by experimentation as the *minReduct* algorithm performed better for the basic matrix generated from this objects.

The second step is to sort in lexicographical order as detailed in *minReduct* but with the difference that we move each of the columns to place each of the ones on the rows as

c_3	c_0	c_1	c_2	c_4	c_5	c_6
c'_0	c'_1	c'_2	c'_3	c'_4	c'_5	c'_6
1	1	0	0	0	0	0
1	0	1	0	0	0	0
1	0	0	0	0	1	0
0	0	1	0	0	0	1
0	0	0	1	1	0	0
0	0	0	0	1	1	1

Figure 2.1: Reduct list search performed over the rearranged basic matrix

close to the left as possible, so that we form an upper-triangular representation. Consider that this matrix is not necessarily square so we only want to ensure this triangular representation as close as we are able. All this column changes must be storaged in order to translate to the original indexes after the algorithm finishes.

The third step is to choose the number of reducts desired to find, this number is required as represent the number of individuals in a population when evolution is performed. A variation to *minReduct* is performed to do it, we first search for a maximum length limit and when found every time that a super-reduct is found we evaluate Proposition 1 to evaluate if it is a reduct or not. Once all the desired number of reducts are found we are able to proced to the next stage of the algorithm where neural networks and evolutionary strategy are combined.

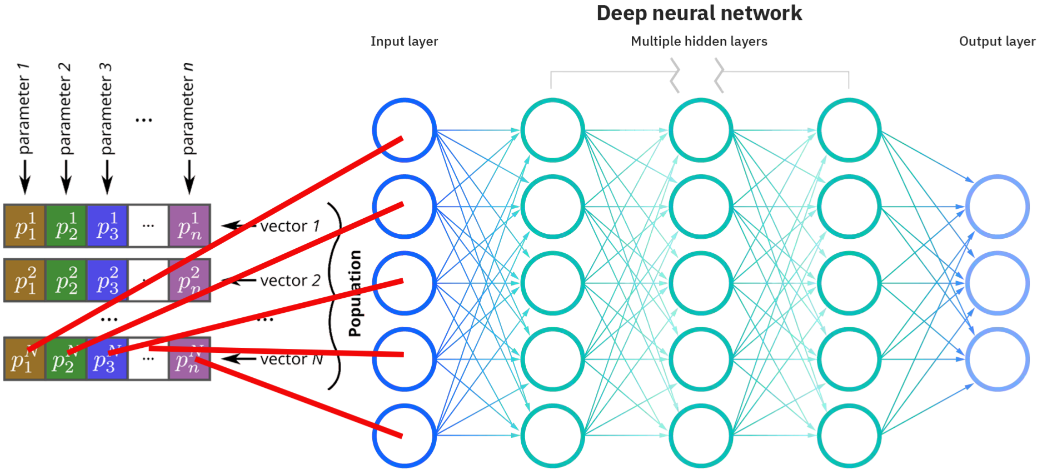


Figure 2.2: Individual relation to neural networks

For evolutionary strategy, as the initial population is the list of reducts found we ensure to mutate each candidate once for allowing enough variation to begin. Then we save the best candidates and use Univariate Marginal Distribution Algorithm (UMDA) [35] for next generation creation. The fitness function is defined as the accuracy on the feed-forward back propagation neural network model. To ensure that we reach a close to a minimal subset of attributes we only perform mutation at a 0.5% rate. Finally, we must define a prediction accuracy threshold as stop criteria or a maximum number of generations so that the algorithm is able to finish.

2.6 Experimental setup

For the sampled sub set of features we decide to choose randomly 10 objects of each class. A class is the group of all digits from 1 to 9 labeled with the same decision attribute. With the threshold established to 100 we proceed to binarize the matrix of comparisons of each pair of selected elements removing all the superfluous rows. After this process the lexicographical order is performed and the matrix is rearranged moving all the ones to the left searching for an upper triangular representation. all the indexes are storaged to be translated at the end.

Once all this has been performed the *minReduct* algorithm is executed but with some changes. The first step is to search for a maximun length bound. For accomplishing it multiple maximum lengths are evaluated under a period of time, once it is reached the minimum found is considered as the general maximum length. With this value, *minReduct* runs again but every time that it founds a super-reduct it evaluates if it is a reduct so that it can be appended to the solution list. A maximum number of desired reducts is previously declared so that when the solution list reaches that quantity the algorithm finishes. If not, it will search until the and and return all of the reducts found.

This solution list is translated to the original indexes and mutated in a rate of 0.5%, or three random features in order to create the initial population for evolutionary strategy. For next generation creation it uses marginal probability in order to keep all those key features and mutation is performed in the same rate in order to grow slowly for just adding the minimum features from generation to generation. this allows to ensure that when we reach a solution the biggest possible variation from another one is at most in the same rate of the mutation 0.5%.

For fitness function, the neural networks play their role. The topology used is an input layer with the same number of neurons as the length of selected features for the current individual to be tested. As activation function *relu* is used. For hidden layers there are two, the first one with 52 neurons and the second one with 26 neurons, both use the same *relu* activation function. The output layer uses $softmax$ as activation function and has 10 response neurons as there are 10 classes in our data set. Also we use as loss function Sparse Categorical Cross-entropy, used 10 epochs and set batch size equal to 1/5 of the training samples. All definitions are described in [36], [23].

The stop criteria is based on the whole data set perform ace considering all the attributes. So we set our threshold for accuracy to the maximum reported accuracy minus 0.04. Once a subset that satisfy this accuracy is found is reported as a solution and the algorithm finishes. In case it is not found at the beginning of the evolution is declared a maximum number of generations. For us this value was set to 100.

With the same topology previously declared the model is trained with some variations when a solution is found. Stratified K-Folds cross-validation are performed for 5 folds over the whole train set each fold with 20 epochs whit the same batch size. For evaluating the performance of the solution one-vs-all multi-class classification metrics are calculated. For achieving it, over the same declared topology, Stratified K-Folds cross-validation is used for model training with only two folds [37]. In general terms, this neural netowrk model is evaluated by accuracy-vs-epochs, loss-vs-epochs, multi-class precision. We also present one-vs-all ROC curves and AUC scores [38], and one vs all precision vs recall metrics [39]. Moreover, the subset of features is evaluated by using it on some of the declared models in the documentation of the MNIST data set.

Python in language version 3.7.10 was used to implement all the source code [40]. Scikit-learn (SKlearn)library [41] and Keras [42] were also used.

Chapter 3

RESULTS AND DISCUSSION

In this chapter the assessment metrics calculated over the proposed model with the selected subset of features are discussed. The initial population had a length of 13 attributes per individual and a total of 20 individuals per population was used. The subset obtained has a length of 152 attributes, which represent a total of 19.38% of the total 784 pixels per image.

3.1 Performance evaluation

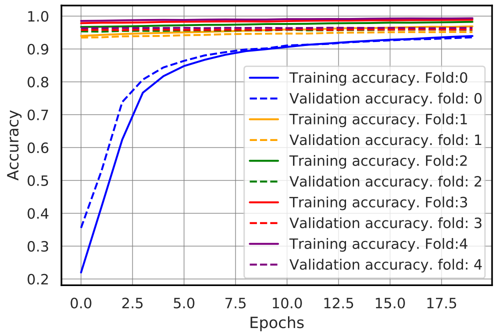


Figure 3.1: Accuracy vs Epochs

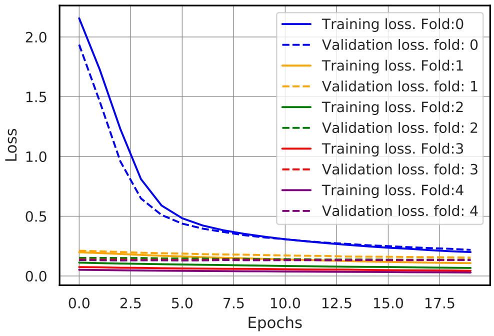
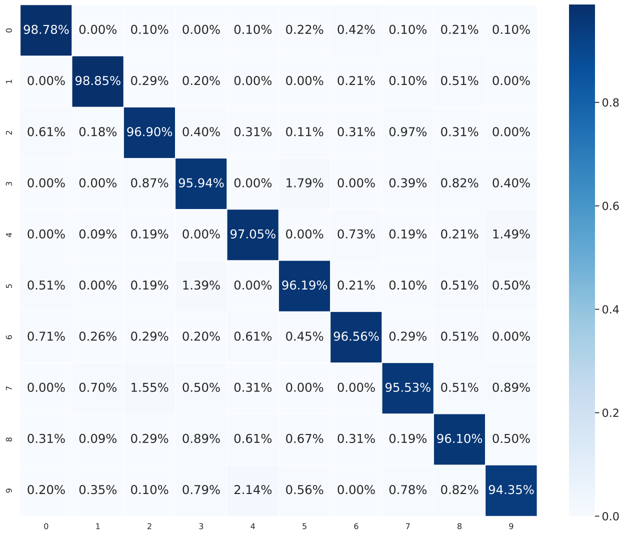


Figure 3.2: Loss vs Epochs

To evaluate accuracy and loss metrics the first step was to reduce the train and test sets considering only those 152 attributes (columns). With this new data set, the training process was performed and the testing over unknown data (10.000 images) returned the following results.

From Figure 3.1 we evidence how on each fold the accuracy in the model increases to higher values really close to 1. In fact, the reported accuracy for the model was 99.36% on training and 97.45% on validation. In terms of loss, it was reported a 0.0297 on training and a 0.0860 on validation. We can also evidence how the model is not over fitted in Figure 3.2. Its also clear that training and validation converge after some epochs, in spite of the curve being slightly different at the beginning. This can be interpreted as a good reason to state that the model fits the data and any variation is not going to be statistical significant.



	0	1	2	3	4	5	6	7	8	9
0	98.78%	0.00%	0.10%	0.00%	0.10%	0.22%	0.42%	0.10%	0.21%	0.10%
1	0.00%	98.85%	0.29%	0.20%	0.00%	0.00%	0.21%	0.10%	0.51%	0.00%
2	0.61%	0.18%	96.90%	0.40%	0.31%	0.11%	0.31%	0.97%	0.31%	0.00%
3	0.00%	0.00%	0.87%	95.94%	0.00%	1.79%	0.00%	0.39%	0.82%	0.40%
4	0.00%	0.09%	0.19%	0.00%	97.05%	0.00%	0.73%	0.19%	0.21%	1.49%
5	0.51%	0.00%	0.19%	1.39%	0.00%	96.19%	0.21%	0.10%	0.51%	0.50%
6	0.71%	0.26%	0.29%	0.20%	0.61%	0.45%	96.56%	0.29%	0.51%	0.00%
7	0.00%	0.70%	1.55%	0.50%	0.31%	0.00%	0.00%	95.53%	0.51%	0.89%
8	0.31%	0.09%	0.29%	0.89%	0.61%	0.67%	0.31%	0.19%	96.10%	0.50%
9	0.20%	0.35%	0.10%	0.79%	2.14%	0.56%	0.00%	0.78%	0.82%	94.35%

Figure 3.3: Confusion Matrix for the model trained with the resultant selected attributes

We present in Figure 3.3 the confusion matrix to analyze how classification and prediction is performed between multiple classes, this allows us to check not just a well performance, it also allows us to know where wrong predictions are happening. As the diagonal is mostly highlighted this shows that the model is able to distinguish between all the objects of the different classes. As error rate predictions in ether case is not grater than 1.55% we consider this value not significant.

For analyzing precision on prediction we present the ROC curves for all one-vs-all with their corresponding area under the curve (AUC). In Figure 3.4 we evidence an AUC approximately to 1 for every one-versus-all cases. This implies that the model has an strong performance in distinguish between all classes. Hence, we are allowed to interpret that those points chosen by the subset of attributes are able to discern, clearly classify, and predict between all classes. A higher precision and recall score is related to a better performance of the model. As presented in Figure 3.5, in average the precision value is close to 0.99 so the ability of the model to predict each of the classes is confirmed.

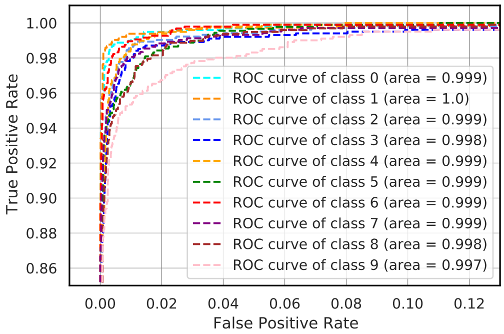


Figure 3.4: ROC Curve one-vs-all

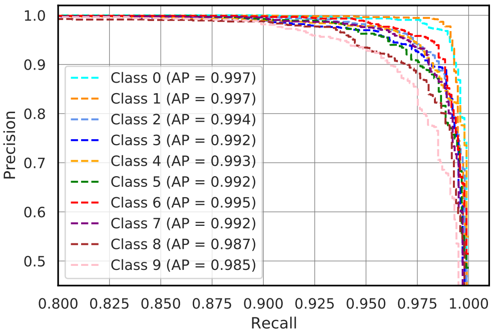


Figure 3.5: Precision-vs-Recall one-vs-all

We would like to emphasize how following this approach we a evidencing of a reduction of more than the 80% of the features is able to keep at least 97% of the accuracy with the ability of fully discern between object and classes.

3.2 State of the art-based comparison

As mentioned we can evaluate the performance of the model against the error rate reported in the MNIST documentation for other models.

Table 3.1: Solution subset of attributes evaluated in different reported models

Classifier Model	**Reported Test Error Rate (%) considering all the features**	**Replicated Test Error Rate (%) considering all the features**	**Replicated Test Error Rate (%) considering selected features only**
Linear classifier (1-layer NN) [43]	12.00%	12.70%	14.29%
K-nearest-neighbors, Euclidean (L2) [43]	5.00%	3.35%	4.35%
40 PCA + quadratic classifier [43]	3.30%	3.74%	5.36%
SVM, Gaussian Kernel [34]	1.40%	3.34%	3.06%
2-layer NN, 800 HU, Cross-Entropy Loss [44]	1.60%	1.86%	3.73%
3-layer NN, 500+300 HU, softmax, cross entropy, weight decay [34]	1.53%	1.79%	2.59%

We are able to evidence from Table 3.1 that all models that used the attributes selected preserved the error rates in a small range no bigger than a 2%. As the classifier models differ significantly in their approaches of training and prediction and the error rates is preserved we can ensure that the selected features are certainly the most relevant for distinguishes purposes over the entire data set. In terms of computational and time complexity every classification model can be considered as less complex and also faster as less operations are preformed and less data is being used.

Even in the case of using a reduction method it performs well, PCA is reducing even more the set of features and the quadratic classifier stills preserves its error rate in range. The same applies for SVM, with the consideration that is the model for which the solution subset of features perform even better than using the entire set of attributes. Our reported sub set of attributes reported an error rate of only 2.14% for neural networks systems while using 19,39% of the total amount of attributes. We consider to have enough evidence to consider this subset as a minimal enough preserver of the discernibility capacity of the whole set of attributes.

Chapter 4

CONCLUSIONS AND FUTURE WORK

This paper proposed a new strategy to find a subset of attributes able to preserve the discernibility capacity of the whole set of attributes in a group of classes. By the theoretical background of Rough Sets we were able to build an initial population of possible solutions for a sample of the entire data set which settles the beginning for an intelligent search. Evolution strategy made possible to extend this subset of attributes to be useful for the entire data set. Mutation also played the role of controller to ensure obtaining a close enough to a the minimum set of attributes. The fitness function was all in the filed of neural networks, which made possible to ensure good accuracy levels. With all this together we found a subset of attributes reduced on more than the 80% of the total amount of attributes. Recalling what was mentioned on the assessment metrics calculation and their interpretation with the subset of features found we can build a model that predicts with an accuracy of over 97%. Moreover, it can discern between all the classes and their objects. Hence, we conclude that the found subset of attributes is able to preserve the discernibility capacity of the whole set of attributes in a group of classes with a minimum length.

Furthermore, experimentation over other models evidence that the computational cost represented by calculating the reducts is worth it as the subset of features can be extended for other techniques. Additionally, the huge attribute reduction shows that not all the information is required for classifying and predicting. As future work, we propose: (1) to use reducts in average per groups of classes to used more information of the data set, (2) compare with other reduction techniques and with other data sets, as well as (3) establishing standard parameters for correctly use of the algorithm just as the density of 1's required in a basic matrix for a good performance.

Bibliography

[1] Wang, M., Wu, C., Wang, L., Xiang, D., & Huang, X. A feature selection approach for hyperspectral image based on modified ant lion optimizer. *Knowledge-Based Systems*, 168:39–48, 2019.

[2] Zhou, H., Zhang, Y., Zhang, Y., & Liu, H. Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy. *Applied Intelligence*, 49(3):883–896, 2019.

[3] Ayesha, S., Hanif, M. K., & Talib, R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. In *Information Fusion*, pages 59:45–58, 2020.

[4] Gao, L., Song, J., Liu, X., Shao, J., Liu, J., & Shao, J. Learning in high-dimensional multimedia data: the state of the art. In *Multimedia Systems*, pages 23(3):303-313, 2017.

[5] Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction In *Journal of Applied Science and Technology Trends*, pages 1(2):56–70, 2020.

[6] Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. Analysis of dimensionality reduction techniques on big data In *IEEE Access*, pages 54776–54788. IEEE, 2020.

[7] Mafarja, M. M., & Mirjalili, S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Computing*, 23(15):6249–6265, 2019.

[8] Saxena, A., Saxena, K., & Goyal, J. Hybrid technique based on dbscan for selection of improved features for intrusion detection system. In *Emerging Trends in Expert Applications and Security*, pages 365–377. Springer, 2019.

[9] Pawlak, Z. Rough sets. In *International journal of computer & information sciences*, 11(5):341–356, 1982.

[10] Pawlak, Z. Classification of objects by means of attributes. In *Polish Academy of Sciences [PAS]*, Institute of Computer Science., 1981.

[11] Pawlak, Z. Rough sets: Theoretical aspects of reasoning about data In *Springer Science & Business Media.*, (Vol. 9), 1991.

[12] Jiang, Y., & Yu, Y. Minimal attribute reduction with rough set based on compactness discernibility information tree. In *Soft Computing*, 20(6):2233–2243, 2016.

[13] Jensen, R., Tuson, A., & Shen, Q. Finding rough and fuzzy-rough set reducts with SAT. In *Information Sciences*, 255:100–120, 2014.

[14] Prasad, P. S., & Rao, C. R. IQuickReduct: an improvement to Quick Reduct algorithm. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pages 152–159, 2009, December.

[15] Chen, Y., Zhu, Q., & Xu, H. Finding rough set reducts with fish swarm algorithm. In *Knowledge-Based Systems*, 81:22–29, 2015.

[16] Chen, Y., Miao, D., & Wang, R. A rough set approach to feature selection based on ant colony optimization. In *Pattern Recognition Letters*, 31(3):152–159, 2010.

[17] Rodríguez-Diez, V., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., Lazo-Cortés, M. S., & Olvera-López, J. A. MinReduct: A new algorithm for computing the shortest reducts In *Pattern Recognition Letters*, 138:177–184, 2020.

[18] Roweis, S. EM algorithms for PCA and SPCA In *EM algorithms for PCA and SPCA*, 626–632, 1998.

[19] Park, C. H., & Park, H. A comparison of generalized linear discriminant analysis algorithms In *Pattern Recognition*, 41(3):1983–1097, 2008.

[20] Baudat, G., & Anouar, F. Generalized discriminant analysis using a kernel approach. In *Neural computation*, 12(10):2385–2404, 2000.

[21] Lazo-Cortes, M., Ruiz-Shulcloper, J., & Alba-Cabrera, E. An overview of the concept of testor. In *Pattern Recognition Journal*, 34(4):753–762, 2000.

[22] Yao, Y., & Zhao, Y. Discernibility matrix simplification for constructing attribute reducts. In *Pattern Recognition Journal*, 179(7):867–882, 2009.

[23] Simon Haykin. *Neural Networks - A Comprehensive Foundation.* 2008.

[24] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *arXiv preprint arXiv:2010.11929*, 2020.

[25] Weytjens, H., Lohmann, E., & Kleinsteuber, M. Cash flow prediction: Mlp and lstm compared to arima and prophet. In *Electronic Commerce Research*, 21(2):371–391, 2021.

[26] Kumar, S. A., Kumar, A., Dutt, V., & Agrawal, R. Multi Model Implementation on General Medicine Prediction with Quantum Neural Networks In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks*,pages 1391–1395, 2021.

[27] Abdi-Khanghah, M., Bemani, A., Naserzadeh, Z., & Zhang, Z. Prediction of solubility of n-alkanes in supercritical co2 using rbf-ann and mlp-ann. In *Journal of CO2 Utilization*, 25:108–119, 2018.

[28] Orru, P. F., Zoccheddu, A., Sassu, L., Mattia, C., Cozza, R., & Arena, S Machine learning approach using mlp and svm algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. In *Sustainability*, 12(11):4776, 2020.

[29] Ghorbani, M. A., Deo, R. C., Karimi, V., Yaseen, Z. M., & Terzi, O. Implementation of a hybrid mlp-ffa model for water level prediction of lake egirdir, turkey. In *Stochastic Environmental Research and Risk Assessment*, 32(6):1683–1697, 2018.

[30] Luo, X. J., Oyedele, L. O., Ajayi, A. O., Akinade, O. O., Delgado, J. M. D., Owolabi, H. A., & Ahmed, A. Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings In *Energy and AI*, 2020.

[31] Svozil, D., Kvasnicka, V., & Pospichal, J. Introduction to multi-layer feed-forward neural networks. In*Chemometrics and Intelligent Laboratory Systems*, 39(1):43–62, 1997.

[32] Eiben, A. E., & Smith, J. E. What is an evolutionary algorithm?. In *Introduction to Evolutionary Computing*, pages 25–48, 2015.

[33] Oliver, K. Genetic algorithms In *Genetic algorithm essentials.* pages 11–19, 2017.

[34] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[35] Alba-Cabrera, E., Santana, R., Ochoa-Rodriguez, A., & Lazo-Cortes, M. Finding typical testors by using an evolutionary strategy. In *Proceedings of the 5th Ibero American Symposium on Pattern Recognition*, page 267, 2000.

[36] Liu, W., Wen, Y., Yu, Z., & Yang, M. Large-margin softmax loss for convolutional neural networks. 2(3):7, 2016.

[37] A Ramezan, C., A Warner, T., & E Maxwell, A. Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sensing*, 11(2):185, 2019.

[38] Wandishin, M. S., & Mullen, S. J. Multiclass roc analysis. *Weather and Forecasting*, 24(2):530–547, 2009.

[39] Powers, D. M Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[40] Van, R. G., & Drake, F. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[41] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[42] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[43] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.3.

[44] Simard, P. Y., Steinkraus, D., & Platt, J. C. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Citeseer, 2003.