International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I2P223
Volume 69 Issue 2, 165-170, February 2021

Feature Selection: An Empirical Study

[*]Vandana C.P, [#]Dr. Ajeet A. Chikkamannur

*Research Scholar-Visvesvaraya Technological University, Asst. Prof, Department of Information Science Engineering, New Horizon College of Engineering, Bangalore, India
#Department of Computer Science and Engineering, R. L. Jalappa Institute of Technology, Doddaballapur, Bangalore, India
vandana.hareesh@gmail.com, ac.ajeet@gmail.com

Abstract *— Feature Selection is inevitable in today's decision-making system due to the enormous amount of heterogeneous, highly volatile data. It is important to choose the correct feature set to avoid Curse of Dimensionality and learn algorithms to behave effectively. If very few elements are chosen, satisfactory results may not be inferred, or if the number of features selected is very high, then performance is an issue. The accuracy can be improved by adding more relevant features. However, this is justifiable only up to a certain number of features. In this paper, we discussed the various types of feature selection techniques and carried out an empirical study.*

Keywords *— Feature Extraction, Entropy, Mutual Information, KNN, Clustering*

I. INTRODUCTION

In this era of IoT, where all the physical things are connected to the cyber world, they become the source of the enormous amount of data that exhibits the characteristics of Velocity, Value, Volume, Variety, and Veracity. This rapid growth of data demands the application of machine learning algorithms and techniques to automatically find knowledge from enormous, heterogeneous data sources. When machine learning is performed on high dimensional data, it leads to a phenomenon called the curse of dimensionality. The algorithm is affected adversely as the data becomes sparser in the high-dimensional space [1].

Feature Selection [2] is a very important phase during the preprocessing step during data analysis in the IoT domain. To facilitate high accuracy, features must provide enough characteristics to separate the data into classes or groups for further inference and decision-making in intelligent systems. The number of features selected is very important, and its selection is a crucial task. If very few elements are chosen, satisfactory results may not be inferred, or if the number of features selected is very high, then performance is an issue. The accuracy can be improved by adding more relevant features. However, this is justifiable only up to a certain number of features. After this critical number of features is selected, beyond this count, the growth of accuracy stagnates or even decreases. This behavior is called a peaking phenomenon (see Figure 1). Furthermore, feature selection can help to avoid the curse of dimensionality.

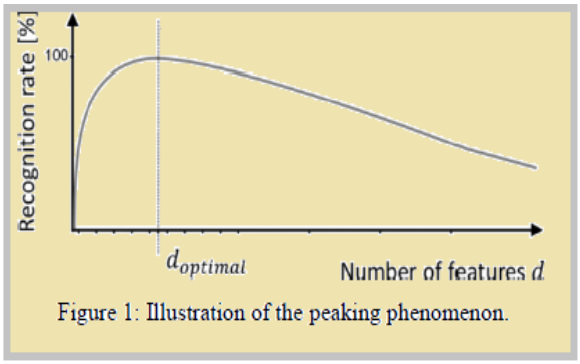

Figure 1: Illustration of the peaking phenomenon.

The feature set contains redundant and/or irrelevant features. In Figure 2a, Feature2 is alone enough to classify the two classes; Feature1 is similar for both classes. So feature I am irrelevant. In Figure 2b, both Feature1 and Feature2 help in classification equally as they carry similar information. Hence it has a redundant feature, and anyone can be removed without causing any loss of information.

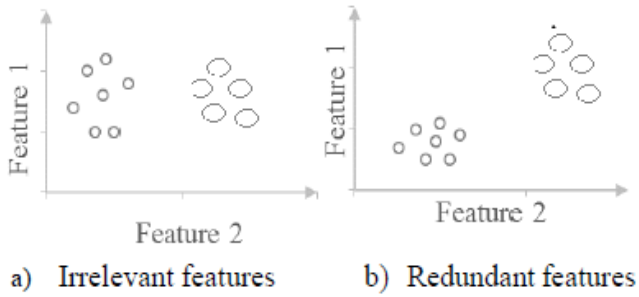

Figure 2: Illustration of irrelevant and redundant features.

Dimensionality Reduction [2][3] can provide a solution with two main techniques: feature removal and choice of features. The extraction process of software transforms the original high-dimensional objects to object space of low dimensionality. Generally, a linear or nonlinear combination of the original features is the newly built feature space. Feature selection generates a subset of relevant features which are representative of the actual feature set for model construction. Given a set of n features, the goal of feature selection is to select a subset of features (p) where p<n.

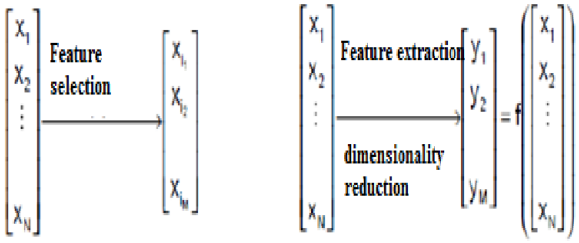

Figure 3: Feature Selection and Extraction

Section 2 discusses the various feature selection techniques in state-of-art. Section 3 exhibits the feature clustering technique. Section 4 represents the comparative study followed by the future work in section5.

II. CLASSIFICATION OF FEATURE SELECTION

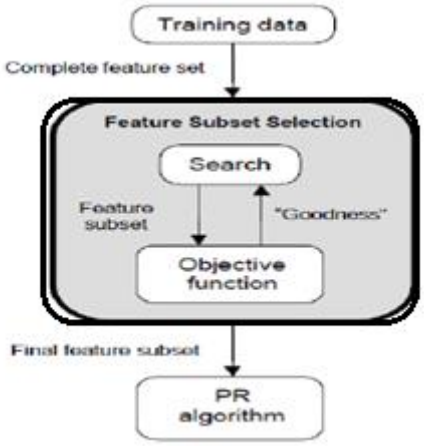

Figure 4: Feature Selection

Based on the availability of class label information in the data set (training data), they are broadly divided into Supervised, unsupervised, and semi-supervised algorithms.

Supervised feature selection aims at selecting features that help to discriminate data sets from other classes (Classification) and also to approximate the values of target variables (regression). Here, the features are selected based on their relevance which is measured in terms of its correlation with the class labels or regression target variable. The data is split into training, and test data and models are trained based on the subsets of features selected. If the feature selection phase is independent of the learning algorithms, it does a pure filter method, features are filtered out using heuristics or the characteristics of the given data.; or it can iteratively apply the learned classifier or regression to measure the quality of the selected feature so far (Wrapper model). Example include sequential forward selection (SFS) or backward feature selection(BFS). If it employs the intrinsic structure of a learning algorithm and embeds the feature selection into an underlying model like a Decision tree classifier, then it is called embedded methods. This trained classifier or regression model predicts class labels or regression targets of unseen samples in the test set with the selected features.

Most of the real-world data are unstructured, and getting them labeled is particularly expensive in both time and effort. Hence unsupervised feature selection[5] usually uses all instances of the data during the feature selection phase. Unsupervised feature selection lacks any class label and hence is generally designed for clustering problems. The feature selection phase can be independent of the training algorithms (filter methods). Alternatively, it iteratively applies training algorithms to improve the quality of selected features (wrapper methods). It integrates the choice of features into unattended learning algorithms (embedded methods). The clustering algorithm is used after the selection phase to derive the cluster shape and structure of all data sets on the selected feature.

Another category of the feature selection, which is the Semi-supervised method, employs both labeled and unlabeled data samples.

Feature Selection Strategy

A two-step wrapper method: check for a subset of characteristics and test the characteristics selected. It repeats iteratively both the two steps mentioned earlier until some of the stop criteria have been met. So 2^d will be a search space of n features that is inefficient when d is increasing. So it is proposed that greedy algorithms consider locally optimal search solutions such as Forward and Backward sequential search, hill-climbing, branch-and-bound methods. Genetic algorithms (GA) is yet another technique proposed to find optimal features.

Filter methods may yield a subset of features that may not be finally optimal with the learning algorithm as it was never guided based on this algorithm. Feature ranking is done independently (univariate) or a combination of features (multivariate). Low-ranked features are filtered out.

Embedded methods are a compromise between methods of filtering and wrapping, which integrates the collection of features into the template that is being studied. Features are selected based on the learning algorithm interaction and are much more efficient compared to wrapper approaches as they do not iteratively test the entire combination of feature sets.

The final aim of the model should be to help in the learning process by reducing the errors caused by the fitness function and making the feature coefficient to zero.

Feature Search strategy

Sequential Forward Selection (SFS)[10]: It is a heuristic search technique that starts with a single best feature selected based on the objective function. The next step is a pair of features is created using one feature from the remaining initial set and this best feature. The best pair is selected. The next triplet of features is formed using one of the remaining features and the best pair already selected. The best triplet is selected. The process repeats until a predefined number of features are selected. It performs best when the optimal subset is small. In SFS, a certain feature that may be useful in initial iterations may become redundant in further iterations, which cannot be found out later.

1. **Let Y=null**
2. **Select the next best feature x^1= arg max(Y_l+x)**
3. **Y_{l+1}=Y_l+ x^1**
4. **l=l+1**
5. **Repeat step 2**

Figure 5. SFS Algorithm

Sequential backward Selection (SBS)[10]: It is a heuristic search approach that starts with a complete feature set as the initial set. The criteria function is first computed with this initial set. At each iteration, one feature is deleted, and the criterion feature is computed for all subsets, and the worst feature is deleted. This repeats until a predefined number of features are left. It works best when the optimal subset is large. In SBS, after a feature is discarded in the initial iteration, it may become more relevant in later iterations; this usefulness cannot be evaluated.

1. **Let full set Y=X**
2. **Select the worst feature x^1= arg max(Y_l-x)**
3. **Y_{l+1}=Y_l- x^1**
4. **l=l+1**
5. **Repeat step 2**

Figure 6. SBS Algorithm

Bidirectional Search (BDS): To bring a trade-off between SFS and SBS, this approach applies SFS and SBS simultaneously to converge to the same solution.

1. **Let full set Y_0=null**
2. **Let full set Y=X**
3. **Select the next best feature x^1= arg max(Y_0+x)**
4. **Y_{l+1}=Y_l+ x^1**
5. **l=l-1**
6. **Select the worst feature x^1= arg max(Y_l-x)**
7. **Y_{l+1}=Y_l- x^1**
8. **l=l+1**
9. **Repeat step 2**

Figure 7. Bidirectional Search (BDS) algorithm

Sequential floating forward selection (SFFS)[10] algorithm will always start with an empty set. After each forward step in the algorithm, SFFS performs an exact backward step if the objective function has to be increased to meet the criteria. The sequential floating backward selection (SFBS) algorithm starts from the full set. After every backward step, SFBS does exactly a forward step if the objective function increases the criteria.

Mutual Information Measure

Let A be a discrete random variable. The probability density function for all event a belongs to the domain of A is represented as p(a). Information entropy H(A) is defined as the uncertainty of A, which are measured as

$$H(A) = -\Sigma_{a\in dom(A)} p(a) \log p(a)$$

H(A) thus represents the information amount of the previously defined variable A. Entropy of a system is the amount of uncertainty or disorder; in this regard, the entropy of a random variable is the amount of uncertainty associated with this variable. H(a) does not depend directly on the actual values of a variable. Let A be a variable with a continuous value, its entropy H(A) is now in the form of an integration form,

$$H(A) = -\int_a p(a) \log p(a) da$$

Assume that A and B are two random variables; their joint entropy H(A, B) is

$$H(A, B) = -\Sigma\Sigma\, p(a, b) \log p(a, b)$$

Conditional entropy [12] is defined as the amount of remaining uncertainty present for one variable or parameter when another variable or parameter is known. Specifically, given the observing values of B, the conditional entropy H(A|B) of variable A with respect to variable B is

$$H(A|B) = -\Sigma\Sigma p(a, b) \log p(a|b)$$

If A relies entirely on B, then H(A) is zero. This implies that when B is understood, no more information is needed to explain A. Otherwise, if they are independent of each other, H(A|B) = H(A), Knowing that B will do nothing to observe A in this scenario.

Mutual information (MI) I(A, B) measures the amount of information shared between A and B parameters

It informs us how much data can be predicted about the other by one function. The higher the two features, the higher the MI. IF I(A; B)=0 means the two variables are completely irrelevant to one another.

$$I(A; Y) = H(A) - H(A|B)$$
$$= H(A)+H(B) - H(A, B)$$

If T = (D, F, C) is a dataset where D = { d1,d2 .. ,dn }, F = { f1, .., fn } and C = { c1c2, .., ck } are both data instances, features and class tags. Every instance di is represented as a combination of F and C value vectors. Based on the instances in D, the learning algorithm must map the input feature space F to the category space C. This means that each input function is relevant to the output class. For a subset S⊂F, this can preserve most of the information contained in the original space

$$S` = \text{argmax}_S I(S;C),$$

Maximizing the value of S is generally an NP-hard optimization scenario/problem as the various combinations of features grow exponentially high. With the assumption that

$$ft = \text{argmax}_{i\notin St} - I(xi;C) - [I(xi; x_{St-1}) - I(xi;x_{St-1}|C)]$$

Normalized mutual information

Mutual Information has a drawback due to its non-comparability between feature pairs that have different

mutual information values in different ranges. To solve this, MI is generalized to a closed array, say [0 1].Symmetric uncertainty in the form of the two uncertainty coefficients weighted average.

$$I(A; B) = \frac{2I(A;B)}{H(A)+H(B)}$$

Laplacian Score for Feature Selection (LSFS)

It selects some top-ranked features with a Laplacian score computed high locality preserving power. The theory is that there are likely to be two nearest data points in the same group. The underlying idea for this is to give more priority to the local data structure over the global structure.

III. CLUSTERING FOR FEATURE SELECTION

Feature selection must generate subsets that contain features highly relevant to the class labels and highly irrelevant (uncorrelated) with each other. The same is the case of the data clustering approach. It is an unsupervised learning algorithm that performs the grouping of the data into different groups (cluster) based on the principle that the members of a cluster are more similar in features to each other than to members in other clusters with respect to features.

In terms of similarity criteria (e.g., correlation coefficient [14], MI [11], and conditional MI [13]), this form of selection method is called feature clustering group characteristics in different clusters.

A subset of features is generated by choosing the most representative function in each cluster, the head of the cluster.

The Clustering distance can be denoted as Sb(C, P); for each cluster selected, P and C are denoted as the cluster for the class labels. Thus the mutual information is given as:

$$Sb(C, S) = \sum_{s \in S} I(s;C)$$

Their degree of relevance for features s and f is defined as the relative amount of s uncertainty reduction when f is known,

$$\text{i.e., } DR(s, f) = \frac{I(s;f)}{H(s)}$$

. Therefore, if s relies entirely on f, then

$$CR(s, f)=1.S(f) = \sum_{s \in S} CR(s; f).$$

Generally, K-means are used in clusters that can be used in FSFC. Difficulty in selecting the k value and the initial centroids, however, is an obvious deficiency of K-means.

In the proposed approach, a feature selection method based on clustering in a hierarchically agglomerative way is performed. Each feature is initially considered as a cluster (singleton), and the between the cluster and within the cluster distances are measured by mutual information (MI) gain and the coefficient of relevancy, respectively. Finally, the aggregated cluster is the result of the feature selection process. This cluster has minimal redundancy among its members and maximal relevancy with the class labels.

The class labels are considered to be a special supercluster cc in the given dataset. Features are grouped into a selected S cluster and a number of clusters of candidates. To separate the classes C, each feature in S has already been selected. There is only one function f in each candidate cluster, which has not yet been picked. In this way, the cluster, which is a candidate (feature) along with the selected cluster S and the total amount of cluster distance within, would be spread with the special cluster and will be combined with the cluster selected.

Finally, the selected cluster S is the final selected subset.

We can employ a pairwise comparison of the features for the Mutual Information gain with the class label and also for the intracluster distance and map them in a matrix as shown below with size n*m.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Sw(S) is the distance within the cluster that is initially assumed to be zero and to be accumulated with S(f) each time the candidate cluster f is merged with the selected one S,

$$Sw(S) = Sw(S) + S(f)$$

Where |S| is the number of elements in S.

IV. EXPERIMENTAL STUDY AND RESULTS

The method adopted here is, therefore, a hierarchical approach to clustering. As discussed earlier, SFS Sequential Forward Selection is the search strategy used. Every feature f in F is initialized as a cluster of candidates, and class labels C are taken as Cluster Cc. The inter-cluster distance Sb(C, f) with C will then be determined for each f. The candidate cluster f with the largest value. The method adopted here is, therefore, a hierarchical approach to clustering. As discussed earlier, SFS Sequential Forward Selection is the search strategy used.

For experimental purposes, the data sets are collected from the UCI machine learning repository.

We used the scikit-feature of the open-source selection feature in the repository. It is based on a scikitlearn system of commonly used machine learning and two Numpy and Scipy packages for scientific computing.

http:/featureselection.asu.edu/ offers several sources to run each algorithm, such as publicly available benchmark datasets, algorithm performance assessment, and test cases.

Table 1: Data set selected

Dataset	No. of Samples	No. of Features	No. of Classes
Colon 2	62	2000	2
Multiple Features	2000	649	10
Ionosphere	355	342	2
Optdigits	5620	64 10	10
Sonar	208	60	2

Confusion matrix

The confusion matrix is a very powerful visualization of the performance of a learning algorithm. A confusion matrix is a matrix that depicts the summary of results predicted from a classification model. The representation of the confusion matrix can be derived from a concept definition of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) terminology for multiple classes. Let Ci be the label of a class. The definitions for TP, FP, FN, and TN for Ci are as follows:

- True Positive TP(Ci) = The number of Ci-classified instances.
- False Positive FP(Ci) = The number of non-Ci instances classified as Ci.
- False Negative FN(Ci) = Number of Ci instances not known as Ci
- True Negative TN(Ci) = The number of non-Ci instances, not Ci.

Accuracy

Accuracy is depicting how correctly the classification is conducted.

Accuracy = (TP+TN)/(TP+FP+TN+FN)

Precision

Precision is a measure of the number of true positive elements produced by the template compared to the number of positive elements that it reports.

Precision= TP/(TP+FP)

Recall

The recall is the actual positive rate, meaning the number of positive claims in the model compared to the actual number of positive claims in the results.

Recall=TP/(TP+FN)

V. CONCLUSION AND FUTURE WORK

Feature selection is an important problem in machine learning for decision making. It enhances the quality of the data set under consideration and improves and assists the final learning and decision-making algorithms. In this paper, we have carried out a preliminary feature selection study and conducted feature selection algorithm performance evaluation based on similarity[15][17].

A hierarchical model was employed, which was supervised by the class information. An empirical study is conducted with 5 data sets of different types. The experimental results are based on proposed feature selection algorithms (it selects less features), and its effectiveness was measured with NaiveBayes, and Decision Tree supervised learning techniques.

In the future, we plan to propose unsupervised feature selection for IoT data sets. And measure its effectiveness by applying unsupervised algorithms.

F1 score

The output of a prototype can also be calculated by the F1 average. It is measured as the model reliability, and product recall weighted average

F1Score=2*TP 2*TP+FP+FN

Receiver operating characteristic curve (ROC Area)

It measures a classifier's performance across all possible thresholds. It is generated by plotting the True Positive Rate against the False Positive Rate. The True Positive Rate and False Positive Rate values range from 0 to 1 for each group.

Table2: Results Observed

DataSet	Precision		Recall		Fmeasure		ROC Area	
	NB	DT	NB	DT	NB	DT	NB	DT
1	0.694	0.704	0.735	0.711	0.712	0.707	0.936	0.831
2	0.211	0.812	0.973	0.932	0.977	0.831	0.988	0.961
3	0.231	0.931	0.676	0.662	0.877	0.731	0.888	0.892
4	0.962	0.932	0.873	0.602	0.917	0.731	0.978	0.811
5	0.781	0.894	0.891	0.786	0.895	0.89	0.774	0.894

Table 2 shows the statistical results in terms of the parameters[13] Precision, Recall, F-measure, ROC Area. Proposed feature selection was run on five data sets discussed in Table1. The selected features were provided as input for Naïve Bayes[8] and Decision Tree[11] supervised algorithms for classification prediction. Respective performance parameters were featured and tabulated.

REFERENCES

[1] Liu H, Yu L., Toward integrating feature selection algorithms for classification and clustering.,IEEE Trans Knowl Data Eng. 17(4)(2005) 491-502.

[2] Liu J, Ranka S, Kahveci T. Classification and feature selection algorithms for multi-class CGH data. Bioinformatics. 24(13)(2008) :i86.

[3] Song L, Smola A, Gretton A, Bedo J, Borgwardt K., Feature selection via dependence maximization. J Mach Learn Res. 13(1)(2012) 1393-1434.

[4] Mitra P, Murthy CA, Pal SK., Unsupervised feature selection using feature similarity., IEEE Trans Pattern AnalMach Intell.24(3)(2002) 301-312.

[5] Song F, Guo Z, Mei D, Feature selection using principal component analysis, International Conference on System Science, Engineering Design and Manufacturing Informatization; 2010; Yichang, China.

[6] Song Q, Ni J, Wang G., A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans Knowl Data Eng. 25(1)(2013) 1-14.

[7] Luo M, Nie F, Chang X, Yang Y, Hauptmann AG, Zheng Q, Adaptive unsupervised feature selection with structure regularization., IEEE Trans Neural Netw Learn Syst. 29(4) (2018) 944-956.

[8] Luo M, Chang X, Nie L, Yang Y, Hauptmann AG, Zheng Q, An adaptive semisupervised feature analysis for video semantic recognition, IEEE Trans Cybern. 48(2)(2018) 648-660.

[9] 29. Ali SI, Shahzad W, A feature subset selection method based on symmetric uncertainty and ant colony optimization, International Conference on Emerging Technologies; (2012).

[10] M. Y. Munirah, M. Rozlini, N. Wahid, A comparative analysis on feature selection techniques for medical datasets, APRN Journal of Engineering and Applied Sciences, 11(22) (2016).

[11] 12. S. Kashef, H. Nezamabadi-pour, An Advanced ACO Algorithm for Feature Subset Selection, Neurocomputing 147(2015) 271279.

[12] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature Selection Algorithm based on Bare Bones Particle Swarm Optimization, Neurocomputing, 148(2015) 150-157.

[13] Y. Shen-Lan, R. Gang, F. Yi-Ping, Multiple kernel learning-based feature selection for process monitoring, IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 24-16(2017) 809-814.

[14] B. Emel, S. Mustafa, Video classification based on ConvNet collaboration and feature selection, 25th Signal Processing and Communications Applications Conference (SIU), 15-18 (2017) 1-4.

[15] Vandana C.P, Ajeet A. Chikkamannur, Study of Resource Discovery trends in Internet of Things, Int. J. Advanced Networking and Applications, 08(03)(2016) ISSN: 0975-0290 3084-3089.

[16] Amit Sagu, Nasib Singh Gill, Preeti Gulia Artificial Neural Network for the Internet of Things Security International Journal of Engineering Trends and Technology 68.11(2020):129-136.

[17] Vandana C.P, Ajeet A. Chikkamannur, Semantic Ontology-Based IoT-Resource Description, Int. J. Advanced Networking and Applications, 11(01)(2019) 4184-4189 ISSN: 0975-0290 10.35444/IJANA.2019.11018