CHAPTER 3

FEATURE SELECTION

3.1 INTRODUCTION

Feature selection is the first technique that is applied on the test suite in test case prioritization. The objective function formed using the objectives and the test suite are given as the input to this feature selection technique. The test cases which satisfy the objective functions are selected from the test suite. Mainly, test case reduction is performed in feature selection using mRMR feature selection algorithm.

3.2 FEATURE

A feature refers to an aspect of the data. Usually, before collecting data, features are specified or chosen. Features are characterized as relevant, irrelevant, redundant, discrete, continuous or nominal.

The features that are related to the given objective and have an influence on the output are known as relevant features, while the rest cannot perform their role. If the features do not have any influence on the output they are irrelevant features. If there are features with duplication that is a feature that takes the role of another is known as redundant and even the feature may be distinct or incessant.
3.3 FEATURE SELECTION

Feature selection is a search problem for finding an optimal or suboptimal subset of $x$ features out of original $X$ features and is also an important technique in data mining and is effective in reducing dimensionality, removing irrelevant data, increasing accuracy, improving performance and result as a pre processing step as shown in Figure 3.1. Feature selection is mainly used to obtain the same or better results with the minimum number of features.

![Figure 3.1 Feature selection](image)

The goal of feature selection is to find a best subset of input features that contains the least number of dimensions that most contribute to accuracy by removing unimportant dimensions that are the most irrelevant and redundant features from the data. A better understanding of data and the important features of data and the relationship between the features can be clearly obtained from feature selection.

In the case of statistics, stepwise regression is the most popular form of feature selection. Greedy algorithm is used to add the best feature or delete the worst feature at each round. The main control issue here is making the decision about the stopping criterion. Also in statistics, there is a problem of nesting since only some criteria are optimized. More robust methods have been explored, such as branch and bound and piecewise linear network.

In machine learning, this complication is rectified by cross-validation. It solves the problem of selecting subset of a input variables, while
ignoring the irrelevant variables. Feature selection (also known as subset selection) is a process commonly used in machine learning, is the process of selecting the best feature among all the features since all the features are not useful in constructing the clusters and some features may be redundant or irrelevant thus not contributing to the learning process.

In many real world problems, feature selection is a must due to the abundance of noisy, irrelevant or misleading features. Many algorithms, approaches and methods are available for feature selection.

### 3.4 FEATURE SELECTION ALGORITHMS

Initially, developed feature selection algorithms are easy, comfortable and informal. The development in technology brought many changes in feature selection also. Many methodical approaches were developed. Theoretically speaking, optimal feature selection requires an exhaustive search of all possible subsets of features of the chosen criteria. But an exhaustive search of all possible subsets of features is really an impractical method if there are a very large number of features are available. So, in practicality it is preferred to have a search for a satisfactory set of features instead of having an optimal feature selection set.

#### 3.4.1 Classification of Feature Selection Algorithms

The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. Feature selection algorithms can be classified into three broad classes. They are

- Filter method
- Wrapper method
- Embedded method.
3.4.1.1 Filter method

Filter methods are used to filter the features based on discriminating criteria. Filtering is done usually based on the characteristics of data and independent of any classification algorithm (Ding and Peng 2005, Yu and Liu 2004) is shown in Figure 3.2.

![Filter method diagram](image)

**Figure 3.2 Filter method**

The features are given as the input to the filter. The filter filters out unreliable features from a given features set for effective prediction using a criteria. The predictor uses this features subset for further prediction. In filters, the features are selected independently of the performances of the learning machine, using a criterion.

The earlier filter-based methods are simple and they evaluated features in isolation and did not consider correlation between features. There are many variations in the earlier filter based methods like simple correlation coefficients similar to Fisher's discriminant criterion (Golub et al 1999, Pavlidis et al 2001), mutual information (Ding and Peng 2005) and statistical tests including t-test and F-test (Ding 2002, Model et al 2001). Later the methods are improved and methods have been proposed to select features with minimum redundancy (Ding and Peng 2005, Yu and Liu 2004).
The method proposed uses a minimum redundancy maximum relevance (mRMR) feature selection approach. The main advantage of this method is that the maximum relevance criteria along with minimum redundancy criteria is used to choose features that are maximally relevant to the criteria and minimally redundant with respect to the criteria. Using mRMR the representative power of the feature set can be expanded and their generalization properties can be improved.

3.4.1.2 Wrapper method

Wrapper methods search through the space of possible features (n feature sets) by using a search algorithm and evaluate each subset by running a model on the subset is shown in Figure 3.3. For this evaluation wrapper methods use the classifier as a black box to score the subsets of features based on their predictive power.

Wrappers usually select feature subsets on the basis of how well a learning machine performs. Wrapper methods are usually based on Support Vector Machines (SVM) have been widely studied in machine-learning (Guyon and Elisseeff 2003, Rakotomamonjy 2003, Weston et al 2000). Another wrapper method which is the variation of SVM that is SVM-RFE (Support Vector Machine Recursive Feature Elimination) (Guyon et al 2002), uses a backward feature elimination scheme to recursively remove insignificant features from subsets of features.
3.4.1.3 Embedded method

In order to combine the advantages of both methods, the embedded method has been proposed and is shown in Figure 3.4. Embedded techniques are embedded in the model and they are specific to a model.
With this model, some good subsets of features will be selected from a high-dimensional data set by using the filter model. The wrapper model will then be applied on those good subsets to get the best one.

These methods may use either feature ranking or subset selection method during its feature selection. Feature ranking ranks the features according to a criterion and eliminates all the other features that do not satisfy the specified criterion. Subset selection searches the set of possible features and selects the features that satisfies the criterion and forms the optimal subset.

### 3.4.2 Approaches in Feature Selection

There are two approaches in Feature selection known as

- Forward selection
- Backward selection.

Forward selection is the simplest data-driven model building approach, in which the variables are added to the model one by one. At each
step, the variables that are not already in the model are tested one by one for inclusion in the model.

Forward selection has a serious drawback that including a new variable may make one or more of the already included variables as non-significant. This disadvantage can be overcome by an alternate approach known as backward selection. This approach starts with fitting a model with all the variables of interest and then the least significant variable that is not significant at the critical level is dropped. The process is continued successively by applying the same rule until all remaining variables are statistically significant. The main aim of the elimination process is to reduce the size of the input feature set and at the same time it also tries to retain the class based information.

3.5 MINIMUM-REDUNDANCY -MAXIMUM-RELEVANCE (mRMR) FEATURE SELECTION

Many feature selection algorithms have been developed in the past and each of the feature selection algorithms was with a clear objective of improving classification accuracy while reducing the dimensionality. The relevance of a feature to the objective is the most important selection criterion because using highly relevant feature improves the accuracy of the system and also the equal concentration should be given for the selected features need to be non-redundant.

So, the minimum-Redundancy-Maximum-Relevance (mRMR) feature or variable or attribute selection method is used here for feature selection. It uses mutual information to analyze relevance and redundancy. In mRMR, the feature that has the maximum relevance measure and minimum redundancy measure with the already selected features is selected and added
to the subset. The mRMR (Hanchuan Peng et al 2005) is the scheme in feature selection is to select the features that correlate the strongest with a classification variable. The selection scheme of mRMR having a high correlation is obtained by combining this scheme with selection features that are mutually different from each other. Using mRMR a feature subset is selected in such a way that best characterizes the statistical property of a target classification variable, subject to the constraint that these features are mutually as dissimilar to each other as possible, but marginally as similar to the classification variable as possible.

### 3.5.1 Feature Selection Problem

The feature selection problem (Hanchuan Peng et al 2005) can be defined as Given the input data D tabled as N samples and M features \(X = \{x_1, x_2, x_3, \ldots, x_m\}\) and the target classification variable \(c\), the feature selection problem is to find from the M-dimensional observation space, \(R^M\), a subspace of \(m\) features, \(R^m\), that “optimally” characterizes \(c\).

The reduction often needs to have the minimal classification error and at the same time it should satisfy the optimal characterization condition. In an unsupervised situation, minimal error usually requires both the maximal statistical dependency of the target class \(c\) on the data distribution in the subspace \(R^m\) and similarly the maximal statistical dependency of the data distribution in the subspace \(R^m\) on the target class \(c\). In this case the classifiers are not specified. This is maximal dependency (Max-Dependency) and the most popular approach to realize Max-Dependency is maximal relevance (Max-Relevance) feature selection.
3.5.1.1 Maximal relevance

Maximal relevance feature selection is an algorithm frequently used in its pairing with minimum redundancy feature selection as minimum Redundancy Maximum Relevance (mRMR). It identifies subsets of data that are relevant to the parameters used and is normally called Maximum Relevance.

In Max-Relevance, the selected features $x_i$ are required, individually, to have the largest mutual information $I(x_i, c)$ with the target class $c$, reflecting the largest dependency on the target class. In terms of sequential search, the $m$ best individual features, i.e., the top $m$ features in the descent ordering of $I(x_i, c)$, are often selected as the $m$ features.

In terms of mutual information, the purpose of feature selection is to find a feature set $S$ with $m$ features $\{x_i\}$, which jointly have the largest dependency on the target class $c$. This scheme, called Max-Dependency, has the following form:

$$\max D(S, c), \quad D = I(\{x_i, i = 1, \ldots, m\}, c)$$ (3.1)

Max-Relevance (Hanchuan Peng et al 2005) is to search features satisfying Equation (3.2), which approximates $D(S, c)$ in Equation (3.1) with the mean value of all mutual information values between individual feature $x_i$ and class $c$.

$$\max D(S, c), \quad D = \frac{1}{|S|} \sum_{x_i \in S} I(X_i, c)$$ (3.2)

Maximal relevance (Max-Relevance) feature selection is selecting the features with the highest relevance to the target class $c$. Either correlation
or mutual information may be the measure used to define dependency of variables.

Relevance with respect to an objective is defined as a feature \( x_i \in X \) is relevant to an objective \( c \) "if there exist two examples \( A, B \) in the instance space \( E \) such that \( A \) and \( B \) differ only in their assignment to \( x_i \) and \( c(A) \neq c(B) \).

### 3.5.1.2 Minimal redundancy

Minimum redundancy feature selection is an algorithm frequently used in a method to accurately identify characteristics of genes and phenotypes and narrow down their relevance and is usually described in its pairing with relevant feature selection as minimum Redundancy Maximum Relevance (mRMR). The subsets that are identified by the maximum relevance are often contain material which is relevant but redundant and mRMR attempts to address this problem by removing those redundant subsets.

The minimal redundancy (Min-Redundancy) condition (Hanchuan peng 2005) can be added to select mutually exclusive features Equation (3.3)

\[
\min R(s) = \frac{1}{|s|^2} \sum_{i \neq j, i \in s} I(X_i, X_j) \tag{3.3}
\]

### 3.5.1.3 Combining minimal redundancy and maximal relevance

The criterion combining the above two constraints is called “minimal-redundancy-maximal-relevance” (mRMR). Minimum Redundancy Maximum Relevance are combined to select features that are mutually far away from each other at the same time having high association with the criterion.
The operator $\phi(D,R)$ is defined as in Equation (3.4) to combine $D$ and $R$ and consider the following simplest form to optimize $D$ and $R$ simultaneously,

$$\max \phi(D,R), \phi = D - R$$

(3.4)

Here as a special case, the "correlation" can be replaced by the statistical dependency between variables. Mutual information can be used to quantify the dependency.

### 3.5.1.4 Mutual information

Mutual information is a basic concept in information theory. The mutual information (MI) of two random variables is a quantity that measures the mutual dependence of the two random variables. The most common unit of measurement of mutual information is the bit, when logarithms to the base 2 are used. The approach is based on the mutual information concept. MI measures the general dependence of random variables without making any assumptions about the nature of their underlying relationships. Consequently, MI can potentially offer some advantages over feature selection techniques that focus only on the linear relationships of variables.

The MI criterion offers some advantages over the other techniques. The MI measures general statistical dependence between variables while compared to the other methods measure linear correlation coefficient. The MI is invariant to monotonic transformations performed on the variables, contrary to linear dimension reducers such as principal component analysis. The MI feature selection approach is independent of the decision algorithm, thus reducing computational complexity.
Mutual information can be calculated either by using entropy or by using probability.

### 3.5.1.4.1 Mutual information using entropy

Mutual information is calculated using entropy as Equation (3.5) if the mutual information has to be calculated for discrete random variables and it is calculated using entropy as Equation (3.7). if the mutual information has to be calculated for continuous random variables, it is calculated using entropy as Equation (3.6).

Given two random variables $X$ and $Y$, the mutual information $I(X;Y)$ is defined as follows:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

(3.5)

$H( )$ is the entropy of a random variable and measures the uncertainty associated with it. For a continuous random variable $X$, $H(X)$ is defined as

$$H(X) = - \int p(x) \log_2 p(x) \, dx$$

(3.6)

If $X$ is a discrete random variable, $H(X)$ is defined as follows:

$$H(X) = - \sum p(x) \log_2 p(x)$$

(3.7)

In both cases $p(X)$ represents the marginal probability distribution of random variable $X$. The entropy is often considered a measure of uncertainty. So, the calculation of mutual information using entropy is done when there is uncertainty.
3.5.1.4.2 Mutual information using probability

Mutual information is also calculated using probability as Equation (3.8) if the mutual information has to be calculated for discrete random variables and it is calculated using probability as Equation (3.9) if the mutual information has to be calculated for continuous random variables.

Mutual Information (MI) measures the information contributed by the presence/absence of a term in making the correct classification decision on the target class c. Given two discrete random variables x and y, their mutual information is defined in terms of their probabilistic density functions p(x), p(y), and p(x,y)

\[
I(x, y) = \sum \sum p(x, y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3.8}
\]

Given two continuous random variables, their mutual information is defined in terms of their probabilistic density functions p(x), p(y), and p(x,y)

\[
I(x, y) = \int_y \int_x p(x, y) \log \frac{p(x,y)}{p(x)p(y)} \, dx \, dy \tag{3.9}
\]

where \( p(x,y) \) is the joint probability distribution function of X and Y, and \( p(x) \) and \( p(y) \) are the marginal probability distribution functions of X and Y respectively. The mRMR uses the mutual information to select best m features that have minimal redundancy and maximal relevance criterion.
3.6 TEST CASE REDUCTION USING MINIMUM-REDUNDANCY -MAXIMUM-RELEVANCE (mRMR) FEATURE SELECTION

The minimum Redundancy-Maximum Relevance (mRMR) approach is used to maximize the joint dependency of top ranking objectives on the target test suite, at the same time the redundancy among the test cases must be reduced.

This is done by incrementally selecting the maximally relevant test cases from the test suite with respect to the objectives while avoiding the redundant ones.

For this initially, the mutual information (MI) between the test cases and the objective is calculated (the relevance term). Then the average MI between the test cases and the remaining test cases that are already selected is computed (the redundancy term).

3.6.1 Definition of mRMR

The general feature selection problem defined for “minimal-redundancy-maximal-relevance” (mRMR) is modified according to test case reduction problem and is defined as

Given the input data D tabled as T test cases and X features (objectives) such as \{x_1, x_2, x_3, x_4\} and the target classification variable c, the feature selection problem is to find from the 4-dimensional observation space, \(R^M\), a subspace of x features, \(R^x\), that “optimally” characterizes c.

Here the dimension is specified as 4 since there are 4 features (objectives). The reduction often needs to have the minimal error and at the same time it should satisfy the optimal categorization condition. Minimal
error usually requires both the maximal statistical dependency of the target class c on the data distribution in the subspace \( R^x \) and similarly the maximal statistical dependency of the data distribution in the subspace \( R^x \) on the target class c. In this case the classifiers are not specified. This is maximal dependency (Max-Dependency) and the most popular approach to realize Max-Dependency is maximal relevance (Max-Relevance) feature selection.

The mRMR feature selection algorithm for test case reduction is given as

1) (Initialization): set \( X \leftarrow \) initial set of 4 objectives, \( X = \{ x_1, x_2, x_3, x_4 \} \),

set \( T \leftarrow \) initial set of test cases, \( T = \{ t_1, t_2, t_3, \ldots t_n \} \) and

Initial test case set \( S \leftarrow \{ \} \) and \( R \leftarrow \{ \} \)

2) (Max-Relevance)

Compute \( I(x_i, t_i) \) as in (3.10) for all \( t_i \in T \) and \( x_i \in X \)

a) select the test case \( t_i \) that maximizes \( I(x_i, t_i) \) and add \( t_i \) to \( S \), \( S \leftarrow \{ t_i \} \)

b) repeat for all \( t_i \in T \)

c) repeat for all \( x_i \in X \)

3) (Min-Redundancy)

Compute \( I(s_i, s_j) \) as in (3.11) for all \( s_i, s_j \in S \)

a) select the test case \( s_i \) that maximizes \( I(s_i, s_j) \)

b) compare \( s_i \) and \( s_j \) in terms of \( I(x_i, s_i), I(x_i, s_j) \)

c) select the \( s_i \) or \( s_j \) with max I and add \( R \leftarrow \{ s_i \} \)
d) repeat for all $s_i \in S$

e) Repeat for all $x_i \in X$

4) Output the mRMR reduced test cases.

### 3.6.2 Stages in mRMR Feature Selection

After forming the objective functions as specified in Section (1.4.1), the test suite is reduced using the mRMR feature selection algorithm. The stages in feature selection are Input, Evaluator and output.

#### 3.6.2.1 Input stage

In the input stage, the test suite $T$ and the objective set $X$ are given to the Evaluator. The objective set $X$ has 4 objectives such as memory usage, execution time, completeness and fault. In mRMR feature selection the function $\text{fsMRRM}(X, T, \text{opfn})$ is used to select the test cases based on the criterion on each step, where $X$ is the set of objectives, $T$ is the test suite and $\text{opfn}$ is the function used for optimization.

![Testcase Reduction](image)

**Figure 3.5 Test case reduction using feature selection**
The test cases with their values specified for memory used, execution time, coverage value and faults are taken as input for feature selection module. The initial screen of feature selection is shown in Figure 3.5.

Each test case and the related information stored inside the notepad are separated by “;”. The function readTestCases() is used for reading the test cases from the notepad. While reading the test cases from the notepad, the symbol “;” indicates the fore coming test case is a new test case.

3.6.2.2 Evaluator stage

The evaluator stage in feature selection uses two techniques.

3.6.2.2.1 The first technique: max-relevance

The first technique Max-Relevance is used to select the test cases with relevant objectives such as maximum fault coverage, minimum execution time, maximum completeness and minimum memory usage from the test suite. The test cases with most relevant features to the objectives that are specified in objective function are selected from the test suite by the technique Maximum Relevance.

For finding the Max-Relevance for test case reduction Equation (3.2) is modified as Equation (3.10) and Equation (3.3) is modified as Equation (3.11).

Max-Relevance (Hanchuan Peng et al 2005) is to search test case with features satisfying Equation (3.10), with the mean value of all mutual information values between individual feature $x_i$ and test suite $T$. 
\[
\max D(X,T), D = \frac{1}{|x|} \sum_{x_i \in x} I(x_i T)
\] 

(3.10)

This first technique acts as an irrelevancy filter, which selects the subset of relevant test cases by removing irrelevant ones. The technique considers the value of the objectives that is available with each test case in the test suite. Then the test cases are ranked based on their values of objectives. The ranking is done based on the importance of the objective to the system. In this system the four objectives are ranked as fault coverage, execution time, completeness and memory usage based on their importance to the test case prioritization system.

The objective value is calculated based on MI between the test cases and the objectives using Equation (3.8). Here calculateMutualInformation function is used for calculating the MI between the test cases and the objectives.

\[P(x_i, t_j)\] is calculated as frequency of both \( x_i \) and \( t_j \) / no of transactions \[P(x_i)\] is calculated as frequency of \( P(x_i) \) / no of transactions and \[P(t_i)\] is calculated as frequency of \( P(t_i) \) / no of transactions

\( t_j \) indicates the test cases in the test suite.

The MI is calculated between each test case and the first objective. Higher MI value between the test case and the objective means more common information content between them. If MI between two the test case and the objective larger than a threshold, denoted by relevancy threshold or RLTH, these two that is the test case and the objective are considered relevant to each other.
Figure 3.6 The mRMR feature selection
The test cases with the RLTH value larger than a threshold, denoted by irrelevancy threshold, are selected and the other test cases are filtered out. This threshold is determined by Equation (1.2).

Figure 3.6 clearly details the stages of mRMR feature selecture algorithm.

3.6.2.2.2 The second technique: min-redundancy

The output from the first technique often contains test cases which are relevant but redundant and mRMR attempts to address this problem by removing those redundant test cases by using the second technique Min-Redundancy.

The minimal redundancy (Min-Redundancy) condition (Hanchuan peng 2005) can be added to select mutually exclusive features using Equation (3.11)

$$\min R(x) = \frac{1}{|x|^2} \sum_{x_i,x_j \in x} I(x_i,x_j)$$  \hspace{1cm} (3.11)

Minimum Redundancy Maximum Relevance (mRMR) feature selection algorithm which is based on mutual-information is used to select test cases from the test suite T, which satisfies Equation (3.4)

The approach utilizes the mutual information concept as the feature selection criterion. By definition, MI measures the information content of a given feature with regard to the decision task at hand.

The selected test cases by the irrelevancy filter may be redundant. The second technique of the proposed feature selection technique filters out redundant test cases from the selected subset of test cases by the first
technique using the irrelevancy filter. The redundant test cases among the selected subset of the first technique are filtered out by the MI technique. Hence, the outcome of the proposed feature selection technique is the most relevant features with minimum redundancy.

The redundancy filter uses the concept of MI technique which is used to measure common information between test cases. Using Equation (3.8) MI is calculated between pair of test cases. Here calculateMutualInformation function calculates the Mutual Information I(X;Y) between two random variables X and Y. In this case it is used for calculating the MI between the test cases.

\[ P(x_i, x_j) \] is calculated as frequency of both \( x_i \) and \( x_j \) / no of transactions

\[ P(x_i) \] is calculated as frequency of \( P(x_i) \) / no of transactions and

\[ P(y_i) \] is calculated as frequency of \( P(y_i) \) / no of transactions

Higher MI value of two test cases means more common information content between them. If MI between two tests cases larger than a threshold, denoted by redundancy threshold or RTH, these two test cases are considered redundant test cases.

If two test cases are redundant, the less relevant one with lower relevance weight value, obtained from the first stage is eliminated and the more relevant one with higher weight value is retained. This process is repeated until no redundant test case is found.

3.6.2.3 Output stage

The selected test cases by the irrelevancy filter are the final result of the proposed feature selection technique. These test cases are considered as the inputs of the next level birds flocking algorithm.
If the feature selection button is pressed in the input stage, then the mRMR feature selection algorithm is used to reduce the input test suite and the reduced test cases set is given as output as shown in Figure 3.7.

![Figure 3.7 Reduced test cases](image)

The test cases are reduced based on the objective function specified. The test cases from feature selection are given as the input to birds flocking.

### 3.7 SUMMARY

Using mRMR feature selection scheme presented above the test suite is reduced based on the objective function. But still the effectiveness of the test suite may be increased by optimizing the test suite. So the optimization of the test suite is done in the next chapter.