FEATURE SELECTION METHODS AND ALGORITHMS

L.Ladha, Research Scholar,

Department Of Computer Science,
Sri Ramakrishna College Of Arts and Science for Women,
Coimbatore, Tamilnadu, India.

T.Deepa, Lecturer,

Department Of Computer Science,
Sri Ramakrishna College Of Arts and Science for Women,
Coimbatore, Tamilnadu, India.

Abstract— Feature selection is an important topic in data mining, especially for high dimensional datasets. Feature selection (also known as subset selection) is a process commonly used in machine learning, wherein subsets of the features available from the data are selected for application of a learning algorithm. The best subset contains the least number of dimensions that most contribute to accuracy; we discard the remaining, unimportant dimensions. This is an important stage of preprocessing and is one of two ways of avoiding the curse of dimensionality (the other is feature extraction). There are two approaches in Feature selection known as Forward selection and backward selection. Feature selection has been an active research area in pattern recognition, statistics, and data mining communities.

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 methods can be decomposed into three broad classes. One is Filter methods and another one is Wrapper method and the third one is Embedded method. This paper presents an empirical comparison of feature selection methods and its algorithms. In view of the substantial number of existing feature selection algorithms, the need arises to count on criteria that enable to adequately decide which algorithm to use in certain situations. This work reviews several fundamental algorithms found in the literature and assesses their performance in a controlled scenario.

Keywords- Feature Selection, Feature Selection Methods, Feature Selection Algorithms.

1. INTRODUCTION

1.1 Feature selection Definition:

A "feature" or "attribute" or "variable" refers to an aspect of the data. Usually before collecting data, features are specified or chosen. Features can be discrete, continuous, or nominal. Generally, features are characterized as:

1. Relevant: These are features which have an influence on the output and their role can not be assumed by the rest.

2. Irrelevant: Irrelevant features are defined as those features not having any influence on the output, and whose values are generated at random for each example.

3. Redundant: A redundancy exists whenever a feature can take the role of another (perhaps the simplest way to model redundancy).

Problem of selecting some subset of a learning algorithms input variables upon which it should focus attention, while ignoring the rest. Feature selection is the process of selecting the best feature among all the features because all the features are not useful in constructing the clusters: some features may be redundant or irrelevant thus not contributing to the learning process.

Feature selection (also known as subset selection) is a process commonly used in machine learning, wherein a subset of the features available from the data are selected for application of a learning algorithm. The best subset contains the least number of dimensions that most contribute to accuracy; we discard the remaining, unimportant dimensions. This is an important stage of preprocessing and is one of two ways of avoiding the curse of dimensionality (the other is feature extraction). The main aim of feature selection is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In many real world problems Feature selection is a must due to the abundance of noisy, irrelevant or misleading features. For instance, by removing these factors, learning from data techniques can benefit. To be completely sure of the attribute selection, we would ideally have to test all the enumerations of attribute subsets, which is infeasible in most cases as it will result in 2n subsets of n attributes. Feature selection has been an active research area in pattern recognition, statistics, and data mining communities.

1.2 Advantages of feature selection:

- It reduces the dimensionality of the feature space, to limit storage requirements and increase algorithm speed;
- It removes the redundant, irrelevant or noisy data.
- The immediate effects for data analysis tasks are speeding up the running time of the learning algorithms.
- Improving the data quality.
- Increasing the accuracy of the resulting model.
- Feature set reduction, to save resources in the next round of data collection or during utilization;
- Performance improvement, to gain in predictive accuracy;
- Data understanding, to gain knowledge about the process that generated the data or simply visualize the data

2. ALGORITHMS FOR FEATURE SELECTION (FSA)

A feature selection algorithm (FSA) is a computational solution that is motivated by a certain definition of relevance. The purpose of a FSA is to identify relevant features according to a definition of relevance.

2.1 Characterization of FSAs:

There exist in the literature several considerations to characterize feature selection algorithms. In view of them it is possible to describe this characterization as a search problem in the hypothesis space as follows:

2.1.1 Search Organization:

General strategy with which the space of hypothesis is explored. This strategy is in relation to the portion of hypothesis explored with respect to their total number. A search algorithm is responsible for driving the feature selection process using a specific strategy. We consider three types of search: exponential, sequential and random.

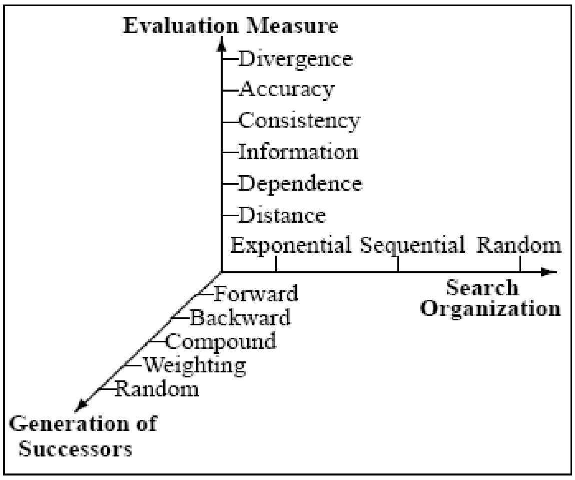
2.1.2 Generation of Successors:

Mechanism by which possible variants (successor candidates) of the current hypothesis are proposed. Up to five different operators can be considered to generate a successor for each state: Forward, Backward, Compound, Weighting, and Random.

2.1.3 Evaluation Measure:

Function by which successor candidates are evaluated, allowing to compare different hypothesis to guide the search process. Some of the evaluation measures are Probability of error, Divergence, Dependence, interclass distance, Information or Uncertainty and consistency.

Characterization of a FSA.



2.2 APPROACHES:

There are two approaches in Feature selection:

1. Forward Selection: Start with no variables and add them one by one, at eachstep adding the one that decreases the error the most, until any furtheraddition does not significantly decrease the error.

2. Backward Selection: Start with all the variables and remove them one by one, at each step removing the one that decreases the error the most (or increases it only slightly), until any further removal increases the error significantly. To reduce over fitting, the error referred to above is the error on a validation set that is distinct from the training set.

3. GENERAL ALGORITHM FOR FEATURE SELECTION

The basic feature selection algorithm is shown in the following.

```
Input:
     S - data sample with features X,|X| = n
     J  - evaluation measure to be maximized
    GS – successor generation operator
Output:
    Solution – (weighted) feature subset
    L := Start_Point(X);
    Solution := { best of L according to J };
repeat
    L := Search_Strategy (L,GS(J),X);
    X’ := {best of L according to J };
    if J(X’)≥J(Solution) or (J(X’)=J(Solution) and |X’| < |Solution|) then Solution :=X’;
until Stop(J,L).
```

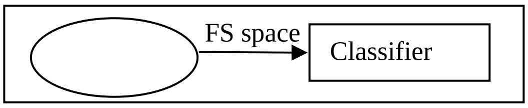
4. GENERAL SCHEMES FOR FEATURE SELECTION

The relationship between a FSA and the inducer chosen to evaluate the usefulness of the feature selection process can take three main forms: Filter,Wrapper and Embedded..

4.1 Filter Methods:

These methods select features based on discriminating criteria that are relatively independent of classification. Several methods use simple correlation coefficients similar to Fisher’s discriminant criterion. Others adopt mutual information or statistical tests (t-test, F-test). Earlier filter-based methods evaluated features in isolation and did not consider correlation between features. Recently, methods have been proposed to select features with minimum redundancy. The methods proposed use a minimum redundancy-maximum relevance (MRMR) feature selection framework. They supplement the maximum relevance criteria along with minimum redundancy criteria to choose additional features that are maximally dissimilar to already identified ones. By doing this, MRMR expands the representative power of the feature set and improves their generalization properties.

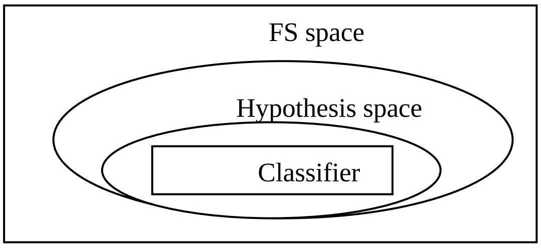
Filter Methods



4. 2 Wrapper Methods:

Wrapper methods utilize the classifier as a black box to score the subsets of features based on their predictive power. Wrapper methods based on SVM have been widely studied in machine-learning community. SVM-RFE (Support Vector Machine Recursive Feature Elimination), a wrapper method applied to cancer research is called, uses a backward feature elimination scheme to recursively remove insignificant features from subsets of features. In each recursive step, it ranks the features based on the amount of reduction in the objective function. It then eliminates the bottom ranked feature from the results. A number of variants also use the same backward feature elimination scheme and linear kernel.

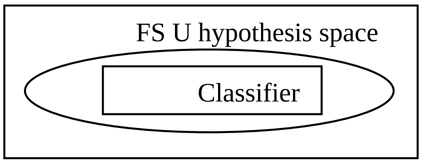
Wrapper Methods



4.3 Embedded Scheme:

The inducer has its own FSA (either explicit or implicit). The methods to induce logical conjunctions provide an example of this embedding. Other traditional machine learning tools like decision trees or artificial neural networks are included in this scheme.

Embedded Scheme



5. DESCRIPTION OF FUNDAMENTAL FSAS

Description for the basic Feature Selection algorithms are as follows:

5.1 CHI (χ 2 STATISTIC):

This method measure the lack of independence between a term and the category. Chi-Squared is the common statistical test that measures divergence from the distribution expected if one assumes the feature occurrence is actually independent of the class value. As a statistical test, it is known to behave erratically for very small expected counts, which are common in text classification both because of having rarely occurring word features, and sometimes because of having few positive training examples for a concept. In statistics, the $\chi 2$ test is applied to test the independence of two events,where two eventsA and B are defined to be *independent* if $P(AB) = P(A)P(B)$ or, equivalently, $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In feature selection, the two events are occurrence of the term and occurrence of the class. Feature selection using the χ2 statistic is analogous to performing a hypothesis test on the distribution of the class as it relates to the values of the feature in question. The null hypothesis is that there is no correlation; each value is as likely to have instances in any one class as any other class. Under the null hypothesis, if p of the instances have a given value and q of the instances are in a specific class, (p · q)/n instances have a given value and are in a specific class (n is the total number of instances in the dataset). This is because p/n instances have the value and q/n instances are in the class, and if the probabilities are independent (i.e.the null hypothesis) their joint probability is their product. Given the null hypothesis, the χ2 statistic measures how far away the actual value is from the expected value:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}.$$

In this equation, r is the number of different values of the feature in question, c is the number of classes in question (in this work, c = 2), Oi,j is the number of instances with value i which are in class j, and Ei,j is the expected number of instances with value i and class j, based on (p·q)/n. The larger this chi-squared statistic, the more unlikely it is that the distribution of values and classes are independent; that is, they are related, and the feature in question is relevant to the class.

5.2 EUCLIDIAN DISTANCE:

Euclidean Distance is the most common use of distance. In most cases when people said about distance , they will refer to Euclidean distance. Euclidean distance or simply 'distance' examines the *root of square differences* between coordinates of a pair of objects. For each feature *Xi* calculate Euclidean distance from it to all other features in sample. Euclidean distance $d(X_i; Y_i)$ between features X_i and Y_i is calculated using the formula: **distance(x,y) = {Σi (xi - yi)2 }½**

Note that Euclidean (and squared Euclidean) distances are usually computed from raw data, and not from standardized data. This method has certain advantages (e.g., the distance between any two objects is not affected by the addition of new objects to the analysis, which may be outliers). However, the distances can be greatly affected by differences in scale among the dimensions from which the distances are computed. For example, if one of the dimensions denotes a measured length in centimeters, and you then convert it to millimeters (by multiplying the values by 10), the resulting Euclidean or squared Euclidean distances (computed from multiple dimensions) can be greatly affected, and consequently, the results of cluster analyses may be very different.

Squared Euclidean distance. One may want to square the standard Euclidean distance in order to place progressively greater weight on objects that are further apart. This distance is computed as (see also the note in the previous paragraph): **distance(x,y) = Σi (xi - yi)2**

5.3 T-TEST:

The t-test assesses whether the means of two groups are statistically different from each other. This analysis is appropriate whenever you want to compare the means of two groups, and especially appropriate as the analysis for the posttest-only two-group randomized experimental design. The formula for the t-test is a ratio. The top part of the ratio is just the difference between the two means or averages. The bottom part is a measure of the variability or dispersion of the scores. The formula for the t-test is shown in the following Figure.

$$T = \frac{X' - Y'}{\sqrt{\frac{S^2_x}{n1} + \frac{S^2_y}{n2}}}$$

5.4 INFORMATION GAIN (IG):

Information gain, of a term measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document. Information Gain measures the decrease in entropy when the feature is given vs absent. This is the application of a more general technique, the measurement of informational entropy, to the problem of deciding how important a given feature is. Informational entropy, when measured using Shannon entropy, is notionally the number of bits of data it would take to encode a given piece of information. The more space a piece of information takes to encode, the more entropy it has. Intuitively, this makes sense because a random string has maximum entropy and cannot be compressed, while a highly ordered string can be written with a brief description of the string's information. In the context of classification, the distribution of instances among classes is the information in question. If the instances are randomly assigned among the classes, the number of bits necessary to encode this class distribution is high, because each instance would need to be enumerated. On the other hand, if all the instances are in a single class, the entropy would be lower, because the bit-string would simply say "All instances save for these few are in the first class." Therefore a function measuring entropy must increase when the class distribution gets more spread out and be able to be applied recursively to permit finding the entropy of subsets of the data. The following formula satisfies both of these requirements: **$H(D) = - \Sigma\ (n_i/n) \log(n_i/n)$ i=1,…l** where dataset D has n =| D | instances and ni members in class c_i, i = 1, . . . , l. The entropy of any subset is calculated as: **$H(D|X) = - \Sigma\ (|X_j|/n)H(D|X\text{-}X_j)$** where H(D|X = Xj) is the entropy calculated relative to the subset of instances that have a value of Xj for attribute X. If X is a good description of the class, each value of that feature will have little entropy in its class distribution; for each value most of the instances should be primarily in one class. The information gain of an attribute is measured by the reduction in entropy: **IG(X) = H(D) − H(D|X)**. The greater the decrease in entropy when considering attribute X individually, the more significant feature X is for prediction.

5.5 CORRELATION-BASED FEATURE SELECTION(CFS):

CFS searches feature subsets according to the degree of redundancy among the features. The evaluator aims to find the subsets of features that are individually highly correlated with the class but have low inter-correlation. The subset evaluators use a numeric measure, such as conditional entropy, to guide the search iteratively and add features that have the highest correlation with the class. The downside of univariate filters for eg information gain is, it does not account for interactions between features, which is overcome by multivariate filters for eg CFS. CFS evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Correlation coefficients is used to estimate correlation between subset of attributes and class, as well as inter-correlations between the features. Relevance of a group of features grows with the correlation between features and classes, and decreases with growing inter-correlation. CFS is used to determine the best feature subset and is usually combined with search strategies such as forward selection, backward elimination, bi-directional search, best-first search and genetic search.

Equation for CFS is given.

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}}$$

where r_{zc} is the correlation between the summed feature subsets and the class variable, k is the number of subset features, r_{zi} is the average of the correlations between the subset features an the class variable, and r_{ii} is the average inter-correlation between subset features.

5.6 MARKOV BLANKET FILTERING:

Koller and Sahami (1996) suggestsed that, Let G be a subset of the overall feature set F. Let fG denote the projection of f onto the variables in G. Markov blanket filtering aims to minimize the discrepancy between the conditional distributions P (C |F=f) and P (C |G=fG), as measured by a conditional entropy.

Conditional Entropy is: **$\Delta_G = \Sigma P(f) D(P(C|F{=}f)||P(C|G{=}f_G))$** whereD(P||Q)= Σ_xP(x)log(P(x)/Q(x)) is the Kullback-Leibler divergence.The goal is to find a small feature set G for which Δ_G is small.

Definition:

Let M be some set of features which does not contain Fi. We say that M is a Markov blanket for Fi if Fi is conditionally independent of G – M – {Fi} given M. (See Pearl 1988).

Corollary:

Let G be a subset of features and Fi be a feature in G. Assume that some subset M of G is a Markov blanket of Fi. Then ∆G' = ∆G , where G' = G – Fi. Once we find a Markov blanket of feature Fi in a feature set G, we can safely remove Fi from G without increasing the divergence to the desired distribution. In a sequential filtering process in which unnecessary features are removed one by one, a feature tagged as unnecessary based on the existence of a Markov blanket Mi remains unnecessary in later stages when more features have been removed.

Approximate Markov Blanket:

$$\boldsymbol{\Delta(F_i|M)=\Sigma P(M=f_M,F_i=f_i)\ D(P(C|M=f_M,F_i=f_i)||(P(C|M=f_M))}$$

If M is a Markov blanket for F_i then $\Delta(F_i|M)=0$. Relax the condition and seek a set M such that $\Delta(F_i|M)$ is small. Those features that form an approximate Markov blanket of feature F_i are most likely to be more strongly correlated to F_i . Construct a candidate Markov blanket for F_i by collecting the k features that have the highest correlations with F_i, where k is a small integer.

ALGORITHM FOR MBF (KOLLER & SAHAMI, 1996):

1. Initialize G = F
2. Iterate
3. For each feature Fi $\in$G,let Mi be the set of k feature Fj$\in$G$-${Fi} for which the correlations between Fi and Fj are the highest.
4. Compute Δ(Fi |M) for each i
5. Choose the i that minimizesΔ(Fi |M) , and defineG=G $-${Fi}

5.7 FAST CORRELATION BASED FS (FCBF):

FCBF (Yu and Liu, ICML 2003) uses also the symmetrical uncertainty measure. But the search algorithm is very different. It is based on the “predominance” idea. The correlation between an attribute X* and the target Y is predominant if and only if $\boldsymbol{\rho_{y,x^*}\geq\delta}$**et**$\boldsymbol{\forall X(X\neq X^*),\ \rho_{x,x^*} < \rho_{y,x^*}}$

Concretely, a predictor is interesting if its correlation with the target attribute is significant (delta is the parameter which allows to assess this one); there is no other predictor which is more strongly correlated to it. Based on this idea, the authors propose a search algorithm which runs in quasilinear time.

ALGORITHM FOR FCBF:

1. S is the set of candidate predictors, M = $\varnothing$ is the set of selected predictors
2. Searching X* (among S) which maximizes its correlation with Y$\rightarrow\rho_{y,x^*}$
3. If $\rho_{y,x^*}\geq\delta$ add X* into M and remove X* from S
4. Remove also from S all the variables X such $\rho_{x,x^*}\geq \rho_{y,x^*}$ (Very important !)
5. If S $\neq\varnothing$ then GOTO (2), else END of the algorithm

This approach is very useful when we deal with a dataset containing a very large number of candidate predictors. About the ability to detect the "best" subset of predictors, as we will see in this tutorial, we note that it is similar to CFS.

5.8 SEQUENTIAL FORWARD SELECTION (SFS):

Sequential Forward Selection is the simplest greedy search algorithm. Starting from the empty set, sequentially add the feature x^+ that results in the highest objective function $J(Y_k+x^+)$ when combined with the features Y_k that have already been selected .

Algorithm:

1.Start with the empty set $Y_0=\{\phi\}$
2.Select the next best feature X^+=argmax[J(Y_k+X)];x¢Y_k
3.Update $Y_{k+1}=Y_k+ X^+$; k=k+1
4.Goto 2

SFS performs best when the optimal subset has a small number of features. When the search is near the empty set, a large number of states can be potentially evaluated. Towards the full set, the region examined by SFS is narrower since most of the features have already been selected. The search space is drawn like an ellipse to emphasize the fact that there are fewer states towards the full or empty sets.As an example, the state space for 4 features is shown. Notice that the number of states is larger in the middle of the search tree. The main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features.

5.9 SEQUENTIAL BACKWARD ELIMINATION (SBE):

Sequential Backward Elimination works in the opposite direction of SFS. Also referred to as SBS (Sequential Backward Selection). Starting from the full set, sequentially remove the feature x$-$ that results in the smallest decrease in the value of the objective function J(Y-x$-$). Notice that removal of a feature may actually lead to an increase in the objective function J(Yk-x$-$)>J(Yk). Such functions are said to be non-monotonic.

Algorithm:

1.Start with the full set Y_0=X
2.Remove the worst feature X^-=argmax[J(Y_k-X)];x Y_k
3.Update $Y_{k+1}=Y_k- X^-$; k=k+1
4.Goto 2

SBS works best when the optimal feature subset has a large number of features, since SBS spends most of its time visiting large subsets. The main limitation of SBS is its inability to reevaluate the usefulness of a feature after it has been discarded.

5.10 PLUS-L MINUS-R SELECTION (LRS):

Plus-L Minus-R is a generalization of SFS and SBS. If L>R, LRS starts from the empty set and repeatedly adds 'L' features and removes 'R' features. If L<R, LRS starts from the full set and repeatedly removes 'R' features followed by 'L' feature additions.

Algorithm:

1.If L>R then start with the empty set $Y=\{\phi\}$else start with the full set Y=X Goto step3

2.Repeat L times $X^{+}=\text{argmax}[J(Y_k+X)];x$¢$Y_k$ and $Y_{k+1}=Y_k+ X^{+}$; k=k+1

3.Repeat R times $X^{-}=\text{argmax}[J(Y_k-X)];x\ \ Y_k$ and $Y_{k+1}=Y_k- X^{-}$; k=k+1

4.Go to 2

LRS attempts to compensate for the weaknesses of SFS and SBS with some backtracking capabilities. Its main limitation is the lack of a theory to help predict the optimal values of L and R.

5.11 BEAM SEARCH AND SMART BEAM SEARCH (SBS):

Although the spread factors of features yield useful information about their goodness, it is possible that features with low values of I be important for classification, as in the case of multi-modal or non-Gaussian feature distributions. Therefore, a more generic feature selection scheme called the beam search has been used. The beam search algorithm proceeds as follows:

1. Compute the classifier performance using each of the n features individually (n I-tuples).
2. Select the best K (beam-width) features based on a pre-defined selection criterion among these I-tuples
3. Add a new feature to each of these K features, forming K(n-l) 2-tuples of features. The tuple-size t is equal to 2 at this stage.
4. Evaluate the performance of each of these t-tuples. Of these, select the best K, based on classification performance.
5. Form all possible (t + 1) tuples by appending these K r-tuples with other features (not already in that t-tuple).
6. Repeat steps 4 to 5 until the stopping criterion is met; the tuple size at this stage is m.
7. The best K m-tuples are the result of beam search.

5.12. SIMULATED ANNEALING (SA):

Simulated annealing is a paradigm method for optimization problems of many types. To form a strong alloy, one heats up the components, mixes them well and lets them cool slowly according to a strict (empirically determined) cooling schedule in order to minimize the number of defects in the solidi_cation process of the material. This physical annealing process is interpreted in the setting of combinatorial optimization (by which we mean _nding the minimum of a cost function C(x) for the vector of variables x in many dimensions) by the following meta-algorithm:

Algorithm: General Simulated Annealing

Data : A candidate solution S and a cost function C(x).

Result: A solution S' that minimizes the cost function C(x).

```
  T←Starting Temperature
While not frozen do
While not at equilibrium do
      S'←perturbation of S.
If C(S')<C(S) or selection criterion then S←S'
End
End
```

In words, we begin with a guessed solution S and heat it up using some starting temperature. While at one temperature, the system is allowed to transit to other states until it reaches equilibrium at that temperature. Transitions that lower cost are always accepted and transitions that increase cost are accepted relative to some selection criterion that usually is a function of both the current temperature and the cost increase of the proposed change. When equilibrium has been reached the temperature is changed and this is continued until the system is at equilibrium at a very low temperature at which time we call it frozen. Due to the equilibration and the guaranteed acceptance of a downhill transition, the state is certain to be a minimum in the cost function. It may however be a local minimum and not the global one. In order to increase the chances of getting the global minimum, the algorithm allows uphill transitions preferentially early in the execution of the algorithm, i.e. at high temperatures.

5.13 RANDIMIZED HILL-CLIMBING:

Hill-climbing is probably the most known algorithm of local search. The idea of hill-climbing is:

1. Start at randomly generated state
2. Move to the neighbor with the best evaluation value

3. If a strict local-minimum is reached then restart at other randomly generated state.

This procedure repeats till the solution is found. In the algorithm, that we present here, the parameter Max_Flips is used to limit the maximal number of moves between restarts which helps to leave non-strict local-minimum.

Algorithm Hill-Climbing:

```
procedure hill-climbing(Max_Flips)
restart: s <- random valuation of variables;
for j:=1 to Max_Flips do
    if eval(s)=0 then return s endif;
    if s is a strict local minimum then
        goto restart
    else
        s <- neighborhood with smallest evaluation value
    endif
endfor
goto restart
end hill-climbing
```

That the hill-climbing algorithm has to explore all neighbors of the current state before choosing the move. This can take a lot of time.

5.14 GENETIC ALGORITHM:

The Genetic Algorithms (GA) are efficient methods for function minimization. In descriptor selection context, the prediction error of the model built upon a set of features is optimized. The genetic algorithm mimics the natural evolution by modeling a dynamic population of solutions. The members of the population, referred to as chromosomes, encode the selected features. The encoding usually takes form of bit strings with bits corresponding to selected features set and others cleared. Each chromosome leads to a model built using the encoded features. By using the training data, the error of the model is quantified and serves as a fitness function. During the course of evolution, the chromosomes are subjected to crossover and mutation. By allowing survival and reproduction of the fittest chromosomes, the algorithm effectively minimizes the error function in subsequent generations. The success of GA depends on several factors. The parameters steering the crossover, mutation and survival of chromosomes should be carefully chosen to allow the population to explore the solution space and to prevent early convergence to homogeneous population occupying a local minimum. The choice of initial population is also important in genetic feature selection. To address this issue, e.g. a method based on Shannon's entropy combined with graph analysis can be used.

Genetic algorithm based on the Darwinian survival of the fittest theory, is an efficient and broadly applicable global optimization algorithm. In contrast to conventional search techniques, genetic algorithm starts from a group of points coded as finite length alphabet strings instead of one real parameter set. Furthermore, genetic algorithm is not a hill-climbing algorithm hence the derivative information and step size calculation are not required. The three basic operators of genetic algorithms are: selection, crossover and mutation. It selects some individuals with stronger adaptability from population according to the fitness, and then decides the copy number of individual according to the selection methods such as Backer stochastic universal sampling. It exchanges and recombines a pair of chromosome through crossover. Mutation is done to change certain point state via probability. In general, one needs to choose suitable crossover and mutation probability time and again via real problems.

5.15 ESTIMATION OF DISTRIBUTION ALGORITHMS:

Estimation of distribution algorithms (EDAs) address broad classes of optimization problems by learning explicit probabilistic models of promising solutions found so far and sampling the built models to generate new candidate solutions. By incorporating advanced machine learning techniques into genetic and evolutionary algorithms, EDAs can scalably solve many challenging problems, significantly outperforming standard genetic and evolutionary algorithms and other optimization techniques. In the recent decade, many impressive results have been produced in the design, theoretical investigation, and applications of EDAs.The pseudocode of an EDA follows:

```
Estimation of Distribution Algorithm (EDA)
t := 0;
generate initial population P(0);
while (not done) {
    select population of promising solutions S(t);
    build probabilistic model P(t) for S(t);
    sample P(t) to generate O(t);
    incorporate O(t) into P(t);
    t := t+1; }
```

EDAs derive inspiration from two areas: genetic and evolutionary computation and machine learning. The remainder of this section discusses these two sources of inspiration.

5.16 DECISION TREES:

Decision tree learner is a tree structure where each non-leaf node represents a test on a feature, each branch denotes an outcome of the test, and each leaf node represents a class label. The Decision Tree classifier became popular due to the fact that the construction of a decision tree classifier does not require any domain knowledge, and the acquired knowledge in a tree form is easy to understand. In addition, the classification step of decision tree induction is simple and fast. Besides the splitting criterion, another interesting challenge of building a decision tree is to overcome the over-fitting of the data. To achieve that, pruning. Different methods exist to build decision trees, which summarise given training data in a tree structure, with each branch representing an association between attribute values and a class label. The most famous and representative amongst these is, perhaps, the algorithm. It works by recursively partitioning the training data set according to tests on the potential of attribute values in separating the classes. The core of this algorithm is based on its original version, named the ID3. So, to have a basic understanding of how this algorithm works, the ID3 method is outlined below. The decision tree is learned from a set of training examples through an iterative process, of choosing an attribute (i.e. feature) and splitting the given example set according to the values of that attribute. The key question here is which of the attributes is the most influential in determining the classification and hence should be chosen first. Entropy measures or equivalently, information gains are used to select the most influential, which is intuitively deemed to be the attribute of the lowest entropy (or of the highest information gain). In more detail, the learning algorithm works by: a) computing the entropy measure for each attribute, b) partitioning the set of examples according to the possible values of the attribute that has the lowest entropy, and c) for each subset of examples repeating these steps until all attributes have been partitioned or other given termination conditions met. In order to compute the entropy measures, frequencies are used to estimate probabilities, in a way exactly the same as with the Naive Bayes approach. Note that although attribute tests are chosen one at a time in a greedy manner, they are dependent on results of previous tests.

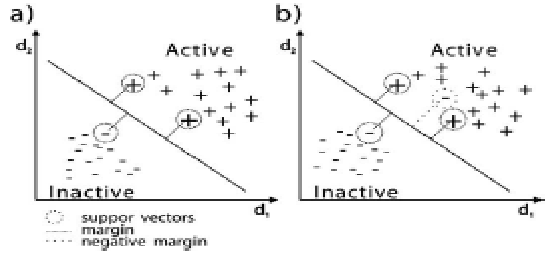
5.17 NAIVE BAYES:

A Naive Bayes classifier can achieve relatively good performance on classification tasks, based on the elementary Bayes' Theorem. It greatly simplifies learning by assuming that features are independent given the class variable. More formally, a Naive Bayes classifier is defined by discriminant functions: $\mathbf{f_i(X)=\prod P(x_j|c_i)P(c_i);}$ where X = (x1; x2; :::; xN) denotes a feature vector and cj ; j = 1; 2; :::;N, denote possible class labels. The training phase for learning a classifier consists in estimating conditional probabilities P(xj jci) and prior probabilities P(ci). Here, P(ci) are estimated by counting the training examples that fall into class ci and then dividing the resulting count by the size of the training set. Similarly, conditional probabilities are estimated by simply observing the frequency distribution of feature xj within the training subset that is labelled as class ci. To classify a class-unknown test vector, the posterior probability of each class is calculated, given the feature values present in the test vector; and the test vector is assigned to the class that is of the highest probability.

The Bayesian classifier is a statistical classifier, which has the ability to predict the probability that a given instance belongs to a particular class. The probabilistic Naive Bayes classifier is based on Bayes's rule and assumes that given the class, features are independent, which is called class conditional independence. In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers. However, this is not always the case in practice, because of the previously mentioned assumption. Even so, the Naïve Bayesian classifier has exhibited high accuracy and high speed when applied to large databases.

5.18 SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine is built on the structural risk minimization principle to seek a decision surface that can separate the data points into two classes with a maximal margin between them. The choice of the proper kernel function is the main challenge when using a SVM. It could have different forms such as Radial Basis Function (RBF) kernel and polynomial kernel. The advantage of the SVM is its capability of learning in sparse, high-dimensional spaces with very few training examples by minimizing a bound on the empirical error and the complexity of the classifier at the same time. WEKA uses the Sequential Minimal Optimization (SMO) algorithm for SVM. The Support Vector Machines (SVM) form a group of methods stemming from the structural risk minimization principle, with the linear support vector classifier as its most basic member. The SVC aims at creating a decision hyper plane that maximizes the margin, i.e., the distance from the hyper plane to the nearest examples from each of the classes. This allows for formulating the classifier training as a constrained optimization problem. Importantly, the objective function is unimodal, contrary to e.g. neural networks, and thus can be optimized effectively to global optimum. In the simplest case, compounds from different classes can be separated by linear hyper plane; such hyper plane is defined solely by its nearest compounds from the training set. Such compounds are referred to as support vectors, giving the name to the whole method. In most cases, however, no linear separation is possible. To take account of this problem, slack variables are introduced. These variables are associated with the misclassified compounds and, in conjunction with the margin, are subject to optimization. Thus, even though the erroneous classification cannot be avoided, it is penalized. Since the misclassification of compounds strongly influences the decision hyper plane, the misclassified compounds also become support vectors.



Support vectors and margins in linearly separable (a) and non-separable (b) problems. In non-separable case, negative margins are encountered and their magnitude is subject to optimization along with the magnitude of the positive margins.

6. TAXONOMY OF FEATURE SELECTION ALGORITHMS

Taxonomy of feature selection techniques is shown in the following table. For each feature selection type, we highlight a set of characteristics which can guide the choice for a technique suited to the goals and resources of practitioners in the field

TAXONOMY OF FEATURE SELECTION ALGORITHMS

MODEL SEARCH		ADVANTAGES	DISADVANTAGES	EXAMPLES
FILTER	UNIVARIATE	Fast, Scalable, Independent of the classifier	Ignores feature dependencies, Ignores interaction with the classifier	X^2, Euclidian distance, t-test, Information gain
	MULTIVARIATE	Models feature dependencies, Independent of the classifier, Better computational complexity than wrapper methods	Slower than univariate techniques, Less sclable than univariate techniques, Ignores interaction with the classifier	Correlation-based feature selection(CFS), Markov blanket filter (MBF), Fast correlation-based feature selection (FCBF)
WRAPPER	DETERMINISTIC	Simple, Interacts with the classifier, Models feature dependencies, Less computationally intensive than randomized methods	Risk of over fitting, More prone than randomized algorithms to getting stuck in a local optimum (greedy search), Classifier dependent selection	Sequential forward selection (SFS), Sequential backward elimination (SBE), Plus L Minus R, Beam search
	RANDOMIZED	Less prone to local optima, Interacts with the classifier, Models feature dependencies	Computationally intensive, Classifier dependent selection, Higher risk of over fitting than deterministic algorithms	Simulated annealing, Randomized hill climbing, Genetic algorithms, Estimation of distribution algorithms
EMBEDDED		Interacts with the classifier, Better computational complexity than wrapper methods, Models feature dependencies	Classifier dependent selection	Decision trees, Weighted naïve Bayes, Feature selection using the weight vector of SVM

7. CONCLUSION:

The task of a feature selection algorithm (FSA) is to provide with a computational solution to the feature selection problem motivated by a certain definition of *relevance*. This algorithm should be reliable and efficient. The many FSAs proposed in the literature are based on quite different principles (as the evaluation measure used, the precise way to explore the search space, etc) and loosely follow different definitions of relevance.

In this work a way to evaluate FSAs was proposed in order to understand their general behaviour on the particularities of relevance, irrelevance, redundancy and sample size of synthetic data sets. To achieve this goal, a set of controlled experiments using artificially generated data sets were designed and carried out. The set of optimal solutions is then compared with the output given by the FSAs (the obtained hypotheses). To this end, a *scoring* measure was defined to express the degree of approximation of the FSA solution to the real solution. The final outcome of the experiments can be seen as an illustrative step towards gaining useful knowledge that enables to decide which algorithm to use in certain situations.

In this vein, it is shown the different behavior of the algorithms to different data particularities and thus the danger in relying in a single algorithm. This points in the direction of using new hybrid algorithms or combinations thereof for a more reliable assessment of feature relevance. As future activities, this work can be extended in many ways to carry up richer evaluations such as considering features strongly *correlated* with the class or with one another, noise in the data sets, other kinds of data (e.g., continuous data), missing values, and the use of combined evaluation measures.

8. REFERENCES:

[1] Efficient IRIS Recognition through Improvement of Feature Extraction and subset Selection, Amir Azizi, Islamic Azad University Mashhad Branch, Hamid Reza Pourreza, Ferdowsi University of Mashhad, Vol 2, June, 2009.

[2] An Unsupervised Feature Selection Method Based On Genetic Algorithm, Nasrin Sheikhi, Amirmasoud Rahmani, Mehran Mohsenzadeh, Department of computer engineering, Islamic Azad University of Iran research and science branch, Ahvaz, Iran Reza Veisisheikhrobat, National Iranian South Oil Company(NISOC), Ahvaz, Iran, Vol 9, no.1 Jan,2011.

[3] An ensemble approach for feature selection of Cyber Attack Dataset, Shailendra Singh, Department of Information Technology, Rajiv Gandhi Technological University, Bhopal, India. Sanjay Silakari, Department of Computer Science & Engineering, Rajiv Gandhi Technological University, Bhopal, India, Vol 6, Nov 2009.

[4] CRS, a Novel Ensemble Construction Methodology, Navid Kardan, Computer Engineering Dep. IUST, Tehran, Iran. Morteza Analoui, Computer Engineering Dep., IUST, Vol 8, Jun,2010.

[5] Five New Feature Selection Metrics In Text Categorization, FENGXI SONG, DAVID ZHANG, YONG XU and JIZHONG WANG, Department of Automation and Simulation 451 Huang Shan Road Hefei, Anhui 230031, P. R. China .

[6] A Survey on Data Mining Techniques for Gene Selection and Cancer Classification, Dr. S. Santhosh Baboo, Reader, PG and Research department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai S. Sasikala, Head, Department of Computer Science, Sree Saraswathi Thyagaraja College, Pollachi,Vol 8, No 1, Apr, 2010

[7] Clustering Time Series Data Stream – A Literature Survey, V.Kavitha, Computer Science Department, Sri Ramakrishna College of Arts and Science for Women, Coimbatore, Tamilnadu, India. M. Punithavalli, Sri Ramakrishna College of Arts & Science for Women, Coimbatore ,Tamil Nadu, India., Vol 8, Apr, 2010.

[8] “comparative study of attribute selection using gain ratio and correlation based feature selection”, asha gowda karegowda1, a. s. manjunath2 & m.a.jayaram3,vol 2,dec,2010.

[9] “feature selection for high-dimensional data: a fast correlation-based filter solution”, lei yu and huan liu.

[10] “A review of feature selection techniques in bioinformatics”, yvan saey, in aki inza and pedro larran aga. vol. 23 no. 19 2007, pages 2507–2517

[11] m. dash, and h. liu, ”feature selection for classification,” *international journal of intelligent data analysis*, vol. 1.

[12] x. he, d. cai, and p. niyogi, ”laplacian score for feature selection,” *nips*, 2005.8] z. zhao, and h. liu, ”spectral feature selection for supervised and unsupervised learning,” *proc. ieee int’l conf. machine learning*(icml’07), 2007.

[13] “Classification and feature selection algorithms for multi-class CGH data”,Jun Liu, Sanjay Ranka and Tamer Kahveci Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA, *Vol. 24 ISMB 2008.*

[14] Araujo DLA, Lopes HS and Freitas AA. A parallel genetic algorithm for rule discovery in large databases. *Proc. 1999 IEEE Systems, Man and Cybernetics Conf.*, v. 3, 940-945. Tokyo.

[15] X.-w. Chen and M. Wasikowski. Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *KDD ’08: Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining*, pages 124–132, New York, NY, USA, 2008. ACM.

[16] Feature Selection in Data Mining Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, June 21st, 2010, Hyderabad, India.Editors: Huan Liu, Hiroshi Motoda, Rudy Setiono, Zheng Zhao

[17] Feature Subset Selection: A Correlation Based Filter Approach, Mark A. Hall, Lloyd A. Smith.

[18] Feature Selection Methods-Data mining to pick predictive variables, Ravi Kumar ACAS, MAAA CAS Predictive Modeling Seminar San Diego,October, 2008.

[19] Adaptive Feature-Space Conformal Transformation for Imbalanced-Data Learning. Gang Wu, Edward Y. Chang, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003).*

[20] Feature Selection in Data Mining, YongSeog Kim, W. Nick Street, and Filippo Menczer, University of Iowa, USA.

[21] Feature Selection, Martin Sewell, 2007.

[22] Feature selection and classification using flexible neural tree, Yuehui Chena, Ajith Abrahama,b, Bo Yanga,c,june 2006.

[23] Study on Feature Selection Techniques in Educational Data Mining, M. Ramaswami and R. Bhaskaran,Vol1,Dec,2009.

[24] Classifier Learning for Imbalanced Data with Varying Misclassification Costs, J ¨org Mennicke,Nov 2006.

[25] Multi-core Design and Memory Feature Selection Survey, Yuval Peress, Major Professor: Dr. Gary Tyson.

[26] Learning Concepts from Large Scale Imbalanced Data Sets Using Support Cluster Machines, Jinhui Yuan, Jianmin Li and Bo Zhang.

[27] Feature Selection in Models For Data Mining, Robert stine,Jan,2005.

[28] t. liu, s. liu, z. chen, and w. ma, ”an evaluation on feature selection for text clustering,” *proc. ieee int’l conf. machine learning (icml’03*, pp. 488-495, 2003.

[29] http://msdn.microsoft.com/en-us/library/ms175382.aspx.

[30] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J.Mach. Learn. Res.*, 3:1289–1305, 2003.