Journal of Machine Learning Research ? (2003) ?-? Submitted /; published ?/?

Enhancing Learning using Feature and Example selection

Baranidharan Raman BARANI@CS.TAMU.EDU
Thomas R.Ioerger IOERGER@CS.TAMU.EDU
Department of Computer Science
Texas A&M University
College Station, Texas -77840, USA.

Abstract

While most of the stable learning algorithms perform well on domains with relevant information, they degrade in the presence of irrelevant or redundant information. Selective or focused learning presents a solution to this problem. Two components of selective learning are selective attention (feature selection) and selective utilization (example selection). We present novel algorithms for feature and example selection and present the benefits of these two approaches independently and as a combined scheme. We propose a sequential search filter approach called Subset selection using Case-based Relevance APproach (SCRAP) for identifying and eliminating irrelevant features. The SCRAP filter addresses the problem of finding a feature subset that provides a balance between defining consistent hypotheses and improving prediction accuracy. SCRAP filter was compared with the RELIEF filter and was found to perform better on three families of learning algorithms. We also propose the learning algorithm using SEarch Ring (LASER) framework to perform example selection for learning algorithms. The naive bayes learner was used as the target learner for our experiments. LASER provides significant improvement in prediction accuracy of the naive bayes learner without example selection. Application of both feature and example selection schemes to the naive bayes learner resulted in better prediction accuracy.

Keywords: Feature Selection, Example Selection, Naive Bayes learner, Nearest Neighbors, Class Imbalance.

1. Introduction

Learning is an important aspect of research in Artificial Intelligence. Many statistical, symbolic, connectionist and case-based algorithms have been proposed with good success [Quinlan, 1993, Rosenblatt, 1962, Aha et al, 1991, Michalski, 1983]. Many of the existing learning approaches consider the learning algorithm as a passive entity that makes use of the information presented to them. Such schemes are called 'Passive Learners' by Cohn et al (1995). Markovitch (1989) identifies irrelevant, noisy and redundant information as the harmful elements of knowledge. The passive learning schemes will degrade performance on domains with these harmful elements.

In this paper, we focus on the issue of selecting relevant information. This involves solving two problems namely, the problem of selecting relevant features and the problem of selecting relevant examples. Markovitch (1989) defines these components of selective learning as 'Selective Attention' and 'Selective Utilization' respectively.

We present the benefits of selective attention and selective utilization independently and as a combined strategy.

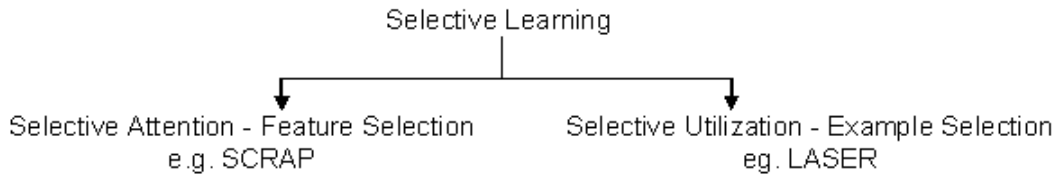

Figure 1: Selective Learning

1.1 Overview of Relevance

The term 'relevance' has a number of definitions in the Machine Learning literature. The objective or goal of features selection or examples selection defines the term 'relevance' [Blum and Langely, 1997]. For example if the objective of our selection scheme is to improve the prediction accuracy then the selected features are relevant to target concept.

The problem of 'relevance' is applicable to domains with irrelevant or redundant information, especially to domains that represent larger and more complex tasks.

1.1.1 Feature Relevance

Finding the smallest possible feature set is NP-Complete [Davies and Russell, 1994]. The presence of irrelevant features makes the task of learning difficult. The predictive accuracy of the learning algorithms decreases on domains with irrelevant features [Langely, 1996, Rendell and Seshu, 1990]. Irrelevant and redundant features masks or obscures the distribution of truly relevant features for the task in hand [Koller and Sahami, 1996]. John(1997) has shown that a single irrelevant feature to credit-approval or diabetes data sets reduced the prediction accuracy of C4.5 by 5%. Langely and Sage (1994) have shown that the naive bayes learner performs sub-optimally on domains with redundant features like the voting data set. These reasons advocates the need for doing some feature filtering before the learning algorithm is used.

Feature preprocessing schemes like Feature Extraction, Feature Construction and Feature Selection have been used to deal with this problem [Liu and Motoda, 1998]. Feature extraction schemes perform linear/non-linear transformation of data and project it to a lower dimensional space in such a way that most of the information is retained. Examples of such schemes are linear discriminant analysis and principal component analysis [Duda, Hart and Stork, 2000]. Feature construction tries to simplify hypothesis search by adding newer features with more information [Matheus and Rendell, 1989]. These two approaches try solving the problem of irrelevant information in the feature space by changing the representation.

Feature selection approaches try finding features that retain the maximum useful information amongst all the given features. The problem of finding relevant features from the given feature space is defined as 'Feature Selection'. There are three kinds of feature relevance being assumed in empirical models in the past.

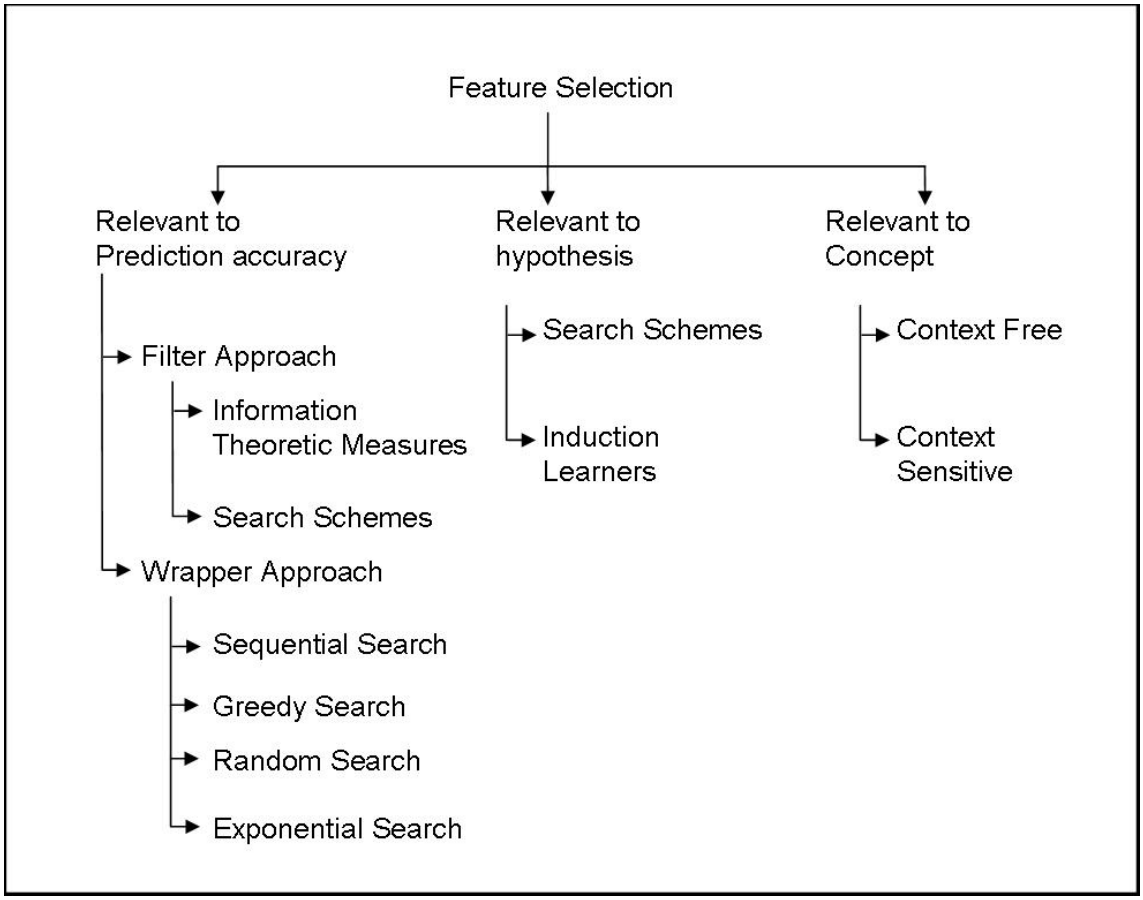

Figure 2: Feature Selection Algorithms Classification

- Relevance of feature to constructing consistent hypothesis.
- Relevance of feature to improving prediction accuracy.
- Relevance of feature to the concept.

Based on these three relevance metrics the feature selection algorithms can be grouped as shown in Figure 2. The sub trees are not mutually exclusive. This means that a feature selection algorithm can be a member of more than one sub tree.

The features relevant to the construction of consistent hypotheses are called MIN-FEATURES [Almuallim and Dietterich, 1991]. A hypothesis is consistent if no two examples agree on all the features in the feature space and have different class labels. There are three algorithms that find try finding MIN-FEATURES namely, ID3 [Quinlan, 1993], FOCUS [Almuallim and Dietterich, 1991], and FRINGE [Pagallo and Haussler, 1986]. Finding MIN-FEATURES has been shown to reduce to the vertex cover Problem in polynomial time making this a NP-Complete problem [Davies and Russell, 1994].

The second type of feature relevance is involved in finding the feature set that optimizes the prediction accuracy of the learning algorithm. There are two sub classes of feature selection algorithms in this category namely, 'Wrapper Approaches' [Kohavi and John, 1998] and 'Filter Approaches'. A detailed description of these methods are provided in Section 2.

The Wrapper approaches are heuristic search procedures that evaluate the quality of the feature subset by using the prediction accuracy of the target learning scheme on the validation set. Search techniques like sequential forward and backward feature selection [Devijver and Kitler, 1982] have been used as Wrappers. The other search schemes used as Wrappers include the greedy variants of hill climbers [Caruana and Freitag, 1994], best-first search [Kohavi and John, 1998] and beam search [Aha and Bankert, 1995]. All these wrapper approaches are computationally expensive but provide greater increase to prediction accuracy. This is mainly due to the fact that they include the bias of the target learning algorithm. Another noticeable observation from these works is that there is no algorithm that performs optimally on all domains, as shown by variability in experimental results. This is understandable as feature selection is a highly domain specific task.

The Filter approach to feature selection attempts to remove the irrelevant features from the feature set before it is used by the learning algorithm [Liu and Motoda, 1998]. The examples of feature evaluating measures are intrinsic properties of the data, probabilistic distance measures, probabilistic dependence measures, interclass distance measures, information theoretic measures like entropy measures etc [Doak, 1992]. FOCUS [Almuallim and Dietterich, 1991], Koller and Sahami's (1996) cross-entropy filter and RELIEF and its variants [Kira and Rendell, 1992, Kononenko, 1994] are some of the well known filter schemes. Filter approaches are computationally less expensive but return a large feature sub-set. As noted above there is not a single filter approach that performs well on all the data sets.

The third type of feature relevance is based on context-sensitivity of the feature. Context-sensitivity refers to the correlation between the feature and the target space [Pedrod, 1996]. Context-sensitivity can be global or local. Backward elimination [Devijver and Kitler, 1982] is a global context-sensitive feature selection approach while RC (Relevance in Context) [Pedrod, 1996] is an example of local context-sensitive approach. Feature selection schemes like forward selection Search [Devijver and Kitler, 1982] are examples of context-free feature selection.

Despite the variety of methods that perform feature selection, there has been little work done to combine these relevance definitions and develop an empirical framework that will strike a balance between the hypothesis consistency and improving prediction accuracy. There are two main problems in developing such a framework. Firstly, finding MIN-FEATURES is NP-Complete. Secondly, the feature set that provides maximum increase in classification accuracy need not define hypotheses consistently. This happens when incrementally useful [Blum and Langely, 1997] features are selected by Wrapper schemes. For example, there might be features that help define few hypotheses consistently, but their inclusion results in a particular target learner to degrade performance. These features are removed by Wrapper approaches to improve performance at the cost of losing hypothesis consistency.

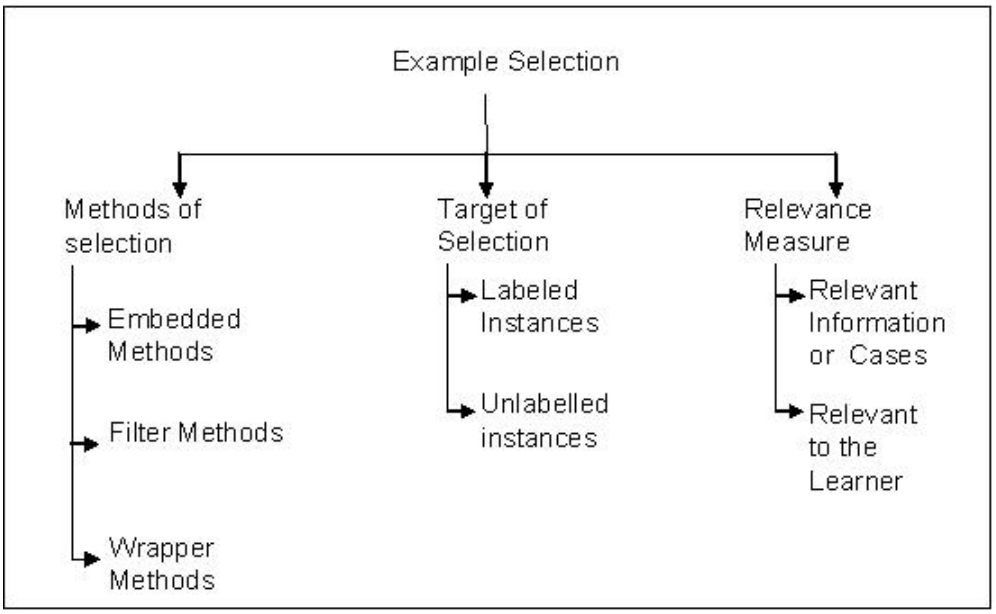

Figure 3: Example Selection Algorithms Classification [Blum and Langely, 1997]

1.2 Example Relevance

Another important component of selective learning is selecting relevant training examples for classifying a test case. Example selection involves selecting examples that aid the learning process [Blum and Langely, 1997]. Such a type of learning can be considered as a selective learning [Markovitch, 1989] or focused learning approach [Blum and Langely, 1997]. Selective learning helps reduce the effect of the harmful elements of information such as noise in the data. Unlike the feature selection problem, example selection does not imply deleting information in the training set, but it means using only the informative instances. There are three reasons for selection examples: computational efficiency, which arises when there are sufficiently large training examples, so learning from a subset will be computationally efficient, high cost of labeling or easy generation of examples and increased rate of learning by focusing attention on informative examples to aid learning algorithms search the hypotheses space [Blum and Langely, 1997].

The example selection schemes can also be classified into filter methods, wrapper methods and methods that are part of the learning algorithm. Other classifications of example selection are shown in Figure 3.

The filter method of example selection acts as a preprocessor to the learning algorithm. Various static sub-sampling methods come in this category. Lewis and Catlett (1994) use one probabilistic classifier to select instances for training another classifier. Wrapper models for example selection are used to iteratively update the models using misclassified data like the windowing technique used for decision trees working on large training sets [Quinlan, 1983]. Dynamic sub-sampling methods come under this category

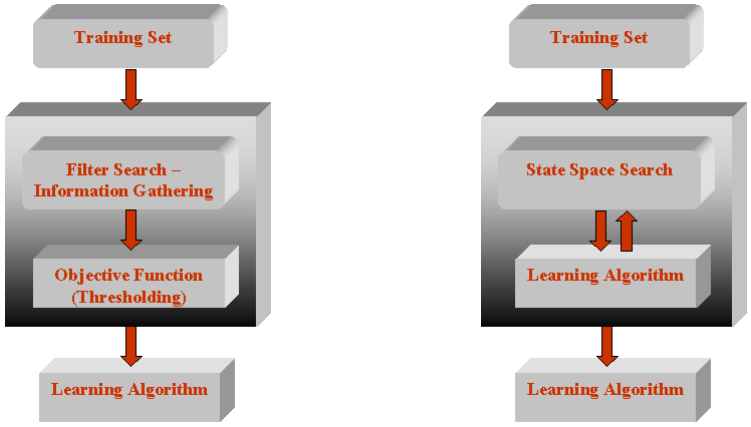

Figure 4: Feature Selection -Filter and Wrapper Approaches

[John and Langley, 1996]. The boosting algorithm [Schapire, 1990] that performs example selection by changing the distribution of the training set also comes in this category.

In the embedded methods of example selection the learning algorithm selects the relevant examples. The perceptron algorithm [Rosenblatt, 1962] uses only those examples that are misclassified to update its weights. The k-nearest neighbor algorithm [Hart and Cover, 1967] selects the 'k' most similar labeled instances in the training set to determine the class label of the unseen example. These methods depend on the test case to select the relevant examples unlike the wrapper approaches like boosting, which perform global selection of training examples independent of test examples. Winston(1975) presents a learning scheme with a tutor who gives informative instances. A embedded method for selecting unlabeled data is provided using 'query by committee' [Seung e tal., 1992].

2. Related Machine Learning Research on Feature and Example Selection

In the first part of this section we will review the existing feature selection algorithms and in the second part we will discuss about the various example selection schemes. We will study the different models of feature selection as either Wrapper approach or Filter approach.

Feature selection is one generic way of improving learning algorithms by adapting or optimizing the representation of the examples. The problem of feature selection [John et al, 1994, Langley, 1994] can be defined as the task of selection of a subset of features that describe the hypothesis at least as well as the original set.

2.1 The Wrapper Approach

The Wrapper approach to feature selection conducts a feature space search for evaluating features. The wrapper approach includes the learning algorithm as a part of their evaluation function. The wrapper schemes perform some form of state space search and select or remove the features that maximize an objective function. The subset of features selected is then evaluated using the target learner. The process is repeated until no improvement is made or addition/deletion of new features reduces the accuracy of the target learner. Wrappers

usually provide better accuracy but are computationally more expensive than the Filter schemes.

2.1.1 Sequential Search techniques

The idea of sequential search for feature selection was introduced by Devijver and Kitler (1982). A search scheme forms the integral part of the feature selection algorithm. The effectiveness of the heuristic used determines the performance of the Wrapper algorithm. Sequential search schemes add or remove features one-at-a-time. The Wrappers that perform sequential search have a vulnerability of being trapped in local minima. The randomized algorithms inject some non-determinism to help the search procedure to escape local minima [Doak, 1992]. We will discuss some well known feature selection search schemes in this section.

Forward Selection search

The forward selection is a simple algorithm that starts with an empty set and adds one feature at a time until all features are added/dropped. The feature is added at each step that most increases the performance of the learner.

Backward Elimination search

Backward elimination works exactly opposite to forward selection. Here we start with the complete feature set and drop each feature and observe the performance of the learner. If the generalization produced with the current set of features is better, then the feature is dropped and we proceed with the next feature.

It is not clear whether backward elimination or forward selection will perform better on a data set with no prior information on feature correlation.

Bi-directional search

In bi-directional search we do both forward selection search and backward elimination search[Doak, 1992]. Convergence of the search procedure is ensured by not adding features eliminated and not eliminating features added. Other, variants include the plus-L minus-R [Devijver and Kitler, 1982] searches where 'R' features are removed after adding 'L' features. If L > R we start with an empty set and if R>L then we start with the full set of features. This scheme attempts to compensate for the weakness of forward selection and backward elimination using back tracking. But optimal L and R values are difficult to set.

Greedy search

Caruana and Freitag (1994) explore some greedy schemes like backward stepwise elimination - SLASH (BSE-SLASH) scheme, which starts like the backward elimination with all the features but after the evaluation using a decision tree, makes use of only those features that were used to build the model by the learner at that step. BSE-SLASH does this at every step during its search. They also explore the use of greedy bi-directional hill climbers. Caruana and Freitag (1994) suggest that the hill climbing searches improves the generalization performance. However, it is not clear whether a particular scheme might work better

than the other on a given domain.

Best first search

The best-first search [Ginsberg, 1993] selects the most promising node generated but not expanded for search. Kohavi and John (1998) used best first search in their wrapper approach. Each node in the search space represents a feature subset. For 'n' features there are 'n' bits in each state and each bit indicates whether a feature is present (1) or absent (0). Compound operators are used to connect states. The operators used are addition or deletion of a single feature. The search is initiated with the goal of finding a state with maximum prediction accuracy. Because of the complexity of the search space $O(2^n)$, the state space search is stopped if there is no improvement to the accuracy after 'k' attempts.

Kohavi and John (1998) found that their wrapper performed better than RELIEF when used with the ID3 and the naive bayes learners.

RC: Relevance in Context

RC [Pedrod, 1996] is context-sensitive feature selection scheme. RC is similar to backward selection Search but makes feature evaluations using local instances. It performs a sequential search and finds nearest neighbor of same class to each instance. All the features that change are hypothesized as irrelevant and are deleted. The current feature set is tested on the target learner like in other Wrapper approaches. If the accuracy improves or remains unaffected then the deleted features are not considered again. This feature evaluation is done in parallel for all instances in the training set.

Beam search

The beam search is like breadth first search(BFS) but unlike BFS only the best 'n' nodes at each level are placed at the head of the search queue and are used for further search. The beam search becomes exhaustive if there are no bounds on queue size. If the queue size becomes one, it reduces to forward selection search. Beam search is extremely powerful on data sets with a small instance space and large number of features [Aha and Bankert, 1995].

2.1.2 Randomized Algorithms

Randomized algorithms prevent the feature selection search from converging to local minima like the sequential searches. We discuss the use of simulated annealing and genetic algorithms for the purpose of feature selection.

Simulated Annealing

Simulated Annealing [Kirk et al, 1983] & [Hayk, 1994] is another application of stochastic optimization search scheme to feature selection. In simulated annealing the system state is subjected to a small random change and we either accept the new state if it is better than the previous state or accept a deteriorating state with a probability $\exp(-\frac{\Delta E}{T})$ where E refers to energy of the state and T temperature.

In case of feature selection the transformation will consist of adding or removing the features.

Genetic Algorithms

Genetic algorithms [Goldberg, 1989] start from a random initial population and create a better population by mating or cross-over between pairs of solutions and mutating solutions to try to improve their fitness or some objective function. The instance space is represented using bit strings with a '1' if a feature was selected for the newer population and a '0' otherwise [Doak, 1992].

2.2 Filter Approaches

Filter approaches for feature subset selection attempt to assess the features and their merits using the data available [Liu and Motoda, 1998]. They remove the irrelevant features before the data is presented to the learning algorithm. The decision tree filter [Clardie, 1993], FOCUS [Almuallim and Dietterich, 1991], RELIEF and its variants [Kira and Rendell, 1992, Kononenko, 1994] are some of the widely known Filter algorithms. The decision tree filter and FOCUS filter try finding MIN-FEATURES. The Filter algorithm evaluates the features independent of the classifiers that use them. Statistical and information-theoretic measures like information gain, cross-entropy, etc., can be used to weigh the relevance of the features [Devijver and Kitler, 1982]. Other measure that has been used are intrinsic properties of the data, probabilistic distance measures, probabilistic dependence measures, interclass distance measures, information theoretic measures like entropy measures etc [Doak, 1992]. These measures capture the relationship of the feature with the target concept.

2.2.1 Decision-Tree Filter

The ID3 decision-tree induction algorithm [Quinlan, 1993] uses information gain computed using the training set to evaluate features. The ID3 builds a top-down hierarchical model of the concept with the most relevant feature as the root and less relevant features at the lower levels (near the leaves) of the decision tree.

$$Gain(f, X) = Entropy(X) - \sum_{v \in values(f)} \frac{|X_v|}{|X|} Entropy(X_v)$$

Entropy is given by the equation

$$Entropy(X) = -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$$

$p_{\oplus}$ is the proportion of positive examples in X and $p_{\ominus}$ is the proportion of negative examples in X.

Decision trees use only those features that are required to completely classify the training set and removes all other features. Cardie (1993) discusses the use of decision tree features with a k-nearest neighbor for learning a natural language processing task. The results show clearly that the quality of the subset generated by decision tree helped the k-nearest neighbor to reduce its prediction error.

2.2.2 Cross-Entropy Filter

Koller and Sahami (1996) present the cross-entropy filter approach. They define the task of feature selection as the task of finding a feature subset G such that Pr(C|G=f_G) is close

to Pr(C|F=f), where C is set of classes and F is the feature set and Pr(C|F=f) is the probability of assigning an unlabelled instance to a particular label in C with features in F assigned to values f. The cross entropy between two distributions measures the extent of error if one distribution is substituted by the other. Suppose, that the distribution of the original feature set is represented as μ and the approximated distribution due the reduced feature set as σ. Then cross entropy is given by,

$$D(\mu, \sigma) = \sum_{x \in \mu} \mu(x) log \frac{\mu(x)}{\sigma(x)}$$

A feature set G that minimizes $\sum_f$ Pr(f)D(Pr(C|f),Pr(C|f_G)) is the optimal subset.

2.2.3 Focus

The FOCUS [Almuallim and Dietterich, 1991] algorithm tries to identify a subset of features that is sufficient to re-construct the hypothesis correctly. Given a training space X, FOCUS tries to find the subset of features called MIN-FEATURES that are sufficient. A subset of features is said to be sufficient iff. there are no two instances that have same values for all the features and have conflicting class labels. Alternatively, MIN-FEATURES is the least number of features with which a consistent hypothesis can be constructed. FOCUS performs exhaustive search to determine the best feature sub-set. The complexity of FOCUS is on the order of O(mpn^p) for a training set with m examples or instances, n features and p relevant features.

2.2.4 RELIEF

The RELIEF algorithm is an instance based filter proposed by Kira and Rendell (1992). The proposed version of the algorithm was not capable of handling instance spaces with more than two class labels. Kononenko (1994) modified RELIEF to overcome this limitation.

Each feature i in the feature set f_i is assigned a weight W_i. If this weight is greater than a threshold τ then the feature is considered relevant to the target concept, else it is dropped.

---

1. Initialize weights W_i =0 $\forall$ i$\in$ F (set of features).
2. for each instance in training set pick a $near - hit^+$ and a $near - miss^-$.
3. Update Weight: $W_i = W_i$ - $diff(x_i, near - hit_i^+)^2$ +$diff(x, near - miss_i^-)^2$.

In the above listing *diff(Feature-Instance1, Feature-Instance2)* calculates the difference in the values of the features. If the feature is nominal then the difference is 0 or 1 depending on whether they are same or different. If the feature is continuous then the difference is normalized.

Kononenko (1994) suggested variants of this algorithm to handle missing values and for multi-class problems.

2.3 Example Selection

The problem of example selection is to identify a subset of examples from the example space that aid the learning process more than others [Blum and Langely, 1997]. Like feature selection, the example selection methods can be grouped as Filter methods, Wrapper methods and Embedded methods.

2.3.1 Filter Methods

Filtering techniques are least common among the example selection schemes. The different sampling techniques like static sampling and random sampling fall in this category. Given a sample, static sampling runs the appropriate hypothesis test on each of its fields to test whether they come from the same distribution as the original database, and reports whether the current sample size is sufficient [John and Langley, 1996]. The 'Probably Close Enough criterion (PCE)' is a way of evaluating a sampling strategy. Another filter approach presented by Lewis and Catlett (1994) suggest the use of one learning algorithm to filter examples for the other.

2.3.2 Wrapper Methods

The best known Wrapper method for example selection is boosting [Schapire, 1990]. The boosting technique modifies the distribution of the data. This is done by assigning increasing probability to misclassified instances, which causes the weaker learning algorithms in an ensemble to train on these highly probable misclassified instances.

Another well known Wrapper example selection method applied to decision trees is windowing [Quinlan, 1983]. Windowing uses a random sub sample of the data for building a initial tree. The remaining examples in the training set are tested for class label using the built tree. A random sample of the examples that were misclassified are added to the example set. The process is iterated until all examples are correctly classified.

2.3.3 Embedded Methods

Embedded methods are those learning algorithms that have the example selection strategy embedded in their learning scheme. For example the perceptron algorithm [Rosenblatt, 1962] selects all the instances that were misclassified for updating its weights. These schemes, which ignore all the examples that were correctly classified are called 'conservative algorithms' [Blum and Langely, 1997]. Lazy learners like k-nearest neighbors perform example selection by identifying the k closest cases to the test example.

Seung et al (1992) proposed an embedded mechanism called query by committee. In this method two random hypotheses amongst a list of consistent hypotheses of an induction algorithm are selected for classifying a random example from the example space. If the classification using both the hypotheses is different, then the example is selected for training.

3. SCRAP Feature Selection

A labelled instance is represented as an ordered pair ($\vec{x}$,y) where $\vec{x}$ is an element vector of X (instance space) and y belongs to set of class labels Y. Each element vector contains

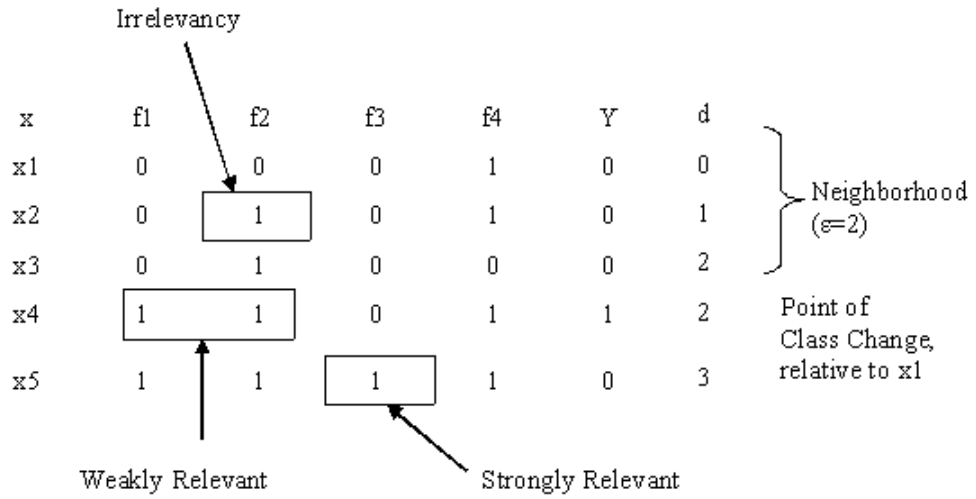

Figure 5: An Example - XOR(f1,f3) Concept

'n' features represented as $(f_1,...,f_n)$. Also the training set is comprised of 'm' instances X=$((\vec{x_1},y_1), ...,(\vec{x_m},y_m))$. We refer to the nth feature of mth element vector as $x_{m,n}$.

Definition 3.0.1 (Hamming distance) *Hamming distance between two instances is the count of the number of features that differ. (excluding the target feature)*

$$dist(\vec{x_i}, \vec{x_j}) = \sum_{1}^{n} \delta(x_{i,k}, x_{j,k}) \tag{1}$$

where δ(a,b)

1 if $a \neq b$
0 if $a = b$

For example the hamming distance between x1 and x4 in Figure 5 is two.

Definition 3.0.2 (Point of Class Change) *"A point of class change" (PoC) is defined as the closest instance with different class label. Let $(\vec{x_i},y_i)$ and $(\vec{x_j},y_j)$ be two labelled instance in X. $(\vec{x_j}, y_j)$ is called point of class change with respect to $(\vec{x_i}, y_i)$iff.*

$$dist(\vec{x_i}, \vec{x_j}) < dist(\vec{x_i}, \vec{x_z})$$

$$\forall\ z \neq j \in X \ \ and\ y_i \neq y_j$$

In Figure 5, the point of class change for x1 is x4.

Definition 3.0.3 (Neighborhood) *A neighborhood of an example $(\vec{x_i}, y_i)$ in instance space is a set of labelled instances $(\vec{x_{i1}}, y_{i1})$,..., $(\vec{x_{ik}}, y_{ik})$ such that*

$$dist(\vec{x_i}, \vec{x_j}) < \varepsilon \ \ \forall \ \ j = (i1), ..., (ik).$$

where

$$\varepsilon = dist(\vec{x_i}, \vec{x_{i'}})$$

$(\vec{x_{i'}}, y_{i'})$ is the point of class change for $(\vec{x_i}, y_i)$. Neighborhood can be thought of as a pure cluster (A pure cluster is one in which all the instances in the cluster belong to the same class). The instances in the neighborhood are called neighbors. In the example in Figure 5, there are 3 neighborhoods with members (x1,x2,x3),x4,x5 respectively.

Definition 3.0.4 (Strongly relevant [Blum and Langely, 1997]) *A feature f_i is strongly relevant to the concept if there are 2 examples $\vec{x_k}$ and $\vec{x_j}$ that differ only in their value of f_i and have different class labels. In Figure 5, the absolutely relevant feature is feature f3.*

Definition 3.0.5 (Irrelevancy of a feature) *Irrelevancy of a feature f_i is the number of pairs of examples $\vec{x_k}$ and $\vec{x_j}$ that differ only in their value of f_i and have the same class labels.*

$$Irrel(f_i) = \sum_j \sum_k \delta(x_{j,i}, x_{k,i}) * (1 - \delta(y_j, y_k))$$

where $\delta(a,b)$

$$1 \text{ if } a \neq b$$
$$0 \text{ if } a = b.$$

and $dist(\vec{x_j}, \vec{x_k})$=1. In the example given in Figure 5, the irrelevancy of feature f2 is one.

Definition 3.0.6 (Weakly relevant [Blum and Langely, 1997]) *A feature f_i is weakly relevant to the concept if it becomes strongly relevant on removal of few features. This means that all features f_k are weakly relevant if for some $\vec{x_i}$, $\vec{x_j}$ is the point of class change and $f_{i,k} \neq f_{j,k}$. In our example in Figure 5, the weakly relevant features are f1 and f2.*

Definition 3.0.7 (Incremental relevancy) *Blum and Langely (1997) define a feature that improves prediction accuracy on inclusion as incrementally useful. We extend the same to relevancy. A weakly relevant feature f_i is incrementally relevant if its more relevant than irrelevant. This means that the feature is in the strongly relevant feature set or appears more in the weakly relevant feature set than its irrelevancy.*

3.1 The SCRAP Feature Selection Algorithm:

SCRAP is a sequential search filter. SCRAP constructs pure clusters called 'neighborhoods'. The entire neighborhood is considered as a single search node for feature evaluation. Each neighborhood is uniquely identified by two points of class change. The first point of class change is where the neighborhood construction started and the second one is its termination point. The two instances at the point of class change are used for determining feature relevancy. The weak and the strong relevancy subsets are updated accordingly. The irrelevancy measure of the features is also updated. The search proceeds from the new point of class change and the process is repeated in until the entire search space is covered. On completion of this sequential search the entire instance space will be organized by nodes (neighborhoods) that are connected only to its adjacent node that belong to a different class. The edges are defined by the set of features that change between the two nodes.

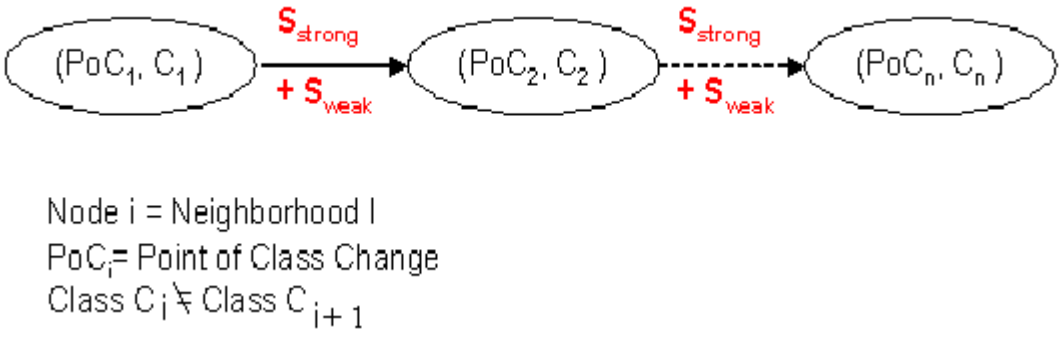

Figure 6: Search Space after SCRAP search

The essence of this feature selection scheme is that it tries to identify all features that are required to define the discriminating hypothesis to distinguish adjacent pure clusters.

All the features in the S_{strong} are selected. S_{strong} + S_{weak} are MIN-FEATURES along this search path. In order to strike balance between feature relevance for forming consistent hypothesis and features that increase prediction accuracy, features in the subset S_{weak} - S_{strong} are checked for incremental relevancy. The features that are in S_{strong} and those in S_{weak} that are incrementally relevant are selected.

3.1.1 SCRAP Algorithm

---

initialize the starting point of search with a random $\vec{x_i}$.
$S_{strong} = \phi$; $S_{weak} = \phi$; Irrelevancy(f_i)=0 ($\forall$ $f_i \in$ f)

while there are still training instances unmarked

form the neighborhood of $\vec{x_i}$ using unmarked examples and identify its point of class change $\vec{x_j}$. Mark $\vec{x_i}$ and all its neighbors

update S_{strong},S_{weak} and irrelevancy of all features

repeat the search with $\vec{x_j}$

$S_{increment}$= $S_{increment}$+f_i ; ($\forall$ $f_i \in S_{weak}$-S_{strong} & n($f_i \in S_{weak}$) > Irrelevancy(f_i))

S_{subset}=S_{strong}+$S_{increment}$

3.1.2 Example: XOR(f1,f3) Concept

Let us consider the XOR concept provided earlier and perform the search conducted by SCRAP on this instance space. SCRAP starts the instance space search with $\vec{x1}$. The point-of-class-change for $\vec{x1}$ is $\vec{x4}$. Features f1 and f2 become weakly relevant features

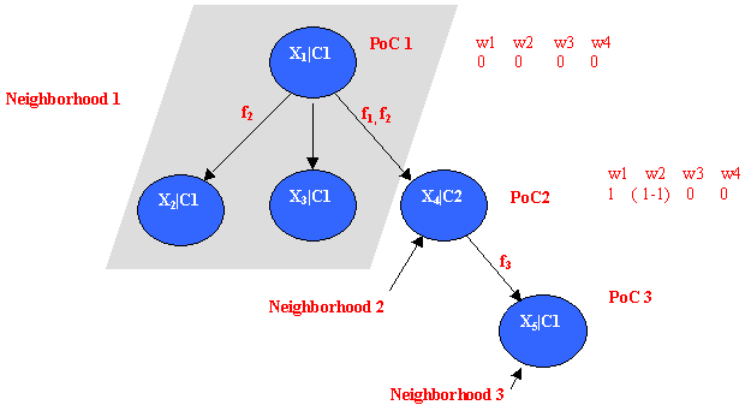

Figure 7: SCRAP's Instance Based Search on XOR(f1,f3) Concept

(S_{weak}={f1,f2}). SCRAP identifies irrelevance of f2 ($(\vec{x1}, \vec{x2})$) and increments irrelevance(f2). The instances (x_1, c_1),(x_2, c_1),(x_3, c_1) after becoming part of neighborhood 1 cannot become part of any other neighborhoods, and are excluded from further search. SCRAP resumes search with $\vec{x4}$ and finds new point-of-class-change as $\vec{x5}$. The feature f3 is identified as strongly relevant feature (S_{strong}={f3}). $\vec{x4}$ becomes neighborhood 2 and $\vec{x5}$ becomes neighborhood 3. $S_{increment}$={f1 } because irrelevancy(f2)=1. SCRAP returns features f1 and f3 as the selected features. Feature f4 did not show any evidence of relationship to the target class determination and hence is ambiguous feature and is discarded. Feature f2 was also filtered because there was nothing to gain on its inclusion.

3.1.3 Complexity Analysis

The neighborhood construction component of the SCRAP algorithm compares the features of each instance with the search instance. So the complexity of one execution of the inner loop is O(fn). The check for instance coverage (while loop) has worst case time of O(n). The total complexity is O(fn^2).

4. LASER Algorithm

Before we discuss the LASER algorithm we will present the intuition behind our example selection strategy.

4.1 Complementary Algorithms

The probability estimate for a non-parametric approach is given by,

$$p(x) = \frac{k}{NV}$$

where 'N' is the total number of examples, and 'V' is the Volume containing the 'k' neighbors. If value of 'V' is fixed and the corresponding 'k' is found then the problem becomes a Kernel Density Estimation problem. On the other hand, if value of 'k' is fixed and the

corresponding 'V' is found then it becomes k-nearest neighbor approach.

k-Nearest Neighbor Classifier

The non-parametric equation and the Bayes rule can be combined to derive one of the powerful instance based learning algorithms, namely k-nearest neighbors.

The probability estimate that the test example belong to the class 'y' is,

$$P(x \mid y) = \frac{k_i}{N_i V}$$

where 'N_i' is the number of training instances belonging to class 'y' and k_i is the number of examples with class label 'y' in volume 'V'.
The prior probability of class 'y' in the volume 'V' is,

$$P(y) = \frac{N_i}{N}$$

Substituting these probabilities in Bayes rule we get:

$$P(y \mid x) = \frac{\frac{k_i}{N_i V}\frac{N_i}{N}}{\frac{k}{NV}} = \frac{k_i}{k}$$

To minimize the probability of misclassification we must choose the class with largest k_i/k. This is k-nearest neighbor classification rule. It is clear that k-NN is a special selective case of Bayesian learner [Bishop, 1995].

The 1-nearest neighbor learner is a non-parametric approach that always performs local search. The 1-nearest neighbor algorithm tries to find evidence in proximity and uses that to categorize instances. The Nearest Neighbor learning algorithms introduce the bias of similarity.

Learners like naive bayes use global information for calculating the prior and conditional probabilities. This form of information gathering can be viewed as a search in global space. The naive bayes classifier introduces bias in assuming conditional independence. Probability of an unlabelled instance x belonging to class y is given by,

$$P(y|x) = P(y) \prod_{i=1..n} P(f_i|y)$$

Whereas 1-nearest neighbor uses one most similar instance in the instance space for class label determination, the naive bayes classifier uses all the instances in the training set. Observing the search, bias, example selection and their relation to Bayes rule, it is clear that the two learners complement each other. We will define them to belong to a new category of learning algorithms called 'complementary algorithms'. This forms the foundation of our LASER framework that exploits this complementary property of these two strategies.

4.2 LASER Framework

LASER is a embedded example selection method that consists of two main components, namely an example selection scheme and the target learner. The example selection scheme is made of two sub components. The example filter method identifies instances similar to the unlabelled instance to be classified. Similarity is measured in Hamming Space rather than the conventional Euclidean space.

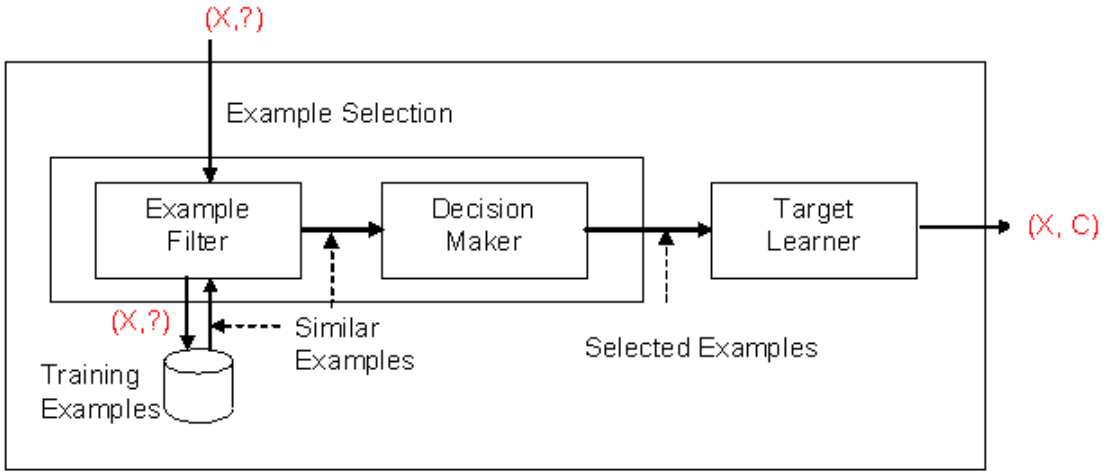

Figure 8: LASER Framework

Definition 4.2.1 (Min-Ham distance) *The Min-Ham distance for a unlabelled instance is the least distance in Hamming space where there is a neighbor. Suppose Min-Ham*$(\vec{x_i}) = \varepsilon$

$$HammingDistance(\vec{x_i}, \vec{x_j}) \geq \varepsilon, \qquad \forall \vec{x_j} \in X \tag{2}$$

A search ring of Min-Ham radius is formed in Hamming space with the example to be classified at its center. All the examples present on the ring are selected by the filter. There are three possible scenarios that arise on selection of these instances.

- Case #1 only one example is at Min-Ham distance.
- Case #2 all the examples (at Min-Ham distance) are of same class.
- Case #3 the selected examples (at Min-Ham distance) come from more than one class.

To evaluate our framework we used the naive bayes classifier. Case #1 shows a test example that is close to a singe training case than all others training examples. Case #2 indicates an instance that has a strong local knowledge. For case #1 and case #2 the examples in the search ring belong to a single class. Case #3 indicates an instance that falls near the decision boundary and needs the global probability distribution of the component classes to determine its class. LASER filters the case #1 and case #2 instances and returns the class of the search ring. This is in effect 1-NN and k-NN in Hamming Space. For case #3 instances, the naive bayes learner is used to predict the class label. LASER in effect uses hybrid examples selection strategy, that is one or k-closest examples for case #1 and case #2 and all examples for Case #3. The reason for choosing a local example selection strategy for case #1 and case #2 is because inclusion of other instances might include more irrelevant examples. The naive bayes is known to perform with degraded performance in presence of irrelevant or redundant instances. So a focused or selective approach is used for these instances. Where as for Case #3 instances the global distribution information is necessary to determine its class membership. Hence all training instances are selected.

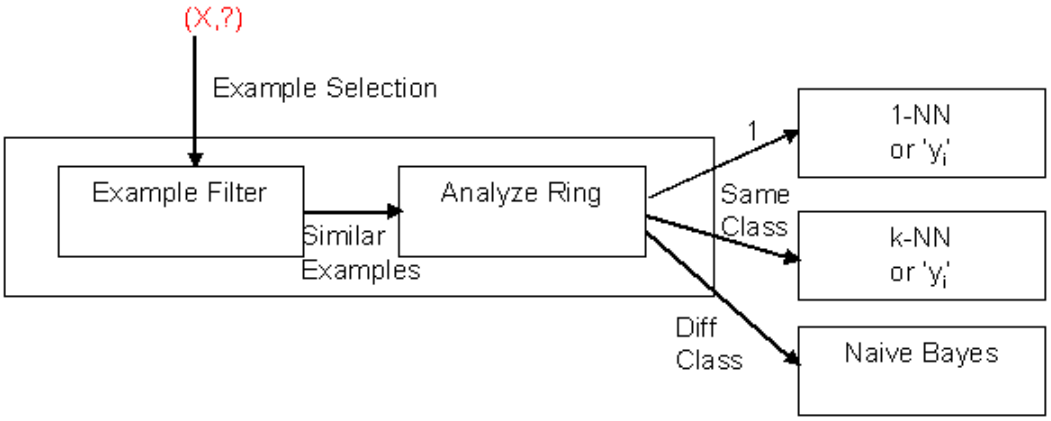

Figure 9: Hybrid Example Selection Strategy

LASER - Example Selection (unlabelled $\vec{x}$, training set X)

Example Filtering

form a search ring of radius $r_{Min-Ham}$.

select all the labelled instances $X_{selected}$ $(x_i, y_j) \in$ (X,Y) such that h(x,x_i)=Min-Ham Distance.

pass the selected examples $X_{selected}$ to the Case-Based Reasoner

Decision Maker(Search Ring)

Case #1 and Case #2 return y_i.

Case #3 select all training instances X.

Naive Bayes Classifier(Selected Examples).

---

5. Results

This section presents the empirical results of our feature selection (SCRAP) and example selection (LASER) schemes. A detailed discussion of the results is presented in the next section. In the first section we present the results of SCRAP feature selection and compare the performance of three families of learning algorithms with/without feature selection. In the second section we show the performance of LASER and compare it with the nearest neighbor and the naive bayes classifiers. Finally we combine the feature selection and example selection approaches and study the benefits of focused learning to the naive bayes learner.

Dataset	Inst. Space	No.of. Feat.	S_{strong}	Irrel	S_{weak}	No.of Neigh -borhoods
breast cancer	217	9	3.875	5.75	3.375	29.375
credit	653	15	0	0.25	15	50.125
diabetes	768	8	0	0	8	93.5
DNA	106	57	0	0	57	10.875
glass	214	9	0	0.125	9	25.125
heart	296	13	0	0	13	25.25
iris	150	4	1	2.375	2.75	28
rotate	1000	2	1	1	1	105
voting	435	16	1.25	12.875	13.375	16.625
>= 3-of-6-of-13	1000	12	4.33	11.67	8	63.33
= 3-of-6-of-13	1000	12	4.375	11.125	8.25	112.75
= 5-of-10-of-13	1000	12	7.675	11.75	4.75	94.625

S_{strong} - Strongly Relevant Features.
S_{weak} - Relatively Relevant Features.
Irrel - # Features with Irrelevance measure > 0.

Table 1: SCRAP-Statistics

Dataset	All	SCRAP	RELIEF
breast cancer	9	4.88	5.25
credit	15	15	11.875
diabetes	8	8	6.63
DNA	57	56	47.13
glass	9	9	5.63
heart	13	13	10.25
iris	4	2.75	3.88
rotate	2	2	1.5
voting	16	4.5	12.14
>= 3-of-6-of-13	12	5.33	8.38
= 3-of-6-of-13	12	5.75	7.25
= 5-of-10-of-13	12	8.625	8.625

Table 2: Feature Selection Details

5.1 SCRAP Feature Selection

For evaluating SCRAP, we used nine real domains mostly from UCI Machine Learning Repository [Blake and Merz] and three artificial data sets. Langley and Sage (1994b) indicate that many of the datasets in UCI repository have just a few completely irrelevant features and no complex feature interactions.

We also used three artificial data sets that were either M-of-N concepts or X-of-N concepts [Murphy and Pazzani, 1991]. The three artificial data sets used for study were at least 3-of-6 among 13 features (>=3-of-6), exactly 3-of-6 among 13 features (=3-of-6) and exactly 5-of-10 among 13 features (=5-of-10). These domains have many irrelevant features and the "exactly" concepts represent a non linearly separable problems [Murphy and Pazzani, 1991].

Table I, shows the different elements of the SCRAP feature selection algorithm. Ten runs with one-third training instance and one-third testing set was used to validate the results. It can be seen that real domains have many weakly relevant features rather than strongly relevant or totally irrelevant features as noted by Langley and Sage (1994b). The number of neighborhoods denotes the pure clusters that are possible on a sequential search.

The SCRAP filter did not reduce feature space on five real domains namely credit, diabetes, glass, heart and rotate data sets (see Table II). All the features were found to be relevant to defining consistent hypotheses. This is an important observation as RELIEF filter did reduce features on these data sets, which reduced the prediction accuracy of many learning algorithms.

5.2 Performance with learning algorithms

We compared the generality and relevancy to prediction accuracy of the feature subsets generated by SCRAP and RELIEF. This was done by comparing their performance on three families of learning algorithms namely Instance Based Learners (k-Nearest Neighbor), Symbolic and Statistical learners (naive bayes and Decision Trees), Connectionist Models (Back-Propagation Neural Network).

5.2.1 k-Nearest Neighbors

The Nearest Neighbor classifier [Hart and Cover, 1967] is a non-parametric approach that selects 'k' similar examples for determining the classification of the unlabelled instance. Instance based schemes like the Nearest Neighbor show degraded performance when there are many low relevance features [Aha and Bankert, 1995]. We tested the selected feature subset on 1-Nearest Neighbor.

SCRAP improved performance of 1-nearest neighbor on two real domains and two artificial domains (see Table III). In comparison, RELIEF improved prediction accuracy on four real domains and one artificial domain but reduced the accuracy on six data sets. Hence we can conclude that SCRAP was the more stable feature selection method amongst the two for this class of learners. SCRAP improved the prediction accuracy of 1-NN on both real and artificial domains.

Dataset	1-NN	1-NN +SCRAP	1-NN +RELIEF
breast cancer	72.75	72.125^{-}	72.375^{-}
credit	83.125	nfr	79.57^{-}
diabetes	69.25	nfr*	69^{-}
DNA	67.75	71.375^{+}	71.5^{+}
glass	71.125	nfr	67.25^{-}
heart	75.375	nfr	77.875^{+}
rotate	72.125	nfr	62^{-}
voting	91.75	93^{+}	93.5^{+}
>= 3-of-6-of-13	81.625	88.25^{+}	88.625^{+}
= 3-of-6-of-13	73.5	92.5^{+}	68.25^{-}
= 5-of-10-of-13	77.50	76.75^{-}	74.625^{-}
Average Real	75.41	75.94^{+}	74.13^{-}
Average Artif	77.54	85.83^{+}	77.17^{-}

nfr - no feature reduction.
+ = better than 1-NN, - = worse than 1-NN.

Table 3: k-Nearest Neighbor Learner kNN (k=1),with/without Feature Selection using SCRAP and RELIEF filters.

5.2.2 Symbolic and Statistical Learners

We used two well known Symbolic and Statistical classifiers namely the ID3 decision trees [Quinlan, 1993] and the naive bayes learner [Mitchell, 1997] for evaluation.

Decision Trees

The ID3 algorithm [Quinlan, 1993] was designed to build the decision tree. The ID3 algorithm has bias to produce the smallest tree structure that explains the training set consistently[Caruana and Freitag, 1994]. In many real domains the relevant feature set selected by SCRAP and RELIEF were larger than the subset selected for constructing the tree by the ID3 learner (Table IV). Hence the ID3 classifier did not gain much from feature selection. But on the artificial domains with large number of irrelevant features, this was not the case. The ID3 tree was larger than the selected subset, and therefore contained irrelevant features. ID3 predicted with higher accuracy when trained on the subset generated by SCRAP. RELIEF was not able to improve ID3 on either real or artificial domains.

ID3 over-fits the data due its inherent bias. This can be rectified by pruning the ID3 tree [Elomaa and Kaariainen, 2001]. We used the reduced error pruning method for pruning the tree. Feature selection and reduced error pruning improved performance on both real and artificial domains as can be seen in Table V. Esposito et al (1997) note that reduced error pruning produces the most accurate subtree with respect to the training set. This augmented with relevant feature set identified by SCRAP produced the optimal prediction

Dataset	# feat*	ID3 % acc	# feat	ID3 +SCRAP	# feat	ID3 +RELIEF
breast cancer	7.125	71.5	4.625	73.25^{+}	4.75	71.75^{+}
credit	8	84.285	8	nfr	8	86.143^{+}
diabetes	7.6	76.4	7.6	nfr	6.625	75.25^{-}
DNA	3.25	79.875	3.0	77.125^{-}	2.875	75.75^{-}
glass	5.125	84.125	5.125	nfr	3.75	82.5^{-}
heart	8.75	79.625	8.75	nfr	7.625	79.938^{+}
iris	1.875	96.125	1.25	94.25^{-}	1.75	94.75^{-}
rotate	1.75	94.75	1.75	nfr	1.75	95.313^{+}
voting	4.125	95.375	3.875	94.125^{-}	6.875	94.625^{-}
>= 3-of-6-of-13	8.625	95.75	5.25	97^{+}	6.5	94.625^{-}
= 3-of-6-of-13	11.75	70.75	5.375	79.25^{+}	6.5	66.5^{-}
= 5-of-10-of-13	9.75	74.375	8.25	74.89^{+}	7.625	73.625^{-}
Average Real		84.67		84.22^{-}		84.00^{-}
Average Artif		80.29		83.71^{+}		78.25^{-}

feat - is the # distinct features in the decision tree.
+ = better than ID3, - = worse than ID3.

Table 4: ID3 with/without Feature Selection using SCRAP and RELIEF filters.

Dataset	# feat	ID3+REP	# feat	ID3+REP +SCRAP	# feat	ID3+REP +RELIEF
breast cancer	7	68.875	4.5	73.75^{+}	4.75	74.375^{+}
credit	8	86.71	8	nfr	8	86.42^{-}
diabetes	7.6	76.1	7.6	nfr	5.625	76.125^{+}
DNA	3.25	77	2.875	78.25^{+}	2.875	82.5^{+}
glass	5	83.875	5	nfr	3.75	83^{-}
heart	8.625	79.5	8.625	nfr	7.625	81^{+}
iris	1.875	94.75	1.25	96.25^{+}	1.75	95.875^{+}
rotate	1.75	94.75	1.75	nfr	1.75	95.5^{+}
voting	4.125	96.25	3.875	95.125^{-}	6.625	95.125^{-}
>= 3-of-6-of-13	8.735	96.125	5.25	96.875^{+}	6.5	94.375^{-}
= 3-of-6-of-13	11.75	74.25	5.375	80.375^{+}	6.625	70.125^{-}
= 5-of-10-of-13	9.5	75.25	8.25	75.25	7.625	74.25^{-}
Real		84.20		84.92^{+}		85.55^{+}
Artificial		81.88		84.17^{+}		79.58^{-}

feat - is the # distinct features in the pruned decision tree.
+ = better than ID3 with Pruning, - = worse than ID3 with Pruning.

Table 5: Pruned Decision Trees with/without Feature Selection using SCRAP and RELIEF filters.

accuracy for ID3.

Naive Bayes Learners

The naive bayes learner is a simple learning scheme based on the Bayes Rule. Langley and Sage (1994) have shown that the naive bayes classifier is affected by presence of irrelevant features and redundant information. This case was observed in our results too (Table VI). Langely and Sage (1994) also note that selective or focused approach (feature selection) improves naive bayes low asymptotic accuracy, on domains with redundant features like the voting data set. On domains where naive bayes already performs well like DNA and breast cancer such efforts does not produce any significant returns. Table VI shows this property of naive bayes. Naive Bayes learner benefited from using only the selected feature subset on the artificial domains with large number of irrelevant features. This clearly shows that focused or selective approaches are required to improve the naive bayes classifier on domains with redundant or irrelevant information.

5.2.3 Neural Networks

There are different architectures of Neural Network like the single layer model or Perceptron [Rosenblatt, 1962], the multi-layer model [Rumelhart et al., 1986] etc. We used a back-propagation neural network with one hidden layer and learning rate being 0.1 for evaluating our feature selection scheme. Neural networks was able to handle the problem of noisy information well(Table VII). The multi-layer network did not benefit from feature selection on the real domains. But on artificial domains with many irrelevant features SCRAP was able to improve the prediction accuracy of back-prop network. Table VII shows considerable improvement to prediction accuracy on at least and exactly 3 out of 6 amongst 13 features concepts. This suggests the importance of feature selection even to stable learning algorithms on domains with many irrelevant features.

5.3 Summary

A comparison of prediction error by various families of algorithms on using the feature sets generated by the SCRAP and RELIEF filters is presented in Table VIII. The δ values are all positive indicating that SCRAP was the better feature selection scheme on all the three classes of learning algorithm.

SCRAP clearly out performed RELIEF for generality of the feature subset produced. By generality, we mean the feature set being independent of any particular learning algorithm. SCRAP feature subsets improved the prediction accuracy of the different families of classifiers more than the RELIEF filter.

Table IX shows the statistical significance of the results presented. The paired t-tests were conducted to test the significance of difference in prediction accuracy between doing feature selection and not doing feature selection and between SCRAP and RELIEF. We used the average performance of learning algorithms 1-NN, ID3 with/without reduced error pruning, naive bayes and back propagation networks on each of the real and the artificial domains to compute the 2-tailed p-Values for determining the significance of difference. The tests show that SCRAP performed significantly better than RELIEF and doing no feature

Dataset	NB	NB +SCRAP	NB +RELIEF
breast cancer	74.46	74.19^{-}	73.12^{-}
credit	78.73	nfr	79.42^{+}
diabetes	74.51	nfr	76^{+}
DNA	79.76	76.37^{-}	72.56^{-}
glass	71.52	nfr	69.96^{-}
heart	82.25	nfr	82.16^{-}
iris	96.06	93.63^{-}	90.45^{-}
rotate	91.92	nfr	90.93^{-}
voting	91.53	93.32^{+}	90.93^{-}
= 3-of-6-of-13	92.52	97.606^{+}	93.70^{+}
= 3-of-6-of-13	68.81	69.46^{+}	68.81
>= 5-of-10-of-13	73.50	76.31^{+}	75.30^{+}
Average Real	82.30	81.83^{-}	80.61^{-}
Average Artif	78.28	81.13^{+}	79.27^{+}

Table 6: Naive Bayes Learner (NB) with/without Feature Selection using SCRAP and RELIEF filters.

Dataset	MNN	MNN +SCRAP	MNN +RELIEF
breast cancer	85.82	80.95^{-}	85.17^{-}
credit	90.20	nfr	89.78^{-}
diabetes	82.37	nfr	70.65^{-}
DNA	90.57	93.40^{+}	92.69^{+}
glass	83.91	nfr	83.57^{-}
heart	69.93	nfr	75.76^{+}
iris	85	84.85^{-}	81.25^{-}
rotate	92.23	nfr	91.43^{-}
voting	93.45	93.27^{-}	92.75^{-}
>= 3-of-6-of-13	91.01	92.23^{+}	91.43^{+}
= 3-of-6-of-13	69.12	71.52^{+}	69.47^{+}
= 5-of-6-of-13	81.23	79.91^{-}	78.03^{-}
Average Real	85.91	85.69^{-}	84.78^{-}
Average Artif	80.45	81.22^{+}	79.64^{-}

Table 7: A Multi-Layer Neural Network(MNN), with/without Feature Selection using SCRAP and RELIEF filters.

Class	Data sets	Base Accuracy	SCRAP	RELIEF	δ^*
Instance Based	Real	75.41	75.94	74.13	+1.80
Learners	Artif	77.54	85.83	77.17	+8.67
Symbolic	Real	83.72	83.66	83.39	+0.27
Statistical Learners	Artif	80.15	83.00	79.03	+3.97
Neural Networks	Real	85.91	85.68	84.78	+0.90
	Artif	80.45	81.22	79.64	+1.58
Average	Real	82.50	82.52	81.81	+0.71
	Artif	79.69	83.21	78.78	+4.43

δ = SCRAP - RELIEF.
Base Accuracy= Accuracy without feature selection.

Table 8: SCRAP vs RELIEF

selection on artificial domains. But on real domains there was no significant differences between doing and not doing feature selection. The Cronbach's reliability coefficients gives how reliable can a conclusion be made using the data. For reliability, the data should have Cronbach alpha score greater than 0.7 for their data.

Data sets	no feature set reduction(nfr)	RELIEF	SCRAP	pVal SCRAP-nfr	pVal SCRAP-RELIEF	alpha
Real	82.50	81.81	82.52	.981	.146	.9832
Artif	79.69	78.78	83.21	.019	.021	.9521

Table 9: Significance of the results

5.4 LASER Example Selection

The same data sets used for evaluating the feature selection algorithm were used for evaluating LASER. Table VIII presents the information about these domains using a 4-tuple (#Inst,#Feat,#Disc,#Cont) which gives the number of instances in the instance space, dimensionality of the domain, number of discrete features and number of continuous features respectively.

Table X summarizes performance of 1-NN, naive bayes (NB), and LASER on benchmark data sets. 1-NN always selects one closest example for determining the class label. The naive bayes learner always uses the entire training space for classification. LASER is hybrid between these two example selection approaches. So, we compare the three approaches namely 1-nearest neighbor, naive bayes and LASER to demonstrate the robustness of our approach. δ_{NB} shows the difference between the predictive accuracy of LASER and naive bayes algorithms. pVal(LASER-NB) gives the significance of the difference between the means of LASER and naive bayes scheme at 95% significance. Ten runs of each algorithm on the data set was used in determining the significance of the difference. This was done by paired-t testing. LASER out-performed 1-NN on all domains. The average difference of

Dataset (Inst,Feat, #Disc,#Cont)	Majority Class	kNN (k=1)	NB	LASER	δ_{NB}	pVal LASER-NB
breastcancer (217,10,10,0)	70.76%	72.75	73.11	78.8^{+}	5.69	0.000
credit (653,16,10,6)	54.67%	83.125	78.30	85.3^{+}	7.0	0.000
diabetes (768,9,0,9)	65.1%	69.25	73.89	78.59^{+}	4.7	0.017
DNA (106,58,58,0)	50%	67.75	73.33	77.71^{+}	4.38	0.235
glass (214,100,0,10)	59.35%	71.125	63.89	76.34^{+}	12.45	0.000
heart (296,14,8,6)	54.05%	75.375	81.41	82.86^{+}	1.45	0.362
iris (150,5,0,5)	33.33%	85	95.69	92.8^{-}	-2.89	0.140
rotate (1000,3,0,3)	76.51%	72.125	91.5	90.57^{-}	-0.93	0.195
voting (435,17,17,0)	61.38%	91.75	90.93	93.51^{+}	2.58	0.349
>= 3-of-6-of-13 (1000,14,14,0)	63.1%	81.625	94.04	92.64^{-}	-1.4	0.152
= 3-of-6-of-13 (1000,14)	68.8%	68.81	67.24	77.06^{+}	9.82	0.000
= 5-of-10-of-13 (1000,14,14,0)	75.8%	77.50	77.233	78.23^{+}	0.9	0.447
Average Real		76.472	80.227	84.053	3.825	
Average Artif		75.978	79.504	82.643	3.139	

+ = better than Naive Bayes, - = worse than Naive Bayes.
p-Val $< .05$ indicates significant difference.

Table 10: Naive Bayes Learner (NB)& LASER performance on benchmark datasets

Dataset	case1%	Acc# case1	case2%	Acc# case2	case3%	Acc# case3
breast cancer	47.17	84.56	7.5	76.81	45.32	73.14
credit	50.27	88.82	6.86	75.84	42.86	82.69
diabetes	41.83	82.63	6.95	66.85	51.25	76.82
DNA	72.57	82.6	6.57	60.87	20.85	65.75
glass	39.58	86.12	9.85	77.14	50.56	68.52
heart	48.87	84.97	6.4	63.49	45	82.7
iris	49.2	91.87	10.6	86.79	40.2	95.5
rotate	38.13	89.13	4.17	90.65	47.69	91.94
voting	44.07	94.2	1.8	88.89	54.2	92.88
at least 3	46.06	91.91	8.2	86.45	45.7	94.5
exactly-3	46.69	86.69	9.1	74.25	44.2	67.46
exactly-5	46.27	82.54	9.7	71.91	44	75.08

Table 11: Case-Wise Performance of LASER.

Dataset	NB C1	NB C2	LASER C1	LASER C2
breast cancer	79.32	53.37	85.86	58.87
credit	82.76	76.21	87.14	84.02
diabetes	78.72	63.88	83.65	68.03
DNA	77.59	69.35	80.95	73.91
glass	71.46	54.26	80.21	72.02
heart	82.11	80.89	85.68	79.31
rotate	95.68	78.97	94.21	79.58
voting	93	90.13	91.01	95.17
>= 3-of-6-of-13	98.88	91.74	93.14	91.95
= 3-of-6-of-13	67.89	43.2	78.32	72
= 5-of-10-of-13	80.12	47.5	80.77	62.05

C1 - Majority Class
C2 - Minority Class

Table 12: Class-wise accuracy of Naive Bayes Learner (NB)& LASER.

prediction accuracy on real domains was 7.58% and on artificial domains was 8.13%. LASER performed better than naive bayes on all but 3 data sets, but these were not significant and the accuracy of both the schemes on those data sets were above 90%. The overall predictive accuracy of LASER was 3.8% more than naive bayes on real domains and 3.1% more on artificial domains.

Table XI gives the percentage of instances that were classified into different cases of example selection and the prediction accuracy of the on those cases. It can be seen that case 2 is infrequent than the other two cases. The test instances that are closer to one training instance than all others in Hamming space are more likely to have the same class label. This can be clearly seen in Table XI that there is a consistent high accuracy of prediction for case1 instances. The case3 presents the more tough classification instances as they fall near the decision boundary. The performance of naive bayes learner on these instances where global probability distribution of classes is required for classification is better on many domains than naive bayes learner performance on all instances. This indicates that naive bayes performance must reduces on case1 and case2 examples due to irrelevant examples that reduces the over all performance and LASER hybrid strategy has helped solve this problem.

Table XII gives the accuracy of predicting each class label by naive bayes and LASER. The case-wise accuracy clearly indicates that the success of the example selecting strategy. Further explanation of how LASER helped naive bayes handle the problem of class imbalance is presented in the next section.

5.5 Summary

The LASER example selection strategy improved the performance of naive bayes on domains with known irrelevant features (that is the artificial domains). LASER was also able to boost naive bayes Performance on the real domains suggesting the presence of redundant information or irrelevant information. Table XIII shows the results on one-tailed hypothesis testing and Cronbach's reliability coefficient (α). A p-Val of .0096 indicates that the probability of naive bayes performing better than LASER is less than 1%. It is clear from the Table XIII that LASER outperformed the naive bayes and the nearest neighbor approaches.

Comparision	p-Val	Reliability α
NB vs LASER	0.0096	0.9295
1-NN vs LASER	0.0001	0.8746

p-Val $< .05$ indicates significant difference.

Table 13: Significance of Results

6. Combining Feature Selection (SCRAP) and Example Selection (LASER)

We combined the SCRAP feature selection algorithm and LASER example selection scheme to improve the performance of the naive bayes learner. Table XIV presents a comparison of naive bayes with/without SCRAP feature selection or/and LASER example selection. Feature selection improved the performance of naive bayes on domains with irrelevant features. Example selection improved performance of naive bayes on both real and artificial domains. The combined approaches of SCRAP and LASER yielded the maximum improvement to the prediction accuracy of naive bayes learner demonstrating the benefits of selective learning. Even ID3 and Neural Networks get at most 80% accuracy on the artificial domains with/without feature selection. LASER was able to comprehensively outperform these approaches by following a focused learning approach on these domains with many irrelevant features.

Dataset	NB	NB +SCRAP	LASER	LASER +SCRAP	pVal Combined-LASER
breast cancer	74.46	74.19	78.8	81.41^{+}	0.117
credit	78.73	78.73	85.3	85.3	-
diabetes	74.51	74.51	78.59	78.59	-
DNA	79.76	76.37	77.71	87.71^{+}	.026
glass	71.52	71.52	76.34	76.34	-
heart	82.25	82.25	82.86	82.86	-
iris	96.06	93.63	92.8	93.8^{+}	.475
rotate	91.92	91.92	90.57	90.57	-
voting	91.53	93.32	93.51	95.52^{+}	.069
>= 3-of-6-of-13	92.52	97.606	92.64	97.87^{+}	.00
= 3-of-6-of-13	68.81	69.46	77.06	99.52^{+}	.00
= 5-of-10-of-13	73.50	76.31	78.23	90.42^{+}	.001
Real	82.30	81.83	84.053	85.79^{+}	
Artificial	78.28	81.13	82.643	95.94^{+}	

Combined - LASER + SCRAP
$+$ = better than LASER.
p-Val $< .05$ indicates significant difference.

Table 14: A comparison of Naive Bayes Learner (NB)with/without Feature Selection(SCRAP) & Example Selection (LASER).

7. Discussion

In this section we present a detailed analysis of the results presented in the previous section. We use the results to analyze the behavior of stable learning algorithms on feature selection. We present a detailed study of LASER's solution to the class imbalance problem.

7.1 Balancing Hypothesis Consistency and Prediction Accuracy

The SCRAP feature selection procedure arranges the instance space as pure clusters (all instances belonging to same class). The feature subset required to differentiate adjacent pure clusters of different class labels are identified and classified as strongly relevant or weakly relevant. Some of the weakly relevant features help defining fewer hypothesis and are not incrementally relevant. Removing such features provides the necessary tradeoff between defining consistent hypothesis and improving prediction accuracy.

The only bias introduced by the SCRAP sequential search filter is the MIN-FEATURE bias [Almuallim and Dietterich, 1991]. The feature subsets selected by SCRAP were found to be general. By general we refer to being beneficial across multiple families of learning algorithms. This is because SCRAP presents features that hold the information for identifying the hypotheses and is algorithm independent. Higher accuracy can be achieved by selecting the features that suit the bias of the learning algorithm. This is done by the Wrapper approaches. But this comes at the cost of losing generality of the feature subset.

7.2 Performance of learning algorithms with Feature Selection

We had presented earlier, the prediction accuracy of different learning algorithms with/ without feature selection. The results present an interesting foundation for a discussion on the benefits of feature selection realized by different learning algorithms. We will use the SCRAP's results for this analysis.

7.2.1 1-Nearest Neighbors

The nearest neighbor schemes use every feature in the feature set for finding the closest neighbors. If there are more irrelevant features, the performance of the k-nearest neighbors is going to be affected severely. On real domains that had no totally irrelevant features, feature selection selects the incrementally useful [Blum and Langely, 1997] features. Due to this reason the nearest neighbor approach was only marginally improved on these domains. But on domains with many irrelevant features (as in the artificial domains), feature selection can be expected to improve the performance by a larger margin. The performance of 1-nearest neighbor on the three artificial domains that had many irrelevant features indicates the utility of feature selection to nearest neighbor approaches.

7.2.2 Decision Trees

The ID3 algorithm build the smallest decision tree that is consistent with the training examples and introduces the MIN-FEATURE Bias[Almuallim and Dietterich, 1991]. This bias of ID3 is also called the Smallest Tree bias [Caruana and Freitag, 1994]. The feature subset selection scheme must return a smaller subset (subset of those selected by ID3) to produce a different decision tree. On real domains where there were no irrelevant features, the feature selection scheme usually returns all the features. Hence the same decision tree was built and there was no improvement to the performance. But on the artificial domains the count of distinct features in the decision tree was larger than the feature subset selected by the SCRAP. The prediction accuracy of ID3 on the three artificial domains improved on feature selection.

7.2.3 The Naive Bayes Learner

The naive bayes learner assumes the existence of a single probability distribution for each class which is sufficient for identifying the membership of each instance and that the features are independent of each other [Langely and Sage, 1994].

The real domains chosen for our experiments were mostly from UCI repository. Many of these domains were noted to be with almost no irrelevant features or complex feature interactions [Langely and Sage, 1994b]. SCRAP was able to improve the performance of naive bayes on the voting data set that has been shown to have redundant features [Langely and Sage, 1994b]. The artificial domains did present both irrelevant features and feature interactions. The naive bayes learner improved its overall performance by 3% on these domains by using the SCRAP filter for feature selection. This validates the fact that the naive bayes classifier is affected by irrelevant and redundant attributes [Langely and Sage, 1994b].

7.2.4 Back Propagation Network

The back propagation neural network proposed by Rumelhart et al. (1986) is a robust model capable of handling noisy information effectively. Bishop (1995) notes that multilayer model of neural networks can be used for dimensionality reduction. Therefore, we might expect Neural Network them to benefit least from feature selection. Table VII shows that the Neural Networks can benefit through feature selection on domains with many irrelevant features, though less so than other algorithms like naive bayes.

7.3 The Class Imbalance Problem

The class imbalance problem corresponds to domains for which one class is represented by a large number of examples while the other of represented only by a few [Japkwicsz, 2000]. Earlier works to overcome this problem includes *re-sampling*, in which the smaller class is re-sampled until there is an equivalent number of instances as the major class, *down-sizing* which removes the instances of majority class until the representation is balanced, and learning by recognition in which a multi-layer perceptron is made to recognize either the majority class or the minority class [Japkwicsz, 2000].

Japkowicz (2000) has shown that the connectionist systems have degraded performance on domains with class imbalance. In the previous section we have shown that the naive bayes learner tends to predict the minority class labels with lower prediction accuracy on such domains.

LASER presents a solution to this problem by adapting the example selection strategy,that is by selective learning. LASER specifically improved the predictive accuracy in the minority class to achieve a gain in overall predictive accuracy. The class-wise prediction accuracy of both LASER and naive bayes show that there is a huge difference (more than 10%) in the class-wise predictive accuracy in breast cancer, diabetes, glass, exactly and exactly(2) data sets. The majority classifiers of these data sets were around 70% (except glass). The δ value for these data sets in Table IX was also high. The minority class instances in these highly imbalanced domains will be wrongly classified if global information is used. This is exactly where a hybrid approach like LASER can comprehensively outperform

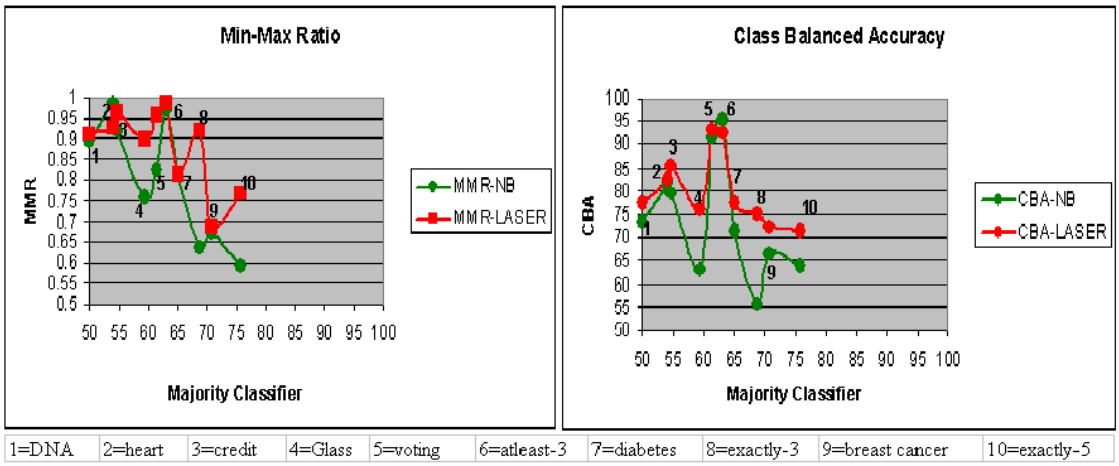

Figure 10: Predictive Measure a. MMR b.CBA

the the traditional scheme due to its adaptability to different example selection strategy at different regions of instance space.

We define two prediction quality metrics to evaluate performance for class imbalanced domains.

Definition 7.3.1 *The Min-Max Ratio (MMR) is the ratio between the Minority Classifiers (Min) predictive accuracy to Majority Classifiers (Max) predictive accuracy.*

$$MMR_i = \frac{MinorityClass(y_{c_i})Accuracy\%}{MajorityClassAccuracy\%} \tag{3}$$

MMR for a classifier, which predicts the entire instance accurately for every class, is 1.0 and which predicts all instance of a single class alone is 0. The higher the MMR, the more the learner is consistent in classifying the instance belonging to various class labels in instance Space.

Definition 7.3.2 *The Class Balance Accuracy (CBA) is the measure of predictive accuracy when corrected for an equal distribution for all class labels in the instance space.*

$$CBA = 1/n \sum_{i=1..n} Accuracy(y_i) \tag{4}$$

where Accuracy(y_i) returns % accuracy for class label $y_i \in Y$.

Figure 11 shows the two metrics MMR and CBA against the majority classifier percentage. The label identifies the data set, its majority class percentage(x-axis) and MMR or CBA value (y-axis) of LASER and naive bayes on that domain. Both MMR and CBA are negatively correlated with the majority classifier. It is clear that the percentage of correctly predicted minority class instances decreases with increase in percentage of majority class. Also, the quality of prediction (CBA) drops with increase in majority classifier's proportion. LASER was able to improve the minority class prediction and maintain the CBA much higher than naive bayes.

7.4 Comparison of Feature Selection and Example Selection

Feature selection (SCRAP) is necessary to improve the performance of all categories of learning algorithms on domains with many totally irrelevant features. On domains with no or few irrelevant features, feature selection does not provide any significant improvement to learning algorithms accuracy. The Examples selection scheme (LASER) was able to improve the performance of the learning algorithm on domains with many irrelevant features and domains with no or few irrelevant features. Table XIII shows that a combined approach of feature and example selection produced the optimal results for the naive bayes learner amongst the three approaches and the simple naive bayes learner.

8. Conclusion

We have presented the SCRAP feature selection algorithm and the LASER example selection scheme. Our feature selection approach is based on the assumption that features that are relevant to defining consistent hypotheses locally along a sequential search path will be a close approximation of the MIN-FEATURES. Also checking for incremental relevancy of weakly relevant features is an acceptable tradeoff between defining consistency and improving prediction accuracy. The SCRAP filter was able to remove most of the irrelevant features on the artificial domains, which had many irrelevant features. SCRAP produced general data sets that were able to improve the performance of three families of learning algorithms: Instance-based learners, Statistical and Symbolic learners and Neural Networks. The limitation of this approach is that the feature subset produced is often large and on few real domains it returned the entire feature set as relevant.

Our LASER example selection scheme presents the other aspect of selective learning. The example selection component was used with naive bayes classifier. LASER was able to improve the prediction accuracy of naive bayes on both real and artificial domains significantly. Analysis suggested that this helps mitigate problems especially in domains with class imbalance. The only limitation of LASER is that its decision strategy is target-learner specific and needs to be modified to suit different learners.

We showed the benefits realized by the components of the selective learning mechanism namely selective attention and selective utilization. We combined the both these schemes to observe their combined effect on naive bayes. The results suggest that the naive bayes learner benefited significantly by taking a focused or selective approach.

References

[Aha et al, 1991] Aha,D., Kibler, D., and Albert, M.K. Instance based learning algorithms. In Machine Learning, volume6, pages 37-66, 1991.

[Aha, 1998] Aha, David, W. Feature Weighting for Lazy Learning Algorithms. In Feature Extraction, Construction, and Selection . A Data Mining Perspective, Huan, Liu and Hiroshi, Motoda (eds.). pages 13-32, ISBN: 079238198X, Kluwer Academic Publisher, 1998.

[Aha and Bankert, 1995] Aha, D. W. and Bankert, R. L., A comparative evaluation of sequential feature selection algorithms, In Proceedings of the Fifth International

Workshop on Artificial Intelligence and Statistics, editors D. Fisher and H. Lenz, pp. 1–7, Ft. Lauderdale, FL.

[Almuallim and Dietterich, 1991] Almauallium, H. and Dietterich, T.G. Learning with many Irrelevant Features, In Proceedings of Ninth National Conference on Artificial Intelligence, Vol.2, AAAI Press, pages 547-552, Anaheim, CA, 1991.

[Bishop, 1995] Neural Networks for Pattern Recognition, Oxford University Press, ISBN 0-19-853864-2, 1995.

[Blake and Merz] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science.

[Blum, 1994] Blum, Avrim. Relevant Examples and Relevant Features: Thoughts from Computational Learning Theory, In AAAI Fall Symposium, pages 18-22, Technical Report FS-94-02, ISBN 0-929280-76-8, 1994.

[Blum and Langely, 1997] Blum, Avrim and Langely, Pat. Selection of Relevant Features and Examples in Machine Learning, In Artificial Intelligence, Vol.97, No.1-2, pages 245-271,1997.

[Cahuvin and Rumelhart, 1995] Chauvin, Y., and Rumelhart,D. BACKPROPAGATION: Theory, architecture and applications(edited collection). Hillsdale, NJ, Lawerence Erlbaum Assoc.

[Caruana and Freitag, 1994] Caruana, Rich and Freitag, Dayne, Greedy Attribute Selection, The Proceedings of the 11th International Conference on Machine Learning, 1994.

[Clardie, 1993] C. Cardie. Using Decision Trees to Improve Case-Based Learning. In Proceedings of the Tenth International Conference on Machine Learning, 25-32, Amherst, MA, 1993. Morgan Kaufmann.

[Cohn etal.,1995] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. Active learning with statistical models. In Tesauro, G., Touretzky, D., and Alspector, J., eds., In Advances in Neural Information Processing, Volume 7. pages 705-712, Morgan Kaufmann, 1995.

[Davies and Russell, 1994] Davies, S. and Russell, S. NP-Completeness of Searches for Smallest Possible Feature Sets. In Proceedings of the 1994 AAAI Fall Symposium on Relevance, pages 37–39. AAAI Press, 1994.

[Devijver and Kitler, 1982] Devijver, P.A and Kittler, J. Pattern Recognition: A Statistical Approach, Englewood Cliffs, NJ:Prentice/Hall.

[Dietterich, 1998] Dietterich, G. Thomas, Machine-Learning Research: Four Current Directions, In AI Magazine, Vol.18, No.4, pages=97–136, 1998.

[Doak, 1992] Doak, J. An Evaluation of feature selection methods and their application to Computer Security", University of California at Davis, Tech Report CSE-92-18.

[Duda, Hart and Stork, 2000] Duda, O.Richard, Hart,E.Peter, Stork, G. David. Pattern Classification, ISBN- ISBN: 0471056693, Wiley, New York, NY 2000.

[Elomaa and Kaariainen, 2001] Elomaa, T. and Kaariainen, M. (2001) An analysis of reduced error pruning.In Journal of Articial Intelligence Research, Volume 15, pages 163–187, 2001.

[Esposito et al, 1997] sposito, F., Malerba, D., Semeraro, G., A Comparitive Analysis of Methods for Decision Trees. In Pattern Analysis and Machine Intelligence, Vol. 19,No.5, pages 476–491, May 1997.

[Freund and Schapire, 1996] Freund, Yoav and Schapire, E. Robert Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, 1996.

[Furnkraz and Widmer, 1994] Furnkranz, Johannes and Widmer, Gerhard, Incremental Reduced Error Pruning,In International Conference on Machine Learning,pages 70-77,1994.

[Ginsberg, 1993] Ginsberg, M.L. Essentials of Artificial Intelligence. Morgan Kaufmann.

[Goldberg, 1989] Goldberg, David E. Genetic Algorithms in Search, Optimization and Machine Learning Addison-Wesley Pub. Co. ISBN: 0201157675. 1989.

[Hart and Cover, 1967] Cover, T. and Hart, P. Nearest Neighbor pattern classification, In IEEE transactions in Information Theory, Vol.13, pages 21-27.

[Hayk, 1994] Haykin, S. Neural Networks. Macmillan College Publishing Company, Inc, New York.

[John et al, 1994] John, H.George and Kohavi, Ron and Pfleger Karl, Irrelevant Features and the Subset Selection Problem, In International Conference on Machine Learning, pages 121-129, 1994.

[John, 1997] John, H. George. Enhancements to the Data Mining Process, PhD Thesis, Computer Science Department, School of Engineering, Stanford University, March 1997.

[John and Langley, 1996] John, G. and Langely, P.Static Versus Dynamic Sampling for Data Mining. In E. Simoudis and J. Han (Eds.) Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI/MIT Press, 1996.

[Kira and Rendell, 1992] Kira, Kenji and Rendell, L.A. A Practical Approach to Feature Selection. In Proceedings of International Conference on Machine Learning, Sleeman,D. and Edwards, J. (eds.), Morgan Kaufmann, pages 249-256, Alberdeen, Italy, July, 1992.

[Kirk et al, 1983] Kirkpatrick, S., Gelatt, C., Jr., and Vecci, M. Optimization by simulated annealing. Science, 220:671–680.

[Kohavi and John, 1998] Kohavi, Ron and John, George.H. Wrappers for Features Subset Selection, In Artificial Intelligence, Vol.97, No.1-2, pages 273-324, 1997.

[Koller and Sahami, 1996] Koller, D. and Sahami, M. Toward Optimal Feature Selection.In Proceedings of the 13th International Conference on Machine Learning (ML), pages 284–292, Bari, Italy, July 1996.

[Kononenko, 1994] Kononenko, Igor. Estimating Attributes: Analysis and Extensions of RELIEF, In European Conference on Machine Learning, pages 171-182, 1994.

[Langley et al., 1992] Langely, P, Iba, W.,& Thompson, K. An analysis of Bayesian Classifiers. In proceedings of the Tenth National Conference on Artificial Intelligence, pages 223–228,SanJose, CA, AAAI-1992.

[Langley, 1994] Langely, P. Selection of relevant features in machine learning, In AAAI Fall Symposium on Relevance, pages 140–144, 1994.

[Langely and Sage, 1994] Langely, P. and Sage, S. Induction of selective Bayesian Classifiers. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pages 399-406, Seattle, WA, Morgan Kaufmann, 1994.

[Langely and Sage, 1994b] Langley, P., and Sage, S. Oblivious decision trees and abstract cases. In Working Notes of the AAAI94 Workshop on Case-Based Reasoning. AAAI Press, 1994.

[Langely, 1996] Langely, P. Elements of Machine Learning. Morgan, B. Michael (eds.), ISBN 1558603018, Morgan Kaufmann, 1996.

[Lewis and Catlett, 1994] Lewis, D.David and Catlett, Jason Catlett, Heterogeneous uncertainty sampling for supervised learning, In Proceedings of ICML-94, 11th International Conference on Machine Learning, pages 148–156, Morgan Kaufmann Publishers, New Brunswick, 1994.

[Liu and Motoda, 1998] Huan, Liu and Hiroshi, Motoda. Feature Extraction, Construction, and Selection . A Data Mining Perspective. ISBN: 079238198X, Kluwer Academic Publisher, 1998.

[Markovitch and Scott, 1993] Shaul Markovitch and Paul D. Scott. Information Filtering: Selection Mechanisms in Learning Systems. Machine Learning, Volume 10, number 2, pages 113–151, February 1993.

[Markovitch, 1989] Shaul Markovitch. Information Filtering: Selection Mechanisms in Learning Systems. PhD thesis, EECS Department, University of Michigan, 1989

[Matheus and Rendell, 1989] Matheus, C.J. and L.A. Rendell, L.A., Constructive induction on decision trees, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pages 645-650, 1989.

[Michalski, 1983] Michalski, R. A theory and methodology of inductive learning. In Artificial Intelligence, volume 20, pages 111-161, 1983

[Minsky and Papert, 1969] Minsky, M. and Papert, S. Perceptrons. Cambridge, MIT Press.

[Mitchell, 1997] Mitchell, Tom. Machine Learning. ISBN 0070428077 ,McGraw Hill, 1997.

[Murphy and Pazzani, 1991] Murphy, P.M. and Pazzani, M.J. ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees. In Proceedings of the Eighth International Workshop on Machine Learning, pages 183–187, Evanston, IL, June 1991.

[Japkwicsz, 2000] Japkowicz,N. The Class Imbalance Problem: Significance and Strategies , In the Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000).

[Pagallo and Haussler, 1986] Pagallo, G. and Haussler, D., Boolean feature discovery in empirical learning. In Machine Learning, Volume 1, number 1, pages 81-106, 1986.

[Pedrod, 1996] Pedrod, D.Context-Sensitive Feature Selection for Lazy learner. In Artificial Intelligence Review, Volume 11,pages 227-253, 1997.

[Quinlan, 1983] Ross Quinlan. Learning efficient classification procedures and their application to chess end games. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, Machine Learning - An Artificial Intelligence Approach, pages 463–482. Tioga, Palo Alto, CA, 1983.

[Quinlan, 1993] Quinlan, J.R. C4.5:Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

[Rendell and Seshu, 1990] Rendell, A. Larry. and Sheshu, Raj. Learning hard concepts through constructive induction: Framework and rationale, Computational Intelligence, Volume 6, pages 247-270. 1990.

[Rosenblatt, 1962] Rosenblatt, F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan, New York, 1962.

[Rumelhart et al., 1986] Rumelhart, D.E., Hinton, G.E., and Williams, R.J. Learning internal representations by error propagation. In DE Rumelhart and JL McClelland, editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1, Vhapter 8. Bradford Books (MIT Press), Cambridge, MA, 1986.

[Schapire, 1990] Schapire ,R.E. The strength of weak learnability. In Machine Learning, Volume 5, pages 197–227, 1990.

[Seung e tal., 1992] H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In Proceedings of the Fifth Workshop on Computaional Learning Theory, pages 287–294, San Mateo, California, 1992.

[Winston, 1975] P.H. Winston. "Learning structural descriptions from examples". In P.H. Winston editor, The psychology of computer vision. McGraw-Hill, New York, 1975.