

פברואר 2003

פרויקט מסכם בנושא:

תנאי עצירה מוקדמים לעצי הכרעה בישומי DATA MINING

עבודת גמר במסגרת לימודים לתואר שני בפקולטה להנדסה – המחלקה
ללימודים בינתחומיים

מגיש: מוטי בן-נון

מנחה: דר' דנה רון, המחלקה להנדסת חשמל - מערכות

Abstract

Optimization of the decision trees induction process and classification process has a major effect on data mining applications that usually deal with very large data bases. Reducing training time or classification time can have a significant effect on the total running time.

There are two main approaches to improve decision trees accuracy: Pruning and Early stopping rules. Although early stopping rules have a good potential of reducing both training time and increasing predictive accuracy (pruning methods cannot reduce training time) they are not in wide use because of early works that showed evidence that early stopping rules reduce predictive accuracy.

This project deals with exploring existing papers regarding pruning and early stopping efficiency and comparing their performance. The conclusions are that pruning is not always helpful. When the problem is complicated because the number of attributes is large or because the classes are cost sensitive or because the relations between the class and the attributes are weak, fully growing the tree is too costly and thus pruning is not effective . Early stopping rules can improve the performance and reduce tree growing time in problems that contain all of the mentioned difficulties.

Table of contents

<i>Abstract</i>	5
<i>Table of contents</i>	6
<i>Chapter 1</i>	7
Background	7
Data mining	7
Machine Learning	9
Decision Trees	10
Pruning (Post Pruning)	16
Early stopping	17
<i>Chapter 2</i>	18
Scope and purpose	18
<i>Chapter 3</i>	20
Related work - Early stopping & pruning effect	20
<i>Chapter 4</i>	25
Experiment Model Description	25
Cost effects	28
Target Function – the “Grade Measure”	32
Early stopping rules	33
Cost Complexity Pruning	35
<i>Chapter 5</i>	39
Experimental results- Early stopping rules	39
Experimental results – Pruning	47
Experimental results – combinations of methods	47
<i>Chapter 6</i>	47
Conclusions	47
<i>BIBLIOGRAPHY</i>	47

Chapter 1

Background

Data mining

Data mining involves fitting models or determining patterns from observed data. The Fitted models play the role of inferred knowledge: Whether or not the models reflect useful or interesting knowledge is part of the overall interactive Knowledge Discovery in Database (KDD) process.

Most data mining methods are based on concepts from machine learning and statistics: Classification, Clustering, Graphical models and so forth.

The two “high-level” primary goals of data mining in practice tend to be prediction and description. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. Description focuses on finding human interpretable patterns describing the data.

The goal of prediction and description are achieved by using the following primary data mining tasks:

- Classification is learning a function that maps (classifies) a data item into one of several predefined classes.
- Regression is learning a function, which maps a data item to a real valued prediction variable.

- Clustering is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data.
- Dependency modeling consists of finding a model that describes significant dependencies between variables.
- Change and Deviation Detection focuses on discovering the most significant changes in the data from previously measured values.

Having outlined the primary tasks of data mining, the next step is to construct algorithms to solve them.

Machine Learning

The object of machine learning is to improve the behavior of a computer over time by having the computer process input, usually in the form of concrete samples of data, and change its behavior so that it can more effectively process similar input in the future.

In this work the application is based on **Inductive learning** which is the process of acquiring generalized knowledge from samples or instances of some class. This form of learning is accomplished through inductive inference, the process of reasoning from a part to a whole, from particular instances to generalization, or from individual to universal. It is a powerful form of learning that humans do almost effortlessly.

The traditional inductive learning tends also to assume that the learning process is divided sharply into temporal phases. The learning process is assumed to start with **training phase**, where the samples are presented to the system and the system computes a general description. Then there's a **validation phase** where the descriptions are tested against new samples not originally in the training set. If the results are not satisfying the training process must restart using different settings to ensure different and hopefully better results. Estimation of the algorithm performance can be done using different set that is referred as **test set**.

The **measure of how "good" that learning algorithm is** -- whether the algorithm correctly classifies a set of samples, whether the algorithm can learn the shortest possible description from a set of samples, whether a generalized algorithm can handle a given set of cases. The algorithm is sometimes evaluated by how many samples it may take to converge on a given description.

Decision Trees

Decision Trees are one of the popular methods that are in use in the field of data mining marked as a machine learning method.

A decision tree is a structure that represents a procedure for classifying objects based on their attributes.

Each object is represented as a set of attribute/value pairs and a classification. For example, a set of medical symptoms might be represented as follows:

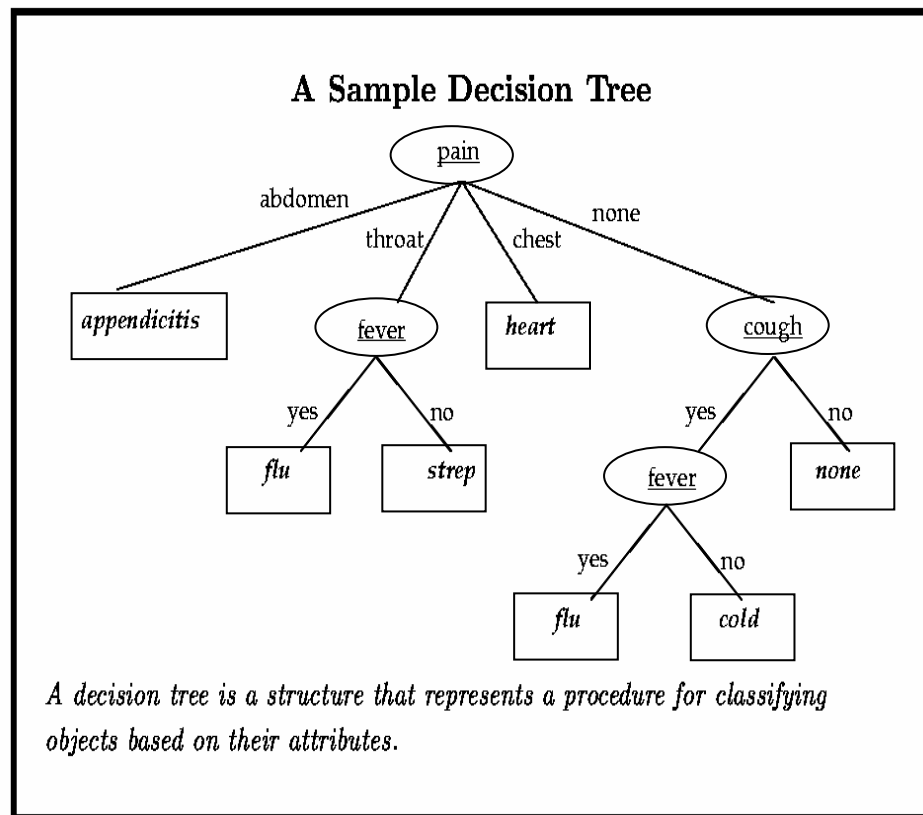


Figure 1 - Example of a Decision Tree

Knowing the decision tree we can easily classify any new instance as seen in the next table.

	Attributes (Features)				Classification
	Cough	Fever	Weight	Pain	
Mary	yes	yes	skinny	none	flu
John	no	no	normal	none	none
Fred	yes	no	normal	none	cold

Table 1 - set of instances and their classification

Decision trees automatically constructed from data have been used successfully in many real world situations. Decision trees have several advantages for classification problem solution:

- Knowledge from pre-classified samples can be easily introduced.
- Decision trees can model wide range of data distribution.
- Trees classification process is easy to understand. While other classifier use complex mathematical model to evaluate the sample class, Decision Tree rely on simple If...Then structure.

Decision trees are constructed from a training set, which consists of samples. Each sample is completely described by set of attributes (features) and a class label. The concept underlying a data set is the true mapping between the attributes

and the class. A decision tree contains zero or more internal nodes and one or more leaf nodes. All internal nodes have two or more child nodes. All internal nodes contain splits, which test the value of expression of the attributes. Each leaf node has a class label associated with it.

The task of constructing a tree from the training set has been called tree induction. Most existing tree induction systems make greedy heuristic choice of one of a set of candidate splits for a data set and then recursively partitions each of the subsets produced by the split. The recursive splitting process terminates when all members of the subset are in the same class or when the set of candidate splits is empty. Those are the natural stopping conditions of the training phase.

Researches studying classification problems based on induced decision trees have found that the model that best fits training data is unlikely to yield optimal performance on fresh data. This kind of model is **overfitted** to the training set in terms that it captures noise and patterns that have no significance for the true model. In Figure 2 the left regression model seems to represent the data while the right model is overfitted to data points and we believe it will have smaller accuracy on new instances although it has higher accuracy on the given data set. We would like Decision trees to be better than the data by capturing only the true characteristics of the data.

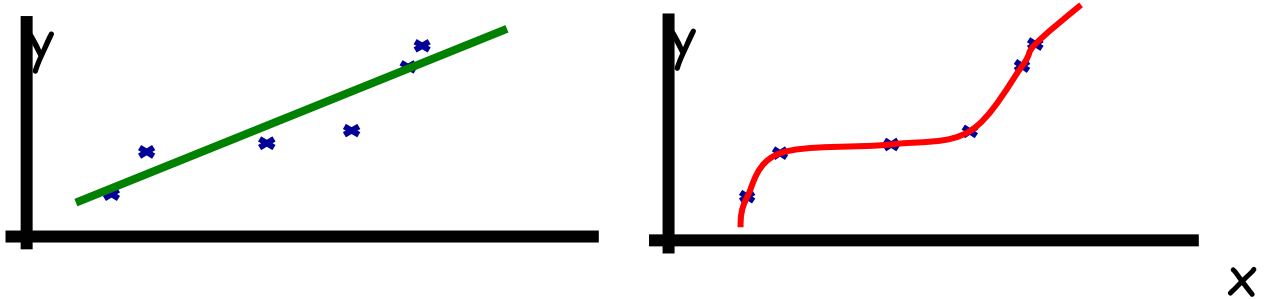


Figure 2 -Overfitting illustration

The basic idea of overfitting avoidance in decision tree induction algorithms is that full grown tree that perfectly fits the training data set is overfitted to data points and removing some leaves and branches from the decision tree can improve the accuracy on unseen data sets. In the early stages of the decision tree development few researches found out that the predictive accuracy on unseen data set is dependent on the tree's depth (see Figure 3).

Overfitting avoidance is supposed to be achieved by removing those leaves and branches that harm the predictive accuracy (see

Figure 4). This strategy is called pruning. The more common approach is “post pruning” that allows the tree to reach to its full size and then removing some leaves and branches and creating new leafs instead.

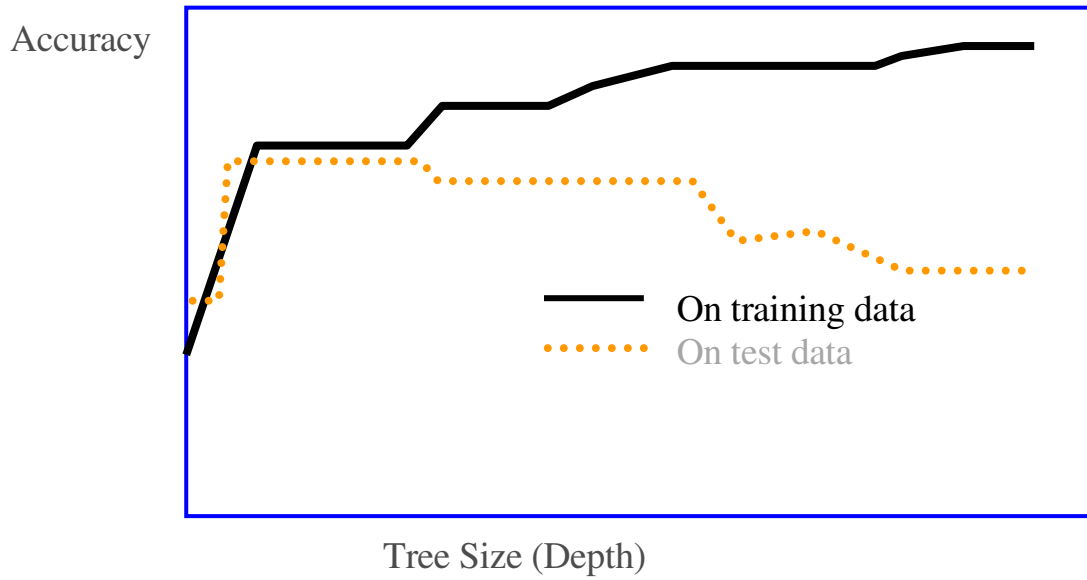


Figure 3 - Accuracy of a decision tree on training set and validation set

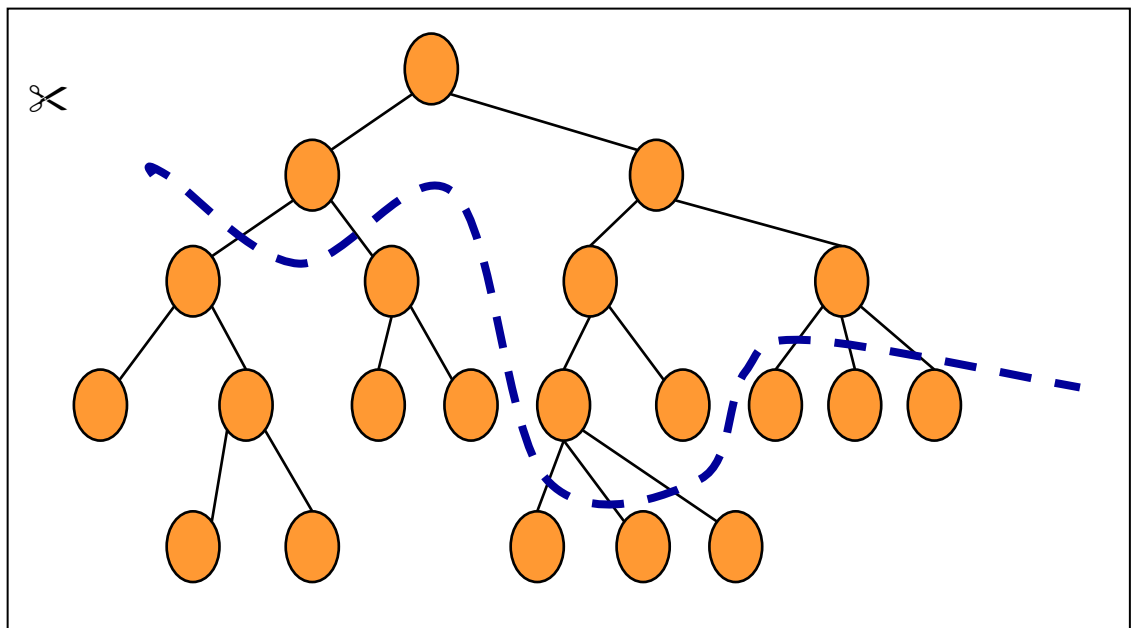


Figure 4 – Decision Tree Pruning

Less common approach is “early stopping” (or pre-pruning) that halts tree growth before it reaches its full size (see Figure 5).

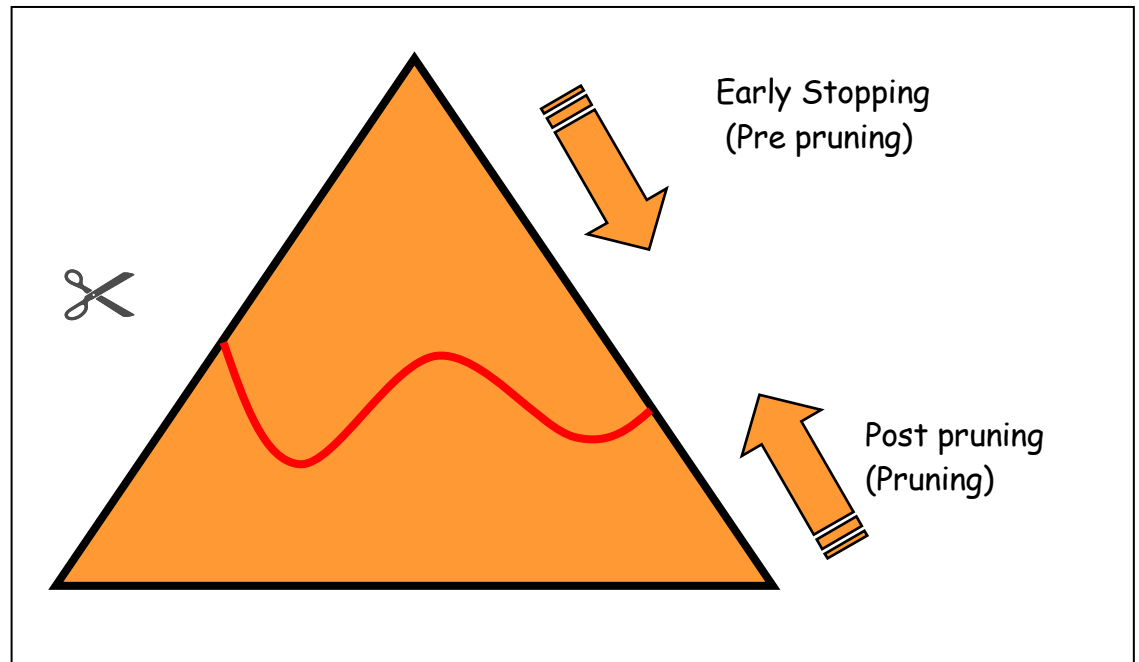


Figure 5 - Early stopping and Post pruning.

Pruning (Post Pruning)

Post pruning methods are supposed to improved the performance of the Tree on fresh data sets. A tree T_{\max} is grown to fit the training set even if it seems worthless and is then retrospectively pruned of those branches that seem to decrease predictive accuracy. Pruning methods aim to simplify those decision trees that overfit the data. While some methods use the training set to evaluate accuracy, others exploit an additional pruning - validation set.

Strategies for increasing predictive accuracy through selective pruning have been widely adopted in decision tree induction.

Early stopping

Early stopping of the decision tree growth can loosen the over-fitted model and improve predictive accuracy on unseen datasets but the greedy splitting algorithms makes it hard to know when to stop the splitting procedures. Early stopping methods establish stopping rules for preventing the growth of those branches that do not seem to improve predictive accuracy of the tree.

Chapter 2

Scope and purpose

The purpose of this work is to study the efficiency of early stopping (pre-pruning) that tend to avoid overfitting and comparing it with pruning (post pruning) algorithm with respect to the practical difficulty that derives from dealing with large data set as acceptable in Data Mining applications.

The Pruning approach adds procedures for replacing one or more of the splits with a terminal node or leaf after the tree reached its full size. These algorithms are costly because of the time spent in building the full grow tree and because of the overhead involved in calculating the pruning potential.

Early stopping reduces complexity of the grown tree by not letting the tree reach its full size. Definition of early stopping conditions that obtain high accuracy on unseen data set can save a lot of calculation time involved in further growing the tree and then pruning it.

Many papers deal with variety of pruning algorithms and their performance on different problems but only few papers examine the performance of early stopping conditions in data mining problems.

Although most of the researches agree on the benefits of using pruning methods, some challenge this approach and investigated the pruning effects in data mining problems.

This project covers existing papers related to comparing early stopping and pruning algorithms. In real life problems the aim is not to achieve 100% accuracy but to solve the problem efficiently meaning that we are willing to pay for running time with accuracy.

Based on the existing papers and other ideas the project compares pruning and early stopping in terms of accuracy.

The purpose of the decision trees grown in this work is to generalize customer's behavior in large store. Each sample is constructed from at least 15 attributes (continuous variables) and single binary class. The class equals 1 if the customer bought the target product, otherwise the class equals zero. The training set size contains thousands of samples.

The distribution of classes is class=0 for more than 90% for almost any item at the database. This is mainly because the large variety of items at the database and typical customer behavior.

A secondary purpose of this project was to study current approaches for dealing with unbalanced class distributions. In Data Mining applications it is common practice to apply Cost factors for different classes (Weighting) but only lately researchers started to study the effect of class distribution and Cost factors on the decision tree induction process.

Chapter 3

Related work - Early stopping & pruning effect

Actual results regarding early stopping methods are very rare since they are related to the early stages of decision tree development in the mid 80's. Those works are usually based on relative simple concept learning based on small number of features and sometimes small databases thus very far from current data mining applications.

But when it comes to pruning effect the things are different. Researches put a lot of effort in finding and characterizing the optimal pruning algorithm. Pruning became a common practice in top down induction of decision trees and every respectable application includes pruning procedure. The reasons for applying pruning are very rational – the tree induction process is based on a greedy algorithm that needs to be corrected by “loosing” the overfitting. On expense of the accuracy in the training set we can get better performance on unseen datasets. The same is not true for early stopping conditions since they are considered to reduce accuracy and generalization by simplifying the tree because the tree did not reach it full potential size and the expected performance are not tested on unseen test set.

In spite of the above a new approach is rising against pruning. An increasing number of researches claim to have strong evidences that pruning reduces predictive accuracy of decision trees especially in problems that are hard to learn for some of the following reasons:

- Having a large number of features.
- The features are not strongly related to the class.
- Class distribution is unbalanced and some classes can be very rare and hard to learn.
- Cost of misclassification of error is not uniform with the class distribution.

All of the above can be found in data mining problems and so pruning is not expected to produce better performance but to simply injure the tree.

The claim is that in problems with one of the above difficulties pruning won't improve accuracy.

Here is a short briefing of some of the papers that oppose pruning:

1. Brieman et. al. – “Classification and Regression Trees”

Brieman found out that tree quality depends more on good stopping conditions than on splitting rules. The following stopping conditions were previously examined:

- Restriction on minimum node size: A node is not split if it has smaller than k samples, where k is a parameter to the

tree induction algorithm. Used on early works but no measure of performance were found.

- Threshold on impurity: A threshold is imposed on the value of splitting criterion, such that if the splitting falls below the tree growth is aborted. The problem is to define threshold that is meaningful at all nodes in the tree.

2. J. Kent Martin and D.S Hirschberg – “Time complexity of tree induction”

The authors investigated different variables that affect decision trees accuracy and time complexity. They found out that early stopping or pruning have little or no impact on accuracy and in some cases where accuracy is significantly affected, the effect is sometimes beneficial, sometimes harmful.

3. J. Kent Martin – “An exact probability Metric for decision trees”

Kent suggests a new approach to induce early stopping by calculating the probability of obtaining the observed data under the null hypothesis using Fishers exact test P_0 . P_0 the null hypothesis probability measure gave more accurate trees than unpruned trees or using other early stopping conditions.

4. Cullen Schaffer – “Over fitting avoidance as bias”

In this paper Schaffer showed that pruned trees have better accuracy only in a certain type of the problems. He uses series of experiments that empirically showed that the success of a pruning algorithm depends on the problem generating environment and not on the pruning method. He performed tests by growing trees using CART and then pruning the tree using cost complexity pruning method. He showed that for large trees the full grown trees outperform the pruned trees in 80% of the test with significance level of 97%.

5. Cullen Schaffer – “when does overfitting decrease prediction accuracy in induced decision trees and rule sets”

Schaffer attempted to find out what are the conditions that make pruned trees better or when does overfitting decrease accuracy. Schaffer showed that pruning can improve accuracy only in exceptional circumstances, depending on class distributions.

Overfitting decreases prediction accuracy in the presence of significant description noise or when attributes are not informative enough for classification. Schaffer claims that pruning usually increases accuracy because descriptive noise is not relevant to most of the problems and that the chosen attributes are often highly relevant for classification.

Schaffer also established empirical evidence showing that pruning isn't only problem dependent but representation dependent as well. Overfitting occurs when a complex model outperforms a simple one on training data,

but not in true accuracy. Since the complexity of a decision tree model depends on the representation of attribute data a bias toward models that are considered simple under one representation exists.

6. Patrik M. Murthy and Michael J. Pazzani – “Exploring the decision tree forest: An empirical investigation of occam’s razor in decision tree induction”.

The researchers investigated the relationship between the size of a decision tree consistent with some training data and the accuracy of the tree on test data. For most of the problems they explored they found that on average, the smallest decision trees consistent with the training data had more error on unseen data than slightly larger tree. While Schaffer allowed the tree not to be consistent with all training data Murthy and Pazzani used on tree that were consistent with all the training data.

The disadvantage of this paper is that it relays on training sets of small size.

Similar conclusions were obtained by Tapio Elomaa – (The biases of decision tree pruning strategies). She found that pruning have better predictive accuracy in simple concepts. In learning complex concepts it is better to allow the tree to reach is full length.

Chapter 4

Experiment Model Description

This part of the work will describe the model of the tree induction process that is applied on a typical data mining database.

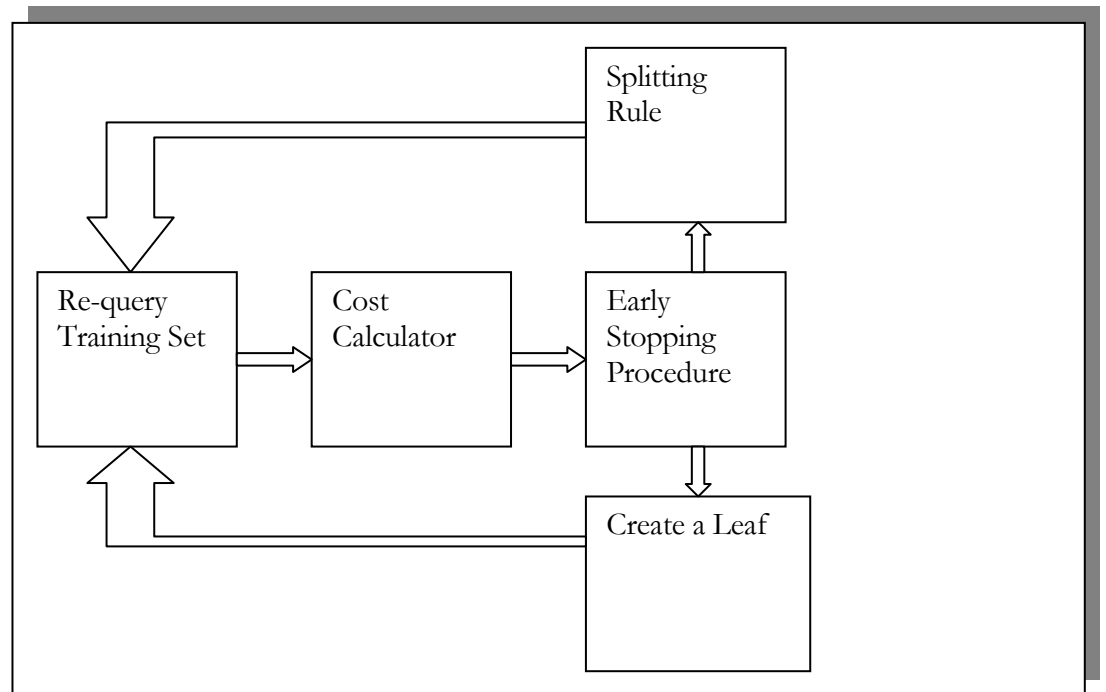


Figure 6 - Basic Tree Induction block diagram

The basic tree induction includes the following procedures:

1. Re-query Training Set. After building the previous node / leaf the procedure selects the relevant portion of the training set the reached the last node.

2. Cost Calculator. On top of the relevant training set the cost calculator computes the relevant weight of each sample using the **Cost matrix**. The Cost Calculator is aimed to discriminate between classes by giving a predefined weight to a certain class. The method of applying cost and the legitimacy of this method will be explained in the next section. Cost calculator uses **Cost matrix**, C of the size $n \times n$. $C(i,j)$ is the cost of assigning a case to class i where it actually belongs in class $=j$. How to determine the values of $C(0,1)$ and $C(1,0)$ will be also explained later on.

	Actual Class is "0"	Actual Class is "1"
Predict "0"	0	X
Predict "1"	Y	0

Table 2 – Cost Matrix Values

3. Splitting Rule. Looks for the most informative split among all possible splits.
4. Early Stopping Procedure. Applies the current active early stopping conditions on top of the natural stopping conditions (reached 100% purity or all samples have the same attributes values).
5. Create Leaf. The leaf value is determined by the majority based on weighted number of samples.

On top of this tree induction program, a pruning procedure can be applied. The advantages of this structure are the flexibility in combining different stopping condition with pruning and different cost models. The cost affects all of the procedures involved in inductive tree learning and pruning: Splitting rule, stopping rules, Leaf creation and pruning. Other works related to cost models tried to find a way to introduce cost into existing popular applications (like C4.5) by changing the natural frequency of the classes in the training set. Since cost is integrated in the data mining process data mining application have must integrated cost capabilities.

Cost effects

Many types of “cost” are involved in the process of data mining. For example the following types of cost (and more) were defined by Peter Turney in “Types of cost in inductive concept learning”: cost of tests which is the cost involved in obtaining the attributes, cost of a teacher which is the cost of determining the class of an instance, cost of computation. In this project, cost, means **Cost of misclassification errors**. That is, the cost of assigning a sample to class i , when it actually belongs to class j . For example, the cost of misclassification failing to diagnose some diseases can lead to irreparable damage by delaying the needed treatment or by giving the inappropriate treatment to the patient. Applying cost into the induction process of the decision tree aimed to classify patient disease creates a risk reduction decision process in the same way an expert doctor would try to reduce risks.

Cost sensitive problems learning are more difficult by the fact that in many domains the expensive errors correspond to the rare cases (in the previous example the disease can be rare and harmful). A cost insensitive learner might therefore decide to ignore these rare events and classify them as negative (in the previous example the classifier will fail to identify the disease but he will have high accuracy rate because the prediction is true for most of the population).

At each step in the induction process the cost calculator calculates the weighted number of samples for each class.

We will define the following in order to explain how to calculate the weighted number of samples (we will continue using them to further to define early stopping rules):

- \mathcal{S} is the whole set of training samples
- l is a leaf in the tree.
- $\mathcal{S}(l)$ is the portion of the given set \mathcal{S} that reached to the leaf l
- $\mathcal{S}_i(l) = \{x \in \mathcal{S}(l) : \text{Class}(x) = i\}$, $i=0,1$

The induction and pruning algorithms use **the weighted number** of samples instead of using **the total number** of samples. For example, in order to determine the leaf class we usually use $\mathcal{S}_i(l)$ to calculate the total number of samples from each class at each leaf. In the weighted case we would use $|\mathcal{S}_0(l)| \cdot C(1,0)$ instead for class=0 and $|\mathcal{S}_1(l)| \cdot C(0,1)$ for class=1. This method resembles changing the class distribution of the training set since we can achieve the same effect by replicating each sample according to the weight calculated using the cost matrix. Few papers proposed to change the class distribution of the training set instead of using cost matrix so that existing tree induction procedures won't have to be changed to include cost effects.

The consequences of changing the given training set class distribution are not trivial. The same is true for using cost matrix in the tree induction process. But as explained above, it stands to reason that in classification problems we want to discriminate between classes. The question is how to determine the values of the cost matrix or what is the "optimal" class distribution. Only recently researches examined the relationship between cost, class distribution and the predictive accuracy of the classifier. The determination of the cost matrix in this work was can be justified using the following papers:

1. Chan and Stolfo [Learning with non uniform class and cost distributions: effects and multi-classifier approach] found out that creating a new class distribution other than the given class distribution can yield more effective results than using the given training set. The reason for that is that using the given class distribution for training creates difficulty to distinguish between minority class instances and noisy data and thus the classifier will have poor performance in prediction of the minority class.
2. Weiss & Provost tried to have better understanding as to what is the optimal value for class distribution.
Under the error rate measure they found that the results vary with minority class distribution **a natural distribution is not the best class distributions for training** (the same is true for 50:50 distribution).

They also used **Receiver Operating Characteristic** (ROC) analysis which represents the false positive on the X axis and the true positive on the Y axis. ROC curves are independent of the naturally occurring class distribution or error cost.

To assess the overall quality of a classifier they measured the fraction of the total area that falls under the ROC curve and defined it as AUC (area under curve) where large AUC values indicate generally better classifier performance and indicate better ability to rank cases.

They found out that when using the AUC measure, the optimum ranges appear to be centered to the right of the 50:50 class distribution.

The Cost matrix that was used in this project was selected to achieve 50:50 distribution of training set based on the natural distribution of the training set:

	Actual Class is 0	Actual Class is 1
Predict "0"	0	Relative frequency of class = 0 in the training set
Predict "1"	Relative frequency of class = 1 in the training set	0

Table 3- **Cost Matrix**

Target Function – the “Grade Measure”

The target Function is the measure that is used to estimate the performance of the classifiers. In the ordinary 1/0 loss case, we can group together all of the different misclassification errors and estimate the total fraction of misclassifications.

Here we want the classifier to have better predictions on the rare class we used the following grade measure: $GradeMeasure = \alpha \cdot TP + (1 - \alpha)TN$

Where TP=True Positive which is the correct predictions of class=1 and TN=True negative is the correct predictions of class=0 and $0 \leq \alpha \leq 1$. To give greater significance to the rare class we used high values of α (greater than 0.85).

In the some of the results analysis the error rate measure is used. The definition of the error rate is:

$$ErrorRate = \frac{FP + FN}{TP + TN + FN + FP}$$

Where TP & TN are as defined above.

FN (False Negative) is the misclassification of sample as 0 where it actually belongs to Class=1.

FP (False Positive) is the misclassification of sample as 1 where it actually belongs to Class=0.

Early stopping rules

Although early stopping is widely referred as basic approach to loosen overfitting it is still rare to find reports on early stopping performance. In this work the following early stopping rules were investigated:

1. Tree Depth.

Turn a node to a leaf if:

Node depth > *TreeDepth* Threshold.

2. Global Purity.

Turn a node to a leaf if the purity of class j > *Global Purity* Threshold :

In unit class models the purity is simply the total fraction of samples that belong to a certain class:

$$GlobalPurity(i) = \frac{|S_i(I)|}{|S(I)|}$$

In the non uniform case the purity will be the weighted number of samples using the cost matrix:

$$GlobalPurity(1) = \frac{|S_1(I)| \cdot C(0,1)}{|S_1(I)| \cdot C(0,1) + |S_0(I)| \cdot C(1,0)}$$

$$GlobalPurity(0) = \frac{|S_0(I)| \cdot C(1,0)}{|S_1(I)| \cdot C(0,1) + |S_0(I)| \cdot C(1,0)}$$

3. Max Class Purity & Min-others

Using the previous definitions for $GlobalPurity(1)$ and $GlobalPurity(0)$ we add another demand for the minimum weight of the other class, $Min-others(0)$ is weighted number of samples that reached the leaf 1 and belong to class=1.

$$Min-others(0) = |S_1(l)| \cdot C(0,1)$$

$$Min-others(1) = |S_0(l)| \cdot C(1,0)$$

Turn node to a leaf if:

$$\begin{aligned} &Purity(i) > MaxClassPurity \\ &and \\ &MinWeight(i) < Min-others \end{aligned}$$

4. Min no. of samples

Turn a node to a leaf if the weighted number of samples reached to the current node $<$ Min no. of samples

$$|S_1(l)| \cdot C(0,1) + |S_0(l)| \cdot C(1,0) \leq MinNoSamples$$

5. Min grade

Stop if the grade measure is higher than Min grade threshold. The grade measure is defined as $GradeMeasure = \alpha \cdot TP + (1 - \alpha)TN$

Cost Complexity Pruning

Cost Complexity pruning is used in CART software that is widely used and considered to have reliable results.

The basic idea is: use a test set to compute the classification error rate of each minimum cost-complexity subtree. Choose the subtree with the minimum test error rate. At the initial steps of pruning, the algorithm tends to cut off large subbranches with many leaf nodes. With the tree becoming smaller, it tends to cut off fewer. The pruning algorithm is presented below.

Let the measured misclassification rate of a tree T on the validation set be $R(T)$.

For any subtree $T \leq T_{\max}$, define its complexity as $|\tilde{T}|$ the number of terminal nodes in T . Let $\alpha > 0$ be a real number called the complexity parameter and define the cost-complexity measure $R_{\alpha}(T)$ as:

$$R_{\alpha}(T) = R(T) + \alpha |\tilde{T}|$$

For each value of α , find the subtree $T(\alpha)$ that minimizes $R_{\alpha}(T)$, i.e.,

$$R_{\alpha}(T(\alpha)) = \min_{T \leq T_{\max}} R_{\alpha}(T)$$

If α is small, the penalty for having a large number of terminal nodes is small and $T(\alpha)$ tends to be large. For α sufficiently large, the minimizing subtree $T(\alpha)$ will consist of the root node only.

Since there are at most a finite number of subtrees of T_{\max} , $R_{\alpha}(T(\alpha))$ yields different values for only finitely many values of α . $T(\alpha)$ continues to be the minimizing tree when α increases until a jump point is reached.

Two questions:

1. Is there a unique subtree $T \leq T_{\max}$ which minimizes $R_{\alpha}(T)$?
2. In the minimizing sequence of trees T_1, T_2, \dots , is each subtree obtained by pruning upward from the previous subtree, i.e., does the nesting $T_1 \succ T_2 \succ T_3 \dots \succ \{t\}$ hold?

Definition: The smallest minimizing subtree $T(\alpha)$ for complexity parameter α is defined by the conditions:

1. $R_{\alpha}(T(\alpha)) = \min_{T \prec T_{\max}} R_{\alpha}(T)$
2. If $R_{\alpha}(T) = R_{\alpha}(T(\alpha))$, then $T(\alpha) \prec T$

If subtree $T(\alpha)$ exists, it must be unique.

It can be proved that for every value of α , there exists a smallest minimizing subtree.

The starting point for the pruning is not T_{\max} , but rather $T_1 = T(0)$, which is the smallest subtree of T_{\max} satisfying

$$R(T_1) = R(T_{\max})$$

Let t_L and t_R be any two terminal nodes in T_{\max} descended from the same parent node t . If $R(t) = R(t_L) + R(t_R)$, prune off t_L and t_R .

Continue the process until no more pruning is possible. The resulting tree is T_1 .

For T_t any branch of T_1 , define $R(T_t)$ by

$$R(T_t) = \sum_{t' \in \tilde{T}_t} R(t')$$

where \tilde{T}_t is the set of terminal nodes of T_t .

For t any nonterminal node of T_1 , $R(t) > R(T_t)$.

Weakest-Link Cutting

For any node $t \in T_1$, set $R_\alpha(\{t\}) = R(t) + \alpha$

For any branch T_t , define $R_\alpha(T_t) = R(T_t) + \alpha|\tilde{T}_t|$

When $\alpha = 0$, $R_0(T_t) < R_0(\{t\})$. The inequality holds

For sufficiently small α . But at some critical value of α , the two cost-complexities become equal. For α exceeding this threshold, the inequality is reversed.

Solve the inequality $R_0(T_t) < R_0(\{t\})$ and get

$$\alpha < \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

Define a function $g_1(t)$, $t \in T_1$ by

$$g_1(t) = \begin{cases} \alpha < \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1} & t \notin \tilde{T}_1 \\ +\infty & t \in \tilde{T}_1 \end{cases}$$

Define the weakest link \bar{t}_1 in T_1 as the node such that

$$g_1(\bar{t}_1) = \min_{t \in T_1} g_1(t)$$

and put $\alpha_2 = g_1(\bar{t}_1)$

When α increases, \bar{t}_1 is the first node that becomes more preferable than the branch $T_{\bar{t}_1}$ descended from it.

α_2 is the value after $\alpha_1 = 0$ that yields a strict subtree of T_1 with smaller cost-complexity at this complexity parameter. That is, for all $\alpha_1 \leq \alpha < \alpha_2$ the tree with the smallest cost-complexity is T_1 .

Let $T_2 = T_1 - T_{\bar{t}_1}$

Repeat the previous steps. Use T_2 instead of T_1 , find the weakest link in T_2 and prune off the weakest link node.

$$g_2(t) = \begin{cases} \alpha < \frac{R(t) - R(T_{2t})}{|\tilde{T}_{2t}| - 1} & t \in T_2, t \notin \tilde{T}_2 \\ +\infty & t \in \tilde{T}_2 \end{cases}$$

$$g_2(\bar{t}_2) = \min_{t \in T_2} g_2(t)$$

$$\alpha_3 = g_2(\bar{t}_2)$$

$$T_3 = T_2 - T_{\bar{t}_2}$$

If at any stage, there are multiple weakest links for instance, if $g_k(\bar{t}_k) = g_k(\hat{t}_k)$ then define:

$$T_{k+1} = T_k - T_{\bar{t}_k} - T_{\hat{t}_k}$$

Chapter 5

Experimental results- Early stopping rules

The results presented here are based on data drawn from a large grocery store. As explained previously in chapter 2 each sample is constructed from at least 15 continuous attributes and a single binary class.

This database has most of the characteristics of data mining datasets: large number of features (15 features), the features are weakly related to the class, class distribution is unbalanced (the frequency of class=1 is less than 10%) and the cost of misclassification is not uniform.

1. *Tree Depth*

Behavior of the early stopping rule “Tree depth” is reported in the literature to illustrate overfitting, the accuracy performance drops when tree depth and complexity are high. Usually tree accuracy increases to a certain optimum and then decreases as tree depth increases (see Figure 3). The explanation for the low accuracy at the full grown tree is related to capturing noise in the model.

We will distinguish between the weighted case that creates the uniform class distribution and the un-weighted class distribution (natural distribution) to study the effect of applying cost into the decision tree induction process.

For the un-weighted case (see Figure 7) there is no peak that gives better accuracy (or grade measure): the more detailed the tree the higher the accuracy (and the grade measure).

For the un-weighted case results are different from expected since the accuracy has weak correlation to tree depth at low tree depths.

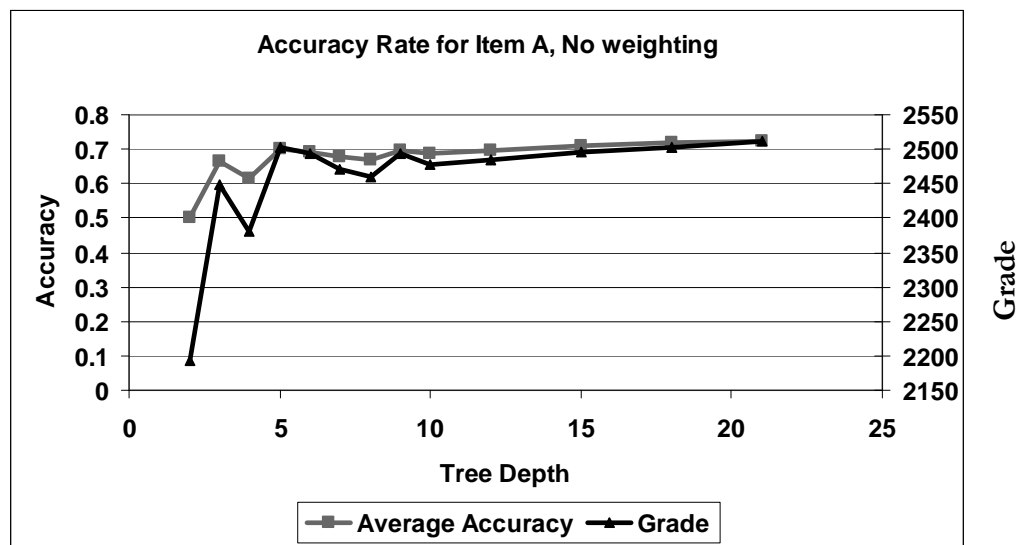


Figure 7 –early stopping= “Tree depth” performance measured in terms of average accuracy and grade measure

When using a cost matrix, accuracy decreases as tree depth increases (Figure 8) but that’s the grade that resembles the results reported in literature. The grade increases to a certain peak and then it decreases. The peak is not that significant but as explained later the peak value and location are dependent on other stopping rules parameters. This by it self is not a strong evidence for preferring the grade measure but if we consider the performance of the average accuracy & the

theoretical background we can say in grater confidence that the grade measure is superior to the accuracy rate for the purpose of this study.

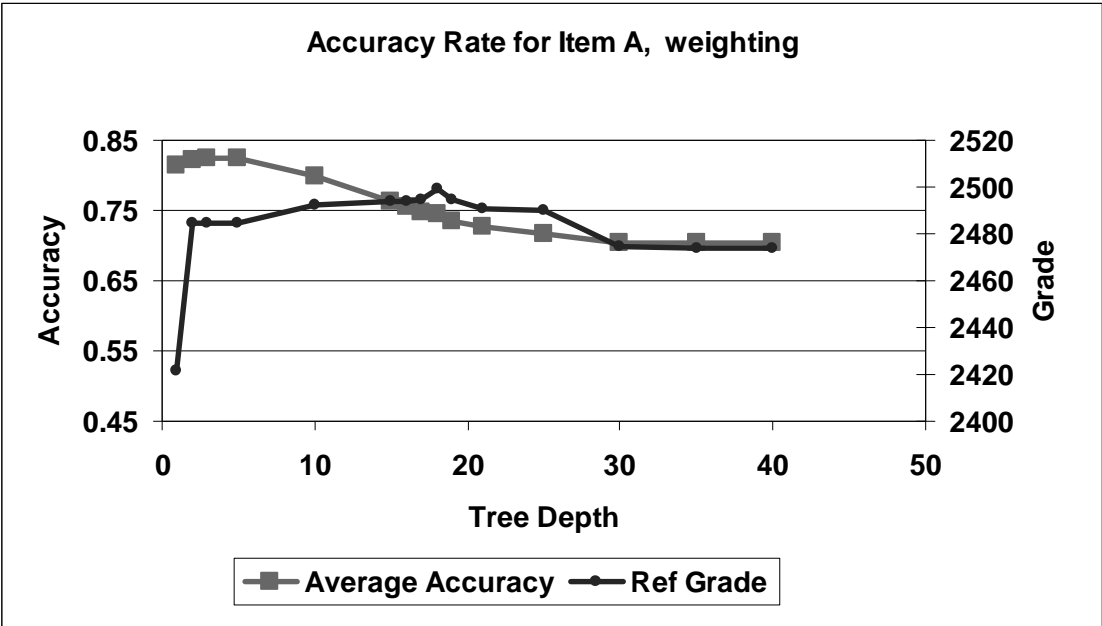


Figure 8 - early stopping= *Tree depth* , Accuracy rate and grade

From Figure 8 we can also get better understanding of the grade measure and the average accuracy. For the uniform cost matrix (natural distribution) Average accuracy and Grade have good correlation but when it comes to the weighted case (using un-uniform cost matrix to create uniform class distribution) they behave differently. The reason for the difference can be understood from Figure 9 and

Figure 10. In these figures, the grade measure has a gain towards the rare class values (Class=1) and thus it is more sensitive than the accuracy measure. In the un-weighted case (Figure 9) the classifier has difficulty to predict class=1 (the rare class) while creating uniform class distribution enables the tree to increase predictive accuracy for the rare class (see class=1 in

Figure 10).

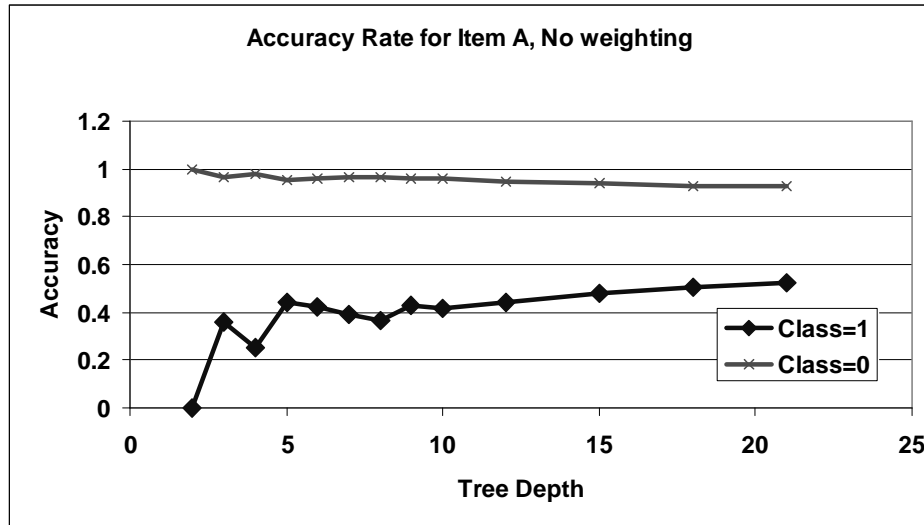


Figure 9 - early stopping= *Tree Depth*, Accuracy Rate

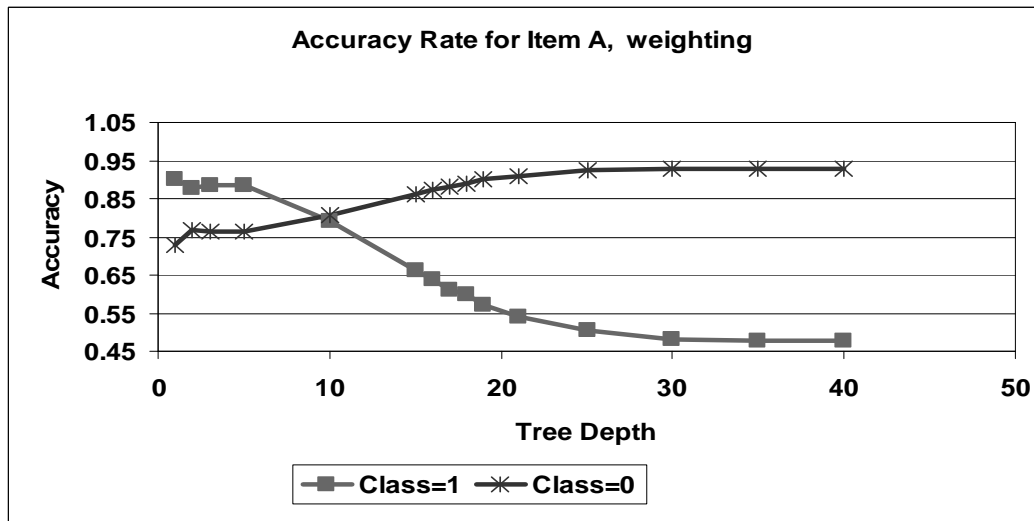


Figure 10 - early stopping= *Tree Depth*, weighting, Accuracy Rate

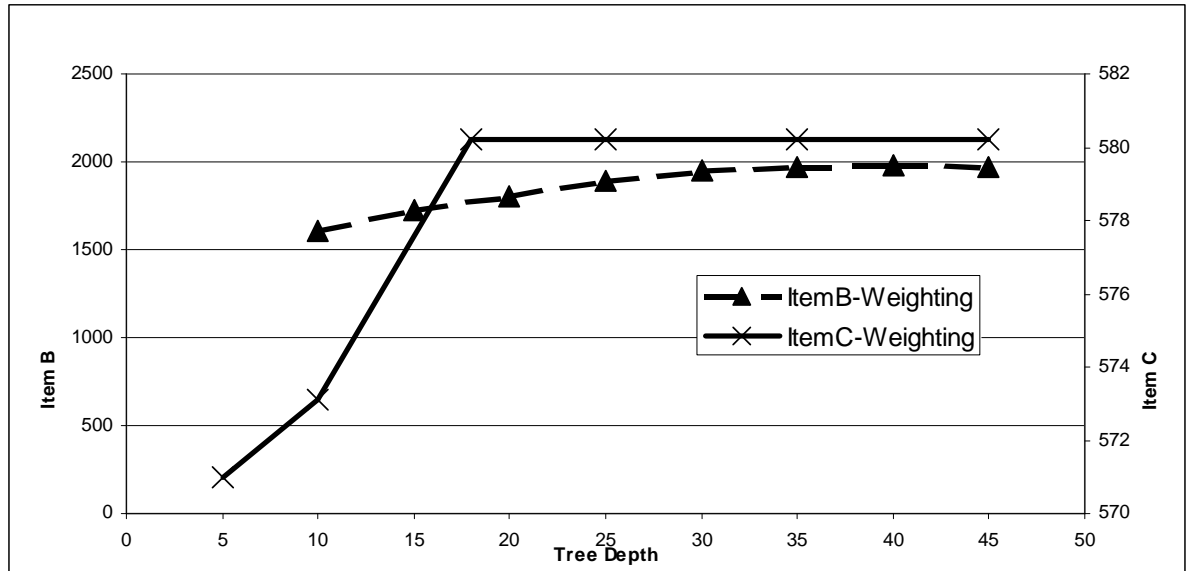


Figure 11 –Grade measure when using Early stopping = *Tree depth* & cost matrix for items B and C

Similar results were obtained for totally different training set (Different item, with different samples and class distribution) as shown in Figure 11 &

Figure 12. In all cases grade measure increases as tree depth increases while the accuracy decreases.

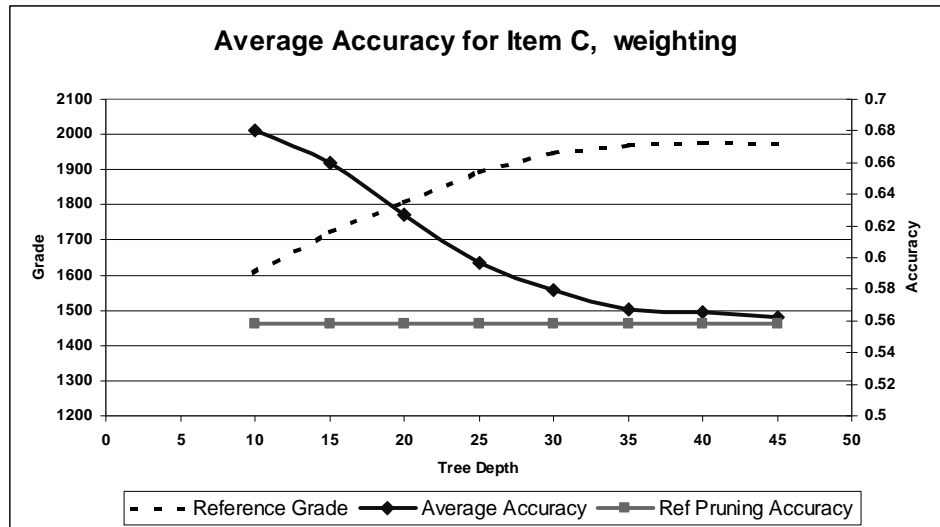


Figure 12 - Grade and Accuracy measures

Another thing that is clear from all of the above charts is that we deal with problems that are difficult to learn using decision trees since a full-grown tree reaches the depth of about 50 binary levels. The difficulty of the problem is a consequence of the number of attributes (about 15 continuous features), the size of the training set (about 50,000 samples) and small correlation between the features and the class.

Tree Depth early stopping condition allows to control training phase running time by bounding it to a certain limit. In this kind of trainings running with Tree Depth threshold of 25-30 can save 10-15 level in tree depth without having significant effect on performance.

2. Global Purity

Global purity early stopping rule is also mentioned in few papers but without detailed documentation. The global purity presented here is for the un-weighted case where class distribution is problematic and class=1 is the rare case (frequency lower then 10%). As seen in Figure 13 & Figure 14 global purity is either fulfilled in the early stages of the training phase or only after reaching extended depth. Since this problem is with two classes only, any purity level below the frequency of the common class will immediately terminate tree growing because the stopping rule is true.

For the weighted case results are expected to be similar to those presented in Figure 15 and it will be explained there.

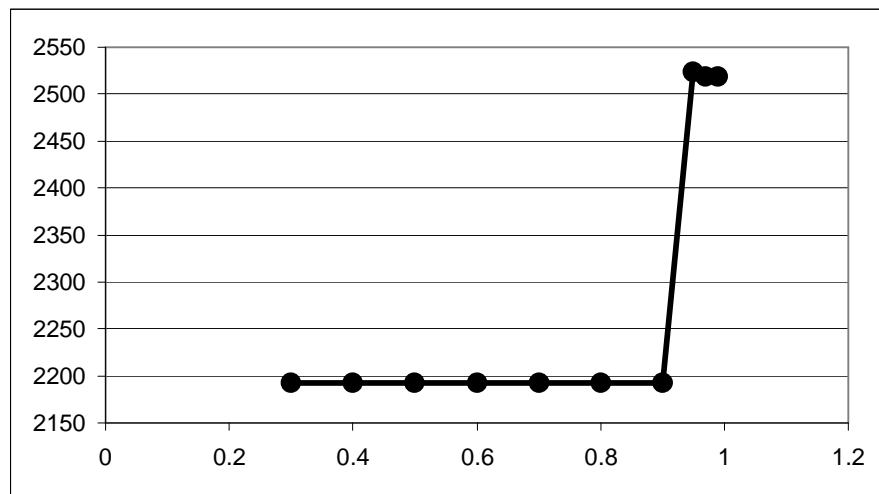


Figure 13 – Grade measure when using early stopping = *Global purity*

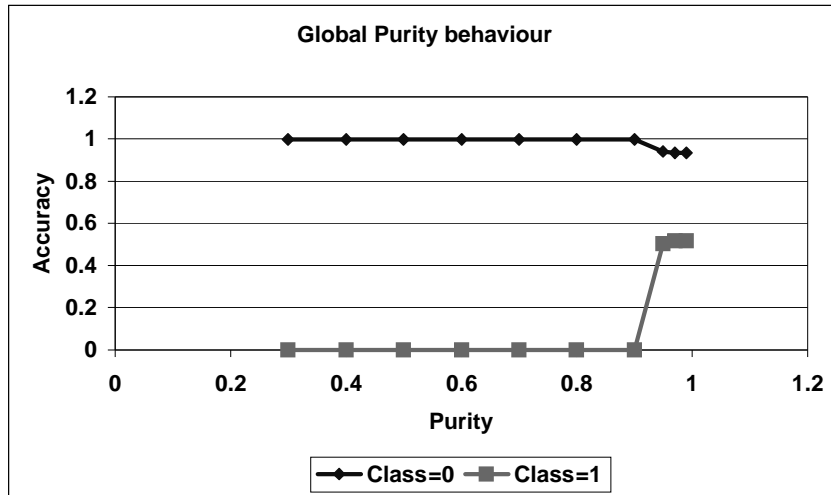


Figure 14 - Accuracy Rate measure obtained when using *global purity* as early stopping

3. Max Class Purity & Min-others

The insensitivity of the global purity method can be avoided when defining individual purity per each class instead of using the same threshold for all classes. On top of the purity demand we add another demand. This demand ensures that when the relative purity condition is true in the very beginning of the induction process growing process will continue if the number of samples from the other class is above the Minimum for the other class weight parameter.

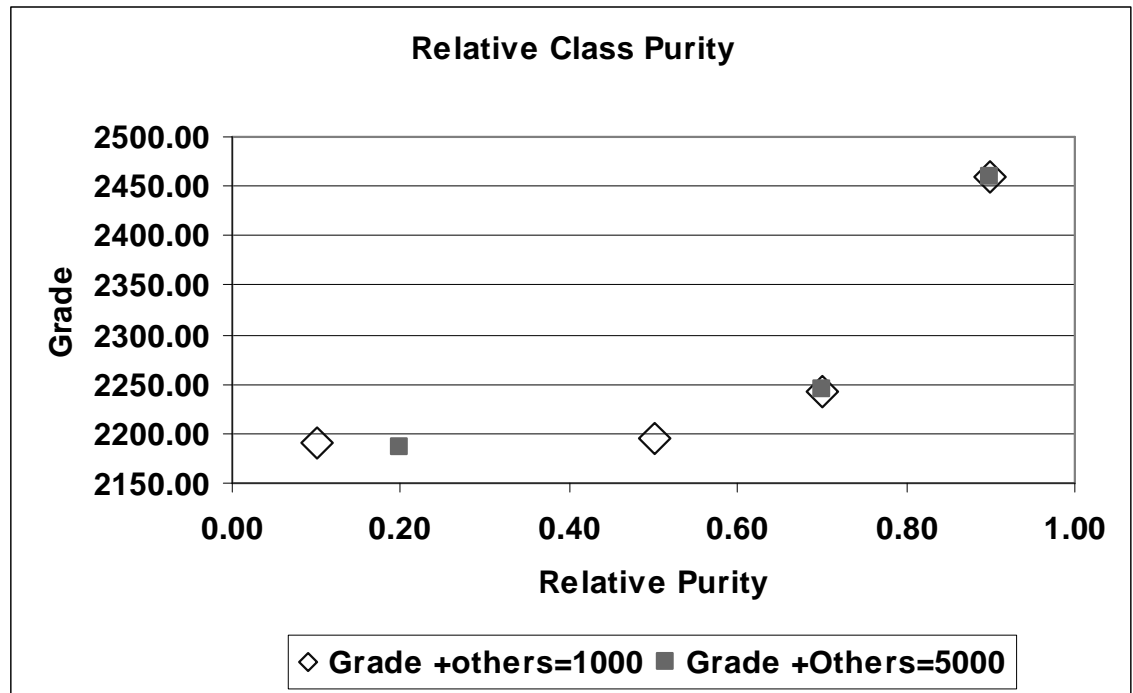


Figure 15 – *Relative Class Purity* early stopping for two values of *min others weight*

Since the current early stopping method involves 4 different parameters (two parameters per each class – Purity and the minimum weighted number of samples from the other class) we need to eliminate some parameters to have better

understanding of the behavior of this early stopping rule. For all of the results presented here $Purity(class=0)=0.99$.

The results of using individual purity threshold and cost matrix (weighting) are shown in Figure 15. In terms of grade measure performance are improving as purity increases.

Because we terminated one of the classes and we look at a constant level of a minimum value for the other class weight, in the weighted case it is the same as running the global purity early stopping rule.

The effect of the threshold on the weighted number of samples from the other class cannot be understood from in Figure 15 since for two different values the measure grade has the same dependency in the relative purity.

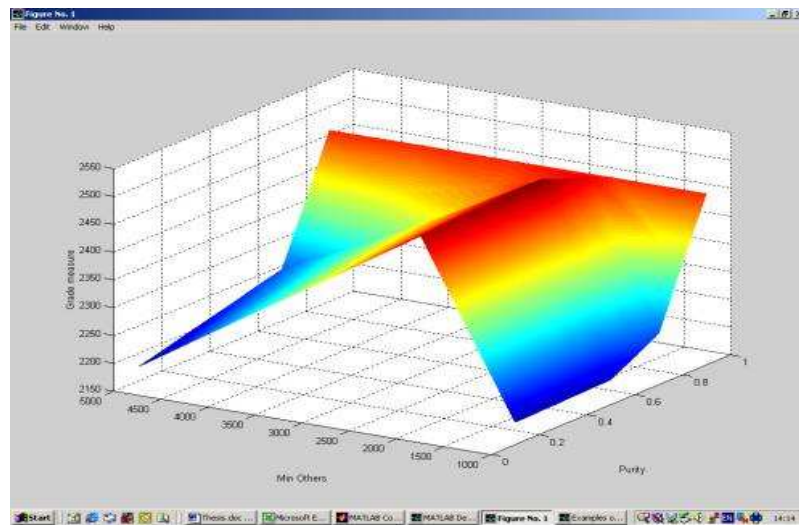


Figure 16 -Grade measure for $Purity(class=1)$ and $Min-Others(Class=1)$ when the other class is eliminated

Figure 16 presents the grade measure surface as a function of $Purity(Class=1)$ and $MinWeight(Class=1)$, the other class parameters were eliminated by defining $Purity(Class=0)=0.99$ that is the natural stopping rule.

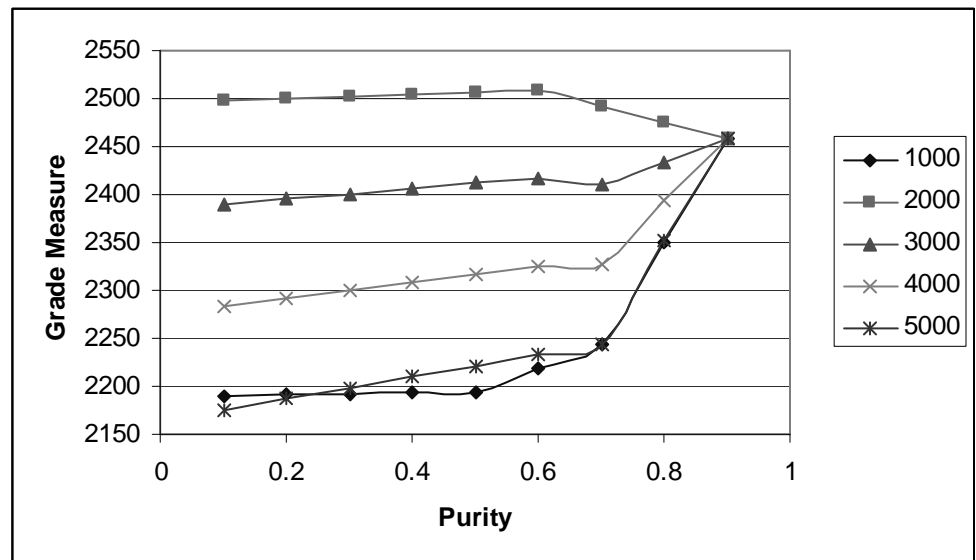


Figure 17 - Grade measure for $Purity(class=1)$ and different $MinOthers(Class=1)$ when the other class is eliminated

Examining the projection of

Figure 16 on the purity dimension (

Figure 17) we can see that the results are sensitive to the min other class weight parameter. Results shown in Figure 15 are just the extreme values of the function while values between the boundaries ($1000 < \textit{Min-others} < 5000$) are not identical to other values. The surprising thing is that the behavior is inconsistent although all lines converge to the same value, the trend is not the same – for some lines the grade measure improves as purity increases while in other the opposite is true, performance are inferior when purity is higher.

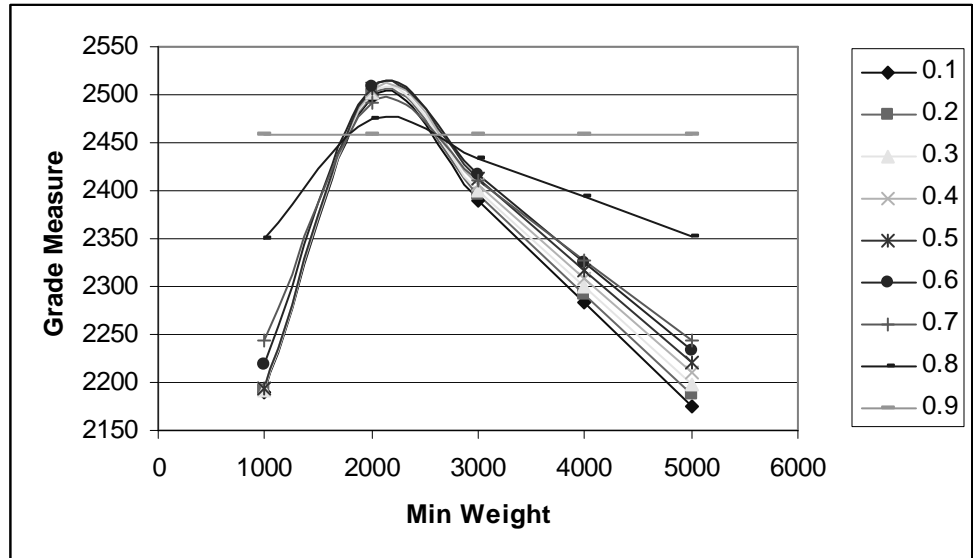


Figure 18 - Grade measure for $Purity(class=1)$ and different $MinOthers(Class=1)$ values when the other class is eliminated

The optimum around a certain $Min-others$ value is clear when looking at the projection of

Figure 16 on the $Min-others$ dimension (Figure 18). For almost all values of purity the peak is around $Min-others= 2000$. It is also clear that results in Figure 15 are not true for other values of $Min-others$. For the optimal value of the $Min-others$ parameter performance is less sensitive to the value of the $Purity$ parameter.

At first sight, it seems surprising that worst results are obtained for the optimal value of *Min-others* and high *Purity* values – one would expect that high purity values will always produce decision trees with better accuracies since they are more consistent with the training set. The explanation for that is that for the high Purity levels results are less sensitive to *Min -others* parameter, high purity values are rare and with the combination of other the *Min-others* parameter the early stopping rule is never assigned as true so the only stopping rule is the natural stopping rule of 100% purity.

The best performance is obtained for combination Purity=0.6 and *Min-others* = 2000.

4. *Min-Weight at each split (node)*

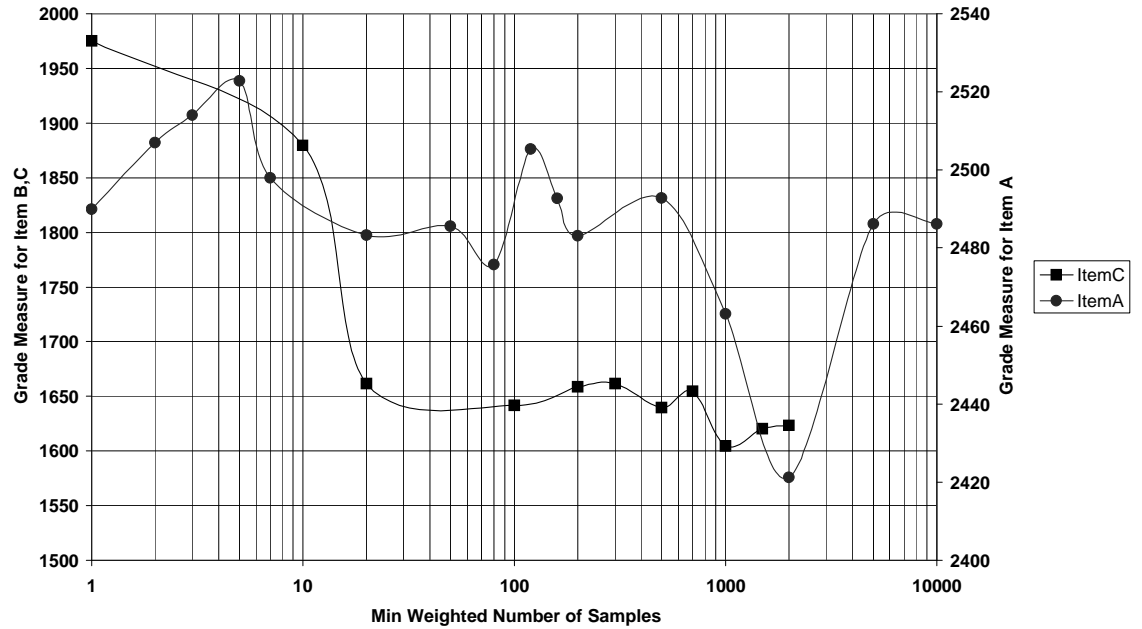


Figure 19 – Grade measure for items A & C when using early stopping = *Min weight*

The minimum weighted number of samples early stopping rule is presented above in Figure 19. The x axis is shown in logarithmic scale to capture all the data and still be able to distinguish between different x axis values.

At first sight no consistent patterns can be drawn from the above chart.

For item A an unexpected peak appears at threshold=5. But for item C the peak appears at threshold=1. The trend of the charts is also unexpected.

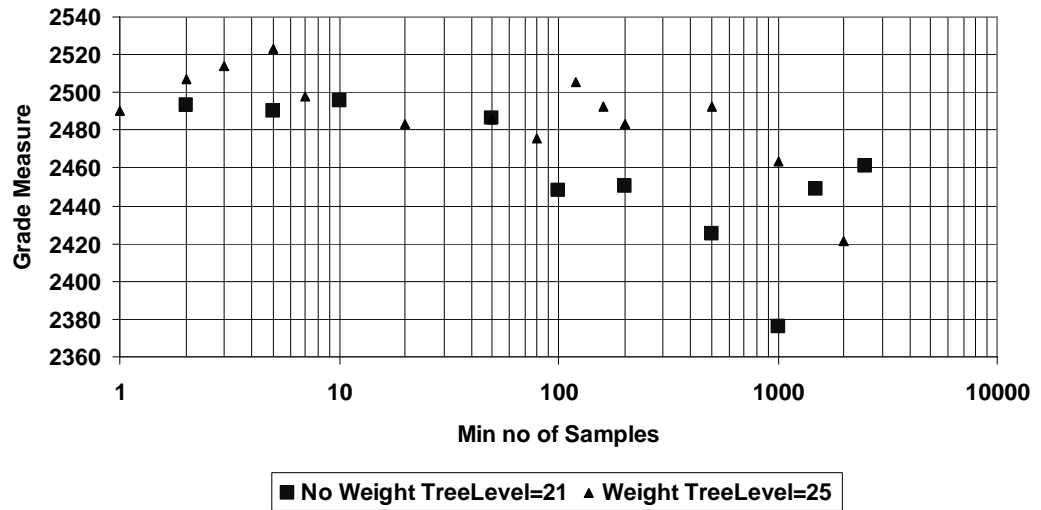


Figure 20 – Grade measure for item A when using early stopping = Min number (weighted and un-weighted) of samples

The un-weighted number of samples has the same noisy appearance (see

Figure 20).

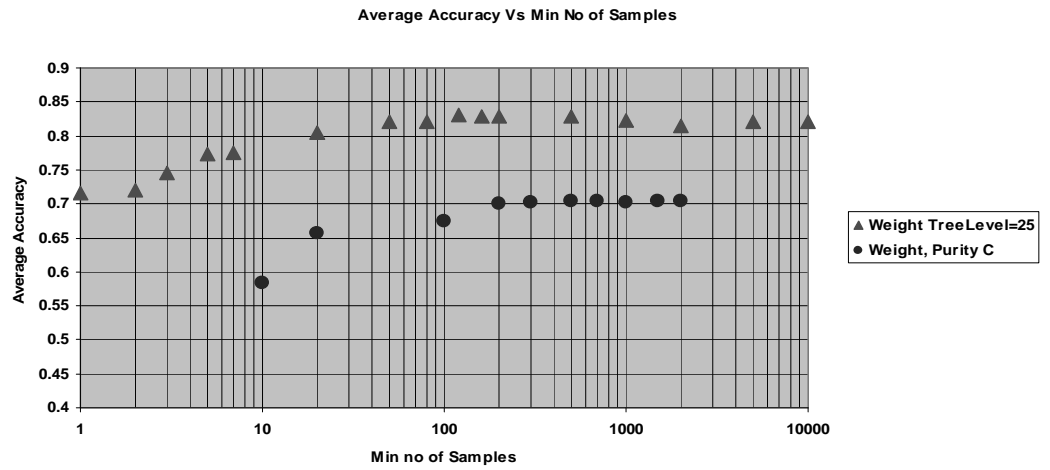


Figure 21- Grade measure for item A when using early stopping = Min number (weighted and un-weighted) of samples

Average Accuracy Vs Min No of Samples

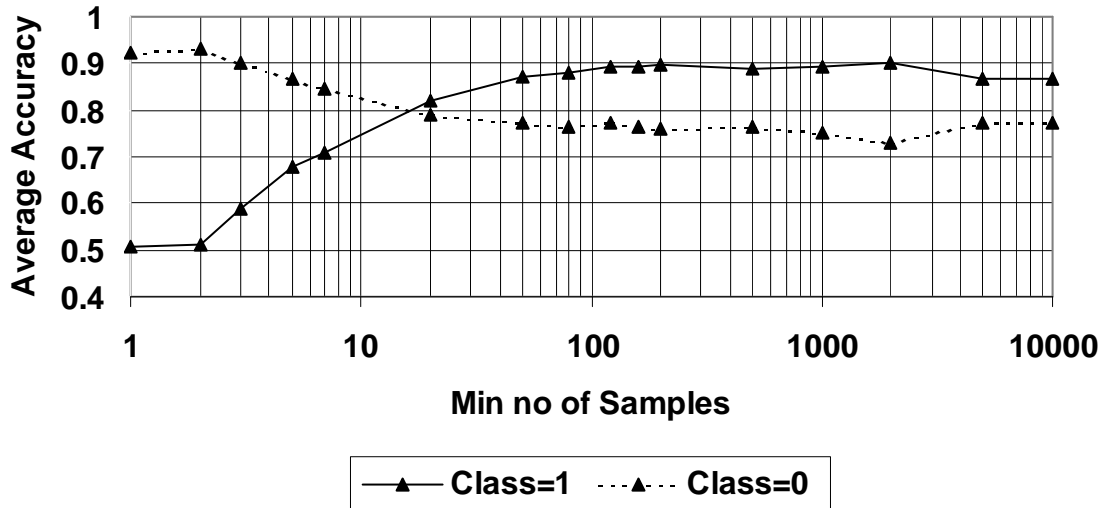


Figure 22- Accuracy for items A when using early stopping = *Min weight*

Average Accuracy Vs Min No of Samples

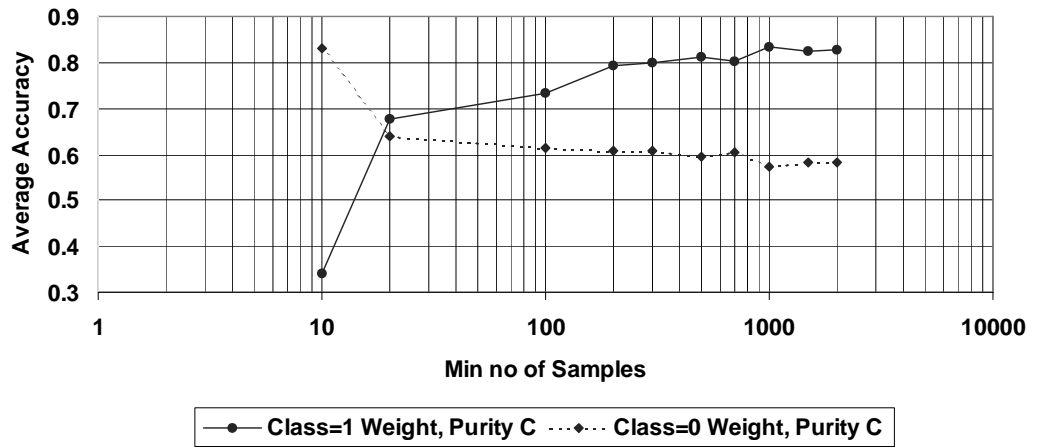


Figure 23- Accuracy for items C when using early stopping = *Min weight*

Only after examining the dependency of the average accuracy in the minimum weighted number of samples (see Figure 21) we can get better understanding of this stopping rule. The higher the threshold the better the average accuracy for both item A and item C.

Drilling in Figure 21 we look at Figure 22 & Figure 23 that show the ingredients of the accuracy measure, accuracy of predicting each class. One can learn that the accuracy rate for the Class=1 (Positive) and Class=0 for both items A and C has similar, rather smooth, behavior. Notice that for both items the accuracy rates of the common class (class=0) is superior to the rare class (class=1) at *min weight* lower than 10. This is because the trees were grown to unlimited depth and as seen previously in Figure 9 &

Figure 10, full grown trees have better accuracy for class=0 (the common class). For low tree depth the opposite is true – they have better accuracies on the rare class (class=1).

The noisy pattern in the grade measure charts (see Figure 19,

Figure 20 & Figure 21) is because the sensitivity of the grade measure to true positive value. The small bouncing in the positive class accuracy rate in Figure 21 & Figure 22 is amplified as a result of the cost matrix values to noisy patterns.

It seems that *Min-weight* stopping rule can improve predictive accuracy if it is used with tree depth stopping rule. Tree depth stopping rule is the starting point of the Min-weight stopping rule chart, choosing starting point at low tree depth can bring to interesting results.

4. Minimum Grade stopping

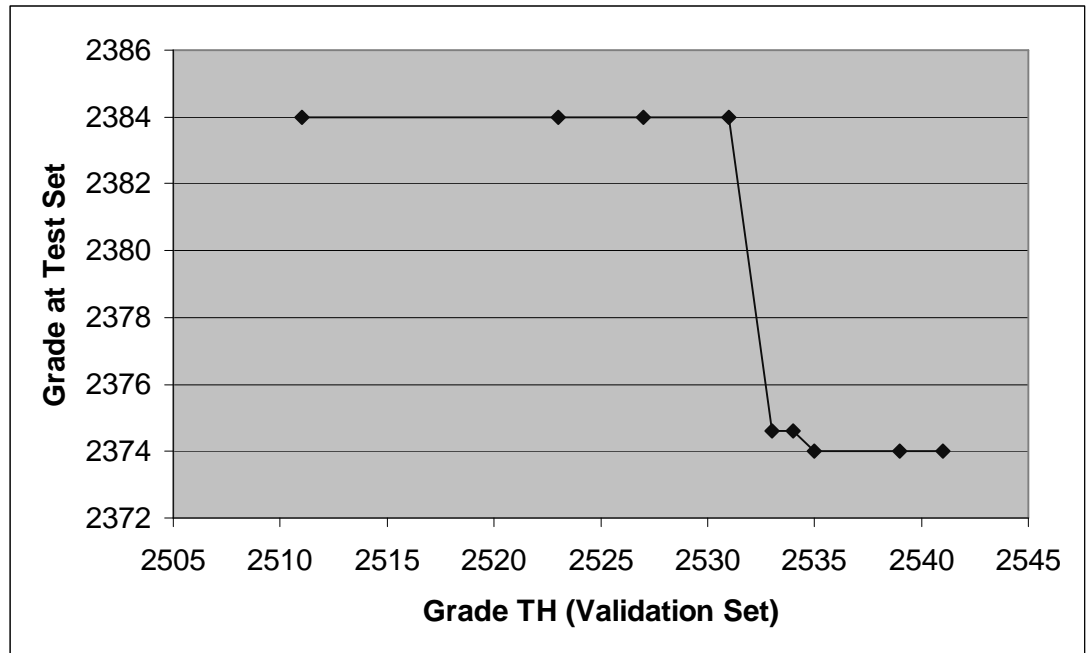


Figure 24 – Grade based stopping rule.

In Minimum grade stopping rule the tree is grown until it reaches a certain target grade on a validation set, the performance is measured on different, unseen, test set (the same approach of training set, validation set and test set appears in pruning algorithms).

One problem in this method is that the grade measure is sensitive to the training set parameters meaning that in the used data mining application, results were strongly dependent on the item, for item A (grade measure between 2594-2186), item B (grade measure between 580-558) and item C (grade measure between 1978-842). The range of the grade measure is between 2594-558 but each item is in located in different parts of the scale.

Another thing is the insensitivity of the method that is demonstrated in Figure 24, only two different grades are obtained for a wide range of target grade.

It is still not enough evidence to fully understand grade based stopping, it seems that in the noisy world of data mining samples it does not work well but still it is worth checking it in more “smooth” data sets. The different range between the validation set and test set implies that the training set has different statistical properties.

Experimental results – Pruning

Pruning did not increase predictive accuracy and it did not help to improve performance in the grade measure method either. Those results are consistent with results reported in other papers that studied the effect of pruning in different conditions and data sets.

In both item A and C (Figure 25 & Figure 26) the performance of pruning on test set were even less that the maximum value. For item C pruning results were even worst then full grown tree.

The combination of pruning with other early stopping rule does not have any benefits either (Figure 27).

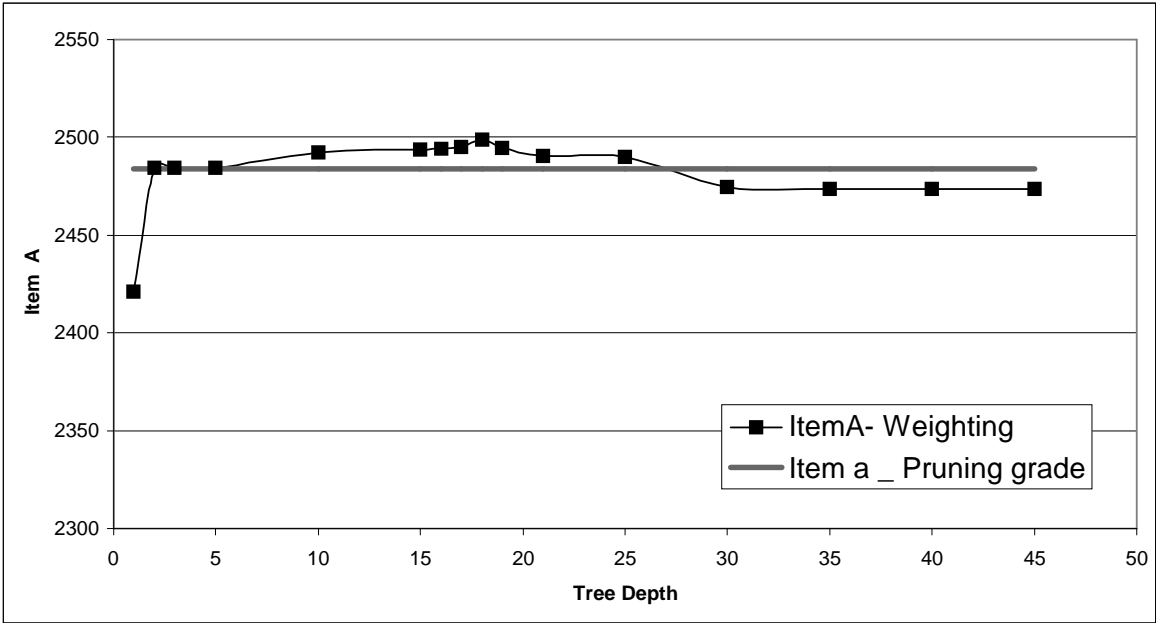


Figure 25 – Item A, Pruning results & Tree Depth stopping

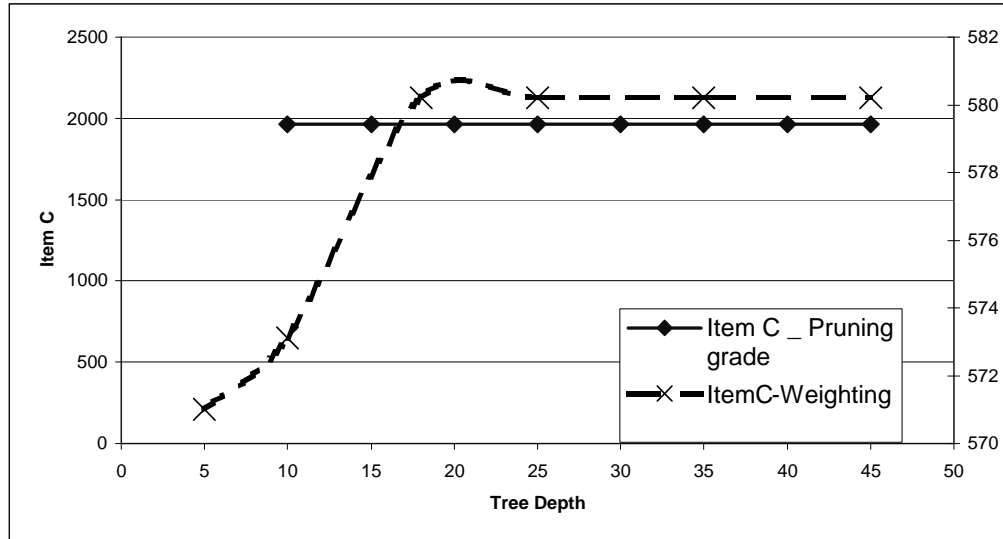


Figure 26 – Item C, Pruning results & Tree Depth stopping

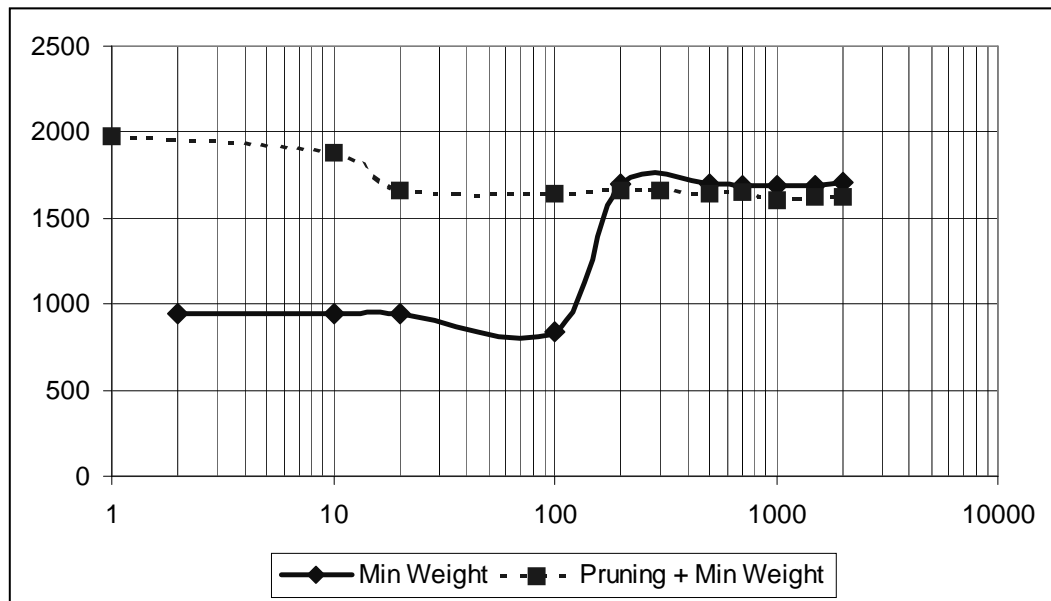


Figure 27 – Item C, Pruning results & Tree Depth stopping

Experimental results – combinations of methods

Additional results are also presented to study the interaction between different methods aiming to find a combination of stopping rules that will yield better results than previously presented when using each early stopping rule and pruning.

Figure 28 presents the combined effect of applying the following stopping rules:

1. Tree Depth
2. Relative purity (notice that relative purity gets two parameters per each class : relative class purity and *min-others*. For the common class the parameters were set to act as natural stopping conditions)
3. Min weight (Minimum weighted number of samples at each split)

As seen previously in Figure 18, high purity is inferior for lower values of purity. Trees grown with early stopping rule of $\text{purity}(\text{class}=1)=0.5$ achieved better grades for two different values of min weighted number of samples. The relative purity stopping rule outperforms natural stopping condition particularly for optimal tree depth.

Adding early stopping rule of $\{\text{Min-weight} = 10\}$ to $\{\text{Purity}(\text{class}=1)=0.5\}$ hurts tree performance and results are getting close to the basic tree depth stopping rule.

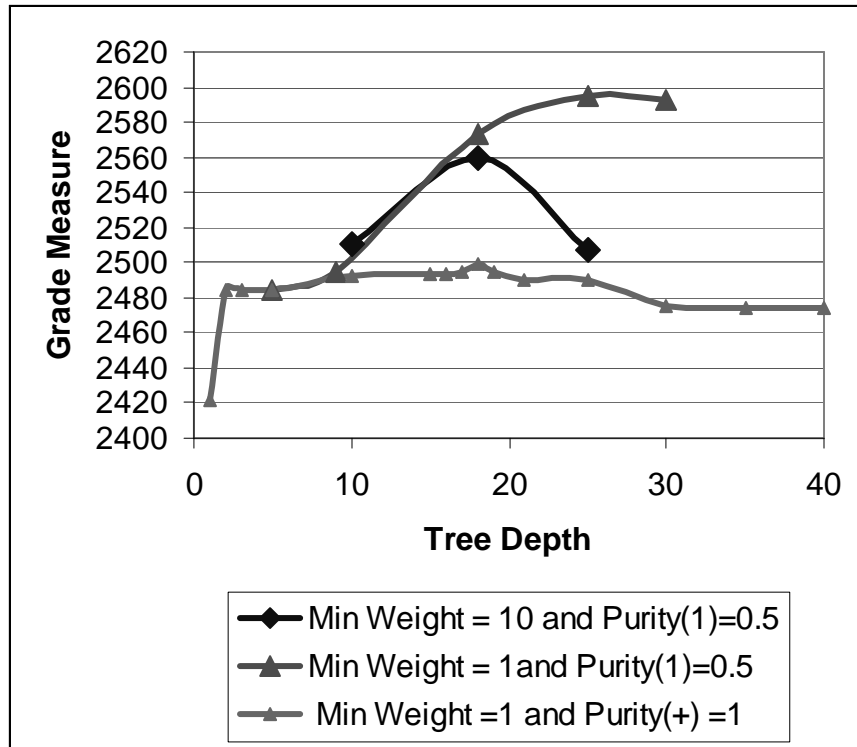


Figure 28 – Item A, Grade measure Vs. Tree depth for two different relative purity values

Tree Depth	Min Weight	Class Purity	Min-Others	Grade Measure
18	10	"+=0.5 -=0.99"	"+=1000 -=1	2511.93
18	10	"+=0.5 -=0.99"	"+=500 -=1	2512.91
18	10	"+=0.5 -=0.99"	"+=3000 -=1	2511.75

Table 4 – Minimum Other Class Weight effect in combination of Tree depth, Minimum Weight and Relative purity

Tree Depth	Min Weight	Class Purity	Min - others	Grade Measure
18	10	"+=0.5 -=0.99"	"+=3000 -=1	2511.75
18	10	"+=0.4 -=0.99"	"+=2000 -=1	2505.83
18	10	"+=0.6 -=0.99"	"+=2000 -=1	2509.32

Table 5 –Class Purity effect in combination of Tree depth, Minimum Weight and Relative purity

Sensitivity study around tree depth = 18 was performed and it was found that results were not sensitive to *Min-Others* (Table 4) nor to relative class purity (Table 5). The former contradicts results in

Figure 17 that show strong dependency in the *Min-Others*. The reason for that can be the interaction with tree depth early stopping rule that was not applied on top of the relative purity halting condition.

The insensitivity to the relative purity value is consistent with Figure 18, close values of purity around 50% don't have significant influence the measured grade.

Chapter 6

Conclusions

It is very important to optimize decision trees induction and classification processes since data mining applications usually deal with very large data bases. Early stopping conditions have good potential of reducing both training time and increasing predictive accuracy. Since early stopping rules stop the tree induction process before full growing the tree they can save time when compared to pruning algorithms that full grow the tree and then perform a process of optimization that is also time consuming. Early stopping can also reduce overfitting of the model to training data.

This work is additional evidence that pruning can not always improve tree induction as most papers claim. These results are consistent with other researches results (Murthy, Martin & Hirschberg, Schaffer, Murthy & Pazzani) that empirically showed that pruning injures the performance of decision trees in problems that are difficult to learn. These researches didn't explore the early stopping conditions, their work focused on evaluating pruning performance. This work showed that simple early stopping rules can both improve predictive accuracy and save training time for data sets drawn from real business data. One reason for this that the full grown tree depth can reach the depth of 50 binary levels and most of the leaves are constructed from one sample only. Since the validation set is from the same order of the training set the probability of covering all the leaves in the full grown tree is very low. And furthermore, if a validation sample reached to a leaf that his label (class) was determined by only one sample

from the training set then there is no way to distinguish between them correctly, each one of them (or both) can be the false (noisy sample) or true class. Using larger sets of validation sets can improve the pruning performance.

The results also imply that early stopping conditions improve the performance of the classifier. A cocktail of early stopping conditions can outperform full grown tree or pruned tree. The most effective early stopping rule is *purity* and *min-others*, pruning didn't have improvement in the performance. I must emphasize that the effect of early stopping rules was only tested on a specific kind of data, which is not necessarily similar or typical to other data sets.

The results presented here are sensitive to the values of the “**Cost Matrix**”. In data mining applications it is accepted to apply different types of costs. The use of the “**Cost Matrix**” to weight the samples was only recently studied. Much of the effort is aimed to find a way to imply “**Cost Matrix**” into existing decision trees application.

It is also important to define general criteria for improving tree induction process. The new generation of data mining applications is integrated in the IT systems of the business. The data mining engine is automatically extracting information from vast amount of data (previous data mining application required expert users to retrieve data mining results from data). The independent data mining engine needs to be set up to optimize performance and afterwards to analyze data independently. This work implies that “good” stopping rules for data sets similar to the data sets presented before then the results can be defined and the “**Cost**

Matrix” values are calculated based on the natural frequency of the data (class distribution).

In order to generalize the results and defining how to determine best early stopping rule further work is needed to examine different values of cost matrix values and different type of data sets and sensitivity of the pruning process to the size of the validation set.

BIBLIOGRAPHY

1. A study of cross validation and bootstrap for accuracy estimation and model selection. Ron Kochavi
2. Automatic construction of decision trees from data: A multi-Disiplinary survey. Sreerama K. Murthy
3. The time complexity of decision tee induction. J. Kent Martin and D.S. Hirschberg
4. An exact probability metric for decision tree splitting.
5. When Does overfitting Decrease prediction accuracy in inducted decision tree and rule sets. Cullen Schaffer
6. Exploring the decision forest. An empirical investigation of occam's razor in decision tree induction. Patrick M. Murphy and Michael J. Pazzani.
7. Feature Selection via discretization. Huan Liu and Rudy Setiono
8. Occam's two razors: The sharp and the blunt. Pedro Domingo 1998
9. A comparative analysis of methods for pruning decision trees.
10. The biases of decision tree pruning strategies. Tapio Elomaa.
11. Overfitting avoidance as bias. Cullen Schaffer.1992
12. Further experimental evidence against the utility of Occam's Razor Geoffrey Webb 1996.
13. Efficient C4.5 Salvatore Ruggieri
14. Microsoft Research Pattern Recognition - Classification and regression trees
15. Goal Directed classification using linear machine decision trees – Bruce A. Draper, Carla E. Brodley, Paul E. Utgoff.
16. Types of cost in inductive concept learning- Peter Turney
17. Bootstrap methods for cost sensitive evaluation of classifiers – Margineantu & Dietterich
18. Cost Sensitive specialization – Geoffrey I. Webb
19. the effect of class distribution on classifier learning: an empirical study – G.M Weiss, F. Provost
20. The Foundations of cost sensitive learning – Charls Elkan
21. Learning with non uniform class & cost distributions: effects & multi classifier approach – Philip K. Chan & Salvatore Stolfo
22. Design and analysis of experiments – D.C. Montgomery.
23. Smith, Chris. Theory and the Art of Communications Design. State of the University Press,