GENERALIZING THE NUMBER OF STATES

IN

BAYESIAN BELIEF PROPAGATION,

AS APPLIED TO

PORTFOLIO MANAGEMENT

Jan Walters Kruger

KRGER-JW@SOREX.VISTA.AC.ZA

Degree of Master of Science degree by coursework

'A research report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfillment of the requirements for the degree of Master of Science.'

Johannesburg, 1996.

DECLARATION

I declare that this research report is my own, unaided work. It is being submitted for the degree of Master in Science in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

Jan Kruger

18th day of MARCH 1. '.7

ABSTRACT.

This research report describes the use of the Pearl's algorithm in Bayesian belief networks to induce a belief network from a database. With a solid grounding in probability theory, the Pearl algorithm allows belief updating by propagating likelihoods of leaf nodes (variables) and the prior probabilities.

The Pearl algorithm was originally developed for binary variables and a generalization to more states is investigated.

The data used to test this new method, in a Portfolio Management context, are the Return and various attributes of companies listed on the Johannesburg Stock Exchange (JSE).

The results of this model is then compared to a linear regression model. The bayesian method is found to perform better than a linear regression approach.

Acknowledgements

I wish to acknowledge the assistance given to me by my supervisor, Prof. David Lubinsky, of the department Computer Science, University of the Witwatersrand.

CONTENTS   Page

List of Figures

List of tables

Chapter 1

INTRODUCTION

1.1 Goal:

The objective of the research was to modify Pearl's [Pearl(3)] algorithm to be appropriate for domains with large numbers of numeric variables, with low correlation coefficients and then compare this empirically to Linear Regressi to do this I developed a heuristic to generalize the number of states of variables.

1.2 Method:

The main result of this report is a modified version of the Pearl [Pearl(3)] algorithm to extract generic models from data. The Pearl [Pearl(3)] algorithm is a fast, O(nlogn) [Ghinea (2), Pearl(3)], algorithm that gives a Bayesian tree network characterizing the probabilistic structure of the input data.

This method gives us the Belief[1] and the probability distribution of the states of each variable. We will use the discrete distributions (relative frequencies of discrete states) as prior distributions. As an example, the posterior probability distributions can be used to find the Expected Return and the Risk (standard deviation of Return) of shares.

---

[1] In this research report Belief is the probability that a variable is in a certain state. This is synonymous to probability. Not to be confused with the belief function as used by Dempster-Shafer [Shafer (8)].

1.3 Motivation:

An essential part of intelligent behavior is the ability to learn from observations and modify our understanding of the world [Lubinsky(4)].

The close relationship between Statistical Methods, specifically Multiple Regression, and the Tree Method used by the Pearl algorithm [Pearl(3)] could lead to some points of synergy. Both these methods estimate the value of a variable from data. The Pearl algorithm estimates the discrete distribution of this variable, while Multiple Regression gives an estimate and under certain assumptions also the standard error.

The Multiple Regression model is an additive model. The model is given as: $y = X\beta + e$, where y is a vector of dependent variables, X is a matrix of independent (predictor) variables (that we get from data), β a vector of parameters, or indices, and e a vector of error terms. It is usually assumed that the error, or random shock terms have a normal distribution, with mean of zero and a standard deviation of σ [Elton and Gruber (5)]. In this research report, I compare this with the Pearl algorithm.

The Pearl algorithm uses the correlation between variables in order to find the latent structure (hidden variables), by applying the law of Total Probability to factor a joint probability distribution according to the Markov properties (d-separation) of a graph. Using these graphs the algorithm predicts the distribution of the dependent variable. Lazarfeld(1) calls the causal variable(s) the latent structure and shows how to use a causal variable to stratify the dependent variable to get significant dependencies. I explain the algorithm in the next chapter.

I found no reference to any empirical comparison between the Pearl algorithm and Multi-Regression models. This is surprising. Regression analysis is one of the core techniques in the economic and social sciences. The implications of any method that could compete with Regression analysis could have major implications on these sciences.

The Belief nets constructed by the Pearl algorithm have richer structures than multiple regression. Multiple Regression only allows a single parent, whereas the Pearl algorithm allows a tree structure.

1.4 Overview of chapters:

Chapter 2, covers an overview of some of the literature on Bayesian decision trees.

Chapter 3, the main section is on the development of a heuristic to generalize the number of states.

Chapter 4, is on the software developed.

Chapter 5, gives the results found by applying the adapted algorithm on real Stock Exchange data.

Chapter 6, gives the conclusions.

1.5 Development during the research

When I started the research, I wanted to apply the Pearl algorithm to Stock exchange data. This kind of Stock exchange data is notoriously difficult to model, with a high degree of accuracy. [Elton and Gruber (5)]

The Pearl Algorithm only works for binary variables. Lazarfeld [Lazarfeld(1)] proves that the variables available are insufficient to calculate the parameters, where each variable has more than two states. The research then turned to finding a heuristic to bypass this problem.

The main contribution of this research is a modification of the Pearl Algorithm to handle more than two states. For this, I developed a heuristic to calculate the parameters of the elements in the link matrices and tested the heuristic on Stock Exchange data. The variables available all have low correlations to the dependent variable (Return).

The heuristic developed, works well in certain Belief nets. Selecting Belief nets based on the performance on a training set of data, eliminates the Belief nets where the heuristic did not perform well. The heuristic performed similarly on both the training set and on the testing set, but I could not find rules to explain why the heuristic sometimes performed poorly in calculating the link matrices.

Chapter 2

LITERATURE STUDY

2.1 Literature on Bayesian Decision Trees.

2.1.1 First an explanation of Bayes' theorem:

This section starts with an elementary approach to the Bayesian theorem. As this section develops, I add more advanced concepts.

The discrete form of Bayes' theorem

Start with a set A of mutually exclusive disjoint events $A = \{A_1, A_2, A_3, ...A_t\}$; whose union has probability one. These disjoint events form a partitioning of the set A. Assume the prior probabilities $p(A_i)$, for all the events, A_i, known. Then event B occurs, for which $p(B \mid A_i)$ (the conditional probability B given A_i) is known for each A_i.

Bayes' theorem states that:

The posterior distribution of A_i, $p(A_i \mid B) = p(B \mid A_i)\, p(A_i) / \sum_{j=1}^{n} p(B \mid A_j)\, p(A_j)$,

with the proviso that B is not empty, therefore $p(B) \neq 0$.

These probabilities reflect our revised opinions about the A_i, in light of the fact that B has occurred.

Berger [(12)] gives an excellent overview of hypothesis testing, risk functions and robustness of the Bayesian methods. The thesis by Musick[(6)] also shows how beta distributions propagate through the Bayesian formula from a-priori to a-posteriori distributions. In this report we find the relative frequencies in the different classes and propagate them.

Example:

Suppose we know that the prevalence of a certain disease A in the population is 1%. Suppose that a screening test to test for this disease shows positi e in 96% of infected individuals, but the test also shows positive in 7% of not infected individuals. After the screening test, take individuals who show positive on the test for further, more detailed observation.

Questions arising: Find the probability of an infected individual, if he shows positive in the test. How many individuals will show positive on the test, out of a population of 30 000 000?

Solution:

Define p(T) as the prior probability that the test will show positive for an individual.

Therefore:

let A_1 stand for the event of an infected individual, and

let A_2 stand for the event of a not infected individual.

$p(T \mid A_1) = 0.96$, $p(T \mid A_2) = 0.07$, $p(A_1) = 0.01$, $p(A_2) = 0.99$

This gives the probability that an individual will show positive on the screening test:

$p(T) = \sum_i p(A_i)\, p(T \mid A_i)$, where T is partitioned according to event i.

$= 0.96 \times 0.01 + 0.07 \times 0.99$

$= 0.159$

What about t probability of an infected individual, if he shows positive in the test?

From Bayes' theorem:

$$p(A_1 | T) = p(A_1)\, p(T \mid A_1) \,/ \sum_i p(A_i)\, p(T \mid A_i)$$

$$= 0.96 \times 0.01 / 0.159$$

$$= 0.604$$

Matr[illegible] representation:

We will use capital letters, P, to represent matrix p[illegible]babilities. In the above example:

Table 2.1 Matrix representation

	A_1	A_2
Hypothesis: Test positive	0.604	0.395
Hypothesis: Test negative	0.356	0.654

$P(A | T) =$

$$\begin{bmatrix} p(A_1 | \text{Test positive}) & p(A_2 | \text{Test positive}) \\ p(A_1 | \text{Test negative}) & p(A_2 | \text{Test negative}) \end{bmatrix}$$

$P(A | T) =$

$$\begin{bmatrix} 0.604 & 0.396 \\ 0.346 & 0.654 \end{bmatrix}$$

$P(A | T)$ is a Stochastic Matrix.

The second part of the question: How many individuals will show positive on the test, out of a population of 30 000 000?

The answer is 30 000 000 $p(T)$ = 30 000 000 x 0.159

= 4.77 million

2.1.2 Graphical models / Belief Networks:

Graphical representations offer a framework for representing and reasoning with probabilities and independencies. The nodes in these graphs represent propositional variables, and the arcs represent local dependencies [Pearl(3)].

The basis for Graphical models is the notion of independence [Buntine(11)]:

Definition 1:

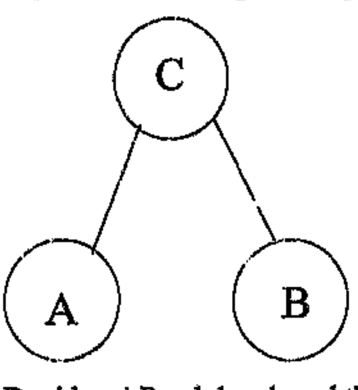
Two events, A and B, are independent if $p(A, B) = p(A)\,p(B)$,

where $P(A, B) \equiv P(A \cap B)$[2].

Definition 2:

A is independent of B given C if $p(A, B \mid C) = p(A \mid C)\, p(B \mid C)$, whenever $p(C) \neq 0$, for all A, B, and C.

Figure 2.2 Graphical representation of definition 2



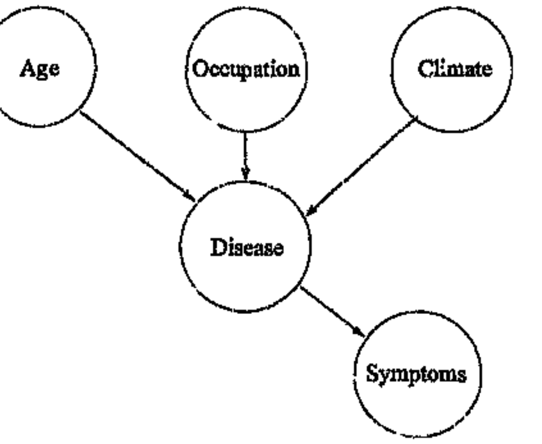
David and Pearl developed the theory of independence as a basic tool for knowledge structuring [Buntine(11), Pearl(3)]. W↑ can equate a graphical model with a set of probability distributions that satisfy the implied constraints. Graphical models are

---

[2] To indicate that sets can be events, Pearl[(3)] and Musick[(6)] prefer to use the logical AND, $\wedge$ instead of the intersection, $\cap$. In this report the intersection, $\cap$, will be used.

equivalent probability models if their corresponding sets of satisfying probability distributions are equivalent.

Figure 2.3 Graphical representation of a simplified medical application



The Figure above [Buntine(11)] is an example of a graphical model of a simplified medical problem. The variables Age, Occupation and Climate are variables that cause Disease. The Disease on the other hand causes the Symptoms.

A directed graph is a graphical model, where all the arcs have direction. Indicate direction in the Figure above: if C is the parent of A and B. A directed graph without cycles is a directed acyclical graph (DAG).

Definition of a belief network

Musick [(6)] defines a **belief network or Bayesian network** as: "A directed acyclical graph consisting of a set of nodes X_i for i in 1...n, a conditional probability table T_i for each node, and a set of directed arcs between the nodes. Each node represents a random variable, and has an associated, possibly infinite, domain $x_{ik} \in D_{xi}$. An arc from X_i to X_j represents a dependency between the two, and establishes X_i as the parent of its child X_j. The conditional probability table (CPT) {An example of a CPT given in the next Section} of every node in the network represents the set of conditional probabilities $Pr(X_i| \Pi_i)$, where Π_i represents the set of parents of X_i. Finally X_i is conditionally independent of every other variable in the network given its parent instantiations (By instantiation we mean that the variable is set to a given value), or $Pr(X_i | \Pi_i, X_j) = Pr(X_i | \Pi_i)$ for any $X_j \neq X_i$ and $X_j \notin \Pi_i$."

Example of a Conditional Probability Table (CPT)

The easiest way to explain what a CPT is, is with an example [Musick(6)]

Represent "Parents Smoke" by P, "Smoker" by S and "Disease" by D.

P has the disjoint possibilities 0, that neither parent smokes, 1, that one of the parents smokes and 2 that both parents smoke.

S is the disjoint event that the patient either smokes or does not smoke.

D stands for the event that the diseases bronchitis, halitosis is present or that neither is present. In the example that follows assume that bronchitis and halitosis are disjoint events.

Table 2.2 Probability that the parents smoke

		p(P)
P	0	0.62
	1	0.23
	2	0.15

Table 2.3 Probability of a smoker given that the parents smoked

p(S | P)				
		P		
		0	1	2
S	0	0.61	0.57	0.53
	1	0.39	0.43	0.47

Table 2.4 Probability of disease given a smoker

p(D | S)			
		S	
		0	1
D	bronchitis	0.75	0.51
	halitosis	0.20	0.41
	no disease	0.05	0.18

Table 2.5 The joint Probability Table for the Smokers example

p(D, P, S)							
		P, S					
		0, 0	0, 1	1, 0	1, 1	2, 0	2, 1
D	bronchitis	0.28	0.13	0.10	0.05	0.06	0.04
	halitosis	0.08	0.10	0.03	0.04	0.02	0.03
	no disease	0.02	0.04	0.01	0.02	0.00	0.01

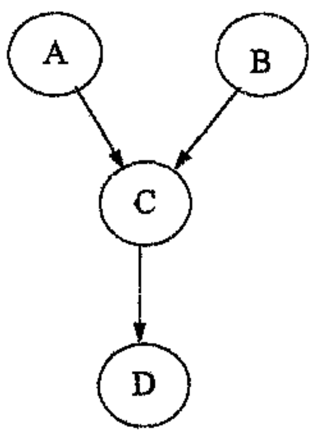
Tables 2.2, 2.3 and 2.4 are examples of conditional probability tables (CPTs). Table 2.5 is a joint probability table.

Directed Acyclical Graphs (DAG)

A Bayesian network is a graphical model that uses directed arcs exclusively to form a directed acyclic graph (DAG).

The forward conditional probabilities give the strengths of the influences between nodes.

Figure 2.4 A simple Bayesian network



The decomposition works as follows:

$p(A, B, C, D) = p(D \mid C)\ p(C \mid A, B)\ p(A)\ p(B)$

If parent(x) is the set of variables with a directed arc into x

[Buntine (11)]:

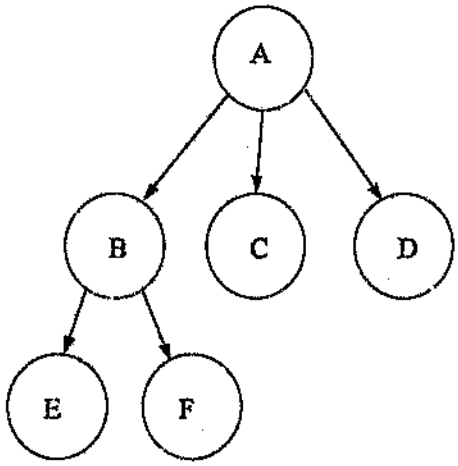
$p(X \mid M) = \prod_{x \in X} p(x \mid \text{parent}(x), M)$, where M is the conditioning context, and $x \in X$.

This is the interpretation of a belief network used in this research report.

2.1.3 Decision Trees

A Bayesian decision tree is a belief network, where every node has only one parent, except the root node that does not have a parent.

Figure 2.5 A tree showing the different levels



In Figure 2.5 [Turban(10)], Nodes A, B, C, D, E and F form a tree. Node A is the **root** and is on level 0. Node A is the parent of Nodes B, C and D on level 1. Nodes E and F are on level 2. The leaf nodes are E, F, C and D.

As can be seen from the section on evidence propagation below, it is convenient to use a Bayesian tree rather than a network that can have multiple parents for nodes. Pearl gave a few methods to deal with multiple-connected nets.

2.1.4 Pearl's algorithm for evidence propagation

The belief of x, BEL(x), is defined as: $P(x \mid e)$, where e is the combined effect of all instantiated variables.

BEL(x) is a vector, where each component represents the belief that x is in that state. Represent the belief that the weather tomorrow will be x_1 = sunny, x_2 = cloudy, x_3 = raining by the vector
$BEL(x) = (BEL(x_1), BEL(x_2), BEL(x_3))$, where x is the vector (x_1, x_2, x_3).

The likelihood vector, or **diagnostic support**, $\lambda(x)$, is $P(e \mid x)$. The likelihood vector is the probability that the evidence, e, will occur given the state of the vector x.

If we have **two nodes**, x and y, with a single arc connecting x to y, such that x is the parent of y, then from the Bayes rule:

Indicate the evidence going from the child, y, to the parent, x, by e, then

$BEL(x) = P(x \mid e) = \alpha\ P(x)\ P(e \mid x) = \alpha\ P(x)\ \lambda(x)$,

where $\lambda(x) = P(e \mid x)$,

$P(x) =$ the prior probability of x

and $\alpha = [P(y)]^{-1}$, a normalizing constant.

NOTE: Under multiplication of vectors we understand term by term multiplication and not the scalar product.

With a **chain** of three nodes $\{x, y, z\}$:

$BEL(x) = P(x \mid e) = \alpha\ P(x)\ \lambda(x)$, as above, but

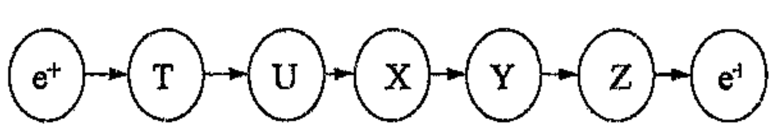
$$\lambda(x) = P(e \mid x) = \sum_{y} P(e \mid y, x)\ P(y \mid x) = \sum_{y} P(e \mid y)\, P(y \mid x) = P(y \mid x)\, \lambda(y)$$

and $\lambda(y) = P(z \mid y)$, the likelihood vector of the middle node.

$P(y \mid x)$ is the transitional probability matrix.

This equation holds for a chain of any length. We can now propagate the likelihood against the arc direction to get the belief of a node on a lower level. To propagate the evidence in the arc direction we use a separate vector.

Figure 2.6 A chain with evidential data at its head (e^+) and tail (e^-)



In Figure 2.6 , let e be the total evidence available. It is more convenient to separate the impact of the evidence into e^+, and e^-. Where e^+ can be considered as the background knowledge one has about T, in which case the prior probability P(t) will be identical to $P(t \mid e^+)$.

If node u is the parent of node x:

$$\pi(x) = P(x \mid e^+) = \sum_u P(x \mid u, e^+)\ P(u \mid e^+) = \pi(u)\ P(x \mid u)$$

which is a vector of prior probabilities propagated down from the parent, or the **causal support.**

The belief now becomes:

$$\begin{aligned} BEL(x) &= P(x \mid e^+, e^-) = \alpha\ P(e^+ \mid x, e^-)\ P(x \mid e^-) = \alpha\ P(e^+ \mid x)\ P(x \mid e^-) \\ &= \alpha\ \pi(x)\ \lambda(x) \end{aligned}$$

Consider a node x, with m children $y_1, y_2, \ldots, y_m$, and parent u:

The rest of Section 2.1.4 considers a tree structure, where a typical node x, has m children $y_1, y_2, \ldots, y_m$, and parent u.

Belief updating:

As with the chain structure above; Using the causal support, $\pi(x)$, and diagnostic support, $\lambda(x)$, the belief distribution of x is:

$$BEL(x) = \alpha\ P(e^+ \mid x)\ P(x \mid e^-) = \alpha\ \lambda(x)\ \pi(x),$$

The difference comes in the way that $\lambda(x)$ is calculated.

$\lambda(x) = P(e^- \mid x) = P(\text{evidence from the children} \mid x) = \prod_i P(y_i \mid x)$, because x separates its children and the siblings are conditionally independent.

Therefore,

$$\lambda(x) = \prod_{i=1}^{m} \lambda_{yi}(x)$$

and $\pi(x) = \sum_{u} P(x \mid u)\, \pi_x(u) = \pi(u)\, P(x \mid u)$

and α is a normalizing constant to make $\sum_{x} BEL(x) = 1$.

Bottom-up propagation:

Node x uses the $\lambda(x)$ message from the children to compute a new message $\lambda_x(u)$ to send to its parent u.

$\lambda_x(u) = \lambda(x)\, P(x \mid u)$

Top-down propagation:

The new π message sent by node x to its j-th child y_j is:

$$\pi(y_j) = \alpha\, \pi(x) \prod_{k \neq j} \lambda_{yk}(x)$$

but,

$BEL(x) = \alpha\, \lambda(x)\, \pi(x)$, therefore $\pi(y_j) = \alpha\, BEL(x) / \lambda(y_j)$.

Readers interested in Belief propagation in more general singly connected networks are referred to Pearl(3).

From the above it is evident that we only need the prior probabilities of the root and the likelihood functions of the leaf nodes to calculate the belief of all the nodes.

2.2 Structure

In this example, the Belief net is the network that is induced by the data. The Belief net will show associations, which may indicate causality. Before evidence can be propagated through the network, as discussed in the previous section, we must find the network. In Section 2.2.1 Pearl's Algorithm to find the Structure is discussed. In Section 2.2.2 conditions limiting the application of the Pearl Algorithm are investigated, followed by an alternative algorithm by Cooper and Herskovits to build and to evaluate a Belief Network in Section 2.2.3. The Pearl algorithm uses the idea of hidden variables, or Latent structure, to build a Belief Network, while the Cooper and Herskovits algorithm only investigates causality of the variables that are actually found in a database.

2.2.1 Pearl's Algorithm to find the Structure of the Belief net

We show below that if $p(a, b \mid c) = p(a \mid c)\, p(b \mid c)$, (i.e. if a and b are conditionally independent) with $p(c) \neq 0$, then

$\rho_{ab} = \rho_{ac} \cdot \rho_{cb}$. This means that node c lies between nodes a and b.

Proof of the product rule for correlations

Morrison [Morrison (8)] gives the conditional correlation between two variables a and b, given c, by:

$$\rho_{ab.c} = \frac{\rho_{ab} - \rho_{ac}\rho_{bc}}{\sqrt{(1-\rho_{ac}^{2})(1-\rho_{bc}^{2})}}$$

but, the variables a and b are independent given c, therefore

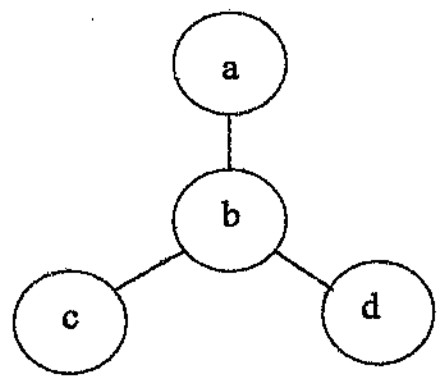
$$\rho_{ab.c} = 0$$

therefore

$\rho_{ab} = \rho_{ac} \cdot \rho_{cb}$.

This proves the product rule for correlations.

Figure 2.7 A star formation:



In Figure 2.7 the product rule for correlations gives:

$$\rho_{ad} = \rho_{ab} \cdot \rho_{bd}$$

$$\rho_{ac} = \rho_{ab} \cdot \rho_{bc}$$

and $$\rho_{cd} = \rho_{cb} \cdot \rho_{bd}$$

In other words any three nodes a, c and d can be presented in a star formation must have a centre node, b with the true correlations as indicated.

Solving by substitution gives:

$$\rho_{bd}^{2} = \rho_{ad} \cdot \rho_{cd} / \rho_{ac}$$

$$\rho_{ab}^{2} = \rho_{ac} \cdot \rho_{ad} / \rho_{cd}$$

$$\rho_{bc}^{2} = \rho_{ac} \cdot \rho_{dc} / \rho_{ad}$$

To link up a fourth node, e, in a star formation, and if the node links on to the,

ba- arc, then: $\rho_{ac} \cdot \rho_{de} = \rho_{ad} \cdot \rho_{ce}$

bc- arc, then: $\rho_{ac} \cdot \rho_{de} = \rho_{cd} \cdot \rho_{ae}$

bd- arc, then: $\rho_{ad} \cdot \rho_{ce} = \rho_{cd} \cdot \rho_{ae}$

If any two of these equations hold (then all three will hold) the common node will be the same as the old centre node.

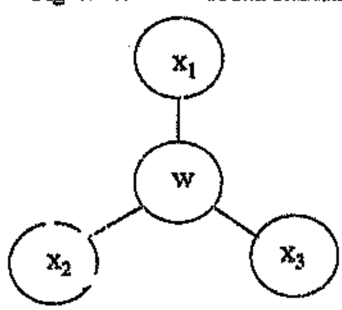
By following this reasoning any new node, that can be presented in a star formation, can be linked onto the belief network. Note this will give an undirected network. To create a tree structure, with direction, any node is selected as the root and all arcs are then directed away from this root.

Musick [(6)] claims that our ability to make qualitative judgements about which variables affect the value or state of another is good. In his dissertation he makes the assumption that the structure has already been defined by knowledge engineers. domain experts, or an automated structure discovery algorithm. Musick [(6)] wants to minimize manual input and do the database mining as an automated process, but he then assumes that a manual structure input must be given. The Pearl Algorithm can be used to find the structure.

2.2.2 Conditions for Star-Decomposability

Assume W is the central variable between the binary nodes x_1, x_2 and x_3.

Figure 2.8 A star formation:



Let $f_i = p(x_i = 1 \mid w = 1)$

and $g_i = p(x_i = 1 \mid w = 0)$

Define the probabilities as:

$p_i = p(x_i = 1)$

$p_{ij} = p(x_i = 1, x_j = 1)$

$p_{ijk} = p(x_i = 1, x_j = 1, x_k = 1)$

The standard deviation of a Bernoulli variable is $\sigma_i = [p_i (1 - p_i)]^{½}$.

and the correlation coefficients: $\rho_{ij} = (p_{ij} - p_i p_j)/ \sigma_i \sigma_j$

Theorem:

A necessary and sufficient condition for three binary random variables to be star-decomposable is that they satisfy the inequalities:

$$p_{ik}\, p_{ij} / p_i \leq p_{ijk} \leq p_{ik}\, p_{ij} / p_i + \sigma_j \sigma_k (\rho_{jk} - \rho_{ij} \rho_{ik})$$

for all i, j, k $\in$ {1, 2, 3} and $i \neq j \neq k$. When these conditions are satisfied, the parameters of the star-decomposed distribution can be determined uniquely, up to a complementation of the hidden variable W,

i.e., $w \to (1 - w)$, $f_i \to g_i$, $\alpha \to (1 - \alpha)$.

Where w is the state of W and α is the prior probability that w is in state 1.

The proof is given by Pearl[Pearl(3)], following previous work done by Lazarfeld [(1)]:

Assuming P is star decomposable then the joint-occurrence probabilities can be expressed in terms of α, f_i and g_i, then:

$$p_i = \alpha f_i + (1 - \alpha) g_i, \qquad i = 1, 2, 3$$
$$p_{ij} = \alpha f_i f_j + (1 - \alpha) g_i g_j \qquad i \neq j,$$
$$p_{ijk} = \alpha f_i f_j f_k + (1 - \alpha) g_i g_j g_k \qquad i \neq j \neq k,$$

By manipulation we can get:

$$p_{ij} - p_i p_j = \alpha (1 - \alpha) (f_i - g_i) (f_j - g_j)$$
$$p_i p_{ijk} - p_{ij} p_{ik} = \alpha (1 - \alpha) f_i g_i (f_j - g_j) (f_k - g_k).$$

Solve $f_i - g_i$, $i = 1, 2, 3$, out of $p_{ij} - p_i p_j$ to get:

$$f_i - g_i = S_i [\alpha (1 - \alpha)]^{-1/2}$$

where

$$S_i = \pm [(p_{ij} - p_i p_j) (p_{ik} - p_i p_k) / (p_{jk} - p_j p_k)]^{1/2}$$

This gives

$$f_i = p_i + S_i [(1 - \alpha) / \alpha]^{1/2}$$

$$g_i = p_i - S_i [(1 - \alpha) / \alpha]^{1/2}$$

from $p_i p_{ijk} - p_{ij} p_{ik}$ we can determine $\alpha = t^2 / (1 + t^2)$

where t is the solution to $t^2 + Kt - 1 = 0$

and K is

$$K = S_i / p_i - p_i / S_i + \mu_i / (S_i p_i),$$

$$\mu_i = (p_i p_{ijk} - p_{ij} p_{ik}) / (p_{jk} - p_j p_k).$$

It can be verified that K obtains the same value regardless of which index i provices the parameters.

The resulting values of α, f_i and g_i determine whether P is star decomposable. We require that: S_i be real,

$$0 \leq f_i \leq 1 \text{ and } 0 \leq g_i \leq 1.$$

These terms are related to the correlation coefficient via

$$\rho_{ij} = [p_{ij} - p_i p_j] [p_i (1 - p_i)]^{-1/2} [p_j (1 - p_j)]^{-1/2} = [p_{ij} - p_i p_j]/\sigma_i \sigma_j,$$

so the condition that S_i is real is equivalent to requiring that all three of the correlation coefficients are non-negative. If two of them are non-negative we can rename two variables by their complements; the newly defined triplet will have all its pairs positively correlated.

f_i can only be negative if S_i is negative. Choosing S_i as the negative square root, will give a symmetrical solution, f_i', g_i' and α', of f_i, g_i and α, with the $g_i = f_i'$, $f_i = g_i'$ and $\alpha = 1 - \alpha'$. Assume S_i non-negative.

The condition $f_i \leq 1$, or equivalently $t \geq S_i / (1 - p_i)$ gives:

$$p_{ijk} \leq p_{ij}\, p_{ik} / p_i + \sigma_k \sigma_j [\rho_{jk} - \rho_{ij} \rho_{ik}]$$

Similarly with $g_i \geq 0$: $p_{ik}\, p_{ij} / p_i \leq p_{ijk}$. This proves the theorem.

In this research report, I attempt to generalize to more than two states for each variable.

2.2.3 The Cooper and Herskovits algorithm

This is a Bayesian method for doing a greedy search among possible network structures in an attempt to get the most likely structure. Musick [(4)] claims this method works very well.

To find the most probable belief structure B_s from a Database D of instances; This means maximize $p(B_s \mid D)$. The number of possible structures is super-exponential in the number of variables, and there is no clear way of effectively pruning the search space as synergistic effects might occur between any set of variables.

Assumptions made for the Cooper Herskovits algorithm:

1. The process that generated the database is representable by a belief network containing only the discrete variables B_s.
2. Under the belief network model, cases occur independently.
3. Cases are complete, that is, there are no cases that have variables with missing values.
4. The conditional probability is independent of a separate conditional distribution. $p(X_3 = 0 \mid X_2 = 0)$ is independent of $p(X_2 = 0 \mid X_1 = 0)$
5. The density function $f(B_p \mid B_s)$ is uniform, where B_p is the quantitative structure. Therefore there is no initial preference as to what probabilities to place on the structure B_s.

The probability of a structure given a database, D is $p(B_s \mid D) = p(B_s, D) / p(D)$.

If we have two structures B_a and B_b, then the ratio will be:

$p(B_a \mid D) / p(B_b \mid D) = p(B_a, D) / p(B_b, D)$.

The **Cooper and Herskovits algorithm** starts with a single node network, and repeats the following steps:

1. Add a node to the set of parents of the current node.
2. Calculate the probability of the new structure and compare it to the old.
3. If the new structure is noticeably better, then keep the newly added node and try another.

Musick [(6)] claims that this algorithm can reproduce models nearly equivalent to those selected by humans on small to medium sized problems. The probabilities $p(B_s \mid D)$ can be estimated by relative frequencies from a database.

The Cooper and Herskovits algorithm can even handle cyclical networks. In this research report we will restrict ourselves to the Pearl Algorithm (as discussed in Sections 2.1.4 and 2.2.1 for acyclical graphs).

2.3 Modern Portfolio Theory:

Any new technique of portfolio management, necessitates a brief discussion of present methods within the portfolio management field. This will also help in comparing this technique to existing techniques.

Investors want maximum Return, for minimum risk (standard deviation of Return). The Pearl [Pearl(3)] algorithm can be used to estimate the distribution of the Return.

For the Capital Asset Pricing Model (CAPM), analysts have to estimate expected Returns, risk and correlations with the market index of the stocks that they follow. [Elton and Gruber(5)]

The regression coefficient of each stock with the market portfolio, is called Beta. Blume's study showed that the historical Beta and the Beta for the next period had a correlation of 0.6, for individual stocks. This gives an indication that the correlation structure is reasonably stable over time (according to Elton and Gruber (5)), if we take into consideration the high volatility of this kind of data. We therefore expect that the correlation estimates from the data and the subsequent Pearl Algorithm [Pearl (3)] to build the belief net to be reasonably stable over time.

The multi-index model:

Given a set of n economic indices, I_j, like Inflation Rate, the All Share index and some other indices the Return on share i, is:

$$R_i = a_i + \sum_{j=1}^{n} b_{ij} I_j + e_i,$$

which is a regression model, regressing the Return on the different indices I_j. The random error term, e_i, is then assumed to be normally distributed with mean zero and standard deviation σ_{ei}. Assume that the indices are orthogonal. Reduce non-orthogonal indices to orthogonal indices, with some extra calculations.

This method, used by Portfolio Managers, can be a standard to evaluate the new method developed in this research report.

The multi-Index model [Elton and Gruber(5)] can be used for passive or active management. With passive management we structure a portfolio for a client to hedge against certain market occurrences. With active management we take a gamble on the market, based on better forecasts of changes in the indices.

The Sharpe and Cooper research [Elton and Gruber(5)] found that the Return, with dividends ignored, had a coefficient of determination of 0.996 to the Returns with dividends taken into account. For the purposes of this research report, Returns, without dividends, will be used.

Notes:

The Multi-Index model assumes a linear model between the independent variables and Return. The Pearl Algorithm does not make the linear assumption, but it does require discretizing the variables (with a corresponding loss of information).

The Multi-Index model gives us a point estimate of the expected Return and, under certain assumptions, the standard error(risk) of this Return. The Pearl Algorithm gives the belief of the different states of Return. We can then determine a point estimate.

For the purposes of analysis the multi-index model usually assumes that the error term has a normal distribution.

Should the Pearl algorithm perform comparably to the Multi-Index model, it would warrant further investigation to improve this method. Multi-Index models have been used for a very long time, while the Pearl algorithm is a relative new concept. Econometrics uses the multi-regression methods in many different ways. Macro-Economic as well as Micro-Economic fields can benefit from any improvement on the Multi-Index models.

Chapter 3

Finding the link matrices.

3.1 Introduction

In the previous chapter, Section 2.2.2, I discussed the method by Lazarfeld to solve the parameters of the link matrices for binary variables. As a portfolio management technique we need more than two states. Most of the independent variables are rational numbers and we lose too much information by limiting ourselves to two states. In this chapter, I develop a heuristic to increase the number of states from two to four. The success of this heuristic leads to the possibility to increase to even more states.

Section 3.2 gives Lazarfeld's algorithm. This works well for binary variables, but the star decomposition criteria, mentioned in Section 2.2.2, makes this infeasible for more states. We would need seven leaf nodes, connected to the same hidden node, to estimate the parameters for a four state model (Section 3.2.2). The Pearl algorithm to find the structure (Section 2.2.1) only allows for more than three nodes to be linked to a hidden node in exceptional cases.

3.2 The binary variable scenario

3.2.1 Latent Structure

A variable θ, that influences a number of measurable variables, ψ_i, is called a latent variable. Therefore $\psi_i = \phi_i(\theta)$. θ could be a vector of variables. ϕ_i is some function that maps θ onto the observable variable ψ_i. The set of latent variables and the relationships to the other variables is called the latent structure.

Lazarfeld [Lazarfeld(1)] defines latent structure as:

"As the first step in the definition of the latent dichotomy, we postulate the existence of two latent classes. While there will be many situations in which the objects of our

analysis are not individual people, we shall consider now, for the sake of clarity, that we study a population of individuals. The division into two classes may be thought of as corresponding to the presence or the absence of a latent attribute or quality in these individuals.

The relation between the latent classes and the observable items in defined by the axiom of local independence, which states that within a class the items are all independent of one another. This definition states mathematically why the observed items are related to one another."

3.2.2 Number of variables needed to solve the parameters

To find the latent structure (hidden variables) with two states, we need three leaf nodes. To find the latent structure with four states, we need seven leaf nodes to solve for the parameters [Lazarfeld (1)].

By making full use of the implied structure [Lazarfeld (1)], using the ascending matrix method we still need five nodes to solve the parameters of the latent structure.

The Pearl algorithm to create the structure, uses star decomposability of three variables. To generalize the method, to use more states, we need to find a heuristic to calculate the parameters of the latent structure, using only three variables.

We could also generalize the Pearl algorithm to create the structure to incorporate more variables at the same node, but the restrictions for star decomposability of four, or more variables are so stringent that they are unlikely to occur [Pearl(3)].

3.2.3 The accounting Equations

In order to get the parameters of the hidden structure, we must solve the accounting equations [Lazarfeld (1)]:

Let w be the hidden variable, and the leaves be i, j and k, then the accounting equations are:

$$1 = \sum_{w} p(w)$$

$$p(i) = \sum_{w} p(w)\ p(i \mid w)$$

$$p(i, j) = \sum_{w} p(w)\ p(i \mid w)\, p(j \mid w) \text{ and}$$

$$p(i, j, k) = \sum_{w} p(w)\ p(i \mid w)\, p(j \mid w)\, p(k \mid w)$$

3.2.4 To solve the parameters for binary variables

We will start with a known model, expand the model to the frequency matrix (combined frequency of the three leaf nodes), and show how Lazarfeld's approach re-creates the model from the frequency matrix.

Take three leaf nodes i, j and k, and a hidden variable, w, in a star formation between the three variables (Figure 3.1).

We have a method to solve the parameters, with binary variables [Lazarfeld(1)] . The parameters are all the entries in the three link matrices and the prior probability of the hidden variable, w.

To give an example, we start with a known model,

1. fix the prior probability for each state of w and
2. initialize all the relationships, link matrices, with w,

In this section, we expand this model to get the frequencies in the different states of the variables, F(i, j, k).

From the frequencies, we then use Lazarfeld's method to get back to the model.

Example for binary variables

Assume the following model:

$$P(w) = [0.2 \;\; 0.8]$$

and $P(i \mid w)$ =

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.3 & 0.7 \end{bmatrix}$$

$P(j \mid w)$ =

$$\begin{bmatrix} 0.1 & 0.9 \\ 0.2 & 0.8 \end{bmatrix}$$

and $P(k \mid w)$ =

$$\begin{bmatrix} 0.1 & 0.9 \\ 0.3 & 0.7 \end{bmatrix}$$

The frequencies, if we assume 10 000 observations, are then calculated from the accounting equations given at the beginning of Section 3.2, giving

$F(i, j, k)$ =

$$\begin{array}{cc} i = 1 & i = 2 \\ \begin{bmatrix} 148 & 372 \\ 612 & 1668 \end{bmatrix} & \begin{bmatrix} 352 & 928 \\ 1488 & 4432 \end{bmatrix} \end{array}$$

Use these frequencies to get back to the model.

Lazarfeld then derives the following theorem:

The roots of the equation

t^2 - $\{1 + ([jk;i] - [jk;\sim i]) / [jk]\}$ t + $[jk;i] / [jk] = 0$, where $[jk]$ is the determinant of $P(j, k)$ and $[jk;i]$ is the determinant of $P(j, k \mid i)$. $[jk;\sim i]$ is the determinant of $P(j, k \mid \text{not } i)$,

This gives the components of the link matrices.

$[jk;i] = 2\,800 \times 148 - 520 \times 760 = 19\,200$

$[jk;\sim i] = 7\,200 \times 352 - 1\,280 \times 1\,840 = 179\,200$

$[jk] = 500 \times 10\,000 - 1800 \times 9600 = 320\,000$

therefore the root is

$t = (0.5 \pm 0.1)/2$

These are the values for $p(i = 1 \mid w = 1)$ and $p(i = 1 \mid w = 2)$, because the correlation between w and i is negativ (because the determinant of $P(i \mid w)$ is negative), the smaller solution will be the value o. $p(i = 1 \mid w = 1)$.

This Returns the model $P(i \mid w)$ =

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.3 & 0.7 \end{bmatrix}$$

which is the link matrix of the model. Find the other link matrices the same method.

3.3 Finding the link matrices

In Section 3.2.2, I investigate a binary-variable example, following the method of Lazarfeld. The problem is now to find a heuristic that will give acceptable results, when using a model with four states instead of two states.

3.3.1 The four state model

In this section, we start with a known model and expand this model to the frequency matrix (Figure 3.1). Sections 3.3.2, 3.3.3 and 3.3.4 create models by simulation, and evaluate the performance of different heuristics to get back to the known model.

... Section 3.3.1, I give an example of a four state model. From this model, I calculate the frequency matrix and the correlation coefficients. The heuristic works, if it takes us from the frequency matrix back to the parameters of the model.

In Section 3.3.2, I test the method suggested, but not implemented, by Pearl [(3)]. The method combines states to use Lazarfeld's algorithm. The tests do not give good results, and we discus why this approach was not persued further.

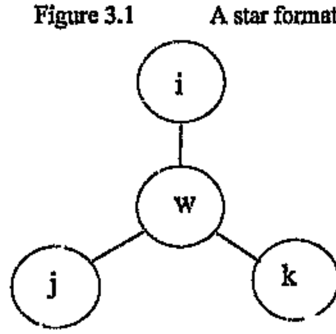
Section 3.3.3 gives the heuristic, that I developed, after realizing that S_i determines the choice of the roots. For the simulation, in this section, I assume that all the correlations are positive.

In Section 3.3.4, I address the problem of which root to choose, depending on the calculated correlation.

Example of a four state model

Take a structure with three variables, connected by a hidden variable, see Figure 3.1.

Figure 3.1 A star formation:



Let w be the hidden node and i, j and k be the leaf nodes:

Take, P(w) as [0.2 0.3 0.1 0.4]

$P(i \mid w) =$

$$\begin{bmatrix} 0.2 & 0.3 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.2 & 0.2 & 0.3 & 0.3 \\ 0.5 & 0.2 & 0.2 & 0.1 \end{bmatrix}$$

$$
P(j \mid w) =
\begin{bmatrix}
0.1 & 0.2 & 0.3 & 0.4 \\
0.2 & 0.1 & 0.3 & 0.4 \\
0.4 & 0.3 & 0.1 & 0.2 \\
0.3 & 0.4 & 0.2 & 0.1
\end{bmatrix}
$$

$$
P(k \mid w) =
\begin{bmatrix}
0.1 & 0.4 & 0.4 & 0.1 \\
0 & 0.2 & 0.4 & 0.4 \\
0 & 0.5 & 0.5 & 0 \\
 & 0.3 & 0.3 & 0.2
\end{bmatrix}
$$

Using the accounting equation, $p(i, j, k) = \sum_{w} p(w)\ p(i \mid w)\, p(j \mid w)\, p(k \mid w)$, where the summation is over all the states of w. Multiply by 10 000 to get frequencies.

$F(i, j, k) =$

$i = 1$

$$
\begin{bmatrix}
124 & 248 & 260 & 148 \\
168 & 308 & 314 & 180 \\
92 & 196 & 214 & 128 \\
56 & 168 & 192 & 104
\end{bmatrix}
$$

$i = 2$

$$
\begin{bmatrix}
54 & 160 & 184 & 102 \\
76 & 186 & 198 & 100 \\
50 & 166 & 202 & 122 \\
'0 & 188 & 236 & 136
\end{bmatrix}
$$

$i = 3$

$$
\begin{bmatrix}
54 & 192 & 228 & 126 \\
76 & 207 & 225 & 112 \\
50 & 189 & 243 & 158 \\
40 & 222 & 294 & 184
\end{bmatrix}
$$

$i = 4$

$$
\begin{bmatrix}
28 & 160 & 208 & 124 \\
40 & 149 & 173 & 88 \\
28 & 159 & 231 & 172 \\
24 & 202 & 298 & 216
\end{bmatrix}
$$

We can calculate the Pearson product moment correlation. For the simulation, I calculated this using the cells in the frequency matrix as grouped data.

Let $\quad G(a,b) = \sum_{k=0}^{3} F(a,b,k)$

then $\quad \bar{a} = \sum_{a=0}^{3} a \sum_{b=0}^{3} G(a,b)$ and $\quad \bar{b} = \sum_{b=0}^{3} b \sum_{a=0}^{3} G(a,b)$

$$r_{ab} = \frac{\sum_{a=0}^{3}\sum_{b=0}^{3} G(a,b)(a-\bar{a})(b-\bar{b})}{\sqrt{\sum_{a=0}^{3}\sum_{b=0}^{3} G(a,b)(a-\bar{a})^2 \cdot \sum_{b=0}^{3}\sum_{a=0}^{3} G(a,b)(b-\bar{b})^2}},$$

where a and b are the numerical numbers of the states {0, 1, 2, 3}.
For the actual application in Chapter 5, I used the pre-discretized data from the companies to calculate the correlation.

The correlation coefficients are:

$r_{ij} = 0.1179$, $r_{ik} = 0.1148$ and $r_{jk} = 0.0776$.

From Pearl's algorithm to find the structure (Section 2.2.1):

$r_{wi} = 0.4176$, $r_{wj} = 0.2823$ and $r_{wk} = 0.2749$.

The rest of this chapter discusses a series of heuristics to find the model, P(w), P(i | w), P(j | w) and P(k | w), from the frequency matrix F(i, j, k).

3.3.2 Combine three states to get a binary situation

In Section 3.2.3, I described an algorithm that works for binary states. In this section I attempt to re-use the same approach by combining three states, to calculate the relavant parameter for the fourth state. In order to calculate parameters in the first row, I combine states two, three and four by adding the frequencies in the frequency matrix. To calculate the parameters in the second row, I add the frequencies of rows one, three and four in the frequency matrix, then in this binary situation, calculate the parameters as in the previous section.

Pearl suggested this method to handle more than two states.

A summary of this method:

1. Fix one state of i, one state of j and one state of k.
2. Add the other three states for each variable together, and solve for the chosen state.

3. Do this for every combination of states of the variables i, j and k.

This gives a binary situation as the one described in Section 3.2.3 for each combination of the states of i, j and k. An example follows.

The combined frequency matrix for the four state example is:

$F(i, j, k) =$

$i = 1$

$$\begin{bmatrix} 124 & 248 & 260 & 148 \\ 168 & 308 & 314 & 180 \\ 92 & 196 & 214 & 128 \\ 56 & 166 & 192 & 104 \end{bmatrix}$$

$i = 2$

$$\begin{bmatrix} 54 & 160 & 184 & 102 \\ 76 & 186 & 198 & 100 \\ 50 & 166 & 202 & 122 \\ 40 & 188 & 236 & 136 \end{bmatrix}$$

$i = 3$

$$\begin{bmatrix} 54 & 192 & 228 & 126 \\ 76 & 207 & 225 & 112 \\ 50 & 189 & 243 & 158 \\ 40 & 222 & 294 & 184 \end{bmatrix}$$

$i = 4$

$$\begin{bmatrix} 28 & 160 & 208 & 124 \\ 40 & 149 & 173 & 88 \\ 28 & 159 & 231 & 172 \\ 24 & 202 & 296 & 216 \end{bmatrix}$$

Take $i = 2$, $j = 3$ and $k = 1$, the combined frequency matrix becomes:

$i = 2$

$$\begin{bmatrix} 50 & 490 \\ 170 & 149 \\ & 0 \end{bmatrix}$$

$i \neq 2$

$$\begin{bmatrix} 170 & 1690 \\ 610 & 5330 \end{bmatrix}$$

Do this for each state of i, j and k. For each of these combinations of states, i, j and k, obtain a different combined frequency matrix. The Lazarfeld method gives two roots, call them t_1 and t_2. Compare these roots to the correct values of the elements of the link matrix:

$P(i \mid w) =$

$$\begin{bmatrix} 0.2 & 0.3 & 0.3 & 0.2 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.2 & 0.2 & 0.3 & 0.3 \\ 0.5 & 0.2 & 0.2 & 0.1 \end{bmatrix}$$

To give a single numerical value as an indication of how well this heuristic could work take the average absolute deviation between the best root and the correct value. Let t_{rc} be the root selected. In Table 3.1 t_{rc} is chosen as the the root that is closest to P(i | w).

$$\text{AAD} = \text{Average absolute deviation} = \frac{\sum \left| P(i \mid w) - t_{rc} \right|}{n},$$ where the

summation goes over all the rows and columns of P(i | w). If the heuristic works perfectly, the average absolute deviation will be 0.0. The two roots t_1 and t_2 of the quadratic equation above from the binary matrices, as in Section 3.2.3, are in Table 3.1.

Table 3.1 The two roots and the correct value for each parameter, with the best choice the AAD = 0.0524.

State of w	State of i	t_1	t_2	correct value
1	1	0.1229	0.6671	0.2
1	2	0.2000	0.2200	0.3
1	3	0.1400	0.3000	0.3
1	4	0.0314	0.3186	0.2
2	1	0.0056	0.3262	0.1
2	2	0.1954	0.2001	0.2
2	3	0.2727	0.3000	0.3
2	4	0.1801	0.5200	0.4
3	1	0.0883	1.8117	0.2
3	2	0.2000	0.4000	0.2
3	3	-0.3000	0.3000	0.3
3	4	-0.8352	0.3352	0.3
4	1	0.0385	0.3967	0.5
4	2	0.1706	0.2000	0.2
4	3	0.2412	0.3000	0.2
4	4	0.1564	0.4966	0.1

The heuristically determined values of the elements in the link matrix give an indication of how well this heuristic works. The estimates are not good, but they are not random either.

In Section 3.2.3 Lazarfeld chose the parameter, by selecting the largest of the two roots for the main diagonal of the link matrix, if the correlation is positive. If the correlation is negative, the smallest of the two roots is the parameter on the main diagonal.

The roots as given in Table 3.1 still deviate too much from the real values. We must improve the estimation of roots, before we need to resolve the problem of choosing which root is relevant. Pearl does not give an indication on how to choose between the roots. Because the roots do not give the correct values for the parameters, I did not persue this matter further.

3.3.3 Heuristic with pair-wise grouping

Reasoning behind this heuristic.

Three variables i, j and k can only be star decomposable, if the correlations r_{ij}, r_{ik} and r_{jk} are all positive, or two of them are negative [Pearl(3)]. If two of the correlations are negative, change them to all positive correlations by reversing the order of the states of one of the variables.

When we calculate r_{iw}, r_{wj} and r_{wk} by the equations:

$$r_{iw}^2 = r_{ij}\, r_{ik} / r_{jk}$$
$$r_{jw}^2 = r_{ij}\, r_{jk} / r_{ik}$$
$$r_{kw}^2 = r_{ik}\, r_{jk} / r_{ij},$$

We get two sets of solutions, one all positive and the other one all negative. To show this is true:

$r_{iw}\ r_{jw} = r_{ij}$, because r_{ij} is positive, both r_{iw} and r_{jw} must be positive, or both negative. The result follows by looking at variable k and variable j. So, r_{iw}, r_{wj} and r_{wk} are all positive or all negative.

This is similar to the choice of S_i in Section 2.2.2, that gives us two symmetrical solutions.

By inter-changing states 1 and 4 and states 2 and 3 of the hidden variable, we get from the positive to the negative correlations. This explains why the quadratic equation links the two states in the binary state case. This shows that we must not combine the states of w, but look at the states of w in pairs, namely states 1 and 4 and states 2 and 3.

Manual calculation, using this heuristic.

To calculate the parameters P(i | w), we use Lazarfeld's algorithm from Section 3.2.3. To calculate row one, column one and row four column one in P(i | w), we look only at the frequencies where j and k are in states one and four. The Frequency matrix from Section 3.3.1 becomes:

F(i, j, k) =

$$
\begin{array}{ll}
i = 1 & \begin{bmatrix} 124 & 148 \\ 56 & 104 \end{bmatrix} \\
i = 2,\ 3 \text{ or } 4 & \begin{bmatrix} 136 & 352 \\ 104 & 536 \end{bmatrix}
\end{array}
$$

The rows for i = 1 are rows (states one and four of j) one and four of F(1, j, k) and the rows for i = 2, 3 and 4 are the total for F(2, j, k), F(3, j, k) and F(4, j, k). The columns are calculated in the same way (states one and four of k).

From the Lazarfeld algorithm:

[jk;i=1] =	124x104 - 56x148	=	4608
[jk;i not =1] =	136x536 - 104x352	=	36288
[jk] =	260x640 - 160x500	=	86400

The roots of the equation t^2 - {1 + ([jk;i] - [jk;~i]) / [jk]} t + [jk;i] /[jk] = 0, or $t^2 - 0.63333t + 0.05333 = 0$ are 0.5333 and 0.1000. From the model, Section 3.3.1 the correct values should be 0.2 and 0.5.

To get the values for rows two and three, of column one of P(i | w), we take F(1, j, k) and the total of F(2, j, k), F(3, j, k) and F(4, j, k), looking only at states two and three of the variables j and k.

To get the values for rows one and four, column two of P(i | w), we take F(2, j, k) and the total of F(1, j, k), F(3, j, k) and F(4, j, k), looking at states one and four of variables j and k. The rest of the elements follow in the same way.

Repeating the calculation of the roots for each parameter of the model, for the time being, take all correlations (Pearson product moment correlations) as positive, we get:

P(i | w) =

$$\begin{bmatrix} 0.53 & 0.20 & 0.19 & 0.09 \\ 0.40 & 0.21 & 0.23 & 0.16 \\ 0.10 & 0.20 & 0.30 & 0.40 \\ 0.10 & 0.19 & 0.30 & 0.40 \end{bmatrix}$$

P(j | w) =

$$\begin{bmatrix} 0.29 & 0.39 & 0.21 & 0.12 \\ 0.20 & 0.30 & 0.25 & 0.25 \\ 0.20 & 0.10 & 0.30 & 0.40 \\ 0.20 & 0.10 & 0.30 & 0.40 \end{bmatrix}$$

P(k | w) =

$$\begin{bmatrix} 0.19 & 0.32 & 0.30 & 0.19 \\ 0.20 & 0.33 & 0.29 & 0.18 \\ 0.00 & 0.10 & 0.39 & 0.52 \\ 0.02 & 0.24 & 0.40 & 0.33 \end{bmatrix}$$

Compare these matrices to the correct matrices as in Section 3.3.1. The maximum deviation from the correct values is 0.21.

(Note the model that we started with has negative correlations with the hidden variable w. Here we chose the positive correlation for each pair, according to the Lazarfeld method, Section 3.2.3. To compare we have to compare row one of the model, with row four of the estimation of the model, etc.)

Simulations using this heuristic.

Each iteration consists of all the steps taken in the manual calculation.

The steps, for each iteration, are:

1. Create a model, with random numbers as prior probabilities and random numbers for the elements of the link matrices (normalized to make them probabilities).
2. From the model, calculate the (4 x 4 x 4) frequency matrix.
3. From the frequency matrix calculate the Pearson correlation coefficients.
4. Apply the heuristic described above to calculate the parameters.
5. Compare the obtained values with the model of the specific iteration.

I ran a simulation, with 34 420 iterations and assumed that the correlations to the hidden variables were positive, irrespective of whether the correlations of the hidden variables in the models were positive:

Table 3.2 Results of the first simulation, with an AAD = 0.082

Row	Column	Mean error	Standard deviation of error
1	1	0.0810248	0.1627995
1	2	0.0827834	0.1633740
1	3	-0.0820516	0.1496301
1	4	-0.0817563	0.1499833
2	1	0.0814020	0.1637623
2	2	0.0821751	0.1630656
2	3	-0.0815376	0.1498768
2	4	-0.0820405	0.1499095
3	1	-0.0828858	0.1448113
3	2	-0.0832196	0.1496353
3	3	0.0834214	0.1605161
3	4	0.0809790	0.1603846
4	1	-0.0813898	0.1502567
4	2	-0.0824145	0.1498112
4	3	0.0815157	0.1631894
4	4	0.0822888	0.1634199

This tells us that by selecting positive or negative correlations to the hidden variable, at random, the average absolute error is about 0.082, with a standard deviation of 0.15. The average absolute error can be reduced by an educated choice between the roots.

The pattern of negative and positive errors, is an indication of an incorrect choice of the sign of the correlation.

3.3.4 Selecting the root

To allocate the roots of, t^2 - {1 + ([jk;i] - [jk;~i]) / [jk]} t + [jk;i] /[jk] = 0, to the states in their pairs, take the elements on the main diagonal of the pair in the link matrices as the largest of the two roots and the smaller root for the other elements.

For the simulation, I made the discriminants and probabilities that were less than zero, are set to zero, and probabilities of more than one, are set to one. This happened in about 5% of the parameters calculated.

For the simulation, each iteration consists of randomly creating a model, and calculating the frequency matrix. Then apply the heuristic described above to calculate the parameters and compare the obtained values with the relevant model.

The results of a simulation with 1185 iterations, using the heuristic are in Table 3.3.

Table 3.3 Results of a simulation with the improved heuristic, with an AAD = 0.0633

Row	Column	Mean error	Standard deviation of error
1	1	0.0563405	0.1502446
1	2	0.0678629	0.1550100
1	3	-0.0666467	0.1428206
1	4	-0.0575959	0.1372334
2	1	0.0622232	0.1580924
2	2	0.0624152	0.1571103
2	3	-0.0633632	0.1422067
2	4	-0.0612752	0.1462479
3	1	-0.0599822	0.1422875
3	2	-0.0690256	0.1412494
3	3	-0.0668639	0.1570861
3	4	-0.0534771	0.1568450
4	1	-0.0634986	0.1388671
4	2	-0.0687006	0.1453125
4	3	0.0632440	0.1581354
4	4	0.0689552	0.1607154

This reduces the average absolute error to 0.0633 and the standard deviation to about 0.145. This can nearly compete with the "best" choice in Table 3.1. The absolute error improved, but the choice of the root is still not perfect. We see this from the pattern in the sign of the errors. The error comes in due the binary choice of the r It will not make sense to adjust the values for the calculated error, because this will introduce errors in the correctly calculated parameters.

With randomly created models, the correlations tended to be very close to zero. I chose 1000 cases from a simulation where the correlations between all the leaf variables were more than 0.20 and found that the absolute mean error and the standard deviation of this error were about the same.

The heuristic consists of two steps. We already know that the choice of root is not perfect. To determine the accuracy of the other step, namely the calculation of the roots, we can assume the correct value known.

The errors, between the best root and the correct value, are either very close to correct, or in about 10% of cases the heuristic failed and produced incorrect roots. There seems to be a tendency that the heuristic fails when the [jk] determinant is close to zero, but often the value is correct even when the [jk] determinant is nearly zero. I could find no clear indication that would explain why some elements are accurate, while others are completely incorrect.

3.3.5 Improving the heuristic

One improvement would be to identify the entries that are obviously incorrect, when the probability is less than zero, or more than one. These are the calculations where we know that the heuristic failed. We can recalculate these obviously incorrect values (both values of the incorrect pair) using the pairs states 1 and 3 and states 2 and 4.

Table 3.4 Simulation results, when obviously incorrect results are recalculated, using 3315 iterations, with an AAD = 0.0473

Row	Column	Mean error	Standard deviation of error
1	1	0.0462413	0.1429633
1	2	0.0516833	0.1425864
1	3	-0.0433038	0.1384373
1	4	-0.0546206	0.1310220
2	1	0.0452210	0.1434841
2	2	0.0454919	0.1418247
2	3	-0.0471996	0.1386019
2	4	-0.0435133	0.1344084
3	1	-0.0438793	0.1352575
3	2	-0.0473363	0.1390441
3	3	-0.0476626	0.1440521
3	4	-0.0435531	0.1424507
4	1	-0.0543001	0.1326349
4	2	-0.0447070	0 1386045
4	3	0.0552383	0.1419775
4	4	0.0437687	0.1421711

This gives a average absolute error of 0.0473, with a standard deviation of about 0.14. The pattern of incorrect choices of the root, is still evident. This is better than the best that could be achieved by the heuristic of Table 3.1.

Recalculating when the discriminant of the quadratic equation is less than zero, does not make any noticeable improvement.

The values heuristically calculated for the parameters tended to be very close to correct, within 0.01 of the correct value, which indicated that the model worked, or else they were not even close. The numbers of errors with completely incorrect values were about 5% of the total number of elements. Incorrect values due to the wrong choice between the roots amounted to about 20% of the elements.

3.3.6 Identify the cause of the errors.

The heuristic contains two components: Firstly, the calculation of the values of the elements of each pair. Secondly, we allocate these values between the two elements of the pair according to the negative or positive correlation between the hidden node and the leaf nodes. From the previous paragraph, we see that to further improve the heuristic, the second component needs more attention.

If we select the best value, we can see how much we lose by assuming all pairs must have the correct correlation sign. (The global correlation could be positive, while some pairs could have a negative correlation)

From Table 3.5, we see that this gives a mean absolute error of about 0.03 with a standard deviation of about 0.11. There is still a pattern in the errors, therefore the choice of errors does no fully explain the pattern.

The heuristic in Section 3.3.5 compares favorably to the best choice of root in this section. The loss, that we make by assuming all pairs positively correlated, is only 0.02 on average, for each element.

Table 3.5 Selecting the best value of each pair, with 22 216 iterations (AAD=0.03)

Row	Column	Mean error	Standard deviation of error
1	1	0.0360689	0.1083372
1	2	0.0372258	0.1077500
1	3	-0.0260772	0.0895031
1	4	-0.0244026	0.0873792
2	1	0.0322831	0.1322003
2	2	0.0328441	0.1274338
2	3	-0.0233486	0.1085415
2	4	-0.0226206	0.1054089
3	1	-0.0271116	0.1082862
3	2	-0.0231264	0.1055908
3	3	-0.0356498	0.1311579
3	4	-0.0361977	0.1301298
4	1	-0.0218543	0.0877087
4	2	-0.0224973	0.0892521
4	3	0.0328204	0.1050006
4	4	0.0320960	0.1028545

3.4 Conclusion for the chapter

In this chapter, I developed a heuristic to generalize the number of states, from two to four states. The first attempt in Section 3.3.2, of just combining states into a binary scenario, did not give acceptable results. We might use this heuristic to generalize to an uneven number of states, or to recalculate the obviously incorrect values remaining after the adjustments to the heuristic in Section 3.3.5. The heuristic in Section 3.3.3 used the possibility of positive and negative correlation structures to get better results. The problem of which of the roots, for each of the elements of the link matrix to choose still remained unresolved.

Section 3 3.4 addresses the choice of roots, and in Section 3.3.5, I address the problem where certain calculated values for the elements in the link matrices were obviously incorrect.

The heuristic developed in this chapter is still not perfect. As the heuristic links more nodes, the errors propagate further. Many Belief nets will give incorrect results. In Chapter five, I select the Belief nets that give acceptable results and do an empirical study of how well the heuristic works.

Chapter 4

Data cleaning and Software Developed

4.1 Introduction

In Chapter three I developed a heuristic to generalize the Pearl algorithm from two to four states. This chapter gives an overview of the software that I developed.

The McGregor database is a commercial database giving information on different companies. This is a flat file database. The attributes are the financial variables of companies for the past five years. Appendix B gives a list of the fields used in this research report.

The McGregor database [McGregor (7)] does not give the variable Return. Return is the ratio between the prices on the last Friday in May 1995 and the last Friday in November 1995. I obtained both these values from the relevant newspapers. I ignored Dividend income.

In Section 4.2 we look at the measures taken to clean the data and the reasoning why I exclude certain variables.

In Section 4.3, I discuss the software, to create the tree structure, and explain how to implement the theory. The software developed uses the frequencies and correlations between variables to build belief nets.

The method used, compares the incoming node to a selection of three leaf nodes in the belief net.

The star decomposability restrictions allow, or do not allow, the linking on of the new node. A different tree structure is possible, based on the evaluation order of nodes. The choice of initial variables dramatically influences the calculation of the link matrices. The link matrices can be substantially different for the belief nets that had different initial variables. All possible sets of three initial variables were used to create belief nets.

In Section 4.4, I discuss the criteria used to evaluate and decide the effectiveness of different Belief nets.

In Chapter five I give the empirical results, with this heuristic on the Stock Exchange data.

4.2 Data cleaning and reducing the number of variables.

I ignored all shares with incomplete data, irrespective of the reason why they were incomplete. The purpose of this research report is to investigate a technique, and not to make a statement regarding real investment opportunities (or any real-world causalities). There is no claim that the results found in this research are relevant for a more general population. Possible parallels between the Belief nets and other Stock exchanges or over other time periods may exist, but I make no claim to that effect in this research.

I reverse the order of the states of the negatively correlated variables. This makes the correlation between all variables and Return positive.

We know that [Curwin and Slater(13)] for the correlation to be significant, at a 5% level of significance,

$$\frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \geq 1.96$$

Therefore $238\, r^2 > 1.96^2 (1 - r^2)$,

or $r > 0.126$

I ignored variables, with a correlation to Return, of less than 0.05[3]. This left 65 variables.

[3] These variables could be brought back if we suspect that they significantly influence some of the variables that are identified as causal variables for the Return.

Comparing the best performing belief nets from chapter five with Appendix A, we see that some of the variables used only marginally made it past the criterion above. This makes an intelligent search difficult.

4.3 Software developed.

I wrote a number of programs, during this research.

The main programs are:

1. A program to calculate the Pearson correlation coefficient and to convert the data into z-values.
2. A simulation program that creates models, with three leaf nodes and a center node, randomly, and then tests the effectiveness of the heuristic to determine the parameters in the link matrices.
3. A program to read in the z-values and correlations to construct the Belief nets, and save the nets. (This program can run either on the training set, the test set or on all companies) See an outline of the subroutine to create the Belief net structure in Section 4.3.1.
4. A program that reads the belief nets, instantiates them for all the companies (for either the training or the test set of companies) and calculates and stores the accuracy.

4.3.1 To develop the tree:

I use three nodes to get the initial tree. Return has to be one of these nodes to force Return into the structure. The other two nodes can be any other two variables. If the initial three nodes are not star decomposable [Pearl (3)] we have to start with a new selection.

For a new node to be added one of the star decomposition conditions $\rho_{ac} \cdot \rho_{de} = \rho_{ad} \cdot \rho_{ce}$, $\rho_{ac} \cdot \rho_{de} = \rho_{cd} \cdot \rho_{ae}$ or $\rho_{ad} \cdot \rho_{ce} = \rho_{cd} \cdot \rho_{ae}$ must be true. Because we are working with data, and possible sampling variability, we must allow a certain variation, ε.

If we take $|r_{ij} r_{km} - r_{ik} r_{jm}| < \varepsilon$, we give variables with a small correlation an advantage. By taking $r_{ij} r_{km} / r_{ik} r_{jm}$ between $1 \pm \varepsilon$, we avoid this problem. In this research I used $\varepsilon = 0.04$. If we take a very small ε, we will get very small Belief nets. A large ε will allow for many-node Belief nets.

Algorithm to link on a new node, k:

Using Section 2.2.1 to design an Algorithm to link on a new node, k, we get the following rules to decide where the node must link on.

Given a star formation as in Figure 2.7, where we have a central node b and leaf nodes a, c and d. To link up a fourth node, e, and if the node links on to the,

if $\left|\dfrac{r_{ac} r_{de}}{r_{ad} r_{ce}} - 1\right| < \varepsilon$ then link onto arc ba.

if $\left|\dfrac{r_{ae} r_{de}}{r_{cd} r_{ae}} - 1\right| < \varepsilon$ then link onto arc bc.

if $\left|\dfrac{r_{ad} r_{ce}}{r_{cd} r_{ae}} - 1\right| < \varepsilon$ then link onto arc bd.

If more than one of these rules is true, it means that the new node must link on at the central node, b. In this research I used the first true rule encountered. The correlation between the central node b and the new node (parent of e) will be one if it links on at node, b. In the research I found that the correlation could be one, but that the link matrix, from the heuristic is not a unit matrix. This confirms that the heuristic is not infallible. In these cases I decided not to remove the arc between the nodes.

Use the variable L to indicate the active node. Link the variable k onto an arc, at a new hidden node, called "Parent of k" in Figures 4.1 and 4.2.

1. Starting at the root we decide which of the three branches is applicable, using the rules above.

   Set L as the root.

2. Move down to the next node on the structure, set L equal to this node.

3. Case 1: Decide whether the incoming node, k, must link onto an arc to a leaf node, refer to Figure 4.1.

   Case 2: Decide whether the incoming node, k, must link onto the arc back towards the root, Figure 4.2.

   Case 3: Decide that the incoming node, k, must link onto a node towards a leaf node, but the next node lower down in the tree, is not a leaf node.

   Go to step 2.

Use the variables i, j and m to indicate existing nodes in the belief net. Add the incoming node (variable k) as well as its parent (variable "parent of k"), to the correct position as determined by the algorithm above.

Figure 4.1: Case 1, Adding k to an arc to a leaf node.

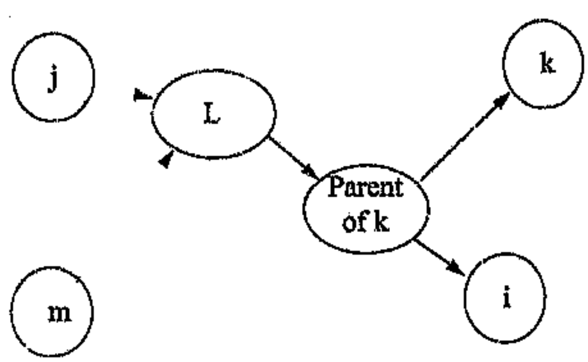
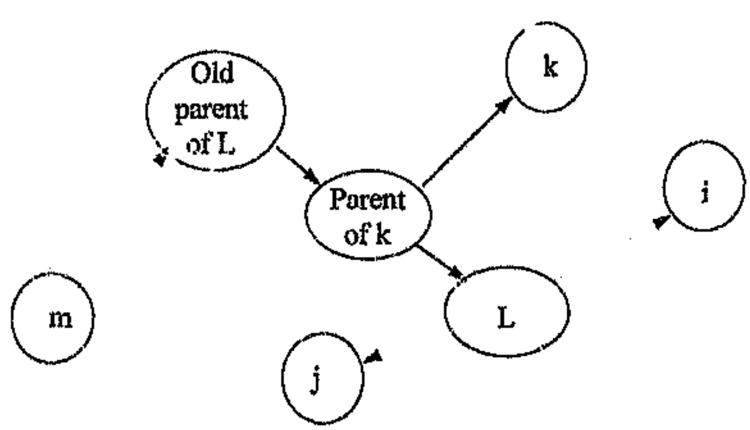


Figure 4.2: Case 2: Adding k to an internal arc.



Use the heuristic developed in Chapter three to calculate the link matrices between variables m, i and k. The new hidden variable is "Parent of k". In the next section, I use these matrices to calculate the link matrices of the individual arcs.

4.3.2 Finding the internal link matrices of the structure:

Now we use the heuristics developed in Chapter three to solve for the link matrices between any three variables in a belief net. The heuristics developed in Chapter three can also be used to derive the link matrix to the incoming variable, k.

Case 1 in Figure 4.1:

Find the link between "parent of k" and L by solving:

$$P(i \mid L) = P(i \mid \text{"parent of k"})\, P(\text{"parent of k"} \mid L).$$

where

$P(i \mid L)$ is the link matrix of i to the old parent of the leaf node i, that we know, or can be recalculated by the methods from Chapter three.

$P(i \mid \text{"parent of k"})$ is the link to the new parent of i.

Case 2, Figure 4.2:

The two link matrices,

- the link from L to "parent of k" and
- the link from "parent of k" to the old parent of L,

must be calculated.

Find the link from L to "parent of k" by solving:

$$P(i \mid \text{"parent of k"}) = P(i \mid L)\ P(L \mid \text{"parent of k"})$$

P(i | L) is known, or can be recalculated by multiplying all the link matrices down the tree towards the leaf node i.

Calculate P(i | "parent of k") with the methods developed in Chapter three.

Find P("parent of k" | old parent of L), by solving:

P(L | old parent L) = P(L | "parent of k") P("parent of k" | old parent of L).

P(L | old parent of L) was calculated when L was the incoming hidden node, and P(L | "parent of k") is the matrix calculated earlier in this section.

4.3.3 Internal correlations

Calculate the correlations, with the variables on all three arcs, to the hidden variable, "parent of k", using the equations given in Section 2.2.1. Calculate these correlations as we create the hidden variables. We need these correlations to decide where incoming nodes must link onto the belief net, in Section 4.3.1.

4.4 Evaluating the belief networks.

All the variables were discritized into four states. We can split these variables in any arbitrary manner, without affecting the working of the heuristic. If too many classes were empty we would be back to binary or three states.

To get a uniform way to discritize the variables, I decided to use the z-value, the number of standard deviations from the mean. The z-values would be convenient to transform back to the variables.

(This does not mean that the variables are normally distributed. With hindsight, quartile splitting could be better. Most of the variables are from very skew distributions. See appendix B.)

$z = \frac{x - \bar{x}}{S_x}$, where the variables have their normal meanings.

1. State one is when $z > 0.6$
2. State two when $0 < z < 0.6$
3. State three when $-0.6 < z < 0$ and
4. State four when $z < -0.6$.

With normally distributed variables, the frequency would have been about equal for all classes.

To initialize a belief network we have to specify three initial variables. I forced Return into the network by specifying this as one of the initial nodes. We have the option of ${}_{65}C_2$ different possibilities as initial variables. For each starting triplet, I create a Belief net, as described above.

Some of these belief nets do not exist, because the initial variables are not star decomposable.

Each Belief net is evaluated by instantiating the states of the leaf nodes and find Return by propagating the likelihood through the net. If the true state of Return is the same as the state with the maximum belief of Return, then count it as a success. The accuracy of a belief net is the proportion of successful predictions. I first did this by setting the likelihood, of each state of Return, equal to the relative frequency of that state, I then repeated this with the likelihood of each state of Return set equal to 0.25. Return is assumed to have a normal distribution [Elton and Gruber(5)] so the expected frequencies for states, with the z-values split at zero and ±0.6, should be about equal.

This gives the proportion of companies, for which the network gave the correct prediction. Belief nets that give us an accuracy of more than 0.25, are better than random. By using this accuracy, I identify the best nets for testing on the testing set of companies, T.

In the next chapter I discuss the results of these prog.'ams and compare the Belief nets with linear regression.

Chapter 5

Results from Stock exchange data

5.1 Introduction:

In the previous chapter, I explained how to create and evaluate the Belief nets.

There are 2080 different possible belief nets for the 65 variables, that will include Return.

To test the Belief nets, we must choose between different prior probabilities for Return. The likelihoods of all the independent variables are instantiated to be equal to the occurrence of the company, but set the dependent variable (Return) t

1. all states have equal likelihood or
2. the states have likelihood equal to the observed prior frequencies of Return on the training set, C.

These choices do not differ much, but the difference in the results shows how sensitive the heuristic is to minor changes in the prior probabilities.

None of the variables have a correlation, to Return, significant on a 1% level. There is a high level of co-linearity between the variables. The correlations between some of these variables are as high as 0.97. I suspect that many of the 65 variables are linear combinations of the other variables.

In Section 5.2 I give the results, using 240 companies to create the Belief networks (the Training set, C) and the remaining companies to test(the testing set, T).

Section 5.3 discusses the accuracy of the predictions as calculated by Linear Regression. The hold-out sample shows that the Pearl Algorithm is better at extrapolation to different companies.

Finally, I give the conclusions and recommendations for future research in Chapter 6.

5.2 The Belief nets

I analysed all the belief nets that I created. Section 5.2.1 gives Belief nets, with the best re-substituted accuracy, using equal prior probabilities for the states. Section 5.2.2 gives the Belief nets with the best accuracy, when using the observed frequencies for the 240 companies as the prior probabilities for the states.

I tested every belief net created, both on the Training set and on the testing set of companies. For each of the nets the difference between the accuracy between the sets was not large. The biggest difference, 9%, occurred in the nets given in Figures 5.3 and 5.4.

5.2.1 Prior with equal probabilities for all states.

By taking the prior probability equal, for all the states of Return, seventeen Belief nets have an accuracy of more than 50%, when measured on the 240 companies. Testing the accuracy by looking at the remaining 117 companies, fourteen of the Belief nets still have an accuracy of more than 50%.

Six Belief nets have accuracies of more than 70% on the 240 companies. Six of these stayed above 70% when tested on the remaining 117 companies. One Belief net dropped from 73% to 69.57%.

The two belief nets that measured the best accuracy on the 240 companies are: The Audfees5-Currat3 Belief net and the Curliab2-Debtass2 Belief net. Both have an accuracy of 79% when measured with on the 240 companies and both drop to 75% when tested on the remaining 117 companies. The Belief nets are shown in Figure 5.1 and in Figure 5.2. The correlation coefficients are indicated on the arcs.

Figure 5.1 The Audfees5-Currat3 Belief net

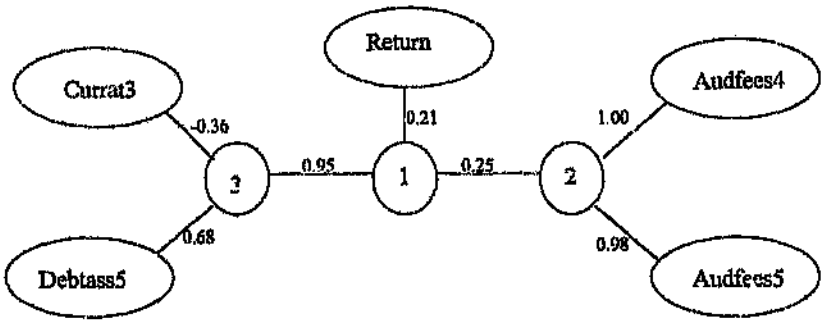


Table 5.1 The variables in Figure 5.1

Currat	Current Ratio
Debtass	Debt asset ratio
Audfees	Audit Fees

Figure 5.2 The Curliab2-Debtass2 Belief net

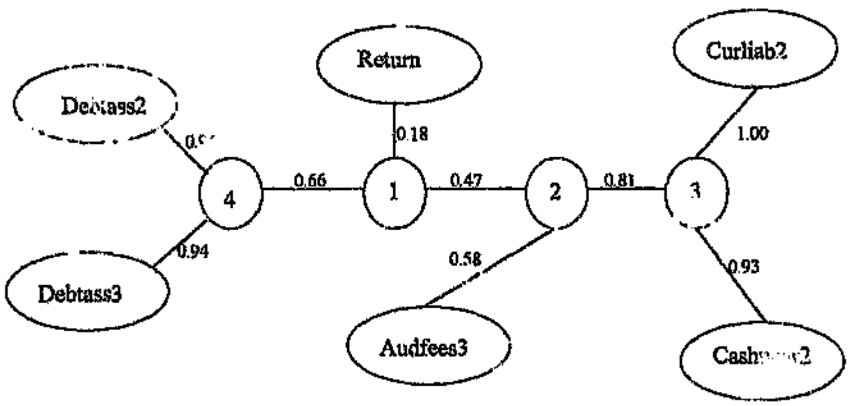


Table 5.2 The variables in Figure 5.2

Debtass	Debt asset ratio
Audfees	Audit Fees
Currliab	Current Liabilities
Cashnear	Cash and near cash

The accuracy of some of the belief nets improve when they are tested on the remaining 117 companies. The Divcovr2-Intcovr1 and the Intbrcl5-Intcovr1 Belief nets both improve their accuracy from 72% to 81%. These Belief nets are in Figures 5.3 and 5.4.

Figure 5.3 The Divcovr2-Intcovr1 Belief net

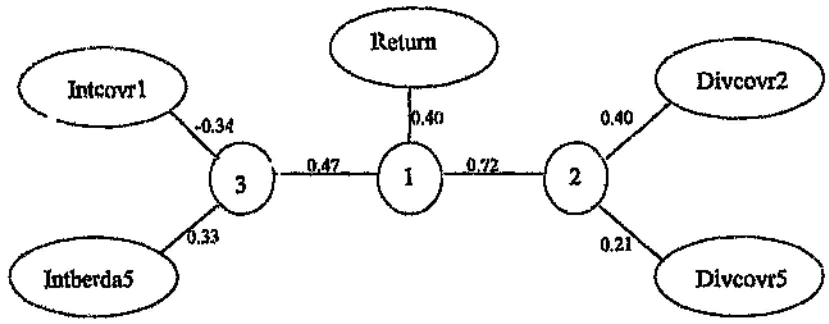


Table 5.3 The variables in Figure 5.3

Intcovr	Interest Cover
Intberda	Int. Bear. Debt/Ass Ratio
Divcovr	Dividend cover

Figure 5.4 The Intbrcl5-Intcovr1 Belief net

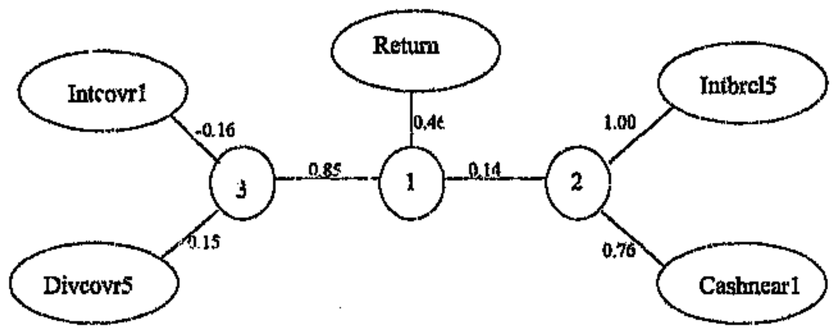


Table 5.4 The variables in Figure 5.4

Intcovr	Interest Cover
Divcovr	Dividend cover
Intbrcl	Interest Bearing-Current Liabilities
Cash near	Cash and near cash

This section shows that the accuracy when using the 240 companies to create the nets and the remaining 117 companies to test the Belief nets, does not change substantially. The Belief nets do not depend on the companies used to create them, but are valid for the remaining companies as well.

5.2.2 Prior as the observed frequencies for 240 companies

The observed relative frequencies for the states of Return are 0.267 for state one, 0.279 for state two, 0.246 for state three and 0.208 for state four.

By taking the observed frequencies as the prior probabilities for the states, on 240 companies, nineteen Belief nets have an accuracy of more than 50%. Testing these Belief nets on the remaining 117 companies, five Belief nets still have an accuracy of more than 50%.

None of the Belief nets have an accuracy of more than 70%, when measured on the 240 companies.

This shows how sensitive the Algorithm is to changes in the prior probabilities.

5.3 Linear Regression

To compare the results of my research with another method, I used a standard Linear Regression package. The data was first discritized into states. The package selected the relevant variables from the 65 independent variables, by stepwise regression. The estimates for the dependent variable, the state of Return, was then rounded of to the nearest state.

An adapted version of the program to measure accuracy, program four in Section 4.3, calculated the accuracy of the rounded Linear Regression estimates.

I did the stepwise Linear Regression after the variables were converted to their respective states (the discritized data). I did this to be more comparable to the states of the belief networks. Clearly this approach puts the Linear Regression at a disadvantage. Further study must be done to get a more complete comparison between Bayes nets and Linear Regression.

Using the same sets of companies for training and testing, the equation found by Linear Regression is

Return = -1.4106 - 0.6952Curasset2 + 0.8636Curasset4 + 0.2007Divcovr2 + 0.5717Intbrcl3 + 0.4810Mintrec5 + 0.1560Mdivyldp2

Table 5.5 The variables in the 240 company Linear regression

Curasset	Current Assets
Divcovr	Dividend cover
Intbrcl	Interest Bearing debt to current liabilities
Mintrec	Interest received (M for states inverted)
Mdivyldp	Dividend yield percentage

The accuracy for the Linear Regression model measured 0.346 on the 240 company Training set, C. On the 117 remaining companies, the accuracy was only 0.188.

When accuracy is measured as the ability to classify Return into the four categories then, all the Belief nets gave a better accuracy than stepwise linear regression. The Belief nets given in Sections 5.2.1 gave an accuracy of more than 0.70 on the test set. To measure accuracy, the penalty for miscalculation was equal, irrespective of the distance from the correct classification. The accuracy is the proportion of times the correct state is predicted.

In this case, the modified Pearl algorithm clearly outperforms Linear Regression.

The fact that the adapted Pearl algorithm performs nearly the same on both sets, confirms the possibility that the adapted Pearl algorithm might model the underlying process better.

CHAPTER 6

Conclusion and recommendations for future Research

The heuristic developed and tested in Chapter three of this research report opens up possibilities to generalize to more than two states. This heuristic is an improvement, but still not perfect. If we can generalize the states of variables from binary to four states, then we can generalize it further. This heuristic lends itself to generalization to any even number of states. The number of data items tended to limit further generalization in this case, because some empty cells caused singular matrices.

Musick [Musick(6)] has an approach that approximates the variables by estimating the probability density function as a Beta distribution. This makes the prior probability a binary probability density function of a given variable. It is possible to extend this thinking to the Pearl algorithm. With this approach, link matrices, that have elements that are functions with a stochastic variable as argument, can be found.

We need to do an in depth sensitivity analysis to find the robustness of the link matrices.

Before promoting the Pearl algorithm as a forecasting technique, we must evaluate the method over different time periods.

To really use the Pearl algorithm as a portfolio management technique, we must qualitatively investigate the statistical dependencies of the Belief networks, in Section 5.2.1.

Some questions that still remain unanswered:
Which are the best values to take as the prior probabilities for the states of the root?
What are the best values to choose as likelihood of the states of the dependent variable?

References:

1. LAZARFELD, P.F., "Latent structure analysis, in measurement and prediction", eds. S. A. Stoufer, L. Guttman, E.A. Suchman, P.F.Lazarfeld, S.A.Star and J.A.Classen, Wiley, New York, 1966
2. GHINEA, Gheorghita., "Structure extraction from data - a case study", Proceedings of the 10th Annual Conference for Masters and Ph.D. Students in Computer Science and Information Systems. 4-5 July 1995, FRD conference at Rhodes University.
3. PEARL, J., "Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference" Morgan Kaufmann, San Mateo, California 1988
4. LUBINSKY, D., "Tutorial on Machine Learning and Data Mining", Proceedings SAICSIT 95, Industry meets Academia, 25 & 26 May 1995, Edited by Steenkamp A.L.
5. ELTON E.J. and GRUBER M.J., "Modern Portfolio Theory and Investment Analysis" Fifth edition, John Wiley & Sons, Inc., 1995
6. MUSICK, C. R., "Belief Network Induction", A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science in the graduate Division of the University of California at Berkeley, 1994.
7. MCGREGOR's Information Services (PTY) LTD, P. O. Box 651302 Benmore South Africa.
8. SHAFER, G., "A Mathematical Theory of Evidence", Princeton University Press, 1976.
9. MAKRIDAKIS, S., WHEELWRIGHT, S. C. and MCGEE, V. E., "Forecasting: Methods and Applications", 2nd ed., John Wiley & Sons, 1983
10. TURBAN, E. in collaboration with Frenzel, L. E. Jr., "Expert Systems and Applied Artificial Intelligence", Maxwell Macmillan International, 1992.
11. BUNTINE, W. L., "Operations for learning with Graphical Models", Journal of Artificial Intelligence Research 2, p159-225, 1994
12. BERGER, J. O., "Statistical Decision Theory Foundations, Concepts and Methods", Springer Verlag, 1980
13. CURWIN J. and SLATER R., "Quantitative Methods for Business Decisions" (third ed.), Chapman and Hall, 1991.
14. The Johannesburg Stock Exchange, "The Investors Guide", ISSN 0250-1732

Appendix A:

Correlation Matrice between the 66 most relevant Variables

	return	AUDFEES3	AUDFEES4	AUDFEES5	CASHNEAR1	CASHNEAR2	CASHNEAR4	CASHNEAR5	CEFFTAX4	CEFFTAX5	CURASSET2	CURASSET3
return	1.00	0.05	0.05	0.05	0.05	0.07	0.05	0.05	0.10	0.09	0.05	0.05
AUDFEES3	0.05	1.00	0.95	0.93	0.41	0.44	0.47	0.38	0.05	0.04	0.28	0.47
AUDFEES4	0.05	0.95	1.00	0.99	0.31	0.34	0.37	0.30	0.07	0.05	0.25	0.45
AUDFEES5	0.05	0.93	0.99	1.00	0.27	0.30	0.34	0.29	0.07	0.06	0.20	0.40
CASHNEAR1	0.05	0.41	0.31	0.27	1.00	0.94	0.83	0.72	-0.08	-0.04	0.65	0.67
CASHNEAR2	0.07	0.44	0.34	0.30	0.94	1.00	0.86	0.76	-0.07	-0.03	0.74	0.77
CASHNEAR4	0.05	0.47	0.37	0.34	0.83	0.86	1.00	0.95	-0.04	-0.02	0.45	0.55
CASHNEAR5	0.05	0.38	0.30	0.29	0.72	0.76	0.95	1.00	-0.02	0.00	0.27	0.37
CEFFTAX4	0.10	0.05	0.07	0.07	-0.08	-0.07	-0.04	-0.02	1.00	0.73	-0.03	-0.02
CEFFTAX5	0.09	0.04	0.05	0.06	-0.04	-0.03	-0.02	0.00	0.73	1.00	0.01	0.01
CURASSET2	0.05	0.28	0.25	0.20	0.65	0.74	0.45	0.27	-0.03	0.01	1.00	0.97
CURASSET3	0.05	0.47	0.45	0.40	0.67	0.77	0.55	0.37	-0.02	0.01	0.97	1.00
CURASSET4	0.05	0.42	0.41	0.37	0.65	0.76	0.54	0.36	-0.02	0.01	0.97	1.00
CURASSET5	0.05	0.71	0.69	0.68	0.71	0.80	0.77	0.66	0.01	0.04	0.73	0.86
CURLIAB1	0.07	0.49	0.34	0.27	0.86	0.93	0.77	0.65	-0.06	-0.02	0.73	0.76
CURLIAB2	0.07	0.47	0.32	0.26	0.85	0.93	0.77	0.64	-0.05	-0.02	0.77	0.80
CURLIAB3	0.07	0.55	0.39	0.33	0.82	0.90	0.80	0.68	-0.05	-0.02	0.68	0.74
CURLIAB4	0.07	0.53	0.39	0.33	0.83	0.92	0.80	0.68	-0.05	-0.02	0.71	0.77
CURLIAB5	0.06	0.61	0.44	0.38	0.76	0.83	0.83	0.77	-0.03	-0.02	0.41	0.52
Ncurrat1	0.11	0.10	0.09	0.08	0.12	0.12	0.11	0.10	-0.02	-0.02	0.07	0.07
Ncurrat3	0.07	0.10	0.09	0.08	0.12	0.10	0.07	0.06	0.02	0.03	0.06	0.06
DEBTASS1	0.08	0.17	0.15	0.13	0.23	0.22	0.17	0.12	0.11	0.05	0.16	0.17
DEBTASS2	0.12	0.17	0.15	0.14	0.23	0.22	0.16	0.12	0.10	0.03	0.16	0.17
DEBTASS3	0.11	0.17	0.14	0.14	0.24	0.24	0.17	0.13	0.15	0.09	0.17	0.17
DEBTASS4	0.12	0.19	0.16	0.15	0.26	0.26	0.19	0.14	0.15	0.07	0.18	0.18
DEBTASS5	0.14	0.19	0.16	0.16	0.26	0.25	0.19	0.15	0.16	0.08	0.16	0.17
Ndeceam4	0.08	-0.16	-0.16	-0.16	-0.11	-0.11	-0.15	-0.15	-0.07	-0.05	-0.06	-0.10
Ndeceam5	0.09	-0.13	-0.13	-0.14	-0.10	-0.09	-0.13	-0.13	-0.04	-0.03	-0.05	-0.08
Ndeftax5	0.05	-0.54	-0.57	-0.56	-0.31	-0.33	-0.35	-0.31	-0.04	-0.02	-0.15	-0.28
DIREMOL3	0.08	0.61	0.64	0.65	0.25	0.27	0.33	0.29	0.10	0.10	0.20	0.34
DIREMOL4	0.07	0.55	0.58	0.60	0.27	0.27	0.31	0.27	0.10	0.12	0.23	0.35
DIREMOL5	0.08	0.53	0.57	0.59	0.20	0.19	0.24	0.22	0.06	0.06	0.13	0.25
DIVCOVR2	0.12	0.03	0.02	0.03	0.06	0.06	0.03	0.04	0.05	0.07	0.04	0.04
DIVCOVR3	0.16	0.05	0.04	0.05	0.04	0.05	0.03	0.04	0.11	0.08	0.03	0.04
DIVCOVR4	0.12	0.06	0.06	0.06	0.04	0.05	0.03	0.04	0.00	0.02	0.03	0.04
DIVCOVR5	0.06	0.02	0.02	0.02	0.01	0.02	0.01	0.01	-0.01	-0.02	0.01	0.01
Ndivyldp1	0.11	0.10	0.10	0.11	0.03	0.03	0.05	0.05	-0.04	-0.03	0.01	0.02
Ndivyldp2	0.08	0.12	0.11	0.12	0.07	0.06	0.07	0.07	0.02	0.05	0.04	0.06
Ndivyldp3	0.12	0.14	0.13	0.14	0.11	0.10	0.10	0.09	0.07	0.10	0.07	0.09
Ndivpaid1	0.08	-0.32	-0.29	-0.27	-0.38	-0.34	-0.40	-0.39	-0.05	-0.06	-0.25	-0.31
Ndivpaid2	0.07	-0.41	-0.40	-0.39	-0.35	-0.35	-0.44	-0.41	-0.02	-0.02	-0.27	-0.37

Correlation Matrice between the 66 most relevant Variables

	return	AUDFEES3	AUDFEES4	AUDFEES5	CASHNEAR1	CASHNEAR2	CASHNEAR4	CASHNEAR5	CEFFTAX4	CEFFTAX5	CURASSET2	CURASSET3
Ndivpaid3	0.07	-0.40	-0.39	-0.38	-0.33	-0.32	-0.44	-0.41	-0.04	-0.05	-0.23	-0.33
Ndivpaid4	0.09	-0.31	-0.29	-0.28	-0.31	-0.29	-0.43	-0.41	-0.04	-0.06	-0.21	-0.29
Ndivpaid5	0.11	-0.34	-0.34	-0.34	-0.25	-0.25	-0.37	-0.37	-0.02	-0.06	-0.16	-0.26
Nemyldp2	0.05	0.06	0.0[illegible]	0.06	0.02	0.01	0.02	0.02	-0.01	-0.02	0.00	0.01
Nfixasset2	0.12	-0.37	-0.39	-0.38	-0.25	-0.24	-0.24	-0.22	0.05	0.03	-0.22	-0.29
Nfixasset3	0.11	-0.55	-0.61	-0.61	-0.22	-0.23	-0.26	-0.23	0.04	0.02	-0.19	-0.33
Nfixasset4	0.11	-0.55	-0.61	-0.61	-0.21	-0.21	-0.25	-0.22	0.03	0.02	-0.19	-0.33
Nfixasset5	0.11	-0.55	-0.61	-0.62	-0.19	-0.19	-0.24	-0.22	0.04	0.02	-0.15	-0.29
Nfundshr5	0.07	-0.16	-0.16	-0.17	-0.08	-0.08	-0.10	-0.11	-0.04	-0.02	-0.04	-0.08
Ngrincm5	0.06	-0.62	-0.67	-0.68	-0.29	-0.30	-0.40	-0.39	-0.08	-0.10	-0.21	-0.37
INTBERDA4	0.06	0.19	0.13	0.11	0.47	0.43	0.35	0.28	-0.09	-0.11	0.29	0.29
INTBERDA5	0.06	0.19	0.14	0.12	0.46	0.43	0.35	0.28	-0.09	-0.12	0.28	0.28
Nint_fin1	0.06	-0.44	-0.50	-0.50	-0.14	-0.12	-0.15	-0.14	-0.03	-0.01	-0.15	-0.24
Nint_fin3	0.06	-0.56	-0.64	-0.65	-0.11	-0.12	-0.15	-0.14	-0.03	-0.02	-0.15	-0.29
INTBRCL1	0.07	0.44	0.28	0.22	0.85	0.92	0.75	0.62	-0.06	-0.03	0.72	0.74
INTBRCL3	0.07	0.43	0.28	0.22	0.83	0.92	0.74	0.61	-0.06	-0.02	0.76	0.79
INTBRCL2	0.07	0.47	0.30	0.23	0.81	0.89	0.76	0.64	-0.06	-0.03	0.69	0.72
INTBRCL4	0.07	0.45	0.29	0.23	0.82	0.9[illegible]	0.76	0.64	-0.06	-0.03	0.72	0.75
INTBRCL5	0.07	0.52	0.33	0.27	0.76	0.82	0.80	0.74	-0.05	-0.03	0.41	0.49
NIntcovr1	0.06	0.05	0.05	0.05	0.02	0.02	0.01	0.01	-0.01	0.03	0.02	0.03
Nintcovr2	0.06	0.05	0.05	0.05	0.03	0.02	0.02	0.02	-0.03	-0.04	0.02	0.03
Nintcovr5	0.11	0.03	0.04	0.04	0.03	0.03	0.03	0.02	0.06	0.04	0.02	0.02
Nintrec5	0.06	-0.31	-0.36	-0.38	-0.07	-0.09	-0.21	-0.25	-0.10	-0.12	-0.09	-0.19
NMayprice	0.10	-0.20	-0.17	-0.16	-0.19	-0.14	-0.22	-0.20	-0.01	-0.04	-0.07	-0.10

Correlation Matrice between the 66 most relevant Variables

	CURASSET4	CURASSET5	CURLIAB1	CURLIAB2	CURLIAB3	CURLIAB4	CURLIAB5	Ncurrat1	Ncurrat3	DEBTASS1	DEBTASS2	DEBTASS3
return	0.05	0.05	0.07	0.07	0.07	0.07	0.06	0.11	0.07	0.08	0.12	0.11
AUDFEES3	0.42	0.71	0.49	0.47	0.55	0.53	0.61	0.10	0.10	0.17	0.17	0.17
AUDFEES4	0.41	0.69	0.34	0.32	0.39	0.39	0.44	0.09	0.09	0.15	0.15	0.14
AUDFEES5	0.37	0.68	0.27	0.26	0.33	0.33	0.38	0.08	0.08	0.13	0.14	0.14
CASHNEAR1	0.65	0.71	0.86	0.85	0.82	0.83	0.76	0.12	0.12	0.23	0.23	0.24
CASHNEAR2	0.76	0.80	0.93	0.93	0.90	0.92	0.83	0.12	0.10	0.22	0.22	0.24
CASHNEAR4	0.54	0.77	0.77	0.77	0.80	0.80	0.83	0.11	0.07	0.17	0.16	0.17
CASHNEAR5	0.36	0.66	0.65	0.64	0.68	0.68	0.77	0.10	0.06	0.12	0.12	0.13
CEFFTAX4	-0.02	0.01	-0.06	-0.05	-0.05	-0.05	-0.03	-0.02	0.02	0.11	0.10	0.15
CEFFTAX5	0.01	0.04	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	0.03	0.05	0.03	0.09
CURASSET2	0.97	0.73	0.73	0.77	0.68	0.71	0.41	0.07	0.06	0.16	0.16	0.17
CURASSET3	1.00	0.86	0.76	0.80	0.74	0.77	0.52	0.07	0.06	0.17	0.17	0.17
CURASSET4	1.00	0.85	0.73	0.77	0.71	0.74	0.47	0.07	0.06	0.16	0.16	0.16
CURASSET5	0.85	1.00	0.73	0.75	0.76	0.78	0.68	0.10	0.08	0.17	0.17	0.18
CURLIAB1	0.73	0.73	1.00	1.00	0.99	0.99	0.91	0.13	0.12	0.25	0.25	0.26
CURLIAB2	0.77	0.75	1.00	1.00	0.99	0.99	0.88	0.13	0.11	0.24	0.24	0.25
CURLIAB3	0.71	0.76	0.99	0.99	1.00	1.00	0.94	0.13	0.12	0.24	0.24	0.25
CURLIAB4	0.74	0.78	0.99	0.99	1.00	1.00	0.92	0.13	0.11	0.24	0.24	0.25
CURLIAB5	0.47	0.68	0.91	0.88	0.94	0.92	1.00	0.13	0.11	0.22	0.22	0.23
Ncurrat1	0.07	0.10	0.13	0.13	0.13	0.13	0.13	1.00	0.57	0.34	0.27	0.22
Ncurrat3	0.06	0.08	0.12	0.11	0.12	0.11	0.11	0.57	1.00	0.24	0.23	0.28
DEBTASS1	0.16	0.17	0.25	0.24	0.24	0.24	0.22	0.34	0.24	1.00	0.90	0.80
DEBTASS2	0.16	0.17	0.25	0.24	0.24	0.24	0.22	0.27	0.23	0.90	1.00	0.90
DEBTASS3	0.16	0.18	0.26	0.25	0.25	0.25	0.23	0.22	0.28	0.80	0.90	1.00
DEBTASS4	0.17	0.19	0.27	0.26	0.26	0.26	0.24	0.21	0.25	0.77	0.87	0.94
DEBTASS5	0.16	0.18	0.26	0.25	0.25	0.25	0.24	0.23	0.25	0.75	0.84	0.88
Ndecearn4	-0.09	-0.18	-0.07	-0.06	-0.07	-0.07	-0.09	-0.01	-0.02	0.08	0.08	0.05
Ndecearn5	-0.08	-0.16	-0.04	-0.04	-0.05	-0.05	-0.07	0.03	0.03	0.12	0.13	0.10
Ndeftax5	-0.25	-0.48	-0.27	-0.25	-0.29	-0.29	-0.34	-0.09	-0.07	-0.06	-0.06	-0.07
DIREMOL3	0.32	0.53	0.18	0.18	0.22	0.22	0.24	0.10	0.10	0.17	0.19	0.20
DIREMOL4	0.33	0.51	0.17	0.17	0.20	0.20	0.20	0.09	0.09	0.14	0.15	0.18
DIREMOL5	0.23	0.43	0.11	0.10	0.14	0.14	0.17	0.08	0.09	0.14	0.15	0.17
DIVCOVR2	0.03	0.05	0.05	0.05	0.05	0.05	0.05	0.08	0.12	0.09	0.13	0.14
DIVCOVR3	0.03	0.06	0.04	0.04	0.04	0.04	0.05	0.10	0.06	0.15	0.14	0.15
DIVCOVR4	0.04	0.07	0.03	0.03	0.03	0.03	0.04	0.11	0.08	0.17	0.17	0.17
DIVCOVR5	0.01	0.02	0.01	0.01	0.01	0.01	0.01	-0.03	-0.05	-0.03	-0.03	-0.03
Ndivyldp1	0.02	0.06	0.02	0.02	0.03	0.02	0.04	-0.02	0.06	-0.01	0.02	0.05
Ndivyldp2	0.06	0.10	0.05	0.05	0.05	0.05	0.05	0.10	0.04	0.00	-0.04	-0.01
Ndivyldp3	0.08	0.13	0.08	0.07	0.07	0.07	0.06	0.11	0.00	0.07	0.03	0.00
Ndivpaid1	-0.30	-0.43	-0.27	-0.27	-0.27	-0.27	-0.27	-0.11	-0.06	0.09	0.12	0.12
Ndivpaid2	-0.35	-0.51	-0.26	-0.26	-0.28	-0.28	-0.27	-0.10	-0.03	0.12	0.13	0.12

Correlation Matrice between the 66 most relevant Variables

	CURASSET4	CURASSET5	CURLIAB1	CURLIAB2	CURLIAB3	CURLIAB4	CURLIAB5	Ncurrat1	Ncurrat3	DEBTASS1	DEBTASS2	DEBTASS3
Ndivpaid3	-0.32	-0.49	-0.22	-0.22	-0.24	-0.24	-0.25	-0.08	-0.02	0.12	0.13	0.13
Ndivpaid4	-0.27	-0.44	-0.19	-0.19	-0.20	-0.20	-0.21	-0.08	-0.02	0.14	0.15	0.15
Ndivpaid5	-0.25	-0.44	-0.15	-0.14	-0.16	-0.17	-0.19	-0.07	-0.01	0.15	0.16	0.16
Nemyldp2	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.02	-0.05	-0.07	-0.07	-0.11
Nfixasset2	-0.27	-0.42	-0.19	-0.18	-0.19	-0.19	-0.20	-0.15	-0.11	0.01	0.01	0.03
Nfixasset3	-0.31	-0.53	-0.16	-0.15	-0.19	-0.19	-0.21	-0.13	-0.11	0.02	0.02	0.02
Nfixasset4	-0.31	-0.53	-0.15	-0.14	-0.18	-0.18	-0.20	-0.12	-0.10	0.03	0.03	0.02
Nfixasset5	-0.27	-0.51	-0.12	-0.12	-0.16	-0.16	-0.19	-0.12	-0.10	0.05	0.04	0.03
Nfundshr5	-0.08	-0.16	-0.04	-0.04	-0.05	-0.05	-0.07	-0.03	-0.02	0.10	0.10	0.04
Ngrincm5	-0.35	-0.63	-0.18	-0.18	-0.22	-0.22	-0.26	-0.08	-0.05	0.06	0.06	0.05
INTBERDA4	0.28	0.29	0.44	0.43	0.42	0.43	0.38	0.14	0.15	0.52	0.58	0.64
INTBERDA5	0.27	0.28	0.43	0.42	0.41	0.42	0.38	0.14	0.15	0.52	0.58	0.62
Nint_fin1	-0.22	-0.42	-0.05	-0.04	-0.06	-0.06	-0.08	-0.10	-0.04	-0.10	-0.08	-0.05
Nint_fin3	-0.26	-0.51	-0.05	-0.04	-0.08	-0.08	-0.10	-0.06	-0.04	-0.06	-0.07	-0.07
INTBRCL1	0.72	0.69	1.00	0.99	0.98	0.98	0.89	0.12	0.11	0.24	0.24	0.25
INTBRCL3	0.76	0.71	0.99	1.00	0.98	0.98	0.87	0.12	0.10	0.23	0.24	0.24
INTBRCL2	0.69	0.69	0.99	0.99	0.99	0.99	0.92	0.12	0.11	0.24	0.24	0.25
INTBRCL4	0.72	0.71	0.99	0.99	0.99	0.99	0.90	0.12	0.11	0.24	0.24	0.25
INTBRCL5	0.45	0.61	0.92	0.90	0.94	0.93	0.99	0.12	0.11	0.22	0.22	0.23
Nintcovr1	0.02	0.03	0.02	0.02	0.02	0.02	0.02	-0.02	0.06	0.14	0.16	0.17
Nintcovr2	0.03	0.04	0.02	0.02	0.02	0.02	0.02	0.11	0.00	0.15	0.13	0.11
Nintcovr5	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.04	-0.06	0.10	0.10	0.11
Nintrec5	-0.20	-0.40	-0.01	-0.01	-0.04	-0.04	-0.06	-0.03	0.00	0.09	0.09	0.09
NMayprice	-0.10	-0.20	-0.11	-0.09	-0.10	-0.10	-0.13	0.03	0.01	0.13	0.15	0.13

Correlation Matrice between the 66 most relevant Variables

	DEBTASS4	DEBTASS5	Ndecearn4	Ndecearn5	Ndeftax5	DIREMOL3	DIREMOL4	DIREMOL5	DIVCOVR2	DIVCOVR3	DIVCOVR4	DIVCOVR5
return	0.12	0.14	0.08	0.09	0.05	0.08	0.07	0.08	0.12	0.16	0.12	0.06
AUDFEES3	0.19	0.19	-0.16	-0.13	-0.54	0.61	0.55	0.53	0.03	0.05	0.06	0.02
AUDFEES4	0.16	0.16	-0.16	-0.13	-0.57	0.64	0.58	0.57	0.02	0.04	0.06	0.02
AUDFEES5	0.15	0.16	-0.16	-0.14	-0.56	[illegible]	0.60	0.59	0.03	0.05	0.06	0.02
CASHNEAR1	0.26	0.26	-0.11	-0.10	-0.31	[illegible]	27	0.20	0.06	0.04	[illegible]	0.01
CASHNEAR2	0.26	0.25	-0.11	-0.09	-0.33	0.27	0.27	0.19	0.08	0.05	0.05	0.02
CASHNEAR4	0.19	0.19	-0.15	-0.13	-0.35	0.33	0.31	0.24	0.03	0.03	0.03	0.01
CASHNEAR5	0.14	0.15	-0.15	-0.13	-0.31	0.29	0.27	0.22	0.04	0.04	0.04	0.01
CEFFTAX4	0.15	0.16	-0.07	-0.04	-0.04	0.10	0.10	0.06	0.05	0.11	0.00	-0.01
CEFFTAX5	0.07	0.08	-0.05	-0.03	-0.02	0.10	0.12	0.08	0.07	0.08	0.02	-0.02
CURASSET2	0.18	0.16	-0.06	-0.05	-0.15	0.20	0.23	0.13	0.04	0.03	0.03	0.01
CURASSET3	0.18	0.17	-0.10	-0.08	-0.28	0.34	0.35	0.25	0.04	0.04	0.04	0.01
CURASSET4	0.17	0.16	-0.09	-0.08	-0.25	0.32	0.33	0.23	0.03	0.03	0.04	0.01
CURASSET5	0.19	0.18	-0.18	-0.16	-0.48	0.53	0.51	0.43	0.05	0.06	0.07	0.02
CURLIAB1	0.27	0.26	-0.07	-0.04	-0.27	0.18	0.17	0.11	0.05	0.04	0.03	0.01
CURLIAB2	0.26	0.25	-0.06	-0.04	-0.25	0.18	0.17	0.10	0.05	0.04	0.03	0.01
CURLIAB3	0.26	0.25	-0.07	-0.05	-0.29	0.22	0.20	0.14	0.05	0.04	0.03	0.01
CURLIAB4	0.26	0.25	-0.07	-0.05	-0.29	0.22	0.20	0.14	0.05	0.04	0.03	0.01
CURLIAB5	0.24	0.24	-0.09	-0.07	-0.34	0.24	0.20	0.17	0.05	0.05	0.04	0.01
Ncurrat1	0.21	0.23	-0.01	0.03	-0.09	0.10	0.09	0.08	0.08	0.10	0.11	-0.03
Ncurrat3	0.25	0.25	-0.02	0.03	-0.07	0.10	0.09	0.09	0.12	0.06	0.08	-0.05
DEBTASS1	0.77	0.75	0.08	0.12	-0.06	0.17	0.14	0.14	0.09	0.15	0.17	-0.03
DEBTASS2	0.87	0.84	0.08	0.13	-0.06	0.19	0.15	0.15	0.13	0.14	0.17	-0.03
DEBTASS3	0.94	0.88	0.05	0.10	-0.07	0.20	0.18	0.17	0.14	0.15	0.17	-0.03
DEBTASS4	1.00	0.94	0.03	0.08	-0.07	0.21	0.19	0.19	0.14	0.16	0.18	-0.03
DEBTASS5	0.94	1.00	0.04	0.08	-0.04	0.20	0.18	0.18	0.16	0.16	0.20	-0.02
Ndecearn4	0.03	0.04	1.00	0.88	0.23	-0.14	-0.16	-0.13	-0.09	-0.21	-0.21	-0.04
Ndecearn5	0.08	0.08	0.88	1.00	0.20	-0.15	-0.17	-0.15	-0.03	-0.12	-0.13	-0.38
Ndeftax5	-0.07	-0.04	0.23	0.20	1.00	-0.35	-0.34	-0.33	-0.01	-0.02	-0.04	-0.01
DIREMOL3	0.21	0.20	-0.14	-0.15	-0.35	1.00	0.94	0.83	0.04	0.08	0.12	0.03
DIREMOL4	0.19	0.18	-0.16	-0.17	-0.34	0.94	1.00	0.90	0.07	0.07	0.13	0.03
DIREMOL5	0.19	0.18	-0.13	-0.15	-0.33	0.83	0.90	1.00	0.05	0.04	0.09	0.03
DIVCOVR2	0.14	0.16	-0.09	-0.03	-0.01	0.04	0.07	0.05	1.00	0.47	0.29	0.09
DIVCOVR3	0.16	0.16	-0.21	-0.12	-0.02	0.08	0.07	0.04	0.47	1.00	0.53	0.11
DIVCOVR4	0.18	0.20	-0.21	-0.13	-0.04	0.12	0.13	0.09	0.29	0.53	1.00	0.11
DIVCOVR5	-0.03	-0.02	-0.04	-0.38	-0.01	0.03	0.03	0.03	0.09	0.11	0.11	1.00
Ndivyldp1	0.05	0.03	0.00	0.02	-0.05	0.17	0.16	0.15	0.10	0.16	0.10	-0.01
Ndivyldp2	0.00	-0.04	-0.08	-0.04	-0.07	0.16	0.14	0.12	0.05	0.12	0.10	-0.02
Ndivyldp3	0.03	0.03	-0.10	-0.08	-0.08	0.23	0.21	0.17	0.12	0.26	0.11	0.03
Ndivpaid1	0.12	0.12	0.20	0.24	0.38	-0.39	-0.45	-0.39	0.03	0.09	0.03	0.00
Ndivpaid2	0.11	0.12	0.23	0.27	0.46	-0.53	-0.56	-0.46	0.06	0.09	0.02	-0.01

Correlation Matrice between the 66 most relevant Variables

	DEBTASS4	DEBTASS5	Ndeceam4	Ndeceam5	Ndeflax5	DIREMOL3	DIREMOL4	DIREMOL5	DIVCOVR2	DIVCOVR3	DIVCOVR4	DIVCOVR5
Ndivpaid3	0.12	0.12	0.19	0.23	0.41	-0.52	-0.53	-0.45	0.06	0.11	0.03	-0.01
Ndivpaid4	0.14	0.14	0.19	0.24	0.36	-0.46	-0.49	-0.40	0.07	0.12	0.05	0.00
Ndivpaid5	0.15	0.15	0.21	0.27	0.38	-0.47	-0.51	-0.44	0.08	[illegible]	0.06	0.01
Nemyldp2	-0.11	-0.12	0.03	0.01	-0.03	0.06	0.06	0.05	-0.37	-0.22	-0.09	-0.04
Nfixasset2	0.03	0.05	0.19	0.18	0.56	-0.30	-0.31	-0.29	0.01	0.03	0.00	0.00
Nfixasset3	0.01	0.04	0.20	0.20	0.67	-0.44	-0.44	-0.43	0.03	0.04	-0.01	0.00
Nfixasset4	0.02	0.05	0.19	0.19	0.68	-0.43	-0.43	-0.42	0.04	0.04	-0.01	0.00
Nfixasset5	0.03	0.06	0.22	0.22	0.64	-0.44	-0.45	-0.44	0.04	0.04	-0.01	0.01
Nfundshr5	0.03	0.04	0.86	0.81	0.27	-0.15	-0.20	-0.15	-0.06	-0.17	-0.17	-0.03
Ngrincm5	0.04	0.05	0.28	0.30	0.56	-0.64	-0.66	-0.61	-0.01	0.01	-0.04	-0.02
INTBERDA4	0.67	0.62	0.03	0.04	-0.10	0.09	0.08	0.09	0.12	0.11	0.08	-0.01
INTBERDA5	0.64	0.68	0.05	0.08	-0.08	0.08	0.07	0.05	0.13	0.11	0.10	-0.02
Nint_fin1	-0.05	-0.03	0.17	0.17	0.49	-0.48	-0.51	-0.47	-0.03	-0.03	-0.06	-0.03
Nint_fin3	-0.07	-0.05	0.16	0.16	0.53	[illegible]	-0.57	-0.53	-0.01	-0.02	-0.05	-0.02
INTBRCL1	0.26	0.25	-0.05	-0.03	-0.23	0.13	0.12	0.06	0.05	0.04	0.03	0.01
INTBRCL3	0.25	0.24	-0.04	-0.02	-0.22	0.14	0.13	0.06	0.05	0.03	0.03	0.01
INTBRCL2	0.26	0.25	-0.05	-0.03	-0.23	0.15	0.13	0.07	0.05	0.04	0.03	0.01
INTBRCL4	0.26	0.25	-0.04	-0.03	-0.23	0.14	0.13	0.07	0.05	0.04	0.03	0.01
INTBRCL5	0.24	0.24	-0.06	-0.04	-0.28	0.15	0.12	0.09	0.05	0.04	0.03	0.01
Nintcovr1	0.15	0.15	0.08	0.06	-0.04	0.09	0.09	0.08	0.05	0.03	0.01	0.02
Nintcovr2	0.13	0.12	[illegible]	0.02	-0.02	0.09	0.09	0.08	0.07	0.05	0.05	0.03
Nintcovr5	0.07	0.11	-0.01	0.01	-0.02	0.07	0.07	0.06	[illegible]	0.03	0.07	0.01
Nintrec5	0.09	0.11	0.21	0.16	0.34	-0.27	-0.24	-0.22	-0.02	-0.05	-0.06	-0.02
NMayprice	0.12	0.12	0.63	0.64	0.22	-0.20	-0.22	-0.17	0.02	-0.02	-0.04	0.02

Correlation Matrice between the 66 most relevant Variables

	Ndivyldp1	Ndivyldp2	Ndivyldp3	Ndivpaid1	Ndivpaid2	Ndivpaid3	Ndivpaid4	Ndivpaid5	Nemyldp2	Nfixasset1	Nfixasset2	Nfixasset3
return	0.11	0.08	0.12	0.08	0.07	0.07	0.09	0.11	0.05	0.11	0.12	0.11
AUDFEES3	0.10	0.12	0.14	-0.32	-0.41	-0.40	-0.31	-0.34	0.06	-0.41	-0.37	-0.55
AUDFEES4	0.10	0.11	0.13	-0.29	-0.40	-0.39	-0.29	-0.34	0.05	-0.42	-0.39	-0.61
AUDFEES5	0.11	0.12	0.14	-0.27	-0.39	-0.38	-0.28	-0.34	0.06	-0.42	-0.38	-0.61
CASHNEAR1	0.03	0.07	0.11	-0.38	-0.35	-0.33	-0.31	-0.25	0.02	-0.26	-0.25	-0.22
CASHNEAR2	0.03	0.06	0.10	-0.34	-0.35	-0.32	-0.29	-0.25	0.01	-0.25	-0.24	-0.23
CASHNEAR4	0.05	0.07	0.10	-0.40	-0.44	-0.44	-0.43	-0.37	0.02	-0.26	-0.24	-0.26
CASHNEAR5	0.05	0.07	0.09	-0.39	-0.41	-0.41	-0.41	-0.37	0.02	-0.25	-0.22	-0.23
CEFFTAX4	-0.04	0.02	0.07	-0.05	-0.02	-0.04	-0.04	-0.02	-0.01	0.06	0.05	0.04
CEFFTAX5	-0.03	0.05	0.10	-0.06	-0.02	-0.05	-0.06	-0.06	-0.02	0.03	0.03	0.02
CURASSET2	0.01	0.04	0.07	-0.25	-0.27	-0.23	-0.21	-0.16	0.00	-0.23	-0.22	-0.19
CURASSET3	0.02	0.06	0.09	-0.31	-0.37	-0.33	-0.29	-0.26	0.01	-0.29	-0.29	-0.33
CURASSET4	0.02	0.06	0.08	-0.30	-0.35	-0.32	-0.27	-0.25	0.01	-0.27	-0.27	-0.31
CURASSET5	0.06	0.10	0.13	-0.43	-0.51	-0.49	-0.44	-0.44	0.02	-0.43	-0.42	-0.53
CURLIAB1	0.02	0.05	0.08	-0.27	-0.26	-0.22	-0.19	-0.15	0.01	-0.21	-0.19	-0.16
CURLIAB2	0.02	0.05	0.07	-0.27	-0.26	-0.22	-0.19	-0.14	0.01	-0.20	-0.18	-0.15
CURLIAB3	0.03	0.05	0.07	-0.27	-0.28	-0.24	-0.20	-0.16	0.01	-0.20	-0.19	-0.19
CURLIAB4	0.02	0.05	0.07	-0.27	-0.28	-0.24	-0.20	-0.17	0.01	-0.20	-0.19	-0.19
CURLIAB5	0.04	0.05	0.08	-0.27	-0.27	-0.25	-0.21	-0.19	0.01	-0.21	-0.20	-0.21
Ncurrat1	-0.02	0.10	0.11	-0.11	-0.10	-0.08	-0.08	-0.07	0.02	-0.15	-0.15	-0.13
Ncurrat3	0.06	0.04	0.00	-0.06	-0.03	-0.02	-0.02	-0.01	-0.05	-0.11	-0.11	-0.11
DEBTASS1	-0.01	0.00	0.07	0.09	0.12	0.12	0.14	0.15	-0.07	0.00	0.01	0.02
DEBTASS2	0.02	-0.04	0.03	0.12	0.13	0.13	0.15	0.16	-0.07	0.01	0.01	0.02
DEBTASS3	0.05	-0.01	0.00	0.12	0.12	0.13	0.15	0.16	-0.11	0.03	0.03	0.02
DEBTASS4	0.05	0.00	0.03	0.12	0.11	0.12	0.14	0.15	-0.11	0.03	0.03	0.01
DEBTASS5	0.03	-0.04	0.03	0.12	0.12	0.12	0.14	0.15	-0.12	0.04	0.05	0.04
Ndecearn4	0.00	-0.08	-0.10	0.20	0.23	0.19	0.19	0.21	0.03	0.21	0.19	0.20
Ndecearn5	0.02	-0.04	-0.08	0.24	0.27	0.23	0.24	0.27	0.01	0.21	0.18	0.20
Ndeftax5	-0.05	-0.07	-0.08	0.38	0.46	0.41	0.36	0.38	-0.03	0.35	0.56	0.67
DIREMOL3	0.17	0.16	0.23	-0.39	-0.53	-0.52	-0.46	-0.47	0.06	-0.33	-0.30	-0.44
DIREMOL4	0.16	0.14	0.21	-0.45	-0.56	-0.53	-0.49	-0.51	0.06	-0.34	-0.31	-0.44
DIREMOL5	0.15	0.12	0.17	-0.39	-0.46	-0.45	-0.40	-0.44	0.05	-0.32	-0.29	-0.43
DIVCOVR2	0.10	0.05	0.12	0.03	0.06	0.06	0.07	0.06	-0.37	0.01	0.01	0.03
DIVCOVR3	0.16	0.12	0.26	0.09	0.09	0.11	0.12	0.12	-0.22	0.04	0.03	0.04
DIVCOVR4	0.10	0.10	0.11	0.03	0.02	0.03	0.05	0.06	-0.09	-0.01	0.00	-0.01
DIVCOVR5	-0.01	-0.02	0.03	0.00	-0.01	-0.01	0.00	0.01	-0.04	0.00	0.00	0.00
Ndivyldp1	1.00	0.26	0.29	0.07	-0.08	-0.08	-0.07	-0.07	0.16	-0.08	-0.07	-0.08
Ndivyldp2	0.26	1.00	0.43	-0.11	-0.11	-0.11	-0.10	-0.10	0.38	-0.10	-0.09	-0.10
Ndivyldp3	0.29	0.43	1.00	-0.14	-0.14	-0.12	-0.13	-0.11	0.31	-0.11	-0.11	-0.11
Ndivpaid1	0.07	-0.11	-0.14	1.00	0.88	0.86	0.88	0.85	-0.08	0.59	0.64	0.60
Ndivpaid2	-0.08	-0.11	-0.14	0.88	1.00	0.93	0.92	0.90	-0.09	0.55	0.62	0.66

Correlation Matrice between the 66 most relevant Variables

	Ndivyldp1	Ndivyldp2	Ndivyldp3	Ndivpaid1	Ndivpaid2	Ndivpaid3	Ndivpaid4	Ndivpaid5	Nemyldp2	Nfixasset1	Nfixasset2	Nfixasset3
Ndivpaid3	-0.08	-0.11	-0.12	0.86	0.93	1.00	0.96	0.94	-0.08	0.54	0.60	0.62
Ndivpaid4	-0.07	-0.10	-0.13	0.88	0.92	0.96	1.00	0.95	-0.09	0.55	0.61	0.59
Ndivpaid5	-0.07	-0.10	-0.11	0.85	0.90	0.94	0.95	1.00	-0.09	0.59	0.63	0.67
Nemyldp2	0.16	0.38	0.31	-0.08	-0.09	-0.08	-0.09	-0.09	1.00	-0.06	-0.05	-0.06
Nfixasset2	-0.07	-0.09	-0.11	0.64	0.62	0.60	0.61	0.63	-0.05	0.87	1.00	0.92
Nfixasset3	-0.08	-0.10	-0.11	0.60	0.66	0.62	0.59	0.67	-0.06	0.80	0.92	1.00
Nfixasset4	-0.07	-0.10	-0.09	0.59	0.64	0.60	0.57	0.66	-0.06	0.77	0.89	0.99
Nfixasset5	-0.07	-0.10	-0.09	0.56	0.63	0.58	0.54	0.65	-0.06	0.79	0.86	0.97
Nfundshr5	-0.02	-0.08	-0.11	0.18	0.27	0.17	0.17	0.21	0.00	0.24	0.21	0.26
Ngrincm5	-0.10	-0.13	-0.15	0.74	0.82	0.85	0.80	0.86	-0.07	0.66	0.70	0.81
INTBERDA4	0.01	-0.05	-0.01	0.03	0.01	0.03	0.05	0.06	-0.13	-0.05	-0.05	-0.05
INTBERDA5	0.00	-0.05	-0.02	0.03	0.02	0.04	0.06	0.08	-0.10	-0.03	-0.02	-0.02
Nint_fin1	-0.09	-0.09	-0.12	0.53	0.54	0.54	0.54	0.55	-0.02	0.72	0.76	0.74
Nint_fin3	-0.09	-0.09	-0.12	0.43	0.55	0.53	0.50	0.56	-0.03	0.63	0.72	0.82
INTBRCL1	0.01	0.04	0.06	-0.23	-0.22	-0.17	-0.14	-0.10	0.01	-0.15	-0.13	-0.10
INTBRCL3	0.01	0.03	0.06	-0.22	-0.22	-0.18	-0.14	-0.10	0.00	-0.14	-0.13	-0.10
INTBRCL2	0.01	0.04	0.06	-0.22	-0.22	-0.18	-0.15	-0.11	0.01	-0.14	-0.13	-0.10
INTBRCL4	0.01	0.03	0.06	-0.22	-0.22	-0.18	-0.15	-0.11	0.00	-0.14	-0.13	-0.10
INTBRCL5	0.02	0.04	0.06	-0.22	-0.21	-0.18	-0.15	-0.11	0.01	-0.14	-0.13	-0.11
Nintcovr1	0.05	-0.02	-0.01	0.08	0.05	0.02	0.03	0.04	-0.03	0.10	0.09	0.08
Nintcovr2	0.01	0.12	0.16	0.00	0.00	-0.02	-0.02	0.00	0.01	0.02	0.02	0.02
Nintcovr5	-0.03	-0.04	-0.04	-0.02	-0.03	-0.02	-0.02	0.00	-0.06	0.01	0.01	0.02
Nintrec5	-0.06	-0.08	-0.10	0.46	0.56	0.51	0.53	0.57	-0.03	0.45	0.44	0.52
NMayprice	-0.12	-0.14	-0.18	0.43	0.40	0.40	0.43	0.42	-0.08	0.32	0.27	0.25

Correlation Matrice between the 66 most relevant Variables

	Nfixasset4	Nfixasset5	Nfundshr5	Ngrincm5	INTBERDA4	INTBERDA5	Nint_fin1	Nint_fin3	INTBRCL1	INTBRCL3	INTBRCL2	INTBRCL4
return	0.11	0.11	0.07	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.07
AUDFEES3	-0.55	-0.55	-0.16	-0.62	0.19	0.19	-0.44	-0.56	0.44	0.43	0.45	0.47
AUDFEES4	-0.61	-0.61	-0.16	-0.67	0.13	0.14	-0.50	-0.64	0.28	0.28	0.29	0.30
AUDFEES5	-0.61	-0.62	-0.17	-0.68	0.11	0.12	-0.50	-0.65	0.22	0.22	0.23	0.23
CASHNEAR1	-0.21	-0.19	-0.08	-0.29	0.47	0.46	-0.14	-0.11	0.85	0.83	0.82	0.81
CASHNEAR2	-0.21	-0.19	-0.08	-0.30	0.43	0.43	-0.12	-0.12	0.92	0.92	0.91	0.89
CASHNEAR4	-0.25	-0.24	-0.10	-0.40	0.35	0.35	-0.15	0.15	0.75	0.74	0.76	0.76
CASHNEAR5	-0.22	-0.22	-0.11	-0.39	0.28	0.28	-0.14	-0.14	0.62	0.61	0.64	0.64
CEFFTAX4	0.03	0.04	-0.04	-0.08	-0.09	-0.09	-0.03	-0.03	-0.06	-0.06	-0.06	-0.06
CEFFTAX5	0.02	0.02	-0.02	-0.10	-0.11	-0.12	-0.01	-0.02	-0.03	-0.02	-0.03	-0.03
CURASSET2	-0.19	-0.15	-0.04	-0.21	0.29	0.28	-0.15	-0.15	0.72	0.76	0.72	0.69
CURASSET3	-0.33	-0.29	-0.08	-0.37	0.29	0.28	-0.24	-0.29	0.74	0.79	0.75	0.72
CURASSET4	-0.31	-0.27	-0.08	-0.35	0.28	0.27	-0.22	-0.26	0.72	0.76	0.72	0.69
CURASSET5	-0.53	-0.51	-0.16	-0.63	0.29	0.28	-0.42	-0.51	0.69	0.71	0.71	0.69
CURLIAB1	-0.15	-0.12	-0.04	-0.18	0.44	0.43	-0.05	-0.05	1.00	0.99	0.99	0.99
CURLIAB2	-0.14	-0.12	-0.04	-0.18	0.43	0.42	-0.04	-0.04	0.99	1.00	0.99	0.99
CURLIAB3	-0.18	-0.16	-0.05	-0.22	0.42	0.41	-0.06	-0.08	0.98	0.98	0.99	0.99
CURLIAB4	-0.19	-0.16	-0.05	-0.22	0.43	0.42	-0.06	-0.08	0.9[illegible]	0.98	0.99	0.99
CURLIAB5	-0.20	-0.19	-0.07	-0.26	0.38	0.38	-0.08	-0.10	0.89	0.87	0.90	0.92
Ncurrat1	-0.12	-0.12	-0.03	-0.08	0.14	0.14	-0.10	-0.06	0.12	0.12	0.12	0.12
Ncurret3	-0.10	-0.10	-0.02	-0.05	0.15	0.15	-0.04	-0.04	0.11	0.10	0.11	0.11
DEBTASS1	0.03	0.05	0.10	0.06	0.52	0.52	-0.10	-0.06	0.24	0.23	0.24	0.24
DEBTASS2	0.03	0.04	0.10	0.06	0.58	0.58	-0.08	-0.07	0.24	0.24	0.24	0.24
DEBTASS3	0.02	0.03	0.04	0.05	0.64	0.62	-0.05	-0.07	0.25	0.24	0.25	0.25
DEBTASS4	0.02	0.03	0.03	0.04	0.67	0.64	-0.05	-0.07	0.26	0.25	0.26	0.26
DEBTASS5	0.05	0.06	0.04	0.05	0.62	0.68	-0.03	-0.05	0.25	0.24	0.25	0.25
Ndeceam4	0.19	0.22	0.86	0.28	0.03	0.05	0.17	0.16	-0.05	-0.04	-0.04	-0.05
Ndeceam5	0.19	0.22	0.81	0.30	0.04	0.08	0.17	0.16	-0.03	-0.02	-0.03	-0.03
Ndeftax5	0.68	0.64	0.27	0.56	-0.10	-0.08	0.49	0.53	-0.23	-0.22	-0.23	-0.23
DIREMOL3	-0.43	-0.44	-0.15	-0.64	0.09	0.08	-0.48	-0.57	0.13	0.14	0.14	0.15
DIREMOL4	-0.43	-0.45	-0.20	-0.66	0.08	0.07	-0.51	-0.57	0.12	0.13	0.13	0.13
DIREMOL5	-0.42	-0.44	-0.15	-0.61	0.09	0.05	-0.47	-0.53	0.06	0.06	0.07	0.07
DIVCOVR2	0.04	0.04	-0.06	-0.01	0.12	0.13	-0.03	-0.01	0.05	0.05	0.05	0.05
DIVCOVR3	0.04	0.04	-0.17	0.01	0.11	0.11	-0.03	-0.02	0.04	0.03	0.04	0.04
DIVCOVR4	-0.01	-0.01	-0.17	-0.04	0.08	0.10	-0.06	-0.05	0.03	0.03	0.03	0.03
DIVCOVR5	0.00	0.01	-0.03	-0.02	-0.01	-0.02	-0.03	-0.02	0.01	0.01	0.01	0.01
Ndivyldp1	-0.07	-0.07	-0.02	-0.10	0.01	0.00	-0.09	-0.09	0.01	0.01	0.01	0.01
Ndivyldp2	-0.10	-0.10	-0.08	-0.13	-0.05	-0.05	-0.09	-0.09	0.04	0.03	0.03	0.04
Ndivyldp3	-0.09	-0.09	-0.11	-0.15	-0.01	-0.02	-0.12	-0.12	0.06	0.06	0.06	0.06
Ndivpaid1	0.59	0.56	0.18	0.74	0.03	0.03	0.53	0.43	-0.23	-0.22	-0.22	-0.22
Ndivpaid2	0.64	0.63	0.27	0.82	0.01	0.02	0.54	0.55	-0.22	-0.22	-0.22	-0.22

Correlation Matrice between the 66 most relevant Variables

	Nfixasset4	Nfixasset5	Nfundshr5	Ngrincm5	INTBERDA4	INTBERDA5	Nint_fin1	Nint_fin3	INTBRCL1	INTBRCL3	INTBRCL2	INTBRCL4
Ndivpaid3	0.60	0.58	0.17	0.85	0.03	0.04	0.54	0.53	-0.17	-0.18	-0.16	-0.13
Ndivpaid4	0.57	0.54	0.17	0.80	0.05	0.06	0.54	0.50	-0.14	-0.14	-0.15	-0.15
Ndivpaid5	0.66	0.65	0.21	0.86	0.06	0.08	0.55	0.56	-0.10	-0.10	-0.11	-0.11
Nemyldp2	-0.06	-0.06	0.00	-0.07	-0.13	-0.10	-0.02	-0.03	0.01	0.00	0.00	0.01
Nfixesset2	0.89	0.86	0.21	0.70	-0.05	-0.02	0.76	0.72	-0.13	-0.13	-0.13	-0.13
Nfixasset3	0.99	0.97	0.26	0.81	-0.05	-0.02	0.74	0.82	-0.10	-0.10	-0.10	-0.10
Nfixasset4	1.00	0.98	0.26	0.80	-0.04	-0.01	0.72	0.80	-0.09	-0.09	-0.10	-0.09
Nfixasset5	0.98	1.00	0.32	0.80	-0.03	-0.01	0.68	0.78	-0.07	-0.07	-0.08	-0.08
Nfundshr5	0.26	0.32	1.00	0.29	0.01	0.04	0.15	0.17	-0.03	-0.02	-0.03	-0.03
Ngrincr..	0.80	0.80	0.29	1.00	0.00	0.02	0.71	0.79	-0.12	-0.12	-0.13	-0.13
INTBERDA4	-0.04	-0.03	0.01	0.00	1.00	0.91	-0.05	-0.08	0.44	0.43	0.43	0.43
INTBERDA5	-0.01	-0.01	0.04	0.02	0.91	1.00	-0.04	-0.05	0.43	0.41	0.42	0.42
Nint_fin1	0.72	0.68	0.15	0.71	-0.05	-0.04	1.00	0.87	0.02	0.02	0.02	0.02
Nint_fin3	0.80	0.78	0.17	0.79	-0.08	-0.05	0.87	1.00	0.02	0.01	0.01	0.01
INTBRCL1	-0.09	-0.07	-0.03	-0.12	0.44	0.43	0.02	0.02	1.00	1.00	0.99	0.99
INTBRCL3	-0.09	-0.07	-0.02	-0.12	0.43	0.41	0.02	0.01	1.00	1.00	1.00	0.99
INTBRCL2	-0.09	-0.08	-0.03	-0.13	0.43	0.42	0.02	0.01	0.99	0.99	1.00	1.00
INTBRCL4	-0.10	-0.08	-0.03	-0.13	0.43	0.42	0.02	0.01	0.99	1.00	1.00	1.00
INTBRCL5	-0.10	-0.09	-0.04	-0.15	0.39	0.39	0.01	0.00	0.91	0.89	0.92	0.94
Nintcovr1	0.09	0.11	0.06	0.01	0.11	0.11	-0.06	-0.05	0.02	0.02	0.02	0.02
Nintcovr2	0.02	0.03	0.04	-0.01	0.10	0.11	-0.06	-0.05	0.02	0.02	0.02	0.02
Nintcovr5	0.02	0.01	0.00	-0.02	0.01	0.10	-0.01	-0.02	0.02	0.01	0.01	0.02
Nintrec5	0.52	0.53	0.19	0.61	0.06	0.08	0.39	0.43	0.03	0.02	0.02	0.02
NMayprice	0.25	0.26	0.52	0.36	0.04	0.06	0.20	0.16	-0.08	-0.07	-0.07	-0.08

Correlation Matrice between the 66 most relevant Variables

	INTBRCL5	Nintcovr1	Nintcovr2	Nintcovr5	Nintrec5	NMayprice
return	0.07	0.06	0.06	0.11	0.06	0.10
AUDFEES3	0.52	0.05	0.05	0.03	-0.31	-0.20
AUDFEES4	0.33	0.05	0.05	0.04	-0.36	-0.17
AUDFEES5	0.27	0.05	0.05	0.04	-0.38	-0.16
CASHNEAR1	0.76	0.02	0.03	0.03	-0.07	-0.19
CASHNEAR2	0.82	0.02	0.02	0.03	-0.09	-0.14
CASHNEAR4	0.80	0.01	0.02	0.03	-0.21	-0.22
CASHNEAR5	0.74	0.01	0.02	0.02	-0.25	-0.20
CEFFTAX4	-0.05	-0.01	-0.03	0.06	-0.10	-0.01
CEFFTAX5	-0.03	0.03	-0.04	0.04	-0.12	-0.04
CURASSET2	0.41	0.02	0.02	0.02	-0.09	-0.07
CURASSET3	0.49	0.03	0.03	0.02	-0.19	-0.10
CURASSET4	0.45	0.02	0.03	0.02	-0.20	-0.10
CURASSET5	0.61	0.03	0.04	0.03	-0.40	-0.20
CURLIAB1	0.92	0.02	0.02	0.02	-0.01	-0.11
CURLIAB2	0.90	0.02	0.02	0.02	-0.01	-0.09
CURLIAB3	0.94	0.02	0.02	0.02	-0.04	-0.10
CURLIAB4	0.93	0.02	0.02	0.02	-0.04	-0.10
CURLIAB5	0.99	0.02	0.02	0.02	-0.06	-0.13
Ncurrat1	0.12	-0.02	0.11	0.04	-0.03	0.03
Ncurrat3	0.11	0.06	0.00	-0.06	0.00	0.01
DEBTASS1	0.22	0.14	0.15	0.10	0.09	0.13
DEBTASS2	0.22	0.16	0.13	0.10	0.09	0.15
DEBTASS3	0.23	0.17	0.11	0.11	0.09	0.13
DEBTASS4	0.24	0.15	0.13	0.07	0.09	0.12
DEBTASS5	0.24	0.15	0.12	0.11	0.11	0.12
Ndecearn4	-0.06	0.03	0.01	-0.01	0.21	0.63
Ndecearn5	-0.04	0.06	0.02	0.01	0.18	0.64
Ndeftax5	-0.28	-0.04	-0.02	-0.02	0.34	0.22
DIREMOL3	0.15	0.09	0.09	0.07	-0.27	-0.20
DIREMOL4	0.12	0.09	0.09	0.07	-0.24	-0.22
DIREMOL5	0.09	0.08	0.08	0.06	-0.22	-0.17
DIVCOVR2	0.05	0.05	0.07	0.10	-0.02	0.02
DIVCOVR3	0.04	0.03	0.05	0.03	-0.05	-0.02
DIVCOVR4	0.03	0.01	0.05	0.07	-0.06	-0.04
DIVCOVR5	0.01	0.02	0.03	0.01	-0.02	0.02
Ndivyldp1	0.02	0.05	0.01	-0.03	-0.06	-0.12
Ndivyldp2	0.04	-0.02	0.12	-0.04	-0.08	-0.14
Ndivyldp3	0.06	-0.01	0.16	-0.04	-0.10	-0.18
Ndivpaid1	-0.22	0.08	0.00	-0.02	0.46	0.43
Ndivpaid2	-0.21	0.05	0.00	-0.03	0.56	0.40

Correlation Matrice between the 66 most relevant Variables

	INTBRCL5	Nintcovr1	Nintcovr2	Nintcovr5	Nintrec5	NMayprice
Ndivpaid3	-0.18	0.02	-0.02	-0.02	0.51	0.40
Ndivpaid4	-0.15	0.03	-0.02	-0.02	0.53	0.43
Ndivpaid5	-0.11	0.04	0.00	0.00	0.57	0.42
Nemyldp2	0.01	-0.03	0.01	-0.06	-0.03	-0.08
Nfixasset2	-0.13	0.09	0.02	0.01	0.44	0.27
Nfixasset3	-0.11	0.08	0.02	0.02	0.52	0.25
Nfixasset4	-0.10	0.09	0.02	0.02	0.52	0.25
Nfixasset5	-0.09	0.11	0.03	0.01	0.53	0.26
Nfundshr5	-0.04	0.06	0.04	0.00	0.19	0.52
Ngrincm5	-0.15	0.01	-0.01	-0.02	0.61	0.36
INTBERDA4	0.39	0.11	0.10	0.01	0.06	0.04
INTBERDA5	0.39	0.11	0.11	0.10	0.08	0.06
Nint_fin1	0.01	-0.06	-0.06	-0.01	0.39	0.20
Nint_fin3	0.00	-0.05	-0.05	-0.02	0.43	0.16
INTBRCL1	0.91	0.02	0.02	0.02	0.03	-0.08
INTBRCL3	0.89	0.02	0.02	0.01	0.02	-0.07
INTBRCL2	0.94	0.02	0.02	0.02	0.02	-0.08
INTBRCL4	0.92	0.02	0.02	0.01	0.02	-0.07
INTBRCL5	1.00	0.02	0.02	0.02	0.01	-0.09
Nintcovr1	0.02	1.00	0.25	0.00	0.02	0.08
Nintcovr2	0.02	0.25	1.00	0.06	-0.02	0.01
Nintcovr5	0.02	0.00	0.06	1.00	-0.01	0.00
Nintrec5	0.01	0.02	-0.02	-0.01	1.00	0.20
NMayprice	-0.09	0.08	0.01	0.00	0.20	1.00

Appendix B:

Explanation of variable names and the Prior probabilities

Variable	Expanded name	CLASS 1	2	3	4
Return	Nov. Price divided by the May Price.	0.28	0.25	0.24	0.22
AUDFEES3	Audit Fees	0.00	0.81	0.12	0.08
AUDFEES4	Audit Fees	0.00	0.80	0.12	0.09
AUDFEES5	Audit Fees	0.00	0.79	0.12	0.08
CASHNEAR1	Cash and Near Cash	0.00	0.85	0.09	0.06
CASHNEAR2	Cash and Near Cash	0.00	0.86	0.08	0.06
CASHNEAR4	Cash and Near Cash	0.00	0.85	0.08	0.07
CASHNEAR5	Cash and Near Cash	0.00	0.85	0.09	0.05
CEFFTAX4	Effective tax rate %	0.35	0.13	0.19	0.33
CEFFTAX5	Effective tax rate %	0.34	0.15	0.20	0.32
CURASSET2	Current Assets	0.00	0.84	0.12	0.04
CURASSET3	Current Assets	0.00	0.83	0.11	0.06
CURASSET4	Current Assets	0.00	0.84	0.10	0.06
CURASSET5	Current Assets	0.00	0.82	0.10	0.08
CURLIAB1	Current Liabilities	0.00	0.91	0.07	0.03
CURLIAB2	Current Liabilities	0.00	0.90	0.07	0.03
CURLIAB3	Current Liabilities	0.00	0.90	0.08	0.02
CURLIAB4	Current Liabilities	0.00	0.90	0.08	0.02
CURLIAB5	Current Liabilities	0.00	0.88	0.10	0.02
Ncurrat1	Current ratio.	0.14	0.33	0.48	0.05
Ncurrat3	Current ratio.	0.12	0.20	0.56	0.12
DEBTASS1	Debt/Asset ratio	0.32	0.19	0.23	0.26
DEBTASS2	Debt/Asset ratio	0.32	0.19	0.22	0.28
DEBTASS3	Debt/Asset ratio	0.33	0.15	0.26	0.26
DEBTASS4	Debt/Asset ratio	0.33	0.16	0.23	0.28
DEBTASS5	Debt/Asset ratio	0.31	0.18	0.26	0.26
Ndecearn4	Declared Earnings per share	0.10	0.17	0.72	0.00
Ndecearn5	Declared Earnings per share	0.09	0.16	0.74	0.01
Ndeftax5	Defered tax	0.06	0.08	0.86	0.00
DIREMOL3	Directors Emoluments	0.29	0.38	0.16	0.17
DIREMOL4	Directors Emoluments	0.31	0.37	0.13	0.19
DIREMOL5	Directors Emoluments	0.21	0.48	0.17	0.15
DIVCOVR2	Dividend Cover	0.25	0.26	0.33	0.16
DIVCOVR3	Dividend Cover	0.28	0.25	0.31	0.16
DIVCOVR4	Dividend Cover	0.20	0.31	0.36	0.13
DIVCOVR5	Dividend Cover	0.00	0.60	0.37	0.03
Ndivyldp1	Dividend yield %	0.15	0.17	0.53	0.15
Ndivyldp2	Dividend yield %	0.15	0.19	0.53	0.13
Ndivyldp3	Dividend yield %	0.17	0.19	0.40	0.24
Ndivpaid1	Dividends Paid	0.13	0.11	0.76	0.00
Ndivpaid2	Dividends Paid	0.13	0.09	0.78	0.00

Explanation of variable names and the Prior probabilities

Variable	Expanded name	CLASS 1	CLASS 2	CLASS 3	CLASS 4
Ndivpaid3	Dividends Paid	0.12	0.10	0.78	0.00
Ndivpaid4	Dividends Paid	0.12	0.11	0.77	0.00
Ndivpaid5	Dividends Paid	0.12	0.10	0.78	0.00
Nernyldp2	Earnings Yield %	0.09	0.27	0.62	0.03
Nfixasset1	Fixed Assets	0.13	0.08	0.79	0.00
Nfixasset2	Fixed Assets	0.12	0.07	0.81	0.00
Nfixasset3	Fixed Assets	0.12	0.07	0.81	0.00
Nfixasset4	Fixed Assets	0.11	0.08	0.81	0.00
Nfixasset5	Fixed Assets	0.12	0.07	0.81	0.00
Nfundshr5	Fund flow per share(cents)	0.11	0.14	0.74	0.01
Ngrincm5	Gross Income	0.13	0.10	0.78	0.00
INTBERDA4	Int. Bear. Debt/Ass Ratio	0.34	0.30	0.18	0.17
INTBERDA5	Int. Bear. Debt/Ass Ratio	0.35	0.28	0.21	0.16
Nint_fin1	Interest and Finance charges	0.11	0.09	0.80	0.00
Nint_fin3	Interest and Finance charges	0.09	0.12	0.79	0.00
INTBRCL1	Interest Bearing-Current Liabilities	0.00	0.95	0.03	0.02
INTBRCL2	Interest Bearing-Current Liabilities	0.00	0.96	0.03	0.02
INTBRCL3	Interest Bearing-Current Liabilities	0.00	0.96	0.03	0.01
INTBRCL4	Interest Bearing-Current Liabilities	0.00	0.96	0.03	0.01
INTBRCL5	Interest Bearing-Current Liabilities	0.00	0.95	0.04	0.01
Nintcovr1	Interest Cover	0.02	0.03	0.94	0.00
Nintcovr2	Interest Cover	0.03	0.02	0.94	0.00
Nintcovr5	Interest Cover	0.03	0.03	0.93	0.00
Nintrec5	Interest Received	0.09	0.07	0.84	0.00
NMayprice	End of May price	0.12	0.15	0.73	0.00

Notes:

* A capital N before a variable indicates that the variable has been inverted to have a positive correlation to return.

** The number at the end of a variable indicates how many financial years ago.