



UNIVERSITY OF PAVIA

Department of Economics and Management

Doctoral Program in Economics and Management of Technology
(DREAMT) – XXIX Cycle

Bayesian Networks Models for Equity Market

Supervisor:

Prof. Maria Elena De Giuli

PhD Dissertation of:

Alessandro Greppi

Contents

Introduction	2
1 Graphical Models	4
1.1 What is a Graph	4
1.1.1 Graphs Recurring Terminology	6
1.2 Directed Acyclic Graphs	7
1.2.1 The Moral Graphs	9
1.2.2 The Wermuth Condition	9
1.3 Conditional Independence	10
1.4 Conditional Independence Graphs	12
1.5 How Nodes Are Connected	13
1.5.1 Serial Connection	13
1.5.2 Diverging Connection	14
1.5.3 Converging Connection	14
1.6 How to Read Independences in a Graph	15
1.6.1 Markov Properties and Undirected Graphs	15
1.6.2 Markov Properties and Directed Acyclic Graphs	17
2 Bayesian Networks and Object Oriented Bayesian Networks	19
2.1 The Expert and the Machine	19
2.2 Dealing with Uncertainty	20
2.3 Message Propagation in the Network	20
2.3.1 Oil Stocks	21
2.3.2 Iron Price and Dividend	22
2.3.3 Why SYN is Up by 8%?	23
2.4 Bayesian Networks	24
2.4.1 Conditional Probabilities	24
2.5 Calculating probabilities	25
2.6 Probabilities and Bayesian Networks	26
2.7 Revisiting the Previous Examples	27
2.7.1 Oil Stocks Example Revisited	27
2.7.2 Iron Price and Dividend Example Revisited	28
2.8 Object Oriented Bayesian Networks	29
2.8.1 Limits of the BNs approach	30
2.9 The OOBN Basic Elements	30
2.10 Features and Elements of the Class	33
2.10.1 Links	34
2.11 Subclasses and Inheritance	34

2.12	Modularity	35
2.13	Making Inference in an OOBN	36
3	Bayesian Networks for Financial Markets Signals	37
3.1	Introduction	37
3.2	Data Description	39
3.2.1	The Variables of the Model	39
3.3	The S&P 500 Bayesian Network	41
3.4	Simulating Market Evolution for the Period 1994-2003	43
3.4.1	Scenario A: The Effects of High/Low Volatility Between 1994 and 2003	43
3.4.2	Scenario B: The Effects of High/Low Price to Earnings Ratio Between 1994 and 2003	44
3.5	Simulating Market Evolution for the Period 2004-2015	48
3.6	Examination of Different Scenarios (2004-2015)	51
3.6.1	Scenario C: The Effects of High/Low Volatility Between 2004 and 2015	51
3.6.2	Scenario D: The Effects of High/Low Price to Earnings Ratio Between 2004 and 2015	53
3.7	Concluding Remarks	56
4	OOBNs for S&P 500	58
4.1	An Application to the S&P 500 Index	58
4.2	The Variables of the OOBN	59
4.3	The OOBN for Market Signals Detection	61
4.4	Macro, Micro, Technical Analysis and Market Sentiment Classes	61
4.4.1	Market Sentiment Area	61
4.4.2	Technical Analysis Area	65
4.4.3	Macroeconomics Area	72
4.4.4	Microeconomic Dimension Area	78
4.5	OOBN What-if Analysis Simulations	92
4.6	What Practitioners Actually Do and Why This Model Is Innovative	109
	Conclusions and Future Extensions	111
	Appendix 1: Introduction to the Hugin Software	113
4.7	Brief History of Hugin	113
4.8	How to Build a BN with Hugin	113
4.8.1	Adding a New Node	113
4.8.2	Adding Directed Arrows Among Nodes	114
4.8.3	Specifying the States Associated to Each Node	115
4.8.4	Entering Values into a CPT	115
4.9	Structural Learning	117
4.9.1	Constraint Based Algorithms	117
4.9.2	Search and Score Based Algorithms	118
4.9.3	Restricted Models	120
4.10	Other Relevant Algorithms	120
4.10.1	The Learning Wizard	121
4.11	How to Learn the BN Directly from the Data	122

Appendix 2: How to Build an OOBN with the Hugin Software	126
4.12 Dynamic Bayesian Networks	126
4.12.1 Building an OOBN Starting from a Static BN	126
4.13 Building the OOBNs Elements	127
4.13.1 Output Nodes	127
4.13.2 Input Nodes	128
4.13.3 Creating an Interface Node	128
4.13.4 OOBN for the Biotech Company Example	128
4.14 Running the OOBN	130
4.15 Learning the OOBN from a Dataset	131
4.15.1 The Financial Markets Example	131
Bibliography	135

List of Figures

1.1	An example of Directed Graph	5
1.2	An Example of Bidirected Graph	5
1.3	An Example of Undirected Graph	5
1.4	An Example of Undirected Graph	7
1.5	A Directed Graph with an Edge from the Node “a” to “b”	8
1.6	A Directed Graph with an Edge from the Node “b” to “a”	8
1.7	A Directed Cyclic Graph with Three Nodes	8
1.8	A Directed Acyclic Graph with Two Parents and a Child	9
1.9	A Directed Acyclic Graph Before the Moralization	9
1.10	The Moralized Graph	9
1.11	A Simple Directed Graph with $V = 4$	10
1.12	A Simple Undirected Graph with $V = 4$	10
1.13	A Directed Graph with a Forbidden Configuration	10
1.14	An Undirected Graph with a Forbidden Configuration	11
1.15	An Example of Graph	12
1.16	An Example of Undirected Graph with Four Vertices	12
1.17	An Independence Graph with Three Separating Subset	13
1.18	A Serial Connection	13
1.19	A Diverging Connection	14
1.20	A Converging Connection	14
1.21	Example for Markov Properties and Undirected Graphs	15
1.22	An Undirected Graph with $V=\{1, 2, 3, 4\}$	16
1.23	An Undirected Graph with a Separating Subset	16
1.24	Example for Markov Properties and Directed Graphs	17
2.1	A Network Model of <i>Low Oil Price</i>	21
2.2	A Network Model of <i>Iron Price and Dividend</i>	22
2.3	A Network Model of <i>SYN Stocks Up by 8%</i>	23
2.4	Probabilities and the Steps that Lead us to a Conclusion	29
2.5	The Yearly Pattern for Dividend Payments and Stock Reactions	30
2.6	The Yearly Pattern for Dividend Payments and Stock Reactions	31
2.7	Output (O) and Input (I) Nodes Representation	32
2.8	The AAPL dividend OOBN with the Encapsulated Nodes Hidden	33
2.9	The network class for each oil stock	35
2.10	The oil stock portfolio OOBN	36
3.1	The 1994-2003 S&P 500 Network	42
3.2	The Effects Referred to High Volatility 1994-2003	45
3.3	The Effects Referred to Low Volatility 1994-2003	46

3.4	The Effects Referred to High Price to Earnings 1994-2003	47
3.5	The Effects Referred to Low Price to Earnings 1994-2003	49
3.6	The 2004-2015 S&P 500 Network	50
3.7	The Effects Referred to High Volatility (2004-2015)	52
3.8	The Effects Referred to Low Volatility (2004-2015)	54
3.9	The Effects Referred to High Price to Earnings Ratio 2004-2015	55
3.10	The Effects Referred to Low Price to Earnings Ratio 2004-2015	57
4.1	The OOBN for S&P 500 signals detection	62
4.2	The Market Sentiment OOBN	63
4.3	Market Sentiment OOBN: High VIX Scenario	63
4.4	Market Sentiment OOBN: Low VIX Scenario	64
4.5	Market Sentiment OOBN: High Put Call Ratio Scenario	64
4.6	Market Sentiment OOBN: Low Put Call Ratio Scenario	65
4.7	Market Sentiment OOBN: Low Volatility Spread Scenario	65
4.8	Market Sentiment OOBN: High Volatility Spread Scenario	66
4.9	The Technical Analysis OOBN	66
4.10	Technical Analysis OOBN: Bullish ROC Scenario	67
4.11	Technical Analysis OOBN: Bearish ROC Scenario	68
4.12	Technical Analysis OOBN: “Golden Cross” Scenario	68
4.13	Technical Analysis OOBN: “Death Cross” Scenario	69
4.14	Technical Analysis OOBN: Overbough RSI Scenario	70
4.15	Technical Analysis OOBN: Oversold RSI Scenario	70
4.16	Technical Analysis OOBN: The Effects of High Vix	71
4.17	Technical Analysis OOBN: The Effects of Low Vix	71
4.18	The Macroeconomics OOBN	72
4.19	Macroeconomics OOBN: High GDP Growth Scenario	73
4.20	Macroeconomics OOBN: Low GDP Growth Scenario	73
4.21	Macroeconomics OOBN: High Unemployment Scenario	74
4.22	Macroeconomics OOBN: Low Unemployment Scenario	75
4.23	Macroeconomics OOBN: High Gold Price Scenario	76
4.24	Macroeconomics OOBN: Low Gold Price Scenario	76
4.25	Macroeconomics OOBN: Strong Dollar Scenario	77
4.26	Macroeconomics OOBN: Weak Dollar Scenario	78
4.27	The Micro Dimension OOBN	79
4.28	Micro Dimension OOBN: High EPS Growth	81
4.29	Micro Dimension OOBN: Low EPS Growth	82
4.30	Micro Dimension OOBN: High ROE	83
4.31	Micro Dimension OOBN: Low ROE	84
4.32	Micro Dimension OOBN: High Ebitda Margin	85
4.33	Micro Dimension OOBN: Low Ebitda Margin	87
4.34	Micro Dimension OOBN: High EV/Ebitda	88
4.35	Micro Dimension OOBN: Low EV/Ebitda	89
4.36	Micro Dimension OOBN: The Influence of High GDP Growth	90
4.37	Micro Dimension OOBN: The Influence of Low GDP Growth	91
4.38	Micro Dimension OOBN: The Influence of the “Death Cross”	93
4.39	Micro Dimension OOBN: The Influence of the “Golden Cross”	94
4.40	Micro Dimension OOBN: High Price to Earnings Ratio Scenario	95
4.41	Micro Dimension OOBN: Low Price to Earnings Ratio Scenario	96
4.42	Scenario A1: Dxy_Index, Pe_Ratio and Rsi are equal to 1	98

4.43	Scenario A2: Pe_Ratio and Rsi equal to 1; Dxy_Index equal to 2	99
4.44	Scenario B1: Dxy_Index, Vix, Rsi and Pe_Ratio equal to 1	100
4.45	Scenario B2: Dxy_Index, Vix and the Rsi equal to 1 and the Pe_Ratio equal to 2	101
4.46	Scenario C1: Vola_Spread and RSI equal to 1; USA_Gdp equal to 2 . . .	103
4.47	Scenario C2: Vola_Spread equal to 1; USA_Gdp and RSI equal to 2 . . .	104
4.48	The Priors for Pe_Ratio, Dxy_Index, Rsi and Vola_Spread	105
4.49	Scenario D1: The Contrarian Buy Scenario	106
4.50	Scenario D2: The Contrarian Sell Scenario	107
4.51	Scenario E1: The Effect of High Gold Price on the Target Variable . . .	108
4.52	Scenario E2: The Effect of Low Gold Price on the Target Variable . . .	108
4.53	An Example of Practitioners' Tool: Bridgewater's Growth-Inflation Matrix	109
4.54	A Screenshot of Hugin's Network Window	114
4.55	Hugin's Network Toolbar. Evidenced by the Squares, from the Left to the Right: the Node Properties Tool, the Discrete Chance Tool and the Arrow Tool	114
4.56	The Nodes Belonging to the BN	115
4.57	The Arrow Tool Button	115
4.58	A Qualitative Representation of the BN	116
4.59	The Add/Remove States Tool	116
4.60	P(Bond_Emission)	116
4.61	P(Capital_Increase)	117
4.62	P(Stocks_Going_Down, Bond_Emission, Capital_Increase)	117
4.63	The Results of the Simulation	118
4.64	How to Open the Learning Wizard	122
4.65	Loading Files with the Learning Wizard	123
4.66	Imposing Structure Constraints	123
4.67	Structural Learning Algorithm Selection	124
4.68	The Snapshot of the BN	125
4.69	The Network Pane with our BN	125
4.70	The Initial Static BN	127
4.71	A DBN with Two Time Slices	127
4.72	An Interface Node for the Biotech Company Example	128
4.73	The Instance Node Tool	129
4.74	Selecting the Instance Node Through the Toolbar	129
4.75	Three Connected Time Slices	130
4.76	The Collapsed Instance Nodes	130
4.77	The Results of Our Toy Example	130
4.78	The Financial Markets Instance Nodes	131
4.79	The BN Used for Learning the CPT	132
4.80	Uploading the Instance Nodes	132
4.81	Importing the CPT Table Referred to B_S_SPX	132
4.82	The CPT referred to B_S_SPX (VOLA, EARN_GR)	133
4.83	The OOBN Leaned from Financial Data	134

List of Tables

2.1	An example of $P(A B)$. All columns sum to 1.	25
2.2	An example of $P(A, B)$. All the entries sum to 1.	25
2.3	Conditional Probabilities for $P(PD = ED OD)$	27
2.4	Joint Probability Table for $P(PD = ED, OD)$	27
2.5	Calculating the Updated $P(ED)$	28
2.6	The CPT for $P(Opec_Fre_Prod Sup_Red)$	35

Introduction

There is no common agreement on the procedures that should be followed by investors when they take buy or sell decisions. However, traders can learn and test different approaches that allow them to improve their understanding of price dynamics. However, the process of returns generation is complicated by the fact that financial markets evolve quickly due to the continuous innovation of investment instruments. In this challenging framework, the ability of taking an efficient decision in a short time can influence the overall performance of the investment.

For this reason, we propose in this work an innovative approach that exploits graphical models in order to provide buy or sell indications on the most capitalized equity market in the world: the S&P500. Adopting a model allows us to deal with a complex framework by generating a reliable approximation of the real world. We decided to analyze the S&P500 because its dynamics are influenced by a large amount of variables whose interpretation represents a challenging task for practitioners. Generally, an investor observes the market and then he makes a decision but procedure is generally time consuming. This is why computers and algorithms are spreading in the last years through financial industry with the objective of supporting fund managers and strategists. The aim of this work is to build a model that is able to perform in a mouse-click simulations on alternative market scenarios by exploiting algorithms potential.

In order to do that, we use graphical models: Bayesian Networks (BNs) and their extension called Object Oriented Bayesian Networks (OOBNs). The attribute “graphical” means that they can both be represented by a graph, a feature that makes complex frameworks easier to interpret. Furthermore, thanks to Hugin, a software that has been designed to deal exclusively with these models, we can exploit the potentiality of some algorithms that allow us to learn directly from the data the network structure and to define prior probabilities. Thanks to these features, it is possible to observe known or unexpected dependence/independence relations among the variables and to simulate the impact of new information across the network.

In summary, we have chosen BNs and OOBNs because graphical models allow showing clearly and intuitively dependence and independence relations. Moreover, they deal efficiently with uncertain situations by exploiting some of the most established probability theories, such as the Bayes’ Rule. Furthermore, by using the Hugin software we can exploit the algorithms implemented in it and learn directly from the data the network structure or simulate in real time different scenarios.

The dissertation is organized as follows:

- In Chapter 1 we present graphs, the theory at their basis and how we can read undirected or directed graphs. This leads us to the introduction of the Direct Acyclic Graphs (DAGs)
- In Chapter 2 we show how an expert system helps a researcher in dealing with reasoning under uncertainty situations. Then, thanks to the support of some toy examples, we introduce BNs, their features and limits. In conclusion, we present OOBNs as a powerful tool that allows us to deal with complex frameworks and overcome some of BNs weaknesses.
- In Chapter 3 we adopt BNs for detecting S&P 500 buy or sell signals. Thanks to our experiment, we demonstrate that including in the same model variables that are generally observed separately provides useful indications that are missed by the tools used by fund managers.
- In Chapter 4 we extend our analysis on the American equity market by introducing the OOBNs. This approach makes easier to read a complex framework and to interpret the results of the simulations.
- In conclusion, Appendix 1 and Appendix 2 contain an extensive guide on the Hugin software that shows step-by-step how to build a BNs and an OOBN.

Chapter 1

Graphical Models

1.1 What is a Graph

Graphical models are multivariate statistical models whose independence structure is represented by a graph. They can be easily interpreted by exploiting the Markov properties, see Sections 1.6 and 1.7 and they can improve the communication between statisticians and researchers. Moreover, graphical models can be subdivided in different modules, making complex problems easier to understand.

In the real world, we can often describe a particular scenario by using a diagram made of vertices and links. We now suppose that an observer wants to understand whether two variables are related. Thanks to a graph, we can represent any relation, which consists in a multivariate statistical model that incorporates independence constraints. Each variable corresponds to the vertices of the graph, while the presence or the absence of an edge connecting two of them provides an important indication on any relation of dependence/independence. More in detail, a graph G is a mathematical object composed by a set of vertices V and a set of edges E , so that $G = \{V, E\}$. The vertices $V = \{1 \dots |V|\}$ are a representation of random variables, while a set of edges E represents the dependencies among variables (Lauritzen, 1995). The origin of the term *graph* descends from the fact that this particular representation allows to represent graphically a problem and its properties. Circles, points or ellipses identify the vertices while the edges can be of different types:

- Directed (i, j) which consists in an arrow $i \longrightarrow j$, see Figure 1.1
- Bidirected (i, j) which consists in an bi-directional arrow $i \longleftrightarrow j$, see Figure 1.2
- Undirected $\{i, j\}$ which consists in a line $i - j$, see Figure 1.3

If a graph has only directed edges it is called *directed* graph, while if it has only undirected edges it is an *undirected* graph. As we can observe, we can obtain the undirected graph in Figure 1.3 by replacing in the directed graph in Figure 1.1 the arrows with lines.

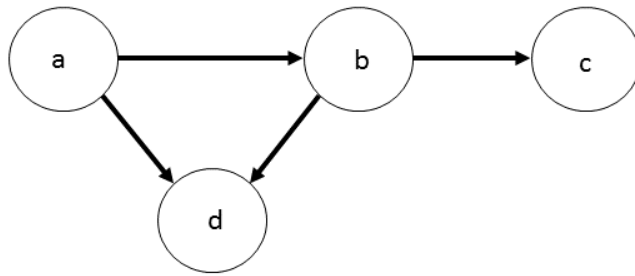


Figure 1.1: An example of Directed Graph

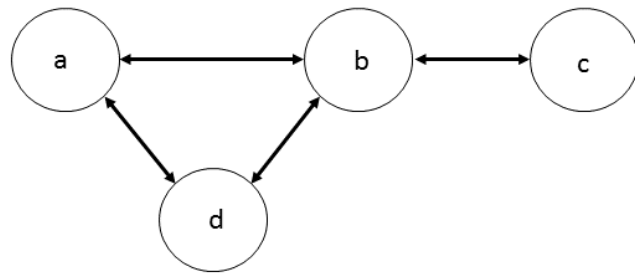


Figure 1.2: An Example of Bidirected Graph

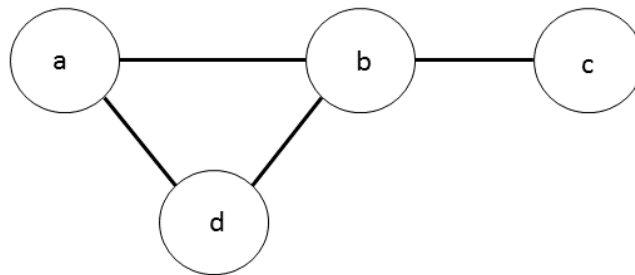


Figure 1.3: An Example of Undirected Graph

1.1.1 Graphs Recurring Terminology

Before introducing new concepts, we provide a brief definition of the most recurring terms referred to graphs theory:

- *Complete Graph and Complete Subset*: a graph is complete when an edge or a line connects all its vertices. In its turn, a subset is complete when it induces a complete subgraph.
- *Adjacent Vertices*: two vertices are *adjacent* if they have an edge in common.
- *Clique*: it represents a complete maximal subset of vertices (Lauritzen, 1996).
- *Parents and Child*: in the case we have an arrow pointing from the vertex a towards vertex b , we call a parent of b . In its turn, b is called child of a .
- *Neighbors*: a subset of vertices separates other two vertices i and j if all the paths connecting i and j include at least a vertex belonging to the subset. By assuming that an hypothetical subset $k \subseteq V$ represents a subset of vertices in V , we call neighbors the ones belonging to V , but not in k , that are adjacent to a vertex belonging to k . The boundary of k , called $bd(k)$, represents the parents and the neighbors of k .
- *Path*: it represents a sequence of different vertices from i_1 to i_m for whom each pair belonging to the set of edges E is connected to the following pair through a directed link. If no subsequence of the sequence is a path, we are dealing with a short path. In the case of undirected graph, every edge is undirected and all the vertices in a path are adjacent.
- *Connected Vertices*: two vertices i and j are connected if there is a path from i to j and from j to i .
- *Cycle*: it is the case in which a path ends in its starting point. This means that we have a cycle if the starting point i_1 is equal to the ending point i_m .
- *Chordless Cycle*: it represents a particular configuration in which only the successive pairs of vertices in the cycle are adjacent.
- *Directed Acyclic Graph (DAG)*: it represents a directed graph with node set V that does not allow directed cycles (acyclic). See further details on the DAGs in Section 1.2.
- *Skeleton*: it is the undirected graph obtained by substituting all the directed edges in a DAG with undirected ones.
- *Ancestral Set*: given two vertices a and b , we call a ancestor of b and b descendent of a if we have a descending path from a to b . We use the following notations to identify different possible situations: $an(a)$, ancestors of a ; $de(a)$, descendants of a ; $nd(a)$, $V \setminus (de(a) \cup a)$, non-descendants of a . Assuming that $A \subset V$, the ancestral set of A , $an(A)$, represents the set of all the ancestors of A . The subset A is called ancestral if and only if the following condition holds: $a \in A, an(a) \subseteq A$ (Lauritzen, 1995).
- *Tree*: it is a connected, acyclic, non-directed graph with a unique path between two vertices.

- *Node*: it represents a variable of the network. This word is often used interchangeably with the term vertex.
- *Link*: it represents the properties of conditional dependence/independence among the nodes belonging to the same network. This word is often used interchangeably with the term edge.

In Figure 1.4 we represent in the same graph some of the elements introduced in the previous list. As we can see, the graph is undirected and with a set $V = \{1, 2, 3, 4, 5, 6, 7\}$ and a set of $E = \{(2,3), (3,4), (4,7), (3,6), (2,6), (5,6), (6,7)\} \cup \{(3,2), (4,3), (7,4), (6,3), (6,2), (6,5), (7,6)\}$. In this particular example, we notice that many paths can lead us from 4 to 5, such as 4, 3, 6, 5. However, the graph in Figure 1.4 is not connected because there is no edge between the vertex 1 and the other vertices of the graph. Moreover, we can identify that the boundary of 4 is represented by $\{3,7\}$, while the one referred to $\{3,4\}$ is the set $\{2, 6, 7\}$. Furthermore, we can observe that the cycle 4, 7, 6, 3, 4 is chordless, while the cycle 4, 7, 6, 2, 3, 4 is not. In conclusion, we can identify the following cliques: $\{2, 3, 6\}$, $\{3,4\}$, $\{4,7\}$, $\{6,7\}$, $\{6,6\}$, $\{1\}$.

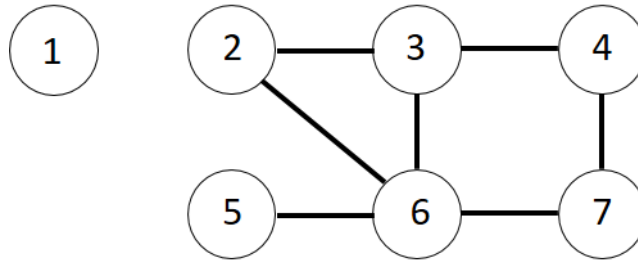


Figure 1.4: An Example of Undirected Graph

1.2 Directed Acyclic Graphs

We now analyze how the variables interact by observing the effects of a variable A on a variable B and their conditional probability density function, $f_{A|B}$. For example, we suppose that the company “Ford Motors” hired a top CEO, variable X_a , and at the same time its revenues increased in the last year, variable X_b . These two events can affect Ford Motors stock performance, variable S_F . Our aim is to understand if S_F depends on X_a , on X_b or on both of them, a task that can be accomplished by observing if $S_F \perp\!\!\!\perp X_a | X_b$ and $S_F \perp\!\!\!\perp X_b | X_a$. We can also check if there is interaction between X_a and X_b , without taking into consideration S_F . This investigation can be performed by using the Directed Graphs.

A directed edge (a,b) looks like the one in Figure 1.5, while a directed edge (b,a) is shown in Figure 1.6.

The inclusion of directed edges in our graph raises some problem related to directed cycles and in particular to the feed-back issue.

In Figure 1.7 we can observe a new scenario where a influences b , in its turn b influences c and in conclusion c influences a . Unfortunately, joint probabilities do not describe this configuration and for this reason, our toy example deals with directed graphs that do not allow loops, the so-called Directed Acyclic Graphs (DAGs).

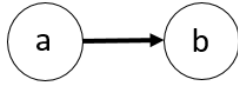


Figure 1.5: A Directed Graph with an Edge from the Node “a” to “b”

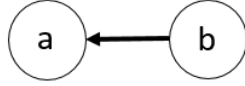


Figure 1.6: A Directed Graph with an Edge from the Node “b” to “a”

More in detail, a directed graph consists in a completely ordered graph $G = (V, E)$, whose edges can assume only one direction and where $V = \{1, 2, \dots, k\}$ and $V(j) = \{1, 2, \dots, j\}$. The directed arrow connecting i and j is not the edge set if and only if $i \perp j | (V(j) \setminus \{i, j\})$ (Wermuth and Lauritzen, 1983).

Moreover, we notice that by dealing with undirected graphs we refer to a single joint distribution, while directed ones refers to a sequence of marginal distributions. This statement allows us to introduce the recursive factorization identity:

$$f_{1,2,\dots,v} = f_{v|V(v)\setminus\{v\}} f_{v-1|V(v)-1\setminus\{v-1\}} \dots f_2|1 f_1$$

By implying that the arrows in our graph can assume only one direction, we state that the variables belonging to the graph are ordered and this allows us to introduce the concept of parenthood:

- in Figure 1.8, the vertices a and b are antecedents of the variable c . Because of this relations we refer to them as the parents of c , in notation we can summarize this statement as follows: $pa(c)$.
- On the other hand, the variable c does not have any outflowing arrow but only inflowing, as pointed out in the previous point. So, we refer to c as the child of a and b .

By assuming that there is a complete ordering of the variables, the introduction of a directed cycle in our graph would violate the irreflexive nature of the structure (Whittaker, 1990). From now on, we consider the graphs in this work as ordered *a priori*, a feature that provides to each variable a past, a present and a future.

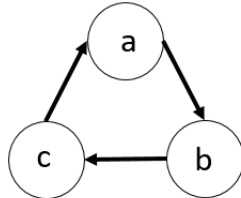


Figure 1.7: A Directed Cyclic Graph with Three Nodes

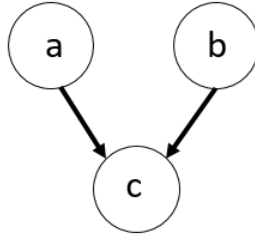


Figure 1.8: A Directed Acyclic Graph with Two Parents and a Child

1.2.1 The Moral Graphs

A moral graph (Lauritzen and Spiegelhalter, 1988) consists in the undirected graph $G^m = (V, E^m)$ associated to the directed graph $G = (V, E)$, where $E^m = \{\{u, v\} \mid u \text{ and } v \text{ are connected or have a child in common}\}$ (Kjaerulff and Madsen, 2013). These graphs are called *moral* because they “marry” the parents that share a child. In Figure 1.9 and 1.10 we show how we obtain G^m starting from a DAG G . First, we add undirected edges between nodes that share a common child and then we substitute all the directed edges with undirected ones. In a moral graph referred to a DAG $G = (V, E)$ the set of vertices connected to a vertex $v \in V$ is called Markov blanket.

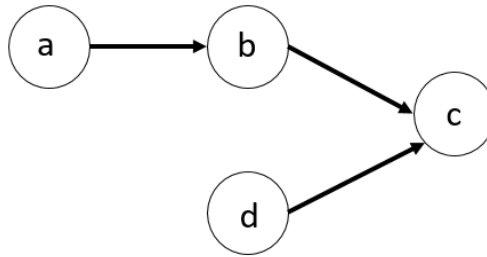


Figure 1.9: A Directed Acyclic Graph Before the Moralization

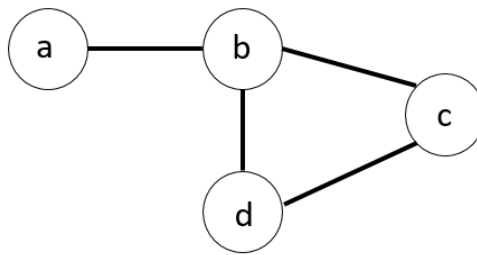


Figure 1.10: The Moralized Graph

1.2.2 The Wermuth Condition

In order to add more detail to graph properties, we now introduce the Wermuth condition (Wermuth and Lauritzen, 1990). First of all we have to observe that

the graph $G = (V, E)$ has an undirected equivalent which is $G^m = (V, E^m)$ that can be obtained just by replacing the directed edge with undirected ones. In Figure 1.11 we represent a simple directed graph whose independences are the following $d \perp\!\!\!\perp b \mid \{a, c\}$, $d \perp\!\!\!\perp a \mid \{b, c\}$ and $c \perp\!\!\!\perp a \mid \{b\}$

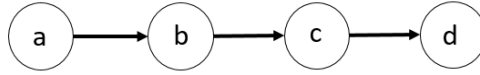


Figure 1.11: A Simple Directed Graph with $V = 4$

From a visual inspection we observe that the path from a to d can be blocked by the vertex b so that $d \perp\!\!\!\perp a \mid \{b\}$. This statement can be proved by introducing the undirected version of our previous diagram, we call it G^m , whose diagram is shown in Figure 1.12.

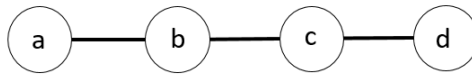


Figure 1.12: A Simple Undirected Graph with $V = 4$

Even in the undirected version of the diagram the following independences hold: $d \perp\!\!\!\perp b \mid \{a, c\}$, $d \perp\!\!\!\perp a \mid \{b, c\}$ and $c \perp\!\!\!\perp a \mid \{b, d\}$. These statements are the same ones introduced when we analyzed the directed version of the graph G .

In order to conclude the presentation of the Wermuth condition we introduce another graph structure, the one represented in Figure 1.13. In this example we have the following independences: $d \perp\!\!\!\perp b \mid \{a, c\}$, $d \perp\!\!\!\perp a \mid \{b, c\}$ and $a \perp\!\!\!\perp b$. If we analyze its respective undirected graph, Figure 1.14, we observe that $a \perp\!\!\!\perp b \mid c$. This statement does not hold for the directed version of the graph because marginal dependence does not imply conditional dependence (Whittaker, 1990).

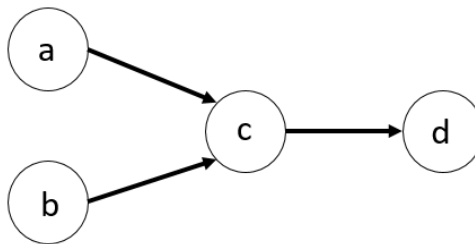


Figure 1.13: A Directed Graph with a Forbidden Configuration

In conclusion, we can observe that the Wermuth condition holds only if the directed graph do not include a forbidden configuration.

1.3 Conditional Independence

The concept of conditional independence represents the corner stone of graphical models theory: two events A and B are conditionally independent given a third event C if the occurrence or non-occurrence of both of them is independent in their

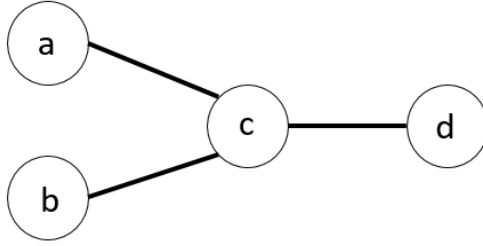


Figure 1.14: An Undirected Graph with a Forbidden Configuration

conditional probability distribution given C. This notion has been introduced by Markov (1906), who considers a set of variables $X_{-1}, X_0, X_1, X_2 \dots X_n$ observed in different moments and that t is the present instance. Then Markov assumes that the conditional probability referred to an observation at $t+1$, given the past and the present, depends neither on the past nor on t but just on the present. This statement can be expressed by the following formula:

$$f_{t+1}(x_{t+1}|x_t, x_{t-1}, \dots, x_1) = f_{t+1}(x_{t+1}|x_t)$$

Equivalently, we can say that the past and the future are conditionally independent given the present (Whittaker, 1990). This theme has been formally treated also by Dawid (1979, 1980), who considers three random variables X, Y and Z with a joint distribution P. The variable X is the one conditionally independent of Y given Z under P, a relation that can be expressed as follows:

$$X \perp\!\!\!\perp Y|Z[P]$$

if for any set A in the same sample space of X there is a conditional probability expressed by $P(A|Y, Z)$. In the case Z is trivial, we can state that X is independent from Y, in notation $X \perp\!\!\!\perp Y$.

On the other hand, if the variables X, Y and Z are discrete and random, the relation among them can be expressed in the following way: $X \perp\!\!\!\perp Y|Z$.

We now introduce the properties referred to the relation $X \perp\!\!\!\perp Y|Z$, where h is an arbitrary function in the sample space of X (Lauritzen, 1996):

1. if $X \perp\!\!\!\perp Y|Z$ then $Y \perp\!\!\!\perp X|Z$
2. if $X \perp\!\!\!\perp Y|Z$ and $U = h(X)$, then $U \perp\!\!\!\perp Y|Z$
3. if $X \perp\!\!\!\perp Y|Z$ and $U = h(X)$, then $X \perp\!\!\!\perp Y|(Z, U)$
4. if $X \perp\!\!\!\perp Y|Z$ and $X \perp\!\!\!\perp W|(Y, Z)$, then $X \perp\!\!\!\perp (W, Y)|Z$
5. if $X \perp\!\!\!\perp Y|Z$ and $X \perp\!\!\!\perp Z|Y$ then $X \perp\!\!\!\perp (Y, Z)$

We underline that the converse to property 4 follows from 2 and 3. In addition, we remark that property 5 holds only if we do not have non-trivial relationship between the variables Y and Z.

1.4 Conditional Independence Graphs

We now introduce a graph with directed and undirected edges, see Figure 1.15. According to the definitions provided in the previous sections, we underline that:

- If the edge between the vertices i and j is directed, the set E contains the pair (i, j) , where i is a parent of j and j is a child of i .
- If the edge between the vertices i and j is undirected, the set E contains both the pairs (i, j) and (j, i) .

In our example, the graph in Figure 1.15 is composed by $V = \{1, 2, 3, 4\}$ and $E = \{(2,1), (1,3), (4,3), (3,4)\}$:

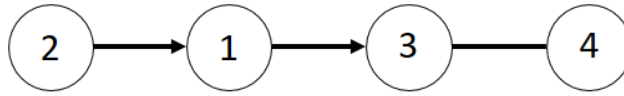


Figure 1.15: An Example of Graph

We can observe that the nodes 3 and 4 are adjacent but neither the pairs $(3,2)$ nor $(3,1)$ are adjacent. In conclusion, we say that a graph is defined as conditionally independent when there is no edge connecting two vertices whenever a pair of variables is independent given all the other ones. More rigorously, we can say that a conditional independence graph of X is an undirected graph $G = (V, E)$ where $V = \{1, 2, 3, \dots, k\}$ and the pair (i, j) does not belong to the set of edges E if and only if $X_i \perp\!\!\!\perp X_j \mid X_{K/\{i,j\}}$ (Whittaker, 1990). We now represent in Figure 1.16 an example of conditional independence graph with four vertices.

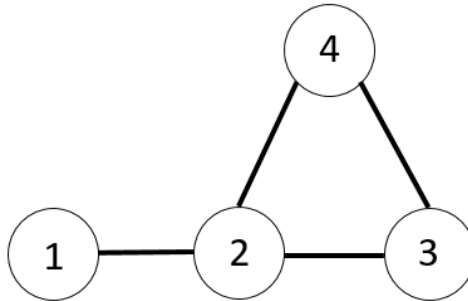


Figure 1.16: An Example of Undirected Graph with Four Vertices

In this particular case we have that $X_1 \perp\!\!\!\perp X_3 \mid (X_2, X_4)$ and that $X_1 \perp\!\!\!\perp X_4 \mid (X_2, X_3)$. The cliques that we can identify in this graph are $\{1, 2\}$ and $\{2, 4, 3\}$.

The Separation Theorem

We now introduce an independence graph with $V = \{a, b, c, d, e\}$, see Figure 1.17. According to the separation theorem (Whittaker, 1990), X_b, X_d, X_e are vectors with

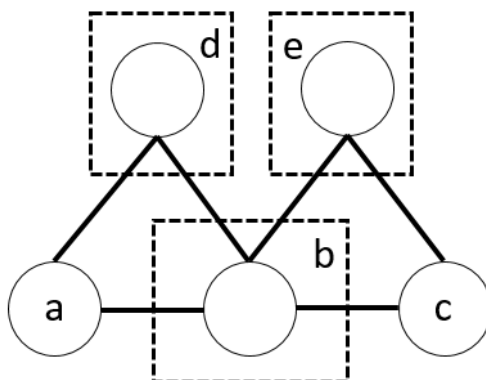


Figure 1.17: An Independence Graph with Three Separating Subsets

disjoint subsets of variables if each vertex belonging to the subset d is separated by b from the subset e . This relation can be written as follows:

$$X_d \perp\!\!\!\perp X_e | X_b$$

On the other hand, when the subset b separates a and c so that $a \perp\!\!\!\perp c | b$, in the case in which a and c are not adjacent, it is true that a and c are conditionally independent given the rest of the subsets. We provide more details on this theme in Section 1.6.1.

1.5 How Nodes Are Connected

In the following paragraphs, we illustrate the possible configurations that can be assumed by the nodes.

1.5.1 Serial Connection

In Figure 1.18 we observe that if the node C1 influences C2, then C2 influences C3. Any evidence on C1 affects the certainty of C2 that in its turn influences the certainty of C3. In the same way, a new evidence on C3 influences the variable C1 by passing through C2. If the state of variable C2 is known, then this channel is blocked, and consequently C1 and C3 become independent variables. This type of reasoning applies to Markov chains (Taroni et al., 2006; Nielsen and Jensen, 2009). In this case, we say that C1 and C3 are d-separated given C2, and when the state of a variable is known, it is instantiated.

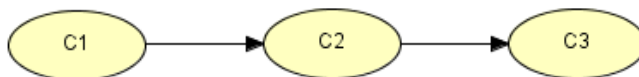


Figure 1.18: A Serial Connection

In conclusion, we can say that any evidence can be transmitted in a serial connection unless the state of a separating variable is known.

1.5.2 Diverging Connection

The scenario represented in Figure 1.19 shows that the information could pass through all C1 children unless the state of C1 is known. In this case, we say that C2, C3 and C4 are d-separated given C1. The variable C1 represents also the root of our network, because it coincides with its origin, while C2, C3 and C4 are called leaves because they are a ramification of the root C1.

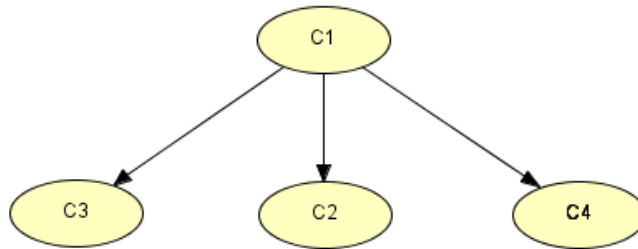


Figure 1.19: A Diverging Connection

In conclusion, we notice that the evidence may be transmitted through a diverging connection unless it is instantiated.

1.5.3 Converging Connection

In this particular case shown in Figure 1.20, the nodes C1, C2, C3 and C4 form a converging connection. If we have no information about C4, with the exception of what can be inferred from the knowledge of its parents C1, C2, and C3, then the parents are independent. In this situation, any evidence on one of them has no effect on the others. If any evidence influences the certainty level of C4, then the parents become dependent. This relation is known as conditional dependence.

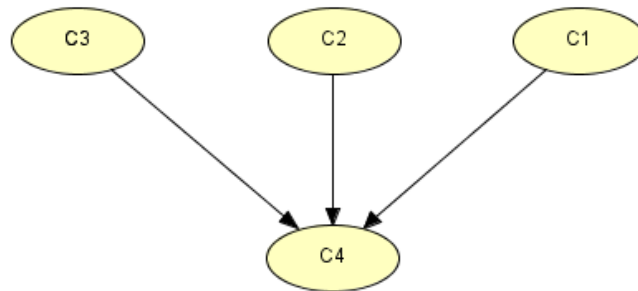


Figure 1.20: A Converging Connection

In conclusion, we can observe that the evidence can be transmitted through a converging connection if a variable of the connection or its descendants receives new evidence. With the term evidence, we refer to a statement on the certainty of the states belonging to a variable. If the evidence provides the exact state we call it hard (instantiation), otherwise we have a soft evidence.

1.6 How to Read Independences in a Graph

1.6.1 Markov Properties and Undirected Graphs

In the following section, we introduce the undirected Markov properties, which allow us to interpret easily a graph. We now consider an undirected graph $G = (V, E)$ and a set of variables X_v belonging to V . Then we assume that P is the probability on X_v that factorizes according to G and that we have a product measure on X (Whittaker, 1990, Cowell et al., 1999, Jensen, 1996) such that $\mu = \otimes_{v \in V} \mu_v$. For example, we can imagine to deal with a non-negative function f_A defined on the group of random variables X_A , which represents a subset of A , and the density p of P factorizes according to this formula:

$$p(x) = \prod_A f_A(x_A)$$

The probability measure P obeys to the following rules:

Pairwise Markov Property (P)

This property states that pairs of not adjacent vertices (a, b) are conditionally independent with respect to the other variables of the graph. This relation can be summarized with the following relation:

$$a \perp\!\!\!\perp b \mid V \setminus \{a, b\}$$

According to the pairwise Markov property, we can observe in Figure 1.21 that $a \perp\!\!\!\perp e \mid \{b, c, d, f, g\}$ and $a \perp\!\!\!\perp e \mid \{b, c, d, f, g\}$.

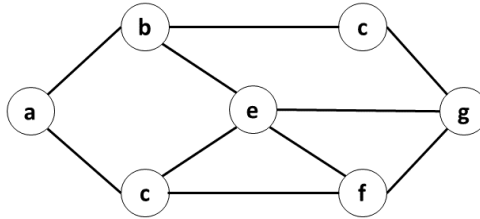


Figure 1.21: Example for Markov Properties and Undirected Graphs

Local Markov Property (L)

The local Markov property states that a vertex is conditionally independent from the other ones belonging to the same graph given its neighbors. By assuming that vertex $a \in V$, we have that

$$a \perp\!\!\!\perp V \setminus cl(a) \mid bd(a)$$

According to the local Markov property, we can observe in Figure 1.21 that $e \perp\!\!\!\perp \{a, d\} \mid \{b, c, f, g\}$ and $g \perp\!\!\!\perp \{a, b, c\} \mid \{d, e, f\}$.

The Concept of Separation

Before introducing the global Markov property, we provide further details on conditional independence graphs, which include some aspects, referred to the interdependence among variables. In order to explain how the separation theorem works we analyze the undirected graph, represented in Figure 1.22.

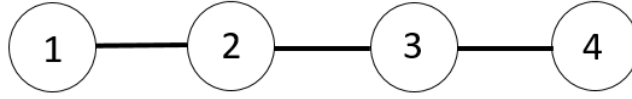


Figure 1.22: An Undirected Graph with $V=\{1, 2, 3, 4\}$

According to the graph structure, we observe that $X_4 \perp\!\!\!\perp X_1 | (X_2, X_3)$. Since including X_2 would be redundant, we can exclude it from our definition and prove the separation theorem by stating that $X_4 \perp\!\!\!\perp X_1 | X_3$.

More in general, we can say that two vertices are separated if the only way to connect them is to pass through the same subset. One of the simplest ways for separating subsets is shown in Figure 1.23.

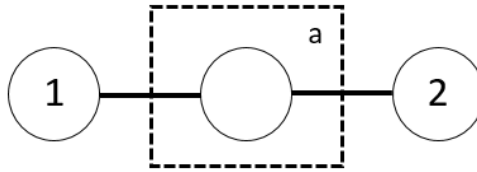


Figure 1.23: An Undirected Graph with a Separating Subset

As we can observe, the vertices 1 and 2 are separated by the subset a , that in this particular case is represented by a dashed rectangle. For a matter of space, we do not introduce all the possible separating subsets. However, we underline that the empty subset can be considered as a separating subset.

Global Markov Property (G)

This property states that, given a triple (A, B, S) of disjoint subsets belonging to V , two subsets A and B are conditionally independent given a third one, S , which separates them.

$$A \perp\!\!\!\perp B | S$$

According to the global Markov property, we can observe in Figure 1.21 that $a \perp\!\!\!\perp f | \{b, e, f\}$ and $b \perp\!\!\!\perp f | \{c, d, e\}$.

Relations Among Undirected Markov Properties

We remark also that the Markov properties are related as follows:

$$(G) \implies (L) \implies (P)$$

This relation means that global Markov property implies local Markov property because $bd(a)$ separates a from $V \setminus cl(a)$. By assuming that L holds, we have that $b \in V \setminus cl(a)$ because a and b are non-adjacent vertices. Consequently, we have that

$$bd(a) \cup ((V \setminus cl(a)) \setminus \{b\}) = V \setminus \{a, b\}$$

and according to the local Markov property and the property 3 in Section 1.2 we have that

$$a \perp\!\!\!\perp V \setminus cl(a) | V \setminus \{a, b\}$$

By applying the property 2 reported in Section 1.2 we obtain that $a \perp\!\!\!\perp b | V \setminus \{a, b\}$, that is the pairwise Markov property. We should also observe that the relation $(G) \implies (L) \implies (P)$ depends only on the properties of conditional independence from 1 to 4 proposed in Section 1.2.

1.6.2 Markov Properties and Directed Acyclic Graphs

We now show that the Markov properties hold even if we assume that the graph $G = (V, E)$ is directed and acyclic (Smith, 1989; Geiger and Pearl, 1990). The recursive factorization on the probability distribution P is admitted if we have finite measures μ_a and non-negative functions k^a , with a representing a subset of V , defined on $X_a \times X_{pa(a)}$. According to the previous assumptions, we have:

$$\int k^a(y_a, x_{pa(a)}) \mu_a(dy_a) = 1$$

then the density distribution p on P , assuming $\mu = \otimes_{a \in V} \mu_a$, can be represented by the following formula

$$p(x) = \prod_a \in V k^a(x_a, x_{pa(a)})$$

We can now state that for P is valid the directed Factorization property.

The Directed Local Markov Property

We can observe that the probability distribution P is subject to the directed local Markov property if any variable, given its parents, is conditionally independent from its non-descendants:

$$a \perp\!\!\!\perp nd(a) | pa(a)$$

According to the directed local Markov property, we can observe in Figure 1.24 that $d \perp\!\!\!\perp \{a, c, e, f\} | b$; $e \perp\!\!\!\perp \{a, d\} | \{b, c\}$; $c \perp\!\!\!\perp \{b, d\} | a$.

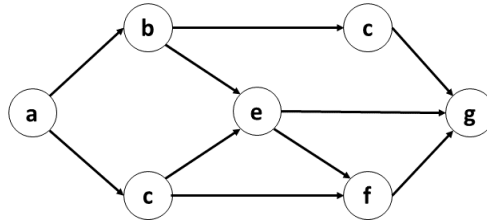


Figure 1.24: Example for Markov Properties and Directed Graphs

The Ordered Markov Property

We now suppose that the vertices of a generic DAG are well ordered, which means that they are linearly ordered so that $a \in pa(b) \implies a < b$. In this case, the ordered Markov property holds for a well-ordered graph if

$$\forall a \in V : a \perp\!\!\!\perp \{pr(a) \setminus pa(a)\} | pa(a)$$

where $pr(a)$ stands for *predecessors* of a .

According to the directed ordered Markov property, we can observe in Figure 1.24 that $d \perp\!\!\!\perp \{a, c\} | b$; $e \perp\!\!\!\perp \{a, d\} | \{b, c\}$; $c \perp\!\!\!\perp b | a$.

d-Separation

Before introducing the directed global Markov property, we now define the concept of d-separation. Two variables A and B are d-separated if for all the paths between them there is an intermediate variable C such that we have a serial or a diverging connection and the state of C is known. We have d-separation also when we have a converging connection and any new evidence has updated neither C nor its descendants. If A and B are d-separated, any change in the certainty of A does not affect our level of certainty on B . If A and B are not d-separated, we say that they are d-connected. In this case, a path $k = [A, \dots, B]$ in a DAG $G = \{V, E\}$ is blocked by $C \subseteq V$ if k contains a vertex $c \in V$ and the edges of the path k do not meet head-to-head in c (Kjaerulff and Madsen, 2013). The concept of d-separation has been analyzed in a seminal book written by Pearl (1988) and subsequently analyzed by Lauritzen (1990).

The Directed Global Markov Property

We now show that the global Markov property holds for directed acyclic graphs too. By assuming that the probability distribution P admits a recursive factorization between the graph G and an ancestral set A , the global Markov property is valid for P_A and G_A . By allowing the recursive factorization, we can write that

$$A \perp\!\!\!\perp B | Z$$

where the moralized graph of the smallest ancestral set includes A , B and Z and A , B are separated by Z (Pearl, 1986). As in the case of the global Markov property for undirected graphs, the directed global Markov property represents an efficient tool for reading dependence/independence relations in a directed graph.

According to the directed global Markov property, we observe in Figure 1.24 that $d \perp\!\!\!\perp f | \{c, e\}$.

Chapter 2

Bayesian Networks and Object Oriented Bayesian Networks

2.1 The Expert and the Machine

It is generally recognized that experts have a deep knowledge about a specific subject. For example, a banker knows the interest rate that should be applied to a customer loan, and a fund manager knows what should be bought or sold on the market. By observing his side of the world, an expert understands and defines with precision its possible states. The banker uses the information available on the borrower in order to measure the probabilities of paying back his debt, while the fund manager is able to interpret a Central Bank announcement and forecast how financial markets can react. Then, by following his interpretation of the states of the world, the expert decides what to do. The banker can decide to apply a higher or a lower interest on a loan, while the fund manager can chose to go long or short on corporate bonds or stocks. For any decision taken, the expert expects an output that only sometimes comes true. We can divide this analytical process in three main steps:

1. The expert observes the state of the world
2. He takes an action according to his personal view
3. In conclusion, he checks the output of his decision and if his expectations have been met.

Because of the inefficiency evidenced by men in the process outlined above, the technological evolution tried to replace any human intervention with expert systems that consists in a computer model of the human expert. This innovation has been modeled around a production rule and it can be summarized as follows (Jensen, 1996):

“if condition then fact; if condition then action”

A rule-based system is based on inference and knowledge. An inference system combines rules and observations in order to obtain a result referred to a specific state of the world and to the actions taken by the expert, while a set of production rules represents the core of a knowledge-based system.

2.2 Dealing with Uncertainty

Despite their contribution to innovation, the expert systems evidenced some limit too, especially in situations of reasoning under uncertainty. For example, a source of uncertainty can be represented by a missing data, a partial observations or the vagueness of the terms involved (i.e. good stock/bad stock; small company/large company...). The production rule previously introduced can be updated by adding an element referred to uncertainty and the relation can be updated as follows (Jensen, 1996):

“if condition, with a certainty x ; then fact, with a certainty $f(x)$ ”

where f is a function.

Then the inference system needs to be updated in order to guarantee coherence when dealing with uncertain situations. However, we should remember that inference rules are not suitable in the most of the cases.

In order to provide a support to human experts and to overcome some of the limits evidenced, normative expert systems have been introduced. The difference between the rule based expert systems and the normative ones is that only the latter models the domain. More in detail, they use common probability and decision theories. Furthermore, they do not substitute the expert, but they support him. The first attempt of introducing classical probability principle in an expert system has been conducted in the 1960s by Gorry and Barnett (1968) but in the 1980s Pearl (1986) revived these principles by adapting BNs to expert systems.

2.3 Message Propagation in the Network

Following the approach proposed by Jensen (1996), the models illustrated in the next sections are causal networks, which consist in a set of variables that are connected by directed links. All these elements represent events and propositions. In the examples shown in Figures 2.1, 2.2 and 2.3, each variable has 2 states, YES or NO, that indicates whether an event happened or not. In general, a variable can have an infinite number of states (countable or continuous) but in this work we consider only nodes with a finite number of states. We use causal network to represent a set of variables connected together by some directed edges, with the explicit requirement that there is a causal dependence between the ones that are linked (Pearl, 2000). In the following sections we exploit three toy examples referred to “Oil Stocks”, “Iron Price and Dividend” and “Why SYN price is up by 8%?” to plug in the probabilities associated to the states referred to each variable and show how a BN works. This step allows us to pass from a purely qualitative representation to a quantitative one.

We can now define a BN, see e.g. Jensen (1996) and Cowell et al. (1999), as a tool for modeling large multivariate probability structures and making inference. It is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a Directed Acyclic Graph (DAG). The variables under investigation are represented through the nodes of DAG, and the joint distribution can be expressed in terms of the product of the conditional table associated to each node given its parents.

The use of a graph as a pictorial representation of the problem at hand simplifies the model interpretation and facilitates communication and interaction among agents with different kind and degrees of information. BNs provide an inferential engine to

make inference on the parameters and the structure of the model, and they can be interactively updated with new information.

As mentioned above, in the next sections we introduce three different financial scenarios that follow the schemes of some well-known example elaborated by Jensen (1996) in order to illustrate some crucial aspects related to the reasoning under uncertainty process.

2.3.1 Oil Stocks

A fund manager holds in his portfolio two oil stocks, ExxonMobil (XOM) and Petrobras (PBR). At the opening bell the stocks are both losing 1.5%, so he starts wondering that the dynamic is justified by the oil price drop to 30\$ per barrel. Consequently, he gets worried because he knows that today there is a high probability that the oil price continues to slide towards new lows. Knowing that PBR and XOM are both oil stocks, if one of the them crashes because of the low barrel price, the other one is supposed to go down too. Therefore, the fund manager wants to monitor closely the situation and then decide what to do with these positions. At 2 p.m., he observes that PBR price is down by 6%, while XOM still loses 1.5%. The fund manager checks the oil price and he observes that it is stable at 30\$ per barrel, so PBR losses are not related to any oil price movement. Now that he knows that PBR performance does not depend on oil dynamics, he believes that XOM price will not collapse on that trading day too. In order to simplify this example, we represent each possible event as a two states variable: YES and NO. Variables with this characteristic are called Boolean. We suppose also that a different level of certainty, represented by a real number, is associated to each event. In this particular example, we have three variables: Oil price goes down (OD), Petrobras stock goes down (PD) and ExxonMobil stock goes down (ED). OD increases the level of certainty associated both to PD and ED. We represent the scenario in Figure 2.1. The arrows on the links model the direct impact while the other black arrows indicates the direction of the impact on certainty.

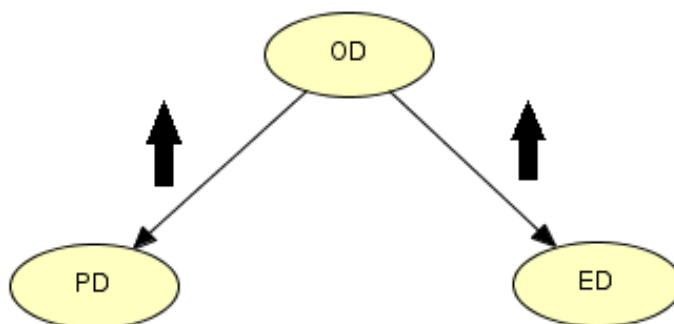


Figure 2.1: A Network Model of *Low Oil Price*

When the fund manager observes that PBR price is down by 6%, he is reasoning in the opposite direction of the directed arrow. Since the impact function that point to PD is increasing, the inverse function is increasing too. Hence, the fund manager gets an increased certainty of OD. Then the increased certainty of OD generates a new expectation, corresponding to an increased certainty of ED. At the same time he

observes that the oil price is stable at 30\$ per barrel, so the fact the PBR is losing 6% cannot change his expectations concerning oil price and, as a consequence, PBR stock crash has no influence on XOM stock performance.

This example shows how dependence/independence among variables changes according to the information gathered. Until the fund manager checks the oil price level during that specific trading day, PBR and XOM are dependent because the information on each of them affects the certainty we have on the other stock. However, when the oil price is known (stable around 30\$ per barrel) then XOM and PBR become independent and any information on one of these stocks do not affect the other and vice versa. This relation is called conditional independence.

2.3.2 Iron Price and Dividend

The hedge fund AQR has a long position on a basic resources stock, RioTinto (RIO). The fund manager knows that commodities prices are going down because of the low demand from the emerging markets. On 12th August 2015, he notices that RIO is down by 10%, so he wonders about the possible causes of the crash. Maybe RIO is down because that day the dividend (D) has been paid or maybe the stock is under pressure because the iron price (I) hit its historical low in the last 25 years. His belief on both the events grows. In order to gather additional information, the fund manager checks how another basic resources stock, Glencore (GLEN), performs on that trading day. He observes that GLEN is losing 15% (G). Now he knows that RIO price is down by 10% because of the iron price and not because of the dividend payment. When the AQR fund manager observes RIO's bad performance, he is reasoning in the opposite direction of the direct arrows. Since both impact functions are pointing towards RIO, which is down by 10%, his certainty in I and D increases. In turn, the increased certainty on I determines an increased certainty that GLEN is performing poorly. Therefore, the fund manager observes that investors are heavily selling both GLEN and RIO so his certainty on I increases immediately. Now that RIO's crash has been justified, there is no longer reason to believe that the stock is down by 10% because of the dividend payment. Hence, the certainty of D is reduced to its initial size.

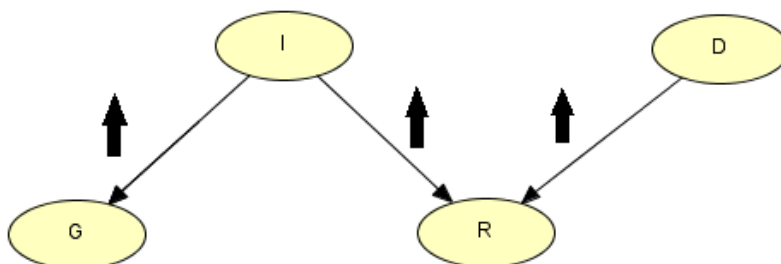


Figure 2.2: A Network Model of *Iron Price* and *Dividend*

Thanks to this example, we can observe that when the fund manager gathers new information on GLEN's performance he understands that RIO stocks are falling because of the falling iron price (I) and not because of the dividend payment (D).

Some Clarification on Causation

The scenarios presented in Figure 2.1 and 2.2 show the newsflow impact among several events. In both these examples, we observe how the level of certainty associated to a variable can be updated when we gather new information. These models suggest a guideline for the reasoning process referred to unknown events. When we reason in the same direction of the links, our model follows this statement (Jensen, 1996):

“An event A causes the event B, with a certainty x ”

From this statement, we can generalize by saying that:

“if we know that an event A occurred,”

then

“the event B has taken place with a certainty x ”

The process of reasoning in the opposite direction of the links is more delicate. Until now we have assumed that the certainty on an event A increases when its consequence B have taken place. In order to perform this inversion, we have to use a quantitative statement known as the Bayes’ rule, an aspect that will be analyzed more in detail in Section 2.4.1 . We conclude by remarking that Bayesian networks are not causal models, but models where information propagates between events (Jensen, 1996).

2.3.3 Why SYN is Up by 8%?

On 28th November 2015, Syngenta (SYN) is up by 8% (S). Tom, a retail investor, checks his bank account and observes that he is making a consistent gain on his investment. He now believes that the Board of Directors has approved the industrial plan he was waiting for (IP) and that the guidance released by the company sets very high revenues targets for the next years. One minute later Jim (J), Tom’s neighbor, phones him because he wants to congratulate with his friend to have guessed Monsanto’s (MON) acquisition target (M). Therefore, the stock is going up because of MON’s takeover offer on SYN and not because of the industrial plan. Jim now can consider selling his stocks. In Figure 2.3 we show the structure of the graph, which is similar to the one represented in Figure 2.2 (Iron Price and Dividend toy example).

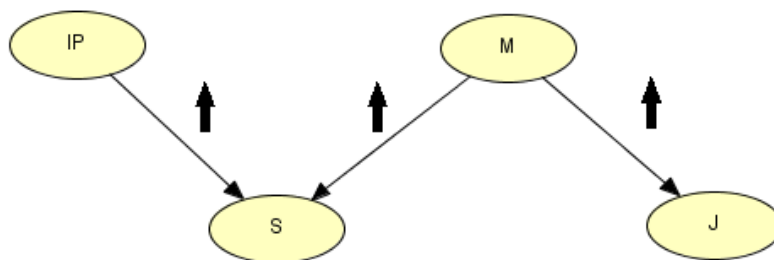


Figure 2.3: A Network Model of *SYN Stocks Up by 8%*

The Relevance of Prior Certainties

From the previous examples, we can notice that if an event is known, then the level of certainty referred to other variables changes. We need to have some certainties prior to any information if we want to calculate the actual certainty referred to a specific event. More in detail, we need priors referred to the events that are not considered as an effect of other variables belonging to the network. By observing the example “Why SYN is up by 8%?”, we know that SYN stocks are going up. The investor wonders that this dynamic is justified by the diffusion of the new industrial plan. This belief is updated when Tom is informed by his friend Jim that MON has just made a takeover offer on SYN, an event that increases the level of certainty associated to M. On the other hand, the prior certainty associated to a guidance upwards revision decreases. In order to follow this reasoning path, prior certainties on M and IP are needed.

2.4 Bayesian Networks

Now that the qualitative features have been examined in detail, we include the quantitative aspects by introducing the BNs. A BN is a probabilistic graphical model that exploits a DAG in order to represent a set of random variables and their conditional dependences (Pearl, 1988; Neapolitan, 1990). More in detail, the basic elements of a BN over X variables are a DAG $G = (V, E)$ and a set of conditional probability distributions P . Each node v corresponds to a random variable X_v belonging to X that generally has a defined number of mutually exclusive states (Jensen, 2001; Uusitalo, 2007; Fenton and Neil, 2012). On the other hand, the directed links E allow us to specify the relations of conditional dependence/independence among the variables belonging to the network by exploiting the concept of d-separation. We underline also that there is a conditional probability distribution $P(X_v|X_{pa(v)}) \in P$ for each variable of the network. Usually the parents of v , $pa(v)$, are also labeled as the conditioning variables while the variable v is called conditioned variable (Lauritzen, 2003; Kjaerulff and Madsen, 2013). There are some basic axioms that represent the building blocks of the BNs. One of the most important states that the probability $P(A)$ referred to an event A is a number comprised between 0 and 1. If $P(A) = 1$, we are certain of the occurrence of the event A , on the other hand if we are sure that A will not happen $P(A) = 0$. If the events A and B are mutually exclusive, then $P(A \cup B) = P(A) + P(B)$.

2.4.1 Conditional Probabilities

Another basic concept about BNs is the one referred to conditional probability. Whenever a statement on the probability $P(A)$ is given, then it is given conditioned by other elements. A conditioned probability statement can be expressed in the following way:

the probability of A, given B, is x

or in notation

$$P(A|B)=x$$

This statement implies that if B is true, and everything else known do not affect A , then the probability referring to A is x . The fundamental rule (Jensen,1996) for calculating probabilities is the following:

$$P(A|B)P(B) = P(A, B)$$

where $P(A, B)$ represents the probability corresponding to the joint event A and B. We should also remember that the probabilities associated to the events A and B can always be conditioned by an event C, so the fundamental rule formula can be updated in the following way:

$$P(A|B, C)P(B|C) = P(A, B|C)$$

Then, we can derive the Bayes' rule from the fundamental rule. We underline that the term "Bayesian" associated to BNs is not referred to the Bayesian inferential paradigm but on the information propagation algorithm based on the Bayes' rule. This formula is considered BNs pillar because it allows updating the probability associated to an event when we gather new information about a particular scenario:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

In the case in which the probabilities associated to the event A and B are conditioned by an event C we have:

$$P(B|A, C) = \frac{P(A|B, C)P(B|C)}{P(A|C)}$$

2.5 Calculating probabilities

The variables belonging to the network are called nodes, and they have a finite number of mutually exclusive states. If we deal with a variable A with states ranging from a_1 to a_n , then its probability $P(A)$ is represented by a probability distribution among them. We can summarize this concept as follows:

$$P(A) = (x_1 \dots x_n); x_i \geq 0; \sum_{i=1}^n x_i = 1$$

Where the probability for A of being in state a_i is x_i . If we add to our example a variable B, with states from b_1 to b_m , then $P(A|B)$ is an $n \times m$ table that contains all the combinations referred to $P(a_i|b_j)$

Table 2.1: An example of $P(A|B)$. All columns sum to 1.

	b_1	b_2	b_3
a_1	0.2	0.1	0.4
a_2	0.8	0.9	0.6

If we consider the joint probabilities for A and B and that $P(B) = (0.2; 0.4; 0.4)$, $P(A, B)$ is again an $n \times m$ table as shown in Table 2.2: Then if we apply the fundamental

Table 2.2: An example of $P(A, B)$. All the entries sum to 1.

	b_1	b_2	b_3
a_1	0.12	0.04	0.16
a_2	0.08	0.36	0.24

rule on A and B, we have to apply the procedure to the $n \times m$ configuration (a_i, b_i) :

$$P(a_i|b_i)P(b_j) = P(A_j, b_j)$$

We can also calculate the probability distribution referred to $P(A)$ from a table $P(A, B)$. For example, we can assume that a_i is one of the possible state of A. The variable A can assume the state a_i in different m situations. These events $(a_i, b_i) \dots (a_m, b_m)$, are mutually exclusive and consequently we obtain $P(a_i) = \sum_{j=1}^m P(a_i, b_j)$. This step is called marginalization and allow us to obtain $P(A)$:

$$P(A) = \sum_B P(A, B)$$

If we marginalize B out of Table 2.2 we obtain that $P(A) = (0.32, 0.68)$

2.6 Probabilities and Bayesian Networks

The quantitative aspect referred to causal relations is called strength. For example, we can assume that a node A is parent of the node B. Thanks to probability calculus we obtain that $P(B|A)$ is the strength associated to the arrow connecting the variables. Furthermore, we have to consider that if there is a variable C that in its turn is parent of B, then considering the two conditional probabilities $P(B|A)$ and $P(B|C)$ alone does not provide any useful information on the interaction between A and B. We need more understanding of $P(B|A, C)$, and BNs can help us in this task. The variables composing the BN belongs to different sources such as empirical data, expert opinions or simulation outputs (McCann et al, 2006, Koski and Noble, 2009). In correspondence of each node, we have a probability table that is determined by the possible states referred to the variable and to the states of parent nodes. The BN provides probabilities for each node according to the factors, interactions and individual conditional probabilities that influence them (Pearl, 1988). Each variable A with parents $B_1 \dots B_n$ has an attached conditional probability table $P(A|B_1 \dots B_n)$. The probability framework referred to the BN provides us information on the strength of the relationships among the variables (Jensen, 1996). The advantage of using a BN is that it allows applying the concept of d-separation. For example, if we introduce a new evidence e in the network and then A and B are d-separated, we can summarize this statement as follows:

$$P(A|B, e) = P(A|e)$$

This axiom shows us that we can use d-separation in order to find conditional independences in the BN.

The Chain Rule

We now consider dealing with a universe of variables $U=(A_1, \dots A_n)$. If we have access to the joint probabilities table $P(U)=P(A_1, \dots A_n)$, we can consequently obtain $P(A_i)$ and $P(A_i|e)$, where e is the evidence. However, we have to take into account that the dimension of our universe influences the value of $P(U)$, that grows exponentially with the number of variables considered. In the eventuality that our sample is too big, it becomes infeasible to treat all these data in a table. BNs help us to draw a more compact representation of $P(U)$, and if the conditional independence holds for U, then $P(U)$ can be obtained from the conditional probabilities of the network. In order to introduce the chain rule, we have to assume that the BN has been built over

the universe $U=(A_1, \dots, A_m)$. Then the joint probability $P(U)$ represents the product of all the conditioned probabilities included in the BN:

$$P(U) = \prod_i P(A_i | pa(A_i))$$

$pa(A_i)$ represents a parent set of A_i .

2.7 Revisiting the Previous Examples

In this paragraph, we revisit the examples referred to “Oil Stocks” and “Iron Price and Dividend” by applying the rules of probability calculus.

2.7.1 Oil Stocks Example Revisited

In the toy example proposed in Figure 2.1 we state that only the oil price dynamics are relevant for PBR and XOM stock prices. After having introduced our scenario from a qualitative perspective, we calculate $P(PD|OD)$, $P(ED|OD)$ and $P(OD)$. In order to perform this quantitative analysis, we have to attribute a level of certainty to OD according to fund manager knowledge on the subject. In this case, he knows that the oil price is sliding because OPEC decided not to cut its production. Therefore, we assume that $P(OD=y)$ is 0.80 and that $P(OD=n)$ is 0.20. Since both PBR and XOM are oil stocks, they suffer if the oil price plunges. For this reason, we give a 85% probability to the case in which both stocks go down if the oil price falls. On the other hand, we associate a 15% probability to the scenario in which PBR and XOM stocks go up when at the same time the barrel price goes down.

Table 2.3: Conditional Probabilities for $P(PD = ED|OD)$

	OD=y	OD=n
PD = ED =y	0.85	0.2
PD = ED =n	0.15	0.8

In order to obtain the initial probabilities for PD and ED we can use the fundamental rule for calculating $P(PD, OD)$ and $P(ED, OD)$:

$$P(PD = ED = y, OD = y) = P(PD = ED = y|OD = y)P(OD = y) = 0.85 \cdot 0.80 = 0.68$$

$$P(PD = ED = n, OD = y) = P(PD = ED = n|OD = y)P(OD = y) = 0.15 \cdot 0.80 = 0.12$$

$$P(PD = ED = y, OD = n) = P(PD = ED = y|OD = n)P(OD = n) = 0.20 \cdot 0.20 = 0.04$$

$$P(PD = ED = n, OD = n) = P(PD = ED = n|OD = n)P(OD = n) = 0.80 \cdot 0.20 = 0.16$$

Table 2.4: Joint Probability Table for $P(PD = ED, OD)$

	OD=y	OD=n
PD = ED =y	0.68	0.04
PD = ED =n	0.12	0.16

In order to get the probabilities for PD and ED we marginalize OD out of Table 2.4 and we get that:

$$P(PD) = P(ED) = (0.72, 0.28)$$

He needs the information that PBR stock is down by 6% at 2 p.m. in order to update the probability of OD. In order to do that, we use the Bayes' rule:

$$\begin{aligned} P(OD|PD = y) &= \frac{P(PD = y|OD)(P(OD))}{P(PD = y)} \\ &= \frac{1}{0.72} \cdot (0.85 \cdot 0.80, 0.20 \cdot 0.20) \\ &= (0.944, 0.056) \end{aligned}$$

To update the probability of ED, we use the fundamental rule to calculate $P(ED, OD)$, see Table 2.5:

Table 2.5: Calculating the Updated $P(ED)$

	OD=y	OD=n
ED =y	0.803	0.011
ED =n	0.142	0.044

In conclusion, we calculate $P(ED)$ by marginalizing OD out of $P(ED, OD)$. The updated result is $P(ED) = (0.814, 0.186)$. This result incorporates the quantitative effect of the information that Petrobras stock crashed. At last, when the fund manager observes that the oil price is stable at 30\$ per barrel, then $P(ED|PD = y) = (0.20, 0.80)$.

2.7.2 Iron Price and Dividend Example Revisited

We now propose to revisit the example referred to Figure 2.2, in which the AQR fund manager wants to understand whether RIO stock is going down because of the dividend payment (D) or because the iron price hit new lows (I). Let the prior probability for R, representing the probability for Rio Tinto stock to go down, be $P(R) = (0.5)$. Consequently, we have that $P(D|R) = (0.10, 0.50)$ and that $P(I|R) = (0.80, 0.20)$. According to these conditioned probabilities, the probability of Glencore to go down (G) is equal to 99% if both D and I occurs, 90% if D happens and I do not or viceversa and 1% if D and I are both false. We underline that in Figure 2.4 we focus on how the information propagates through the network and that we are not representing the BN structure shown in Figure 2.2.

Moreover, by exploiting the conditional independence relationships, we can write the BN joint probabilities as follows

$$p(G, D, I, R) = p(G|D, I)p(D|R)p(I|R)p(R)$$

For example, in the case every variable is TRUE we have that

$$p(T, T, T, T) = 0.99 * 0.1 * 0.8 * 0.5 = 0.0396$$

while if R is FALSE and the other variables are TRUE we have

$$p(F, T, T, T) = 0.01 * 0.1 * 0.8 * 0.5 = 0.0004$$

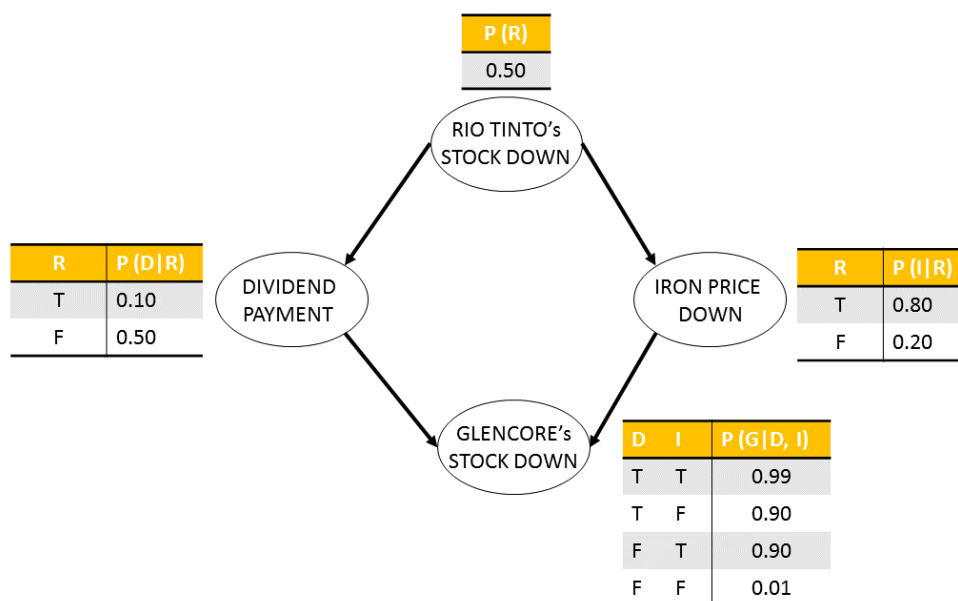


Figure 2.4: Probabilities and the Steps that Lead us to a Conclusion

2.8 Object Oriented Bayesian Networks

In the previous sections, we evidenced the potentiality of BNs as a modeling language. Even if they represent an efficient tool for dealing with reasoning under uncertainty situations, they appear to be more suitable for small and medium domains but not for large and complex ones (Mahoney and Laskey, 1996). In a BN, each node corresponds to a variable or an attribute built for a specific domain and because of that, we cannot use multiple times the same repetitive pattern. In order to overcome this issue, we now propose the use of OOBNs because they allow describing more complex domains through inter-related objects (Koller and Pfeffer, 1997). This particular BNs extension is called *object oriented* because it is based on Object Oriented programming language (Goldberg and Robson, 1983) that provides a flexible framework.

The OOBNs are more efficient with respect to BNs because:

- they allow the user to adopt a top down or a bottom up approach
- they exploit elements and useful features belonging to the Object Oriented programming (see Section 2.9)
- they use classes and instance nodes in order to build the network
- they allow to simplify a framework by subdividing a complex problem in simpler ones, making easier the communication between the expert and the users
- they allow to replicate recursive patterns easily

According to Kjaerulff and Madsen (2013), the concept of object-orientation can be defined as the combination of “objects + inheritance”. An object represents the instance of a class, while the inheritance provides indications on the relation among classes.

2.8.1 Limits of the BNs approach

We start this section by introducing an example referred to the Apple stock (AAPL) that every year pays a cumulative dividend of a certain dollar amount. Our analysis is focused on a three years range, and for this reason we show in Figure 2.5 three equal time slices. According to the management decision about the entity of the dividend, represented by the nodes D1 or D2, the stock has two different reactions, nodes SR1 or SR2. From a visual inspection, we notice that the final BN consists in the repetition of the Year 0 pattern for three times, with links connecting the nodes referred to the previous year dividend with the ones corresponding to the following year dividend. In this case, the network is composed by three sub-BNs that are merged in the same structure. However, this representation does not allow a top down modeling, which consists in treating a sub-BN as a slice of the whole BN. In order to avoid this issue, we can identify the sub-BNs with a smart positioning of their nodes. Moreover, this process do not allow to “copy and paste” the same time slice all the times we need. Thanks to this example, we can also observe that splitting a problem in several parts

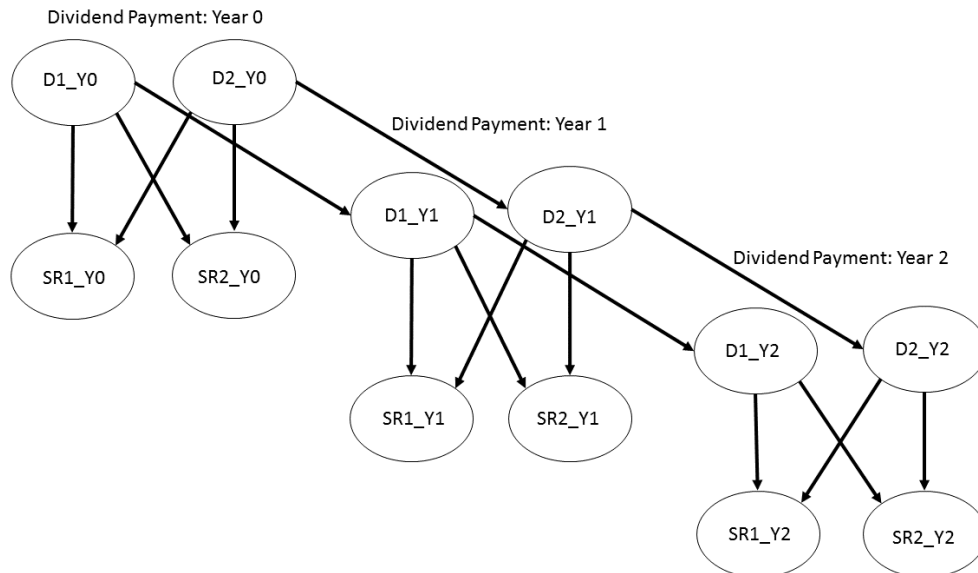


Figure 2.5: The Yearly Pattern for Dividend Payments and Stock Reactions

allows us to improve our understanding and simplifies the modeling task.

2.9 The OOBN Basic Elements

We now analyze in detail the basic elements of an OOBN by providing detailed descriptions and definitions.

The Object

At the base of the OOBN, we have the object. This element can be simple or complex. The former category corresponds to a random single variable; on the other hand, a group of simple objects is at the basis of the latter one. Even a BN can be identified as

an object. The object represents a set of properties associated to a variable belonging to the domain analyzed. The models can be based on physical entities, for example, they can be referred to the entire stock market, or they can be based on abstract entities, such as investors' behavior and psychology. We now define the different types of objects according to the ones proposed and elaborated by Koller and Pfeffer (1997):

- Basic Type: it represents a set of values or a user defined set that can be Boolean, Integer, Real or belonging to a finite set defined by the user: e.g. Sector = {Banks, Utilities, Mining}.
- Structured Type: it consists in a set of values defined by a tuple $\{A_1 : t_1 \dots A_i : t_i\}$. The values $A_1 \dots A_i$ are attribute labels, while $t_i \dots t_1$ are corresponding basic or structured types. For example, a structured type referred to a company CEO can be Age, Experience, and Retribution etc. . . . The Age attribute has its own defined set {45-50; 51-65; 66+yrs}; the experience can have the following labels {10-15; 16-20; 20+yrs}; and the Retribution can take values belonging to the following ranges {100k-150k; 151k-200k; 201k-250k; 251k+}

In conclusion, we remark that an OOBN can also contain instances of other networks encapsulated in the model.

The Class and the Attributes

By dealing with the OOBNs, we have also to introduce the concept of class, a basic element that encapsulates some nodes and consequently that makes them invisible from the outside. We can also refer to a class by calling it Object Oriented Network Fragment (OONF) (Koller and Pfeffer, 1997). In order to exploit class features we have to instantiate it. If the process occurs inside another class a particular node called instantiation represents it. In the example in Figure 2.6, we observe the effects of AAPL dividend OOBN and its recursive pattern. Each year represents an instantiation of the same class, for this reason we can say that AAPL dividend OOBN is composed by three instantiations.

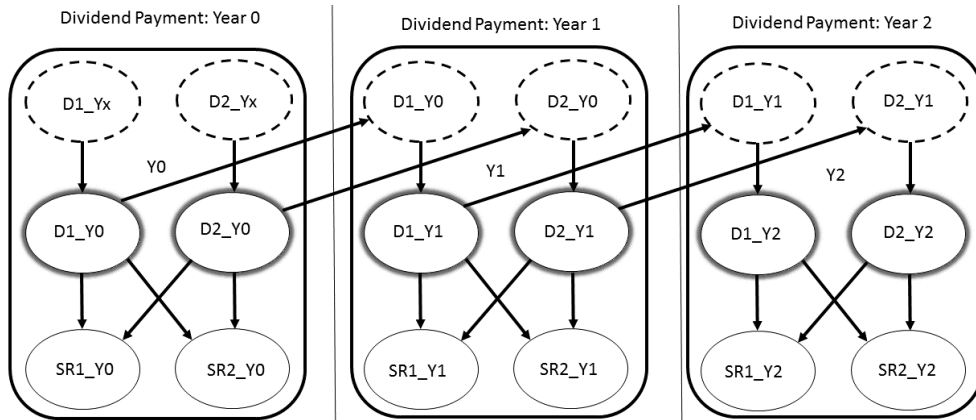


Figure 2.6: The Yearly Pattern for Dividend Payments and Stock Reactions

Even if we can look at a class as a single element, we need interactions among instantiations. For example, a node inside a class can have parents outside of it or

we can have a node outside the class that has its parents on the inside. However, the former relation is problematic because the potential of inner nodes should not be influenced by any instantiation of the class (Bangso and Wullemin, 2000). In order to solve this issue, we can introduce a reference node, corresponding to the shaded ones in the first time slice (D1_Y0 or D2_Y0), with the objective of enlarging the visibility of the node. More in detail, a reference node has its own specific characteristics, such as position or label, while it shares the type and the potential with its referenced node, represented for example by the dashed ones in the second time slice (D1_Y0 or D2_Y0). The reference nodes are visible outside of the class instantiations and they are linked through a reference link to their respective referenced node. We remark that it is not possible to change a reference node potential and this statement implies that it cannot have parents. On the other hand, it is possible to observe interactions between nodes that have their parents inside of a reference node. This is possible because every node belonging to a class can be designated as an output node and consequently it becomes visible from outside the class. This implication allows to an output node to be parent of a node outside of its class. Input and output nodes together represent the class interface and they are the only part visible from the outside of each class. More in detail, we can define a class as a part of a BN that contains both instantiations and reference nodes. In this framework, we can identify three different kinds of nodes:

- The input nodes, which are parameters to the object (the dashed ones belonging to instantiation k2 in Figure 2.7)
- The output nodes, which are visible outside of the class and to the rest of the model (the shaded one belonging to instantiation k1 in in Figure 2.7)
- The encapsulated nodes, known also as internal nodes, that are only visible within the class (the nodes C3 and C4 belonging to instantiation k2 in Figure 2.7)

In Figure 2.7 we graphically represent the input/output scheme: the node O belongs to the instantiation $k1$ and it is parent of the nodes c1 and C2 outside $k1$; on the other hand X is a parent of nodes C3 and C4 inside the instantiation $k2$ and thanks to the input node I it is connected to its children.

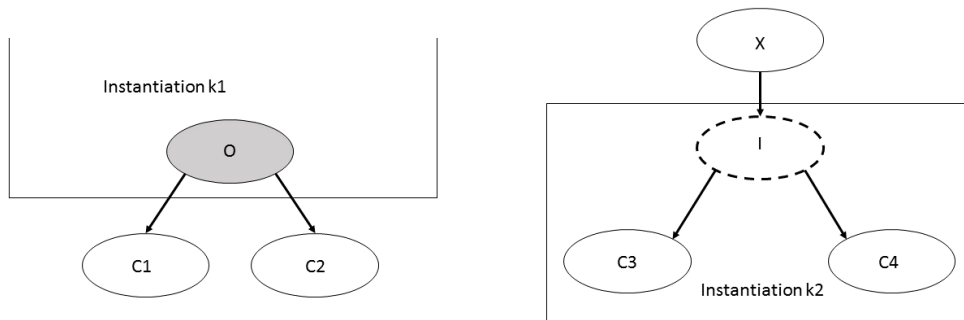


Figure 2.7: Output (O) and Input (I) Nodes Representation

Output and encapsulated attributes belong to the complex objects category. On the other hand, input attributes are not objects but only parameters. Given that the objective of our analysis is to model the levels of uncertainty associated to an object,

we have to introduce a stochastic function by exploiting the recursive composition of the class. This procedure allows us to specify a conditional distribution referred to a set of value attributes (encapsulated or output nodes) given a set of input ones. In its simplest form, a class can be represented as a CPT. In addition, a class deals with a chain rule that is very similar to the one adopted by BNs (Pearl, 1988). This analogy is possible because a class over its value attributes and input nodes consists in a DAG whose nodes are the input, the output and the encapsulated attributes. When we are dealing with OOBNs, a stochastic function describes the relations between input attributes and the value ones. In order to associate a stochastic function with a specific object their type should match. Generally, we can notice that complex models deal with similar objects that share almost identical stochastic functions. Thanks to this observation, we can identify a generic class that can be used multiple times as shown in Figure 2.6. Furthermore, we can optimize how we visualize the class and make easier to analyze the OOBN by hiding the encapsulated nodes as shown in Figure 2.8. However, we remark that the structure referred to each time slice must be the same for every replicated fragment.

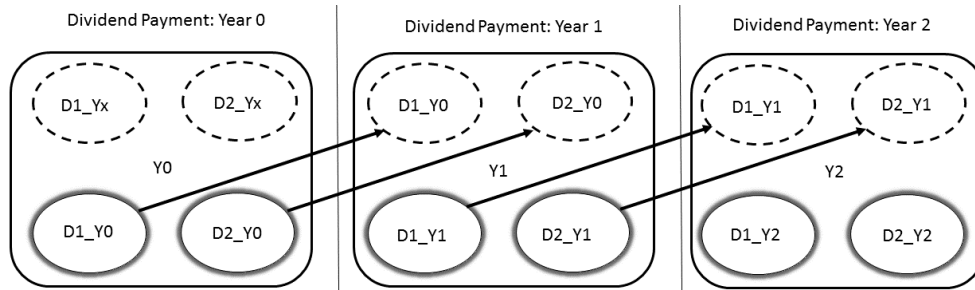


Figure 2.8: The AAPL dividend OOBN with the Encapsulated Nodes Hidden

2.10 Features and Elements of the Class

A class represents a fragment of an OOBN and it can contain instantiations belonging to other classes. It can include a sets of input (I), encapsulated (E) and output (O) nodes for whom the following statements hold:

- the identified categories of nodes (I, E and O) are pairwise disjoint
- an input node (I) cannot have parents in an instantiation k or a children outside k but it can have a referenced node outside k
- an output node (O) does not have any of its parents outside the instantiation k
- an encapsulated node (E) does not have any parent or children outside of the instantiation k

Moreover, when we deal with OOBNs we can distinguish two further types of node:

- Reference nodes: only input (I) and output (O) nodes can be labelled as reference nodes. For each reference node, we can have only one referenced node. They cannot have any parents but they can have children. A reference node do not have an encapsulated/internal node but it must be designated as input or output node.

- Real nodes: they represent a variable. For example, an encapsulated node can be labelled as a real node.

2.10.1 Links

Another basic element of the OOBNs is represented by the links connecting the nodes. We can identify two types of links:

- Directed links: this category represents the simple links that connect real nodes among themselves.
- Reference links: they are the ones connecting a reference node to its respective referenced one.

We underline that input nodes cannot be referenced nodes of a variable belonging to the same class, while they can be referenced nodes of variables belonging to other instance nodes. On the other hand, output nodes can be referenced nodes of encapsulated nodes belonging to the same class and for input nodes belonging to other instance nodes. It is not possible that two output nodes belonging to the same instantiation share the same referenced node. However, we should pay attention that the output nodes must have children outside of the instance node where they belong. Thanks to the graphical representation of the OOBNs, we can easily check this aspect. In conclusion, we remark that the encapsulated nodes can be used as referenced nodes only by the input ones.

2.11 Subclasses and Inheritance

One of the most important features of the OOBNs is represented by the possibility of building subclasses that inherit the properties of other existing classes. Because of this, classes are organized according to a is-a hierarchy (Koller and Pfeffer, 1997), which implies that the instance of a subclass is at the same time an instance of its parent class. The concept of hierarchy is another feature borrowed from the Object Oriented Programming, together with to the concepts of input, output and encapsulated nodes. More in detail, we can say that a subclass supplies all the outputs to the parent class but it cannot require any input attribute that is not supplied to the parent class. The concept of is-a hierarchy allow us to use a class as an approximated version of its subclasses. Furthermore, it allows to simplify specifications of similar classes and to organize our knowledge by following a top-down approach.

We should also consider that even if the class and subclasses describe the same type of object, the subclasses could be more detailed. We can consider the class referred to “Stocks” in general as an abstract class, that in its turn can have two subclasses called “Financial stocks” and “Utilities stocks”. As we can observe from this simple example, the subclasses add more details about the object. The depth of the information about an object can differ according to the needs of the user. For example, we can add more information to the Company Revenues type that is composed by the following intervals {100k-200k; 200k-250k; 250-300k, 300k+} and turn it into a more granular representation: {100k-150k; 150k-200k; 200k-250k, 250k-300k; 300k+}. The new type defined represents a subtype of Company Revenues. Then we can use this new type as a value type for the Company Revenues attribute of a new class Big Company that represents a subclass of the class Companies. The possibility of subclassing is a tangible proof of the hierarchical structure of the OOBNs. Furthermore, by observing

the model for a subclass we can notice that it has a lot in common with the model for a superclass. For example, the distribution over the *Financial Instruments* attribute is the same for the Stocks class and for the Financial Stocks class. The hierarchical relations among classes allow that subclasses can inherit parts from a superclass, whose structure must be a DAG.

2.12 Modularity

Another important feature of OOBNS is represented by the modularity, that allows solving the problems referred to models with numerous repetitive structures or too cluttered. We now assume that a fund manager holds in his portfolio three oil stocks: ExxonMobil (XOM), Petrobras (PBR) and Total (FP). Because of the recent oil price volatility, he wants to analyze each stock performance according to the information about OPEC oil output decisions and the workers union positions.

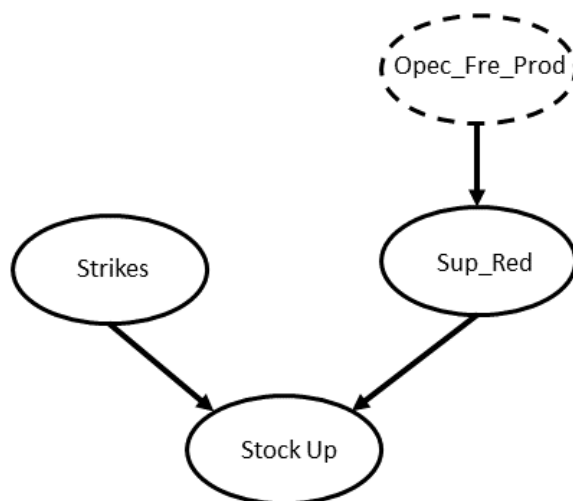


Figure 2.9: The network class for each oil stock

In Figure 2.9 we show the oil stocks network class. We assume that the prior referred to Strikes is fixed, while the amount of oil supply is determined by OPEC decisions. The variable “OPEC Freeze Production” represents an input attribute: if oil production is cut its supply decreases and this affects positively XOM, PBR and FP. The prior on “OPEC Freeze Production” (*Opec.Fre.Prod*) is $P(Opec.Fre.Prod) = (0.9, 0.1)$ while the conditional distribution of the variable “Supply Reduction” (*Sup_Red*) is shown in Table 2.8. In Figure 2.10 we show the network class for the oil stocks portfolio. As we

Table 2.6: The CPT for $P(Opec.Fre.Prod|Sup_Red)$

	Sup_Red	
OPEC_Fre_Prod	No	Yes
No	0.85	0.15
Yes	0.35	0.65

can observe, the oil stocks portfolio network consists of three different instantiations of each oil stock network. Moreover, the Opec_Fre_Prod input node referred to XOM, PBR and FP is bound to the Opec_Fre_Prod variable that belongs to the oil stocks portfolio class.

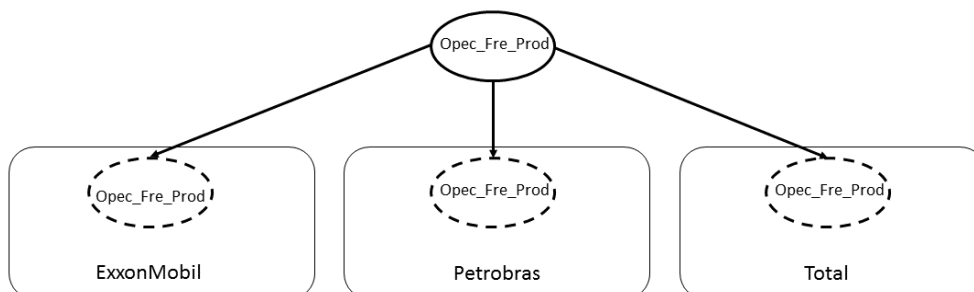


Figure 2.10: The oil stock portfolio OOBN

2.13 Making Inference in an OOBN

As outlined in the previous paragraphs, a class belonging to an OOBN has a structure comparable to a BN. Because of this analogy, we can apply on them the same inference algorithm used for BNs (Lauritzen and Spiegelhalter, 1988). The only way to make effectively inference in an OOBN is to group together simple objects belonging to the same area. In this way, it is easier for the user to identify among the encapsulated attributes the ones corresponding to input and output nodes. Furthermore, we can observe that the most of the nodes belonging to an object are encapsulated, and intuitively we can state that the variables that matter the most are the input and output made that communicate outside of the instantiation. This set of variables is enough to d-separate the encapsulated variables from the rest of the network. Since the set of input and output nodes d-separate the elements belonging to the object from the other variables of the model we can connect these two cliques in two junction trees and obtain a hierarchical model (Srinivas, 1994). In order to reduce the size of the set of input and output nodes, we can exploit the Multiply-Section Bayesian Network (MSBN) (Xiang et al., 1993). The construction procedure follows the one outlined before; however, it builds the junction trees in a way that allows decomposing the set of interface nodes. Therefore, we can build junction trees with smaller cliques and then make inference more efficiently. A MSBN divides all the variables belonging to a BN into non-disjoint subnets, however, the complexity of inference grows if the number of objects included in an OOBN increases. Even so, it is highly unlikely that an object defines an extremely large number of single objects. Considering the subclasses as an approximation of a class allows also adding more details to simple objects. In conclusion, we remember that OOBNs allow us to answer to several queries, including the ones involving encapsulated attributes, even if they are not visible outside the object and within the model.

Chapter 3

Bayesian Networks for Financial Markets Signals

3.1 Introduction

Fund managers activity is continuously evolving and adapting to financial markets. Independently from the time horizon, from the risk profile (the investor can be a risk lover or risk adverse) and from which asset class they are dealing with (i.e. stocks, bonds, derivatives, options. . .), market operators should constantly update their portfolio management techniques. Moreover, they have to deal with and elaborate a large amount of variables belonging to different areas: qualitative (i.e. based on articles from newspapers or TV interviews) or quantitative. The latter category includes macroeconomic data (i.e. GDP, inflation, and industrial production) and micro economic ones (i.e. single company information, M&A announcements, quarterly earnings results, insolvency problems. . .). Furthermore, one should also incorporate in the analysis more specific information deriving from the actors of financial community, such as brokers' report on listed companies or studies that provide economic estimates on a country or a specific sector. The main source for practitioners is represented by the data providers such as Bloomberg (www.bloomberg.com), Factset (www.factset.com) or Datastream (<http://financial.thomsonreuters.com>). Thanks to the internet diffusion, fund managers gather further information from articles and "rumors" reported and spread by specialized websites.

According to the framework outlined, the biggest challenge for an investor is to collect systematically various inputs and consider only the ones that have an impact on prices. All these assumptions imply that the market is a dynamic and chaotic system, at least from a mathematical perspective, because it crosses several points of unstable equilibrium. Because of this, one of the hardest task for a fund manager is to separate market rumors from the signal that indicates the trend.

The investment approaches developed by practitioners for interpreting market fluctuations are basically two: the fundamental and the quantitative one. The fundamental approach consists in an economic analysis of parameters and indicators regarding an asset valuation. According to the quantitative approach, the fund manager builds and uses mathematical tools by organizing his results in rankings, consisting in ordered group of variables with respect to a parameter, that allow him to determine if an asset is expensive or cheap, on relative or historical basis.

In addition, we have to consider that the framework is complicated further by investors simultaneous interaction because they can differ on the investment horizons and (or) risk/returns profiles. By assuming the existence of various kinds of investors, we imply that the same information has a different significance for each of them. This depends mainly on investors' utility function and expected return for a certain asset class. The interference among actors' objectives determine various adjustments and reactions on the market. This type of information is known as the "market flow". Furthermore, the interaction among agents determines the so-called "sentiment", a phenomenon representing the crowd psychology of the market that is reflected into price movements. The rising prices are an indication of the bullish sentiment (positive and optimistic) while decreasing prices reflect a bearish sentiment (negative and pessimistic). The sentiment pushes the market in a direction and it simply represents investor's common belief, known also as consensus. However, we underline that market sentiment is not always supported by fundamentals.

Sometimes markets complexity do not allow the agents to evaluate and interpret the existing relationships among variables by using the common statistical methods (i.e. regressions, descriptive statistics...) and the results obtained appear to be non-significant. In such a complex framework, it becomes fundamental to introduce new variables representing factors borrowed from behavioral finance. The objective of this psychology-based theory is to explain markets behavior and their anomalies by studying investors' rationality / irrationality. By including these new variables, we have now a wider and more detailed view of the market.

The objectives of our approach are to provide to fund managers a tool that allows analyzing different contents and their non-linear interactions and to prevent them from using methods and approaches selected arbitrarily and not based on a rigorous methodology.

Because of the large amount of variables available, fund managers' new challenge is represented by the integration in the same framework of variables belonging to different areas. For this reason we need new tools that are able to elaborate all these various aspects. The objective of the approach proposed in this work is to provide a new and wider perspective about financial markets and to find what "moves" assets prices. We remark that our aim is not only to provide ex-post indications but also to give operative signals that can be used in the future. By exploiting BNs we catch market common sense, that consists in what the financial community considers as the price drivers in a particular moment, but also which are the factors that affects market valuations in a certain period. Furthermore, BNs allow the user to identify unknown or unexpected relations among variables belonging to different areas by exploiting their complementarity. Because of this, they can be used as a useful support for fund managers and investment community as a whole. To our knowledge, this is the first time that these techniques are used for market signal detection.

The use of BNs is not limited to the integration and analysis of all the information available. They also allow simplifying the market framework by evidencing which are the market drivers and how they interact within the same framework. Furthermore, BNs allow simulating alternative scenarios. In this way, we observe how the markets behave when we introduce new evidences that modify the previous equilibrium. However, our aim is not only to identify market drivers but also to observe what happens when we simulate a new scenario by introducing new information in the BN.

Thanks to this model we are able to understand the relevance of a certain variable in the market, a task that cannot be easily done by exploiting the most common tools. BNs allow catching information that are very different among them, generally

observed separately and that are hard to include in the same framework. Furthermore, they allow us to observe how market equilibria and dynamics evolve by showing how the propagation flow changes among the variables. Thanks to this approach, it is possible to update market operators' knowledge that sometimes is too much anchored to past experiences. Fund managers bias in this case is represented by thinking that "if something in the past determined a market reaction, the same cause will determine this dynamic in the future".

3.2 Data Description

We examine the behavior of the S&P 500 in two different periods: the first one ranges from 1994 to 2003, while the second one from 2004 to 2015. We split the interval under analysis in two parts because the majority of the recent financial crisis (e.g. South-East Asia, Dot Com Bubble...) characterize the years between 1994 and 2003 and they deserve a stand-alone analysis, while in the period 2004-2015 we observed only one major event represented by the Great Depression. Moreover, this division allows to observe financial instruments evolution, a phenomenon mainly occurred in the last decade. The data are collected from the Bloomberg database, on a weekly basis. The objective of this work is to use BNs to identify the variables that affect the buy/sell signals provided by the largest American stock exchange.

There is no agreement among financial practitioners regarding which information should be considered when we analyze the S&P 500. In our analysis, we follow the approach proposed by Credit Suisse (Patel et al., 2011) that identifies the following categories of variables: value, growth, profitability, sentiment, momentum and technical analysis. In addition, we consider a further category that includes contrarian variables, which are generally used to provide indications against the price trend of an index: B_S_SPX and B_S_CRB. The first one refers to the S&P 500 while the second one to the Commodities Research Bureau (CRB) index. When the market is performing well, they indicate to reduce market exposure by providing a sell signal; on the other hand, when the market is underperforming, they suggest to buy. These categories contain exhaustively all the information needed for the analysis of a market index, both from a quantitative and a qualitative perspective.

3.2.1 The Variables of the Model

We now provide a brief definition of the variables belonging to the BN for financial markets signal detection.

Value Variables

- Price to Earnings Ratio (PE_RATIO): it is calculated by dividing the last index price by the S&P 500 Trailing Weighted EPS. The Trailing Weighted EPS (Market Convention index earnings), is calculated by summing up the equity member contributions (Trailing 12 Month Earnings Per Share of the member company equity multiplied by the number of shares of the equity in the index) and dividing the sum by the index divisor.
- Price to Sales (P_SALES): it consists in the ratio between the index price and the market cap weighted sum of the sales generated over one year by all the

listed companies. A low multiple, with respect to a benchmark, suggests that the index is undervalued.

- Price to Cash Flow (P_CF): it is calculated by dividing the index price by the market cap weighted sum of the operating cash flow referred to all the S&P 500 companies.
- Enterprise Value to Ebitda (EV_EBITDA): it is obtained by dividing the market cap weighted sum of the enterprise values referred to the companies listed in the S&P 500 by the market cap weighted sum of their Ebitda.
- Enterprise Value to Sales (EV_SALES): it is the ratio between the market cap weighted sum of the enterprise values referred to the companies listed in the S&P 500 and the market cap weighted sum of their Sales.
- Price to Book Value (P_BV): it is obtained by dividing the index price by the market cap weighted sum of the book values referred to the companies listed in the S&P 500. A low value, compared to a benchmark, indicates that the index is *cheap* but it can also indicate that the market is already pricing some negative issue.
- Sales/Ebitda per Share (SALES_PS; EBITDA_PS): it is calculated by dividing the market cap weighted sum of Sales/Ebitda generated by the companies listed in the S&P 500 by the total amount of shares outstanding in the index.
- Dividend Yield (DVD_YLD): it is the ratio between the market cap weighted sum of the yearly dividends per share and the last index price.

Growth Variables

- Sales/Ebitda/Earnings growth (SALES_GR; EBITDA_GR; EARN_GR): they indicate the market cap weighted sum of Sales/Ebitda/Earnings growth over the last year associated to the all the companies listed in the S&P 500.

Profitability Variables

- Ebitda Margin (EBITDA_MRG): it measures the operating profitability of all the companies listed in the S&P 500 and it is calculated by dividing the market cap weighted sum of all the Ebitda by the market cap weighted sum of all the revenues.
- BuyBack Yield (BB_YLD): it is the ratio between the market cap weighted amounts paid by all the companies belonging to the S&P 500 for purchasing their own shares over the index capitalization.

Market Sentiment Variables

- Implied Volatility / VIX (VOLA): it represents the estimated volatility derived from the option prices. It generally goes up when the market has negative expectations, while it decreases when investors are optimistic on future prices movements. According to this behavior, the volatile markets are considered riskier than low volatility ones.

- Put Call Volume Ratio (PC_RATIO): it is the ratio between the traded volumes of put to call options. When the amount of call options exceeds the volume of put options we have a bullish (optimistic) signal, while in the case the put options traded are more than the calls, it is a bearish (pessimistic) signal.

Momentum and Technical Analysis Variables

- Relative Strength Index (RSI): it compares the magnitude of recent gains to recent losses in order to determine whether an asset is overbought or oversold.
- Rate of Change (ROC): it measures the speed at which the index changes over a certain period.
- Index Value Above or Below its 52 weeks mean value (P_UP_DOWN): it measures whether the current market price is above or below its mean price calculated over the previous 52 weeks.

Market operators usually reason in terms of a discretized version of the previous variables. We consider the following classification. Variable P_UP_DOWN assumes two states: 0 when S&P 500 value is lower than its one-year average, 1 otherwise. Variables referring to value, growth, profitability, momentum and technical analysis and market sentiment are classified in three states. The scheme followed is based on how the values of each variable differ from its median (me). The label 1 is assigned to all values greater than $(me + \frac{1}{4} \sigma)$ (where σ is the standard deviation); label 2 refers to all values smaller than $(me - \frac{1}{4} \sigma)$. We label with 0 all remaining values.

For the contrarian variables, the labeling has been inverted. More precisely, the sell signal is given by state 2 (high value), while the buy signal is provided by state 1 (low value). State 0 provides a neutral indication.

3.3 The S&P 500 Bayesian Network

The BN structure is learned directly from the data by using the Hugin software (www.hugin.com). We considered the following two steps procedure. First, we ran the Chow-Liu algorithm (see Appendix 1, Section 10.2) to draw a tree structure maximizing the data likelihood and that is used as initial draft of the network (Chow and Liu, 1968). Then we run the NPC algorithm (see Appendix 1, Section 10.1) by Steck (2001), using as an initial constraint the tree obtained in the previous step with the addition of other restrictions deriving from our financial market knowledge. By exploiting the tree obtained in previous step, we reduce the dimensionality of the model space to explore with the NPC algorithm. Moreover, the NPC algorithm allows us to choose, among independence equivalent models, the most suitable for the problem under analysis. The model construction is completed by estimating the conditional probability tables (the parameters of the BN) from the data by exploiting the EM (Expectation-Maximization) algorithm (see Appendix 1, Section 11), whose version for BNs has been proposed by Lauritzen (1995). This algorithm is divided in two steps: the expectation step (E-step) and the maximization step (M-step). In the first one, we calculate the expected data frequencies given the current value of the parameters; in the second one, we maximize the log-likelihood of the parameters. These steps are alternated iteratively until a predefined criterion is satisfied.

We show in Figure 3.1 the network referred to 1994-2003 data. We used different colors in order to identify alternative groups of variables: *Green*, value; *Brown*, momentum

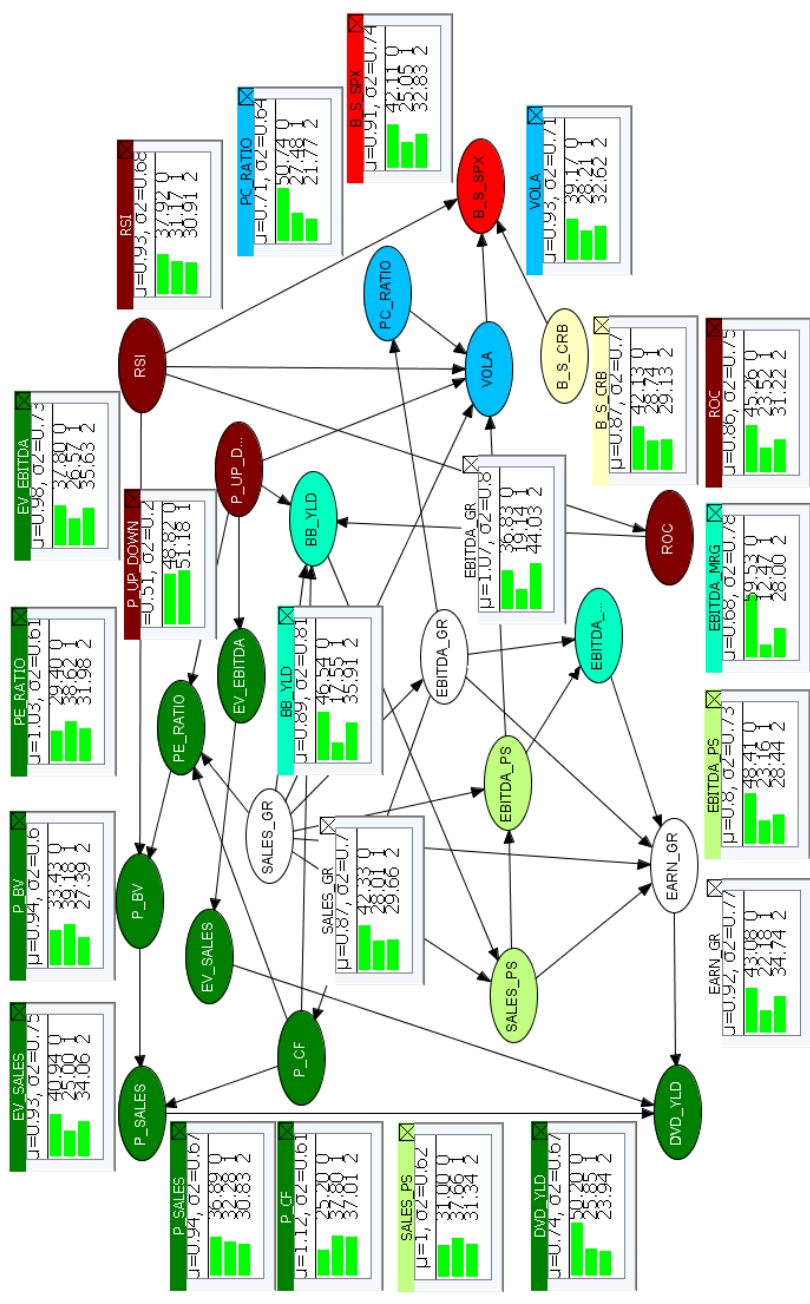


Figure 3.1: The 1994-2003 S&P 500 Network

and technical analysis; *Light Blue*, profitability; *Cyan*, sentiment; *White*, growth. Finally for the signal variables we used different colors: *Yellow* for the B.S.CRB and *Red* for the B.S.SPX.

The structural learning procedure allows us to observe interesting results on market dynamics that will be described in the following sections.

As expected, the VOLA variable plays an important role in the BN because it collects information from the neighbor ones and it is directly connected to the output node (B.S.SPX). Studies conducted by Goldman Sachs (Nielsen, 2011) on the equity market evidenced that market volatility represents a good risk measure in the short term. The PC_RATIO, a sentiment variable that provides indications on investors' mood, is indirectly connected to B.S.SPX through the VOLA node. A value above 1 reveals negative expectations (bearish), since the volume of put options traded is higher than the one referred to the call options. On the other hand, a value below 1 indicates positive expectations (bullish), since investors are buying more call options than put options. This confirms that investors are often driven by their expectations. The RSI represents another variable directly and indirectly connected through the VOLA node to the target one. This technical oscillator compares the magnitude of recent gains to recent losses, and it indicates if the market is overbought, which corresponds to a sell signal according to our contrarian approach, or oversold, a condition in which our investment strategy triggers a buy signal.

B.S.CRB influences directly the B.S.SPX but it does not have any inflowing arrow. This BN configuration suggests that B.S.CRB is only directly connected to B.S.SPX. According to the common knowledge, the PE_RATIO is the key variable for market valuation. From a visual inspection of the network, we can notice that this variable is not directly linked to B.S.SPX. Furthermore, it only mildly influences the output variable.

3.4 Simulating Market Evolution for the Period 1994-2003

Once the model has been estimated, the network can be used to study the effects of different market evolutions. The evidence propagation algorithm allows us to evaluate in a mouse-click time alternative scenarios. In this chapter we present only the results referring to two alternative market situations: a volatility shock and the role of P/E. The theme of market volatility is recently dominating the media headlines while an analysis on the P/E can provide interesting insights. P/E is considered by market operators as a key variable for any valuation based on fundamentals.

Our analysis is conducted by observing how the conditional probabilities changes after the introduction of a new information in the BN. This task is accomplished by comparing the value referred to each state before and after the simulation.

3.4.1 Scenario A: The Effects of High/Low Volatility Between 1994 and 2003

In this scenario, we study the effects of VOLA changes on the other market variables. We represent in Figure 3.2 and 3.3 our findings. We observe that the impact of the new information do not involve only the neighbor variables but all of them with the exception of B.S.CRB. This can be the proof that commodities have been used by the investors as a safe haven in the period 1994-2003: this is why volatility do not impact on

B.S.CRB. In line with the common financial knowledge, the momentum and technical analysis variables, RSI, ROC and P_UP_DOWN, are affected by a change in the sentiment variable VOLA. More precisely, we find that in the case of high VOLA (state 1) the probability referred to low RSI (state 2) increases from 30.91% to 52.06%, low (bearish) ROC (state 2) probability goes from 31.22% to 46.12%, while P_UP_DOWN state 0 jumps from 48.82% to 64.20%. When VOLA is low (state 2) the probability distribution corresponding to high (overbought) RSI (state 1) increases from 30.91% to 43.71%, the one referred to high (bullish) ROC (state 1) goes from 23.52% to 29.58%, while P_UP_DOWN state 1 probability jumps from 51.18% to 65%. When the VOLA is high, the probability referred to high (state 1) PE_RATIO (expensive market) increases from 38.62% to 55.71%, while in the low volatility scenario (state 2) the conditioned probability referred to low (state 2) PE_RATIO (cheap market) increases from 31.98% to 56.85%. The BN shows us that in the period 1994-2003 in correspondence of a high PE_RATIO, there is a higher probability of turbulences on the markets, while low volatility periods occurred when the market expressed low PE_RATIO. Another interesting result involving the valuation variables involves the DVD_YLD. When VOLA is high (state 1), the probability of low (state 2) DVD_YLD increases from 23.94% to 27.67%. In the scenario of low VOLA (state 2), the conditional probability referred to high DVD_YLD goes from 18.01% to 36.65%. This interesting result is in line with the financial knowledge but no indicator is currently able to catch and graphically show this result as our BN does. On the profitability side, we notice that the BB_YLD is sensible to VOLA changes. During high VOLA phases (state 1) the conditioned probability referred to high BB_YLD increases from 17.55% to 22.83%. On the other hand, when VOLA is low (state 2) the low BB_YLD conditioned probability (state 2) increases from 35.91% to 67.61%. These results confirm that companies wait for volatility spikes for accelerating their buy back operations. For what concerns the growth variables, it is interesting to observe that when VOLA is high (state 1) the probability referred to low EARN_GR (state 2) goes from 34.74% to 36.40%. This result is the evidence that we have a higher probability of experiencing volatile markets when earnings show a disappointing growth. In the case of low VOLA the BN do not provide clear signals, even if the probability corresponding to high EARN_GR increases from 22.18% to 32.83%. Finally, we analyze the impact of a VOLA change on B.S.SPX, the BN target variable. Generally, we observe that in the period 1994-2003 different VOLA scenarios do not provide clear buy or sell signals. When VOLA is high (state 1) the probability of a buy signal (state 1) for B.S.SPX increases from 25.05% to 37.26%, while the sell signal (state 2) probability increases from 32.83% to 39.71%. In a low VOLA scenario, the B.S.SPX conditioned probability referred to the buy signal (state 1) goes from 25.05% to 17.42% while the one referred to the sell signal (state 2) decreases from 32.83% to 23.86%.

3.4.2 Scenario B: The Effects of High/Low Price to Earnings Ratio Between 1994 and 2003

In the following scenario, we measure the effects of a change in PE_RATIO on the other variables. We represent in Figure 3.4 and 3.5 the most interesting findings. Our analysis allows us to observe that the RSI, ROC and B.S.CRB are invariant with respect to any change in PE_RATIO. This result represents the striking evidence of the irrationality of stock prices during the financial bubbles occurred between 1994 and 2003.

In the high PE_RATIO scenario (state 1) we observe a general impact on the other

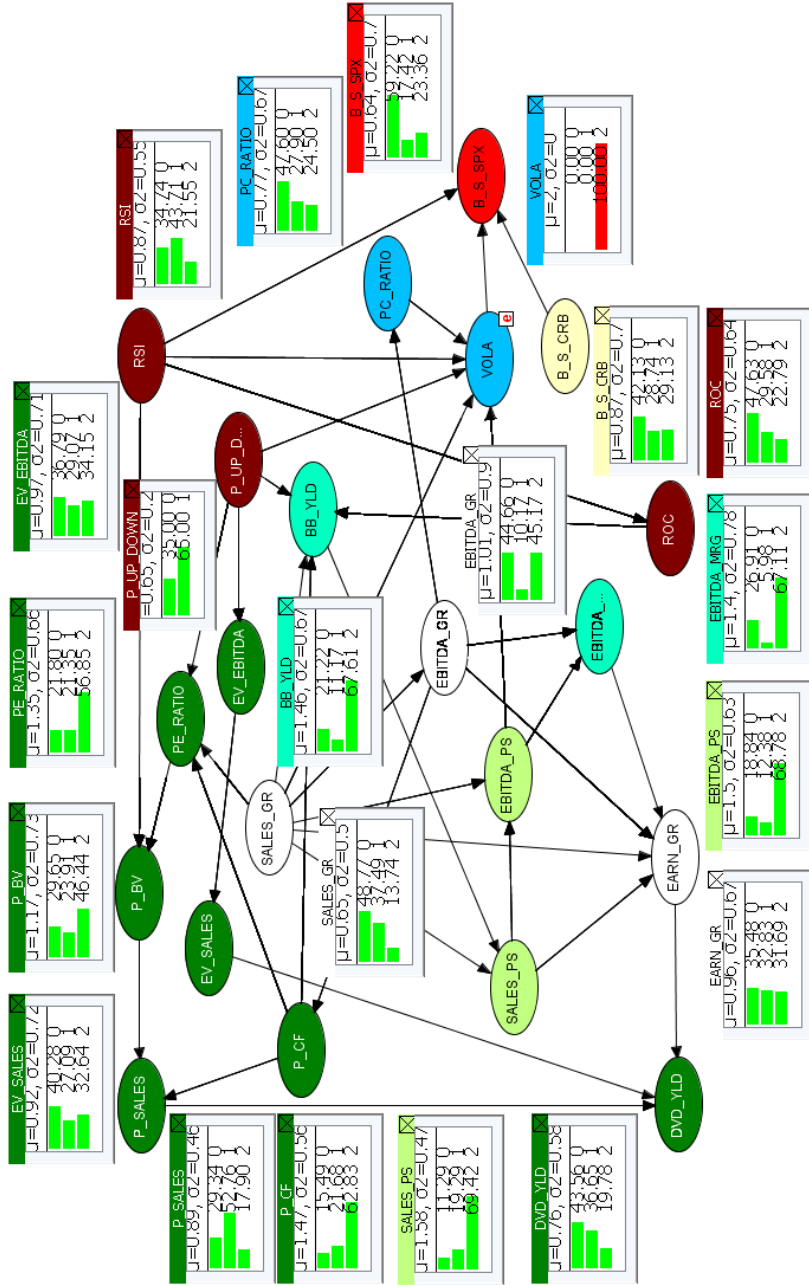


Figure 3.3: The Effects Referred to Low Volatility 1994-2003

fundamental analysis variables. Because of its relevance in the calculation of a stock total return, we focus on the signals provided by the DVD_YLD. When the PE_RATIO is high (state 1) the probability referred to low DVD_YLD increases from 23.94% to 37.37%. On the other hand, in the case of low PE_RATIO (state 2) the probability of incurring in a high DVD_YLD increases from 25.85% to 47.19%. These signals explain an important dynamic of fundamental analysis: if the price goes up too much, we have a high PE_RATIO and consequently a low DVD_YLD because this valuation metric is obtained by dividing the dividend per share by the stock price. For what concerns growth variables, it is interesting to observe that if the PE_RATIO is high (state 1) the probability of incurring in low EARN_GR increases from 34.74% to 56.88%. On the other hand, when market PE_RATIO is low (state 2) the conditioned probability referred to high EARN_GR goes from 22.18% to 32.37%. These dynamics reflect what we can expect when we observe an increase or a decrease in the earnings growth. According to the fact that the PE_RATIO is obtained by dividing the share price by the earnings per share, we can expect that when EARN_GR is high, the ratio is low and that if EARN_GR is low, the PE_RATIO is high. On the profitability side, we notice that when the PE_RATIO is high the probability of having a high (state 1) EBITDA_MRG increases from 12.47% to 26.75%. On the other hand, when the PE_RATIO is low (state 2) the conditioned probability referred to low EBITDA_MRG increases from 28% to 67.10%. This signal explains that investors buy stocks when EBITDA_MRG is high and consequently the stock price increases and pushes upwards the PE_RATIO. We now observe how a change in PE_RATIO affects the sentiment variables. When PE_RATIO is high (state 1) the PC_RATIO provides a bullish signal (state 2) because the probability referred to this state increases from 21.77% to 26.55%. This signal is in line with the investor behavior during financial bubbles: even if the market is expensive in terms of valuation, any decisions is driven by exuberance and the investors buy stocks without considering valuation metrics. On the other hand, when the market is cheap (state 1), low PE_RATIO, the PC_RATIO provide us a bearish signal (state 1) because the probability associated to this state goes from 27.48% to 28.78%. If we consider how a change in the PE_RATIO affects the VOLA, we observe that when PE_RATIO is high we have a higher probability of high VOLA (state 1) because its conditioned probability increases from 28.21% to 40.59%. When PE_RATIO is low (state 2) the probability referred to low VOLA increases from 32.62% to 57.99%. These results tell us that when stock market is expensive there is a higher probability of experiencing high volatility phases, while when the market is cheap the stock prices are less subject to sudden changes. Finally, we observe that the impact of a change in PE_RATIO on B_S_SPX is not significant and it does not provide a clear trading indication. This is another proof that, in the period 1994-2003, buy or sell signals provided by the S&P500 do not reflect market fundamentals. Then, when the financial bubble bursts, the prices converge to their fair value and the overvalued stocks crash.

3.5 Simulating Market Evolution for the Period 2004-2015

In Figure 3.6 we represent the network referred to the period 2004-2015. We use the same color classes used in Figure 3.1. The structural learning procedure allows us to observe that the learned structure is different from the one referred to 1994-2003 data; we describe in the following part the most important findings. The most striking

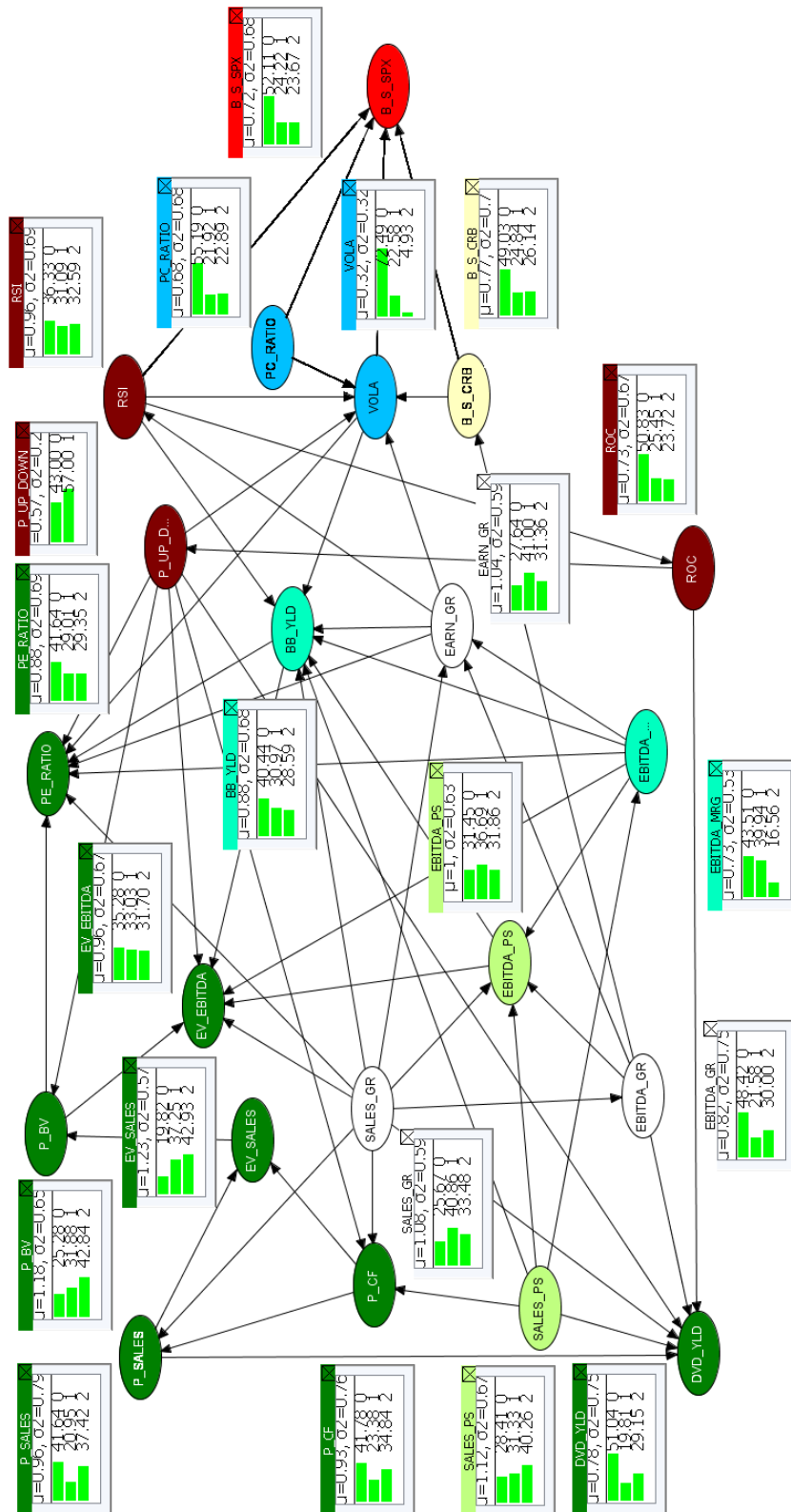


Figure 3.6: The 2004-2015 S&P 500 Network

difference is referred to the variable PE_RATIO. In this case it is directly connected to VOLA and it has only inflowing arrows. Another interesting result is referred to the VOLA variable that plays again an important role in the BN because it collects information from the neighbor variables and transfers it directly to B_S_SPX. We can also observe that VOLA in the 2004-2015 BN is directly connected to BB_YLD and to PE_RATIO. Furthermore, EBITDA_PS is not anymore directly connected to VOLA. The sentiment variable PC_RATIO in the 2004-2015 BN is directly and indirectly connected to B_S_SPX, while in the 1993-2003 BN it is just indirectly connected to the target variable through the VOLA node. As shown in the 1994-2003 BN, we observe that the RSI is connected directly and indirectly, through the VOLA node, to the target variable. We notice again that B_S_CRB influences directly the B_S_SPX. The difference with the 1994-2003 BN is that in addition B_S_CRB has an outflowing arrow towards the VOLA node and an inflowing one from EBITDA_GR. According to the results obtained from our simulation, the PE_RATIO continues to show some limit: any change in its states do not provide a clear buy or sell signal on the S&P 500. In addition, we discovered that even if the DVD_YLD is considered a relevant indicator by the investors, since it consists in an important portion of a stock total return, it affects only marginally VOLA and B_S_SPX variables. This finding is consistent with the results presented by Goldman Sachs (Nielsen, 2011) that evidences that the effect of dividend yield on short-term returns is not significant. On the other hand, a higher dividend yield should lift the average return in the long run.

3.6 Examination of Different Scenarios (2004-2015)

When the BN has been learned, we can perform simulations by exploiting the propagation algorithm and then evaluate different scenarios referred to the period 2004-2015. We propose again two alternative market situations: a volatility shock and the role of P/E. In order to perform a comprehensive analysis, we describe how the conditioned probabilities change after the introduction of new information and then we compare the results obtained with the ones associated to the period 1994-2003.

3.6.1 Scenario C: The Effects of High/Low Volatility Between 2004 and 2015

In this simulation, we consider the effect of a VOLA change on the other variables. The most relevant findings involve the following neighbor variables: PE_RATIO, BB_YLD, RSI, ROC, P_UP_DOWN, EARN_GR and B_S_SPX, and they are shown in Figure 3.7 and Figure 3.8.

Our analysis evidences that technical analysis and momentum variables are the most sensible to VOLA dynamics. This result is in line with common financial knowledge; a change in VOLA determines a market price fluctuation that is consequently reflected by RSI, ROC and P_UP_DOWN. More in detail, we observe that in the case of high VOLA (state 1) the probability distribution referred to low (oversold) RSI (state 2) jumps from 32.59% to 61.44%, low (bearish) ROC (state 2) probability increases from 23.71% to 43.99%, while P_UP_DOWN state 0 goes from 43% to 69.35%. On the other hand, low VOLA (state 2) triggers opposite responses in our BN. The probability distribution referred to high (overbought) RSI increases from 32.59% to 60.97%, the one corresponding to high (bullish) ROC jumps from 25.45% to 41.13%, while P_UP_DOWN state 1 goes from 67% to 76.72%.

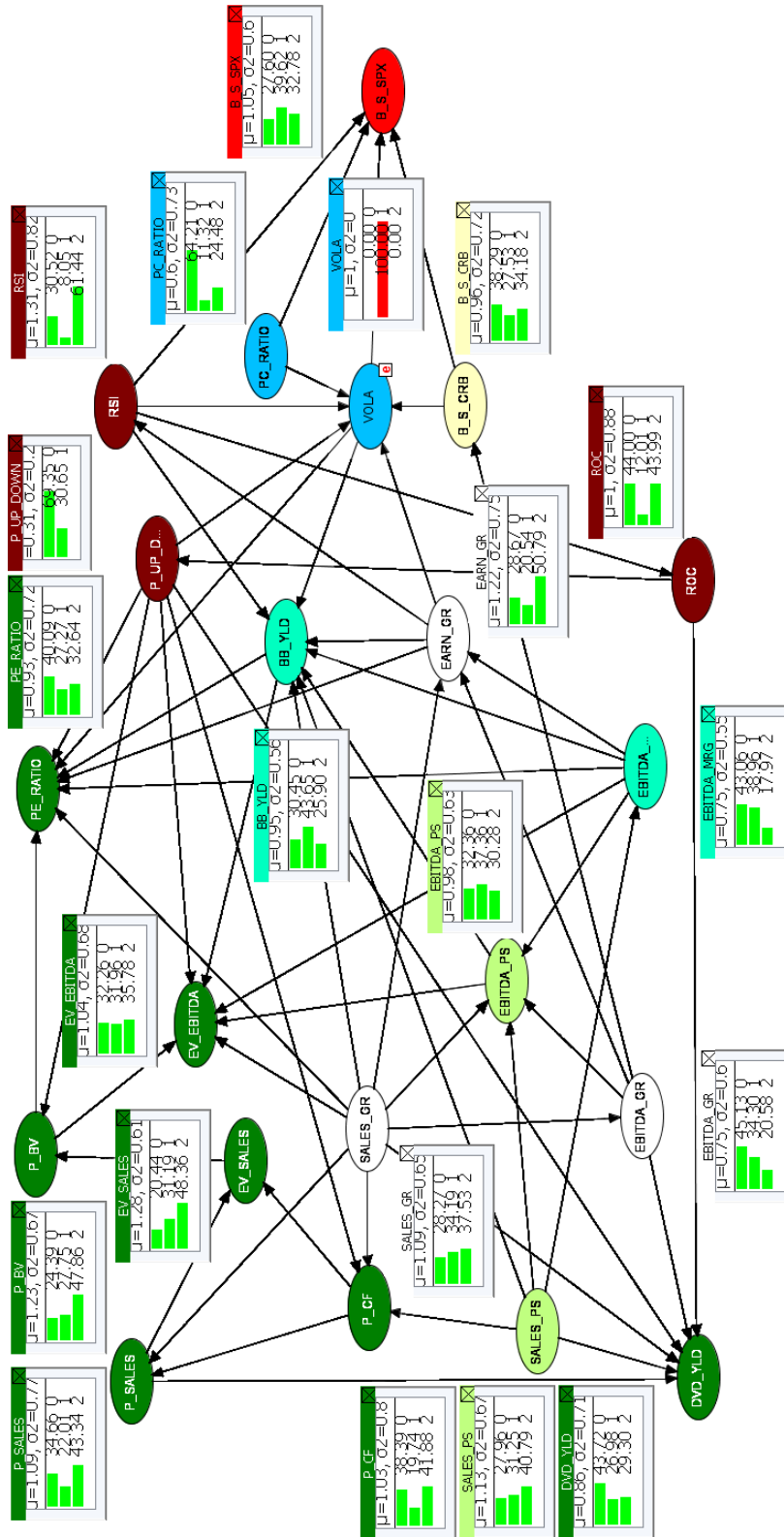


Figure 3.7: The Effects Referred to High Volatility (2004-2015)

The output of this simulation evidences a behavior in line with the result obtained for the period 1993-2004. The advantage of using the BN is that they allow us to observe market reactions that are not usually caught by the standard tools applied by institutional and retail investors, such as regressions or scorecards.

When VOLA is high, the probability associated to low (bullish) PC_RATIO goes from 22.89% to 24.48%. This change implies that there is a higher probability to experience high volatility when the market has positive expectations. In the low VOLA scenario the probability corresponding to state 1 is the highest (bearish) compared to the one referred to state 2, even if its value decreases from 21.92% to 20.93%. This output evidences that markets have negative expectations when VOLA is low.

For what concerns profitability, we observe that the BB_YLD is sensible to any VOLA increase. The probability referred to a high BB_YLD (state 1) increases from 30.97% to 43.65% when VOLA is high. In low VOLA phases (state 2), market prices are more stable and the companies slow down their repurchase activity, waiting for buy opportunities in more volatile periods. The strategy adopted by the companies is caught by the BN: the probability referred to high BB_YLD decreases from 30.97% to 29.68%, while the one referred to low BB_YLD goes from 28.59% to 23.13%.

Other interesting results are referred to the growth variable EARN_GR. In the high VOLA scenario the probability referred to low EARN_GR (disappointing growth) increases from 31.36% to 50.79%, revealing us that the S&P 500 is more subject to sell-offs when listed companies earnings are not growing. During low VOLA phases the BN does not provide clear indications on EARN_GR: the probability referred to high EARN_GR (state 1) increases from 41% to 41.97%, while the one referred to low EARN_GR (state 2) goes from 31.36% to 43.28%.

Our analysis provides also information on how a change in VOLA is reflected on the PE_RATIO states. In a high VOLA (state 1) scenario the probability referred to low PE_RATIO (state 2) increases from 29.35% to 32.64%, while in the case of low VOLA (state 2) the probability of high PE_RATIO (state 1) increases from 29.01% to 31.72%. In this case, BN provides a new insight on how to interpret market dynamics. Finally we observe that a change in VOLA impacts also the contrarian variables B_S_CRB and B_S_SPX.

In the case of high VOLA (state 1) we observe that B_S_CRB provides a sell signal because the state 2 probability increases from 26.14% to 34.18%, while B_S_SPX shows a buy signal because its state 1 probability goes from 24.22% to 39.62%. When VOLA is low (state 2) B_S_CRB still suggests a sell signal because the state 2 probability increases from 26.14% to 29.76%, while B_S_SPX provides a sell signal because the probability associated to state 2 decreases from 23.67% to 16.36% but it is still the higher than the one referred to state 1.

3.6.2 Scenario D: The Effects of High/Low Price to Earnings Ratio Between 2004 and 2015

In the following scenario, we measure how a change in PE_RATIO affects the other variables. We analyze the findings of this simulation in Figure 3.9 and 3.10.

According to our results and in contrast with the common financial belief, we observe that a change in the PE_RATIO, have a sensible impact only on the profitability variable BB_YLD. The impact of PE_RATIO on BB_YLD confirms that the listed companies repurchase their own shares according to their P/E multiple. In the high PE_RATIO scenario (state 1) the conditioned probability referred to low BB_YLD (state 2) increases from 28.59% to 43.39%, evidencing that the buyback operations

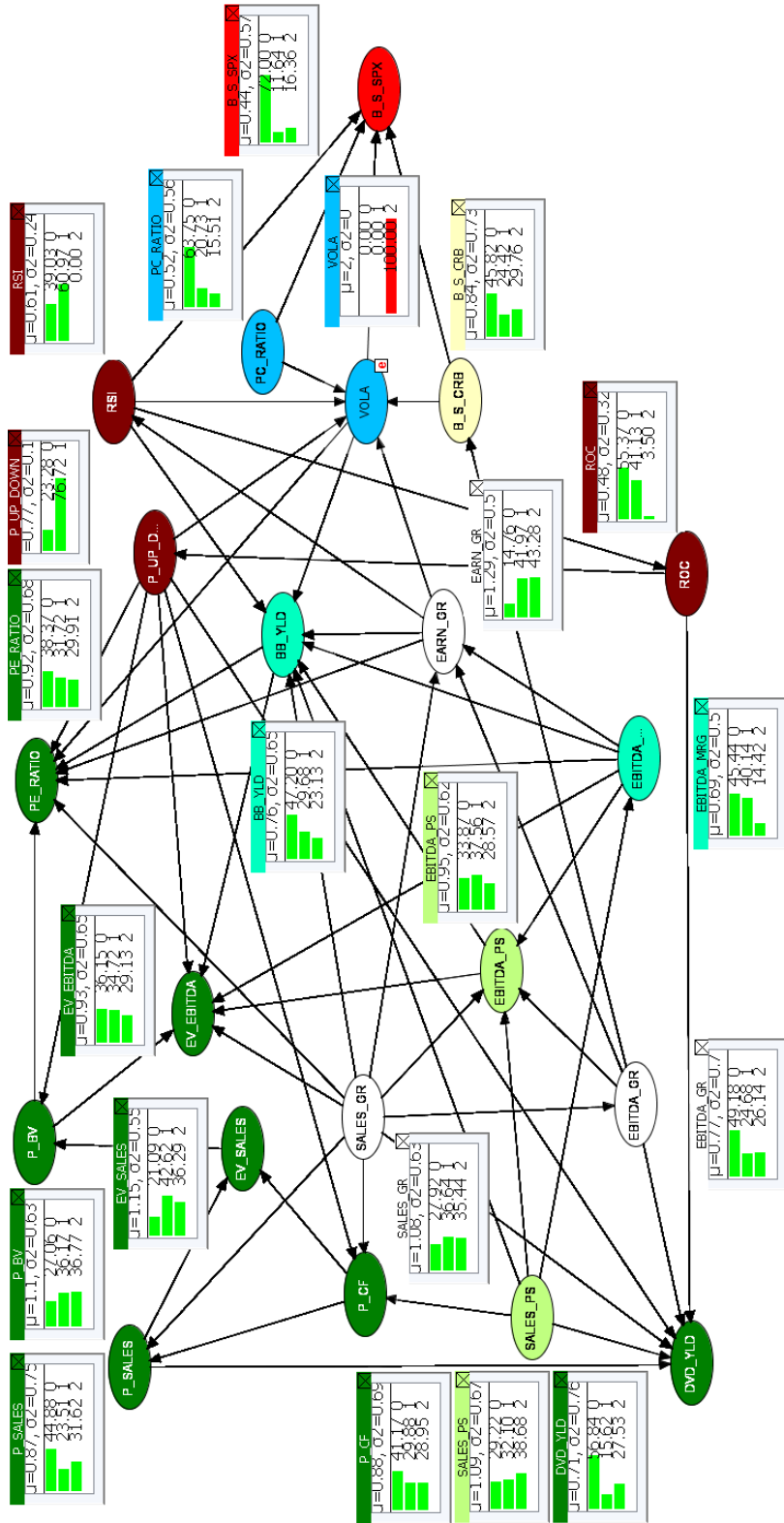


Figure 3.8: The Effects Referred to Low Volatility (2004-2015)

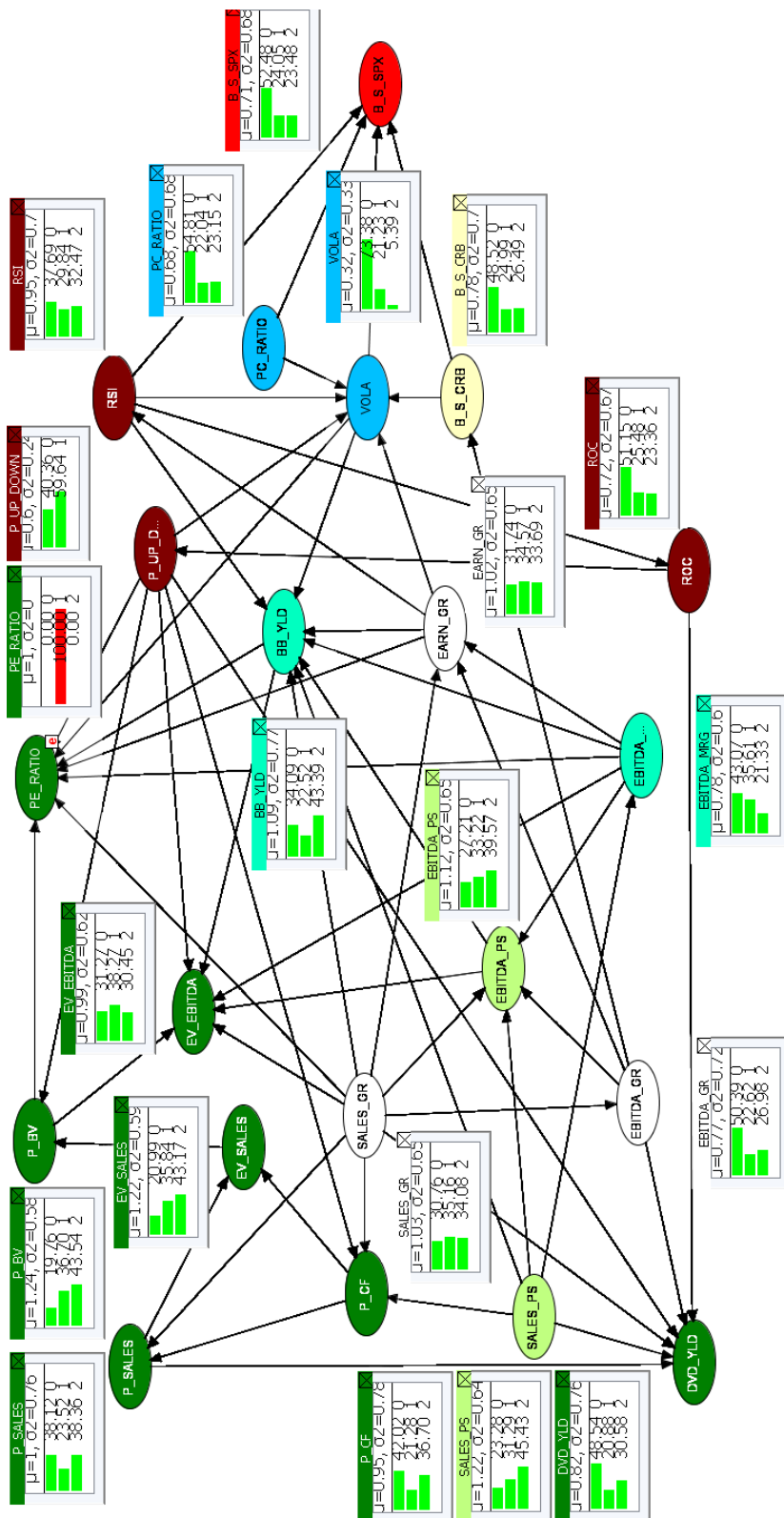


Figure 3.9: The Effects Referred to High Price to Earnings Ratio 2004-2015

slow down when the market PE_RATIO is high (expensive). On the other hand, when the market is “cheap”, low PE_RATIO low (state 1), the probability referred to high BB_YLD goes from 30.97% to 31.61%.

We observe that the PE_RATIO does not provide clearly a buy or a sell signal because it mildly affects the BN variables, with the exception of BB_YLD. This finding is in contrast with the common financial belief because market operators look at the PE_RATIO as a key valuation metric.

3.7 Concluding Remarks

The results illustrated in this chapter underline the potentiality of BNs in the analysis of financial markets. According to the results, we evidence that market efficiency does not always depend on financial news but also on information coming from other areas. By exploiting BN potentiality, we can observe in a mouse-click time how new information affects market dynamics. This approach allows us to reveal dependences that otherwise would not be evidenced by the common tools used every day by financial operators.

For example, the results provided by the 1994-2003 BN evidence the irrationality of the markets during financial bubbles. These simulations show that any PE_RATIO change do not affect RSI and ROC, variables referred to momentum and technical analysis, and B_S_CRB, the commodities index signal. Thanks to our BN we are able to observe this independence that otherwise we cannot read. In addition, the extension of our analysis allows us to observe that B_S_CRB is not affected by any VOLA change, a dynamic that reveals that between 1994 and 2003 commodities were considered as a “safe haven” by investors. Furthermore, our results suggest that when VOLA is high there is a higher probability of experiencing a high PE_RATIO, while when VOLA is low the probability of low PE_RATIO is the highest. On the other hand, we observe that in the period 2004-2015 the P/E ratio cannot anymore be considered as the key valuation metric, because we have stronger signals originated by other variables. In order to fully understand market equilibrium the investors should use both economic variables and sentiment ones. Our results suggest that in the period 2004-2015 market signals are principally driven by sentiment variables, these in their turn affect also the valuation variables such as the P/ E ratio. Consequently, P/E impacts on the companies buy back that accelerates when the market is “cheap” (low P/E) or slows down when the stocks are “expensive” (high P/E).

These results are in contrast with the findings obtained by analyzing the BN learned from the 1994-2003 data where in correspondence of high VOLA we have higher probability of high PE_RATIO and when the VOLA is low we observe a higher probability of low PE_RATIO. This is the evidence that the market equilibrium and drivers changed across the last 21 years. When we are looking for a market signal we should not only consider financial factors; our analysis reveals that market equilibrium have changed compared to conventional thinking and also behavioral factors should be taken into consideration. As expected, the BNs learned from different data, 1994-2003 and 2004-2015, show different market behaviors since market drivers changed across time.

Chapter 4

OOBNs for S&P 500

As underlined previously, BNs evidence some limit in dealing with complex networks, the ones composed by a large number of variables. In order to overcome this issue, we adopt the OOBNs for conducting an analysis on S&P 500. Thanks to this more advanced representation, we can add more detail to our analysis on equity markets buy/sell signals detection without compromising the model efficiency.

4.1 An Application to the S&P 500 Index

As observed in Chapter 3, financial markets analysis is a complex task because there is a large number of variables and indicators that drive securities trends and prices. Moreover, we have noticed that in some occasions the tools generally used by practitioners fail in providing a wide and comprehensive view of the markets.

In order to include in the same model variables belonging to micro, macro, sentiment and technical analysis areas we propose to use the Object Oriented Bayesian Networks (Koller and Pfeffer, 1997). Through an application to the most capitalized American equity market, we show how OOBNs support efficiently an investor by providing information on which variable influences more the S&P 500 dynamics. Thanks to OOBNs, we are able to represent in the same model aspects that are hard to analyze at the same time by performing regressions, statistical analysis or by consulting the most used data providers such as Bloomberg, Factset or Datastream. Thanks to OOBNs, a fund manager identifies the market drivers and how the environment is evolving. We underline that the blocks in our model correspond to a specific Bayesian Network (BN) and the user can perform simulations on each area as shown in Chapter 3. Moreover, OOBNs allow increasing the efficiency of inference process (Koller and Pfeffer, 1997), notwithstanding the complexity of the structure inside each instance node. This result is possible because the encapsulated nodes are conditionally independent from the rest of the OOBN given the input and the output ones. The OOBNs are ideal to deal with such frameworks because they allow including a large number of variables in the same model but they also make easier to interpret the results thanks to their graphical output and the instance nodes features. The variables used in this experiment have been downloaded from the Bloomberg database and they range from January 1996 (when available, otherwise we use the first available data) to May 2016, on monthly basis. The most of them have been already introduced in Chapter 3, while the new ones will be defined in Section 4.2. They are discretized by following the same methodology adopted in the BNs experiment (see Chapter 3, Section 2). The only exception is

represented by MACD_DIF and CROSS_MA20_50, which have been discretized in two states because investors generally observe if they are in positive (state 1) or negative (state 0) territory. Overall, the model has been built by learning a BN for each single area (macro, micro, technical analysis, market sentiment) (e.g. Patel et al. 2011) and by including them in the same OOBN. These categories contain all the information needed for the S&P 500 analysis, both from a quantitative and a qualitative perspective. We now list all the variables grouped by area of belonging:

- Market Sentiment: Vix (VIX); Put Call Volume Ratio (Pc_Ratio); Implied/Realized Volatility Spread (Vola_Spread).
- Technical Analysis: Relative Strength Index (RSI); Rate of Change (ROC); MACD (Macd_Dif); 20-50 Periods Moving Averages Cross (Cross_MA20_50).
- Macroeconomics: Unemployment rate (Unemploy); ISM Manufacturing Index (ISM_Man); Consumer Price Index (CPI); American GDP (USA_Gdp); DXY Index (DXY_Index); Oil; Wheat; Copper; Gold.
- Micro Dimension: Dividend Yield (Dvd_Yld); Price to Earnings Ratio (PE_Ratio); Price to Sales (P_S); Price to Cash Flow (P_Cf); Enterprise Value to Ebitda (Ev_Ebitda); Ebitda Margin; Profit Margin (Profit_Mrg); Price to Book Value (P_Bv); Sales Growth (Sales_G); EBITDA Growth (Ebitda_G); Earnings Growth (Earn_G); Sales per Share (Sales_S); Ebitda per Share (Ebitda_S); Earnings per Share (Eps_S); Return on Equity (ROE).

4.2 The Variables of the OOBN

We now provide a brief definition of the variables that we have added to the ones already used in the experiment in Chapter 3.

Market Sentiment Area

- Implied/Realized Volatility Spread: it is the difference between the implied volatility (derived from the options prices) and the realized volatility (calculated as the standard deviation of the log returns of the index). The realized volatility, also known as the historical volatility, is calculated as the standard deviation of market returns over a certain period.

Technical Analysis Area

- MACD: it is one of the most used momentum indicators. MACD is calculated by subtracting the 12 day exponential moving average referred to the market price from its 26 day exponential moving average. Then, we use a 9 day moving average as a “signal line”: if it crosses from below to above the longer term moving average the signal is bullish. On the other hand, if it crosses from above to below the slowest moving average the movement triggers a sell signal.
- 20-50 Periods Moving Averages Cross: we have a “golden” cross when a short-term moving average (i.e. 20 days moving average), crosses from below to above a long-term moving average, (i.e. 50 days moving average). This movement is considered a bullish (optimistic) signal. On the other hand, if the short-term moving average crosses from above to below the long-term moving average we

have a “death” cross, that is considered a bearish (pessimistic) signal for the prices.

Macroeconomic Area Area

- **Unemployment rate:** it represents the percentage of the total unemployed labor force that is actively looking for a job and willing to work.
- **ISM Manufacturing Index:** this index is based on surveys conducted by the Institute of Supply Management (ISM) on more than 300 manufacturing firms. It monitors employment, production inventories, new orders and supplier deliveries.
- **Consumer Price Index:** it measures the price level of a basket of goods and services that are generally purchased by households. By observing its changes we can monitor inflation.
- **Gross Domestic Product:** it represents the monetary value of all the finished goods and services produced in a country in a defined period (generally one year). It includes private and public consumption, government expenditure, investments and the trade balance. The GDP is used as a measure of the economic activity of a nation.
- **DXY Index:** this index compares the value of the US Dollar to a basket of foreign currencies. More in detail DXY index is calculated as the weighted geometric mean of the value of the USD against the following currencies: Euro (57.6% weight), Yen (13.6% weight), Pound (11.9% weight), Canadian Dollar (9.1% weight), Swedish Krona (4.2% weight), Swiss Franc (3.6% weight). DXY goes up when the US Dollar strengthens.
- **Gold:** it is a precious metal and the most popular one used as investment. Investors uses it as a safe haven during high volatility periods or when they fear rising inflation.
- **Oil:** Crude oil is an unrefined petroleum product. Its price per barrel heavily depends on the demand for it and on the evolution of its reserves. Crude oil oversupply or a decreasing demand can be a signal of a slowdown in the manufacturing sector.
- **Wheat:** this agricultural commodity is generally used to hedge against inflation. Its price is sensible to weather conditions that can favor or put in danger the crop. Wheat price depends also on global economy health.
- **Copper:** it is one of the most versatile industrial metals. Even if the copper shares many similarities with the most precious metals, such as gold and silver, it is extremely cheap compared to them because of its wide applications (i.e. wires, pipes, and telecommunications ...).

Microeconomic Dimension Area

- **Return on Equity (ROE):** it is obtained by dividing the market cap weighted sum of the net incomes referred to all the companies listed in the S&P 500 by the market cap weighted sum of their shareholders' equity.

- Earnings per Share: it is the ratio between the market cap weighted sum of the earnings generated by the companies listed in the S&P 500 and the total amount of shares outstanding in the index.
- Profit Margin: it is obtained by dividing the market cap weighted sum of the net incomes referred to all the companies listed in the S&P 500 by the market cap weighted sum of their revenues.

The Response Variable: B_S_SPX

The variable B_S_SPX, provides buy/sell indications on the American equity market according to a *contrarian* investment approach. It has been built by following the same procedure outlined in Chapter 3 Section 2.1.

4.3 The OOBN for Market Signals Detection

In Figure 4.1, we represent our OOBN that is composed by several instance nodes corresponding to different economic areas: Macroeconomics, Micro_Dimension, Technical_Analysis and Market Sentiment. In Section 4.4 we analyze in detail how the blocks have been built and their inner dynamics.

From a visual inspection, we notice that the areas communicate among them through the output nodes: the Macroeconomics block is connected to the Micro_Dimension node through the output node USA_Gdp; the Technical_Analysis area communicates with the Micro_Dimension via Cross_MA20_50; Market_Sentiment instance node is connected to the Technical_Analysis area by the Vix. On their turn, all these areas are connected to the response variable B_S_SPX through the following output nodes: Dxy_Index, PE_ratio, RSI and Vola_Spread. In Section 4.5 we perform some relevant simulation that illustrates OOBNs potentiality in finance.

4.4 Macro, Micro, Technical Analysis and Market Sentiment Classes

We estimated the structure of each area directly from the data by following the same two-step approach (Chow-Liu algorithm and NPC algorithm) adopted in Chapter 3. Then we perform some simulation on each instance node. As already observed with BNs, a financial expert could expect some of the results obtained, however, the model provides additional insights thanks to the large amount of information included that help investors in their decision making process. The simulations performed in this chapter represent relevant scenarios that an investor can face during his daily activity.

4.4.1 Market Sentiment Area

This section deals with the variables that provide indications on the market sentiment, an aspect reflecting market operators' attitude and behavior, see Figure 4.2. More precisely, we refer to the effect of the crowd psychology on buy/sell activity and price movements. If investors feel optimistic (bullish) the prices are supposed to go up, on the contrary if they are pessimistic (bearish) the pressure is downwards. In this block, we identify two output attributes: Vola_Spread and Vix. According to our prior financial markets knowledge, the former is directly connected to the B_S_SPX response

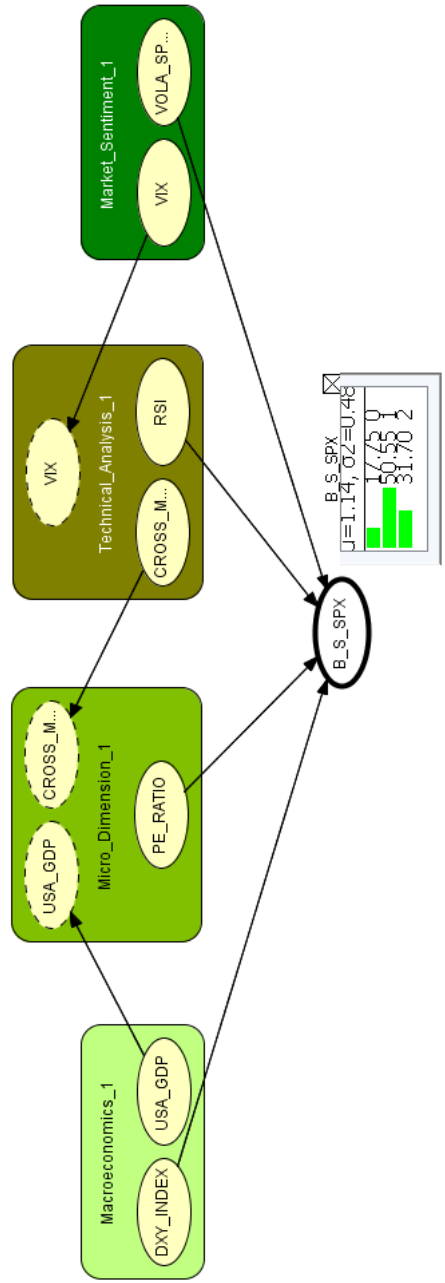


Figure 4.1: The OBN for S&P 500 signals detection

variable, while the latter influences the technical analysis area. The monitor referred to each node shows the marginal probability referred to its states expressed in percentage. The network structure shows that Vix and Pc_Ratio directly influence Vola_Spread.

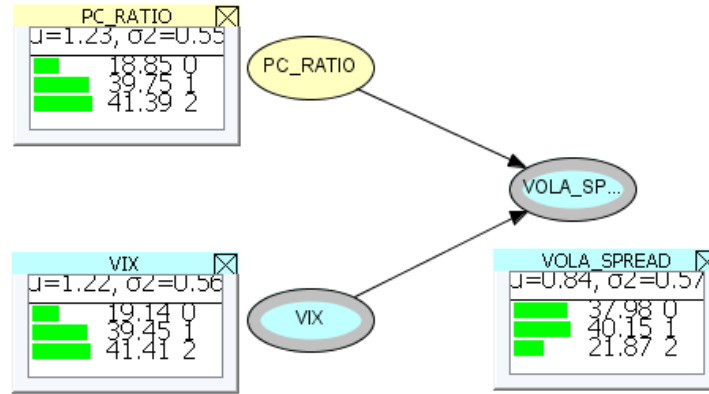


Figure 4.2: The Market Sentiment OOBN

The marginal probabilities indicate that in 40.15% of the cases Vola_Spread is in the highest range. Thanks to the information propagation algorithm, we can now perform some simulation in order to observe which variable influences more the Vola_Spread.

Scenario A: The Impact of Vix on the Market Sentiment Area

We now show in Figure 4.3 that Vix has a strong influence on Vola_Spread. In the case volatility is equal to 1, Vola_Spread state 1 jumps from 40.15% to 78.05%. On the other hand, if the volatility is equal to 2, see Figure 4.4, Vola_Spread state 2 increases by 17.4%. Pc_Ratio do not reflect any change of Vix, showing that the hedging strategies involving the purchase of options contract do not depend on the implied volatility.

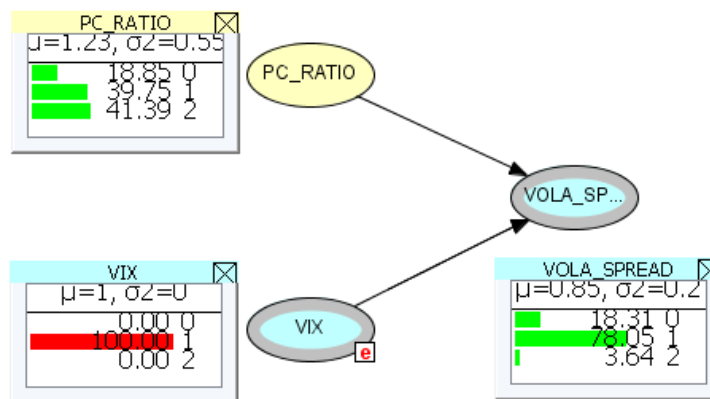


Figure 4.3: Market Sentiment OOBN: High VIX Scenario

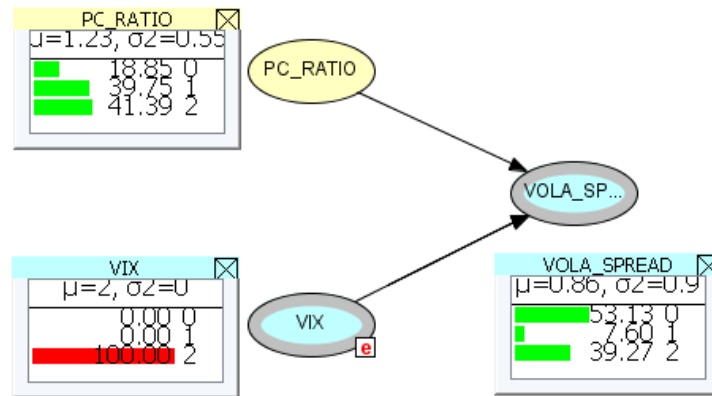


Figure 4.4: Market Sentiment OOBN: Low VIX Scenario

Scenario B: The Impact of Put/Call Ratio on the Market Sentiment Area

We now analyze the impact of Pc.Ratio on Vola.Spread, see Figure 4.5 and 4.6. Thanks to our simulation we observe that investors bullishness, incorporated by Pc.Ratio state 2, or bearishness, Pc.Ratio state 1, have a smaller impact on Vix.Spread than Vix. These indications and insights cannot be observed with the tools generally adopted by practitioners. Thanks to our simulation, we notice that investors' bearishness (state 1) determines a small reduction of the probability associated to high Vola.Spread, whose probability decreases from 40.15% to 39.46%. Despite of this reduction, high Vola.Spread is still the most likely outcome. On the other hand, we observe that a bullish Pc.Ratio, which corresponds to state 2, makes Vola.Spread state 0 to increase. These results reveal that the Vix has a stronger impact on Vola.Spread than the Pc.Ratio

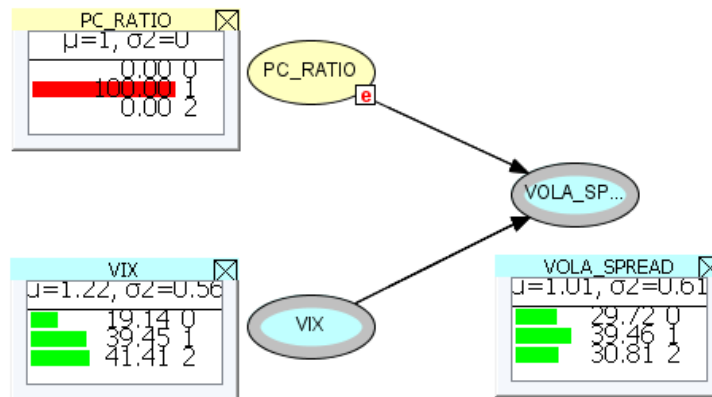


Figure 4.5: Market Sentiment OOBN: High Put Call Ratio Scenario

Scenario C: The Impact of Volatility Spread on the Market Sentiment Area

We conclude the analysis of this area by simulating a scenario of high and low Vola.Spread. As shown in Figure 4.7 in a low Vola.Spread scenario (state 2), Vix is

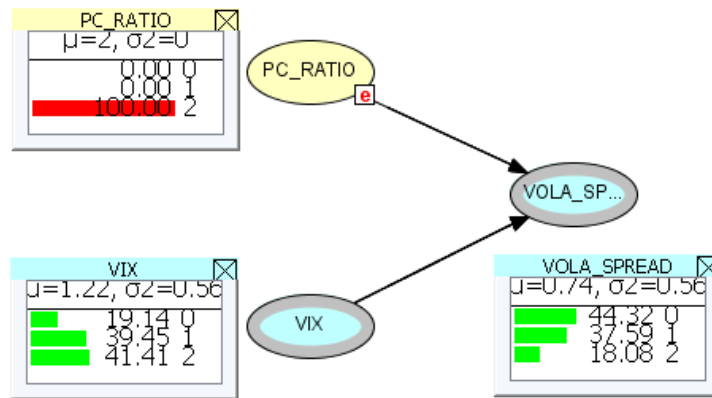


Figure 4.6: Market Sentiment OOBN: Low Put Call Ratio Scenario

more likely to be high because state 1 probability increases from 41.41% to 76.70%, while the Pc.Ratio do not provide a clear indication because both state 1 and 2 are close to the 39%. In a high Vola.Spread scenario, Figure 4.8, our model reveals a higher probability for low Vix, state 2, while the Pc.Ratio indicates a higher bearish likelihood (state 1), whose probability increases by 16.3%. This dynamic shows the investors' propensity to protect their gains through put options during low volatility phases. In conclusion, we can state that our results show a strong dependence between Vix and Vola.Spread.

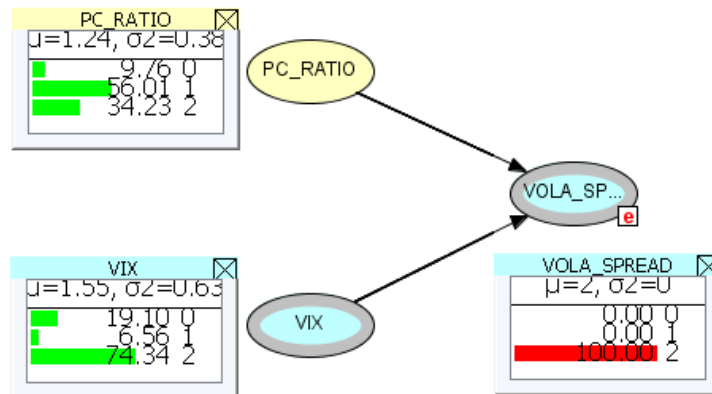


Figure 4.7: Market Sentiment OOBN: Low Volatility Spread Scenario

4.4.2 Technical Analysis Area

This block includes the most frequently used technical analysis oscillators. These indicators are based on the analysis of past prices with the objective of predicting their future direction and without considering the fair value of securities. More in detail, this discipline observes how the supply and demand dynamics define the financial markets trends (upwards or downwards). The instance node referred to the Technical Analysis area, see Figure 4.9. The BN structure shows that Vix, the input attribute belonging to the market sentiment area, is directly connected to Cross_MA20.50 and to

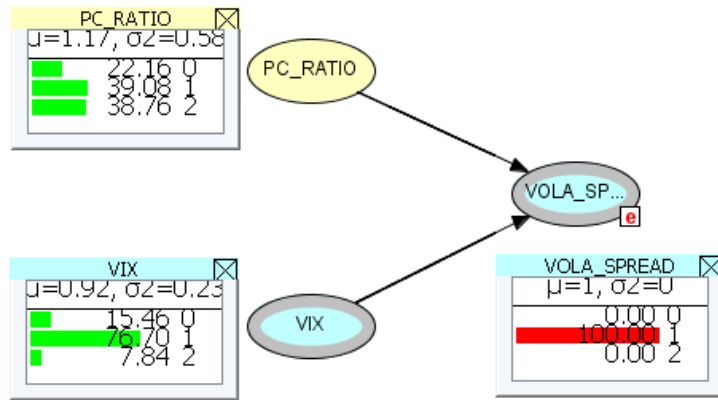


Figure 4.8: Market Sentiment OOBN: High Volatility Spread Scenario

Roc, that in its turn has an edge towards the RSI and the Macd_Dif. The designated output nodes in this block are Cross_MA20.50, that influences the “price multiples” in the Micro Dimension area (see Section 4.4.4), and the RSI, that is connected to the B.S.SPX response variable. We now perform some simulation in order to access the dependencies among the variables belonging to the Technical Analysis area.

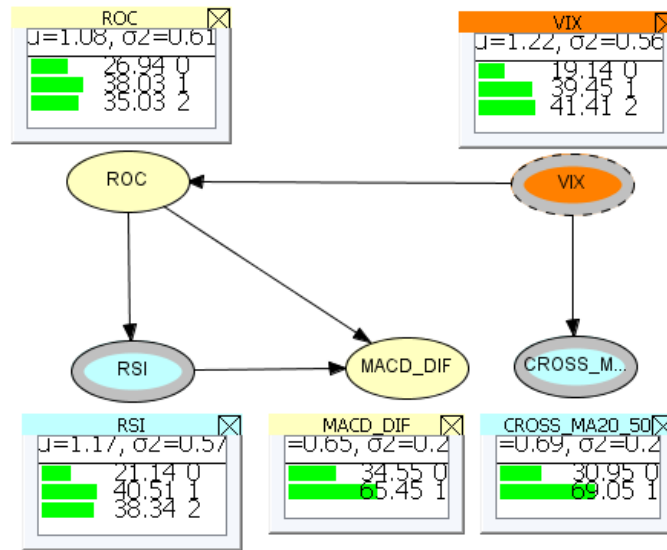


Figure 4.9: The Technical Analysis OOBN

Scenario A: The Impact of ROC on the Technical Analysis Area

In Figure 4.10, we simulate the impact of bullish Roc, state 1, on the other nodes. In particular, we observe that the marginal probability associated to the overbought Rsi, state 1, jumps from 40.51% to 70.67%. Furthermore, we notice that the probability referred to a positive Macd_Dif increases by 31.4%. At the same time, we register and increase of low Vix probability, that goes from 41.41% to 48.64%.

On the other hand, when we simulate a bearish Roc scenario, Figure 4.11, the marginal probability associated to the oversold Rsi, goes from 38.34% to 79.56% while Macd_Dif state 0 jumps from 34.55% to 78.34%. We conclude our analysis by observing that in this scenario there is a higher probability to experience high volatility because Vix state 1 increases by 25.2%.

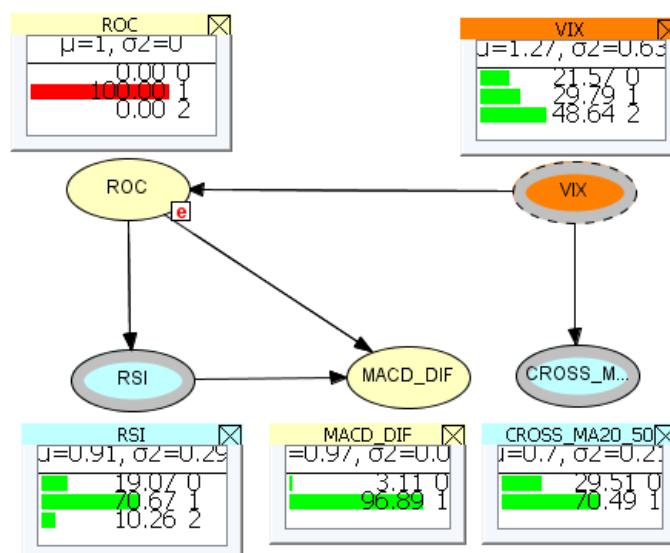


Figure 4.10: Technical Analysis OOBN: Bullish ROC Scenario

Scenario B: The Impact of the 20-50 Moving Averages Cross on the Technical Analysis Area

In Figure 4.12 we simulate an event known by practitioners as “the golden cross”, which occurs when Cross_MA20_50 is equal to 1. This pattern determines a slight increase of the marginal probability associated to overbought Rsi, bullish Roc and positive Macd.Dif, corresponding to their states 1. At the same time, we observe that low Vix probability, state 2, increases from 41.41% to 48.09%. This evidence strengthens the operators’ expectation that prices will go up. In Figure 4.13 we simulate a “death cross” scenario, corresponding to Cross_MA20_50 state 0. This event determines a slight increase of the probabilities associated to oversold Rsi and bearish Roc, while the one associated to positive Macd.Dif, state 1, decreases from 65.45% to 62.26%. Moreover, we observe that Vix state 1 increases in this particular situation. The results obtained confirm that both the “golden cross” and the “death cross” influence practitioners investment approach. A bullish configuration, Cross_MA20_50 state 1, makes investors more prone to purchase stocks and this action determines a Vix decrease. On the other hand, a bearish configuration, Cross_MA20_50 state 0, induces the operators to liquidate or reduce their positions with the result of increasing market volatility.

Scenario C: The Impact of the RSI on the Technical Analysis Area

We now simulate two alternative scenarios, corresponding to overbought and oversold Rsi. When the Rsi is overbought, state 1, the marginal probability associated to

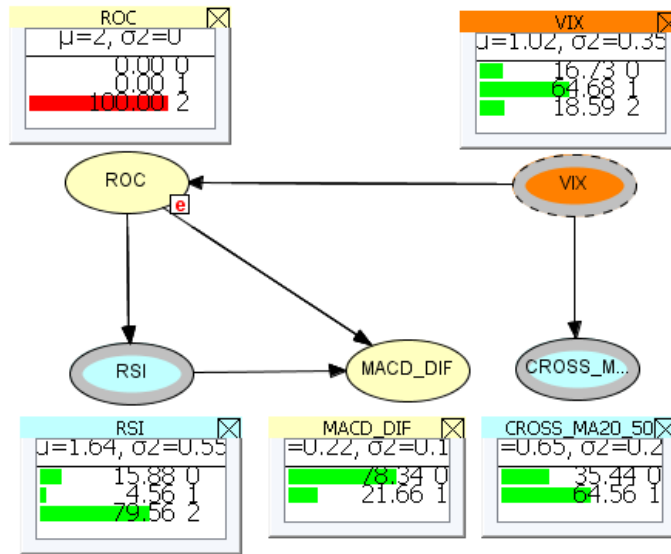


Figure 4.11: Technical Analysis OOBN: Bearish ROC Scenario

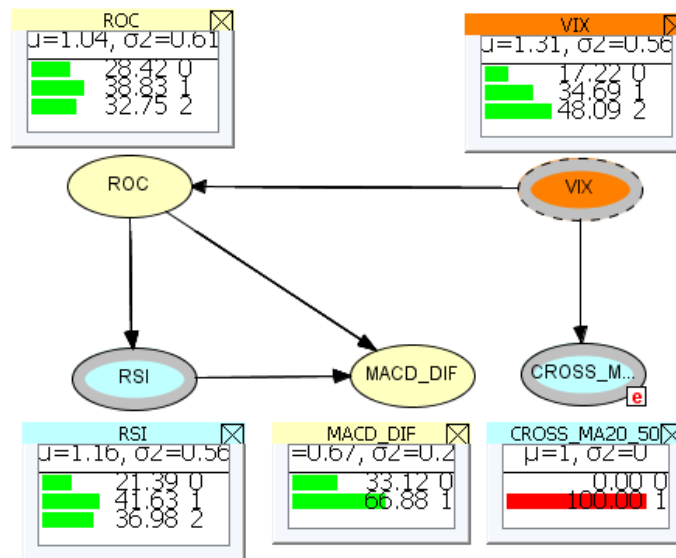


Figure 4.12: Technical Analysis OOBN: "Golden Cross" Scenario

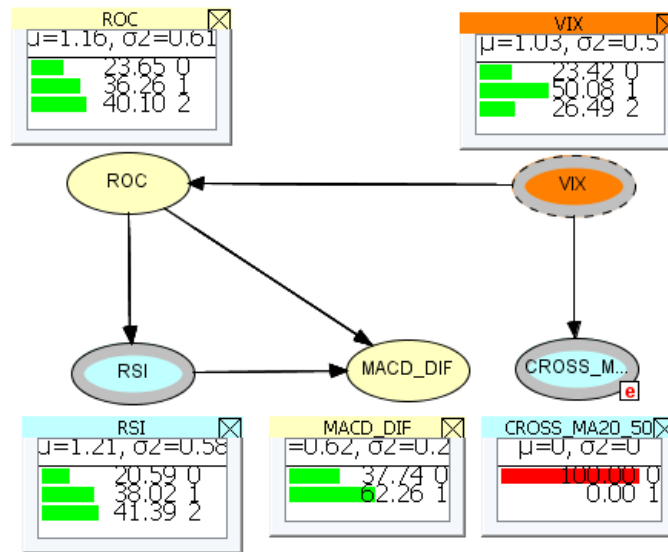


Figure 4.13: Technical Analysis OOBN: “Death Cross” Scenario

positive Macd.Dif jumps from 65.45% to 94.15%, while the state associated to bullish Roc increases from 38.03% to 66.34%, see Figure 4.14. At the same time, low Vix probability, state 2, increases from 41.41% to 51.09%. On the other hand, the oversold Rsi, state 2, determines an increase of Macd.Dif state 0 probability from 34.55% to 73.66%, while the bearish Roc likelihood, state 1, doubles, see Figure 4.15. In conclusion, we observe that the probability associated to high Vix state 1 goes from 39.45% to 53.52%. This evidence confirms that when the market goes down and it reaches the oversold level, the probability of high Vix increases.

Scenario D: The Impact of Vix on the Technical Analysis Area

We conclude our analysis by observing the impact of the input attribute Vix on the nodes belonging to the Technical Analysis area. In a scenario of high volatility, Figure 4.16, we notice that bearish Roc state 2 probability increases from 35.03% to 57.43%, the one referred to oversold Rsi, state 2, goes from 38.34% to 52.02%, while Macd.Dif state 0 probability increases from 34.55% to 48.93%. Our analysis shows that the probability associated to the “golden cross”, Cross_MA20_50 state 1, decreases from 69.05% to 60.71%. However, this result still indicates a higher probability for a bullish 20 and 50 moving averages pattern. In conclusion we simulate how low Vix, state 2, influences the Technical Analysis variables. In Figure 4.17, we notice a bullish configuration for both the Roc and the Rsi, whose states 1 increase respectively from 38.03% to 44.68% and from 40.51% to 49.98%. The mode allows observing that low Vix has a stronger impact than high Vix on Macd.Dif and Cross_MA20_50 because positive Macd.Dif probability goes from 66.45% to 77.58%, while the one referring to Cross_MA20_50 jumps from 69.05% to 80.19%.

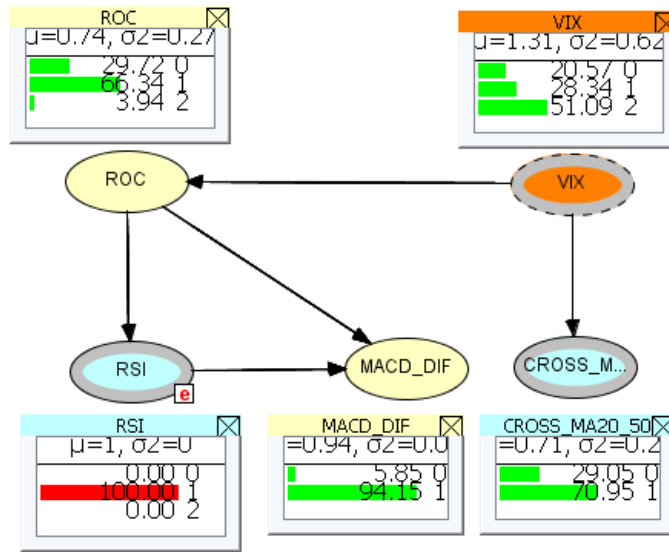


Figure 4.14: Technical Analysis OOBN: Overbought RSI Scenario

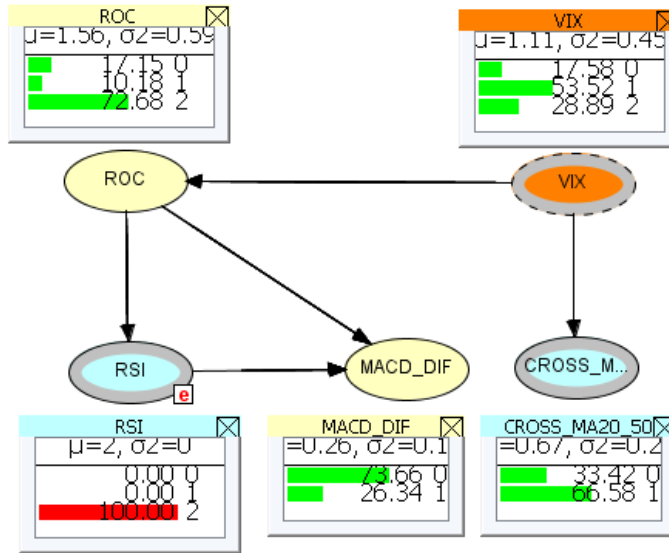


Figure 4.15: Technical Analysis OOBN: Oversold RSI Scenario

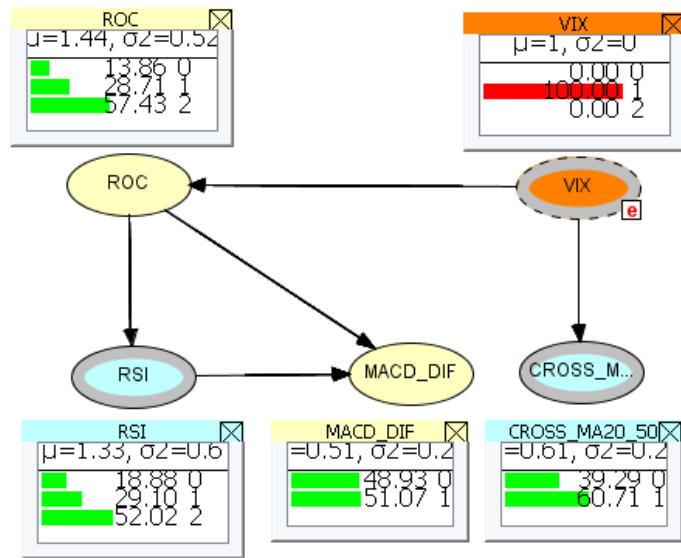


Figure 4.16: Technical Analysis OOBN: The Effects of High Vix

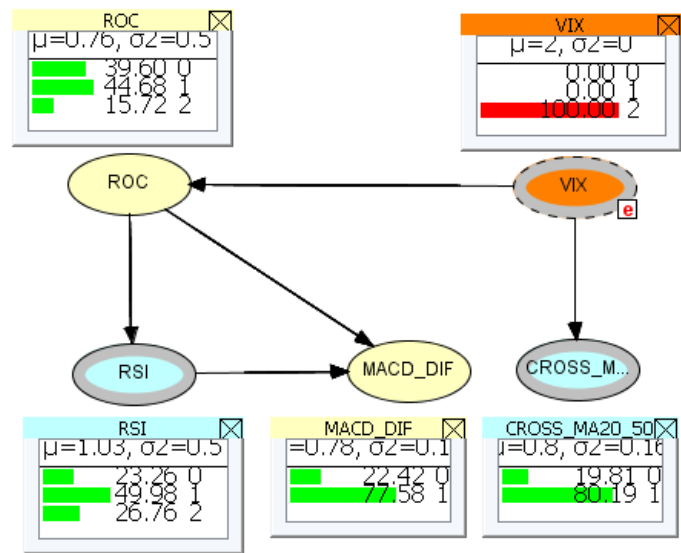


Figure 4.17: Technical Analysis OOBN: The Effects of Low Vix

4.4.3 Macroeconomics Area

The variables belonging to the Macroeconomic area, see Figure 4.18, can influence market dynamics and fund managers decisions. The output nodes belonging to this block are Usa.Gdp and Dxy_Index: the former influences the Micro Dimension area, while the latter is directly connected to the OOBN response variable.

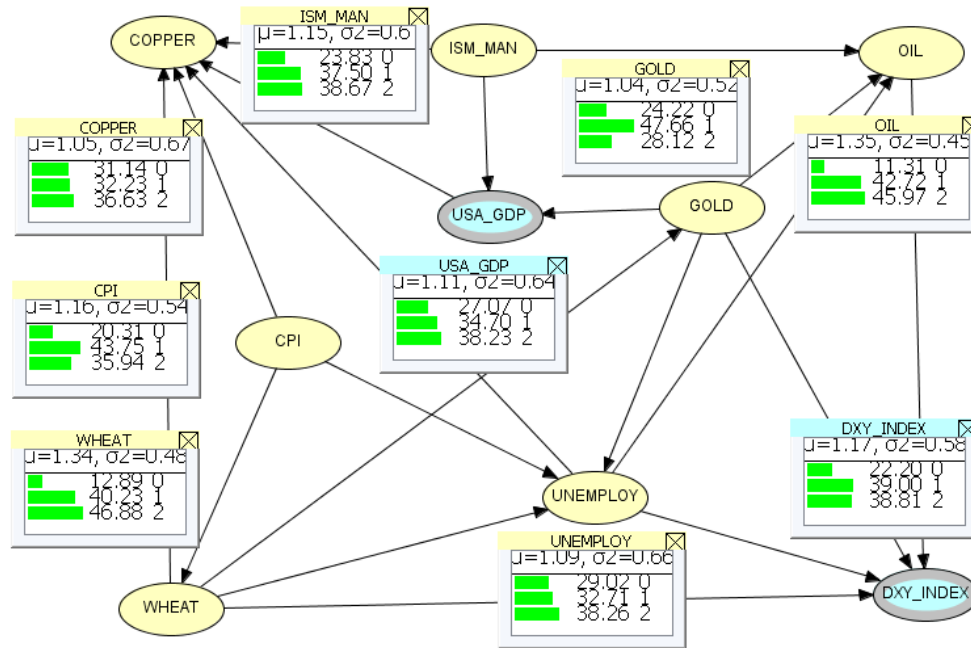


Figure 4.18: The Macroeconomics OOBN

Scenario A: The Impact of American GDP on the Macroeconomic Area

We now simulate the influence of high American GDP growth, corresponding to state 1, on the other variables, see Figure 4.19. According to our financial knowledge, the model shows that the probability associated to a strong dollar, Dxy_Index equal to state 1, increases from 39% to 49.22%. At the same time, an expert investor could expect a general decline of commodities prices as shown by state 2 probability associated to Copper, Oil, Gold and Wheat. Fund managers know the dynamic “American Dollar up / commodities down” but thanks to our graphical representation, we are able to provide further insights, comprehensive real time results and quantitative support. The benefits of a growing GDP are reflected also by the unemployment decrease, whose state 2 goes from 38.26% to 42.33%, by the Ism_Man, whose state 1 jumps from 37.50% to 53.09%, and by the Cpi, whose state 1 goes from 43.75% to 46.81%. On the other hand, in a low Gdp scenario, see Figure 4.20, we observe that the probability associated to a weak dollar, Dxy_Index state 2, increases from 38.81% to 51.63%. At the same time, the marginal probability referred to high commodities prices (Oil, Gold, Copper, and Wheat) increases. Thanks to our model we can also observe the impact of low Gdp on unemployment, whose state 1 increases from 32.71% to 46.51%, while the Ism_Man state 2 jumps from 38.67% to 59.46%.

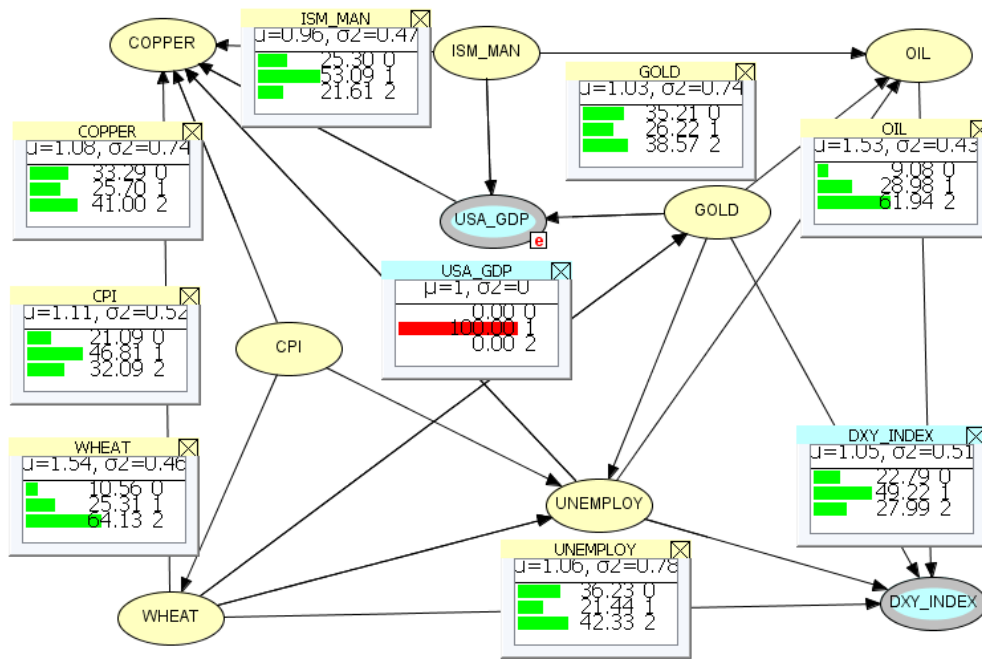


Figure 4.19: Macroeconomics OOBN: High GDP Growth Scenario

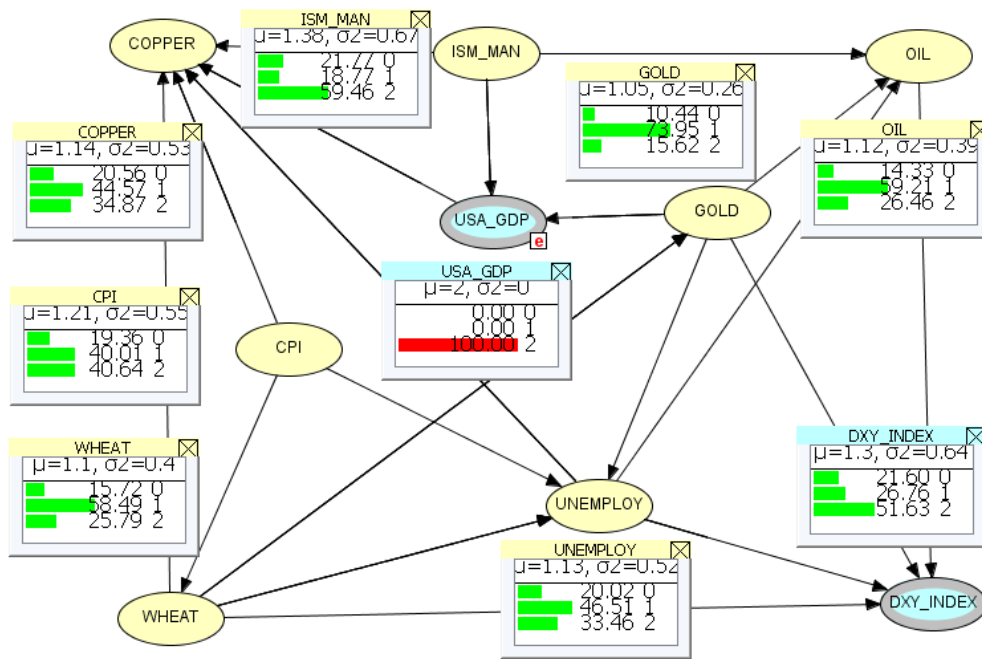


Figure 4.20: Macroeconomics OOBN: Low GDP Growth Scenario

Scenario B: The Impact of Unemployment on the Macroeconomic Area

In Figure 4.21 we simulate a scenario of high unemployment, corresponding to state 1. From a visual inspection we observe that the marginal probability associated to a weak dollar, Dxy_Index state 2, jumps from 38.81% to 76.26%. At the same time, Usa_Gdp marginal probability, state 2, increases from 38.23% to 54.35%, while Cpi state 2 probability goes from 32.09% to 47.49%. Our model shows that the Ism_Man is not influenced by changes in the unemployment rate. This finding reveals us that production inventories, new orders and suppliers' deliveries to the manufacturing sector represent the core part of Ism_Man. As we have seen in the simulation referred to the American GDP, commodities prices are more likely to go up when the economy is slowing. This scenario shows us a similar dynamic, with the exception of copper. This industrial metal has a different and inverse trend because its demand is mainly driven by economic expansion. This finding reveals us that when unemployment is rising there is a higher probability that copper demand decreases and consequently its price is supposed to drop.

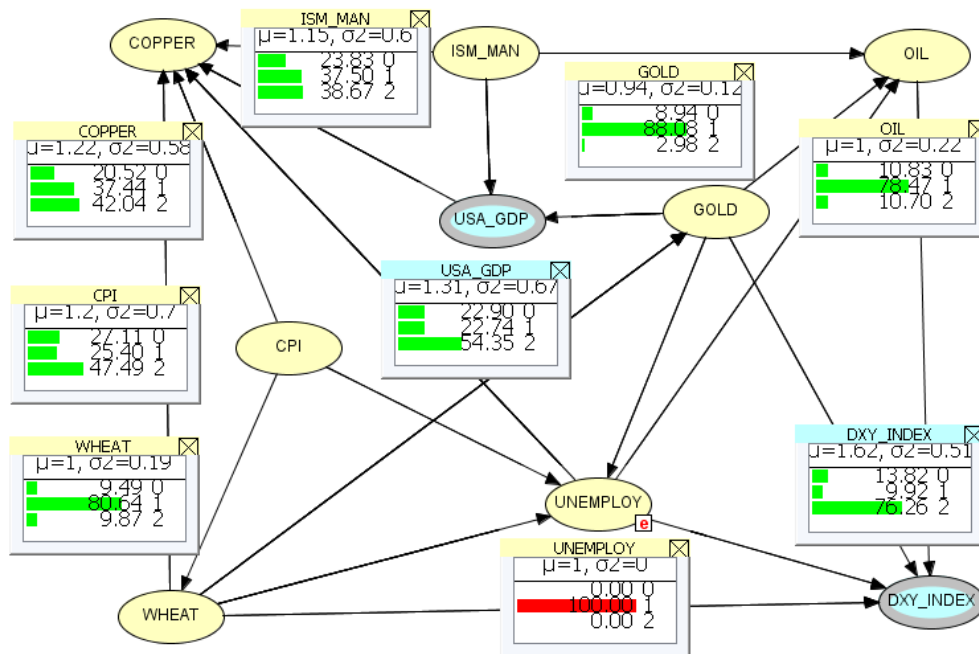


Figure 4.21: Macroeconomics OBN: High Unemployment Scenario

In Figure 4.22 we simulate the impact of low unemployment, corresponding to state 2. This positive scenario determines a strengthening of the dollar, Dxy_Index state 1, that jumps from 39% to 63.79% and a higher probability of high Usa_Gdp, whose probability increases from 34.70% to 38.40%. At the same time we observe that Cpi state 1 increases from 43.75% to 60.58%, while the Ism_Man confirms to be unaffected by unemployment rate changes. On the commodities side, the model shows that in a scenario of high employment there is a higher probability of low commodities prices, state 2. The only exception is represented by copper, which shows an increase of state 0 probability because of its higher correlation to economic cycle.

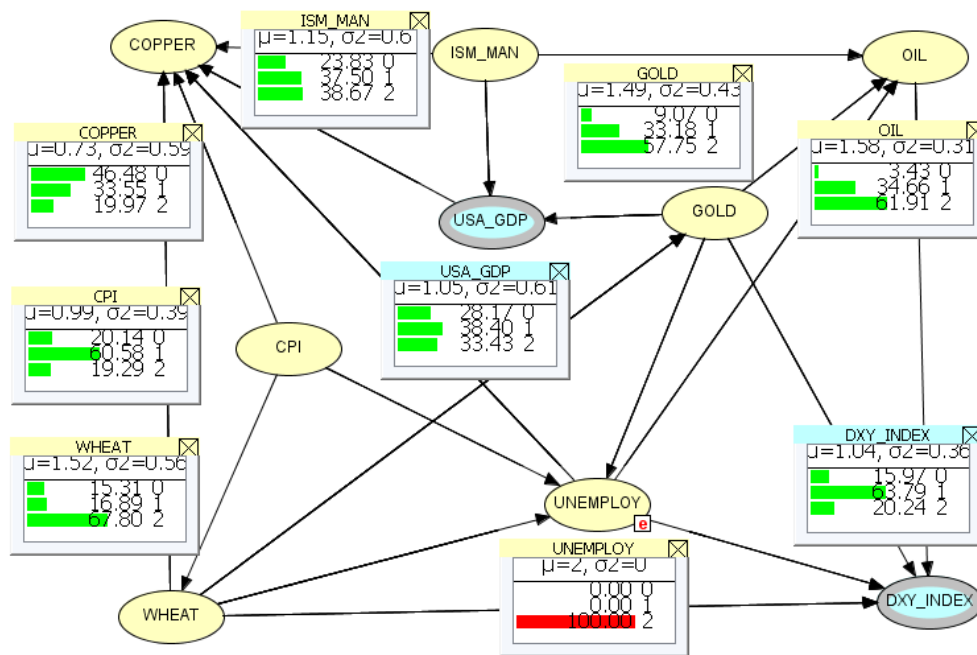


Figure 4.22: Macroeconomics OOBN: Low Unemployment Scenario

Scenario C: The Impact of Gold Price on the Macroeconomic Area

We now we simulate in Figure 4.23 the impact of high Gold price, state 1, on the variables belonging to the Macroeconomics instance node. From a visual inspection, we observe that when gold price rallies the probability of low Gdp growth, state 2, increases by 21%. thanks to its safe haven characteristics. At the same time, we have a higher probability to have high unemployment, whose state 1 grows from 32.71% to 60.47%. In addition, we notice that high Gold makes low Dxy_Index state 2 probability jump from 38.81% to 70.11%, while low Cpi likelihood increases by 10%. All these behaviors are justified by gold safe haven characteristics.

On the commodities side we observe that when Gold price is high, we have a high probability that also the other goods are their highest range, corresponding to state 1. More in detail, Oil state 1 probability doubles from 42.72% to 83.16%, Copper state 1 goes from 32.23% to 40.83%, while Wheat state 1 increases from 40.23% to 77.05%. In conclusion, we observe that the Ism_Man is conditionally independent from Gold price dynamics. On the other hand, in a low gold price scenario, see Figure 4.24, the Oil state 2 probability increases from 45.97% to 98.58%, Wheat state 2 goes from 46.88% to 97.22%, while Copper seems to be more resilient by showing a higher probability associated to its state 0. Furthermore, we can observe that when Gold price is low, Usa_Gdp state 1 probability goes from 34.70% to 47.59%, and the unemployment state 2 probability doubles. We conclude our analysis by observing that we have a higher probability of strong Dollar, state 1 increases by 59.3%, when gold price is low. In this particular situation, the American currency represents the safe haven asset.

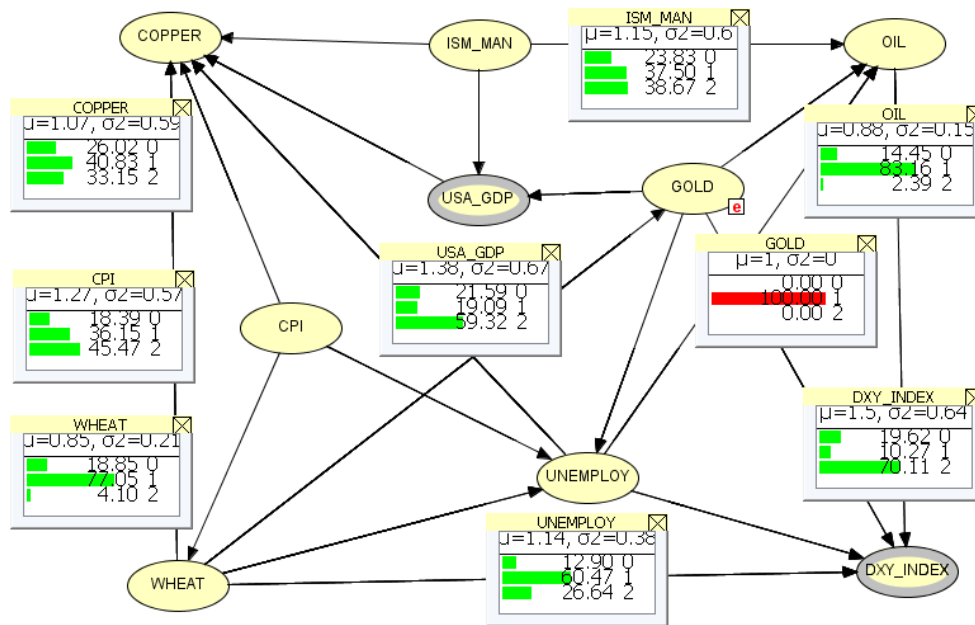


Figure 4.23: Macroeconomics OOBN: High Gold Price Scenario

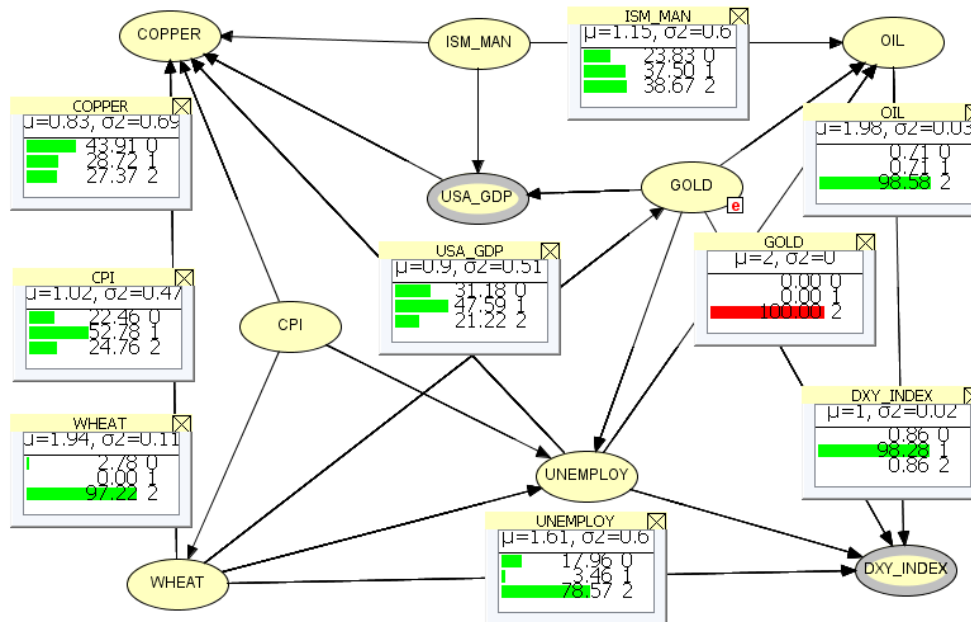


Figure 4.24: Macroeconomics OOBN: Low Gold Price Scenario

Scenario D: The Impact of DXY Index on the Macroeconomic Area

We conclude our analysis on the macroeconomic area by performing simulations on the Dxy_Index. As shown in Figure 4.25, a strong dollar determines a general increase of state 2 probabilities referred to Gold, Oil and Wheat.

The only exception is represented by Copper, whose state 0 evidences the largest increase. This particular dynamic is justified by the fact that industrial metals price depends on Gdp growth as the dollar strength does. Moreover, we observe that when Dxy_Index is high, low unemployment probability increases from 38.26% to 62.59%, while high Usa_Gdp probability goes 34.70% to 43.80%. On the other hand, the strong currency seems to affect negatively the Ism_Man, whose state 2 probability increases from 38.67% to 41.08%. At the same time, we observe that the likelihood of high inflation increases because Cpi state 1 goes from 43.75% to 47.71%. In the weak dollar

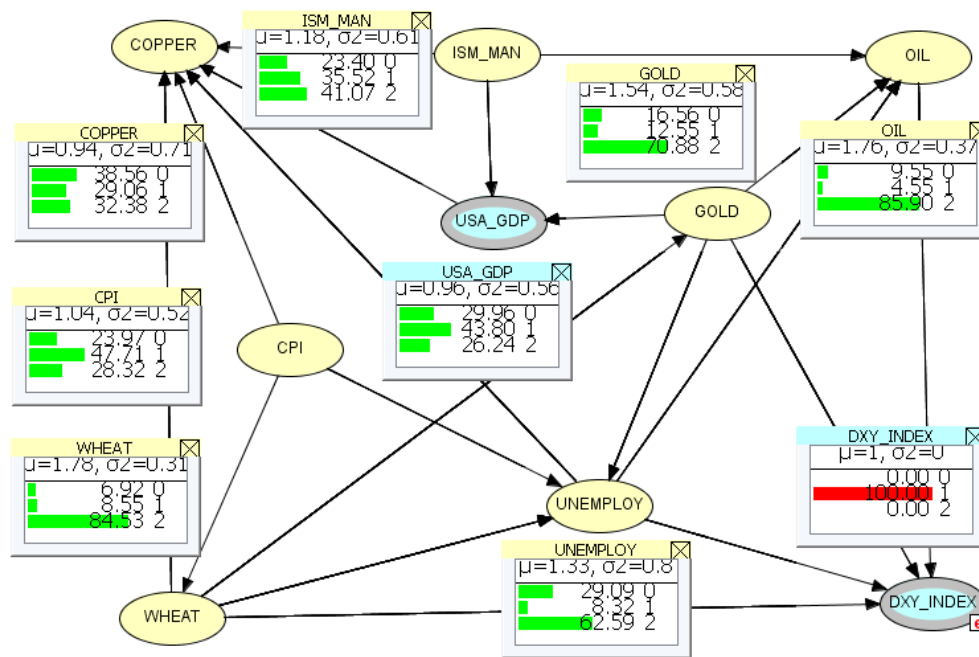


Figure 4.25: Macroeconomics OOBN: Strong Dollar Scenario

simulation, Figure 4.26, we observe that state 1 probabilities associated to Gold, Oil and Wheat strongly increase. At the same time, low Dxy_Index is the symptom of a weakening economy, as reflected by Usa_Gdp state 2 probability that increases from 38.23% to 50.86%, and high unemployment state 1 probability that jumps from 32.71% to 64.29%. In this particular situation, Copper does not provide clear indications because state 1 probability goes from 32.23% to 36.54%, while state 2 probability increases from 36.63% to 37.29%. On the other hand, inflation evidences a slow down reflected by the Cpi state 2 probability that increases from 35.94% to 43.45%. Finally, we observe that a weak dollar affects positively the Ism_Man as shown by the increase of the marginal probability associated to its state 1 that goes from 37.50% to 41.81%.

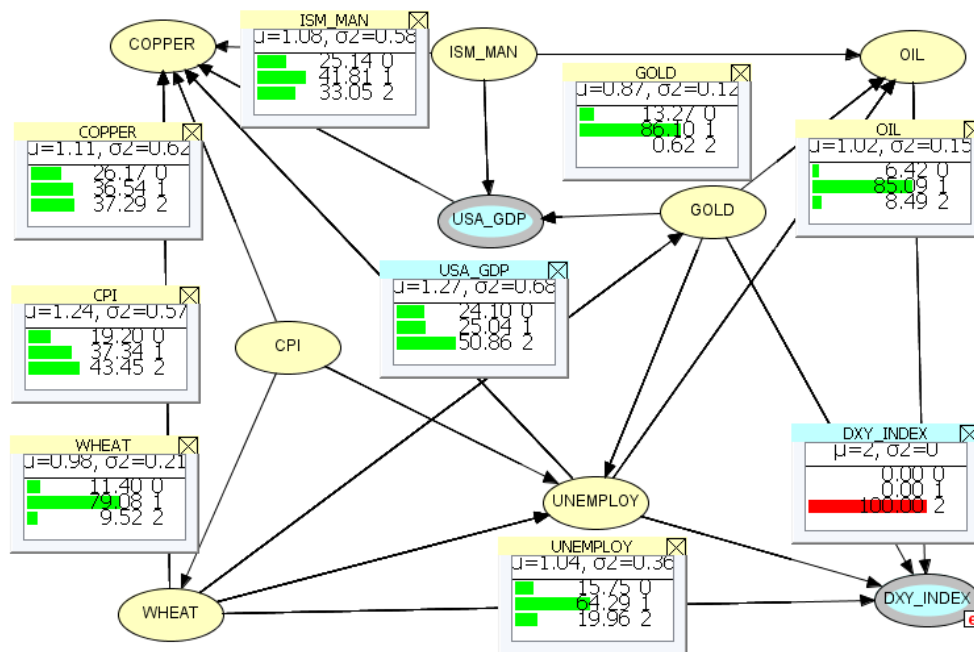


Figure 4.26: Macroeconomics OOBN: Weak Dollar Scenario

4.4.4 Microeconomic Dimension Area

The block shown in Figure 4.27 deals with a large range of variables associated to the firm dimension. We underline that because of its complexity, the learning procedure does not indicate clearly an output node and consequently we appoint as target variable the *Pe_Ratio* by following practitioners' consensus. From a visual inspection, we observe that *Ev_Ebitda*, *P_S*, *Cross_MA20_50*, *Dvd_Yld* and *Profit_Mrg* directly influence the *Pe_Ratio*.

We also underline that the Microdimension block includes two input nodes: *USA_Gdp* belongs to the macro area and it influences the variables connected to *Sales*, *Ebitda* and *Earning growth*, while *Cross_MA20_50* is linked to the relative valuation nodes that are sensible to price movements.

Scenario A: The Impact of EPS Growth on the Micro Dimension Area

We now simulate the impact of high *Eps_G*, state 1, see Figure 4.28. Our model shows that the probability associated to high *Sales_G* increases from 67.65% to 68.94%, while the one associated to high *Ebitda_G* goes from 69.23% to 82.02%. In addition, high *Pe_Ratio* state 1 represents the most likely outcome. This indication suggests that when *Eps* grow, investors buy more stocks driven by positive expectations and consequently the prices goes up. Moreover, high *Eps_G* increases the probability of high *USA_Gdp*, state 1, that goes up from 34.29% to 44.82%, This result shows the relation between economic cycle and firms earnings. Surprisingly, the model shows an increase of the probability associated to low *Sales_Ps*, low *Ebitda_Ps*, and *Earn_Ps*. In conclusion, we observe that *Dvd_Yld* and *Cross_Ma20_50* are independent from *Eps_G*. This output indicates that stocks profitability and shares price patterns do not depend from the

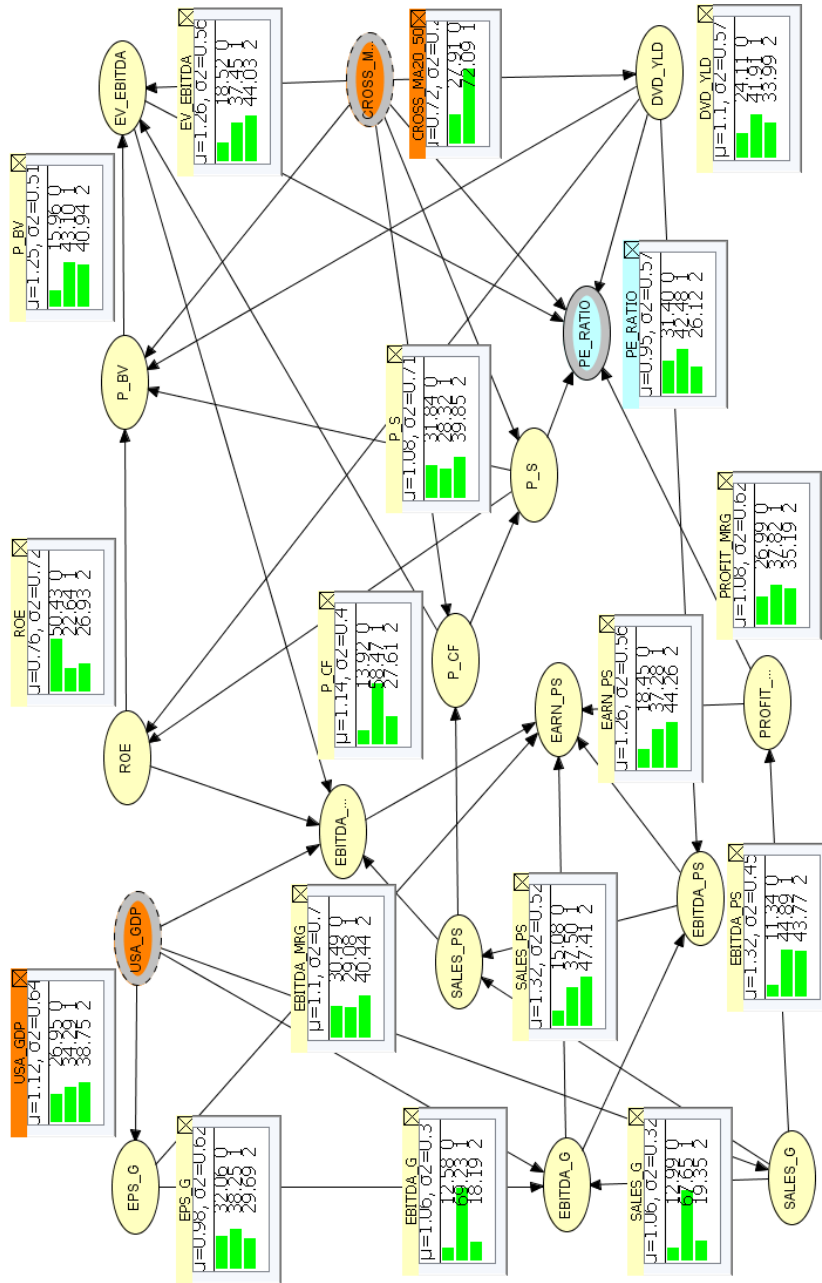


Figure 4.27: The Micro Dimension OBN

earnings trend. In Figure 4.29 we observe the impact of low Eps_G, state 2, on the other variables belonging to the Microdimension area. Also in this case, state 1 probability referred to Sales_G and Ebitda_G is still the most likely outcome. However, we can observe a general decrease of their respective likelihood. This evidence reveals us that the balance sheet “bottom line” can be heavily influenced by costs, depreciations, and financial decisions. Moreover, this simulation reveals us that high Pe_Ratio probability decreases from 42.48% to 41.78%, showing that company fundamentals are not always reflected in markets valuation. In addition, our model confirms the strong relation between earnings and economic cycle because low Eps_G makes low USA_Gdp decrease from 38.75% to 55.47%. At the same time, when Eps_G is low, we observe a general increase of per share values for Sales, Ebitda and Eps. In conclusion, we observe again that Eps_G does not influence Dvd_Yld and Cross_Ma20_50.

Scenario B: The Impact of ROE on the Micro Dimension Area

We now simulate the impact of high Roe, see figure 4.30. First, we notice that the marginal probability associated to Pe_Ratio state 0 increases from 31.40% to 54.07%, a result indicating again that fundamentals are not always reflected by valuation metrics. By continuing our visual inspection, we observe that a high Roe scenario increases the probability of high P_Bv, whose state 1 probability goes from 43.10% to 65.22%. At the same time, Dvd_Yld state 2 probability jumps from 33.99% to 66.92%. In conclusion, we notice that high Roe determines an increase of the probability associated to Ev_Ebitda state 1 from 37.45% to 49.53%.

The simulation in Figure 4.31 shows us that low Roe increases the probability associated to low Pe_Ratio from 26.12% to 55.98%. The impact of low Roe on the Microdimension area is confirmed by the consistent increase of low Ebitda_Mrg probability, state 2, that goes from 44.03% to 83.23%. Moreover, we can observe that P_Bv state 2 marginal probability increases from 40.94% to 97.70%. This dynamic is justified by the fact that the Roe is included in the numerator of P/Bv formula. In conclusion, we notice that the dividend yield is more likely to be high, state 1, because its probability grows from 34.67% to 47.81%.

Scenario C: The Impact of Ebitda Margin on the Micro Dimension Area

We now simulate in Figure 4.32 the high Ebitda_Mrg, state 1, scenario. Our model shows us that Pe_Ratio state 1 probability increases from 42.48% to 54.26%, while the one referred to P_Bv state 2 goes from 48.76% to 56.13%. These dynamics indicate that the buyers' pressure is higher when market marginality grows. On the growth variables side (Sales_G, Ebitda_G and Eps_G), we notice a general state 1 increase. Thanks to this finding, we can derive that the higher the sales, the higher the probability of growing margins. Consequently, high Sales_G increases the probability of higher Ebitda_G and Eps_G. In conclusion, our simulation shows that high USA_Gdp probability increases in case of high Ebitda_Mrg, and that Cross_MA20_50 state 1 increases by 18.4%. All these indications are positive for market dynamics. On the other hand, when we simulate a low Ebitda_Mrg scenario, see Figure 4.33, the probability referred to low Pe_Ratio goes from 26.12% to 37.10%. This output reveals that P/E ratio is low when companies' marginality decreases. Moreover, we observe that P_Bv state 2 probability increases from 40.94% to 53.09%, a result indicating that investors' selling pressure is higher when Ebitda_Mrg is low. By looking at the growth variables (Sales_G, Ebitda_G and Eps_G) we notice a general decrease of their state 1 probabilities. This outcome indicates that low Ebitda_Mrg is a signal of slowing down businesses.

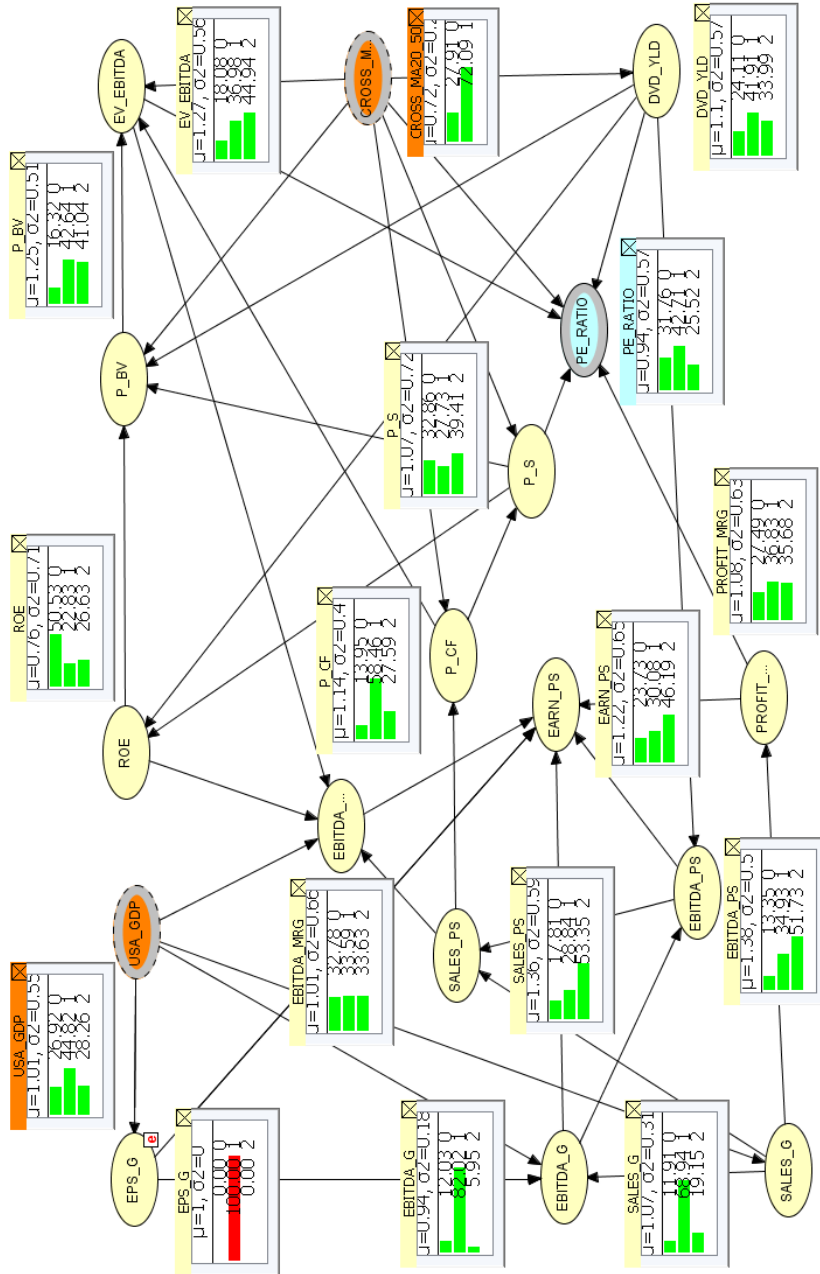


Figure 4.28: Micro Dimension OOBN: High EPS Growth

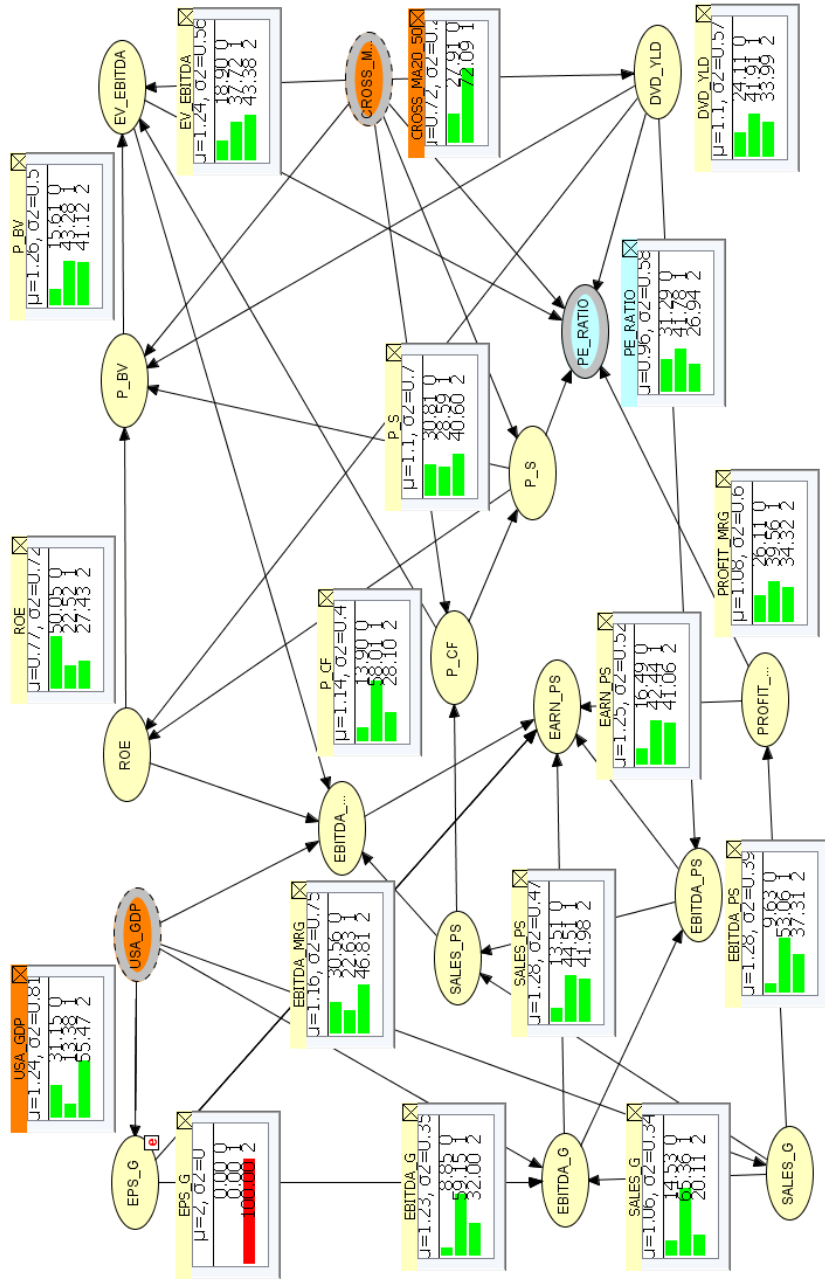


Figure 4.29: Micro Dimension OOBN: Low EPS Growth

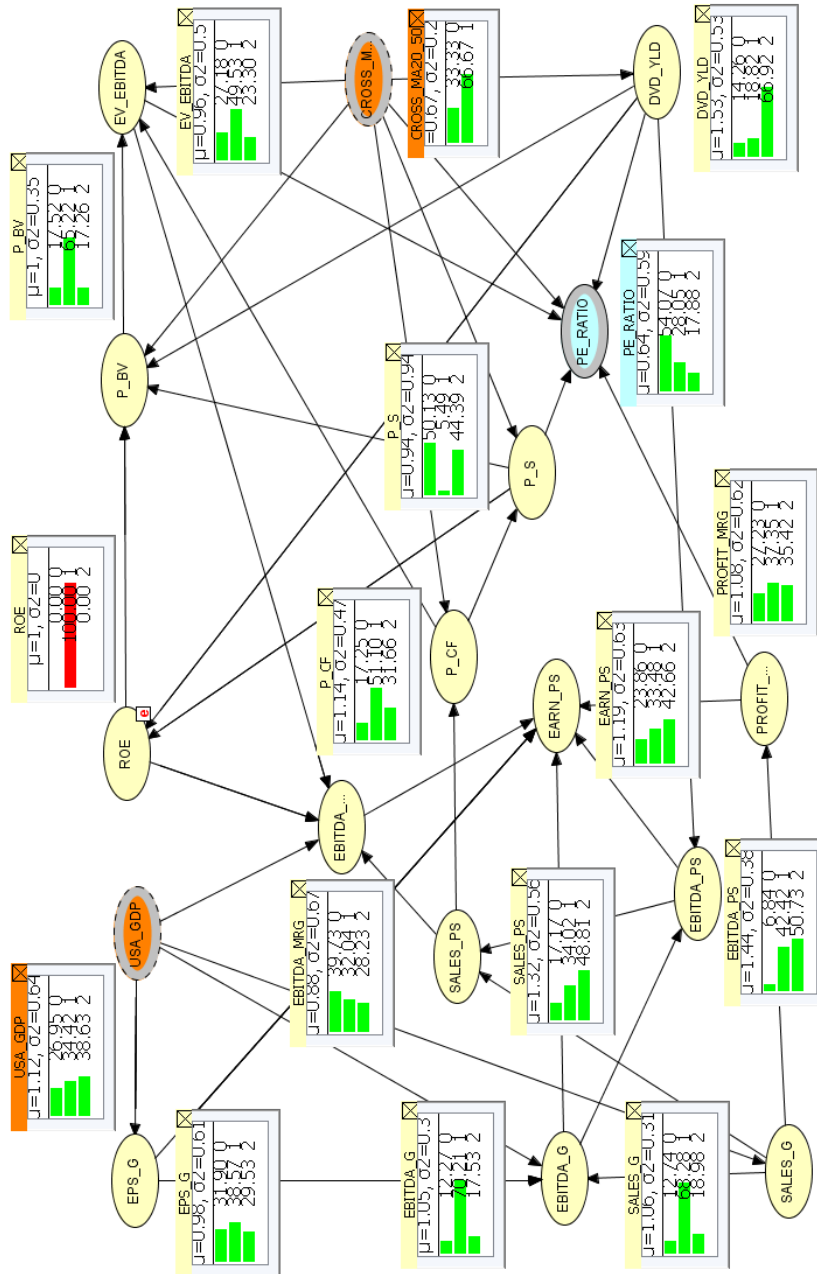


Figure 4.30: Micro Dimension OBN: High ROE

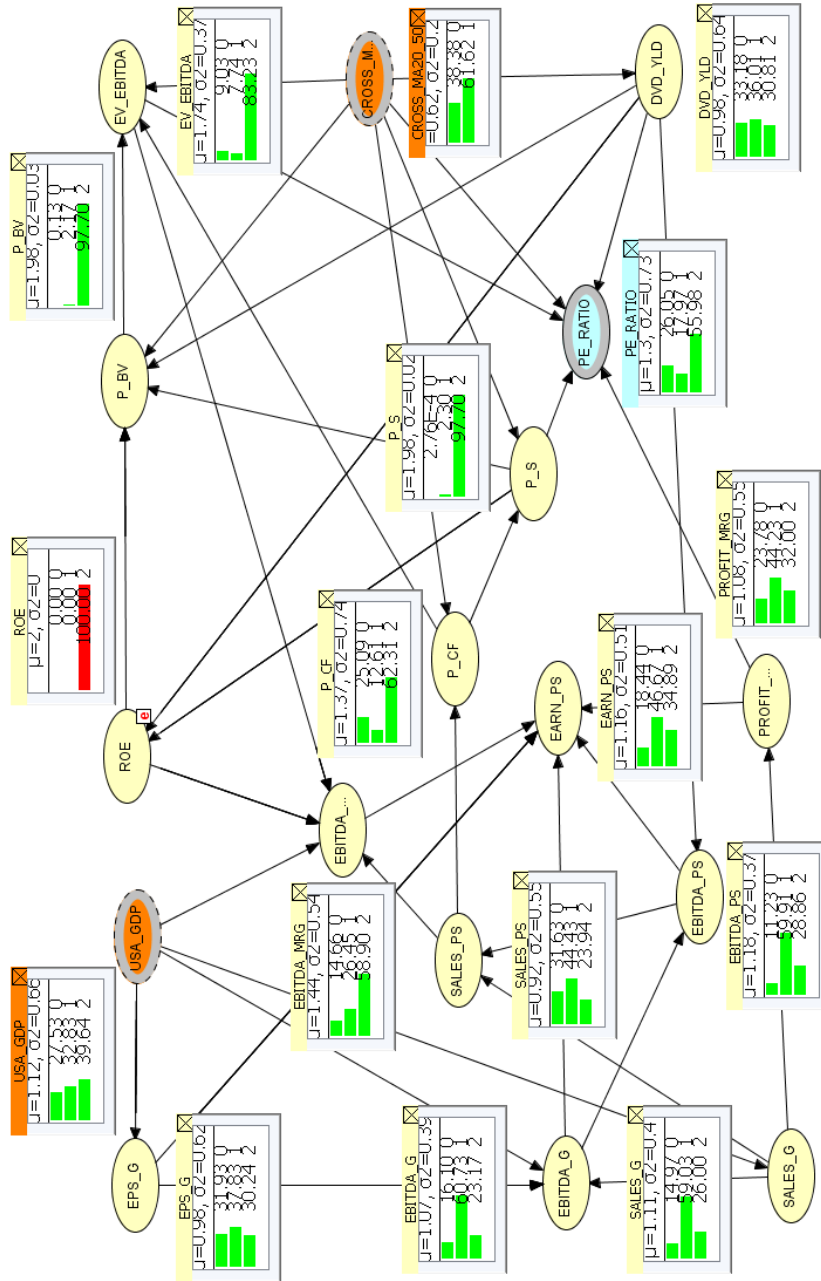


Figure 4.31: Micro Dimension OOBN: Low ROE

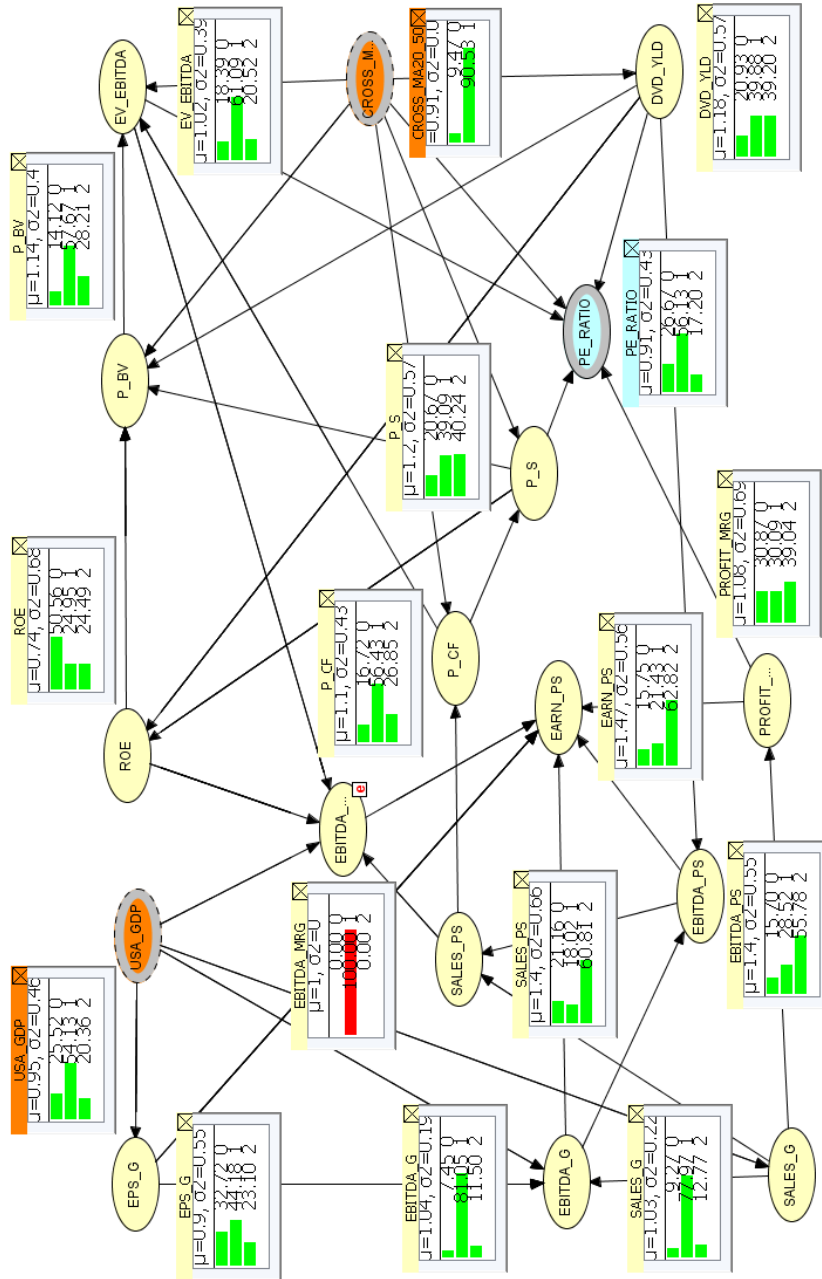


Figure 4.32: Micro Dimension OOBN: High Ebitda Margin

In addition, thanks to this model we find that Ebitda.Mrg depends on economic cycle as reflected by USA.Gdp, whose state 2 probability increases by 12%. We conclude the low Ebitda.Mrg simulation by observing that the probability associated to the “golden cross”, corresponding to Cross.MA20_50 state 1, decreases slightly and consequently it indicates a higher probability that the prices will go down.

Scenario D: The Impact of EV/Ebitda on the Micro Dimension Area

In Figure 4.34 we show the impact of high Ev.Ebitda on the other variables. The results suggest that high Ev.Ebitda increases the probability of observing a “golden cross” in the S&P 500 chart, a pattern corresponding to Cross.MA20_50 state 1. In line with this finding, we have a general state 1 increase for the variables that are sensible to price movements: Pe.Ratio state 1 probability increases from 42.48% to 66.45%, the one associated to P_S goes from 28.32% to 55.23%, P_Cf state 1 probability increases from 58.47% to 73.37%, while P_Bv jumps from 43.10% to 81.61%. On the other hand, Dvd.Yld state 2 goes up by 12.5%, reflecting that any price increase has a negative impact on this variable obtained by dividing the dividend by the stock price. In Figure 4.35 we simulate the impact of low Ev.Ebitda, corresponding to its state 2. Thanks to our model, we observe that the “death cross” probability, corresponding to Cross.MA20_50 state 2, increases from 27.91% to 53.66%. At the same time, the probability to observe low valuation variables, state 2, increases as the probability of a “death cross” goes up: Pe.Ratio state 2 probability goes from 26.12% to 48.28%, the one referred to P_S increases from 39.85% to 59.20%, P_Bv jumps from 40.94% to 79.99%. In conclusion, we notice that Dvd.Yld state 1 increases from 41.91% to 48.89% reflecting the positive effects of a price drop.

Scenario E: The Impact of American GDP on the Micro Dimension Area

We now simulate the impact of high USA.Gdp on the variables belonging to the Microdimension block, see Figure 4.36. In line with our financial knowledge, we observe that Sales_G, Ebitda_G and Eps_G state 1 probabilities increase respectively from 67.65% to 74.01%, from 69.23% to 76.03% and from 38.25 to 49.99%. At the same time, high USA.Gdp increases high P/E ratio probability that goes from 42.48% to 44.81%, while the Cross.MA20_50 and Dvd.Yld reveal to be independent from the economic trend. We now simulate the impact of low Gdp_Usa on the Microdimension block, see figure 4.37. In this scenario, we observe that Sales_G, Ebitda_G state 1 probabilities decrease respectively from 67.65% to 61.79% and from 69.23% to 64.05%. On the other hand, the Eps_G state 2 probability jumps from 29.69% to 42.49%, revealing that the Earnings are more sensitive to the economic cycle than the other balance sheet aggregates. Moreover, low USA.Gdp determines a small contraction of high P/E ratio probability, that goes from 42.48% to 40.44%. This result shows again that market valuations sometimes do not incorporate the fundamentals of an economy. In conclusion, the model confirms that Cross.MA20_50 and Dvd.Yld are influenced by any USA.Gdp change.

Scenario F: The Impact of 20-50 Moving Averages Cross on the Micro Dimension Area

We now simulate how the Cross.MA20_50 node, belonging to the Technical Analysis area influences the micro dimension block, see Figure 4.38. By simulating a “death cross” scenario, Cross.MA20_50 state 0, we notice that state 2 probability associated to

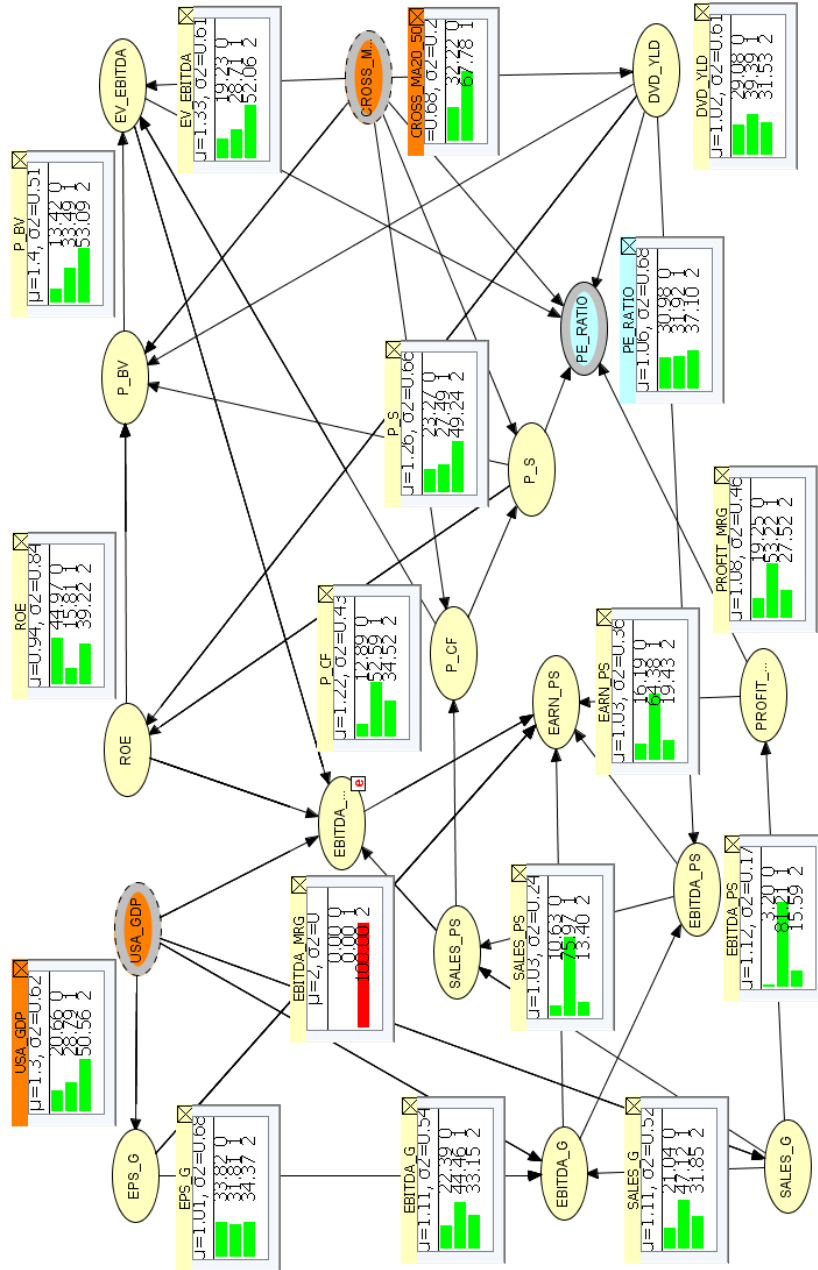


Figure 4.33: Micro Dimension OBN: Low Ebitda Margin

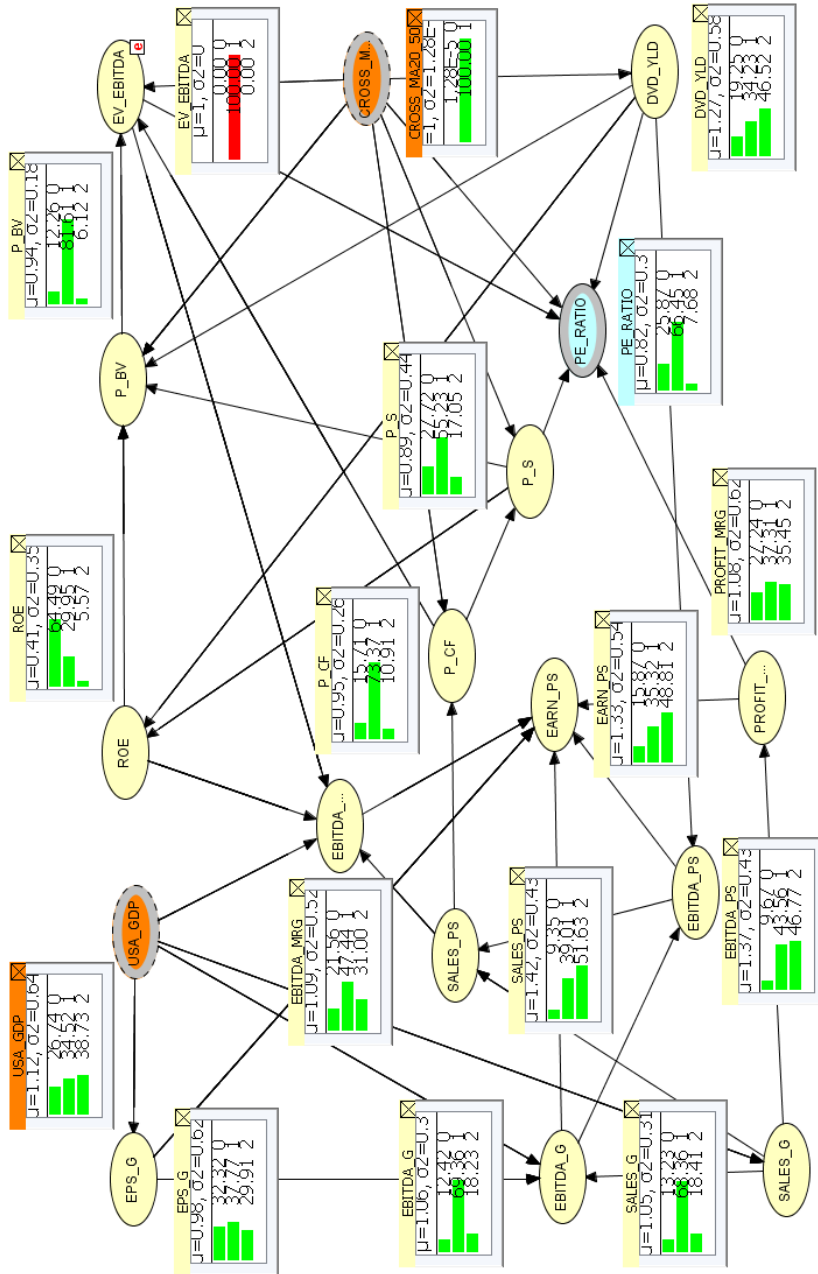


Figure 4.34: Micro Dimension OOBN: High EV/Ebitda

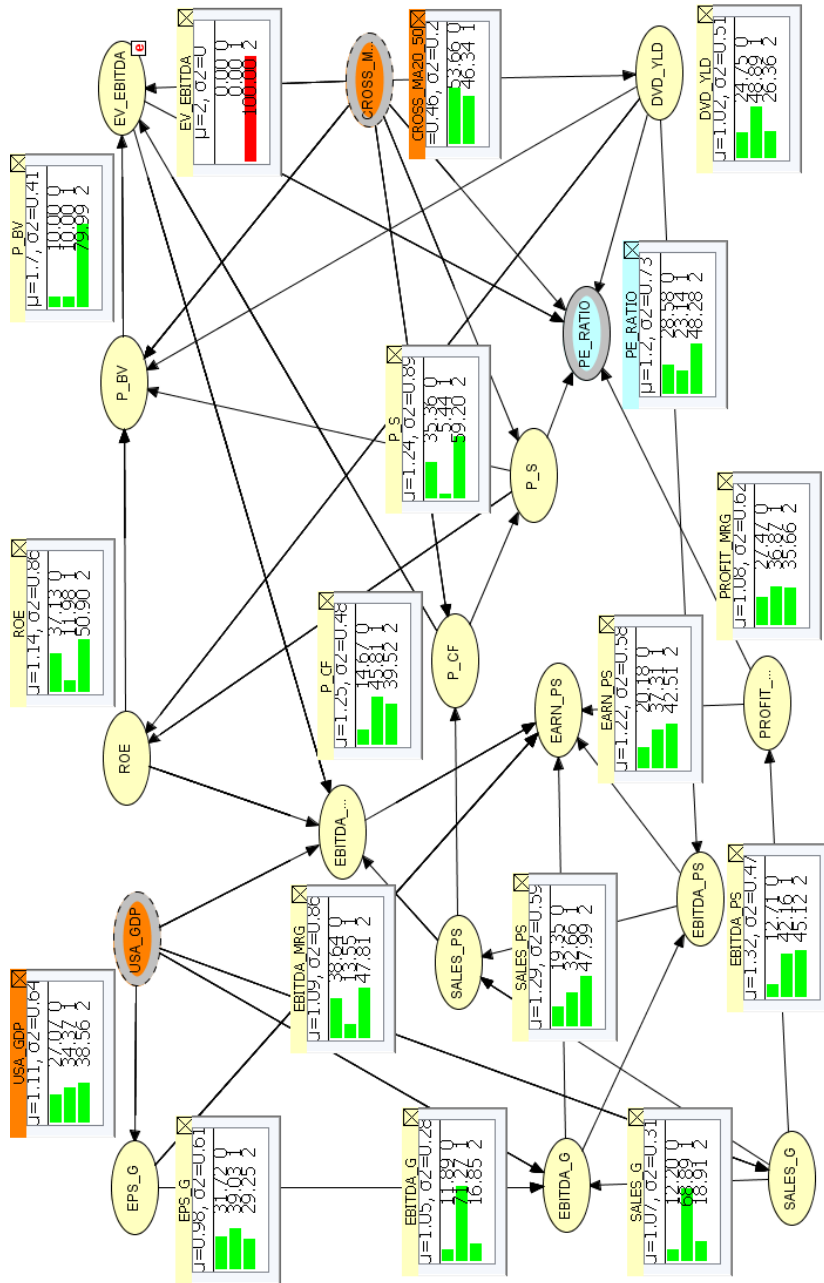


Figure 4.35: Micro Dimension OOBN: Low EV/Ebitda

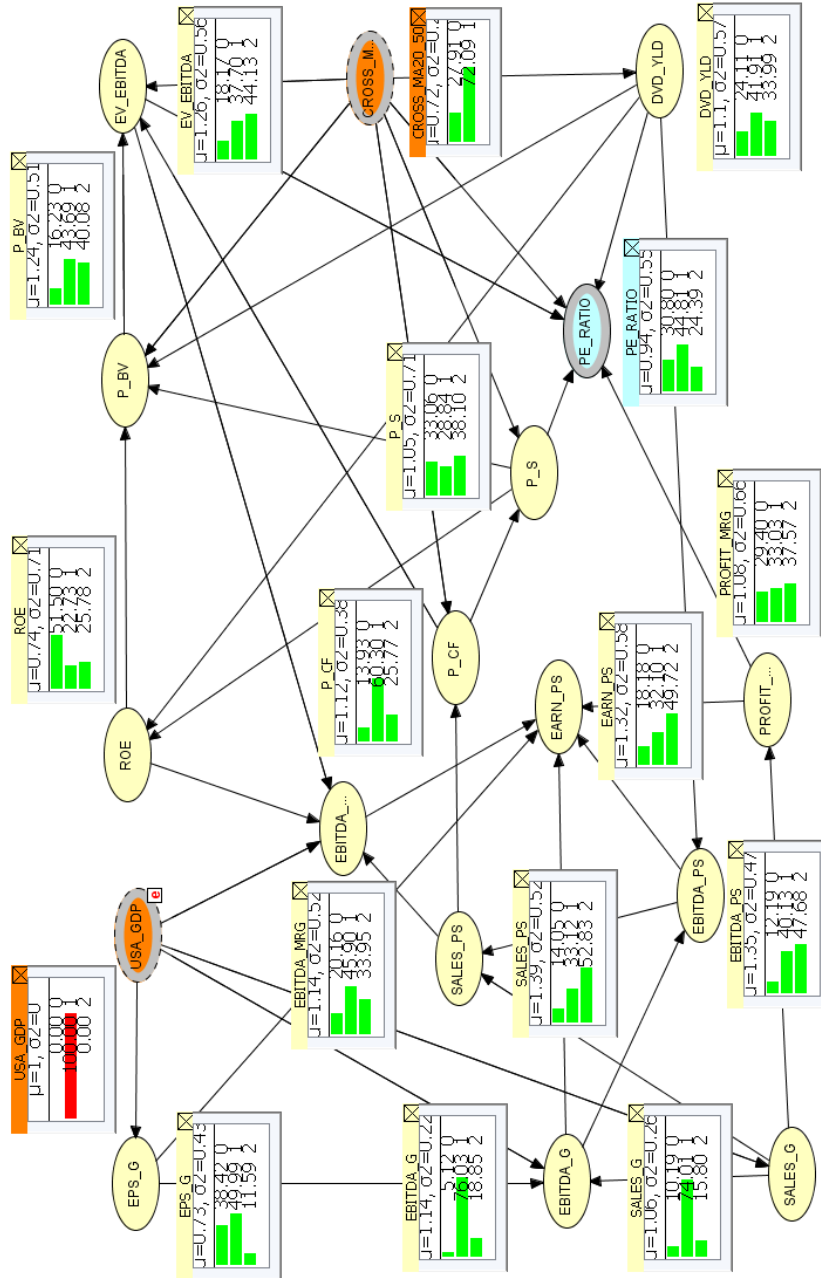


Figure 4.36: Micro Dimension OBN: The Influence of High GDP Growth

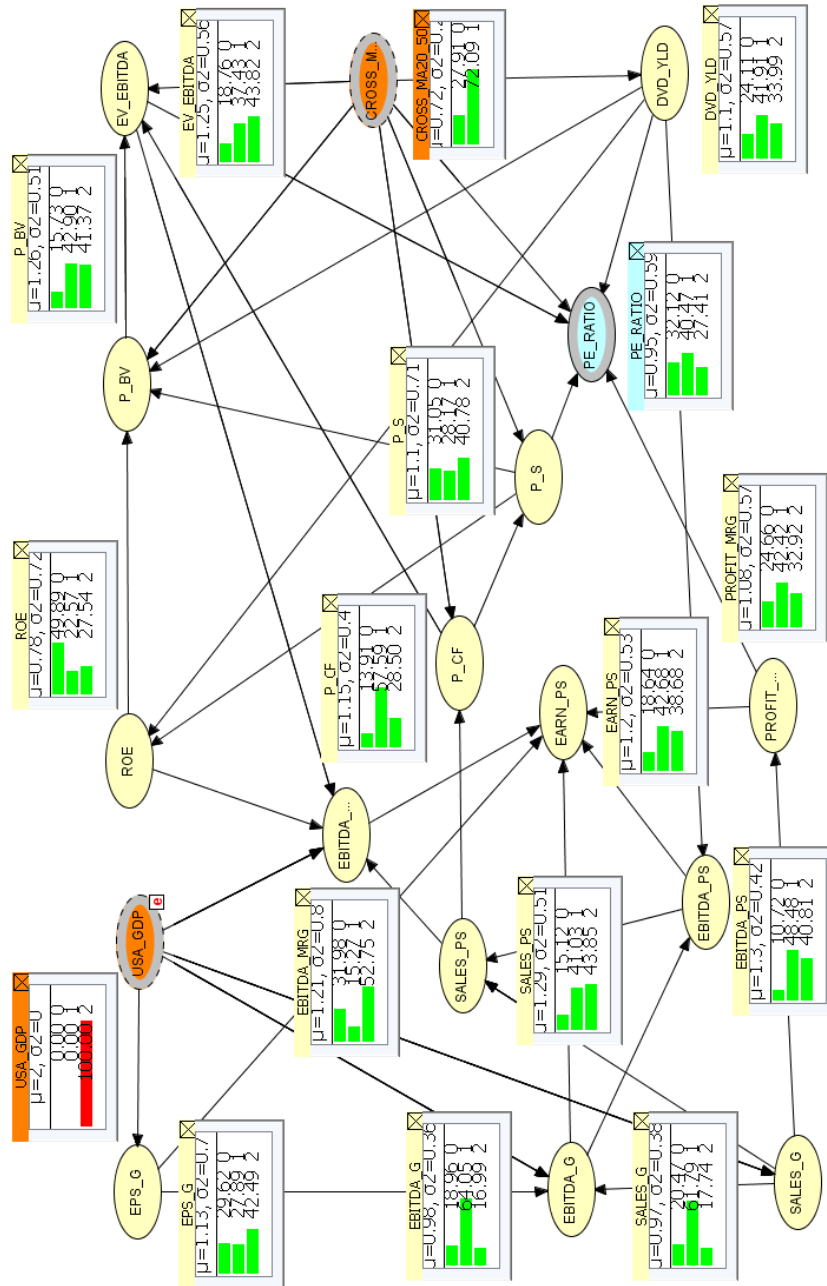


Figure 4.37: Micro Dimension OoBN: The Influence of Low GDP Growth

relative valuation nodes increases. More in detail, we observe that the Pe_Ratio state 2 probability goes from 26.12% to 47.42%. This result suggests a strong relation between stock prices and this multiple. Thanks to our model, we observe that Profit_Mrg, Sales_G, Ebitda_G, Eps_G and USA_Gdp are independent from technical analysis. In Figure 4.39 we simulate a “golden cross” scenario, corresponding to Cross_MA20_50 state 1. The scenario reveals that state 1 probability generally associated to “price multiples” increases because they heavily depend on market trend. For example, the PE_Ratio state 1 probability increases from 42.48% to 54.13%. In conclusion, this simulation allows us to observe that Cross_MA20_50 is conditionally independent from the nodes Profit_Mrg, Sales_G, Ebitda_G, Eps_G and USA_Gdp.

Scenario G: The Impact of P/E Ratio on the Micro Dimension Area

We now simulate the impact of high Pe_Ratio, see Figure 4.40. From a visual inspection, we notice an increase of state 2 probability referred to Sales, Ebitda and Earnings per share. This dynamic, reflects an accountability dynamic: if the turnover goes down, Ebitda and the Eps are more likely to decrease. By observing the nodes associated to balance sheet aggregates growth, we notice an increase of state 1 probability referred to Sales_G, Ebitda_G, while Eps_G probability remains stable. Moreover, our model allows to observe that high Pe_Ratio determines an increase of state 1 probability associated to P_Cf and P_S, that goes up respectively from 58.47% to 77.39% and from 28.32% to 50.41%. In conclusion, we notice that the probability to observe a “golden cross”, corresponding to Cross_MA20_50 state 1, jumps from 72.09% to 91.87%, while USA_Gdp reveals to be mildly affected by high Pe_Ratio. We conclude our analysis by simulating a low Pe_Ratio scenario, see Figure 4.41. Our model allows us to observe an increase of the probability associated to high sales, high Ebitda and high earnings per share. At the same time, we have a state 1 reduction referred to Sales_G, Ebitda_G, while Eps_G probability is substantially unaffected by a Pe_Ratio change. From a macro perspective, low Pe_Ratio determines an increase of low USA_Gdp probability from 38.75% to 40.68%. In conclusion, this simulation shows that low P/E increases the probability of a “death cross” because Cross_MA20_50 state 0 probability jumps from 27.91% to 50.67%.

4.5 OOBN What-if Analysis Simulations

Now that the market OOBN, see Section 4.3, and its instance nodes, see Section 4.4, have been introduced, we can simulate different scenarios that allows us to identify the variables or areas that trigger buy/sell signals.

In the following paragraphs, we analyze some relevant combinations of events from a financial perspective. The following simulations allow us to show OOBNs efficiency in supporting investors:

- A) In this case, we simulate a scenario where Dxy_Index, Pe_Ratio and Rsi are equal to 1. Then the recommendation obtained is compared with the one provided by setting Pe_Ratio and Rsi equal to 1 and Dxy_Index equal to 2.
- B) In this simulation, we compare the indication provided by a scenario where Dxy_Index, Vix, Rsi and Pe_Ratio are equal to 1 with the one where Dxy_Index, Vix and Rsi are equal to 1 and Pe_Ratio equal to 2.

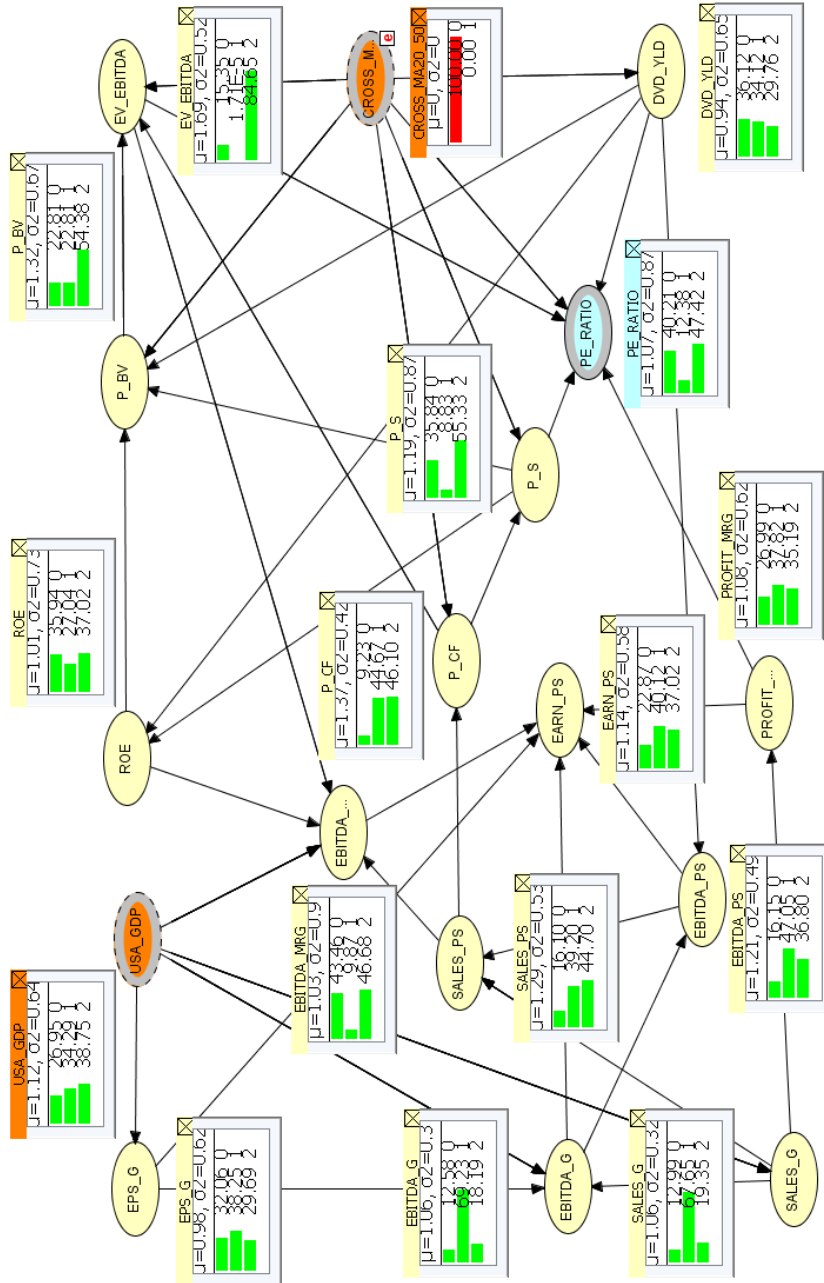


Figure 4.38: Micro Dimension OOBN: The Influence of the “Death Cross”

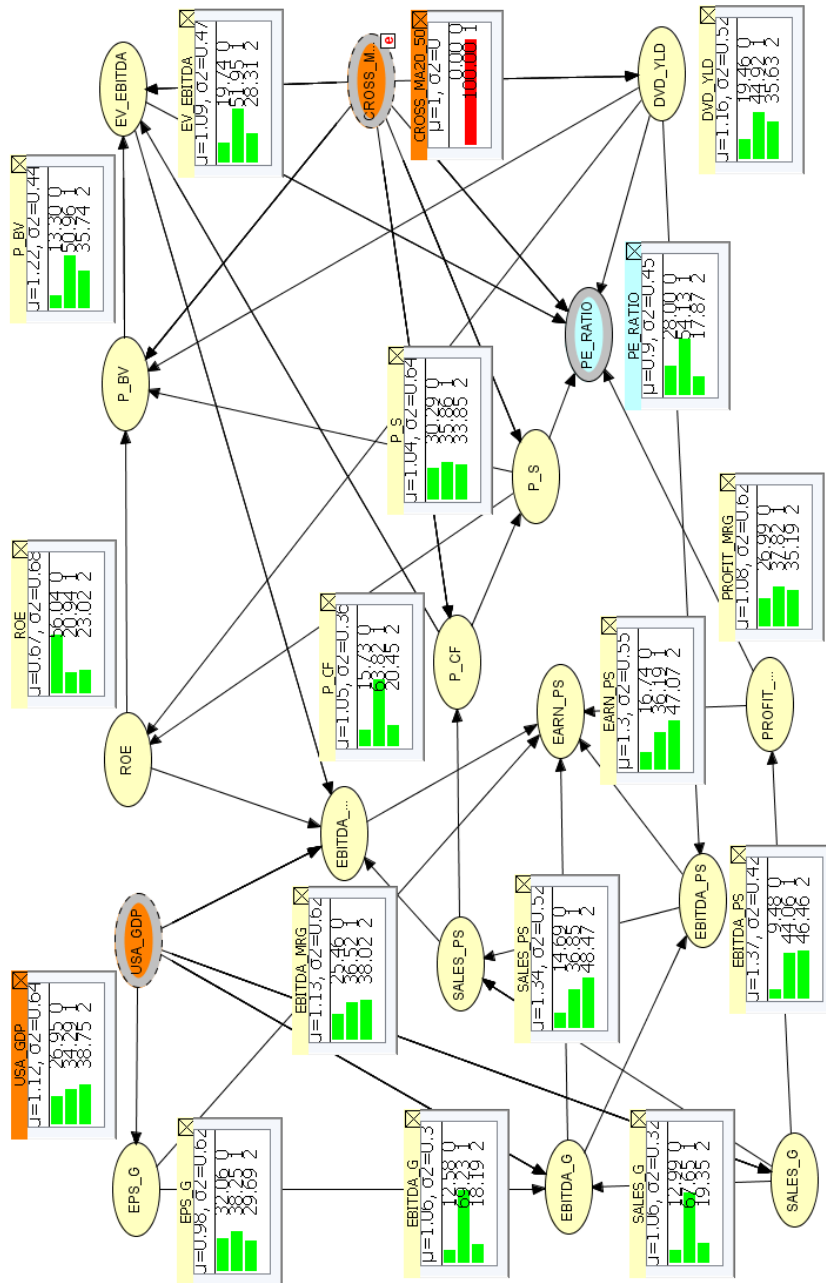


Figure 4.39: Micro Dimension OOBN: The Influence of the “Golden Cross”

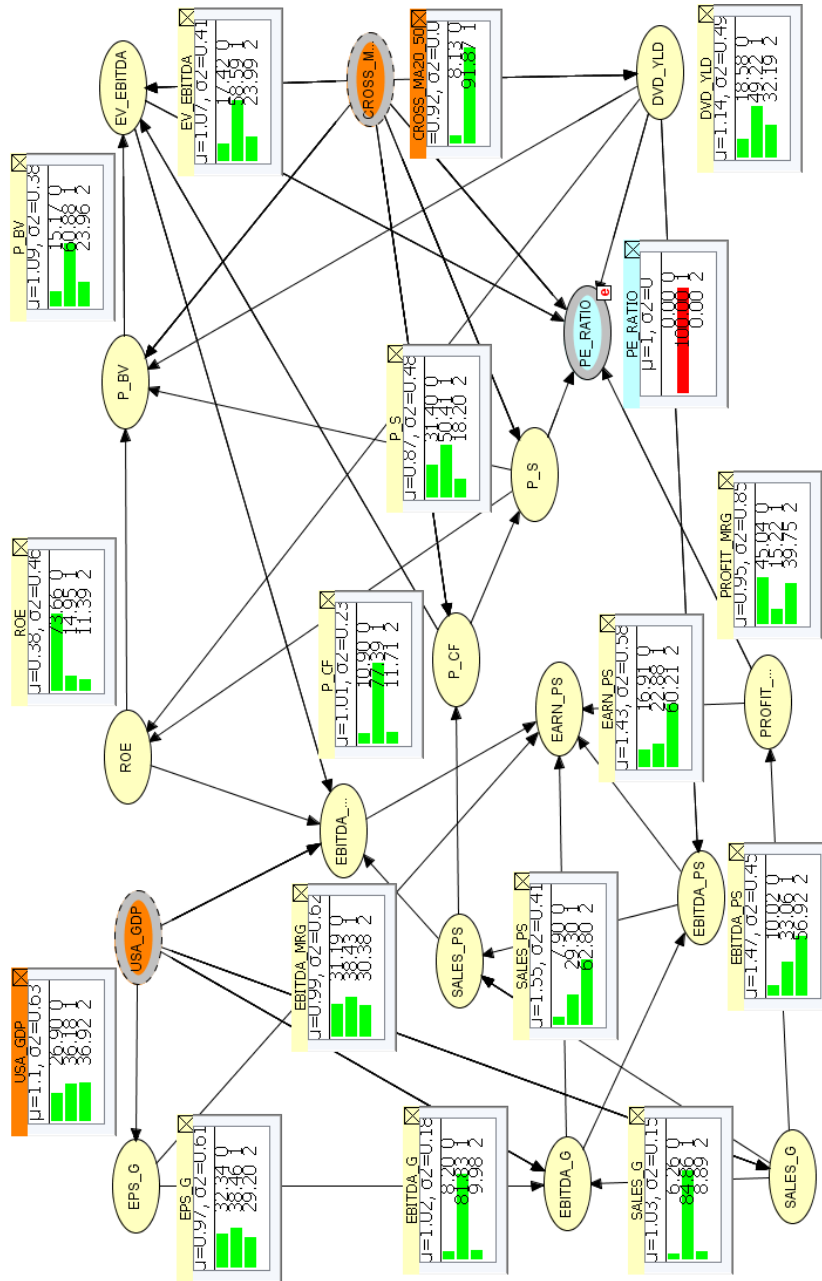


Figure 4.40: Micro Dimension OOBN: High Price to Earnings Ratio Scenario

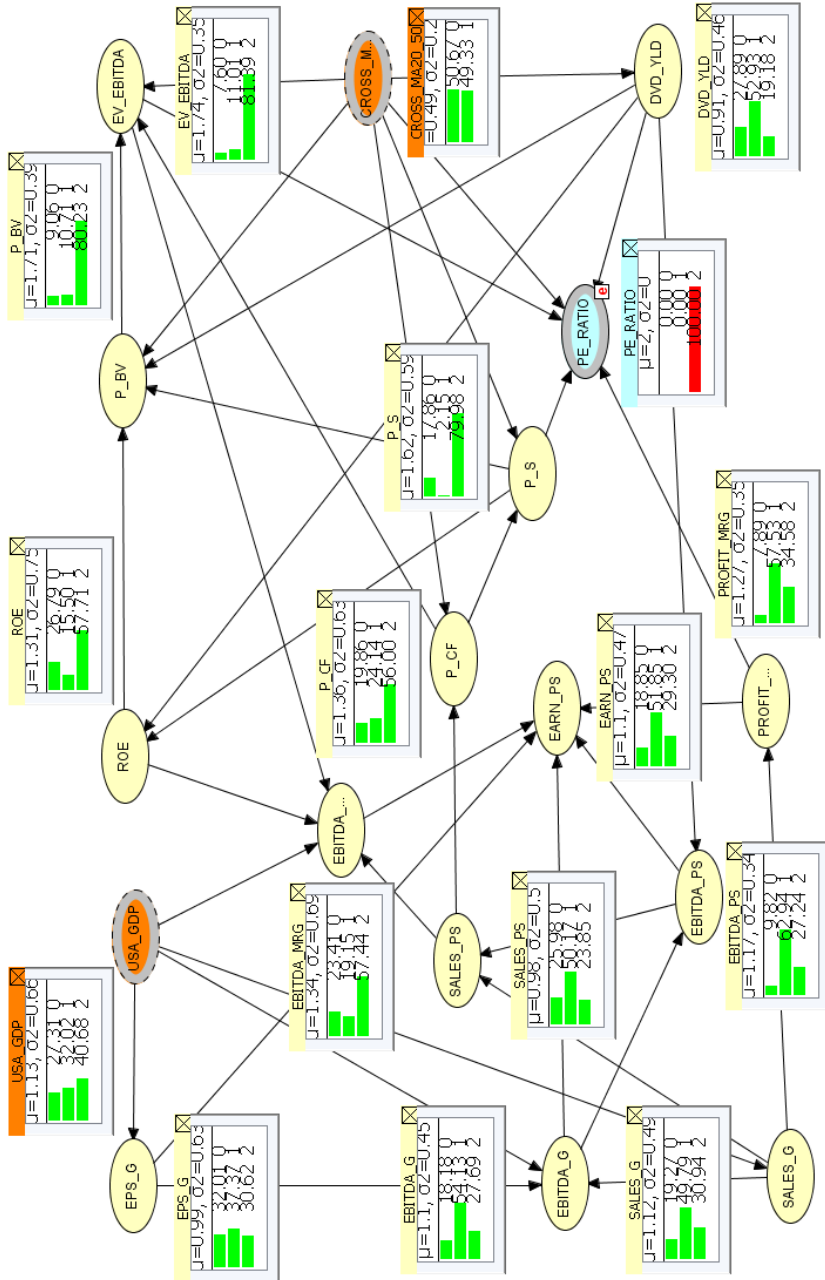


Figure 4.41: Micro Dimension OOBN: Low Price to Earnings Ratio Scenario

- C) In this scenario, Rsi and Vola_Spread are equal to 1, while USA_Gdp is equal to 2. Then we compare the previous trading indication with the one obtained when Vola_Spread is equal to 1, while USA_Gdp and Rsi are equal to 2.
- D) In this case, we observe how the probability referred to Rsi, Pe_Ratio, Vola_Spread and Dxy_Index changes when B_S_SPX is equal to 1 or 2.
- E) In conclusion, we observe how gold price, an encapsulated node belonging to the macroeconomic area, influences B_S_SPX

Simulating Scenario A

We now simulate a scenario where Dxy_Index, Pe_Ratio and Rsi equal to 1, see Figure 4.42. These implications mean that we have at the same time a strong dollar, a condition that reflects the strength of the American economy, an “expensive” market (high P/E) and an overbought configuration, corresponding high Rsi. Thanks to our model, we obtain a buy recommendation as suggested by the increase of the probability associated to B_S_SPX state 1 that increases from 50.55% to 79.11%. As we can notice, the OOBN allow us to have a real time feedback on a complex scenario that takes into account a large amount of information belonging to different areas. On the other hand, Figure 4.43 represents a scenario where Pe_Ratio and Rsi are equal to 1, while Dxy_Index is equal to 2. These evidences imply that the stock market is expensive and overbought, a context where practitioners generally expect a market sell-off. In addition, we add to this scenario the information about the weak dollar. Thanks to the OOBN, we observe that the weak dollar, low Dxy_Index, is enough to trigger a sell signal, corresponding to B_S_SPX state 2, whose probability jumps from 31.70% to 88.93%. According to our previous analysis, we have observed that generally Dxy_Index weakens if USA_Gdp growth slows. Then, we can derive that the market can tolerate high valuations (high P/E) only if the American Gdp keep on growing, otherwise a sell indication is triggered.

Simulating Scenario B

In Figure 4.44, we simulate the effects Dxy_Index, Vix, Rsi and Pe_Ratio equal to 1, that correspond respectively to a strong dollar, high volatility, overbought and expensive market. More in detail, the information about the dollar (equal to 1) indicates that the American economy is in good shape; high volatility is generally a bearish signal for equity markets; high Rsi indicates that the market has been “heavily bought” in the previous days, while the high Pe_Ratio indicates that investors’ exuberance drove high the stock prices. In this framework, the model provides a buy indication with the 89.96% of probability. We now observe how the recommendation changes if Pe_Ratio is equal to 2 (“cheap market”) and Dxy_Index, Vix and the Rsi are equal to 1, see Figure 4.45. In this simulation, the model provides a sell indication. This result indicates that Pe_Ratio can discount in the present the optimism for future growth. Thanks to our model, we can simplify this complex framework and obtain in real time an indication that would be hard to reach by using the tools generally adopted by practitioners.

Simulating Scenario C

We now simulate a scenario where we have Rsi and Vola_Spread equal to 1, while USA_Gdp is equal to 2, see Figure 4.46. Our assumptions implies that the negative macroeconomic outlook (low USA_Gdp) is negative, the spread between implied and

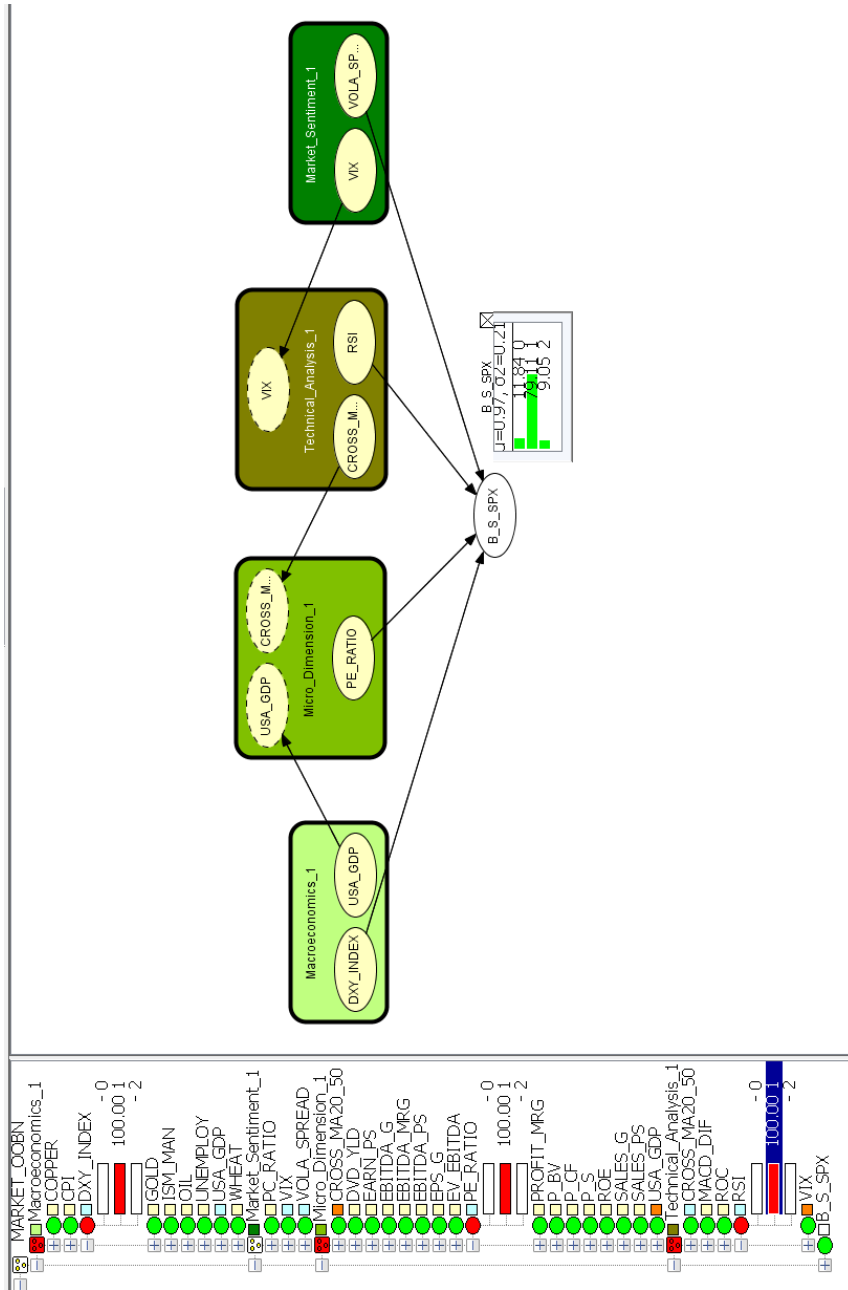


Figure 4.42: Scenario A1: Dxy_Index, Pe_Ratio and Rsi are equal to 1

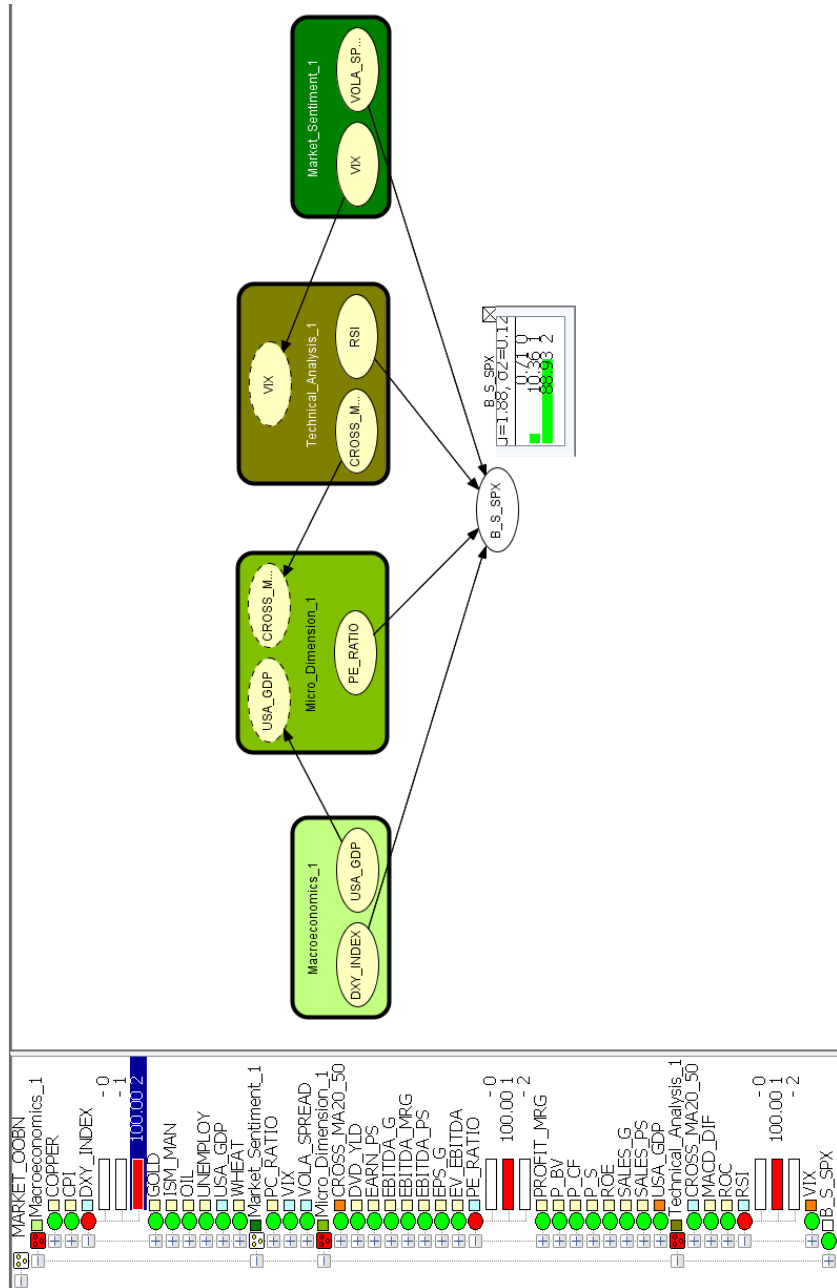


Figure 4.43: Scenario A2: Pe.Ratio and Rsi equal to 1; Dxy.Index equal to 2

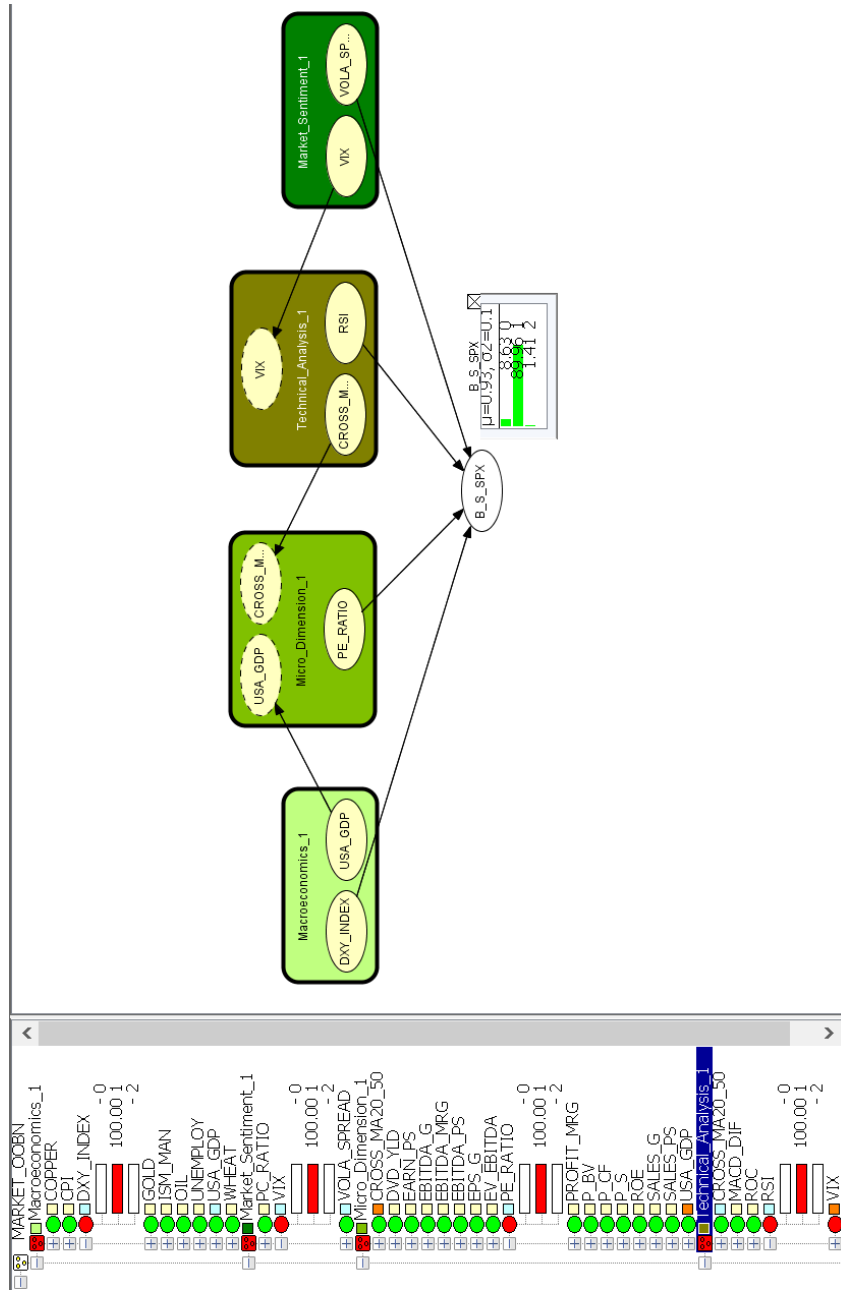


Figure 4.44: Scenario B1: Dxy_Index, Vix, Rsi and Pe.Ratio equal to 1

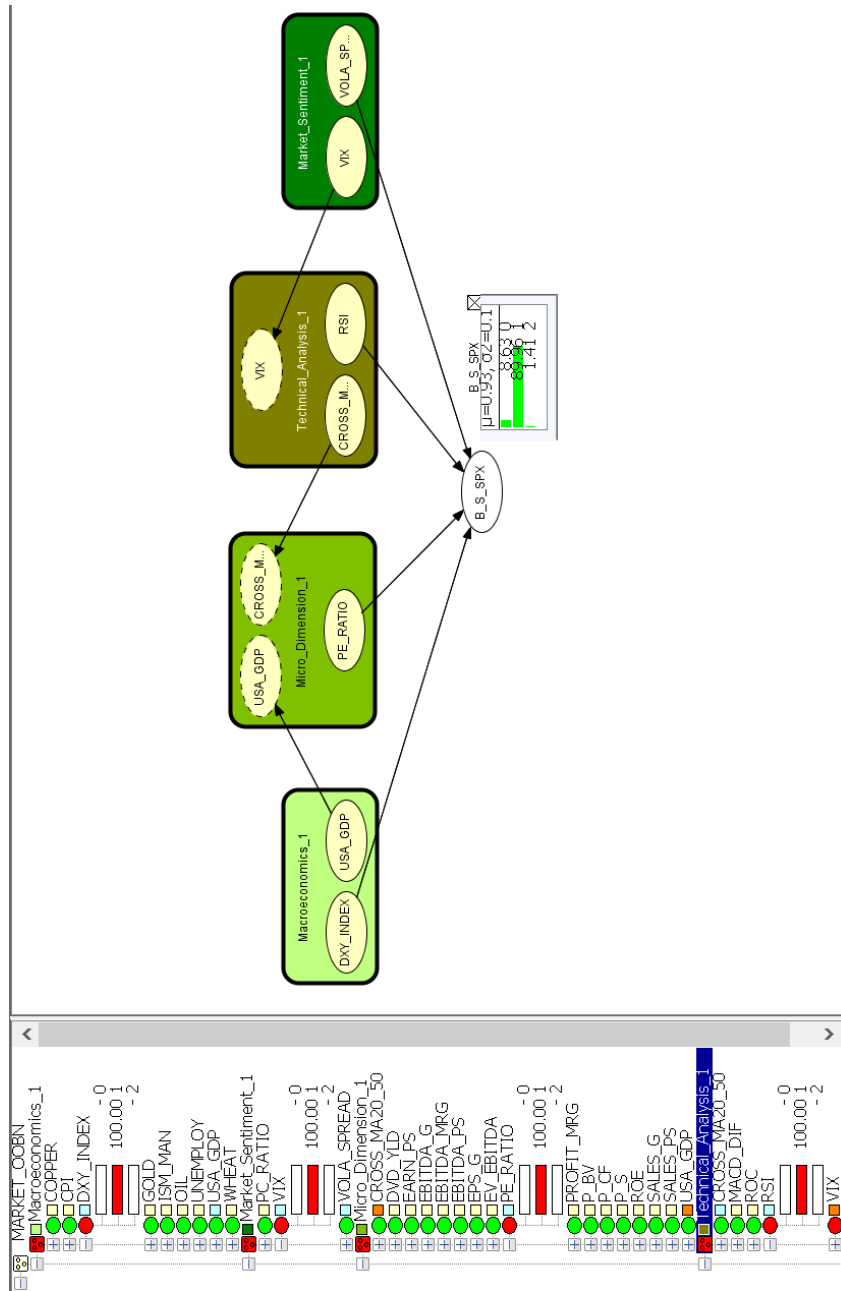


Figure 4.45: Scenario B2: Dxy_Index, Vix and the Rsi equal to 1 and the Pe_Ratio equal to 2

realized volatility is high and that the market is currently overbought (high Rsi). In this case, the output provided by our model is a sell recommendation with a 51% probability, while it associates a 44.15% probability to the buy indication. We now observe what happens if Vola_Spread is equal to 1 and USA_Gdp and Rsi are equal to 2, see Figure 4.47. This scenario implies that the realized/implied volatility spread is high, while the economic growth is weak and the market is oversold. Our model provides a buy recommendation, B.S.SPX state 1 increases from 50.55% to 52.37%, that incorporates quickly and efficiently all the evidences outlined above. This result reveals that the RSI drives investors' operative decision even if the macroeconomic environment is adverse. Consequently, the oversold market seems to represent a buy opportunity where investors can take advantage if the market rebounds.

Simulating Scenario D

In this scenario we observe how the probability associated to the output nodes directly connected to the response variable (Pe_Ratio, Dxy_Index, Rsi, Vola_Spread) changes when we set B.S.SPX equal to 1 (buy) or 2 (sell). A full picture of the probabilities associated to Pe_Ratio, Dxy_Index, Rsi and Vola_Spread before introducing any evidence about B.S.SPX is provided in Figure 4.48. We now simulate the impact of B.S.SPX equal to 1 (buy indication) on Pe_Ratio, Dxy_Index, Rsi, Vola_Spread, see Figure 4.49. In line with the results observed in other scenarios, the probability associated to Dxy_Index state 1 increases from 39% to 48.87%, Vola_Spread state 0 slightly increases from 37.98% to 40.73% (state 1 and 2 are substantially stable), Pe_Ratio state 1 probability decreases from 41.23% to 35.93% and its state 0 goes from 31.77% to 49.72%, while Rsi state 2 probability increases from 38.34% to 46.50%.

In conclusion, our model allows us to observe that we should buy S&P 500 when the dollar is strong, a situation compatible with growing American economy, and the Pe_Ratio is high, a scenario that suggests investors' optimism. In the following scenario we observe how B.S.SPX equal to 2 (sell indication) influences the Pe_Ratio, Dxy_Index, Rsi, Vola_Spread, see Figure 4.50. The model allows us to observe that Dxy_Index state 2 probability increases from 38.81% to 49.88%, Pe_Ratio state 1 probability goes from 41.23% to 51.07%, Rsi state 1 probability goes from 40.51% to 56.34%, while the probability associated to Vola_Spread state 1 slightly increases from 40.15 to 41.65%. The indications provided in this simulation show us that we should sell or reduce the exposure to S&P 500 outlook when the macroeconomic outlook is deteriorating.

Simulating Scenario E

We conclude our simulations by showing that the Hugin software allows us to perform simulations by changing the states associated to encapsulated nodes. For example, we can intervene on the Gold node, a safe haven commodity that generally is purchased by the investors when market volatility is too high or when inflation is rising. In the case Gold is equal to 1, see Figure 4.51, we observe that the probability associated to B.S.SPX state 1, corresponding to the buy signal, decreases from 50.55% to 44.22%, while the one associated to the sell indication, B.S.SPX state 2, goes from 31.70% to 37.05%. On the other hand, if Gold is equal to 2, see Figure 4.52, the probability associated to B.S.SPX state 1 jumps from 50.55% to 63%.

This result indicates that low Gold represents a positive signal for the American equity market because investors will be more prone to invest in risky assets.

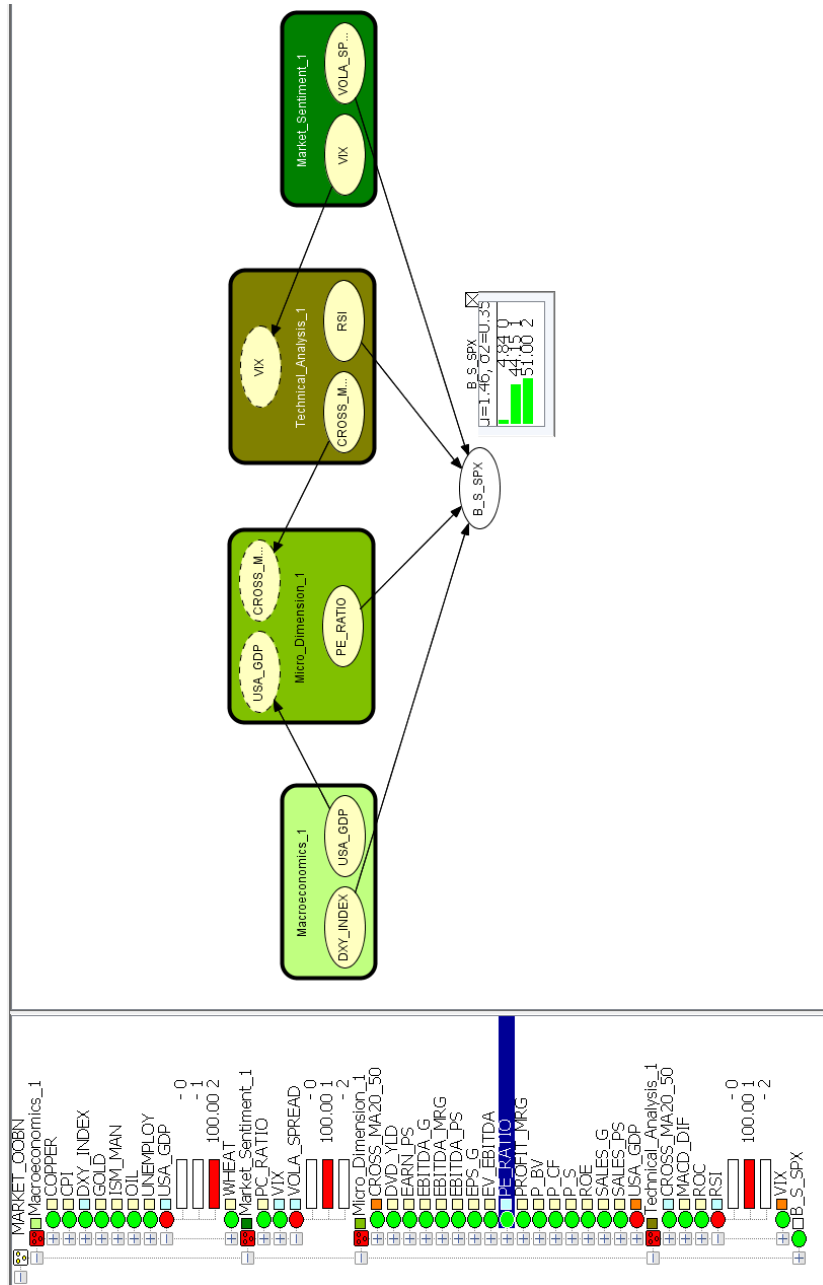


Figure 4.46: Scenario C1: Vola_Spread and RSI equal to 1; USA.Gdp equal to 2

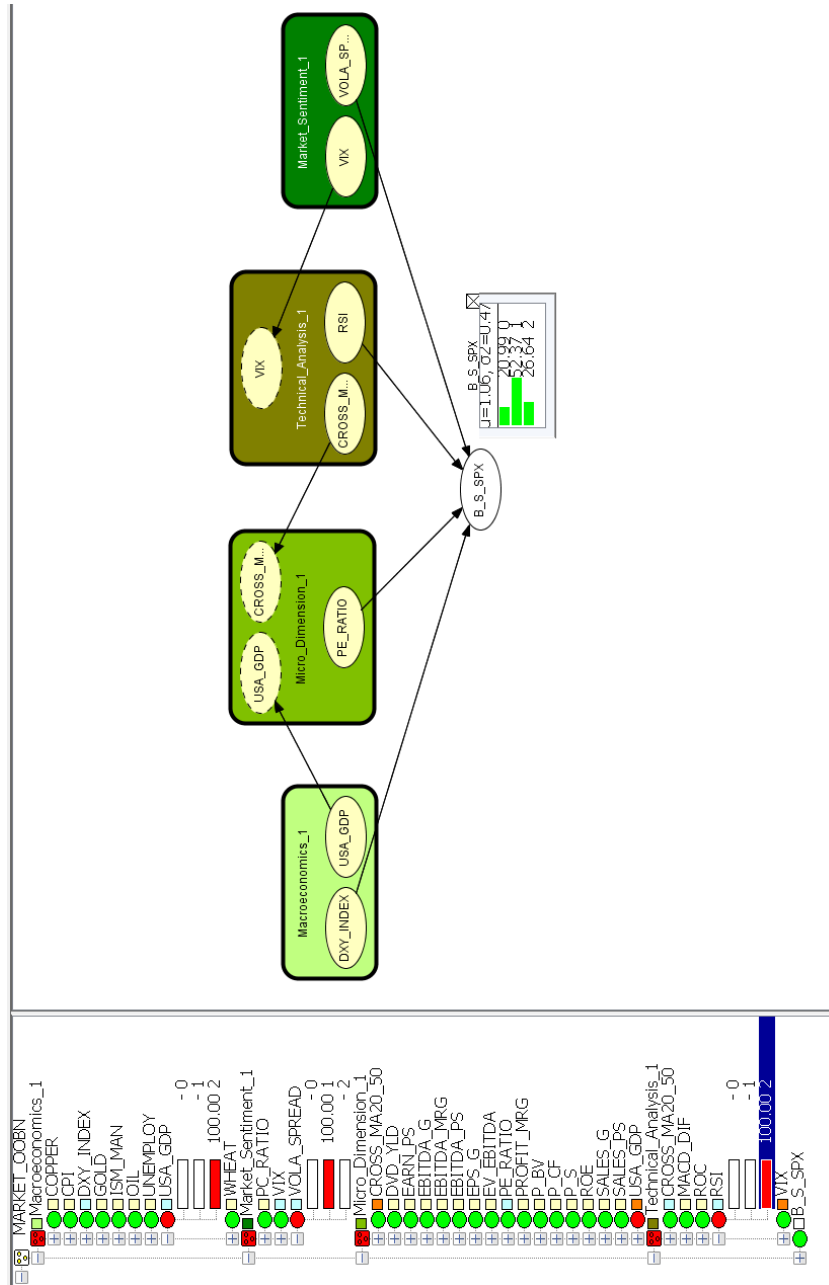


Figure 4.47: Scenario C2: Vola_Spread equal to 1; USA_Gdp and RSI equal to 2

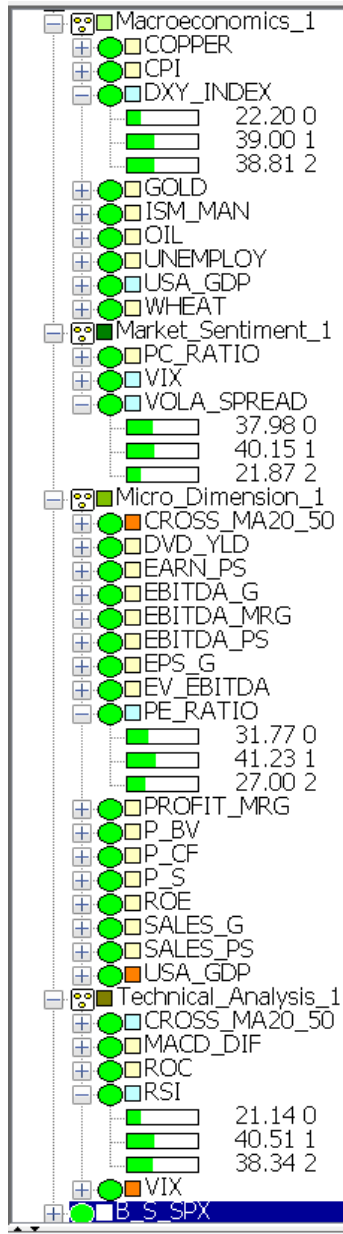


Figure 4.48: The Priors for Pe_Ratio, Dxy_Index, Rsi andVola.Spread

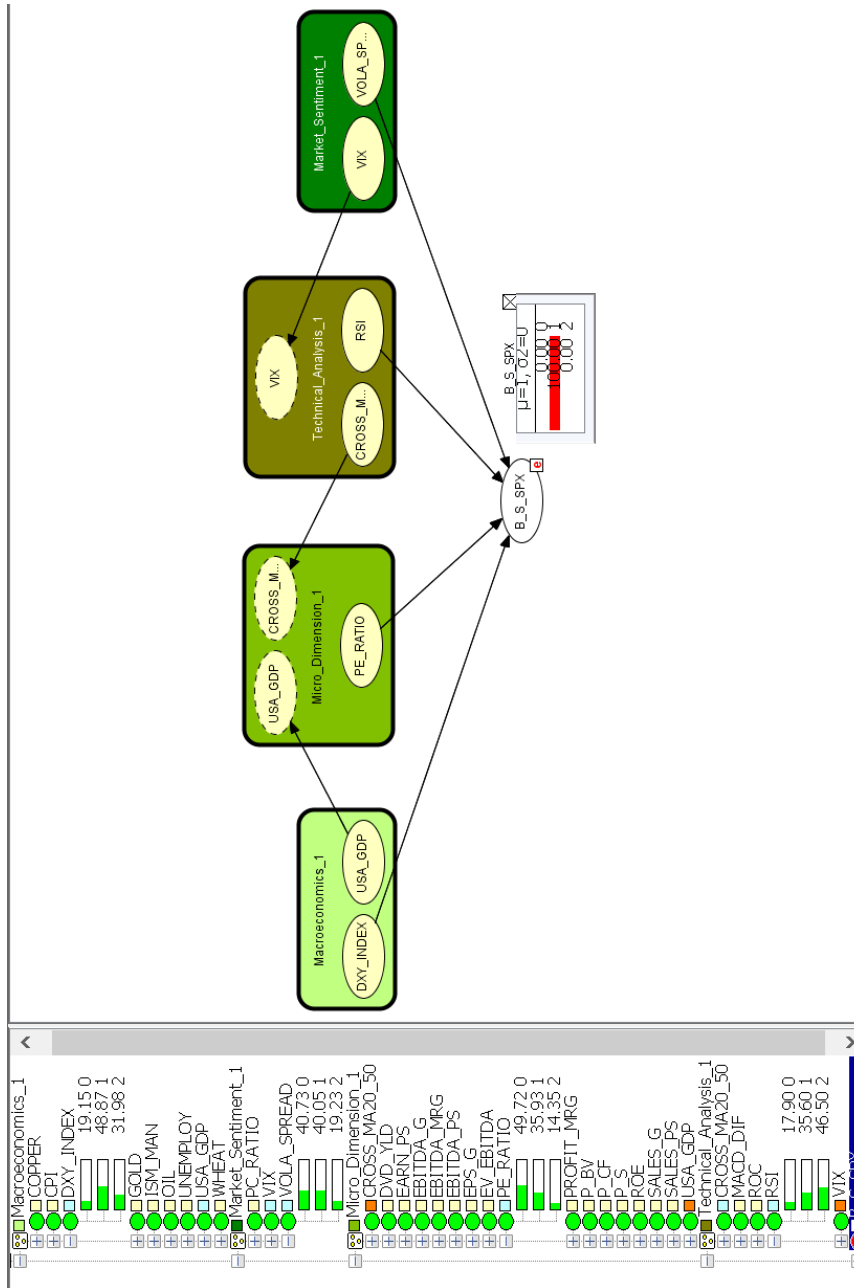


Figure 4.49: Scenario D1: The Contrarian Buy Scenario

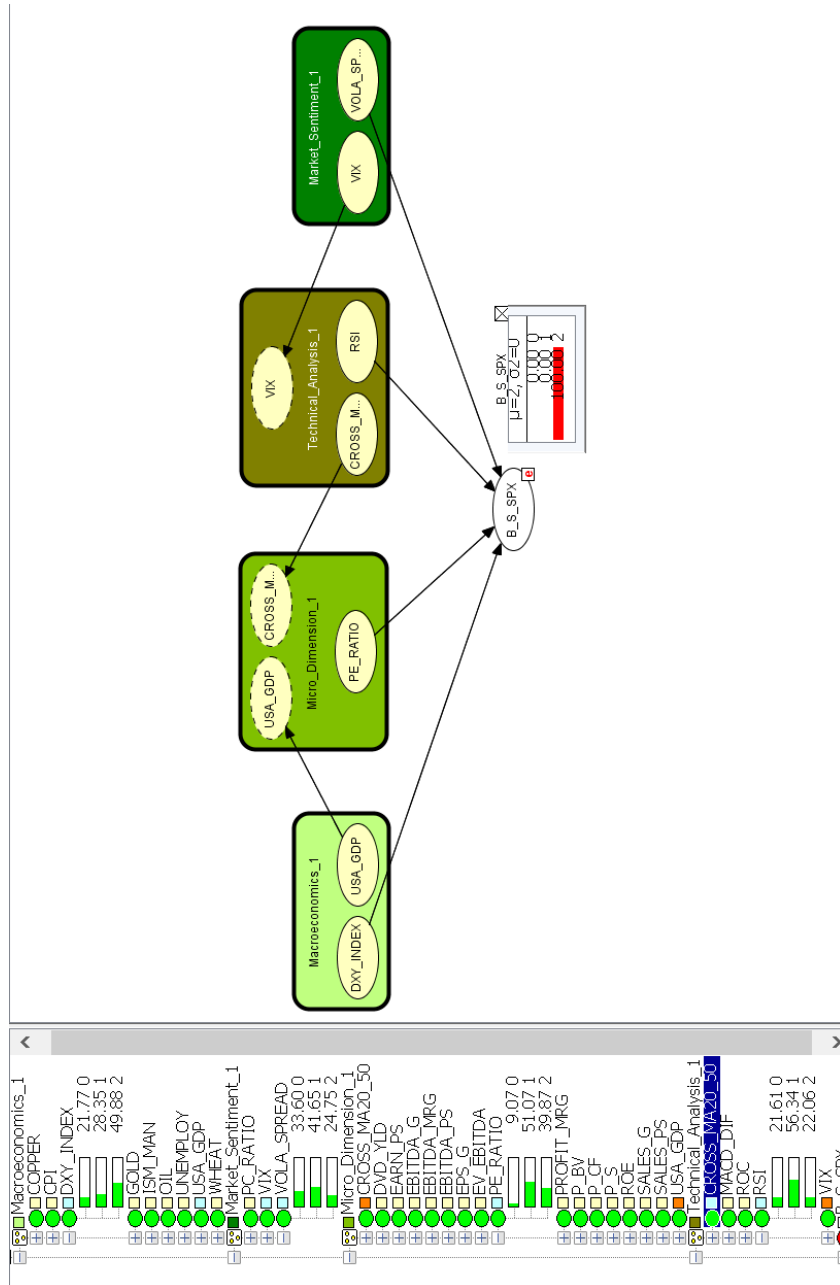


Figure 4.50: Scenario D2: The Contrarian Sell Scenario

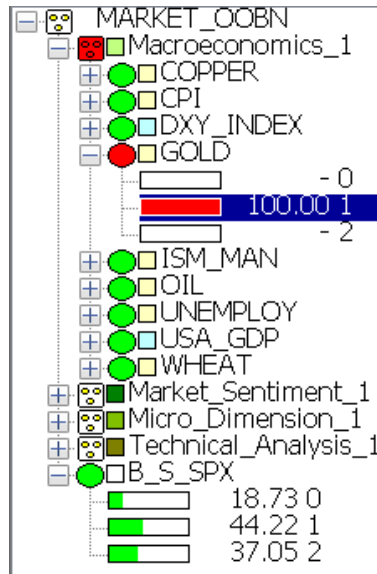


Figure 4.51: Scenario E1: The Effect of High Gold Price on the Target Variable

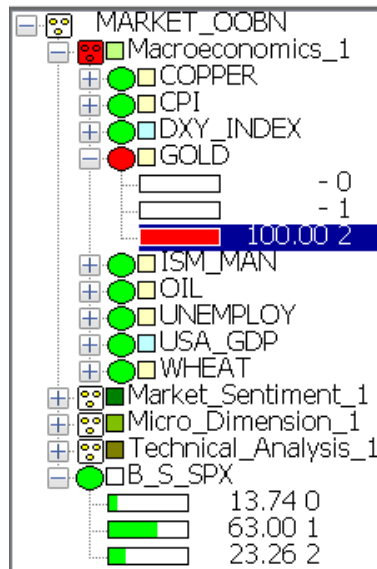


Figure 4.52: Scenario E2: The Effect of Low Gold Price on the Target Variable

4.6 What Practitioners Actually Do and Why This Model Is Innovative

We dedicate this section to compare our model with the analytical tools generally used by professional investors. In the following list, we propose the common approaches followed by practitioners during their decision making process:

- they perform some calculations (basic statistics or regressions)
- they meet strategists and sell-side analysts
- they study research papers on specific asset classes, regions or sectors
- they exploit the tradable information that appears on their terminals (Bloomberg, Factset, Datastream)

The operations listed above are time consuming if compared to the speed of computers and algorithms. Moreover, these elaborations are generally focused on a single portion of the market (i.e. a particular sector, a geographical area ...) and consequently the investment indication provided does not incorporate all the information available in a particular moment. For example, we now analyze the matrix elaborated by Bridgewater, a well-known asset manager, for its “All Weather” fund, see Figure 4.53. (<http://www.bwater.com/Uploads/FileManager/research/All-Weather/All-Weather-Story.pdf>). Thanks to this representation, we observe that different expectations on the macro scenario (a combination of high/low growth and high/low inflation) drive the investment decisions and the money allocation among asset classes.

		Growth	Inflation			Growth	Inflation
MARKET EXPECTATIONS	Rising			Rising	25% OF RISK Equities Commodities Corporate Credit EM Credit	25% OF RISK IL Bonds Commodities EM Credit	
	Falling			Falling	25% OF RISK Nominal Bonds IL Bonds	25% OF RISK Equities Nominal Bonds	

Figure 4.53: An Example of Practitioners’ Tool: Bridgewater’s Growth-Inflation Matrix

This matrix represents a powerful tool because it makes easier to understand complex dynamics. However it provides a static representation of financial markets and any buy or sell decision relies on the ability of a fund manager to interpret few variables. In addition, this model does not allow any customization or real time simulation.

On the other hand, our work underlines the potentiality of OOBNs in improving investors’ decision by providing not only insightful recommendations but also real time signals that deal efficiently with markets complexity and dynamics. We should also consider that practitioners’ decisions are often based on the financial industry consensus

and for this reason they are potentially biased. The anchoring bias consists in relying too much on a benchmark when asset managers take an operative decision. Because of this, institutional investors are prone to adopt very similar strategies counting on the fact that if they are wrong on that trade, the majority of the fund managers is in troubles too. Being the “voice out of the choir” could be too risky in certain situations even if an institutional investor is right. However, thanks to our graphical representations we make easier to interpret the financial markets and observe how they evolve quickly. For this reason, the fund managers need a tool that shows how the environment is changing and that reduces the impact of cognitive biases. Moreover, thanks to OOBNs they observe not only financial market dynamics but also they are able to know the likelihood of a specific outcome. Market operators can expect some of the indications obtained, however our model provides further strength to common believes by attaching a probability to the results. By introducing OOBNs for market signals detection, we propose a solution that overcomes some of the limits evidenced by the common tools. In addition, it represents a step beyond because it allows performing real time simulations on a complex framework. However, the objective of this work is to provide an advisory tool for fund managers and not to substitute traders. More in detail, this work deals with augmented intelligence because the market operator exploits the computer calculation power in order to improve the quality of its decisions. Augmenting human intelligence consist in exploiting a computer to compensate practitioners’ cognitive limits in order to obtain better and more precise indications (Schmitt, 1998). In conclusion, we state that the introduction of technological innovation in the financial world can change how practitioners actually work by reducing market inefficiencies and the impact of information asymmetries.

Conclusions and Future Extensions

Nowadays, financial markets are more interconnected than ever and their complexity does not allow understanding who are the factors that drive assets prices. The tools currently adopted by practitioners provide only a partial view of the whole framework because they focus on a limited amount of variables. Moreover, these procedures are generally time consuming and the results obtained can be biased by the irrationality of the investor.

In order to support fund managers, we have proposed an innovative model that supports fund managers in handling a continuously evolving environment. More in detail, in this work we exploit the potential of graphical models, first by adopting BNs (Chapter 3) and then the OOBNs (Chapter 4). A BN allows simplifying the financial market framework thanks to its graphical representation. However, they show some limit in dealing with very large and complex problems. In order to overcome this weakness, we introduce OOBNs that still allow combining at the same time different objects and sources of information (i.e. the economic areas and indicators that influence equity prices). The OOBNs exploit some of the basic elements of object-oriented programming, such as the concepts of object and hierarchy, and they allow us to simplify the visualization of the problem at hand by dividing it in several blocks, called instance nodes. Thanks to this approach, the graphical representation is less cluttered and easier to read, with some benefits also for the information propagation algorithms performance (Koller and Pfeffer, 1997). We remark that our work provides new contribution to the financial market analysis because:

1. Looking for financial markets signals detection by exploiting graphical models is an innovative approach to a well-known problem.
2. The tools currently adopted by practitioners do not consider at the same time all the variables or areas that we included in our simulations.
3. Graphical models allow performing scenarios in a mouse click and to obtain quantitative results in real time.
4. Even if some of the results obtained can be expected, our model indicates in real time their likelihood to happen.

As anticipated in point 1, the previous BNs applications to economics do not focus on detecting equity markets inner dynamics. In Chapter 3 we exploit BNs to understand how the market evolved in the last 20 years by comparing two periods (pre and post Lehman Brother bankruptcy) characterized by different maturity of financial instruments. One of the most important findings is that the Price to Earning Ratio, generally

considered as the key ratio for market valuation, has a limited impact on the buy and sell signals. Moreover, the model allows us to observe that the Vix, a variable known as the proxy of investors' fear, gained relevance in the recent years in determining market dynamics. These relevant results drove us to extend our analysis and to adopt the OOBNs. This extension allows us to simplify further the graphical representation of the market. Instead of visualizing the network as a "cloud" of nodes, thanks to the OOBNs the user visualizes only the input and output attributes. This approach allows us to represent in the same model four different relevant areas (Macroeconomic, Micro Dimension, Sentiment Area and Technical Analysis) and our target variable B_S_SPX. As a consequence of OOBNs adoption, we can add more variables to our model but at the same time the graphical representation remains tidy and clear.

As underlined in point 2, graphical representations allow to deal at the same time with consistent amounts of variables. For example, thanks to the BN introduced in Chapter 3 we observe that the inclusion in the same framework of information belonging to different areas allow us to update investors' knowledge. In Chapter 4, we show that OOBNs allow us to increase the amount of data analyzed without losing the immediacy provided by the graphical representation.

According to the results obtained in this work, summarized in point 3, markets evolve quickly and fund manager should be able to make a decision in a short time. The approaches followed by practitioners are generally time consuming and the interpretation relies on a subjective decision. By proposing our work, we offer to strategists or fund managers a tool that increases the amount of information elaborated simultaneously and that potentially improves their decisions.

As anticipated in point 4, some of the results obtained with our model are consistent with the common financial knowledge. However, we provide more strength to this knowledge by attaching a probability to the result obtained. Moreover, the model allows us to observe dynamics and relations among variables that are not caught by following standard approaches. The results obtained in Chapter 3 and 4, are supported by a rigorous algorithmic approach and this confirms the positive impact of technology in finance, a phenomenon born in 2009 and called *fintech*. In light of the results obtained, some interesting research direction can be the following:

- Exploring more in detail market dynamics, for example the ones referred to single sectors (i.e. financials; utilities; industrials ...).
- Performing stock picking on single stocks. This task consists in determining which security is overpriced or underpriced. This analysis allows the investor to make arbitrage, a strategy that consist in buying low and then selling high an asset.
- Expanding this model to other asset classes such as bonds, currencies and commodities.

Innovation in finance is spreading at a fast pace because information on markets and retail customers are easier to collect and this allows to offer "tailor-made" products. Consequently, even a Silicon Valley startup can challenge well-established financial players that are generally anchored to obsolete approaches.

Appendix 1: Introduction to the Hugin Software

We now propose in Appendix 1 and Appendix 2 a detailed guide that shows how to build BNs and OOBNs with Hugin. This material has been included because learning how to use the software represented a key part of the work. We remark that currently it is not available a structured guide on Hugin but only a brief tutorial that do not show all the passages required to build graphical models. For this reason, the material collected in Appendix 1 and 2 represents an important source of knowledge that can be exploited in other works dealing with Bayesian Networks.

4.7 Brief History of Hugin

Leading researchers in the field of graphical models have founded in the late 1980s in Aalborg (Denmark) the company that initially developed Hugin under the EU sponsored program ESPRIT. This software is a decision support tool that allows building models and finding solutions in various fields where is necessary to reason in a context of uncertainty such as diagnostics, risk management or anti-fraud. More in detail, the software has been optimized to deal with statistical models such as Bayesian Networks and influence diagrams. Over the years, the system has been updated by including new models including the Object Oriented Bayesian Networks. Thanks to its easy to use graphical interface, it allows to learn directly from the data.

4.8 How to Build a BN with Hugin

When the user runs the software for the first time, the window shown in Figure 4.56 opens up. In the main screen, we have a menu bar, a tool bar, and a document pane. When the software is launched, the document pane is empty.

4.8.1 Adding a New Node

The first step for building a BN is to generate new nodes. In order to create them, we click in the toolbar on the discrete chance tool. Once this feature has been selected, we can click in the document pane (blank area) and then a node C1 appears. The software allows personalizing the name of each variable belonging to the network. This task can be accomplished by double-clicking on the node that we want to rename. Then a new window opens, and by selecting the “Node” label, we can change the name and the label associated. Then we click on the “Ok” button. If we do not specify name

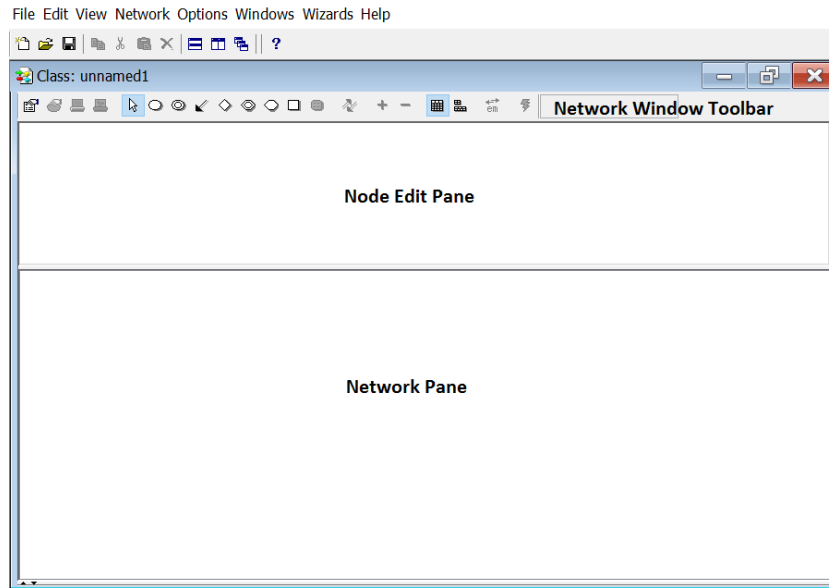


Figure 4.54: A Screenshot of Hugin's Network Window

and label, the software uses the same one associated to the internal name of the node. Now we propose a toy example where we generate a node named “Bond_Emission”. In Figure 4.55 we show were to find in Hugin's toolbar the node properties tool, the discrete chance tool and the arrow tool.



Figure 4.55: Hugin's Network Toolbar. Evidenced by the Squares, from the Left to the Right: the Node Properties Tool, the Discrete Chance Tool and the Arrow Tool

Then the nodes “Capital_Increase” and “Stocks_Goes_Down” are added to the network by following the same procedure illustrated above. A trick that allows us to avoid selecting for each variable the discrete chance node is to press and hold down the Shift key and then click all the times needed on Hugin's network pane in order to generate a new node for each click.

4.8.2 Adding Directed Arrows Among Nodes

Once we have created the nodes, the BN looks like the one in Figure 4.56. In order to add a directed link from a variable to another one, we press the arrow tool (Figure 4.57) and then we drag a directed arrow from the node “Bond_Emission” to “Stocks_Going_Down”. If we hold down the Shift key, we avoid selecting every time the arrow tool and we can generate all the links needed just by connecting the nodes. Then, we complete our BN by connecting “Capital_Increase” to “Stocks_Going_Down”. Now we have a qualitative representation of our problem. The next step is to add the quantitative element to our BN by filling the conditional probability tables (CPTs) associated to each node.

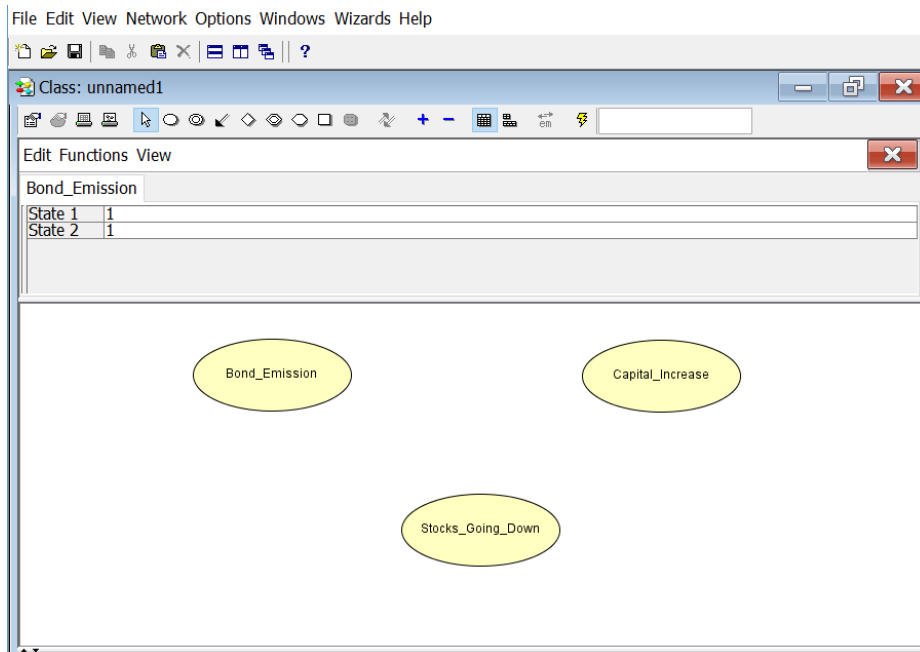


Figure 4.56: The Nodes Belonging to the BN



Figure 4.57: The Arrow Tool Button

4.8.3 Specifying the States Associated to Each Node

In our toy example, we assume that each variable of the network (“Capital_Increase”, “Bond_Emission”, and “Stocks_Going_Down”) has two states: Yes or No. Now we specify how we set the states for the “Capital_Increase” variable. The same procedure is conducted for the other nodes belonging to the BN. First, we click on the “Capital_Increase” node. Now that the variable is active, we click on add/remove states tool. Then, by double clicking on the node, we can explore the states label where we could change the name associated to a state. The procedure consists in selecting and then editing the text referred to State 1 and State 2. In this toy example, we type Yes and No in correspondence of each state. The same procedure should be followed for the other nodes belonging to the network. We underline that the CPT associated to “Stocks_Going_Down” is larger than the other ones because it has two parent nodes: “Capital_Increase” and “Bond_Emission”. On the other hand, the parent nodes cannot be the children of any other one.

4.8.4 Entering Values into a CPT

The last step is represented by attributing to each state its conditional probability. We these values for each node in Figures 4.60, 4.61 and 4.62.

All these values should be entered in the CPT table by using the Hugin GUI. We now illustrate how to fill the CPT for the “Bond_Emission” node. First, we should select

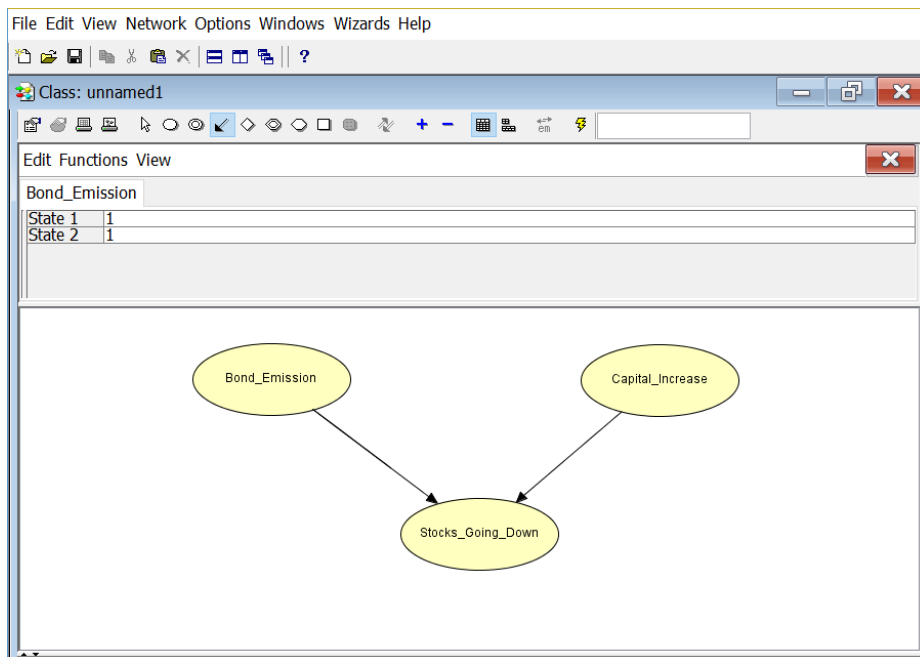


Figure 4.59: The Add/Remove States Tool

the “Bond_Emission” node, then we right click on it and then we select the option “Open tables”. By double clicking on the field representing the state Yes, we can change the probability associated to this event. By entering the value 0.1. We perform the same action with the No state and we type the value 0.9. The same procedure is followed for the other nodes, “Capital_Increase” and “Stocks_Going_Down”. Now that we have built our BN with Hugin, we save our output. By clicking on the top left corner the “File” label, and then by selecting “Save As” we can choose the directory where we want to save our BN. Now that our work has been saved, we can run our simulation.

Bond_Emission	
Yes	0.1
No	0.9

Figure 4.60: $P(\text{Bond_Emission})$

Capital_Increase	
Yes	0.1
No	0.9

Figure 4.61: P(Capital_Increase)

Stocks_Going_Down				
Edit Functions View				
Capital...	Yes		No	
Bond E...	Yes	No	Yes	No
Yes	0.95	0.85	0.9	0.02
No	0.05	0.15	0.1	0.98

Figure 4.62: P(Stocks_Going_Down, Bond_Emission, Capital_Increase)

4.9 Structural Learning

In the previous paragraphs, we have presented how to build from the scratch a BN with Hugin. Another important feature of this software is represented by the possibility of structural learning by exploiting several algorithms.

4.9.1 Constraint Based Algorithms

The objective of a constraint based algorithm is to determine the validity of an independence relation by exploiting any possible information available. Furthermore, it allows the expert to exploit its knowledge before starting the testing process by letting him include some constraints. For example, the user can indicate the absence or the presence of a hedge between two variables and the orientation of the link (Kjaerulff and Madsen, 2013).

The PC Algorithm

The PC algorithm (Spirtes and Glymour, 1991; Spirtes et al. 2000) steps can be summarized by introducing a list of necessary and sufficient rules (Madsen et al, 2003):

1. The software tests for conditional independences among a set of variables
2. The skeleton of the network is drawn
3. Colliders are identified
4. The direction of the arrows is determined

The output of the PC algorithm is a PDAG. In the first step, we assume that the variables X and Y are independent given S_{xy} . The software tests statistically by adopting sets S_{xy} of size 0, 1, 2, 3. If the assumption $X \perp\!\!\!\perp Y \mid S_{xy}$ is significantly

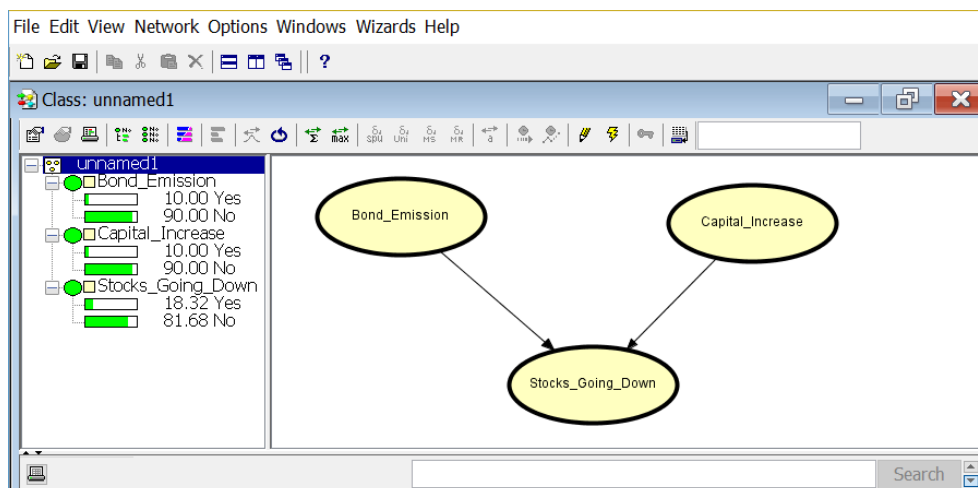


Figure 4.63: The Results of the Simulation

satisfied, the software stops testing for independence the variables X and Y . In the case that the software does not find a conditional dependence among the variables X and Y these variables will be connected by an undirected link. This procedure allows the user to obtain the skeleton of the network. The skeleton is the undirected graph obtained by removing the direction from all the arcs belonging to the same BN (Nielsen, T. D. and Jensen, F. V., 2009). Then, a collider is created at Y if X and Y are neighbor variables, Z and Y are neighbors too, X and Z are not close to each other, and it is true that $Y \in S$ for any S_{xy} (the condition $X \perp\!\!\!\perp Z \mid S_{xy}$ is satisfied) (Madsen et al., 2003). Implicitly we are assuming that if the first rule does not hold, and there is not an identified collider, the other ones are the guarantee that we have not created a loop in the network.

The NPC Algorithm

The NPC algorithm is an extension of the PC algorithm that includes the concept of necessary path condition (NPC). The NPC condition states that two variables X and Y are conditionally independent on a set S and no subset $s \subset S$, if there is a path between X and every $Z \in S$, that does not cross Y , and between Y and every $Z \in S$, that does not cross X (Steck, 1999). We remember that the Hugin software allows the user to solve the NPC algorithm uncertainties by choosing if two nodes should be connected. Moreover, the NPC allows introducing some prior constraints according the user prior knowledge.

4.9.2 Search and Score Based Algorithms

These algorithms require for their optimization a scoring function and a search strategy. Using a search or a score based approach implies that we are dealing with a graph $G = \{V, E\}$ and that during the structural learning we are looking for the DAG structure with the optimal score.

Search Algorithms

All the existing search procedures use strategies based on greedy hill climbing that consists in selecting during each step the neighbor of the structure with the highest score at that moment. By following this approach, we are looking to improve the structure quality by reaching the highest possible score. The hill climbing algorithm can be summarized in the following steps (Kjaerulff and Madsen, 2013):

1. We start with an initial graph structure
2. A set of structures is generated using the search operators
3. The score for each structure is computed
4. The structure with the highest scores is chosen
5. We use the selected structure for running the next iteration
6. The steps from 2 to 4 are repeated until the score cannot be improved anymore
7. When the process is ended we obtain the graph structure with the best score possible

The Chow Liu Algorithm

This algorithm belongs to the category of the search procedures and it allows the user to obtain a tree model that maximizes the data likelihood. Furthermore, thanks to the Chow-Liu algorithm is possible to obtain the best possible approximation of the network such that the arrows connecting the nodes are directed in the opposite direction of the root of the network. The approximation quality is measured by using the Kullback-Leibler distance between the distributions obtained by running the Chow-Liu algorithm and the true one. The Kullback-Leibler distance consists in an information-theoretic distance measure (Johnson and Sinanovic, 2001) and it calculated as follows:

$$D(p_1||p_0) = \int p_1(x) \log \frac{p_1(x)}{p_0(x)} dx$$

where $p_1(\cdot)/p_0(\cdot)$ represents the likelihood ratio.

In the eventuality that we are learning the distribution directly from the data, we consider as true the one obtained from the frequencies associated to each observation. More in detail, the Chow-Liu tree is the maximum-weight spanning tree over all the variables considered. The weight associated to each edge corresponds to the mutual information between the nodes connected (Chow and Liu, 1968). The steps of the algorithm are the following:

1. It calculates the mutual information $MI(X_i, X_j)$ for each pair (X_i, X_j)
2. It considers the mutual information-weighted graph, that corresponds to the complete undirected graph over $\{X_1, \dots, X_n\}$, where the edges (X_i, X_j) have weights $MI(X_i, X_j)$
3. It builds a maximum-weighted spanning tree for the complete mutual information-weighted graph
4. It provides a direction to the tree that goes in the opposite direction of the root

Score Functions

As already mentioned before, several functions measure the quality of the graph structure according to the data available. For example, we can associate a score function to a likelihood function (Darwiche, 2009). In this case, the score function relies on the best possible guess based of the conditional probability distribution associated to a given graph structure and a dataset. The first step is to use the maximum likelihood estimation in order to determine the conditional probability distributions referred to our graph. Then we have to solve the overfitting issues referred to the likelihood function. In order to do that we use a score function because it considers the complexity of the representation: the higher the complexity, the higher the gain measured by the likelihood function.

Another example of score function is the Bayesian Information Criterion (BIC) that penalizes the likelihood by a term defined by the network structure. More in detail, if the database size increases, the penalty associated to the complexity goes down. A further example of score function that penalizes the complexity is the Akaike's Information Criterion (AIC). However, BIC penalty is higher than the one referred to AIC.

4.9.3 Restricted Models

In the case we are dealing with extremely large conditional probability distributions, the score functions can fail in their task of reducing the model complexity when we are dealing with a large amount of data. In order to solve this issue, we can learn a graph by adopting a restricted model such as naive Bayes or the already mentioned Chow-Liu trees.

Naive Bayes Model

This approach represents the simplest restricted model but it is also known for its high quality performances in the classification tasks. Similarly to Chow-Liu trees, we start the process by identifying a root variable while the remaining nodes are considered its children. Then the naive Bayes model determines the conditional dependences between the children of the root variable. Consequently, the model generates a DAG by drawing a directed edge from the root to the other variables and it defines the conditional probability distributions by using data frequencies or by exploiting the EM algorithm.

4.10 Other Relevant Algorithms

The EM Algorithm

The objective of an optimization algorithm is to estimate the parameters of a DAG. By using Hugin, this task is supported by the EM algorithm that calculates maximum likelihood and a posteriori it makes estimates when we have missing data (Lauritzen, 1995; Cowel and Dawid, 1992). In this section, we illustrate how we can use data in order to estimate conditional probability distributions and densities when we have only the graphical structure. This methodology is known as batch learning and it requires the availability of all the data at the beginning of the process. In this procedure, the *case* is an assignment of values to the nodes of the domain. If we have a value in correspondence of each node, the case is defined as complete; on the other hand, we

have an incomplete case if the values have not been attached to each node. Before the learning procedure starts, we should define the nodes on which we want to estimate probability distributions and densities. The input of the EM algorithm is represented by the set of cases. By exploiting Hugin's inferential engine, we can update the conditional probabilities or densities and consequently the tables associated to each node belonging to the domain. Whenever the experience tables show only zeros, the algorithm provides the maximum likelihood probability table. On the other hand, if the experience table is not made of zeros, the table is used to form prior counts that will be added to the ones derived from the data. Before running the EM algorithm, we have to specify conditional probability tables and their densities. Then the launched algorithm performs some iteration in order to calculate and optimize the log-likelihood of the data probability associated to the case given the current joint probability distribution. The number of iteration can be controlled in the following ways:

1. We can explicitly specify a number of iterations, where 0 means no maximum
2. If the maximum number of iterations specified is not exceeded, the EM algorithm stops running when the difference between the log-likelihood of two consecutive iterations is considered small enough

The Adaptation Algorithm

The Hugin software exploits also the adaptation algorithm, which updates the conditional probability distributions according to the evidence introduced (i.e. new information). This algorithm applies only to the discrete chance nodes and it allows performing what-if analysis and simulating different scenarios. This process allows performing simulations by changing the input information and by observing its impact on the other nodes. Thanks to this approach, we can evaluate if the result deviates from our expectations. Adaptation learning is useful when the model is incomplete, the domain is drifting over or it does not reflect properly the model. Another important feature adaptation algorithm feature is represented by the notion of experience, which represents a quantitative memory based both on expert valuations and on information gathered in the past (Spiegelhalter and Lauritzen, 1990). The experience is determined by running a set of counts. The counts should not be necessarily integers and they are contained in the so-called experience table. At the beginning, the table is filled with zeros that do not correspond to a valid count. Before running the adaptation algorithm, these values should be replaced by positive values. This approach is used to turn "on" or "off" the adaptation algorithm at the level of individual parent configurations: if the experience count is a positive number the algorithm is turned on; on the other hand if we have all zeros or negative values the algorithm is turned off.

4.10.1 The Learning Wizard

It is possible to be assisted during the learning procedure by Hugin's "Learning Wizard". We can open it by selecting in the main window the "Wizard" label and then by selecting "Learning Wizard". This feature allows to read data directly from text files (i.e. ".txt", ".csv"), and it provides some basic tools for preprocessing the data by excluding some variable, replace some value or discretize them. When the data treatment has been completed, the Learning Wizard allows the user to use choose some learning algorithm (i.e. Chow-Liu, PC, NPC...). This process can be divided in Structural Learning and Data Analysis. In the former, we can choose the

algorithm to adopt or add some constraints that incorporates our prior knowledge on a subject. This task can be very complex if the environment under analysis has a large number of variables. The wizard facilitates this procedure by letting the user to save all the information available about the network (nodes positions, colors, labels, and constraints). In the case the user adopts the NPC algorithm, Hugin allows to solve ambiguous regions or unsolved link directions by intervening directly on the network. In conclusion, the parameters estimation phase let us introduce arbitrarily an initial value and the parameters for the EM algorithm.

4.11 How to Learn the BN Directly from the Data

Hugin’s learning wizard allows the user to learn the BN directly from data. We now propose an example with a small database based on real financial data referred to S&P 500 implied volatility (Vix) and technical analysis area (Rsi, Roc, Macd_Diff, Cross_MA20_50). First, we click on the label “Wizards” and then select the option “Learning Wizard”, as shown in Figure 4.66.

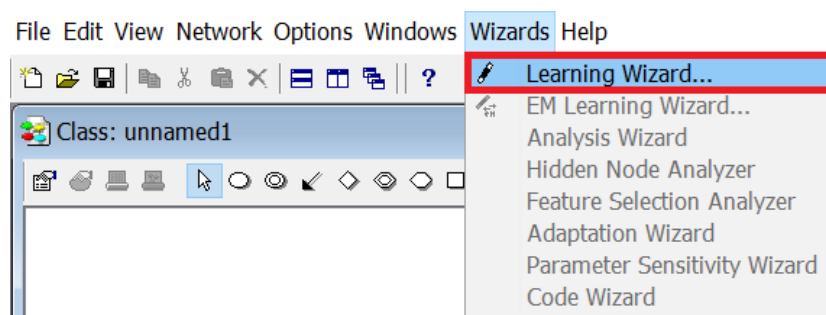


Figure 4.64: How to Open the Learning Wizard

Then a new window opens (Figure 4.65). Now we can select the file we want to upload by browsing it from its directory. We can also choose which type of encoding to use and the decimal separator symbol.

When the file has been successfully uploaded, we click on the “Next” button and the guided procedure lead us to “Node Summary” where we decide to treat our variables as numeric or continuous.

The following step leads us to define the structural constraints, see Figure 4.66.

According to our financial knowledge we impose a link connecting Vix, Roc and the Cross_MA20_50 nodes. Furthermore, we added two non-directed arrows connecting Roc with Rsi and Macd_Diff. This allows the software to understand the information flow by processing a learning algorithm. Then, we can notice an arrow with a red cross on it directed from Macd_Dif to Vix. This representation means that we are imposing the constraint that for no reason there should be a link with this direction. Anyway, our choice allows having a link from Vix to Macd_Diff.

In addition, this window allow us saving our progresses by clicking on the “Save” button (the floppy disk in the commands bar), while on its left we find the button that allows us uploading a previously saved file. In this way, we avoid repeating all the time the delicate procedure of imposing constraints. When the structure constraints have been set, we can click on the next button in order to decide which structural learning algorithm to adopt: NPC (Necessary Path Condition), PC (Path Condition), Greedy

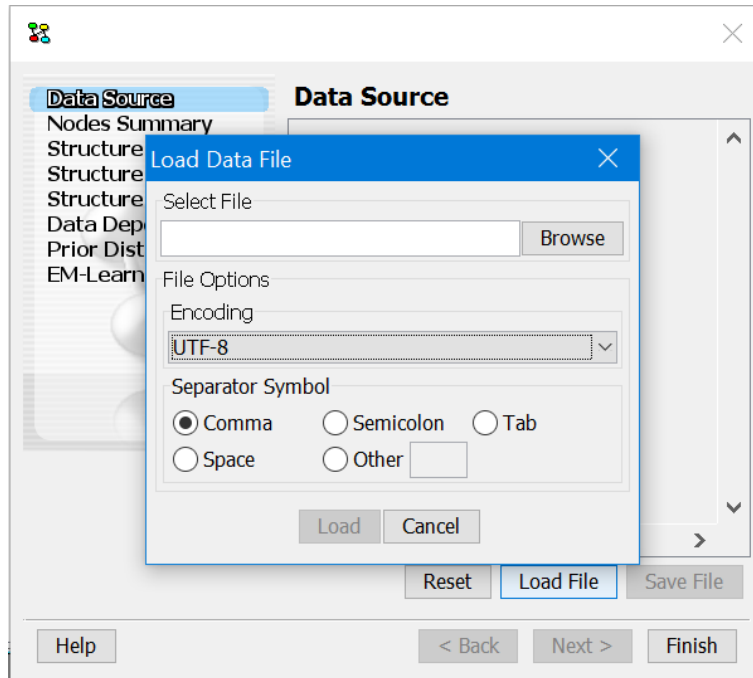


Figure 4.65: Loading Files with the Learning Wizard

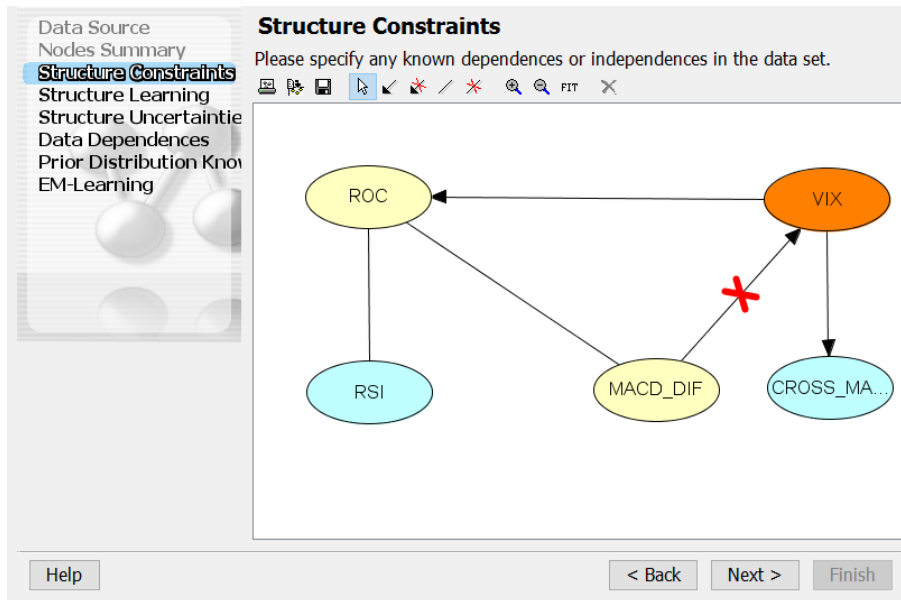


Figure 4.66: Imposing Structure Constraints

search-and-score, Chow-Liu tree, Rebane-Pearl polytree, Tree Augmented and Naïve Bayes (Figure 4.67).

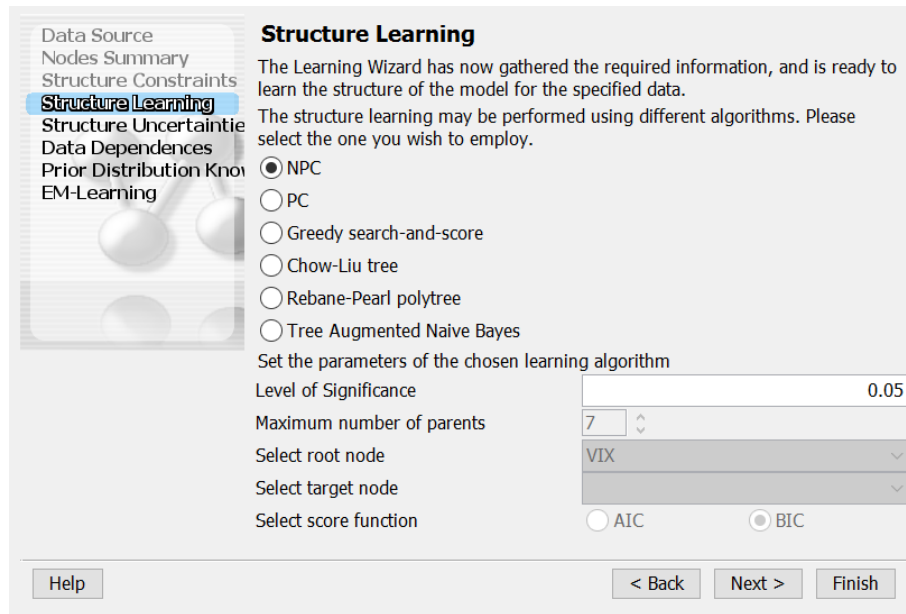


Figure 4.67: Structural Learning Algorithm Selection

We now suppose to chose the NPC algorithm. Then we click on the Next button and the Structure Uncertainties window opens, allowing us to solve eventual uncertain paths among the nodes of the BN. At the end of the procedure, Hugin provides a snapshot, Figure 4.68, of the BN structure. At this stage, the software allows us to define some prior knowledge on the distribution, but if we do not have any we can just click on the next button.

The procedure ends with the EM-Learning. In this final step, the software extracts directly from the data the conditional distributions. When the elaboration is concluded, we click on the Finish button and our BN appears in the Network Pane.

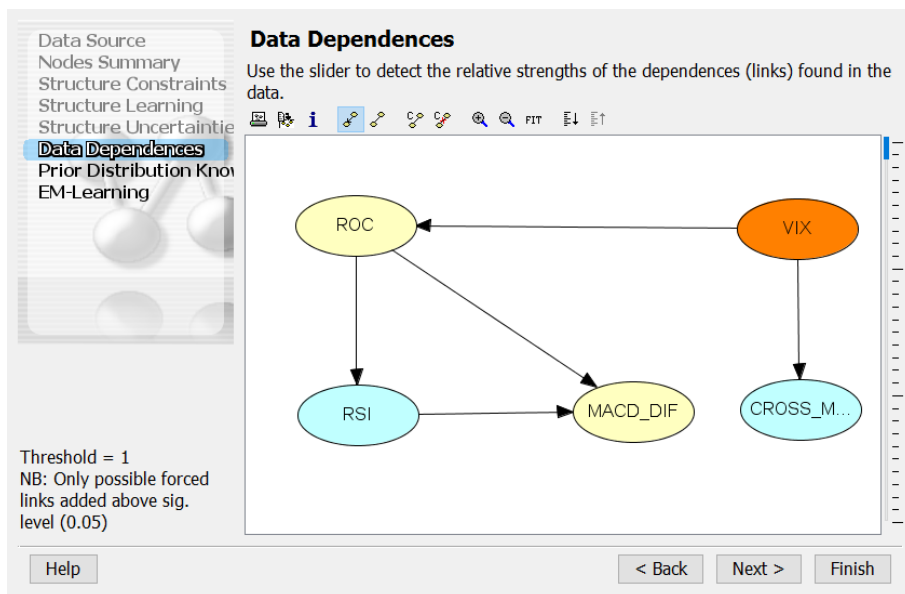


Figure 4.68: The Snapshot of the BN

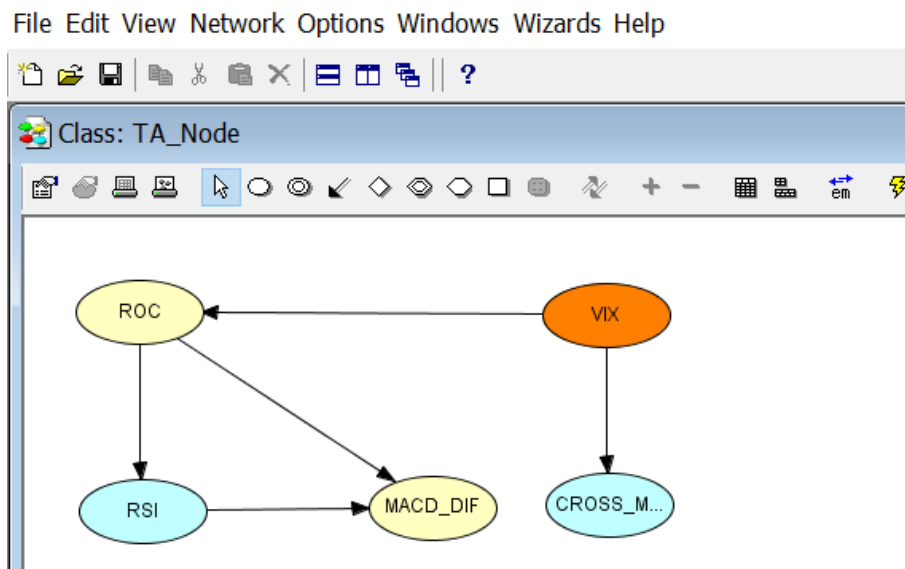


Figure 4.69: The Network Pane with our BN

Appendix 2: How to Build an OOBN with the Hugin Software

We now show in the following sections how to build an OOBN with Hugin. More in detail, we are going to show how to build it from the scratch and how to learn it directly from a small dataset.

4.12 Dynamic Bayesian Networks

The need of introducing the temporal dimension in a DAG derives from the fact that static BNs deal only with a single representation of the present. They do not consider any evolution through the time of the variables belonging to the network. On the other hand, Dynamic Bayesian Networks (DBNs hereafter) represent an intermediate step between static BNs and OOBNs. We now propose an example referred to a biotech company with 3 billion USD market capitalization that is subject to frequent capital increases because its financial position deteriorates very quickly. According to the entity of the operation, the stock price can react in different ways. We suppose that the Board of Directors chooses between two capital increase solutions:

- 0.5 billions USD capital increase. This operation would not solve definitely the financial issues. Future capital injections can be required.
- 3 billions USD capital increase, very dilutive because its amount is equal to the market cap of the company.

4.12.1 Building an OOBN Starting from a Static BN

First, we have to build a generic time slice containing the following nodes: C1 and C2, referred to the entity of the Capital Increase, and SD1 and SD2, representing the stock reactions to the earnings per share dilution. In this example, we ignore any specification about the CPTs. It is evident that the static BN shown in Figure 4.70 cannot provide indications on the evolution of the company financial conditions through the time. In this particular case, the introduction of the temporal dimension can provide us additional and more insightful information: this is why we extend our model to a DBN.

In order to do that, we connect multiple instances of the same static BN, where each instance represents a time slice. The simplest way to represent a DBN with Hugin is

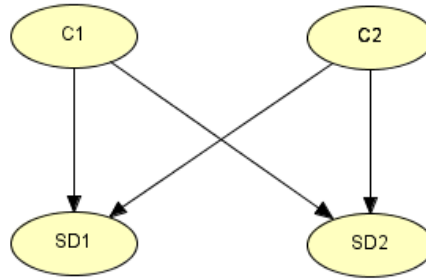


Figure 4.70: The Initial Static BN

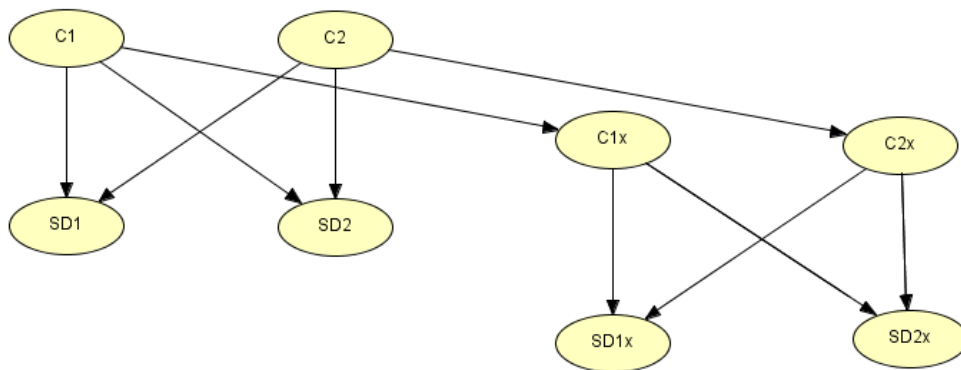


Figure 4.71: A DBN with Two Time Slices

to connect multiple instances as shown in Figure 4.71.

Thanks to this representation, we can measure the impact of two alternative capital increases on the probability of incurring in future in another capital injections. Therefore, there is a directed arrow between “Capital Increase” (nodes C1 and C2) and “Capital Increase Next Year” (nodes C1x and C2x). The nodes SD1 and SD2 represent the impact on the shares of the capital increase announcement while SD1x and SD2x represent the impact of the announcement of a capital increase in the next time slice. By exploiting the network shown in Figure 4.71 it is possible to predict the evolution of the company financial needs for two consecutive periods.

4.13 Building the OOBNs Elements

Before illustrating how to build an OOBN, we introduce its characteristic elements: the output nodes and the input nodes.

4.13.1 Output Nodes

The introduction of the output nodes represents a step beyond the static BNs because they allow communicating with the nodes outside of the instance node. Thanks to the output nodes, we can propagate the information from a time slice to another one. In our example, C1 and C2 have been chosen as parents of the nodes C1 and C2 belonging

to the following instance. In order to do that, we set the Capital Increase nodes as output nodes.

4.13.2 Input Nodes

In our example, the temporal aspect is introduced through the input nodes C1_prev and C2_prev. They are placeholders for C1 and C2 and for this reason, they should not be confused with real nodes because they are only parameters. As shown in Figure 4.72, an input node is connected by a directed link to a real one.

4.13.3 Creating an Interface Node

The first step is to declare the output nodes. In order to do that we right click on a variable, we select among the available properties “Set Input/Output” and then we choose “Set as Output”. On the other hand, right clicking on C1_prev and C2_prev and then selecting “Set as Input” in the “Set Input/Output” panel designates the input nodes. The input nodes borders are dashed in order to underline that they are not real. Finally, we connect C1_prev to C1 and C2_prev to C2.

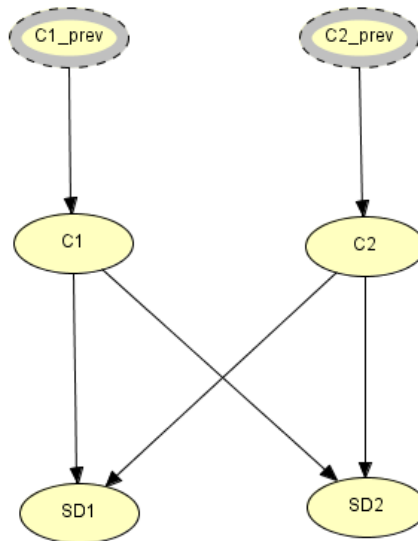


Figure 4.72: An Interface Node for the Biotech Company Example

Whenever a model contains a high number of variables, we need a representation that makes easier the visual inspection. In addition, BNs show some limit when we have in the same network several recurring fragments. In order to solve these issues, we introduced OOBNs that allow overcoming efficiently these drawbacks.

4.13.4 OOBN for the Biotech Company Example

In Figure 4.73 we represent two time slices: one is referred to Year 0 and the other one to the following year. The operation of replicating multiple times the same network structure can result tedious and time consuming. For this reason, we illustrate how to build an OOBN with Hugin. First, we create a single time slice, as shown in Figure

4.74. The nodes C1_prev and C2_prev have been set as input nodes, while C1 and C2 as output nodes. Then we save this network in *.oobn format. A network is “object oriented” when it includes instance nodes. In order to build them, we open a new file and we click on Hugin’s Window Toolbar on the Instance Node Tool, see Figure 4.73.



Figure 4.73: The Instance Node Tool

Then Hugin shows the list of possible instances that can be uploaded and used. We select the one saved before and referred to the capital increase. Then we click on the network pane: a white rectangular node has been generated and it appears in the Network Pane, see Figure 4.74.

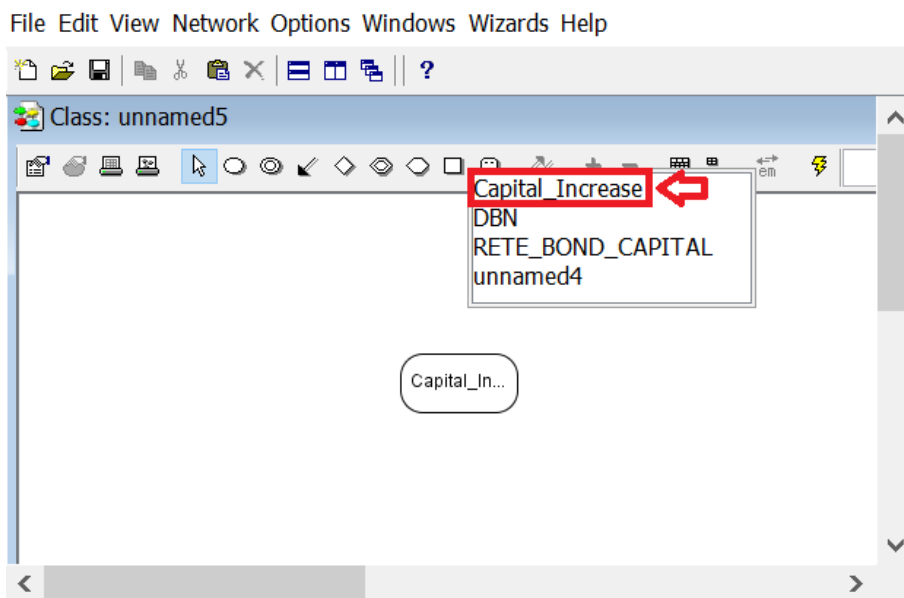


Figure 4.74: Selecting the Instance Node Through the Toolbar

By clicking on this node, we can expand it and observe the content of the instance. By following this step, we can replicate the time slice many times we need. In this toy example, we cover an interval of 3 years. Then we connect C1 to C1_prev through the years and when we conclude our procedure the OOBN looks like the one shown in Figure 4.75. The nodes previously declared as output or input nodes (interface nodes) are visible in each instance. At the top, we have the inputs while on the bottom of each “block” there are the output ones.

Then we have to connect the time slices: Capital_Increase_1 with Capital_Increase_2, and Capital_Increase_2 with Capital_Increase_3. We perform this task by connecting C1 and C2 belonging to Capital_Increase_1 respectively to C1_prev and C2_prev contained in the instance node Capital_Increase_2, and C1 and C2 belonging to Capital_Increase_2 with C1_prev and C2_prev included in the instance node Capital_Increase_3. Thanks to this operation, we pass an entire set of probabilities from an input node to an output one. We show the OOBN with the instance nodes collapsed in Figure 4.76.

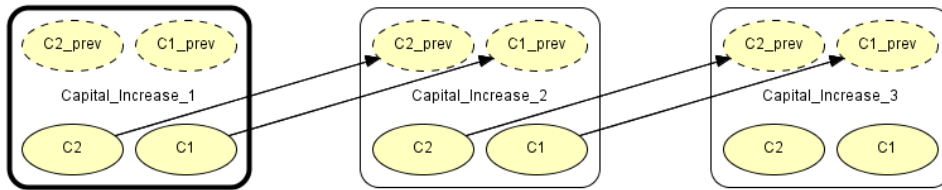


Figure 4.75: Three Connected Time Slices



Figure 4.76: The Collapsed Instance Nodes

4.14 Running the OOBN

We compile the network by clicking on the Run Mode button. Then, the Hugin window is split in two parts as shown in Figure 4.77. On the right side, we have the Network Pane; while on the left side, we have the Node List Pane.

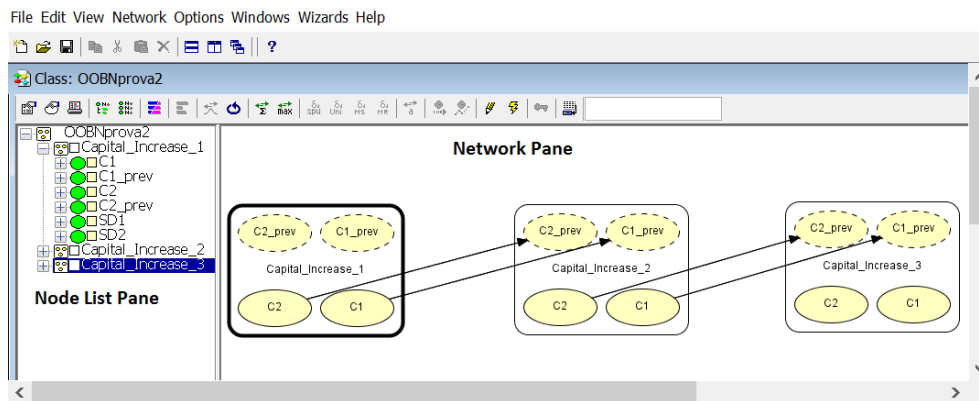


Figure 4.77: The Results of Our Toy Example

We observe that unlike the basic nodes, the instance ones do not have belief monitors associated. However, we can explore the belief monitors referred to the nodes belonging to each instance node by expanding them in the Node List Pane. If an instance contains many nodes it can be hard to find the one we are looking for. Hugin interface simplify this research. We can right click on an instance node in the Node List Pane, select “Traverse Instance” and then a new window containing the complete class referred to that instance opens.

4.15 Learning the OOBN from a Dataset

The procedure of learning the OOBN directly from a dataset shares many steps with the one outlined in the previous paragraphs.

4.15.1 The Financial Markets Example

In our toy example, the dataset is represented by some market variables:

- Sentiment Variables: Volatility (VOLA) and Put/Call Volume Ratio (PC_RATIO)
- Growth Variables: Sales Growth (SALES_GR, EBITDA_GR and EARN_GR)
- Target Variable: B_S_SPX

Each variable is discretized in three different states: 0, 1, 2. The nodes belonging to Sentiment and Growth blocks belongs to the same instance node. Each fragment of the OOBN has been learned separately by adopting a two steps procedure. First, we identify the skeleton of each network by running the Chow Liu Algorithm (Chow and Liu, 1968), and then we connect the nodes with undirected edges by following the suggested structure. We also introduce some constraints deriving from our prior knowledge of the financial markets. Finally, we run the NPC algorithm and we obtain a BN referred to each single area, as the ones shown in Figure 4.78.

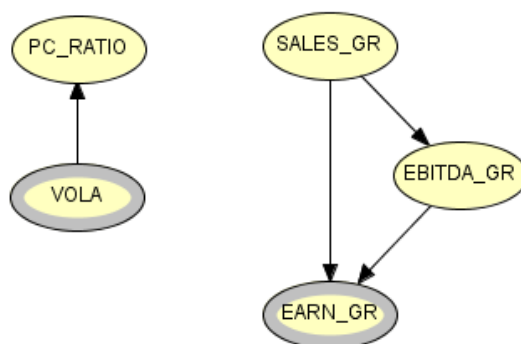


Figure 4.78: The Financial Markets Instance Nodes

Then we set the VOLA node as the output node for the Sentiment Variables instance node, while the output node for the Growth Instance node is the EARN_GR. In this specific example, we do not have input nodes. Now that each instance node has been introduced, we can build our OOBN with Hugin. First, we have to build the CPT for the output nodes (Vola and Earn.Gr). By exploiting OOBN properties, connecting the output nodes to the target variable is enough to obtain the CPT associated to B_S_SPX, see Figure 4.79.

The next step is to right click on B_S_SPX and select the command “Open Table”, that shows the CPT referred to the target variable. We now open Microsoft Excel and we save an empty *.csv file in a random directory. This passage is necessary in order to export the probability table. We now go back to Hugin and we click on the Functions label, then we select Export table and we save the data by overwriting the file on *.csv created in the previous step. Now that we have obtained the CTP associated to

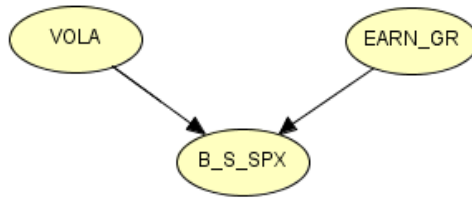


Figure 4.79: The BN Used for Learning the CPT

B_S_SPX we can build our OOB. First, we upload each instance node and leave it open in the background. This procedure allows the software to gather the information about the instance nodes. Then we upload in the network pane the B_S_SPX variable by using Hugin's learning wizard. The final step consists in clicking on the instance node button, see Figure 4.80. By clicking on these instance nodes we are including them in the network.

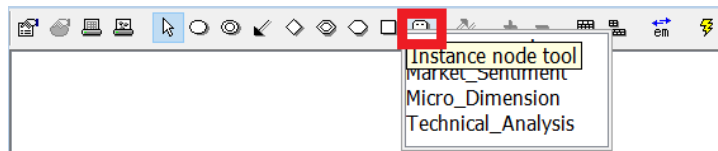


Figure 4.80: Uploading the Instance Nodes

We now connect the output variables to B_S_SPX. When this task is completed, we right click on the response variable and when its CPT opens we select the Function label and then we select "Import Table". We now import the probability table previously saved on the *.csv file, as shown in Figure 4.81.

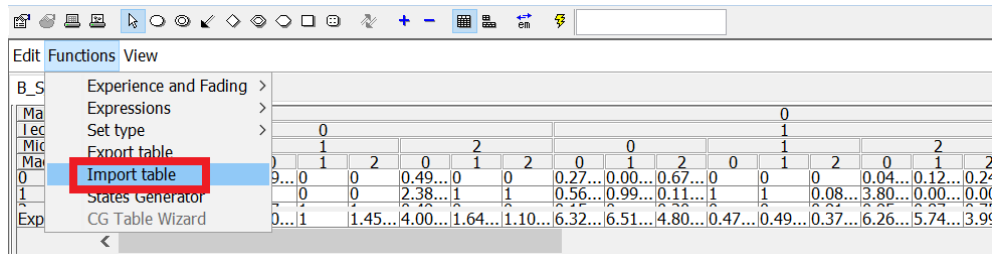


Figure 4.81: Importing the CPT Table Referred to B_S_SPX

Now we can run the model by clicking on the Run button.

B_S_SPX	0	1	2	0	1	2	0	1	2	0	1	2
Senti19...	0	0	0	0	0	0	0	0	0	0	0	0
Growth...	0.366489	0.298764	0.651488	0.621384	0.279996	0.404892	0.404892	0.279996	0.404892	0.404892	0.19346	0.490162
0	0.224777	0.253937	0.253937	0.253937	0.373978	0.271274	0.271274	0.373978	0.271274	0.271274	0.406948	0.311264
1	0.408734	0.318898	0.318898	0.171451	0.346026	0.323834	0.323834	0.346026	0.323834	0.323834	0.399691	0.198573
2	0.408734	0.318898	0.318898	0.171451	0.346026	0.323834	0.323834	0.346026	0.323834	0.323834	0.399691	0.198573
Experie...	508	508	508	508	508	508	508	508	508	508	508	508

Figure 4.82: The CPT referred to B_S.SPX | (VOLA, EARN_GR)

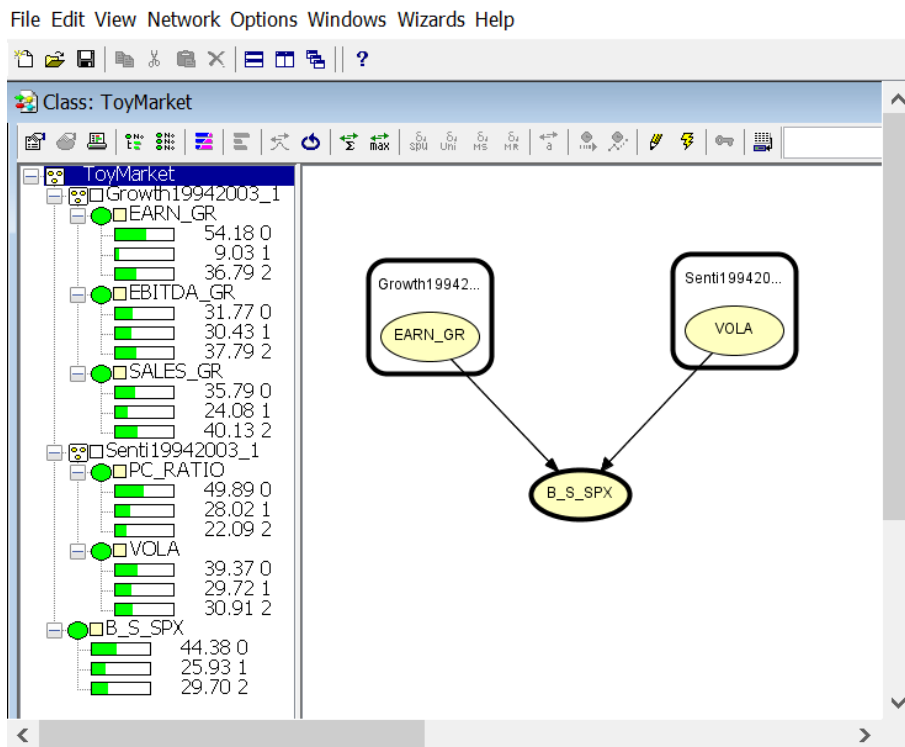


Figure 4.83: The OOBN Leaned from Financial Data

Bibliography

- [1] Andersen, D. 2006. *Investing Using HUGIN Software - An Easy Way to Use Quantitative Investment Techniques*. Working Paper.
- [2] Bangso, O., & Wuillemin, P. H. (2000, May). *Top-Down Construction and Repetitive Structures Representation in Bayesian Networks*. In FLAIRS Conference (pp. 282-286).
- [3] Brooks, R. A. (1991). *Intelligence without representation*. Artificial intelligence, 47(1), 139-159.
- [4] Chow, C., & Liu, C. (1968). *Approximating discrete probability distributions with dependence trees*. IEEE transactions on Information Theory, 14(3), 462-467.
- [5] Cowell, R. G., & Dawid, A. P. (1992). *Fast retraction of evidence in a probabilistic expert system*. Statistics and Computing, 2(1), 37-40.
- [6] Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. ISO 690
- [7] Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*. Cambridge University Press.
- [8] Dawid, A. P. (1979). *Conditional independence in statistical theory*. Journal of the Royal Statistical Society. Series B (Methodological), 1-31.
- [9] Dawid, A. P. (1980). *Conditional independence for statistical operations*. The Annals of Statistics, 598-617.
- [10] Demirer, R., Mau, R., & Shenoy C. 2006 *Bayesian Networks: A Decision Tool to Improve Portfolio Risk Analysis*. Journal of Applied Finance (16:2), pp 106-133.
- [11] Geiger, D., Verma, T., & Pearl, J. (1990). *Identifying independence in Bayesian networks*. Networks, 20(5), 507-534.
- [12] Gemela, J. (2001). *Financial analysis using Bayesian networks*. Applied Stochastic Models in Business and Industry, 17(1), 57-67.
- [13] Goldberg, A., & Robson, D. (1983). *Smalltalk-80: the language and its implementation*. Addison-Wesley Longman Publishing Co., Inc..
- [14] Gorry, G. A., & Barnett, G. O. (1968). *Experience with a model of sequential diagnosis*. Computers and Biomedical Research, 1(5), 490-507.

- [15] Jensen, F. V. (1996). *An introduction to Bayesian networks* (Vol. 210). London: UCL press.
- [16] Jensen, F. V. (1996). *Bayesian networks basics*. AISB quarterly, 9-22. Chicago
- [17] Jensen, F.V. (2001). *Bayesian Networks and Decision Graphs*. Series for Statistics for Engineering and Information Science.
- [18] Jensen, F.V. and Nielsen, T.D. (2007). *Bayesian Nets and Decision Graphs*. 2nd edn, Springer Verlag, Berlin.
- [19] Johnson, D., & Sinanovic, S. (2001). *Symmetrizing the kullback-leibler distance*. IEEE Transactions on Information Theory.
- [20] Koller, D., & Pfeffer, A. (1997, August). *Object-oriented Bayesian networks*. In Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence (pp. 302-313). Morgan Kaufmann Publishers Inc..
- [21] Kjærulff, U. B., & Madsen, A. L. (2013). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. New York, NY: Springer New York, 17(382), 186.
- [22] Koski, T. and Noble, J.M. 2009. *Bayesian Networks - An introduction*. Wiley Series in Probability and Statistics.
- [23] Lauritzen, S. L. (1995). *The EM algorithm for graphical association models with missing data*. Computational Statistics & Data Analysis, 19(2), 191-201.
- [24] Lauritzen, S. L. (1996). *Graphical models* (Vol. 17). Clarendon Press.
- [25] Lauritzen, S. L., & Sheehan, N. A. (2003). *Graphical models for genetic analyses*. Statistical Science, 489-514.
- [26] Lauritzen, S. L., & Spiegelhalter, D. J. (1988). *Local computations with probabilities on graphical structures and their application to expert systems*. Journal of the Royal Statistical Society. Series B (Methodological), 157-224.
- [27] Lauritzen, S. L., Dawid, A. P., Larsen, B. N., & Leimer, H. G. (1990). *Independence properties of directed Markov fields*. Networks, 20(5), 491-505.
- [28] Madsen, A. L., Lang, M., Kjærulff, U. B., & Jensen, F. (2003, July). *The Hugin tool for learning Bayesian networks*. In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (pp. 594-605). Springer Berlin Heidelberg.
- [29] Mahoney, S. M., & Laskey, K. B. (1996, August). *Network engineering for complex belief networks*. In Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence (pp. 389-396). Morgan Kaufmann Publishers Inc..
- [30] Markov, A. A. (1906). *Extension of the law of large numbers to dependent quantities* (in Russian), Izvestiia Fiz.-Matem. Obsch. Kazan Univ., (2nd Ser.), 15, pp. 135-156
- [31] McCann, R. K., Marcot, B. G., & Ellis, R. (2006). *Bayesian belief networks: applications in ecology and natural resource management*. Canadian Journal of Forest Research, 36(12), 3053-3062.

- [32] Nielsen, A. E. (2011). *Goal - Global Strategy Paper No. 1*, Goldman Sachs Global Economics - Commodities and Strategy Research.
- [33] Nielsen, T. D., & Jensen, F. V. (2009). *Bayesian networks and decision graphs*. Springer Science & Business Media.
- [34] Neapolitan, R. E. (1990). *Probabilistic reasoning in expert systems: theory and applications*. New York: Wiley.
- [35] Neil, M. & Fenton, N., (2012). *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press.
- [36] Patel, P.N., Yao, S., Carlson, R., Banerji, A., Handelman, J. (2011) *Quantitative Research - A Disciplined Approach*, Credit Suisse Equity Research.
- [37] Pearl, J. (1986). *On evidential reasoning in a hierarchy of hypotheses*. Artificial Intelligence, 28(1), 9-15.
- [38] Pearl, J. (1986). *Fusion, propagation, and structuring in belief networks*. Artificial intelligence, 29(3), 241-288.
- [39] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible reasoning*.
- [40] Pearl, J. (2000). *Causal inference without counterfactuals: Comment*. Journal of the American Statistical Association, 95(450), 428-431.
- [41] Pfeffer, A., Koller, D., Milch, B., & Takusagawa, K. T. (1999, July). *SPOOK: A system for probabilistic object-oriented knowledge representation*. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 541-550). Morgan Kaufmann Publishers Inc..
- [42] Rebonato, R., & Denev, A. (2014). *Portfolio Management Under Stress: A Bayesian-net Approach to Coherent Asset Allocation*. Cambridge University Press.
- [43] Schmitt, G. (1998). *Design and construction as computer-augmented intelligence processes*. CAADRIA'98 Third International on CAAD Education, Japan.
- [44] Smith, J. Q. (1989). *Influence diagrams for statistical modeling*. The Annals of Statistics, 654-672.
- [45] Spiegelhalter, D. J., & Lauritzen, S. L. (1990). *Sequential updating of conditional probabilities on directed graphical structures*. Networks, 20(5), 579-605.
- [46] Spirtes, P., & Glymour, C. (1991). *An algorithm for fast recovery of sparse causal graphs*. Social science computer review, 9(1), 62-72.
- [47] Spirtes, P., Glymour, C. N., & Scheines, R. (2000). *Causation, prediction, and search*. MIT press.
- [48] Shenoy, C., and Shenoy, P.P. 1998. *Bayesian networks: a decision tool to improve portfolio risk analysis*. Working paper, School of Business, University of Kansas.
- [49] Shenoy, C., & Shenoy, P. P. (2000). *Bayesian network models of portfolio risk and return*.

- [50] Srinivas, N., & Deb, K. (1994). *Muiltiobjective optimization using nondominated sorting in genetic algorithms*. Evolutionary computation, 2(3), 221-248.
- [51] Steck, H., & Tresp, V. (1999, September). *Bayesian belief networks for data mining*. In Proceedings of the 2. Workshop on Data Mining und Data Warehousing als Grundlage moderner entscheidungsunterstutzender Systeme (pp. 145-154).
- [52] Taroni, F., Aitken, C., Garbolino, P., & Biedermann, A. (2006). Book Tools. Chicago
- [53] Uusitalo, L. (2007). *Advantages and challenges of Bayesian networks in environmental modelling*. Ecological modelling, 203(3), 312-318.
- [54] Wermuth, N. & Lauritzen, S.L. (1983). *Graphical and recursive models for contingency tables*, Biometrika, 70, 537-552.
- [55] Wermuth, N., & Lauritzen, S. L. (1990). *On substantive research hypotheses, conditional independence graphs and graphical chain models*. Journal of the Royal Statistical Society. Series B (Methodological), 21-50.
- [56] Whittaker, J. (1990). *Graphical models in applied multivariate analysis*. Chichester New York et al: John Wiley & Sons.
- [57] Xiang, Y., Poole, D., & Beddoes, M. P. (1993). *Multiply sectioned Bayesian networks and junction forests for large knowledge based systems*. Computational Intelligence, 9(2), 171-220.