

Improving the Quality of Indicator Systems by MoSi – Methodology and Evaluation

Inaugural-Dissertation
zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaft
des Fachbereichs Wirtschaftswissenschaft der
Freien Universität Berlin

vorgelegt von

Dipl.-Volkswirt Veit Köppen

aus

Berlin

September 2008

Dekan: Univ.-Prof. Dr. Jörg Sydow

Erstgutachter: Univ.-Prof. Dr. Hans-J. Lenz

Zweitgutachter: Univ.-Prof. Dr. Hans-Knud Arndt

Tag der Disputation: 18.07.2008

Abstract

Long term business success highly depends to how fast a company reacts on changes in the market situation. Those who want to be successful need relevant, up-to-date, and accurate information. Business or economic decisions rely on indicators. One facet of data quality is the integrity of data. Most of the main business and economic indicators suffer from statistical discrepancies. These indicators are based on non-linear equation systems and are normally not crisp, but random due to measurement errors. Consequently, computation of the corresponding probability distributions is usually not trivial.

Handling uncertainty within indicator systems is a major challenge for improved decision making. Different approaches exist for dealing with uncertainty, e.g., Fuzzy set theory and the probabilistic method. The shortcomings of both approaches can be reduced by the use of simulation.

As the Gaussian distribution is not closed under all four arithmetic operations, there is the need for Markov Chain Monte Carlo (MCMC) simulation to determine the probability distributions. A combination of data, generated by MCMC simulation, which is based on prior knowledge about a fully specified non-linear, stochastic balance equation system with noisy measurements, is proposed for handling uncertainty within indicator systems. The Metropolis Hastings algorithm enables the use of any computable target probability function. SamPro is the algorithm that implements the MCMC simulation approach for indicator systems.

The estimation of unobservable quantities of such models is improved by SamPro and data inconsistencies to the equation system are revealed. MoSi is proposed as a software tool for the modelling of indicator systems as well as their simulation. The implementation of the SamPro algorithm is consequently included in MoSi, as well. MoSi can be used efficiently in the processes of planning, decision making, and controlling.

Contents

List of Figures	v
List of Tables	ix
List of Abbreviations	xi
Nomenclature	xiii
1 Introduction	1
1.1 Problem Statement	2
1.2 Contribution	2
1.3 Thesis Overview	3
2 Indicator Systems	5
2.1 Modelling	7
2.1.1 System and Models	7
2.1.2 Visualisation	11
2.2 Classical Business Indicator Systems	14
2.2.1 DuPont System of Financial Control	14
2.2.2 ZVEI - indicator system	17
2.2.3 RL business indicator system	19
2.2.4 Further Classical Business Indicator Systems	21
2.3 Balanced Scorecard	24
2.3.1 Background	24
2.4 Economic Indicator Systems	28
2.4.1 Measures of National Income and Output	28
2.4.2 United Nations System of National Accounts	31
2.4.3 Petroleum Economic Indicators	32
2.5 Challenges for Indicator Systems	34
3 Methodology	35
3.1 The Possibilistic Approach	36
3.1.1 Fuzzy sets	36
3.1.2 Operations on Fuzzy Sets	38

3.1.3	FuzzyCalc	41
3.2	The Probabilistic Approach	44
3.2.1	Single-PeakDistributions	44
3.2.2	Symmetrical Distributions	46
3.2.3	Heavy Tail Distributions	47
3.2.4	Skewed Distributions	48
3.2.5	Contaminated Distributions	51
3.2.6	Models with Errors in the Variables	53
3.2.7	Quantor	54
3.3	Simulation	57
3.3.1	Monte Carlo Simulation	57
3.3.2	Particle Filters	58
3.3.3	Markov Chain Monte Carlo Simulation	66
3.3.4	Metropolis Algorithm	69
3.3.5	Metropolis-Hastings Algorithm	73
3.3.6	Gibbs-Sampler	76
3.3.7	Missing Values	81
3.3.8	M-Consistency Check	83
3.4	Robustifying the Metropolis Hastings Algorithm	90
3.4.1	Noise	90
3.4.2	Transition Kernels	90
3.4.3	Burn-In Problem	98
3.4.4	Number of Particles	101
3.4.5	Achieving Equation Fulfilment using SamPro	102
3.4.6	Outliers	104
4	Comparative Study	113
4.1	Phases	115
4.1.1	Simulation Planning	115
4.1.2	Simulation Experiments	118
4.1.3	Simulation Analysis	118
4.2	Comparing FuzzyCalc, Quantor and SamPro	122
4.3	Effects of Nonlinearity, Skewness, and Correlation	129
4.3.1	Experimental Group A	132
4.3.2	Experimental Group B	138
4.3.3	Experimental Group C	142
4.3.4	Results	149
4.4	Balanced Scorecard Example	150
4.4.1	Model	150
4.4.2	Input Data	150
4.4.3	Simulation and Results	150
4.4.4	Results of BSC example	153
4.5	Tool Evaluation	155
4.5.1	FuzzyCalc	155

4.5.2	Quantor	156
4.5.3	MoSi	157
5	Software Design and Implementation	159
5.1	Design	160
5.2	Implementation	165
5.3	Software Demonstration	170
6	Conclusion	175
6.1	Overview	175
6.2	Comparison of Methods	176
6.3	Contribution	177
6.4	Future Work	179
	Bibliography	181
	Index	193
	Appendices	195

List of Figures

2.1	Hierarchy of Indicators	9
2.2	Equation Graph (Parser Tree) for Equation $A = (B + C) * D - E/F$	12
2.3	Another Equation Graph for Equation $A = (B + C) * D - E/F$. .	13
2.4	The Original DuPont Measurement System	15
2.5	The Extended DuPont Measurement System	15
2.6	The Simplified DuPont Measurement System	16
2.7	The Equation Graph of the simplified DuPont Measurement System	16
2.8	The ZVEI Business Indicator System	18
2.9	Structural Analysis of the ZVEI Business Indicator System	19
2.10	The Simplified RL Business Indicator System	20
2.11	Ratio au Tableau de Bord	21
2.12	Managerial Control Concept	22
2.13	Pyramid Structure of Ratios	22
2.14	BSC Strategy Approach	24
2.15	Translating a Strategy into Operation	26
2.16	Model Graph of “Measures of National Income and Output” . . .	29
2.17	EcoOil Model	33
3.1	A Fuzzy Set A with Triangular Membership Function and a crisp set B	37
3.2	An α -cut Fuzzy Set	37
3.3	A Fuzzy Set A with Complement A^C	39
3.4	Union of Fuzzy Set A and B	39
3.5	Union and Intersection of Fuzzy Set A and Complement A^C	39
3.6	Union of Fuzzy Set A and B	39
3.7	Gaussian Distribution	44
3.8	Bivariate Gaussian Distribution	45
3.9	Triangular Distribution	45
3.10	Uniform Distribution	46
3.11	Laplace Distribution	46
3.12	Student’s T Distribution	47
3.13	Pareto Distribution	48
3.14	Cauchy Distribution	48

3.15 Exponential Distribution	49
3.16 Gamma Distribution	49
3.17 Beta Distribution	50
3.18 Inverse Gaussian Distribution	50
3.19 Multivariate Skew Normal Distribution	51
3.20 Contaminated Bivariate Gaussian Distribution	52
3.21 Double Gamma Distribution	52
3.22 Double Weibull Distribution	53
3.23 Model Graph of GLS Model	53
3.24 DuPont Model in QR	54
3.25 Profit within QR	55
3.26 Graph of DuPont Model in QR	56
3.27 Particles and Filtering Function	58
3.28 Generating Function q and Proposal Function π	59
3.29 Weights in Sequential Importance Sampling at $t = 0$ and $t = 1$. .	64
3.30 Idea Behind the Particle Filter Bootstrap	65
3.31 Bootstrapping AR-1 Process Starting with Uniform Distribution .	66
3.32 Bootstrapping AR-1 Process Starting with Gaussian Distribution .	67
3.33 Target (π) and Enveloping ($M \cdot q$) Function in Rejection Sampling	70
3.34 Histograms of Metropolis Iterations at $t = 10, 100, 1000$	72
3.35 Histograms of Metropolis Iterations at $t = 10, 100, 1000$ with Non-Symmetrical Proposal Function	73
3.36 Histograms of Metropolis-Hastings Iterations at $t = 10, 100, 1000$ with Non-Symmetrical Proposal Function	76
3.37 Time Series Plot of Coal-Mining Disasters in Great Britain	78
3.38 True and Approximated MPD for λ_1 and λ_2	80
3.39 True and Approximated MPD for m	80
3.40 Acceptance Rates Corresponding to Tuning Parameter τ	81
3.41 M-Inconsistent and M-Consistent Densities	83
3.42 Joint Distribution of two Estimates for the Same Variable	86
3.43 Corrected Estimator	86
3.44 A Function, that has no L_2 -Norm	87
3.45 Projection for a Common Sampling Distribution	88
3.46 A Bivariate Gaussian Mixture	92
3.47 Random Walk for Bivariate Mixture with Initial Parameters and Chain Length Effects	93
3.48 Different Proposal Functions (Deviation of $\sigma_{proposal}$ from σ_{target}) .	94
3.49 Different Shapes of Proposal Functions for Gaussian Target	95
3.50 Different Shapes of Proposal Functions for Cauchy Target	95
3.51 Independent MH for Bivariate Mixture with Initial Parameters and Chain Length Effects	96
3.52 Time Series Plots of Random Walk MH	99
3.53 Independence Chain with Different Number of Particles	102
3.54 Uniform Distribution with Different Numbers of Samples	103

3.55	Trajectory of Profit using Prior Information and Equation System	104
3.56	Estimate of Mean by SamPro	105
3.57	Estimated Distributions for P, CT, PM and ROI (no trimming) . .	106
3.58	Estimated Distributions for P, CT, PM and ROI with Marginal Ordering and Trimming ($\gamma = 5\%$)	108
3.59	Estimated distributions for P, CT, PM and ROI with Reduced Or- dering and Trimming ($\gamma = 5\%$)	109
3.60	Mean and Variance as a Function of γ for P, CT, PM and ROI with Reduced Ordering	110
3.61	Convex Hull for Some Points of CT and P	111
4.1	Gaussian Distribution (μ and σ^2 given)	119
4.2	Statistical Estimators for Cauchy Distribution	120
4.3	Histogram of 10000 Samples from a Gaussian Distribution	121
4.4	Exact and Simulated Estimates for Profit	130
4.5	Estimates for Profit	131
4.6	Normal Distributed Variables with no Correlation	133
4.7	Effect of Skewed Distributions of all (Observed) Variables, no Cor- relation	134
4.8	Normality and Negative Correlation ($\rho = -0.7$)	135
4.9	Normality and Positive Correlation ($\rho = 0.7$)	136
4.10	Skewness and Negative Correlation	137
4.11	Skewness and Negative Correlation	137
4.12	Scatter Plot of Simulated Profit Values for $\rho = -0.4$	139
4.13	Scatter Plot of Simulated Profit Values for $\rho = 0$	139
4.14	Scatter Plot of Simulated Profit Values for $\rho = 0.4$	139
4.15	Scatter Plot of Simulated Profit Values for $\rho = -0.4$ (Weak M- Consistent)	141
4.16	Scatter Plot of Simulated Profit Values for $\rho = 0$ (Weak M-Consistent)	141
4.17	Scatter Plot of Simulated Profit Values for $\rho = 0.4$ (M-Inconsistent)	141
4.18	Prior Distributions $\rho = -0.4$ (Weak M-Consistent)	143
4.19	Prior Distributions $\rho = 0$ (Weak M-Consistent)	143
4.20	Prior Distributions $\rho = 0.4$ (Weak M-Consistent)	143
4.21	Result Distributions $\rho = -0.4$ (Weak M-Consistent)	145
4.22	Result Distributions $\rho = 0$ (Weak M-Consistent)	145
4.23	Result Distributions $\rho = 0.4$ (Weak M-Consistent)	145
4.24	Scatter Plots of Profit for $\rho = 0.4$	147
4.25	3D Scatter Plot of Simulated Profit Values for $\rho = -0.4$ (Weak M-Consistent)	148
4.26	3D Scatter Plot of Simulated Profit Values for $\rho = 0$ (Weak M- Consistent)	148
4.27	3D Scatter Plot of Simulated Profit Values for $\rho = 0.4$ (M-Inconsistent)	148
4.28	Balanced Scorecard Model Graph	151
4.29	Simulation Results of BSC Characteristic Indicators (Kiviat Diagram)	153

5.1	Architecture of MoSi	160
5.2	Class Diagram for MoSi-Modelling	161
5.3	Activity Diagram for MoSi-Simulation	162
5.4	Class Diagram for MoSi-Simulation	164
5.5	Package Diagram for MoSi	166
5.6	MoSi Project Files	167
5.7	Screenshot of DuPont Model within MoSi	170
5.8	Screenshot of Table View of the DuPont Model	171
5.9	Variable Property Dialog	172
5.10	Simulation Start Dialog	172
5.11	Simulation Results (a)	173
5.12	Simulation Results (b)	173

List of Tables

3.1	Coal-Mining Disaster data in Great Britain by years 1851-1962 . . .	79
3.2	Geweke Z-Scores on Random Walk Scenarios	100
3.3	Raftery-Lewis Test Results of Random Walk Scenarios	101
4.1	Prior Information of 15 DuPont Test Scenarios	122
4.2	FuzzyCalc Results of 15 DuPont Test Scenarios	124
4.3	Quantor Results of 15 DuPont Test Scenarios	125
4.4	SamPro first Iteration Results of 15 DuPont Test Scenarios	127
4.5	Means and Standard Deviations of all observed and simulated variables, Gaussian distributions, no correlation, no missing values, M-consistent observations.	139
4.6	Simulation Results after 5 SamPro iterations with M-consistency ($\rho = 0$)	140
4.7	Estimates for $\rho = 0.4$	144
4.8	Prior Information of the BSC Characteristic Indicators	152
4.9	Evaluation of Tools for Indicator Systems with Errors in the Variables	155
4.10	Evaluation symbols	155

List of Acronyms

BSC	Balanced Scorecard
CT	Capital Turnover
EVA	Economic Value Added
GDP	Gross Domestic Product
GLS	General Least Squares
GNI	Gross National Income
GNP	Gross National Product
GUI	Graphical User Interface
JVM	Java Virtual Machine
LHS	Left Hand Side
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
MoSi	Modelling and Simulation
MPD	marginal posterior distribution
MSN	Multivariate Skew Normal
NI	National Income
NNP	Net National Product
pdf	probability distribution function
PM	Profit Margin
PRNG	Pseudorandom Number Generator
RHS	Right Hand Side
ROCE	Return on Capital Employed
ROE	Return on Equity

ROI	Return on Investment
SamPro	Sampling and Projection
sd	standard deviation
SIS	Sequential Importance Sampling
SNA	United Nations System of National Accounts
UML	Unified Modeling Language
XML	Extensible Markup Language
ZVEI	German Electrical and Electronic Manufacturer's Association

Nomenclature

a, b, c, d	Real Values
n, m	Integer Values
A, B, G, X, Y, Z	Sets
A^C	Complement Set of A
H, P, R, Q	Matrices
I_d	d Dimensional Identity Matrix
t	Iteration Index
T	Total Number of Iterations
$x^{(n)}$	Sample of size n
f_{xz}	Common Density Function of x and z
$\Gamma()$	Gamma Function
γ	Fraction of Eliminated Samples (γ -Trimming)
$d_M()$	Mahalanobis Distance
\mathbb{R}	Set of Real Numbers
$\mathbb{R}_{\geq 0}$	Set of Non-Negative Real Numbers
$\mu_A()$	Membership Function of Fuzzy Set A
A_α	α -Cut of Fuzzy Set A
Ψ	Kruse-Meyer Similarity Measure
MoA	Measure of Area
$N_d()$	d Dimensional Gaussian Distribution
$U(a, b)$	Uniform Distribution in $(a, b]$

$MSN(\cdot)$	Multivariate Skew Normal Distribution
u	Random Number from $U(a, b)$
μ	Mean
$\hat{\mu}$	Estimated Mean
μ_γ	γ -Trimming Mean Estimator
$\check{\mu}$	Mean Estimator for d_M
σ	Standard Deviation
$\hat{\sigma}$	Estimated Standard Deviation
σ_γ	γ -Trimming Standard Deviation Estimator
Σ	Covariance Matrix
$\check{\Sigma}$	Covariance Estimator for d_M
ρ	Correlation Coefficient
$\tilde{\mu}$	Median
$\hat{\tilde{\mu}}$	Estimated Median
S	State Space of a Markov Chain
s_i	State in a Markov Chain
$P(x_t)$	Markov Chain Distribution Estimate in t
$P_M(x_t, x_{t-1})$	Transition Kernel of Markov Chain
p_{ij}	Probability to move from s_i to s_j
T_{MH}	Transition Kernel of Metropolis Hastings Algorithm
α	Error Rate
$\omega(x)$	Weighting Function
$\hat{\omega}$	Normalised Weights
$\Pi(x)$	Target Distribution Function
$\pi(x)$	Target Density Function
$Q(x)$	Filter Function, Proposal Distribution
$q(x)$	Proposal Density Function

$\tilde{\Pi}(x)$	Estimator of Π for Particle Filters
x, y, z	Observable Variables
v, w	Unobservable Noise
ξ, ζ	Unobservable Variables
$\hat{\xi}, \hat{\zeta}$	Estimators for ξ and ζ
\underline{q}_{max}	Maximum of all Lower Quantiles
\overline{q}_{min}	Minimum of all Upper Quantiles
$I_{\underline{q}_{max}\overline{q}_{min}}$	Interval $[\underline{q}_{max}, \overline{q}_{min}]$
$\hat{f}(x)$	SamPro Estimator for Merged Density Function

Chapter 1

Introduction

“You can’t control what you can’t measure.” (DeMarco 1982, p. 3) This frequently cited statement has been widely used in project management but is also valid in almost every management context. Quantitative methods are very powerful in sciences and the evolution of computer techniques has enhanced the usage of Business Intelligence in almost every area of decision making and control.

Although the usage of business indicators as quantitative elements for the description of a firm’s state has had a great impact in the last decades, there is also a part that is indeterminate. McDonough (1963, p. 108) defines this decision gap “as a conceptual distance between the point where we stop further formal analysis of a particular phase of a problem, take our best view of what we see, and leap through or over what still remains unknown to a decision.” In order to improve the information basis for a decision and reduce the decision gap, the information degree should be optimised. However, this optimisation is restricted by the principle of economics (Albach (1961, p. 364); Staehle (1969, p. 35)).

Currently, indicator systems in controlling, decision making, or planning use crisp data. Excel spreadsheets as well as special software are important companions of managers. In case of different possible values for a specific indicator, a decision has still to be taken for further investigations. Thus, an important question is, how to deal with uncertainty of indicators?

Aitchison (1986) introduced the theory of compositional data, where econometric methods are useful only if all variables are linear related and sum up to a unity. Nevertheless, the linear hypothesis is rarely valid. Therefore the relaxation of this assumption is a fundamental objective of this thesis.

Kraemer (1991) deals with the problem of uncertainty by introducing classes of tolerance. The implementation as a control centre that handles such differences is discussed in Kraemer (1993). An open question of this decision tool is how to choose the classes, its bounds, and avoid model inconsistencies.

1.1 Problem Statement

Several kinds of uncertainty can occur when using indicator systems. In the following, a short description of some selected factors and application areas is given that are causes for stochastic information.

- **Uncertainty.** Uncertainty occurs if only limited knowledge for a state is available. Predictions of the future, physical measurements or unaccountable features for a formal model create for instance uncertainty.
- **Measurement.** Errors measuring is always connected with errors. This may result from physical processes of the event that has to be measured or from the measurement instrument.
- **Fraud.** Fraud and fraud detection are application fields in business control. Typically, frauds are not considered in common business models. A decision tool might be helpful to assist fraud detection in a company.
- **Forecasting.** In order to make decisions based on future outcomes, it is often necessary to predict future values.

In order support decisions, the usage of all available information is necessary. This includes stochastic or imprecise information. The following very simplified example should illustrate the problem.

In a company two major cost factors C_1 and C_2 exist. For the next period the sales amount is estimated with 100 million €. This value has an error of 10%. The overall cost are estimated with 80 million €. C_1 will be approximately 45 million € and C_2 40 million €. The sum of both cost is not equivalent to the overall cost of 80 million €.

Therefore two problem solution tasks arise for the management: on the one hand uncertainty within indicators should be reduced and inconsistency should be revealed. On the other hand unknown values have to be computed. So the question arises: how to handle data with uncertainty given a fully specified model?

Addressing and solving such problems is the subject of this thesis.

1.2 Contribution

The main objective of this work is the analysis and the usage of indicator equation systems with imprecise data. Mathematically necessary restrictions should be relaxed as far as possible.

A widely known and used methodology dealing with complex stochastic systems is the Markov Chain Monte Carlo (MCMC). MCMC algorithms have been extensively described, c.f. Tanner (1993), Chib and Greenberg (1995), Gilks et al. (1996b), Liu (2001), Robert and Casella (2003), Gamerman and Lopez (2006). All authors use their own notations, descriptions, and examples of their specific

domains. A consistent notation of our own is used in this thesis. Examples from the business area are given and the pseudo code for the most important algorithms is presented. Furthermore, a brief introduction of the methodology is given.

The major contribution of this thesis is the introduction of the Sampling and Projection (SamPro) algorithm. As an intermediate result of the simulation process several estimates for a single variable are available. SamPro merges this information and generates an improved estimate. The main properties and aspects of this MCMC algorithm are discussed in detail as well. It is important to underline though, that SamPro is not restricted to MCMC methods. Any other random number generator can be used as well.

The second contribution of this thesis is Modelling and Simulation (MoSi), a software prototype for stochastic simulation. Two aspects were essential during software development. On the one hand, the resulting tool should be easy to use but not restricted to specific classes of distributions. On the other hand, the tool should have an acceptable performance while generating precise results. Therefore, a prototype with a graphical user interface written in Java was developed. The simulation process is implemented using the statistical programming language R.

A further objective of this thesis is the comparison of the developed method with other available tools that deal with the topic of decision making using indicator systems. The developed approach is compared to two other software packages. The comparison presents an analysis and outlines the achieved optimisation of the MCMC algorithm.

1.3 Thesis Overview

The remaining part of the document is structured as follows.

Chapter 2 introduces business indicator as well as economic indicator systems. In the literature these systems are presented in different and sometimes contradictory ways. Hence the main indicator systems are described. The developed method for handling uncertainty in indicator systems can deal with all described systems and many more.

Firstly, Chapter 2 delivers a brief introduction of the area of modelling. Then, classical business indicator systems are described. Moreover, this chapter discusses Balanced Scorecards, introduced by Norton and Kaplan in the early 90s. Their indicator system not only considers financial indicators, but also non financial indicators.

The developed methods are not restricted to business indicator systems. Economic indicator systems can also be used with methods for handling uncertainty. Therefore, economic indicator systems are included in Chapter 2.

Modelling of indicator systems raises the question how uncertainty is handled. Chapter 3 describes several related methodologies. Firstly, Fuzzy set theory and its application to equation systems are introduced. Secondly, FuzzyCalc[®], an Excel Plug-In is briefly described, which is based on this theory.

Stochastic models can be utilised for univariate or multivariate functional description. Some selected cases are given as well as the basic model for errors in the variables. Furthermore, an approximation approach is presented.

In order to relax the assumptions made in the stochastic (general least squares) approach, simulation is an appropriate choice. The basic concepts and defiances of Monte Carlo simulation are presented in Chapter 3. This includes Markov Chain Monte Carlo techniques that facilitate more generalised algorithms for random number generation.

In the context of indicator systems it is necessary to deal with more than one estimate for a given variable. An algorithm merging several estimates per variable is presented.

The presented Metropolis Hastings algorithm is very powerful if it is applied in its frame of discernment. Robustness of the developed MCMC technique is a mandatory requirement in order to reduce misinterpretations. This leads to an algorithm for stochastic indicator system simulation called SamPro.

A comparison of three different approaches is presented in Chapter 4. In the first study the Fuzzy set, GLS approximation, and simulation approach are evaluated. Another comparative study deals with different effects of stochastic properties: (1) skewness, (2) dependency, and (3) non-linearity. The chapter is closed with an example of a stochastic Balanced Scorecard implemented with the SamPro algorithm.

Chapter 5 addresses the issue of software design and implementation. A short description of the tool MoSi is given. For this purpose Unified Modeling Language (UML) 2.0 notation is used, and a software demonstration is given. The implementation itself is done in the programming languages Java and R.

Finally, Chapter 6 concludes the thesis. The main contributions are summarised and additionally a discussion on further issues is given.

Chapter 2

Business and Economic Indicator Systems

This chapter discusses several business and economic indicator systems. Most of them are widely used or adapted to the needs of companies or countries. However, none of the described indicator systems is able to capture the uncertainty of the indicators. This is partly due to the fact that they just represent the past and no measurement errors are made explicit. However, if immediate reaction on variations or differences of current values is required, then the introduction of probability or possibility is inevitable. A further advantage of using indicators with uncertainty is the projection into the future. In this way the vision of a company or the goals of a country or organisation can be adjusted more precisely. Nevertheless, indicator systems with uncertainty will not prevent that indicators cannot be matched with an underlying indicator system. If the uncertainty is too large, new or more precise information will not be delivered.

Data might violate an underlying equation system. That means the data is in conflict with the rules that generate this data. Two possible error sources are plausible. The underlying model is not true or the data might be wrong. Definition or balance equations can be seen as error free and therefore conflicting data has to be investigated. A major assumption is that the underlying equation system holds true. For this reason indicator systems are presented that are widely used and build up from definitions, balance or behavioural equations. Despite the usage of business indicator systems economic indicators systems are presented as well.

In this chapter, the indicators are presented independent of being crisp or random indicators. First, general requirements are presented together with the notation and introduction of general indicator systems. After that, several classical business indicator systems are presented. The Balanced Scorecard as a management tool is thoroughly described. In conclusion, a description of a collection of economic indicator systems is given.

The management of a company is in need of indicators due to the following

facts (Aichele 1997, p. 4),(Schott 1988, pp. 13):

- Separating the relevant from the nonrelevant in the complex and abundance business environment and obtaining aggregated information from unsorted data.
- To think of relationships rather than monetary factors and to mind productivity instead of production.
- Looking on causal relations (cause and effect) and the interaction of positive and negative factors.
- To benchmark the company's competing position and to react in-time to market changes.
- To implement a goal oriented management tool for thinking and action.

Business indicators are quotients, ratios and indices that represent business factors, such as production amount, sales, or capital. Economic indicators are build up to represent main factors of economies, such as Gross Domestic Product (GDP), investments, or consumption.

2.1 Modelling

An indicator system is an abstract, isomorphic model of a part of the real world. In order to reduce the complexity of information systems, the basic principles of correct modelling have to be considered. This is also important for the modelling of indicator systems.

2.1.1 System and Models

The basic principle of correct modelling is founded on accounting. The following six principles build the frame of the essential model quality criteria: (Rosemann et al. 2005, p. 48)

- accuracy
- relevance
- economy
- clarity
- comparableness
- methodical layout.

The basic principle of **accuracy** means that the represented view on the reality is correct. A qualitative model requires both the structure and the behaviour to be correct. In other words it has to be semantically sounds. In the case of an indicator system, the indicators and the connection between them have to be properly modelled. On the other hand, the syntax can be distinguished from this point.

The basic principle of **relevance** says that a model has to map all relevant circumstances from the user's perspective. Irrelevant information has to be avoided. A business indicator model should contain all indicators that are relevant for the objectives of the managers. An indicator that is directly required for a report is just as relevant as all indicators that generate information for this particular indicator like right hand side variables of an equation where the left hand side variable is the report indicator. Irrelevant indicators do not have any relations to relevant ones. Nevertheless, an over determined indicator system is not a model with irrelevant information because an over determined system can reveal model inconsistencies or reduces uncertainty. A good rule for the relevance of a model is that there is an information loss of the user if some part of the model is eliminated.

The basic principle of **economy** tries to balance the cost and the benefit of a model. In the case of modelling, the amount of work of generating the model should be reasonable compared to the usage of the indicator system. Already existing models reduce the cost. Such models are intensively discussed in the literature. Efficiency determination is a complex task and practically only an

estimation of the benefit is feasible. Another aspect is the time or computational effort and the additional value of different simulation runs. In order to optimise simulation runs, indicators and equations that do not add new information to the user can be omitted. However, indicators that generate additional information even on an intermediate level might be considered as elements of the model.

The basic principle of **clarity** states that a model is only useful if the user of the model is able to understand it. In other words, a clear model should have a reasonable degree of intuitive readability. Methodical requirements should be low. In the context of indicator systems mathematical requirements such as the representation of equation systems can be set as standard. However, the meaning of the indicators and the units should be given to the user as meta information, i.e., with comments or definition of the indicator because different user groups might interpret indicators in different ways. A formal, mathematical description is usually self-explanatory.

The basic principle of **comparableness** brings model comprehensive usage into account. This might be necessary if a complex model is created in different teams and the stand alone models have to be integrated. There are two main options for the creation of comparable models. It is generated either by using a meta model which does not only describe the elements of a model but also the requirements, or by following guidelines that describe the steps of the process of model generation. While in the first approach all objects can be validated, the second approach commits the model architect to the guidelines.

The basic principle of **methodical layout** is based on the fact that the complexity of the real world is reduced by models but the model complexity has to be handled, too. Well defined interfaces are necessary for corresponding models. Input data of a model that is generated by another model should refer to the data description part of the exporting model.

These six principles can be implemented in many modelling ways. The highest level of specificity is reached in specific modelling techniques or in modelling conventions. A modelling technique describes how circumstances are represented in a model. A modelling convention is on the other hand an individual form of modelling technique.

In the following the basic elements of business indicators and indicator systems are defined. Indicators appear in every area of business and economy. In production output quantity is measured, in marketing client contacts are used and an investment is evaluated by the generated cash flow. However, in economics indicators such as personal consumption, corporate taxes and Gross Domestic Product (GDP) are used. For more details on economic indicator see Section 2.4. Business and economic indicators condense the complex reality. However, it is necessary to know the structure and the properties of the indicators. Due to the fact that this work uses indicators and indicator system in the context of simulation, a definition of indicators and indicator systems is given. Business and economic indicators are used interchangeably from here on. The terms indicator, ratio, characteristic or index, measure, parameter, or benchmark will be used synonymously.

Although business and economic indicators have been used for a very long time, a generally accepted definition does not exist. There are several opinions as to what a business indicator is exactly. Antoine (1958), Staehle (1969), Reichmann (2006), Gladen (2003) all define business indicators in different manners. The main discrepancy is the use of absolute indicators.

(Staehle 1969, p. 52) defines business indicators as ratios and absolute indicators that refer to numerical ascertainable business facts in a concentrated mode.

Gladen (2003) distinguishes between indicators in the broader sense and indicators in the narrower sense. “Business indicators in the narrower sense are measurements that are deliberately highly consolidated as absolute or relative values to report in a concentrated form about a numerical ascertainable circumstance. [...] Business indicators in the broader sense are quantitative information that are pre-processed for specific requirements of the analysis and controlling of the company” (Gladen 2003, transl. p.12). Indicators in the narrow sense are a subset of indicators in the broader sense.

Pre-processing and consolidation leads to an information loss. Nevertheless this loss is intended and called an “unloading of information” (Gladen 2003, p. 12).

Indicators can be characterised by different modes. The following three modes are taken from Meyer (1976). A categorisation of business indicators into business functions like supply, production or finance can be applied. A structural mode can have a quantitative aspect, a focus on time structure or reference to topics. Under the statistical methodical aspect indicators are handled as absolute or relative numbers with additional information about the error term.

In case indicators are treated with a statistical method, a further differentiation into absolute and relative indicators is necessary. In the English literature business indicators are often equated with ratios, which is too restrictive. A business indicator can have absolute or relative values. Fig. 2.1 gives a taxonomy of possible business indicators.

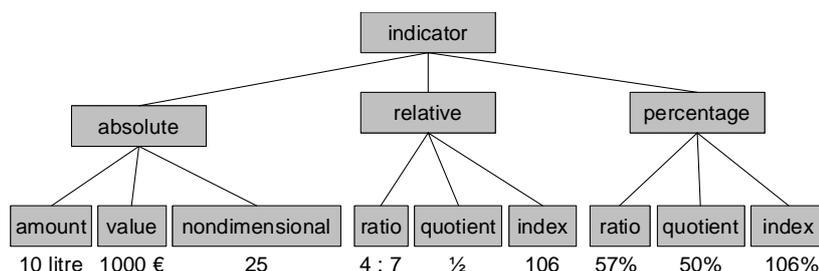


Figure 2.1: Hierarchy of Indicators

An absolute indicator can represent an amount, i.e., refined barrels of oil, a value like available cash or a simple fact. Absolute indicators can be stand alone indicators, means, sums or differences. Although stand alone indicators

have limited explanatory power, they are the most used indicators in business and economics. Examples for the absolute indicators are given below:

categorisation	example
stand alone indicator	number of employees
mean	average inventory on hand
difference	profit
sum	balance sheet total

Relative indicators can be distinguished into indicators of quotients, ratios and indices. Quotients show structural dependencies and are often used in benchmarks. They can be represented as

$$\text{quotient} = \frac{\text{indicator}_{\text{subset}}}{\text{indicator}_{\text{total quantity}}}.$$

Ratios are used for the comparison of characteristics that are similar but have different dimensions. A formal definition of a ratio is:

$$\text{ratio} = \frac{\text{quantity type 1}}{\text{quantity type 2}}.$$

Indices are useful to compare conditions to another time point or to total volume. An index in time t can be written as

$$\text{index}_t = \frac{\text{quantity}_t}{\text{quantity}_0} \cdot 100.$$

Quantity_0 is the basis indicator.

Examples of relative indicators within a business area are given below:

categorisation	example
quotient	fraction of equity to total capital
ratio	ROI
index	price index

Definition 2.1.1 (Business Indicator). Let G be the set of all goods and $k : G \rightarrow \mathbb{R}$ a measurable mapping. k is called business indicator for every subset of goods $q \subset G$ with $q \mapsto k(q)$.

In this work, a business indicator is a statistic on business that allows analysis of business performance and predictions of future performance.

A single business indicator might be of relevance for executing a planning or controlling function. However, most business cases can be adequately managed if several indicators are employed (Müller 1990, p. 56). Moreover, heavily condensed business indicators tend to diverge from reality, thus necessitating further indicators. Such a group and partial ordering of indicators constitutes an indicator system with a decreasing degree of compression (Gladden 2003, p. 12).

Due to the fact that indicators are interrelated, the relationships should also be qualified. Cause-and-effect relations, correlations or equations can model these relationships between indicators. Figures and their relations provide information in their totality (Horváth 1994, p. 555).

An indicator system should be clear and easy to use. The clarity of the model is another restriction for the use of further indicators, and the basic principles of correct modelling play an important role. In the case of an indicator system with a decreasing degree of aggregation, subordinated indicators are only used if required.

Definition 2.1.2 (Indicator System). (K, r) is called indicator system where K is the set of indicators and $\forall k \in K$ are indicators as defined in Def. 2.1.1. $r : K^n \rightarrow K^m$ with $n, m \in \mathbb{N}_+$. r is a relation between a certain number of indicators $k \in K$.

Def. 2.1.2 formalises indicator systems. In the following a more concrete representation, necessary for the simulation of indicator systems, is given.

A fundamental assumption of this work is that all necessary indicators are quantified, i.e. have a metric scale. It is not necessary that all indicators of an indicator system are characterised by a stochastic distribution function. Due to the fact that the indicators are connected within a system, as stated in Def. 2.1.2, further stochastic elements occur due to error propagation.

Simulation of indicator systems requires an extension of Def. 2.1.1 and Def. 2.1.2. A business or economic indicator can be part of the measurement process or the result of using the indicator system.

An indicator system consists of m business indicators. The true but unobservable state variables for these business indicators are given by the m -dimensional vector $\xi = (\xi_1, \dots, \xi_m)$. The relations between the variables are defined with a (n, m) matrix H where $\text{rank}(H) = n$ and $n \leq m$. $\zeta = H\xi$ are balance equations and ξ as well as ζ are only observable with additive errors that are latent:

$$x = \xi + v \quad (2.1.1)$$

$$z = H\xi + w. \quad (2.1.2)$$

The errors $u = \begin{bmatrix} v \\ w \end{bmatrix} \sim F(\cdot)$ are distributed according to a distribution function F . Further assumptions are, that $v \perp w$, $E[u] = 0$, H is measurable. This means that the system H of relations between ξ, ζ is error free. The only errors that occur are given by u . A possible distribution function F might be a Gaussian distribution $N(0, \sum_{uu})$. Note, that the matrix H can be generalised to a function, $\zeta = H(\xi)$, as described in Lenz and Rödel (1991).

2.1.2 Visualisation

In this section, a graphical style for showing and using business indicator systems is introduced and discussed. The equation graph is explained, which is a useful model for deriving the steps of solving the business model.

Model Graph

In this section, graph representation is used to describe equation systems. Using restriction on the operators allows resolving the equation system with algorithms from graph theory. Two kinds of nodes exist in an equation graph. One node represents variables and the other operators. A restriction of mathematical operators is, that they are separable. This means for such an operator the equation can be uniquely solved for all variables of the equation.

An arc connects an operator node with a variable node. The direction of an arc states whether a variable is a Left Hand Side (LHS) or Right Hand Side (RHS) variable. An arc pointing to the operator node denotes an input variable or RHS variable. An arc with direction to the variable node implies a LHS variable. Furthermore, the arcs are numbered to obtain the structure of the equation. The arcs directed to an operator are numbered starting by 1 and the outgoing arcs are also numbered starting by 1 if more than one arc exists.

The following example illustrates an equation graph.

Example 2.1.1 (Equation Graph). The equation graph for the following equation is given in Fig. 2.2:

$$A = (B + C) * D - E/F.$$

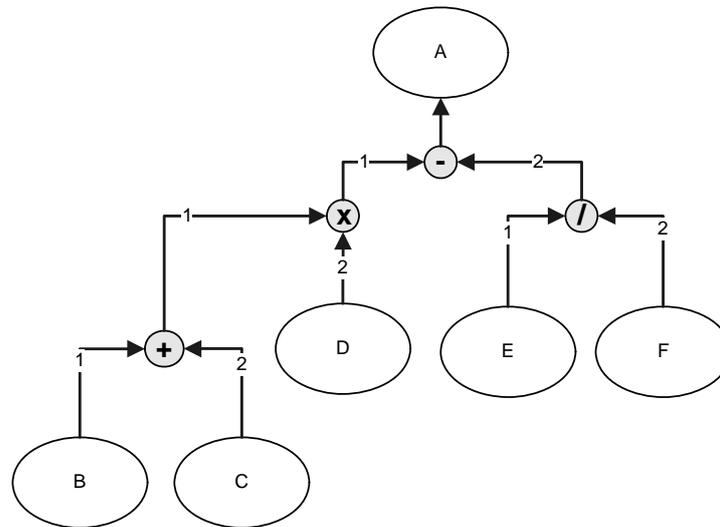


Figure 2.2: Equation Graph (Parser Tree) for Equation $A = (B + C) * D - E/F$

The division operator takes as input E and F where arc number 1 determines the dividend and arc number 2 constitutes the divisor. Division has only one result (quotient), which is the subtrahend for the subtraction operation. Addition of B and C is also numbered here but is due to commutative property of addition not really necessary. The sum of B and C is one factor of multiplication whereas D is the other. The product is minuend and the difference is A .

In the case of commutative operators, the preorder traversal can be omitted in order to reduce the complexity of this graphical representation and to obtain a more user friendly representation. A further, simplifying assumption is that each equation consists of only one operator. In the case of an equation with more than one operator, this assumption can be achieved if the equation is split into several equations and intermediate or dummy results are introduced.

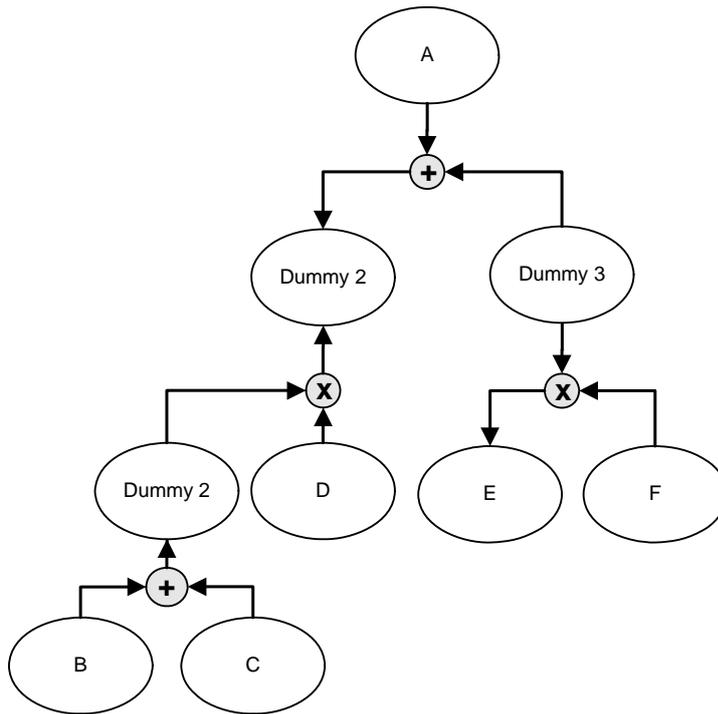


Figure 2.3: Another Equation Graph for Equation $A = (B + C) * D - E / F$

In Fig. 2.3, the same equation as in the example described above is used. Substitution and introduction of three dummy variables guarantee that each equation consists of one operator. Division and subtraction are exchanged by rearrangement so that only commutative operations occur.

The advantage of one operator per equation and the usage of commutative operators lies within the reduction of graphical interpretation. This even simplifies the algorithm structure to solve the equation system. However, this is connected with the introduction of further (dummy) variables and the operator node restriction. Consequently, the original equation system might not be obvious anymore.

A fundamental advantage of the graphical representation is that the relational structure is more obvious. Disjoint graph elements suggest that there is no relationship between these parts of the model.

2.2 Classical Business Indicator Systems

Business indicator systems have been traditionally used in order to understand, evaluate and rate the current or future state of a business. The evaluation of companies or even countries has become more important than ever. Reasons for that are, that knowing the current state is the basis for improvements. Internal decisions and decision making processes for a business' future rely upon evaluations.

The evaluation of a company depends upon the view of the analyst. Financial indicators are often used for business analyses for both internal and external users. There is no restriction to a single company or business sector. Even countries have to evaluate their current states, for instance to prove credit status or report past activities.

Due to the multitude of activities and events within a single company it is necessary to abstract from details. This is achieved by indicators that represent meaningful aggregates.

This section elaborates on classical business indicator systems. Subsequently, a new indicator system is described.

2.2.1 DuPont System of Financial Control

The chemical company E.I. DuPont de Nemours developed a key indicator system for financial aggregation in 1919. (Schwarz 2002) The top key indicator in this system is the Return on Investment (ROI)(Staeble 1969, p. 69). This indicator stands for the periodical success of the company. It is the quotient of the profit and the capital that was used in the same specific time period. Another interpretation of the return on investment can be drawn if the complete profit of an investment is used. However, this interpretation is in most cases not applicable for the DuPont system. The reason is that the DuPont system is used for periods (i.e., quarterly or annual reports) and most investments will be applied to overlapping periods. ROI can be computed by the following formulas:

- $ROI = \frac{\text{profit}}{\text{sales}} \cdot \frac{\text{sales}}{\text{equity}}$
- $ROI = \text{profit margin} \cdot \text{capital turnover}$
- $ROI = \frac{\text{profit}}{\text{equity}}$

The DuPont measurement system is a financial indicator system. The original Model developed by DuPont is shown in Fig. 2.4. The top business indicator is ROI, which can be computed as a product of capital turnover and sales profitability. The business indicator capital turnover is a quotient of sales and capital, which is divided in current assets and capital assets.

The DuPont measurement system is still used. However, definitions of some business indicators were changed and several new business indicators were added because businesses grow more complex and the measurement has to be done on

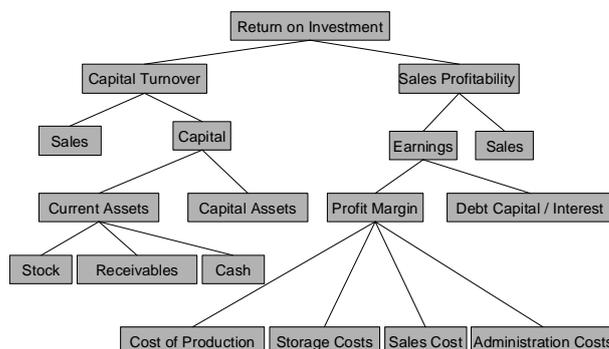


Figure 2.4: The Original DuPont Measurement System
adapted and translated from Aichele (1997, p.93)

a lower level, cf. Fig. 2.5. For instance, the ROI was modified over time and

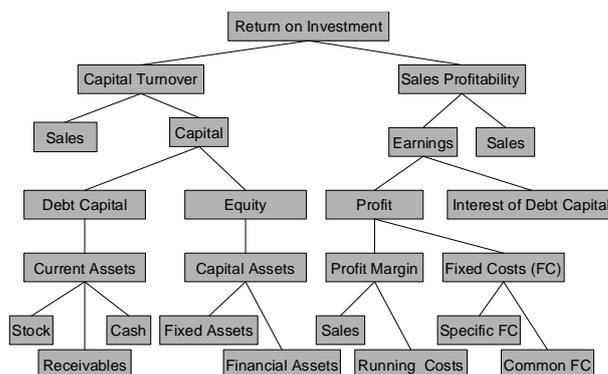


Figure 2.5: The Extended DuPont Measurement System
adapted and translated from from Staehle (1969, p.69)

key indicators like Economic Value Added (EVA) or Return on Capital Employed (ROCE) are also used nowadays.

In this work the simplified DuPont system is used due to the fact that all necessary operations exist but the complexity of the indicator system is manageable. The simplified DuPont system is shown in Fig. 2.6. This simplified business indicator system consists of seven business indicators that are related as follows:

$$\begin{aligned}
 \text{transaction volume} &= \text{profit} + \text{cost} \\
 \text{capital turnover} &= \text{transaction volume} \div \text{capital} \\
 \text{profit margin} &= \text{profit} \div \text{transaction volume} \\
 \text{return on investment} &= \text{profit} \div \text{capital}
 \end{aligned}$$

Note, that all indicators are separable. Further equations like $\text{capital} = \text{transaction volume} / \text{capital turnover}$ can be used for computing the variables.

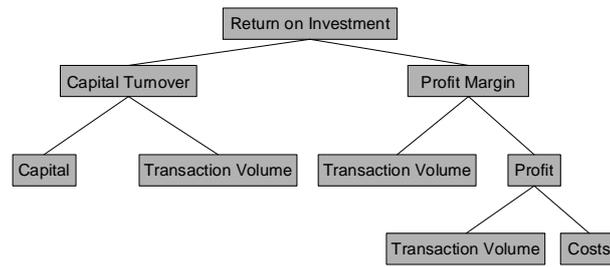


Figure 2.6: The Simplified DuPont Measurement System
adapted and translated from Müller and Lenz (2003)

Another representation for the simplified DuPont Systems is given in Fig. 2.7. Here the notation is used as described in Section 2.1.2. The operators in the DuPont Equation Graph are restricted to addition and multiplication. However, due to the fact that these indicators are separable, all other equations can be resolved.

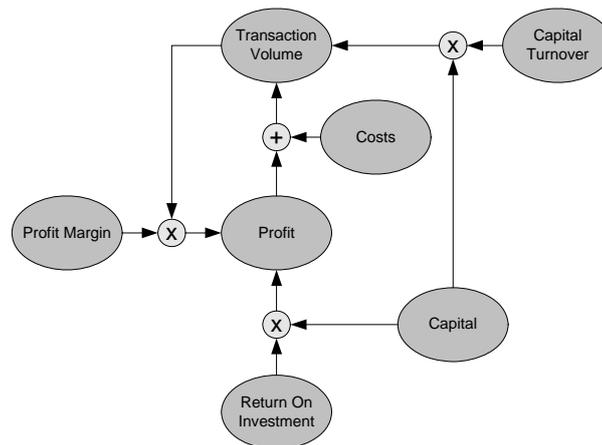


Figure 2.7: The Equation Graph of the simplified DuPont Measurement System

Advantages of the DuPont system

- it can be used for a whole company or only for a fraction or department.
- it has a control and allocating function.

Disadvantages of the DuPont system

- the system can only be used for companies where a profit can be acquired.
- the system is orientated in the short run only. Long term potentials will not be considered.
- due to the usage of mean values it might be aspiring to reach only the average.

2.2.2 ZVEI - indicator system

The German Electrical and Electronic Manufacturer's Association extended the DuPont system in the 70s. The actual version of their business indicator system was released in 1989. The system consists of 128 business (Zentralverband Elektrotechnik- und Elektronikindustrie e.V. ZVEI 1989, pp. 185–207) and 121 auxiliary indicators (Zentralverband Elektrotechnik- und Elektronikindustrie e.V. ZVEI 1989, pp. 209–220). All main business indicators are part of balance sheet or profit and loss account. The system is independent from industries but allows a detailed business analysis. Both growth and structure of a company can be evaluated. While growth is analysed with comparing absolute indicators, structure analysis uses and compares absolute and relative indicators. The top indicator of interest in the German Electrical and Electronic Manufacturer's Association (ZVEI) system is Return On Equity (ROE).

Fig. 2.8 summarises the main focusses in the ZVEI system. The upper part represents the growth analysis where periods or different time points of the company are evaluated. The lower part in the indicator depicts the structural analysis. Here the return on equity is the top indicator. Further indicators can also be incorporated into the ZVEI business indicator system.

In the growth analysis business volume, employees and business success are evaluated. The Business indicators are all part of annual accounts. The nine key indicators of the growth analysis are:

- business backlog,
- sales,
- added value,
- personnel cost,
- employee,
- earnings before interest, taxes, depreciation and amortisation (EBITDA),

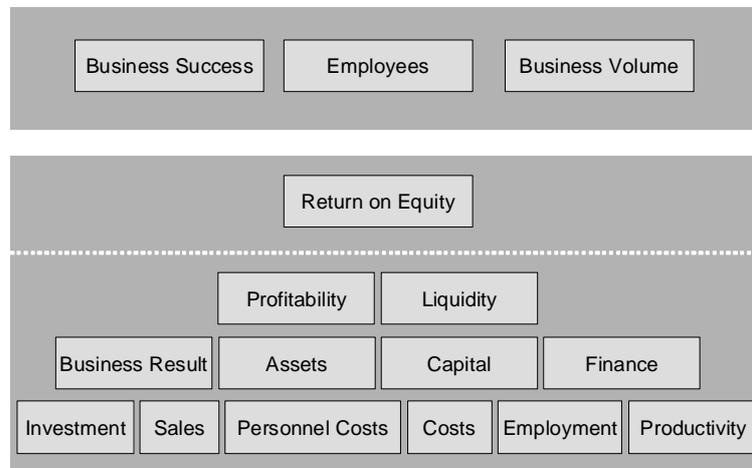


Figure 2.8: The ZVEI Business Indicator System
translated from Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
ZVEI (1989)

- profit or loss on ordinary activities,
- profit for the year, and
- cash flow.

In the structural analysis the current indicators and the respective indicators from the previous period are investigated. The business indicators are related via a mathematical equation system and ordered in a hierarchical way like in the DuPont system. Nevertheless the top indicator of interest is return on equity. Analogues to the DuPont system business indicators are segmented into elements that describe the indicators. Thus, the cause-and-effect relation can be investigated. Due to the fact that the ZVEI system consists of more indicators, a better differentiation and a more precise cause-and-effect analysis can be done. The main focus is set on the following indicators and corresponding formulas:

- investment in per cent of equity/assets
- constitution in per cent of assets/current assets
- liquidity on sales in per cent current assets/current liability
- liquidity on collection in per cent (cash + receivable)/current liability
- absolute liquidity ratio in per cent cash/current liability
- finance in per cent equity/bonded capital

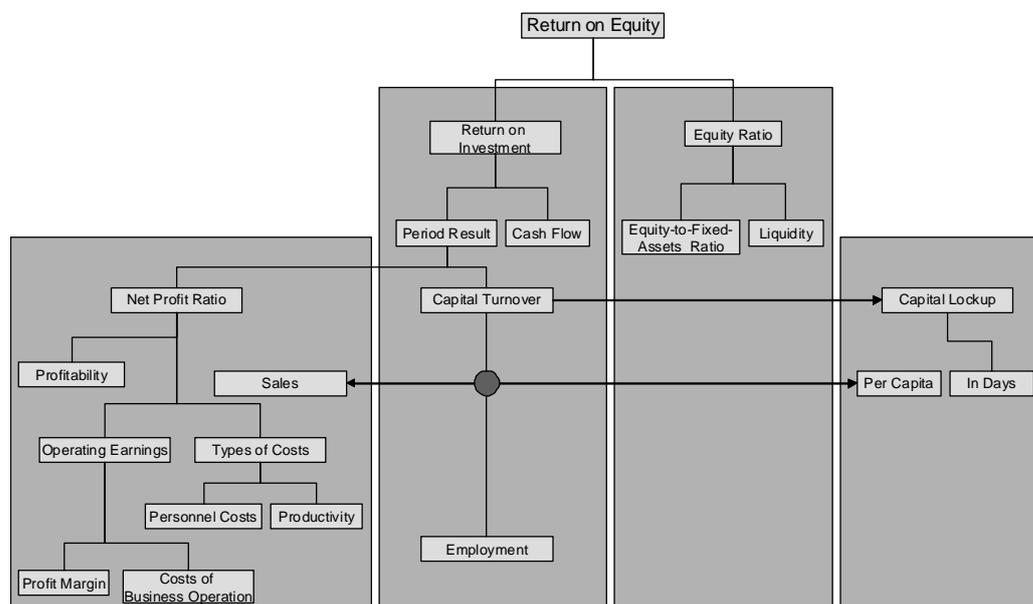


Figure 2.9: Structural Analysis of the ZVEI Business Indicator System
translated from Zentralverband Elektrotechnik- und Elektronikindustrie e. V.
ZVEI (1989, p. 118)

Restrictions of the ZVEI system are large number of auxiliary indicators that are required in a formal way but do not have a business background, the compilation of indicators is arbitrary and the system is mostly used for internal purposes. (März 1983, pp. 75)

2.2.3 RL business indicator system

The RL system was developed by Reichmann and Lachnit (1976). It focuses on profitability and liquidity. This system aims to assist the management control of the corporation. The key indicators in this system are profit and liquidity. The system is divided into a common and a special part. The common part is permanently used within the management for planning and controlling. The special part is used for completion depending on the company. Fig. 2.10 simplifies the RL business indicator system (Aichele 1997, p. 103). The characteristics of the RL system are:

- The system is useful for controlling liquidity and profit.
- Only a few fiscal business indicators are involved.
- The system is not designed for making a connection to business processes.
- Most of the indicators are evaluated once a year or quarterly.

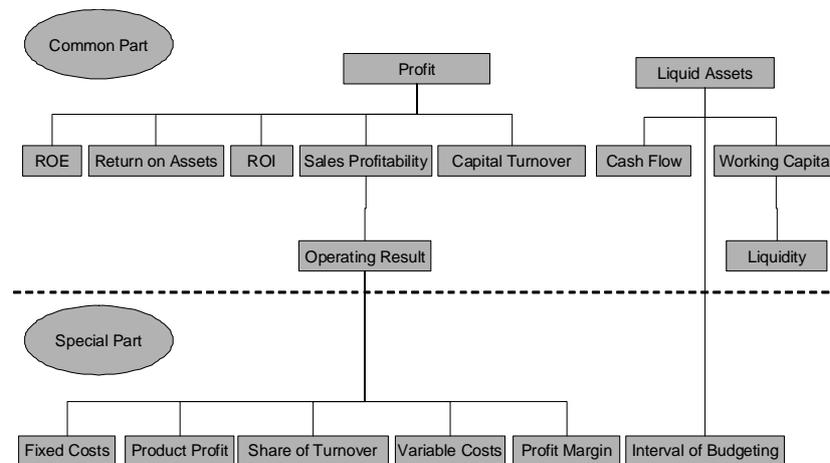


Figure 2.10: The Simplified RL Business Indicator System
translated from Aichele (1997, p. 103)

The common part refers to the whole corporation, whereas the special part is divided into a product specific and a division specific business management. The system aims to control profit and finance at the same time.

In the common part business indicators are used as management information that is independent from the type of business (Lachnit and Müller 2006, p. 269). It is not sufficient to manage a company just by looking at profitability like ROI. This is only one view on the investment profitability. Further indicators such as Return on Equity (ROE) and return on assets on the complete annual profit and loss (Lachnit and Müller 2006, p. 270) have to be involved in the evaluation of a company. The key indicators within the common part are ordinary operating result and capital employed (Lachnit 1998, p. 37). A company can only survive if the ability to meet financial obligations is given at anytime. This ability is determined by revenues and expenditure dependent on time, where flow and stock indicators are used to summarise this information (Lachnit 1998). The top business indicator in the liquidity part is the stock of liquid assets.

The special part is used to supplement the controlling system with specific company needs and concrete implementation of the RL system (Lachnit 1998, p. 32). This extension can be based on the industry sector, the structure of the company or size of the business. Under these characteristics it is reasonable to divide the special part into RL of products and of organisational units (Lachnit and Müller 2006, p. 280). The key indicators within the product specific part have to go more into detail and ordinary operating result and capital employed should be decomposed dependent on individual products, product groups or organisational framework. The business indicator cash flow profit margin makes the effect of payment for products and product groups clear. This underlines the liquidity in the product specific part. The organisational specific part divides the profit and

finance components into the organisational framework (Lachnit and Müller 2006, p. 280). This might be departments, burden centre, position or profit centres. Unexpected and exceptional circumstances can be reported in a case based special part. In this part all changes are specified that are not element of the other parts. Problem oriented modules have to be developed for the particular abnormality (Lachnit 1998, p. 40).

2.2.4 Further Classical Business Indicator Systems

The **Ratio au Tableau de Bord** is a business indicator system developed in 1959 by Lauzel and Cibert (1973). It gives more in-depth information than the DuPont system and includes a comprehensive analysis of balance sheet and P&L. (Aichele 1997, p. 98) This system is aligned to management of business. It is

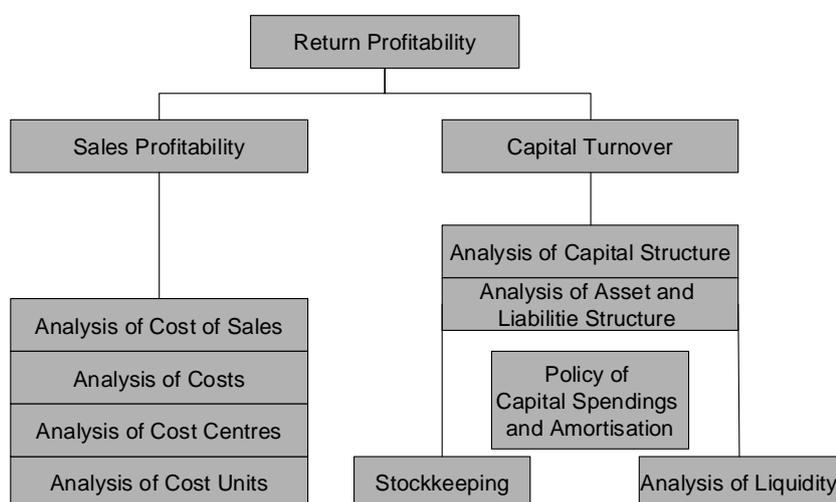


Figure 2.11: Ratio au Tableau de Bord
adapted and translated from (Lauzel and Cibert 1973, p. 26)

useful for internal controlling and for delegating decisions. Fig 2.11 summarises the system. The business indicators of this system are elements of a matrix that relates factors of production with reasons of changes.

The **Managerial Control Concept** is the result of a heuristic case study where 200 US-American businesses were involved (Tucker 1961). This system is divided into the following functional areas: production, sales and finance. Different management levels and therefore different indicators for analysing are assigned to each area. The indicators are differentiated into elementary indicators, optimised indicators and aggregated indicators. While elementary indicators establish simple relations between operational measures (Tucker 1961, p. 29), optimised indicators connect specific operational circumstances to obtain a deeper insight into a certain area (Tucker 1961, p. 51). Aggregated indicators are obtained by combining elementary and optimised indicators. They are used in the management process

as a decision basis. Fig. 2.12 summarises the Managerial Control Concept from Tucker.

Production	Sales	Finance
Elementary	Elementary	Elementary
Optimised	Optimised	Optimised
Elementary	Elementary	Elementary

Figure 2.12: Managerial Control Concept

The **Pyramid Structure of Ratios** was developed by Ingham and Harrington in 1958 (Ingham and Harrington 1962). This system divides the indicators into primary, supporting and descriptive indicators. The system is very similar to the DuPont system but uses only relative indicators. It was developed assuming that the indicators would build the basis for permanent international comparison. The primary ratio is comparable to the ROI in the DuPont system or the ROE

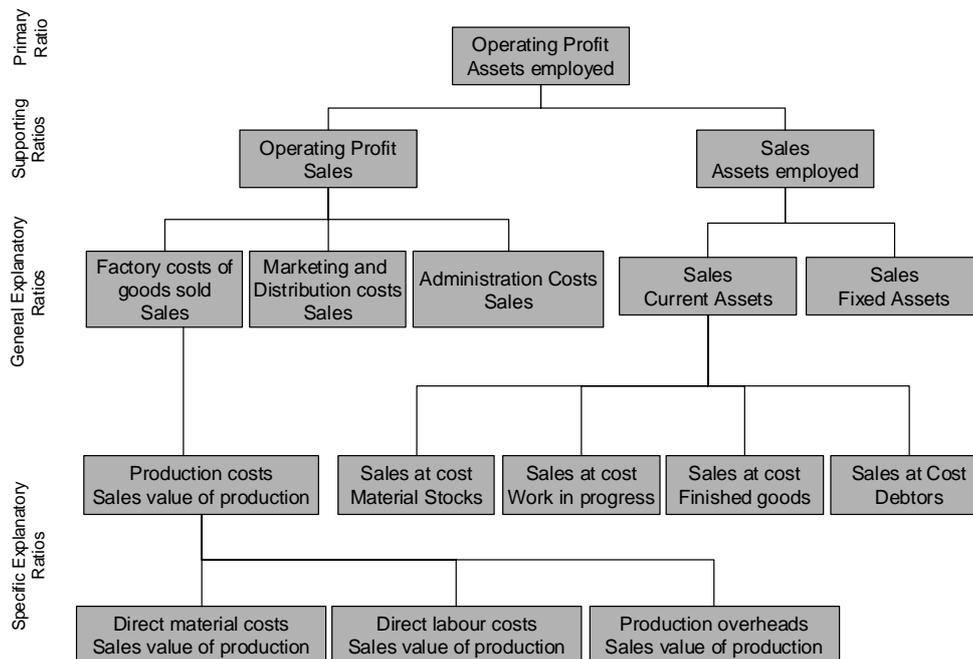


Figure 2.13: Pyramid Structure of Ratios
adapted from (Ingham and Harrington 1962, p. 467)

in the ZVEI system. It represents the top indicator for decision making. This indicator, also called Operating Profit Assets Employed, is directly derived by two other ratios, which are called supporting ratios. Indicators from the supporting

ratio level might be compounded by other indicators that grant a deeper insight into business work. Here two levels are differentiated, the general explanatory ratio level and indicators from the specific explanatory level. Fig. 2.13 represents a specification of the Pyramid Structure of Ratios (Ingham and Harrington 1962, p. 467).

2.3 Balanced Scorecard

2.3.1 Background

“If you can’t measure it, you can’t manage it.” (Kaplan and Norton 1996, p. 21) With this sentence the Balanced Scorecard (BSC) inventors Robert S. Kaplan and David P. Norton made a statement that describes a common problem in the industry: you cannot manage a company if you do not have the performance indicators to manage and control your company. So what about indicators like the mental satisfaction or the motivation of your staff. Can those be measured? With the BSC, Kaplan and Norton represent a management tool for bringing the current state of the business and the strategy of the company together. It is a result of previous indicator systems such as the DuPont system. All these systems use key indicators to represent a company’s business. However, a BSC is more than a business indicator system (Friedag and Schmidt 2002). Kaplan and Norton (2004) emphasise this in their further development of Strategy Maps. Fig. 2.14 shows an overall involvement in a company’s strategy by managing the company with a BSC.



Figure 2.14: BSC Strategy Approach

The Balanced Scorecard as a management method is a top down process, where the company’s vision is formulated in the upper level. Two strategic management levels build up the strategies to reach the vision. The company’s vision describes where the company will go, whereas the company’s strategy characterises the way how the vision is reached. In the strategic controlling process the company strategies are worked out and are transported into the company. The foundation on which the strategies are built is measurement. Due to the complex structure

of businesses, measures are depicted to describe the current state of the company in dependence of perspectives. The strategy is translated into target values for the measurements. On the other hand, aggregation from measures to business is used to reduce the convoluted numbers of measurements. The strategic planning process evaluates the strategies and set targets to fulfil the strategies. Here it is defined which measures are used and collected. Target values that should be reachable by a given period of time are set. Aggregation with business indicator systems are done and a selection of indicators that fit best to the strategies is conducted in order to have a breakdown of the company. Fig 2.14 illustrates the BSC managing structure.

Kaplan and Norton (1996) introduced in the early 90s their concept of evaluating business with the Balanced Scorecard method. This method is used to integrate all relevant aspects into the business success. The Balanced Scorecard is a measurement system that is built up not only by financial business indicators but also by all indicators from key aspects of the company. Kaplan and Norton see this system also for communicating the business strategy into the business processes or units. The relevant aspect in the Balanced Scorecard approach is the linkage between business indicators and the business strategy. A company's strategy is not explained in a single value. Kaplan and Norton suggest the combination of key indicators from different perspectives. In their book (Kaplan and Norton 1992, p. 72), the perspectives Finance, Customers, Internal Business, and Innovation and Learning are pointed out as the most relevant. Friedag and Schmidt (2002) states that a Balanced Scorecard should not be restricted to these four perspectives. Instead, the most relevant company perspectives should be taken. Other perspectives for a company's Balanced Scorecard are Supplier, Creditor, Public sector, Communications, Organisation and Implementation. However, the business indicators have to integrate the whole business. Furthermore, a balancing of the scorecard indicators is necessary.

However, before a decision is made which indicators are used to build up the BSC and the corresponding perspectives, the importance of the indicators has to be examined. Kaplan and Norton divide indicators into hard and soft objectives (depending on the measurability of objectives, e. g., net gain versus a social status of an employee) and short and long-term objectives (which gives the management the possibilities to predict the company's future, e. g., employee fluctuation or customer satisfaction). Kaplan and Norton also consider cause and effect. The three main aspects are:

1. all indicators that do not make sense do not have to be included in the BSC,
2. while building a BSC, a company should be differentiated between performance and result indicators, and
3. all non-monetary values should influence monetary values.

Based on these indicators, a complete system of indicators which turns into or influences each other can be created. Reichmann (2006) describes this mea-

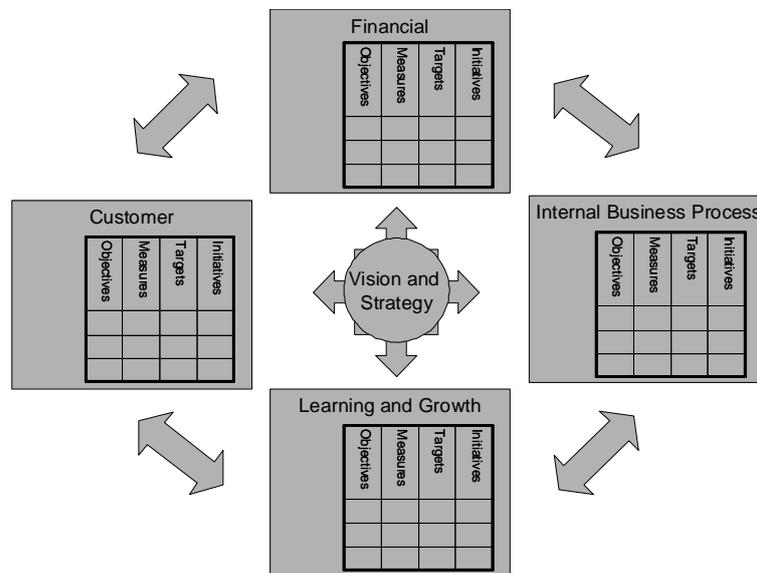


Figure 2.15: Translating a Strategy into Operation
adapted from Kaplan and Norton (1996, p. 9)

surement as an essay of bringing different performance indicators together in a sensible way, so that indicators complete their information and result in a common goal. E. g., all indicators of the well known Du-Pont-Schema result into the common goal of return of investment. One of the disadvantages of traditional indicator systems (e. g., the Du-Pont-Schema) is that a company's management concentrates on short-term periodical gains, without regarding the financial future of their company. Additionally, non-monetary indicators like customer satisfaction are excluded. Omitting these values stands for losing qualified information that can help the management to drive a company's strategy better. Tackling these disadvantages Kaplan and Norton published the BSC, a milestone of modern performance measurement.

The BSC considers the following four perspectives:

- Financial Perspective to reflect the financial performance like the return on investment
- Customer Perspective to sum all indicators of the customer/company relationships
- Business Process Perspective to give an overview about key business processes
- Learning and Growth Perspective which measures the company's learning curve

By splitting a company into four different views, the management of a company gets the chance to have a quick overview over the main perspectives of their company and divide these into usable and unnecessary layers. The management can focus on its strategic goal. It is able to connect qualitative performance indicators with one or all business indicators.

2.4 Economic Indicator Systems

The National accounting system is one of the most well-known economic indicator systems. Countries all over the world use this system to report their financial status to organisations such as the United Nations (UN). The system of national accounts was developed in the 1940s. Both the Measures of National Income and Output and the United Nations System of National Accounts are derived from that system.

The Measures of National Income and Output are used to describe an economy in a past period. These measures are often used in order to control the goals that were originally set by policy. United Nations System of National Accounts (SNA) is an international standard. In the next sections, both systems are briefly described. This is followed by a discussion of relevant challenges.

2.4.1 Measures of National Income and Output

The Measures of National Income and Output are part of macroeconomics. They show an ex post quantitative overview of the economic development of an economy.

The main indicator of the system is GDP (Kuznets 1971). It was developed in the 1940s by Simon S. Kuznets for the Bureau of Economic Analysis, United States Department of Commerce (2000). Other important indicators are Gross National Product (GNP), National Income and Disposable Income (DI). The GDP evaluates all produced goods and services. It can be computed in three possible ways. From a methodological point of view they are equivalent:

- Origination Account by all sectors of the economy,
- Application Account of all produced goods and services and
- Distribution Account of all components of income.

In Germany the GDP is computed only with the Origination Account and the Application Account. The Distribution Account is not applicable since the information on profits of companies is not sufficiently available (Statistisches Bundesamt 2006). In the United States the Origination Account is not used for the computation of the GDP.

Another way to represent the measures of national income and output is given in Fig. 2.16, where a graph is used. This is analogous to the depiction of the DuPont model as in Section 2.1.2 and 2.2.1. Here the operators are only of type addition. The variables are the indicators of the economic system. It can be easily seen, that the GDP can be resolved on different ways. Chapter 3 explains how the information on more than one equation can be exploited.

The ways in which the GDP can be computed are explained in the following. Usually, available data determines the methods used for computation of GDP. If it is possible to compute the GDP in at least two ways, the values of GDP have to be made consistent with each other.

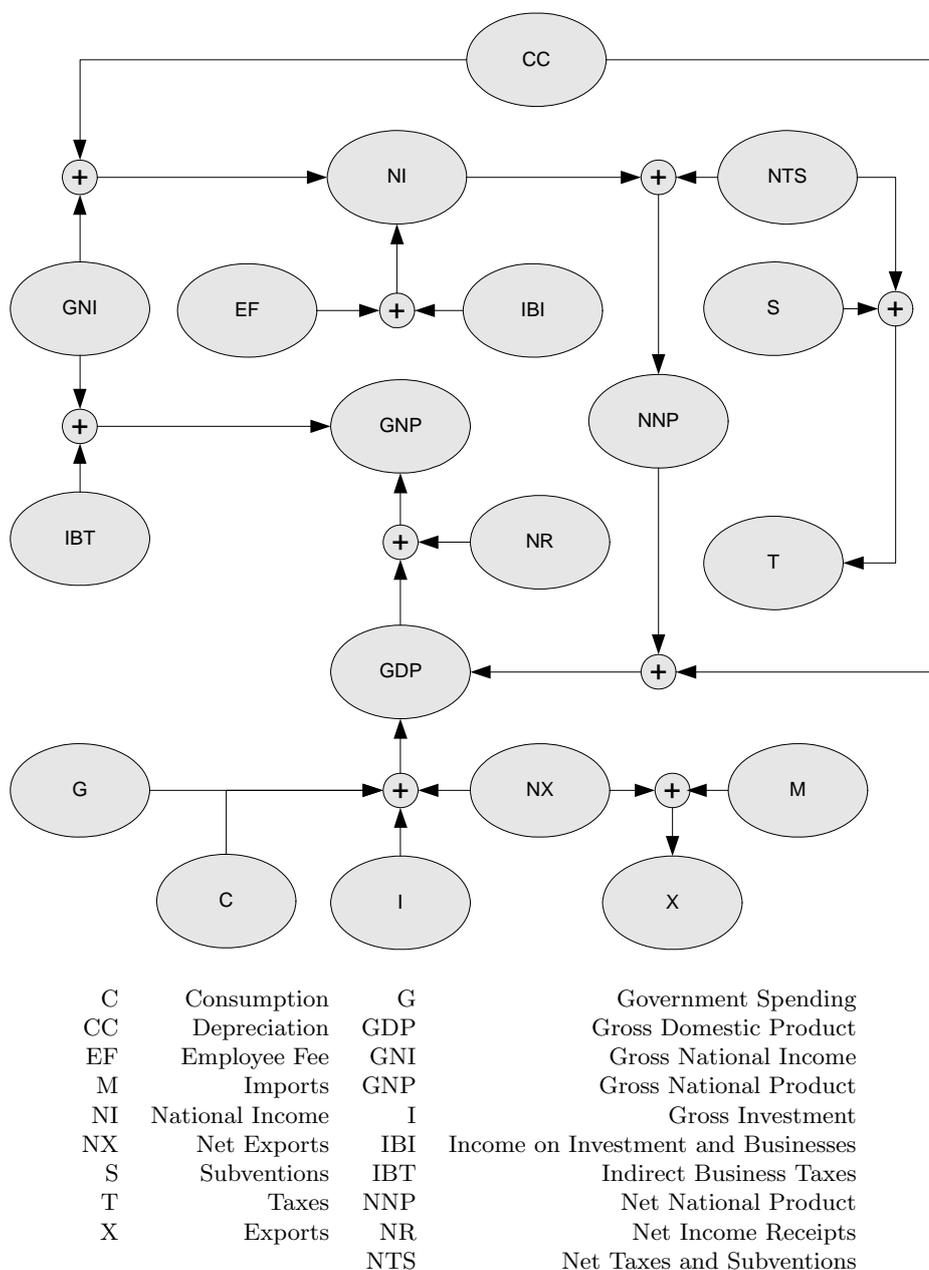


Figure 2.16: Model Graph of “Measures of National Income and Output”

All indicators are evaluated at fair market values because they reflect the relative value of goods. In a good running economy the fair market values reflect the relative satisfaction that the consumers gain from the goods. The fair market value is the result of fair market prices times the sold quantity of products.

Origination Account

In the Origination Account the GDP resolves from the output value minus the payments in advance. This intermediate result is called gross value added. The GDP is the result of adding taxes on goods (indirect taxes) and subtraction of subventions. The origination account can be depicted as follows:

+	output value	
-	payments in advance	
<hr style="width: 50%; margin: 0 auto;"/>		
=	gross value added	
+	indirect taxes	
-	subventions	
<hr style="width: 50%; margin: 0 auto;"/>		
=	gross domestic product	

Application Account

This approach is also called flow of goods approach. The application account uses private consumption, business investments in capital in the economy, government expenditures on final goods and services as well as imports and exports of the economy. This account is often used in defining the GDP. Symbols for the variable names like private consumption (C), private investment (I), public spending (G) and net exports (NX) are common. Private consumption can easily be calculated because it is restricted to the end products that the consumers buy or consume. However, private investments are the spending of households that are used for company assets, bonds or other capital investments. Public spending are the expenditures that the government expend. This can be subventions or transfers but also public investments. The net exports are the difference between the exports (X) into and the imports (M) from other countries.

+	consumption	C
+	investment	I
+	public spending	G
+	exports	X
-	imports	M
<hr style="width: 50%; margin: 0 auto;"/>		
=	gross domestic product	Y

Distribution Account

The Distribution account is also called flow of income and cost approach. Here the cost of business activities are used. The cost can be salaries and wages, economic rent and capital profit. Starting point in this account is national income which is generated by production of goods and services by the citizens. The national income is the sum of fee of employees and the capital income. In order to obtain the net national income from the national income, taxes on manufacturing goods and taxes on imports have to be added and subventions have to be

subtracted. Adding depreciation as the consumption of fixed assets delivers the GDP.

$$\begin{array}{rcl}
 & + & \text{employee fee} \\
 + & \text{income on investments and businesses} & \\
 \hline
 = & & \text{national income} \\
 + & \text{taxes on imports and manufacturing} & \\
 - & & \text{subventions} \\
 \hline
 = & & \text{net national income} \\
 + & \text{consumption of fixed capital} & \\
 \hline
 = & & \text{gross domestic product}
 \end{array}$$

The GDP reflects all produced goods and services in a country in a fixed time period. Usually, this time period is a year. However, the GDP is not the only indicator of interest. In Fig. 2.16, further important indicators like $GNP = GDP + NR$ are given.

The Gross National Product (GNP) represents the total value of all end products and services produced by the factors of production within the country. Only sold elements are included. Other indicators are Gross National Income (GNI), National Income (NI) and Net National Product (NNP). The GDP was developed in the 1940s to check if the American economy can be sustained during World War II. Criticism on the system of Measures of National Income and Output can be found in Jesinghaus (2000).

2.4.2 United Nations System of National Accounts

The United Nations System of National Accounts (SNA) (another abbreviation is UNSNA) was initially established in 1953. It was revised in 1968 and 1993 and will be revised once again in 2008. The system “is a conceptual framework that sets the international statistical standard for the measurement of the market economy“ (UN Statistics Division 1993).

Carson and Honsa (1990) describe the system and emphasise that it creates a comprehensive, integrated picture of the economy. The system integrates financial and nonfinancial stock and flow accounts of an economy. A comparison between other countries or economies that use this system is more feasible and therefore an improvement for all users of the system. The main accounts of the SNA are the following:

- The production account inherits components of the gross output of an economy.
- Incomes generated by production are consolidated in the primary distribution of income account.
- Transfers including the social spending are combined in a separate account.

- The expenditures of households build another account.
- The capital account records the values of the non-financial assets and shows the change in net worth due to saving and capital transfers.
- The Financial transactions account involves all financial assets and liabilities.
- Changes in the asset values account are concerned about results of flows that are not transactions.
- The Balance Sheet provides an overview of assets and liabilities at a certain time.
- The external transactions account captures transactions between resident institutional sectors of the total economy and transactions with non-resident units in the rest of the world.

Two main aims are connected with the SNA system. On the one hand, it facilitates comparisons between countries. On the other hand, it is a guide for countries in developing their own economic accounting systems.

Transactors and transactions are grouped to enable analysis, forecasting and policy. Transactors are governments, enterprises, and households. Further transactors are either treated separately or combined with others. Production, income and outlay, capital accumulation and finance build up transactions in the SNA accounts. The SNA is more fully integrated and more comprehensive in coverage compared to the U.S. economic accounts, cf. Carson and Honsa (1990).

2.4.3 Petroleum Economic Indicators

This section presents an economic indicator system that is used in the field of oil or gas exploration. The system is used by a Petro-Canada company. It can be built up on annual, quarterly or monthly data and be used for decision support like what-if analysis. The considered model of main petroleum indicators consists of 12 attributes that are connected with 8 equations. Related models can be found in Kemp and Kasim (2003), Moroney and Berg (1999).

The model graph is shown in Fig. 2.17. For example, *Profit* depends upon *Revenue* and *Royalty*.

Variables in Fig. 2.17 are of two types. If data for variables is available these variables have a darker colour. The equation system consists of balance, behaviour, as well as definition equations. Information about variables upon which no prior knowledge is available can be computed. These variables are light-coloured in Fig. 2.17.

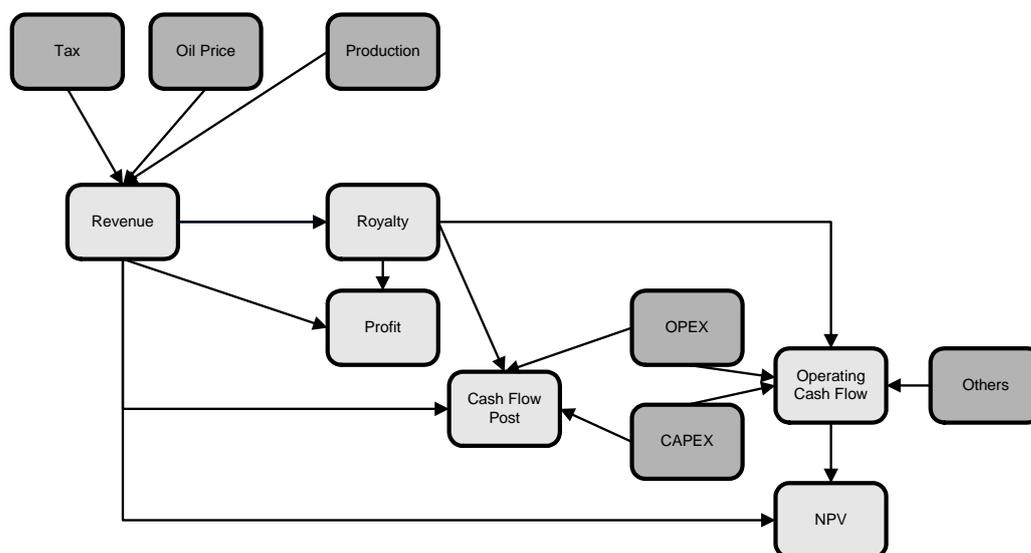


Figure 2.17: EcoOil Model

adapted from Kemp and Kasim (2003) and Moroney and Berg (1999)

$$\begin{aligned}
 \text{Capital Expenditures} &= \text{Total Exploration Cost} + \text{Total Development Cost} \\
 \text{Operating Expenditures} &= \text{Fixed Operating Cost} + \text{Well Repairs} + \text{Administrative} \\
 &\quad \text{Operating Expenditures} + \text{other Operating Expenditures} \\
 \text{Revenue} &= 0.4 \cdot \text{Production} \cdot (\text{Oil Price} - \text{Tax}) \\
 \text{Royalty} &= 0.125 \cdot \text{Revenue} \\
 \text{Operating Cash Flow} &= 0.725 \cdot (\text{Revenue} - \text{Royalty}) - 0.4 \cdot (\text{Capital Expenditures} \\
 &\quad + \text{Operating Expenditures}) - \text{Contractor Tariff} \\
 \text{Cash Flow Post} &= 0.725 \cdot (\text{Revenue} - \text{Royalty}) - 0.4 \cdot (\text{Capital} \\
 &\quad \text{Expenditures} + \text{Operating Expenditures}) \\
 \text{Profit} &= 0.275 \cdot (\text{Revenue} - \text{Royalty}) \\
 \text{Net Present Value} &= 0.91 \cdot \text{Operating Cash Flow}
 \end{aligned}$$

In this model no longitudinal interactions of the variables are assumed and therefore it is sufficient to view only one period at a time. With the help of this indicator system a realistic decision upon oil and gas explorations and production can be easily achieved.

2.5 Challenges for Indicator Systems

Two different effects occur by using equation systems and data. On the one hand, new information can be generated, e. g., aggregated information for controlling. On the other hand, it can be checked whether the data fits to the model or not. If the equation system consists of definitions or balance equations, the model can be assumed as valid. In this case, a validation of a data set is possible.

Data quality validation plays an important role. When data is collected, measurement errors may happen. Moreover, the aggregated data has due to error propagation a stochastic character as well. Therefore, the corresponding variables can be considered stochastic.

Using crisp data instead of stochastic data generates two major problems. For one, it is necessary to enforce that the collected crisp numbers reflect the indicator states as good as possible. Second, the underlying model should not be violated by the data. In the ideal case both problems do not occur. However, measurement intervals instead of real values, imprecise instead of precise data, or even fraud instead of honesty reflect reality better. For that reason aggregations of complex systems should be used as uncertain indicators.

Uncertainty can be expressed in several ways and uncertainty about numbers is, for instance, possible by a description as random variables or by the use of Fuzzy sets or numbers. By using such representations the validation checks become more complex. Therefore new methods or algorithms have to be used for handling imprecise data within indicator systems. Another goal besides data validation should be the reduction of uncertainty of variables.

In the context of a full information the problem of inconsistent data does not occur. However, in reality only partial information is available. The more complex the indicator system is the more likely the information is stud. Partial information may lead to different estimates or computation for the same indicator. This results in inconsistencies. Therefore a method for merging the partial information and to resolve inconsistent data is required.

Methods will only be used in practice if the usage is user-friendly: the method should be implemented in tools that facilitate its usage and results must be delivered “in-time”.

Chapter 3

Methodology of Handling Imprecise Data

This chapter presents the methodology for handling imprecise data. Both, the probabilistic and the possibilistic approach are considered. The probabilistic approach uses a statistical model. An analytical solution for statistical models is presented. Due to restrictions of the approach with respect to the probability distribution family involved, simulation is as an alternative for solving equation systems with error in the variables. The special characteristics of simulation are explored. Fuzzy set theory is another methodology to deal with imprecise data and is considered, too.

3.1 The Possibilistic Approach

This section presents the possibilistic approach. Fuzzy set theory is used to describe imprecise data. Zadeh's extension principle is the basis for operations with Fuzzy sets (Zadeh 1965, 1975a,b,c).

Objects and elements can be grouped in sets. Same property values of interest group elements together in one set, different property values partition elements into different groups. In classical set theory an element can be a member of a group or not. This can be expressed by the characteristic function μ .

Definition 3.1.1 (Characteristic Function). A characteristic function μ_A of a set $A \subseteq X$ maps all elements of a basic set X as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{else.} \end{cases} \quad (3.1.1)$$

3.1.1 Fuzzy sets

Fuzzy set theory goes back to Zadeh (1965). In contrast to classical set theory the codomain is not restricted to $\{0; 1\}$, but to the interval $[0, 1]$. The characteristic function is called membership function.

Definition 3.1.2 (Membership Function). A Fuzzy set A on X is defined by the pair $(x, \mu_A(x))$ and

$$A = \{(x, \mu_A(x)) | x \in X, \mu_A(x) \in \mathbb{R} \wedge \mu_A(x) \in [0, 1]\}. \quad (3.1.2)$$

The properties of a membership function $\mu_A(x)$ are:

- $\mu_A(x) \geq 0 \forall x \in X$.
- The range of $\mu_A(x)$ should be normalised so that $\sup_{x \in X} [\mu_A(x)] = 1$.

The support $\text{supp}(A)$ of a Fuzzy set A is a set of all elements x where $\mu_A(x) > 0$. It is essential that

$$\text{supp}(A) = \{x \in X | \mu_A(x) > 0\} \text{ and } \text{supp}(A) \subseteq X. \quad (3.1.3)$$

If $\text{supp}(A)$ consists of exactly one element x_1 with $\mu_A(x_1) = 1$ x_1 is called a crisp value or a singleton. In Fig. 3.1 a Fuzzy set A with a triangular membership function is shown. Other forms of membership function are possible. A crisp data set B is also given in the figure.

The empty set \emptyset can be defined $\forall x \in X : \mu_{\emptyset}(x) = 0$. The identity of two Fuzzy sets A and B with the same basic set X is given iff $\forall x \in X \mu_A(x) = \mu_B(x)$.

The α -cut of a Fuzzy set A_α is defined as:

$$A_\alpha = \{x \in X | \mu_A(x) > \alpha\} \text{ with } 0 \leq \alpha \leq 1. \quad (3.1.4)$$

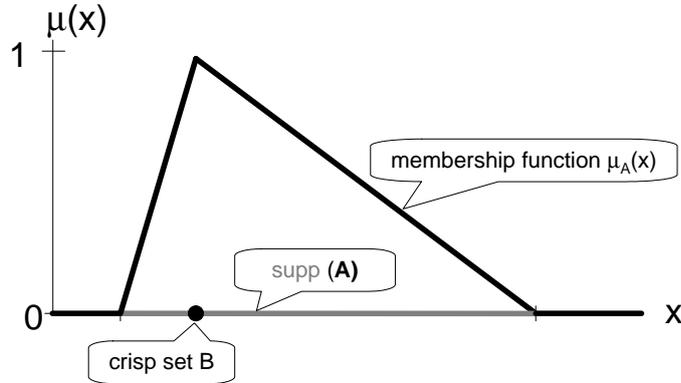


Figure 3.1: A Fuzzy Set A with Triangular Membership Function and a crisp set B

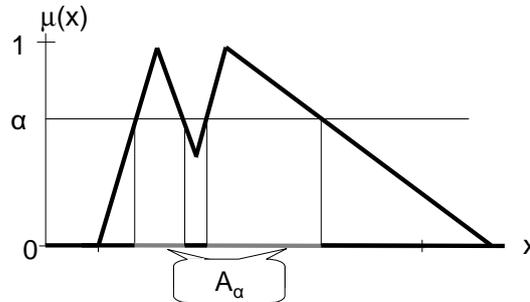


Figure 3.2: An α -cut Fuzzy Set

It delivers all elements that have a membership function value greater than α .

Fig. 3.2 shows the α -cut A_α of a Fuzzy set. Here the α -cut consists of two intervals.

A Fuzzy set A is a subset of a Fuzzy set B ($A \subseteq B$) if for all $x \in X$ $\mu_A(x) \leq \mu_B(x)$. A is called a proper subset of B ($A \subset B$) if for all $x \in X$ $\mu_A(x) < \mu_B(x)$.

It is essential that (Rommelfanger 1994, Zimmermann 1991):

$$\begin{aligned}
 A &\subseteq A \\
 A \subseteq B \wedge B \subseteq A &\Leftrightarrow A = B \\
 A \subseteq B \wedge B \subseteq C &\Leftrightarrow A \subseteq C \\
 A \subseteq B &\Rightarrow \text{supp}(A) \subseteq \text{supp}(B) \\
 A \subseteq B &\Rightarrow A_\alpha \subseteq B_\alpha
 \end{aligned}$$

Definition 3.1.3 (Convex Fuzzy Set). A Fuzzy set A with a membership function $\mu_A(x)$ is called convex if

$$\mu_A(c) \geq \min\{\mu_A(a), \mu_A(b)\} \quad \forall a, b, c \in X \text{ and } a \leq c \leq b.$$

A Fuzzy set is convex iff all α cuts are connected (Bothe 1995, p:35). Fig. 3.1 shows a convex Fuzzy set, whereas in Fig. 3.2 a non-convex Fuzzy set is depicted.

Definition 3.1.4 (Cardinality of Fuzzy Set). The cardinality of a Fuzzy set A is analogous defined to cardinality of a set:

$$|A| = \sum_{x \in X} \mu_A(x) \quad \text{with } X \text{ finite.}$$

3.1.2 Operations on Fuzzy Sets

In the previous section Fuzzy sets were introduced. In this section operations on Fuzzy sets are presented. First, logical operations like negation, union and intersection are defined. This is followed by Zadeh's Extension principle. The last part of this section deals with Fuzzy arithmetic.

Operations on Fuzzy Sets

The most important logical operators are *not*, *and* and *or*. They are used to connect information. An adaption of these operators to Fuzzy sets with complement, union and intersection is possible (Bothe 1995, p. 25).

The creation of complement means a negation of the statement corresponding to a set. The complement of the statement "the apple is green" is "the apple is not green".

Definition 3.1.5 (Complement of a Fuzzy Set). The complement A^C of a Fuzzy set A on the basic set X with the corresponding membership functions μ_{A^C} and μ_A is defined as:

$$\mu_{A^C}(x) = 1 - \mu_A(x) \quad \forall x \in X.$$

Fig. 3.3 shows a Fuzzy set A and its complement A^C .

The following statements can be drawn from Def. 3.1.5:

$$\begin{aligned} [A^C]^C &= A \\ A \subseteq B &\Leftrightarrow B^C \subseteq A^C. \end{aligned}$$

Whereas for the union and for the intersection of sets different but equivalent functional descriptions are available, Fuzzy set theory according to Zadeh (1965) uses *min*-criterion for union and a *max*-criterion for the intersection.

Definition 3.1.6 (Union of Fuzzy Sets). Given two Fuzzy sets A and B with their corresponding membership functions μ_A and μ_B the union $A \cup B$ is given by:

$$\mu_{A \cup B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad \forall x \in X.$$

The union of a Fuzzy set A and B is given in Fig. 3.4.

Definition 3.1.7 (Intersection of Fuzzy Sets). For two Fuzzy sets A and B with their corresponding membership functions μ_A and μ_B the intersection $A \cap B$ is defined as:

$$\mu_{A \cap B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad \forall x \in X.$$

The intersection of a Fuzzy set A and B is given in Fig. 3.6.

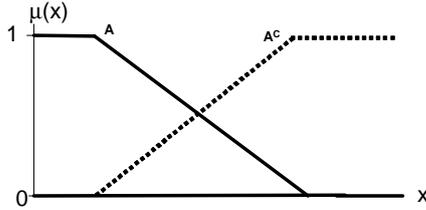


Figure 3.3: A Fuzzy Set A with Complement A^C

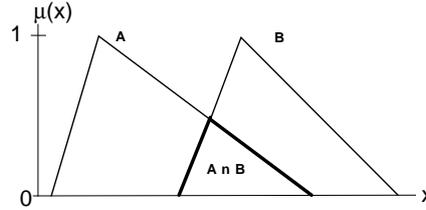


Figure 3.4: Union of Fuzzy Set A and B

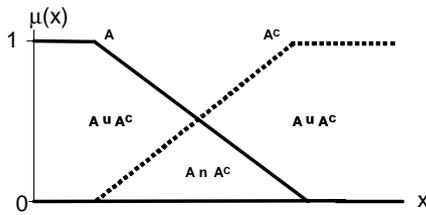


Figure 3.5: Union and Intersection of Fuzzy Set A and Complement A^C

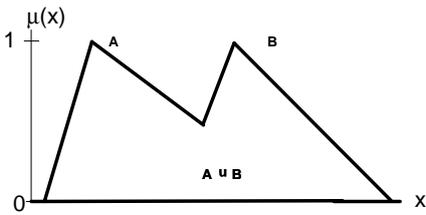


Figure 3.6: Union of Fuzzy Set A and B

For union and intersection of Fuzzy sets the following laws exist (Rommelfanger 1994, p. 20):

commutative law	$A \cap B = B \cap A$
	$A \cup B = B \cup A$
associative law	$(A \cap B) \cap C = A \cap (B \cap C)$
	$(A \cup B) \cup C = A \cup (B \cup C)$
distributive law	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
absorption law	$A \cap (B \cup A) = A$
	$A \cup (B \cap A) = A$
De Morgan's theorem	$[A \cap B]^C = A^C \cup B^C$
	$[A \cup B]^C = A^C \cap B^C$

The law of complements does not hold for union and intersection of of Fuzzy sets. This can be seen in Fig. 3.5. This means that the following statements hold:

$$A \cap A^C \neq \emptyset$$

$$A \cup A^C \neq \text{basic set } X.$$

Extension Principle

Before Zadeh's extension principle is explained, it is necessary to define Fuzzy numbers.

Definition 3.1.8 (Fuzzy Number). A convex Fuzzy set A is called Fuzzy number if (Bothe 1995, p. 58):

- basic set $X = \mathbb{R}$
- $\exists x \in X : \mu_A(x) = 1$ and
- $\mu_A(x)$ is piecewise continuous.

If there exists exactly one $a \in X$ so that $\mu_A(a) = 1$ then a is called mean value of the Fuzzy number. A nonempty Fuzzy number is called positive if:

$\forall x < 0 : \mu_A(x) = 0$. In an analogous way a negative Fuzzy number can be defined by $\forall x > 0 : \mu_A(x) = 0$.

In order to obtain calculation rules for Fuzzy numbers, the Extension principle (Zadeh 1965, 1975a,b,c) is used.

Definition 3.1.9 (Extension Principle). Let f be a function $f : X \rightarrow Y$ with basic sets X, Y and A is a Fuzzy set defined on X . The extension of f is Fuzzy set B with $B = f(A)$ where:

$$\mu_B(y) = \begin{cases} \sup\{\mu_A(x); y = f(x), x \in X\} \\ 0 \end{cases} \quad \text{if } \nexists x \in X \text{ with } f(x) = y. \quad (3.1.5)$$

Mathematical concepts of crisp numbers are transferred to Fuzzy numbers. The foundations of Fuzzy set by set theory enables the definition of functions on Fuzzy numbers. In the beginning, the cartesian product of Fuzzy sets is defined.

Definition 3.1.10 (Cartesian Product of Fuzzy Sets). Let be A_1, \dots, A_n Fuzzy sets with the basic sets X_1, \dots, X_n . The cartesian product $A_1 \times \dots \times A_n$ is a Fuzzy set with membership function $\mu_c(x_1, \dots, x_n)$ on the basic set $X_c = X_1 \times \dots \times X_n$ where

$$\mu_c(x_1, \dots, x_n) = \min\{\mu_{A_i}(x_i) \mid x_i \in X_i; i = 1, \dots, n\}.$$

With Def. 3.1.10 and help of the extension principle (Def. 3.1.9) the following calculation rules can be derived:

Suppose there are:

- n Fuzzy sets A_i with basic sets X_1, \dots, X_n and membership functions $\mu_1(x), \dots, \mu_n(x)$,
- a mapping function $f : X_1 \times \dots \times X_n \rightarrow Y$ where $y = f(A_1, \dots, A_n)$ and $y \in Y$.

The Fuzzy set Z is then given by the mapping function $f(x_1, \dots, x_n)$ on Y by:

$$Z = \{(y; \mu_Z(y)) \mid y = f(x_1, \dots, x_n), (x_1, \dots, x_n) \in X_1 \times \dots \times X_n\}$$

whereas the corresponding membership function evaluates as:

$$\mu_Z(y) = \begin{cases} \sup_{y=f(x_1, \dots, x_n)} [\min\{\mu_1(x_1), \dots, \mu_n(x_n)\}] & \forall y \in Y \\ 0 & \text{otherwise.} \end{cases}$$

According to the case of adding two Fuzzy numbers subtraction, multiplication and division of Fuzzy numbers can be calculated.

Fuzzy Basic Arithmetic Operations

In Kruse et al. (1995), it is shown that the basic arithmetic operations \oplus , \ominus , \odot , and \oslash can be realised by combining the intervals of α -cuts. Thus, the operations can be reduced to interval arithmetic.

The following interval operations are applicable (Kruse et al. 1995, pp. 38):

$\forall a, b, c, d \in \mathbb{R} :$

$$[a, b] + [c, d] = [a + c, b + d]$$

$$[a, b] - [c, d] = [a - d, b - c]$$

$$[a, b] \cdot [c, d] = \begin{cases} [ac, bd] & a \geq 0 \wedge c \geq 0 \\ [bd, ac] & b < 0 \wedge d < 0 \\ [\min\{ad, bc\}, \max\{ad, bc\}] & ab \geq 0 \wedge cd \geq 0 \wedge ac < 0 \\ [\min\{ad, bc\}, \max\{ac, bd\}] & ab < 0 \vee cd < 0 \end{cases}$$

$$\frac{1}{[a, b]} = \begin{cases} [\frac{1}{b}, \frac{1}{a}] & 0 \notin [a, b] \\ [\frac{1}{b}, \infty) \cup (-\infty, \frac{1}{a}] & a < 0 \wedge b > 0 \\ [\frac{1}{b}, \infty) & a = 0 \wedge b > 0 \\ (-\infty, \frac{1}{a}] & a < 0 \wedge b = 0 \end{cases}$$

3.1.3 FuzzyCalc

The FuzzyCalc algorithm was developed as an Excel Plug-In (Müller (1999), Lenz and Müller (2000) and Müller et al. (2003)). The main idea is the analysis of Fuzzy data related to an equation system. In case of conflicting data two scenarios are possible:

- strong model inconsistency of the data, or
- weak model inconsistency, where an adjustment of data is possible to solve the conflict.

Fuzzy set theory is used to solve an algebraic equation system given expert knowledge about data described by Fuzzy numbers. The FuzzyCalc[®] algorithm is used to solve a Fuzzy equation system. Algorithm 1 shows how the FuzzyCalc algorithm works.

Algorithm 1 Fuzzy Calc Algorithm

Input: one observation per Fuzzy variable (number),
 equation system,
 fully specified membership functions (missing values allowed)

Output: adjusted Fuzzy sets (variables)

- 1: **repeat**
- 2: **for** all equations **do**
- 3: **for** all variables in current equation **do**
- 4: **Resolve** (separation step) equation for each variable
- 5: **Compute** Fuzzy set of variable using Zadeh’s extension principle
- 6: **end for**
- 7: **end for**
- 8: **for** all variables **do**
- 9: **Compute** the intersection of Fuzzy sets of variable for all related equations
- 10: **if** intersection is empty **then**
- 11: **notify** user: “M-inconsistent!”
- 12: **else**
- 13: **re-normalise** the corresponding membership function
- 14: **end if**
- 15: **end for**
- 16: **until** (M-inconsistency notification) **or** (no corrections of any Fuzzy sets occur)

The algorithm works as follows: In the first part, the equation system is separated for each variable and alternative values are calculated. For the calculation of alternatives Fuzzy arithmetic is used. In the second part of the algorithm, the intersection between all alternative values for each variable and the prior value (Fuzzy number of the related variable) is calculated. If this intersection is empty, the algorithm stops. Otherwise the value of the intersection is re-normalised and a new iteration begins. The looping stops if no further changes in any variable occur.

Interpreting a membership function with more than one peak is not trivial. Therefore only convex membership functions are considered as input. FuzzyCalc has a number of useful properties. Several of these properties are identical with properties of the general-least-squares estimation under a Gaussian regime. These properties are:

- supports have a monotone contraction, when Fuzzy variables are arithmetically combined,

- the peaks of adjusted Fuzzy variables fulfill the equation system,
- invariance of adjusted data,
- shift of peak positions (values) depend on length of support, and
- shift depends on support.

3.2 The Probabilistic Approach

In the probabilistic approach data are described by density or distribution functions.

In the next sections a selection of distributions are described. Finite parameter distributions are described by their probability distribution function (pdf). Moreover, several classes of probability (continuous) distributions are given.

3.2.1 Single-Peak Distributions

One-peak distributions have a single maximum of their density function.

The following distributions f are representatives of the class of single-peak distributions. They might also be member of the other classes as explained below.

Gaussian Distribution

The Gaussian distribution is also called normal distribution. Due to the Central Limit Theorem the sum of identically and independently distributed random variables tends to be normal distributed. In reality errors of real-valued observations are often Gaussian distributed. The standard Gaussian distribution has mean $\mu = 0$ and standard deviation (sd) $\sigma = 1$.

$$\text{pdf: } \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad x \in \mathbb{R}$$

Parameters: μ - location
 σ - scale

Moments: $Mean = Median = \mu$
 $Variance = \sigma^2$
 $Skewness = Kurtosis = 0$

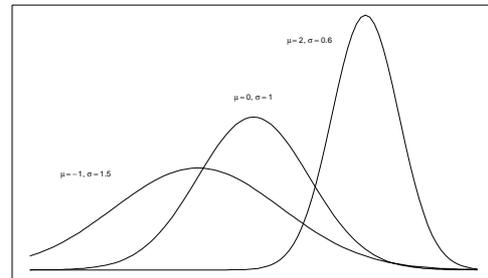


Figure 3.7: Gaussian Distribution

Bivariate Gaussian Distribution

The Bivariate Gaussian distribution is a special case of the multivariate Gaussian distribution. It can be seen as a generalisation of the Gaussian distribution. To stress that X is bivariate distributed, the notation $X \sim N_2(\mu, \Sigma)$ is used.

pdf: $\frac{1}{2\pi|\Sigma|^{1/2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \quad x \in \mathbb{R}^2$
 Parameters: $\mu = (\mu_1, \mu_2)$ – location
 $\Sigma = E[(x - \mu)(x - \mu)^T]$ –
 Covariance matrix
 Moments: $Mean = Median = \mu$
 $Variations = \sigma_1^2, \sigma_2^2$
 $Skewness = Kurtosis = 0$

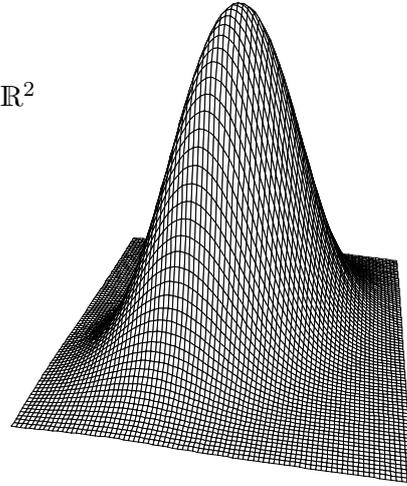


Figure 3.8: Bivariate Gaussian Distribution

Triangular Distribution

The Triangular distribution can also be interpreted as the convolution of two uniform distributions. If $X_1 \sim U(a, b)$ and $X_2 \sim U(c, d)$ and $(b - a) = (d - c)$ then $(X_1 + X_2)$ is triangular distributed. This distribution has a finite support on a bounded interval. In the case of $b = (a + c)/2$, this distribution is symmetric. The Triangular distribution is often used, if only small information about the distribution of data are available, i.e., upper and lower bound.

pdf:
$$\begin{cases} \frac{2(x-a)}{(c-a)(b-a)} & \text{if } 0 < x < b \\ \frac{2(c-x)}{(c-a)(c-b)} & \text{if } b \leq x < c \\ 0 & \text{otherwise} \end{cases}$$

Parameters: a - lower bound
 b - mode
 c - upper bound

Moments: $Mean = \frac{a+b+c}{3}$
 $Variance = \frac{a^2+b^2+c^2-ab-ac-bc}{18}$
 $Skewness = \frac{\sqrt{2}}{5} \frac{2(a^3+b^3+c^3)-3(a^2b+ab^2+a^2c+ac^2+b^2c+bc^2)+12abc}{\sqrt{(a^2+b^2+c^2-ab-ac-bc)^3}}$
 $Kurtosis = -\frac{3}{5}$

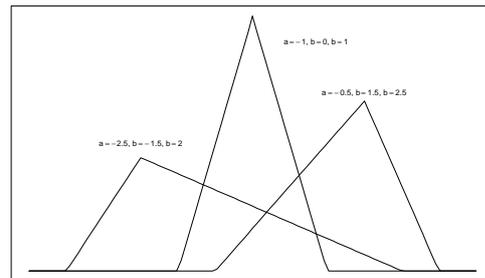


Figure 3.9: Triangular Distribution

3.2.2 Symmetrical Distributions

The class of symmetrical distributions has the property that the mean and the median have the same value. Three symmetric distributions are described.

Uniform Distribution

The Uniform distribution is also called rectangular distribution. Pseudorandom Number Generators (PRNG) generally return random numbers standard uniformly distributed values as a primary output. The standard Uniform distribution has $a = 0$ and $b = 1$.

pdf: $\frac{1}{b-a} x \in \mathbb{R}_{[a,b]}$
 Parameters: a - lower bound
 b - upper bound
 Moments: $Mean = Median = \frac{a+b}{2}$
 $Variance = \frac{1}{12}(b-a)^2$
 $Skewness = 0$
 $Kurtosis = -\frac{6}{5}$

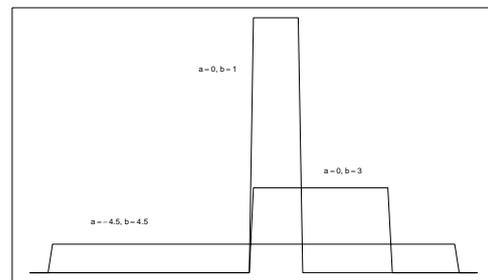


Figure 3.10: Uniform Distribution

Laplace Distribution

The Laplace distribution is often called double exponential distribution and also known as bilateral exponential distribution.

pdf: $\frac{1}{2b} e^{-\frac{|x-a|}{b}} x \in \mathbb{R}$
 Parameters: a - location
 b - scale
 Moments: $Mean = Median = a$
 $Variance = 2b^2$
 $Skewness = 0$
 $Kurtosis = 3$

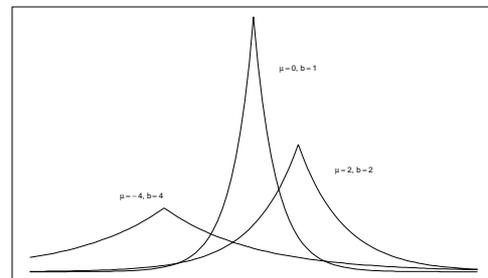


Figure 3.11: Laplace Distribution

Student's t Distribution

The Student's t distribution is also simply referred as t-distribution. It is a specialisation of the generalised hyperbolic distribution (Müller 1975). For large values (≥ 100) of degrees of freedom t-distribution approaches the Gaussian distribution. The k moment exists only if $c > k$.

pdf:
$$\frac{\Gamma(\frac{c+1}{2})}{b\sqrt{\pi c}\Gamma(\frac{c}{2})} \left[1 + \frac{(x-a)^2}{c} \right]^{-\frac{c+1}{2}} x \in \mathbb{R}$$

Parameters: a - location
 b - scale
 c - degrees of freedom

Moments: $Mean = Median = a$
 $Variance = (\frac{c}{c-2})b^2$
 $Skewness = 0$
 $Kurtosis = 3 \left[\frac{(c-2)^2\Gamma(\frac{c}{2}-2)}{4\Gamma(\frac{c}{2})} - 1 \right]$

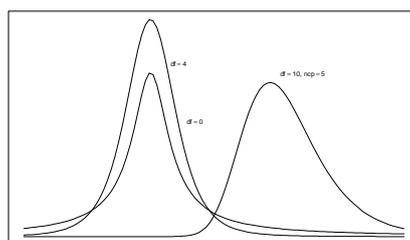


Figure 3.12: Student's T Distribution

3.2.3 Heavy Tail Distributions

A distribution is said to be heavy tailed (Crovella and Bestavros 1997) if:

$$Pr[X > x] \sim x^{-\alpha} \text{ as } x \rightarrow \infty \text{ and } 0 < \alpha < 2.$$

A density function with a hyperbolic shape is distribution with heavy tails. A characteristic of heavy tail distributions is that the log-log plot is approximately linear.

In the following the Pareto and the Cauchy distribution are briefly described as members of the class of heavy tail distributions.

Pareto Distribution

The Pareto distribution is right-skewed. It is defined on the interval $[a, \infty)$ where $a > 0$. This distribution is used to describe income, where a random income exceeds a minimum, a , is Pareto distributed. The moment k exists if $b > k$.

pdf: $\frac{ba^b}{x^{b+1}} \quad x \in \mathbb{R}_{[a,\infty)}$

Parameters: a - location
 b - shape

Moments: $Mean = \frac{ab}{b-1}$
 $Median = a\sqrt[2]{2}$
 $Variance = \frac{a^2b}{(b-2)(b-1)^2}$
 $Skewness = \frac{2(b+1)\sqrt{b-2}}{(b-3)\sqrt{b}}$
 $Kurtosis = \frac{6(b^3+b^2-6b-2)}{b(b^2-7b+12)}$

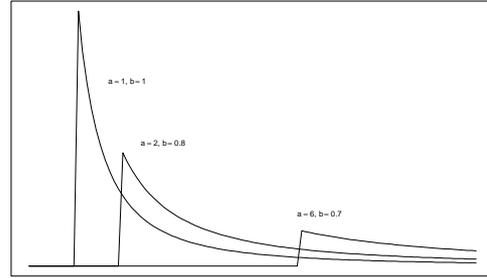


Figure 3.13: Pareto Distribution

Cauchy Distribution

The Cauchy distribution is also known as Lorentz distribution.

The Cauchy distribution has no finite moments due to the fact that the corresponding integrals do not exist. Given X_1 and X_2 as standard Gaussian distributed random numbers the X_1/X_2 is standard Cauchy distributed. The Cauchy distribution is equal to the Student's t distribution with one degree of freedom.

The standard Cauchy distribution has location 0 and scale 1.

pdf: $\frac{1}{\pi s [1 + (\frac{x-l}{s})^2]} \quad x \in \mathbb{R}$

Parameters: l - location
 s - scale

Moments: $Median = a$

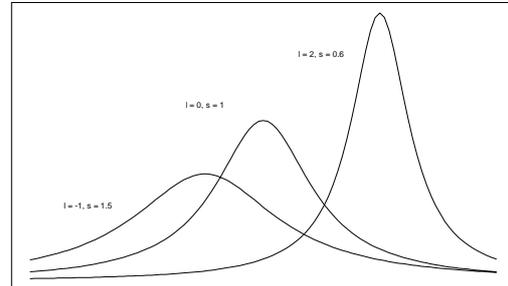


Figure 3.14: Cauchy Distribution

3.2.4 Skewed Distributions

Skewness is a measure of the asymmetry of a probability function. It is the third moment divided by the cubic standard deviation. In contrast to symmetric distributions, where a centre exists, the mode, mean and median values are different.

A right skewed or positive skewed distribution has a longer tail on the right side of the maximum or the mass of the distribution is concentrated on the left side. In the case of a left skewed distribution it is the other way around. In the following some typical skewed distributions are briefly explained.

Exponential Distribution

The Exponential distribution is sometimes called Negative Exponential distribution. It is the continuous version of the Geometric distribution. The support of this distribution is $[0, \infty)$. The Exponential distribution has the memoryless property which means that:

$$Pr(X > t + s | X > s) = Pr(X > t) \quad \forall s, t.$$

pdf: $\lambda e^{-\lambda x} \quad x \in \mathbb{R}_+$
 Parameters: λ - rate
 Moments: $Mean = \frac{1}{\lambda}$
 $Median = \frac{\ln 2}{\lambda}$
 $Variance = \frac{1}{\lambda^2}$
 $Skewness = \frac{2}{\lambda}$
 $Kurtosis = \frac{9}{\lambda^2}$

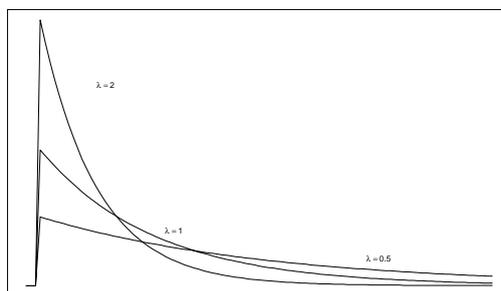


Figure 3.15: Exponential Distribution

Gamma Distribution

The Gamma distribution is a right-skewed distribution. The function is defined on the interval $[0, \infty)$. It approaches the Normal distribution if c goes to infinity. In the case that shape parameter c is an integer, the distribution is also called Erlang distribution. With $c = 1$ it is the above described Exponential distribution. Gamma distribution with parameters $a = 0, b = 2, c = d/2$ is the Chi-square distribution with d degrees of freedom.

pdf: $\frac{1}{b\Gamma(c)} \left(\frac{x-a}{b}\right)^{c-1} e^{-\frac{x-a}{b}} \quad x \in \mathbb{R}_+$
 Parameters: a - location
 b - scale
 c - shape
 Moments: $Mean = a + bc$
 $Variance = b^2c$
 $Skewness = \frac{2}{\sqrt{c}}$
 $Kurtosis = \frac{6}{c}$

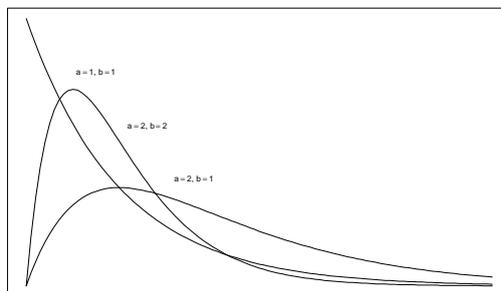


Figure 3.16: Gamma Distribution

Beta Distribution

The Beta distribution is a skewed distribution if parameters $\alpha \neq \beta$. The support of the Beta distribution is restricted to the interval $[0, 1]$. An application area is the modelling of events that take place in the interval $[\alpha, \beta]$. In the case of $\alpha = \beta = 1$, this is the standard uniform distribution. A generalisation of the Beta distribution is the Dirichlet distribution with more than two parameters.

pdf: $\frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$
 $x \in \mathbb{R}_{[0,1]}$

Parameters: α - shape
 β - shape

Moments: $Mean = \frac{\alpha}{\alpha+\beta}$
 $Variance = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
 $Skewness = \frac{2(\beta-\alpha)\sqrt{\alpha+\beta+1}}{(\alpha+\beta+2)\sqrt{\alpha\beta}}$
 $Kurtosis = 6 \frac{\alpha^3 - \alpha^2(2\beta-1) + \beta^2(\beta+1) - 2\alpha\beta(\beta+2)}{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)}$

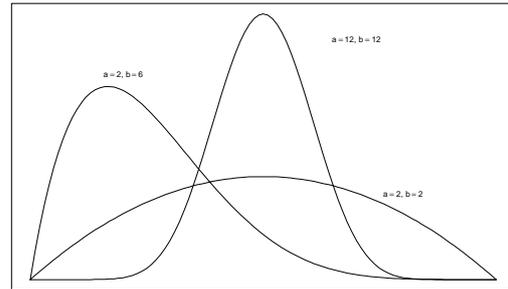


Figure 3.17: Beta Distribution

Inverse Gaussian Distribution

The Inverse Gaussian Distribution is also called Wald distribution. It is defined on the interval $[0, \infty)$. If λ goes to infinity, the distribution approaches the Gaussian distribution. While the Gaussian distribution corresponds to the distances at a fixed time in Brownian motion the Inverse Gaussian describes the time of Brownian motion to reach a fixed level.

pdf: $\sqrt{\frac{\lambda}{2\pi x^3}} e^{-\frac{\lambda}{2x} \left(\frac{x-\mu}{\mu}\right)^2} x \in \mathbb{R}$

Parameters: μ - location
 λ - scale

Moments: $Mean = \mu$
 $Variance = \frac{\mu^3}{\lambda}$
 $Skewness = 3\sqrt{\frac{\mu}{\lambda}}$
 $Kurtosis = \frac{15\mu}{\lambda}$

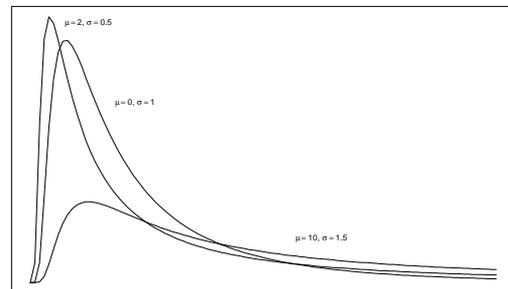


Figure 3.18: Inverse Gaussian Distribution

Multivariate Skew Normal Distribution

The Multivariate Skew Normal (MSN) distribution was developed by Azzalini and Dalla Valle (1996) and extended by Azzalini and Capitanio (1999). It is defined on \mathbb{R}^d . It is described by three parameters for location, scale and shape. If the shape parameter α is zero, then the distribution is a multivariate Normal distribution. A property of MSN distribution is that all marginal components have a skew-normal distribution. An obvious application area of this distribution is the case of empirical distributions that have a shape close to the Gaussian but lack symmetry.

$$\text{pdf: } \frac{2}{\Omega\sqrt{2\pi}} \cdot e^{\left(-\left(\frac{x-\xi}{\Omega}\right)^2/2\right)} \int_{-\infty}^{(x-\xi)/\Omega} e^{-\frac{(x-\xi)^2}{2}} dt$$

$$x \in \mathbb{R}^d$$

Parameters: ξ - location

Ω - scale

α - shape

$$\text{Moments: } \text{Mean} = \xi + \Omega\sqrt{2/\pi} \frac{\alpha}{\sqrt{1+\alpha^2}}$$

$$\text{Variance} = \Omega^2 \left(1 - 2 \frac{\alpha^2}{(1+\alpha^2)\pi}\right)$$

$$\text{Skewness} = \frac{4-\pi}{2} \frac{(2/\pi)^{3/2} \cdot \frac{\alpha^3}{(1+\alpha^2)^{3/2}}}{\left(1 - 2 \frac{\alpha^2}{(1+\alpha^2)\pi}\right)^{3/2}}$$

$$\text{Kurtosis} = 2(\pi - 3) \frac{(2/\pi)^2 \cdot \frac{\alpha^4}{(1+\alpha^2)^2}}{\left(1 - 2 \frac{\alpha^2}{(1+\alpha^2)\pi}\right)^2}$$

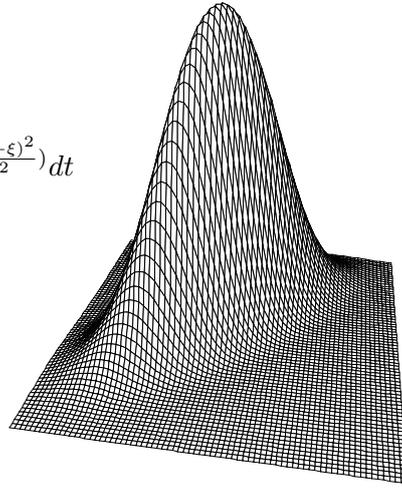


Figure 3.19: Multivariate Skew Normal Distribution

3.2.5 Contaminated Distributions

Contaminated distributions are probability functions with several modes. Such a contaminated function can be built up by a mixture of different probability functions. The first example shows a contaminated bivariate Gaussian distribution. Two other examples follow in this section.

Contaminated Bivariate Gaussian Distribution

The Contaminated Bivariate Gaussian Distribution is built up from several Bivariate Gaussian distributions. Each of these density functions effect the contaminated distribution with $0 < p_i \leq 1$ where $\sum p_i = 1$.

pdf:
$$\sum_i p_i \frac{1}{2\pi|\Sigma_i|^{1/2}} e^{(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i))}$$

 $x \in \mathbb{R}^2$

Parameters: μ_i - location parameter vector
of distribution i
 Σ_i - Covariance matrix vector
of distribution i

Moments: $Mean = Median = \sum_i p_i \mu_i$
 $Variance = \sum_i p_i^2 \Sigma_i = \Sigma$

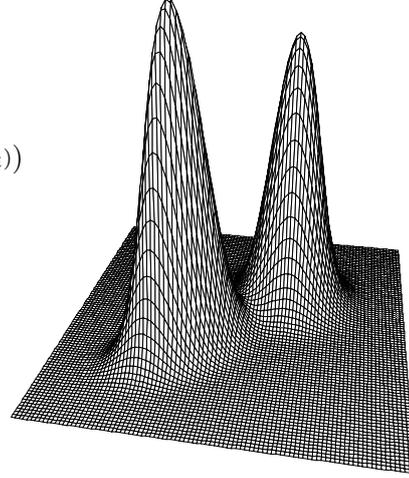


Figure 3.20: Contaminated Bivariate Gaussian Distribution

Double Gamma Distribution

The Double Gamma distribution is a bimodal distribution. It can be interpreted as a version of the Gamma distribution.

pdf:
$$\frac{1}{2b\Gamma(c)} \left| \frac{x-a}{b} \right|^{c-1} e^{-\left| \frac{x-a}{b} \right|}$$

 $x \in \mathbb{R}$

Parameters: a - location
 b - scale
 c - shape

Moments: $Mean = Median = a$
 $Variance = c(c+1)b^2$
 $Skewness = 0$

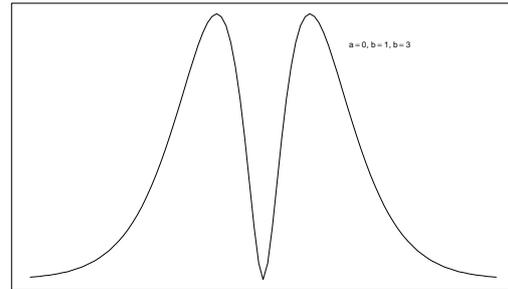


Figure 3.21: Double Gamma Distribution

Double Weibull Distribution

The Double Weibull distribution is a bimodal distribution. It can be interpreted as the signed version of the Weibull distribution.

pdf: $\frac{c}{2b} \left| \frac{x-a}{b} \right|^{c-1} e^{-\left| \frac{x-a}{b} \right|^c} x \in \mathbb{R}$
 Parameters: a - location
 b - scale
 c - shape
 Moments: $Mean = Median = a$
 $Variance = \Gamma\left(\frac{c+2}{c}\right)b^2$
 $Skewness = 0$

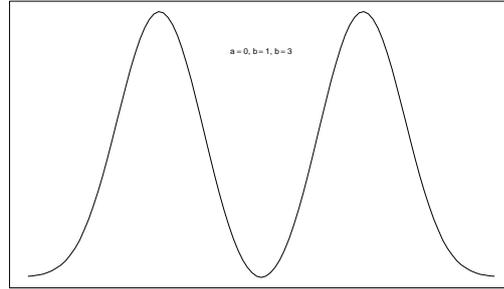


Figure 3.22: Double Weibull Distribution

3.2.6 Models with Errors in the Variables

In the following a model of structural relationships between variables is presented. Here the relationships are described by a linear model. This makes it feasible to estimate the latent variables as described by Schmid (1979) and Lenz and Rödel (1991).

The observable state variables are stated as $x = (x_1, \dots, x_p); z = (z_1, \dots, z_q)$ and the structural relationships are described with the $(q \times p)$ matrix H , which has $rank(H) = q$. The true but unobservable vector is given by ξ, ζ . The noise is given by v, w , where the expectations of noise are zero. Equation 3.2.1 represents the model with error in the variables and Fig. 3.23 gives a short overview of the dependencies.

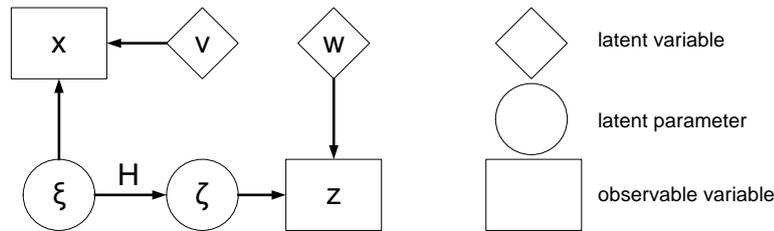


Figure 3.23: Model Graph of GLS Model

$$\begin{aligned}
 x &= \xi + v \\
 z &= \zeta + w \quad \text{where} \quad \zeta = H\xi
 \end{aligned} \tag{3.2.1}$$

$$E \left(\begin{pmatrix} v \\ w \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$Cov \left(\begin{pmatrix} v \\ w \end{pmatrix} \right) = \begin{pmatrix} P & 0 \\ 0 & R \end{pmatrix}$$

$$Rank(P) = p \quad Rank(R) = q$$

In Fig. 3.23, the observable variables are plotted with a rectangle. The unobservable (true) values are visualised by circles. Errors are indicated by a diamond. The unobservable variables can be estimated by a General Least Squares (GLS)-estimators as stated in Lenz and Rödel (1991):

$$\hat{\xi} = x + PH'(HPH' + R)^{-1}(z - Hx) \text{ with } \Sigma_{\hat{\xi}\hat{\xi}} = P - PH'(HPH' + R)HP \quad (3.2.2)$$

and

$$\hat{\zeta} = H\hat{\xi} \text{ with } \Sigma_{\hat{\zeta}\hat{\zeta}} = H\Sigma_{\hat{\xi}\hat{\xi}}H'. \quad (3.2.3)$$

The restrictions within this class of models lie in the linearity and Gaussian assumption. Indicator systems, as described in Chapter 2, usually include nonlinear relations and therefore this method is not exact.

Another approach is given in Spremann and Bamberg (1984) which deals with business analysis of information challenges. Three categories are differentiated there: (1) Retrieval and processing of information for documentation and coordination; (2) Decision based evaluation of information systems with uncertainty or risk; (3) simplification of complex data sets for better operability purposes. But again this approach is restricted to linear systems which is not satisfactory.

3.2.7 Quantor

An implementation of the probabilistic approach is the package PRTI as developed by Schmid (1988). The software tool QR, which was developed by Schmid and Müller, is an implementation of this approach. In this thesis QR version 2.03 from 1993 is used. QR is an abbreviation for Quantitative Reasoning. An introduction to the software abilities is given in Müller (1991).

The input mask is comparable to a Microsoft Excel spreadsheet. Accordingly, each variable is placed in a cell. Fig. 3.24 shows the DuPont model in QR.

Transaction Volume 100,000 ± 10,000 100,000 ± 10,000		Costs 80,000 ± 8,000 80,000 ± 8,000	Capital 80,000 ± 8,000 80,000 ± 8,000
	Profit ? 20,000 ± 12,806		Return on Investment ? 0,250 ± 0,165
Profit Margin ? 0,200 ± 0,106			Capital Turnover ? 1,250 ± 0,191

Figure 3.24: DuPont Model in QR

The variables are shown with their names (line one), their prior knowledge with a number (line two) or a question mark if unknown and the computed values in bold (line three). If all prior knowledge is Gaussian distributed, the computed

values can be interpreted as mean and standard deviation of a corresponding Gaussian distribution. Mostly, only the first two moments of unknown distributions are given and correlation is ignored.

Fig. 3.25 shows the dialog for a variable in a cell. In this case profit, P , is visualised. The relation between profit, transaction volume and cost is expressed with the relation $TV - Co$. The value of Profit is apriori unknown and thus a question mark is stated as input. After the computation, the estimated values for profit are $mean = 20$ and $deviation = 12.8$.

The screenshot shows a dialog box titled 'Zelle' with the following fields and controls:

- Name:** Profit
- Kürzel:** P
- Verknüpfung:** TV - Co
- Eingabe:** ?
- Ausgabe:** 20 ± 12,806249
- Zugehöriges Feld sichtbar** (with an upward arrow icon)
- Fernverbindung:** (empty field)
- Dezimalstellen:** 3
- Zellengrenze überschreiben**

Buttons on the right side include OK, Abbrechen, Löschen, and Notizen.

Figure 3.25: Profit within QR

In order to successfully implement a model in QR, it is helpful to visualise the model in a graph before. Three elements can be used: a basis variable which is either countable or measurable, a quotient and a linear combination element. Restrictions upon the connections of these elements are the following:

- a quotient can only be connected to exactly two basis variables,
- a linear combination has at least two basis variables as input and exactly one basis variable as output,
- other connections are not allowed.

These restrictions ensure that the implemented model follows the assumptions of the theory. In Fig. 3.26, the DuPont model is visualised.

The basis variables are depicted in a rectangular with rounded edges, a quotient in a rectangular and linear combination in a circle. With help of the model graphs, the restrictions of underlying theory may become more obvious. Large indicator systems, such as the ZVEI indicator system, are difficult or maybe even impossible to implement.

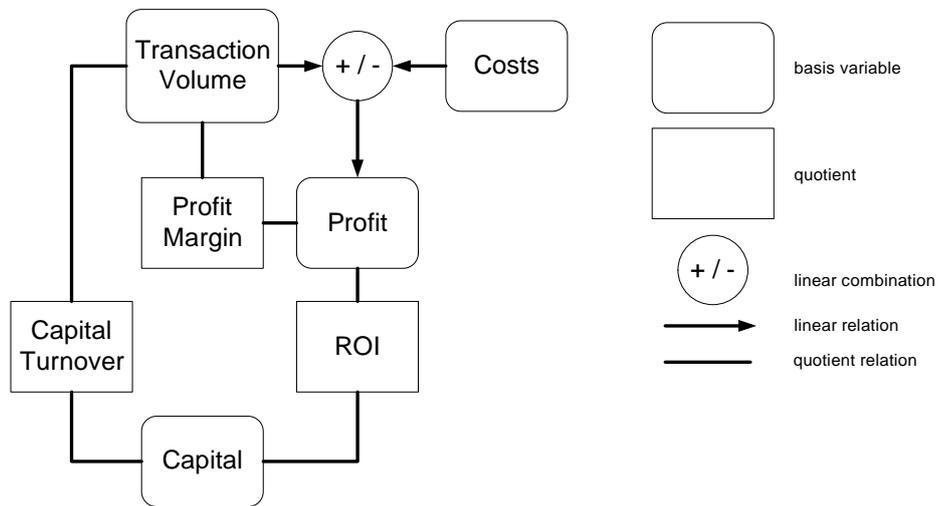


Figure 3.26: Graph of DuPont Model in QR

3.3 Simulation

3.3.1 Monte Carlo Simulation

Monte Carlo (MC) methods are a class of computational algorithms for simulating the behaviour of real systems (Metropolis and Ulam 1949). The name for this method was chosen due to the use of random numbers at the time random numbers were obtained by playing roulette. Stanislaw Ulam suggested the name because his uncle gambled the family's assets away (Metropolis 1987, p. 127).

The objective of the Monte Carlo method is to solve the following problems with the help of target function $\Pi(x)$:

I Generate a sample $\{x^{(r)}\}_{r=1}^R$ corresponding to the target function $\Pi(x)$.

II Evaluate the integral

$$\Phi = \int \phi(x)\Pi(x)d^n x \quad (3.3.1)$$

with

n - dimension of the distribution

x - (x_1, \dots, x_n)

$\phi(x)$ - function of interest that is not fully specified and known.

Indicator systems as described in Chapter 2 are one possible area of application for these methods. Monte Carlo techniques are useful in studying systems with a large number of coupled variables or for modelling phenomena with significant uncertainty in inputs. Risk calculation in business is a further application area.

Randomness occurs in almost all real situations and describes the fact that there exists a lack of predictability. By repeating a random process, the results are not ensuing a deterministic pattern. The sequence of outcomes can be represented by a probability distribution. Wolfram (2002) describes three mechanisms that are accountable for randomness:

- Randomness from the environment, i.e., the Brownian motion
- Randomness from initial conditions, i.e., convergence of heuristic search
- Randomness from system intrinsics, i.e., Cellular automaton

For randomness in science all three mechanisms can be used. Randomness from the environment can be achieved with hardware random number generator is used. In this case, a physical process generates the random numbers. One drawback of using such generators are that they are expensive. Another downside is the fact that the produced sequence has to be checked whether or not the numbers fulfil the desired probability distribution. The second mechanism can be used to generate a random number sequence, e. g., by rolling a dice. Galton (1890) described how to generate randomness by dice rolls. This method is particularly time consuming because the sequence has to be entered manually. The third mechanism is cheaper

than the first mechanism and much faster in generating random numbers than the second mechanism. This mechanism is also called pseudo randomness. “Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin.”(Neumann 1951, p. 36). Due to the better properties of generating random sequences, pseudo random numbers are widely used.

Pseudo random numbers are a sequence of numbers that appear to be random at least in a short sequence. However, they are generated by a deterministic algorithm, so called PRNG. With the same initial value of the PRNG the same sequence of numbers is generated. The randomness is interpreted via the statistical properties of the sequence like uniformly distributed or statistical independence. The problem of generating a sequence of pseudo random numbers can be split into two parts. First a uniformly distributed pseudo random number in the interval $(0, 1]$ is generated. Second, this random number is transformed so that the number follows the desired distribution.

3.3.2 Particle Filters

Particle filter is a method used for solving Monte Carlo problems. It uses a function to filter candidates. There are two possible approaches to select a filtering function: a rejection function and a weighting function. The rejection function discards some elements, whereas the weighting function boosts elements. The rejection techniques was first used by Neumann (1951), (see also Liu (2001, p. 24)). The Metropolis algorithm and the extension by Hastings as well as the Gibbs Sampler use a rejection technique. Importance sampling, sequential importance sampling and bootstrap filters use weighting techniques. All methods have in common the fact that the sample $\{x_t\}$ with $t = 1, \dots, T$ evolves over t . While importance sampling and bootstrapping are Monte Carlo techniques, Gibbs Sampler and Metropolis Hastings Algorithm use Markov Chains. Here $P(x_0)$ is called starting distribution and the transition kernel of the Markov chain is $P_M(x_t, x_{t-1})$.

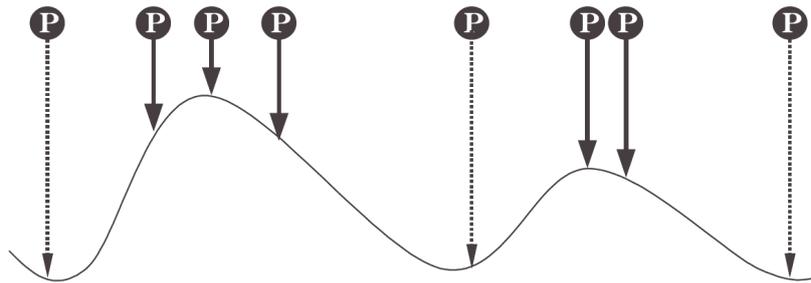


Figure 3.27: Particles and Filtering Function
rejects particles with dotted arc, accepts others

Fig. 3.27 shows the basic concept. The random numbers in the sample are called particles. The generated sample $\{x^{(r)}\}_{r=1}^R$ with R number of particles is

evaluated by the filter function $Q(x^{(r)})$. The filtering function is also known as proposal function.

In the case of a rejection function, the samples with the dotted line in Fig. 3.27 are eliminated from the sample. With Q being a weighting function the particles with the solid lines are weighted stronger than the dotted line particles.

Importance Sampling

Importance sampling estimates the expectation of the target function $\pi(x)$. No samples are generated by this method. It is rather used to reduce the computational effort (Marshall 1956). The idea behind importance sampling is to eliminate elements from the sample $\{x^{(r)}\}_{r=1}^R$ that are *over-represented* by sampling from the proposal function $q(x)$ and to add elements to the sample that are *under-represented* (MacKay 2003, p. 362). This is done by introducing weights on the elements of the sample that is generated by $q(x)$. These weights can be represented by the weighting function:

$$\omega_r(x) \equiv \frac{\pi(x^{(r)})}{q(x^{(r)})}. \quad (3.3.2)$$

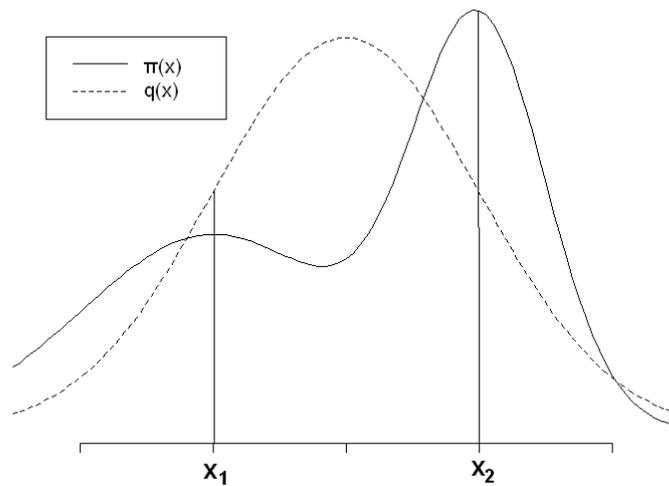


Figure 3.28: Generating Function q and Proposal Function π

Fig. 3.28 shows an example for functions q and π . Here x_1 is over-represented by q and x_2 is under-represented. By applying the weighting function as given in Equation 3.3.2, the sample is adjusted.

The estimator of the target function Π results into:

$$\tilde{\Pi}(x) \equiv \frac{\sum_r \omega_r(x) \pi(x^{(r)})}{\sum_r \omega_r(x)}.$$

A major drawback of importance sampling is the estimation of reliability of $\tilde{\Pi}$ (Doucet et al. 2001, pp. 8). The variance is dependent on the integral over x including function $\pi(x)$. Nevertheless, the strong law of large numbers holds asymptotically and $\tilde{\Pi} \xrightarrow[R \rightarrow \infty]{a.s.} \pi$.

A further problem is the estimation of the weights. For each estimation step these weights have to be computed. In case of a high-dimensional function $\pi(x)$, this is computationally expensive (Doucet et al. 2001, p. 9), see also (MacKay 2003, pp. 363).

The idea of the importance sampling algorithm as given in (Liu 2001, p. 33) can be seen in Algorithm 3.3.2. The estimation of the integral function Π of π is

Algorithm 2 Importance Sampling Algorithm

Input: $\pi()$ – target function

$q(x)$ – trial function

Output: $\tilde{\Pi}$ – estimator for Π

- 1: **for** $i := 1$ to R **do**
 - 2: Generate $x^{(i)} \sim q(\cdot)$
 - 3: **end for**
 - 4: **for** $i := 1$ to R **do**
 - 5: Compute $\omega_i(x) := \frac{\pi(x^{(i)})}{q(x^{(i)})}$
 - 6: **end for**
 - 7: Estimate $\tilde{\Pi} := \frac{\sum_r \omega_r(x) \pi(x^{(r)})}{\sum_r \omega_r(x)}$
-

done by adjusting a sample x that is drawn by the trial function q . The adjustment is achieved by weighting the samples. Drawbacks of the method are:

- q should produce samples that are in the proxi set of Π . If q is a good approximation of Π this is achieved more easily.
- The weights might vary by large factors. Again a good approximation will reduce this problem.

Concluding, the importance sampling method is only useful if a function q is available from which samples can be drawn and which is a near-perfect approximation of function Π . In a high-dimensional space it is a non-trivial task to find such a function q (Liu 2001, p. 46).

Sequential Importance Sampling

Sequential Importance Sampling is a further development of Importance Sampling. Here the estimation of Π is achieved by a recursive algorithm. A recursive Bayesian

filter is constructed. The required posterior pdf is represented as a set of random samples with respective weights. With a large number of samples it is an equivalent representation of the functional description of the pdf (cf. Arulampalam et al. (2002)).

In this method the trial function is built up sequentially. A decomposition of the sample x into (x_1, \dots, x_d) constructs the trial function q as follows:

$$\begin{aligned} q_n(x) &= q_1(x_1) \cdot q_2(x_2|x_1) \cdot q_3(x_3|x_1, x_2) \cdot \dots \cdot q_d(x_d|x_1, \dots, x_{d-1}) \\ &= q_1(x_1) \prod_{i=2}^d q_i(x_i|x_1, \dots, x_{i-1}) \\ &= q_k(x_1, \dots, x_k) \prod_{i=k+1}^d q_i(x_i|x_1, \dots, x_{i-1}). \end{aligned} \quad (3.3.3)$$

The great advantage of this approach is the reduction of the original problem to d “simpler” problems.

The target density function π can be rewritten corresponding to the decomposition of x as:

$$\pi(x) = \pi(x_1) \prod_{i=2}^d \pi(x_i|x_1, \dots, x_{i-1}). \quad (3.3.4)$$

The weights for sequential importance sampling can be constructed according to Equation 3.3.2:

$$\omega(x) = \frac{\pi(x)}{q(x)} = \frac{\pi(x_1) \prod_{i=2}^d \pi(x_i|x_1, \dots, x_{i-1})}{q(x_1) \prod_{i=2}^d q(x_i|x_1, \dots, x_{i-1})}. \quad (3.3.5)$$

The formula 3.3.5 suggests a recursive way of computing and monitoring the weights as a function at time t with $t \in [2, d]$. The weight ω_t is the product of weight ω_{t-1} and the quotient of target function and trial function at time t :

$$\omega_t(x) = \omega_{t-1}(x) \frac{\pi(x_t|x_1, \dots, x_{t-1})}{q(x_t|x_1, \dots, x_{t-1})} \quad (3.3.6)$$

The function $\omega_d(x)$ is equal $\omega(x)$ as given in Equation 3.3.5.

The algorithm for a sequential importance sampling is given in Algorithm 3.

An improvement of this method corresponding to importance sampling is that if there are no changes in the weights, the generation of new samples can be stopped. The marginal density function $\pi(x_t)$ could be used to generate the sample $x^{(d)}$. Nevertheless, in most cases it is impossible to compute the integral:

$$\Pi(x_t) = \int \pi(x) dx$$

Algorithm 3 An easy Sequential Importance Sampling Algorithm**Input:** $\pi(\cdot)$ – target function $q_i(x)$ $i = 1, \dots, d$ trial functions**Output:** $\tilde{\Pi}$ – estimator for Π

- 1: Sample $X_1 \sim q_1(\cdot)$
- 2: $\omega_1 := \frac{\pi(X_1)}{q_1(X_1)}$
- 3: **for** $i := 2$ to d **do**
- 4: Generate $X_i \sim q_i(\cdot | X_1, \dots, X_{i-1})$
- 5: $\omega_i(X_1, \dots, X_i) := \omega_{i-1}(X_1, \dots, X_{i-1}) \frac{\pi(X_i | X_1, \dots, X_{i-1})}{q_i(X_i | X_1, \dots, X_{i-1})}$
- 6: **end for**

since x is a high-dimensional vector.

Although the Algorithm 3 is quite simple, typically it cannot be implemented. This is due to the fact that $\pi(x_d | x_1, \dots, x_{d-1})$ is unknown. In order to solve this problem, a modification of the algorithm is implemented. Now the target function $\pi(\cdot)$ can be represented by a sequence of intermediate target functions $\pi_i(x_i)$. It is assumed that:

$$\pi(x) = \pi_d(x) = \pi_d(x_1, \dots, x_d). \quad (3.3.7)$$

The intermediate target functions build a sequence

$\pi_1(x_1), \pi_2(x_1, x_2), \pi_3(x_1, x_2, x_3), \dots, \pi_{d-1}(x_1, \dots, x_{d-1})$ that moves smoothly towards $\pi_d(x_1, \dots, x_d)$. At each time t an importance sampling approximation of $\pi_t(x_1, \dots, x_t)$ is done.

The importance function is the same as given in Equation 3.3.3, but now $q_t(x_t | x_1, \dots, x_{t-1})$ corresponds to $\pi_t(x_t | x_1, \dots, x_{t-1})$.

Algorithm 4 Sequential Importance Sampling Algorithm**Input:** $\pi_i(\cdot)$ $i = 1, \dots, d$ a sequence that build target function π $q_i(x)$ $i = 1, \dots, d$ trial functions**Output:** $\tilde{\Pi}$ – estimator for Π

- 1: Sample $X_1 \sim q_1(\cdot)$
- 2: $\omega_1 := \frac{\pi(X_1)}{q_1(X_1)}$
- 3: **for** $i := 2$ to d **do**
- 4: Generate $X_i \sim q_i(\cdot | X_1, \dots, X_{i-1})$
- 5: $\omega_i(X_1, \dots, X_i) := \omega_{i-1}(X_1, \dots, X_{i-1}) \frac{\pi_i(X_1, \dots, X_i)}{\pi_{i-1}(X_1, \dots, X_{i-1}) \cdot q_i(X_i | X_1, \dots, X_{i-1})}$
- 6: **end for**

Algorithm 4 produces an approximation of $\pi_t(x_1, \dots, x_t)$ at each time t :

- at time $t = 1$:

$$\omega_1(x_1) = \frac{\pi(x_1)}{q_1(x_1)}.$$

- at all other times:

$$\begin{aligned}\omega_t(x_1, \dots, x_t) &= \frac{\pi_1(x_1)}{q_1(x_1)} \prod_{i=2}^t \frac{\pi(x_1, \dots, x_i)}{\pi_{i-1}(x_1, \dots, x_{i-1})q_i(x_i|x_1, \dots, x_{i-1})} \\ &= \frac{\pi_t(x_1, \dots, x_t)}{q_1(x_1)q_2(x_2|x_1) \cdots q_t(x_t|x_1, \dots, x_{t-1})} \\ &= \frac{\pi_t(x_1, \dots, x_t)}{q_t(x_1, \dots, x_t)}.\end{aligned}$$

Different approaches exist for multitude of applications. Arulampalam et al. (2002) describes particle filters and the Sequential Importance Sampling (SIS) in a appropriate way. An overview of SIS in Bayesian filtering is given in Doucet et al. (2000). Applications of sequential Filtering are given in Doucet et al. (2001).

A major problem remains: how to select $\pi_t(x_1, \dots, x_t)$? A further problem is stated in the following example.

Example 3.3.1 (Weights in SIS). In this example the target function π is the exponential function with $\lambda = 1$. Standard Gaussian distribution serves as sampling function. In this case, the dimensionality is one. In Fig. 3.29, the histograms of weights at time $t = 0$ and $t = 1$ can be seen. Obviously, most of the weights are equal to zero. With increasing t this becomes more extreme. After a few steps only the minority has a weight different from zero.

More critics on SIS can be found in (Doucet et al. 2001, pp. 10).

Bootstrapping

Bootstrapping tries to overcome the problems of Importance Sampling and SIS by resampling. The most popular algorithm was proposed in Gordon et al. (1993), but goes back to suggestions of Efron (1982) and Smith and Gelfand (1992). This method is often also called particle filter. It makes use of the idea of sampling importance resampling Rubin (1987).

The basic idea behind the approach is that samples with a very low weight ω are disregarded. Instead, samples with a high weight are included in the sample several times. Fig. 3.30 illustrates the basic idea.

First, the particles $x^{(1)}, \dots, x^{(8)}$ are generated by the proposal function. In the next step, the weights $\omega^{(1)}, \dots, \omega^{(8)}$ corresponding to the particles are computed. Particles with small weight reduce the sampling size effectively. While in SIS this problem cannot be eliminated, bootstrapping extends in a further step the SIS procedure. Here the weights are considered. In Fig. 3.30, $x^{(1)}$ has a small weight $\omega^{(1)}$. That is the reason why $x^{(1)}$ is eliminated from the sample. On the contrary, the weight $\omega^{(3)}$ is quite large. Therefore, an insertion of a multitude of the value $x^{(3)}$ is necessary to ensure the same sampling size. This is also called resampling step.

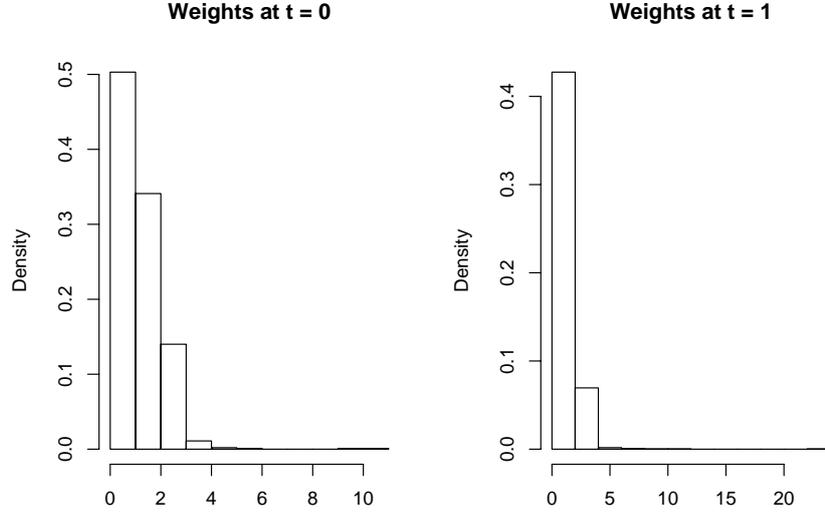


Figure 3.29: Weights in Sequential Importance Sampling at $t = 0$ and $t = 1$

The resampling step ensures that the sampling size does not converge to a single value (see Arulampalam et al. (2002)). The filter generates a sample x corresponding to the target function π . Thus, it is also possible to estimate the integral Π . The sampling procedure is an extension of the SIS procedure, where the weights are computed. On the other hand according to Eq. 3.3.5. Π can be estimated at time t as:

$$\tilde{\Pi}_t = \sum_{i=1}^d \dot{\omega}_t^{(i)} \cdot \pi_t(x_t^{(i)}) \text{ where } \dot{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{j=1}^d \omega_t^{(j)}}. \quad (3.3.8)$$

where $\dot{\omega}_t^{(i)}$ are the normalised weights according to Doucet et al. (2001, p. 8).

In order to obtain a constant sampling size at each iteration t , weights have to be normalised and must be integer values (c.f. Doucet et al. (2001, p. 8)).

On the one hand it is essential that the weights ω are displaced by integer values. On the other hand it holds that

$$\sum_{i=1}^d w_t^{(i)} = d.$$

These normalised weights are computed by the following equation:

$$\dot{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^d w_t^{(j)}} = \frac{1}{d} \cdot w_t^{(i)}. \quad (3.3.9)$$

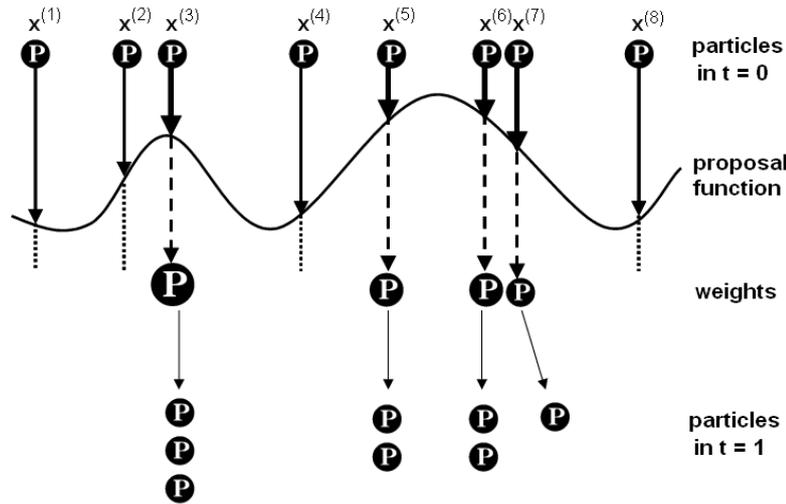


Figure 3.30: Idea Behind the Particle Filter Bootstrap

For each π_t the weights are corrected in such a way that the following relation holds:

$$\sum_{i=1}^d w_t^{(i)} \cdot \pi_t(x_t^{(i)}) \approx \sum_{i=1}^d \hat{w}_t^{(i)} \cdot \pi_t(x_t^{(i)}) \quad (3.3.10)$$

Eq. 3.3.10 ensures that particles with a weight $w = 0$ are eliminated from the sample and that particles with a weight $w > 1$ are multiple inserted into the sample. This guarantees that the sampling size at each step is d . The procedure of resampling is described in Arulampalam et al. (2002). A detailed description of the adaptation of the weights and the sample set can be found in Smith and Gelfand (1992).

The Bootstrap algorithm has the following positive properties:

- it is easily implemented,
- it is quite modular because only the expressions of importance distribution and weights have to be recoded for different problems,
- parallel implementation is possible,
- the resampling step only requires one dimensional quantities.

Example 3.3.2 (Bootstrapping AR-1 Process). The following example illustrates the bootstrap algorithm for a special autoregressive process of first order (AR-1):

$$\begin{aligned} x_t &= x_{t-1} + u_t & \text{where } u_t &\approx U(-1, 1) \\ y_t &= \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x_t - \mu)^2}{2\sigma^2}} & \text{where } \mu &= 80, \sigma = 4. \end{aligned}$$

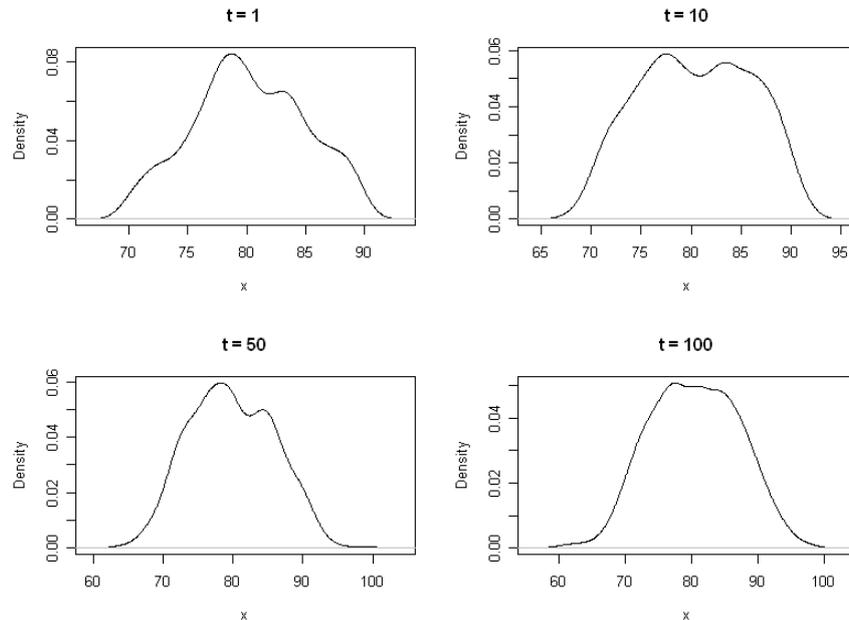


Figure 3.31: Bootstrapping AR-1 Process Starting with Uniform Distribution

In Fig. 3.31, the density estimates of function $\pi(x_t|y_t)$ are shown at the times $t = 1, 10, 50$, and 100 . The Bootstrap is used with 1000 particles and 100 iterations. The initialisation was implemented with a Uniform distribution in the interval $[70, 90]$. It can be seen that the convergence to the Gaussian distribution is slow and depends on the initial values.

In Fig. 3.32, all simulation parameters are the same as above but the initial values $x_{(0)}$ are distributed corresponding to a Gaussian distribution with mean $\mu = 80$ and sd $\sigma = 2$. Here the convergence is much faster than in the example shown in Fig. 3.31.

Another problem of the Bootstrap algorithm is the possible overestimation of the mean value. Although the convergence in Fig. 3.32 is quite fast the mean value could have a too strong weight.

In Fig. 3.31, a drawback of the bootstrap method can be illustrated: If the distribution of the initial sample differs strongly from the target density many iterations are necessary to reach the target function, i.e., $t > 100$. Otherwise, the convergence is fast, cf. $t = 10$ in Fig. 3.32. Overestimation is another drawback. The efficiency may decrease due to excessive use of resampling (Liu 2001, p. 66).

3.3.3 Markov Chain Monte Carlo Simulation

In this section, MCMC techniques are discussed. These techniques are used in order to generate samples from complex density functions where the samples are

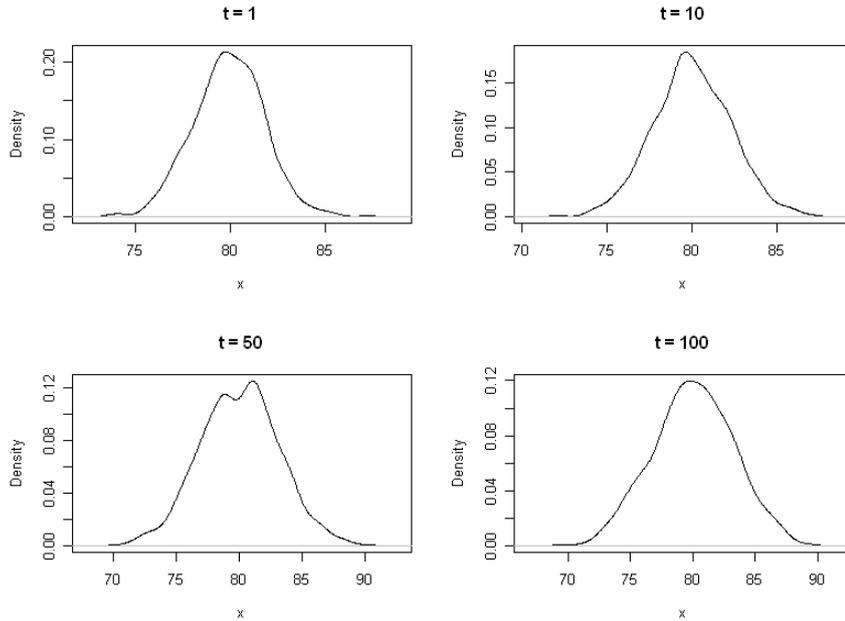


Figure 3.32: Bootstrapping AR-1 Process Starting with Gaussian Distribution

produced by evolving a Markov chain Liu (2001, p.4), see also Gilks et al. (1996a), Gamerman and Lopez (2006).

Definition 3.3.1 (Markov Chain Monte Carlo). Markov Chain Monte Carlo (MCMC) techniques are used for simulation of a distribution function f . They produce an ergodic Markov chain X_t that has a stationary distribution according to f .

Before elaborating on Markov Chain Monte Carlo (MCMC) algorithms, a short introduction to Markov chains is provided. Thereafter the most common MCMC techniques are described. They are the cornerstone for the simulation of indicator systems, which is the topic of Chapter 4.

Markov Chains

Definition 3.3.2 (Markov-Chain). A sequence of random variables X_t with state space $R = \{1, \dots, n\}$ is called Markov chain of $(t-m+1)$ th order, iff a $(n \times n)$ matrix P with $p_{ij} \geq 0 \forall i, j \in \{1, \dots, n\}$ and $\sum_{j=1}^n p_{ij} = 1 \forall i \in \{1, \dots, n\}$ exists, so that

$$\begin{aligned} Pr(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_0 = x_0) = \\ Pr(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_m = x_m) = p_{x_m x_{t+1}}. \end{aligned} \quad (3.3.11)$$

Usually first order Markov chains are used. Therefore, the Markov criterion 3.3.11 simplifies to:

$$\begin{aligned} Pr(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_0 = x_0) \\ = Pr(X_{t+1} = x_{t+1} | X_t = x_t) = p_{x_t x_{t+1}}. \end{aligned} \quad (3.3.12)$$

Discrete Markov chains describe finite states at discrete events. In the following, the term Markov chain is used for first order Markov chains. $X = (X_t)$ is discrete in t and $S = \{s_0, s_1, \dots, s_m\}$ represents a finite state space.

A probability vector ν with $\nu_i = Pr(X_0 = s_i)$ is called initial distribution. Transition probabilities are given by

$$p_{ij}(t) = Pr(X_{t+1} = s_j | X_t = s_i).$$

The transition probabilities can be pooled in the transition matrix P_M :

$$(p_{ij}(t)) = P_M(t) = \begin{pmatrix} p_{11}(t) & \dots & p_{1m}(t) \\ \vdots & \ddots & \vdots \\ p_{m1}(t) & \dots & p_{mm}(t) \end{pmatrix}.$$

If a transition probability is not defined, a zero is placed in the matrix M . If all transition probabilities are independent of t , the Markov chain is homogeneous. Otherwise the chain is called inhomogeneous. In the homogeneous case the change from state s_i to state s_j in n steps can be computed as follows:

$$p_{ij}^n = P_M^n(i, j).$$

This results in the unique determination of the distribution of X_n with M and ν :

$$Pr(X_n = i) = (P_M^n \cdot \nu)(i).$$

A distribution ν^* is called stationary distribution iff

$$\nu^* = P_M \times \nu^*.$$

It might occur, that more than one stationary distribution exists. In the case that M is the identity matrix, all initial distributions are stationary and thus all possible distributions are stationary in this degenerated extreme case. Several classes of Markov chains exhibit only one stationary distribution.

Lemma 3.3.3 (Stationary Distribution). If a homogeneous Markov chain is irreducible and aperiodic, there exists exactly one stationary distribution ν^* .

Because these Markov chains play an important role in the MCMC algorithms, a more intensive analysis of these sequences is necessary, c.f. Gilks et al. (1996a), Liu (2001), Gamerman (2002), Robert and Casella (2003).

A state s_j in a Markov chain (X_t) is accessible from state s_i , if starting in state s_i the state s_j is reached after a finite number of transitions with a probability greater than zero. This is denoted with $i \rightarrow j$. A state s_i is said to communicate with state s_j if both states are accessible from the other.

Definition 3.3.4 (Irreducibility). A Markov chain is **irreducible** iff all states in the state space are accessible from the others.

Definition 3.3.4 indicates that the probability to get from state s_i to state s_j for all $s_i, s_j \in S$ is positive.

Definition 3.3.5 (Aperiodicity). A state s_i has the period k with

$$k = \text{greatest common divisor } \{n : Pr(X_n = i | X_0 = i) > 0\}.$$

A state s_i is aperiodic if its period $k = 1$. An irreducible Markov chain is **aperiodic** if all of its states are aperiodic.

Definition 3.3.6 (Recurrence). The time T_i is the next return time to state s_i defined as

$$T_i = \min \{n : X_n = s_i | X_0 = s_i\}.$$

The state s_i is transient if T_i is not finite with some probability:

$$Pr(T_i < \infty) < 1.$$

A state is recurrent if it is not transient. If the expected return means time is positive, the state is positive recurrent. Definition 3.3.6 can also be formulated so that a state is recurrent if and only if $\sum_{n=0}^{\infty} p_{ij}^{(n)} = \infty$.

Definition 3.3.7 (Ergodicity). If a state s_i is positive recurrent and aperiodic than the state is said to be ergodic. A Markov chain with only ergodic states is said to be **ergodic**.

Based on these definitions it can be checked whether a given Markov chain has a steady state or not. These definitions are the basis for the presented algorithms in the following sections.

3.3.4 Metropolis Algorithm

In 1953 Metropolis, Rosenbluth & Rosenbluth and Teller & Teller developed an algorithm to draw random numbers from a general (multivariate) distribution. They describe a general algorithm nowadays known as the Metropolis algorithm. An introduction to this method can be found in (Hammersley and Handscomb 1964, p. 117). Algorithm 3.3.4 presents the Metropolis algorithm. The transition kernel q has to be symmetrical and suggests a possible move. This move is accepted or rejected by the α rule. The Metropolis method is a rejection method that generates a sample and can be used to estimate an integral.

The idea of rejection was developed by Neumann (1951). The main idea is that a generated value called candidate is rejected with a certain probability. The function $\pi(x)$ is a complex function where the normalising constant might be unknown. The proposal distribution $q(x)$ is a function from which samples can be drawn and $M \in R_{\geq 0}$ is a “covering constant” (Liu 2001, p. 24) so that:

$$M \cdot q(x) \geq \pi(x) \quad \forall x \in R^d. \quad (3.3.13)$$

The rejection sampling algorithm that was suggested by von Neumann is described in (Liu 2001, p. 24). Fig. 3.33 illustrates the relation of the involved functions.

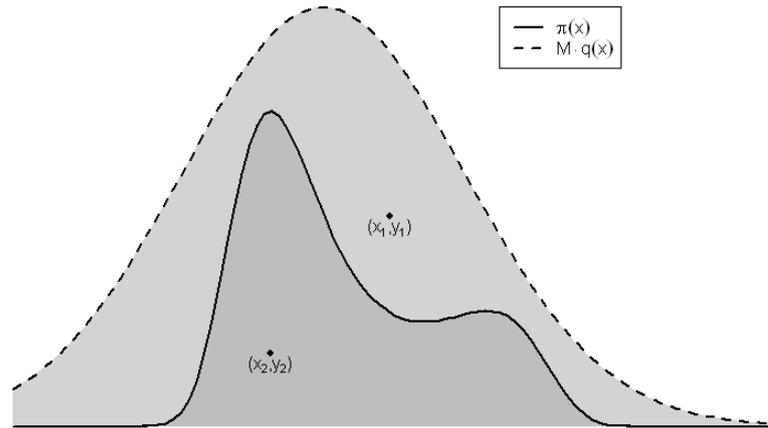


Figure 3.33: Target (π) and Enveloping ($M \cdot q$) Function in Rejection Sampling
adapted from (MacKay 2003, p. 364)

A candidate is drawn with the help of the proposal function q . This candidate is rejected iff it is in the area of $M \cdot q(x)$ but not in the area of $\pi(x)$. In Fig. 3.33 two candidates from the proposal function are shown. Point (x_1, y_1) is rejected and point (x_2, y_2) is accepted because (x_2, y_2) lies below the target function $\pi(x)$.

Obviously, the rejection method works well if the proposal function is similar to the target function. The acceptance rate can be expressed as:

$$\alpha(x) = \frac{\int \pi(x)}{M \cdot \int q(x)}.$$

The Metropolis algorithm picks up the idea of rejection techniques. However, the acceptance rate α is independent of the proposal distribution. The basic idea is the creation of a Markov chain with a stationary density equal to the target function $\pi(x)$.

A candidate y_t at time t only depends on the generated sample x_{t-1} at time $t - 1$. The proposal distribution is now $q(y_t, x_{t-1})$. The basic assumption is that q is a symmetrical function (Metropolis et al. 1953), which means:

$$q(y_t, x_{t-1}) = q(x_{t-1}, y_t).$$

The acceptance rate α is given as:

$$\alpha(y_t, x_{t-1}) = \frac{\pi(y_t)}{\pi(x_{t-1})}. \quad (3.3.14)$$

Algorithm 3.3.4 describes the procedure. The target function π , the function from which a sample is desired, and the symmetrical transition kernel q constitute the required input. The algorithm produces a sample x .

Algorithm 5 Metropolis Algorithm

Input: $\pi()$ – target function

$q(\cdot, \cdot)$ – proposal distribution, transition kernel

$U(0, 1)$ – uniform distribution in $[0, 1]$

t – iteration index

T – number of iterations

Output: $x = \{x_1, \dots, x_T\}$ – sample

```

1:  $t = 0$ 
2: initialise  $x_t$ 
3: repeat
4:   increment  $t$ 
5:   generate  $y \sim q(x_{t-1}, \cdot)$ 
6:   generate  $u \sim U(0, 1)$ 
7:   calculate  $\alpha(x_{t-1}, y) \leftarrow \min \left\{ 1, \frac{\pi(y)}{\pi(x_{t-1})} \right\}$ 
8:   if  $\alpha(x_{t-1}, y) \geq u$  then
9:      $x_t \leftarrow y$ 
10:  else
11:     $x_t \leftarrow x_{t-1}$ 
12:  end if
13: until  $t = T$ 

```

A candidate y at time t is accepted if its target value $\pi(y)$ is greater than $\pi(x_{t-1})$. Otherwise, a uniformly distributed random number u is drawn and the candidate y is accepted only if $\alpha(y_t, x_{t-1}) \geq u$. Otherwise, the candidate is rejected and the last accepted value is inserted in the sample.

The Metropolis method is a cornerstone in MCMC techniques and is particularly used in statistical physics (Liu 2001, p. 107). Its efficiency mainly depends on the initial value x_0 . Although the values can be arbitrarily chosen, they should be close to the target function. Otherwise, the algorithm needs some iterations to guarantee that the sample follows the target function. This is called the Burn-In phase. If the initial values are not in the area of the target function, the total iteration number should be increased and the first part of the sample should be eliminated from further computations.

A basic assumption of the initial values is that $\pi(x_0) > 0$. The acceptance ratio cannot be equal to one due to the fact that in this case direct sampling from the target function is necessary (Gamerman 2002, p. 175).

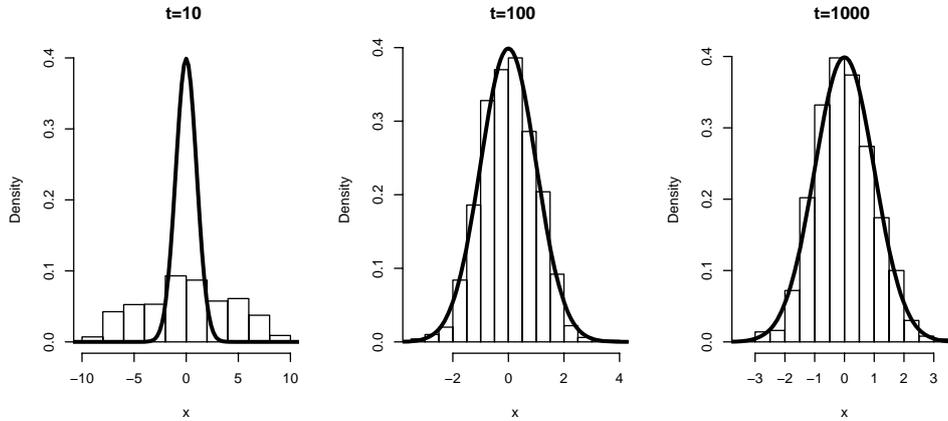


Figure 3.34: Histograms of Metropolis Iterations at $t = 10, 100, 1000$

Example 3.3.3 (Metropolis Algorithm and Burn-In). The Burn-In problem is visualised in the following example. The target function π is equal to the standard Gaussian density. The number of particles used at time t to estimate the density is set to $N = 1000$. The total iteration number of the algorithm is 5000. An AR1 process is used as a proposal function $q(y, x_{t-1})$, where $y = x_{t-1} + u$ and $u \sim U(-1, 1)$.

If the initial values x_0 are close to the target function, eliminating the Burn-In is not necessary. If the initial values are significantly different, the first iterations of the sampling sequence have to be eliminated. In the following the initial values are uniformly distributed in the interval $[-10, 10]$.

The histograms of generated samples at time points $t = 10, 100$, and 1000 are given in Fig. 3.34. Moreover, the target function is plotted. At iteration $t = 10$, the initial values still have a high impact. At $t = 100$, the sample is obviously already following the target function. This state is stable as can be seen at time $t = 1000$. The Burn-In is thoroughly discussed in Section 3.4.3.

The following example illustrates the problem of choosing a non-symmetrical proposal function.

Example 3.3.4 (Metropolis Algorithm and non-Symmetry). The simulation operations are the same as in example 3.3.3. The only change involves the proposal function, where the added noise now has an exponential distribution with $\lambda = 1/2$ and a shift of minus two.

In Fig. 3.35, it can be seen that again at iteration $t = 10$ the initial values dominate. At $t = 100$, the values are closer to the target function but shifted to the left. This state is also stable as it can be seen at time $t = 1000$.

If the proposal function is not symmetrical, the Metropolis algorithm fails. It also fails if the target function is non-symmetrical (Metropolis et al. 1953). This is a severe drawback. The generalisation of Metropolis' method by Hastings

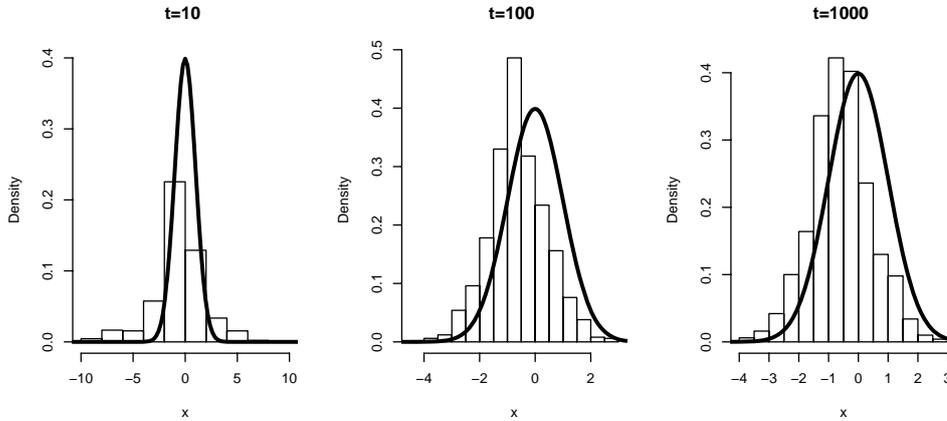


Figure 3.35: Histograms of Metropolis Iterations at $t = 10, 100, 1000$ with Non-Symmetrical Proposal Function

is presented in the following section. This generalisation manages to solve the conflict of non-symmetry in both cases.

3.3.5 Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH)-algorithm (Hastings 1970) is a further development of the Metropolis algorithm. Hastings combined the rejection method used in Metropolis et al. (1953) with a weighting component. This is realised by changing the acceptance ratio by a factor that considers the impact of sampled candidates. The acceptance rule α changes to:

$$\alpha(y, x_{t-1}) = \frac{\pi(y) \cdot q(x_{t-1}, y)}{\pi(x_{t-1}) \cdot q(y, x_{t-1})}. \quad (3.3.15)$$

In the case of a symmetrical proposal function ($q(y, x_{t-1}) = q(x_{t-1}, y)$), the ratio is equal to Metropolis acceptance ratio (see Eq. 3.3.14).

While the Metropolis algorithm is restricted to symmetrically density functions, the MH algorithm can be used for any density function $\pi(\cdot)$. The restriction on the proposal function to the symmetrical case is neutralised. The only restriction in Hastings generalisation is that the transition kernel $q(x, y) > 0$ if and only if $q(y, x) > 0$ (Liu 2001, p. 111). For regularity conditions of the transition kernel see Roberts (1996). The second important restriction for the MH algorithm is identical to the restriction of initial values in the Metropolis algorithm ($\pi(x_0) > 0$).

The MH algorithm is described for instance in Hastings (1970), Chib and Greenberg (1995), Gilks et al. (1996a) and Liu (2001). Implementation deviations arise due to different dealings of the above restrictions. Chib (2004) for instance, changes the ratio α in the way that $\alpha = 1$ for $\pi(x_{t-1}) \cdot q(x_{t-1}, y) \leq 0$. (Liu 2001,

Algorithm 6 Metropolis-Hastings Algorithm

Input: $\pi(\cdot)$ – target function
 $q(\cdot, \cdot)$ – proposal distribution, transition kernel
 $U(0, 1)$ – uniform distribution in $[0, 1]$
 t – iteration index
 T – number of iterations

Output: $x = \{x_1, \dots, x_T\}$ – sample

- 1: $t = 0$
- 2: initialise x_t
- 3: **repeat**
- 4: increment t
- 5: generate $y \sim q(x_{t-1}, \cdot)$
- 6: generate $u \sim U(0, 1)$
- 7: calculate $\alpha(x_{t-1}, y) \leftarrow \min \left\{ 1, \frac{\pi(y) q(x_{t-1}, y)}{\pi(x_{t-1}) q(y, x_{t-1})} \right\}$
- 8: **if** $\alpha(x_{t-1}, y) \geq u$ **then**
- 9: $x_t \leftarrow y$
- 10: **else**
- 11: $x_t \leftarrow x_{t-1}$
- 12: **end if**
- 13: **until** $t = T$

p. 112) suggests a different acceptance ratio:

$$\alpha(x_{t-1}, y) = \frac{\delta(x_{t-1}, y)}{\pi(x_{t-1}) \cdot q(x_{t-1}, y)}$$

where $\delta(x_{t-1}, y)$ is any symmetric function chosen so that $\alpha(x_{t-1}, y) \leq 1 \forall x_{t-1}, y$.

In Algorithm 6, the MH method is presented. This algorithm is analogous to the Metropolis algorithm (Alg. 3.3.4). Let us prove that the MH-algorithm generates a Markov chain with a stationary distribution equal to π . The Transition kernel T_{MH} of the sequence proposed by the MH algorithm is:

$$\begin{aligned} T_{MH}(x_{t+1}, x_t) &= q(x_{t+1}, x_t) \alpha(x_{t+1}, x_t) \\ &+ \delta(x_{t+1} - x_t) \int q(s, x_t) (1 - \alpha(s, x_t)) ds \end{aligned} \quad (3.3.16)$$

where $\alpha(z, x) = \min \left\{ 1, \frac{p(z) q(x, z)}{p(x) q(z, x)} \right\}$.

Stationarity of the MH-Algorithm. π is the stationary density of the Markov chain,

because

$$\begin{aligned}
\int T_{MH}(s_{t+1}, s_t) \pi(s_t) ds_t &= \int q(s_{t+1}, s_t) \alpha(s_{t+1}, s_t) \pi(s_t) ds_t \\
&+ \int \delta(s_{t+1} - s_t) \int q(s, s_t) (1 - \alpha(s, s_t)) ds ds_t \\
&= \int \min\{\pi(s_t) q(s_{t+1}, s_t), \pi(s_{t+1}) q(s_t, s_{t+1})\} ds_t \\
&+ \int q(s, s_{t+1}) (1 - \alpha(s, s_{t+1})) \pi(s_{t+1}) ds \\
&= \int \min\{\pi(s_t) q(s_{t+1}, s_t), \pi(s_{t+1}) q(s_t, s_{t+1})\} ds_t \\
&+ \int \pi(s_{t+1}) q(s, s_{t+1}) ds \\
&- \int \pi(s_{t+1}) \alpha(s, s_{t+1}) q(s, s_{t+1}) ds \\
&= \int \min\{\pi(s_t) q(s_{t+1}, s_t), \pi(s_{t+1}) q(s_t, s_{t+1})\} ds_t \\
&+ \pi(s_{t+1}) \\
&- \int \min\{\pi(s_t) q(s_{t+1}, s_t), \pi(s_{t+1}) q(s_t, s_{t+1})\} ds_t \\
&= \pi(s_{t+1})
\end{aligned}$$

□

The following example underlines the fact that the MH algorithm works also in the case of a non-symmetric proposal function.

Example 3.3.5 (Metropolis Hastings Algorithm and non-Symmetry). In this example the same simulation data as in example 3.3.4 is used. By running the simulation with the MH algorithm it can be seen that the generated sample is close to the desired target function.

In Fig. 3.36, again three different time steps are visualised with the histogram plots of the generated sample x_t . It can be seen that at $t = 10$ the sample is closer to the initial uniform distribution compared to Example 3.3.4. Yet at $t = 100$, the target function is well approximated. This status again is stable.

Comparing Fig. 3.35 and Fig. 3.36, a further effect can be seen. The sample in MH algorithm converges much faster to the target function.

Despite the computational overhead compared to Metropolis algorithm, the MH algorithm is much more efficient (MacKay 2003, p. 365). This is due to the fact that the candidates are chosen better. Challenges of the MH method are described in more detail in Section 3.4, where the issues of choosing proposal function, Burn-In and other relevant simulation environment parameters are discussed.

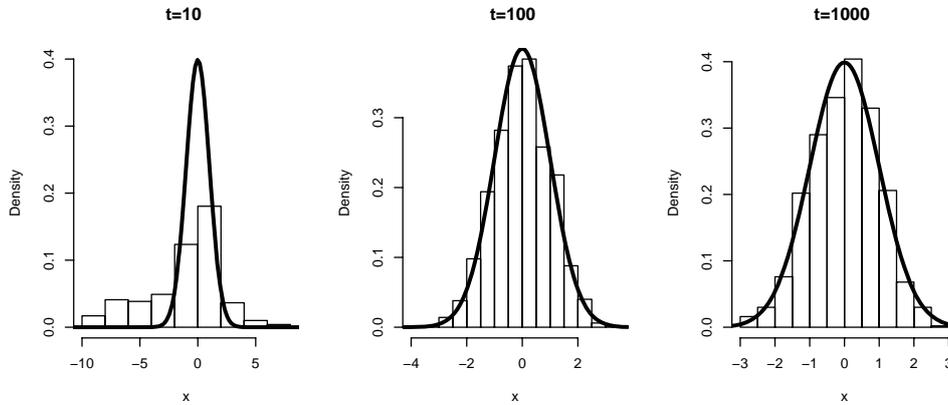


Figure 3.36: Histograms of Metropolis-Hastings Iterations at $t = 10, 100, 1000$ with Non-Symmetrical Proposal Function

The Gibbs Sampler, described in the following section, is a special case of the MH algorithm.

3.3.6 Gibbs-Sampler

The Gibbs-Sampler was proposed by Geman and Geman (1984). This method was proposed in the field of image processing and is also quite popular in statistics. The method works in the area of missing data as well (Tanner and Wong 1987). The Gibbs Sampler is the most widely used MCMC method (Gilks et al. 1996a, p. 12). Although the name is misleading since the method is not restricted to Gibbs distributions, the Gibbs Sampler is more common than earlier publications in physics, such as Glauber (1963), Huang (1973) and Creutz (1980).

The main idea is the decomposition of a multivariate distribution (d dimensions). A sample consists of T values (iterations). The decomposition is easy to be implemented because a sequence of conditional distributions is simpler to obtain than to compute the marginal of the joint distribution by integration.

It is essential that:

$$x = (x_1, \dots, x_c) \text{ with } c = 1, \dots, C \quad (3.3.17)$$

where C is the number of components of sample x . This number can represent the dimensions of the sample as suggested by Liu (2001, p. 130). However, it is not necessary that $C = d$. Some dimensions might be merged if it is useful in the implementation of the Gibbs Sampler. Due to the decomposition of the sample, the Gibbs method might be interesting in cases where multivariate sampling is quite complex.

Two Gibbs Sampler methods can be distinguished: A systematic approach and a random-scan Gibbs Sampler (Liu 2001, pp. 129). At time $t + 1$, a value for $x^{(c)}$

is generated with the proposal distribution q . This is done according to MacKay (2003, p. 371):

$$x_{t+1}^{(c)} = q(x^{(c)} | x_{t+1}^{(1)}, x_{t+1}^{(2)}, \dots, x_{t+1}^{(c-1)}, x_t^{(c+1)}, \dots, x_t^{(C)})$$

A candidate generated in this way is always accepted (Gilks et al. 1996a, pp. 10-12). The authors describe the combination of Metropolis-Hastings and Gibbs Sampler as well.

The following algorithm (Alg. 7) is taken from Gamerman and Lopez (2006, pp. 142–143) and describes the Gibbs Sampler.

Algorithm 7 Gibbs Sampler

Input: $\pi()$ – target function

$\pi_i(x^i | x^{(-i)})$ – full conditional distribution, from which samples are drawn

$U(0, 1)$ – uniform distribution in $[0, 1]$

t – iteration index

T – number of iterations

Output: $x = \{x_1, \dots, x_T\}$ – sample

1: $t = 0$

2: initialise $x_t = (x_0^1, x_0^2, \dots, x_0^c)$

3: **repeat**

4: increment t

5: **for** $c := 1$ TO C **do**

6: generate $x_t^c \sim \pi(x_t^c | x_t^1, x_t^2, \dots, x_t^{c-1}, x_{t-1}^{c+1}, \dots, x_{t-1}^C, \cdot)$

7: **end for**

8: **until** $t = T$

The following example shows that this algorithm is indeed a MH algorithm (Gamerman and Lopez 2006, p. 211). Setting the proposal distribution q in the MH Algorithm 6 with $q_i = \pi_i$, the candidate is drawn from the full conditional distribution and is accepted with $\alpha = 1$.

Obviously, the computational effort of the Gibbs Sampler compared to the MH algorithm is lower. This is justified by the fact that the ratio given in Eq. 3.3.15 does not have to be computed.

The following example should emphasise the fact that MH outperforms Gibbs sampling.

Example 3.3.6 (Gibbs sampling vs. MH). This example is taken from Carlin et al. (1992) and (Gamerman and Lopez 2006, p. 211–213). A sample $x = (x_1, \dots, x_n)$ should be generated from a Poisson distribution with a change point at m . This means: $x_i | \lambda_1 \sim Poi(\lambda_1)$, $i = 1, \dots, m$ and $x_j | \lambda_2 \sim Poi(\lambda_2)$, $j = m + 1, \dots, n$. The independent prior distributions for λ are $\lambda_1 \sim G(a, b)$, $\lambda_2 \sim G(c, d)$. The change point m is uniformly distributed over all possible sample states ($m \sim U(1, n)$).

The posterior density is:

$$\pi(\lambda_1, \lambda_2, m) \propto \lambda_1^{a+(\sum_{i=1}^m x_i)-1} e^{-(b+m)\lambda_1} \lambda_2^{c+(\sum_{i=m+1}^n x_i)-1} e^{-(d+n-m)\lambda_2}$$

The full conditional densities are easily obtained:

$$\begin{aligned} \pi_{\lambda_1}(\lambda_1) &= G\left(a + \sum_{i=1}^m x_i, b + m\right) \\ \pi_{\lambda_2}(\lambda_2) &= G\left(c + \sum_{i=m+1}^n x_i, d + n - m\right) \\ \pi_m(m) &= \frac{\lambda_1^{a+(\sum_{i=1}^m x_i)-1} e^{-(b+m)\lambda_1} \lambda_2^{c+(\sum_{i=m+1}^n x_i)-1} e^{-(d+n-m)\lambda_2}}{\sum_{j=1}^n \lambda_1^{a+(\sum_{i=1}^j x_i)-1} e^{-(b+j)\lambda_1} \lambda_2^{c+(\sum_{i=j+1}^n x_i)-1} e^{-(d+n-j)\lambda_2}} \end{aligned} \quad (3.3.18)$$

The analytical marginal posterior distributions can be found in Gamerman and Lopez (2006, p. 144).

The above-mentioned model can be applied to $n = 112$ observations of counts of coal mining disasters in Great Britain by the years 1851 to 1962 as collected by Maguire et al. (1952) and corrected by Jarrett (1979). Tab. 3.1 shows the observations.

In Fig. 3.37 a times series plot of the data is given, where the assumed change point is marked (year = 1891).

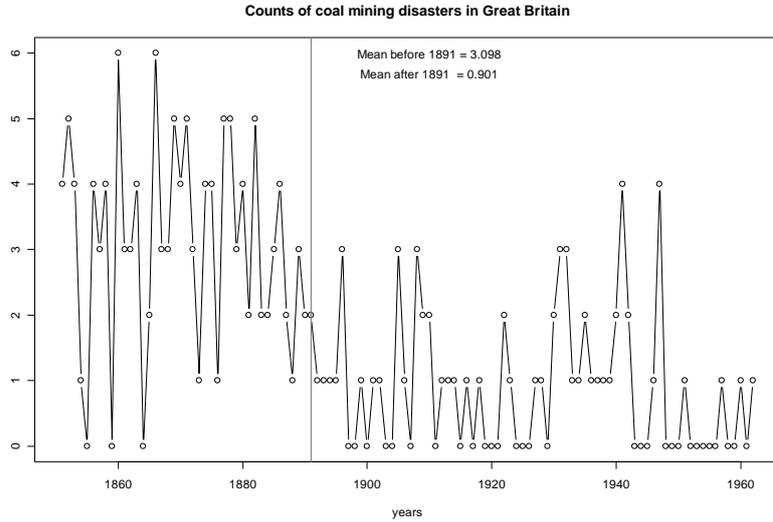


Figure 3.37: Time Series Plot of Coal-Mining Disasters in Great Britain

The simulation is done with the following parameters:

Table 3.1: Coal-Mining Disaster data in Great Britain by years 1851-1962
(taken from Carlin et al. (1992))

Year	Count								
1851	4	1874	4	1897	0	1920	0	1943	0
1852	5	1875	4	1898	0	1921	0	1944	0
1853	4	1876	1	1899	1	1922	2	1945	0
1854	1	1877	5	1900	0	1923	1	1946	1
1855	0	1878	5	1901	1	1924	0	1947	4
1856	4	1879	3	1902	1	1925	0	1948	0
1857	3	1880	4	1903	0	1926	0	1949	0
1858	4	1881	2	1904	0	1927	1	1950	0
1859	0	1882	5	1905	3	1928	1	1951	1
1860	6	1883	2	1906	1	1929	0	1952	0
1861	3	1884	2	1907	0	1930	2	1953	0
1862	3	1885	3	1908	3	1931	3	1954	0
1863	4	1886	4	1909	2	1932	3	1955	0
1864	0	1887	2	1910	2	1933	1	1956	0
1865	2	1888	1	1911	0	1934	1	1957	1
1866	6	1889	3	1912	1	1935	2	1958	0
1867	3	1890	2	1913	1	1936	1	1959	0
1868	3	1891	2	1914	1	1937	1	1960	1
1869	5	1892	1	1915	0	1938	1	1961	0
1870	4	1893	1	1916	1	1939	1	1962	1
1871	5	1894	1	1917	0	1940	2		
1872	3	1895	1	1918	1	1941	4		
1873	1	1896	3	1919	0	1942	2		

- number of iterations $n = 2500$
- initial value $m_0 = 1891$
- system parameters $a = b = c = d = 0.0001$.

The results can be seen in Fig. 3.38 and Fig. 3.39. Both figures give the marginal posterior distribution (MPD), the true value and the histogram approximation.

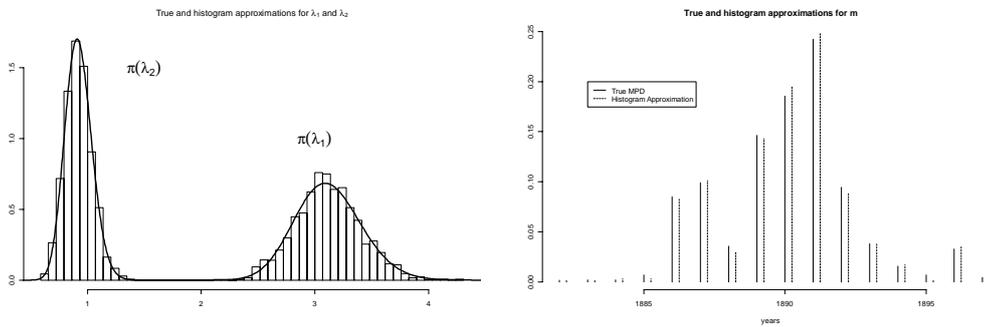


Figure 3.38: True and Approximated MPD for λ_1 and λ_2

Figure 3.39: True and Approximated MPD for m

Evidently, the Poisson rates are distinct (Fig. 3.38). The change point is with a probability 25% in the year 1891, which is deduced from Fig. 3.39.

This simulation can also be done with MH algorithm. Instead of sampling m' from its full conditional density function $\pi_m(m)$, a sampling can be done using proposal function $q(m, m')$. MH shows a better performance than Gibbs Sampler since Eq. 3.3.18 does not have to be evaluated at all possible values.

In the simulation study the sample m' is generated with a proposal function $q(m, m') \propto e^{-\tau \cdot |m - m'|}$. The tuning parameter $\tau \in [0.05; 0.95]$ and all other simulation parameters are equal to the values in the Gibbs simulation study.

The computational effort in this simulation is approximately 40% of Gibbs sampler. In Fig. 3.40, the impact of tuning parameter τ on the acceptance rate can be seen. The average acceptance rates for different tuning parameters τ are plotted.

The Gibbs Sampler is not further considered in the rest of this document, because the MH algorithm is a generalisation of the Gibbs Sampler. Although the Gibbs Sampler is more efficient than the MH method, because it accepts every move, it is also possible to find an MH algorithm that outperforms Gibbs Sampler. This certainly requires a more “intelligent” and complex simulation environment for the outperforming MH as can be seen in Example 3.3.6.

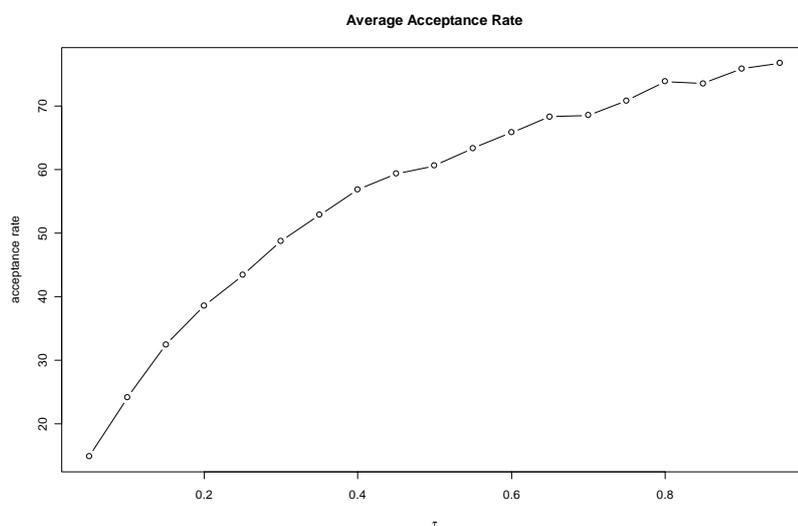


Figure 3.40: Acceptance Rates Corresponding to Tuning Parameter τ

3.3.7 Missing Values

This section addresses the issue of missing values. Values might be missing due to several reasons. They can severely affect results. If missing data are ignored or it is assumed that excluding missing data is sufficient, the risk of obtaining invalid results is not negligible.

In the case of equation systems with stochastic variables, it is possible to impute missing values with the help of an equation system. There are three relevant cases for missing data of an equation system based on:

1. prior information of LHS variables,
2. LHS variables, which can be computed according to case 1, and
3. LHS variables, which cannot be computed.

In order to illustrate all three cases recall the simplified DuPont system from Section 2.2.1. Prior information is at hand only for cost and transaction volume. Profit can be directly computed by the equation system (case 1). Profit Margin can be computed if profit is already available (case 2). Information about capital turnover cannot be computed (case 3).

In the following, imputation is explained and resulting defiances are addressed. Prior information can be used in order to generate unknown data with the given equation system. There are two possible cases. Either one or more (than one) estimates are generated for a variable. In the first case, new data can be directly used to obtain further estimates for variables with help of the equation system. In

the second case, a combined estimate as described in Eq. 3.3.20 is useful. Nevertheless, if estimates for a variable are M-inconsistent an imputation is not feasible and the complete procedure stops.

In order to obtain an algorithm for imputation, the maximum iteration of the procedure is equal to the number of equations involved in the equation system. The following algorithm describes the imputation steps.

Algorithm 8 Imputation of Missing Values

Input: Equation System, Prior information

Output: combined estimates for variables with missing values

```

1: Set of Variables with known values ( $Set_{known}$ ) := Prior information
2: Set of missing values ( $Set_{Missing}$ ) := Set of all Variables -  $Set_{known}$ 
3: iteration counter ( $ic$ ) := number of equations -  $|Set_{known}|$ 
4: for  $ic$  to number of equations do
5:   for all equations do
6:     if LHS in  $Set_{known}$  then
7:       Resolve RHS
8:     if RHS variable in  $Set_{missing}$  then
9:       add RHS to  $Set_{known}$ 
10:      remove RHS variable from  $Set_{missing}$ 
11:     else
12:       combine estimates of RHS variable according to Eq. 3.3.20
13:     end if
14:   end for
15: end for
16: if  $|Set_{known}| > (ic + 1)$  then
17:    $ic := |Set_{known}| - 1$ 
18: end if
19: end for

```

Algorithm 8 returns improved estimates for all state variables of the equation system. Due to the fact that most business indicator systems are ordered in a hierarchical structure, it might be necessary to resolve the equation system with loops.

Example 3.3.7 (Missing Value Imputation). Given the equation system M with two equations and indicators A, B, C, D, E :

1. $A = B + C$
2. $D = A * E$

and prior information B, C , and E , then in the first step A has to be estimated. The values of A, D can be resolved.

A further option in order to impute missing values is to resolve desired equations and substitute missing values by equations. However, this procedure ends

with equations that have more than one operator and the data structure for this equation solver is much more complex. Therefore, this approach is not pursued.

3.3.8 M-Consistency Check

This section deals with the problem of competing estimators. When using prior information and a system of equations, different estimates per variable might be gained for all equations where this variable shows up. Yet which information should be used?

When using crisp data, different estimates will certainly lead to model inconsistency (M-inconsistent). In the case of M-inconsistency, there can be only one choice. All prior information has to be checked and revised. This includes not only variables where M-inconsistency is detected but also all variables that were used for generation of estimates for such a variable.

The M-consistent case occurs if the prior information and all estimates are equal per variable. There are two possible states of M-inconsistency: strong M-inconsistency and weak M-inconsistency. These are defined in the following.

Definition 3.3.8 (M-Inconsistent Data). Given equation system M , let $k > 1$ be samples for a variable $x \in \mathfrak{X}$ and $\alpha \in \mathbb{R}_{[0,1]}$. For each sample the quantiles $q_{\frac{\alpha}{2}}$ and $q_{1-\frac{\alpha}{2}}$ are computed. Let $\underline{q_{max}}$ be the maximum of all $q_{\frac{\alpha}{2}}$ and $\overline{q_{min}}$ be the minimum of all $q_{1-\frac{\alpha}{2}}$. The interval $I_{q_{max}q_{min}}$ is defined as $[\underline{q_{max}}, \overline{q_{min}}]$.

The system of equations and the prior information is called strong M-inconsistent iff there exists a variable $x \in \mathfrak{X}$ where

$$I_{q_{max}q_{min}} = \emptyset. \tag{3.3.19}$$

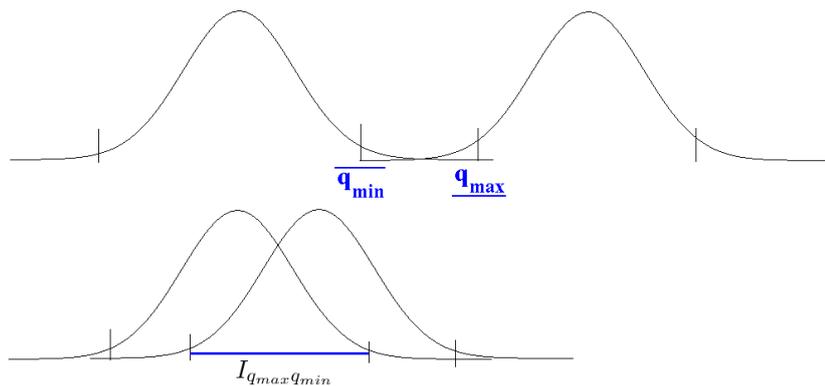


Figure 3.41: M-Inconsistent and M-Consistent Densities

A stochastic variable is M-consistent within an equation system if it is not M-inconsistent. Strong M-consistency is only present iff all estimates of a variable

are equal. Due to sampling and estimation errors, this is almost never the case in simulation experiments. In such cases the state is called weak M-consistent.

Identification of M-Inconsistent Indicators

M-Inconsistency of an indicator systems causes a problem for application. It is not reasonable to further work with such data. Therefore it is very important to detect (location) the variables that generate the M-inconsistency. The next section tries to illustrate this.

In the first step, all M-inconsistent variables have to be detected. For these variables either prior information is given or estimates are computed by the equation system as RHS variables. In the second case, the variables that are used in the equation system to generate an estimate (LHS variables) have to be evaluated as well. This means that for variables marked as M-inconsistent, all generating information (prior or LHS variables) have to be marked for further investigation. These variables build a M-inconsistency set.

If only one variable causes M-inconsistency, all marked variables have to be checked and prior information of these variables have to be adjusted before another simulation can be conducted.

If more than one variable is detected as M-inconsistent, it might be useful to first focus the effort of auditing to those variables, that occur in more than one M-inconsistent set.

The identification of M-inconsistency is explained in detail in Alg. 9.

All variables marked M-inconsistent are used as input for the detection. This set of variables is determined by the corresponding interval $I_{q_{max}q_{min}}$ as given in Eq. 3.3.19. Although the prior information of these variables might be a cause for M-inconsistency, the prior information that generates further estimates is also a possible origin of M-inconsistency. Therefore, all involved variables (RHS and LHS) of M-inconsistent variables have to be checked.

Solution Set of M-Consistent Indicator Systems

If the indicator system is M-consistent, different estimates for one variable should be combined to only one estimate. This is described in the following part.

If all variables are M-consistent, the solution set can be computed. In the first step, all estimates of a variable have to be merged into a new estimate. This is done with the help of joint distribution of all samples of a variable. This joint distribution can be seen as a function on the product space. In Fig. 3.42, such a joint distribution function for a variable is shown. In this case, it is a bivariate function, and thus yields two estimates. The combined estimator is then a function of all single estimates where all values are equal. Fig. 3.43 shows this case. Note, that a renormalisation has to be applied such that the new improved estimate for the considered variable fulfills the prerequisites of a density function.

Algorithm 9 Detection of M-Inconsistent Indicators**Input:** all detected M-inconsistent variables**Output:** set of variables for auditing

```

1: initialise set of all marked variables
2: for all M-inconsistent variables do
3:   initialise set with current M-inconsistent variable
4:   add current variable to set of all marked variables
5: end for
6: for all equations do
7:   for all M-inconsistent variables do
8:     if RHS(current equation) = current M-inconsistent variable then
9:       add to set of current M-inconsistent variable LHS(current equation)
10:      add LHS(current equation) to set of all marked variables
11:     end if
12:   end for
13: end for
14: for v in set of marked variables do
15:   counter := 0
16:   for all sets of M-inconsistent variables do
17:     if v in current set then
18:       Increment(counter)
19:     end if
20:   end for
21:   set priority to check for selected variable to counter
22: end for

```

The combined estimate for a M-consistent variable X with more than one estimator $(\hat{X}_1, \dots, \hat{X}_n)$ is:

$$\hat{f}_X(x) = c \cdot f_{(\hat{X}_1, \dots, \hat{X}_n)}(y) \quad \forall y_i = y_j \text{ with } i, j \in [1, \dots, n], \quad c \in \mathbb{R}_+. \quad (3.3.20)$$

The function $f_{(\hat{X}_1, \dots, \hat{X}_n)}$ is the joint distribution of all single estimates for the same variable. y is a vector in the interval $I_{q_{max}q_{min}}$. Properties of Eq. 3.3.20 are that the combined estimate has a variance lower or equal to the lowest estimate. This is due to the fact that the new estimate has only values greater 0 in the interval $I_{q_{max}q_{min}}$. Therefore, the mean of the improved estimate also lies in this interval. Factor c is the renormalising constant and has to be evaluated, for instance with numerical integration methods.

In order to ensure that the joint distribution exists, the corresponding functions have to fulfil the Lebesgue requirement.

Definition 3.3.9 (L1-norm). A function f with $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ where $n, m \in \mathbb{N}_+$

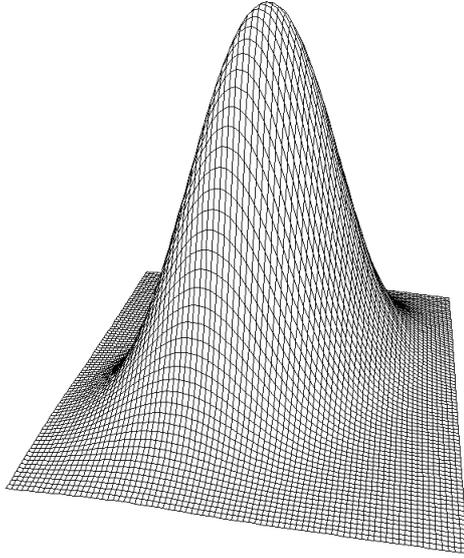


Figure 3.42: Joint Distribution of two Estimates for the Same Variable

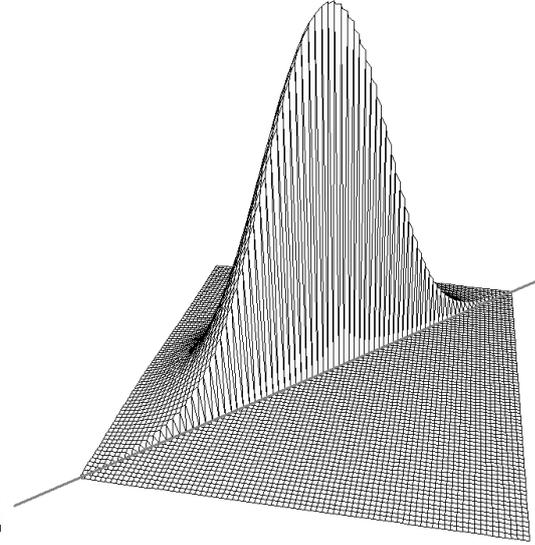


Figure 3.43: Corrected Estimator

has L1-norm:

$$\|f\|_{L1} = \int_{-\infty}^{+\infty} |f(x)| dx. \quad (3.3.21)$$

A function for which the L1-norm exists is a density function.

Definition 3.3.10 (Lk-norm). A function f with $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ where $n, m \in \mathbb{N}_+$ has Lk-norm:

$$\|f\|_{Lk} = \sqrt[k]{\int_{-\infty}^{+\infty} |f(x)|^k dx}. \quad (3.3.22)$$

If k estimates exist for a variable, then it is sufficient for the existence of the joint distribution that all functions have Lk-norm. This is in respect to the fact that the k th product space is used for the joint distribution and a missing Lk-norm implies that there is no definite number c .

The following example illustrates a density function that does not fulfil the L2-norm.

Example 3.3.8 (Lesbesgue). The function

$$f(x) = \begin{cases} \frac{1}{\sqrt{x}} - 1 & 0 < x < 1 \\ 0 & \text{else} \end{cases} \quad (3.3.23)$$

is a density function that does not have $L2$ -norm. This function cannot be used if a second estimate exists and has to be combined in the consistency step. Fig. 3.44 visualises the function.

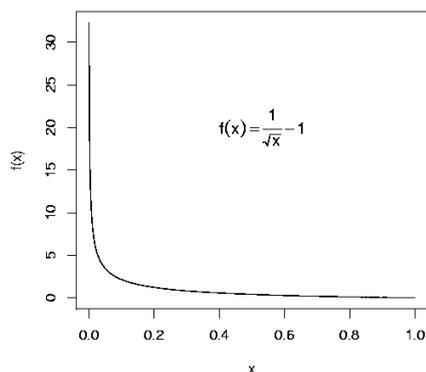


Figure 3.44: A Function, that has no $L2$ -Norm

SamPro

The above described functionalities and algorithms are combined in SamPro, (Alg. 10). This algorithm uses the model and prior information to impute missing values and to improve estimates for all involved variables.

Algorithm 10 SamPro Algorithm

Input: stochastic equation system M

one observation per variable (missing values allowed)

Output: estimates for all variables, i.e., marginal densities, means and standard deviations

- 1: resolve (set LHS \equiv RHS) for all variables in all equations
 - 2: sample from the joint density function of all RHS variables for LHS
 - 3: estimate all LHS variables
 - 4: estimate $\alpha/2, 1 - \alpha/2$ - quantiles $\underline{q}_{max}, \overline{q}_{min}$ for each variable.
 - 5: **if** $\underline{q}_{max} > \overline{q}_{min}$ **then**
 - 6: M-inconsistency found
 - 7: **else**
 - 8: compute the distribution f_{xz} restricted to the subspace $x - z = 0$
 - 9: **end if**
-

Two different approaches are possible for the computation of the distribution f_{xz} under the restriction $x - z = 0$ ($x \equiv$ RHS and $z \equiv$ LHS). If samples from a common distribution of all involved estimates of a variable exist, these can be used to

obtain the improved estimate. Due to the fact that condition $x - z = 0$ is not numerically fulfilled exactly by these samples, a margin should be applied in almost all cases. This corridor spans a hyper cylinder with the central line $x = z$. The radius of the corridor cylinder specifies both accuracy and computational effort. All values outside the corridor are eliminated from the sample and the remaining samples are used for density estimation. Before this can be done, the multidimensional sample is projected onto the central line of the cylinder. This mapping constitutes the improved estimate and can be used, for example, to generate a histogram, cf. Fig. 3.45. Three different estimates span the sample space.

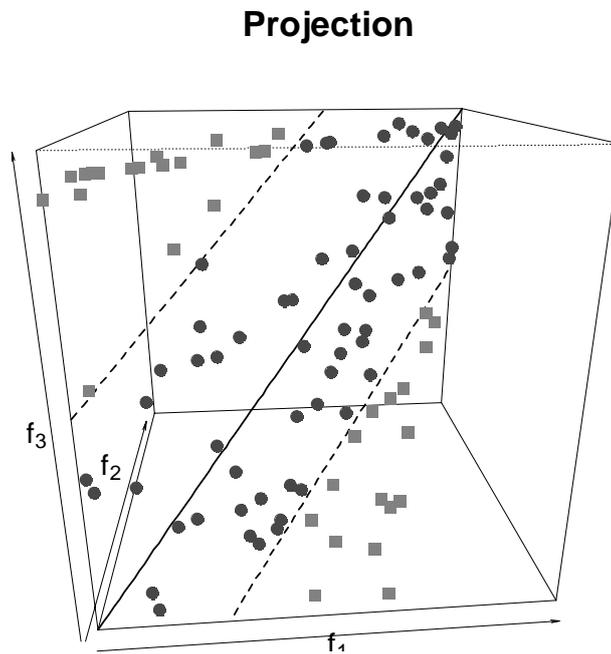


Figure 3.45: Projection for a Common Sampling Distribution

The dashed lines represent the boundary of the cylinder. All points outside (plotted with squares) are excluded from the combined estimate. The projection of the points, plotted with a circle onto the solid line, contributes to the distribution of the combined estimator.

All estimates are exclusively assumed to be either independent or dependent. The independence assumption seems to be consistent with the application of business indicator systems because most equations are definition, balance or behavioural equations and a dependency structure between variables of these equations might not be obvious. An intensive study on the effect of the independence assumption including robust analysis is given in Chapter 4.

For example, in the case of independence of x and z , the joint distribution is defined as $f_{xz} = f_x \cdot f_z$. After the computation of the joint function, it has to be renormalised to obtain a distribution function, as shown in Fig. 3.43.

Outliers are discussed in Section 3.4. Robustification of simulation samples can be applied in two kinds: one robustification is applied for an estimate (\hat{x}) and one is applied for the interval $I_{q_{max}q_{min}}$. Explanations of robustification are given in the following section.

3.4 Robustifying the Metropolis Hastings Algorithm

In this paragraph, the most common challenges of using the MCMC algorithm are discussed. This addresses not only a general problem that occurs when using this MCMC technique, but also problems that caused by the equation systems considered here.

3.4.1 Noise

In real world business and economic indicator systems effects of noise in the indicators are almost never investigated. Crisp data are used instead. Data may be erroneous. This is a result from estimation, measuring or forecasting. If part of the data has some added noise, error propagation exuberates the problem.

Two aspects of noise are of interest. One is that noise has no impact on the mean value of the indicator. The other that it influences the mean value of the data. In the case of an indicator system, only noise with a mean value of zero of the indicators is of interest to obtain M-consistent data. If the added noise has a mean value significantly different from 0, data may become M-inconsistent as described in Def. 3.3.8.

Error distributions can be of several forms. It can be symmetrical or skewed, it can have a small or large variance. Both, heavy tails or bounded domains are possible. Examples of distribution functions and their characteristics are already given in Sections 3.2.1–3.2.5.

SamPro achieves a noise reduction in almost all cases. This is due to the fact that the interval $I_{q_{max}q_{min}}$ shrinks the data space. This is also the reason why an increase in the noise is not possible. If two concurring estimates for a variable exist and both have the same first moment but differ in the second moment, then the smaller of both variances results in the new improved estimator. If the first moment of both estimates is different but the interval $I_{q_{max}q_{min}}$ is non-empty, the resulting estimate has a standardised first moment that is bounded by the first moments of both input estimates. If an estimate of *Indicator_A* has a higher second moment than the estimate of *Indicator_B*, then the combined estimate has a first moment closer to the first moment of *Indicator_B* than to that of *Indicator_A*.

3.4.2 Transition Kernels

The Metropolis-Hastings algorithm does not specify the transition kernel. The only restriction of the transition kernel is that $q(x, y) > 0$ if and only if $q(y, x) > 0$. In the literature several transition kernels have been proposed. The most important kernels are presented in the following sections.

All kernels should have some common properties. An acceptance rate of 25 - 35 % should be reached, which was determined by Gelman et al. (1996), see also Liu (2001, p. 115). This might vary from one transition kernel to another but gives a lower bound for the acceptance rate. Other authors suggest to find

transition kernels that reach acceptance rates from 20 - 50 %, see for example Besag et al. (1995), Bennett et al. (1996).

Random Walk

The random walk transition kernel q generates the candidate values as a random walk with $y \sim x_t + \epsilon$, where all ϵ are independent and identically distributed from a sampling function $Q()$. The function $Q()$ should be a spherically symmetric distribution (Liu 2001, 114). This generates candidates following a random walk of the accepted values x_t . This also simplifies the acceptance rule for the values x_{t+1} :

$$x_{t+1} = \begin{cases} y & u \leq \frac{\pi(y)}{\pi(x_t)} \\ x_t & \text{otherwise} \end{cases} \quad (3.4.1)$$

In a random walk chain the proposal values are the previous values plus a disturbance. A symmetrical disturbance with zero mean implies a symmetric chain, as in the Metropolis algorithm. Gaussian or Student's t distribution centred around zero are often used (Gamerman and Lopez 2006, p. 198). Tierney (1994) suggests that the covariance matrix is set to cS where c is a tuning parameter and S is the approximation of the posterior variance covariance matrix. The value of c should lie between 0.5 and 3.

Example 3.4.1 (Random Walk Metropolis Hastings). In this example the target function is a Bivariate Gaussian mixture:

$$\pi(x) = 0.6 \cdot f_2 \left(x, \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \right) + 0.4 \cdot f_2 \left(x, \begin{pmatrix} 0.3 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix} \right)$$

where f_2 is bivariate normal density. Fig. 3.46 plots the function and shows the bimodal aspect.

The random walk proposal function is set to $q(x, y) = f_2(y, x, I_2)$. In Fig. 3.47, different initial values and different lengths of the chain are visualised. The acceptance rate suggests that the chain is mixing efficiently. This is independent of the initial values and the length of the chain. Due to the fact that the initial values are in the domain of the true density function and the proposal q covers the target space quite well this chain works adequate. A contour plot of the true function is added in each part of Fig. 3.47.

Comments on the selection of space range for the proposal and the associated acceptance rates can be found in a similar example in Gamerman and Lopez (2006, p. 199).

Independence Chain

The independence chain is the simplest possible transition kernel to use. There is no dependency of x_{t-1} on the candidate y_t , i.e. $q(y, x) = q(y)$. The candidates are

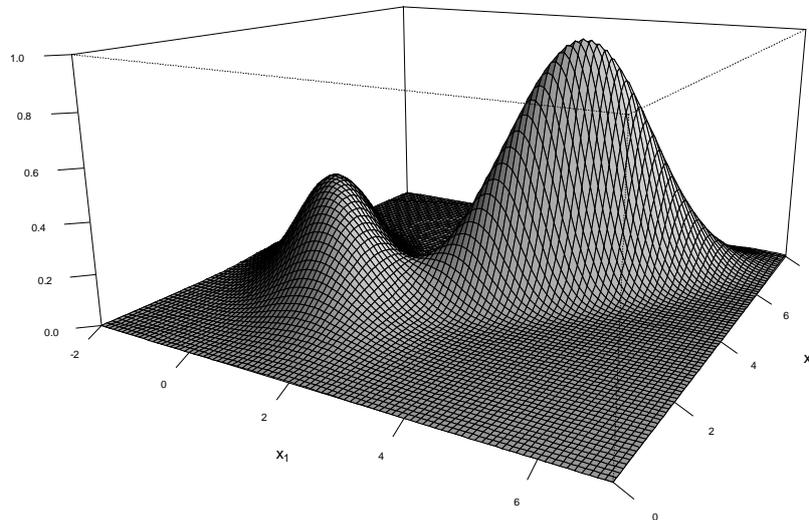


Figure 3.46: A Bivariate Gaussian Mixture

drawn from a proposal distribution Q . This also simplifies the acceptance ratio α to

$$\alpha(y, x_t) \leftarrow \min \left\{ 1, \frac{\pi(y) \cdot q(x_t)}{\pi(x_t) \cdot q(y)} \right\}. \quad (3.4.2)$$

The efficiency of the independence chain is closely connected to the differences of the shapes between target function π and proposal distribution q . Obtaining good performance results requires that q covers the domain of interest. Thus, a heavy tail distribution such as the long tailed t-distribution is a good choice. In order to obtain a good performance of the independence chain, the shape of the transition kernel should be quite similar to the target function but the variance should be higher.

The independence chain has a kernel where the new candidate is drawn independently from the last accepted value. This may imply that the generated chain is not a Markov chain. However, the combination of candidate generation and acceptance of proposed values create a Markov chain (Gamerman and Lopez 2006, pp. 199).

The proposal should be close to the density function (Liu 2001, p. 115). If the proposal distribution is equal to the target distribution, the acceptance ratio is equal to 1. In such a case, the Metropolis-Hastings (MH) algorithm should not be used because it only slows down the simulation. In order to generate candidates that are in the domain of the target's distribution tails, the proposal distribution should cover a wider domain. This is illustrated in Fig. 3.48. It shows the results for the simulation of a standard Gaussian distribution. The simula-

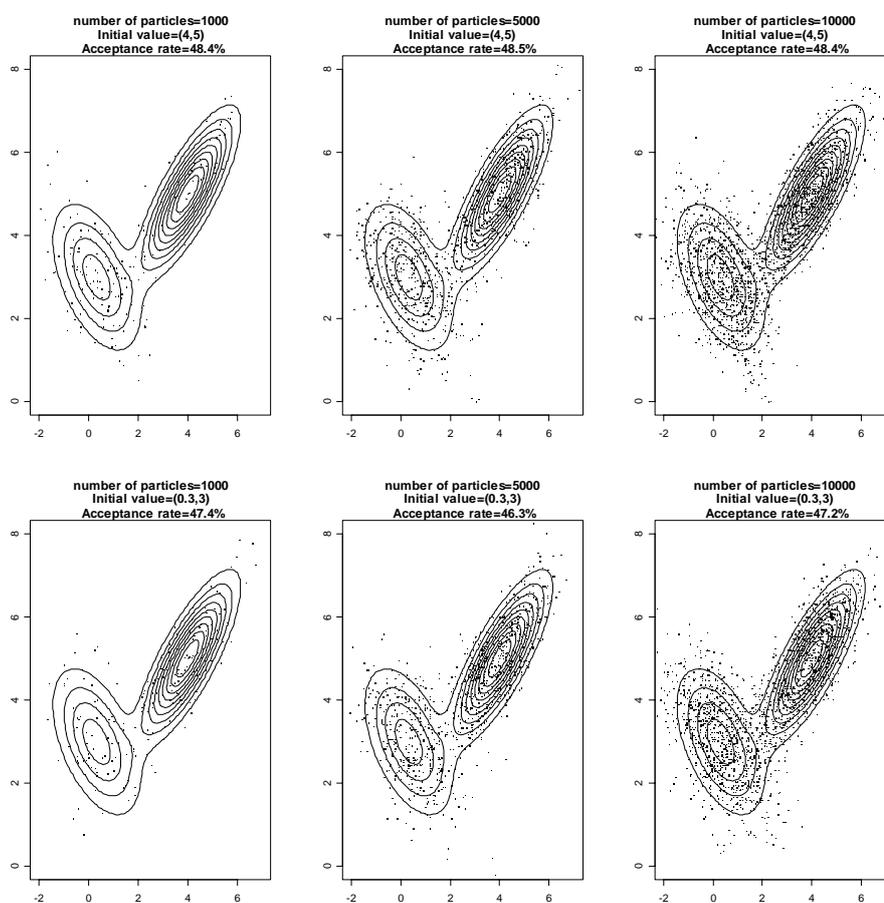


Figure 3.47: Random Walk for Bivariate Mixture with Initial Parameters and Chain Length Effects

(solid lines represent true density function)

tion is performed with different proposal distributions that generate independence chains. Each proposal distribution is a Gaussian distribution with zero mean and a standard deviation varying from 0.5 to 2.5. Rejection rate and minimum square deviation from the target distribution are plotted. It can be seen that the rejection rate is zero in the case that the proposal distribution is equal to the target distribution. By increasing the variance of the proposal distribution, the convergence rate of the chain increases, too.

Fig. 3.49 and 3.50 illustrate the fact that the shape of the proposal distribution should be close to the target function.

The target function in Fig. 3.49 is again a standard Gaussian distribution. Three different proposal distributions are used. In the first simulation, the proposal distribution is a Gaussian with zero mean and standard deviation of 1.1. The second simulation uses a Cauchy distribution with location set to 0 and scale set

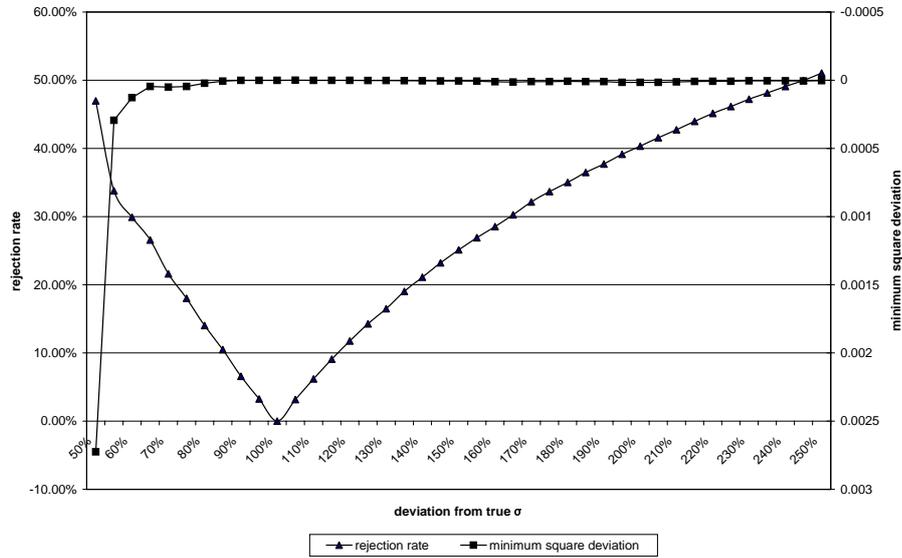


Figure 3.48: Different Proposal Functions (Deviation of $\sigma_{proposal}$ from σ_{target})

to 1. The third simulation involves a Laplace distribution with location parameter 0 and scaling parameter 1. The MH algorithm generates 1000 samples. The shape of the Gaussian proposal fits the target function best and the generated samples fit the target function well. By increasing the number of samples, the results of all proposal kernels are getting equal. This is due to the fact that the proposals fulfil the criteria for independence proposals.

In Fig. 3.50, a standard Cauchy distribution is the target function. In the first simulation experiment, a standard Gaussian is used as proposal distribution. The second proposal distribution is a Cauchy distribution located at 0 with a scale equal to 1.1. The Laplace distribution with the same parameters as before is used in the last simulation experiment. It seems that a Cauchy proposal is not a good candidate generator. However, a Kolmogorov Smirnov test on this data states that it fits best to this kernel. The second best fit is reached with a Laplace proposal. Gaussian proposal is also accepted. However, the Gaussian distribution is not a good candidate because there are not adequate samples in the generated tails.

Gamerman (2002) states that a general rule for independence chains is the avoidance of large variations which can be obtained by using proposal distributions that make ratio π/q as constant as possible. Tierney (1994) remarks that density functions with thin tails should be avoided for independence proposals. Mengersen and Tweedie (1996) describes properties for convergence (see also Robert and Casella (2003, pp. 276)).

Example 3.4.2 (Independent Metropolis Hastings). This example uses the same Bivariate Gaussian mixture target function as Example 3.4.1.

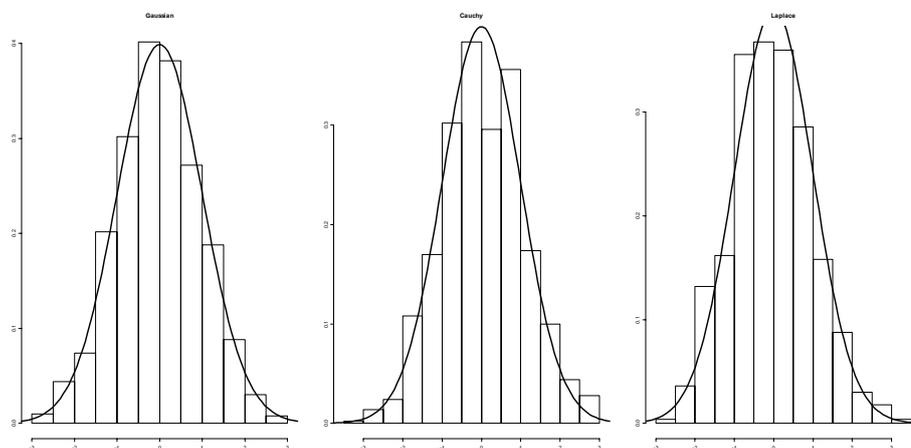


Figure 3.49: Different Shapes of Proposal Functions for Gaussian Target

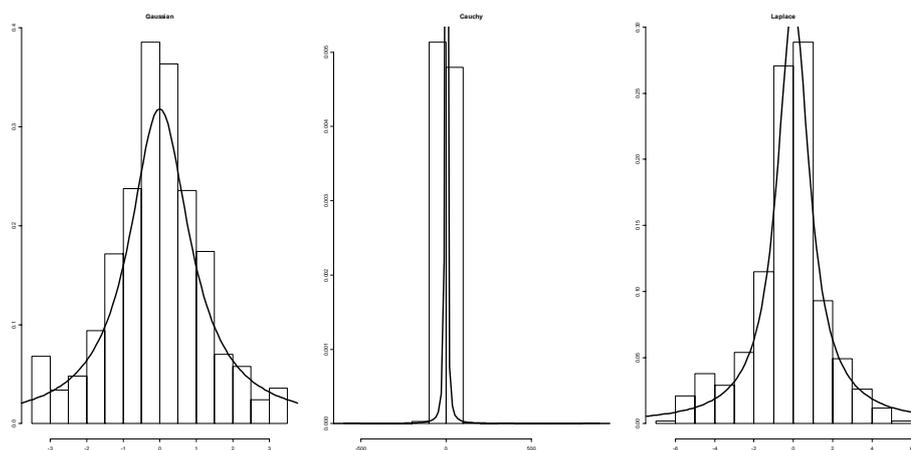


Figure 3.50: Different Shapes of Proposal Functions for Cauchy Target

The proposal function changes to $q(x, y) = f_2 \left(y, \begin{pmatrix} 2.52 \\ 4.2 \end{pmatrix}, \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix} \right)$.

Here the proposal is a Bivariate Gaussian with the mean of the target function and a covariance matrix that covers the desired sampling space.

In Fig. 3.51, independent simulations with different length and initial values are plotted. A comparison with Fig. 3.47 reveals that the acceptance rate with independence MH is lower, because the proposal generates more values that will be rejected, compared to the random walk algorithm.

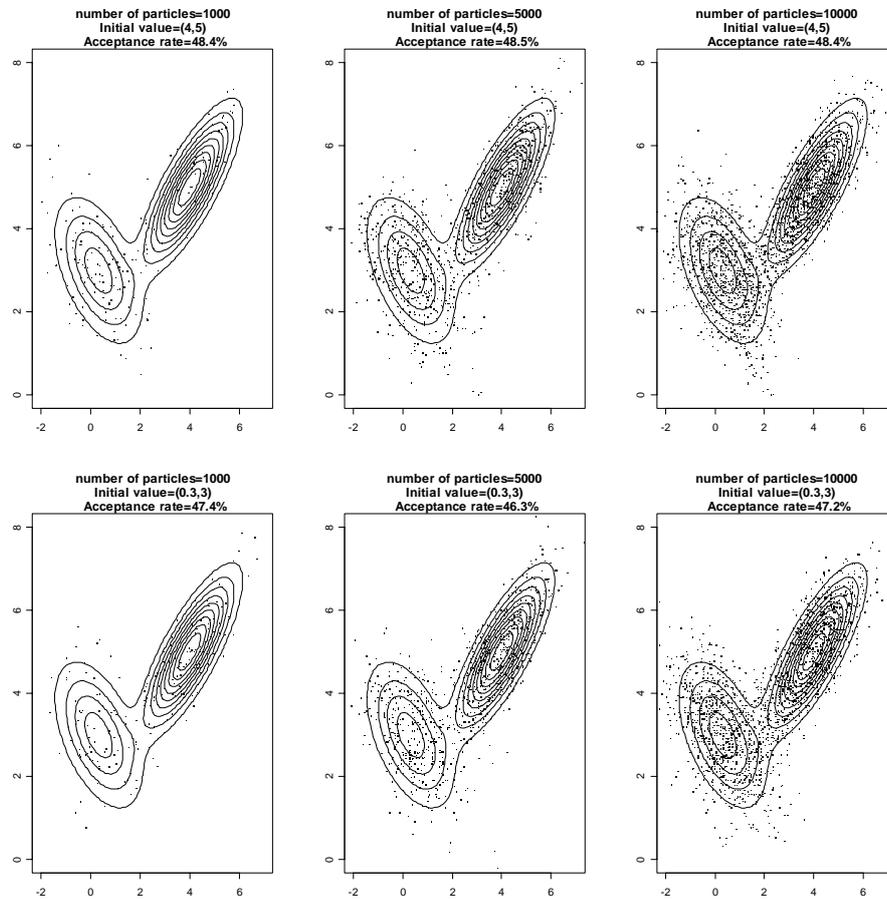


Figure 3.51: Independent MH for Bivariate Mixture with Initial Parameters and Chain Length Effects

Tuning parameters of the independence Chain algorithm can be found in Gamerman and Lopez (2006, pp. 202).

Simulated Annealing

A defiance of the above chains is that they can end up in a local extremum. In that case, the problem might be that the real target function will not be represented by the samples from the Metropolis Hastings algorithm. Although the generated chain has the stationary distribution π , the samples might stay in a local optimum due to the fact that the generation process has finished. To escape local optima simulated annealing is used, which is abutted to the metallurgic process of cooling. In physics annealing tries to minimise the energy, while temperature is the control factor.

The idea of the simulated annealing algorithm was introduced by Metropolis et al. (1953). The aim of the algorithm is to minimise the criterion function on a finite set. It is also applicable to optimisation and simulation (Robert and Casella 2003, p. 163). The change of scale (often called temperature) ensures that faster moves on the surface are possible. A positive feature of simulated annealing is that candidates may be accepted which have a lower probability without the temperature effect. This additional randomisation effect makes escapes from local minima possible.

The standard implementation of simulated annealing algorithm modifies temperature at each iteration. Algorithm 11 (Kirkpatrick et al. 1983) provides more insight.

Algorithm 11 Simulating Annealing Algorithm

Input: Temperature T_1 (reasonable large)
 $h(x)$ function that has to be minimised
 n - Simulation runs m - Length of Simulation chain $q_i(\cdot)$ - transition kernel

- 1: Initialise x_0
- 2: **for** i in 1 to n **do**
- 3: **for** j in 1 to m **do**
- 4: Generate candidate according to $q_i(\cdot)$
- 5: Accept candidate with probability α and target function $\pi_i(x) \propto e^{-h(x)/T_i}$.
- 6: **end for**
- 7: $T_{i+1} := \text{Decrease}(T_i)$ but ensure $T_{i+1} > 0$
- 8: **end for**

The sequence of temperature T should fulfil two assumptions: First, it has to be a monotone decreasing and second, T_1 should be reasonably large whereas $\lim_{i \rightarrow \infty} T_i = 0$. For convergence diagnostics see Geman and Geman (1984) and Liu (2001, p. 210).

During the first iterations of simulated annealing the particles can move relatively freely over the domain. This automates and improves the search for an adequate starting model. By cooling down, the particles become more restricted and for temperatures close to zero, good samples for $\pi(\cdot)$ are obtained for sufficiently large numbers, n , of iterations. The decrease of temperature should be of

order $O(\log(m \cdot n)^{-1})$ in order to reach the global minimum of $h(x)$ with probability 1 (Geman and Geman 1984, Holley et al. 1989).

A typical cooling process can be stated as (Walsh 2004):

$$T_i = \max \left\{ T_1 \left(\frac{T_k}{T_1} \right)^{i/n}, T_k \right\}. \quad (3.4.3)$$

The process generates n temperatures, the last temperature is T_k .

Another technique related to simulated annealing is simulated tempering (Marinari and Parisi (1992), see also Geyer and Thompson (1995)). The main idea behind this technique is the creation of a family of target functions π that vary only in their temperatures. The initial target function corresponds to the function with the lowest temperature.

There exists a wide range of specialised algorithms such as Parallel Tempering (Geyer 1991), Ensemble Simulation (Berg and Neuhaus 1992, Hesselbo and Stinchcombe 1995), and Tempering with dynamic Weighting (Liu 2001, pp. 219–224). All of these algorithms have to be carefully adjusted. However, they prove more efficient than more general methods in their application domain.

In the following sections, the common problems of all presented MCMC algorithms are described. Furthermore, solutions for specific defiances are given.

3.4.3 Burn-In Problem

The first samples generated by the Metropolis Hastings algorithm might not represent the desired target function. This occurs if the initial values are not in the domain of the target function. These first iterations have to be discarded to ensure that the generated chain has converged to the invariant distribution. The first sequence is called Burn-In. The theoretical calculation of the Burn-In is not a trivial task (Chib 2004). Computational performance issues require minimisation of the Burn-In. Therefore, the following assumptions on the proposal distribution are taken into account:

- the shape of the proposal distribution should be similar to the target function,
- the moves of the proposal distribution should make large moves on the support of the target function, and
- high numbers of rejected candidates should be avoided.

This results into requirements of the acceptance rate as well as the initial value. The initial value does not play any role in the case of the independence chain. This is due to the fact that after the first iteration, a value with a higher proposal function value is accepted. The only desired condition in such a case is that the initial target value is greater than zero.

In other Metropolis Hastings algorithms the first 1000 - 5000 iterations are typically thrown out of the sample. Different convergence tests exist to assess whether or not the chain has reached stationarity. A chain is said to be poorly mixing if no candidate is accepted at a large number of iterations. In contrast, a well mixing chain has a much higher acceptance rate.

A first hint of a well mixing chain that reached stationarity can be found in a time series plot of the samples over the iterations. In Fig. 3.52, four time series plots for different Random walk proposals are given. All chains have a Gaussian function with $\mu = 10$ and $\sigma = 1$ as a target function. The initial value is set to zero. Furthermore, the proposal functions are also Gaussian functions but with standard deviation ranging from 10 to 0.01.

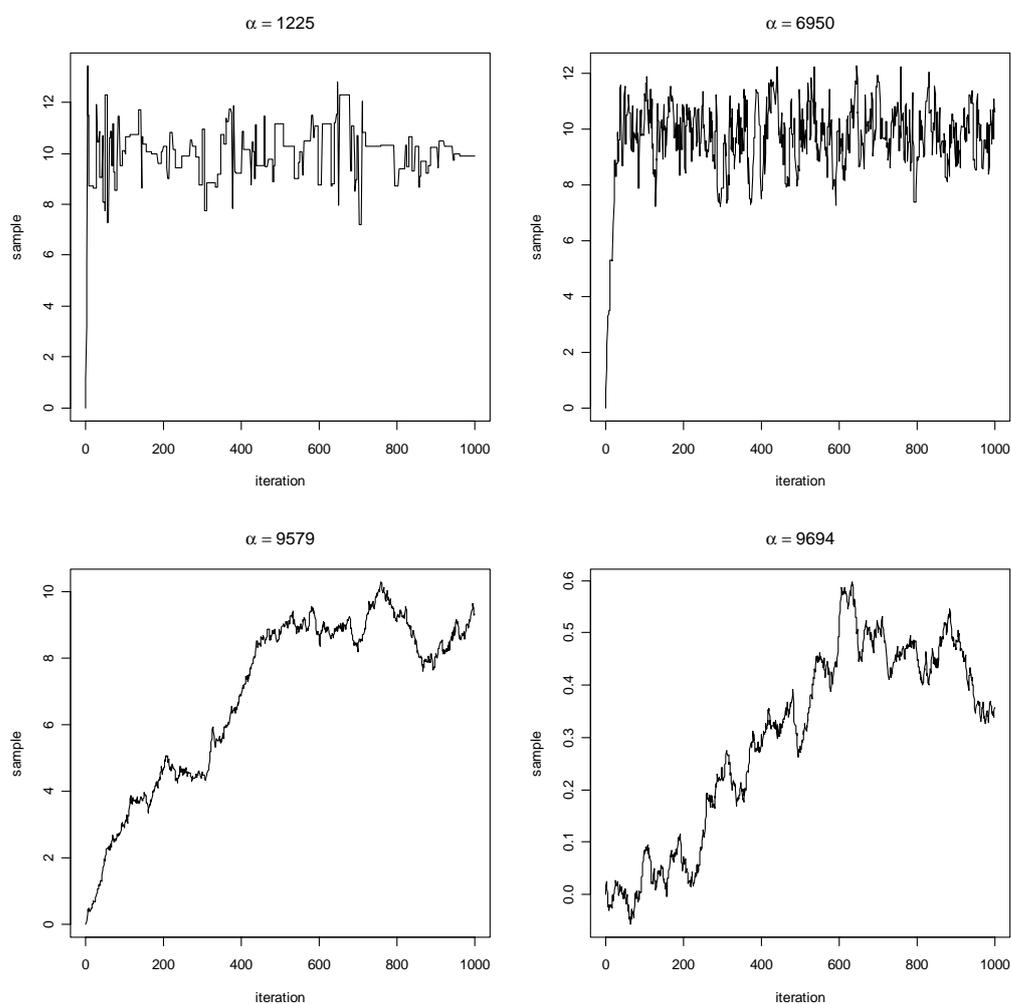


Figure 3.52: Time Series Plots of Random Walk MH

The upper left time series plot is generated by a random walk with $\sigma_{proposal} = 10$. This ensures that the Burn-In is reduced compared to all other presented random walk chains. Moreover, this chain is not well mixing. This can be seen from jumps and local steady states in the time series plot. The acceptance rate of $m = 10000$ generated candidates is only 12.25%. The second Random walk chain uses a proposal with variance equal to 1. The first iterations move from the initial value to the domain of the target function. After less than 50 iterations, the chain has reached stationarity. The acceptance rate is about 69.5% and therefore this is a well mixing chain. The lower left time series plot represents a chain with a standard deviation of 0.1. Here it is not sure that the Burn-In is finished after the first 1000 iterations. The acceptance rate of the chain is 95.8% and the chain is very well mixing. The best mixing chain is the lower right with an acceptance rate of 96.9%. Nevertheless, this chain is not reaching the stationary distribution after 10,000 iterations. This can be concluded by the fact that the time series plot slowly moves away from the initial point.

Other graphs that are helpful for estimating the Burn-In period use autocorrelation between the generated samples (Walsh 2004), where AR-1 and AR-k processes are observed. Plotting partial autocorrelation function might also be helpful in investigating the Markov Chain Monte Carlo sampler in more detail.

There also exist formal tests for stationarity (i.e. Geyer (1992), Gelman and Rubin (1992), Raftery and Lewis (1992b), Robert (1995)). In the following, two formal tests are presented. The first is easily implemented, whereas the second test is a more informative approach.

Geweke (1992) suggests a modified z-test on two samples of the generated chain, for instance the first 5% and the last 20%. If the Burn-In is already finished, both samples should have the same mean. The resulting score of the modified Z-test is referred to as Geweke Z-score. If the absolute value is greater than two, this is an indicator for drift of the mean, which means that the chain has not reached stationarity. This test procedure is also called Geweke test. In the random walks shown in Fig. 3.52 the first 1000 elements were identified as the Burn-In period. On the remaining 9000 elements the Geweke Z-score was computed (see Tab. 3.2). The standard score (Z-score) is dimensionless and it represents the distance between the raw score and the population mean in units of the standard deviation.

Table 3.2: Geweke Z-Scores on Random Walk Scenarios

Case	Fractions: 0.1 and 0.5	Fractions: 0.05 and 0.2
1 ($\sigma = 10$)	2.398	2.477
2 ($\sigma = 1$)	-0.3091	0.08316
3 ($\sigma = 0.1$)	-2.277	4.007
4 ($\sigma = 0.01$)	-24.96	-81.98

Raftery and Lewis (1992a) propose a further formal test. It is based on the estimation of the quantiles. The parameters of the Raftery-Lewis test are the quantiles of interest, the error margin of the estimate and the probability of obtaining an estimate in the error interval. For computational issues a precision is required as well. The test procedure works in such a way that a $(0,1)$ sequence is generated — 1 if $\theta_t < q$ and zero otherwise. This generated sequence is used to estimate the transition probabilities. The estimated transition kernel makes it possible to compute estimates on the Burn-In, thinning ratio and total number of chain elements need to reach stationarity. Again the above described MH random walk chains are tested with this scenario. The quantile of interest is set to 2.5%. The accuracy is set to 0.5% with 95% power of the test. Precision is fixed at 0.001. The estimated results are presented in Tab. 3.3. Note, that a dependence factor is estimated as well. This factor influences the sampling size due to autocorrelation. With this parametrisation the tested chain has to consist of at least 3746 values, see Raftery and Lewis (1992a).

Table 3.3: Raftery-Lewis Test Results of Random Walk Scenarios

Case	Estimated Burn-In	Estimated Total Sampling Size	Dependence Factor
1	37	40397	10.8
2	21	22253	5.94
3	342	309586	82.6
4	851	476726	127

3.4.4 Number of Particles

The number of particles plays a significant role for the quality of the function representation. Nevertheless, there is a trade-off between accuracy and computational effort. The larger the number of particles the more computational effort is necessary but there is also an increase in accuracy.

In Fig. 3.53, six different Independence chains are drawn, all of them generated from a standard Gaussian as the proposal distribution and the standard uniform distribution as the target function. The initial value is set to zero and the Burn-In is set to 1000. Evidently, the smaller the number of particles, the less the target function can be identified.

Fig. 3.54 shows the Uniform distribution with different numbers of samples. A comparison between the Uniform distribution and the Independence chain shows that if only a few samples in the Independence chain are used they differ too much from the desired function of interest. Anyhow, in the case that the number of particles is high enough the samples describe the target function very well.

With increasing number of dimensions it is necessary to increase the number of elements as well. By a rule of thumb there is an exponential dependence of the sample size on the dimension.

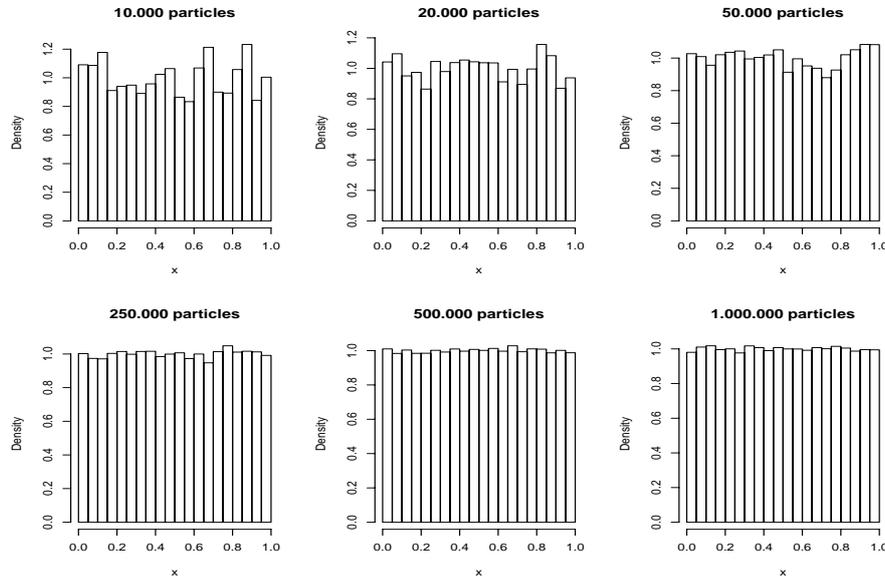


Figure 3.53: Independence Chain with Different Number of Particles

A further issue regarding the number of particles is whether one long or multiple smaller chains should be generated. A long chain has a higher probability to reach the steady state. In contrast, smaller chains can be started from different initial values. In the literature both approaches are discussed (c.f. Geyer (1992), Raftery and Lewis (1992b)) for long chains and Gelman and Rubin (1992) for several smaller chains).

If parallel processing machines are available, running several chains is much more efficient than generating one long chain. If long burning-in periods are required or the dependency structure of the chain is highly correlated, smaller chains might not reach the steady state. With the help of the above discussed testing procedures it might be possible to decide which approach is most useful.

3.4.5 Achieving Equation Fulfilment using SamPro

If data do not fulfil an equation system, a SamPro iteration might also not finish with estimates that fulfil the equation system. Although, the gap between estimates and computed values is decreased, it might still be unacceptable.

A short example illustrates the complexity of the problem. This example is presented in more detail in Section 4.3.2.

In order to simplify the notation $x \sim N(\mu, \sigma^2)$ is used instead of $x = \mu + u$ with $u \sim N(0, \sigma^2)$. For instance, $cost \sim N(80, 8^2)$.

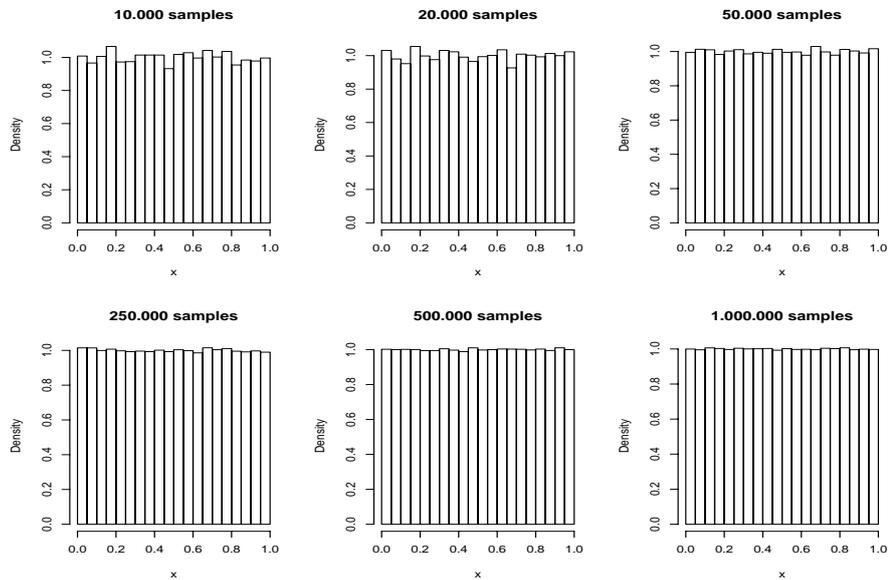


Figure 3.54: Uniform Distribution with Different Numbers of Samples

Example 3.4.3 (Equation Fulfilment). Given the following equation system:

$$\begin{aligned} \text{Transaction Volume} &= \text{Profit} + \text{Cost} \\ \text{ROI} &= \text{Profit}/\text{Capital} \end{aligned}$$

and full prior information:

$$\begin{aligned} \text{Profit} &\sim N(30, 3^2) \\ \text{Cost} &\sim N(80, 8^2) \\ \text{Capital} &\sim N(60, 6^2) \\ \text{Sales} &\sim N(100, 10^2) \\ \text{ROI} &\sim N(0.333, 0.333^2). \end{aligned}$$

Profit can be computed from both equations and, furthermore, is as prior information at hand. The three estimates for the mean of variable profit are:

$$\text{Profit} = 30 \quad (3.4.4)$$

$$\text{Sales} - \text{Cost} = 20 \quad (3.4.5)$$

$$\text{Capital} \cdot \text{ROI} = 19.98 \quad (3.4.6)$$

After a first iteration of SamPro the new estimated (adjusted) means of profit are:

$$\text{Profit} = 24.92 \quad (3.4.7)$$

$$\text{Sales} - \text{Cost} = 29.48 \quad (3.4.8)$$

$$\text{Capital} \cdot \text{ROI} = 25.01 \quad (3.4.9)$$

This reveals a convergence tendency of all three estimates of profit, cf. Fig. 3.55.

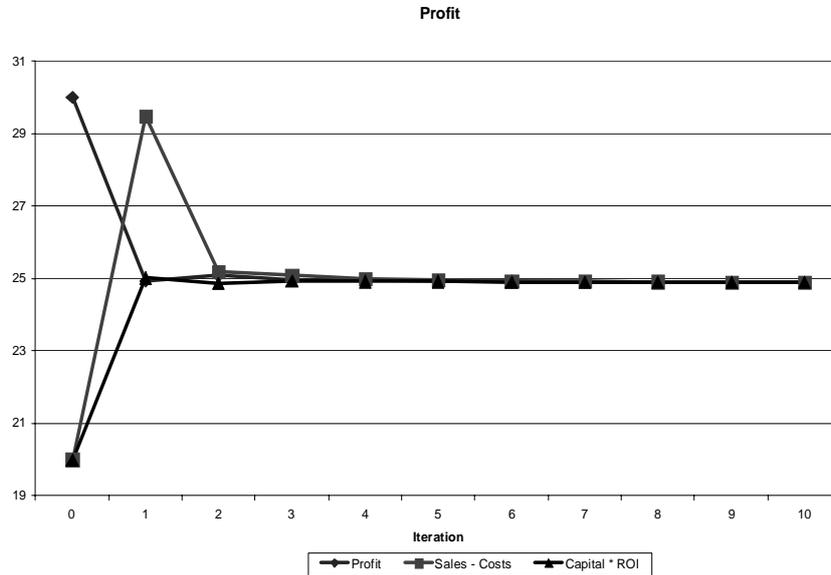


Figure 3.55: Trajectory of Profit using Prior Information and Equation System

Fig. 3.56 shows the trajectories of the estimated means for the DuPont indicators. After only three iterations a stable state is reached.

3.4.6 Outliers

An outlier is an extreme value relative to model and data. Estimates based on data sets with outliers might be misleading. Outliers can result from a different population than all other data. Hawkins (1980) defines outliers in the following way: “The intuitive definition of an outlier would be an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism”. Büning (1991) states the following two reasons for outliers: data are generated by a distribution with heavy tails or the data originated by two distributions, where one creates “good” and the other creates “bad” observations.

Outliers play an important role in the case of division of variables of an equation system. This refers to the first reason stated by Büning because division might result in heavy tail distributions. The following simple example illustrates some difficulties with outliers in the context of equation systems.

Example 3.4.4 (Outliers generated by an Equation System). The simple DuPont system as described in section 2.2.1 is used with the following data:

- Transaction Volume $\sim N(100, 50^2)$
- Cost $\sim N(80, 40^2)$

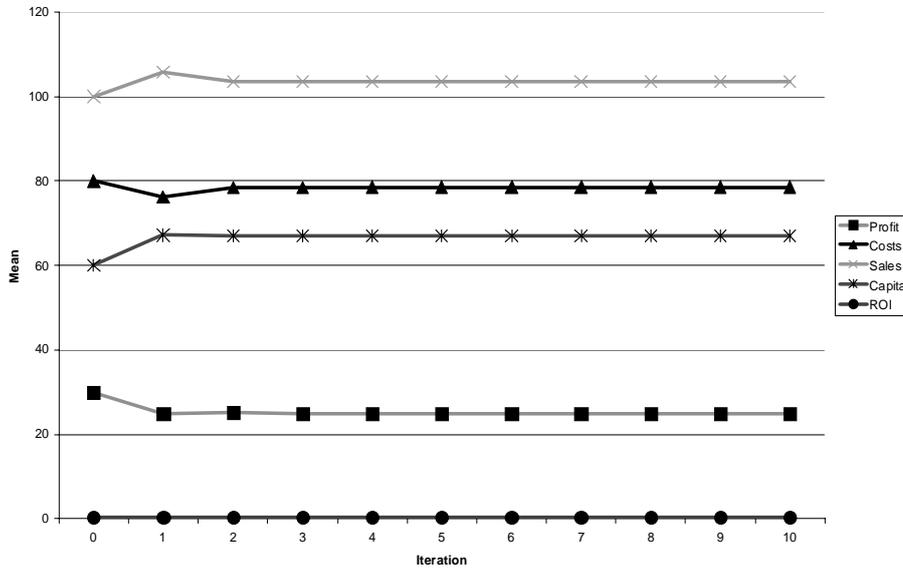


Figure 3.56: Estimate of Mean by SamPro

- Capital $\sim N(80, 40^2)$.

From the equation system Profit (P), Capital Turnover (CT), Profit Margin (PM) and ROI can be computed. The results are presented in Fig. 3.57. Some outliers are marked in the figure with crosses. These outliers result into a large variance and also influence the mean value estimation. For instance, the variance of PM is estimated by 15305. Although the median of CT is 1.2 the estimated mean is 2.3 which is 90% higher.

The treatment of outliers is part of robust methods.

Robust Methods in the Univariate Case

Robust methods can be divided into two areas: one is the detection of outliers and the second their elimination. In this section, only univariate data sets are considered. Due to their order structure multivariate data sets are more complex. They will be discussed in the next section.

In the univariate case, a sample X of size n can be seen as a vector. The vector has length n . The most common robust method is γ -trimming. $\gamma\%$ of lower values and $\gamma\%$ of upper values are eliminated from the set. In the first step, the sample X has to be ordered ($X_{[j]}$). The trimmed data set $X_\gamma = (X_{[\gamma \cdot n + 1]}, \dots, X_{[(1-\gamma) \cdot n]})$ consists of $(100 - 2 \cdot \gamma)\%$ of elements from original data set X .

The estimator of the mean is now defined as:

$$\hat{\mu}_\gamma = \frac{1}{(1 - 2\gamma) \cdot n} \sum_{r=\gamma \cdot n + 1}^{n - \gamma \cdot n} x_{[r]}. \quad (3.4.10)$$

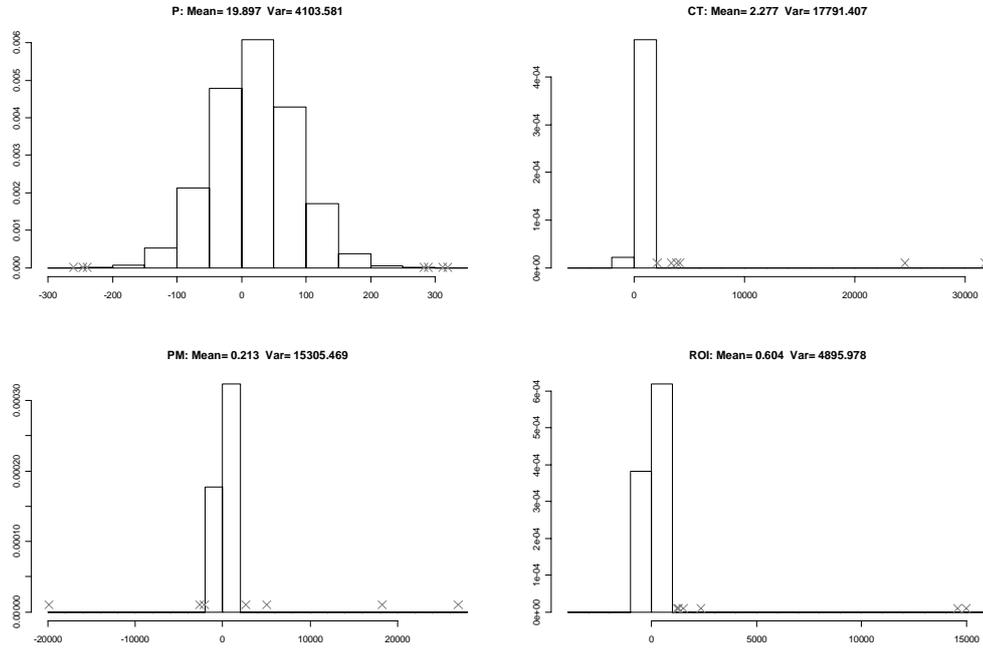


Figure 3.57: Estimated Distributions for P, CT, PM and ROI (no trimming)

The trimmed estimator for the variance is:

$$\hat{\sigma}_\gamma^2 = \frac{1}{(1-2\gamma) \cdot n} \sum_{r=\gamma \cdot n+1}^{n-\gamma \cdot n} x_{[r]} - \mu_\gamma. \quad (3.4.11)$$

Considering each variable to be pairwise independent might be the easiest way to make a robust estimation and this way univariate robustification is possible. The underlying dependency structure is not regarded in this case. This might result in in-appropriate estimates. If the γ values are very low or only one variable is of interest, then univariate trimming is easy to implement. In the case of equation systems described in Chapter 2, the dependency structure has to be taken into account in order to reduce errors of estimation. Therefore robust methods in a multivariate case are the appropriate choice, as explained in the next section.

Robust Methods in the Multivariate Case

The main problem of implementing robust methods on multivariate data is the order structure of these data.

Although Barnett (1976) states: “One cannot really order multivariate data”, he differentiates between following multivariate orderings:

- marginal ordering,

- reduced or aggregate ordering,
- partial ordering, and
- conditional or sequential ordering.

These four orderings and their respective robust approach are given in the following. The data are given as a $p \times n$ data matrix:

$$X = (X_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pn} \end{pmatrix}.$$

It is a necessary requirement that there are the same number of observations n for all p dimensions.

Marginal ordering treats the multivariate data as in the independent univariate case. All dimensions are considered separately. The ordering of each dimension gives:

$$x_{i[1]} \leq x_{i[2]} \leq \dots \leq x_{i[n]} \quad \text{for } i = 1, \dots, p$$

This is a single value oriented ordering.

For each dimension the data are trimmed as in the univariate case. Estimated means are derived from Eq. 3.4.10 as:

$$\hat{\mu}_{i\gamma} = \frac{1}{(1 - 2\gamma) \cdot n} \sum_{r=\gamma \cdot n+1}^{n-\gamma \cdot n} x_{i[r]} \quad \text{for } i = 1, \dots, p.$$

The marginal ordering estimate for the multivariate data is:

$$\hat{\mu}_{\gamma} = (\hat{\mu}_{1\gamma}, \hat{\mu}_{2\gamma}, \dots, \hat{\mu}_{p\gamma}).$$

Using marginal ordering in example 3.4.4 creates different estimates. Using for instance $\gamma = 0.05$ results in Fig.3.58. Variance and mean estimates are adjusted. For example, with marginal ordering the estimate of Capital Turnover (CT) is 1.43. The variance of Profit Margin (PM) is reduced from 15305 to 0.36.

Marginal ordering can be easily computed but at the cost of the loss of the dependency structure. Each dimension is handled separately and this might produce wrong results. In order to use the dependency structure of the given multivariate data other orderings have to be considered.

Reduced ordering is based on distance functions. Several distance measures might be useful and the choice also depends on the data. For instance, the Mahalanobis distance might be a useful distance measure. If the moments are unknown

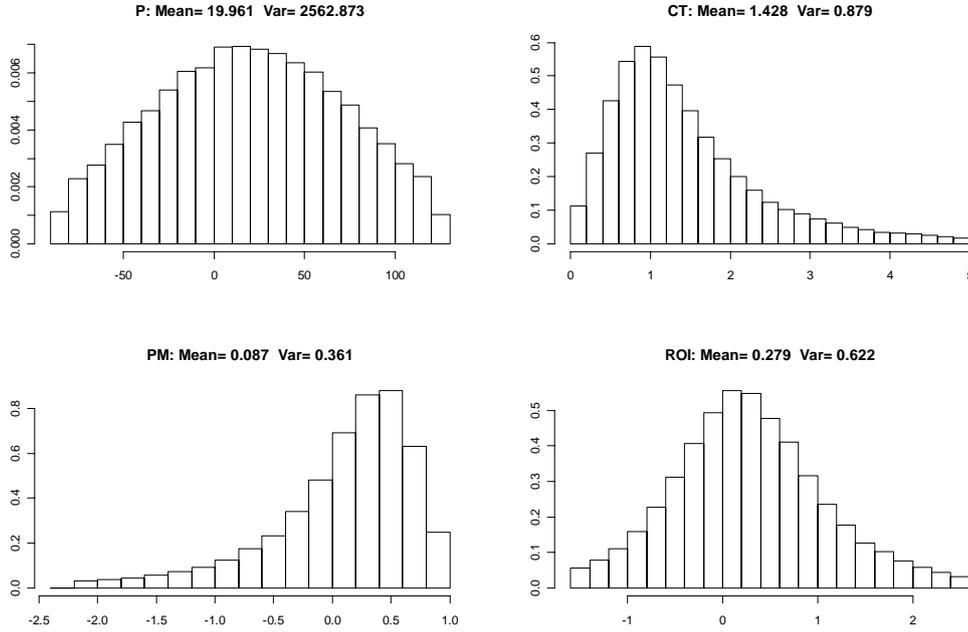


Figure 3.58: Estimated Distributions for P, CT, PM and ROI with Marginal Ordering and Trimming ($\gamma = 5\%$)

they have to be estimated. This results in an estimated Mahalanobis distance:

$$\begin{aligned}
 \check{\mu} &= \frac{1}{n} \sum_{i=1}^n x_i \\
 \check{\Sigma} &= \frac{1}{n-1} \sum_{i=1}^n (x_i - \check{\mu})(x_i - \check{\mu})' \\
 d_M(x, \check{\mu}) &= (x_i - \check{\mu})' \check{\Sigma}^{-1} (x_i - \check{\mu}) \quad (3.4.12)
 \end{aligned}$$

The d -dimensional elements are ordered corresponding to their distance measures, that means, iff

$$d_i(x_i, \check{\mu}) \leq d_j(x_j, \check{\mu}) \implies x_i \leq_M x_j$$

The order structure of the distance orientated approach is the given as:

$$\begin{array}{ccccccc}
 d_{[1]} & \leq & d_{[2]} & \leq & \dots & \leq & d_{[n]} \\
 | & & | & & & & | \\
 x_{[1]} & \leq_M & x_{[2]} & \leq_M & \dots & \leq_M & x_{[n]}.
 \end{array}$$

Which distance measure is used depends on the implementation and the structure of variables, which is not further investigated here.

Applying the Mahalanobis distance function the corresponding estimates for Example 3.4.4 are given below. The centroid is estimated by:

$$\begin{aligned} \textit{Profit} &= 19.900 \\ \textit{Capital Turnover} &= 2.277 \\ \textit{Profit Margin} &= 0.213 \\ \textit{ROI} &= 0.604 \end{aligned}$$

Trimming 5% with the highest distance measure of the multivariate data results in an histogram estimate as given in Fig. 3.59. In comparison to Fig. 3.57, the variance of PM is reduced to 15.14 and the estimated mean of CT is 1.65. With respect to the dependency structure of all involved variables, this reduction is significantly smaller than in the case of marginal ordering.

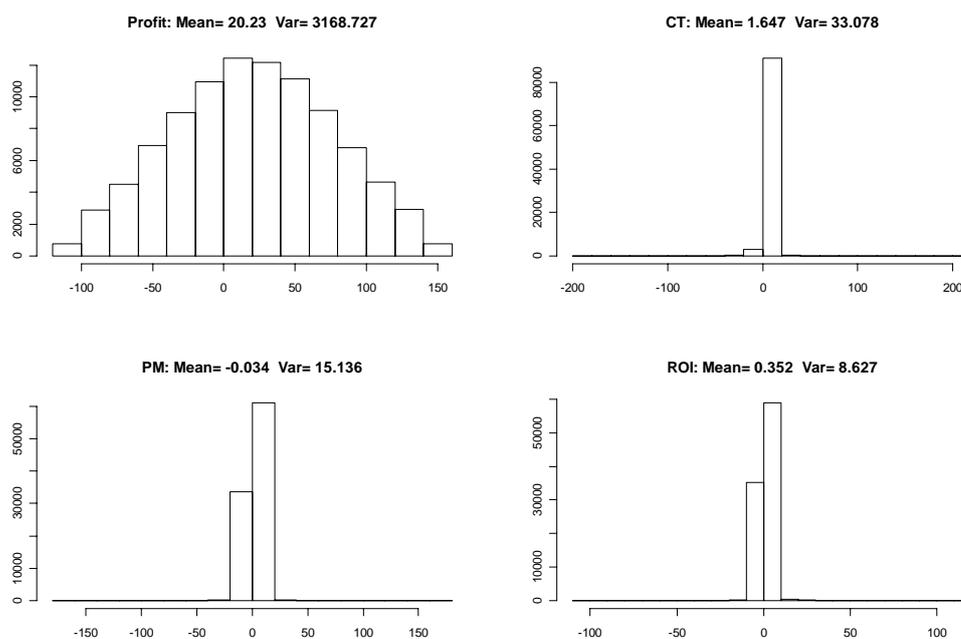


Figure 3.59: Estimated distributions for P, CT, PM and ROI with Reduced Ordering and Trimming ($\gamma = 5\%$)

Fig. 3.60 shows the function of estimated mean and variance with respect to the trimming level γ . It can be seen that a large variance reduction is achieved. High potential outliers are eliminated. However, not all variables lead to the same reduction of the estimated variance. CT for instance has the most influenced

outliers according to the Mahalanobis distance and the variance is reduced much more than the variance of Profit.

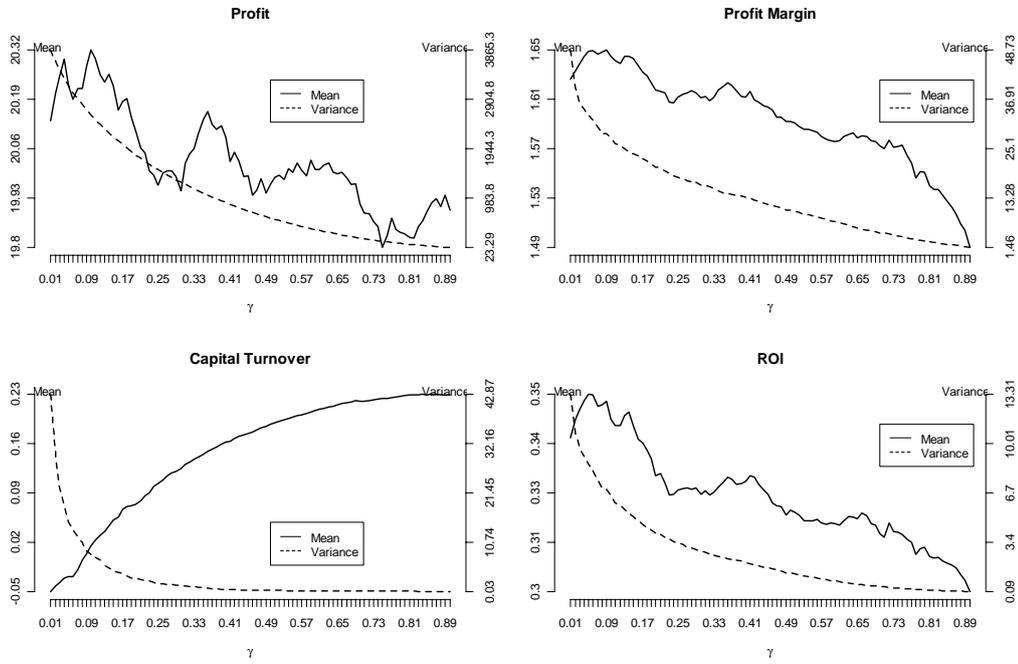


Figure 3.60: Mean and Variance as a Function of γ for P, CT, PM and ROI with Reduced Ordering

In contrast, the estimated means do not behave similarly. While CT is increasing, PM and ROI are decreasing. Profit is more fluctuating. However, the range of all estimated means of the variables lies in a small interval.

Another ordering that takes account of dependency structure uses a geometric principle. This ordering is called partial ordering. The concept of convex envelopes is used.

Definition 3.4.1 (Convex Envelope). A convex envelope is the minimum convex set that encloses all data points.

All elements of the same group have the same order. Fig. 3.61 shows the convex envelope of Profit and CT for 20 data points. The outer convex envelope is called H_0 , the second outer envelope is H_1 and so on.

This ordering inherits two main problems. First, it is not possible to fix the amount of outliers as done in the above orderings. For instance, if all data points are on a circle, these data are all outliers or none. Adding one new data point outside the circle, some of the old data points are element of H_1 and some are in the set H_0 , which contains the new data point, too.

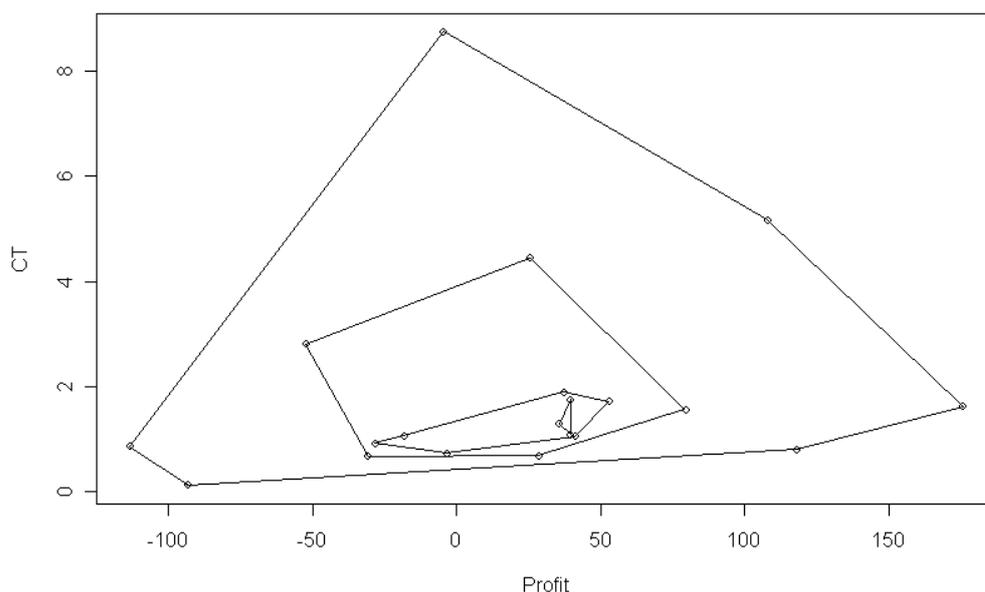


Figure 3.61: Convex Hull for Some Points of CT and P

The second problem is the complexity of algorithms that implement partial ordering. Preparata and Hong (1977) shows convex hull sets for two and three dimensional data. For data with more than three dimensions the complexity of the algorithms is high.

For these two reasons partial ordering is not considered in the rest of this study.

The last ordering in multivariate dimensionality considered here is called conditional ordering. Here, one dimension is selected as the order criteria. However, this method is inapplicable for outlier detection and useless for trimming, as stated in Büning (1991). Therefore, conditional ordering is not examined in detail.

Summing up the challenges of outlier detection and robust estimation, marginal ordering and the corresponding elimination do not account for the dependency structure and should be discarded. Partial ordering is not preferred because its implementation is very complex. Conditional ordering cannot be used for outlier detection. Reduced ordering uses distance measures as order structure and delivers acceptable results when using Mahalanobis distance. However, there are also other possible distance measures.

Chapter 4

Comparison of MCMC Simulation, Probabilistic and Possibilistic Approach

This chapter presents comparative studies and aims to evaluate the approaches described in Chapter 3: GLS approach, Fuzzy set approach and SamPro. For each approach the corresponding software is used. In the case of the GLS estimation approach the software package QR is used. QR is based on the work of Schmid (1979) and Müller (1988). The Fuzzy set approach is evaluated with the Excel Plug-In “FuzzyCalc” which is described in Müller and Lenz (2003). SamPro is implemented in MoSi which is presented in Chapter 5. The evaluation of the three approaches includes a comparison of the underlying methods.

There are a number of criteria that can be used to evaluate software tools. For the purpose of this thesis, five criteria are selected for tool evaluation. These are: (1) ease of (graphical) modelling, (2) quality of information on errors, (3) graphic visualisation of results, (4) existing restrictions in models, and (5) performance. The criteria are explained in detail here below. A synopsis is given at the end of this chapter.

Graphical Modelling

Modelling indicator systems is essential for understanding and identifying relations between indicators of an equation system. These equations represent balance or behavioural equations, as well as equations simply given by definitions. In such cases, there are no uncertainties in the model and conflicts only occur due to errors in the variables. Consequently, it can be expected that the graphic representation of a model is easy to generate. The advantage of combining mathematical formulation and graphic visualisation is that it simplifies usage.

Information on errors

Type and amount of errors in the variables depend on underlying model assumptions. In fact, limited prior knowledge restricts the area of application and potentially results in inappropriate conclusions.

Result Presentation

It should be a primary goal to ensure easy use, easy interpretation and to provide useful insight as well as accurate values. This is best realised through curves or histograms, as well as graphic presentations of meaningful estimates, such as median or confidence intervals. Whether or not an analysis is possible depends on the structure of the indicator model and given prior knowledge. Furthermore, the user should be assisted by the application rather than assist the application.

Restriction of Models

Real world indicator systems can be very large and complex. An example is the ZVEI business indicator system in Section 2.2.

Arithmetic operations are the basis of all approaches, including indicator systems. For example, behavioural, balance or definition equations are based in almost all cases on $+$, $-$, \cdot , or $/$.

Performance

In real time decision making, performance is a crucial factor. How fast computation and estimation techniques are completed and results are obtained depends on the complexity of the model (structural equations) and prior knowledge. The smaller the computational effort, the higher the number of analyses and what-if scenarios that can be run. Consequently, the quality of a decision support tool is determined by both required time and accuracy level.

4.1 Phases

Simulation can be divided into three main phases: planning, execution, and analysis. These three phases are described in detail in the following sections.

4.1.1 Simulation Planning

Simulation planning regards all information required for the simulation execution. First, a simulation model is created. It is then necessary to mark all indicators of interest. The indicators will be analysed in the third phase. Additional indicators, called intermediate indicators, might be included in the model as well, and information about these indicators is then also part of the simulation.

These intermediate indicators occur for instance as measurements that aggregate an indicator of interest or intermediate result that will be used for computing indicators of interest.

The simulation planning phase consists of the following steps:

1. indicator model creation,
2. indicator prior knowledge description,
3. simulation method selection, and
4. experimental environment parameters setting.

Indicator Model

An indicator model represents a priori relationships between the indicators. Such relationships can be balance equations, definitions or behavioural equations. A fundamental assumption of the simulation is that the relationships between indicators as well as the complete model are correct. Potentially incorrect relationships should not be taken into account.

Relationships between indicators are expressed by mathematical operations. An equation system is such a model. A further crucial assumption for reliable simulations is that all operators that connect indicators are separable, i.e. each equation can uniquely be solved for each indicator existing in the operation. Examples for operators are operations such as addition, subtraction, multiplication, or division. These are the most frequent operators in an indicator system.

Such an equation system can be expressed with mathematical formulae. Yet, with increasing equation complexity and with an increasing number of indicators, it becomes more and more complicated. It is extremely difficult to keep track of indicators that have multiple definitions, e. g. the indicator *Profit* within the DuPont system.

As stated in section 2.1.2, the model graph is a useful graphic representation. Furthermore, it is necessary for simulation and computation to establish start nodes for the computation of equations. Every node has equal rights and the

computation hierarchy is only dependent on the available information. In the next section it will be discussed how available information affects computation.

Input Data

There are three kinds of available information on indicators:

- Relationships between indicators,
- Expert knowledge about distribution of indicators, and
- Data.

Relationships between indicators are modelled in the indicator model. Mathematical formulations of equation graphs should be used in this case. An alternative solution is to use a dependency structure between indicators. Combining such a structure with expert knowledge about indicators allows generating multivariate density functions. For instance, using Bayes' theorem, full conditional distributions can be derived, c.f. Gamerman and Lopez (2006, pp. 41).

However, a univariate distribution of an indicator is needed for indicator system simulation. If the prior knowledge contains contradicting information about an indicator, the M-consistency check, as explained in Section 3.3.8 should be applied. If the result of the check is "not M-inconsistent", it should be used as prior knowledge.

Density estimations should be considered for obtaining density functions if expert knowledge is not available but sufficient data for the indicators exists. For an introduction to multivariate density estimation, see Scott (1992, 2004).

Simulation can generate the best results if the prior information is completely entered into the model. Furthermore missing values are allowed. Dependence structures related to multivariate distributions are to be considered if the generation of random numbers according to the density function is possible. This is also connected to the choice of simulation method which will be discussed in the next section.

Simulation Method

For some simulation cases there exist several different random number generation methods and an appropriate method is to be selected. As described in chapter 3, the random number generator has a significant impact on simulation time. More general methods require an increased computational effort and the generated total set of random is only partially used (Robert and Casella 2003, p. 295). There is no general method that both minimises the computational effort and is applicable to all selected density functions. This explains why a selection of simulation methods is required. However, an automatisation of this selection can be used for standard density functions.

Whenever there are algorithms available for frequently used distributions, random number generation should be performed with these algorithms. Mersenne twister, for instance, should be chosen as the PRNG for uniformly distributed values. Other common distributions require different PRNGs.

If there is no algorithm available, the Metropolis Hastings method can be applied. Gibbs Sampler, SIS or acceptance rejection methods can be used if their performance is better than the one of the general method of MH.

If the Metropolis Hastings method is chosen, a kernel selection is required. A kernel selection is another factor that affects the performance of random number generation.

The three most common kernels were be discussed in the Section 3.4. They are: Independence Chain, Random Walk, and Simulated Annealing. With regard to a number of scenarios, all three kernels outperform other existing kernels. The choice of a kernel does also have an impact on the quality of the proposal distribution. The acceptance rate of the MH-method is one possible criterion to measure performance. However, one should keep in mind that a high acceptance rate may correspond to a slower convergence (Robert and Casella 2003, p. 295).

Experimental Parameters

Experimental parameters specify the environmental procedure. The main point of interest are the simulation effort and the possibility of experiment repetition. In addition, precision measures are derived from these parameters as well.

Due to the fact that pseudo random sequences are deterministic and reproducible, it is necessary to set the initial seed an algorithm given. This seed determines the pseudo random numbers and is therefore an experimental parameter. The role of the seed is discussed in section 3.3.1.

Another crucial parameter for simulation is the number of generated random values. Increasing the number of values leads to both a higher precision of the statistical statements and an increased computational effort. For instance, a higher number of random numbers implies a higher memory usage for computing medians, leading to a longer simulation time due to more repetitions. The number of generated random values needs to be reduced the more variables are used in the simulation.

In the case of the MH algorithm, another parameter is of interest for the simulation: the Burn-In parameter as described in section 3.4.3. The Burn-In parameter increases at the same time the number of random numbers increases because the initial values will be dropped. In order to obtain the desired number of random values, more numbers have to be generated. The Burn-In parameter only increases the simulation time because all initial random numbers will be removed.

4.1.2 Simulation Experiments

Simulation experiments depend on the given model, the corresponding data, and the experimental setup. Experimental parameters define the performance of the simulation. Initialising the simulation requires to fix the random number seeds as well as the number of (desired) random numbers.

In a first step, all variables with existing prior information are generated. How the process is performed depends on the distribution and the selected simulation method. In the case of Metropolis Hastings simulation, the generated random numbers have a length of the desired random numbers plus Burn-In. After this initial simulation step, the first elements are taken out of consideration.

The next step within the simulation procedure is the separation of all equations within the model. Equations are eliminated where no prior information for LHS are given. RHS variables can then be computed with the help of the equation system.

This second step is performed by selecting the variables with more than one sample. In such a case a M-consistency check as suggested in section 3.3.8 is required in order to obtain a combined estimate per variable. A M-consistency check can either result in a merged sample with an enhanced estimate or in the discovery of a M-inconsistency issue. In case of M-inconsistency problems, the entire simulation stops and the user is notified. Otherwise, one sample is available for each variable upon which an analysis can be performed.

4.1.3 Simulation Analysis

The simulation can be started from two different results of experiments. In case that the simulation experiments show M-inconsistencies it has to be located, which part of the input data lead to this problem.

Statistic Indicators

Most decision tools based on stochastic elements use two estimators, mean and standard deviation (or variance). Both indicators can be easily computed but they inherit assumptions upon the underlying distribution of samples. In the case that the sampling distribution is a Gaussian both estimators are intuitive and correspond to the parameters of the distribution. Mean estimator of skewed distributions over or under estimated the central point. In contrast heavy tailed distributions do not have a standard deviation. For these reasons more robust estimators have to be used for reasonable interpretation. The location parameter of a univariate distribution is estimated by the median as a robust estimator. Confidence interval estimate give a cover of the range of the population parameter. The confidence level is often called $1 - \alpha$ and defines how likely the interval covers the parameter.

In Fig. 4.1 the Gaussian distribution is plotted with mean μ , standard deviation σ , and quantiles of $\alpha = 2.5\%$ and $\alpha = 97.5\%$. Note, that for a Gaussian

distribution the median is equal to mean and the quantile interval is equal to $[-1.96, 1.96] \cdot \sigma$ with $1 - \alpha = 95\%$.

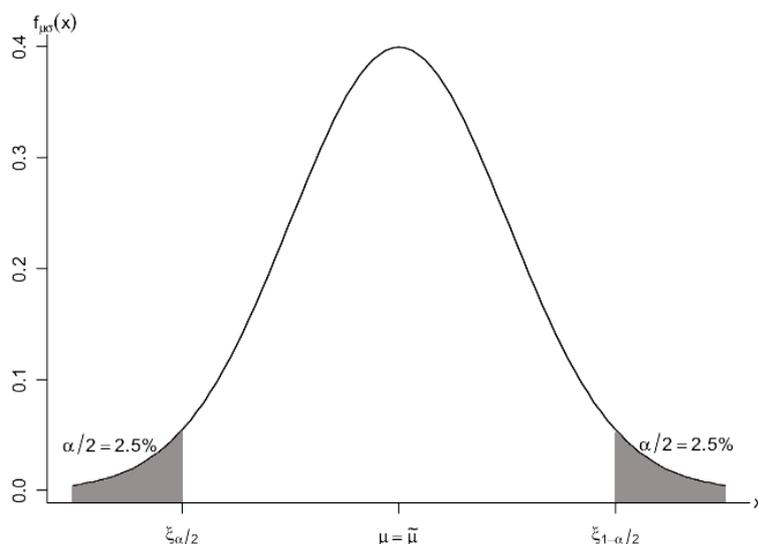


Figure 4.1: Gaussian Distribution (μ and σ^2 given)

In contrast in Fig. 4.2 a Cauchy distribution is plotted. Estimates are obtained from a sample of 10 million elements. Another sample produces different estimates in mean ($\hat{\mu}$) and standard deviation (sd) ($\hat{\sigma}$) but median and confidence intervals are quite stable. The interval $[-\hat{\sigma}, +\hat{\sigma}]$ spans more than hundred times the confidence interval.

In the case of multivariate distributions, median and confidence intervals cannot be unambiguously obtained. This is due to the fact of ordering in multivariate dimensions.

Graphical Visualisation

To get a general idea of the resulting distribution, a graphical visualisation is helpful. A histogram is a graphical representation of a sample. It shows the relative frequencies of specified categories, also called bins. These categories are non-overlapping intervals. Further graphics are Pareto chart, check sheet, control chart, cause-and-effect diagram, flowchart, and scatter diagram.

Definition 4.1.1 (Histogram). A histogram is the graphical representation of a mapping M that counts the observations that are elements of disjoint bins. For continuous data, a true density histogram is defined over a grid on the real line.

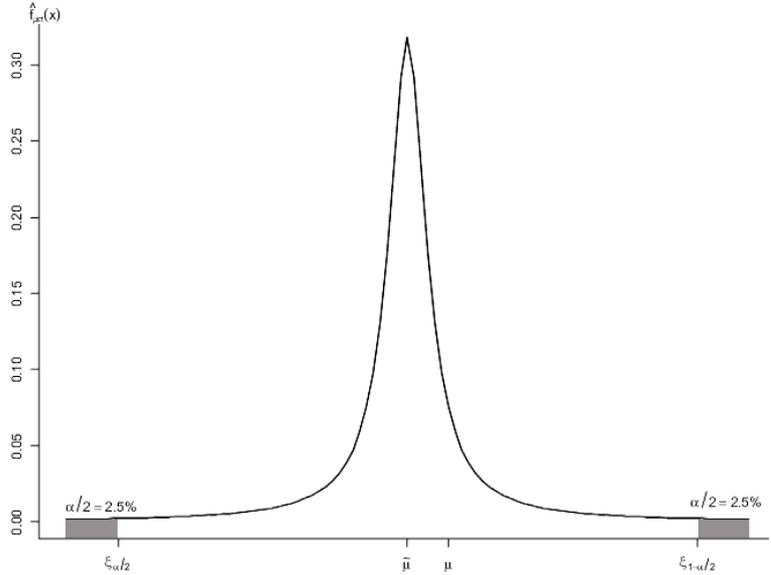


Figure 4.2: Statistical Estimators for Cauchy Distribution

The i -th bin is denoted as $B_i = [b_i; b_{i+1})$. The density estimate is given by

$$\hat{f}(x) = \frac{c(x)}{n(b_{i+1} - b_i)} \quad \forall x \in B_i. \quad (4.1.1)$$

with $c(x)$ as the number of samples that are element of bin B_i and n is the number of all samples.

Def. 4.1.1 corresponds to definition given by Scott and Sain (2005). Note, that the histogram itself is a density estimator.

There exists not a “best” partition of bins. It should be adjusted depending on the shape of the underlying distribution width and the number of bins. There exists a trade off between bandwidth and variance, cf. kernel estimation techniques, Silverman (1986).

A histogram is depicted in Fig. 4.3 of the Gaussian distribution with a sample size of 10000 elements.

Another approach is kernel density estimation. The estimated density can be plotted. There exist several methods for density estimation. A good overview of univariate density estimation is given in Silverman (1986). Multivariate density estimation is explained in Scott (1992).

Other approaches to present multivariate distributions with dependencies are correlation matrices, scatter plots, or three dimensional plots. Partial correlation graphs are a further possibility to obtain a first view on the dependency structure of the evaluated variables.

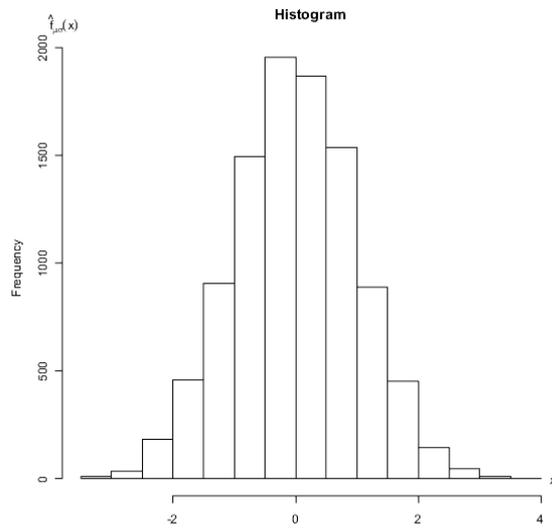


Figure 4.3: Histogram of 10000 Samples from a Gaussian Distribution

To get an overview on the fulfilment of target values for a given number of indicators Kiviat diagrams are useful. Those diagrams have a central point and each indicator has its own axis. The axes can be scaled according to their parameter values. Another possible scaling is the ratio between actual and target value. Concentric circles make the chart easy to read. To further improve the readability of such a chart, it is reasonable for indicators where a higher value denotes a worse result, this ratio can be inverted. In Section 4.4.1 Fig. 4.29 shows a Kiviat Diagram of a Balanced Scorecard.

4.2 Comparing FuzzyCalc, Quantor and SamPro

In this section a comparative study from Köppen and Lenz (2006) is presented. Three different approaches to analyse imprecise data are used. On the one hand Fuzzy arithmetic which has been detailed on in Section 3.1.3. The Microsoft Excel Plug-In “FuzzyCalc” serves as the tool for the computation. On the other hand the GLS approach as described in Section 3.2.6 and the MCMC algorithm SamPro (Algorithm 10) are used.

The comparative study is based on the DuPont system as described in Section 2.2.1. 15 different test cases are designed, see Tab. 4.1. These test cases were developed in corporation with an automobile company as a benchmark.

Table 4.1: Prior Information of 15 DuPont Test Scenarios

Case	TV	Co	Ca	P	PM	ROI
1	100 ± 5	80 ± 4	80 ± 4			
2	100 ± 10	80 ± 8	80 ± 8			
3	100 ± 50	80 ± 40	80 ± 40			
4		80 ± 4	80 ± 4	20 ± 1		
5		80 ± 8	80 ± 8	20 ± 2		
6		80 ± 40	80 ± 40	20 ± 10		
7	100 ± 10	80 ± 8	80 ± 8	20 ± 2		
8	100 ± 10	80 ± 8	80 ± 8	30 ± 3		
9	100 ± 10	80 ± 8	80 ± 8	40 ± 4		
10	100 ± 10	80 ± 8	80 ± 8			0.25 ± 0.025
11	100 ± 10	80 ± 8	80 ± 8			0.4 ± 0.04
12	100 ± 10	80 ± 8	80 ± 8			0.5 ± 0.05
13	100 ± 5	80 ± 4	80 ± 4	30 ± 1.5	0.2 ± 0.01	0.4 ± 0.02
14	100 ± 10	80 ± 8	80 ± 8	30 ± 3	0.2 ± 0.02	0.4 ± 0.04
15	100 ± 50	80 ± 40	80 ± 40	30 ± 15	0.2 ± 0.1	0.4 ± 0.2

The 15 test scenarios are divided into the following categories:

- missing values (cases 1 – 6);
- missing values and over determination for one equation (cases 7 – 12);
- complete information.

In the first six test cases imputation of missing values is needed. The effects of deviation can be studied. In scenario 7 – 12 missing values can be computed and adjustment of data is achieved, which combines a reduction of deviation on the one hand and fitting values to obtain M-consistency with the equation system on the other hand. The cases 13 – 15 are designed to study M-inconsistency. All test cases assume Gaussian distributed data for the GLS approach and the simulation. For the Fuzzy set approach the data from Tab. 4.1 is interpreted as triangular

Fuzzy membership functions with a given peak value equal to the observation. The lower and upper bounds are derived by subtracting or adding the second values. The resulting membership functions are all symmetrical.

These 15 test cases are performed in FuzzyCalc, Quantor, and with SamPro algorithm.

Transaction volume in test case 1, for instance, is represented with the triangular function where the lower bound is set to 90, upper bound is 110 and the peak value is equal to 100.

The results obtained by FuzzyCalc are given in Tab. 4.2.

They have the following properties:

- all peak values fulfil the equation system,
- a reduction of the uncertainty intervals (uncertainty decreases), and
- M-inconsistent data are pointed out.

The results can be interpreted again as Fuzzy membership functions. These functions are not necessary symmetric, see, for instance, case 8. FuzzyCalc imputes unknown values using the equation system. This can be easily seen in case 1. Comparing cases 13 to 15 the domain of uncertainty is not large enough in the first two cases to derive a M-consistent Fuzzy interval for each involved variable. Increasing the area of uncertainty, as done in case 15, enlarges the domain for M-consistent data and FuzzyCalc delivers an improved solution set.

In the following these results will be compared with the results computed by Quantor. In Tab. 4.3 the results from Quantor are given. An extract of the comparison between Quantor and FuzzyCalc is also shown in Lenz et al. (2006).

Quantor assumes Gaussian distributions with no correlation. The GLS approach has the following properties:

- mean values fulfil the equation system,
- symmetric confidence intervals with reduced intervals, and
- no M-inconsistency is revealed due to infinite interval.

The GLS approach does not reveal M-inconsistencies. This is caused by the fact that the underlying prior information is not restricted to a domain and therefore a solution exists always. It is possible to specify M-inconsistency at this point by testing prior information (uncertainty intervals) for a variable against its estimate, cf. Lenz and Rödel (1991). In case the overlap of the intervals is very small the variable should be marked as M-inconsistent.

Note, that the intervals contract and that all balance equations are fulfilled. While the probabilistic approach produces symmetric confidence intervals, the corresponding intervals using the possibilistic approach are asymmetric.

Table 4.2: FuzzyCalc Results of 15 DuPont Test Scenarios

Case	TV	Co	Ca	P
1	100 ± 5	80 ± 4	80 ± 4	20 ± 9
2	100 ± 10	80 ± 8	80 ± 8	20 ± 18
3	100 ± 50	80 ± 40	80 ± 40	20 ± 90
4	100 ± 5	80 ± 4	80 ± 4	20 ± 1
5	100 ± 10	80 ± 8	80 ± 8	20 ± 2
6	100 ± 50	80 ± 40	80 ± 40	20 ± 10
7	100 ± 10	80 ± 8	80 ± 8	20 ± 2
8	105.4 [-6.4, +4.6]	76 [-4, +7]	81.2 [-9.2, +6.8]	30.3 [-3.3, +2.7]
9	109.8 [-1.8, +0.2]	72.5 [-0.5, +1.5]	80.7 [-8.7, +7.3]	37.4 [-1.4, 0.6]
10	100 ± 10	80 ± 8	80 ± 8	20 [-3.8, +4.2]
11	105.1 [-7.2, +4.9]	76.2 [-4.2, +7.8]	79.2 [-4.2, +7.8]	29.2 [-3.3, +8.8]
12	108.5 [-4.1, +1.5]	74 [-2, +3.6]	76.5 [-4.5, +8]	34.8 [-2.4, +3.2]
13		system M-inconsistent		
14		system M-inconsistent		
15	107 [-52, +43]	79.7 [-39.7, +40.3]	74.8 [-34.8, +45.2]	27.6 [-12.6, +17.4]
Case	PM	ROI	CT	
1	0.2 ± 0.1	0.25 ± 0.12	1.25 ± 0.12	
2	0.2 [-0.18, +0.22]	0.25 [-0.23, +0.18]	1.25 [-0.23, +0.13]	
3	0.2 [-1.6, +2]	0.25 [-2, +2.5]	1.25 [-0.8, +2.5]	
4	0.2 ± 0.02	0.25 [-0.02, +0.03]	1.25 [-0.12, +0.13]	
5	0.2 ± 0.04	0.25 [-0.05, +0.06]	1.25 [-0.23, +0.28]	
6	0.2 [-0.13, +0.4]	0.25 [-0.17, +0.5]	1.25 [-0.8, +2.5]	
7	0.2 ± 0.04	0.25 ± 0.05	1.25 [-0.23, +0.28]	
8	0.29 ± 0.04	0.37 [-0.07, +0.09]	1.29 [-0.17, +0.23]	
9	0.34 ± 0.01	0.46 ± 0.06	1.36 [-0.33, +0.17]	
10	0.2 [-0.05, +0.07]	0.25 ± 0.025	1.25 [-0.23, +0.28]	
11	0.28 [-0.04, +0.11]	0.38 [-0.02, +0.06]	1.31 [-0.19, +0.23]	
12	0.32 [-0.02, +0.05]	0.46 [-0.01, +0.07]	1.4 [-0.17, +0.12]	
13		system M-inconsistent		
14		system M-inconsistent		
15	0.23 [-0.13, +0.07]	0.37 [-0.17, +0.23]	1.33 [-0.87, +2.42]	

Table 4.3: Quantor Results of 15 DuPont Test Scenarios

Case	TV	Co	Ca	P	PM	ROI	CT
1	100 ± 5	80 ± 4	80 ± 4	20 ± 6.4	0.2 ± 0.055	0.25 ± 0.082	1.25 ± 0.092
2	100 ± 10	80 ± 8	80 ± 8	20 ± 12.8	0.2 ± 0.106	0.25 ± 0.165	1.25 ± 0.191
3	100 ± 50	80 ± 40	80 ± 40	20 ± 64	0.2 ± 0.44	0.25 ± 1.027	1.25 ± 1.527
4	100 ± 4	80 ± 4	80 ± 4	20 ± 1	0.2 ± 0.012	0.25 ± 0.018	1.25 ± 0.084
5	100 ± 8	80 ± 8	80 ± 8	20 ± 2	0.2 ± 0.024	0.25 ± 0.038	1.25 ± 0.176
6	100 ± 41	80 ± 40	80 ± 40	20 ± 10	0.2 ± 0.156	0.25 ± 0.305	1.25 ± 1.449
7	100 ± 6	80 ± 6.29	80 ± 8	20 ± 1.976	0.2 ± 0.022	0.25 ± 0.038	1.25 ± 0.162
8	100 ± 6	76.3 ± 6.35	80 ± 8	29.5 ± 2.9	0.279 ± 0.029	0.368 ± 0.056	1.322 ± 0.17
9	111 ± 7	72.9 ± 6.42	80 ± 8	38.2 ± 3.8	0.344 ± 0.034	0.478 ± 0.073	1.389 ± 0.177
10	100 ± 6	80 ± 6.3	80 ± 8	20 ± 2.76	0.2 ± 0.028	0.25 ± 0.025	1.25 ± 0.147
11	107 ± 7	75.8 ± 6.5	78 ± 8	30.7 ± 4.27	0.29 ± 0.037	0.39 ± 0.04	1.36 ± 0.152
12	110 ± 4	73.5 ± 6.6	77 ± 8	36.7 ± 5.18	0.33 ± 0.042	0.479 ± 0.05	1.44 ± 0.158
13	110 ± 3	84.7 ± 2.8	72 ± 3	25.57 ± 0.9	0.23 ± 0.007	0.355 ± 0.016	1.53 ± 0.073
14	110 ± 6	84.7 ± 5.5	72 ± 6	25.6 ± 1.8	0.232 ± 0.0015	0.355 ± 0.033	1.532 ± 0.152
15	110 ± 31	84.4 ± 27.7	72 ± 31	25.96 ± 9.2	0.233 ± 0.24	0.355 ± 0.24	1.519 ± 1.136

The assumptions and interpretation of both approaches are rather different. While the probabilistic approach needs a Gaussian regime for mathematical convenience of testing, the possibilistic approach considers membership functions and ignores any assumption about dependencies between variables. Due to a theorem of Dubois and Prade (2006), the benchmark study illustrates, that possibility measures are an upper bound of probability measures. The question arises which effects are caused by the basic assumption of Gaussian distributions. This is investigated by a comparative study in Section 4.3.

Next, the model computed by the SamPro algorithm will be presented. These results will be compared to the two other result sets.

In Tab. 4.4 the results of the first iteration of SamPro algorithm are given.

These are the estimated median values and quantiles at $\alpha/2 = 2.5\%$ and $1 - \alpha/2 = 97.5\%$. The median is a more robust estimate and equal to the median in the case of a symmetric distribution. Therefore, the median corresponds to the Quantor results of mean and will be compared to it. For non-symmetric distributions quantile estimates are the better choice than the sd which is suitable for symmetrical distributions. If the standard deviation given by Quantor is multiplied by 1.96 this results in quantile estimates at level $\alpha = 5\%$.

Note, that compared to FuzzyCalc the cases 12 and 13 are also stated as M-inconsistent. Due to the fact that probability mass is very small in the area result obtained by Quantor, a sample is not obtained in this domain and so the system is stated as M-inconsistent. In the first view, another distressing result is that the estimated medians using SamPro do not always fulfil the equation system. This is further discussed in Section 3.4.5.

Several iterations of the SamPro algorithm ensure equation fulfilment but at the same time increase the computational effort. Due to the definition of M-inconsistency for the SamPro algorithm it can be ensured that M-inconsistency is revealed after the first iteration. That means that higher computational effort is only spent for M-consistent cases to obtain equation fulfilment.

Case 11 illustrates that after the first iteration, the gap between estimated profit and the equation system, using *transaction volume – cost*, *profit margin · transaction volume* and *roi · capital*, is about 50%. After the second iteration this gap is reduced to 8%. The result of iteration 10 of case 11 has a gap of less than 1‰.

Comparing the SamPro results with the FuzzyCalc results, the estimates cover a smaller interval. This corresponds to the results of Dubois and Prade (1992), Dubois et al. (2004), Dubois and Prade (2006). To compare both approaches in a fair way the prior information used in SamPro is the same triangular function as used by FuzzyCalc. These results can be also found in Hausmann (2004).

In the following the results are compared in more detail.

For results on imputation case two is selected. The results of Quantor compared to SamPro for linear equation are equal, see *Profit* as an example. FuzzyCalc computes the same central point but the uncertainty interval is larger, for *Profit* FuzzyCalc computes ± 18 compared to ± 12.8 . For a non-linear equation

Table 4.4: SamPro first Iteration Results of 15 DuPont Test Scenarios

Case	TV		Cost		Capital		Profit	
	2.5%	50%	2.5%	50%	2.5%	50%	2.5%	50%
1	90.2	100.0	72.2	80.0	72.2	80.0	7.4	20.0
2	80.4	100.0	64.3	80.0	64.3	80.0	-5.1	20.0
3	1.9	100.0	1.6	80.0	1.7	80.0	-105.6	20.0
4	91.9	100.0	72.2	80.0	72.2	80.0	18.0	20.0
5	83.8	100.0	64.3	80.0	64.3	80.0	16.1	20.0
6	19.2	100.0	1.6	80.0	1.7	80.0	0.4	20.0
7	87.5	100.0	67.6	80.0	64.3	80.0	16.1	20.0
8	93.0	105.8	63.8	76.3	64.3	80.0	23.8	29.5
9	99.4	111.9	60.0	72.4	64.3	80.0	35.7	39.5
10	80.4	100.0	64.3	80.0	64.6	80.0	14.8	19.9
11	80.4	100.0	64.3	80.0	63.1	78.2	22.8	30.5
12	80.4	100.0	64.3	80.0	61.6	76.7	27.6	36.6
13	M-inconsistent							
14	M-inconsistent							
15	43.9	104.2	14.1	76.3	19.7	70.4	5.1	22.4
Case	PM		ROI		CT		97.5%	
	2.5%	50%	2.5%	50%	2.5%	50%	2.5%	50%
1	0.080	0.200	0.093	0.250	0.412	1.088	1.250	1.437
2	-0.061	0.200	-0.064	0.250	0.584	0.943	1.250	1.657
3	-4.142	0.231	-2.742	0.237	4.074	-0.575	1.201	8.033
4	0.179	0.200	0.218	0.250	0.287	1.102	1.250	1.422
5	0.159	0.200	0.189	0.250	0.331	0.972	1.250	1.627
6	-0.011	0.198	-0.115	0.240	1.607	-0.205	1.205	7.864
7	0.151	0.200	0.189	0.250	0.331	0.943	1.250	1.657
8	0.226	0.300	0.283	0.375	0.497	0.943	1.250	1.657
9	0.325	0.400	0.407	0.500	0.633	0.943	1.250	1.657
10	-0.061	0.200	0.202	0.250	0.298	0.943	1.250	1.657
11	-0.061	0.200	0.314	0.391	0.468	0.943	1.250	1.657
12	-0.061	0.200	0.384	0.479	0.574	0.943	1.250	1.657
13	M-inconsistent							
14	M-inconsistent							
15	0.061	0.211	0.099	0.352	0.684	-0.575	1.201	8.033

the results of PM are selected. SamPro estimates a skewed distribution which can be seen at the difference of the quantile distance. This SamPro result coincides with the FuzzyCalc result which also states non-symmetry.

For results on imputation and correction case 8 is selected. For a linear equation $Profit$ is selected again. SamPro estimates the same median as the Quantor result for mean. The skewness is negligible. FuzzyCalc computes quite different values due to the different approach. For the non-linear case PM is selected again. All three approaches compute or estimate different central values. Because of the fact that the distribution is skewed the median estimated by SamPro is better than the mean value computed by Quantor. The difference of peak computed by FuzzyCalc and median is smaller.

Summing up both cases SamPro results are comparable to the Quantor results in the linear case. Non-linear equation lead to skewed or heavy-tail distributions which require robust estimates.

4.3 Effects of Nonlinearity, Skewness, and Correlation

In this section the effects of skewness and prior information on dependency structures are investigated. Due to the fact that the calculus of Fuzzy set theory cannot handle dependency structures this section only focuses on comparison of Quantor and SamPro.

In first part of this section effects of correlation for an addition equation and a multiplication equation are separately investigated. This is followed by a comparison study of Quantor and SamPro.

Exact estimates as proposed in Lenz and Rödel (1991) can be compared with simulation in the case of linear equations. As an example take:

$$\textit{Transaction Volume} = \textit{Cost} + \textit{Profit} \quad (4.3.1)$$

Prior information is given for all three indicators as:

$$\begin{aligned} \textit{Transaction Volume} &\sim N(100, 10^2) \\ \textit{Cost} &\sim N(80, 8^2) \\ \textit{Profit} &\sim N(30, 3^2) \end{aligned}$$

Note, that the given prior knowledge does not fulfill the balance equation (4.3.1). Using model information and prior knowledge leads to the following matrices as stated in Section 3.2.6:

$$H = [1 \ 1] \quad P_{P \ C_o} = \begin{bmatrix} 64 & \textit{Cov}_{P \ C_o} \\ \textit{Cov}_{P \ C_o} & 9 \end{bmatrix} \quad R_{TV} = [100]. \quad (4.3.2)$$

$\textit{Cov}_{P \ C_o}$ is functional dependent upon the correlation parameter. The comparison of the estimated real values ξ and ζ as computed by Equation 3.2.2 and Equation 3.2.3 is done with simulation results using 1000000 particles per indicator.

In Fig. 4.4 the estimated values for Profit against the correlation between Profit and Cost are shown. Increasing correlation between Profit and Cost reduces the estimated value for Profit. The simulation results are very close to the computed real values. Increasing the number of samples reduces the deviation further.

As a first conclusion from this comparison it can be assumed that SamPro algorithm achieves same results as the theoretical approach for linear equations.

In the next comparison the equation operator is changed to multiplication. As an example a DuPont equation is taken again:

$$\textit{Profit} = \textit{ROI} \cdot \textit{Capital} \quad (4.3.3)$$

Prior information is given for all three indicators as:

$$\begin{aligned} \textit{Profit} &\sim N(30, 3^2) \\ \textit{ROI} &\sim N(0.4, 0.04^2) \\ \textit{Capital} &\sim N(80, 8^2) \end{aligned}$$

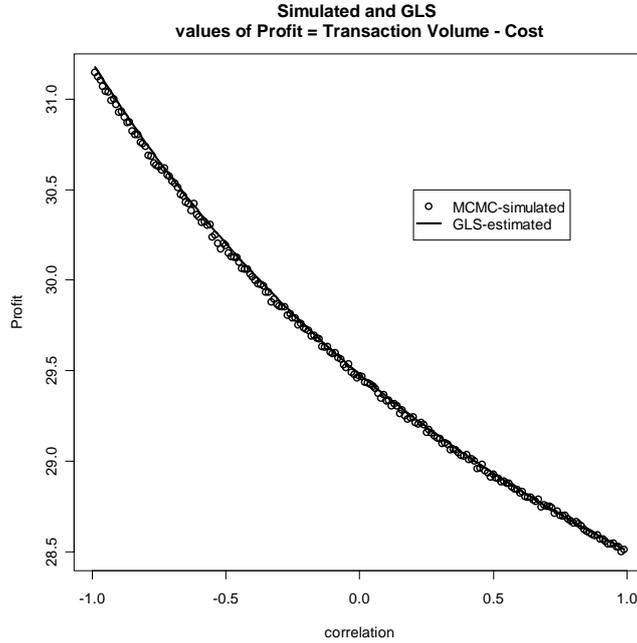


Figure 4.4: Exact and Simulated Estimates for Profit

Note, that the given prior knowledge does not fulfill the balance equation (4.3.3).

As multiplication is a nonlinear operation, the relation $ROI \cdot Capital$ must be approximated with a first order Taylor series. The following specification is used:

$$H = [80 \ 0.4] \quad P_{ROI, Ca} = \begin{bmatrix} 0.04^2 & Cov_{ROI \ Ca} \\ Cov_{ROI \ Ca} & 64 \end{bmatrix} \quad R_P = [9]. \quad (4.3.4)$$

The comparison of the estimated real values ξ and ζ as computed by Equation 3.2.2 and Equation 3.2.3 is done with SamPro simulation results using 1000000 particles per indicator as before.

In Fig. 4.5 the estimated values for Profit depend upon the correlation between Profit and Cost are shown. Increasing correlation between Profit and Cost reduces the estimated value of Profit. The simulation results lie below the computed estimates. This can be interpreted as a result of the Taylor approximation, i.e., the computed values are overestimated and can be justified by the fact that the Taylor approximation is taken at point $\begin{pmatrix} ROI \\ Capital \end{pmatrix} = \begin{pmatrix} 0.4 \\ 80 \end{pmatrix}$ although the computed estimates for both are smaller. Achieving a better outcome of computed values for profit requires an adjustment at the point where the Taylor approximation takes place. However this changes the method and will not be further discussed here.

As a further conclusion from this comparison it can be assumed that SamPro algorithm achieves better results than the GLS approach due to the Taylor

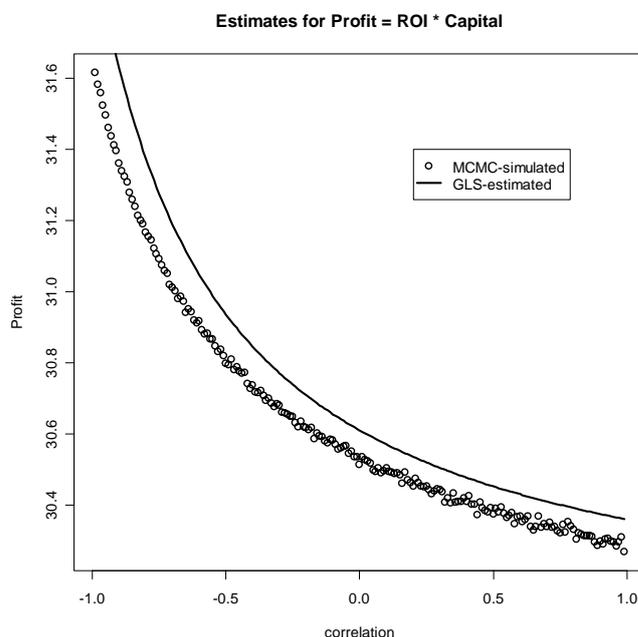


Figure 4.5: Estimates for Profit

approximation.

In the following several experiments are performed. The first two scenarios use the following equation system M :

$$\begin{aligned} \text{Transaction Volume} &= \text{Profit} + \text{Cost} \\ \text{ROI} &= \text{Profit}/\text{Capital} \end{aligned} \quad (4.3.5)$$

Evidently, there are two endogenous and three exogenous random variables. The first equation represents a linear and the second a non-linear equation. The reduction to two equations makes it easier to study effects of correlation and skewness. To study the effects of correlation and skewness in more complex systems, experimental group C uses the DuPont system with four equations and seven variables.

In order to simplify the notation $x \sim N(\mu, \sigma^2)$ is used instead of $x = \mu + u$ with $u \sim N(0, \sigma^2)$. For instance, $\text{cost} \sim N(80, 8^2)$. The mean is either estimated from the observations or is known. The variance is also assumed to be a-priori known. The distributions considered include the Normal (Gaussian), Exponential, Gamma and the Dirichlet distribution. MSN as described by Azzalini and Dalla Valle (1996) is also used in some experiments. The absolute values of correlation coefficients between pairs of variables can vary from 0.7, 0.6, 0.4 to 0.0.

Scenarios are given the two-equation model M , i.e., $\text{sales} = \text{profit} + \text{cost}$

as a linear equation and $ROI = profit/cost$ as a nonlinear relation. In all of our experiments with up to 2.5 million replications each, it is assumed that a realisation of the random variables *profit*, *cost* and *capital* is at hand.

Moreover, the types of distributions are varied. This implies that at least the mean of the various distributions can be determined. Furthermore, prior information is available about the standard deviation or variance of the measurement errors. The other two variables, i.e., sales and ROI, are handled in experimental group A as variables with missing values (null values) and later in group B as (correctly or noisy) observed values. If missing values of variables exist, they must be estimated, i.e., imputed.

In experimental group A the first three experiments analyse the effect of skewed distributions compared with Gaussian distributions if all variables are not correlated. The next two experiments separately investigate the effects of cross-correlation. Finally, the interaction between non-normality and correlation is of concern.

In experimental group B the data set is complete and Gaussian distributions are assumed, that means no missing values exist. The effects of negative, zero and positive correlation are studied for two cases: The measurement of the variables fulfill (“M-inconsistent variables”) or do not fulfill the balance equation system.

4.3.1 Experimental Group A

Effects of non-normality or correlation in the case of missing values are studied in this group.

Scenario 1: Normality; no correlation

Specification of the distributions:

$$\begin{aligned} Profit &\sim N(20, 2^2) \\ Cost &\sim N(80, 8^2) \\ Capital &\sim N(60, 6^2) \end{aligned}$$

Missing values: Sales, ROI

Result:

The MCMC simulation and the GLS estimation result for sales as part of a linear relation are compatible with respect to mean and sd. The same is true for the imputation of ROI as a nonlinear relationship. Note, that the Gaussian hypothesis is valid about the distribution of all variables.

In Fig. 4.6 the results for the imputed data are given. The histogram represents the simulation result whereas the solid line shows the result obtained using Quantor.

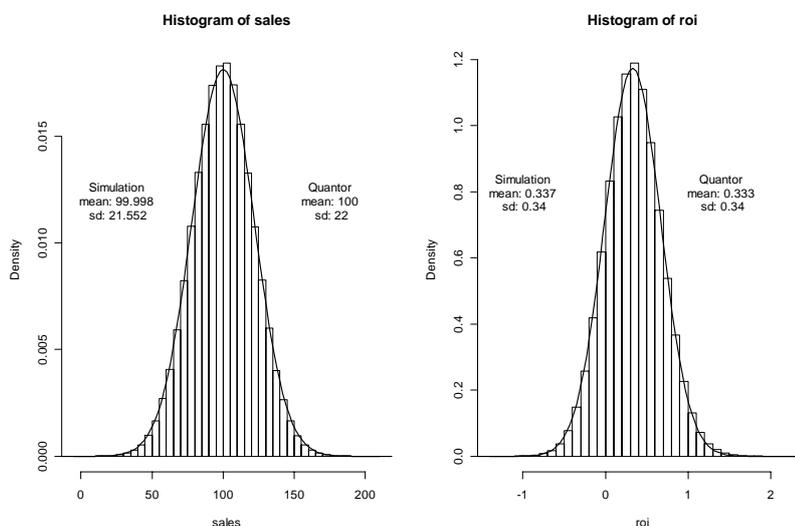


Figure 4.6: Normal Distributed Variables with no Correlation

Scenario 2: Effect of Skewness; no correlation**Specification of the distributions:**

$Profit \sim Exp(1/20)$ vs. $N(20, 20^2)$ [Simulation vs. GLS approach]

$Cost \sim Gamma(8, 0.1)$ vs. $N(80, 28^2)$

$Capital \sim Gamma(15, 0.25)$ vs. $N(60, 8.6^2)$

Missing values: Sales, ROI

Result:

In Fig. 4.7 the results for the imputed data are given. In the linear case the mean and the standard deviation are similar as the experiments for the variable sales show. In the nonlinear case the mean and the standard deviation (sd) are overestimated by about 15%. This overestimation increases with the increase of skewness in the prior information.

Scenario 3: Normality; negative cross-correlation**Specification of the distributions:**

$Profit \sim N(20, 2^2)$

$Cost \sim N(80, 8^2)$

$Capital \sim N(60, 6^2)$

Correlation used for simulation:

$$\rho(Profit, Cost) = -0.7$$

$$\rho(Profit, Capital) = -0.7$$

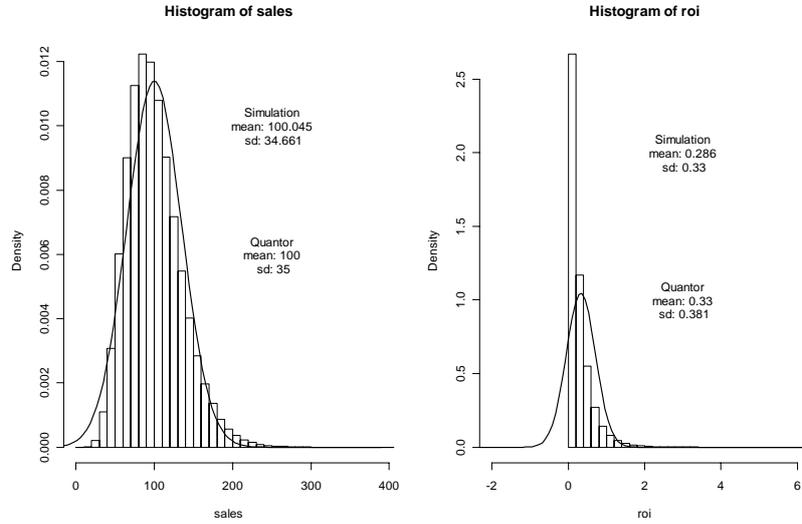


Figure 4.7: Effect of Skewed Distributions of all (Observed) Variables, no Correlation

Missing values: Sales, ROI

Result:

In Fig. 4.8 the results for the imputed data are shown. While the means are nearly equal the standard deviations differ between +18% and -16%. Quantor overestimates sd in the linear case and underestimates sd in the nonlinear equation.

Scenario 4: Normality; positive cross-correlation

Specification of the distributions:

$$Profit \sim N(20, 2^2)$$

$$Cost \sim N(80, 8^2)$$

$$Capital \sim N(60, 6^2)$$

Correlation used for simulation:

$$\rho(Profit, Cost) = 0.7$$

$$\rho(Profit, Capital) = 0.7$$

Missing values: Sales, ROI

Result:

Although the positive sign of the correlation coefficients is contra intuitive from a manager's point of view, it is used here more formally as an opposite case to negative correlation in scenario 3. In Fig. 4.9 the results for the imputed data are shown. While the means are nearly about the same values, the percentage

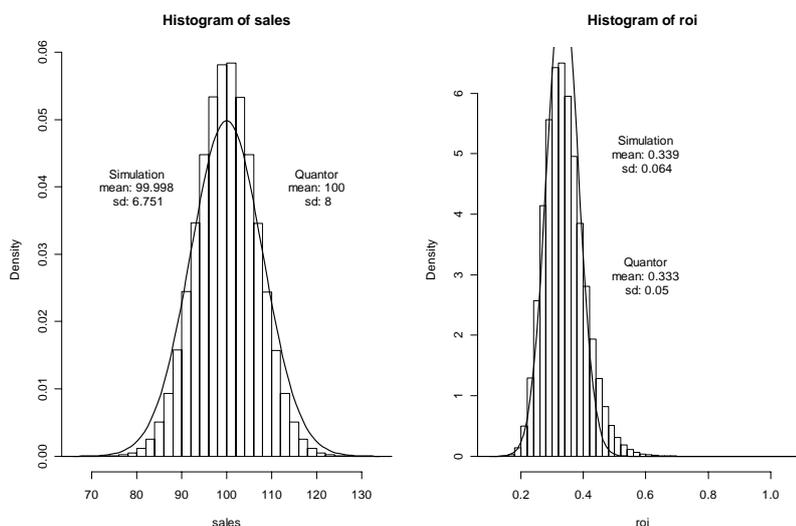


Figure 4.8: Normality and Negative Correlation ($\rho = -0.7$)

of differences changes sign: -19% for sales vs. +16% for ROI. Compared to scenario 3 the divergent estimation of sd is underestimated in the linear case and overestimated in the nonlinear case. This is quite intuitive because of

$$\begin{aligned} \text{Var}(X + Y) &= \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y) \\ &= \sigma^2(X) + \sigma^2(Y) + 2\rho(X, Y)\sigma(X)\sigma(Y) \end{aligned}$$

and the fact that the GLS approach does not make use of correlation.

Scenario 5: Skewness; negative cross-correlation

Specification of the distributions:

$$\begin{aligned} (\text{Profit}, \text{Cost}, \text{Capital}) &\sim \text{Dir}(30, 40, 8) \text{ vs.} \\ \text{Profit} &\sim N(20, 2.8^2) \\ \text{Cost} &\sim N(80, 8.8^2) \\ \text{Capital} &\sim N(60, 20^2) \end{aligned}$$

Correlation used for simulation: This correlation is imposed by the Dirichlet (Dir) distribution with the above given parameters.

$$\begin{aligned} \rho(\text{Profit}, \text{Cost}) &= -0.8 \\ \rho(\text{Profit}, \text{Capital}) &= -0.3 \end{aligned}$$

Missing values: Sales, ROI

Result:

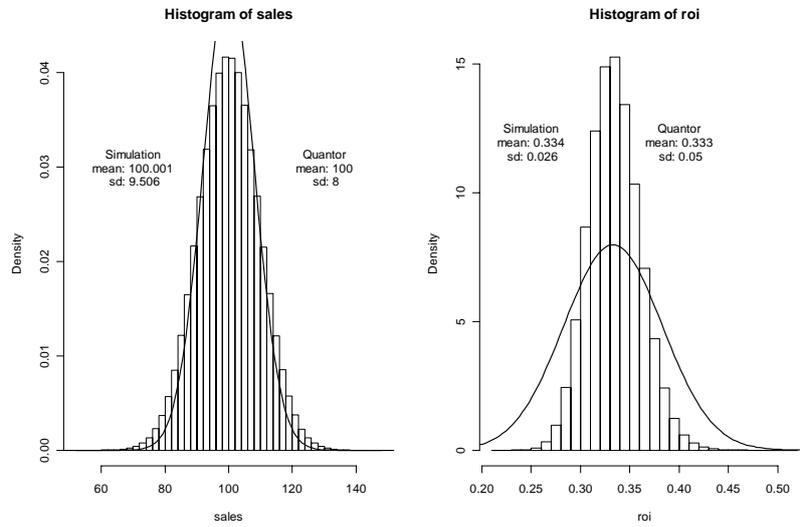


Figure 4.9: Normality and Positive Correlation ($\rho = 0.7$)

In Fig. 4.10 the results for the imputed data are shown. While the means of variable 'sales' are identical this is not true for ROI. The GLS approach over-estimated the simulated (exact) values about 12%. Quite opposite, the standard deviation of sales is over-estimated by GLS by about 65% while sd of ROI is about the same.

Scenario 6: Skewness; positive cross-correlation

Specification of the distributions:

$$\begin{aligned}
 & (Profit, Cost, Capital) \\
 & \sim MSN \left((2, 8, 6), \begin{pmatrix} 1 & 0.6 & 0.7 \\ 0.6 & 1 & 0.2 \\ 0.7 & 0.2 & 1 \end{pmatrix}, (2500, 80, 300) \right) \text{ vs.} \\
 & Profit \sim N(20, 4.3^2) \\
 & Cost \sim N(80, 8.3^2) \\
 & Capital \sim N(60, 7.3^2)
 \end{aligned}$$

Correlation used for simulation: This correlation is imposed by the Multivariate Skew Normal (MSN) distribution with the above given parameters. Note, that the parameterisation is adopted from Azzalini and Capitanio (1999).

$$\begin{aligned}
 \rho(Profit, Cost) &= 0.4 \\
 \rho(Profit, Capital) &= 0.5
 \end{aligned}$$

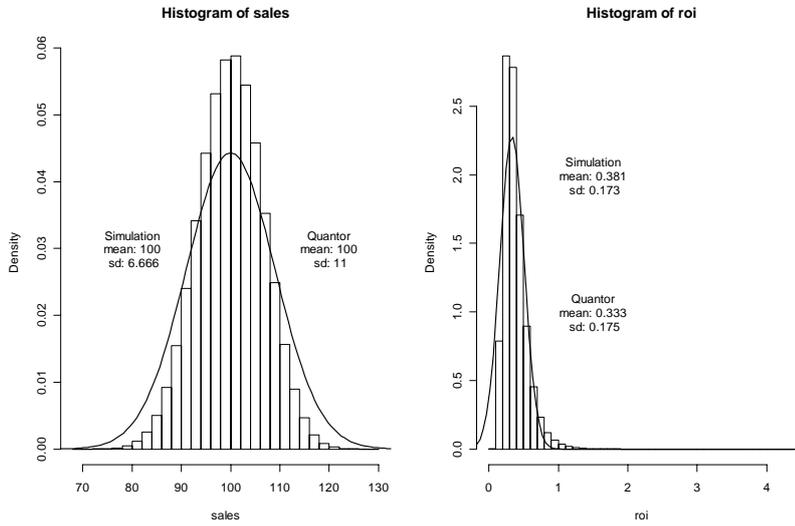


Figure 4.10: Skewness and Negative Correlation

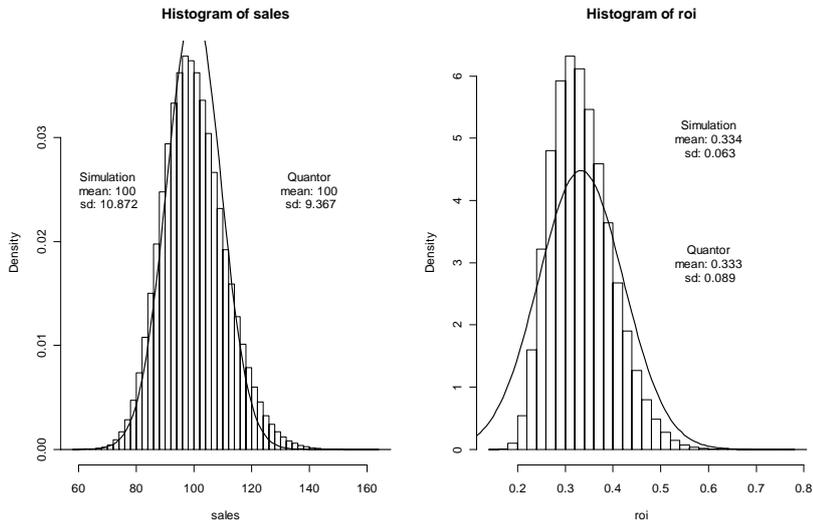


Figure 4.11: Skewness and Negative Correlation

Missing values: Sales, ROI

Result:

In Fig. 4.11 the results for the imputed data are shown. While the means are nearly about the same for both variables this is not true for the sd. For sales simulation results $sd = 10.87$ and $sd' = 9.4$ is the result under the Gaussian assumption. The corresponding values for ROI are $sd = 0.063$ vs. $sd' = 0.069$.

4.3.2 Experimental Group B

In this group two scenarios are studied given full prior information. A Gaussian regime with correlation is assumed.

Scenario 1: M-consistent prior information

Specification of the distributions:

$$\begin{aligned} Profit &\sim N(20, 2^2) \\ Cost &\sim N(80, 8^2) \\ Capital &\sim N(60, 6^2) \\ Sales &\sim N(100, 10^2) \\ ROI &\sim N(0.333, 0.333^2) \end{aligned}$$

Note, that as above the distribution of profit is threefold determined by the prior $N(20, 2^2)$, by equation $Profit = Sales - Cost$ and by equation $Profit = ROI \cdot Capital$.

Missing values: no

Correlation Matrix is specified as: $\rho \in \{-0.4, 0.0, 0.4\}$. In order to parsimoniously specify the correlation matrix R , it depends upon only one parameter, ρ . This expert knowledge might vary from company or business sector and represents in this example only one possible specification out of many. The lower and upper bounds are necessary to ensure a positive definite correlation matrix R .

$$R_\rho = \begin{pmatrix} & Profit & Cost & Capital & Sales & ROI \\ Profit & 1 & \rho & 0 & -\rho & -\rho \\ Cost & \rho & 1 & 0 & -\rho & \rho \\ Capital & 0 & -\rho & 1 & 0 & \rho \\ Sales & -\rho & -\rho & 0 & 1 & -\rho \\ ROI & -\rho & \rho & \rho & -\rho & 1 \end{pmatrix}$$

Results:

The variance of cost, capital, sales and ROI is proportional to ρ . The variance of the estimated profit is non monotonic in ρ and has its maximum at $\rho = 0.2$. The means of all variables are more or less constant.

Scenario 1 is closed by presenting three three-dimensional scatter plots showing the simulated values of the variable profit determined from the prior distribution and the two RHS of the model equations for $\rho \in \{-0.4, 0.0, 0.4\}$.

Table 4.5: Means and Standard Deviations of all observed and simulated variables, Gaussian distributions, no correlation, no missing values, M-consistent observations.

Variable	Prior		Posterior	
	Mean	Sd	Mean	Sd
Profit	20	2	19.87	1.58
Cost	80	8	79.95	6.29
capital	60	6	59.81	4.89
sales	100	10	99.96	6.37
ROI	0.33	0.33	0.33	0.03

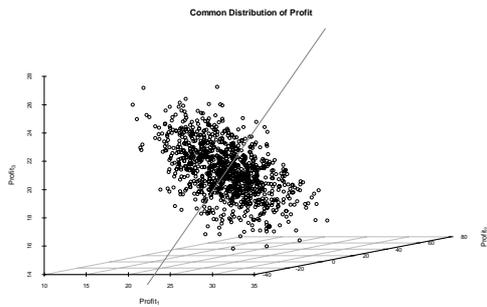


Figure 4.12: Scatter Plot of Simulated Profit Values for $\rho = -0.4$

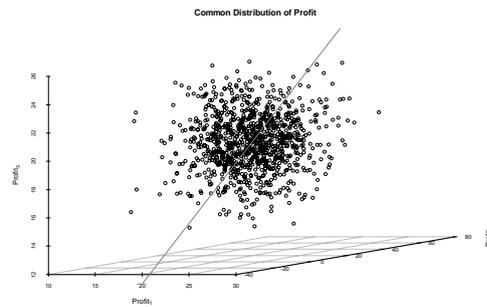


Figure 4.13: Scatter Plot of Simulated Profit Values for $\rho = 0$

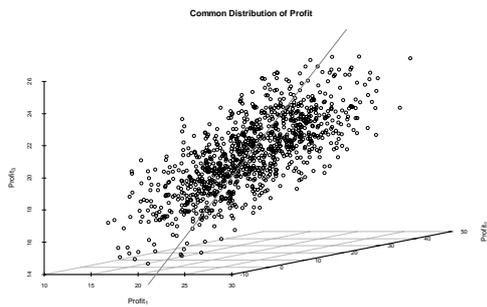


Figure 4.14: Scatter Plot of Simulated Profit Values for $\rho = 0.4$

Table 4.6: Simulation Results after 5 SamPro iterations with M-consistency ($\rho = 0$)

Variable	Prior		Posterior	
	Mean	Sd	Mean	Sd
profit	30	3	24.93	0.85
cost	80	8	78.51	1.84
capital	60	6	67.03	2.33
sales	100	10	103.46	1.84
ROI	0.333	0.333	0.372	0.013

Scenario 2: M-Inconsistent prior information

Specification of the distributions:

$$Profit \sim N(30, 3^2)$$

$$Cost \sim N(80, 8^2)$$

$$Capital \sim N(60, 6^2)$$

$$Sales \sim N(100, 10^2)$$

$$ROI \sim N(0.333, 0.333^2)$$

Note, as in scenario 1 the distribution of profit is threefold determined by the prior $N(30, 3^2)$, and by the equations $profit = sales - cost$ and $profit = ROI * capital$. The mean and standard deviation of profit are increased from $N(20, 2^2)$ to $N(30, 3^2)$. This implies M-inconsistency of the observed values (means) of $profit = sales - cost$ and $profit = ROI \cdot capital$.

Missing values: no

Correlation Matrix as specified with $\rho \in \{-0.4, 0.0, 0.4\}$.

Results:

The variance of cost, capital, sales and ROI is proportional to ρ . The variance of profit is non monotonic and has a maximum at $\rho = 0.2$. The mean of profit is monotonically increasing; the means of the remaining variables are more or less constant. The case $\rho = 0.4$ leads to incoherency of profit, cf. Fig. 4.17, thus implying the incoherency of the whole equation system with the data set. Note, that the observed value of each variable is equal to its corresponding (estimated) mean. To ensure that the first moments fulfil the equation system in a case of weak M-consistency it might be necessary to iterate SamPro algorithm. Nevertheless this is achieved after a few iterations. In Tab. 4.6 the results of the 5th iteration are given for the case $\rho = 0$. The first moments fulfil the equation system up to a small error.

Finally, three scatter plots are presented in Fig. 4.15, 4.16 and 4.17 for the variable profit with $\rho \in \{-0.4, 0.0, 0.4\}$. Note, the outlier effect, i.e. a “too large” mean of profit, i.e., $N(30, 3^2)$, on the overlap of the point cloud and the (linear) subspace spanned by simulated values of $profit$, $profit_1$ and $profit_2$.

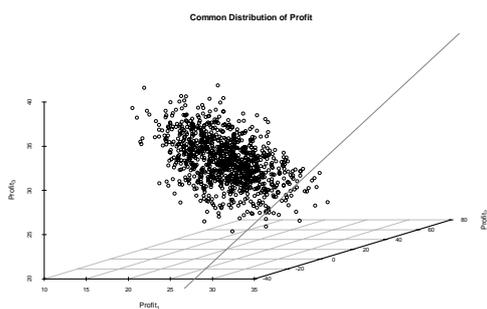


Figure 4.15: Scatter Plot of Simulated Profit Values for $\rho = -0.4$ (Weak M-Consistent)

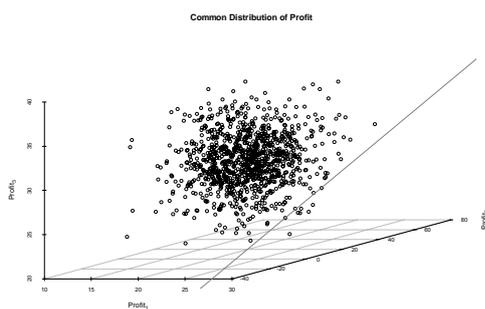


Figure 4.16: Scatter Plot of Simulated Profit Values for $\rho = 0$ (Weak M-Consistent)

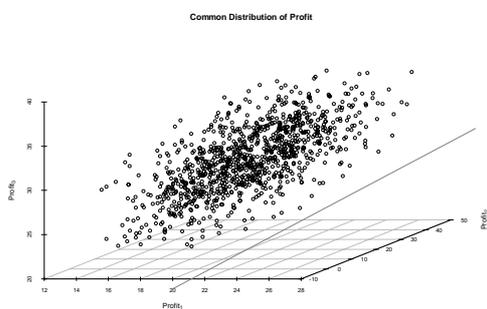


Figure 4.17: Scatter Plot of Simulated Profit Values for $\rho = 0.4$ (M-Inconsistent)

4.3.3 Experimental Group C

In this experimental group the equation model is extended to the DuPont system as described in Section 2.2.1. As prior knowledge a contaminated multivariate Gaussian distribution is used. Capital turnover is unknown, whereas for all other indicators prior knowledge is given as follows:

$$\mu_{prior} = E \left[\begin{pmatrix} Sales \\ Cost \\ Capital \\ Profit \\ ProfitMargin \\ ROI \end{pmatrix} \right] = \begin{pmatrix} 100 \\ 80 \\ 60 \\ 20 \\ 0.2 \\ 0.333 \end{pmatrix}$$

$$sd_{Prior}^2 = Var \left[\begin{pmatrix} Sales \\ Cost \\ Capital \\ Profit \\ ProfitMargin \\ ROI \end{pmatrix} \right] = \begin{pmatrix} 100 \\ 64 \\ 36 \\ 9 \\ 0.0004 \\ 0.001109 \end{pmatrix}$$

$$cor_{Prior} = Cor \left(\begin{pmatrix} Sales \\ Cost \\ Capital \\ Profit \\ ProfitMargin \\ ROI \end{pmatrix} \right) = \begin{bmatrix} 1 & \rho & 0 & \rho & -\rho & \rho \\ \rho & 1 & 0 & -\rho & 0 & -\rho \\ 0 & 0 & 1 & 0 & 0 & -\rho \\ \rho & -\rho & 0 & 1 & \rho & \rho \\ -\rho & 0 & 0 & \rho & 1 & 0 \\ \rho & -\rho & -\rho & \rho & 0 & 1 \end{bmatrix}$$

The mixture of two multivariate Gaussian distributions for this experimental group is defined as:

$$0.9 \cdot N_6(\mu_{Prior}, sd_{Prior} \cdot cor_{Prior} \cdot sd_{Prior}) + 0.1 \cdot N_6(\mu_{Prior} + 3 \cdot sd_{Prior}, 0.25 \cdot sd_{Prior} \cdot cor_{Prior} \cdot sd_{Prior}).$$

N_6 is the six-dimensional Gaussian distribution.

Correlation ρ varies in the interval $[-0.5, 0.5]$ to analyse the influence of correlation. Two simulation scenarios are presented in the following. In the first scenario, prior knowledge is M-consistent with the equation model. In scenario two, the expected mean of profit is changed to 30 and therefore the prior knowledge becomes M-inconsistent with the DuPont system.

Scenario 1: M-Consistent Prior Knowledge

The marginal prior distributions are shown in Fig. 4.18 to 4.20 for $\rho = -0.4$, $\rho = 0$, and $\rho = 0.4$.

Using SamPro new estimates for all indicators of the DuPont system are generated. The density estimates for each indicator are presented in Fig. 4.21 to 4.23

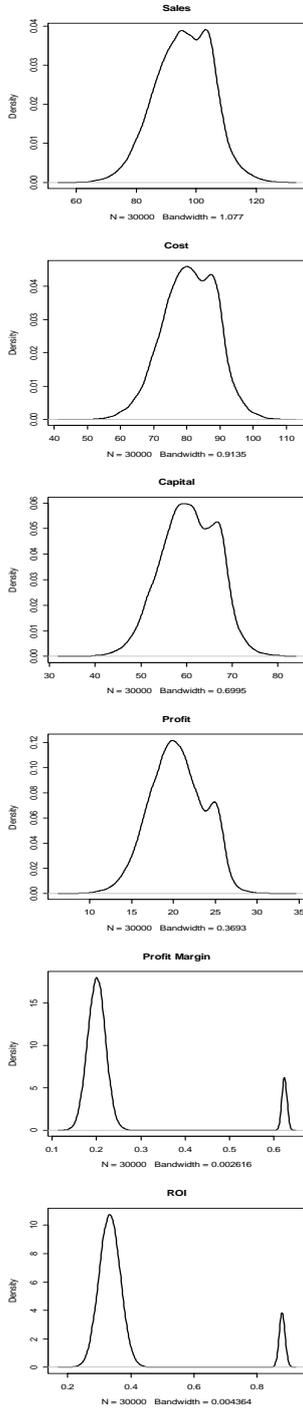


Figure 4.18: Prior Distributions $\rho = -0.4$ (Weak M-Consistent)

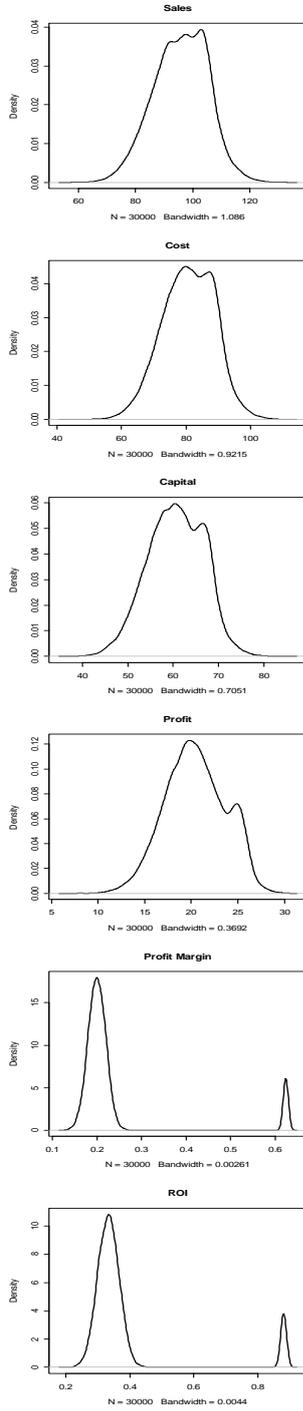


Figure 4.19: Prior Distributions $\rho = 0$ (Weak M-Consistent)

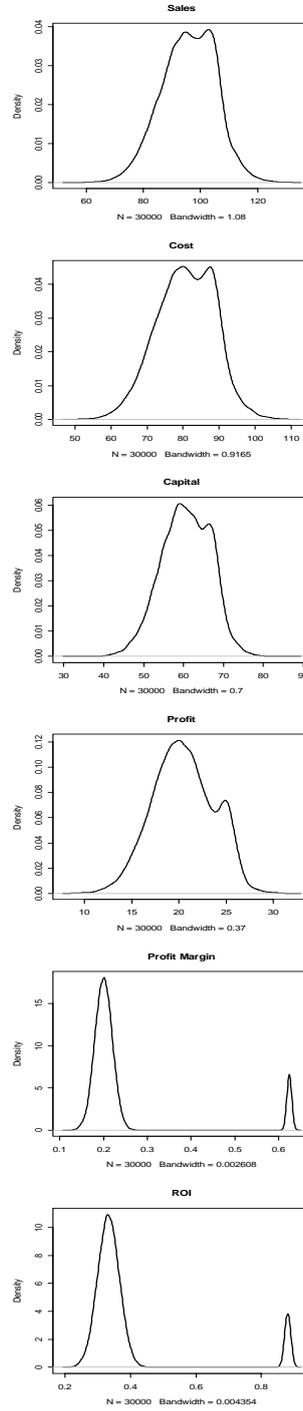


Figure 4.20: Prior Distributions $\rho = 0.4$ (Weak M-Consistent)

Table 4.7: Estimates for $\rho = 0.4$

Indicator	Quantiles			mean	sd
	$\alpha = 2.5\%$	$\alpha = 50\%$	$\alpha = 97.5\%$		
Sales	89.09	99.88	110.67	99.88	5.51
Cost	70.20	81.64	89.95	81.32	5.45
Capital	49.64	59.80	70.22	59.83	5.25
Profit	16.87	19.88	23.08	19.91	1.59
Profit Margin	0.165	0.199	0.235	0.199	0.018
ROI	0.275	0.332	0.390	0.332	0.029
Capital Turnover	1.266	1.654	2.194	1.678	0.232

with respect to the correlation parameter ρ . Density plots for cost reveal the perpetuation of a bimodal distribution function. Furthermore, the density plots for capital turnover show a non-Gaussian estimate. This is due to the division operation for capital turnover. As a result a Cauchy distribution is obtained by simulation.

Comparing the new estimates with the prior information it can be seen that a reduction in the uncertainty is achieved. Tab. 4.7 shows the results for $\rho = 0.4$. This simulation is representative to the other simulation experiments. The reduction of uncertainty can be easily seen.

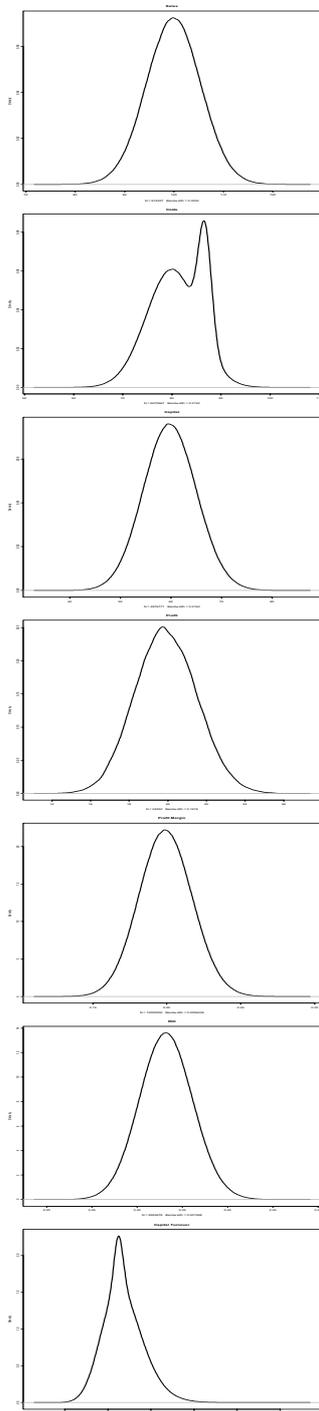


Figure 4.21: Result Distributions $\rho = -0.4$ (Weak M-Consistent)

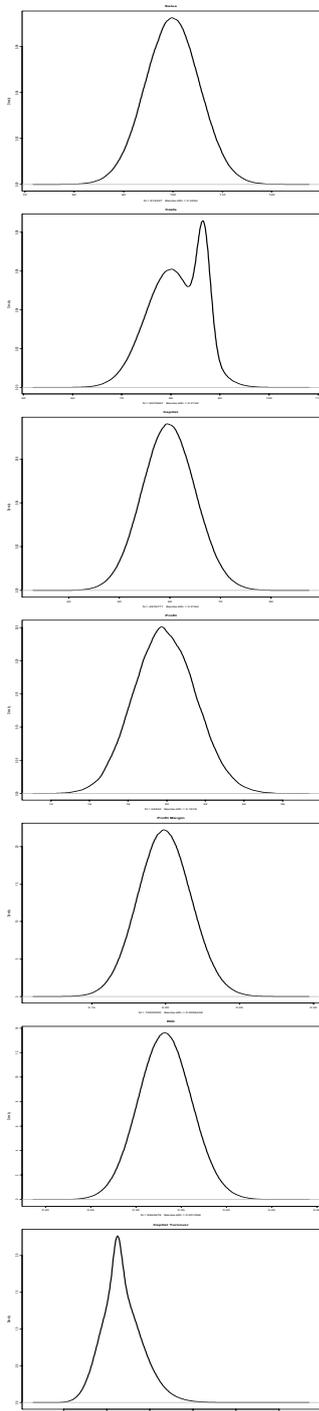


Figure 4.22: Result Distributions $\rho = 0$ (Weak M-Consistent)

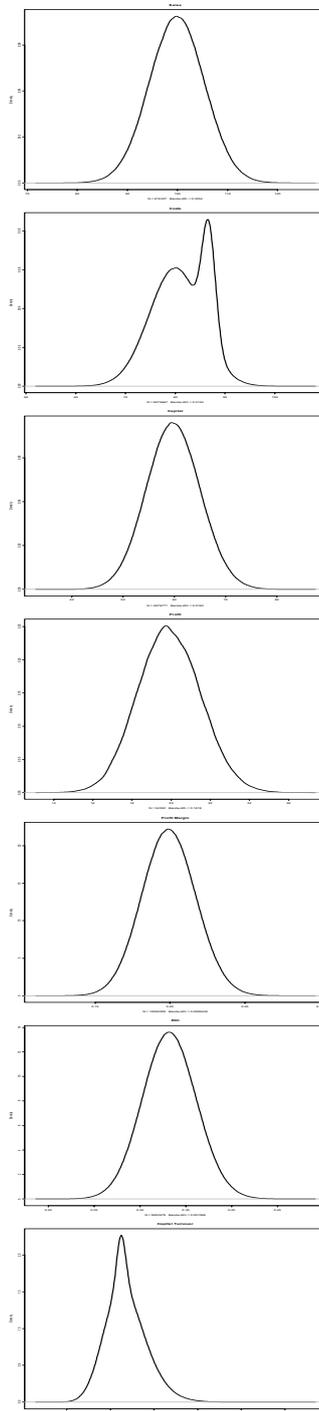


Figure 4.23: Result Distributions $\rho = 0.4$ (Weak M-Consistent)

Scenario 2: M-Inconsistent Prior Knowledge

In the next scenario the prior knowledge is slightly changed. The expected mean of profit is set to 30 instead of 20. This implies M-inconsistency of prior knowledge or, in other words, the data do not coincide with the equation model.

The simulation study is again performed for different values of ρ . The influence of correlation is studied as in experimental group B.

Due to the fact that the prior knowledge only differs in the expected value of profit, it is not given here again.

Performing the simulation with the SamPro-algorithm for $\rho = -0.4$ and $\rho = 0$ new estimates can be obtained. Running the simulation experiment with $\rho = 0.4$ the system is stated as M-inconsistent. The reason is that the different estimates of profit cannot be combined. In Fig. 4.24 a scatter plot for all profit estimates is given. Merging profit and the result of equation sales \cdot profit margin reveals that both estimates do not map to the same subspace (no points on the line). Therefore the system is marked as M-inconsistent.

The influence of correlation can be easily seen in Fig. 4.25, 4.26, and 4.27. In the last three dimensional scatter plot ($\rho = 0.4$) points or samples do not exist on the desired subspace. That is the reason why in this simulation experiment the system is stated as M-inconsistent.

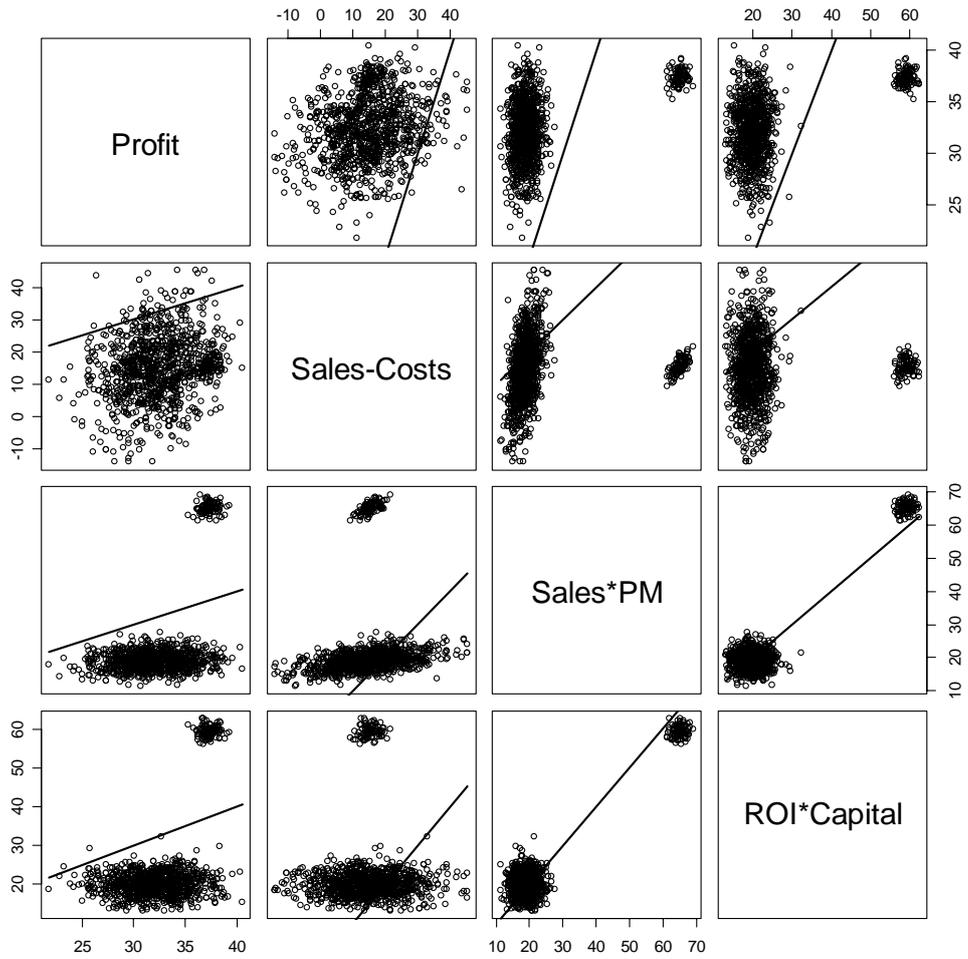


Figure 4.24: Scatter Plots of Profit for $\rho = 0.4$

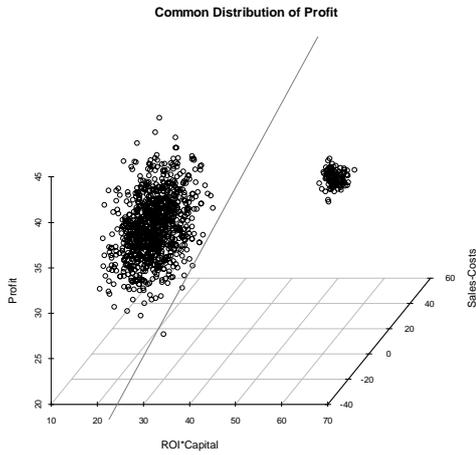


Figure 4.25: 3D Scatter Plot of Simulated Profit Values for $\rho = -0.4$ (Weak M-Consistent)

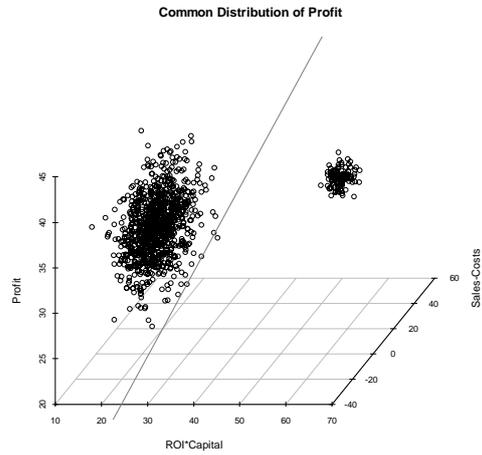


Figure 4.26: 3D Scatter Plot of Simulated Profit Values for $\rho = 0$ (Weak M-Consistent)

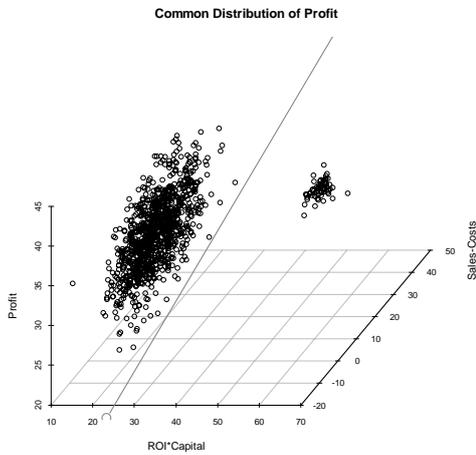


Figure 4.27: 3D Scatter Plot of Simulated Profit Values for $\rho = 0.4$ (M-Inconsistent)

4.3.4 Results

A summary on the above described experiments can be given as follows:

In the uncorrelated case (the means of) the simulated indicators are about the same as the GLS estimates.

Skewness of distributions mostly has only a small effect on the estimates.

Cross-correlations of the variables can lead to severe problems: The equation system may become M-inconsistent with respect to a given data set, i.e., the overlap of the sets of simulated values of at least one variable, determined from all equations where it is part of, may become empty. Under a Gaussian regime with infinite domains and under the independence assumption, this effect cannot happen.

Using a GLS approach is compared to MCMC simulation computational cost-effective. However, skewness and correlation may lead to quite different estimates and the introduction of robust estimators, like median, improves all estimates even more. In the case of M-inconsistency it might be necessary to iterate the simulation algorithm several times to satisfy a given balance equation system. Of course, any iteration will increase the computational effort. Furthermore, note that if a given data set is contradictive to the corresponding equation system, M-inconsistency is revealed by our MCMC simulation algorithm, which is quite in contrast to the GLS approach used by Quantor.

4.4 Balanced Scorecard Example

In this section a BSC example is illustrated. This is taken from Köppen et al. (2007). Here the software Balanced Scorecard Simulator is used which is presented in Allgeier (2006).

4.4.1 Model

The MCMC simulation is based on a stochastic model of the selected BSC indicators, cf. Fig. 4.28.

The model is characterised by the following three features:

- 24 indicators (variables) and three system constants are assigned to the four BSC perspectives (employee and innovation, internal business process, customer and financial perspective).
- All variables of the model are connected to each other by the four basic arithmetical operations;
- 15 equations with up to three variables exist. Obviously, each equation can be uniquely solved for each existing variable.

4.4.2 Input Data

In our simulation example full prior information is provided for all 24 characteristics, cf. Tab. 4.8 columns 2 and 3. Fully specified prior distributions for the current state are shown in column 2. Column 3 shows given target values for each indicator.

The simulation is primarily used to identify M-inconsistencies of the BSC data. Additionally, a new indicator is established, namely “cost“ related to the financial perspective, which is defined as $cost = turnover - profit$. The indicator *turnover of regular customers of the previous period* is constant and set to 1900.

4.4.3 Simulation and Results

The CPU operating time of a test run with a number of iterations $T = 100.000$ is approx. 5 min. The data set does not contradict the given model. The columns $\hat{\mu}$, $\hat{\sigma}$ and adjusted target values in Tab. 4.8 show the result of the simulation. An analysis of the computed expected values and standard deviations for every indicator, provides evidence that the imprecision (error intervals) of the simulated indicators are reduced. The observed values are adjusted in the way, that a shift is proportional to the prior variance of a variable.

The Balanced Scorecard Simulator computes model consistent target values as described by Allgeier (2006, p. 61) and prior information for the current state is used with the SamPro algorithm. The results of the simulation lead to estimates $\hat{\mu}$ and $\hat{\sigma}$.

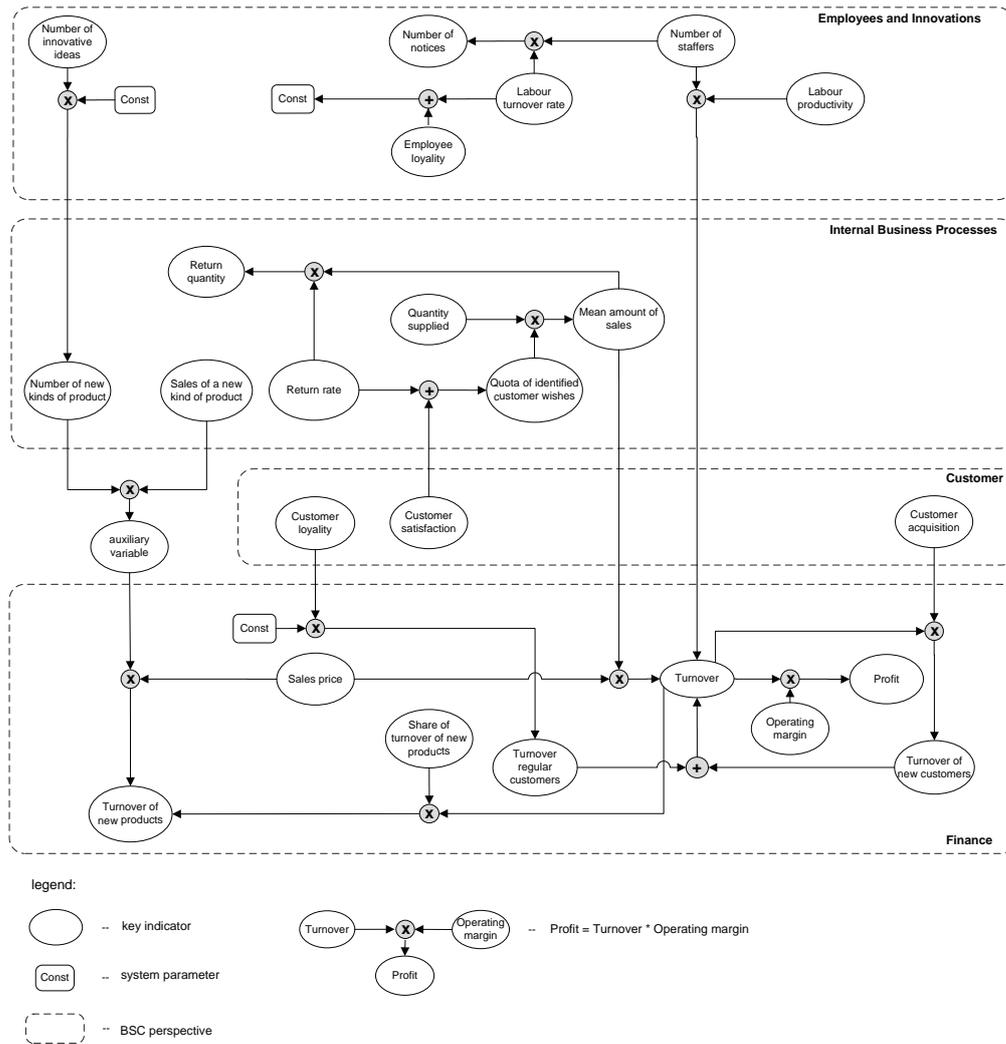


Figure 4.28: Balanced Scorecard Model Graph

Table 4.8: Prior Information of the BSC Characteristic Indicators

BSC indicator	prior distribution	given target value	$\hat{\mu}$	$\hat{\sigma}$	adjusted target value
Number of innovative ideas	$N(18, 2^2)$	18	18.00	1.86	20.00
Employee loyalty	$U(0.97, 0.99)$	0.99	0.99	0.0004	0.99
Number of notices	$U(1, 3)$	2	2.06	0.24	2.05
Number of staffers	$N(205, 5^2)$	205	201.14	3.76	203.70
Labour turnover rate	$N(0.01, 0.001^2)$	0.01	0.01	0.0006	0.01
Labour productivity	$N(20, 2^2)$	20	19.94	0.42	20.00
Number of new kinds of product	$U(3, 5)$	4.5	4.47	0.86	5.00
Sale set of a new kind of product	$U(45, 50)$	50	47.49	1.43	50.00
Returns quantity	$N(89, 5^2)$	90	89.75	4.40	100.00
Return rate	$N(0.19, 0.02^2)$	0.2	0.17	0.01	0.20
Quantity supplied	$N(550, 10^2)$	556	548.73	9.30	625.00
Mean amount of sales	$N(440, 10^2)$	445	439.45	9.41	500.00
Quota of identified customer wishes	$N(0.8, 0.01^2)$	0.8	0.80	0.01	0.80
Customer loyalty	$N(2, 0.01^2)$	2	2.00	0.01	2.04
Customer satisfaction	$U(0.6, 0.7)$	0.6	0.65	0.03	0.60
Customer acquisition	$N(0.05, 0.01^2)$	0.05	0.05	0.0003	0.05
Turnover of new products	$U(1990, 2010)$	2000	1999.40	5.67	2037.04
Sales price	$N(8, 2^2)$	9	8.50	0.57	8.15
Share of turnover of new products	$N(0.45, 0.001^2)$	0.5	0.45	0.00	0.50
Turnover regular customers	$N(3800, 50^2)$	3800	3795.20	44.36	3874.07
Turnover of new customers	$N(199, 10^2)$	200	198.78	6.09	200.00
Turnover	$N(4000, 25^2)$	4000	3994.38	20.54	4074.07
Operating margin	$N(0.25, 0.02^2)$	0.27	0.25	0.01	0.27
Profit	$N(1000, 25^2)$	1100	999.25	22.09	1100
Cost	unknown	unknown	3013.77	129.07	2974.07

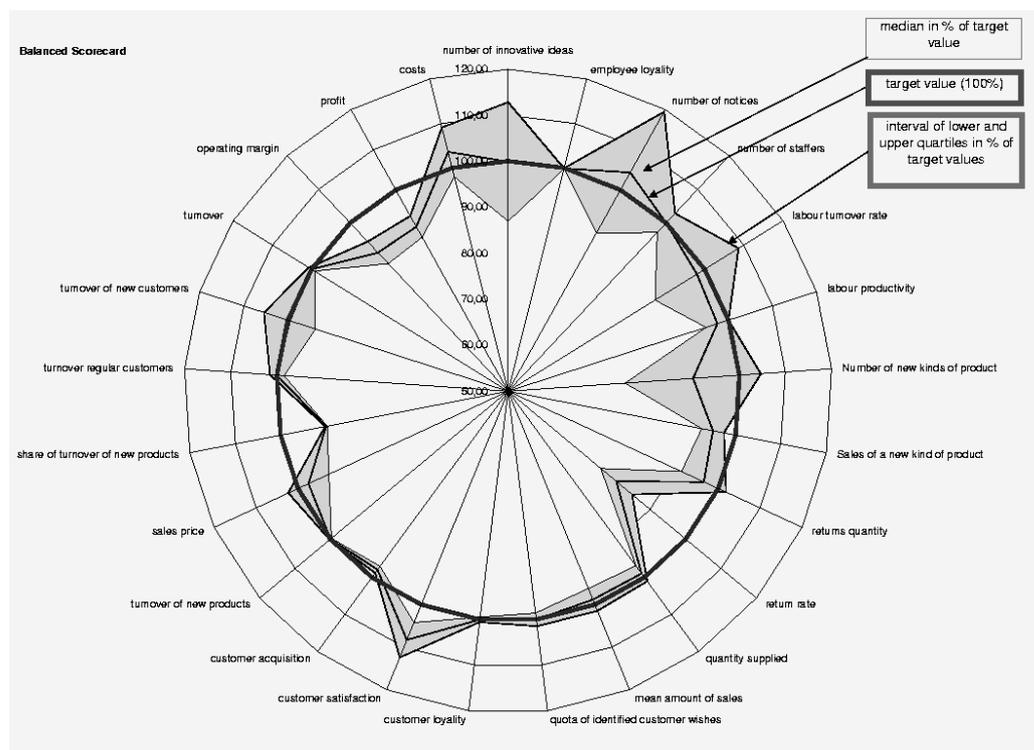


Figure 4.29: Simulation Results of BSC Characteristic Indicators (Kiviati Diagram)

In Tab. 4.8 the results of the Balanced Scorecard Simulator are presented. The given target value of indicator *ReturnsQuantity* was in conflict to the equation system given by Fig. 4.28. An adjustment of the target value from 90 to 100 was achieved for this indicator which does not conflict with the BSC model. The current state of indicator *Returnrate* was adjusted from 0.19 to 0.17 by using the SamPro algorithm. Other indicators like *Quantitysupplied* or *Turnover* were adjusted in the current as well as target values.

The simulation results are visualised in a Kiviati diagram, cf. Fig. 4.29. This chart type is suitable for the representation of multiple indicators. If a system of equations is classified as M-inconsistent, only the number and the names of the incorrect indicators are reported by a result notification. This makes it easier for the user to identify the causes for an M-inconsistency within the data.

4.4.4 Results of BSC example

The adaptation and modelling of key indicators as random variables in a BSC enhances the information content and brings the BSC closer to reality, cf. Müller et al. (2003). Randomness happens due to a kind of 'natural' longitudinal derivation of indicators, errors in observations and measurements of actual data of BSC

indicators, or evaluation tolerance in target data.

The software *Balanced Scorecard Simulator* is a management tool for planning, decision making, and controlling, i.e., it allows the computation of M-consistent target values, produces coherent, analytical data, and supports the controller to detect incoherencies in the values of the BSC, c.f. Köppen et al. (2007), Allgeier (2006).

Additionally, it reduces uncertainty within the prior information on the current state for each indicator by the usage of the SamPro algorithm.

The *Balanced Scorecard Simulator* is useful in the context of stochastic indicators where quantitative relationships are known. A basic strength of the BSC is the fact that it also applicable for qualitative factors. However, a quantitative representation of these soft facts has to be taken into account and therefore the tool is an improvement of the current state. In the case that the relations of the soft facts of a BSC are unknown the tool cannot be used in an appropriate way.

4.5 Tool Evaluation

In this section a brief evaluation of the software tools is given. This evaluation focuses on the criteria described at the beginning of this chapter. The results of the comparison can be summarised as given in Tab. 4.9 and will be explained in more detail for each software tool in the following.

Table 4.9: Evaluation of Tools for Indicator Systems with Errors in the Variables

	FuzzyCalc	Quantor	MoSi
Graphical Modelling	+	0	++
Information on errors	0	0	++
Result presentation	+	0	++
Restriction of Models	+	0	+
Performance	++	++	--

Symbols used for the evaluation are given in Tab. 4.10.

Table 4.10: Evaluation symbols

Value	Symbol
Excellent	++
Good	+
N/a or medium	0
Poor	-
Very poor	--

4.5.1 FuzzyCalc

Graphical Modelling

FuzzyCalc states the equations clearly arranged in a mathematical form. These equations are represented as a block. The relations between variables in a complex system are difficult to identify and the number of involved equations per variable is hard to discover at first.

Information on errors

FuzzyCalc supports for triangular and trapezoid membership functions. These are the most common membership functions. The interpretation of a multi peak membership function might be quite difficult. Dependency structures between variables cannot be modelled. Multivariate membership functions would be useful, but are unfortunately not implemented in the tool.

Result Presentation

Results of the Fuzzy set approach are Fuzzy membership functions themselves. FuzzyCalc handles these functions by the use of a graphical and table view. Moreover prior knowledge is visualised in the same view as the new and improved values. In the case that a strong M-inconsistency is detected a dialog box appears to inform the user.

Restriction of Models

Restrictions on the model are caused by two reasons. One is that only basic arithmetic operators are handled and each equation consists of exactly one of these four operators. The other limitation is due to the software tool MS-Excel limited abilities, which in turn limit the models complexity.

Performance

FuzzyCalc solves the models in matter of a few seconds, which is relatively short and acceptable amount of time. The visualisation of the results is also performed very quickly.

To sum up the above components, FuzzyCalc is a software tool for Fuzzy indicator systems, which is not only useful in decision making but also easily accessible for the user. Accuracy is defined with the 100 α -cuts that are implemented in the software.

4.5.2 Quantor

Graphical Modelling

Quantor hides the information of relations in dialog boxes. This makes it difficult for the user to see equations, but a structure can be organised using the cell grid provided by Quantor. The tool has no graphical elements for modelling uncertainty and uses a tabular environment with dialog boxes.

Information on errors

Information on errors is restricted to Gaussian distributions. This simplifies the user interaction. Mean and standard deviation of a Gaussian distribution are used as prior knowledge. Estimates of mean and standard deviation are computed, if all means fulfill the equation system and standard deviation is reduced as far as possible.

Result Presentation

Quantor only gives estimates on the first and second moments. These should be treated quite carefully and interpreted in cases of violation of the Gaussian

regime, i.e., quotients of two Gaussian variables. Due to the unlimited domain of a Gaussian distribution, M-inconsistency cannot occur. Results very far from prior values, should carefully be checked if the corresponding estimated variance is small.

Restriction of Models

The four arithmetic operations $+$, $-$, \cdot , and $/$ are allowed. Quantor works with another restriction that division cannot be applied on quotients, due to the assumption of Gaussian approach. This clearly restricts the model class. Another drawback is that for each indicator used, only one equation is allowed. Rearrangements and introduction of intermediates have to be used if several equations per variable exist.

Performance

Quantor solves models the fastest compared to the other two applications. The results are often computed within a few seconds. Accuracy is limited to the CPU number precision.

4.5.3 MoSi

The developed prototype MoSi is evaluated in the following paragraphs. A description of design and implementation of MoSi is given in Chapter 5.

Graphical Modelling

Mathematical formulation and graphical visualisation are both used in MoSi. It is possible to formulate the model with mathematical equations where in each equation a basic arithmetic operator is used. These equations are visualised in a graph mode, where only the commutative operators $+$ and \cdot appear. Relations induced by the equation system can easily be seen by the user in the graphical visualisation. A feature that is currently in demand is an equation parser that relaxes the assumption of only one operator per equation.

Information on errors

MoSi can handle every type of density function by the use of a general MCMC method. Several improved PRNG are implemented and can be used. It is furthermore possible to enrich the prior knowledge by own functions that have to be written in R. Another restriction is overcome by using the MH algorithm. The target function does not need to be normalised but the integral should be finite. Multivariate distributions are also possible and new algorithms implemented in R packages can be easily used.

Result Presentation

MoSi provides statistical analysis of the results including moment estimates and graphical visualisation by using histogram plots. To supply the user with a more detailed analysis of the obtained results they are saved to a R file.

Restriction of Models

Restrictions on models are only dependent upon the main memory of the computer MoSi is running on. In the actual version the four basic arithmetic operations are implemented. This restriction can be relaxed and further operators can be implemented. The necessary condition for admitting a new operator is the principle of separation. It is not necessary that the operator is dyadic.

Performance

Simulation generally needs more of a computational effort. However, the performance depends to a significant extent on the random number generator and the dimensions within the model. These dimensions are build up from the number of variables and equations as well as the number of equations per variable. Whereas FuzzyCalc and Quantor compute a DuPont model within seconds, MoSi can take from several minutes or hours up to even days. To increase accuracy, more computational effort is necessary which directly translates into an increase of running time.

Chapter 5

Software Design and Implementation

This chapter deals with the aspects of the design and implementation of the Software tool MoSi. MoSi stands for *Modelling and Simulation* and MoSi enables the user to model indicator systems as described in chapter 2. The simulation environment can be used to obtain improved stochastic estimates. Due to the fact that different existing packages are included into the project, each of them will be briefly explained.

It was one of the design goals to make the software almost entirely independent from the operating system. Thus, the Programming language Java was an obvious candidate. However, simulation requires a high computational effort and the performance losses due to the Java VM Technology are not within acceptable limits. This obvious disadvantage led to the decision to use the statistical programming language R for the simulations.

Licensing issues play another role in this decision but will not be further discussed here. For the design UML 2.0 is used to craft the Java architecture of the software. The description of R algorithms is realised with pseudo code.

5.1 Design

There are two major requirements for an indicator systems simulation software: user friendliness and performance.

This is the reason why two different programming languages are used. With the Java Graphical User Interface (GUI)s can be easily produced, but the performance of Java applications when it comes to random number generation and simulation is unfortunately unsatisfactory. GNU R is a fast environment for statistical computing, and simulation tasks are quickly done.

Using Java as the technology for GUI makes it possible to use the software on every system where the Java Virtual Machine in version 1.5 or higher is installed. Due to the fact that the other involved software is also available for different platforms the complete MoSi software can be used on a wide selection of operating systems. The architecture of MoSi is presented in Fig. 5.1.



Figure 5.1: Architecture of MoSi

Indicator system should be modelled keeping two aspects in mind. One being that the input of equations should be possible and the other, that a graph visualisation of the indicator system is also of interest. The graph representation has the advantage that connections between variables are discovered more easily.

The design concept of model view controller will be used. The use of this pattern isolates the business logic (simulation) or model from user interface considerations the controller and view. Therefore, modification either on the visual appearance or the underlying business rules are easier implemented and, if necessary, changed. The class diagram of the main business logic and graphic classes is presented in Fig. 5.2.

Since the simulation of such equation graph systems is done in two different language environments, an interface is necessary for the communication between both languages. Additionally, it should be possible to redo the simulation at another time by obtaining the same results. That is the reason why for the simulation purpose files are also stored. The generation of simulation files is done from the equation graph within the Java environment, whereas the execution of these script files is performed within R. The results from the simulation process are transferred to Java and visualised. Fig. 5.3 reflects the simulation flow in an activity diagram.

The activity diagram is from the view of the Java language. Three main activities can be distinguished: simulation preparation, simulation execution, and simulation analysis.

In the preparation phase the necessary parameters are obtained from the user and the R simulation script files are generated from the equation graph model.

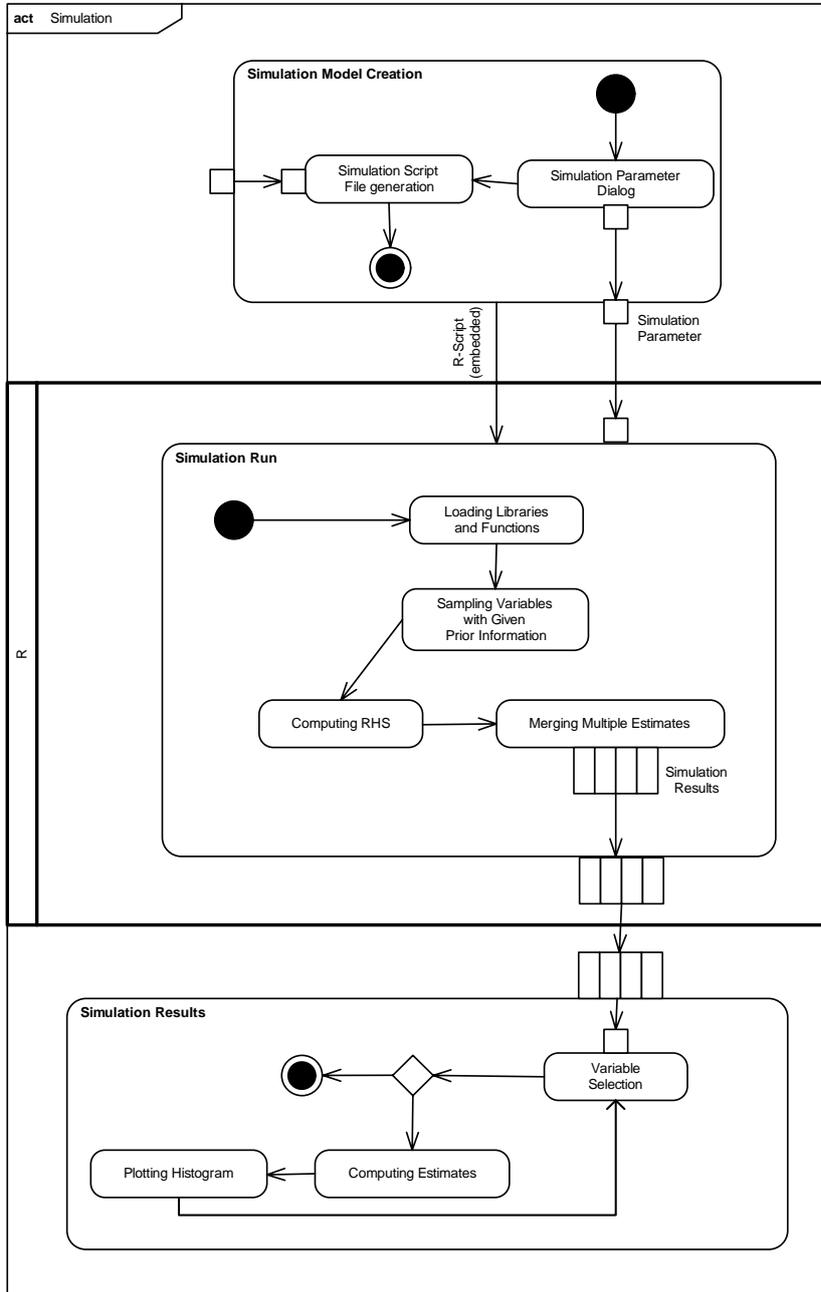


Figure 5.3: Activity Diagram for MoSi-Simulation

This graph model serves on the one hand as a business indicator model and on the other hand as a source from which prior information for variables can be extracted. This helps in creating the simulator which is written into a text file. Prior information is used here to create samples for unknown variables as well as to generate multiple estimates with equations extracted from the business indicator model.

Simulation takes place in R and the generated file as well as simulation parameters, like seed and sample length, are used within R. After Sampling LHS variables and generating RHS variables multiple estimates per variable are merged according to Alg. 10. Simulation results are stored in one vector of samples per variable.

In the simulation analysis phase, a variable selection is performed which includes a complete communication of the corresponding vector from R to Java. This vector is used to generate an interactive histogram plot where the user has the possibility to set the number of breaks. Furthermore statistical measures like estimated mean, median and standard deviation are computed and displayed.

The structure of simulation process is presented in a class diagram as given in Fig. 5.4. Two main parts can be distinguished here. Model representation is given in the upper part and classes for the simulation are placed in the lower part of the figure. The equation system is build up on equations. These equations consist of variables and operators. If prior knowledge is available it is connected to the variable as a distribution. In Fig. 5.4 only two distribution classes as representatives are visualised. All distributions classes are derived from a super class called *Distribution*. A connection between the logical model and the simulation model is implemented by two associations. On the one hand, for each *Variable* in the logic view, a complementary *RVariable* is generated. The *RVariable* objects also contain the simulation result vector. On the other hand *RSimulation* is associated to the *EquationSystem* class. These two classes build and run the simulation. The communication between Java and R is implemented in the class *RCommunication* which uses an interface of the JRI package (Urbanek 2007). The simulation results are prepared for the graphical presentation in the class *RSimulationResult*. This class also creates the histogram.

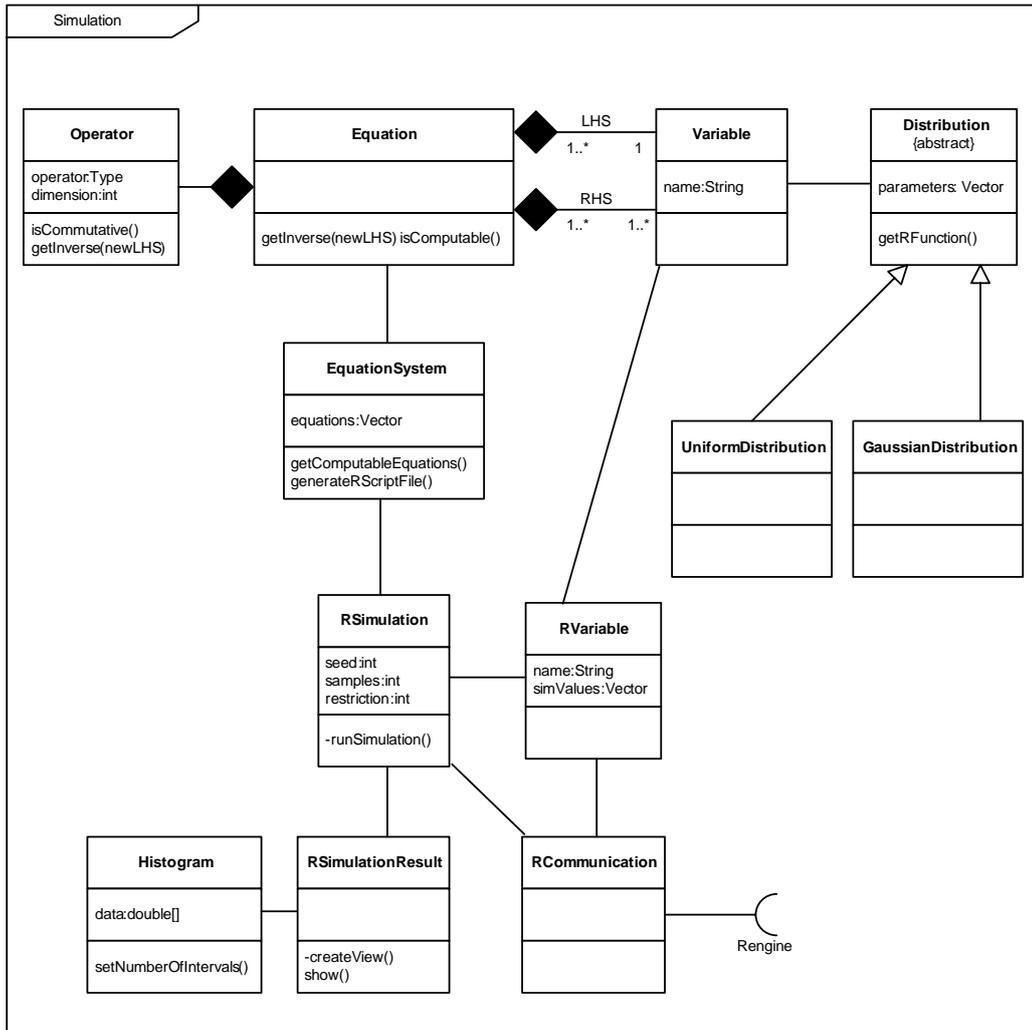


Figure 5.4: Class Diagram for MoSi-Simulation

5.2 Implementation

In this section the implementation of MoSi is presented. MoSi consists of a two programming environments. The user interacts with a GUI that is developed using Java. As a template the UML-editor “Violet” (C. Horstmann) is used. Here the main aspects of the graph editor framework, which is explained in Horstmann (2005), is of interest.

The implementation of the GUI is done with Java 1.5 and uses Swing libraries. To obtain a persistent layout independent from the operating system a common *look and feel* is used.

As described in Chapter 3, simulation needs a lot of computational power. To increase performance for the simulation, R (R Development Core Team 2007) is used. “R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files” (Hornik 2007). From the available technical possibilities, running script files is of main interest. Communication between Java and R is implemented by using JRI package (Urbanek 2007).

An overview of the project structure is given in the package diagram in Fig. 5.5.

The package “editor” includes all GUI relevant classes. The sub-packages “framework” and “common” are adopted from the editor “Violet” by Horstmann (2005). The sub-package “equation” includes the corresponding GUI classes for the equation models. The communication interface between Java and R is placed in the package “JRI”.

The “logic” package contains all relevant classes for simulation of indicator systems. This includes the preparation of simulation which is organised in the sub-package “equation”. The actual simulation as well as the statistical evaluation take place in the sub-package “simulation”. In the sub-package “fuzzy” the FuzzyCalc algorithm is implemented.

A list of Java classes is given in Fig. 5.6.

While in Java the GUI and model creation are implemented, the simulation is done in R. For communication between both languages, script execution as well as interface communications are used. The option for using the script file facility for a simulation rerun is left open. On the other hand experiment results are not saved into files due to possibility of simulation reruns that yield the same results.

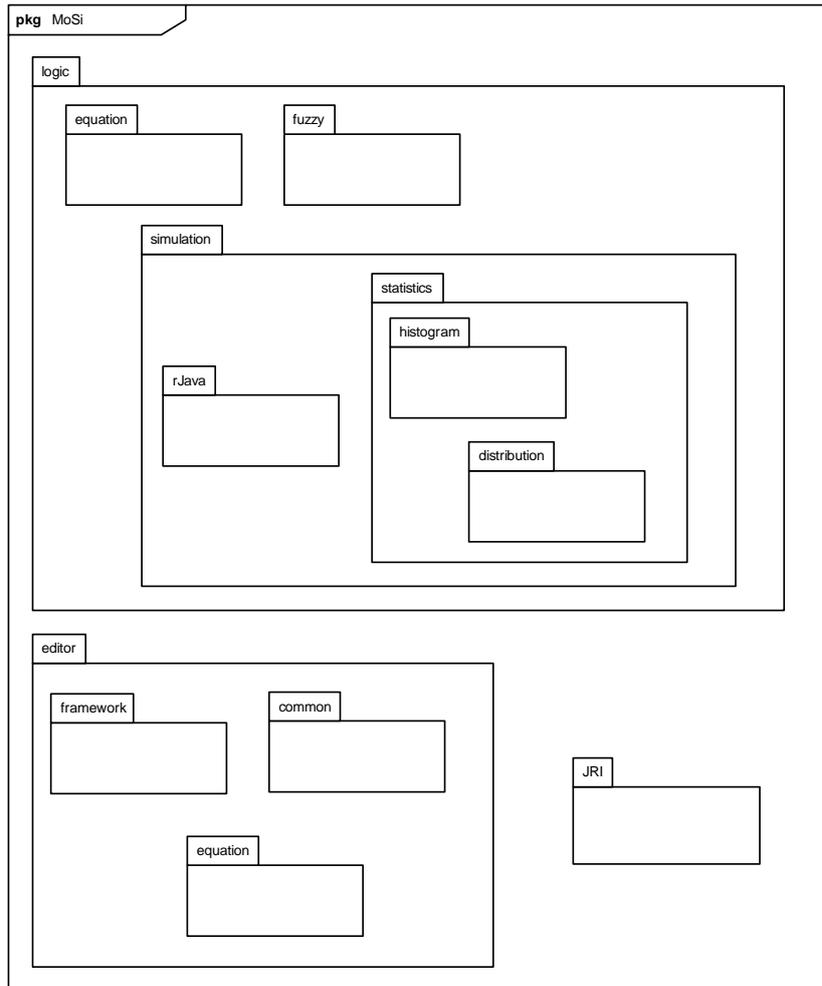


Figure 5.5: Package Diagram for MoSi

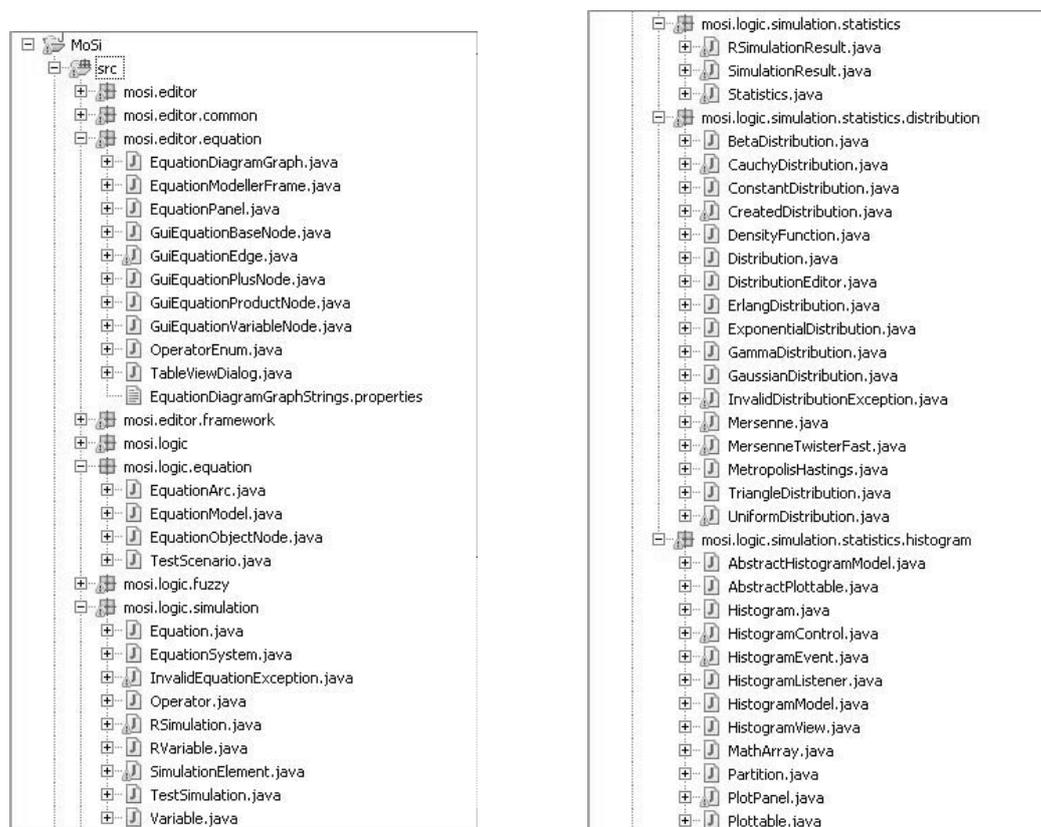


Figure 5.6: MoSi Project Files

Listing 5.1: R-Simulation Script

```

# Simulation Parameter Initialisation #

n <- 100000
set.seed(12345)
source("Functions.R")

# Initialising Variables #

variable1 <- rnorm(n, 0, 1)
variable2 <- mh(n, densityvariable2, type = "independence")
variable3 <- rtriangular(n, -1, 0, 1)

# Computing new Values #

variable4equation1 <- variable1 + variable2
variable4equation2 <- variable2 / variable3

# Folding new Values #

variable4 <- merging(matrix(c(variable4equation1, variable4equation2), ncol=2)
rm(c(variable4equation1, variable4equation2))

```

In Listing 5.1 an equation graph, consisting of four variables (variable1 to variable4) and two operators, represents the underlying model. The sum of variable1 and variable2 is variable4. The product of variable4 and variable3 is variable2. As prior information variable1, variable2 and variable3 are given. At first, the initialisation is done. This is followed by LHS sampling using prior information. For these samples all possible RHS variables are computed in a third step. In case more than one estimate exists per variable these estimates are folded. This is the last step of the simulation within R. As a result, one vector of samples exists for each variable. If a vector is empty, M-inconsistency is the result of the simulation experiment.

The script file *Functions.R* which is used in the simulation script loads several functions and R packages that are necessary for the simulation experiment. At this point the implementation of the MH algorithm for prior knowledge can also be done. A good introduction on implementing MH algorithm is given by Chib and Greenberg (1995). Due to the fact that R is a functional programming language only target functions for the MH algorithm have to be implemented in the *Functions.R* script and sampling can be used as depicted for *variable2* in listing 5.1. *densityvariable2* stands for the function from which samples should be drawn. In this example a independence chain is used. Furthermore, parameters of the *mh* function are used as standard values. Two samples for *variable4*

have to be merged. The algorithm is described in Section 3.3.8. In the case of M-inconsistency the result of function merging is 0. Otherwise a vector of the merged values is returned.

As a result for each involved indicator, a vector of SamPro results is transferred to Java. Here again the graphical representation of the simulation is given. A histogram which is adjustable to the number of uniform breaks is used for each indicator. Statistical indicators such as mean, median and standard deviation are also given. In the case of indicator M-inconsistency, the user is informed by a dialog box.

In the following section a short software demonstration for the DuPont system is given.

5.3 Software Demonstration

In this section a short demonstration of MoSi is given. To ensure that MoSi runs, R should be installed and the requirements, such as installed R packages and environmental variables, should be fulfilled. A Java Virtual Machine (JVM) with at least version 1.5 is also required to run the software.

After starting MoSi it is possible to create a new equation graph, or to load MoSi specific Extensible Markup Language (XML) files. The equation graph is the model used within MoSi for simulation of equation systems. Fig. 5.7 contains such an equation graph for the DuPont model. It shows all standard elements

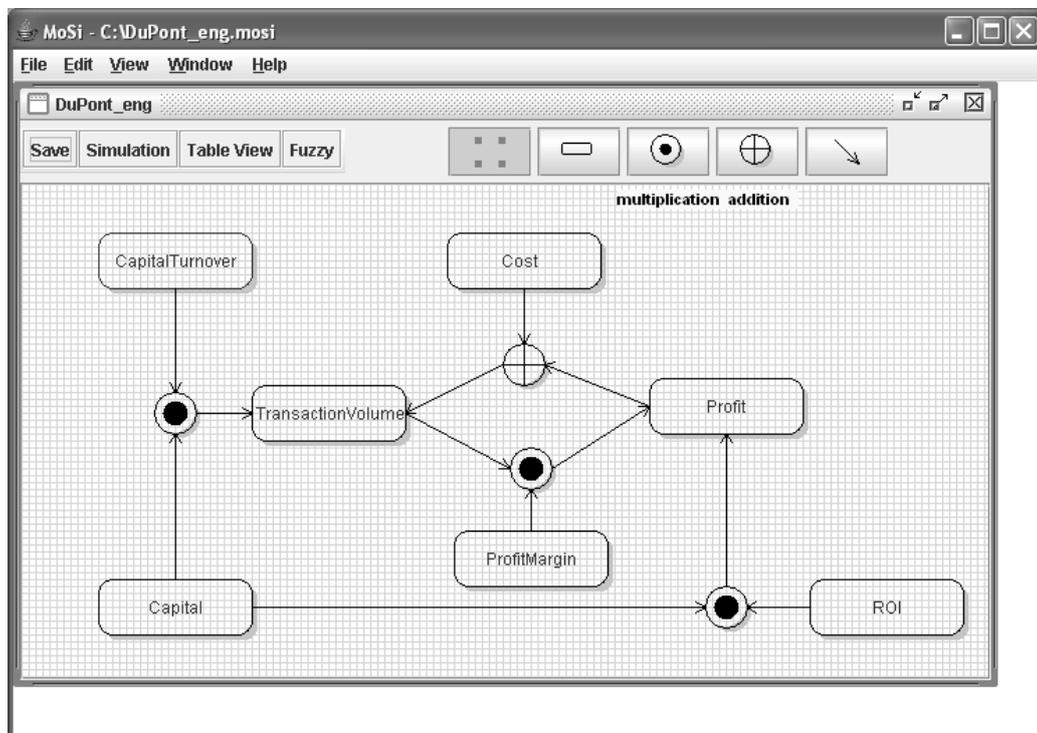


Figure 5.7: Screenshot of DuPont Model within MoSi

of the equation graph. In the menu bar, File dialog can be used for loading and saving models. The creation of new models is also possible via this menu item.

In an internal frame the equation graph is given. This frame consists of a button bar and the modelling area. The button bar consists of two parts. In the left part four standard actions can be easily started by pressing the corresponding button. A click on the “Save” button saves the model in a XML file. The “Simulation” button starts the process of simulation as described in Section 5.1. “Table View” button transfers the graph representation in a mathematical formulation.

In Fig. 5.8 the table view of the DuPont model is presented. By adding new

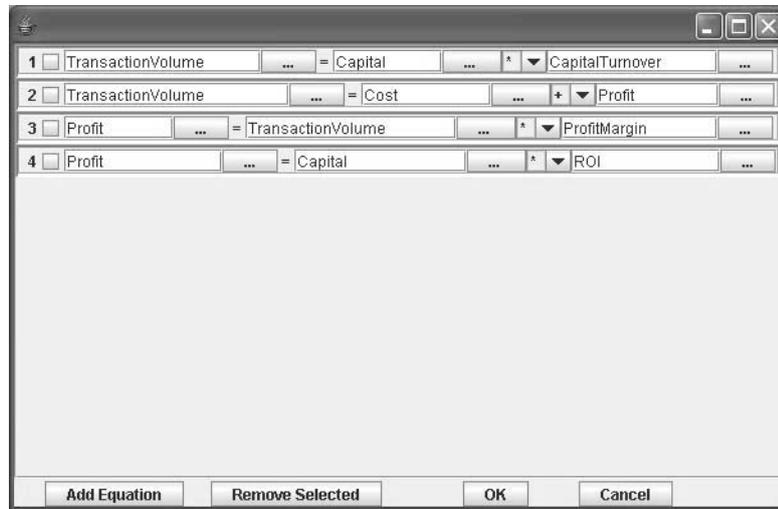


Figure 5.8: Screenshot of Table View of the DuPont Model

equation operators +, -, ·, and / are possible. Transformation from or into the equation graph only uses the commutative operators. In the table view dialog the deletion and addition of equations in mathematical formulae is possible.

In the second part of the button bar five buttons for the graphical modelling of the equation graph are placed. The first button enables the selection of existing elements. The second button adds new variables into the equation graph. Button three represents the multiply operator node and button four the addition operator node. To insert a directed relation between an operator and a variable node the last button can be used.

To model an equation in the graphical environment three variables and one operator are necessary. Two variables link into the operator. These are the input or LHS variables. A third relation is directed from the operator to another variable. This is the RHS or output variable. In Fig. 5.7 for instance the equation

$$\textit{Capital Turnover} \cdot \textit{Capital} = \textit{Transaction Volume}$$

is visualised on the left side.

Each variable node has properties that are available via double click on the graphical element. In the property dialog the name of a variable can be changed. It should be noted that equal names refer to equal variables in the simulation. Although two rectangles in the modelling graph are used in the simulation phase these are constituted as one variable.

In Fig. 5.9 a property view is shown. Three tabs exist here. In the common tab the name of a variable can be given. In the second tab the simulation properties are specified. In a drop down menu the selection of distribution for the variable is possible. Corresponding to the selected distribution function, the necessary parameters are also editable in this tab. The third tab deals with the Fuzzy

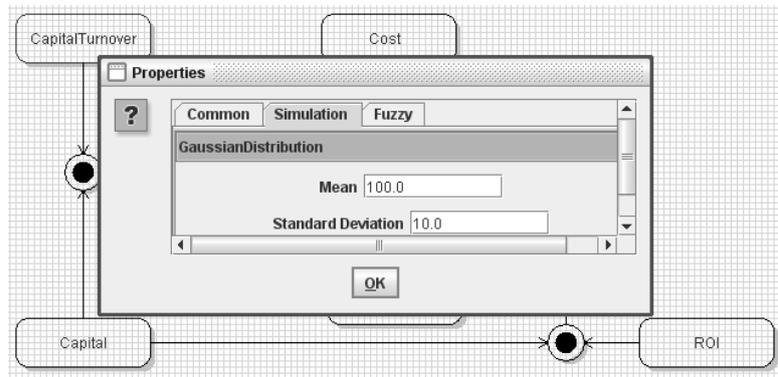


Figure 5.9: Variable Property Dialog

representation of the variable and is of no further interest at this point.

Starting the simulation will open a small simulation dialog box, cf. Fig. 5.10.

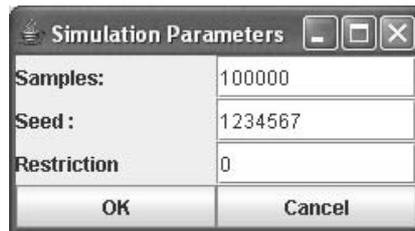


Figure 5.10: Simulation Start Dialog

Three parameters of the simulation are adjustable within this dialog. On the first entry the number of generated samples per variable is set. Increasing this number not only increases simulation run time but also memory usage. On the other hand the accuracy depends upon this parameter. To reproduce a simulation run the initial seed can be set in the second parameter of the simulation dialog. The third parameter will only be used in the graphical report. This restriction defines the number of samples which is used for the histogram plot. Setting this parameter to zero means no restriction and all generated simulation elements will be used.

The analysis of the simulation can be done in the Java environment. A histogram and some statistical estimates will be given here. Fig. 5.11 and 5.12 show two histogram views of the same simulation result.

In Fig. 5.11 the estimated values of mean, median and sd as well as the number of samples used for the histogram are shown. The histogram itself can be adjusted by setting the number of uniquely distributed breaks. In Fig. 5.12 this number is increased. Here it can be also seen, that for each computable variable a selection via a drop down menu is possible.

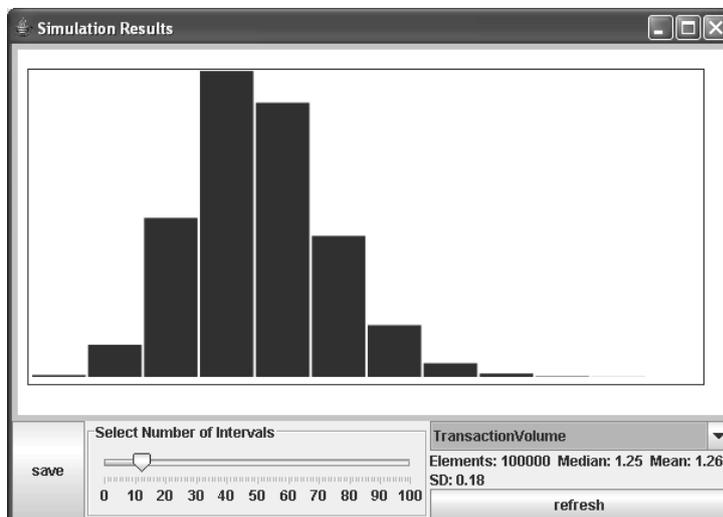


Figure 5.11: Simulation Results (a)

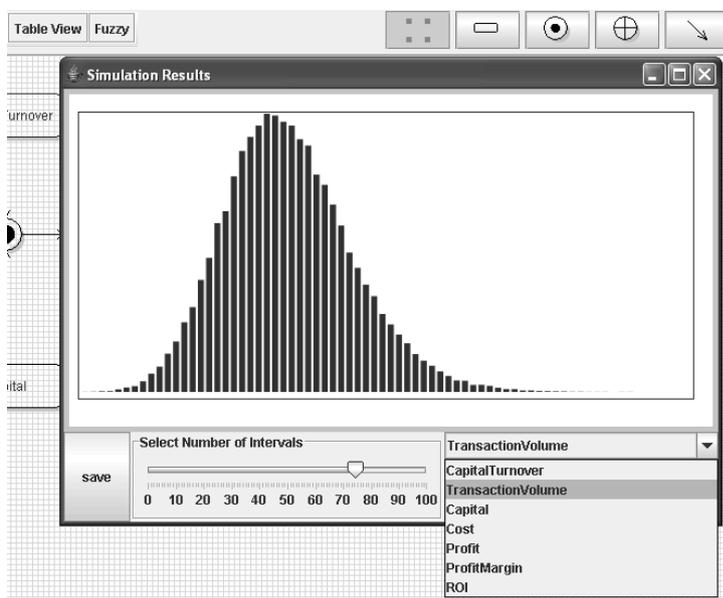


Figure 5.12: Simulation Results (b)

Chapter 6

Conclusion

This chapter summarises the results of this thesis and provides an outlook on future work in the area of stochastic indicator systems.

6.1 Overview

Controlling needs a quantitative basis to guarantee that decisions are made according to the states of the examined area. Allowing errors in the variables leads to more realistic results, which consequently enhances decisions. Read (1898) stated: “It is better to be vaguely right than exactly wrong.”¹

The research field of control and management with uncertainty encompasses different approaches and methodologies. The choice of the indicator’s error representation depends on the available information and the underlying objectives of the system.

Fuzzy set theory is useful in the context of linguistic statements of the indicators or errors that occur in gathering necessary information. When measurements are possible or the constitutive generating errors can be estimated, it is beneficial to use stochastic representation. This is justified by the fact that Fuzzy descriptions build up an upper bound.

Using stochastic information complicates the process of evaluating indicator systems due to error propagation. Non-linearity causes even higher complexity. Complex dependency structures should be evaluated and used to enhance results for decision making.

Although approximation of non-linear stochastic equation systems is feasible, it increases the possibility of misinterpretation, e. g., inappropriate estimates, such as mean values where no moment exists, or divergences in the estimated values from the real ones.

This thesis shows that simulation is the tool of choice to overcome these problems. Complex dependency structures between variables of the equation system

¹John Maynard Keynes stated it equivalent with: “It is better to be vaguely right rather than precisely wrong.”

can be handled. However, there is an inevitable trade-off. Using simulation instead of approximation increases data accuracy but at the cost of an increased computation effort. Depending on the complexity structure of the model and the number of involved variables, as well as on the stochastic nature of prior knowledge, a decision between accuracy and computation time has to be made.

6.2 Comparison of Methods

One focus of this work is a comparative study of existing methods for errors in the variables models with the new developed simulation approach. The existing selected techniques are Fuzzy set theory and a GLS approximation. Tools exist for both methods. The comparison is made against the developed Software Prototype of MoSi, which is based on the simulation algorithm SamPro. In the following a brief overview of the criteria and their characteristics within the tools is given. This evaluation is a summary as stated in Chapter 4.

Graphical Modelling

Modelling of indicator systems is essential for a good overview of a indicator system. Mathematical formulation and graphical visualisation are a useful combination. Only MoSi provides both model representations. FuzzyCalc states the equations clearly arranged in mathematical formulation, whereas Quantor hides the information of relations in dialog boxes.

Information on errors

Information on errors in the variables depend on the underlying model assumptions. Fuzzy set uses membership functions, whereas GLS and the simulation approach use stochastic information. Approximation with the Kalman filtering technique restricts statistical distribution to a Gaussian regime. FuzzyCalc bounds the domain of membership functions to the most common. MoSi can handle every type of density function by the use of a general MCMC method.

Result Presentation

Results of the approach should warrant an easy access for the user. Fuzzy sets are closed under arithmetic operations. FuzzyCalc handles these functions in a graphical and tabular view. The results generated by Quantor only give estimates on the first and second moments. In case that the Gaussian regime is violated, i.e., quotients of two Gaussian variables, they have to be treated and interpreted very carefully. MoSi provides statistical analysis of the results including moment estimation and graphical interpretation by using histogram plots.

Restriction of Models

Real world indicator systems can be very huge and complex. Model assumptions restrict the usage of approaches. The four basic arithmetic operators are the essential basis for all indicator systems. This implies nonlinearity on the one hand but a separation of equations on the other. All three approaches apply basic arithmetic operations. However, Quantor works with a further restriction that division cannot be applied on quotients. FuzzyCalc and MoSi handle indicator systems similar. A further restriction is the available memory (RAM).

Performance

Performance is a crucial factor in real time decision making. FuzzyCalc and Quantor solve models within a relatively short time. Quantor slightly outperforms FuzzyCalc. Simulation takes a much larger computational effort. Moreover, the performance of MoSi depends on the random number generator as well as the dimensions within the model. Increasing precision is equivalent with an increase of sample size and therefore with the computational effort.

In this thesis a simulation approach for handling indicator systems with uncertainty has been presented. It is not only an appropriate method but also outperforms existing methods whereas it relaxes made assumptions. The SamPro method represents reality in a better way. A higher accuracy for decision makers is therefore possible. A disadvantage of the simulation method is the computation effort. A reduction of the computation effort might be a future research area. The simulation approach should be applied for long term planning and strategy as well as for critical application scenarios where precision plays an important role.

MoSi is a user friendly modelling and simulation tool for indicator systems. It's core is the SamPro algorithm and it enables easily prior knowledge integration and comprehensive result presentation. In short, managers obtain a suitable tool for controlling and decision making.

6.3 Contribution

In this section the main contributions of the thesis are presented.

State of the Art of Indicator Systems

Controlling companies or economies is based on indicators. Due to the increasing complexity managers are in the need for decision support systems. Presently, these indicators are treated as crisp values, although they inherent uncertainty due to measurement or prediction errors. To relax this unrealistic view, indicator systems with uncertainty are investigated in this thesis.

A further topic in indicator systems usage is data validation where discrepancies within the data and the indicator system are revealed and corrected.

Approaches for handling indicator systems with uncertainty are proposed. (Müller and Lenz 2003) implemented a Fuzzy set approach in the Excel-Plugin FuzzyCalc. Schmid (1979) developed a GLS approach which is implemented in Quantor (Müller 1991). Both approaches make assumptions that are relaxed in the simulation approach SamPro which is developed in this thesis. The relaxation leads to an improved mapping of reality.

SamPro: Simulation Algorithm

In this thesis a method for simulation of indicator systems with uncertainty is developed based on Markov Chain Monte Carlo (MCMC) simulation.

MCMC simulation is a field where complex distributions can be investigated. Hastings (1970) developed an algorithm (MH algorithm) that generates samples for multivariate, complex distributions. This algorithm is used and embedded in the context of indicator systems. Challenges of the MH algorithm, like kernel selection and burn-in are discussed and the SamPro algorithm is developed. SamPro enables the use of indicator system with uncertainty. Correlation between indicators, non-normality of distributions, and non-linear equations within the indicator system are considered.

The properties of SamPro are:

- Revelation of inconsistencies between indicator system and given data.
- Estimation of consistent indicators.
- Imputation of missing values of indicators.
- Reduction of uncertainty for indicators.

Robustification

The simulation of nonlinear equation systems might generate distributions where robust methods have to be applied. Robust methods for the SamPro algorithm are discussed. Two major robustification approaches can be applied in the algorithm. On the one hand for each indicator an outlier elimination can be done. On the other hand, to maintain the structure within the indicator system, a multivariate outlier elimination can be performed.

Outlier elimination reduces the uncertainty intervals and estimation based on the indicators becomes more stable.

Comparison Study

Chapter 4 presents a comparative study between the new developed SamPro algorithm and two other approaches for handling indicator systems with uncertainty.

The results of this study are:

- SamPro achieves a more effective reduction of uncertainty intervals than the Fuzzy set approach which is due to the results given by Dubois and Prade (2006).
- SamPro achieves better estimates in the context of indicator systems with nonlinear equations.
- SamPro identifies inconsistencies between the indicator model and data in contrast to the GLS approach.
- SamPro represents reality better than the two other approaches due to the use of manifold classes of distributions.
- SamPro considers correlations between indicators, which is not possible with the other two approaches.

Software Tool MoSi

In order to achieve acceptance and usage of the developed MCMC simulation method in practice it is necessary to implement the SamPro algorithm in an easy to use software tool.

For this reason MoSi is developed. The user interface (GUI) is based on Java, the simulation algorithm is implemented in R. This combination enables platform independence and yet fast simulation processing. The simulation results can be easily visualised with the Java based interface.

Indicator systems with uncertainty become applicable in practice by the use of MoSi. The decision maker is enabled to use consistent indicators or to reveal inconsistencies. A reduction of uncertainty is also achieved and decisions can be put on a firm footing.

6.4 Future Work

In this section, hints on further development in the area of stochastic indicator systems are provided.

One particularly interesting domain of further research is simulation efficiency and algorithmic improvements. Using general methods, such as the Metropolis Hastings algorithm, makes the use of more simulation information as prior knowledge possible, but increases the simulation's running time. The implementation of more efficient algorithms for specific, but frequently used density functions can reduce the simulation time, while maintaining the quality of the simulation results. Another area of improvement is the implementation of multi threading random number generators. This can further reduce computation time and improve simulation in contrast to other approaches.

The equation model assumption for simulation is the separation principle. Therefore, further operations might be investigated. In addition to that, implementation of the domain restriction for variables might be useful. This not

only improves interpretation and validity checking but also expands the domain of application. Introducing for instance monotone transformations like the logarithm or the exponential function broadens the area of indicator models.

Another hot topic is prior or expert knowledge. This is split into two areas. On the one hand, available data of indicators can be used for multivariate density estimation. On the other hand, knowledge is encapsulated in the structure of an equation system. Using existing business or economic data leads to new relations and equations of indicators. This abolishes the assumption of the validity of the model, which is a basic assumption of this work, but might reveal other relations between indicators that are not yet clearly visible.

In conclusion it can be said that a sign hanging in Albert Einstein's office at Princeton sums up the core thesis of this work most aptly: "Not everything that counts can be counted, and not everything that can be counted counts".

Bibliography

- Aichele, C.: 1997. *Kennzahlenbasierte Geschäftsprozeßanalyse*. Schriften zur EDV-orientierten Betriebswirtschaft. Betriebswirtschaftlicher Verlag Dr. Th. Gabler, Wiesbaden.
- Aitchison, J.: 1986. *The Statistical Analysis of Compositional Data*. Chapman and Hall. London.
- Albach, H.: 1961. Entscheidungsprozeß und Informationsfluß in der Unternehmensorganisation. *in* Schnauffer and Agthe (1961). pp. 335–402.
- Allgeier, M.: 2006. *MCMC Simulation der Balanced Scorecard-Kenngrößen*. Master's thesis. Freie Universität. Berlin.
- Antoine, H.: 1958. *Kennzahlen, Richtzahlen, Planungszahlen*. 2nd edn. Betriebswirtschaftl. Verl. Gabler. Wiesbaden.
- Arndt, H.-K., Graubitz, H. and Köppen, V.: 2007. Topic maps for representing balanced scorecards. *in* G. M. Dimirovski and F. Ulengin (eds), *11th IFAC Symposium on Computational Economics and Financial and Industrial Systems*. Istanbul, Turkey. pp. 120–125.
- Arulampalam, S., Maskell, S., Gordon, N. and Clapp, T. 2002. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* **50**(2), 174–188.
- Azzalini, A. and Capitanio, A. 1999. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61**(3), 579–602.
- Azzalini, A. and Dalla Valle, A. 1996. The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.
- Barnett, V. 1976. The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A (General)* **139**(3), 318–355.
- Becker, J., Kugeler, M. and Rosemann, M. (eds): 2005. *Prozessmanagement - Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. 5 edn. Springer, Berlin, Heidelberg, New York.

- Bennett, J., Racine-Poon, A. and Wakefield, J.: 1996. *Markov Chain Monte Carlo in Practice*. in Gilks et al. (1996b). chapter MCMC for nonlinear hierarchical models, pp. 339–357.
- Berg, B. A. and Neuhaus, T. 1992. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters* **68**(1), 9.
- Besag, J., Green, P., Higdon, D. and Mengersen, K. 1995. Bayesian computation and stochastic systems (with discussion). *Statistical Science* **10**, 3–66.
- Bothe, H.-H.: 1995. *Fuzzy Logic - Einführung in Theorie und Anwendungen*. Springer, Berlin.
- Büning, H.: 1991. *Robuste und adaptive Tests*. de Gruyter, Berlin, New York.
- Bureau of Economic Analysis, United States Department of Commerce 2000. Gdp: One of the great inventions of the 20th century. *Survey of Current Business* (1), 6–14.
- Carlin, B. P., Gelfand, A. E. and Smith, A. F. M. 1992. Hierarchical bayesian analysis of changepoint problems. *Applied Statistics* **41**(2), 389–405.
- Carson, C. S. and Honsa, J. 1990. The united nations system of national accounts: An introduction. *Survey of Current Business* (6), 20–30.
- Chib, S.: 2004. *Markov Chain Monte Carlo Technology*. in Gentle et al. (2004). chapter Markov Chain Monte Carlo Technology, pp. 71–102.
- Chib, S. and Greenberg, E. 1995. Understanding the metropolis-hastings algorithm. *American Statistician* **49**(49), 327–335.
- Creutz, M. 1980. Monte carlo study of quantized SU(2) gauge theory. *Physical Review D* **21**(8), 2308 – 2315.
- Crovella, M. E. and Bestavros, A. 1997. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.* **5**(6), 835–846.
- DeMarco, T.: 1982. *Controlling Software Projects: Management, Measurement and Estimation*. Yourdon Press. Englewood Cliffs, NJ.
- Doucet, A., de Freitas, N. and Gordon, N. (eds): 2001. *Sequential Monte Carlo Methods in Practice*. Springer, New York.
- Doucet, A., Godsill, S. and Andrieu, C. 2000. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing* **10**, 197–208.
- Dubois, D. and Prade, H. 1992. When upper probabilities are possibility measures. *Fuzzy Sets and Systems* **49**, 65–74. DP164.

- Dubois, D. and Prade, H.: 2006. Possibility Theory and its applications: a retrospective and prospective view.. *in* G. D. Riccia, D. Dubois, R. Kruse and H.-J. Lenz (eds), *Decision theory and multi-agent planning*. CISM Course and Lectures n°482. Springer. <http://www.springerlink.com>. pp. 89–100.
- Dubois, D., Foulloy, L., Mauris, G. and Prade, H. 2004. Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities. *Reliable Computing* **10**, 273–297.
- Efron, B.: 1982. *The Jackknife, the Bootstrap and Other Resampling Plans*. CBMS-NSF Regional Conference Series in Applied Mathematics, Monograph 38. SIAM. Philadelphia.
- Ermakov, S., Melas, V. and Pepelyshev, A. (eds): 2005. *Proceeding of the Fifth Workshop on Simulation*. NII Chemistry Saint Petersburg University Publishers.
- Friedag, H. R. and Schmidt, W.: 2002. *Balanced Scorecard*. TaschenGuide ; 61. 2 edn. Haufe. Planegg.
- Galton, F. 1890. Dice for statistical experiments. *Nature* **42**(1070), 13–14.
- Gamerman, D.: 2002. *Markov Chain Monte Carlo Stochastic Simulation for Bayesian Inference*. Chapman & Hall / CRC.
- Gamerman, D. and Lopez, H. F.: 2006. *Markov Chain Monte Carlo : Stochastic Simulation for Bayesian Inference*. 2nd edn. Chapman & Hall / CRC.
- Gelman, A. and Rubin, D. B. 1992. Inference from iterative simulation using multiple sequences. *Statistical Science* **7**(4), 457–472.
- Gelman, A., Roberts, G. O. and Gilks, W. R.: 1996. Efficient metropolis jumping rules. *in* J. M. Bernardo, J. O. Berger, A. Dawid and A. F. M. Smith (eds), *Bayesian Statistics 5*. Clarendon Press. Oxford.
- Geman, S. and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**(6), 721–741.
- Gentle, J. E., Härdle, W. and Mori, Y. (eds): 2004. *Handbook of Computational Statistics - Concepts and Methods*. Springer, Heidelberg, Berlin, New York.
- Geweke, J.: 1992. Evaluating the accuracy of sampling based approaches to the calculation of posterior moments. *in* J. O. Berger, J. Bernardo, A. Dawid and A. Smith (eds), *Bayesian Statistics 4*. Oxford University Press. Oxford. pp. 169–194.

- Geyer, C. J.: 1991. Markov chain monte carlo maximum likelihood.. in E. Kerami-gas (ed.), *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface*. Interface Foundation. Fairfax. pp. 156–163.
- Geyer, C. J. 1992. Practical markov chain monte carlo. *Statistical Science* **7**(4), 473–483.
- Geyer, C. J. and Thompson, E. A. 1995. Annealing markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association* **90**(431), 909–920.
- Gilks, W. R., Richardson, S. and Spiegelhalter, D. J.: 1996a. *Introducing Markov Chain Monte Carlo*. in . chapter Introducing Markov Chain Monte Carlo, pp. 1–19.
- Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds): 1996b. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.
- Gladen, W.: 2003. *Kennzahlen- und Berichtssysteme. Grundlagen zum Performance Measurement*. 2nd edn. Gabler Verlag, Wiesbaden.
- Glauber, R. J. 1963. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics* **4**(2), 294–307.
- Gordon, N. J., Salmond, D. J. and Smith, A. F. M. 1993. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F (Radar and Signal Processing)* **140**(2), 107–113.
- Hammersley, J. M. and Handscomb, D. C.: 1964. *Monte Carlo Methods*. Methuen, London.
- Hastings, W. K. 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**, 97–109.
- Hausmann, A.: 2004. *Kennzahlensysteme in der BWL: Simulation mittels Particle Filter in Anwendung*. Master’s thesis. Freie Universität Berlin. Berlin.
- Hawkins, D. M.: 1980. *Identification of Outliers*. Monographs on Statistics and Applied Probability. Chapman and Hall. London ; New York.
- Hesselbo, B. and Stinchcombe, R. B. 1995. Monte carlo simulation and global optimization without parameters. *Physical Review Letters* **74**(12), 2151–2155.
- Holley, R. A., Kusuoka, S. and Stroock, D. W. 1989. Asymptotics of the spectral gap with applications to the theory of simulated annealing. *Journal of Functional Analysis* **83**(Issue 2), 333–347.
- Hornik, K.: 2007. *The R FAQ*. ISBN 3-900051-08-9.

- Horstmann, C.: 2005. *Object Oriented Design and Patterns*. 2nd edn. Wiley & Sons. New York.
- Horváth, P.: 1994. *Controlling*. Verlag Vahlen, München.
- Huang, H. W. 1973. Time-dependent statistics of the ising model in a magnetic field. *Physical Review* **A8**(5), 2553.
- Ingham, H. and Harrington, L. T.: 1962. *Interfirm Comparison for Management*. 2nd edn. Millbrook Press.
- Jarrett, R. G. 1979. A note on the intervals between coal-mining disasters. *Biometrika* **66**(1), 191–193.
- Jesinghaus, J.: 2000. On the art of aggregating apples & oranges. *EVE Workshop*.
- Kaplan, R. S. and Norton, D. P. 1992. The balanced scorecard - measures that drive performance. *Harvard Business Review* **1**, 71–79.
- Kaplan, R. S. and Norton, D. P.: 1996. *The Balanced Scorecard. Translating Strategy Into Action*. Harvard Business School Press.
- Kaplan, R. S. and Norton, D. P.: 2004. *Strategy Maps: Converting Intangible Assets Into Tangible Outcomes*. Harvard Business School Press.
- Kemp, A. and Kasim, S. 2003. An econometric model of oil and gas exploration development and production in the uk continental shelf: a systems approach. *The Energy Journal* **24**(2), 113–141.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* **220**(4598), 671–680.
- Köppen, V., Allgeier, M. and Lenz, H.-J.: 2007. Balanced scorecard simulator – a tool for stochastic business figures. in R. Decker and H.-J. Lenz (eds), *Advances in Data Analysis*. Springer. Berlin, Heidelberg. pp. 457–464.
- Köppen, V. and Lenz, H.-J.: 2005. Simulation of non-linear stochastic equation systems. in A. P. S.M. Ermakov, V.B. Melas (ed.), *Proceeding of the Fifth Workshop on Simulation*. NII Chemistry Saint Petersburg University Publishers. St. Petersburg, Russia. pp. 373–378.
- Köppen, V. and Lenz, H.-J.: 2006. A comparison between probabilistic and possibilistic models for data validation. in V. M. Rizzi A (ed.), *Proceeding in Computational Statistics COMPSTAT 2006*. Rome. pp. 1533–1541.
- Kraemer, W.: 1991. Wissensbasierter Controlling-Leitstand: Zeitnahe Kostenanalysen zur Reduzierung der Unsicherheit. in Scheer (1991). pp. 388–418.
- Kraemer, W.: 1993. *Effizientes Kostenmanagement: EDV-gestützte Datenanalyse und -interpretation durch den Controlling-Leitstand*. Gabler. Wiesbaden.

- Kruse, R., Gebhardt, J. and Klawonn, F.: 1995. *Fuzzy-Systeme*. 2nd edn. Teubner, Stuttgart.
- Kuznets, S. S.: 1971. *Economic Growth of Nations: Total Output and Production Structure*. Harvard University Press.
- Lachnit, L.: 1998. *Zukunftsfähiges Controlling*. Vahlen Verlag, München. chapter Das Rentabilitäts-Liquiditäts-(R/L) Kennzahlensystem als Basis controllinggestützter Managementkonzepte, pp. 19–44.
- Lachnit, L. and Müller, S.: 2006. *Unternehmenscontrolling*. Betriebswirtschaftlicher Verlag Dr. Th. Gabler, Wiesbaden.
- Lauzel, P. and Cibert, A.: 1973. *Des Ratios au Tableau de Bord*. 5 edn. Enterprise Moderne D’Edition. Paris.
- Lenz, H.-J. and Müller, R. M.: 2000. On the solution of fuzzy equation systems. in G. D. Riccia, H.-J. Lenz and R. Kruse (eds), *Computational Intelligence in Data Mining*. CISM Courses and Lectures. Springer, New York.
- Lenz, H.-J. and Rödel, E.: 1991. Statistical quality control of data. in P. Gritzmann, R. Hettich, R. Horst and E. Sachs (eds), *16th Symposium on Operations Research*. Physica Verlag, Heidelberg. pp. 341–346.
- Lenz, H.-J., Köppen, V. and Müller, R. M.: 2006. Edits - data cleansing at the data entry to assert semantic consistency of metric data. *SSDBM '06: Proceedings of the 18th International Conference on Scientific and Statistical Database Management*. IEEE Computer Society. Washington, DC, USA. pp. 235–240.
- Liu, J. S.: 2001. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, Berlin.
- MacKay, D. J. C.: 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
- Maguire, B. A., Pearson, E. S. and Wynn, A. H. A. 1952. The time intervals between industrial accidents. *Biometrika* **39**(1/2), 168–180.
- Marinari, E. and Parisi, G. 1992. Simulated tempering: a new monte carlo scheme. *Europhysics Letters* **19**(6), 451–458.
- Marsaglia, G. and Zaman, A. 1993. Monkey tests for random number generators. *Computers & Mathematics with Applications* **26**(9), 1–10.
- Marshall, A.: 1956. The use of multi-stage sampling schemes in monte carlo computations. in H. A. Meyer (ed.), *Symposium on Monte Carlo Methods*. John Wiley, New York. pp. 123–140.

- März, T.: 1983. *Interdependenzen in einem Kennzahlensystem: eine empirische Untersuchung zur Aussagefähigkeit von Kennzahlen bei der Unternehmensanalyse*. Hochschulschriften zur Betriebswirtschaftslehre. V. Florentz GmbH. München.
- Matsumoto, M. and Kurita, Y. 1992. Twisted gfsr generators. *ACM Trans. Model. Comput. Simul.* **2**(3), 179–194.
- Matsumoto, M. and Nishimura, T. 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30.
- McDonough, A.: 1963. *Information Economics and Management Systems*. MCGraw-Hill Books, New York.
- Mengersen, K. L. and Tweedie, R. L. 1996. Rates of convergence of the hastings and metropolis algorithms. *The Annals of Statistics* **24**(1), 101–121.
- Metropolis, N. 1987. The beginning of the monte carlo method. *Los Alamos Science* **15**, 125–130.
- Metropolis, N. and Ulam, S. 1949. The Monte Carlo Method. *Journal of American Statistical Association* **44**, 335–341.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. 1953. Equations of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**(6), 1087–1092.
- Meyer, C.: 1976. *Betriebswirtschaftliche Kennzahlen und Kennzahlen-Systeme*. Poeschel Verlag, Stuttgart.
- Moroney, J. R. and Berg, M. D. 1999. An integrated model of oil production. *The Energy Journal* **20**(1), 105–124.
- Müller, P. H. (ed.): 1975. *Lexikon der Stochastik*. 2nd edn. Akademie Verlag. Berlin.
- Müller, R.: 1990. *Computerunterstützte Planung- und Kontrollrechnung auf der Basis des ZVEI-Kennzahlensystems*. PhD thesis. Justus-Liebig-Universität Gießen.
- Müller, R. A.: 1988. Is there any need for long-term forecasting methods?. in Schmid and Signer (1988). pp. 9–28.
- Müller, R. A.: 1991. Kostenplanung mit unscharfen Daten. in Scheer (1991). pp. 365–387.
- Müller, R., Lenz, H.-J. and Ruhnke, K. 2003. Ein fuzzybasierter Ansatz zur Durchführung analytischer Prüfungen bei der Existenz von Schätzspielräumen. *Die Wirtschaftsprüfung* **56**(10), 532–541.

- Müller, R. M.: 1999. *Controlling, Planung und Prognose mit unscharfen Daten*. Master-thesis. Freie Universität Berlin.
- Müller, R. M. and Lenz, H.-J.: 2003. FuzzyCalc - Analytische Prüfung mit Fuzzymethoden: Theorie, Anwendung, Beispiele. in J. Geldermann and H. Rommelfanger (eds), *Einsatz von Fuzzy Sets, Neuronalen Netzen und Künstlicher Intelligenz in industrieller Produktion und Umweltforschung*. number 725 in *Fortschritt-Berichte VDI, Reihe 10*. VDI-Verlag. Düsseldorf. pp. 123–142.
- Neumann, J. 1951. Various techniques used in connection with random digits. *National Bureau of Standards applied mathematics series* **12**, 36–38.
- Preparata, F. P. and Hong, S. J. 1977. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM* **20**(2), 87–93.
- Probst, H.-J.: 2001. *Balanced Scorecard leicht gemacht: Warum Sie mit weichen Faktoren hart rechnen sollten*. Wirtschaftsverlag Ueberreuter.
- R Development Core Team: 2007. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.
- Raftery, A. E. and Lewis, S. M.: 1992a. How many iterations in the gibbs sampler. in J. O. Berger, J. Bernardo, A. Dawid and A. Smith (eds), *Bayesian Statistics 4*. Oxford University Press. Oxford. pp. 763–773.
- Raftery, A. E. and Lewis, S. M. 1992b. [practical markov chain monte carlo]: Comment: One long run with diagnostics: Implementation strategies for markov chain monte carlo. *Statistical Science* **7**(4), 493–497.
- Read, C.: 1898. *Logic: Deductive and Inductive*. Grant Richards. London.
- Reichmann, T.: 2006. *Controlling mit Kennzahlen und Management-Tools : die systemgestützte Controlling-Konzeption*. 7th edn. Vahlen, München.
- Reichmann, T. and Lachnit, L. 1976. Planung, Steuerung und Kontrolle mit Hilfe von Kennzahlen. *ZfbF* **28**, 705–723.
- Robert, C. P. 1995. Convergence control methods for markov chain monte carlo algorithms. *Statistical Science* **10**(3), 231–253.
- Robert, C. P. and Casella, G.: 2003. *Monte Carlo statistical methods*. Springer texts in statistics. 2 edn. Springer. New York et.al.
- Roberts, G. O.: 1996. *Markov Chain Concepts related to Sampling Algorithms*. in Gilks et al. (1996b). chapter Markov Chain Concepts related to Sampling Algorithms, pp. 45–57.

- Rommelfanger, H.: 1994. *Fuzzy Decision Support-Systeme - Entscheiden bei Unschärfe*. 2nd edn. Springer, Berlin.
- Rosemann, M., Schwegmann, A. and Delfmann, P.: 2005. *Vorbereitung der Prozessmodellierung*. in Becker et al. (2005). 5 edn. chapter 3, pp. 45–103.
- Rubin, D. B. 1987. The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association* **82**(398), 543–546.
- Scheer, A.-W. (ed.): 1991. *Rechnungswesen und EDV - 12. Saarbrücker Arbeitstagung 1991*. Physica-Verlag. Heidelberg.
- Schmid, B.: 1979. Bilanzmodelle. Simulationsverfahren zur Verarbeitung unscharfer Teilinformationen. *Technical Report 40*. ORL-Institut. ETH Zürich.
- Schmid, B.: 1988. Procedure for the representation and transformation of information on complex systems (PRTI). in Schmid and Signer (1988).
- Schmid, B. and Signer, R. (eds): 1988. *Information on Complex Systems – Representation and Inference*. number 61 in *Berichte zur Orts-, Regional- und Landesplanung*. Institut für Orts-, Regional- und Landesplanung ETH Zürich. Verlag der Fachvereine. Zürich.
- Schnauffer, E. and Agthe, K. (eds): 1961. *Organisation*. Vol. 1 of *TFB-Handbuchreihe*. Deutscher Betriebswirte-Verlag GmbH.
- Schott, G.: 1988. *Kennzahlen. Instrument der Unternehmensführung*. 5th edn. Forkel. Wiesbaden.
- Schwarz, R.: 2002. *Controlling-Systeme*. Die Wirtschaftswissenschaften. Betriebswirtschaftlicher Verlag Dr. Th. Gabler GmbH. Wiesbaden.
- Scott, D. W.: 1992. *Multivariate density estimation : theory, practice, and visualization*. Wiley. New York.
- Scott, D. W.: 2004. *Multivariate Density Estimation and Visualization*. in Gentle et al. (2004). pp. 517–538.
- Scott, D. W. and Sain, S. R.: 2005. *Multi-dimensional density estimation*. Elsevier Publishing. Amsterdam, The Netherlands, The Netherlands. chapter Multi-dimensional density estimation, pp. 229–261.
- Silverman, B. W.: 1986. *Density Estimation for Statistics and Data Analysis*. Vol. 26 of *Monographs on Statistics and Applied Probability*. Chapman & Hall.

- Smith, A. F. M. and Gelfand, A. E. 1992. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician* **46**(2), 84–88.
- Spremann, K. and Bamberg, G. 1984. Repräsentative Informationen in linearen Systemen. *OR Spektrum* **6**, 23–37.
- Staehle, W. H.: 1969. *Kennzahlen und Kennzahlensysteme*. Betriebswirtschaftlicher Verlag Dr. Th. Gabler, Wiesbaden.
- Statistisches Bundesamt: 2006. Qualitätsbericht Volkswirtschaftliche Gesamtrechnungen (VGR).
- Tanner, M. A.: 1993. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*. Springer Verlag, New York, Berlin, Heidelberg.
- Tanner, M. A. and Wong, W. H. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* **82**, 528–549.
- Tierney, L. 1994. Markov chains for exploring posterior distributions. *The Annals of Statistics* **22**(4), 1701–1728.
- Tucker, S. A.: 1961. *Successful Managerial Control by Ratio-Analysis*. McGraw Hill Book Company.
- UN Statistics Division: 1993. *System of National Accounts 1993*. Commission of the European Communities, International Monetary Fund, Organisation for Economic Co-operation and Development, United Nations and World Bank.
- Urbanek, S.: 2007. *rJava: Low-level R to Java interface*. R package version 0.5-0.
- Walsh, J.: 2004. Markov chain monte carlo and gibbs sampling. Lecture Notes for EEB 581.
- Wolfram, S.: 2002. *A new Kind of Science*. Wolfram Media.
- Zadeh, L. A. 1965. Fuzzy sets. *Information and Control* **8**, 338–353.
- Zadeh, L. A. 1975a. The concept of a linguistic variable and its application to approximate reasoning - i. *Information and Control* **8**, 199–249.
- Zadeh, L. A. 1975b. The concept of a linguistic variable and its application to approximate reasoning - ii. *Information and Control* **8**, 301–357.
- Zadeh, L. A. 1975c. The concept of a linguistic variable and its application to approximate reasoning - iii. *Information and Control* **9**, 43–80.

Zentralverband Elektrotechnik- und Elektronikindustrie e.V. ZVEI (ed.): 1989. *ZVEI-Kennzahlensystem*. 4th edn. Betriebswirtschaftlicher Ausschuß des Zentralverbandes Elektrotechnik- und Elektronikindustrie (ZVEI) e.V., Frankfurt am Main.

Zimmermann, H.-J.: 1991. *Fuzzy Set Theory – and its Applications*. 2nd and revised edn. Kluwer Academic Publishers, Boston.

Index

- Balanced Scorecard (BSC), 24, 150
Bootstrapping, 63
- Distribution
- Beta, 50
 - Bivariate, 44
 - Cauchy, 48
 - Exponential, 49
 - Gamma, 49
 - Gaussian, 44
 - Inverse Gaussian, 50
 - Laplace, 46
 - Multivariate Skew Normal, 51
 - Pareto, 47
 - Students't, 47
 - Triangular, 45
 - Uniform, 46
- DuPont, *see* Indicator System, 16, 104
- Fuzzy set, 36
- Gibbs Sampler, 76, 117
- Importance Sampling, 59
- Indicator, 5, 9
- Business, 8, 10, 14
 - Economic, 8, 28
- Indicator System, 5, 11, 14, 28
- DuPont, 14, 22, 54, 81, 104, 122, 129, 169
 - Managerial Control Concept, 21
 - Measures of National Income and Output, 28
 - Pyramid Structure of Ratios, 22
 - Ratio au Tableau de Bord, 21
 - RL business indicator system, 19
- United Nations System of National Accounts, 31
- ZVEI, 17, 22
- Managerial Control Concept, *see* Indicator System
- Markov Chain, 70, 74
- Markov chain, 67
- Markov Chain Monte Carlo (MCMC), 67, 71, 74, 76, 90, 122, 149, 157, 178
- Measures of National Income and Output, *see* Indicator System
- Metropolis algorithm, 69
- Metropolis Hastings algorithm, 73–75, 77, 92, 117, 168, 178
- Independence Chain, 91
 - Random Walk, 91
 - Simulated Annealing, 97
- Monte Carlo Simulation, 57
- Pyramid Structure of Ratios, *see* Indicator System
- QR, 54
- Ratio au Tableau de Bord, *see* Indicator System
- RL business indicator system, *see* Indicator System
- Sequential Importance Sampling, 60
- United Nations System of National Accounts, *see* Indicator System
- ZVEI, *see* Indicator System

Appendices

Appendix A

Mersenne Twister

Mersenne Twister (Matsumoto and Nishimura 1998) is a fast pseudo random number generator with a period of $2^{19937} - 1$. The name is derived from the Mersenne prime which has the form $2^n - 1$. It is a twisted generalised feedback shift register (Matsumoto and Kurita 1992). Two different implementations, one for 32 bit and one for 64 bit word length are available and generate different sequences. Generated sequences have a negligible serial correlation between successors. Algorithm 12 depicts the algorithm for 32 bit length.

This algorithm is very efficient and fast. Mersenne Twister is the fastest generator compared to all other PRNGs with adequate statistical properties. The numbers of the pseudo random sequence are uniformly distributed. Due to these facts the MT19937 will be used in the former simulation for generating uniformly distributed numbers although the sequence can be determined if at least 624 but maximal 1247 numbers in a row are known. The most important fact for the usage of the Mersenne Twister is that the algorithm generates random numbers that pass the Diehard tests (Marsaglia and Zaman 1993) but also other tests for statistical randomness.

Algorithm 12 Mersenne Twister - MT19937

Input: *seed* – starting seed*n* – number of random numbers**Output:** $x = \{x_1, \dots, x_n\}$ – sample

```
1: set  $MT[ ] := \text{int}[0..623]$ 
2:  $t := 1$ 
3: set  $MT[0] := \text{seed}$ 
4: for  $i := 1$  to 623 do
5:    $MT[i] := \text{last 32 bit of } (69069 \cdot MT[i - 1]) + 1$ 
6: end for
7: repeat
8:   if  $t \equiv_{624} 0$  then
9:     for  $i := 0$  to 622 do
10:       $y := \text{32nd bit of } MT[i] + \text{last 31 bits of } M[i+1]$ 
11:      if  $y \equiv_2 0$  then
12:         $MT[i] := MT[(i+397) \% 624] \text{ XOR right shift } y$ 
13:      else
14:         $MT[i] := MT[(i+397) \% 624] \text{ XOR right shift } y \text{ XOR } 2567483615$ 
15:      end if
16:    end for
17:     $y := \text{32nd bit of } MT[623] + \text{last 31 bits of } M[0]$ 
18:    if  $y \equiv_2 0$  then
19:       $MT[623] := MT[396] \text{ XOR right shift } y$ 
20:    else
21:       $MT[623] := MT[396] \text{ XOR right shift } y \text{ XOR } 2567483615$ 
22:    end if
23:  end if
24:   $y := MT[t \% 624]$ 
25:   $y := y \text{ XOR right shift by 11 bits of } y$ 
26:   $y := y \text{ XOR (left shift by 7 bits of } y \text{ AND } 2636928640)$ 
27:   $y := y \text{ XOR (left shift by 15 bits of } y \text{ AND } 4022730752)$ 
28:   $y := y \text{ XOR right shift by 18 bits of } y$ 
29:   $x_t := y$ 
30:  increment  $t$ 
31: until  $t = n$ 
```

Appendix B

German Summary

Ausgangslage und Zielsetzung

Die Steuerung von Unternehmen erfolgt heutzutage auf Basis von Kennzahlen. Die Komplexität von Unternehmen ist dabei soweit angewachsen, dass ein Überblick über die Gesamtheit des Unternehmens einem einzelnen Manager nicht möglich ist. Entscheidungsunterstützende Systeme bereiten die operativen Daten eines Unternehmens nicht nur in aggregierter tabellarischer Form auf, sondern sind immer häufiger in mobilen Umgebungen platziert und nutzen grafische Darstellungsweisen. Doch die beste Visualisierung nutzt nichts, wenn die berechneten Kennzahlen widersprüchlich bzw. fehlerhaft sind. Zwei mögliche Quellen für fehlerhafte Kennzahlen sind die Manipulation der zugrunde liegenden Daten oder die Nichtberücksichtigung der in den Daten inheränten Unsicherheit. Dies sind die beiden Problemfelder, mit denen sich die vorliegende Arbeit beschäftigt.

Ob Daten manipuliert wurden, lässt sich durch eine Datenvalidierung ermitteln. Bei der Zusammenführung von Modell und Daten kann es zu einem Konflikt kommen, wenn die Daten nicht mit den Regeln, die durch das Modell festgelegt sind, übereinstimmen. In solch einem Fall können entweder die Daten manipuliert worden sein oder aber das Modell genügt nicht in ausreichender Weise der Abbildung der Realität. Da Kennzahlensysteme Aggregationen in wohldefinierten Funktionsräumen darstellen, ist von der Glaubwürdigkeit der Modelle auszugehen. Dies ist auch eine elementare Annahme für die vorliegende Arbeit. D.h. in dieser Arbeit wird davon ausgegangen, dass nicht valide Kennzahlen durch fehlerhafte Daten verursacht werden.

Herausforderungen bei der Steuerung mittels Kennzahlensystemen bereiten jedoch nicht nur manipulierte Daten, sondern auch Daten unter Unsicherheit. Kennzahlensysteme zur Unternehmensführung oder im Vergleich zwischen Volkswirtschaften berücksichtigen zumeist nicht die Tatsache, dass die vorliegenden Werte mit Unsicherheit belastet sind. Die Ausgangsdaten unterliegen Unsicherheiten, wie zum Beispiel Mess- und Prognosefehlern, Risiko oder Ungewissheit. Die

aggregierten Daten werden zur Erleichterung der Entscheidungsprozesse durch eine Zahl repräsentiert, obwohl eigentlich Zahlenbereiche vorliegen. Die Verwendung exakter Werte vermittelt fälschlicher Weise den Eindruck einer hohen Genauigkeit, jedoch wird ein Informationsverlust erzielt. Die Erstellung von Berichtskennzahlen läuft nach wohldefinierten Regeln ab. Dies können Bilanzgleichungen, Definitionsgleichungen oder Verhaltensgleichungen sein. Durch Fehlerfortpflanzung unterliegen aber auch die aggregierten Kenngrößen dieser Unsicherheit. Auf der einen Seite liegt das Problem also in der nicht ausreichenden Abbildung der Realität durch einzelne Zahlen und auf der anderen Seite im Einfluss von Unsicherheit. Für die Darstellung von numerischer Unsicherheit gibt es bereits verschiedene Modellansätze.

Die von Zadeh (1965) begründete Fuzzy Set Theorie ist eine Möglichkeit zur Darstellung unscharfer Zahlen. Die in Müller (1999) entwickelte Software-Lösung nutzt diesen Ansatz und erreicht neben der Behandlung possibilistischer Kennzahlendarstellungen auch eine Identifikation von Inkonsistenzen zwischen den Daten und den Systemgleichungen.

Ein probabilistischer Ansatz für die Behandlung von Kennzahlensystemen unter Unsicherheit ist entwickelt worden in Schmid (1979). Unter der Normalverteilungsannahme werden hierbei ebenfalls Reduktionen der Unsicherheitsbereiche durch die Verwendung der durch das Gleichungssystem gegebenen Informationen in den Daten erreicht. Dieser Ansatz ist durch ein General-Least-Squares (GLS) Verfahren in Lenz und Rödel (1991) dargestellt.

Während der possibilistische Ansatz nur eine obere Schranke für die vorliegenden Unsicherheiten darstellt (Dubois und Prade 2006), ist die Beschränkung auf die Normalverteilung im zweiten Ansatz sehr restriktiv. Dadurch können Abhängigkeitsstrukturen in den Daten nicht näher berücksichtigt werden. Mehrgipfelige und hochdimensionale Unsicherheitsbereiche werden in den beiden Ansätzen ebenfalls ausser Acht gelassen.

In der vorliegenden Arbeit wird ein Ansatz zur Bewältigung der dargestellten Probleme entwickelt. Dabei sollen die Beschränkungen der bereits bestehenden Ansätze überwunden werden. Aufgrund der Komplexität von linearen und nicht-linearen Funktionen in Kennzahlensystemen und dem Einsatz hochdimensionaler Verteilungen wird hierbei Simulation eingesetzt.

Im operativen Geschäft fallen Daten nicht unabhängig voneinander an. Vielmehr bestehen Zusammenhänge zwischen den Messwerten. Aus diesem Grund ist eine Einschränkung der stochastischen Eingangsgrößen für die Kennzahlenermittlung nicht adäquat. Deshalb müssen für die Simulation ebenfalls multivariate Verteilungen genutzt werden. Dies erhöht unweigerlich die Komplexität bei der Erzeugung der für die Simulation notwendigen Zufallszahlen. Neben den Monte Carlo Verfahren zur Generierung einfacher Zufallszahlen haben sich Markov-Chain-Monte-Carlo (MCMC) Verfahren für komplexe Verteilungen bewährt. Ein allgemeines Verfahren für die Erzeugung von Zufallszahlen aus beliebigen und komplexen Verteilungsfunktionen wurde durch Hastings (1970) entwickelt. Der Metropolis Hastings Algorithmus stellt einen zentralen Bestandteil

der Kennzahlensimulation dar.

Der Metropolis Hastings Algorithmus ist eine allgemeine MCMC-Methode. Aufgrund dieser Allgemeinheit treten die folgenden Herausforderungen auf:

- Die Wahl der Startwerte spielt eine entscheidende Rolle im Laufzeitverhalten des MCMC-Verfahrens. Dies hängt mit dem Erreichen der stationären Verteilung zusammen.
- Die Wahl der kandidatengenerierenden Funktion (Vorschlagsfunktion) hat einen bedeutenden Einfluss auf die Güte der erzeugten Zufallszahlen. Multimodale Verteilungen erfordern beispielweise komplexere Vorschlagsfunktionen.
- Die Bestimmung der Einschwingphase ist für hochdimensionale Verteilungen nicht trivial. Zufallszahlen innerhalb der Einschwingphase dürfen nicht für die Kennzahlensimulation verwendet werden, da diese nicht der eigentlichen Verteilung folgen.

Die Anwendung der Methode hat den Vorteil, dass Zufallszahlenfolgen beliebiger Verteilungsfunktionen unabhängig von ihrer Komplexität erzeugt werden können. Um zufriedenstellende Resultate zu erhalten, ist es nicht ausreichend nur die Güte der einzelnen Simulationsergebnisse zu beachten, sondern auch den Simulationaufwand effizient zu gestalten. Die Wahl der Länge der Simulationsläufe spielt hierbei eine wichtige Rolle. Unter Beachtung der Herausforderungen der MCMC-Methode erzielt man hinreichend gute Simulationsmittel.

Die Anwendung der Modellinformationen, wie zum Beispiel nichtlineare Transformationen, erfordert jedoch zusätzliches Augenmerk während der Simulation. So sind beispielweise die Zwischenergebnisse auf Ausreißer zu untersuchen und diese aus den Simulationsexperimenten zu entfernen. Während der Simulation müssen mithilfe weiterer Plausibilitätsprüfungen ebenfalls fehlerhafte Werte entfernt werden.

Kennzahlensysteme beschränken sich nicht nur auf eine Darstellungsform für jede Kennzahl. Vielmehr ist durch das Einsetzen von Zwischenergebnissen und Umstellen der Gleichungen eine Kennzahl auf vielfältige Weise berechenbar. Dadurch kann es jedoch bei unterschiedlichen Ausgangsdaten zu divergierenden Ergebnissen kommen. Hierbei ist dann zu entscheiden, ob die Ausgangsdaten im Widerspruch zum Kennzahlensystem stehen oder ob ein gemeinsamer Ergebnisraum existiert.

Der Simulationsansatz soll daher zwei Herausforderungen bewältigen. Auf der einen Seite sollen Widersprüche in den Daten zum Modell aufgezeigt werden. An dieser Stelle kann dann keine weitere Aussage zu den Kennzahlen erfolgen, sondern eine Überprüfung der Ausgangsdaten ist zwingend erforderlich. Andererseits sollen stochastische Kennzahlen geliefert werden, die nicht nur den Regeln des Kennzahlenmodells genügen, sondern auch eine geringere Unsicherheit aufweisen sollen.

Kennzahlensimulation

Im Folgenden wird dargestellt, auf welche Weise die Simulation von Kennzahlensystemen erfolgt. Ausgangspunkt ist hierbei ein Kennzahlensystem mit linearen und nichtlinearen Operatoren. In der Praxis treten hierbei fast ausschließlich die Operationen Addition, Subtraktion, Multiplikation und Division auf. Aus diesem Grund erfolgt eine Beschränkung auf diese vier Grundrechenarten. Die Kennzahlensimulation erfordert eigentlich nur Operationen die separierbar sind, d.h. Gleichungen die nach all ihren auftretenden Variablen umgeformt werden können. Somit wären auch weitere Operationen wie Hyperbelfunktionen möglich.

Das Kennzahlensystem stellt ein Wissenssystem dar. Durch die Notwendigkeit der Verwendung von Computern zur Simulation ist die digitale Darstellung der Kennzahlensysteme unabdingbar. Die Verwendung von Topic Maps als Standard der Wissensrepräsentation ermöglicht nicht nur die digitale Darstellung und Navigation durch Kennzahlensysteme, sondern ist auch für komplexere Systeme wie Balanced Scorecards geeignet (Arndt et al. 2007).

Häufig wird in der Literatur der Kennzahlenbaum als Darstellungsform für Kennzahlensysteme gewählt. Hierbei tritt jedoch das Problem auf, dass die gleiche Kennzahl mehrfach dargestellt wird. Um in der Simulation jeder Kennzahl einen konsistenten Wert zuweisen zu können, wird demgegenüber die Datenstruktur des Graphen genutzt.

SamPro

Der Algorithmus zur Simulation von Kennzahlensystemen besteht im Wesentlichen aus zwei Teilen. Zuerst werden Zufallszahlen gemäß den Vorinformationen und des Regelwerks erzeugt. Hierbei werden alle Informationen verwendet, die neues Wissen erzeugen. Somit ist mit dem Regelwerk und den vorhandenen Daten eine Imputation für unbekannte Kennzahlen möglich. Dieser erste Teil der Simulation wird als Sampling-Phase bezeichnet. Darauf schließt sich die Phase der Informationszusammenführung an. Liegen für eine Kennzahl mehrere stochastische Informationen vor, so müssen diese zusammengeführt werden. Hierzu wird der Raum einer jeden Kennzahl über jede ihrer stochastischen Schätzräume aufgespannt. Der Projektionsraum stellt den Bereich dar, in welchem die einzelnen Schätzungen gleich sind. Da im stetigen Intervall die Wahrscheinlichkeit Null beträgt, dass zwei Werte gleich sind, muss ein Toleranzbereich gewählt werden, in dem die Werte der unterschiedlichen Schätzungen als gleich betrachtet werden können. Dieser zweite Schritt wird als Projektionsschritt bezeichnet.

In Abbildung B.1 ist die Projektion von drei Schätzungen f_1 , f_2 und f_3 abgebildet. Die durchgezogene Gerade stellt die Gleichheit der einzelnen Schätzer dar ($f_1 = f_2 = f_3$). Der Korridor ist durch die gestrichelten Linien angedeutet. Alle Werte innerhalb des Bereiches werden für den zusammengeführten Schätzer akzeptiert und auf die durchgezogene Gerade projiziert. Die Werte außerhalb des Bereiches werden für das weitere Vorgehen außer Acht gelassen. Für den Fall, dass

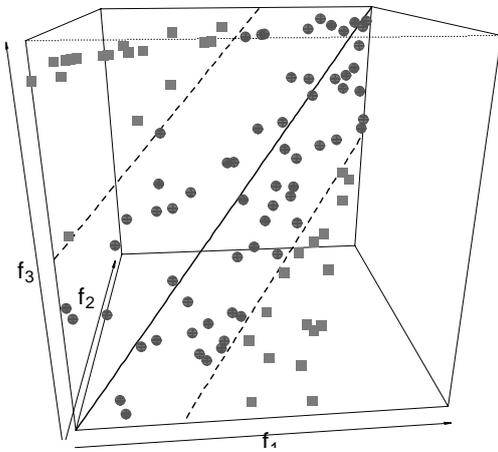


Abbildung B.1: Projektion von drei Schätzungen

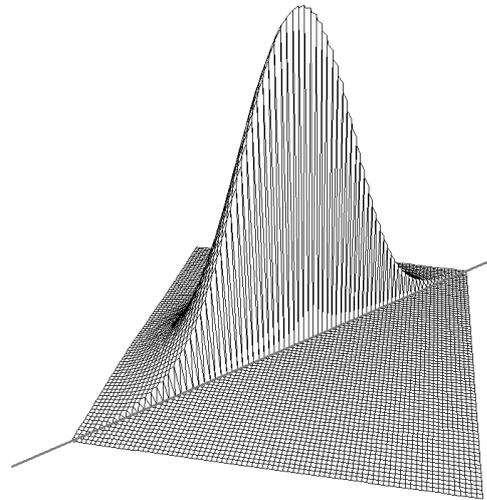


Abbildung B.2: Korrigierte Schätzung

kein Wert im Bereich liegt wird das gesamte Modell als M-inkonsistent gekennzeichnet.

Abbildung B.2 zeigt das Ergebnis der Projektion für zwei Schätzer einer Kennzahl. Die Projektion kann hierbei auch als Projektion auf den Unterraum mit $x = y$ angesehen werden, wobei x die Vorinformationen einer Kennzahl repräsentiert und y die Informationen, die anhand anderer Vorinformationen und des Gleichungssystems gewonnen werden. Es ist dabei zu beachten, dass der zusammengeführte Schätzer normalisiert werden muss.

Für den Fall, dass eine Kennzahl während des Projektionsschrittes als M-inkonsistent ermittelt wurde, muss das Gesamtsystem als M-inkonsistent betrachtet und die Simulation beendet werden. Eine Identifikation der beteiligten Kennzahlen, die zur Inkonsistenz geführt haben, vereinfacht die Analyse der Informationen.

Nachdem die Projektion verschiedener Schätzer für jede Variable verbesserte Schätzer ermittelt hat, kann es sein, dass eine Robustifizierung der Daten notwendig ist. Dies kann dann für jede Variable einzeln erfolgen, wodurch jedoch die Abhängigkeitsstruktur zwischen allen Kennzahlen verloren geht. Eine Ausreißereliminierung im multivariaten Fall stellt sich hingegen aufgrund der Ordnungsstruktur komplizierter dar. Die Verwendung geeigneter Abstände, wie die Mahalanobis Distanz ist für die Robustifizierung im multivariaten Fall notwendig.

Als Resultat des SamPro Algorithmus erhält man somit entweder die Identifizierung einer Inkonsistenz zwischen den Vorinformationen und dem Kennzahlensystem oder für jede aus den Daten berechenbare Kennzahl wird ein zum Kennzahlenmodell konsistenter Schätzer ermittelt.

Die Eigenschaften des SamPro Algorithmus können wie folgt zusammengefasst

werden:

- Verschiebung der Mittelwerte zur Erfüllung des Kennzahlensystems,
- Verschiebung ist abhängig von der Varianz der einzelnen Kennzahlen,
- Verringerung der Unsicherheitsbereiche und
- Invarianzeigenschaft bei Kennzahlen mit Varianz von Null.

Das Softwaretool MoSi

Zur Unterstützung von Entscheidungen mittels Kennzahlensystemen ist die Verwendung von Softwaretools ein wichtiges Mittel. Aus diesem Grund wurde für die Simulation von Kennzahlensystemen das Softwaretool MoSi entwickelt. MoSi steht hierbei für **M**odellierung und **S**imulation. Die Erstellung und Anpassung von Kennzahlensystemen wird ebenso unterstützt wie die Eingabe von Vorinformationen für Kennzahlen. Die Simulation läuft dabei automatisch ab. Visuelle Darstellungen wie z.B. Histogramme und Berechnungen wichtiger statistischer Schätzer ermöglichen einen leichteren Zugang zu den Simulationsergebnissen.

Da Nutzerfreundlichkeit ein wichtiges Kriterium für erfolgreiche Software ist, wurde die grafische Benutzerschnittstelle in Java implementiert. Dies ermöglicht die Unabhängigkeit der Software vom Betriebssystem. Die Simulationskomponente ist hingegen in der statistischen Programmiersprache R entwickelt. Der Hauptgrund für die Verwendung ist die hohe Zahlenverarbeitung während der Simulation. Zusätzlich können bereits in R implementierte Zufallszahlengeneratoren verwendet werden, was den Simulationsprozess weiter beschleunigt. Die Kommunikation zwischen den beiden Programmiersprachen erfolgt über die Schnittstelle JRI (Urbanek 2007).

Vergleich der Simulation mit bestehenden Verfahren

In diesem Abschnitt wird der SamPro Algorithmus mit bereits existierenden Verfahren verglichen.

Müller et al. (2003) haben zum Umgang von Unsicherheit in Kennzahlensystemen die Fuzzy Set Theorie verwendet. Als Ergebnis entstand das Excel-Plug-In FuzzyCalc. Gleichungssysteme werden wie auch die Fuzzy Set Variablen im Excel-Tabellenblatt abgelegt. Das Resultat der FuzzyCalc Implementierung können dann entweder eine Inkonsistenzmeldung oder aber verbesserte Fuzzy Set Variablen sein, deren Gipfelpunkte das Gleichungssystem erfüllen und einen meist verringerten Unsicherheitsbereich aufweisen.

Ein zweiter, von Schmid (1979) entwickelter Ansatz nimmt normalverteilte Kennzahlen als Grundlage. Mittels Kalman Filter werden verbesserte Schätzer auf Grundlage des Gleichungssystems ermittelt. Nichtlineare Gleichungen werden hierbei jedoch durch Approximation dargestellt.

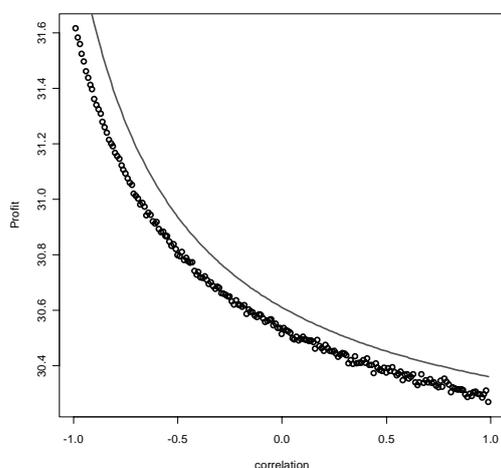


Abbildung B.3: Schätzungen einer Kennzahl in Abhängigkeit der Korrelation

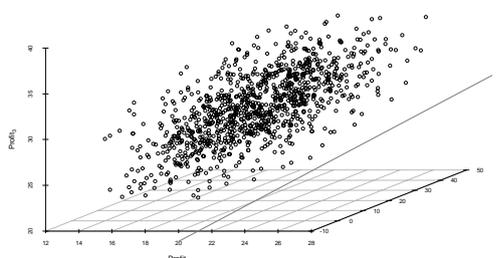


Abbildung B.4: Scatterplot für Gewinn mit korrelierten Vorinformation

Alle drei Ansätze ermöglichen die Imputation von Kennzahlen. Die Aufdeckung von Dateninkonsistenzen zum Kennzahlensystem wird durch den possibilistischen Ansatz und die Simulation gewährleistet.

Der SamPro Algorithmus hebt die Beschränkungen dieser beiden Methodiken auf. Anhand des DuPont Kennziffersystems erfolgt in Köppen und Lenz (2006) ein Vergleich der unterschiedlichen Methodiken. Es zeigt sich, dass die Simulation unter gleichen Verteilungsannahmen ähnliche Ergebnisse wie der GLS-Ansatz liefert. Die Resultate des possibilistischen Ansatzes stellen wie in Dubois und Prade (2006) dargestellt, eine obere Schranke dar. Damit sind die dort erzielten Unsicherheitsbereiche größer als im probabilistischen Ansatz bzw. der Simulation. Somit kann mit der Simulation eine höhere Genauigkeit erreicht werden.

Unter Aufgabe der Normalverteilungsannahme ergibt sich im Vergleich zum GLS-Ansatz das Bild, dass die Nichtlinearität durch die Approximation zu einer Unterschätzung der eigentlichen Mittelwerte bei der GLS-Schätzung führt. Korrelation kann dabei einen entscheidenden Einfluss auf die Modellkonsistenz haben.

In Abbildung B.3 ist der GLS-Schätzer in Abhängigkeit der Korrelation als durchgezogene Linie dargestellt. Die Simulationsergebnisse für das gleiche Modell sind als Kreise in der Abbildung ebenfalls eingezeichnet. Die beständige Überschätzung durch den GLS-Schätzer ist deutlich zu sehen.

Der Einfluss der Korrelation auf die Kennzahlensimulation ist in Abbildung B.4 ersichtlich. Die Gerade stellt den Unterraum für die drei Schätzer einer Kennzahl dar. Durch die Korrelation erreicht kein Element der Simulationsstichprobe den konsistenten Korridor. Dies führt zur Inkonsistenz des Kennzahlen.

Die Kennzahlensimulation kann aber ebenso zur Unternehmenssteuerung durch

Balanced Scorecard Systeme eingesetzt werden. Die Verwendung stochastischer Kennzahlen in der Balanced Scorecard ermöglicht eine bessere Planung und Steuerung, da dadurch Daten realitätsnäher abgebildet werden. Köppen et al. (2007) stellen hierzu ein in Excel entwickeltes Softwaretool vor. Mittels der Simulation können dann zum Modell konsistente Daten in den Berichten verwendet werden.

Zusammenfassung

Die Verwendung von Unsicherheit in Kennzahlensystemen stellt nicht nur ein geeignetes Mittel für die Prognose dar, sondern hilft auch bei kurzfristig zu treffenden Entscheidungen. Das entscheidungsunterstützende System muss dabei in der Lage sein, inkonsistente Daten aufzudecken sowie eine Verringerung der Unsicherheit innerhalb der Informationen zu erreichen. Die Imputation zusätzlich relevanter Kennzahlen zum Beispiel für das Berichtswesen ist ein weiterer Aspekt, den das System leisten soll.

Der entwickelte SamPro Algorithmus zur Simulation von Kennzahlensystemen imputiert fehlende Daten, deckt Modell-Inkonsistenzen von Daten auf und ermittelt verbesserte Kennzahlenschätzungen. Diese Kennzahlenschätzungen erfüllen das Kennzahlensystem und weisen eine verringerte Unsicherheit auf. Mit MoSi steht nun dem Anwender ein benutzerfreundliches System zur Modellierung und Simulation von Kennzahlensystemen zur Verfügung. Durch den Einsatz des Tools werden die durch SamPro möglichen Kennzahlen visuell aufbereitet und dem Benutzer leicht zugänglich gemacht.

Der vorgestellte Simulationsansatz zum Umgang mit Unsicherheiten in Kennzahlensystemen stellt ein geeignetes Vorgehen dar. Der Simulationsansatz kann hierbei existierende Methoden übertreffen, da einschränkende Annahmen nicht vorausgesetzt werden müssen. Es ist gelungen, eine bessere Abbildung der Realität und somit größere Genauigkeiten in den stochastischen Kennzahlensystemen zu erreichen. Ein Nachteil der Simulation besteht jedoch im Rechenaufwand. Für langfristige Planungen und in kritischen Anwendungsszenarien, in denen die Genauigkeit eine entscheidende Rolle spielt, sollte auf den Simulationsansatz zurückgegriffen werden.