View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Infoscience - École polytechnique fédérale de Lausanne

Low-rank tensor methods for large Markov chains and forward feature selection methods

THÈSE N^{O} 7718 (2018)

PRÉSENTÉE LE 8 JANVIER 2018

À L'ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
À LA FACULTÉ DES SCIENCES DE BASES
ALGORITHMES NUMÉRIQUES ET CALCUL HAUTE PERFORMANCE - CHAIRE CADMOS
ET
À L'INSTITUTO SUPERIOR TÉCNICO (IST) DA UNIVERSIDADE DE LISBOA

PROGRAMME DOCTORAL EN MATHÉMATIQUES
ET
DOUTORAMENTO EM ESTATISTICA E PROCESSOS ESTOCÁSTICOS

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES (PhD)

PAR

Francisco SANTOS PAREDES QUARTIN DE MACEDO

acceptée sur proposition du jury:

Prof. A. B. Ferreira Cruzeiro Zambrini, présidente du jury
Prof. D. Kressner, Prof. A. M. Pacheco Pires, directeurs de thèse
Dr P. Milheiro-Oliveira, rapporteuse
Dr N. Antunes, rapporteur
Dr C. Alves, rapporteur

TÉCNICO LISBOA

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Low-rank tensor methods for large Markov chains and forward feature selection methods

Francisco Santos Paredes Quartin de Macedo

Supervisor: Doctor António Manuel Pacheco Pires
Co-Supervisor: Doctor Daniel Kressner

Thesis approved in public session to obtain the PhD Degree in
Statistics and Stochastic Processes

Jury final classification: Pass with Distinction and Honour

Jury:

Chairperson: Doctor Ana Bela Ferreira Cruzeiro Zambrini, Instituto Superior Técnico da Universidade de Lisboa
Members of the committee:
Doctor António Manuel Pacheco Pires, Instituto Superior Técnico da Universidade de Lisboa
Doctor Carlos José Santos Alves, Instituto Superior Técnico da Universidade de Lisboa
Doctor Paula Manuela Lemos Pereira Milheiro de Oliveira, Faculdade de Engenharia da Universidade do Porto
Doctor Nelson Gomes Rodrigues Antunes, Faculdade de Ciências e Tecnologia da Universidade do Algarve
Doctor Maria do Rosário de Oliveira Silva, Instituto Superior Técnico da Universidade de Lisboa

Funding institutions - Fundação para a Ciência e a Tecnologia

2018

Acknowledgements

This thesis is based on my research conducted at École Polytechnique Fédérale de Lausanne and Instituto Superior Técnico from October 2012 until September 2016.

There are many people who are responsible for the fact that this thesis was finished successfully. I will try to be brief by only mentioning people who were more directly related to the result that is obtained in this document. I however assure that those who were less directly present will also not be forgotten either.

First of all, I want to thank my thesis advisor from IST: Prof. António Pacheco; and my advisor in EPFL: Prof. Daniel Kressner. I will start with Prof. Daniel Kressner since this long marathon started in 2012 in EPFL under his supervision. Daniel, it was an extreme pleasure to learn from you in a field that you completely dominate while I was completely clueless when I first arrived in EPFL. Also in terms of writing, my skills were extremely far from where they are today, and you were responsible for this clear evolution. After a very hard first year where I was struggling in many ways, it is impossible not to look back and be deeply thankful, mostly for your patience. It motivated me even more to never give up. Prof. António Pacheco, I want to thank you for the support, in particular during the last year in the conclusion of our joint work that truly made me proud for all that was achieved. I feel sorry that we did not have enough time in the end to concretize all the ideas that we had, but it was still very enjoyable to be part of our long meetings. While they were sometimes heavy psychologically when things did not seem to go our way, the final result was really satisfying. In the same context, I want to thank, from the bottom of my heart, Prof. M. Rosário Oliveira and Prof. Rui Valadas, who were part of these long meetings and also strongly responsible for the mentioned achievements.

Likewise, I want to thank all the researchers with whom I collaborated, whose work is also reflected in this thesis. I have enjoyed the collaboration with Wuppertal, in particular with Sonja Sokolovic. This work went on for quite a while and if our relation was not cool and relaxed, it would have become complicated. Moreover, I want to thank Prof. Peter Buchholz and Prof. Christian Mazza for the help with creating my broad collection of models that was very helpful for testing and comparing algorithms.

I am also thankful to all the people that created a nice group environment in EPFL. In particular, I want to mention the PhD students that were part of the group during this period: Ana, Cedric, Francesco, Meiyue, Michael, Petar, and Thalia. I also want to mention the people already in their postdocs: André, Christine, Jonas and Robert. I want to particularly thank Christine for her precious help, sometimes very late in the

evening, when I was struggling to become more knowledgeable in the field of tensors during the first year.

Financial support by Fundação para a Ciência e a Tecnologia during these four years is thankfully acknowledged.

It is time to address the personal component. In fact, while I tried to be as productive as possible during all these years, sometimes trying to act as a complete machine, I am proud to finally see that nothing is worth any type of effort and that motivation fades without personal interactions.

In this context, I have to start with my family. Family does not necessarily imply only blood relations. Family is what I call people with whom we always feel home, even if I have only recently started to understand what this really means. I have however clearly felt this with different people during these years, mostly the amazing group of friends I had back in Lausanne. However, I will refrain from listing names. They know who they are, anyway.

There are however concrete people that I cannot skip.

I must start with a person that not only was fundamental in personal terms as she additionally did all she could, and more, in order to help me reach this goal. Furthermore, I know that this person is not a person who expects something in return, but just did it because she has an unbelievably pure heart. Thank you, Marina.

The next person is my mother, who did just as much effort in helping me reach this goal. Her work is frequently underappreciated and I hope this message helps compensating for the times when that happened.

Still in the context of people who helped me have conditions for developing this thesis in the obsessive way I did, I cannot forget my father and my sister, as I have normally worked from home during my stay in Portugal and asked them to adjust to my needs, and I am aware this was very hard at times.

I was not only gone during the four years of this PhD, but even before, during my whole studies, so that my interactions with people were significantly reduced compared to what some people needed from me. I thus want to apologize to those who needed to be with me while I was not present as I was chasing this goal. As I said, many people are responsible for what I have achieved, and people in these circumstances suffered sometimes even more than me, and so they deserve at least as much credit for it. I would like to particularly mention my grandparents and my godfather. I will definitely make it up to you now.

Finally, I have to thank the person who entered my life to finally make me fully understand the beauty of it: *my Conchinha.* Concerning this thesis in a more direct way, her help in tuning the text was also unmeasurable.

Lisbon, the 28th of January 2017. Francisco

Abstract

In the first part of this thesis, we present and compare several approaches for the determination of the steady-state of large-scale Markov chains with an underlying low-rank tensor structure. Such structure is, in our context of interest, associated with the existence of interacting processes. The state space grows exponentially with the number of processes. This type of problems arises, for instance, in queueing theory, in chemical reaction networks, or in telecommunications.

As the number of degrees of freedom of the problem grows exponentially with the number of processes, the so-called *curse of dimensionality* severely impairs the use of standard methods for the numerical analysis of such Markov chains. We drastically reduce the number of degrees of freedom by assuming a low-rank tensor structure of the solution.

We develop different approaches, all considering a formulation of the problem where all involved structures are considered in their low-rank representations in *tensor train* format.

The first approaches that we will consider are associated with iterative solvers, in particular focusing on solving a minimization problem that is equivalent to the original problem of finding the desired steady state. We later also consider tensorized multigrid techniques as main solvers, using different operators for restriction and interpolation. For instance, aggregation/disaggregation operators, which have been extensively used in this field, are applied.

In the second part of this thesis, we focus on methods for feature selection. More concretely, since, among the various classes of methods, sequential feature selection methods based on mutual information have become very popular and are widely used in practice, we focus on this particular type of methods. This type of problems arises, for instance, in microarray analysis, in clinical prediction, or in text categorization.

Comparative evaluations of these methods have been limited by being based on specific datasets and classifiers. We develop a theoretical framework that allows evaluating the methods based on their theoretical properties. Our framework is based on the properties of the target objective function that the methods try to approximate, and on a novel categorization of features, according to their contribution to the explanation of the class; we derive upper and lower bounds for the target objective function and relate these bounds with the feature types. Then, we characterize the types of approximations made by the methods, and analyse how these approximations cope with the good properties of the target objective function.

We also develop a distributional setting designed to illustrate the various deficiencies of the methods, and provide several examples of wrong feature selections. In the context of this setting, we use the minimum Bayes risk as performance measure of the methods.

Key-words: Markov chains, curse of dimensionality, low-rank structure, tensor train format, entropy, mutual information, forward feature selection methods, performance measure, minimum Bayes risk.

Resumo

Na primeira parte da tese, apresentamos e comparamos algoritmos para a determinação da distribuição estacionária de cadeias de Markov de larga escala com uma estrutura subjacente de baixa característica. Tal estrutura está, no nosso contexto de interesse, associada à existência de processos que interagem entre si. O espaço de estados cresce exponencialmente com o número de processos. Aplicações podem ser encontradas, por exemplo, em teoria de filas de espera, em análise de redes de reacções químicas, ou em telecomunicações.

Como o número de graus de liberdade cresce exponencialmente com o número de processos, a tão chamada *maldição da dimensionalidade* prejudica severamente o uso de métodos standard para a análise numérica de tais cadeias de Markov. No nosso caso, reduzimos o número de graus de liberdade assumindo que a solução tem uma estrutura tensorial de baixa característica.

Desenvolvemos diferentes abordagens, que partilham o facto de o problema ser formulado considerando que as estruturas envolvidas nas suas representações em termos do formato *tensor train*.

As primeiras abordagens que vamos considerar estão associadas a métodos iterativos, em particular focando-nos na resolução de um problema de minimização que é equivalente ao problema de encontrar a distribuição estacionária desejada. Mais tarde, consideramos também métodos multigrelha tensorizados, variando os operadores considerados para restrição e interpolação, em particular considerando técnicas de agregação/desagregação, que são particularmente adequadas para o problema que pretendíamos resolver.

Na segunda parte da tese, concentramo-nos em métodos de selecção de variáveis. Mais concretamente, como, entre as várias classes de métodos, métodos de selecção de variáveis sequenciais baseados em informação mútua tornaram-se bastante populares e amplamente utilizados na prática, concentramo-nos neste tipo de métodos. Aplicações podem ser encontradas, por exemplo, em análise de microarranjos, em predição clínica, ou em categorização de textos.

Avaliações comparativas destes métodos têm sido limitadas pelo facto de se basearem em conjuntos de dados e classificadores específicos. Neste trabalho, desenvolvemos uma configuração teórica que permite avaliar os métodos baseando-nos nas suas propriedades teóricas. A nossa configuração é fundamentada pelas propriedades teóricas da função objectivo alvo que os métodos tentam aproximar, e numa nova categorização de variáveis, de acordo com a sua contribuição para a explicação da classe; derivamos limites superiores e inferiores para a função objectivo alvo e relacionamos estes limites com os tipos de variáveis. Em seguida, caracterizamos os tipos de aproximação considerados pelos métodos, e analisamos como estas aproximações lidam com as boas propriedades da função objectivo alvo.

Desenvolvemos também uma configuração distribucional projectada para ilustrar as várias deficiências dos métodos, e fornecemos vários exemplos de selecções de variáveis erradas. No contexto desta configuração, utilizamos o risco de Bayes mínimo como medida de performance dos métodos.

Palavras chave: cadeias de Markov, maldição da dimensionalidade, estrutura de baixa característica, formato *tensor train*, entropia, informação mútua, métodos de selecção de variáveis sequenciais, medida de performance, risco de Bayes mínimo.

Résumé

Dans la première partie de la thèse, nous présentons et comparons plusieurs approches pour déterminer l'état stationnaire de chaînes de Markov à grande échelle ayant une structure sous-jacente de tenseur à rang faible. Dans le notre cadre d'intérêt, une telle structure est associée avec l'existence d'interactions entre processus. L'espace d'état grandit de manière exponentielle en fonction du nombre de processus. Ce type de problèmes se pose par exemple dans la théorie des files d'attente, l'analyse de réseaux de réactions chimiques, ou dans les télécommunications.

Comme le nombre de degrés de liberté du problème grandit de manière exponentielle en fonction du nombre de processus, le *fléau de la dimension* pose des problèmes lors de l'utilisation de méthodes standard d'analyse numérique pour de telles chaînes de Markov. Nous réduisons drastiquement le nombre de degrés de liberté en supposant une structure de tenseur à faible rang de la solution.

Nous développons différentes approches qui ont en commun le fait que le problème est formulé en considérant toutes les structures impliquées en représentations à rang faible au format *tensor train*.

Les approches initiales que nous avons considérées étaient associées avec des solveurs itératifs, en particulier se focalisant sur la résolution d'un problème de minimisation équivalent au problème de trouver l'état stationnaire désiré. Ensuite, des techniques multigrille tensorisées ont également été considérées comme des solveurs principaux en utilisant différents opérateurs pour la restriction et interpolation, par exemple des opérateurs d'aggrégation/désaggrégation, qui ont été utilisés intensivement dans ce domaine.

Dans la deuxième partie de cette thèse, nous nous concentrons sur des méthodes de sélection de caractéristiques. Plus concrètement, puisque les méthodes de sélection de caractéristiques progressives basées sur l'information mutuelle parmi les différentes classes de méthodes sont devenues très populaires et utilisées en pratique, nous nous focalisons sur ce type de méthodes en particulier. Ce type de problèmes se pose par exemple dans l'analyse de microréseaux, la prédiction clinique, ou la catégorisation de texte.

Les évaluations comparatives de ces méthodes ont été limitées par l'utilisation de jeux de données et de classificateurs spécifiques. Nous développons un cadre théorique qui permet d'évaluer les méthodes basées sur leurs propriétés théoriques. Notre cadre est basé sur les propriétés de la fonction objectif cible que les méthodes essaient d'approximer et sur la catégorisation originale de caractéristiques selon leur contribution à l'explication de la classe; nous dérivons des bornes supérieures et inférieures pour la fonction objectif et nous mettons ces bornes en relation avec le type de caractéristiques. Ensuite, nous caractérisons les types d'approximations faites par les méthodes et analysons comment ces approximations se comportent avec les bonnes propriétés de la fonction objectif cible. De plus, nous développons un cadre distributionnel construit pour illustrer les différentes déficiences des méthodes et donnons plusieurs exemples de mauvaises sélections de caractéristiques.

Mots clés : Chaînes de Markov, fléau de la dimension, structure à rang faible, format train de tenseurs, entropie, information mutuelle, méthodes de sélection de caractéristiques progressives, mesure de performance, risque Bayésien minimal.

Acronyms

The following table describes the significance of various acronyms used throughout the thesis. The page on which each one is defined or first used is also given.

Acronym	Full definition	Page
ALS	alternating least squares	26
AMEn	alternating minimal energy	34
CP	CANDECOMP/PARAFAC	4
MI	mutual information	5
SVD	singular value decomposition	4
TMI	triple mutual information	86
TT	tensor train	4

Contents

Acknowledgements **i**

Abstract (English/Français/Português) **iii**

Acronyms **ix**

1 Introduction **1**

1.1 Low-rank tensor methods for structured large-scale Markov chains 1
1.2 Foundational topics in forward feature selection methods 5
1.3 Contributions of this thesis 6

2 Tensors and tensor train format **11**

2.1 Tensors 11
2.1.1 Basic operations on tensors 12
2.2 Tensor train format 13
2.2.1 Operations on tensors in TT format 16
2.2.2 Linear operators acting on tensors in TT format 20

3 Alternating optimization schemes **25**

3.1 Truncated power method 27
3.1.1 Power method 27
3.1.2 Power method in TT format 28
3.2 Alternating least squares 29
3.3 Alternating minimal energy 34
3.4 Numerical experiments 36
3.4.1 First test case: an overflow queuing network 37
3.4.2 Second test case: Kanban control model 40
3.5 Conclusion 42

4 A tensorized multigrid scheme combined with AMEn **43**

4.1 A tensorized multigrid scheme 44
4.1.1 Multigrid 44
4.1.2 Tensorized multigrid 45
4.2 Multigrid-AMEn 50

4.2.1 Limitation of AMEn 50
4.2.2 Limitations of the tensorized multigrid scheme 51
4.2.3 Combination of the two methods 51
4.3 Numerical experiments 53
4.3.1 Model problems 53
4.3.2 Numerical results 55
4.4 Conclusion 59

5 Aggregation/disaggregation techniques 63
5.1 Aggregation/disaggregation schemes 64
5.1.1 Tensorized algorithm pairing states in each subsystem 64
5.1.2 Aggregating all states from fixed subsystems 65
5.2 Proposed algorithms 66
5.2.1 Algorithm from Section 5.1.1 in TT format 66
5.2.2 Algorithm from Section 5.1.2 in TT format 69
5.3 Numerical experiments 71
5.3.1 First test case: converging metabolic pathways 72
5.3.2 Second test case: cyclic metabolic pathways 73
5.3.3 Third test case: wireless network with handoff 75
5.3.4 Choosing the algorithm to use 77
5.4 Conclusion 78

6 Entropy and mutual information 81
6.1 Entropy 81
6.2 Differential entropy 82
6.3 Mutual information 83
6.3.1 Discrete case 83
6.3.2 Continuous case 84
6.3.3 Combination of continuous with discrete random vectors 84
6.4 Triple mutual information and conditional mutual information 85
6.4.1 Conditional mutual information 85
6.4.2 Triple mutual information 86

7 Forward feature selection methods based on MI 87
7.1 The forward feature selection problem 87
7.1.1 Target objective functions 88
7.1.2 Feature types and their properties 89
7.1.3 Complementarity 92
7.2 Representative feature selection methods 93
7.2.1 Methods and their drawbacks 94
7.2.2 Properties of the methods 99
7.2.3 Other methods 102
7.3 Comparison of feature selection methods on a distributional setting 103

7.3.1 Minimum Bayes risk 103
7.3.2 Setting description 105
7.3.3 Required quantities 106
7.3.4 Applying the different feature selection methods 108
7.4 Conclusion 116

8 Conclusion **121**

A Some analytical relevant computations **125**
A.1 Computation of the terms in the tables of Section 7.3.3 125
A.1.1 Values in Table 7.3 125
A.1.2 Values in Table 7.4 126
A.1.3 Values in Table 7.5 131
A.1.4 Values in Table 7.6 131
A.2 Calculations of MBR values in Section 7.3.4 140

Bibliography **152**

Curriculum Vitae **153**

1 Introduction

In this chapter, we introduce the two main independently treated problems addressed in this thesis: finding the stationary distribution of structured large-scale Markov chains using low-rank tensor methods and evaluating a particular subclass of methods for feature selection. The first problem is discussed in Section 1.1 while the second is discussed in Section 1.2. We finish the chapter by briefly discussing the contributions in this thesis, in Section 1.3.

1.1 Low-rank tensor methods for structured large-scale Markov chains

In the first part of the thesis, we focus on the computation of the stationary distribution of continuous–time Markov chains.

The problem to be solved is:

$$Q^T x = 0 \text{ with } e^T x = 1, \tag{1.1}$$

where Q is the generator matrix of the Markov chain of interest and e denotes the vector of all ones. For a certain ordering of the states of the Markov chain, the entry in the i-th row and j-th column of Q contains the rate of transition from the state i to the state j. Matrix Q^T is non-symmetric, singular and verifies $e^T Q^T = 0$.

This problem can be formulated as a constrained least squares problem:

$$\min_x \{\|Q^T x\|_2^2 : \, e^T x = 1\}. \tag{1.2}$$

For an irreducible – thus, also ergodic since we work with the continuous case – Markov chain, the problem has an unique solution [Ros00, Ch. 4]. For reducible Markov chains, which also appear quite frequently, this does not hold. We will however also deal with this latter case.

We focus on Markov processes that feature high-dimensional state spaces arising from modelling subsystems that interact between them. The considered Markov chain consists of d interacting subsystems (processes). Q consequently has a tensor – Kronecker – structure of the form

$$Q = \sum_{t=1}^{V} \bigotimes_{k=1}^{d} E_k^{(t)}, \tag{1.3}$$

where each term in the summation represents a possible transition between states. The number of possible states of subsystem k corresponds to the number of rows (or columns) of the different $E_k^{(t)}$. Note that Q^T has the same structure as we just need to replace each $E_k^{(t)}$ in (1.3) by $(E_k^{(t)})^T$.

Authors typically divide the possible transitions into three classes; see, for instance, [Ste94, Ch. 9]: local transitions, associated with exclusively one subsystem; functional transitions, whose rates involve the current state of other subsystems; and synchronized transitions, which are transitions that occur simultaneously in different subsystems.

Therefore, the matrix can be decomposed in:

$$Q = Q_L + Q_I, \tag{1.4}$$

where Q_L is the part representing the local transitions while Q_I represents the interactions between subsystems – associated with functional and synchronized transitions. The local part takes the form

$$Q_L = \sum_{\mu=1}^{d} I_d \otimes \dots I_{\mu+1} \otimes L_\mu \otimes I_{\mu-1} \otimes \cdots \otimes I_1, \tag{1.5}$$

where L_μ is the matrix representing the local transitions in the μ-th subsystem. Note that such local part can be simply represented using a Kronecker sum given that all possible transitions that exclusively concern a particular subsystem can be considered together. This is due to the fact that the matrices associated with the remaining subsystems are simply identity matrices since such subsystems are completely independent of the related transition.

Applications of such models can be found, e.g., in queueing theory [Kau83, Cha87]; stochastic automata networks [LS04a, PS97]; analysis of chemical reaction networks [ACK10, LH07]; or telecommunications [PSS96, AFRT06].

Assuming a low-rank structure in the transition rate matrix is extremely useful when

Markov chains with an underlying Kronecker structure, as in (1.3), are considered. Moreover, this becomes unavoidable if the goal is to study Markov chains associated with an increasing number of processes, given that problems of this kind are known for their so called state space explosion [BD07a] – exponential growth of the state space dimension on the number of subsystems. Such state space explosion severely impairs the numerical analysis of such Markov processes. In fact, as the number of processes grows, the associated number of states easily becomes such that classical solvers are completely inefficient. For example, if the number of states per subsystem is 5 and the number of subsystems/processes is 10, we already have $5^{10} = 500.000$ states. The mentioned classical solvers include all standard iterative solvers [BBC+94] for addressing the linear system (1.1) or the equivalent eigenvalue problem; in fact, they have a complexity that scales at least linearly with the state space dimension.

Low-rank tensor techniques can be used instead if we see the vector x from (1.1) as a tensor. Denoting the number of states in the μ-th process, $\mu = 1, \dots, d$, by n_μ, x has dimension $n_1 n_2 \cdots n_d$. Quite naturally, the entries of this vector can be rearranged into an $n_1 \times \cdots \times n_d$ array, defining a d-th order tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. The entries of $\mathbf{X}$ are denoted by

$$\mathbf{X}(i_1, i_2, \dots, i_d), \quad 1 \le i_\mu \le n_\mu, \quad \mu = 1, \dots, d,$$

where n_μ is defined as the size of the μ-th mode of the tensor.

Recalling the example involving 10 subsystems with 5 possible states each, if x can be simply represented as

$$x = x_d \otimes x_{d-1} \otimes \cdots \otimes x_1, \quad x_i \in \mathbb{R}^{n_i}, \quad i = 1, ..., d, \tag{1.6}$$

there are only $5 \times 10 = 50$ degrees of freedom, instead of the original $5^{10} = 500.000$.

Product form solutions represent a particularly popular reduction technique that has been successfully used in a wide range of applications for solving (1.3). The basic idea of this reduction is to yield a system for which the stationary distribution factorizes into a product of distributions for the individual processes – factorization of the form (1.6). This reduced system naturally allows a much less expensive numerical treatment. General techniques for arriving at product form solutions are described, e.g., in [Kul11, Ch. 6]. Extensive work has been done on finding conditions under which such product form approach applies; see [FPS08, Fou08] for some recent results. However, its practical range of applicability is still limited to very specific subclasses. A rather different approach is based on the observation that the transition rate matrix of a communicating Markov process can often be represented by a short sum of Kronecker products [PFL89]. This property can then be exploited when performing matrix-vector multiplications or constructing preconditioners [LS04b, LS04c] to reduce the cost of iterative solvers significantly. However, the complexity of such linear solvers still scales linearly with the state space dimension.

Considering the simple product structure in (1.6), more sophisticated types of low-rank tensor formats have been developed, which generalize the matrix rank to the tensor case. The matrix rank is associated with the $d = 2$ case. $\mathbf{X}$ becomes a matrix and there is an unique notion of rank that can be computed by the singular value decomposition (SVD) [GL96]. The extension of this concept to $d > 2$ is then by no means unique, originating different low-rank decompositions; see [KB09] for an overview.

The *CANDECOMP/PARAFAC (CP) decomposition* takes the form

$$\operatorname{vec}(\mathbf{X}) = \sum_{r=1}^{R} u_r^{(1)} \otimes u_r^{(2)} \otimes \cdots \otimes u_r^{(d)} = \sum_{r=1}^{R} \bigotimes_{\mu=1}^{d} u_r^{(\mu)}, \tag{1.7}$$

where each $u_r^{(\mu)}$ is a vector of length n_μ. The tensor rank of $\mathbf{X}$ is the smallest R admitting such decomposition. The relation to (1.6) is clear: the corresponding x can be represented as a tensor with tensor rank 1. Such representation clearly fits the considered structure of the generator matrix; recall (1.3). This allows that efficient algorithms are developed when combining the two representations, for instance, for defining the important matrix-vector multiplication. In fact, such an operation can be then performed in a particularly efficient way.

The introduced decomposition has been, in fact, used in [Buc10] for the specific problem we aim to solve. More concretely, a version of (1.2), defined in terms of this format, is solved. The algorithm uses an alternating optimization scheme. Despite certain theoretical drawbacks [KB09], CP decomposition has been observed to perform fairly well in practice. This decomposition may, however, not always be the best choice because it does not exploit the topology of interactions. Furthermore, approximations (truncations) are a crucial part of our proposed algorithms and this format does not allow performing them efficiently. This is not the case for the related *tensor train* (TT) format, since it has an associated truncation procedure – TT-SVD algorithm [Ose11b] – that is based on singular value decomposition (SVD), which is known to have quasi-optimal upper bounds for the error that is done in an approximation. We focus on this format instead for dealing with the low-rank structure in high-dimensional tensors. This ansatz was developed in the numerical linear algebra community [OT09, Ose11b] but had already been used earlier in the physics community to represent quantum states of 1D spin chains [AKLT87, Whi92, Sch11].

While TT format has been used, for instance, for simulating stochastic systems [JCJ10, KKNS13], we explicitly target the computation of the associated stationary distributions, developing and comparing different algorithmic approaches.

1.2 Foundational topics in forward feature selection methods

Concerning the second part of the thesis, we focus on forward feature selection methods.

In an era of data abundance, of a complex nature, it is of utmost importance to extract, from the data, useful and valuable knowledge for real problem solving. Companies seek, in the pool of available information, commercial value that can leverage them among competitors or give support for making strategic decisions. One important step in this process is the selection of relevant and non-redundant information in order to clearly define the problem at hand and aim for its solution [BCSMAB15]; the importance of this step, in the context of this new era of data abundance, has been also noted, for instance, in [YWDP14, LDC$^+$15].

Feature selection problems arise in a variety of applications, reflecting their importance. Instances can be found in: microarray analysis [XJK$^+$01, SIL07, BCSMAB13, LZO04, LLW02], clinical prediction [BKRA$^+$15, LZO04, LLW02], text categorization [YP97, RY02, VMAF13, KQB16], image classification and face recognition [BCSMAB15], multi-label learning [SS00, CS02], and classification of internet traffic [POV$^+$].

Feature selection techniques can be categorized as classifier-dependent (*wrapper* and *embedded* methods) and classifier-independent (*filter* methods). Wrapper methods [KJ97] search the space of feature subsets, using the classifier accuracy as the measure of utility for a candidate subset. There are clear disadvantages in using such approach. The computational cost is huge, while the selected features are specific for the considered classifier. Embedded methods [GGNZ08, Ch. 5] exploit the structure of specific classes of classifiers to guide the feature selection process. In contrast, filter methods [GGNZ08, Ch. 3] separate the classification and feature selection procedures, and define a heuristic ranking criterion that acts as a measure of the classification accuracy.

Filter methods differ among them in the way they quantify the benefits of including a particular feature in the set used in the classification process. Numerous heuristics have been suggested. Among these, methods for feature selection that rely on the concept of *mutual information* are the most popular. Mutual information (MI) captures linear and non-linear association between features, and is strongly related with the concept of *entropy*. Since considering the complete set of candidate features is too complex, filter methods usually operate sequentially and in the forward direction, adding one candidate feature at a time to the set of selected features. In each step, the selected feature is the one that, among the set of candidate features, maximizes an objective function expressing the contribution of the candidate to the explanation of the class. A unifying approach for characterizing the different forward feature selection methods based on MI has been proposed in [BPZL12]. An overview of the different feature selection methods is also provided in [VE14], adding a list of open problems in the field.

Among the forward feature selection methods based on MI, the first proposed group [Bat94, PLD05, HLLC08, Pas13] is constituted by methods based on assumptions that were originally introduced in [Bat94]. These methods attempt to select the candidate feature that leads to: maximum *relevance* between the candidate feature and the class; and minimum *redundancy* of the candidate feature with respect to the already selected features. Such redundancy, which we call *inter-feature redundancy*, is measured by the level of association between the candidate feature and the previously selected features. Considering inter-feature redundancy in the objective function is important, for instance, to avoid later problems of collinearity. In fact, selecting features that contain repeated information, considering the information in the already selected ones, in terms of class explanation should be avoided.

A more recently proposed group of methods based on MI considers an additional term, resulting from the accommodation of possible dependencies between the features given the class [BPZL12]. This additional term is disregarded by the previous group of filter methods. Examples of methods from this second group are the ones proposed in [LT06, YM99, Fle04]. The additional term expresses the contribution of a candidate feature to the explanation of the class, when considering that the information contained in the already selected features is known, which corresponds to a *class-relevant redundancy*. The effects captured by this type of redundancy are also called *complementarity* effects.

We build a theoretical framework for the evaluation of the most relevant forward feature selection methods based on mutual information.

1.3 Contributions of this thesis

Chapter 2. This chapter starts the first part of the thesis. We first introduce tensors in their full representation and some basic operations. We then discuss the tensor train format and how the most relevant operations are done on it. These operations are the ingredients needed in our proposed approaches for the solution of (1.1). The main concern when performing such operations is that the resulting structures still have a low-rank factorization.

Chapter 3. We propose and compare different algorithms for the solution of (1.1). All these algorithms have their structures in TT format. We start with a simple iterative solver that formulates (1.1) as an eigenvalue problem, proceeding then to an alternating optimization scheme that uses a formulation based on the equivalent problem (1.2). The importance of allowing the ranks to be flexible during this procedure motivated the introduction of a third algorithm. Such algorithm is based on the same core principles as the second, but adding a step that includes a natural rank adaptivity scheme, using TT-SVD algorithm [Ose11b]. In the end, the first of the three algorithms is mostly added as a reference for comparison, being an algorithm based on a simple iterative solver, while the main idea of the chapter is to consider alternating optimization schemes. This is the core idea of the two remaining proposed algorithms.

We illustrate the use of the developed algorithms on two networks, which are queuing networks that were used in the past for this same purpose. The associated numerical experiments show the efficiency of our approaches when compared with a relevant existing one. Furthermore, we extensively explore the differences between the proposed approaches, concluding that those considering alternating optimization schemes are clearly better. It is then also clear that the algorithm where rank adaptivity is allowed is clearly the best of these two. For this particular algorithm, we show its scalability with respect to the number of processes.

The content of this chapter is based on the paper [KM14].

Chapter 4. We propose a new algorithm for the solution of (1.1), combining the advantages of considering a tensorized multigrid method. This allows reducing the mode sizes and, as a consequence, the condition number of the matrix of the problem; solved with an alternating scheme, similar to the one proposed in Chapter 3. Moreover, all structures involved in the algorithm are again in TT format. Such a tensorized multigrid should be a good main solver, as it takes advantage of the knowledge that the generator matrix has the representation (1.3) on the definition of the corresponding restriction and interpolation operators. Its coarsest grid problem however still suffers from the curse of dimensionality, as the number of modes of the corresponding tensor is not reduced from one grid to another. In this context, applying an algorithm, such as the mentioned alternating scheme that is designed to deal with large d, is necessary.

We illustrate the use of the developed algorithm on a variety of models from different fields taken from a broad benchmark collection that is also part of the contributions of the work that was developed in the context of this thesis: [Mac15]. The main novelty about this approach is that, as the numerical experiments show, it allows dealing efficiently not only with a large number of possible states per process (mode sizes) but also with a large number of processes (d). Furthermore, by considering models from different fields, robustness is also verified.

The content of this chapter is based on the paper [BKK$^+$16].

Chapter 5. We propose two variants of an additional algorithm for the solution of (1.1), again based on multigrid methods with restriction and interpolation operators that take advantage of the structure of the problem. The main difference is that such operators are now of a different type – based on aggregation/disaggregation. Restriction and interpolation are chosen in a very particular way, called aggregation and disaggregation, respectively. Such an approach has been used in the past to solve (1.1). Moreover, the structure (1.3) has been considered when choosing the aggregation and disaggregation operators. The difference of our proposed algorithms is that, again, all involved structures are in TT format. The two variants result from considering aggregation and disaggregation done in two different ways.

The use of the developed algorithms is again illustrated on a variety of models from the benchmark collection [Mac15].

The main object of comparison in the numerical experiments is the algorithm from Chapter 4. Such experiments show that the first variant performs similarly to the algorithm of reference. This is expected as they are both based on tensorized multigrid schemes that reduce the mode sizes from one grid to the next in a similar way. The performance is however slightly better for the newly proposed one. Furthermore, the second variant is the only algorithm that is able to deal with a very particular type of models. In fact, such type of models was rarely addressed in the existing literature associated with this context. It would, in particular, cause difficulties if we tried to apply the algorithms proposed in the two chapters that come before this one in the thesis to it.

In the end, robustness is obtained in the sense that we can partition the models in two parts and for each part, we always have a method that is extremely efficient: we can use the first variant for the generality of the models, while the second variant for the remaining mentioned models that cannot be properly addressed with any other of the proposed algorithms. Robustness is also emphasized by the fact that even a reducible model is considered. In fact, while the solution is not unique for this type of model, if we only focus on each connected component of states, the solution becomes again unique, so that a concrete solution is actually possible to find.

The content of this chapter is based on the paper [Mac16].

Chapter 6. This chapter marks the beginning of the second part of the thesis. We discuss the core concepts associated with forward feature selection methods based on MI. More concretely, the fundamental concept of entropy is explored in detail, for discrete and continuous random variables, along with its main properties. This is important since the relevant concepts associated with MI can be also written in terms of intuitive expressions containing entropy terms. Such concepts related to MI are then presented, along with their associated properties.

Chapter 7. We describe the general context of forward feature methods based on MI, introducing standard objective functions to be maximized in each step. Then, some relevant concepts concerning feature selection are defined, and we prove some properties of the mentioned standard objective functions that concern such concepts.

We focus on explaining the general context concerning forward feature selection methods based on mutual information. We introduce a target objective function to be maximized in each step; we then define important concepts and prove some properties of such target objective functions.

The mentioned target objective function cannot be used in practice as there is a term that is required when evaluating them that is quite complex. It requires, in particular, the knowledge of a high-dimensional term that is hard to estimate accurately. The common solution is to use approximations, leading to different feature selection methods. For the analysis in this thesis, we selected a set of methods representative of the main types of approximations to the target objective function. We will describe the representative methods, and discuss drawbacks resulting from their underlying approximations; we then discuss how these methods cope with the some desirable properties that hold for the target objective function.

We compare the feature selection methods using a distributional setting, based on a specific definition of class, features, and a performance metric. The setting provides an ordering for each of the methods, which is independent of specific datasets and estimation methods, and is compared with the ideal feature ordering – the one obtained for the target objective function. The aim of the setting is to illustrate how the drawbacks of the methods lead to incorrect feature ordering and to the loss of the good properties of the target objective function.

We are clearly able to identify, based on the developed work, the methods that should be avoided, and the methods that have the best performance.

The content of this chapter is based on the paper [MOPV17].

2 Tensors and tensor train format

In this chapter, we present basic properties of tensors and discuss some basic operations. We then also present the corresponding elementary properties of the representation in tensor train format, while discussing how some operations are done in this format. The main concern when performing such operations is that the resulting structures still have a low-rank factorization. We focus on the operations needed in the algorithms for solving (1.1), introduced in the chapters of the first part of the thesis that follow.

This chapter is the fundamental introduction to what follows in the first part of the thesis.

2.1 Tensors

A tensor is simply a multidimensional array. If $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots n_d}$, then the tensor is said to be of dimension d, while the different n_i, $i = 1, ..., d$, are its mode sizes – i is the mode, while n_i is the size of this mode. In particular, if $d = 2$, we have a matrix. We exemplify the notation for individual entries of the tensors for this particular case: $\mathbf{X}(i_1, i_2)$ denotes the element that is in the row i_1 and the column i_2 of the matrix.

From here on, Matlab's colon notation will be used to designate a range of indices.

It is common to dispose the entries of the tensor in matrices or even vectors. The resulting representations are called *matricization* and *vectorization*, respectively. In particular, we will see that it is then possible to represent the most relevant operations on tensors through operations on such matrices and vectors, making them more intuitive.

Vectorization is first exemplified for $d = 2$. The vectorization of $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ is:

$$\mathrm{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{X}(:,1) \\ \mathbf{X}(:,2) \\ \vdots \\ \mathbf{X}(:,n_2) \end{bmatrix}.$$

We now also discuss the generalization of the case $d = 2$ given that it may not be straight-forward. If $x = \mathrm{vec}(\mathbf{X})$, then $x(\xi(i_1, ..., i_d)) = \mathbf{X}(i_1, ..., i_d)$, where ξ is a function that gives the index map

$$\xi(i_1, ..., i_d) = 1 + \sum_{\mu=1}^{d} (i_\mu - 1) \prod_{\nu=1}^{\mu-1} n_\nu. \tag{2.1}$$

This function naturally goes from $\{1, ..., n_1\} \times \cdots \times \{1, ..., n_d\}$ to $\{1, ..., \prod_{\mu=1}^{d} n_\mu\}$. In practice, we are considering that the multi-indices of the form $(i_1, i_2, ..., i_d)$ are traversed in reverse lexicographical order.

As for matricizations, the idea is, as already noted, to organize the entries of $\mathbf{X}$ in a matrix. A particularly used type of matricization consists of considering the matrix resulting from associating the μ-th mode with the rows and the remaining modes with the columns. This is called the μ-th matricization. The resulting matrix $\mathbf{X}_{(\mu)} \in \mathbb{R}^{n_\mu \times n_1 \cdots n_{\mu-1} n_{\mu+1} \cdots n_d}$ can be also formally defined through an index map. This index map is now defined from $\{1, ..., n_1\} \times \cdots \times \{1, ..., n_d\}$ to $\{1, ..., n_\mu\} \times \{1, ..., \prod_{\nu=1, \nu \neq \mu}^{d} n_\nu\}$, and it is given by

$$\xi_{(\mu)}(i_1, ..., i_d) = (i_{(\mu)}, i_{(\neq\mu)}), \text{ where } i_{(\neq\mu)} = 1 + \sum_{\nu=1, \nu\neq\mu}^{d} (i_\nu - 1) \prod_{\eta=1, \eta\neq\mu}^{\nu-1} n_\eta.$$

The column index $i_{(\neq\mu)}$ is, in coherence with the vectorization case, associated with a reverse lexicographical order. It should be emphasized that the matricizations exemplified above are just particular cases, and there are many other possible ways to obtain matrices from reorganizing the entries of a tensor. In particular, while these are the most interesting matricizations when we think of tensors in their full representation, the ones used in the context of TT format are of a different type, as we see later in Section 2.2.

2.1.1 Basic operations on tensors

We now focus on operations that are important to understand in the context of what follows in the first part of the thesis. This will, in particular, facilitate the perception of the operations in TT format, discussed in Section 2.2.1. In this context, operations that are straight-forward generalizations of the matrix case are omitted.

Multiplication with a matrix. It is possible to define the multiplication of a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots n_d}$ by a matrix $A_\mu \in \mathbb{R}^{m \times n_\mu}$. A natural restriction is that it must be associated with the μ-th dimension for coherence in the mode sizes. The μ-th mode multiplication is defined, along the μ-th mode, as follows

$$\mathbf{Y} = \mathbf{X} \times_\mu A_\mu \Leftrightarrow \mathbf{Y}_{(\mu)} := A_\mu \mathbf{X}_{(\mu)}.$$

Note that the size of the μ-th mode of $\mathbf{Y}$ differs from that of $\mathbf{X}$, being m instead of n_μ.

A property that connects the μ-th mode product with the Kronecker product for matrices follows. Given the matrices $A_\mu \in \mathbb{R}^{m_\mu \times n_\mu}$, $\mu = 1, ..., d$,

$$\mathbf{Y} = \mathbf{X} \times_1 A_1 \cdots \times_d A_d \Leftrightarrow \mathbf{Y}_{(\mu)} = A_\mu \mathbf{X}_{(\mu)} (A_d \otimes \cdots A_{\mu+1} \otimes A_{\mu-1} \otimes \cdots A_1)^T.$$

Note that $\mathbf{Y} \in \mathbb{R}^{m_1 \times m_2 \times \cdots m_d}$.

Inner product. In the tensor case, the inner product is defined by the sum of the element-wise product. It should be clear that this can be written, for tensors $\mathbf{X}$ and $\mathbf{Y}$, in terms of their μ-th matricizations, and also of their vectorizations. We have

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \langle \mathrm{vec}(\mathbf{X}), \mathrm{vec}(\mathbf{Y}) \rangle = \mathrm{trace}(\mathbf{X}_{(\mu)}^T, \mathbf{Y}_{(\mu)}),$$

where μ can be any mode – $\mu = 1, ..., d$.

In the matrix case, $d = 2$, we get the trace inner product.

As for the induced norm $||\mathbf{X}|| = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}$, it is, for $d = 2$, the Frobenius norm.

2.2 Tensor train format

We now explore the tensor train format.

The number of degrees of freedom associated with $\mathbf{X} \in \mathbb{R}^{n \times \cdots \times n}$ is n^d, if we consider that the access to each individual entry is allowed. Reducing the number of degrees of freedom can be done by generalizing the notion of low-rank format from matrices to tensors.

A matrix of rank r, $X \in \mathbb{R}^{n_1 \times n_2}$, can be written as $X = USV^T$, where $U \in \mathbb{R}^{n_1 \times r}$ and $V \in \mathbb{R}^{n_2 \times r}$ are matrices with orthonormal columns, while $S \in \mathbb{R}^{r \times r}$ is a diagonal matrix with non-negative entries. Such a decomposition can be obtained using SVD.

When we move to tensors, the n^d storage can be reduced using low-rank tensor formats. We focus on a particular one in this thesis – the *tensor train* (TT) format [OT09, Ose11b]. The main feature of this format is that the exponential dependency on d disappears. This format had been first used in the physics community under the name *matrix product states* (MPS) [AKLT87, Whi92, Sch11].

We next represent a tensor in this format, followed by the description of how some relevant operations are performed in such representation. We particularly focus on the operations needed in the context of the algorithms that are described in the following chapters of the first part of the thesis.

The matricization of interest, concerning the μ-th mode, when considering TT format is not the μ-th mode matricization. It is, instead, the called μ-th mode unfolding. For a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, the μ-th mode unfolding consists of arranging the entries in a matrix $\mathbf{X}^{\langle \mu \rangle} \in \mathbb{R}^{(n_1 \cdots n_\mu) \times (n_{\mu+1} \cdots n_d)}$ associated with the index map $\xi(i_1, ..., i_d) = (i_{\text{row}}, i_{\text{col}})$, where

$$i_{\text{row}} = 1 + \sum_{\nu=1}^{\mu} (i_\nu - 1) \prod_{\eta=1}^{\nu-1} n_\eta \quad \text{and} \quad i_{\text{col}} = 1 + \sum_{\nu=\mu+1}^{d} (i_\nu - 1) \prod_{\eta=\mu+1}^{\nu-1} n_\eta .$$

Such an index map is defined from $\{1, ..., n_1\} \times \cdots \times \{1, ..., n_d\}$ to $\{1, ..., \prod_{\nu=1}^{\mu} n_\nu\} \times \{1, ..., \prod_{\nu=\mu+1}^{d} n_\nu\}$.

The notion of rank in this format is related with the typical definition of rank for matrices through different unfoldings. The so-called *TT rank* is given by

$$\text{rank}_{\text{TT}}(\mathbf{X}) = (r_0, r_1, ..., r_d) := (1, \text{rank}(\mathbf{X}^{\langle 1 \rangle}), ..., \text{rank}(\mathbf{X}^{\langle d-1 \rangle}), 1).$$

An individual entry of the original tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ can be written as a product of d matrices

$$\mathbf{X}(i_1, ..., i_d) = U_1(i_1) U_2(i_2) \cdots U_d(i_d), \tag{2.2}$$

where the matrices $U_\mu(i_\mu) \in \mathbb{R}^{r_{\mu-1} \times r_\mu}$, $i_\mu = 1, ..., n_\mu$, are the so-called *TT cores*, which will frequently be called cores for simplicity.

Remark 1. *Tensor train decomposition is expected to be more suitable for networks with an underlying topology of the subsystems associated with a train, in the sense that the existing interactions should concern consecutive subsystems after these are suitably ordered; see* (2.2)*.*

Each individual core can be also defined through three-dimensional tensors $\mathbf{U}_\mu \in$

$\mathbb{R}^{r_{\mu-1}\times n_\mu \times r_\mu}$ that verify $\mathbf{U}_\mu(:, i_\mu, :) = U_\mu(i_\mu)$ instead, again for $i_\mu = 1, ..., n_\mu$. This provides an alternative way to represent the TT cores. Therefore, $\mathbf{X}(i_1, ..., i_d)$ can be also represented as

$$\mathbf{X}(i_1, ..., i_d) = \sum_{k_1=1}^{r_1} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathbf{U}_\mu(1, i_1, k_1)\mathbf{U}_\mu(k_1, i_2, k_2) \cdots \\ \mathbf{U}_\mu(k_{d-2}, i_{d-1}, k_{d-1})\mathbf{U}_\mu(k_{d-1}, i_d, 1). \quad (2.3)$$

Some additional concepts are worth introducing in the context of exploring the structure of the format. In particular, these concepts will be useful when defining the operations of interest associated with this format.

We start with left and right *unfoldings*. They are associated with particular reshapings of $\mathbf{U}_\mu$ into matrices, related to the corresponding μ-th mode unfolding: $\mathbf{U}_\mu^L = \mathbf{U}_\mu^{\langle 2\rangle}$ and $\mathbf{U}_\mu^R = \mathbf{U}_\mu^{\langle 1\rangle}$ represent the left and right unfoldings, respectively.

The tensor can be additionally separated in left and right parts. The resulting structures are called *interface matrices*:

$$\mathbf{X}_{\le\mu}(i_1, ..., i_\mu) = U_1(i_1)U_2(i_2)\cdots U_\mu(i_\mu);$$
$$\mathbf{X}_{\ge\mu}(i_\mu, ..., i_d) = [U_\mu(i_\mu)U_{\mu+1}(i_{\mu+1})\cdots U_d(i_d)]^T.$$

Note that $\mathbf{X}_{\le\mu}$ is in $\mathbb{R}^{n_1 n_2 \cdots n_\mu \times r_\mu}$ and $\mathbf{X}_{\ge\mu}$ is in $\mathbb{R}^{n_\mu n_{\mu+1}\cdots n_d \times r_{\mu-1}}$.

The unfoldings and the interface matrices are related as follows:

$$\mathbf{X}_{\le\mu} = (I_{n_\mu} \otimes \mathbf{X}_{\le\mu-1})\mathbf{U}_\mu^L;$$
$$\mathbf{X}_{\ge\mu}^T = \mathbf{U}_\mu^R(\mathbf{X}_{\ge\mu+1}^T \otimes I_{n_\mu}). \quad (2.4)$$

Using (2.4), the μ-th core can be isolated. In fact, it holds that

$$\mathrm{vec}(\mathbf{X}) = (\mathbf{X}_{\ge\mu+1} \otimes I_{n_\mu} \otimes \mathbf{X}_{\le\mu-1})\mathrm{vec}(\mathbf{U}_\mu).$$

Note that $\mathbf{X}_{\ge\mu+1} \otimes I_{n_\mu} \otimes \mathbf{X}_{\le\mu-1} \in \mathbb{R}^{n_1 n_2 \cdots n_d \times r_{\mu-1} n_\mu r_\mu}$.

Adopting the shorthand notation $\mathbf{X}_{\ne\mu} = \mathbf{X}_{\ge\mu+1} \otimes I_{n_\mu} \otimes \mathbf{X}_{\le\mu-1}$, we obtain an expression that is crucial, as we will see in Chapter 3, for the definition of alternating optimization schemes in this format:

$$\mathrm{vec}(\mathbf{X}) = \mathbf{X}_{\ne\mu}\mathrm{vec}(\mathbf{U}_\mu). \quad (2.5)$$

2.2.1 Operations on tensors in TT format

We now focus on operations that are intrinsic to the algorithms that will be defined later, in the remaining chapters of the first part of the thesis. We go through some operations that were skipped in Section 2.1.1 as they are straight-forward when dealing with tensors in their full representation but not when this concrete format is considered. Additionally, even for operations that are straight-forward, we want to at least present them so that we go through all the ingredients needed in the algorithms.

Scaling. Given a TT tensor, and recalling that it can be represented in terms of core tensors as in (2.2), it is clear that multiplying a constant by the tensor is the same as multiplying it by one of the cores that compose it.

Orthogonalization. The representation of a given tensor in the TT format is not unique. If, given the decomposition $\mathbf{U}_\mu^L = QR$, we define that $\mathbf{U}_\mu^L$ becomes Q while $\mathbf{U}_{\mu+1}^R$ becomes $R\mathbf{U}_{\mu+1}^R$, the representation is still associated with the same tensor. If Q has orthonormal rows, this is called the *left-orthogonalization* of the μ-th core. Similarly, the *right-orthogonalization* of a core is obtained by using the decomposition $\mathbf{U}_\mu^R = (QR)^T = R^T Q^T$, setting then $\mathbf{U}_\mu^R$ to Q^T while $\mathbf{U}_{\mu-1}^L$ is changed to $\mathbf{U}_{\mu-1}^L R^T$.

In this context, if left-orthogonalization is performed on the first core, then for the second and so on, we obtain a left-orthogonal tensor, meaning that $(\mathbf{U}_\mu^L)^T \mathbf{U}_\mu^L = I_{r_\mu}$. As a result, it is also true that $\mathbf{X}_{\leq\mu}\mathbf{X}_{\leq\mu}^T = I_{r_\mu}$, for $\mu = 1, ..., d-1$. Similarly, if right-orthogonalization is performed on the second core, then for the third and so on, we obtain a right-orthogonal tensor, meaning that $\mathbf{U}_\mu^R(\mathbf{U}_\mu^R)^T = I_{r_\mu}$. In that case, we also have $\mathbf{X}_{\geq\mu}\mathbf{X}_{\geq\mu}^T = I_{r_\mu}$, for $\mu = 2, ..., d$.

There is a more general definition concerning orthogonality than the possibilities of a tensor being either left-orthogonal or right-orthogonal. If $(\mathbf{U}_\mu^L)^T \mathbf{U}_\mu^L = I_{r_\mu}$ for $\mu = 1, ..., \nu - 1$, and $\mathbf{U}_\mu^R(\mathbf{U}_\mu^R)^T = I_{r_\mu}$ for $\mu = \nu + 1, ..., d$, then the tensor is said to be ν-orthogonal. In particular, being left-orthogonal is equivalent to being d-orthogonal while right-orthogonal is to being 1-orthogonal.

Addition. Consider two tensors $\mathbf{X}$ and $\mathbf{Y}$ such that $\mathrm{rank}_{\mathrm{TT}}(\mathbf{X}) = (r_0, r_1, ..., r_d)$ and $\mathrm{rank}_{\mathrm{TT}}(\mathbf{Y}) = (\tilde{r}_0, \tilde{r}_1, ..., \tilde{r}_d)$, represented in the format from (2.2) as

$$\mathbf{X}(i_1, ..., i_d) = U_1(i_1)U_2(i_2)\cdots U_d(i_d),$$
$$\mathbf{Y}(i_1, ..., i_d) = \tilde{U}_1(i_1)\tilde{U}_2(i_2)\cdots \tilde{U}_d(i_d).$$

The result of adding the two tensors, $\mathbf{Z} = \mathbf{X} + \mathbf{Y}$, can be represented as a TT tensor of TT rank $(r_0 + \tilde{r}_0, r_1 + \tilde{r}_1, ..., r_d + \tilde{r}_d)$,

$$\mathbf{Z}(i_1, ..., i_d) = V_1(i_1)V_2(i_2) \cdots V_d(i_d),$$

where

$$V_1(i_1) = \begin{bmatrix} U_1(i_1) & \tilde{U}_1(i_1) \end{bmatrix}, \quad V_d(i_d) = \begin{bmatrix} U_d(i_d) \\ \tilde{U}_d(i_d) \end{bmatrix},$$

and

$$V_\mu(i_\mu) = \begin{bmatrix} U_\mu(i_\mu) & 0 \\ 0 & \tilde{U}_\mu(i_\mu) \end{bmatrix}, \ \mu = 2, ..., d-1.$$

Adding two tensors results in a tensor with summed entries of the corresponding TT rank vectors.

Inner product. The inner product between two tensors represented in TT format consists of the Euclidean inner product of their vectorizations.

Noting that

$$\text{vec}(\mathbf{X}) = \mathbf{X}_{\neq 1}\text{vec}(\mathbf{U}_1) = (\mathbf{X}_{\geq 2} \otimes I_{n_1})\text{vec}(\mathbf{U}_1) = \text{vec}(\mathbf{U}_1 \times_3 \mathbf{X}_{\geq 2}),$$

the inner product can be efficiently computed reformulating it as follows:

$$\begin{aligned}\langle \mathbf{X}, \mathbf{Y} \rangle &= \text{vec}(\mathbf{U}_2)^T \mathbf{X}_{\neq 2}^T \mathbf{Y}_{\neq 2}\text{vec}(\mathbf{V}_2) \\ &= \text{vec}(\mathbf{U}_2)^T (\mathbf{X}_{\geq 3}^T \otimes I_{n_2} \otimes I_{r_1})(\mathbf{Y}_{\geq 3} \otimes I_{n_2} \otimes \mathbf{X}_{\leq 1}^T \mathbf{Y}_{\leq 1})\text{vec}(\mathbf{V}_2) \\ &= \text{vec}(\mathbf{U}_2 \times_3 \mathbf{X}_{\geq 3})^T \text{vec}(\mathbf{V}_2 \times_1 \mathbf{X}_{\leq 1}^T \mathbf{Y}_{\leq 1} \times_3 \mathbf{Y}_{\geq 3}) \\ &= \langle \mathbf{U}_2 \times_3 \mathbf{X}_{\geq 3}, \mathbf{V}_2 \times_1 \mathbf{X}_{\leq 1}^T \mathbf{Y}_{\leq 1} \times_3 \mathbf{Y}_{\geq 3} \rangle.\end{aligned}$$

Therefore, we can take instead the inner product of two $(d-1)$-dimensional tensors, obtained by removing the first core of each involved tensor and changing the second core of the second appropriately – $\mathbf{V}_2$ becomes $\mathbf{V}_2 \times_1 \mathbf{X}_{\leq 1}^T \mathbf{Y}_{\leq 1} = \mathbf{V}_2 \times_1 (\mathbf{U}_1^L)^T \mathbf{V}_1^L$.

As a result, the inner product can be obtained by successively reducing the dimensionality of the involved tensors. The idea is to obtain the matrices $(\mathbf{U}_\mu^L)^T \mathbf{V}_\mu^L$ and multiply the result to the next core $\mathbf{V}_{\mu+1}$, starting from $\mu = 1$ and repeating the procedure until the last case, $\mu = d$, is reached.

Concerning the norm, induced by the inner product, if $\mathbf{X}$ is ν-orthogonal, then $\mathbf{X}_{\neq \nu}^T \mathbf{X}_{\neq \nu} =$

$I_{r_{\nu-1}n_\nu r_\nu}$, so that

$$||\mathbf{X}|| = \sqrt{\mathrm{vec}(\mathbf{U}_\nu)^T \mathbf{X}_{\neq\nu}^T \mathbf{X}_{\neq\nu} \mathrm{vec}(\mathbf{U}_\nu)} = \sqrt{\mathrm{vec}(\mathbf{U}_\nu)^T \mathrm{vec}(\mathbf{U}_\nu)} = ||\mathbf{U}_\nu||.$$

This means that obtaining the norm of the tensor only requires the computation of the norm of the ν-th core.

Truncation. In order to obtain a TT tensor with a certain TT rank, a generalization of the truncated SVD procedure for matrices can be used. The procedure is called *TT-SVD* [Ose11b] and it basically requires that truncated SVD is applied to the different μ-th mode unfoldings of the tensor, for $\mu = 1, ..., d-1$.

Given a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, with TT rank $(r_0, r_1, ..., r_d)$, we successively apply the best rank-r_μ approximation to the dimension μ, which we denote by $P^\mu_{r_\mu}$, starting from the first mode and finishing in the mode $d-1$. Such a projection is described by $(P^\mu_{r_\mu}\mathbf{X})^{\langle\mu\rangle} = QQ^T\mathbf{X}^{\langle\mu\rangle}$, where $Q \in \mathbb{R}^{n_\mu \times r_\mu}$ contains the first r_μ left singular vectors of $\mathbf{X}^{\langle\mu\rangle}$. The whole projection can be represented as $P^{\mathrm{TT}}_{\mathbf{r}} = P^{d-1}_{r_{d-1}} \circ \cdots \circ P^1_{r_1}$. Note that this projection goes from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to the manifold containing the tensors of TT rank $(r_0, r_1, ..., r_d)$.

While, as opposed to truncated SVD, TT-SVD does not fulfil the best approximation property, it still fulfils a quasi-best approximation one. This is stated in [Ose11b], and we also state it now, noting that $\mathscr{M}_r$ stands for the manifold with tensors of TT-rank $\mathbf{r}$.

Theorem 1. *The result of applying the projection $P^{TT}_{\mathbf{r}}$ to tensor $\mathbf{X}$, represented by $P^{TT}_{\mathbf{r}}(\mathbf{X})$, is such that*

$$||\mathbf{X} - P^{TT}_{\mathbf{r}}(\mathbf{X})|| \leq \sqrt{d-1}||\mathbf{X} - P_{\mathscr{M}_\mathbf{r}}(\mathbf{X})||, \tag{2.6}$$

where $P_\mathbf{r}(\mathbf{X})$ is the projection associated with the best approximation of $\mathbf{X}$ within the set of tensors with TT rank $\mathbf{r}$.

If $\mathbf{X}$ is already given in TT format, with TT rank $\mathbf{r}$, TT-SVD truncation to a prescribed TT rank $\tilde{\mathbf{r}}$ can be obtained efficiently. Assuming that $\mathbf{X}$ is d-orthogonal, it will become $(d-1)$-orthogonal if the SVD of $\mathbf{U}^R_d$, QSV^T, is used to define new cores by setting $\mathbf{U}^R_d$ to V^T and $\mathbf{U}^L_{d-1}$ to $\mathbf{U}^L_{d-1}QS$. Furthermore, if only $\tilde{r}_{d-1}$ singular values are kept, r_{d-1} reduces to $\tilde{r}_{d-1}$. This needs to be repeated until the core $\mathbf{U}_1$ is reached in order for the TT rank to become $\tilde{\mathbf{r}}$ instead of the original $\mathbf{r}$.

Note that such truncation only makes sense if each individual entry of $\tilde{\mathbf{r}}$ is smaller than the corresponding entry of $\mathbf{r}$.

An alternative to the upper bound in (2.6) for $||\mathbf{X} - P^{\mathrm{TT}}_{\mathbf{r}}(\mathbf{X})||$ that is more convenient follows.

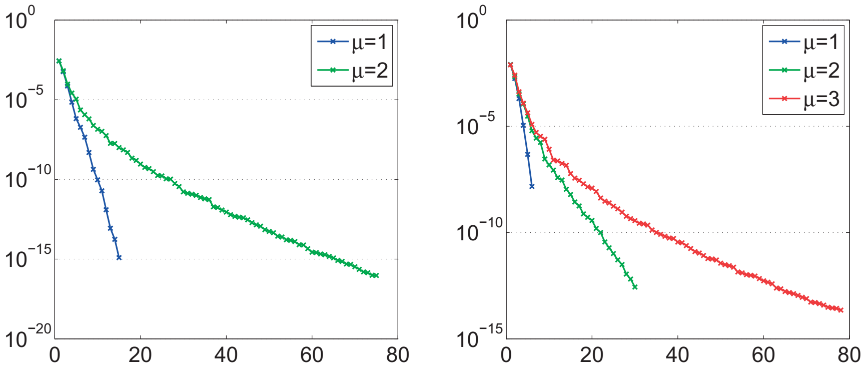


Figure 2.1: Singular values of $\mathbf{X}^{\langle\mu\rangle}$ for the stationary distribution of the large overflow model, for $d = 4$ (left plot) and $d = 6$ (right plot).

Theorem 2. *The result of applying the projection* $P_{\mathbf{r}}^{TT}$ *to tensor* $\mathbf{X}$*, represented by* $P_{\mathbf{r}}^{TT}(\mathbf{X})$*, is such that*

$$||\mathbf{X} - P_{\mathbf{r}}^{TT}(\mathbf{X})| \leq \sum_{\mu=1}^{d-1} \sum_{j=r_{\mu+1}}^{n_\mu} \sigma_j\left(\mathbf{X}^{\langle\mu\rangle}\right)^2, \tag{2.7}$$

where $\sigma_j(\cdot)$ *denotes the* j*-th largest singular value of a matrix.*

This upper bound is in fact strongly related with the TT-SVD procedure, in which the goal is to obtain the smallest possible TT rank entries for a given desired accuracy. This is a fundamental operation in the algorithms proposed in this format as we will explain in the following chapters when discussing them.

As in the matrix case, the success of the application of low-rank tensor methods strongly relies on the data – in our case, the stationary probability distribution – being well approximated by a low-rank tensor or not. According to (2.7), this can be quantified by considering the singular values of the different unfoldings. Good accuracy can only be expected when their singular values decay sufficiently fast.

We now exemplify this for a concrete model of an overflow queuing network that has been extensively used; see, e.g., [Buc00]; to test algorithms for solving the exact same problem that we address in the first part of this thesis.

Figure 2.1 displays the singular values of the relevant matricizations of the mentioned model. We consider two cases: $d = 4$ with $n_\mu = 20$ states per queue; and $d = 6$ with $n_\mu = 6$. Note that only the first half of the matricizations are considered since the singular values of the second half display a similar behaviour. It turns out that the singular values have a very fast decay, showing that this model can be well approximated with very low entries of the TT rank. This was expected since this model has a strong underlying Kronecker structure; see a more detailed description of it in Section 3.4.1.

In general, statements on the size of the values of the TT rank are hard to find. Such statements can be found in [KS15] for a particular type of model. Furthermore, we here show, again for a particular model, that the exact solution can be approximated with very small TT rank entries.

2.2.2 Linear operators acting on tensors in TT format

It is crucial to define the multiplication of a tensor with a matrix in the context of the need for applying a linear operator to a tensor in the algorithms that we will propose later. The goal is to keep the low-rank structure after the operation is done, while trying to apply the operator efficiently. In order for this to be possible, the linear operator needs a suitable representation.

Given a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, and a linear operator $\mathscr{A}$ that goes from $\mathbb{R}^{n_1 \times \cdots \times n_d}$ to $\mathbb{R}^{m_1 \times \cdots \times m_d}$, whose matrix representation is A, the goal is to obtain the result $\mathbf{Y} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ such that

$$\mathbf{Y} = \mathscr{A}\mathbf{X} \Leftrightarrow \text{vec}(\mathbf{Y}) = A\text{vec}(\mathbf{X}).$$

Representation of a matrix in TT format The convenient representation of a matrix in TT format is obtained through the so-called *operator TT format*, introduced in [Ose11a]. Similarly to what is done in the representation of a tensor, recall (2.2), the entries of $A \in \mathbb{R}^{m_1 m_2 \cdots m_d \times n_1 n_2 \cdots n_d}$ are represented as

$$\mathbf{A}_{\xi(i_1,\ldots,i_d),\xi(j_1,\ldots,j_d)} = A_1(i_1, j_1) A_2(i_2, j_2) \cdots A_d(i_d, j_d), \tag{2.8}$$

where the index map ξ is associated with the index ordering of vectorization; recall (2.1).

Representing the associated TT rank vector, whose entries are defined exactly as those of a TT tensor, by $(R_0, R_1, \ldots, R_d)$, each $A_\mu(i_\mu, j_\mu)$, $\mu = 1, \ldots, d$, is a matrix in $\mathbb{R}^{R_{\mu-1} \times R_\mu}$. It is alternatively possible to see each $A_\mu(i_\mu, j_\mu)$ in terms of a four-dimensional tensor $\mathbf{A}_\mu(:, i_\mu, j_\mu, :) = A_\mu(i_\mu, j_\mu)$, where $\mathbf{A}_\mu(:, i_\mu, j_\mu, :) \in \mathbb{R}^{R_{\mu-1} \times m_\mu \times n_\mu \times R_\mu}$. This reminds of the alternatives associated with the representation of a tensor, where (2.3) could be used instead of (2.2).

Note that such an operator is expected to have small entries of the corresponding TT rank if there is an underlying Kronecker structure, emphasizing how this format is particularly oriented for this type of structure.

The tensor rank of a matrix in CP format is defined following the same reasoning as the corresponding definition when considering a tensor instead, as in TT format. In sequence of the discussion based on (1.7) in Section 1.1, it is the smallest V that allows a representation as (1.3). An interesting relation between the TT rank and the tensor rank associated with the representation of a given matrix in TT and CP format, respectively, follows. Given a matrix with tensor rank R, we can represent this matrix through an operator TT format whose associated entries of the TT rank are not larger than R. In fact, if $A = \sum_{i=1}^{R} L_d^{(i)} \otimes \cdots \otimes L_1^{(i)}$, the cores of the corresponding operator TT format are

$$A_1(i_1, j_1) = \begin{bmatrix} L_1^{(1)}(i_1, j_1) & L_1^{(2)}(i_1, j_1) & \dots & L_1^{(R)}(i_1, j_1) \end{bmatrix}, \quad A_d(i_d, j_d) = \begin{bmatrix} L_d^{(1)}(i_d, j_d) \\ L_d^{(2)}(i_d, j_d) \\ \vdots \\ L_d^{(R)}(i_d, j_d) \end{bmatrix},$$

and

$$A_\mu(i_\mu, j_\mu) = \begin{bmatrix} L_\mu^{(1)}(i_\mu, j_\mu) & & & \\ & L_\mu^{(2)}(i_\mu, j_\mu) & & \\ & & \ddots & \\ & & & L_\mu^{(R)}(i_\mu, j_\mu) \end{bmatrix}, \ \mu = 2, \dots, d-1.$$

In particular, for a simple Kronecker product, all entries of the TT rank are equal to 1. This is clear given that the cores are just the different matrices involved in the Kronecker product.

Our problems of interest are those for which the generator rate matrix is possible to write as (1.3) with small V. Such cases are also known to provide a simple TT representation. Moreover, the TT rank entries can be, in general, further reduced. We now exemplify this, again for the already mentioned overflow queuing network.

For the mentioned model, recalling (1.4), the matrices $L_\mu \in \mathbb{R}^{n_\mu \times n_\mu}$ associated with (1.5) contain arrival and departure rates. $\mathbf{Q}_I$, in turn, is given by

$$\sum_{1 \le \mu_1 < \mu_2 \le d} I_d \otimes \cdots \otimes I_{\mu_2+1} \otimes B_{\mu_2} \otimes C_{\mu_2-1} \otimes \cdots \otimes C_{\mu_1+1} \otimes D_{\mu_1} \otimes I_{\mu_1-1} \otimes \cdots \otimes I_1$$

for some matrices $B_\mu, C_\mu, D_\mu \in \mathbb{R}^{n_\mu \times n_\mu}$. Therefore, the tensor rank of this operator is $\frac{d(d+1)}{2}$.

Obtaining the corresponding operator TT format with the smallest possible TT rank entries, given the representation in CP format, can be done using a strategy that is detailed in [KK12]. This is also the approach used to obtain the representation in operator TT format of the different models in [Mac15]. Having a representation of the model as in (1.3), Kronecker representation, is then enough to allow the corresponding TT representation to be obtained.

By direct calculation, using the mentioned strategy, it can be shown that the cores that characterize the operator TT format are, for this model,

$$A_1(i_1, j_1) = \begin{bmatrix} L_1(i_1, j_1) & B_1(i_1, j_1) & I_1(i_1, j_1) \end{bmatrix}, \quad A_d(i_d, j_d) = \begin{bmatrix} I_d(i_d, j_d) \\ D_d(i_d, j_d) \\ L_d(i_d, j_d) \end{bmatrix},$$

and

$$A_\mu(i_\mu, j_\mu) = \begin{bmatrix} I_\mu(i_\mu, j_\mu) & 0 & 0 \\ C_\mu(i_\mu, j_\mu) & B_\mu(i_\mu, j_\mu) & 0 \\ L_\mu(i_\mu, j_\mu) & D_\mu(i_\mu, j_\mu) & I_\mu(i_\mu, j_\mu) \end{bmatrix}, \quad \mu = 2, \ldots, d-1.$$

The TT rank associated with this operator is $(1, 3, ..., 3, 1)$ – no entry of the TT rank exceeds the value 3. This is independent of the choice of d. The complexity of storing and performing operations will be significantly reduced when such entries of the TT rank remain modest. This should be the case, as we already noted and tried to exemplify here, when structured models are considered.

If there is additionally an underlying topology of the processes associated with a train, then particularly small entries in the TT rank of the corresponding operator TT format are expected, in sequence of Remark 1.

In practice, it can be additionally observed from the operator TT format representations of different models in the benchmark collection [Mac15] that the entries of the associated TT rank are in fact very small even for less favourable underlying topologies. This suggests that algorithms in this format should be possible to apply efficiently even in such contexts. We will in fact verify this in the experiments of the algorithms proposed in this first part of the thesis.

Applying an operator TT format to a tensor in TT format After extensively introducing how to represent linear operators in this format, it now becomes easy to describe how to efficiently apply them to a TT tensor. The fact that the result of applying the linear operator $\mathscr{A}$ in operator TT format to the TT tensor $\mathbf{X}$ is still in TT format is crucial.

The resulting tensor $\mathbf{Y}$ can be computed element-wise as described in [Ose11b]:

$$\begin{aligned}\mathbf{Y}(i_1,...,i_d) &= \sum_{j_1=1}^{n_1}\cdots\sum_{j_d=1}^{n_d}\mathscr{A}(i_1,...,i_d,j_1,...,j_d)\mathbf{X}(j_1,...,j_d)\\ &= \sum_{j_1=1}^{n_1}\cdots\sum_{j_d=1}^{n_d}A_1(i_1,j_1)\cdots A_d(i_d,j_d)U_1(j_1)\cdots U_d(j_d)\\ &= \sum_{j_1=1}^{n_1}\cdots\sum_{j_d=1}^{n_d}\left(A_1(i_1,j_1)\otimes U_1(j_1)\right)\cdots\left(A_d(i_d,j_d)\otimes U_d(j_d)\right).\end{aligned}$$

Therefore, the μ-core of the resulting tensor can be obtained by $\sum_{j_\mu=1}^{n_\mu} A_\mu(i_\mu,j_\mu)\otimes U_\mu(j_\mu)$.

For an efficient implementation of this operation, the explicit Kronecker products $A_\mu(i_\mu,j_\mu)\otimes U_\mu(j_\mu)$ are not formed. The summation is performed instead by multiplying two matrices associated with suitable matricizations: $(\mathbf{A}_\mu)_{(3)}^T(\mathbf{U}_\mu)_{(2)}$, where $(\mathbf{A}_\mu)_{(3)}^T \in \mathbb{R}^{R_{\mu-1}m_\mu R_\mu \times n_\mu}$; and $(\mathbf{U}_\mu)_{(2)} \in \mathbb{R}^{n_\mu \times r_{\mu-1}r_\mu}$. The corresponding core, the μ-th core, is then obtained by appropriately reordering the entries of the obtained matrix of size $R_{\mu-1}m_\mu R_\mu \times r_{\mu-1}r_\mu$ into a tensor of size $R_{\mu-1}r_{\mu-1} \times m_\mu \times R_\mu r_\mu$. In particular, the corresponding entry of the TT rank is bounded by $R_\mu r_\mu$, $\mu = 1,...,d-1$.

3 Alternating optimization schemes

This chapter concerns the first proposed techniques for solving (1.1). We propose and compare different algorithms. All these algorithms have their structures in TT format. We start with a simple iterative solver that formulates (1.1) as an eigenvalue problem, proceeding then to an alternating optimization scheme that uses a formulation based on the equivalent problem (1.2). Our idea is exactly to focus on alternating optimization schemes, while the simple iterative solver is mostly used as reference for comparison. A third algorithm that allows the ranks to be flexible is then proposed; it is based on the same core principles as the second, again based on an alternating optimization scheme, but adding a step that includes a natural rank adaptivity scheme. The use of these algorithms is then illustrated on two networks, which are queuing networks that were used in the past for this same purpose.

We start by introducing the historical context of the application of alternating optimization schemes combined with low-rank tensor formats to our problem.

An important property of all tensor formats is their *multilinearity*. This allows extending powerful tools from linear algebra to the tensor setting.

In particular, the optimization problem in (1.2) can be formulated in terms of tensor formats, making use of the underlying low-rank structure. This has been done in [Buc10], where CP format was used. The idea is to solve the original problem but under the additional restriction that $\mathbf{x}$, tensor, represented in CP format, that represents the vector x from the original formulation, is in the manifold $\mathscr{M}_R$ of tensors of a fixed tensor rank R:

$$\min_{\mathbf{x}\in\mathscr{M}_R} \{||\mathbf{Q}^T\mathbf{x}|| : \mathbf{e}^T\mathbf{x} = 1\}, \tag{3.1}$$

where $\mathbf{e}$ is the representation of the vector of all ones, also in CP format. Recall that if Q has a structure of the form (1.3), then Q^T also does, as noted in Section 1.1.

Note that while we used the notation $\mathbf{X}$, with capital letter, for a tensor in CP format in (1.7), we now use $\mathbf{x}$ instead. The idea is that it represents a tensor obtained from reshaping what initially was a vector in the original formulation, so that this notation should be more intuitive. The same will be done for tensors in TT format that represent vectors.

Under this formulation, the algorithm applied for finding an approximation of the solution was the standard *alternating least squares* (ALS) scheme [Hit27]. Such methods first appeared in quantum physics; for instance the so-called DMRG method for addressing eigenvalue problems for strongly correlated quantum lattice systems, see [Sch11] for an overview. The basic idea is to, given a least squares problem, (3.1) in this case, partition the set of degrees of freedom of the problem in subsets, performing the optimization associated with each subset at a time, while all other degrees of freedom are fixed; see [Buc10] for details concerning the way to suitably partition the degrees of freedom in this particular problem associated with CP format. These ideas have been extended to linear systems in the numerical analysis community [OD12, HRS12], so that in the end we can think of a more global class of alternating optimization schemes.

Our main argument for using TT format instead of CP is that, when CP format is used, the tensor rank needed to get a good accuracy in the approximation of the solution is usually too large, as observed in the numerical experiments in [Buc10]. In turn, when using TT format, we expect the TT rank associated with the TT tensor representing the exact stationary distribution to have small entries, as already exemplified in Section 2.2 for a model among those considered in [Buc10]. This is seen in more detail later in the experiments of this chapter. Such large tensor rank is explained by the argument for preferring TT format over CP, described in Section 1.1, that CP format does not consider the topology of the interactions. This in fact explains such large tensor ranks. Another argument for preferring TT format over CP that was also described in Section 1.1, the fact that truncation cannot be done efficiently in CP format, affects two of the three algorithms that are proposed in this chapter. The algorithm that it does not affect is the core alternating least squares scheme, since one of its main drawbacks is the fact that there is no rank adaptivity, as we will see. However, the importance of including rank adaptivity will emphasize the importance of such truncation steps.

The content of this chapter is based on the paper [KM14].

Before exploring such alternating scheme in detail, we introduce an algorithm that will be used as reference for comparison. It is a simple algorithm for addressing (1.1), and which we next adapt to TT format.

3.1 Truncated power method

We will first consider an adaptation of the simplest method to compute a single extremal eigenvalue. This method is then adjusted to work when all involved structures are in TT format.

3.1.1 Power method

We first introduce the application of the method to the matrix case. The method is slightly adapted to fit our particular problem of interest.

In the context of the original problem, applying a time discretization with time step $\Delta t > 0$ to the matrix Q, we obtain

$$P = I + \Delta t Q. \tag{3.2}$$

The problem (1.1) becomes equivalent to the eigenvalue problem

$$P^T x = x, \quad e^T x = 1. \tag{3.3}$$

Furthermore, the eigenvector we are interested in is the one associated with the eigenvalue with the largest magnitude if $\Delta t > 0$ is sufficiently small to guarantee that P is a non-periodic stochastic matrix. In fact, this guarantees that the largest eigenvalue of P, and equally of P^T, is 1; see [Ste94, Ch. 1]. The condition on Δt is, as also noted in [Ste94, Ch. 1],

$$0 < \Delta t < (\max_i |Q(i,i)|)^{-1}. \tag{3.4}$$

In this context, the well-known *power method* can be used. This method is the simplest possible for computing the eigenvector associated with the largest eigenvalue of a matrix $A \in \mathbb{R}^{n \times n}$, given an initial guess $x^{(0)} \in \mathbb{R}^n$, by forming the sequence $\{x^{(k)}\}_{k=0}^{+\infty}$, where

$$x^{(k)} := Ax^{(k-1)}, \quad k = 1, 2, ... \tag{3.5}$$

It is clear that

$$\mathbf{x}^{(k)} = A^k \mathbf{x}^{(0)}.$$

As explained above, in our particular problem, we know that this eigenvalue takes the value 1.

Since we are interested not only in the direction of the vector but also on its length, we might need to normalize the iterates from (3.5), recalling the extra constraint $e^T x = 1$ in

(3.3), restriction on the 1-norm of the vector. However, this is not the case since this property is kept during an iteration.

Δt should be as large as possible since the convergence rate of this algorithm increases with the ratio between the first and the second eigenvalue [Ste94, Ch. 1]. Recalling (3.4), this is equivalent to stating that the value should be as close to $(\max_i |Q(i,i)|)^{-1}$ as possible.

3.1.2 Power method in TT format

Truncated power method is the name of the version of power method where all involved structures are represented in terms of a low-rank tensor format. More concretely, as in the case of ALS, this algorithm was first proposed in the context of CP format; see [BM05]. The word *truncated* is associated with the need for adding truncation steps to the original algorithm in order to prevent the ranks from becoming too large. The same core idea can be applied but considering TT format instead. The particular advantage of TT format is that it allows truncations to be done efficiently, given that the format is based on SVD. Such truncations are based on the TT-SVD algorithm described in Section 2.2.1.

The core of the algorithm is exactly as described for the matrix case, in Section 3.1, but Q is now represented in terms of the corresponding operator TT format (2.8), so that the same holds for P in (3.2) and in (3.3). As for e and x in (3.3), it is also their representation as TT tensors that is considered.

Concerning the choice of Δt, since the matrix Q is only given implicitly, the approximation of $\max_i |Q(i,i)|$ that is possible to obtain may not be very accurate. In any case, recalling (3.4), we at least have to be sure that we consider a value that is an upper bound for this quantity. As already noted, the representation of Q in terms of an operator TT format is strongly associated with the corresponding representation in the form (1.3), Kronecker representation. Therefore, an inexpensive upper bound is given by

$$\prod_{\mu=1}^{d} \max_i |E_\mu^{(1)}(i,i)| + \cdots + \prod_{\mu=1}^{d} \max_i |E_\mu^{(T)}(i,i)|.$$

Initial approximation of the solution. The algorithm is initialized considering a solution based on the uniform distribution – the probability associated with all states is the same.

Truncations. Truncation is needed during a cycle to prevent excessive rank growth, as already noted. We use the TT-SVD algorithm, which truncates the tensor back to lower entries of the TT rank, within a specified tolerance. Such algorithm is performed after each step where the TT rank entries are allowed to increase.

We use an adaptive scheme where the target accuracy is a function of the residual norm after the previous iteration. More concretely, it coincides with its value. The motivation is that we should be more and more accurate as we become closer to an approximate solution of the problem that has the desired accuracy, while being too accurate in the initial stages of the algorithm would make the iterations unnecessarily expensive.

Normalization. In (3.3), we have the restriction that the sum of the entries of the solution must be 1. While this property would be preserved by the power method in general, it is lost in our particular setting, due to the introduced truncations. As a result, we need to normalize the obtained approximation after each iteration. This normalization could be performed only in the last step of the algorithm, instead of in each iterate. Moreover, even if this constraint did not exist, it would be likely to observe in practice that the iterates would, at some point, underflow or overflow – for small or large $||A||$, respectively. Therefore, some kind of normalization should in fact be done in any case to prevent this.

The required inner product between the TT representations of the vectors e and the one associated with the approximate solution obtained after the iteration is inexpensive since the TT tensor representing e has all entries of the TT rank equal to 1 [Ose11b].

3.2 Alternating least squares

As introduced in the beginning of this chapter, defining an alternating optimization scheme in TT format has already been done. However, such optimization scheme has only been formulated for the solution of linear systems. In fact, a formulation that particularly suits our problem, recall (3.1), has only been developed in CP format in [Buc10].

Alternating schemes are expected to be suitable for TT format since the required partition of the degrees of freedom of the problem is naturally defined taking into account that each individual TT core enters the TT decomposition (2.2) linearly, recall (2.5). As a consequence, the optimization with respect to a single TT core (while keeping all other cores fixed) should pose no problem.

The formal way to represent (1.2) in TT format is, again viewing x as a tensor $\mathbf{x}$,

$$\min_{\mathbf{x}\in\mathscr{M}_{\mathbf{r}}} \{||\mathbf{Q}^T\mathbf{x}|| : \langle \mathbf{e}, \mathbf{x}\rangle = 1\}, \tag{3.6}$$

where $\mathscr{M}_r$ is the manifold of tensors having fixed TT rank $\mathbf{r}$; $\mathbf{e}$ is again the representation of the vector of all ones, now in TT format. As for CP format, the representation of $\mathbf{Q}^T$

in TT format is trivial if the one for $\mathbf{Q}$ also is.

This constrained optimization problem, because of the rank constraint, is a highly nonlinear problem with multiple minima, whose solution is by no means simple. This was also the case for the corresponding formulation in CP format, in (3.1).

In sequence of the detailed introduction to TT format from Section 2.2, we know that it is easy to isolate one core, so that ALS should be particularly suitable to this format. More concretely, the partition of the degrees of freedom of the problem should be such that the variables that are together are those belonging to the same core. Thus, in a given step, the k-th core should be optimized, while the others remain fixed; this will be called the *subproblem* of the ALS procedure from here on.

Assume that the representation of x in TT format, $\mathbf{x}$, is, entry-wise,

$$\mathbf{x}(i_1, ..., i_d) = G_1(i_1)G_2(i_2)\cdots G_d(i_d), \tag{3.7}$$

which means that the TT cores that compose it are G_1, ..., G_d. The formulation of the optimization that must be solved in the k-th core, $k = 1, ..., d$, can be easily obtained recalling that, given the left and right interface matrices $\mathbf{x}_{\leq k-1}$ and $\mathbf{x}_{\geq k+1}$, respectively, and the vectorization of the degrees of freedom of the k-th core, $\mathbf{g}_k$, we know, from (2.5), that

$$\mathrm{vec}(\mathbf{x}) = \mathbf{x}_{\neq k}\mathbf{g}_k,$$

where $\mathbf{x}_{\neq k} = \mathbf{x}_{\geq k+1} \otimes I_{n_k} \otimes \mathbf{x}_{\leq k-1}$.

It is assumed that the columns of the mentioned left and right interface matrices are orthonormal.

Inserting the relation above into (3.6) yields

$$\min_{\mathbf{x}_{\neq k}\mathbf{g}_k \in \mathscr{M}_{\mathbf{r}}} \{||\mathbf{Q}^T\Big(\mathbf{x}_{\neq k}\mathbf{g}_k\Big)|| : \langle \mathbf{e}, \mathbf{x}_{\neq k}\mathbf{g}_k\rangle = 1\}. \tag{3.8}$$

In turn, $||\mathbf{Q}^T\Big(\mathbf{x}_{\neq k}\mathbf{g}_k\Big)||$ can be written as $\mathbf{g}_k^T\Big(\mathbf{x}_{\neq k}^T\mathbf{Q}\mathbf{Q}^T\mathbf{x}_{\neq k}\Big)\mathbf{g}_k$, while $\langle \mathbf{e}, \mathbf{x}_{\neq k}\mathbf{g}_k\rangle$ can be written as $(\mathbf{x}_{\neq k}^T\mathbf{e})^T\mathbf{g}_k$. For this reason, the minimization problem is formulated in terms of the vectorization of the k-th core, $\mathbf{g}_k$. This is the subproblem that must be solved on the k-th core.

After the subproblem (3.8) has been solved, the k-th TT core of $\mathbf{x}$ is updated by reshaping $\mathbf{g}_k$ into its k-th core.

Algorithm 1 describes one sweep of ALS. A half sweep (forward sweep) of ALS consists of processing all cores from the left to the right until reaching $k = d$. Similarly, the second

Algorithm 1: ALS sweep for solving (3.6)

```
for $k = 1, 2, ..., d-1$ do
    Replace core $G_k$ by the solution of (3.8), $\mathbf{g}_k$, after suitable reshaping
    Apply orthogonalization to ensure that the updated core $\mathbf{G}_k^L$ is left-orthogonal
end
for $k = d, d-1, ..., 2$ do
    Replace core $G_k$ by the solution of (3.8), $\mathbf{g}_k$, after suitable reshaping
    Apply orthogonalization to ensure that the updated core $\mathbf{G}_k^R$ is right-orthogonal
end
```

half sweep (backward sweep) of ALS consists of processing all cores from the right to the left until reaching $k = 1$. Two subsequent half sweeps constitute a full sweep of ALS, and this is from here on considered as the reference measure of one iteration in such alternating schemes. In turn, the sequence of steps that concern a particular core are called *microiterations.* A microiteration includes not only the described update of a core, but also an orthogonalization step, performed to ensure that the interface matrices are again orthogonal in the subsequent optimization step. This step is also local – it only applies to the updated core.

Solution of the subproblem. The constrained minimization (3.8) is an equality-constrained quadratic program as defined in [NW06, Ch. 16]. According to the same reference, a solution of the minimization problem satisfies the following system of equations:

$$\begin{bmatrix} \mathbf{x}_{\neq k}^T \mathbf{Q}\mathbf{Q}^T \mathbf{x}_{\neq k} & \tilde{\mathbf{e}} \\ \tilde{\mathbf{e}}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{g}_k \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{3.9}$$

where $\mathbf{0}$ is a vector of all zeros and $\tilde{\mathbf{e}} = \mathbf{x}_{\neq k}^T \mathbf{e}$, which is cheap to compute. Note that λ is associated with the vector of Lagrange multipliers; in this context it consists of a single value because there is only one constraint.

The problem to be solved in order to find the local solution, for a certain core, is (3.9). This step is crucial in terms of efficiency since it is, together with orthogonalization, one of the computationally most expensive parts of the algorithm. In fact, the linear system has size $r_{k-1} r_k n_k + 1$. There are two main options: solving the linear system directly or using an iterative solver.

For the efficient application of a direct solver, it is convenient to transform the linear system in a symmetric positive definite one. In fact, it is indefinite but there is just one negative eigenvalue, associated with the linear constraint. This can be done by standard manipulation based on the tools provided in [NW06, Ch. 16] as we next detail.

According to the mentioned reference, given a symmetric positive definite matrix H and a vector a (a is originally a matrix in the reference but it concerns the linear constraints so that, since we only have one, it simply becomes a vector),

$$\begin{bmatrix} H & a \\ a^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} C & E \\ E^T & F \end{bmatrix},$$

$C = H^{-1} - H^{-1}a(a^T H^{-1}a)^{-1}a^T H^{-1}$, $E = H^{-1}a(a^T H^{-1}a)^{-1}$, and $F = -(a^T H^{-1}a)^{-1}$. As a result, the solution of a linear system involving this type of matrix is equivalent to the matrix-vector multiplication between the inverted matrix above and the vector in the right side of the equation of the linear system of interest.

The solution will consist of a vector that is the desired one, plus a last entry λ that is not relevant, associated with the constraint, so that we only need the remaining entries. Such entries are those of $\mathbf{g}_k$. Moreover, our right side is a vector of zeros until the last position, in which we find the value 1. This allows simplifying the expression for the entries associated with $\mathbf{g}_k$. We obtain

$$\mathbf{g}_k = \frac{H^{-1}a}{a^T H^{-1}a}. \tag{3.10}$$

This means that, assuming that the right side of the linear system is as described above, we can simply solve a linear system on the symmetric positive definite matrix H, with an associated right side that is the vector a. After solving this linear system, a normalization step, which requires a simple inner product, is needed.

In our concrete problem, H is $\mathbf{x}_{\neq k}^T \mathbf{Q}\mathbf{Q}^T \mathbf{x}_{\neq k}$, while a is $\mathbf{x}_{\neq k}^T \mathbf{e}$.

As the linear system now concerns a symmetric positive definite matrix, a direct solver as Cholesky decomposition can be applied, and the associated cost is reduced by a factor of $\frac{2}{3}$ compared to the cost of a common direct solver.

The main problem is that the matrix $\mathbf{x}_{\neq k}^T \mathbf{Q}\mathbf{Q}^T \mathbf{x}_{\neq k}$ has size $n_k r_{k-1} r_k$ so that, using a Cholesky decomposition or not, the cost when using a direct solver easily becomes infeasible when the TT rank entries start to increase (or even the mode sizes alone). In fact, the obtained linear system still has a complexity $\mathcal{O}(\hat{r}^6 \hat{n}^3)$, where $\hat{r}$ and $\hat{n}$ are bounds for the TT rank entries and for the mode sizes, respectively.

The solution is to apply an iterative solver. Not only is this the natural solution to the problem associated with the cost of the direct solver, as it additionally allows a speed up associated with the fact that all involved matrix-vector multiplications can be done without explicitly forming the matrix $\mathbf{x}_{\neq k}^T \mathbf{Q}\mathbf{Q}^T \mathbf{x}_{\neq k}$. In fact, this matrix can be also written as a short sum of Kronecker products even though it is not sparse, since the Kronecker structure is inherited by the small TT rank entries of $\mathbf{Q}$; see [KSU14].

Given that the linear system can be converted into symmetric positive definite, as already explained, it might be advantageous to consider this when deciding which iterative solver to use. An immediate idea would be to use conjugate gradient (CG). While this was in fact the iterative solver that was used in the paper on which this chapter is based, [KM14], it was noted that convergence was rarely obtained. This is mostly due to the fact that, unless an effective preconditioner for accelerating convergence is found, its performance is not satisfying, and in fact designing such preconditioners is not simple. In fact, even the knowledge of a preconditioner for the problem in the full-space (1.1) is generally not sufficient, and it has only been possible to convert a preconditioner for the global problem in an effective preconditioner for the subproblems via a very particular construction for Laplace-like operators [KSU14], which is however not relevant for the problem under consideration.

Even without a preconditioner, if a good initial guess for the solution is available, convergence might still be obtained. While such a good initial guess is in fact available in theory, which would be the current approximation for the core that is being optimized, such guess does not have the desired effect due to the normalization step that is then needed; recall (3.10). In fact, the associated term, $a^T H^{-1} a$, is verified to be in general of very large order, so that the initial guess is always extremely far from any good enough approximation of the solution.

A solution is to apply the iterative solver MINRES [Gre97, Saa03] directly to (3.9). In fact, this iterative solver only requires that the matrix is symmetric, which holds in that case. While the problem concerning the fact that no effective preconditioner exists is still a concern, the fact that a good initial guess is available should significantly help reaching convergence. The initial guess that is used consists of considering the previous approximation of the core for the entries associated with $\mathbf{g}_k$ while the initial guess for λ can be simply 0 since it is easy to deduce that its value, using the knowledge that it is the entry associated with the Lagrange multipliers, is $\frac{1}{a^T H^{-1} a}$. As a consequence, the order of the denominator is what is large in this case so that 0 approximates the value well.

Summing up, we can either use a direct solver based on Cholesky factorization taking into account that the solution of the problem can be obtained from (3.10) so that the main step consists of solving a symmetric positive definite linear system; or an iterative solver, which would be applied to (3.9) instead, and for which MINRES should be particularly suitable since the matrix is symmetric, while a good initial guess is available. While the direct solver is expected to be inefficient when the TT rank entries start to grow, the iterative solver has the limitation that convergence may be problematic if the condition number of the matrix of the linear system is large because no good preconditioner is available.

Initial approximation of the solution. An initial approximation must be provided to Algorithm 1. Moreover, it must be a right-orthogonal TT tensor. We consider again the tensor associated with the uniform distribution.

3.3 Alternating minimal energy

There are two main limitations associated with ALS: the entries of the TT rank of the approximate solution have to be set *a priori* and convergence can be slow. However, the core concept behind this algorithm is strong, so that we would like to keep it. In this context, a new algorithm has been proposed that addresses the two mentioned drawbacks while keeping the core structure of ALS. The *alternating minimal energy* (AMEn) method was proposed in [DS14] as a solution for the mentioned problems of alternating schemes, in the context of the already mentioned application of such schemes to the solution of linear systems in TT format. The idea is to enrich the TT cores by adding gradient information, which potentially yields faster convergence than ALS while it allows for rank adaptivity since such enrichments increases the TT rank entries. The mentioned enrichment is only done in the core that is currently being optimized in the ALS sweep, so that the conceptual advantage of ALS that everything is done locally, only in terms of a particular core, is kept, thus allowing to keep the computational effort small.

Such a core enrichment had also been proposed in [Whi05] in the context of the already mentioned DMRG algorithm.

For $d = 2$, considering (3.7) as reference, the TT representation of $\mathbf{x}$ is associated with the cores $G_1 \in \mathbb{R}^{n_1 \times r_1}$ and $G_2 \in \mathbb{R}^{n_2 \times r_2}$ – $\mathbf{x} = G_1 G_2^T$. Suppose that the first step of ALS has been performed and G_1 has been optimized. We then consider a low-rank approximation of the negative gradient of $\frac{1}{2}||\mathbf{Q}^T\mathbf{x}||$:

$$\mathbf{r} = -\mathbf{Q}^T\mathbf{x} \approx R_1 R_2^T.$$

In practice, an approximation of $\mathbf{r}$ with a small rank is typically used. Then the method of steepest descent applied to minimizing $\frac{1}{2}||\mathbf{Q}^T\mathbf{x}||$ would compute

$$\mathbf{x} + \alpha\mathbf{r} \approx \begin{bmatrix} G_1 & R_1 \end{bmatrix} \begin{bmatrix} G_2 & \alpha R_2 \end{bmatrix}^T$$

for some suitably chosen scalar α. We now fix (and orthonormalize) the first augmented core $\begin{bmatrix} G_1 & R_1 \end{bmatrix}$. An increase in an entry of the TT rank results from this enrichment. However, instead of using $\begin{bmatrix} G_2 & \alpha R_2 \end{bmatrix}$, we apply the next step of ALS to obtain an optimized second core via the solution of a linear system of the form (3.9). As a result we obtain a new approximation that is at least as good as the one obtained from one forward sweep of ALS without augmentation; and, at the same time, ignoring the truncation error in $\mathbf{r}$, at least as good as one step of steepest descent. The described procedure

Algorithm 2: AMEn sweep for solving (3.6)

```
for $k = 1, 2, ..., d-1$ do
    Replace core $G_k$ by the solution of (3.8), $\mathbf{g}_k$, after suitable reshaping
    Augment cores $G_k$ and $G_{k+1}$ with cores $R_k$ and $R_{k+1}$, respectively, according to (3.11)
    Apply orthogonalization to ensure that the updated core $\mathbf{G}_k^L$ is left-orthogonal
end
for $k = d, d-1, ..., 2$ do
    Replace core $G_k$ by the solution of (3.8), $\mathbf{g}_k$, after suitable reshaping
    Augment cores $G_k$ and $G_{k+1}$ with cores $R_k$ and $R_{k+1}$, respectively, according to (3.11)
    Apply orthogonalization to ensure that the updated core $\mathbf{G}_k^R$ is right-orthogonal
end
```

is repeated by augmenting the second core and optimizing the second core, and so on. In each step, the rank of $\mathbf{x}$ is adjusted by performing low-rank truncation, meaning in particular that the TT rank entries are also allowed to decrease. This rank adaptivity is one of the major advantages of AMEn.

The generalization to $d > 2$ is straight-forward. Moreover, it follows analogously to [KSU14] by applying the case $d = 2$ to neighbouring cores. We first generalize the representation in TT format of the correction

$$\mathbf{r}(i_1, ..., i_d) = R_1(i_1)R_2(i_2)\cdots R_d(i_d).$$

In the k-th step of the forward ALS sweep, after the k-th core has been optimized, the described procedure must be applied to the k-th and $(k+1)$-th cores. The two cores are augmented with the corresponding cores of the correction $\mathbf{r}$, R_k and R_{k+1}, respectively. The cores G_k and G_{k+1} become $\tilde{G}_k$ and $\tilde{G}_{k+1}$, respectively, where

$$\tilde{G}_k^L = \begin{bmatrix} G_k^L & R_k^L \end{bmatrix} \quad \tilde{G}_{k+1}^R = \begin{bmatrix} G_{k+1}^R & R_{k+1}^R \end{bmatrix}^T. \tag{3.11}$$

In particular, the value of r_k changes. In turn, for the backward sweep, the $(k-1)$-th core is the one that is updated together with the k-th, in a similar manner.

Algorithm 2 describes one sweep of AMEn.

We skip the details about the algorithm that are common to ALS, since in those cases they are simply the same that were already described in Section 3.2. In particular, the initial approximation of the solution and the way to solve the subproblem are the same.

Truncations. As in the context of truncated power method, recall Section 3.1.2, truncations are done during a sweep to prevent excessive rank growth. We use again TT-SVD algorithm. It is again applied after each step where the TT rank entries are allowed to increase.

Again as in truncated power method, we use an adaptive scheme to define the target accuracy of such truncations. Once again, a dependency on the last residual norm that was computed is the reference. In this case, such residual norm concerns the previous microiteration. The target accuracy is in this case the mentioned value divided by 100.

Enrichment rank. As noted before, the typically used values for the rank considered in the approximation of the residual that is used to augment the cores, which is exactly the quantity by which the corresponding entry of the TT rank is allowed to increase, is small. In fact, while we want convergence to be fast, choosing a value that is too large might lead to unnecessarily expensive microiterations. In our experiments, we consider an enrichment rank of 3.

3.4 Numerical experiments

We now investigate the performance of the different proposed methods in TT format: truncated power method, ALS, and AMEn. For reference, we have also implemented the mentioned ALS method for CP decomposition [Buc10], based on functionality from the *tensor toolbox* [BK$^+$12]. To distinguish between the two different ALS algorithms, we will denote them by “ALS-TT” and “ALS-CP”. All algorithms in this first part of the thesis (including those associated with the two chapters that follow) were implemented in Matlab version 2013b, using functions from *TT-Toolbox* [Ose11a].

Concerning the subproblems to be solved in ALS-TT and AMEn, in (3.9), we used a direct solver in general, but we also tried to use the iterative solver MINRES in the context of AMEn for comparison effects. When a direct solver is used, we denote the algorithm simply by “AMEn”; while in the cases where an iterative solver is used, we denote it by “AMEn (MINRES)”.

As for truncated power method, we simply call it “PM”.

Given that the ranks associated with ALS algorithms have to be fixed *a priori*, some criterion had to be used to define what to use in ALS-CP and ALS-TT. In the case of ALS-TT, we used the same value in all entries of the TT rank, corresponding to the maximum among the entries of the TT rank obtained naturally when applying AMEn. This way, we are sure that the algorithm converges if AMEn also does. As for ALS-CP, we tried different tensor ranks, choosing the one exhibiting the best performance.

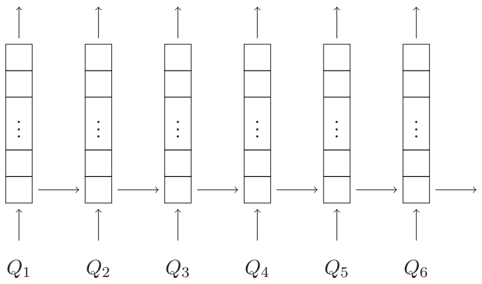


Figure 3.1: Structure of the model `overflow`.

All computation times in this part of the thesis (including those in the next two chapters) were obtained on a 12-core Intel Xeon CPU X5675, 3.07GHz with 192 GB RAM running 64-Bit Linux version 2.6.32.

Throughout all experiments in this thesis (including, again, those in the next two chapters), the stopping criterion is defined in terms of the ratio between the current residual norm, considering the approximate solution at that given step, and the residual norm associated with the tensor of all ones (scaled so that the sum of its entries is one), which is the initial guess for all algorithms tested in this chapter. We denote this measure by relative accuracy from here on.

In the tables that follow: "Time" stands for the computation time, in seconds; "Rank" stands for the maximal entry of the TT rank for the algorithms based on TT format, while for the tensor rank in the case of ALS-CP, of the approximate solution.

3.4.1 First test case: an overflow queuing network

We first consider the model that was used to test ALS-CP in [Buc10] – the already introduced overflow queuing network; recall Section 2.2. All benchmark problems used in this thesis are taken from the already mentioned benchmark collection [Mac15], which not only provides a detailed description of the involved matrices but also Matlab code. In this case, the model is named `overflow` and its structure, for $d = 6$ as in the default example in [Mac15], is depicted in Figure 3.1. Customers which arrive at a full queue try to enter subsequent queues until they find one that is not full. After trying the last queue, they leave the system.

In a first case study, we consider the default parameters from [Mac15], which had been also considered in the experiments in [Buc10]: arrival rates for queues 1 to 6 of 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, respectively; and departure rates 1 for all queues. The maximum capacity of each queue that we consider is 10 so that the mode sizes are all 11, for a total of $11^6 \approx 1.77 \times 10^6$ states.

Table 3.1: Obtained execution times (in seconds) and corresponding ranks for the large overflow model with respect to different relative accuracies.

Relative accuracy	1.0×10^{-1}		4.6×10^{-2}		2.2×10^{-2}		1.0×10^{-2}	
	Time	Rank	Time	Rank	Time	Rank	Time	Rank
ALS-CP	139.1	120	365.8	180	1102.1	240	2612.7	300
PM	35.7	16	48.5	19	65.6	23	85.3	26
ALS-TT	8.0	18	10.9	19	17.3	21	22.7	22
AMEn	4.6	18	5.1	19	6.6	21	9.6	22
AMEn (MINRES)	51.1	19	66.5	21	84.9	23	110.5	27

We have applied ALS-CP to the large overflow model for tensor ranks ranging from 120 to 300 – values associated with the best performances of the algorithm for the different relative accuracies of interest. The considered relative accuracies are displayed in the first row of Table 3.1. We then iterate the algorithms in TT format until the same relative accuracies are reached. The obtained results are shown in rows 4–7 of Table 3.1. Note that all algorithms attained the target relative accuracy.

Due to the lack of an effective preconditioner for MINRES, and despite the fact that a good initial guess is available; recall the related discussion in Section 3.2; the subproblem could not be solved to sufficient accuracy in AMEn (MINRES), thus resulting in a slow convergence – the computation times are clearly worse and even the entries of the TT rank are different. In fact, the associated matrix is very ill-conditioned, so that the convergence of MINRES is severely impaired, often leading to stagnation. Furthermore, we have explored the causes for the ill-conditioning in more detail and verified that it is directly associated with the mode sizes. In this context, noting that the considered mode sizes are as small as 11, this shows how there is no margin in AMEn for increasing the mode sizes if the idea is to be able to apply an iterative solver. Therefore, the reference should be the version of AMEn that solves the subproblem directly. As a consequence, we do not consider AMEn (MINRES) in the experiments that follow.

The execution times of ALS-CP are much larger than those of the TT-based algorithms. The theoretical reasons for this have already been discussed and they mostly concern the fact that the required tensor ranks are particularly large, while the difference in the ranks associated with the operators that represent the generator matrix in each format also play a role. Such ranks have been explicitly stated in Section 2.2.2 for this model.

Among the TT-based algorithms, AMEn is clearly the best and the power method appears to offer the poorest performance. This picture changes, however, when demanding that the approximate solution is more accurate. In fact, this results in larger TT rank entries and consequently makes the solution of the subproblems in AMEn and ALS-TT more

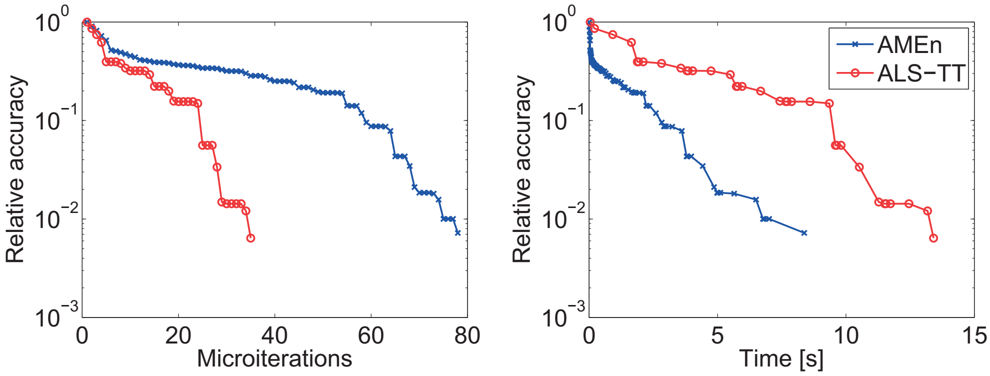


Figure 3.2: Evolution of the relative accuracy with respect to the number of microiterations and with respect to the accumulated execution time (in seconds) for ALS-TT and AMEn applied to the model overflow.

expensive. For example, when demanding a relative accuracy of 10^{-5}, the truncated power method requires 538.2 seconds, ALS-TT 3016.0 seconds, and AMEn 1106.6 seconds.

The left plot of Figure 3.2 shows how the relative accuracy evolves for AMEn and ALS-TT during the microiterations, for the case corresponding to the last column of Table 3.1 – stopping when the relative accuracy is smaller than 10^{-2}. Not surprisingly, ALS-TT converges faster; in fact, it uses the maximal TT rank entry from the final approximation of AMEn right from the first sweep. This picture changes significantly when considering the evolution with respect to the accumulated execution time in the right plot of Figure 3.2. AMEn operates with much smaller TT rank entries during the first sweeps, making them less expensive and resulting in a smaller total execution time despite the fact that the total number of microiterations, and equivalently also of sweeps, is larger. This is representative of the conceptual differences between the two algorithms, and in the end a better computation time is expected when AMEn is considered.

Exploring AMEn. Since the experiments above clearly reveal the advantages of AMEn, we investigate its performance for high-dimensional problems in more detail. For this purpose, we reduce the maximum capacity to 2 customers in each queue, so that all mode sizes are 3, and target a relative accuracy of 10^{-1}. We vary the number of queues from $d = 7$ to $d = 24$. Departure rates are again 1 for all queues while the arrival rates have been adjusted to $\frac{12-0.1\times i}{8}$ for the i-th queue.

Figure 3.3 reveals how the execution time and the maximal TT rank entry grow as d increases. The maximal TT rank entry appears to grow approximately with d^3, while the execution time seems to grow proportionally with d^4. Note that the first statement is important for explaining the second since the cost of the algorithm is strongly associated with the TT rank entries through the cost of the subproblem, which is the main cost of

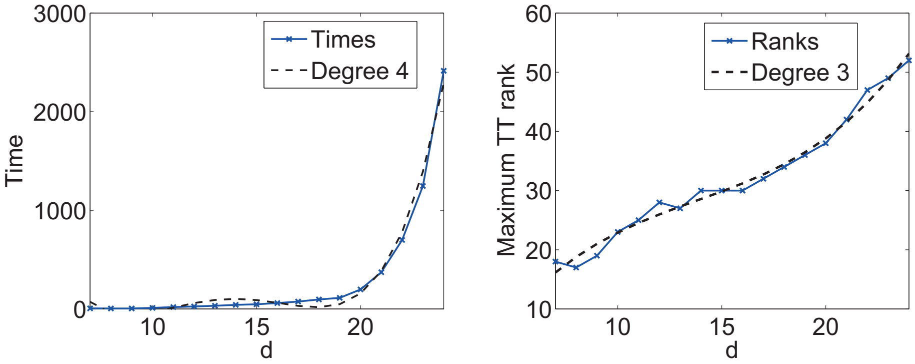


Figure 3.3: Execution time in seconds (left plot) and maximal TT rank entry (right plot) for AMEn applied to the model `overflow` with $d = 7, 8, \ldots, 24$ queues.

the whole algorithm given that a direct solver is used to solve it; as discussed in detail in Section 3.2.

Note that the largest Markov chain is associated with a total of $3^{24} \approx 2.82 \times 10^{11}$ states, which is clearly infeasible for standard solvers, whose cost grows, as already noted in Chapter 1, linearly with this number. In contrast, AMEn requires less than 3000 seconds to obtain a good approximation of the solution.

3.4.2 Second test case: Kanban control model

We now consider the Kanban control model [Buc99], which is another queuing network, where customers arrive in the first queue, being then served in sequence until the last queue, leaving the network afterwards. We assume an infinite source – there is automatically a new arrival when the first queue is not full. A customer that finishes the service and experiences that the next queue is full needs to wait in the current queue. This model is denoted by `kanban` in the benchmark collection [Mac15].

This study is intended to be more brief than for the previous model since we just want to show that the most important conclusions do not depend on the considered model.

We choose $d = 12$ queues, each with a maximum capacity of one customer. The rate of service is 1 for all queues. The time spent travelling from one queue to the next follows an exponential distribution with expected value $\frac{1}{10}$. For the queues 2 to 11, one needs to distinguish the type of customer (already served, waiting to move to the next queue, or no customer), so that there are 3 possible cases, as opposed to what happens in the generality of the models where customers are *indistinguishable*, in which there are only two cases. In fact, in models with indistinguishable customers, the state of a given subsystem is simply characterized by the number of customers since there are only customers of one type so that all that matters is if they are in the queue or not. In

Table 3.2: Obtained execution times (in seconds) and corresponding ranks for the model `kanban`, for a relative accuracy of 10^{-1}.

	Time	Rank
ALS-CP	3024.7	200
PM	3.3	12
ALS-TT	1.4	12
AMEn	0.7	12

this case, customers are *distinguishable* – they can either be customers that were served or that are waiting to be served. The two subclasses of models together, models with distinguishable and indistinguishable customers, naturally form a partition of the class of models associated with customers. In the presence of distinguishable customers, a vector of numbers is needed instead of a simple number to characterize the state of a particular subsystem. This follows the idea in [Buc99], where the states are ordered considering a decreasing lexicographic ordering of such vector. The size of this vector is the number of types of customers plus one, since the first entry is the number of available places in the subsystem, difference between the capacity and the sum of the number of customers of the different types, while the remaining entries are simply associated with the number of customers of the different types. The concept can naturally be generalized to the other types of models – it is not necessarily for queues, so that it does not necessarily concern customers. In this particular case, we have, in total, $2 \times 3^{10} \times 2 \approx 2.36 \times 10^5$ states. The resulting operator TT format has TT rank $(1, 4, ..., 4, 1)$, while the tensor rank that concerns the representation in CP format is $3d - 2 = 34$; as in the previous model, the TT rank entries do not depend on d while this is not the case for the tensor rank.

Table 3.2 shows, for a relative accuracy of 10^{-1}, just as Table 3.1, that, as long as the subproblems that must be solved by AMEn and ALS-TT do not get too expensive, which is the case given that the mode sizes are small but mostly because the entries of the TT rank also remain small, the concept behind the two algorithms, the alternating optimization scheme, is very hard to beat. Furthermore, AMEn beats ALS-TT again. The main information to be extracted from this table is however the clear conclusion that the algorithm based on CP format is outperformed by the algorithms based on TT format that we propose in this chapter, which can be again explained by the exact same theoretical reasons that were discussed for the previous model, in Section 3.4.1.

For this particular model, the entries of the TT rank are naturally even smaller than in the experiments done on the previous model because of the suitable topology that underlies the network, in the context of Remark 1. In fact, this model only features interactions between consecutive queues.

3.5 Conclusion

We have proposed three algorithms for approximating stationary distributions by low-rank TT decompositions: truncated power method, ALS and AMEn. The provided numerical experiments, which feature topologies of interactions that are expected to be particularly suitable to TT format, demonstrate that these methods, in particular AMEn, can perform remarkably well for very high-dimensional problems. In particular, they clearly outperform an existing approach based on CP decompositions.

A bottleneck of ALS-TT and AMEn is that they use a direct solver for the subproblems, which becomes rather expensive for larger entries of the TT rank. Since finding a good preconditioner for the problem does not seem feasible, while we verified that applying an iterative solver without preconditioner easily brings convergence problems, the work that we develop next, presented in the next chapter, was focused on building a new algorithm that incorporates AMEn but in a way that the iterative solver can already be applied efficiently. We also leave for the next two chapters the illustration of the fact that even for models with less suitable underlying topologies, our algorithms in TT format are able to find an approximation of the solution efficiently.

4 A tensorized multigrid scheme combined with AMEn

In this chapter, we consider a tensorized multigrid scheme for solving (1.1). The idea of considering a tensorized scheme is motivated by the structure of the generator matrices of the models of interest, associated with (1.3). Such a scheme has already been proposed in [MB14], but it has the important limitation that the curse of dimensionality still affects it. This can be solved by combining this method with the AMEn method. As we will see, this can be done in a way that the good properties of AMEn are reflected while its drawbacks, discussed in the previous chapter, are not present.

The method proposed in this chapter basically combines the advantages of the two mentioned methods. The tensorized multigrid method from [MB14] allows reducing the mode sizes n_k. This then allows an effective use of the low-rank tensor method AMEn since both the size and, more importantly, the condition number of the subproblems get reduced, allowing the iterative solver MINRES to be effectively applied.

The two algorithms were individually designed to deal with structures in TT format, which is something that is desired since our idea is to build algorithms in TT format. This idea is partially supported by the promising results obtained in the experiments in Section 3.4. Such idea will be even more emphasized in the experiments of both this and the next chapters.

In the end of the chapter, we illustrate the use of this new algorithm on a variety of models from different fields taken from the already introduced benchmark collection [Mac15].

The content of this chapter is based on the paper [BKK$^+$16].

Algorithm 3: Multigrid V-cycle

```
$v_\ell = \mathrm{MG}(b_\ell, v_\ell)$
if coarsest grid is reached then
    Solve coarsest grid equation $A_\ell v_\ell = b_\ell$
else
    Update $v_\ell$ by $\nu_1$ smoothing steps for $A_\ell v_\ell = b_\ell$ with initial guess
    Compute coarse right-hand side $b_{\ell+1} = S^{(\ell)}(b_\ell - A_\ell v_\ell)$
    $e_{\ell+1} = \mathrm{MG}(b_{\ell+1}, 0)$
    $v_\ell = v_\ell + P^{(\ell)} e_{\ell+1}$
    Perform $\nu_2$ smoothing steps for $A_\ell v_\ell = b_\ell$ with initial guess $v_\ell$
end
```

4.1 A tensorized multigrid scheme

We first recall the multigrid method from [MB14] for solving (1.1) considering that the generator matrix has the tensor structure (1.3). Before this, we introduce the generic components of a multigrid method for the matrix case.

4.1.1 Multigrid

Multigrid methods [Hac03] use a set of recursively coarsened representations of the original setting to achieve accelerated convergence. They initially appeared for fast solution of partial differential equations.

A multigrid approach has the following ingredients: the smoothing scheme; the set of coarse variables; the transfer operators (restriction and interpolation operators); the coarsest grid operator.

Algorithm 3 is a prototype of a V-cycle for a general linear system $Ax = b$ that includes the mentioned ingredients. In this algorithm: A_1 is A, b_1 is b, while an initial guess v_1 is considered. For a detailed description we refer the reader to [RS86, TOS01].

In particular, for a two-grid approach, i.e., $\ell = 1, 2$, one can describe the realization as follows: the method performs a certain number ν_1 of smoothing steps, using an iterative solver that can be, for instance, weighted Jacobi or Gauss-Seidel; the residual of the current iterate is computed and restricted by a matrix-vector multiplication with the restriction matrix $S \in \mathbb{R}^{m \times m_c}$; the operator $A_1 = A$ is restricted via a Petrov-Galerkin construction to obtain the coarse-grid operator $A_2 = SA_1P \in \mathbb{R}^{m_c \times m_c}$, where $P \in \mathbb{R}^{m_c \times m}$ is the interpolation operator; then we have a recursive call where we solve the coarse grid equation, which is the residual equation; after this, the error is interpolated and again some smoothing iterations are applied.

We now focus on how to choose m_c and how to obtain the weights for the restriction and interpolation S and P. The value m_c is obtained by specifying coarse variables. Geometric coarsening [TOS01] or compatible relaxation [Bra00, BF10] are methods which partition the given m variables into fine variables $\mathscr{F}$ and coarse variables $\mathscr{C}$ – $m = |\mathscr{C}| + |\mathscr{F}|$. If such a splitting is given, $m_c = |\mathscr{C}|$ and the operators are defined in

$$S : \mathbb{R}^{|\mathscr{C} \cup \mathscr{F}|} \to \mathbb{R}^{|\mathscr{C}|}, \quad P : \mathbb{R}^{|\mathscr{C}|} \to \mathbb{R}^{|\mathscr{C} \cup \mathscr{F}|}.$$

To obtain the entries for these operators, one can use methods like linear interpolation [TOS01] or direct interpolation [RS86, TOS01]. Another possible approach for choosing a coarse grid is aggregation [BMM$^+$10], where one defines a partition of the set of variables and each subset of this partition is associated with one coarse variable.

Instead of stopping at the second grid, because the matrix may still be too large, one can solve the residual equation via a two-grid approach again. By this recursive construction one obtains a multilevel (multigrid) approach, see Figure 4.1; in this case, we chose to represent a particular type of multilevel construction, V-cycle, but we could alternatively, for instance, think of W or F-cycles [TOS01]. The reason for this choice is that the core concept is easier to understand using a V-cycle. Additionally, for our particular context of interest, we verified through experiments that the difference in performance, in case we use such more complex alternative schemes, is often negligible.

We now explain this generalization in words: on the way down, at level ℓ, the method performs a certain number ν_1 of smoothing steps, using an iterative solver; the residual of the current iterate is computed and restricted by a matrix-vector multiplication with the restriction matrix for level ℓ, $S^{(\ell)} \in \mathbb{R}^{m_\ell \times m_{\ell+1}}$, where m_ℓ is the number of states at level ℓ; the operator A_ℓ is also restricted via Petrov-Galerkin to get $A_{\ell+1} = S^{(\ell)} A_\ell P^{(\ell)}, S^{(\ell)} A_\ell P^{(\ell)} \in \mathbb{R}^{m_{\ell+1} \times m_{\ell+1}}$, where $P^{(\ell)} \in \mathbb{R}^{m_{\ell+1} \times m_\ell}$ is the interpolation operator at level ℓ; then we have a recursive call where we solve the coarsest grid equation, which is a residual equation just as all equations except the one associated with the first level; then, on the way up the grids, the error is interpolated and again some smoothing iterations are applied.

4.1.2 Tensorized multigrid

In what follows, m_c represents the set of coarse variables, $m_c = |\mathscr{C}|$, associated with m, in sequence of the two-grid approach from Section 4.1.1.

As in the transition from Section 3.1.1 to Section 3.1.2, we now convert the version of an algorithm desgined for matrices, in this case Algorithm 3, so that it becomes applicable to a tensor-structured problem. The tensor structure must be preserved along the multigrid hierarchy. We follow the approach taken in [MB14] and define interpolation and restriction in the following way.

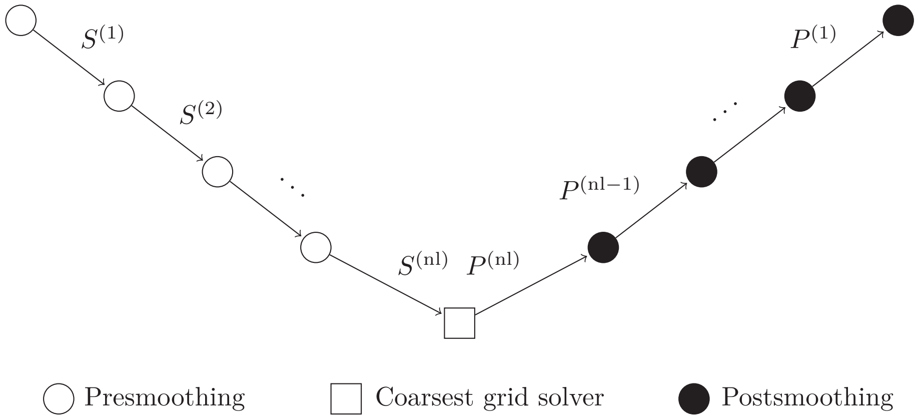


Figure 4.1: Multigrid V-cycle with nl levels: on each level, a presmoothing iteration is performed before the problem is restricted to the next coarser grid. On the smallest grid, the problem is typically solved exactly by a direct solver. When interpolating back to the finer grids, postsmoothing iterations are applied on each level.

Proposition 1. *Let* Q *of the form* (1.3) *be given, with* $E_k^{(t)} \in \mathbb{R}^{m_k \times m_k}$. *Let* $P = \bigotimes_{k=1}^d P_k$ *and* $S = \bigotimes_{k=1}^d S_k$ *with* $P_k \in \mathbb{R}^{m_k \times (m_k)_c}$ *and* $S_k \in \mathbb{R}^{(m_k)_c \times m_k}$ *where* $(m_k)_c < m_k$. *Then the Petrov-Galerkin operator corresponding to* Q^T *satisfies*

$$SQ^T P = \sum_{t=1}^{V} \bigotimes_{k=1}^{d} S_k (E_k^{(t)})^T P_k.$$

Thus, the task of constructing interpolation and restriction operators becomes a local task, i.e., each part P_k of the interpolation $P = \bigotimes_{k=1}^d P_k$ is exclusively associated with the k-th subsystem.

Another important ingredient of the multigrid method is the smoothing scheme. In our setting, it should fulfil two main requirements; it should:

(i) be applicable to non-symmetric, singular systems;

(ii) admit an efficient implementation in TT format.

Requirement (ii) basically means that only the operations that were mentioned in Section 2.2 should be used by the smoother, since most other operations are far more expensive. In this context, even though it is an atypical choice in a multigrid context because it is a non-stationary method, one logical choice is the Krylov subspace method GMRES [Saa03, SS86] (which also fulfills requirement (i)), given that it basically consists of matrix-vector products and orthogonalization steps, i.e., inner products and additions. In turn, Gauss-Seidel and Jacobi do not fulfil requirement (ii), see [MB14]. The use of GMRES as smoother is discussed, for instance, in [RVZ10], having been successfully applied to tensor structured Markov chains in [MB14].

Restriction and interpolation. As suggested in [MB14], we choose the interpolation operator P_k as direct interpolation based on the matrices describing the local part; the different L_μ, $\mu = 1, ..., d$, matrices from (1.5).

The choice for interpolation mentioned above was defined for a particular model, again the overflow queuing network that was first introduced in Section 2.2, which is a model that has a non-trivial local part in all modes. Such non-trivial part exists when there are interactions that are local, and this is the case as its structure, depicted in Figure 3.1, represents. In fact, in each queue, there are local arrivals and local departures. The corresponding matrices L_μ, $\mu = 1, ..., d$, which have been mentioned for this exact model in Section 2.2 while not explicitly written down, then have non-zero entries in the subdiagonal and in the superdiagonal, respectively. In order to understand this, we should recall from Section 3.4.2 that the state of a given subsystem, in a model with customers that are indistinguishable as this one, is simply characterized by a numerical quantity that represents the current number of customers in the corresponding queue. Therefore, the matrices associated with the operator in each queue consider a natural ordering of the states, starting from the empty queue and finishing with the full queue. For more details, see [Mac15]. As a consequence, and recalling that our matrix of interest is the generator matrix transposed, arrivals of individual customers are represented in the form of subdiagonal non-zero entries while departures lead to superdiagonal entries.

In general, however, the local part associated with a model can easily be trivial, at least in some of the modes. In fact, it is rare that the local parts of all modes are non-trivial, considering the broad benchmark collection [Mac15] as reference. Direct interpolation cannot be applied to modes that do not have such a non-trivial local part. Furthermore, we observed in practice that for models whose local part is associated with a superdiagonal matrix, which in sequence of the discussion in the previous paragraph is associated with departures for models with indistinguishable customers, applying direct interpolation also leads to problems in convergence of the corresponding multigrid cycle.

Thus, we consider, in modes with a non-trivial local part that is also not associated with a superdiagonal matrix, direct interpolation; while for all remaining cases, we consider linear interpolation. In fact, linear interpolation can always be applied, so that it is a possible backup plan. Note that linear interpolation should be particularly suitable in the presence of models with indistinguishable customers given the intrinsic 1D topology in the local transitions associated with the described natural ordering of the states – the associated matrix is tridiagonal. In turn, it is clearly not suitable for models with distinguishable customers, as described in Section 3.4.2, since the ordering of the states gets totally changed, being in particular associated with higher dimensions than one (the number of dimensions will depend on the number of types of customers that is considered in the model).

Concerning the discussion about the partition of the class of models in models with indistinguishable or distinguishable customers, it should be added that we initially referred to customers given that the models addressed in the experiments in Section 3.4 were all associated with queuing networks, so that the numerical quantity that defines the state is in fact associated with a number of customers. However, the general context of indistinguishable/distinguishable is easy to extend to any other type of model. While, as noted in [Mac15], this numerical quantity refers instead to a number of particles for chemical networks, a number of users for telecommunications, or a number of deficiencies in quality control; the separation in terms of whether the referred elements are indistinguishable or distinguishable applies exactly as for customers in queuing networks. In this context, for simplicity and in order to leave the concepts general and not restricted to a certain type of model, we start referring to indistinguishable models or distinguishable models. In particular, the comments made until this point about models with indistinguishable/distinguishable customers apply in general to indistinguishable/distinguishable models, respectively.

As for the restriction operator, we simply consider the individual matrices S_k to be the transpose of the matrices P_k, $k = 1, ..., d$. By considering tensorized restriction and interpolation operators as described in Proposition 1, the TT ranks associated to their representations in TT format have all entries equal to 1, as explained in Section 2.2. The idea is that the small matrices S_k and P_k, $k = 1, ..., d$, from their respective Kronecker representations are simply the matrices A_k, $k = 1, ..., d$, in the corresponding TT representations (2.8).

When using the multigrid scheme described in Algorithm 3, the only important property of the matrix that should be kept from one level to another is that the sum of the columns is still 0. Given the possible choices for interpolation, and corresponding choices for restriction, introduced above, it can be easily checked that this holds.

Smoother. For smoothing, we use three steps of GMRES in each grid.

Given the small number of smoothing steps, the associated requirements in storage and computation are negligible.

Normalization. In (1.1) we have the restriction that the sum of the entries of the solution is 1. This is not naturally kept during a cycle. For this reason, we normalize the obtained approximation after each cycle. Such an explicit normalization step is also done in truncated power method in the end of each iteration; recall Section 3.1.2.

As noted when describing the mentioned normalization step for truncated power method, this step is inexpensive since the inner product that is required is between the solution we obtain after the cycle and a vector of all ones, which has the simplest possible TT representation – all entries of the TT rank equal to 1.

Truncations. As in truncated power method and AMEn, algorithms proposed in the previous chapter, truncation is needed during a cycle to prevent excessive rank growth. We use again the TT-SVD algorithm, which truncates the tensor back to lower entries of the TT rank, within a specified tolerance, as described in Section 2.2.1. Once again, it is performed after each step where the TT rank entries are allowed to increase.

In particular, truncation has to be performed after lines 6 and 8 of Algorithm 3. Concerning the truncation of the restricted residual in line 6, we have observed that we do not need a very strict accuracy to obtain convergence of the global scheme, so that we set the target accuracy to 10^{-1}. As for the truncation of the updated iterates v_l after line 8, we note that they have very different norms on the different levels, so that the target accuracy of the truncation should depend on the level. Additionally, a dependency on the cycle, following the idea introduced in the context of truncated power method, in Section 3.1.2, which was then also used in AMEn, in which such an adaptive scheme is applied to the different iterations, is also included. Precisely, the accuracy depends on the residual norm after the previous cycle (which can be seen as the measure of an iteration for such multigrid schemes). Summarizing, the target accuracy of the truncation of the different v_l is the norm of v_l divided by v_1 (dependency on the level), times the residual norm after the previous cycle (dependency on the quality of the current approximate solution), times the value 10. This double adaptivity is also used within the GMRES smoother, again in the steps after which the TT rank entries may increase. Note that the adaptivity in terms of the residual norm after the previous cycle can be motivated just as the adaptive scheme associated with truncated power method, in Section 3.1.2, was motivated – we should be more and more accurate as we approach an approximate solution with the desired accuracy.

We also impose an upper bound on the TT rank entries that are allowed after each truncation. This bound is initially set to 15 and grows by a factor of $\sqrt{2}$ after each cycle for which the new residual norm is larger than $\frac{9}{10}$ times the residual norm obtained considering the solution from the previous cycle, signalling stagnation. This differs from the value $\frac{3}{4}$ proposed in [MB14], but it seems more suitable; in fact, our experiments demonstrated that, using the value $\frac{3}{4}$, the TT rank entries might easily still grow more than what is needed.

Coarsest grid solver. The direct solver that is considered for solving the coarsest grid problem is the Moore-Penrose pseudoinverse.

Initial approximation of the solution. The algorithm is initialized with the tensor that results from solving the coarsest grid problem and then bringing it up to the finest level using interpolation.

Allowing the number of levels to depend on the mode. We allow the possibility that a different number of levels for different modes is considered. For a certain level, if there are modes for which we do not want to restrict further, we simply set the corresponding core to identity. Note that this is possible due to the local nature of the operators for restriction and interpolation, in the sense that there is one of each per mode.

This is particularly useful if the initial mode sizes are different since the optimal sizes to consider in the coarsest grid are expected to be the same for all modes.

4.2 Multigrid-AMEn

We have introduced a tensorized multigrid scheme (Section 4.1) and AMEn (Section 3.3) for solving (1.1) considering that the generator matrix has the tensor structure (1.3). They are expected to perform well but they both have limitations that are in turn expected to lead to not so good performances in certain contexts. We first go through such limitations and then describe a novel combination that potentially overcomes these limitations.

4.2.1 Limitation of AMEn

The limitations of AMEn were extensively discussed in the previous chapter already, so that we just sum them up now.

The cost of its subproblems becomes prohibitively large when a direct solver is used for solving the subproblem. This was discussed in Section 3.2 and confirmed in the experiments in Section 3.4 when demanding a relative accuracy of the solution that led to large TT rank entries.

In turn, the alternative use of an iterative solver is problematic given that the matrix of the subproblem easily becomes ill-conditioned which, despite the fact that a good initial guess can be considered when MINRES is the chosen solver as noted in Section 3.2, can lead to bad convergence of the algorithm, as verified again in Section 3.4, more concretely in Table 3.1. Also in sequence of this table, it was noted that the cause for the ill-conditioning is associated with the mode sizes.

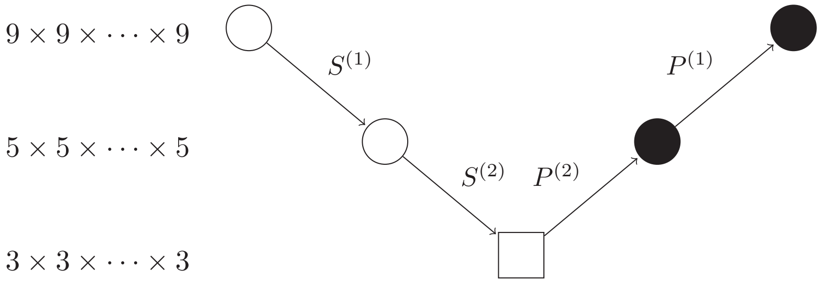

$9 \times 9 \times \cdots \times 9$
$5 \times 5 \times \cdots \times 5$
$3 \times 3 \times \cdots \times 3$
$S^{(1)}$
$S^{(2)}$
$P^{(2)}$
$P^{(1)}$

Figure 4.2: Coarsening process for a problem with mode sizes 9.

4.2.2 Limitations of the tensorized multigrid scheme

The described tensorized multigrid method is limited to modest values of d, simply because of the need for solving the problem on the coarsest grid. The size of this problem grows exponentially in d, so that it is still affected by the curse of dimensionality. In fact, only the mode sizes are reduced from one level to the next, while the value of d remains unchanged. Figure 4.2 illustrates the coarsening process if one applies full coarsening to each $(E_j^{(t)})^T$ according to the possible operators of interpolation and restriction discussed in Section 4.1.2, assuming a capacity of 8, associated with mode sizes 9. In the case of three levels, a problem of size 3^d would need to be addressed by a direct solver on the coarsest grid. While such constructions would not allow that we coarse the problem to a single variable in each dimension, given that the mode sizes must be of the form $2^q + 1$, $q = 0, 1, ...$, it would still be possible to reduce the mode sizes further to the value 2.

4.2.3 Combination of the two methods

Instead of using a direct method for solving the coarsest grid problem, as generally done in the context of multigrid methods as noted in Figure 4.1, and as done in particular in the tensorized multigrid method discussed in Section 4.1.2, we propose the use of AMEn. We expect that it becomes much simpler to solve the subproblems (3.9) within AMEn, in particular since this should allow the iterative solver MINRES to be used effectively, in sequence of the fact that, as noted in the previous chapter and reminded in Section 4.2.1, the problems on MINRES only occur if the mode sizes are allowed to increase. In fact, using this new algorithm, we can force them to be as small as needed. Moreover, in a context where such drawback of AMEn is avoided, AMEn should be very hard to beat as concluded in the experiments in Section 3.4 for problem sizes that still allow an effective application of a direct solver on the subproblems.

Note that the problem to be solved on the coarsest grid constitutes a correction (or residual, as already called) equation, thus differing from the original problem (1.1) in having a nonzero right-hand side and incorporating a different linear constraint. To address this problem, we apply the version of AMEn that was proposed in [DS14], designed for addressing linear systems, instead of the version that is proposed in Section 3.3, to the normal equations and ignore the linear constraint. The linear constraint is fixed only at the end of the cycle by explicitly normalizing the obtained approximation, as done in [MB14] and already described in Section 4.1.2.

We now focus on some particularities that characterize our approach, focusing only on those that are related with the coarsest grid problem given that those associated with the main core of the algorithm, the tensorized multigrid scheme, are just as those already described in Section 4.1.2. More concretely, the two algorithms share the details about: the choice of the restriction and interpolation operators, the smoother, the normalization step, the truncations, and the way to allow that the number of levels is set to different values for different modes.

Parameters of AMEn in the coarsest grid problem. AMEn targets an accuracy that is at the level of the residual norm after the previous multigrid cycle. More concretely, we stop AMEn once this value is reached. The motivation is the same as when we defined the target accuracies of the truncations in truncated power method, AMEn, and then also in the tensorized multigrid that we use here as main solver, recall Section 4.1, in an adaptive way.

The enrichment rank, rank of the approximation of the negative gradient that is then used to augment the cores, is 3, as in our AMEn algorithm; recall Section 3.3. This approximation is obtained by ALS as suggested in [DS14].

A crucial reason for the success of the method that is proposed in this chapter is, as already noted, the fact that the difficulties in solving the subproblems inside AMEn using an iterative solver are associated with the mode sizes. In fact, since the tensorized multigrid scheme that is considered allows reducing the mode sizes, the coarsest grid then becomes perfect for the application of AMEn. In this context, MINRES is used on the subproblems of AMEn. To be more precise, given that the direct solver can still be efficient while the problem size is small, we set a threshold on the value 1000, applying a direct solver for problems with sizes up to this value, using MINRES otherwise.

Size of the coarsest grid problem. By construction of restriction and interpolation, the mode sizes in the coarsest grid can only be of the form $2^q + 1$, $q = 0, 1, ...$, as already noted. In sequence of the study that is mentioned in Section 3.4.1, even if not explicitly shown, concerning the idea that a possible ill-conditioning of the matrix in the coarsest grid will depend on the mode sizes, we verified that mode sizes 5 might still lead to too ill-conditioned subproblems (in the mentioned experiments, we only showed this explicitly for mode sizes 11). This emphasizes even further how there is almost no margin for increasing the mode sizes so that the application of such a method as this tensorized multigrid for reducing the mode sizes is really needed in order for an iterative solver to be possible to effectively apply in the subproblems of AMEn.

Thus, the number of levels is chosen such that the coarsest grid problem has mode sizes 3, as in the example that is represented in Figure 4.2.

Initial approximation of the solution. The approach for defining the initial approximation of the solution is the same as in Section 4.1.2 – the tensor that results from solving the coarsest grid problem is brought up to the finest level using interpolation. However, the coarsest grid problem should now be solved differently. In fact, we again need to avoid the curse of dimensionality that is the main drawback of the original tensorized multigrid scheme. A good solution is, in the context of the main scheme of this new algorithm, to use AMEn; considering its variant proposed in this thesis.

4.3 Numerical experiments

We now illustrate the efficiency of the newly proposed algorithm.

4.3.1 Model problems

The benchmark problems that are used are all taken, as already noted, from the benchmark collection [Mac15]. In total, we consider six different models, which can be grouped into three categories. The considered parameters are the natural generalizations of the default ones that are considered in the mentioned benchmark collection.

All considered models are indistinguishable models as we have explained in Section 4.1.2 that the construction associated with interpolation (and restriction) does not suit models associated with a local topology that is not 1D, which is the case for distinguishable models.

Overflow queuing models. The first class of benchmark models includes an already extensively discussed overflow queuing network; recall, for instance, Section 3.4.1, where its structure is depicted in Figure 3.1. Additionally, two variations of it are considered. The arrival rates, the service rates and the capacity depend on the queue. We consider, for the arrival rate of the k-th queue, $1.2 - (k-1) \cdot 0.1$, for $k = 1, \ldots, d$; while we set all service rates to 1. The variations of the model differ in the way the queues interact. For the already introduced `overflow`, the following description has been provided in Section 3.4.1: customers which arrive at a full queue try to enter subsequent queues until they find one that is not full; after trying the last queue, they leave the system. As for its two variations that are also considered:

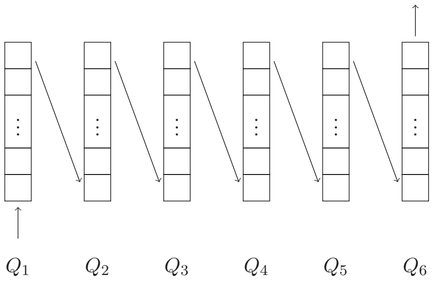


Figure 4.3: Structure of the model kanbanalt2.

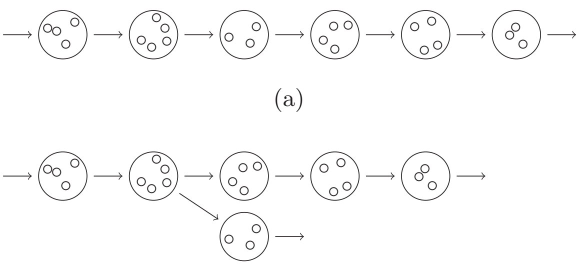
- overflowsim: As overflow, but customers arriving at a full queue try only one subsequent queue before leaving the system;
- overflowpersim: As overflowsim, but if customers arrive at the last queue and this queue is full, they try to enter the first queue instead of immediately leaving the system.

For these models, it is possible to consider direct interpolation based on the local part of the operator, as already noted for the model overflow when discussing such operators in Section 4.1.2. This extends to the other two models because their local parts are common.

Simple tandem queuing network (kanbanalt2). A network of queues has to be passed through by customers, one after the other. Each queue has its own service rate and its own capacity. We set the service rate of all queues to 1, while arrivals only occur at the first queue with a rate of 1.2. The service in queue k can only be finished if queue $k+1$ is not full, so that a served customer can immediately enter the next queue. Figure 4.3 illustrates this model.

The only subsystems with a non-trivial local part are the first and the last, but the last is associated with a superdiagonal matrix given that, as seen in Figure 4.3, it corresponds to a departure; recall the discussion concerning the association between the type of transitions and the type of non-zero entries that occur in the matrices that represent such transitions for indistinguishable models in Section 4.1.2. In the end, we can only apply direct interpolation based on the local matrix associated with the first mode, L_1 from (1.5). As a result, P_1 is constructed via direct interpolation, while linear interpolation is considered in $P_2, \ldots, P_d$.

Metabolic pathways. The next model problems we consider come from the field of chemistry, describing stochastic fluctuations in metabolic pathways. In Figure 4.4(a)



(a)

(b)

Figure 4.4: Structure of the models directedmetab (a) and divergingmetab (b).

each node of the given graph describes a metabolite. A flux of substrates can move along the nodes being converted by means of several chemical reactions (an edge between node k and ℓ in the graph means that the product of reaction k can be converted further by reaction ℓ). The rate at which the k-th reaction happens is given by

$$\frac{v_k m_k}{m_k + K_k - 1},$$

where m_k is the number of particles of the k-th substrate, while v_k and K_k are constants which we choose as 0.1 and 1000, respectively, for all $k = 1, \ldots, d$. Each substrate has a capacity, and such capacities can differ from one substrate to another. This model will be called directedmetab.

Model divergingmetab is a variation of directedmetab. In this case, one of the metabolites in the reaction network can be converted into two different metabolites, meaning that the reaction path splits into two independent ones, as shown in Figure 4.4(b).

The interpolation and restriction operators for these models are chosen in the same way as for kanbanalt2 given that the local transitions that exist in the different modes are of the same type.

4.3.2 Numerical results

We next report the results of the experiments we performed on the models from Section 4.3.1 to test the proposed method, "Multigrid-AMEn". We compare its performance against: algorithm "AMEn", proposed in Section 3.3; and also against the original tensorized multigrid scheme proposed in [MB14], and presented in this thesis in Section 4.1.2, which we simply denote by "Multigrid".

Throughout all experiments, we stop an iteration when the relative accuracy, defined in Section 3.4 as the current residual norm divided by the residual norm when considering the tensor of all ones (scaled so that the sum of its entries is one), is smaller than 10^{-2}.

Table 4.1: Execution time (in seconds), number of iterations and maximal TT rank entry of the computed approximations for overflow with mode sizes 17 and varying the number of subsystems d. The symbol “—” indicates that the desired accuracy could not be reached within 3600 seconds.

	AMEn			Multigrid			Multigrid-AMEn		
d	Time	Iter	Rank	Time	Iter	Rank	Time	Iter	Rank
4	4.5	7	16	4.6	13	13	4.2	13	13
5	36.3	9	23	6.4	11	20	7.0	11	20
6	239.4	12	28	24.7	17	29	20.4	17	29
7	1758.4	14	36	252.4	24	29	38.3	24	29
8	—	—	—	—	—	—	98.4	28	41
9	—	—	—	—	—	—	214.8	36	57
10	—	—	—	—	—	—	718.8	40	80
11	—	—	—	—	—	—	2212.2	45	113

Such normalized tensor of all ones happens to be our initial guess considered by AMEn, as noted in Section 3.4, but it does not correspond to the initial guesses of Multigrid and Multigrid-AMEn.

In the tables that follow, in coherence with the experiments from Section 3.4: “Time” stands for the computation time, in seconds; “Rank” stands for the maximal entry of the TT rank of the approximate solution. As for “Iter”, it will stand for the required number of iterations, which concerns the number of sweeps for AMEn while the number of cycles for the two multigrid algorithms. A limit on the allowed computation time of 3600 seconds (one hour) is imposed.

Scaling with respect to the number of subsystems. In order to illustrate the scaling behaviour of the three methods, we first choose in all models a capacity of 16 in each subsystem; i.e., mode sizes 17; which is again the default choice in [Mac15]; and vary d, the number of subsystems. Figure 4.5 displays the obtained execution times.

To provide more insight into the results depicted in Figure 4.5, we also give the required number of iterations and the maximal TT rank entry of the computed approximation for the overflow model in Table 4.1. For the other models, the observed behaviour is similar and we therefore refrain from providing more detailed data.

In Figure 4.5, we observe that Multigrid and Multigrid-AMEn behave about the same up to $d = 6$ subsystems. For larger d, the cost of solving the coarsest grid problem of size 3^d with a direct method becomes prohibitively large within Multigrid. Multigrid-AMEn is almost always faster than AMEn even for $d = 4$ or $d = 5$. To which extent Multigrid-AMEn is faster depends on the growth of the TT rank entries of the solution with respect to d, since these have the largest influence on the performance of AMEn; recall the detailed discussion in Chapter3.2 about the cost of AMEn when applying a direct solver in the subproblems.

In the context of Remark 1, the TT format is a degenerate tree tensor network, thus matching the topology of interactions in the models overflowsim, kanbanalt2, and directedmetab; recall the figures that represent these models in Section 4.3.1. The influence of the mentioned topology is expected to be mostly felt in the sense that a good approximation of the solution should be possible to obtain with a TT tensor with small TT rank entries, as already noted in the context of the small TT rank entries obtained in the models tested in Section 3.4, specially in the model associated with Table 3.2 which is particularly suitable to the format, just as the models mentioned above. This strongly influences the computation times, in particular when applying AMEn, as emphasized in the previous paragraph, or as for instance noted in the context of Figure 3.3. Note that another factor that helps justifying the good performances of the algorithms is the fact that the TT ranks associated with the operators TT format that represent the models are small and independent of d, just as for the models tested in Section 3.4; for details about the operators, check [Mac15]. Comparing the mentioned models (overflowsim, kanbanalt2, and directedmetab) to overflowsim, the performance is slightly worse for kanbanalt2 and directedmetab. This should be caused by the fact that linear interpolation must replace direct interpolation in most modes (all except the first) for the latter two models as noted in Section 4.3.1. This suggests that the solution of considering linear interpolation is not as efficient as we would expect, recall the discussion about the suitability of this interpolation operator for indistinguishable models in Section 4.2.3. In contrast, overflowsim, as well as overflow and overflowpersim, consider direct interpolation in all modes, which clearly seems to be a very efficient choice. This seems to in fact be a relevant factor, noting that the second best performance is observed for overflowpersim even though its underlying topology contains a cycle – the topology can be represented as in Figure 3.1 but adding also an arrow from the last to the first queue – thus not matching the TT format. It becomes clear that there is robustness with respect to the topology, as in fact the performances are not that different for topologies that were expected to be completely out of context when considering TT format. This is also reflected by the good results obtained for divergingmetab; see the underlying topology from Figure 4.4(b).

We see in Table 4.1 that the upper bound that is considered on the TT rank entries in each truncation of the multigrid approaches; described in Section 4.1.2; works particularly well, given that we obtain similar maximal TT rank entries to those obtained in AMEn, whose rank adaptivity procedure is more natural.

The maximum problem size that is considered is $17^{13} \approx 9.9 \times 10^{15}$. Multigrid-AMEn easily deals with larger d, but this is the largest configuration for which execution times below 3600 seconds can still be obtained.

Table 4.2: Execution time (in seconds), number of iterations and maximal TT rank entry of the computed approximations for overflow with $d = 6$ and varying mode sizes. The symbol "—" indicates that the desired accuracy could not be reached within 3600 seconds.

	AMEn			Multigrid			Multigrid-AMEn		
n	Time	Iter	Rank	Time	Iter	Rank	Time	Iter	Rank
5	0.7	4	13	5.9	8	15	6.2	8	15
9	3.8	6	19	6.1	8	15	3.9	8	15
17	239.4	12	28	24.8	17	29	19.5	17	29
33	—	—	—	102.9	17	41	104.6	17	41
65	—	—	—	882.1	20	57	904.1	20	57

Scaling with respect to the mode sizes. To also illustrate how the methods scale with respect to increasing mode sizes, we next perform experiments where we fix, for all models, $d = 6$ subsystems, which is again the default choice in [Mac15] for most of the involved models, and vary their capacity. The execution times are presented in Figure 4.6, while more detailed information about the model overflow is provided in Table 4.2. The capacity is considered to be the same for all subsystems, implying that the mode sizes are also all the same, so that for simplicity we can state them as scalar values n; in particular in Table 4.2; which represent such common values.

Figure 4.6 shows that AMEn outperforms the two multigrid methods (except for kanbanalt2) for small mode sizes. Depending on the model, the multigrid algorithms start to be faster for mode sizes 9 or 17, since the subproblems to be solved in AMEn become too expensive at this point, given again the fact that a direct solver is used. In fact, the bad performance of AMEn for kanbanalt2 can be explained by the fact that the steady state distribution of this model has rather large TT rank entries already for small mode sizes. This again emphasizes the idea that the quality of the performance is not only a function of how suitable the underlying topology is to the format given that this model has, in theory, a perfect topology.

Concerning the comparison between the two multigrid methods, no significant difference is visible in Figure 4.6. We have already seen in Figure 4.5 that $d = 6$ is not large enough to let the coarsest grid problem solver dominate the computational time in Multigrid. As a consequence, Figure 4.6 nicely confirms that using AMEn for solving the coarsest grid problem also does not have an adverse effect on the convergence of multigrid, in which case the times obtained for Multigrid-AMEn would be worse than those of Multigrid.

The already mentioned robustness that concerns the idea that the format seems to be able to deal even with less suitable topologies is again confirmed. In fact, the scaling behaviour observed in Figure 4.6 is very similar for all models.

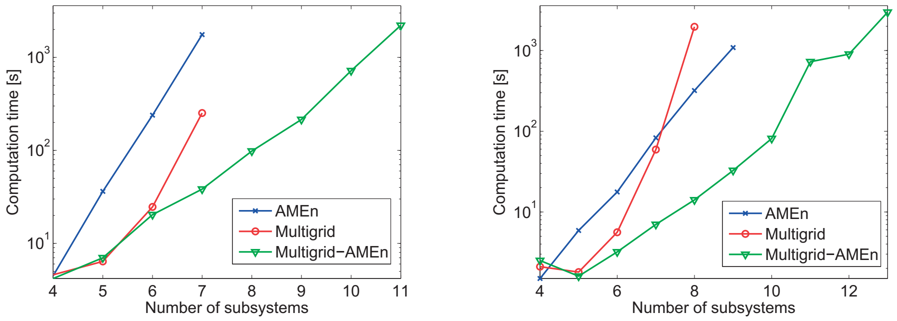
We again verify that the TT rank entries remain in control in the multigrid schemes by looking at the results in Table 4.2.

The maximum problem size that is considered is $129^6 \approx 4.6 \times 10^{12}$. Additionally, once again, no problem should arise from going further on the mode sizes; the fact that we stop is simply because the computations would start taking longer than 3600 seconds.

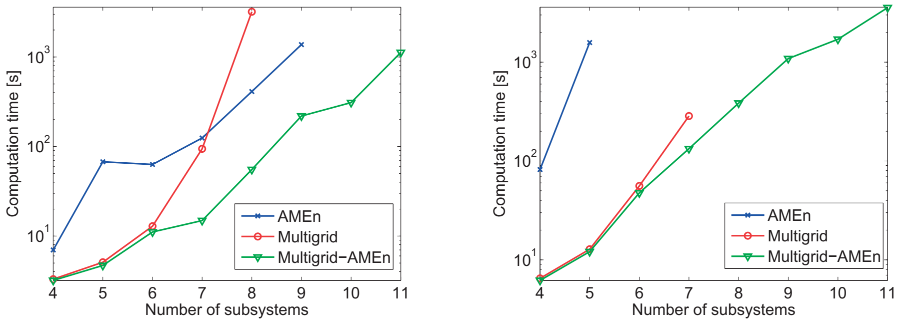
4.4 Conclusion

We have proposed a novel combination of two methods, AMEn and an existing tensorized multigrid scheme, for computing the stationary distribution of large-scale tensor structured Markov chains. Our numerical experiments confirm that this combination truly combines the advantages of both methods. In particular, it addresses the main difficulty of AMEn that it cannot deal with mode sizes that are not extremely small by only applying this method after they have been reduced on the way down the grids. As a result, we can address a much wider range of problems not only in terms of the number of subsystems but also in terms of the number of possible states per subsystem. Our experiments also demonstrate the robustness of TT format in the sense that it is capable of dealing with a larger variety of applications and topologies, compared to what has been reported in the literature, including Section 3.4.

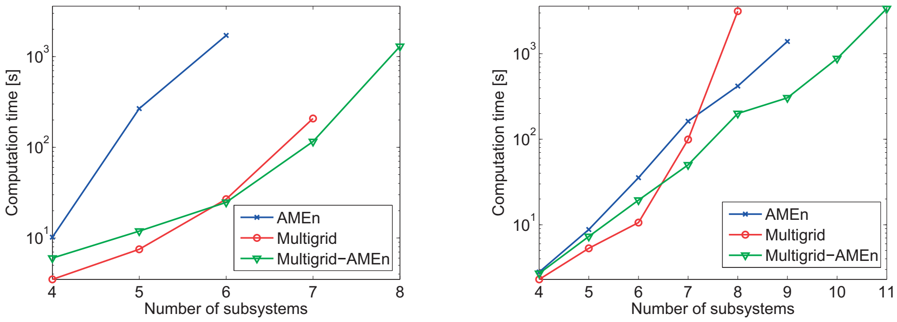
Further improvement is required in the choice of the restriction and interpolation operators. In fact, while applying direct interpolation on the local part of the operator representing the generator matrix seems to be extremely efficient, it is rare that all modes have a non-trivial local part that allows such procedure to be applied. In such cases, linear interpolation must be used instead, and we verified in the numerical experiments that the algorithms are then significantly less efficient. Furthermore, this choice for interpolation is here applied to indistinguishable models, associated with a local 1D topology, but it would not suit distinguishable models, so that robustness still fails in the sense that this type of model cannot be addressed.



(a) overflow (b) overflowsim

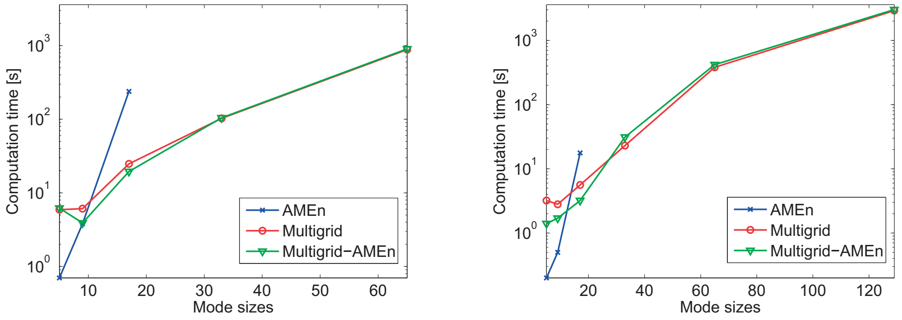


(c) overflowpersim (d) kanbanalt2

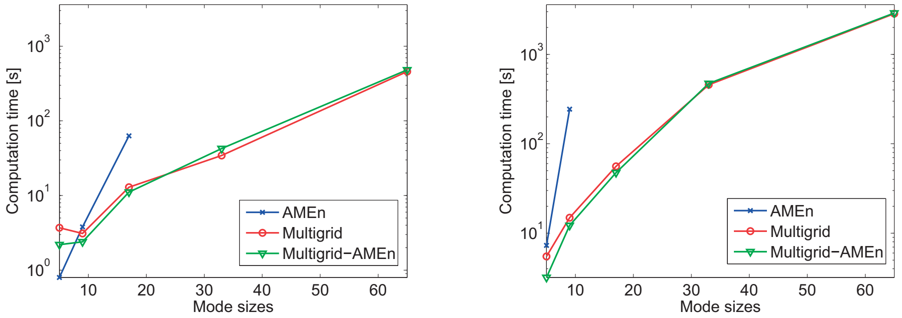


(e) directedmetab (f) divergingmetab

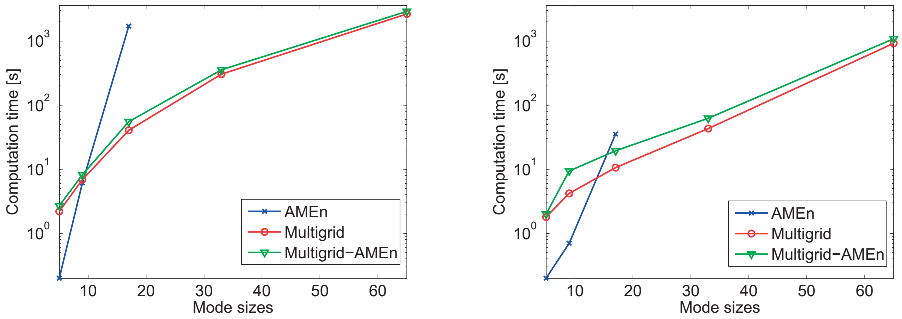
Figure 4.5: Execution time (in seconds) needed to compute an approximation of the steady state distribution for the benchmark models from Chapter 4.3.1. All mode sizes are set to 17.



(a) overflow (b) overflowsim



(c) overflowpersim (d) kanbanalt2



(e) directedmetab (f) divergingmetab

Figure 4.6: Execution time (in seconds) needed to compute an approximation of the steady state distribution for the benchmark models from Chapter 4.3.1. All cases consider $d = 6$.

5 Aggregation/disaggregation techniques

In this chapter, we consider a specific subclass of multigrid schemes for solving (1.1). The difference to the approach that was considered in the previous chapter is that restriction and interpolation are done in a very particular way, which is well-established in the solution of our problem of interest of finding steady states of Markov chains. Our main concern was to, in sequence of the conclusions of the previous chapter, try to find operators for restriction and interpolation that are more efficient for indistinguishable models while also distinguishable models to be effectively addressed.

Aggregation/disaggregation techniques, which can be seen as a subclass of multigrid methods, have also been used to solve (1.1). A reduced version of (1.1) where states have been aggregated in groups is solved and then a solution for the original problem needs to be extrapolated using some disaggregation scheme. Thus, these techniques are associated with a certain way to choose the restriction and interpolation operators. Aggregation, already mentioned in this context in Section 4.1.1, can be defined, for instance: assuming an underlying Kronecker structure of the generator matrix [HL94, BD07a, PM11, Day12]; or imposing specific relations between the rates of transition between states, entries of the generator matrices, of the original and the aggregated processes – lumpability [DHS03, HMRP13].

The approaches proposed in this chapter consist of using aggregation/disaggregation techniques that are particularly convenient to adapt to TT format. In particular, as in the algorithm proposed in the previous chapter, restriction (aggregation) and interpolation (disaggregation) are again tensorized. Two types of operators for aggregation, which give two algorithms, are considered: one based on a tensorized aggregation scheme [HL94] that is particularly similar to that proposed in the previous chapter; and another where full subsystems are aggregated [BD04, BD07b, Day12].

The goal is again to target the computation of the stationary distribution for finite-dimensional communicating Markov processes, developing and comparing different algorithmic approaches in TT format, testing their sensitivity to different models by using a broad benchmark collection for testing. Such tests will be also used to emphasize even more that building algorithms based on TT format is generally a good option.

The content of this chapter is based on the paper [Mac16].

5.1 Aggregation/disaggregation schemes

Having already described the core ideas behind multigrid; recall Section 4.1.1; we directly move to the introduction of their subclass that considers a particular type of restriction and interpolation, called aggregation and disaggregation, respectively. We then describe particular variants that are expected to be effective for Markov chains characterized by interacting subsystems.

The original idea can be found in [Tak75], in an algorithm called *iterative aggregation/disaggregation* (IAD), where only two levels/grids were used. The number of states in the coarse level is typically much smaller than the original number of states given that many states are merged (aggregated) into one. An important drawback is that a big amount of information then gets lost. This is a limitation of methods with two levels since the coarse grid problem has to be small enough to be solved efficiently, thus implying such excessive merging. This is the same idea that was briefly mentioned in Section 4.1.1 for justifying that considering a two-grid approach might be problematic.

The general concept of multigrid, with more than two levels being considered, started to be later combined with the core idea of IAD, of aggregating and disaggregating states. Many different possibilities for the way to define aggregation were proposed, where some of them focus on the case in which there is an underlying Kronecker structure of the generator matrix, as already noted in the beginning of this chapter. This is our case of interest. We next go into detail on such variants. The idea is that the chosen aggregation (restriction) operator has a simple Kronecker representation. This is the case when assuming the existence of subsystems inside the network.

5.1.1 Tensorized algorithm pairing states in each subsystem

This first scheme was proposed in the numerical experiments of [HL94]. A Petrov-Galerkin construction, recall Proposition 1, is again considered, meaning that restriction and interpolation are again tensorized. More concretely, as in the algorithm proposed in [MB14] and described in Section 4.1.2 of this thesis, restriction (aggregation) results in a reduction of the number of states of each subsystem by merging the states associated with consecutive pairs of numbers. As a result, this method should be particularly suitable for networks for which the states are naturally ordered in each subsystem according to the numerical quantity they represent, as described for indistinguishable models in Section 4.1.2. This is the same local topology that should suit the tensorized multigrid

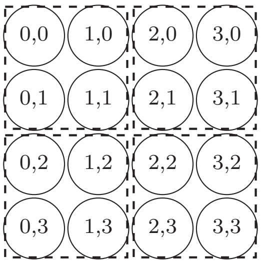


Figure 5.1: Example of how aggregation works for $d = 2$ and 4 possible states per subsystem.

scheme proposed in the previous chapter also, in particular when linear interpolation is considered, recall again Section 4.1.2.

The matrix S_k, in the context of Proposition 1, associated with each subsystem, $k = 1, ..., d$, is, for an example where the number of states of each subsystem is 4,

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^T . \tag{5.1}$$

In particular, defining, as in Section 4.1.1, m_ℓ as the number of states at level ℓ, $m_{\ell+1} = m_\ell/2^d$ as the number of states per subsystem is divided by 2.

The way states are aggregated is represented in Figure 5.1, for an example with $d = 2$ and again assuming 4 states per subsystem.

The suitability of a setting with an underlying ordering of the states in each subsystem associated with a 1D topology is clear in Figure 5.1.

5.1.2 Aggregating all states from fixed subsystems

Aggregation can be also done, again assuming the existence of interacting subsystems, through aggregating all states of one particular subsystem in each level [BD04, BD07b, Day12].

Once again, restriction (aggregation) is tensorized, meaning that it is represented by a simple Kronecker product – it follows a Petrov-Galerkin construction, recall again Proposition 1. The difference is that all matrices S_k, $k = 1, ..., d$, are now identity matrices except for the one associated with the subsystem whose states are aggregated. For that particular dimension, the matrix is a column vector of ones instead. Therefore, this is a tensorized scheme that has the particularity that only one of the small matrices that compose it is not identity. Note that $m_{\ell+1} = m_\ell/n_i$, where i is the mode associated with the subsystem whose states are aggregated from level ℓ to $\ell + 1$.

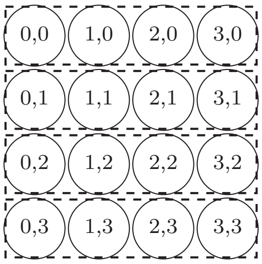


Figure 5.2: Example of how aggregation works for $d = 2$ and 4 possible states per subsystem, assuming that the first dimension is the one associated with the subsystem whose states are aggregated.

The way states are aggregated is represented in Figure 5.2, again for an example with $d = 2$ and 4 states per subsystem.

Two states are aggregated from one level to the next if they are in the same state in a subset of the subsystems that includes all except one – the one whose states are aggregated in that level. In particular, any local topology is now ignored. In Figure 5.2, the subsystem whose state is not relevant for defining the states that are aggregated is the first.

Figure 5.2 clearly suggests how, when the mode sizes start to increase, the distance between states that are aggregated together increases, in case the local topology is associated with naturally ordered states – 1D topology. This means, in particular, recalling the discussion in Section 4.1.2 concerning the fact that such a 1D topology is intrinsic to modes associated with such indistinguishable models, that this algorithm is not expected to behave particularly well for this type of model. In particular, problems in the convergence of this algorithm are expected when, given a model of this type, the mode sizes start to increase.

5.2 Proposed algorithms

As done, for instance, in the transition from Section 3.1.1 to Section 3.1.2, the idea is to now convert the particular variants of aggregation/disaggregation techniques for matrices described in Section 5.1 into algorithms in TT format. In particular, as noted in the context of the construction of the tensorized multigrid scheme proposed in Section 4.1.2, the tensor structure must be preserved along the multigrid hierarchy.

5.2.1 Algorithm from Section 5.1.1 in TT format

Our first proposed algorithm combines the algorithm described in Section 5.1.1 with TT format by having all involved structures, in particular matrices and vectors, in this format.

As noted in Section 5.1.1, this new algorithm is a similar tensorized multigrid scheme to the one proposed in Section 4.1.2 (both reducing the size of the different modes in each grid considering an underlying 1D local topology). This should become even more clear now that we explicitly discuss the adaptation of the algorithm to TT format. Furthermore, they are similar in the fact that they both aim at reducing the mode sizes in each grid. As a consequence, it is logical that the particularities of the tensorized multigrid scheme that we consider are very similar to those that were considered for the other mentioned scheme. We now go through such particularities, and the mentioned similar algorithm is referred to as algorithm of reference.

Restriction and interpolation. Because the corresponding aggregation is represented in a tensorized way, as seen in Section 5.1.1, it allows a TT representation with all entries of the TT rank equal to 1. The same holds for disaggregation. A similar statement was made for the restriction and interpolation operators associated with the algorithm of reference. The idea is that the small matrices involved in the Kronecker representation are simply the matrices A_μ, $\mu = 1, ..., d$, in the corresponding TT representation (2.8). The difference is that such matrices are now of the form (5.1) (adapted to the mode sizes).

As noted in the context of the algorithm of reference, when using the multigrid scheme described in Algorithm 3, the only important property of the matrix that should be kept from one level to another is that the sum of the columns is 0. This holds independently of how interpolation is chosen as, assuming that the sum of the columns at level ℓ is 0, $\mathbf{1}^T A_{\ell+1} = \mathbf{1}^T (S^{(\ell)} A_\ell P^{(\ell)}) = (\mathbf{1}^T S^{(\ell)}) A_\ell P^{(\ell)} = \mathbf{1}^T A_\ell P^{(\ell)} = \mathbf{0}$ (where $\mathbf{0}$ denotes, again, a vector of zeros) using the fact that $\mathbf{1}^T S^{(\ell)} = \mathbf{1}^T$ – the sum of the columns of the restriction operator S_ℓ is 1. In this context, we use the transpose of restriction for interpolation, which corresponds to considering, as in the context of the corresponding operators for the algorithm of reference, each P_k, $k = 1, ..., d$, to be the transpose of S_k. The reason was explained in when justifying in Section 1.1 how simple it is to obtain a representation of Q^T of the form (1.3) given the same type of representation for Q.

Smoother. The fulfilments that must be verified by the smoother are exactly the same as in the algorithm of reference. In this context, we choose again GMRES.

We use three steps of GMRES in the finest grid while one step in the remaining grids. In fact, while we used three steps in all other grids for the algorithm of reference, in this case we verified that the extra cost associated with such additional steps was not worth the corresponding gain in convergence. Note that the separation is done depending on the level and not on whether we are in a presmoothing or postsmoothing stage, as typically done.

Given that we have even less smoothing steps than in the algorithm of reference, the associated requirements in storage and computation are now even more negligible.

Coarsest grid solver. The coarsest grid is still affected by the curse of dimensionality. In fact, just as in the algorithm of reference, the mode sizes are reduced from one level to the next while the number of modes remains unchanged. In this context, the same solution that was adopted in the algorithm proposed in this thesis to solve this problem from the algorithm of reference, recall Section 4.2.3, can be adopted: we use AMEn [DS14] as coarsest grid solver.

Normalization. In (1.1) we have the restriction that the sum of the entries of the solution is 1. This is not naturally kept during a cycle, so that, again as in the algorithm of reference, we normalize the obtained approximation after each cycle.

Truncations. Truncation is again needed during a cycle to prevent excessive rank growth. TT-SVD algorithm is again used as in the algorithm of reference, being applied in the exact same steps of Algorithm 3.

In particular, truncation is again done after lines 6 and 8 of Algorithm 3. As for the parameters that are considered in these truncations, the restricted residual in line 6 is again truncated with constant accuracy 10^{-1}. As for the truncation of the updated iterates v_ℓ after line 8, it is now less complicated since the dependency of their norms on the level is not as strong as in the algorithm of reference, so that we do target an accuracy that depends on the level. The only adaptive scheme that is considered is related to the residual norm after the previous cycle. In fact, the target accuracy is that value times a constant, 10. This is again also the accuracy that is considered for the truncations inside the GMRES smoother.

The same upper bound on the TT rank entries that are allowed after each truncation as in the algorithm of reference are imposed: it is initially set to 15 and grows by a factor of $\sqrt{2}$ after each cycle for which the new residual norm is larger than $\frac{9}{10}$ times the residual norm obtained considering the solution from the previous cycle.

Parameters of AMEn in the coarsest grid problem. For the same reason that the particularities of this algorithm are in general similar to those from the algorithm of reference, it makes sense that the parameters associated with the application of AMEn in the coarsest grid are similar to those of Multigrid-AMEn; recall Section 4.2. In fact, the parameters are the same: AMEn targets an accuracy that is the residual norm after the previous multigrid cycle; the enrichment rank is 3, and the approximation of the associated residual is obtained by ALS as suggested in [DS14]; the subproblems are solved with a direct solver for problems of size up to 1000, while MINRES is used otherwise.

Size of the coarsest grid problem. By construction of restriction and interpolation, the mode sizes in the coarsest grid can only be powers of 2 (except $2^0 = 1$ since it is not possible to reduce the problem to a single variable per mode, which was also the case for the algorithm of reference, even if for different reasons, recalling that the operators for restriction and interpolation are different). In sequence of the comments concerning the size of the coarsest grid problem for Multigrid-AMEn; recall Section 4.2; it was noted that mode sizes 5 would still be too large while mode sizes 3 are already small enough for AMEn to be possible to apply effectively. The question is whether mode sizes 4 are still possible to use or we need to reduce the mode sizes to the value 2. With a similar type of study that led to the mentioned conclusions for Multigrid-AMEn, we concluded that mode sizes 4 would still be too large.

Thus, the number of levels is chosen such that the coarsest grid problem has mode sizes 2, which is the minimum possible value.

Initial approximation of the solution. The algorithm is initialized with the tensor that results from solving the coarsest grid problem, which is then brought up to the finest level using interpolation, and such problem suffers from the curse of dimensionality so that we cannot apply a direct solver, just as for the algorithm of reference. In this context, as in Multigrid-AMEn; recall Section 4.2; our variant of AMEn is used.

Allowing the number of levels to depend on the mode. We allow the possibility that a different number of levels for different modes is considered, as in Multigrid-AMEn; recall Section 4.2. For a certain level, if there are modes for which we do not want to restrict further, we simply set the corresponding core to identity.

5.2.2 Algorithm from Section 5.1.2 in TT format

The second proposed algorithm combines the algorithm discussed in Section 5.1.2 with TT format by again considering the algorithm with all structures in this format.

We now refer to important particularities of the algorithm. It will be possible to observe that most of them coincide with those in the other variant proposed in Section 5.2.1, which is logical since the core of the two algorithms is the same. Looking at the parameters that were tuned to get an optimal performance of the algorithms, we even note that the similarities between the two variants proposed in this chapter are more clear than those between the variant proposed in Section 5.2.1 and Multigrid-AMEn. This was not expected since these latter two are very similar as already noted before and particularly emphasized in Section 5.2.1. For instance, the number of smoothing steps or the parameters of the TT-SVD truncations are the same, as we see next, for the two variants proposed in this chapter while they were seen to differ between the algorithm proposed in Section 5.2.1 and Multigrid-AMEn.

Restriction and interpolation. As the corresponding aggregation is represented in a tensorized way, as in the variant proposed in Section 5.2.1, it again allows a TT representation with all entries of the TT rank equal to 1. The same holds again for disaggregation. The operators TT format associated with aggregation and disaggregation are obtained by setting the different cores to the small matrices from the Kronecker representation. Such matrices, in the case of aggregation, are now as described in Section 5.1.2: for a given level, they are all identity except for the one associated with the mode whose states are aggregated in that level.

We use the transpose of aggregation as disaggregation, again as in the variant proposed in Section 5.2.1.

Smoother. The fulfilments that must be verified by the smoother are, as in the context of the variant proposed in Section 5.2.1, the same as in the tensorized multigrid scheme proposed in Section 4.1.2. In this context, we choose again GMRES.

The number of smoothing steps that is considered is the same as in the mentioned variant: three steps in the finest grid while one step in the remaining grids. This results again in a storage and computation that is negligible.

Coarsest grid solver. While in the multigrid scheme from Section 5.2.1, the mode sizes are reduced from one level to another, while the value of d is unchanged; in this one we maintain the mode sizes but reduce the number of modes. Therefore, AMEn is not so suitable and it is replaced with the direct solver that was also used in the scheme described in Section 4.1.2 – Moore-Penrose pseudoinverse. This is not a problem since using such a direct solver is no longer problematic because there is no curse of dimensionality, as opposed to the case for the algorithm of reference, motivating the creation of Multigrid-AMEn back in Section 4.2. In fact, we can reduce the number of modes as much as desired.

Normalization. As in the variant proposed in Section 5.2.1, the sum of the entries must be 1 and this is not kept during a cycle, so that we normalize the obtained approximation after each cycle.

Truncations. Truncation is again performed to avoid excessive rank growth using TT-SVD algorithm, and it is done in the exact same steps of Algorithm 3 as in the variant proposed in Section 5.2.1. Additionally, it is applied with the same target accuracies. Furthermore, the imposed upper bound on the TT rank entries is defined in the same way.

Size of the coarsest grid problem. Since the expensive Moore-Penrose pseudoinverse is the coarsest grid solver, we define the number of levels as the smallest value for which the number of states on the coarsest grid is smaller than 350.

Initial approximation of the solution. The algorithm is again initialized with the tensor that results from solving the coarsest grid problem, after bringing it up the grids using interpolation. The difference is that the coarsest grid solver must be coherent with the solver that is used in general in the coarsest grid of the main cycle; the reasoning is the same as when we changed the coarsest grid solver from Section 4.1.2 to Section 4.2.

Allowing the number of levels to depend on the mode. As in the variant proposed in Section 5.2.1, we allow the possibility that a different number of levels for different modes, by considering identity matrices in the cores associated with modes that we do not want to restrict further.

5.3 Numerical experiments

We now analyse the performance of the two algorithms proposed in this chapter. The algorithms of reference for comparison are: AMEn, from Chapter 3; Multigrid-AMEn, from Chapter 4.

The algorithms from Sections 5.2.1 and 5.2.2 are denoted by “TensorizedAggregation” and “ModeAggregation”, respectively.

As in the experiments in Section 4.3, we stop an iteration when the relative accuracy, current residual norm divided by the residual norm associated with the tensor of all ones (scaled so that the sum of its entries is one), is smaller than 10^{-2}. This only corresponds to the initial guess in the case of AMEn. It is not the initial guess of the multigrid-based algorithms proposed in this thesis, which share a particular procedure for determining an initial guess.

Table 5.1: Comparison of the algorithms for model convergingmetab – $4^{10} \approx 1.05 \times 10^6$ states.

	Time	Rank	Iter
Multigrid-AMEn	37.4	15	13
AMEn	1.8	16	4
TensorizedAggregation	31.0	29	13
ModeAggregation	14.9	15	8

In the tables that follow, in coherence with the experiments from Section 4.3: "Time" stands for the computation time, in seconds; "Rank" stands for the maximal entry of the TT rank of the approximate solution; "Iter" stands for the required number of iterations, which are defined differently depending on whether they refer to AMEn or to a multigrid-based method, representing the number of sweeps and the number of cycles, respectively. We use "—" in the rows of the tables associated with algorithms that do not converge.

As in the experiments from Section 4.3, we will consider common mode sizes in each test case. As a consequence, for simplicity, we again state them as scalar values n, which represent such common values.

The benchmark problems that are used are all taken, as already noted, from the benchmark collection [Mac15]. We consider three different models as test cases.

5.3.1 First test case: converging metabolic pathways

We consider a model [LH07], named convergingmetab in [Mac15], from the field of chemical networks, which has a topology of interactions that should not suit TT format particularly well, in the context of the ideal underlying train topology introduced in Remark 1.

We consider $n = 4$ and $d = 10$; see Table 5.1.

AMEn has an excellent performance. It is, in fact, expected to be hard to beat, as concluded in the experiments in Section 3.4 and emphasized in the beginning of Section 4.2.3 when motivating the use of AMEn in the coarsest grid of Multigrid-AMEn algorithm, unless the entries of the TT rank are large since this would imply a significant increase on the cost of the subproblem that must be repeatedly solved; recall the discussion about the cost of applying a direct solver to such subproblems in Section 3.2. This was already noted in the previous experiments done in this thesis; more concretely, in Sections 3.4 and 5.3.

As for the proposed algorithms, we see that the restriction and interpolation operators that are used in TensorizedAggregation are particularly effective against those used in Multigrid-AMEn, recalling that the two algorithms are totally similar in terms of their concept, as introduced in Section 5.1.1 and emphasized in Section 5.2.1, while the concretizations differ exclusively on the way such operators are selected. They are both expected to perform well since this is an indistinguishable model, recall the discussion relating Multigrid-AMEn and indistinguishable models in Section 4.1.2 and then also the one associated with TensorizedAggregation in the beginning of this chapter. The idea is that this type of model has an intrinsic 1D topology in each subsystem, but the results of the experiments in Section 4.3 already suggested that the performance of Multigrid-AMEn should be possible to improve. TensorizedAggregation seems to be a method that allows achieving such desired improvement.

ModeAggregation also behaves particularly well in this context with small mode sizes. This is expected since this algorithm is only expected to have problems when the mode sizes are large given the local 1D topology that is intrinsic to such indistinguishable models; recall the associated comment in sequence of the representation in Figure 5.2.

Globally, the algorithms, all in TT format, seem to be efficient even when the underlying topology is not theoretically suitable, in coherence with the conclusions from the experiments in Section 4.3.

5.3.2 Second test case: cyclic metabolic pathways

We now consider another model [LH07], named `convergingmetab` in [Mac15], from the field of chemical networks, with an underlying topology that is even further from the ideal one. In fact, the train topology is destroyed by an interaction between the last and first subsystems, which introduces a cycle. Additionally, this model is reducible, meaning in particular that the steady state is not unique, which implies that it is a subclass of models that algorithms for finding steady states of Markov chains tend to avoid. We however note that, if we only focus on each connected component of states, the solution is unique again, so that a solution should in fact be possible to find. Basically, we can find a solution for each connected component since they are associated with different problems (in particular, the linear constraint associated with probabilities summing to one must be imposed on each connected component instead). Therefore, we can in fact deal with such models, as introduced in Section 1.1.

This model is again an indistinguishable model.

We go further on n and d, one at each time, considering two cases.

We first consider $n = 4$ and $d = 18$. The corresponding results can be found in Table 5.2.

Table 5.2: Comparison of the algorithms for model cyclemetab – $4^{18} \approx 6.87 \times 10^{10}$ states.

	Time	Rank	Iter
Multigrid-AMEn	525.5	80	22
AMEn	418.7	31	6
TensorizedAggregation	82.2	57	17
ModeAggregation	80.9	41	10

The proposed algorithms perform globally well despite the mentioned particularities of this model. The performances are worse than in Table 5.1 but this is not necessarily bad given that the number of modes that are now considered is also larger. The resulting large TT rank entries lead to a bad performance of AMEn, while Multigrid-AMEn is again outperformed by TensorizedAggregation. It is additionally outperformed by ModeAggregation, whose good performance is again justified by the small mode sizes that are considered.

The performance of the proposed algorithms might be even better in case there was more control on the rank growth. In fact, the entries of the TT rank could be much smaller while still leading to the same quality in the approximation of the solution; just see the maximum entry of the TT rank of the approximate solution obtained with AMEn. The same holds for Multigrid-AMEn. In fact, the upper bound that is imposed on the TT rank entries that are obtained after each truncation on all the multigrid-based algorithms (all except AMEn); recall for instance Section 4.1.2, where it is first described; in order to avoid such entries to grow too much does not work that well in this case, even if it did in the experiments in Section 4.3.

The fact that the algorithms are effective even when the underlying topology is not suitable to the format is even more emphasized, recalling that the topology associated with this model is particularly bad because of the existing cycle. Such topology is similar to that of model overflowpersim tested in Section 4.3, for which the performance of the algorithms in TT format was also surprisingly good.

The second case consists of taking $n = 32$ and $d = 6$. The corresponding results are in Table 5.3.

The performance of ModeAggregation is strongly affected by increasing the mode sizes, as expected; see the related argument in sequence of Table 5.1, noting that there is again an underlying 1D local topology because this is an indistinguishable model. As for AMEn, it is also more suited for problems with a large number of subsystems than for problems with a large number of possible states per subsystem; recall from Section 3.2 that there is a direct influence of the mode sizes on the cost of the subproblem, which is

Table 5.3: Comparison of the algorithms for model `cyclemetab` – $32^6 \approx 1.07 \times 10^9$ states.

	Time	Rank	Iter
Multigrid-AMEn	432.1	80	19
AMEn	1956.5	31	6
TensorizedAggregation	64.2	57	15
ModeAggregation	—	—	—

the main cost associated with this algorithm when a direct solver is used for solving it. This explains that its performance is worse than in Table 5.2 despite the smaller global problem size.

5.3.3 Third test case: wireless network with handoff

This model [SS97, AFRT06], named `handoff2class1d` in [Mac15], from the telecommunications field has a topology of interactions that suits TT format perfectly, again in the context of Remark 1, given that a linear local geometry is considered. Its main particularity is the fact that we consider two different classes of users. Thus, this model is in the subclass of distinguishable models, which significantly differ from indistinguishable models as explained in the context of queuing networks for the previously explored distinguishable model; Kanban control model, in the experiments in Section 3.4.2; and then generalized for models that are not necessarily queuing networks, while also explained in more detail, in Section 4.1.2. As noted in the same context in Section 4.1.2, this is a subclass in which the application of Multigrid-AMEn and TensorizedAggregation is not a good option since they consider restriction and interpolation operators that are only suitable for models with a local 1D topology. These ideas were noted in Section 4.1.2 and in the beginning of Section 5.1.1, respectively. Just as in the case of reducible models, algorithms for finding steady states of Markov chains tend to avoid testing on such models; an exception is [Buc99], where the mentioned Kanban control model is considered.

While Multigrid-AMEn and TensorizedAggregation are not suitable for this type of model, in a certain sense we expect the opposite from ModeAggregation. As noted in sequence of Figure 5.2, a 1D local topology, associated with indistinguishable models, is particularly bad for this algorithm when the mode sizes start to increase, as confirmed in Table 5.3. However, when considering this type of model, a distinguishable model, this bad effect on increasing mode sizes is expected to be much less significant. In fact, the distance between states is not as straight-forward in the new underlying topology, associated with the decreasing lexicographic ordering of the states in terms of the vector that characterizes them, as introduced in Section 3.4.2, but it is at least clear that it is

Table 5.4: Comparison of the algorithms for model `handoff2class1d` – 10^{28} states.

	Time	Rank	Iter
AMEn	4.7	4	3
ModeAggregation	149.7	21	4

Table 5.5: Comparison of the algorithms for model `handoff2class1d` – $210^4 \approx 1.94 \times 10^9$ states.

	Time	Rank	Iter
AMEn	231.4	8	6
ModeAggregation	134.7	29	5

not as in Figure 5.2. More concretely, the distance between states that are aggregated is now undoubtedly smaller.

We go even further on the mode sizes and number of dimensions, considering again two cases.

We first consider $n = 10$ and $d = 28$. The corresponding results can be found in Table 5.4.

AMEn and ModeAggregation can easily deal with 28 modes, which is expected since they are both particularly suited for large d. This was already explained in sequence of Table 5.3 for AMEn, and it is clear from its conceptual definition for ModeAggregation, recall Section 5.1.2, since the number of modes is what is reduced along the grids.

In the case of AMEn, because the TT rank entries are very small, in sequence of the already mentioned suitable underlying topology of the network, this algorithm is very hard to beat, in coherence with what had also been concluded for Table 5.1.

The reason why the computation time is much worse for ModeAggregation is the problem that was mentioned in the context of Table 5.2 concerning the fact that the TT rank entries of the multigrid-based methods easily grow too far.

Note that the problem size that is being addressed is clearly the largest that was tested – total of 10^{28} states.

The results for the second case, in which we consider $n = 210$ and $d = 4$, are in Table 5.5.

Despite the smaller problem size, when compared with Table 5.4, AMEn behaves worse because of the larger mode sizes that are now considered; by the same argument as in the comparison of the performances of AMEn in the two tables from Section 5.3.2.

We see that ModeAggregation can deal with such a value of n. In fact, while mode sizes 32 are too large for this algorithm to converge in Section 5.3.2, recall Table 5.3, we now get good convergence for mode sizes 210. This confirms the expectation that the problems that this algorithm has with large mode sizes for indistinguishable models disappear when considering distinguishable models.

In the end, despite not using any information about the topology of each subsystem (local topology), which is more complex in the presence of distinguishable models, ModeAggregation seems to perfectly address such models. Note that it would be even more competitive if the adopted rank adaptivity scheme would not generate far too large entries of the TT rank of the approximate solution.

5.3.4 Choosing the algorithm to use

In sequence of the performed experiments, the way to decide which algorithm to use, depending on the type of model that is given, should be clear. In particular, we will conclude that the two algorithms proposed in this chapter cover all possible models.

AMEn would be the best option in general, as noted in sequence of Tables 5.1 and 5.4, if its performance was not so strongly affected by increasing TT rank entries of the approximate solution. The problems when the TT rank entries increase are justified by the cost of the subproblems; recall the related discussion in Section 3.2. Additionally, in sequence of the mentioned discussion, the TT rank entries do not even need to be particularly large for the cost of the subproblems to become prohibitively large. Moreover, it is common that such large TT rank entries are found, for instance, for models with not so suitable underlying topologies or when considering mode sizes that are not particularly small; recall Tables 5.3 and 5.5, respectively.

As for Multigrid-AMEn, while it is clearly a good option given that it performs well for a large and representative collection of models as seen in the experiments in Section 4.3, the restriction and interpolation that are chosen seem to be possible to outperform, as we had also predicted in the context of the mentioned experiments. This in fact happens when replacing them with the aggregation and disaggregation operators proposed in TensorizedAggregation, as repeatedly observed in the tables throughout the experiments that were performed. This way we can see TensorizedAggregation as an improved version of Multigrid-AMEn. In fact, as already noted in sequence of Table 5.1 and emphasized when describing TensorizedAggregation in Section 5.2.1, these two algorithms are conceptually very similar, being both particularly suited for inistinguishable models, while not for distinguishable models, so that the comparison of their performances directly reflects the quality of the chosen restriction and interpolation operators.

Therefore, for indistinguishable models, TensorizedAggregation is the best option, noting that ModeAggregation has the problem associated with the intrinsic 1D local topology that has been particularly noted in Table 5.3.

Concerning distinguishable models, TensorizedAggregation, and also Multigrid-AMEn, are not suitable, as explained in the beginning of Section 5.3.3. The only option that remains is ModeAggregation. This is however not a problem since this method has a particularly good performance when applied to models of this subclass. In fact, the problem associated with the 1D local topology that affects it does not apply now.

Summing up, we have proposed two algorithms that seem to perfectly complement each other. While TensorizedAggegation is ideal for indistinguishable models but cannot be applied to distinguishable models effectively, ModeAggregation has problems with the underlying 1D local topology of indistinguishable models but behaves particularly well for distinguishable models. The most important fact to be noted is that, for each of the two subclasses in which we have partitioned the set of all models, indistinguishable/distinguishable models, there is one algorithm that perfectly suits it.

It should be added that being distinguishable or indistinguishable may, in some cases, not be a characteristic of the model, but instead of its individual subsystems. This is the case of the model analysed in Section 3.4.2, the Kanban control model, which is not completely distinguishable since its first and last subsystems are in fact indistinguishable. The solution, in such contexts, is to combine the two proposed algorithms. This can be done in two ways. We can apply aggregation as in ModeAggregation until all modes that remain have a 1D local topology, so that TensorizedAggregation can be then effectively applied in the remaining levels. We can alternatively go in the opposite direction and first apply aggregation as in TensorizedAggregation, only restricting the modes that do not have a 1D local topology, and then apply ModeAggregation in the remaining levels since all modes are then associated with 1D local topology but with a small number of states (those that have been processed by TensorizedAggregation), or with different local topologies.

5.4 Conclusion

We have proposed algorithms for approximating steady states for structured Markov chains combining the concepts of aggregation/disaggregation with TT decompositions.

Numerical experiments demonstrate that, for general problem sizes, we go further in terms of performance comparing the two proposed algorithms against the main algorithms proposed in the previous two chapters: AMEn and Multigrid-AMEn. They can, furthermore, perform remarkably well for very high-dimensional problems. The largest problem size that is addressed is 10^{28}; the largest mode sizes are 210; the maximum number of modes is 28.

We take the definition of robustness that was used to characterize the algorithm Multigrid-AMEn in the previous chapter to a whole new level. With the proposed algorithms, all models deriving from Markov chains with a simple Kronecker representation should be possible to address. In fact, when choosing the algorithm to use, depending on the type of model, the choice is quite natural, as seen in Section 5.3.4. While after the previous chapter, we only had an algorithm to deal with all possible indistinguishable models, we can now also find a solution efficiently for distinguishable models, so that robustness is now a word that perfectly applies. Furthermore, the performance of the algorithm for indistinguishable models was improved.

Still in the context of robustness, besides covering different fields of interest, the tests done in this chapter include: a reducible model, for which the performances of the algorithms, both in comparative and absolute terms, seem to be similar to the performances for a general irreducible model; a distinguishable model, which should be particularly challenging but for which we have now an algorithm that perfectly deals with it. These two subclasses of models are in fact extremely important because of their range of applications and also because of the difficulties, in the past, in addressing them. Additionally, the idea from the conclusions of Chapter 4 that we obtain good performances of the algorithms even for models with underlying topologies that are not, in theory, suitable for applying TT format is emphasized.

In the end, the two main problems mentioned in the conclusions of the previous chapter were addressed: we proposed an algorithm that improves the performance of Multigrid-AMEn on indistinguishable models by considering other operators for restriction and interpolation, even though we maintain the exact same concept; we proposed an algorithm that can deal particularly well with distinguishable models.

One important limitation that the proposed algorithms have, and which also affects Multigrid-AMEn, is on the control of the rank growth. Even imposing an upper bound on the TT rank entries allowed after each truncation done inside each cycle, such entries can grow far too much, clearly influencing the global performances. This should be further investigated.

6 Entropy and mutual information

In this chapter, we present the main ideas behind the concepts of entropy and mutual information, along with their basic properties. Moreover, we discuss definitions and important properties associated with conditional MI and MI between three random vectors. In what follows, $\mathscr{X}$ denotes the support of a random vector $\boldsymbol{X}$. Additionally, we assume the convention $0 \ln 0 = 0$, justified by continuity since $x \ln x \to 0$ as $x \to 0^+$.

This chapter is thus the fundamental introduction to what follows in the second part of the thesis.

6.1 Entropy

The concept of entropy [Sha48] was initially motivated by problems in the field of telecommunications. Introduced for discrete random variables, the entropy is a measure of uncertainty. In the following, $P(A)$ denotes the probability of A.

Definition 1. *The entropy of a discrete random vector* $\boldsymbol{X}$ *is:*

$$H(\boldsymbol{X}) = -\sum_{\boldsymbol{x} \in \mathscr{X}} P(\boldsymbol{X} = \boldsymbol{x}) \ln P(\boldsymbol{X} = \boldsymbol{x}).$$

Given an additional discrete random vector $\boldsymbol{Y}$*, the conditional entropy of* $\boldsymbol{X}$ *given* $\boldsymbol{Y}$ *is*

$$H(\boldsymbol{X}|\boldsymbol{Y}) = -\sum_{\boldsymbol{y} \in \mathscr{Y}} \sum_{\boldsymbol{x} \in \mathscr{X}} P(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{Y} = \boldsymbol{y}) P(\boldsymbol{Y} = \boldsymbol{y}) \ln P(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{Y} = \boldsymbol{y}).$$

Note that the entropy of $\boldsymbol{X}$ does not depend on the particular values taken by the random vector but only on the corresponding probabilities. It is clear that entropy is non-negative since each term of the summation in (1) is non-positive. Additionally, the value 0 is only obtained for a degenerate random variable.

An important property that results from Definition 1 is the so-called *chain rule* [CT06, Ch. 2]:

$$H(\boldsymbol{X}_1, ..., \boldsymbol{X}_n) = \sum_{i=2}^{n} H(\boldsymbol{X}_i | \boldsymbol{X}_{i-1}, ..., \boldsymbol{X}_1) + H(\boldsymbol{X}_1), \tag{6.1}$$

where a sequence of random vectors, such as $(\boldsymbol{X}_1, ..., \boldsymbol{X}_n)$ and $(\boldsymbol{X}_{i-1}, ..., \boldsymbol{X}_1)$ above, should be seen as the random vector that results from the concatenation of its elements.

6.2 Differential entropy

A logical way to adapt the definition of entropy to the case where we deal with an absolutely continuous random vector is to replace the probability (mass) function of a discrete random vector by the probability density function of an absolutely continuous random vector, as next presented. The resulting concept is called *differential entropy*. We let $f_{\boldsymbol{X}}$ denote the probability density function of an absolutely continuous random vector $\boldsymbol{X}$.

Definition 2. *The differential entropy of an absolutely continuous random vector $\boldsymbol{X}$ is:*

$$h(\boldsymbol{X}) = -\int_{\boldsymbol{x} \in \mathscr{X}} f_{\boldsymbol{X}}(\boldsymbol{x}) \ln f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{6.2}$$

Given an additional absolutely continuous random vector $\boldsymbol{Y}$, such that $(\boldsymbol{X}, \boldsymbol{Y})$ is also absolutely continuous, the conditional differential entropy of $\boldsymbol{X}$ given $\boldsymbol{Y}$ is

$$h(\boldsymbol{X}|\boldsymbol{Y}) = -\int_{\boldsymbol{y} \in \mathscr{Y}} f_{\boldsymbol{Y}}(\boldsymbol{y}) \int_{\boldsymbol{x} \in \mathscr{X}} f_{\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}}(\boldsymbol{x}) \ln f_{\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}}(\boldsymbol{x}) d\boldsymbol{x}\, d\boldsymbol{y}.$$

It can be proved [CT06, Ch. 9] that the chain rule (6.1) still holds replacing entropy by differential entropy.

The notation that is used for differential entropy, h, is different from the notation used for entropy, H. This is justified by the fact that entropy and differential entropy do not share the same properties. For instance, non-negativity does not necessarily hold for differential entropy. Also note that $h(\boldsymbol{X}, \boldsymbol{X})$ and $h(\boldsymbol{X}|\boldsymbol{X})$ are not defined given that the pair $(\boldsymbol{X}, \boldsymbol{X})$ is not absolutely continuous. Therefore, relations involving entropy and differential entropy need to be interpreted in a different way.

Example 1. *If $\boldsymbol{X}$ is a random vector, of dimension n, following a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{X} \sim \mathscr{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the value of the corresponding differential entropy is $\frac{1}{2}\ln\left((2\pi e)^n |\boldsymbol{\Sigma}|\right)$ [CT06, Ch. 9], where $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$. In particular, for the one-dimensional case, $X \sim \mathscr{N}(\mu, \sigma^2)$, the differential entropy is negative if $\sigma^2 < 1/2\pi e$, positive if $\sigma^2 > 1/2\pi e$, and zero if $\sigma^2 = 1/2\pi e$. Thus, a zero differential entropy does not have the same interpretation as in* *the discrete case. Moreover, the differential entropy can take arbitrary negative values.*

In what follows, when the context is clear, we will refer to differential entropy simply as entropy.

6.3 Mutual information

We now introduce mutual information (MI), which is a very important measure since it measures both linear and non-linear associations between two random vectors.

6.3.1 Discrete case

Definition 3. *The MI between two discrete random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ is:*

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) = \sum_{\boldsymbol{x}\in\mathscr{X}} \sum_{\boldsymbol{y}\in\mathscr{Y}} P(\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y}) \ln \frac{P(\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y})}{P(\boldsymbol{X}=\boldsymbol{x})P(\boldsymbol{Y}=\boldsymbol{y})}.$$

MI satisfies the following; see, e.g., [CT06, Ch. 9]:

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) = H(\boldsymbol{X}) - H(\boldsymbol{X}|\boldsymbol{Y}); \tag{6.3}$$

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) \geq 0; \tag{6.4}$$

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{X}) = H(\boldsymbol{X}). \tag{6.5}$$

Equality holds in (6.4) if and only if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent random vectors.

According to (6.3), $\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y})$ can be interpreted as the reduction in the uncertainty of $\boldsymbol{X}$ due to the knowledge of $\boldsymbol{Y}$. Note that, applying (6.1), we also have

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) = H(\boldsymbol{X}) + H(\boldsymbol{Y}) - H(\boldsymbol{X},\boldsymbol{Y}). \tag{6.6}$$

Another important property that immediately follows from (6.3) is

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) \leq \min(H(\boldsymbol{X}), H(\boldsymbol{Y})). \tag{6.7}$$

In sequence, in view of (6.3) and (6.4), we can conclude that, for any random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$,

$$H(\boldsymbol{X}|\boldsymbol{Y}) \leq H(\boldsymbol{X}). \tag{6.8}$$

This result is again coherent with the intuition that entropy measures uncertainty. In fact, if more information is added, about $\boldsymbol{Y}$ in this case, the uncertainty about $\boldsymbol{X}$ will not increase.

6.3.2 Continuous case

Definition 4. *The MI between two absolutely continuous random vectors* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$*, such that* $(\boldsymbol{X}, \boldsymbol{Y})$ *is also absolutely continuous, is:*

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) = -\int_{\boldsymbol{y}\in\mathscr{Y}} \int_{\boldsymbol{x}\in\mathscr{X}} f_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) \ln \frac{f_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y})}{f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{Y}}(\boldsymbol{y})} d\boldsymbol{x}\, d\boldsymbol{y}.$$

It is straight-forward to check, given the similarities between this definition and Definition 3, that most properties from the discrete case still hold replacing entropy by differential entropy. In particular, the only property from (6.3) to (6.5) that cannot be restated for differential entropy is (6.5) since Definition 4 does not cover $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{X})$, again because the pair $(\boldsymbol{X}, \boldsymbol{X})$ is not absolutely continuous. Additionally, restatements of (6.6) and (6.8) for differential entropy also hold.

On the whole, MI for absolutely continuous random vectors verifies most important properties from the discrete case, including being symmetric and non-negative. Moreover, the value 0 is obtained if and only if the random variables are independent. Concerning a parallel of (6.7) for absolutely continuous random vectors, there is no natural finite upper bound for $h(\boldsymbol{X})$ in the continuous case. In fact, while the expression $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) = h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{Y})$, similar to (6.3), holds, $h(\boldsymbol{X}|\boldsymbol{Y})$ and $h(\boldsymbol{Y}|\boldsymbol{X})$ are not necessarily non-negative. Furthermore, as noted in Example 1, differential entropies can be become arbitrarily small, which applies, in particular, to the terms $h(\boldsymbol{X}|\boldsymbol{Y})$ and $h(\boldsymbol{Y}|\boldsymbol{X})$. As a result, $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y})$ can grow arbitrarily.

6.3.3 Combination of continuous with discrete random vectors

The definition of MI when we have an absolutely continuous random vector and a discrete random vector is also important in later stages of this article. For this reason, and despite the fact that the results that follow are naturally obtained from those that involve only either discrete or absolutely continuous vectors, we briefly go through them now.

Definition 5. *The MI between an absolutely continuous random vector* $\boldsymbol{X}$ *and a discrete random vector* $\boldsymbol{Y}$ *is given by either of the following two expressions:*

$$\begin{aligned}\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) &= \sum_{\boldsymbol{y}\in\mathscr{Y}} P(\boldsymbol{Y} = \boldsymbol{y}) \int_{\boldsymbol{x}\in\mathscr{X}} f_{\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}}(\boldsymbol{x}) \ln \frac{f_{\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y}}(\boldsymbol{x})}{f_{\boldsymbol{X}}(\boldsymbol{x})} d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in\mathscr{X}} f_{\boldsymbol{X}}(\boldsymbol{x}) \sum_{\boldsymbol{y}\in\mathscr{Y}} P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x}) \ln \frac{P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{Y} = \boldsymbol{y})} d\boldsymbol{x}.\end{aligned}$$

The majority of the properties stated for the discrete case are still valid in this case. In particular, analogues of (6.3) hold, both in terms of entropies as well as in terms of differential entropies:

$$\begin{aligned} \mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) &= h(\boldsymbol{X}) - h(\boldsymbol{X}|\boldsymbol{Y}) \qquad (6.9) \\ &= H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}). \qquad (6.10) \end{aligned}$$

Furthermore, $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) \leq H(\boldsymbol{Y})$ is the analogue of (6.7) for this setting. Note that (6.10), but not (6.9), can be used to obtain an upper bound for $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y})$ since $h(\boldsymbol{X}|\boldsymbol{Y})$ may be negative.

6.4 Triple mutual information and conditional mutual information

We next discuss definitions and important properties associated with conditional MI and MI between three random vectors. Random vectors are considered to be discrete in this section as the generalization of the results for absolutely continuous random vectors would follow a similar approach.

6.4.1 Conditional mutual information

Conditional MI is defined in terms of entropies as follows, in a similar way to property (6.3); cf. [Fle04, MB06].

Definition 6. *The conditional MI between two random vectors* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *given the random vector* $\boldsymbol{Z}$ *is written as*

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) = H(\boldsymbol{X}|\boldsymbol{Z}) - H(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{Z}). \qquad (6.11)$$

Using (6.11) and an analogue of the chain rule for conditional entropy, we conclude that:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) = H(\boldsymbol{X}|\boldsymbol{Z}) + H(\boldsymbol{Y}|\boldsymbol{Z}) - H(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}). \qquad (6.12)$$

In view of Definition 6, developing the involved terms according to Definition 3, we obtain:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) = E_{\boldsymbol{Z}}[\mathrm{MI}(\tilde{\boldsymbol{X}}(\boldsymbol{Z}), \tilde{\boldsymbol{Y}}(\boldsymbol{Z})], \qquad (6.13)$$

where, for $\boldsymbol{z} \in \mathscr{Z}$, $(\tilde{\boldsymbol{X}}(\boldsymbol{z}), \tilde{\boldsymbol{Y}}(\boldsymbol{z}))$ is equal in distribution to $(\boldsymbol{X}, \boldsymbol{Y})|\boldsymbol{Z} = \boldsymbol{z}$.

Taking (6.4) and (6.13) into account,

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) \geq 0, \tag{6.14}$$

and $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) = 0$ if and only if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given $\boldsymbol{Z}$.

Moreover, from (6.11) and (6.14), we conclude the following result similar to (6.8):

$$H(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{Z}) \leq H(\boldsymbol{X}|\boldsymbol{Z}). \tag{6.15}$$

6.4.2 Triple mutual information

The generalization of the concept of MI to more than two random vectors is not unique. One such definition, associated with the concept of total correlation, was proposed in [Wat60]. An alternative one, proposed in [Bel03], is called *triple MI* (TMI). We will consider the latter since it is the most meaningful in the context of objective functions associated with the problem of forward feature selection.

Definition 7. *The triple MI between three random vectors $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ is defined as*

$$\begin{aligned}\mathrm{TMI}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \sum_{\boldsymbol{x}\in\mathscr{X}} \sum_{\boldsymbol{y}\in\mathscr{Y}} \sum_{\boldsymbol{z}\in\mathscr{Z}} & P(\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{Z}=\boldsymbol{z}) \times \\ & \ln \frac{P(\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y})P(\boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{Z}=\boldsymbol{z})P(\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Z}=\boldsymbol{z})}{P(\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{Z}=\boldsymbol{z})P(\boldsymbol{X}=\boldsymbol{x})P(\boldsymbol{Y}=\boldsymbol{y})P(\boldsymbol{Z}=\boldsymbol{z})}.\end{aligned}$$

Using the definition of MI and TMI, we can conclude that TMI and conditional MI are related in the following way, which provides extra intuition about the two concepts:

$$\mathrm{TMI}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) - \mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}). \tag{6.16}$$

The TMI is not necessarily non-negative. This fact is exemplified and discussed in detail in the next chapter.

7 Forward feature selection methods based on MI

In the first part of this chapter, we focus on explaining the general context concerning forward feature selection methods based on mutual information. More concretely, target (ideal) objective functions to be maximized in each step are presented and studied. We then define important concepts and prove some properties of such target objective functions. These target objective functions cannot be used in practice as there is a term that is required when evaluating them that is quite complex. It requires, in particular, the knowledge of a high-dimensional term that is hard to estimate accurately. The common solution is to use approximations, leading to different feature selection methods. For the analysis in this thesis, we selected a set of methods representative of the main types of approximations to the target objective function. In the second part of the chapter, we explore such methods: we describe them, we discuss drawbacks resulting from their underlying approximations, and we discuss how they cope with some desirable properties that hold for the target objective function. A third and last part consists of a distributional setting, based on a specific definition of class, features, and a performance metric. The setting provides an ordering for each of the methods, which is independent of specific datasets and estimation methods, and is compared with the ideal feature ordering. The aim of the setting is to illustrate how the drawbacks of the methods lead to incorrect feature ordering and to the loss of the good properties of the target objective functions.

The content of this chapter is based on the paper [MOPV17].

7.1 The forward feature selection problem

In this section, we focus on explaining the general context concerning forward feature selection methods based on mutual information. We first introduce target objective functions to be maximized in each step; we then define important concepts and prove some properties of such target objective functions. In the rest of this section, features are considered to be discrete for simplicity. The name target objective functions comes from the fact that, as we will argue, these are objective functions that perform exactly as we would desire ideally, so that a good method should reproduce its properties as well as possible.

7.1.1 Target objective functions

Let C represent the class, which identifies the group each object belongs to. $\boldsymbol{S}$ ($\boldsymbol{F}$), in turn, denote the set of selected (unselected) features at a certain step of the iterative algorithm; in fact, $\boldsymbol{S} \cap \boldsymbol{F} = \emptyset$, and $\boldsymbol{S} \cup \boldsymbol{F}$ is the set with all input features. In what follows, when a set of random variables is in the argument of an entropy or MI term, it stands for the random vector composed by the random variables it contains.

Given the set of selected features, forward feature selection methods aim to select a candidate feature $X_j \in \boldsymbol{F}$ such that

$$X_j = \arg \max_{X_i \in \boldsymbol{F}} \mathrm{MI}(C, \boldsymbol{S} \cup \{X_i\}).$$

Therefore, X_j is, among the features in $\boldsymbol{F}$, the feature X_i for which $\boldsymbol{S} \cup \{X_i\}$ maximizes the association (measured using MI) with the class, C. Note that we choose the feature that maximizes $\mathrm{MI}(C, X_i)$ in the first step (i.e., when $\boldsymbol{S} = \emptyset$).

Since $\mathrm{MI}(C, \boldsymbol{S} \cup \{X_i\}) = \mathrm{MI}(C, \boldsymbol{S}) + \mathrm{MI}(C, X_i|\boldsymbol{S})$ [HLLC08], in view of (6.16), the objective function evaluated at the candidate feature X_i can be written as

$$\begin{aligned} \mathrm{OF}(X_i) &= \mathrm{MI}(C, \boldsymbol{S}) + \mathrm{MI}(C, X_i|\boldsymbol{S}) \\ &= \mathrm{MI}(C, \boldsymbol{S}) + \mathrm{MI}(C, X_i) - \mathrm{TMI}(C, X_i, \boldsymbol{S}) \\ &= \mathrm{MI}(C, \boldsymbol{S}) + \mathrm{MI}(C, X_i) - \mathrm{MI}(X_i, \boldsymbol{S}) + \mathrm{MI}(X_i, \boldsymbol{S}|C). \end{aligned} \tag{7.1}$$

The feature selection methods try to approximate this objective function. However, since the term $\mathrm{MI}(C, \boldsymbol{S})$ does not depend on X_i, the reference objective function for most feature selection methods is the simplified form of objective function given by

$$\mathrm{OF}'(X_i) = \mathrm{MI}(C, X_i) - \mathrm{MI}(X_i, \boldsymbol{S}) + \mathrm{MI}(X_i, \boldsymbol{S}|C).$$

This objective function has distinct properties from those of (7.1) and, therefore, deserves being addressed separately.

The objective functions OF and OF′ can be written in terms of entropies, which provides a useful interpretation. Using (6.3), we obtain for the first objective function:

$$\mathrm{OF}(X_i) = H(C) - H(C|X_i, \boldsymbol{S}). \tag{7.2}$$

Maximizing $H(C) - H(C|X_i, \boldsymbol{S})$ provides the same candidate feature X_j as minimizing $H(C|X_i, \boldsymbol{S})$, for $X_i \in \boldsymbol{F}$. This means that the feature to be selected is the one leading to the minimal uncertainty of the class among the candidate features. As for the second objective function, we obtain, using again (6.3):

$$\mathtt{OF}'(X_i) = H(C|\boldsymbol{S}) - H(C|X_i, \boldsymbol{S}). \tag{7.3}$$

This emphasizes that a feature that maximizes (7.2) also maximizes (7.3). In fact, the term that depends on X_i is the same in the two expressions.

We now provide bounds for the target objective functions.

Theorem 3. *Given a general candidate feature X_i:*

1. $H(C) - H(C|\boldsymbol{S}) \leq \mathtt{OF}(X_i) \leq H(C)$.
2. $0 \leq \mathtt{OF}'(X_i) \leq H(C|\boldsymbol{S})$.

Proof. Using the corresponding representations (7.2) and (7.3) of the associated objective functions, the upper bounds follow from $H(C|X_i, \boldsymbol{S}) \geq 0$. As for the lower bounds, in the case of statement 1, it comes directly from the fact that $\mathtt{OF}'(X_i) = \mathrm{MI}(C, X_i|\boldsymbol{S}) \geq 0$. As for statement 2, given that, from (7.2), $\mathtt{OF}(X_i) = H(C) - H(C|X_i, \boldsymbol{S}) = H(C) - H(C|\boldsymbol{S}) + \mathrm{MI}(C, X_i|\boldsymbol{S})$, we again only need to use the fact that $\mathrm{MI}(C, X_i|\boldsymbol{S}) \geq 0$. $\square$

The upper bound for $\mathtt{OF}$, $H(C)$, corresponds to the uncertainty in C, and the upper bound on $\mathtt{OF}'$, $H(C|\boldsymbol{S})$, corresponds to the uncertainty in C not explained by the already selected features, $\boldsymbol{S}$. This is coherent with the fact that $\mathtt{OF}'$ ignores the term $\mathrm{MI}(C, \boldsymbol{S})$. The lower bound for $\mathtt{OF}$ corresponds to the uncertainty in C already explained by $\boldsymbol{S}$.

7.1.2 Feature types and their properties

Features can be characterized according to their usefulness in explaining the class at a particular step of the feature selection process. There are two broad types of features, those that add information to the explanation of the class, i.e. for which $\mathrm{MI}(C, X_i|\boldsymbol{S}) > 0$, and those that do not, i.e. for which $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0$. However, a finer categorization is needed to fully determine how the feature selection process should behave. We define four types of features: irrelevant, redundant, relevant, and fully relevant.

Definition 8. *Given a subset of already selected features, $\boldsymbol{S}$, at a certain step of a forward sequential method, where the class is C, and a candidate feature X_i, then:*

- *X_i is irrelevant given $(C, \boldsymbol{S})$ if* $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0 \wedge H(X_i|\boldsymbol{S}) > 0$;

- *X_i is redundant given $\boldsymbol{S}$ if $H(X_i|\boldsymbol{S}) = 0$;*
- *X_i is relevant given $(C, \boldsymbol{S})$ if $\mathrm{MI}(C, X_i|\boldsymbol{S}) > 0$;*
- *X_i is fully relevant given $(C, \boldsymbol{S})$ if $H(C|X_i, \boldsymbol{S}) = 0 \wedge H(C|\boldsymbol{S}) > 0$.*

If $\boldsymbol{S} = \emptyset$, then $\mathrm{MI}(C, X_i|\boldsymbol{S})$, $H(X_i|\boldsymbol{S})$, $H(C|\boldsymbol{S})$, and $H(C|X_i, \boldsymbol{S})$ should be replaced by $\mathrm{MI}(C, X_i)$, $H(X_i)$, $H(C)$, and $H(C|X_i)$, respectively.

Under this definition, irrelevant, redundant, and relevant features form a partition of the set of candidate features $\boldsymbol{F}$. Note that fully relevant features are also relevant since $H(C|X_i, \boldsymbol{S}) = 0$ and $H(C|\boldsymbol{S}) > 0$ imply that $\mathrm{MI}(C, X_i|\boldsymbol{S}) = H(C|\boldsymbol{S}) - H(C|X_i, \boldsymbol{S}) > 0$.

Our definition introduces two novelties regarding previous works: first, we separate non-relevant features in two categories, of irrelevant and redundant features; second, we introduce the important category of fully relevant features.

Our motivation for separating irrelevant from redundant features is that, while a redundant feature remains redundant at all subsequent steps of the feature selection process, the same does not hold necessarily for irrelevant features. The following example illustrates how an irrelevant feature can later become relevant.

Example 2. *We consider a class $C = (X + Y)^2$ where X and Y are two independent candidate features that follow uniform distributions on $\{-1, 1\}$. C follows a uniform distribution on $\{0, 4\}$ and, as a result, the entropies of X, Y and C are $\ln(2)$. It can be easily checked that both X and Y are independent of the class. In the feature selection process, both features are initially irrelevant since, due to their independence from C, $\mathrm{MI}(C, X) = \mathrm{MI}(C, Y) = 0$. Suppose that X is selected first. Then, Y becomes relevant since $\mathrm{MI}(C, Y|X) = \ln(2) > 0$, and it is even fully relevant since $H(C|Y, X) = 0$ and $H(C|X) = \ln(2) > 0$.*

The following theorem shows that redundant features always remain redundant.

Theorem 4. *If a feature is redundant given $\boldsymbol{S}$, then it is also redundant given $\boldsymbol{S}'$, for $\boldsymbol{S} \subset \boldsymbol{S}'$.*

Proof. Suppose that X_i is a redundant feature given $\boldsymbol{S}$, so that $H(X_i|\boldsymbol{S}) = 0$, and $\boldsymbol{S} \subset \boldsymbol{S}'$. This implies that $H(X_i|\boldsymbol{S}') = 0$ by (6.15). As a result, X_i is also redundant given $\boldsymbol{S}'$. $\square$

This result has an important practical consequence: features that are redundant at a certain step of the feature selection process can be immediately removed from the set of candidate features $\boldsymbol{F}$, alleviating in this way the computational effort associated with the feature selection process.

Regarding relevant features, note that there are several levels of relevancy, as measured by $\mathrm{MI}(C, X_i|\boldsymbol{S})$. Fully relevant features form an important subgroup of relevant features since, together with already selected features, they completely explain the class, i.e. $H(C|\boldsymbol{S})$ becomes 0 after selecting a fully relevant feature. Thus, all remaining unselected features are either irrelevant or redundant and the algorithm must stop. This also means that detecting a fully relevant feature can be used as a stopping criterion of forward feature selection methods. The condition $H(C|\boldsymbol{S}) > 0$ in the definition of fully relevant feature is required since an unselected feature can no longer be considered of this type if it already holds that $H(C|\boldsymbol{S}) = 0$.

A stronger condition that could be considered as a stopping criterion is $H(C|\boldsymbol{S}) = H(C|\boldsymbol{S}, \boldsymbol{F})$, meaning that the (complete) set of candidate features $\boldsymbol{F}$ has no further information to explain the class. As in the previous case, the candidate features will all be irrelevant or redundant. However, since forward feature selection algorithms only consider one candidate feature at each iteration, and the previous condition requires considering all candidate features simultaneously, such condition cannot be used as a stopping criterion.

Regarding the categorization of features introduced by other authors, only one category of non-relevant features was considered in [BPZL12], named irrelevant, consisting of the candidate features X_i such that $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0$. Both irrelevant and redundant features have been considered in [MSB08, VE14]. The definition of irrelevant feature is the one in [BPZL12]; redundant features are defined as features such that $H(X_i|\boldsymbol{S}) = 0$. Since the latter condition implies that $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0$ by (6.3) and (6.15), it turns out that redundant features are only a special case of irrelevant ones, which is not in agreement with our definition.

According to the feature types introduced above, a good feature selection method must select, at a given step, a relevant feature, preferably a fully relevant one, keep irrelevant features for future consideration and discard redundant features. The following theorem relates these desirable properties with the values taken by the target objective functions.

Theorem 5.

1. *If X_i is a fully relevant feature given $(C, \boldsymbol{S})$, then $\mathrm{OF}(X_i) = H(C)$ and $\mathrm{OF}'(X_i) = H(C|\boldsymbol{S})$, i.e., the maximum possible values taken by the target objective functions are reached; recall Theorem 3.*
2. *If X_i is an irrelevant feature given $(C, \boldsymbol{S})$, then $\mathrm{OF}(X_i) = H(C) - H(C|\boldsymbol{S})$ and $\mathrm{OF}'(X_i) = 0$, i.e., the minimum possible values of the target objective functions are reached; recall Theorem 3.*
3. *If X_i is a redundant feature given $\boldsymbol{S}$, then $\mathrm{OF}(X_i) = H(C) - H(C|\boldsymbol{S})$ and $\mathrm{OF}'(X_i) = 0$, i.e., the minimum possible values of the target objective functions are reached; recall Theorem 3.*

4. *If* X_i *is a relevant feature, but not fully relevant, given* $(C, \boldsymbol{S})$*, then* $H(C) - H(C|\boldsymbol{S}) < \mathrm{OF}(X_i) < H(C)$ *and* $0 < \mathrm{OF}'(X_i) < H(C|\boldsymbol{S})$.

Proof. The two equalities in statement 1 are an immediate consequence of equations (7.2) and (7.3), using the fact that $H(C|X_i, \boldsymbol{S}) = 0$ if X_i is fully relevant given $(C, \boldsymbol{S})$.

Suppose that X_i is an irrelevant feature given $(C, \boldsymbol{S})$, so that $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0$. Then, the relation $\mathrm{OF}'(X_i) = 0$ results directly from $\mathrm{OF}'(X_i) = \mathrm{MI}(C, X_i|\boldsymbol{S})$. Conversely, the relation $\mathrm{OF}(X_i) = H(C) - H(C|\boldsymbol{S})$ follows from the fact that $\mathrm{OF}(X_i) = H(C) - H(C|\boldsymbol{S}) + \mathrm{MI}(C, X_i|\boldsymbol{S})$. As a result, statement 2 is verified.

The equalities in statement 3 follow likewise since $\mathrm{MI}(C, X_i|\boldsymbol{S}) = 0$ if X_i is a redundant feature given $\boldsymbol{S}$.

As for statement 4, we need to prove that the objective functions neither take the minimum nor the maximum value for a relevant feature that is not fully relevant. We start by checking that the minimum values are not reached. The proof is similar to that of statement 2. Since $\mathrm{OF}'(X_i) = \mathrm{MI}(C, X_i|\boldsymbol{S})$, and since the assumption is that $\mathrm{MI}(C, X_i|\boldsymbol{S}) > 0$, then $\mathrm{OF}'(X_i)$ is surely larger than 0. Concerning $\mathrm{OF}(X_i)$, since $\mathrm{OF}(X_i) = H(C) - H(C|\boldsymbol{S}) + \mathrm{MI}(C, X_i|\boldsymbol{S})$ and $\mathrm{MI}(C, X_i|\boldsymbol{S}) > 0$, $\mathrm{OF}(X_i)$ must be larger than $H(C) - H(C|\boldsymbol{S})$. Concerning the upper bounds, the proof is now similar to that of statement 1. If the feature X_i is not fully relevant given $(C, \boldsymbol{S})$, meaning that $H(C|\boldsymbol{S}, X_i) > 0$, the desired conclusions immediately follow from (7.2) and (7.3). ☐

Thus, fully relevant (irrelevant and redundant) features reach the maximum (minimum) of the objective functions, and relevant features that are not fully relevant reach a value between the maximum and the minimum values of the objective functions. These properties assure that the ordering of features at a given step of the feature selection process is always correct. Note that irrelevant and redundant features can be discriminated by evaluating $H(X_i|\boldsymbol{S})$.

7.1.3 Complementarity

The concept of complementarity is associated with the TMI term of the target objective function, given by $\mathrm{TMI}(C, X_i, \boldsymbol{S}) = \mathrm{MI}(X_i, \boldsymbol{S}) - \mathrm{MI}(X_i, \boldsymbol{S}|C)$; recall (6.16). Following [MB06], we say that X_i and $\boldsymbol{S}$ are *complementary* with respect to C if $-\mathrm{TMI}(C, X_i, \boldsymbol{S}) > 0$. Interestingly, in [CQF$^+$11], complementarity is referred to as the existence of *positive interaction*, or *synergy*, between X_i and $\boldsymbol{S}$ with respect to C.

Given that $\mathrm{MI}(X_i, \boldsymbol{S}) \geq 0$, a negative TMI is necessarily associated with a positive value of $\mathrm{MI}(X_i, \boldsymbol{S}|C)$. This term expresses the contribution of a candidate feature to the explanation of the class, when considering that the information contained in the already selected features is known. Following [LT06, VZCB15], we call this term *class-relevant redundancy*. This term is called *conditional redundancy* in [BPZL12]. Class-relevant redundancy is sometimes coined as the *good* redundancy since it expresses an association that contributes to the explanation of the class. In [GGNZ08], it is highlighted that *correlation does not imply redundancy* to stress that association between X_i and $\boldsymbol{S}$ is not necessarily bad.

The remaining term of the decomposition of TMI, $\texttt{MI}(X_i, \boldsymbol{S})$, measures the association between the candidate feature and the already selected features. Following [LT06], we call this term *inter-feature redundancy*. It is sometimes coined as the *bad* redundancy since it expresses the information of the candidate feature already contained in the set of already selected features.

Note that TMI takes negative values whenever the class-relevant redundancy exceeds the inter-feature redundancy, i.e. $\texttt{MI}(X_i, \boldsymbol{S}|C) > \texttt{MI}(X_i, \boldsymbol{S})$. A candidate feature X_i for which $\texttt{TMI}(C, X_i, \boldsymbol{S})$ is negative is a relevant feature, i.e. $\texttt{MI}(C, X_i|\boldsymbol{S}) \geq 0$, since $\texttt{MI}(C, X_i|\boldsymbol{S}) = \texttt{MI}(C, X_i) - \texttt{TMI}(C, X_i, \boldsymbol{S})$ by (6.16), and $\texttt{MI}(C, X_i) \geq 0$. Thus, a candidate feature may be relevant even if it is strongly associated with the already selected features. Moreover, class-relevant redundancy may turn a feature that was initially irrelevant into a relevant feature, as illustrated in Example 2. In that example, the candidate feature Y was independent of the already selected one, X, i.e. $\texttt{MI}(X_i, \boldsymbol{S}) = \texttt{MI}(Y, X) = 0$, but Y taken together with X had a positive contribution to the explanation of the class (indeed it fully explained the class), since the class-relevant redundancy is positive, i.e. $\texttt{MI}(X_i, \boldsymbol{S}|C) = \texttt{MI}(Y, X|C) = \ln(2) > 0$.

An interesting interpretation was obtained, noting that

$$-\texttt{TMI}(C, X_i, \boldsymbol{S}) = \texttt{MI}(\{X_i\} \cup \boldsymbol{S}, C) - \texttt{MI}(X_i, C) - \texttt{MI}(\boldsymbol{S}, C),$$

in [MB06]: if $-\texttt{TMI}(C, X_i, \boldsymbol{S}) > 0$, then $\texttt{MI}(\{X_i\} \cup \boldsymbol{S}, C) > \texttt{MI}(X_i, C) + \texttt{MI}(\boldsymbol{S}, C)$. Therefore, $-\texttt{TMI}(C, X_i, \boldsymbol{S}) > 0$ measures the gain resulting from considering X_i and $\boldsymbol{S}$ together, instead of considering them separately, when measuring the association with the class C.

7.2 Representative feature selection methods

The target objective functions discussed in Section 7.1 cannot be used in practice since they require joint distributions associated with $\boldsymbol{S}$, which are not known and have to be estimated. This becomes more and more difficult as the cardinality of $\boldsymbol{S}$, denoted by $|\boldsymbol{S}|$ from here on, increases.

The common solution is to use approximations, leading to different feature selection methods. For the analysis in this chapter, we selected a set of methods representative

Table 7.1: Objective functions of the representative feature selection methods, evaluated at candidate feature X_i.

Method	Objective function evaluated at X_i
MIM	$\mathrm{MI}(C, X_i)$
MIFS	$\mathrm{MI}(C, X_i) - \beta \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s)$
mRMR	$\mathrm{MI}(C, X_i) - \frac{1}{\lvert\boldsymbol{S}\rvert} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s)$
maxMIFS	$\mathrm{MI}(C, X_i) - \max_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s)$
CIFE	$\mathrm{MI}(C, X_i) - \sum_{X_s \in \boldsymbol{S}} (\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s \lvert C))$
JMI	$\mathrm{MI}(C, X_i) - \frac{1}{\lvert\boldsymbol{S}\rvert} \sum_{X_s \in \boldsymbol{S}} (\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s \lvert C))$
CMIM	$\mathrm{MI}(C, X_i) - \max_{X_s \in \boldsymbol{S}} \{\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s \lvert C)\}$
JMIM	$\mathrm{MI}(C, X_i) - \max_{X_s \in \boldsymbol{S}} \{\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s \lvert C) - \mathrm{MI}(C, X_s)\}$

of the main types of approximations to the target objective functions. In what follows, we first describe the representative methods, and discuss drawbacks resulting from their underlying approximations; we then discuss how these methods cope with the desirable properties given by Theorem 3 and Theorem 5; finally, we briefly refer to other methods proposed in the literature and how they relate to the representative ones. In this section, features are considered to be discrete for simplicity.

7.2.1 Methods and their drawbacks

The methods selected to represent the main types of approximations to the target objective functions are: MIM [Lew92], MIFS [Bat94], mRMR [PLD05], maxMIFS [POPV16], CIFE [LT06], JMI [YM99], CMIM [Fle04], and JMIM [BHS15]. These methods are listed in Table 7.1, together with their objective functions. The objective function at the first step of all methods, including mRMR and JMI (their objective functions are not well-defined for $\boldsymbol{S} = \emptyset$), is simply $\mathrm{MI}(C, X_i)$. This implies, in particular, that the first feature to be selected is the same in all methods.

The methods differ in the way their objective functions approximate the target objective functions. All methods except JMIM have objective functions that can be seen as approximations of the target OF'; the objective function of JMIM can be seen as an approximation of the target OF. The approximations made by the methods are essentially of three types: approximations that ignore both types of redundancy (inter-feature and class-relevant), approximations that ignore class-relevant redundancy but consider an approximation for the inter-feature redundancy, and approximations that consider an approximation for both the inter-feature and class-relevant redundancies.

These approximations introduce drawbacks in the feature selection process with different degrees of severity, discussed next. The various drawbacks are summarized in Table 7.2.

The simplest method is MIM. This method discards the TMI term of the target objective function $\mathtt{OF}'$, i.e.

$$\mathtt{OF}'(X_i) \approx \mathtt{MI}(C, X_i). \tag{7.4}$$

Thus, MIM ranks features accounting only for relevance effects, and completely ignores redundancy. We call the drawback introduced by this approximation *redundancy ignored*.

The methods MIFS, mRMR, and maxMIFS ignore complementarity effects, by approximating the TMI term of $\mathtt{OF}'$ through the inter-feature redundancy term only, i.e. by discarding the class-relevant redundancy –

$$\mathtt{OF}'(X_i) \approx \mathtt{MI}(C, X_i) - \mathtt{MI}(X_i, \boldsymbol{S}). \tag{7.5}$$

In this case, the TMI can no longer take negative values, since it reduces to the term $\mathtt{MI}(X_i, \boldsymbol{S})$. As discussed in Section 7.1.3, the complementarity expresses the contribution of a candidate feature to the explanation of the class, when considering that the information contained in the already selected features is known, and ignoring this contribution may lead to gross errors in the feature selection process. This drawback will be called *complementarity ignored*, and it was noted in [BPZL12]. These methods include an additional approximation, to calculate the TMI term $\mathtt{MI}(X_i, \boldsymbol{S})$, which is also used by the methods that do not ignore complementarity, and will be discussed next.

The methods that do not ignore complementarity, i.e. CIFE, JMI, CMIM, and JMIM, approximate the terms of the objective functions that depend on the set $\boldsymbol{S}$, i.e. $\mathtt{MI}(C, \boldsymbol{S})$, $\mathtt{MI}(X_i, \boldsymbol{S})$, and $\mathtt{MI}(X_i, \boldsymbol{S}|C)$, which are difficult to estimate, through a function of the already selected features X_s, $X_s \in \boldsymbol{S}$, taken individually. Considering only individual associations neglects higher order associations, i.e. between a candidate and two or more already selected features. Specifically, for CIFE, JMI, and CMIM,

$$\mathtt{OF}'(X_i) \approx \mathtt{MI}(C, X_i) - \Gamma(\mathtt{TMI}(C, X_i, \boldsymbol{S}))$$

and for JMIM,

$$\mathtt{OF}(X_i) \approx \mathtt{MI}(C, X_i) - \Gamma(\mathtt{TMI}(C, X_i, \boldsymbol{S}) - \mathtt{MI}(C, \boldsymbol{S})),$$

where Γ denotes an approximating function. This type of approximation is also used by the methods that ignore complementarity. Hereafter, we denote an already selected feature $X_s \in \boldsymbol{S}$ simply by X_s. Three types of approximating functions have been used: a sum of X_s terms scaled by a constant (MIFS and CIFE), an average of X_s terms (mRMR and JMI), and a maximization over X_s terms (maxMIFS, CMIM, and JMIM).

MIFS and CIFE approximate the TMI by a sum of X_s terms scaled by a constant. In particular, for CIFE,

$$\begin{aligned}\mathtt{TMI}(C, X_i, \boldsymbol{S}) &\approx \sum_{X_s \in \boldsymbol{S}} \mathtt{TMI}(C, X_i, X_s) = \sum_{X_s \in \boldsymbol{S}} [\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s|C)] \\ &= \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s) - \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C).\end{aligned}$$

The MIFS approximation is similar, but without the class-relevant redundancy terms, and with the sum of inter-feature redundancy terms scaled by a constant β. In both cases, a problem arises because the TMI is approximated by a sum of terms which individually have the same scale as the term they try to approximate. This results in an approximation of the TMI that can have a much larger scale than the original term. Since these terms are both redundancy terms, we will refer to this as the *redundancy overscaled* drawback. It becomes more and more severe as $\boldsymbol{S}$ grows. This drawback was also noted in [BPZL12], referring to it as the problem of not balancing the magnitudes of the relevancy and the redundancy.

Two other approximating functions were introduced to overcome the redundancy overscaled drawback. The first function, used by mRMR and JMI, replaces the TMI by an average of X_s terms. In particular, for JMI,

$$\begin{aligned}\mathtt{TMI}(C, X_i, \boldsymbol{S}) \approx& \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathtt{TMI}(C, X_i, X_s) = \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} [\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s|C)] \\ =& \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s) - \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C).\end{aligned}$$

The mRMR approximation is similar, but without the class-relevant redundancy terms. This approximation solves the overscaling problem but introduces another drawback. In fact, since $\mathrm{MI}(X_i, \boldsymbol{S}) \geq \mathrm{MI}(X_i, X_s)$, implying that $\mathrm{MI}(X_i, \boldsymbol{S}) \geq \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s)$, the approximation undervalues the inter-feature redundancy; at the same time, given that $\mathrm{MI}(X_i, \boldsymbol{S}|C) \geq \mathrm{MI}(X_i, X_s|C)$, implying $\mathrm{MI}(X_i, \boldsymbol{S}|C) \geq \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C)$, it also undervalues the class-relevant redundancy. We call this drawback *redundancy undervalued*.

The second approximating function introduced to overcome the redundancy overscaled drawback is a maximization over X_s terms. This approximation is used differently in maxMIFS and CMIM, on one side, and JMIM, on the other. Methods maxMIFS and CMIM just replace the TMI by a maximization over X_s terms. In particular, for CMIM,

$$\mathtt{TMI}(C, X_i, \boldsymbol{S}) \approx \max_{X_s \in \boldsymbol{S}} \mathtt{TMI}(C, X_i, X_s) = \max_{X_s \in \boldsymbol{S}} \left(\mathrm{MI}(X_i, X_s) - \mathrm{MI}(X_i, X_s|C)\right).$$

The maxMIFS approximation is similar, but without the class-relevant redundancy terms.

The discussion regarding the quality of the approximation is more complex in this case. We start by maxMIFS. In this case, since $\mathrm{MI}(X_i, \boldsymbol{S}) \geq \mathrm{MI}(X_i, X_s)$,

$$\mathrm{MI}(X_i, \boldsymbol{S}) \geq \max_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s) \geq \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s). \tag{7.6}$$

Thus, this approximation still undervalues inter-feature redundancy, but is clearly better than the one considering an average. Additionally, it is clearly the best possible approximation, under the restriction that only one X_s is considered.

Regarding CMIM, we first note that a relationship similar to (7.6) also holds for the class-relevant redundancy, i.e.

$$\mathrm{MI}(X_i, \boldsymbol{S}|C) \geq \max_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C) \geq \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C),$$

since $\mathrm{MI}(X_i, \boldsymbol{S}|C) \geq \mathrm{MI}(X_i, X_s|C)$. However, while it is true for the two individual terms that compose the TMI that $\mathrm{MI}(X_i, \boldsymbol{S}) \geq \max_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s)$ and $\mathrm{MI}(X_i, \boldsymbol{S}|C) \geq \max_{X_s \in \boldsymbol{S}} \mathrm{MI}(X_i, X_s|C)$, it is not true that $\mathrm{TMI}(C, X_i, \boldsymbol{S}) \geq \max_{X_s \in \boldsymbol{S}} \mathrm{TMI}(C, X_i, X_s)$. Thus, the maximization over X_s terms of CMIM is not as effective as that of maxMIFS; as a matter of fact, it becomes senseless. Moreover, applying a maximization jointly to the difference between the inter-feature and the class-relevant redundancy terms clearly favours X_s features that together with X_i have small class-relevant redundancy, i.e. a small value of $\mathrm{MI}(X_i, X_s|C)$. This goes against the initial purpose of methods that, like CMIM, introduced complementarity effects in forward feature selection methods. We call this drawback *complementarity penalized.* We now give an example that illustrates how this drawback may impact the feature selection process.

Example 3. *Assume that we have the same features as in Example 2, plus two extra features W and Z, independent of any vector containing other random variables of the set $\{W, Z, X, Y, C\}$. Moreover, consider the objective function of CMIM.*

In the first step, the objective function value is 0 for all features. We assume that W is selected first. In this case, at the second step, the value taken by the objective function is again 0 for all features. We assume that X is selected. At the third step, Y should be selected since it is fully relevant and Z is irrelevant. At this step, the objective function value at Z is 0. The objective function at Y requires more attention. Since Y is independent of the class, $\mathrm{MI}(Y, C) = 0$, *the target objective function evaluated at Y is*

$$\begin{aligned} -\mathrm{TMI}(C, Y, \{W, X\}) &= -\left[\mathrm{MI}(Y, \{W, X\}) - \mathrm{MI}(Y, \{W, X\}|C)\right] \\ &= -\left[0 - (H(Y|C) - H(Y|W, X, C))\right] = -(0 - \ln(2)) = \ln(2), \end{aligned}$$

and the objective function of CMIM evaluated at Y *is*

$$\begin{aligned}&-\max\{\text{TMI}(C,Y,W),\text{TMI}(C,Y,X)\}\\&=-\max\{\text{MI}(Y,W)-\text{MI}(Y,W|C),\text{MI}(Y,X)-\text{MI}(Y,X|C)\}\\&=-\max\{0-0,0-(H(Y|C)-H(Y|X,C))\}=-\max\{0-0,0-\ln(2)\}\\&=-\max\{0,-\ln(2)\}=0.\end{aligned}$$

This shows that, according to CMIM, both Y *and* Z *can be selected at this step, whereas* Y *should be selected first, as confirmed by the target objective function values. The problem occurs because the class-relevant redundancy* $\text{MI}(Y,X|C)$ *brings a negative contribution to the term of the maximization that involves* X*, leading to* $\text{TMI}(C,Y,X)=-ln(2)$*, thus forcing the maximum to be associated with the competing term, since* $\text{TMI}(C,Y,W)=0$*. As noted before, the maximum applied in this way penalizes the complementarity effects between* Y *and* X *that, as a result, are not considered in the objective function of candidate* Y*; contrarily, the term that corresponds to an already selected feature that has no association with* Y*, i.e. the term involving* W*, is the one that is reflected in the objective function of candidate* Y*.*

Note that since $\text{MI}(X_i,\boldsymbol{S})\geq\text{MI}(X_i,X_s)$ and $\text{MI}(X_i,\boldsymbol{S}|C)\geq\text{MI}(X_i,X_s|C)$, this approximation also undervalues both the inter-feature and the class-relevant redundancies. However, since the maximum is applied to the difference of the terms, it can no longer be concluded, as in the case of maxMIFS, that the approximation using a maximum is better than the one using an average (the case of JMI). The inter-feature redundancy term still pushes towards selecting the X_s that leads to the maximum value of $\text{MI}(X_i,X_s)$, since it contributes positively to the value inside the maximum operator; contrarily, the class-relevant redundancy term pushes towards selecting X_s features that depart from the maximum value of $\text{MI}(X_i,X_s|C)$, since it contributes negatively.

JMIM uses the approximation based on the maximization operator, like maxMIFS and CMIM. However, the maximization embraces an additional term. Specifically,

$$\text{TMI}(C,X_i,\boldsymbol{S})-\text{MI}(C,\boldsymbol{S})\approx\max_{X_s\in\boldsymbol{S}}\{\text{TMI}(C,X_i,X_s)-\text{MI}(C,X_s)\}.$$

The additional term of JMIM, i.e. $\text{MI}(C,X_s)$, tries to approximate a term of the target objective function that does not depend on X_i, i.e. $\text{MI}(C,\boldsymbol{S})$, and brings additional problems to the selection process. We call this drawback *unimportant term approximated.* Moreover, the extra term adds a negative contribution to each X_s term, favouring X_s features with small association with C, which goes against the whole purpose of the feature selection process. Additionally, JMIM inherits the drawbacks of CMIM, complementarity penalized and redundancy undervalued.

The representations of the objective functions of CMIM and JMIM in the references

Table 7.2: Drawbacks of the representative feature selection methods.

Drawback	MIM	MIFS	mRMR	maxMIFS	CIFE	JMI	CMIM	JMIM
Redundancy ignored	X							
Complementarity ignored		X	X	X				
Complementarity penalized							X	X
Redundancy undervalued			X	X		X	X	X
Unimportant term approximated								X
Redundancy overscaled		X			X			

where they were proposed originally [Fle04, BHS15] differ from the ones in Table 7.1. More concretely, their objective functions were originally formalized in terms of minimum operators:

$$\mathrm{OF}_{\mathrm{CMIM}}(X_i) = \min_{X_s \in \boldsymbol{S}} \mathrm{MI}(C, X_i|X_s); \tag{7.7}$$

$$\mathrm{OF}_{\mathrm{JMIM}}(X_i) = \min_{X_s \in \boldsymbol{S}} \left\{ \mathrm{MI}(C, X_s) + \mathrm{MI}(C, X_i|X_s) \right\}. \tag{7.8}$$

The representations in Table 7.1 result from the above ones using simple algebraic manipulation; recall (6.16). They allow a nicer and unified interpretation of the objective functions. For instance, they allow noticing much more clearly the similarities between maxMIFS and CMIM, as well as between CMIM and JMIM.

7.2.2 Properties of the methods

The drawbacks presented in Section 7.2.1 have consequences in terms of the good properties that forward feature selection methods must have, as expressed by Theorem 3 and Theorem 5: (i) the existence of meaningful bounds for the objective function, and (ii) the fact that fully relevant candidate features are the only ones that reach the maximum value of the objective function, while irrelevant and redundant features are the only ones to reach the minimum, which guarantees a perfect ordering of the features.

With a few exceptions, the approximations made by the various methods make them lose these properties.

Concerning the preservation of the bounds stated by Theorem 3, it can be shown that MIFS, mRMR, maxMIFS, and CIFE, do not preserve neither the lower bound nor the upper bound. Indeed, the objective function of CIFE is unbounded, both superiorly and inferiorly, due to the overscaled redundancy drawback. Moreover, the objective functions of MIFS, mRMR, and maxMIFS are unbounded inferiorly, due to the complementarity ignored drawback, i.e. due to the lack of the compensation provided by the class-relevant redundancy. For these methods, the upper bound of the objective function becomes $H(C)$. This bound is meaningless since it no longer expresses the uncertainty in C not explained by already selected features; in fact, it is independent of the set of already selected features, $\boldsymbol{S}$.

MIM, JMI, CMIM, and JMIM preserve one of the bounds: JMIM preserves the upper bound and the remaining methods preserve the lower bound. MIM preserves the lower bound but just because its objective function is too simplistic.

In order to see that JMI preserves the lower bound, note that, using (6.16), its objective function can also be written as

$$\mathrm{OF}_{\mathrm{JMI}}(X_i) = \frac{1}{|\boldsymbol{S}|} \sum_{X_s \in \boldsymbol{S}} \mathrm{MI}(C, X_i|X_s). \tag{7.9}$$

For any candidate feature X_i, since $\mathrm{MI}(C, X_i|X_s) \geq 0$ for all $X_s \in \boldsymbol{S}$, it follows that $\mathrm{OF}_{\mathrm{JMI}}(X_i) \geq 0$. Similarly, using (7.7), it follows immediately from the non-negativity of $\mathrm{MI}(C, X_i|X_s)$ that $\mathrm{OF}_{\mathrm{CMIM}}(X_i) \geq 0$, again for any candidate feature X_i.

To see that JMIM preserves the upper bound, note that, using (7.8), its objective function can also be written as

$$\begin{aligned} \mathrm{OF}_{\mathrm{JMIM}}(X_i) &= \min_{X_s \in \boldsymbol{S}} \{\mathrm{MI}(C, X_s) + \mathrm{MI}(C, X_i|X_s)\} \\ &= \min_{X_s \in \boldsymbol{S}} \{H(C) - H(C|X_i, X_s)\} \\ &= H(C) - \max_{X_s \in S} H(C|X_i, X_s). \end{aligned}$$

Hence, the objective function of JMIM has $H(C)$ as upper bound for any candidate feature X_i since $H(C|X_i, X_s) \geq 0$ for all $X_s \in \boldsymbol{S}$. This is the desired bound since this method has target objective function OF as reference.

Despite maintaining one of the bounds stated by Theorem 3, MIM, JMI, CMIM, and JMIM, do not preserve the bound that involves the conditional entropy $H(C|\boldsymbol{S})$. For MIM the upper bound becomes $H(C)$ which, as in the case of methods ignoring complementarity, is meaningless. For the remaining methods, the bound is lost due to the approximation that replaces the terms of the objective function that depend on set $\boldsymbol{S}$ by a function of the already selected features X_s taken individually. As for the lower bound of JMIM and the upper bounds of JMI and CMIM, they are now functions of $H(C|X_s)$. As in the case of MIFS, mRMR, and maxMIFS, the new bounds become meaningless: the upper bounds of JMI and CMIM no longer express the uncertainty in C that is not explained by the *complete* set of already selected features; and the lower bound of JMIM no longer expresses the uncertainty in C already explained by the *complete* set of already selected features.

In Section 7.3 we will illustrate the loss of bounds by the various methods.

Regarding the connections between the bounds of the objective functions and the feature types, stated by Theorem 5, these connections are lost for all methods, despite the fact that some bounds are preserved. It is no longer possible to assure that fully relevant features reach the maximum value of the objective function (when it exists) and that irrelevant and redundant features reach the minimum (when it exists). In particular, the stopping criterion is lost. We will provide several examples in Section 7.3. This is again due to the approximation that replaces the dependencies on the whole set of already selected features, $\boldsymbol{S}$, by dependencies on individual features $X_s \in \boldsymbol{S}$, which is shared by all methods.

It would be useful to have results similar to those of Theorem 5, if their validity given X_s would imply their validity given $\boldsymbol{S}$. Unfortunately, this is only true for redundant features. In fact, according to Theorem 4, a feature that is redundant given X_s will also be redundant given $\boldsymbol{S}$. The same does not hold for irrelevant and relevant features since, as discussed in Section 7.1.2, as $\boldsymbol{S}$ grows, relevant features can become irrelevant, and vice-versa. Thus, only properties concerning redundancy given X_s are worth being considered. In this respect, a weaker version of Theorem 5.3 can be proved for CMIM.

Theorem 6. *If there exists $X_s \in \boldsymbol{S}$ such that X_i is a redundant feature given $\{X_s\}$, then $\mathrm{OF}_{\mathrm{CMIM}}(X_i) = 0$, i.e., the minimum possible value taken by the objective function of CMIM is reached.*

Proof. Since X_i is a redundant feature given $\{X_s\}$, then $\mathrm{MI}(C, X_i|X_s) = 0$ by (6.3) and (6.15). As a result, $\mathrm{OF}_{\mathrm{CMIM}}(X_i) = 0$ follows from (7.7). In fact, in order for $\min_{X_s \in \boldsymbol{S}} \mathrm{MI}(C, X_i|X_s)$ to be 0, it is enough that $\mathrm{MI}(C, X_i|X_s)$ is 0 for one particular X_s, since the terms involved in the minimization are all non-negative. $\square$

Theorem 6 states that the objective function of CMIM reaches the minimum for a feature that is redundant given X_s. Note that a feature can be redundant given $\boldsymbol{S}$ but not redundant given X_s, which renders this result weaker than that of Theorem 5.3. Theorems analogous to Theorem 6 cannot be proved for the remaining methods, and we provide counter-examples in Section 7.3. In particular, the possibility to discard redundant features from the set of candidate features is lost, except for CMIM in the weaker context of Theorem 6.

To summarize, the approximations made by all methods make them lose the good properties exhibited by the target objective functions, namely the guarantee that features are correctly ordered, the existence of a stopping criterion, and the possibility to discard redundant features (here the exception is CMIM, in the weaker context of Theorem 6).

7.2.3 Other methods

We now briefly discuss other methods that have appeared in the literature, explaining why they have not been included as part of the representative methods presented previously.

MIFS-U [KC02] differs from MIFS in the estimation of the MI between the candidate feature and the class – this is a meaningless difference for the type of theoretical properties of the methods that we intend to address, in which estimation does not play a role. MIFS-ND [HBK14] considers the same reference terms as mRMR, employing a genetic algorithm to select the features, thus again not changing anything in theoretical terms. ICAP [Jak05] is similar to CIFE, while forcing the terms $\texttt{TMI}(C, X_i, X_s)$, $X_s \in \boldsymbol{S}$, to be seen as redundancy terms by only considering their contribution when they are positive (negative for the objective function). IGFS [AOA08] chooses the same candidate features in each step as JMI; and CMIM-2 [VE10] is also just the same as JMI, as its objective function is defined exactly as (7.9).

A particular type of methods that were also not considered as representative is characterized by considering similar objective functions to those of the introduced representative methods, with the difference that all MI terms are replaced by corresponding normalized MI terms. More concretely: NMIFS [ETPZ09] is an enhanced version of MIFS, MIFS-U, and mRMR; DISR [MB06] is adapted from JMI, and considers a type of normalization called symmetrical relevance; NJMIM [BHS15] is adapted from JMIM, using also symmetrical relevance. Past experiments [BPZL12, BHS15] show that such normalizations make the methods more expensive, due to associated extra computations, with no compensation in terms of performance. In fact, it is argued in the mentioned experiments that the performance of such methods is actually worse than the performance of the corresponding methods that do not use normalized MI, which should be, as added in [BPZL12], related to the additional variance introduced by the estimation of the extra normalization term.

7.3 Comparison of feature selection methods on a distributional setting

This section compares the feature selection methods using a distributional setting, based on a specific definition of class, features, and a performance metric. The setting provides an ordering for each of the methods, which is independent of specific datasets and estimation methods, and is compared with the ideal feature ordering. The aim of the setting is to illustrate how the drawbacks of the methods lead to incorrect feature ordering and to the loss of the good properties of the target objective functions.

We start by introducing a performance measure for feature selection methods that does not rely on the specificities of a fixed classifier – the *minimum Bayes risk* (MBR). We then describe the characteristics of the setting, namely the definitions of class and features, and show how the quantities required to calculate the objective functions of the methods, i.e. the various types of MI, are calculated. Finally, we present and discuss the results.

7.3.1 Minimum Bayes risk

Commonly, the performance measures used to compare forward selection methods depend on how a particular classifier performs for certain data sets. As a result, it is not clear if the obtained conclusions are exclusively explained by the characteristics of the feature selection method, or if the specificities of the classifier and/or the data under study create confounding effects. To overcome this limitation, we consider a different type of performance measure that is computed at each step of the forward selection method under consideration. Using the set of selected features until a given step, we obtain, for a fixed classifier, the associated *Bayes risk* (BR) or *total probability of misclassification* [JW07, Ch. 11]. Bayes risk is a theoretical measure in the sense that it does not rely on data but instead directly on the, assumed to be known, distributions of the involved features into consideration; see, for practical contexts where it was used, [KB04, GB00]. The *Bayes classifier*; see [HLLC08, Ch. 1]; is a classifier that defines a classification rule associated with the minimum Bayes risk, which will be our performance measure. The suitability of this measure to our setting results from the fact that it relies on the distributions of the features, and also on their class-conditional distributions.

Bayes risk and Bayes classifier For a given class C, with values on the set $\{0, 1, ..., c\}$, and a set of selected features $\boldsymbol{S}$, with support $\mathscr{S}$, a $(C, \boldsymbol{S})$-classifier g is a (Borel-)measurable function from $\mathscr{S}$ to $\{0, 1, ..., c\}$, and $g(\boldsymbol{s})$ denotes the value of C to which the observation $\boldsymbol{s}$ is assigned by the classifier. Then, the Bayes risk of the $(C, \boldsymbol{S})$-classifier g is given by

$$\mathrm{BR}(C, \boldsymbol{S}, g) = P(g(\boldsymbol{S}) \neq C) = \sum_{j=0}^{c} P(g(\boldsymbol{S}) \neq j | C = j) P(C = j).$$

The proposed performance evaluation measure consists of the minimum possible value of the BR, called *minimum Bayes risk* (MBR). Thus, for a given class C and a set of selected features $\boldsymbol{S}$, the associated minimum Bayes risk, $\mathrm{MBR}(C, \boldsymbol{S})$, is given by:

$$\mathrm{MBR}(C, \boldsymbol{S}) = \min_{g} \mathrm{BR}(C, \boldsymbol{S}, g).$$

The minimum Bayes risk corresponds to the Bayes risk of the so-called *Bayes classifier*. The $(C, \boldsymbol{S})$ Bayes classifier assigns an object $\boldsymbol{s} \in \mathscr{S}$ to the value that C is most likely to take given that $\boldsymbol{S} = \boldsymbol{s}$. That is, the $(C, \boldsymbol{S})$ Bayes classifier g is such that

$$\begin{aligned} g(\boldsymbol{s}) &= \operatorname*{argmax}_{j \in \{0,1,\ldots,c\}} P(C = j | \boldsymbol{S} = \boldsymbol{s}) \\ &= \operatorname*{argmax}_{j \in \{0,1,\ldots,c\}} P(C = j) f_{\boldsymbol{S}|C=j}(\boldsymbol{s}). \end{aligned}$$

Note that, in particular, when there are two possible values for the class, i.e. $c = 1$, the Bayes classifier g is such that:

$$g(\boldsymbol{s}) = 1 \iff \frac{f_{\boldsymbol{S}|C=0}(\boldsymbol{s})}{f_{\boldsymbol{S}|C=1}(\boldsymbol{s})} \leq \frac{P(C=1)}{P(C=0)}. \tag{7.10}$$

see [JW07, Ch. 11].

Properties of the minimum Bayes risk We now discuss a few properties of the minimum Bayes risk, the proposed performance evaluation criterion. In the following, measurable should be read as Borel-measurable.

Theorem 7. *If C is a measurable function of $\boldsymbol{S}$, then* $\mathrm{MBR}(C, \boldsymbol{S}) = 0$.

Proof. Let g be the measurable function such that $C = g(\boldsymbol{S})$. As $C = g(\boldsymbol{S})$, it follows that $\mathrm{BR}(C, \boldsymbol{S}, g) = P(g(\boldsymbol{S}) \neq C) = 0$. As $\mathrm{MBR}(\mathrm{C}, \boldsymbol{S})$ is non-negative and $\mathrm{MBR}(\mathrm{C}, \boldsymbol{S}) \leq \mathrm{BR}(\mathrm{C}, \boldsymbol{S}, \mathrm{g})$, we conclude that $\mathrm{MBR}(C, \boldsymbol{S}) = 0$, as intended. ☐

Note that C being a measurable function of $\boldsymbol{S}$ is equivalent to saying that features in $\boldsymbol{S}$ fully explain the class.

Theorem 8. *If X_i is a measurable function of $\boldsymbol{S}$, then* $\mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\}) = \mathrm{MBR}(C, \boldsymbol{S})$.

Proof. Let ξ be the measurable function such that $X_i = \xi(\boldsymbol{S})$, and g be the $(C, \boldsymbol{S} \cup \{X_i\})$ Bayes classifier, so that, in particular, $\mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\}) = \mathrm{BR}(C, \boldsymbol{S} \cup \{X_i\}, g)$. Let g' be the $(C, \boldsymbol{S})$ classifier such that, given an observation $\boldsymbol{s} \in \mathscr{S}$, $g'(\boldsymbol{s}) = j$ when $g(\boldsymbol{s}, \xi(\boldsymbol{s})) = j$. Then $\mathrm{BR}(C, \boldsymbol{S} \cup \{X_i\}, g) = \mathrm{BR}(C, \boldsymbol{S}, g')$. As a consequence, by transitivity, $\mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\}) = \mathrm{BR}(C, \boldsymbol{S}, g')$. This implies that $\mathrm{MBR}(C, \boldsymbol{S}) \leq \mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\})$.

In turn, it always holds that $\mathrm{MBR}(C, \boldsymbol{S}) \geq \mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\})$ since $\boldsymbol{S} \subset \boldsymbol{S} \cup \{X_i\}$. Therefore, $\mathrm{MBR}(C, \boldsymbol{S} \cup \{X_i\}) = \mathrm{MBR}(C, \boldsymbol{S})$. $\square$

Note that X_i being a measurable function of $\boldsymbol{S}$ is equivalent to saying that X_i is redundant given $\boldsymbol{S}$.

7.3.2 Setting description

We now describe the distributional setting used to illustrate the various deficiencies of the feature selection methods. The class chosen for this setting is a generalization of the one proposed in [POPV16], which was based on the scenario introduced in [KC02] and later used in [HLLC08]. It is defined as

$$C_k = \begin{cases} 0, & X + kY < 0 \\ 1, & X + kY \geq 0 \end{cases}, \tag{7.11}$$

where X and Y are independent features with standard normal distributions and $k \in \,]0, +\infty[$.

According to the discussion in Section 7.1, our scenario includes fully relevant, relevant, redundant, and irrelevant features. Specifically, our features are X, $X - k'Y$, $k' > 0$, Z and X_{disc}. X and $X - k'Y$ were chosen as relevant features that, taken together, fully explain the class. As irrelevant feature, we chose Z, independent of X and Y, which for simplicity is considered to follow a Bernoulli distribution with success probability 1/2. Finally, as redundant feature we chose

$$X_{\text{disc}} = \begin{cases} 0, & X < 0 \\ 1, & X \geq 0 \end{cases}, \tag{7.12}$$

The first selected feature is the candidate X_i that has the largest value of $\mathrm{MI}(C_k, X_i)$. The possible candidates are X, $X - k'Y$, and X_{disc}, which are the initially relevant features. Z is an irrelevant feature and, therefore, will not be selected first. We want X to be selected first to assure that, at the second step of the algorithm, there will be, as candidates, one fully relevant, one redundant, and one irrelevant feature. This provides an extreme scenario, where the relevancy level of the relevant feature is the maximum possible, making a wrong decision the hardest to occur. We next discuss the conditions for selecting X before $X - k'Y$ and before X_{disc}.

X is selected before $X - k'Y$ if $\mathrm{MI}(C_k, X) > \mathrm{MI}(C_k, X - k'Y)$, which is equivalent to the condition

$$\arctan k < (\pi - \arctan k') - \arctan k,$$

where the left term represents the angle between the lines $X = 0$ and $X + kY = 0$ and the right term represents the angle between the lines $X - k'Y = 0$ and $X + kY = 0$, in the context of the two-dimensional space defined by the pair of orthogonal vectors (X, Y). The condition can be written in terms of k as

$$k < \tan\left(\frac{\pi - \arctan k'}{2}\right). \tag{7.13}$$

Feature X_{disc} is never selected before X since $\text{MI}(C_k, X_{\text{disc}}) \leq \text{MI}(C_k, X)$ for all $k > 0$. To see this, note that this inequality can be written, using (6.3), as $H(C_k) - H(C_k|X_{\text{disc}}) \leq H(C_k) - H(C_k|X)$, which is equivalent to $H(C_k|X_{\text{disc}}) \geq H(C_k|X)$. This is equivalent to $H(C_k|X_{\text{disc}}) \geq H(C_k|X_{\text{disc}}, X)$, which holds by (6.15). In turn, $H(C_k|X) = H(C_k|X_{\text{disc}}, X)$ is equivalent to $\text{MI}(C_k, X_{\text{disc}}|X) = 0$ by (6.11). Finally, since X_{disc} is redundant given $\{X\}$, equations (6.11) and (6.15) can be used to verify that $\text{MI}(C_k, X_{\text{disc}}|X) = 0$.

In view of the above discussion, the ideal feature ordering coming out of the distributional setting is X in first place and $X - k'Y$ in second place. Ideally, the feature selection method should stop at this step, since a fully relevant feature has been added. However, further steps need to be considered, since actual methods do not preserve the stopping criterion, as noted in Section 7.2.2. Then, at the third step, the remaining features, Z and X_{disc}, must be equally likely to be selected since, according to Theorems 5.2 and 5.3, both their target objective functions reach the lower bound.

7.3.3 Required quantities

In order to be able to determine the order in which the features are selected by the different methods, we have to derive expressions, depending on k and k', needed for evaluating the corresponding objective functions. We need: the MI between each candidate feature and the class, the MI between different pairs of candidate features, and the class-conditional MI between pairs of candidate features. The computation of these quantities require obtaining the univariate entropies of the candidate features and of the class. The derivations of such expressions are provided in Appendix A.1.1 and their final forms are available in Tables 7.3 to 7.6.

Univariate entropies. We start with a summary of the univariate entropies of the different features and of the class presented in Table 7.3. The corresponding derivations can be found in Appendix A.1.1.

Table 7.3: Entropies of the class, C_k, and the input features.

	C_k	X	$X - k'Y$	Z	X_{disc}
Entropy	$\ln(2)$	$\frac{1}{2}\ln(2\pi e)$	$\frac{1}{2}\ln(2\pi e(1 + k^{'2}))$	$\ln(2)$	$\ln(2)$

Table 7.4: MI between each input feature and the class, C_k.

A	$\text{MI}(C_k, A)$
X	$\frac{1}{2}\ln(2\pi e) - \frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X\vert C_k=j}(u)\ln f_{X\vert C_k=j}(u)du$
$X - k'Y$	$\frac{1}{2}\ln(2\pi e(1 + k^{'2})) - \frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X-k'Y\vert C_k=j}(u)\ln f_{X-k'Y\vert C_k=j}(u)du$ [a]
Z	0
X_{disc}	$2\ln(2) + \frac{\arctan k}{\pi}\ln(\frac{\arctan k}{2\pi}) + (1 - \frac{\arctan k}{\pi})\ln(\frac{1}{2} - \frac{\arctan k}{2\pi})$ [b]

[a] $X|C_k = j \sim \text{SN}(0, 1, \frac{(-1)^{j+1}}{k})$, $j = 0, 1$.

[b] $X - k'Y|C_k = j \sim \text{SN}(0, \sqrt{1 + k'^2}, (-1)^{j+1}(\frac{1-kk'}{k+k'}))$, $j = 0, 1$.

MI between input features and the class. As for the MI between input features and the class, they are provided in Table 7.4. The corresponding derivations can be found in Appendix A.1.2.

It must be added that the notation $W \sim \text{SN}(\mu, \sigma, \alpha)$ (where $\mu \in \mathbb{R}$, $\sigma > 0$, and $\alpha \in \mathbb{R}$), means that the random variable W follows a skew-normal distribution, so that it has probability density function [Azz86]

$$f_W(w) = \frac{2}{\sigma}\phi(\frac{w - \mu}{\sigma})\Phi(\frac{\alpha(w - \mu)}{\sigma}), \quad w \in \mathbb{R}, \tag{7.14}$$

where $\Phi(z)$ denotes the value of the standard normal distribution function at point z, while $\phi(z)$ denotes the probability density function, for the same distribution, also at z.

MI between pairs of input features. As for the MI between the different pairs of input features, they are provided in Table 7.5. The corresponding derivations can be found in Appendix A.1.3.

Class-conditional MI between pairs of input features. As for the class-conditional MI between the different pairs of input features, they are provided in Table 7.6. The corresponding derivations can be found in Appendix A.1.4.

Table 7.5: MI between pairs of input features.

A	B	$\mathrm{MI}(\cdot,\cdot)$
X	$X-k'Y$	$\frac{1}{2}\ln(1+\frac{1}{k'^2})$
X	X_{disc}	$\ln(2)$
$X-k'Y$	X_{disc}	$\frac{1}{2}\ln(2\pi e)-\frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X\|C_{k'}=j}(u)\ln f_{X\|C_{k'}=j}(u)du$ [a]
Z	B	$0, \quad B\in\{X, X-k'Y, X_{\text{disc}}\}$

[a] $X|C_{k'}=j \sim \mathrm{SN}(0,1,\frac{(-1)^{j+1}}{k'}),\ j=0,1.$

Table 7.6: Class-conditional MI between pairs of input features.

A	B	$\mathrm{MI}(\cdot,\cdot\|C_k)$
X	$X-k'Y$	$\frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X\|C_k=j}(u)\ln f_{X\|C_k=j}(u)du+$ $\frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X-k'Y\|C_k=j}(u)\ln f_{X-k'Y\|C_k=j}(u)du-$ $(1+\ln\pi+\ln k')$ [a,b]
X	X_{disc}	$-\frac{\arctan k}{\pi}\ln(\frac{\arctan k}{\pi})-(1-\frac{\arctan k}{\pi})\ln(1-\frac{\arctan k}{\pi})$
$X-k'Y$	X_{disc}	$\frac{1}{2}\sum_{j=0}^{1}\int_{\mathbb{R}} f_{X-k'Y\|C_k=j}(u)\ln f_{X-k'Y\|C_k=j}(u)du - h(X-k'Y\|X_{\text{disc}},C_k)$ [b,c]
Z	B	$0, \quad B\in\{X, X-k'Y, X_{\text{disc}}\}$

[a] $X|C_{k'}=j \sim \mathrm{SN}(0,1,\frac{(-1)^{j+1}}{k'}),\ j=0,1.$

[b] $X-k'Y|C_k=j \sim \mathrm{SN}(0,\sqrt{1+k'^2},(-1)^{j+1}(\frac{1-kk'}{k+k'})),\ j=0,1.$ [c] $h(X-k'Y|X_{\text{disc}},C_k)$ in (A.5).

7.3.4 Applying the different feature selection methods

We now present the results of applying the various feature selection methods to the distributional setting. The feature ordering will be discussed for different values of k and k'. Taking the objectives of this study into consideration, for fixed k', the most interesting case is that where X alone leaves the largest possible amount of information undetermined about the class; this leads to $X-k'Y$ having the most importance in the explanation of the class, making the error of not choosing it after X the worst possible. According to the performance metric introduced in Section 7.3.1, we want to choose a value of k that leads to a large MBR when X is the only selected feature, $\mathrm{MBR}(C_k,\{X\})$, which is given by (see Appendix A.2):

$$\mathrm{MBR}(C_k,\{X\}) = \frac{\arctan k}{\pi}. \tag{7.15}$$

Since $\mathrm{MBR}(C_k,\{X\})$ is an increasing function of k, we want k to be as large as possible, under the restriction (7.13). We consider $k=\tan\big((\pi-\arctan k'-10^{-6})/2\big)$.

Given that, in our setting, the features X and $X - k'Y$ fully explain the class, so that, according to Theorem 7, $\text{MBR}(C_k, \{X, X - k'Y\}) = 0$, it makes sense to take the MBR based on the first two selected features as performance measure for characterizing each forward feature selection method. This measure is denoted by MBR_2.

We will carry out two different studies. In the first one, we concentrate on the methods that ignore complementarity; recall (7.5); i.e. MIFS, mRMR, and maxMIFS, and study the feature ordering as a function of k'. The purpose of this study is, in fact, to highlight the consequences of ignoring complementarity. In the second study, we compare the feature ordering of all methods under analysis, for fixed k'. The goal is to provide examples showing wrong decisions made by the various methods, highlighting the corresponding drawbacks.

The consequences of ignoring complementarity We start by focusing on the consequences of ignoring complementarity, as a function of k'; thus, we concentrate on methods MIFS ($\beta = 1$), mRMR, and maxMIFS. The motivation for scanning k' is that it provides different levels of association between the already selected feature X and the candidate feature $X - k'Y$. For small values of k', X and $X - k'Y$ are strongly associated, and the level of association decreases as k' increases.

Scanning k' from 0 to $+\infty$ defines three regions, each corresponding to a specific feature ordering. This is shown in Figure 7.1, where k' was scanned with a step size of 0.01, starting at 0.01. To complete the discussion, we include the objective function values in the second step of the algorithms in Figure 7.2, and the corresponding MBR_2 values in Figure 7.3. Note that, at this step, for each candidate feature, the objective function takes the same value for all methods. As a result, all methods will select the same feature and, in particular, the MBR_2 is the same for all methods.

For small values of k', smaller than 0.565 for MIFS and maxMIFS, and than 0.575 for mRMR – region (a) – the feature ordering is X, Z, X_{disc}, $X - k'Y$. In this region, $X - k'Y$ is chosen last due to a large inter-feature redundancy with X. As shown in Figure 7.2, in this region, the objective function values of $X - k'Y$ and X_{disc} at the second step are negative and smaller than that of Z, which explains why Z is selected in second place. At the third step, the objective function values of $X - k'Y$ and X_{disc} are exactly the same as in the second step for MIFS and maxMIFS, and only slightly different for mRMR. Thus, in this region, the objective functions of $X - k'Y$ are more negative than those of X_{disc}, which explains why X_{disc} is selected in third place.

For intermediate values of k', smaller than 2.115, and larger than 0.565 for MIFS and maxMIFS and than 0.575 for mRMR – region (b) – the feature ordering is X, Z, $X - k'Y$, X_{disc}. In this region, the objective functions of $X - k'Y$ are larger than those of X_{disc}, but smaller than those of Z.

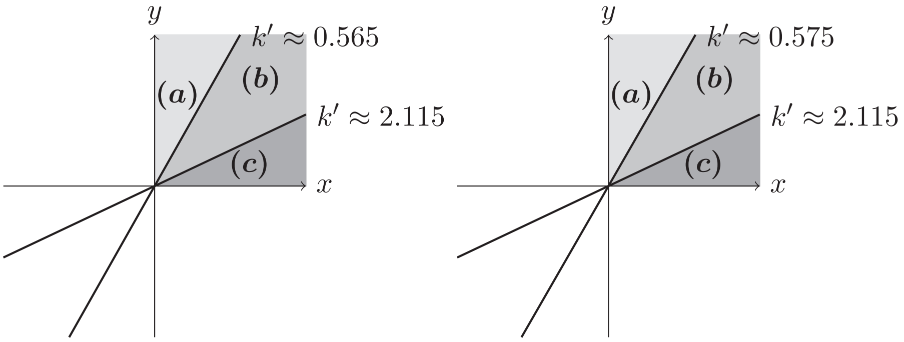


Figure 7.1: Regions associated with a specific ordering of the features, defined by the values of k'. In this representation, k' is defined through the line $x - k'y = 0$. For region (a), the ordering is $\{X, Z, X_{\text{disc}}, X - k'Y\}$; for (b), it is $\{X, Z, X - k'Y, X_{\text{disc}}\}$; and for (c), the ordering is already correct since X is chosen first and $X - k'Y$ second. For methods MIFS and maxMIFS (mRMR), represented in the left (right), (a) is associated with $0 < k' < 0.575$ $(0 < k' < 0.565)$ and (b) with $0.575 < k' < 2.115$ $(0.565 < k' < 2.115)$. Region (c) is associated with $k' > 2.115$ for the three methods.

For large values of k', larger than 2.115 – region (c) – the correct feature ordering is achieved since $X - k'Y$ is selected in second place. Note that in this region, there are two possible orderings for Z and X_{disc}, but this issue is not relevant for our discussion.

The problem of these methods in regions (a) and (b) is due to the lack of the class-relevant redundancy term in their objective functions, which expresses the complementarity effects. In fact, the association between X and $X - k'Y$, as measured by $\text{MI}(X - k'Y, X)$, grows significantly as k' approaches 0, but so does the class-relevant redundancy, which is given by $\text{MI}(X - k'Y, X|C_k)$; recall Tables 7.5 and 7.6, respectively. Ignoring the compensation given by the latter term leads to objective function values that can take negative values; this explains why the lower bound of 0 from Theorem 3, associated with the target objective function, is lost for these methods. Also, in contradiction with the good properties of the target objective function, the objective functions of these methods do not take the same (minimum) values at X_{disc} and Z. The MBR_2 values of these methods (see Figure 7.3) confirm that the performance is very poor in regions (a) and (b): it is above 0.4 in region (a) and above 0.3 in region (b). These results show that ignoring complementarity is a severe drawback that can lead to gross errors in the feature selection process.

Figure 7.2 also shows that results analogous of Theorem 6 do not hold for MIFS, mRMR, and maxMIFS. In fact, the objective function at X_{disc}, a redundant feature, is not necessarily the minimum; in particular, this happens for small values of k', where the objective function at $X - k'Y$ takes lower values than that at X_{disc}.

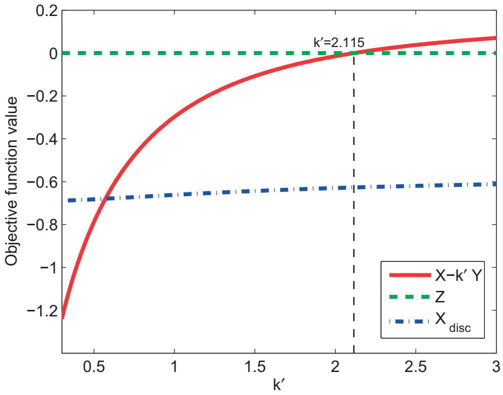


Figure 7.2: Evaluation of the objective function for the different candidate features in the second step of the algorithms (MIFS, mRMR, and maxMIFS) depending on the value of k'.

Feature ordering for all representative methods We now compare the feature ordering of all methods, for fixed k and k'. We want that a mistake in the selection of the second feature becomes particularly severe, so that we use k and k' values that maximize $\mathrm{MBR}(C_k, \{X\})$ – worst possible case. Per (7.15) we need to maximize k and per (7.13) we need to minimize k'. We choose for k' the first value of the grid used in the context of Figure 7.1, i.e. $k' = 0.01$. In this case, $k = \tan(\frac{\pi - \arctan k' - 10^{-6}}{2}) = 199.985$ and $\mathrm{MBR}(C_k, \{X\}) \approx 0.498$. Recall that, since (7.13) holds, and therefore $\mathrm{MI}(C_k, X) > \mathrm{MI}(C_k, X - k'Y)$, X is always selected in first place.

Table 7.7 shows the feature ordering and the associated values of MBR_2. The ordering of features relates to the concrete values of the terms that compose the objective functions. These are provided in Table 7.8, which contains the values of MI between each candidate feature and the class; Table 7.9, which contains the values of MI between the different features; and Table 7.10, which contains the values of the class-conditional MI between the different input features. Note that, in Table 7.8, $\mathrm{MI}(C_k, X)$, $\mathrm{MI}(C_k, X - k'Y)$, and $\mathrm{MI}(C_k, X_{\text{disc}})$ are all shown as taking approximately the value 0, but actually $\mathrm{MI}(C_k, X)$ is the largest one.

Table 7.7 shows that all methods, except MIFS, mRMR, and maxMIFS, achieve an MBR_2 of 0. However, the third step of the algorithm is only completely correct for CMIM. In fact, it should be equally likely to choose Z or X_{disc}, but CIFE, JMI, JMIM, and MIM select X_{disc} first.

MIM suffers from redundancy ignored drawback. The fact that the selection is correct at the first two steps of the feature selection process is meaningless; it only happens because $\mathrm{MI}(C_k, X - k'Y)$ is slightly larger than $\mathrm{MI}(C_k, X_{\text{disc}})$.

The methods that ignore complementarity, i.e. MIFS, mRMR, and maxMIFS, fail at the

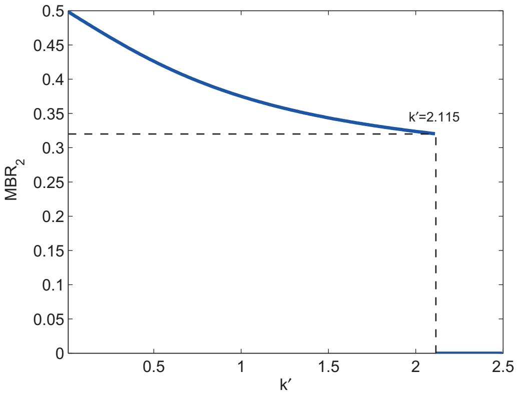


Figure 7.3: MBR_2 for the different algorithms (MIFS, mRMR, and maxMIFS) depending on the value of k'. For $k' < 2.115$, $\text{MBR}_2 = \frac{\pi - \arctan k' - 10^{-6}}{2\pi}$ using $k = \tan\left((\pi - \arctan k' - 10^{-6})/2\right)$ and (7.15); for $k' > 2.115$, the right second feature is chosen, $X - k'Y$, so that $\text{MBR}_2 = 0$.

Table 7.7: Feature ordering and corresponding MBR_2, for $k = 199.985$ and $k' = 0.01$.

Methods	Order of feature selection				MBR_2
MIM	X	$X - k'Y$	X_{disc}	Z	0
MIFS ($\beta = 1$)	X	Z	X_{disc}	$X - k'Y$	0.498
mRMR	X	Z	X_{disc}	$X - k'Y$	0.498
maxMIFS	X	Z	X_{disc}	$X - k'Y$	0.498
CIFE	X	$X - k'Y$	X_{disc}	Z	0
JMI	X	$X - k'Y$	X_{disc}	Z	0
CMIM	X	$X - k'Y$	Z/X_{disc}	X_{disc}/Z	0
JMIM	X	$X - k'Y$	X_{disc}	Z	0

second step of the feature selection process, by not selecting $X - k'Y$. For all methods, the objective function is 0 for Z, $\text{MI}(C_k, X - k'Y) - \text{MI}(X - k'Y, X) = -4.605$ for $X - k'Y$, and $\text{MI}(C_k, X_{\text{disc}}) - \text{MI}(X_{\text{disc}}, X) = -0.693$ for X_{disc}, which explains why Z is selected at this step. Adding the class-relevant redundancy term to the objective functions, would make them take the value $\ln(2)$ for $X - k'Y$ and 0 for X_{disc}, leading to the selection of $X - k'Y$. In fact, the class-relevant redundancy term is $\text{MI}(X - k'Y, X|C_k) = 5.298$ for $X - k'Y$, and $\text{MI}(X_{\text{disc}}, X|C_k) = 0.693$ for X_{disc}. Note that $\ln(2)$ is precisely the maximum of the target objective function, which is achieved for fully relevant features (the case of $X - k'Y$), and the minimum is 0, achieved by irrelevant and redundant features (the cases of Z and X_{disc}). Thus, accounting for the class-relevant redundancy compensates the potentially large negative values associated with the inter-feature redundancy.

With the exception of CMIM, the methods that do not ignore complementarity, fail at

Table 7.8: MI between the class and each input feature, for $k = 199.985$ and $k' = 0.01$.

	X	$X - k'Y$	Z	X_{disc}
$\text{MI}(\cdot, C_k)$	≈ 0	≈ 0	0	≈ 0

Table 7.9: MI between pairs of input features, for $k = 199.985$ and $k' = 0.01$.

$\text{MI}(\cdot, \cdot)$	X	$X - k'Y$	Z
$X - k'Y$	4.605		
Z	0	0	
X_{disc}	0.693	0.686	0

the third step of the feature selection process since their objective functions take larger values at X_{disc} than at Z, implying that X_{disc} is preferred over Z as shown in Table 7.7.

As discussed in Section 7.2, CIFE suffers from overscaled redundancy drawback. At the third step of the feature selection process, after selecting X and $X - k'Y$, the objective function for candidate feature X_i is

$$\text{MI}(C_k, X_i) - \text{MI}(X_i, X) + \text{MI}(X_i, X|C_k) - \text{MI}(X_i, X - k'Y) + \text{MI}(X_i, X - k'Y|C_k), \tag{7.16}$$

while the associated target objective function is

$$\text{MI}(C_k, X_i) - \text{MI}(X_i, \{X, X - k'Y\}) + \text{MI}(X_i, \{X, X - k'Y\}|C_k). \tag{7.17}$$

Both objective functions take the value 0 for the candidate feature Z. For the candidate X_{disc}, the target objective function (7.17) can be written as

$$\text{MI}(C_k, X_{\text{disc}}) - \text{MI}(X_{\text{disc}}, X) + \text{MI}(X_{\text{disc}}, X|C_k), \tag{7.18}$$

given that $\text{MI}(X_{\text{disc}}, \{X, X - k'Y\}) = \text{MI}(X_{\text{disc}}, X)$ and $\text{MI}(X_{\text{disc}}, \{X, X - k'Y\}|C_k) =$

Table 7.10: Class-conditional MI between pairs of input features, for $k = 199.985$ and $k' = 0.01$.

$\text{MI}(\cdot, \cdot|C_k)$	X	$X - k'Y$	Z
$X - k'Y$	5.298		
Z	0	0	
X_{disc}	0.693	0.689	0

$\text{MI}(X_{\text{disc}}, X|C_k)$. Concerning the first condition, note that $\text{MI}(X_i, \boldsymbol{S}) = \text{MI}(X_{\text{disc}}, \{X, X-k'Y\}) = H(X_{\text{disc}}) - H(X_{\text{disc}}|X, X-k'Y) = H(X_{\text{disc}}) - H(X_{\text{disc}}|X) = \text{MI}(X_{\text{disc}}, X)$, since $H(X_{\text{disc}}|X) = 0$ implies that $H(X_{\text{disc}}|X, X-k'Y) = 0$ also, by (6.15); a similar reasoning can be used to show that the second condition also holds.

Thus, in the case of X_{disc}, we see that, when comparing the objective function of CIFE, given by (7.16), with the target objective function, given by (7.18), CIFE includes an extra part with two terms, $-\text{MI}(X_i, X-k'Y) + \text{MI}(X_i, X-k'Y|C_k)$, which is responsible for increasing the redundancy scale. In our case, the extra term takes the value $-\text{MI}(X_{\text{disc}}, X-k'Y) + \text{MI}(X_{\text{disc}}, X-k'Y|C_k) = -0.686 + 0.689 = 0.003$; recall Tables 7.9 and 7.10. This is exactly the value of the objective function at X_{disc}, since the remaining terms sum to 0, which explains why X_{disc} is selected before Z. To see that the remaining terms sum to 0, note that these terms correspond exactly to the evaluation of the target objective function OF', and recall that the target objective function value must be 0 for a redundant feature.

The overscaling effect is relatively modest in this example but, clearly, the problem gets worse as $\boldsymbol{S}$ increases, since more terms are added to the objective function. We also note that, while in this case the objective function has been overestimated, it could have equally been underestimated. This fact together with the overscaling problem is what makes the objective function of CIFE not bounded, neither from below nor from above.

JMI tried to overcome the problem of CIFE by introducing the scaling factor $1/|\boldsymbol{S}|$ in the TMI approximation. However, as discussed in Section 7.2, this leads to redundancy undervalued drawback. At the third step of the feature selection process, when X and $X-k'Y$ have been selected, the objective function of JMI is

$$\text{MI}(C_k, X_i) - \frac{1}{2}\text{MI}(X_i, X) + \frac{1}{2}\text{MI}(X_i, X|C_k) - \frac{1}{2}\text{MI}(X_i, X-k'Y) + \frac{1}{2}\text{MI}(X_i, X-k'Y|C_k)$$

for the candidate X_i. Its value equals 0 for the candidate Z, but for candidate X_{disc} it equals $0 - 0.5 \times 0.693 + 0.5 \times 0.693 - 0.5 \times 0.689 + 0.5 \times 0.686 = 0.0015$, which explains why X_{disc} is selected before Z.

This results directly from the undervaluing of the terms $\text{MI}(X_i, \boldsymbol{S})$ and $\text{MI}(X_i, \boldsymbol{S}|C)$ of the target objective function at $X_i = X_{\text{disc}}$. In fact, $\text{MI}(X_i, \boldsymbol{S}) = \text{MI}(X_{\text{disc}}, X) = 0.693$, but JMI approximates it by a smaller value, i.e. $\frac{1}{2}\text{MI}(X_{\text{disc}}, X) + \frac{1}{2}\text{MI}(X_{\text{disc}}, X-k'Y) = \frac{1}{2} \times 0.693 + \frac{1}{2} \times 0.686 = 0.6895$. Similarly, $\text{MI}(X_{\text{disc}}, \boldsymbol{S}|C_k) = \text{MI}(X_{\text{disc}}, X|C_k) = 0.693$, but again JMI approximates it by a smaller value, i.e. $\frac{1}{2}\text{MI}(X_{\text{disc}}, X|C_k) + \frac{1}{2}\text{MI}(X_{\text{disc}}, X-k'Y|C_k) = \frac{1}{2} \times 0.693 + \frac{1}{2} \times 0.689 = 0.691$.

JMIM introduced an additional term in the objective function which, as discussed in Section 7.2, is unimportant and may lead to confusion in the selection process – unimportant term approximated drawback. At the third step of the selection process, when X and $X - k'Y$ have been selected, the objective function of JMIM is

$$\begin{aligned}\mathrm{MI}(C_k, X_i) - \max\{&\mathrm{MI}(X_i, X) - \mathrm{MI}(X_i, X|C_k) - \mathrm{MI}(C_k, X),\\ &\mathrm{MI}(X_i, X - k'Y) - \mathrm{MI}(X_i, X - k'Y|C_k) - \mathrm{MI}(C_k, X - k'Y)\},\end{aligned}$$

for candidate feature X_i. In this case, the objective function for candidate feature Z is $0 - \max\{0 - 0 - \mathrm{MI}(C_k, X), 0 - 0 - \mathrm{MI}(C_k, X - k'Y)\}$, and for candidate X_{disc} it is $0 - \max\{0.693 - 0.693 - \mathrm{MI}(C_k, X), 0.686 - 0.689 - \mathrm{MI}(C_k, X - k'Y)\}$. We first note that $\mathrm{MI}(C_k, X)$ and $\mathrm{MI}(C_k, X - k'Y)$ are both approximately 0, while $\mathrm{MI}(C_k, \{X, X - k'Y\})$, the quantity they try to approximate, takes the value $\ln(2)$, since it is $H(C_k) - H(C_k|X, X - k'Y) = H(C_k)$. Since, per design of our experiment $\mathrm{MI}(C_k, X) > \mathrm{MI}(C_k, X - k'Y)$, it turns out that the objective function of Z equals $\mathrm{MI}(C_k, X - k'Y)$, and that of X_{disc} equals $\mathrm{MI}(C_k, X)$, leading to the selection of X_{disc}. There are two observations that should be pointed out. First, contrarily to the previous cases of CIFE and JMI, the objective function for Z takes a value that is no longer according to the corresponding target objective function, which in this case should be $\mathrm{MI}(C_k, \{X, X - k'Y\}) = \ln(2)$. Second, the choice between the two features, Z and X_{disc}, is being done by two terms, $\mathrm{MI}(C_k, X)$ and $\mathrm{MI}(C_k, X - k'Y)$, that try to approximate a term that does not depend on the candidate features, $\mathrm{MI}(C, \boldsymbol{S})$, and therefore should take the same value for both features and not become a deciding factor.

The results regarding MIM, CIFE, JMI, and JMIM provide counter-examples showing that theorems analogous to Theorem 6 do not hold for these methods. Indeed, in all cases, the objective function at the third step of the algorithms, after X and $X - k'Y$ have been selected, at X_{disc}, a redundant feature, takes values different from the minimum of the corresponding objective function.

CMIM is the only method that performs correctly in the distributional setting. At the third step of the feature selection process, the objective function is 0 for both Z and X_{disc}. The latter result can be obtained from Theorem 6, since X_{disc} is redundant given $\{X\}$. This can be confirmed numerically. The objective function of CMIM at the third step of the feature selection process is

$$\mathrm{MI}(C_k, X_i) - \max\{\mathrm{MI}(X_i, X) - \mathrm{MI}(X_i, X|C_k), \mathrm{MI}(X_i, X - k'Y) - \mathrm{MI}(X_i, X - k'Y|C_k)\},$$

for candidate feature X_i. In this case, the objective function for candidate feature Z is 0, and the same holds for X_{disc} since $0 - \max\{0.693 - 0.693, 0.686 - 0.689\} = 0$. Note however that this does not mean that CMIM always performs correctly. As discussed in Section 7.2.1, CMIM suffers from the problems of redundancy undervalued and complementarity penalized, and Example 3 provides a case where CMIM decides incorrectly.

7.4 Conclusion

We have carried out an evaluation and a comparison of forward feature selection methods based on mutual information. For this evaluation we selected methods representative of all types of feature selection methods proposed in the literature, namely MIM, MIFS, mRMR, maxMIFS, CIFE, JMI, CMIM, and JMIM. The evaluation was carried out theoretically, i.e. independently of the specificities of datasets and classifiers; thus, our results establish unequivocally the relative merits of the methods.

Forward feature selection methods iterate step-by-step and select one feature at each step, among the set of candidate features: the one that maximizes an objective function expressing the contribution each candidate feature to the explanation of the class. In our case, the mutual information (MI) is used as the measure of association between the class and the features. Specifically, the candidate feature selected at each step is the one that maximizes the MI between the class and the set formed by the candidate feature and the already selected features.

Our theoretical evaluation is grounded on target objective functions that the methods try to approximate and on a categorization features according to their contribution to the explanation of the class. The features are categorized as irrelevant, redundant, relevant, and fully relevant. This categorization has two novelties regarding previous works: first, we introduce the important category of fully relevant features; second, we separate non-relevant features in two categories of irrelevant and redundant features. Fully relevant features are features that fully explain the class and, therefore, its detection can be used as a stopping criterion of the feature selection process. Irrelevant and redundant features have different properties, which explains why we considered them separately. In particular, we showed that a redundant feature will always remain redundant at subsequent steps of the feature selection process, while an irrelevant feature may later turn into relevant. An important practical consequence is that redundant features, once detected, may be removed from the set of candidate features.

We derive upper and lower bounds for the target objective functions and relate these bounds with the feature types. In particular, we showed that fully relevant features reach the maximum of the target objective functions, irrelevant and redundant features reach the minimum, and relevant features take a value in between. This framework (target objective functions, feature types, and objective function values for each feature type) provides a theoretical setting that can be used to compare the actual feature selection methods. Under this framework, the correct decisions at each step of the feature selection process are to select fully relevant features first and only afterwards relevant features, leave irrelevant features for future consideration (since they can later turn into relevant), and discard redundant features (since they will remain redundant).

Besides the theoretical framework, we defined a distributional setting, based on the definition of specific class, features, and a performance metric, designed to highlight the various deficiencies of methods. The setting includes four features, each belonging to one of the feature types defined above, and a class with two possible values. As performance metric, we introduced the minimum Bayes risk, a theoretical measure that does not rely on specific datasets and classifiers. The metric corresponds to the minimum total probability of misclassification for a certain class and set of selected features.

Actual feature selection methods are based on approximations of the target objective functions. The target objective function OF', in particular, comprises three terms, expressing the association between the candidate feature and the class (the relevance), the association between the candidate feature and the already selected features (the inter-feature redundancy), and the association between the candidate feature and the already selected features given the class (the class-relevant redundancy). The class-relevant redundancy is sometimes coined as the *good* redundancy, since it expresses the contribution of the candidate feature to the explanation of the class, when considering that the information contained in the already selected features is known. We also say that this term reflects the *complementarity* between the candidate and the already selected features with respect to the class.

Method MIM was the first method to be proposed, and completely ignored redundancy. Methods MIFS, mRMR, and maxMIFS ignored complementary effects, i.e. they did not include the class-relevant redundancy term in their objective functions. These methods lose both the upper and lower bounds of the target objective function and, more importantly, lose the connection between the bounds and the specific feature types, i.e. it is no longer possible to guarantee that fully relevant and relevant features are selected before redundant and irrelevant features, or that fully relevant come before relevant.

Methods CIFE, JMI, CMIM, and JMIM considered complementarity effects, but in different ways. The main difference between these methods lies in the approximation of the redundancy terms (the ones related with inter-feature and class-relevant redundancies). These terms depend on the complete set of already selected features and are difficult to estimate. To overcome this difficulty, the methods approximate the redundancy terms by a function of already selected features taken individually. In particular, CIFE uses the sum of the associations with the individual already selected features, JMI uses the average, and both CMIM and JMIM use the maximum. In relation to other methods, JMIM introduced an extra term in its objective function, which is unimportant and leads to confusion in the selection process. The approximations of the remaining methods lead to the following problems: CIFE overscales the redundancy, JMI undervalues the redundancy, and CMIM undervalues the redundancy in a lower extent than JMI but penalizes the complementarity. The consequences of these approximations are that CIFE loses both the upper and lower bound of the target objective function, JMI and CMIM preserve only the lower bound, and JMIM preserves only the upper bound. Moreover, as in the case of the methods that ignore complementary, the methods lose the connection between the bounds of the target objective function and the specific feature types, except in a specific case for CMIM. The drawbacks of the various methods were summarized in Table 7.2.

These results show that, for all methods, it is always possible to find cases where incorrect decisions are produced, and we have provided several examples throughout this chapter and as part of our distributional setting. However, the drawbacks of the methods have different degrees of severity. MIM is a very basic method that we only considered for reference purposes. Ignoring complementary is a severe drawback that can lead to gross errors in the selection process. Thus, MIFS, mRMR, and maxMIFS, should be avoided. Regarding the methods that include complementarity effects, CIFE and JMIM should also be avoided, CIFE because its objective function is unbounded both inferiorly and superiorly due to the overscaled redundancy drawback, and JMIM because its objective function includes a bad approximation of an unimportant term that leads to confusion. Thus, the methods that currently have superior performance are JMI and CMIM. There is no clear-cut decision between these two methods, since both present drawbacks of equivalent degree of severity. JMI undervalues both the inter-feature and the class-relevant redundancy. CMIM also undervalues both types of redundancy. However, it tends to approximate better the inter-feature redundancy, but worse the class-relevant redundancy due to the problem of complementarity penalized.

A question that might arise is whether it makes sense to consider such methods based on MI on a setting with both absolutely continuous and discrete features. Recall that the properties of entropy and differential entropy differ significantly – for instance, they are defined in different ranges. We used features of both types in our theoretical setting. Absolutely continuous features were useful in particular for emphasizing the potential limitations concerning methods suffering from complementarity ignored drawback. However, if this drawback, or the remaining discussed drawbacks of the methods, did not exist, there would be no problem in considering features of the two types. In fact, the values obtained by the target objective functions are exactly as they should, independently of the type of feature considered.

A related theoretical aspect that should be addressed is how to extend the definitions of differential entropy and MI for absolutely continuous variables to include quantities such as $h(\boldsymbol{X}, \boldsymbol{X})$, $h(\boldsymbol{X}|\boldsymbol{X})$, and $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{X})$. It has been argued that $\mathrm{MI}(\boldsymbol{X}, \boldsymbol{X}) = +\infty$, using limiting arguments; see, for instance, [Kot66]. By a similar argument, we would have, from (6.3), $h(\boldsymbol{X}|\boldsymbol{X}) = -\infty$, which is coherent with the idea that the value of the differential entropy decreases as the uncertainty decreases, recall (6.8). In fact, we have noted that arbitrarily small values can be, in fact, reached by differential entropy, even for random variables on which there is uncertainty. An example that enlightens this fact is the univariate normal distribution, considering a variance that tends to 0. However, in the discrete case, no uncertainty is associated with entropy 0, and no uncertainty corresponds to having a degenerate random variable, which is in the end a discrete random variable.

Concerning further open problems, the need to estimate high-dimensional MI terms is mentioned in [VE14]: *An important challenge is developing more efficient methods for estimating MI in high-dimensional spaces.* Until this is possible, it is unavoidable that approximations as the one that is considered by all methods, where terms on individual features of $\boldsymbol{S}$ replace the original terms that contain the whole $\boldsymbol{S}$ in the objective functions of the different methods, are considered. It was clearly emphasized throughout this work how considering such approximations restricts the properties of interest of the associated methods.

8 Conclusion

In the first part of this thesis, we have proposed different types of algorithms for approximating steady states of structured Markov chains. Our main achievement is that, for any model that we are given, we have an algorithm that is able to find the corresponding approximate steady state efficiently, even for very high-dimensional problems. This includes models that are associated with topologies that, in theory, would not suit TT format, which is the format associated with all proposed algorithms. In order to apply the algorithm, we only need the Kronecker representation (1.3) of the transition rate matrix of the model of interest, given that the corresponding required representation associated with TT format can be then easily extracted. We next go in more detail through the contributions from the first part of the thesis chapter by chapter.

In Chapter 3, we have proposed and compared different algorithms for the solution of (1.1), whose structures are all in TT format: after having proposed a simple iterative method adapted from the well-known power method for matrices, using an eigenvalue problem formulation; we have focused on alternating optimization schemes, considering an equivalent least squares formulation. Since such alternating schemes do not allow that ranks are adapted during the iteration, a third algorithm, AMEn, was proposed. It combines the advantages of such alternating schemes with a rank adaptive strategy based on incorporating information about the residual that reminds of the steepest descent method. By considering the residual information, convergence should be improved, which is also relevant because alternating schemes tend to converge slowly. We have verified that the three algorithms, because of being built in terms of TT format, perform particularly well when compared to an existing algorithm in a different tensor format. We have additionally verified that the last algorithm performs better than the remaining two, while it behaves particularly well for high-dimensional problems associated with a large number of subsystems. Such experiments were performed on two queuing networks from our benchmark [Mac15].

In Chapter 4, we have proposed an algorithm of a different type for the solution of (1.1), based on an existing tensorized multigrid method, which is itself particularly suited for Kronecker structured Markov chains given the way restriction and interpolation operators are defined. We have combined the advantages of this method with the advantages of AMEn, while trying to avoid the drawbacks of each. The two algorithms had already been implemented in TT format so that combining them to obtain an algorithm in TT format was natural. While the existing tensorized multigrid had problems in the coarsest grid as the mode sizes get reduced on the way down the grids but not the number of subsystems, AMEn is perfectly suited for being applied in the corresponding coarsest grid problem since its main limitation concerns cases where the mode sizes are not particularly small. In turn, the fact that the number of modes is the same as in the finest grid is not a problem since this algorithm is particularly suited for problems with a large number of modes. We have verified on a variety of models from our benchmark [Mac15] that the proposed method consistently beats both AMEn and the original tensorized multigrid scheme as the number of subsystems and the number of possible states per subsystem are increased. We have also verified how algorithms in TT format in general can perform remarkably well even for topologies that are not, in theory, suitable.

In Chapter 5, we have proposed two alternative algorithms for the solution of (1.1) that are both based again on multigrid approaches. Restriction and interpolation have however been chosen differently, based on the well-known aggregation/disaggregation techniques. Such an approach has been in fact used in the past to solve (1.1), in particular with aggregation (name given to restriction) operators that take the Kronecker structure of the generator matrix (1.3) into account, but never considering all structures in TT format as we have here done. The two algorithms are associated with two variants (two possible choices for the aggregation and disaggregation, name given to interpolation, operators). In the end, we have shown, again considering the broad benchmark collection [Mac15], that these two variants allow covering the efficient computation of steady states of all types of models. The first variant considers aggregation and disaggregation operators that are improved versions of the restriction and interpolation operators considered in the algorithm proposed in Chapter 4, by serving the same purpose of dealing with models with local 1D topology particularly well but performing better. Therefore, this variant should be used for indistinguishable models. In turn, the second variant is able to address the complementary subclass of models – distinguishable models – particularly well. Such models are not possible to address with neither the first variant nor the algorithm proposed in Chapter 4. The experiments that have been performed in the context of this chapter, again considering models from the benchmark [Mac15], demonstrate the expected robustness of the proposed methods in this deeper level than that observed in the experiments performed in Chapter 4, in particular with distinguishable models also being possible to address; while we also consider a reducible model, for which one can in fact find a logical unique solution by isolating the connected components of states even though in theory there is no goal unique solution for the problem given a model of this type. Both mentioned subclasses of models are typically avoided in the literature given their tricky particularities, but they are in fact extremely relevant and associated with wide ranges of applications.

In the second part of the thesis, we have carried out an evaluation and a comparison of forward feature selection methods based on mutual information. The methods are based on approximations of target objective functions, and we have selected a set of methods representative of the various types of approximations. We have discussed the various drawbacks introduced by such approximations. We have introduced a theoretical setting based on the target objective functions and on a categorization features according to their contribution to the explanation of the class. The features are categorized as irrelevant, redundant, relevant, and fully relevant. This categorization has two novelties regarding previous works: first, we have introduced the important category of fully relevant features; second, we have separated non-relevant features in two categories of irrelevant and redundant features. Fully relevant features are features that fully explain the class and, therefore, its detection can be used as a stopping criterion of the feature selection process. Irrelevant and redundant features have different properties, which explains why we considered them separately. In particular, we have showed that a redundant feature will always remain redundant at subsequent steps of the feature selection process, while an irrelevant feature may later turn into relevant. An important practical consequence is that redundant features, once detected, may be removed from the set of candidate features. We have derived upper and lower bounds for the target objective functions and related these bounds with the feature types: we have showed that fully relevant features reach the maximum of the target objective functions, irrelevant and redundant features reach the minimum, and relevant features take a value in between. We have then analysed how each method copes with the good properties of the target objective functions. Additionally, we have defined a distributional setting, based on a specific definition of class, features, and a novel performance metric; it provides a feature ranking for each method that is compared with the ideal feature ranking coming out of the theoretical framework. The setting has been designed to challenge the feature selection methods, and illustrate the consequences of their drawbacks. Based on our work, we have identified clearly the methods that should be avoided, and the methods that have the best performance.

A Some analytical relevant computations

In this chapter, we deduce some results associated with analytical computations that are relevant but whose inclusion in the core of the thesis would become heavy, while it would tend to distract the reader from the main content. Such results concern the distributional setting developed in Section 7.3.

A.1 Computation of the terms in the tables of Section 7.3.3

In this section, we derive the expressions required for completing the tables given in Section 7.3.3.

A.1.1 Values in Table 7.3

Univariate differential entropies (continuous features). The entropy of X is obtained from Example 1, in Section 6.2, considering $n = 1$. As for the entropy of the other continuous feature, $X - k'Y$, the same expression can be used since it is widely known that linear or affine combinations of independent univariate features following normal distributions also follow normal distributions. All we need are the variances of these two features. The variance of X is 1 and the variance of $X - k'Y$ is $1 + k^{'2}$. Therefore, the corresponding entropies are $h(X) = \frac{1}{2}\ln(2\pi e)$ and $h(X - k'Y) = \frac{1}{2}\ln(2\pi e(1 + k^{'2}))$, respectively.

Univariate entropies (discrete features and class). Concerning Z, applying Definition 1, $H(Z) = \ln(2)$.

We now discuss the values of $H(X_{\text{disc}})$ and $H(C_k)$. Given that X and Y are independent and individually follow standard normal distributions, the joint density function of (X, Y)

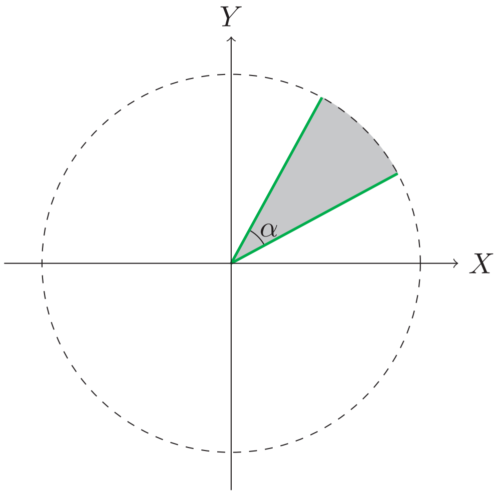


Figure A.1: Angle between two rays starting from the origin; $\alpha/(2\pi)$ is the probability that (X, Y) belongs to the region delimited by the two rays, when X and Y are two independent random variables following standard normal distribution.

is

$$f_{X,Y}(x, y) = \frac{1}{2\pi} \exp(-(x^2 + y^2)) = \phi(x)\phi(y).$$

Therefore, the density at point (x, y) only depends on the distance from this point to the origin, $\sqrt{x^2 + y^2}$, in the context of the two-dimensional space defined by (X, Y). As a consequence, the probability of (X, Y) taking values in a region limited by two rays having the origin as starting point is given by $\alpha/(2\pi)$, with α denoting the angle between the two rays, as illustrated in Figure A.1. The circle is dashed in the figure since we can consider an infinite radius.

Considering an infinite radius is in fact what we need. Both C_k and X_{disc} are characterized by a partition of $\mathbb{R}^2$ in two regions separated by a line that crosses the origin. Therefore, each region covers an angle $\alpha = \pi$, so that each region has associated probability 1/2. Thus, C_k and X_{disc} follow a Bernoulli distribution with success probability 1/2, just as Z. As a result, $H(C_k) = H(X_{\text{disc}}) = H(Z) = \ln(2)$.

A.1.2 Values in Table 7.4

Continuous features. In order to derive $\text{MI}(C_k, X)$ and $\text{MI}(C_k, X - k'Y)$, expression (6.9) should be, in general, preferred over (6.10). In fact, the entropy of a feature given the class can be obtained through the corresponding probability density functions; recall (6.2). These, in turn, are possible to derive easily in our setting. Therefore, we calculate the MI of interest using the representation

$$\mathrm{MI}(X_i, C_k) = h(X_i) - \sum_{j=0}^{1} \int f_{X_i|C_k=j}(u) P(C_k = j) \ln f_{X_i|C_k=j}(u) du, \qquad (A.1)$$

where $P(C_k = j) = 1/2$, $j = 0, 1$. It all comes down to determining $f_{X_i|C_k=j}(u)$, $j = 0, 1$, as $h(X_i)$ is known (vide Table 7.3).

Given that the features follow a normal distribution, the conditional distribution of interest is the well-known skew-normal distribution [Pas13, Ch. 5]. Therefore, we only need to determine, in each case, the parameters of the mentioned distribution; recall (7.14).

In the case of $\mathrm{MI}(C_k, X)$, it was proved in [Pas13, Ch. 5] that $X|C_k = j$, $j = 0, 1$, follow skew-normal distributions with parameters $(0, 1, \frac{(-1)^{j+1}}{k})$; i.e. $X|C_k = j \sim \mathrm{SN}(0, 1, \frac{(-1)^{j+1}}{k})$, $j = 0, 1$.

As for $\mathrm{MI}(C_k, X - k'Y)$, we use the procedure used for the determination of $\mathrm{MI}(C_k, X)$ in [Pas13, Ch. 5] to prove that $X - k'Y|C_k = j \sim \mathrm{SN}(0, \sqrt{1 + k'^2}, (-1)^{j+1}(\frac{1-kk'}{k+k'}))$, $j = 0, 1$. The procedure consists of obtaining the conditional distribution functions of the feature given the two different possible values of the class, taking then the corresponding derivatives in order to obtain the associated probability density functions.

In this context, we will need the probability density function $f_{X+kY, X-k'Y}(z, w)$. This can be obtained from the joint density of the pair (X, Y). In fact, there is a way to obtain the probability density function of $g(X, Y)$, with g being a bijective function, from the probability density function of (X, Y), using the general well-known expression [Kar93, Ch. 2]

$$f_{g(X,Y)}(z, w) = f_{X,Y}(g^{-1}(z, w)) \left| \frac{dg^{-1}(z, w)}{d(z, w)} \right|, \qquad (A.2)$$

where $|\frac{dg^{-1}(z,w)}{d(z,w)}|$ denotes the absolute value of the Jacobian of the inverse of the function g.

As for the inverse function of the transformation $g(X, Y) = (X + kY, X - k'Y)$, it is given by

$$g^{-1}(z, w) = \left(\frac{kz + k'w}{k + k'}, \frac{w - z}{k + k'} \right).$$

The absolute value of its Jacobian is $|1/(k + k')|$. As both k and k' are non-negative, this can be simply written as $1/(k + k')$.

As a result, we have

$$f_{X+kY,X-k'Y}(z,w) = \frac{1}{k+k'}\phi\left(\frac{kz+k'w}{k+k'}\right)\phi\left(\frac{w-z}{k+k'}\right), \quad (z,w) \in \mathbb{R}^2.$$

We can now proceed with the derivation of the distribution functions of interest. From now on, the distribution function of $\boldsymbol{Z}$ at $\boldsymbol{z}$ will be represented by $F_{\boldsymbol{Z}}(z)$.

We start with the case $C_k = 0$:

$$\begin{aligned}
&F_{X-k'Y|X+kY<0}(u) = P(X-k'Y \leq u|X+kY<0)\\
&= \frac{P(X-k'Y \leq u, X+kY<0)}{P(X+kY<0)}\\
&= 2\int_{-\infty}^{0}\int_{-\infty}^{u} f_{X+kY,X-k'Y}(z,w)dw\,dz\\
&= 2\int_{-\infty}^{0}\int_{-\infty}^{u} \frac{1}{k+k'}\phi(\frac{kz+k'w}{k+k'})\phi(\frac{w-z}{k+k'})dw\,dz\\
&= 2\int_{-\infty}^{0}\int_{-\infty}^{u} \frac{1}{k+k'}\frac{1}{2\pi}\exp\left\{-\frac{1}{2}\frac{(k^2+1)z^2}{(k+k')^2}\right\}\exp\left\{-\frac{1}{2}(k'^2+1)[w-\frac{z(1-kk')}{1+k'^2}]^2+\right.\\
&\qquad\qquad \left. z^2[\frac{(1+k^2)(1+k'^2)-(1-kk')^2}{1+k'^2}]\right\}dw\,dz\\
&= \int_{-\infty}^{u}\frac{1}{\sqrt{\pi}}\exp\left\{-\frac{1}{2}\frac{(k+k')^2z^2}{(k+k')^2}\right\}\int_{-\infty}^{0}\frac{1}{\sqrt{\pi}(k+k')}\times\\
&\qquad\qquad \exp\{-\frac{1}{2}\frac{(1+k'^2)(w-\frac{z(1-kk')}{1+k'^2})^2}{(k+k')^2}\}dz\,dw\\
&= \sqrt{2}\int_{-\infty}^{u}\frac{1}{\sqrt{2\pi}}\frac{\sqrt{2}}{\sqrt{1+k'^2}}\int_{-\infty}^{0}\frac{1}{\frac{\sqrt{2\pi}(k+k')}{\sqrt{1+k'^2}}}\exp\left\{-\frac{1}{2}\frac{z^2(k+k')^2}{1+k'^2}\right\}dz\,dw\\
&= \frac{\sqrt{2}}{\sqrt{1+k'^2}}\int_{-\infty}^{u}\frac{1}{\sqrt{2\pi}\sqrt{1+k'^2}(k+k')}\exp\left\{-\frac{1}{2}\frac{z^2}{1+k'^2}\right\}\Phi(-\frac{(1-kk')z}{(k-k')\sqrt{1+k'^2}})dz\\
&= \int_{-\infty}^{u}\frac{2}{\sqrt{1+k'^2}}\phi(\frac{z}{\sqrt{1+k'^2}})\Phi(-\frac{(1-kk')z}{(k-k')\sqrt{1+k'^2}})dz.
\end{aligned}$$

Hence, $X-k'Y|X+kY<0 \sim \mathrm{SN}(0, \sqrt{1+k'^2}, \frac{-(1-kk')}{k+k'})$.

Some auxiliary steps were required in the first step of the derivation above in which $\frac{1}{k+k'}\phi(\frac{kz+k'w}{k+k'})\phi(\frac{z-w}{k-k'})$ was transformed significantly. The main technicality about such steps was the algebraic manipulation

$$
\begin{aligned}
&(kz+k'w)^2+(w-z)^2\\
&=(1+k^2)z^2-2wz(1-kk')+(k'^2+1)w^2\\
&=(1+k'^2)\{w^2-2w\frac{z(1-kk')}{1+k'^2}+[\frac{z(1-kk')}{1+k'^2}]^2-[\frac{z(1-kk')}{1+k'^2}]^2\}+(1+k^2)z^2\\
&=(1+k'^2)\{[w-\frac{z(1-kk')}{1+k'^2}]^2-\frac{[z(1-kk')]^2}{(1+k'^2)^2}\}+(1+k^2)z^2\\
&=(1+k'^2)[w-\frac{z(1-kk')}{1+k'^2}]^2+z^2\frac{(k+k')^2}{1+k'^2}.
\end{aligned}
$$

As for the conditional case in which $C_k = 1$, we provide a briefer version of the computation as most steps are the same as for $C_k = 0$.

$$
\begin{aligned}
&F_{X-k'Y|X+kY\geq 0}(u)\\
&=P(X-k'Y\leq u|X+kY\geq 0)\\
&=\frac{P(X-k'Y\leq u, X+kY\geq 0)}{P(X+kY\geq 0)}\\
&=2\int_{0}^{+\infty}\int_{-\infty}^{u} f_{X+kY,X-k'Y}(z,w)dw\,dz\\
&=2\int_{0}^{+\infty}\int_{-\infty}^{u}\frac{1}{k+k'}\phi(\frac{kz+k'w}{k+k'})\phi(\frac{w-z}{k+k'})dw\,dz\\
&=2\int_{-\infty}^{0}\int_{-\infty}^{u}\frac{1}{k+k'}\frac{1}{2\pi}\exp\left\{-\frac{1}{2}\frac{(k^2+1)z^2}{(k+k')^2}\right\}\exp\left\{-\frac{1}{2}(k'^2+1)[w-\frac{z(1-kk')}{1+k'^2}]^2+\right.\\
&\qquad\left. z^2[\frac{(1+k^2)(1+k'^2)-(1-kk')^2}{1+k'^2}]\right\}dw\,dz\\
&=\int_{-\infty}^{u}\frac{2}{\sqrt{1+k'^2}}\phi(\frac{z}{\sqrt{1+k'^2}})[1-\Phi(-\frac{(1-kk')z}{(k-k')\sqrt{1+k'^2}})]dz.
\end{aligned}
$$

Given the symmetry of the normal distribution, we know that $(1-\Phi(-x))=\Phi(x)$. This allows reducing the expression to

$$
\int_{-\infty}^{u}\frac{2}{\sqrt{1+k'^2}}\phi(\frac{z}{\sqrt{1+k'^2}})\Phi(\frac{(1+kk')z}{(k-k')\sqrt{1+k'^2}})dz.
$$

Thus, $X-k'Y|X+kY\geq 0\sim \text{SN}\left(0,\sqrt{1+k'^2},\frac{(1-kk')}{k+k'}\right)$.

Discrete features. In order to obtain $\text{MI}(C_k, X_{\text{disc}})$, we can use (6.6) in the form $\text{MI}(C_k, X_{\text{disc}}) = H(C_k) + H(X_{\text{disc}}) - H(C_k, X_{\text{disc}})$. In this case, we only need to compute $H(C_k, X_{\text{disc}})$ since the required univariate entropies are known from Table 7.3. From Definition 1, this requires obtaining the probabilities of the four possible combinations

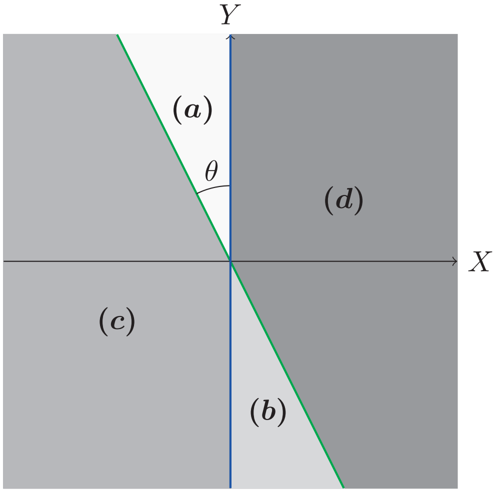


Figure A.2: Four regions associated with the joint distribution of (X_{disc}, C_k), where $\theta = \arctan k$, when X and Y are two independent random variables following standard normal distribution.

of values associated with the pair (C_k, X_{disc}). We represent the regions associated with such values in Figure A.2, considering the two-dimensional space defined by the pair (X, Y). Considering the reasoning used to obtain $H(C_k)$ and $H(X_{\text{disc}})$, associated with Figure A.1, we only need the four angles covered by the associated four regions in order to compute their corresponding probabilities. The determination of such angles only requires the knowledge of θ, represented in Figure A.2 since the remaining angles consist of its supplementary, its opposite, and the opposite of its supplementary.

In the end, there are two angles (associated with regions (a) and (b) in Figure A.2) whose value is $\arctan k$, while the other two (associated with regions (c) and (d) in Figure A.2) have the value $(\pi - \arctan k)$, implying that

$$P(X_{\text{disc}} = u, C_k = j) = \begin{cases} \frac{\pi - \arctan k}{2\pi}, & u = 0, j = 0 \text{ and } u = 1, j = 1 \\ \frac{\arctan k}{2\pi}, & u = 0, j = 1 \text{ and } u = 1, j = 0 \end{cases}. \tag{A.3}$$

As a result,

$$H(C_k, X_{\text{disc}}) = -2 \times \left(\frac{\arctan k}{2\pi} \ln(\frac{\arctan k}{2\pi}) + (\frac{1}{2} - \frac{\arctan k}{2\pi}) \ln(\frac{1}{2} - \frac{\arctan k}{2\pi}) \right).$$

We obtain

$$\text{MI}(C_k, X_{\text{disc}}) = 2\ln(2) + \frac{\arctan k}{\pi} \ln(\frac{\arctan k}{2\pi}) + (1 - \frac{\arctan k}{\pi}) \ln(\frac{1}{2} - \frac{\arctan k}{2\pi}).$$

As for $\text{MI}(C_k, Z)$, its value is 0 since C_k and Z are independent.

A.1.3 Values in Table 7.5

We start with $\mathrm{MI}(X, X_{\text{disc}})$. X_{disc} is redundant given $\{X\}$, so that $H(X_{\text{disc}}|X) = 0$. Additionally, $H(X_{\text{disc}}) = \ln(2)$ (vide Table 7.3), so that, by (6.3), $\mathrm{MI}(X, X_{\text{disc}}) = H(X_{\text{disc}}) - H(X_{\text{disc}}|X) = \ln(2) - 0 = \ln(2)$.

As for $\mathrm{MI}(X - k'Y, X_{\text{disc}})$, we first note that the deduction of $\mathrm{MI}(C_k, X)$ in [Pas13, Ch. 5] can be similarly done for the case where k in (7.11) is allowed to be negative, from which one would conclude that $\mathrm{MI}(C_{-k'}, X) = \mathrm{MI}(C_{k'}, X)$. In turn, X_{disc} and $X - k'Y$ are obtained from $C_{k'}$ and X, respectively, by rotating k' degrees anticlockwise, considering the two-dimensional space defined by the pair (X, Y). As a result, $\mathrm{MI}(X_{\text{disc}}, X - k'Y) = \mathrm{MI}(C_{k'}, X)$. In fact, MI is invariant under one-to-one transformations; see [DMS10] for more details. Finally, by transitivity, $\mathrm{MI}(X - k'Y, X_{\text{disc}}) = \mathrm{MI}(C_{k'}, X)$.

As for $\mathrm{MI}(X, X - k'Y)$, we use (6.6). We have $\mathrm{MI}(X, X - k'Y) = h(X) + h(X - k'Y) - h(X, X - k'Y)$. We only need to compute $h(X, X - k'Y)$ since the univariate entropies are known (vide Table 7.3). As both features follow normal distributions, the joint distribution is a bivariate normal distribution, whose entropy depends on the determinant of the covariance matrix, as described in Example 1. The value of the mentioned determinant is k'^2, so that $\mathrm{MI}(X, X - k'Y) = \frac{1}{2}\ln\left(1 + 1/k'^2\right)$.

The MI terms involving Z do not require any calculation given that Z is independent of X, X_{disc}, and $X - k'Y$, implying that $\mathrm{MI}(Z, X) = \mathrm{MI}(Z, X_{\text{disc}}) = \mathrm{MI}(Z, X - k'Y) = 0$.

A.1.4 Values in Table 7.6

For deriving $\mathrm{MI}(X, X_{\text{disc}}|C_k)$, we use (6.11). We have $\mathrm{MI}(X, X_{\text{disc}}|C_k) = H(X_{\text{disc}}|C_k) - H(X_{\text{disc}}|X, C_k)$. Noting that $H(X_{\text{disc}}|X, C_k) = 0$ since $H(X_{\text{disc}}|X) = 0$, using (6.15), we have $\mathrm{MI}(X, X_{\text{disc}}|C_k) = H(X_{\text{disc}}|C_k)$. In turn, $H(X_{\text{disc}}|C_k) = H(X_{\text{disc}}) - \mathrm{MI}(X_{\text{disc}}, C_k)$, where the values $H(X_{\text{disc}})$ and $\mathrm{MI}(X_{\text{disc}}, C_k)$ have been obtained; recall Tables 7.3 and 7.5, respectively. We conclude that

$$\mathrm{MI}(X, X_{\text{disc}}|C_k) = -\frac{\arctan k}{\pi}\ln\left(\frac{\arctan k}{\pi}\right) - \left(1 - \frac{\arctan k}{\pi}\right)\ln\left(1 - \frac{\arctan k}{\pi}\right).$$

Concerning $\mathrm{MI}(X, X - k'Y|C_k)$, we use (6.12). We have $\mathrm{MI}(X, X - k'Y|C_k) = h(X|C_k) + h(X - k'Y|C_k) - h(X, X - k'Y|C_k)$. The first two terms were obtained in the context of the determination of $\mathrm{MI}(X, C_k)$ and $\mathrm{MI}(X - k'Y|C_k)$; they consist of the second term in (A.1).

As in the computation of class-conditional entropies for univariate continuous features, recall (A.1), we only need the joint probability density functions of $(X, X - k'Y)$ conditioned on the two possible values of the class to determine $h(X, X - k'Y|C_k)$. These are obtained from the corresponding conditional probability density functions associated with the pair (X, Y), using (A.2).

We first need to derive the probability density functions associated with the distributions $(X, Y)|C_k = j$, $j = 0, 1$. These are obtained starting from the corresponding distribution functions, as done in the context of the determination of the probability density functions needed in sequence of (A.1).

We start with $C_k = 1$. We have

$$\begin{aligned}F_{(X,Y)|C_k=1}(x, y) =& \frac{P(X \leq x, Y \leq y, X + kY \geq 0)}{P(X + kY \geq 0)}\\ =& 2P(X \leq x, Y \leq y, X + kY \geq 0).\end{aligned}$$

In order to proceed, we separate the expression in two cases. In fact, if $x < -ky$, the value is simply 0. If, instead, $x \geq -ky$, we have

$$\begin{aligned}2P(X \leq x, Y \leq y, X + kY \geq 0) =& 2\int_{-\infty}^{x}\int_{-\frac{u}{k}}^{y} \phi(u)\phi(v)dv\,du\\ =& 2\int_{-\infty}^{x} \phi(u)[\Phi(v) - \Phi(-\frac{u}{k})]du\\ =& 2\Phi(y)\Phi(x) - F_{\mathrm{SN}(0,1,-\frac{1}{k})}(x).\end{aligned}$$

Thus, the corresponding density is given by

$$f_{(X,Y)|C_k=1}(x, y) = \left\{\begin{array}{cc} 2\phi(y)\phi(x), & x \geq -ky \\ 0, & x < -ky \end{array}\right. .$$

We now make the same type of deduction for $C_k = 0$. We have

$$\begin{aligned}F_{(X,Y)|C_k=0}(x, y) =& \frac{P(X \leq x, Y \leq y, X + kY < 0)}{P(X + kY < 0)}\\ =& 2P(X \leq x, Y \leq y, X + kY < 0).\end{aligned}$$

If $x < -ky$, we obtain

$$\begin{aligned}2P(X \leq x, Y \leq y, X + kY < 0) =& 2\int_{-\infty}^{x}\int_{-\infty}^{y} \phi(u)\phi(v)dv\,du\\ =& 2\Phi(y)\Phi(x).\end{aligned}$$

If, instead, $x \geq -ky$, we have

$$\begin{aligned} 2P(X \leq x, Y \leq y, X + kY \geq 0) =& 2\left[\int_{-\infty}^{x}\int_{-\infty}^{y}\phi(u)\phi(v)du\,dv - \int_{-\infty}^{x}\int_{-\frac{u}{k}}^{y}\phi(u)\phi(v)dv\,du\right] \\ =& 2\Phi(y)\Phi(x) - [2\Phi(y)\Phi(x) - F_{\mathrm{SN}(0,1,-\frac{1}{k})}(x)] \\ =& F_{\mathrm{SN}(0,1,-\frac{1}{k})}(x). \end{aligned}$$

Thus, the corresponding density is given by

$$f_{(X,Y)|C_k=0}(x,y) = \begin{cases} 0, & x \geq -ky \\ 2\phi(y)\phi(x), & x < -ky \end{cases}.$$

We can now obtain $f_{X,X-k'Y}(u,v)$ using (A.2). In this case, $g^{-1}(z,w) = (z, \frac{z-w}{k'})$. The determinant of the corresponding Jacobian is $1/k'$. Therefore, we have $f_{(X,X-k'Y)|C_k=c}(u,v) = f_{(X,Y)|C_k=c}(u, \frac{u-v}{k'})\frac{1}{k'}$, $c = 0, 1$. Thus, we obtain

$$f_{(X,X-k'Y)|C_k=1}(u,v) = \begin{cases} 2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}, & u > \frac{k}{k'+k}v \\ 0, & u \leq \frac{k}{k'+k}v \end{cases}$$

and

$$f_{(X,X-k'Y)|C_k=0}(u,v) = \begin{cases} 0, & u > \frac{k}{k'+k}v \\ 2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}, & u \leq \frac{k}{k'+k}v \end{cases},$$

where we note that $\phi(\frac{v-u}{k'})\frac{1}{k'}$ can be also seen as the density for a normal distribution with parameters (u, k'^2) evaluated at v.

We have

$$h(X, X-k'Y|C_k = 1) = -\int_{-\infty}^{+\infty}\int_{\frac{k}{k'+k}v}^{+\infty} 2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}\ln(2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'})dv\,du$$

and

$$h(X, X-k'Y|C_k = 0) = -\int_{-\infty}^{+\infty}\int_{-\infty}^{\frac{k}{k'+k}v} 2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}\ln(2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'})dv\,du.$$

This implies that

$$h(X, X-k'Y|C_k) = -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}\ln(2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'})dv\,du.$$

The part inside the logarithm can be re-written considering the explicit expression of the density of a standard normal distribution. We have

$$\begin{aligned}\ln(2\phi(u)\phi(\frac{v-u}{k'})\frac{1}{k'}) &= \ln(\frac{1}{\pi k'}\exp(-\frac{1}{2k'^2}[(1+k')u^2-2uv+v^2])) \\ &= -\ln(\pi k') - \frac{1}{2k'^2}[(1+k')u^2-2uv+v^2].\end{aligned}$$

We can still write this as

$$h(X, X-k'Y|C_k) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\phi(u)\phi(v)[\ln(\pi k') + \frac{1}{2k'^2}[(1+k')u^2-2uv+v^2]]dv\,du.$$

The final result is $\ln(\pi k') + \frac{1}{2} + \frac{1}{2} = \ln\pi + \ln k' + 1$. In fact, the first term is a double integral in the whole space of a probability density function, associated with (X, Y), times a constant, so that the result is such constant, $\ln(\pi k')$; while the remaining terms are also easy to obtain since they consist of constants multiplied with first and second order moments from the normal distribution with parameters (u, k').

As for $\mathrm{MI}(X-k'Y, X_{\text{disc}}|C_k)$, we compute it, using (6.11), through its representation $\mathrm{MI}(X-k'Y, X_{\text{disc}}|C_k) = h(X-k'Y|C_k) - h(X-k'Y|X_{\text{disc}}, C_k)$. Note that $h(X-k'Y|C_k)$ has already been derived, in the context of the determination of $\mathrm{MI}(X-k'Y, C_k)$; recall (A.1). Thus, we only need to derive $h(X-k'Y|X_{\text{disc}}, C_k)$.

We need to obtain $f_{X-k'Y|X_{\text{disc}}=u, C_k=j}(v)$ for each possible combination of pairs (u, j). We first note that

$$\begin{aligned}f_{X-k'Y|X_{\text{disc}}=u,C_k=j}(v) &= \frac{d}{dv}F_{X-k'Y|X_{\text{disc}}=u,C_k=j}(v) \\ &= \frac{d}{dv}\frac{P(X-k'Y\le v, X_{\text{disc}}=u, C_k=j)}{P(X_{\text{disc}}=u, C_k=j)} \\ &= \frac{1}{P(X_{\text{disc}}=u, C_k=j)}\frac{d}{dv}P(X-k'Y\le v, X_{\text{disc}}=u, C_k=j).\end{aligned}$$

The values $P(X_{\text{disc}}=u, C_k=j)$, $u=0,1$ and $j=0,1$, can be found in (A.3).

Therefore, we only need to obtain $P(X-k'Y\le v, X_{\text{disc}}=u, C_k=j)$. We start with

$u = 1$ and $j = 1$. We have to split in two cases. For $v \geq 0$,

$$
\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 1, C_k = 1)\\
&= \int_{-\frac{v}{k'+k}}^{0} \int_{-kz}^{v+k'z} \phi(w)\phi(z)dw\,dz + \int_{0}^{+\infty} \int_{0}^{v+k'z} \phi(w)\phi(z)dz\,dw\\
&= \int_{-\frac{z}{k'+k}}^{0} \phi(z)[\Phi(v+k'z) - \Phi(-kz)]dz + \int_{0}^{+\infty} \phi(z)[\Phi(v+k'z) - \frac{1}{2}]dz\\
&= \int_{-\frac{v}{k'+k}}^{0} \phi(z)\Phi(v+k'z)dz - \frac{1}{2}[F_{\text{SN}(0,1,-k)}(0) - F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})]\\
&\qquad + \int_{0}^{+\infty} \phi(z)\Phi(v+k'z)dz - \frac{1}{4}\\
&= \int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\Phi(v+k'z)dz - \frac{1}{2}F_{\text{SN}(0,1,-k)}(0) + \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k}) - \frac{1}{4}.
\end{aligned}
$$

In turn, for $v < 0$,

$$
\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 1, C_k = 1)\\
&= \int_{-\frac{v}{k'}}^{+\infty} \int_{0}^{v+k'z} \phi(w)\phi(z)dz\,dw\\
&= \int_{-\frac{v}{k'}}^{+\infty} \phi(z)[\Phi(v+k'z) - \frac{1}{2}]dz\\
&= \int_{-\frac{v}{k'}}^{+\infty} \phi(z)\Phi(v+k'z)dz - \Phi(\frac{v}{k'}).
\end{aligned}
$$

We now need to take the derivative of the two expressions with respect to v to obtain the corresponding conditional density functions. In the case of $v \geq 0$,

$$
\begin{aligned}
&\frac{d}{dv}\left[\int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\Phi(v+k'z)dz - \frac{1}{2}F_{\text{SN}(0,1,-k)}(0) + \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k}) - \frac{1}{4}\right]\\
&= \int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\phi(v+k'z)dz + \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k}\\
&\qquad - \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k}\\
&= \int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\phi(v+k'z)dz;
\end{aligned}
$$

while, for $v < 0$,

$$
\begin{aligned}
&\frac{d}{dv}\left[\int_{-\frac{v}{k'}}^{+\infty} \phi(z)\Phi(v+k'z)dz - \frac{1}{2}\Phi(\frac{v}{k'})\right] \\
&= \int_{-\frac{v}{k'}}^{+\infty} \phi(z)\phi(v+k'z)dz + \frac{1}{2}\phi(\frac{v}{k'})\frac{1}{k'} - \frac{1}{2}\phi(\frac{v}{k'})\frac{1}{k'} \\
&= \int_{-\frac{v}{k'}}^{+\infty} \phi(z)\phi(v+k'z)dz.
\end{aligned}
$$

Note that the following important result; cf. [AS64, Ch. 3]; was required in order to obtain both final expressions above:

$$
\frac{d}{dx}\int_{a(x)}^{b(x)} g(x,y)dy = \int_{a(x)}^{b(x)} \frac{dg(x,y)}{dx}dy + g(x,b(x))b'(x) - g(x,a(x))a'(x). \tag{A.4}
$$

This result was applied to $\frac{d}{dv}\int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\Phi(v+k'z)dz$, in the expression for $v \geq 0$, and also to $\frac{d}{dv}\int_{-\frac{v}{k'}}^{+\infty} \phi(z)\Phi(v+k'z)dz$, concerning the case $v < 0$.

The desired probability density function is

$$
f_{X-k'Y|X_{\text{disc}}=1,C_k=1}(v) = \begin{cases} \frac{2\pi}{\pi-\arctan k}\int_{-\frac{v}{k'+k}}^{+\infty} \phi(z)\phi(v+k'z)dz, & v \geq 0 \\ \frac{2\pi}{\pi-\arctan k}\int_{-\frac{v}{k'}}^{+\infty} \phi(z)\phi(v+k'z)dz, & v < 0 \end{cases}.
$$

We now consider $u = 0$ and $j = 1$. We have to split again in two cases. For $v \geq 0$,

$$
\begin{aligned}
P(X-k'Y \leq v, X_{\text{disc}} = 0, C_k = 1) &= \int_0^{+\infty}\int_{-kz}^{0} \phi(w)\phi(z)dz\,dw \\
&= \int_0^{+\infty} \phi(z)[\frac{1}{2} - \Phi(-kz)]dz \\
&= \frac{1}{2}F_{\text{SN}(0,1,-k)}(0) - \frac{1}{4}.
\end{aligned}
$$

For $v < 0$,

$$\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 0, C_k = 1) \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \int_{-kz}^{v+k'z} \phi(w)\phi(z)dw\,dz + \int_{-\frac{v}{k'}}^{+\infty} \int_{-kz}^{0} \phi(w)\phi(z)dz\,dw \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)[\Phi(v + k'z) - \Phi(-kz)]dz + \int_{-\frac{v}{k'}}^{+\infty} \phi(z)[\frac{1}{2} - \Phi(-kz)]dz\,dw \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\Phi(v + k'z)dz - \frac{1}{2}[F_{\text{SN}(0,1,-k)}(-\frac{v}{k'}) - F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})] \\
&\qquad + \frac{1}{2}[1 - \Phi(-\frac{v}{k'})] - \frac{1}{2}[1 - F_{\text{SN}(0,1,-k)}(-\frac{v}{k'})] \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\Phi(v + k'z)dz + \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k}) + \frac{1}{2}\Phi(\frac{v}{k'}).
\end{aligned}$$

We now need to take the derivative of the two expressions with respect to v to obtain the corresponding conditional density functions. In the case of $v \geq 0$, it is simply 0 since there is no dependency on v. As for $v < 0$,

$$\begin{aligned}
&\frac{d}{dv}\left[\int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\Phi(v + k'z)dz + \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k}) + \frac{1}{2}\Phi(\frac{v}{k'})\right] \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\phi(v + k'z)dz + \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k} - \frac{1}{2}\phi(\frac{v}{k'})\frac{1}{k'} \\
&\qquad - \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k} + \frac{1}{2}\phi(\frac{v}{k'})\frac{1}{k'} \\
&= \int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\phi(v + k'z)dz.
\end{aligned}$$

Once again, (A.4) was applied, in this case to $\int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\Phi(v + k'z)dz$.

The desired probability density function is

$$f_{X-k'Y|X_{\text{disc}}=0,C_k=1}(v) = \begin{cases} 0, & v \geq 0 \\ \frac{2\pi}{\arctan k}\int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \phi(z)\phi(v + k'z)dz, & v < 0 \end{cases}.$$

We now consider $u = 1$ and $j = 0$. For $v \geq 0$,

$$
\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 1, C_k = 0)\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \int_0^{v+k'z} \phi(w)\phi(z)dw\,dz + \int_{-\frac{v}{k'+k}}^{0} \int_0^{-kz} \phi(w)\phi(z)dz\,dw\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)[\Phi(v + k'z) - \frac{1}{2}]dz + \int_{-\frac{v}{k'+k}}^{0} \phi(z)[\Phi(-kz) - \frac{1}{2}]dz\,dw\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\Phi(v + k'z)dz - \frac{1}{2}[\Phi(-\frac{v}{k'+k}) - \Phi(-\frac{v}{k'})]\\
&\qquad\qquad + \frac{1}{2}[F_{\text{SN}(0,1,-k)}(0) - F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})] - \frac{1}{2}\Phi(\frac{v}{k'+k})\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\Phi(v + k'z)dz - \frac{1}{2}\Phi(\frac{v}{k'+k})\\
&\qquad\qquad + \frac{1}{2}F_{\text{SN}(0,1,-k)}(0) - \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k}).
\end{aligned}
$$

As for the case $v < 0$, $P(X - k'Y \leq v, X_{\text{disc}} = 1, C_k = 0) = 0$.

We now need to take the derivative with respect to v of the expression obtained for $v \geq 0$ (the derivative of the one for $v < 0$ is 0) to obtain the corresponding conditional density functions. We have

$$
\begin{aligned}
&\frac{d}{dv}\left[\int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\Phi(v + k'z)dz - \frac{1}{2}\Phi(-\frac{v}{k'+k}) + \frac{1}{2}F_{\text{SN}(0,1,-k)}(0) - \frac{1}{2}F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\right]\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\phi(v + k'z)dz - \frac{1}{2}\phi(-\frac{v}{k'+k}) - \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k}\\
&\qquad\qquad + \frac{1}{2}\phi(-\frac{v}{k'+k}) + \frac{1}{2}f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{1}{k'+k}\\
&= \int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\phi(v + k'z)dz.
\end{aligned}
$$

We again applied (A.4), in this case to $\int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\Phi(v + k'z)dz$.

The desired probability density function is

$$
f_{X-k'Y|X_{\text{disc}}=1,C_k=0}(v) = \begin{cases} \frac{2\pi}{\pi - \arctan k}\int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \phi(z)\phi(v + k'z)dz, & v \geq 0\\ 0, & v < 0 \end{cases}.
$$

We finally consider $u = 0$ and $j = 0$. For $v \geq 0$,

$$
\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 0, C_k = 0) \\
&= \int_{-\infty}^{0} \int_{-\infty}^{+\infty} \phi(w)\phi(z) dw\, dz - \int_{0}^{+\infty} \int_{-kz}^{0} \phi(w)\phi(z) dz\, dw \\
&\qquad - \int_{-\infty}^{-\frac{v}{k'}} \int_{v+k'z}^{0} \phi(w)\phi(z) dz\, dw \\
&= \frac{1}{2} - \int_{-\infty}^{0} \phi(z) \left[\Phi(-kz) - \frac{1}{2}\right] dz - \int_{-\infty}^{-\frac{v}{k'}} \left[\frac{1}{2} - \Phi(v + k'z)\right] \phi(z) dz \\
&= \frac{3}{4} - \frac{1}{2} F_{\text{SN}(0,1,-k)}(0) - \frac{1}{2}\Phi\left(-\frac{v}{k'}\right) + \int_{-\infty}^{-\frac{v}{k'}} \Phi(v + k'z)\phi(z) dz.
\end{aligned}
$$

As for the case $v < 0$,

$$
\begin{aligned}
&P(X - k'Y \leq v, X_{\text{disc}} = 0, C_k = 0) \\
&= \int_{-\infty}^{-\frac{v}{k'+k}} \int_{-\infty}^{v+k'z} \phi(w)\phi(z) dw\, dz + \int_{-\frac{v}{k'+k}}^{+\infty} \int_{-\infty}^{-kz} \phi(w)\phi(z) dz\, dw \\
&= \int_{-\infty}^{-\frac{v}{k'+k}} \Phi(v + k'z)\phi(z) dw\, dz + \int_{-\frac{v}{k'+k}}^{+\infty} \Phi(-kz)\phi(z) dz \\
&= \int_{-\infty}^{-\frac{v}{k'+k}} \Phi(v + k'z)\phi(z) dz + \frac{1}{2} - \frac{1}{2} F_{\text{SN}(0,1,-k)}\left(-\frac{v}{k'+k}\right).
\end{aligned}
$$

We again need to take the derivative of the two expressions with respect to v to obtain the corresponding conditional density functions. In the case of $v \geq 0$,

$$
\begin{aligned}
&\frac{d}{dv}\left[\frac{1}{4} + \frac{1}{2} F_{\text{SN}(0,1,k)}(0) - \frac{1}{2}\Phi(-\frac{v}{k'}) + \int_{-\infty}^{\frac{v}{k'}} \Phi(v + k'z)\phi(z) dz\right] \\
&= \int_{-\infty}^{\frac{v}{k'}} \phi(v + k'z)\phi(z) dz - \frac{1}{2}\phi(-\frac{v}{k'})\frac{1}{k'} + \frac{1}{2}\phi(-\frac{v}{k'})\frac{1}{k'} \\
&= \int_{-\infty}^{\frac{v}{k'}} \phi(z)\phi(v + k'z) dz;
\end{aligned}
$$

while, for $v < 0$,

$$
\begin{aligned}
&\frac{d}{dv}\left[\int_{-\infty}^{-\frac{v}{k'+k}} \Phi(v + k'z)\phi(z) dz + \frac{1}{2} - \frac{1}{2} F_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\right] \\
&= \int_{-\infty}^{-\frac{v}{k'+k}} \phi(z)\phi(v + k'z) dz - \frac{1}{2} f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{v}{k'+k} \\
&\qquad + \frac{1}{2} f_{\text{SN}(0,1,-k)}(-\frac{v}{k'+k})\frac{v}{k'+k} \\
&= \int_{-\infty}^{-\frac{v}{k'+k}} \phi(z)\phi(v + k'z) dz.
\end{aligned}
$$

We applied (A.4) to $\int_{-\infty}^{\frac{v}{k'}} \Phi(v+k'z)\phi(z)dz$ and $\int_{-\infty}^{-\frac{v}{k'+k}} \Phi(v+k'z)\phi(z)dz$.

The desired probability density function is

$$f_{X-k'Y|X_{\text{disc}}=0,C_k=0}(v) = \begin{cases} \frac{2\pi}{\pi-\arctan k} \int_{-\infty}^{\frac{v}{k'}} \phi(z)\phi(v+k'z)dz, & v \geq 0 \\ \frac{2\pi}{\pi-\arctan k} \int_{-\infty}^{-\frac{v}{k'+k}} \phi(z)\phi(v+k'z)dz, & v < 0 \end{cases}.$$

We can finally obtain an expression for $h(X-k'Y|X_{\text{disc}}, C_k)$:

$$\begin{aligned} -\frac{\pi-\arctan k}{2\pi} & \left(\int_{-\infty}^{0} (\int_{-\frac{v}{k'}}^{+\infty} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz) \ln(\int_{-\frac{v}{k'}}^{+\infty} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz)dv + \right. \\ & \int_{0}^{+\infty} (\int_{-\frac{v}{k'+k}}^{+\infty} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz) \ln(\int_{-\frac{v}{k'+k}}^{+\infty} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz)dv + \\ & \int_{-\infty}^{0} (\int_{-\infty}^{\frac{v}{k'+k}} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz) \ln(\int_{-\infty}^{\frac{v}{k'+k}} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz)dv + \\ & \left. \int_{0}^{+\infty} (\int_{-\infty}^{\frac{v}{k'}} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz) \ln(\int_{-\infty}^{\frac{v}{k'}} \frac{2\pi}{\pi-\arctan k} \zeta(z,v)dz)dv \right) \\ -\frac{\arctan k}{2\pi} & \left(\int_{-\infty}^{0} (\int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \frac{2\pi}{\arctan k} \zeta(z,v)dz) \ln(\int_{-\frac{v}{k'+k}}^{-\frac{v}{k'}} \frac{2\pi}{\arctan k} \zeta(z,v)dz)dv + \right. \\ & \left. \int_{0}^{+\infty} (\int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \frac{2\pi}{\arctan k} \zeta(z,v)dz) \ln(\int_{-\frac{v}{k'}}^{-\frac{v}{k'+k}} \frac{2\pi}{\arctan k} \zeta(z,v)dz)dv \right), \end{aligned} \tag{A.5}$$

where $\zeta(z,v)$ is the function $\phi(v+k'z)\phi(z)$.

As for the class-conditional MI values that involve Z, $\text{MI}(Z, X|C_k) = \text{MI}(Z, X - k'Y|C_k) = \text{MI}(Z, X_{\text{disc}}|C_k) = 0$. This requires checking that pairwise class-conditional independence holds for the three involved pairs. This follows, as argued in Section 7.3.2, from the fact that Z is independent of the pair composed by C_k and any other input feature.

A.2 Calculations of MBR values in Section 7.3.4

In this section, we start by obtaining the value of $\text{MBR}(C_k, \{X\})$. We then prove that $\text{MBR}(C_k, \{X, Z\}) = \text{MBR}(C_k, \{X\})$.

Concerning the computation of $\text{MBR}(C_k, \{X\})$, the $(C, \{X\})$ Bayes classifier assigns x, $x \in \mathscr{X}$, to 1 if and only if, recall (7.10),

$$\frac{f_{X|X+kY<0}(x)}{f_{X|X+kY\geq 0}(x)} \leq 1.$$

The required densities $f_{X|X+kY\geq 0}$ and $f_{X|X+kY<0}$ are known from Appendix A.1.2. We take this into account to re-write the expression above as

$$\frac{2\exp\left\{\frac{-x^2}{2\sigma_X^2\sigma_{X+kY}^2}\right\}\frac{1}{\sqrt{2\pi\sigma_X}}\Phi\left(-\frac{\rho\sqrt{\sigma_{X+kY}^2}x}{\sqrt{\sigma_X^2}}\frac{1}{\sqrt{\sigma_{X+kY}^2(1-\rho^2)}}\right)}{2\exp\left\{\frac{-x^2}{2\sigma_X^2\sigma_{X+kY}^2}\right\}\frac{1}{\sqrt{2\pi\sigma_X}}\Phi\left(\frac{\rho\sqrt{\sigma_{X+kY}^2}x}{\sqrt{\sigma_X^2}}\frac{1}{\sqrt{\sigma_{X+kY}^2(1-\rho^2)}}\right)} \leq 1,$$

where $\sigma_X^2 = 1$ is the variance of X, $\sigma_{X+kY}^2 = 1 + k^2$ is the variance of $X + kY$, and $\rho = \frac{1}{\sqrt{1+k^2}}$ is the correlation between X and $X + kY$.

Many terms cancel out, and we get simply

$$\Phi\left(-\frac{x}{k}\right) \leq \Phi\left(\frac{x}{k}\right).$$

As Φ is a non-decreasing function, this condition is the same as

$$-\frac{x}{k} \leq \frac{x}{k}.$$

After further cancellations of the denominators, we simply obtain the condition $-x \leq x$, which holds if and only if $x \geq 0$. As a result, the classifier assigns x to 1 if x is non-negative, and to 0 otherwise; note that the classifier applied to X gives X_{disc}.

Therefore, $\mathsf{MBR}(C_k, \{X\})$ depends on the angle, in the two-dimensional space defined by the pair (X, Y), between the lines associated with X and $X + kY$, $\arctan k$. Note that the only knowledge of X needed concerns the value that X_{disc} takes; recall (7.12). As a result, Figure A.2 also illustrates the regions where a wrong classification will occur, which are the regions (a) and (b). The probabilities associated with the different regions have been given in (A.3), allowing us to obtain

$$\mathsf{MBR}(C_k, \{X\}) = 2\frac{\arctan k}{2\pi} = \frac{\arctan k}{\pi}.$$

We now verify that $\mathsf{MBR}(C_k, \{X, Z\}) = \mathsf{MBR}(C_k, \{X\})$. By (7.10), the $(C_k, \{X, Z\})$ Bayes classifier associated with the MBR takes the value 1 if and only if

$$\frac{f_{(X,Z)|X+kY<0}(x, z)}{f_{(X,Z)|X+kY\geq 0}(x, z)} \leq 1.$$

Using the facts that Z and X are class-conditionally independent and that Z is independent of C_k, the condition above reduces to

$$\frac{f_{X|X+kY<0}(x)}{f_{X|X+kY\geq 0}(x)} \leq 1.$$

As a result, the points (x, z) assigned by the classifier to the value 1 are those that verify $x \geq 0$. As a result, $\text{MBR}(C_k, \{X, Z\}) = \text{MBR}(C_k, \{X\})$.

Bibliography

[ACK10] D. F. Anderson, G. Craciun, and Th. G. Kurtz, *Product-form stationary distributions for deficiency zero chemical reaction networks*, Bull. Math. Biol. **72** (2010), no. 8, 1947–1970.

[AFRT06] N. Antunes, C. Fricker, P. Robert, and D. Tibi, *Analysis of loss networks with routing*, Annals of Applied Probability **16** (2006), no. 4, 2007–2026.

[AKLT87] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, *Rigorous results on valence-bond ground states in antiferromagnets*, Physical review letters **59** (1987), no. 7, 799.

[AOA08] A. El Akadi, A. El Ouardighi, and D. Aboutajdine, *A powerful feature selection approach based on mutual information*, International Journal of Computer Science and Network Security **8** (2008), no. 4, 116.

[AS64] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55, Courier Corporation, 1964.

[Azz86] A. Azzalini, *Further results on a class of distributions which includes the normal ones*, Statistica (Bologna) **46** (1986), no. 2, 199–208. MR 877720 (88d:62025)

[Bat94] R. Battiti, *Using mutual information for selecting features in supervised neural net learning*, IEEE Transactions on Neural Networks **5** (1994), 537–550.

[BBC$^+$94] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*, vol. 43, Siam, 1994.

[BCSMAB13] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *A review of feature selection methods on synthetic data*, Knowledge and information systems **34** (2013), no. 3, 483–519.

[BCSMAB15] ———, *Recent advances and emerging challenges of feature selection in the context of big data*, Knowledge-Based Systems **86** (2015), 33–45.

[BD04] P. Buchholz and T. Dayar, *Comparison of multilevel methods for Kronecker-based Markovian representations*, Computing **73** (2004), no. 4, 349–371.

[BD07a] ———, *On the convergence of a class of multilevel methods for large sparse Markov chains*, SIAM J. Matrix Anal. Appl. **29** (2007), no. 3, 1025–1049.

[BD07b] ———, *On the convergence of a class of multilevel methods for large sparse Markov chains*, SIAM J. Matrix Analysis Applications **29** (2007), no. 3, 1025–1049.

[Bel03] A. J. Bell, *The Co-Information Lattice*, ICA 2003 (Nara, Japan), April 2003.

[BF10] J. Brannick and R. Falgout, *Compatible relaxation and coarsening in algebraic multigrid*, SIAM J. Sci. Comput. **32** (2010), no. 3, 1393–1416.

[BHS15] M. Bennasar, Y. Hicks, and R. Setchi, *Feature selection using joint mutual information maximisation*, Expert Systems with Applications **42** (2015), no. 22, 8520–8532.

[BK$^+$12] B. W. Bader, T. G. Kolda, et al., *Matlab tensor toolbox version 2.5*, Available online, January 2012.

[BKK$^+$16] M. Bolten, K. Kahl, D. Kressner, F. Macedo, and S. Sokolović, *Multigrid methods combined with amen for tensor structured Markov chains with low rank approximation*, arXiv preprint arXiv:1605.06246 (2016).

[BKRA$^+$15] F. Bagherzadeh-Khiabani, A. Ramezankhani, F. Azizi, F. Hadaegh, E. W. Steyerberg, and D. Khalili, *A tutorial on variable selection for clinical prediction models: feature selection methods in data mining could improve the results*, Journal of clinical epidemiology (2015).

[BM05] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM Journal on Scientific Computing **26** (2005), no. 6, 2133–2159.

[BMM$^+$10] M. Brezina, T. A. Manteuffel, S. F. McCormick, J. Ruge, and G. Sanders, *Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems*, SIAM J. Sci. Comput. **32** (2010), no. 1, 14–39.

[BPZL12] G. Brown, A. Pocock, M. Zhao, and M. Luján, *Conditional likelihood maximisation: A unifying framework for information theoretic feature selection*, J. Mach. Learn. Res. **13** (2012), 27–66.

[Bra00] A. Brandt, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal. **10** (2000), 1–20.

[Buc99] P. Buchholz, *Structured analysis approaches for large Markov chains*, Appl. Numer. Math. **31** (1999), no. 4, 375–404.

[Buc00] ______, *Multilevel solutions for structured Markov chains*, SIAM J. Matrix Anal. Appl. **22** (2000), no. 2, 342–357.

[Buc10] ______, *Product form approximations for communicating Markov processes*, Performance Evaluation **67** (2010), no. 9, 797–815.

[Cha87] R. Chan, *Iterative methods for overflow queueing networks I*, Numer. Math. **51** (1987), 143–180.

[CQF$^+$11] H. Cheng, Z. Qin, C. Feng, Y. Wang, and F. Li, *Conditional mutual information-based feature selection analyzing for synergy and redundancy*, ETRI Journal **33** (2011), no. 2, 210–218.

[CS02] K. Crammer and Y. Singer, *A new family of online algorithms for category ranking*, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2002, pp. 151–158.

[CT06] T. M. Cover and J. A. Thomas, *Elements of information theory*, second ed., Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2006. MR 2239987 (2007h:00002)

[Day12] T. Dayar, *Analyzing Markov chains using Kronecker products: theory and applications*, Springer Science & Business Media, 2012.

[DHS03] S. Derisavi, H. Hermanns, and W. H. Sanders, *Optimal state-space lumping in markov chains*, Information Processing Letters **87** (2003), no. 6, 309–315.

[DMS10] K. Dadkhah, H. Midi, and O. S. Sharipov, *The performance of mutual information for mixture of bivariate normal distributions based on robust kernel estimation*, Applied Mathematical Sciences **4** (2010), no. 29, 1417–1436.

[DS14] S. Dolgov and D. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM J. Sci. Comput. **36** (2014), no. 5, A2248–A2271. MR 3262607

[ETPZ09] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, *Normalized mutual information feature selection*, Neural Networks, IEEE Transactions on **20** (2009), no. 2, 189–201.

[Fle04] F. Fleuret, *Fast binary feature selection with conditional mutual information*, The Journal of Machine Learning Research **5** (2004), 1531–1555.

[Fou08] J. M. Fourneau, *Product form steady-state distribution for stochastic automata networks with domino synchronizations.*, EPEW (Nigel Thomas and Carlos Juiz, eds.), Lecture Notes in Computer Science, vol. 5261, Springer, 2008, pp. 110–124.

[FPS08] J. M. Fourneau, B. Plateau, and W. J. Stewart, *An algebraic condition for product form in stochastic automata networks without synchronizations*, Perform. Eval. **65** (2008), no. 11-12, 854–868.

[GB00] V. Goel and W. J. Byrne, *Minimum bayes-risk automatic speech recognition*, Computer Speech & Language **14** (2000), no. 2, 115–135.

[GGNZ08] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207, Springer, 2008.

[GL96] G. H. Golub and C. F. Van Loan, *Matrix computations*, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[Gre97] A. Greenbaum, *Iterative methods for solving linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[Hac03] W. Hackbusch, *Multi-grid methods and applications*, Springer, 2003.

[HBK14] N. Hoque, D. Bhattacharyya, and J. K. Kalita, *Mifs-nd: a mutual information-based feature selection method*, Expert Systems with Applications **41** (2014), no. 14, 6371–6385.

[Hit27] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Contributions from the Department of Mathematics, sn., 1927.

[HL94] G. Horton and S. T. Leutenegger, *A multi-level solution algorithm for steady-state Markov chains*, Proceedings of the ACM SIGMETRICS 1994 Conference on Measurement and Modeling of Computer Systems (B. D. Gaither, ed.), 1994, pp. 191–200.

[HLLC08] J. Huang, N. Lv, S. Li, and Y. Cai, *Feature selection for classificatory analysis based on information-theoretic criteria*, Acta Automat. Sinica **34** (2008), no. 3, 383–392. MR 2422423 (2009g:68180)

[HMRP13] J. Hillston, A. Marin, S. Rossi, and C. Piazza, *Contextual lumpability*, Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 194–203.

[HRS12] S. Holtz, T. Rohwedder, and R. Schneider, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM Journal on Scientific Computing **34** (2012), no. 2, A683–A713.

[Jak05] A. Jakulin, *Machine learning based on attribute interactions*, Ph.D. thesis, Univerza v Ljubljani, 2005.

[JCJ10] T. H. Johnson, S. R. Clark, and D. Jaksch, *Dynamical simulations of classical stochastic systems using matrix product states*, Phys. Rev. E **82** (2010), 036702.

[JW07] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis (6th Edition)*, Pearson, April 2007.

[Kar93] A. F. Karr, *Probability*, Springer Texts in Statistics, Springer-Verlag, New York, 1993. MR 1231974 (94g:60002)

[Kau83] L. Kaufman, *Matrix methods for queuing problems*, SIAM J. Sci. Statist. Comput. **4** (1983), 525–552.

[KB04] S. Kumar and W. Byrne, *Minimum bayes-risk decoding for statistical machine translation*, Tech. report, DTIC Document, 2004.

[KB09] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review **51** (2009), no. 3, 455–500.

[KC02] N. Kwak and C. Choi, *Input feature selection for classification problems*, IEEE Transactions on Neural Networks **13** (2002), no. 1, 143–159.

[KJ97] R. Kohavi and G. H. John, *Wrappers for feature subset selection*, Artificial intelligence **97** (1997), no. 1, 273–324.

[KK12] V. Kazeev and B. Khoromskij, *Low-rank explicit QTT representation of the Laplace operator and its inverse*, SIAM J. Matrix Anal. Appl. **33** (2012), no. 3, 742–758.

[KKNS13] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, *Direct solution of the chemical master equation using quantized tensor trains*, Tech. Report 2013-04, Seminar for Applied Mathematics, ETH Zürich, 2013.

[KM14] D. Kressner and F. Macedo, *Low-rank tensor methods for communicating Markov processes*, Quantitative Evaluation of Systems (Gethin Norman and William Sanders, eds.), Lecture Notes in Computer Science, vol. 8657, Springer International Publishing, 2014, pp. 25–40.

[Kot66] S. Kotz, *Recent results in information theory*, J. Appl. Probability **3** (1966), 1–93. MR 0209068 (34 #8876)

[KQB16] F. H. Khan, U. Qamar, and S. Bashir, *Swims*, Know.-Based Syst. **100** (2016), no. C, 97–111.

[KS15] V. Kazeev and C. Schwab, *Tensor approximation of stationary distributions of chemical reaction networks.*, SIAM J. Matrix Analysis Applications **36** (2015), no. 3, 1221–1247.

[KSU14] D. Kressner, M. Steinlechner, and A. Uschmajew, *Low-rank tensor methods with subspace correction for symmetric eigenvalue problems*, SIAM J. Sci. Comput. **36** (2014), no. 5, A2346–A2368.

[Kul11] V. G. Kulkarni, *Introduction to modeling and analysis of stochastic systems*, second ed., Springer Texts in Statistics, Springer, New York, 2011. MR 2743408 (2011j:60133)

[LDC$^+$15] S. Li, S. Dragicevic, F. A. Castro, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, et al., *Geospatial big data handling theory and methods: A review and research challenges*, ISPRS Journal of Photogrammetry and Remote Sensing (2015).

[Lew92] D. D. Lewis, *Feature selection and feature extraction for text categorization*, Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics, 1992, pp. 212–217.

[LH07] E. Levine and T. Hwa, *Stochastic fluctuations in metabolic pathways*, Proc. Natl. Acad. Sci. U.S.A. **104** (2007), no. 22, 9224–9229.

[LLW02] H. Liu, J. Li, and L. Wong, *A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns*, Genome informatics **13** (2002), 51–60.

[LS04a] A. N. Langville and W. J. Stewart, *The Kronecker product and stochastic automata networks*, J. Comput. Appl. Math. **167** (2004), no. 2, 429 – 447.

[LS04b] ———, *The Kronecker product and stochastic automata networks*, J. Comput. Appl. Math. **167** (2004), no. 2, 429–447. MR 2064701 (2005c:15046)

[LS04c] ———, *A Kronecker product approximate preconditioner for SANs*, Numer. Linear Algebra Appl. **11** (2004), no. 8-9, 723–752. MR 2089470 (2005k:65068)

[LT06] D. Lin and X. Tang, *Conditional infomax learning: An integrated framework for feature extraction and fusion.*, ECCV (1) (Ales Leonardis, Horst Bischof, and Axel Pinz, eds.), Lecture Notes in Computer Science, vol. 3951, Springer, 2006, pp. 68–82.

[LZO04] T. Li, C. Zhang, and M. Ogihara, *A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression*, Bioinformatics **20** (2004), no. 15, 2429–2437.

[Mac15] F. Macedo, *Benchmark problems on stochastic automata networks in tensor train format*, Tech. report, MATHICSE, EPF Lausanne, Switzerland, 2015.

[Mac16] ———, *Finding steady states of communicating Markov processes combining aggregation/disaggregation with tensor techniques*, Computer Performance Engineering: 13th European Workshop, EPEW 2016, Chios, Greece, October 5-7, 2016, Proceedings (Dieter Fiems, Marco Paolieri, and N. Agapios Platis, eds.), Springer International Publishing, Cham, 2016, pp. 48–62.

[MB06] P. E. Meyer and G. Bontempi, *On the use of variable complementarity for feature selection in cancer classification*, Applications of Evolutionary Computing, Springer, 2006, pp. 91–102.

[MB14] S. Sokolović M. Bolten, K. Kahl, *Multigrid methods for tensor structured Markov chains with low rank approximation*, arXiv preprint arXiv:1412.0937 (2014).

[MOPV17] F. Macedo, M. Rosário Oliveira, A. Pacheco, and R. Valadas, *A theoretical framework for evaluating feature selection methods based on mutual information*, ArXiv e-prints (2017), arXiv:1701.07761 [stat.ML].

[MSB08] P. E. Meyer, C. Schretter, and G. Bontempi, *Information-theoretic feature selection in microarray data using variable complementarity*, Selected Topics in Signal Processing, IEEE Journal of **2** (2008), no. 3, 261–274.

[NW06] J. Nocedal and S. J. Wright, *Numerical optimization*, second ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006. MR 2244940 (2007a:90001)

[OD12] I. V. Oseledets and S. V. Dolgov, *Solution of linear systems and matrix inversion in the TT-format*, SIAM J. Sci. Comput. **34** (2012), no. 5, A2718–A2739.

[Ose11a] I. V. Oseledets, *MATLAB TT-Toolbox Version 2.2*, 2011, Available at `http://spring.inm.ras.ru/osel/?page_id=24`.

[Ose11b] ———, *Tensor-Train decomposition*, SIAM J. Sci. Comput. **33** (2011), no. 5, 2295–2317.

[OT09] I. Oseledets and E. Tyrtyshnikov, *Breaking the curse of dimensionality, or how to use svd in many dimensions*, SIAM Journal on Scientific Computing **31** (2009), no. 5, 3744–3759.

[Pas13] C. Pascoal, *Contributions to variable selection and robust anomaly detection in telecommunications*, Ph.D. thesis, Instituto Superior Técnico, 2013.

[PFL89] B. Plateau, J.-M. Fourneau, and K.-H. Lee, *PEPS: A package for solving complex Markov models of parallel systems*, Modeling Techniques and Tools for Computer Performance Evaluation (R. Puigjaner and D. Potier, eds.), Springer US, 1989, pp. 291–305 (English).

[PLD05] H. Peng, F. Long, and C. Ding, *Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005), 1226–1238.

[PM11] I. Pultarová and I. Marek, *Convergence of multi-level iterative aggregation–disaggregation methods*, Journal of computational and applied mathematics **236** (2011), no. 3, 354–363.

[POPV16] C. Pascoal, M. R. Oliveira, A. Pacheco, and R. Valadas, *Theoretical evaluation of feature selection methods based on mutual information*, Neurocomputing (2016).

[POV$^+$] C. Pascoal, M. R. Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and pages=1755–1763 year=2012 organization=IEEE A. Pacheco, booktitle=INFOCOM, 2012 Proceedings IEEE, *Robust feature selection and robust pca for internet traffic anomaly detection*.

[PS97] B. Plateau and W. J. Stewart, *Stochastic automata networks*, Computational Probability, Kluwer Academic Press, 1997, pp. 113–152.

[PSS96] B. Philippe, Y. Saad, and W. J. Stewart, *Numerical methods in Markov chain modelling*, Operations Research **40** (1996), 1156–1179.

[Ros00] S. M. Ross, *Introduction to probability models*, seventh ed., Harcourt/Academic Press, Burlington, MA, 2000. MR 1766683 (2001b:60002)

[RS86] J. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid Methods (McCormick, S.F., ed.) (1986).

[RVZ10] B. Reps, W. Vanroose, and H. B. Zubair, *Gmres-based multigrid for the complex scaled preconditoner for the indefinite helmholtz equation*, arXiv preprint arXiv:1012.5379 (2010).

[RY02] M. Rogati and Y. Yang, *High-performing feature selection for text classification*, Proceedings of the eleventh international conference on Information and knowledge management, ACM, 2002, pp. 659–661.

[Saa03] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[Sch11] U. Schollwöck, *The density-matrix renormalization group in the age of matrix product states*, Annals of Physics **326** (2011), no. 1, 96–192.

[Sha48] C. E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27** (1948), 379–423, 623–656. MR 0026286 (10,133e)

[SIL07] Y. Saeys, I. Inza, and P. Larrañaga, *A review of feature selection techniques in bioinformatics*, bioinformatics **23** (2007), no. 19, 2507–2517.

[SS86] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput. **7** (1986), no. 3, 856–869.

[SS97] M. Sidi and D. Starobinski, *New call blocking versus handoff blocking in cellular networks*, Wirel. Netw. **3** (1997), no. 1, 15–27.

[SS00] R. E. Schapire and Y. Singer, *Boostexter: A boosting-based system for text categorization*, Machine learning **39** (2000), no. 2, 135–168.

[Ste94] W. J. Stewart, *Introduction to the numerical solution of Markov chains*, Princeton University Press, 1994.

[Tak75] Y. Takahashi, *A lumping method for numerical calculations of stationary distributions of Markov chains*, B-18, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan (1975).

[TOS01] U. Trottenberg, C. Osterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.

[VE10] J. R. Vergara and P. A. Estévez, *CMIM-2: an enhanced conditional mutual information maximization criterion for feature selection*, Journal of Applied Computer Science Methods **2** (2010).

[VE14] ———, *A review of feature selection methods based on mutual information*, Neural Computing and Applications **24** (2014), no. 1, 175–186.

[VMAF13] P. Varela, A. Martins, P. Aguiar, and M. Figueiredo, *An empirical study of feature selection for sentiment analysis*, 9th conference on telecommunications. Conftele 2013, Castelo Branco, 2013.

[VZCB15] N. X. Vinh, S. Zhou, J. Chan, and J. Bailey, *Can high-order dependencies improve mutual information based feature selection?*, Pattern Recognition (2015).

[Wat60] S. Watanabe, *Information theoretical analysis of multivariate correlation*, IBM Journal of research and development **4** (1960), no. 1, 66–82.

[Whi92] S. R. White, *Density matrix formulation for quantum renormalization groups*, Physical Review Letters **69** (1992), no. 19, 2863.

[Whi05] ———, *Density matrix renormalization group algorithms with a single center site*, Phys. Rev. B **72** (2005), 180403.

[XJK$^+$01] E. P. Xing, M. Jordan, R. M. Karp, et al., *Feature selection for high-dimensional genomic microarray data*, ICML, vol. 1, Citeseer, 2001, pp. 601–608.

[YM99] H. H. Yang and J. Moody, *Data visualization and feature selection: New algorithms for nongaussian data*, in Advances in Neural Information Processing Systems, MIT Press, 1999, pp. 687–693.

[YP97] Y. Yang and J. O. Pedersen, *A comparative study on feature selection in text categorization*, ICML, vol. 97, 1997, pp. 412–420.

[YWDP14] K. Yu, X. Wu, W. Ding, and J. Pei, *Towards scalable and accurate online feature selection for big data*, Data Mining (ICDM), 2014 IEEE International Conference on, IEEE, 2014, pp. 660–669.

Curriculum Vitae

Francisco Macedo
Born: December 5th, 1990 in Lisbon, Portugal
Nationality: Portuguese

Education

Oct 2012 - ongoing	Doctoral studies Advisors: Prof. D. Kressner and Prof. A. Pacheco École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland Instituto Superior Técnico, Lisbon, Portugal Topic: *Low-rank tensor methods for large Markov chains and forward feature selection methods*
Oct 2011 - Sep 2012	Master in Mathematics and Applications Instituto Superior Técnico, Lisbon, Portugal Master thesis with Prof. C. Nunes: *Investment policies in competitive products*
Oct 2008 - Jul 2011	Bachelor in Applied Mathematics and Computation Instituto Superior Técnico, Lisbon, Portugal

Publications

2017	F. Macedo, M. Rosário Oliveira, A. Pacheco, R. Valadas: *A theoretical framework for evaluating forward feature selection methods based on mutual information* Online at arxiv.org and submitted to J. Mach. Learn. Res.
2016	M. Bolten, K. Kahl, D. Kressner, F. Macedo, S. Sokolović: *Multigrid methods combined with low-rank approximation for tensor structured Markov chains* To appear in Electron. Trans. Numer. Anal.
2015	F. Macedo: *Benchmark problems on stochastic automata networks in tensor train format* Technical report 26.2015 in Mathematics Institute of Computational Science and Engineering (MATHICSE)

Conference proceedings

2016	F. Macedo: *Finding steady states of communicating Markov processes combining aggregation/disaggregation with tensor techniques* Proceedings of the 13th European Workshop on Performance Engineering (EPEW), Chios, Greece, October 5 – 7
2014	D. Kressner, F. Macedo: *Low-rank tensor methods for communicating Markov processes* Proceedings of the 11st International Conference on Quantitative Evaluation of Systems (QEST), Florence, Italy, September 8 – 10

Conference contributions

Oct 5–7, 2016	13th European Workshop on Performance Engineering, Chios, Greece Talk: *Finding steady states of communicating Markov processes combining aggregation/disaggregation with tensor techniques*
Nov 9–12, 2015	PhD Open Days, Lisbon, Portugal Poster: *Low-rank tensor methods for communicating Markov processes*
Jul 13–15, 2015	MATHICSE Retreat, Leysin, Switzerland
Apr 17, 2015	Swiss Numerical Analysis Day 2015, Université de Genève, Geneva, Switzerland Poster: *Low-rank tensor methods for communicating Markov processes*
Jan 18–20, 2015	Rigi Workshop, Rigi, Switzerland Poster: *Low-rank tensor methods for communicating Markov processes*
Feb 19, 2014	Seminar of Research in Probability and Statistics II, Instituto Superior Técnico, Lisbon, Portugal Invited talk: *A low-rank tensor method for structured large-scale Markov Chains*
Nov 29 – Dec 2, 2013	XXI Congresso Anual da Sociedade Portuguesa de Estatística, University of Aveiro, Aveiro, Portugal Talk: *A low-rank tensor method for structured large-scale Markov Chains*
Jul 14–16, 2014	Joint ALAMA-GAMM/ANLA meeting, Barcelona, Spain
Aug 26–30, 2013	ENUMATH 2013, EPF Lausanne, Lausanne, Switzerland Talk: *A low-rank tensor method for structured large-scale Markov Chains*
Jul 1–5, 2013	Preconditioning of Iterative Methods, Prague, Czech Republic Talk: *A low-rank tensor method for structured large-scale Markov Chains*
Apr 5, 2013	Swiss Numerics Colloquium, EPF Lausanne, Lausanne, Switzerland
Sep 26–29, 2012	XX Congresso Anual da Sociedade Portuguesa de Estatística, Porto, Portugal
Apr 17–20, 2012	XXXIII Congreso Nacional de Estadística e Investigación Operativa, Madrid, Spain Talk: *Optimal investment policies*

Oct 12, 2011	Seminário Diagonal, Instituto Superior Técnico, Lisbon, Portugal Invited talk: *Sistemas de votação justos em eleições parlamentares*
Sep 28 – Oct 1, 2011	IX Congresso Anual da Sociedade Portuguesa de Estatística, Nazaré, Portugal Poster: *Leis da genética de Mendel: a enriquecedora controvérsia*
Mar 23, 2011	Seminário Diagonal, Instituto Superior Técnico, Lisbon, Portugal Invited talk: *Coisas verdes que andam à roda*