FRIEDRICH-SCHILLER-UNIVERSITY JENA

Faculty of Mathematics and Computer Science

FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Multivariate Models with Higher-Order Interactions: Theory and Algorithms

Master Thesis

A thesis submitted by

Christoph Saffer, born on January 12, 1993 in Bamberg

in fulfillment of the requirements

for the degree Master of Science (M. Sc.)

Supervisors: Prof. Dr. Joachim Giesen
Frank Nussbaum

Jena, December 12, 2019

Zusammenfassung

In dieser Arbeit wird ein theoretischer und praktischer Ansatz zur Modellierung von binären Daten vorgestellt. Genauer gesagt untersuchen wir multivariate Interaktionsmodelle auf binären Variablen, die auf den Interaktionen zwischen einer Gruppe von K Variablen basieren. Im Gegensatz zu vollständigen kategorischen Modellen, bei denen jedes mögliche Ergebnis durch einen Parameter modelliert wird, wächst die Komplexität der hier vorgestellten Modellklasse nur polynomiell in der Anzahl der Variablen.

Wir geben eine grundlegende Ausarbeitung zur Theorie der multivariaten Interaktionsmodelle auf binären Variablen und untersuchen den abstrakten Übergang von paarweisen Interaktionen zu Interaktionen höherer Ordnung. Darüber hinaus unterscheiden wir zwischen verschiedenen Möglichkeiten der Darstellung der Modellverteilung, was für die Implementierung von Schätzverfahren oder Inferenzabfragen wichtig ist, da die Parameter in Tensoren höherer Ordnung gespeichert werden und damit in einer anderen Weise darauf zugegriffen wird. Dies ermöglicht uns moderne Berechnungstechniken wie Vektorisierung oder Parallelisierung anzuwenden, um eine höhere Performance zu erlangen, was bei großen Datenmengen immer eine wesentliche Rolle einnimmt.

Später behandeln wir auch das Schätzen der Parameter des Modells, d. h. wir gehen auf die Konstruktion der Pseudo-Likelihood Funktion und des entsprechenden Optimierungsproblems ein. In diesem Zusammenhang liefern wir geeignete Konzepte für die Regularisierung, um die Komplexität des Modells weiter zu reduzieren. Dabei vergleichen wir verschiedene Ansätze wie ℓ_1-norm und overlapped Schatten-1-norm Regularisierung. Am Ende passen wir den Frank-Wolfe Optimierungsalgorithmus für konvexe Probleme an unser Problem an und leiten ihn insbesondere für paarweise Interaktionen her. Wir zeigen auch warum dieser Algorithmus sich für Interaktionen höherer Ordnung schwieriger anpassen lässt und woher strukturelle Probleme (insbesondere im Hinblick auf die Regularisierung) kommen.

Abstract

In this work, a theoretical and practical approach to model binary data is presented. More accurately, we examine multivariate interaction binary models that are based on the interactions between a group of K variables. In contrast to entire categorical models where each possible outcome is modeled by an individual parameter, the complexity of the model class presented here only grows polynomially in the number of variables.

We give a fundamental elaboration of the theory of multivariate interaction models on binary variables and highlight the abstract transition from pairwise to higher-order interactions. Furthermore, we distinguish between various methods of representing the model, which is important in terms of implementing its likelihood or inference queries, since the parameters are stored within tensors of higher orders. This allows us to apply modern computational techniques such as vectorization or parallelization with high performance, which is always an essential issue when it comes to large amounts of data.

Later, we also go into model selection which means the construction of the likelihood and the corresponding optimization problem in order to estimate the parameters of the model. In this context we provide suitable concepts for regularization to take complexity from the model. Hereby we compare different approaches such as ℓ_1-norm and overlapped Schatten-1-norm regularization. In the end, we adjust the Frank-Wolfe optimization algorithm for convex optimization problems to our problem and derive it for pairwise interactions in particular. We also demonstrate why this becomes difficult for higher-order interactions, as well as where structural problems (especially in terms of regularization) stem from.

Contents

Chapter 1

Introduction

In statistical theory and more applied disciplines as machine learning, it is a fundamental research goal to find methods to reduce the number of parameters in probabilistic models. In this work, we will study a class of multivariate models on categorical variables, in other words, models that include variables with a discrete set of outcomes. More specifically, we will consider multivariate binary models or so-called Ising Models, whose variables only have two possible outcomes. In the standard approach, the number of parameters of an entire categorical model grows exponentially with the number of variables. For instance, a model with n binary variables (corresponding to n features in the sample) has $2^n - 1$ parameters, because each combination of all variables is encoded within one parameter. To fit a model to a set of data, the general opinion suggests that roughly ten times more observations than parameters are necessary in order to get a descriptive, meaningful model, but this requirement can often not be satisfied.

To meet the problem of a growing number of parameters, on the one hand, we have the option to use regularization methods as sparsity or low-rank constraints, while fitting the model, with the purpose to set insignificant parameters to zero or find dependencies within the parameters. On the other hand, we can use a different approach to model the data with fewer parameters from the beginning.

We will concentrate on a specific class of probabilistic models, where the focus is set on the interactions between a group of K variables. Here, the number of parameters grows polynomially with the number of variables n, where the exact growth rate depends on how many interactions between variables we consider. This model type combined with methods to reduce the number of parameters through deactivation or dependencies delivers a well parameterized model. Its formulation, analysis and optimization are the goal of this work.

1.1 Idea of Variable Interactions

The idea of variable interactions is based on the assumption that the outcome of one variable depends on the outcomes of the remaining variables. In a statistical model this is encoded by an additional term that models the interaction. When we consider pairwise interactions, an interaction term of the variables x_i and x_j would

be given by

$$q_{ij}\ x_i\ x_j$$

with the interaction parameter q_{ij}. The interaction parameter indicates how much the interaction between x_i and x_j influences the model. An interaction term of the variables x_i, x_j, x_k has the shape

$$q_{ijk}\ x_i\ x_j\ x_k$$

with the interaction parameter q_{ijk}. Let $K = 2$ be the interaction order and $n = 3$ the dimension (number of variables). The entire model with the variables x_1, x_2 and x_3 would include the terms

$$q_1x_1 + q_2x_2 + q_3x_3 + q_{12}x_1x_2 + q_{13}x_ix_3 + q_{23}x_2x_3$$

with the parameter vector $\theta = (q_1,\ q_2,\ q_3,\ q_{12},\ q_{13},\ q_{23})$. Here, we get six parameters for interaction order $K = 2$ and dimension $n = 3$. When we increase the interaction order to $K = 3$, we would have an additional interaction term $q_{123}x_1x_2x_3$ and seven parameters in total, the same as for the entire categorical model ($2^3 - 1 = 7$). In this context, we already get a first impression that the number of interactions K should be lower than the number of variables n in order to gain a benefit in terms of model complexity.

Remark. The complexity of a model is equal to the size of its parameter vector θ (number of parameters) of the respective model.

Lemma 1.1. *Let K be the interaction order, n the number of variables and $K \leq n$, then the complexity of the corresponding multivariate binary model of interaction order K is calculated with*

$$|\theta| = \sum_{i=1}^{K} \binom{n}{i}.$$

Proof. Let $K \leq n$. We count all univariate parameters and get $\binom{n}{1} = n$ parameters. Then, we count all pairwise interactions where each variable is combined with one of the others. We get $\binom{n}{2}$. In general, for $i \leq K$, we count all possible combinations of i variables from n variables as $\binom{n}{i}$. We continue until K interactions. □

Conclusion 1.2. *Let K be the interaction order, n the number of variables and $K = n$. Then, the complexity of the corresponding multivariate binary model of interaction order K is equal to the complexity of the entire categorical model and grows exponentially in n:*

$$|\theta| = \sum_{i=1}^{n} \binom{n}{i} \ = \ 2^n - 1$$

Hence, to model higher-order interactions between variables with an interaction order equal to the number of variables, the number of parameters grows as quickly as the complexity of the entire binary model, as illustrated in Table 1.1. Consequently, the order of interactions K should be kept relatively small in comparison to the number of variables n to get a significant benefit, otherwise the entire categorical model simply can be used. Therefore, we will mostly consider models with a rather low order of interaction.

Var. n \ Order K	2	3	4	5	6	$2^n - 1$
2	3	-	-	-	-	3
3	6	7	-	-	-	7
4	10	14	15	-	-	15
5	15	25	30	31	-	31
6	21	41	56	62	63	63

Table 1.1: Complexity of an multivariate model of interaction order K and dimension n. In comparison, the complexity of an entire binary model on the right.

1.2 Multivariate Binary Models of Pairwise Interactions

1.2.1 Concept and Definitions

In the following sections, we define and introduce a multivariate model of pairwise interactions on binary variables (see [6]) and try to gain a general understanding as to why it is reasonable to use that model class.

Definition. Let $X = (X_1, \ldots, X_n)$ be a vector of random variables with the set of outcomes $\Omega = \{0, 1\}^n$. The distribution $p \colon \Omega \to [0, 1]$ of a *multivariate binary model of pairwise interactions* has the form

$$p(x) \propto \exp\Big(\sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n} \sum_{j \neq i}^{n} q_{ij} x_i x_j \Big)$$

with parameters $q_{ij} \in \mathbb{R}$ if the normalization condition

$$\sum_{x \in \Omega} p(x) = 1$$

is satisfied.

Only to create clarity about the way the parameters in the binary model work, it holds for the interaction term $q_{ij} x_i x_j$ that

$$q_{ij} x_i x_j = \begin{cases} q_{ij} & , x_i = 1 \wedge x_j = 1 \\ 0 & , x_i = 0 \vee x_j = 0 \end{cases}.$$

It means, the parameter q_{ij} only influences the model, if both of its input variables x_i and x_j are activated (equal to 1).

Remark. The parameter q_{ij} models the interaction between the variables x_i and x_j. Evidently, the parameter q_{ji} models the same interaction. Therefore, parameters with permuted index are assumed to be equal. It is

$$q_{ij} = q_{ji}.$$

Furthermore, it is p_θ the model distribution corresponding to the parameter vector $\theta = \{q_1, q_2, \ldots, q_{n-1n}\}$ containing the unique parameters of the model.

Moreover, we find another representation of the distribution of a multivariate binary model of pairwise interactions. We aim to have a more intuitive, compact description of the parameter vector θ. Therefore, we write the parameters q_{ij} within a matrix

$$Q = Q(\theta) = \{q_{ij}\}_{i,j=1,\dots,n} \in \mathbb{R}^{n\times n},$$

whereby $q_{ii} := q_i$. Then, for the representation of the model p_θ, we also get

$$p_\theta(x) \propto \exp\Big(\sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n}\sum_{j\neq i}^{n} q_{ij} x_i x_j \Big) = \exp(x^{\mathrm{T}} Q x)$$

as the distribution p_Q with parameter matrix Q. That is equivalent to the distribution p_θ with parameter vector θ.

Remark. The matrix Q, containing the parameters of the respective model p_Q, is symmetric, since we assumed parameters with permuted index to be equal.

The matrix representation also promises advantages regarding the implementation of the model class, since matrix-vector multiplication can be evaluated much faster than computing over sums with restrictions (for-loops with if-clauses).

Before we proceed with the identifiability of the model, we have a closer look at the necessary normalization coefficient. The model distribution $p(x)$, as defined above, contains a normalization coefficient z to satisfy the normalization condition. The value of the normalization coefficient can be identified by rearranging the equation of the condition:

$$\begin{aligned} 1 = \sum_{x\in\Omega} p(x) &= \sum_{x\in\Omega} \frac{1}{z} \exp\Big(\sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n}\sum_{j\neq i}^{n} q_{ij} x_i x_j\Big) = \sum_{x\in\Omega} \frac{1}{z} \exp(x^{\mathrm{T}} Q x) \\ \iff z &= \sum_{x\in\Omega} \exp(x^{\mathrm{T}} Q x) \end{aligned}$$

The normalization coefficient z is the sum of all possible combinations of the input vector x plugged into the model term. Here, possible numerical difficulties for higher n must be considered, since z has 2^n summands. Now, we would like to discuss the identifiability of the model.

Lemma 1.3. *Let p_θ be a multivariate binary model of pairwise interactions. Then, there is an injective mapping $\theta \to p_\theta$, i.e. the model p_θ is identifiable.*

Proof. Let p_θ and $p_{\theta'}$ be two multivariate binary models of pairwise interactions. One must show that $\theta = \theta'$ follows from the assumption

$$p_\theta = p_{\theta'}. \tag{1.1}$$

It is (1.1) equivalent to

$$\begin{aligned} 0 &= \log(p_\theta) - \log(p_{\theta'}) \\ &= -\log(z_\theta) + \sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n}\sum_{j\neq i}^{n} q_{ij} x_i x_j + \log(z_{\theta'}) - \sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n}\sum_{j\neq i}^{n} q_{ij} x_i x_j \\ &= \log(z_{\theta'}) - \log(z_\theta) + \sum_{i=1}^{n} (q_i - q_i') x_i + \sum_{i=1}^{n}\sum_{j\neq i}^{n} (q_{ij} - q_{ij}') x_i x_j. \end{aligned}$$

We show that (1.1) is satisfied for all $x \in \{0,1\}^n$ if $q_i = q'_i$ for all i and $q_{ij} = q'_{ij}$ for all i, j, since all summands become zero. Since the normalization terms z_θ, $z_{\theta'}$ are independent from x, we define $\Delta := \log(z_{\theta'}) - \log(z_\theta)$. Let $x = e_i$ be a vector with only zeros but one at the i-th position. It follows from $p_\theta(x) = p_{\theta'}(x)$ for all i that

$$0 = \Delta + q_i - q'_i. \tag{1.2}$$

Let $x = e_{ij}$ be a vector with only zeros but ones at the i-th and j-th position. Since $-\Delta = q_i - q'_i$, it follows from $p_\theta(x) = p_{\theta'}(x)$ for all i, j that

$$0 = \Delta + q_i - q'_i + q_j - q'_j + q_{ij} - q'_{ij} = -\Delta + q_{ij} - q'_{ij}. \tag{1.3}$$

Let $x = e_{ijk}$ be a vector with only zeros but ones at the i-th, j-th and k-th position. Since $-\Delta = q_i - q'_i$ and $\Delta = q_{ij} - q'_{ij}$, it follows from $p_\theta(x) = p_{\theta'}(x)$ for all i, j, k that

$$\begin{aligned} 0 &= \Delta + q_i - q'_i + q_j - q'_j + q_k - q'_k + q_{ij} - q'_{ij} + q_{ik} - q'_{ik} + q_{jk} - q'_{jk} \\ &= \Delta - 3\Delta + 3\Delta \\ &= \Delta. \end{aligned}$$

So, $\Delta = 0$. It follows from (1.2) that $q_i = q'_i \; \forall i$ and then, it follows from (1.3) that $q_{ij} = q'_{ij} \; \forall i, j$. That is, we have $\theta = \theta'$. Therefore, the mapping is injective and the model is identifiable. □

After introducing the general shape of a multivariate binary model of pairwise interactions, we want to fit the parameters Q based on data that is distributed by a binary distribution p_Q on a probability space Ω.

1.2.2 Pseudo-Likelihood Approach

We are given d data points $x^{(1)}, \ldots, x^{(d)}$ with each $x^{(i)} \in \{0,1\}, i = 1, \ldots, d$ and $\Omega = \{0,1\}^n$ that are independently drawn from a multivariate binary distribution p on Ω. To find the optimal choice of parameters, we minimize the negative likelihood function

$$\mathbf{p}_{ML} = \arg\min_Q -L(Q) = \arg\min_Q -\prod_{i=1}^{d} p_Q(x^{(i)}),$$

or equivalently minimizing the negative log-likelihood function

$$\mathbf{p}_{ML} = \arg\min_Q -\ell(Q) = \arg\min_Q -\sum_{i=1}^{d} \log p_Q(x^{(i)}).$$

Unfortunately, the minimization problem does not have an analytical solution, therefore we will use optimization algorithms to find a solution for the parameter estimation problem. Because the normalization factor is usually the bottleneck task regarding the computation and implementation of a correctly working and high-performance optimization algorithm on categorical models, we will use the pseudo-likelihood function instead. The pseudo-likelihood function is an approach to approximate the likelihood function, but with less complexity because the normalization constants are easier to track.

In essence, we assume that

$$p_Q(x) = p_Q(x_1, \ldots, x_n) \approx p_Q(x_1|x_{-1}) \ \ldots \ p_Q(x_n|x_{-n}) = \prod_{r=1}^{n} p_Q(x_r|x_{-r}),$$

where $p(x_r|x_{-r})$ is the conditional probability of x_r given the variables $x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$:

$$p_Q(x_r|x_{-r}) = p_Q(x_r|x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) = \frac{\exp(2 \sum_{j=1}^{n} q_{rj} x_r x_j - q_{rr} x_r x_r)}{1 + \exp(q_{rr} + 2 \sum_{j=1, j\neq r}^{n} q_{rj} x_j)}$$

The conditional probability includes all terms of the probability distribution that contain the index r. The normalization in the denominator also includes all terms that contain the index r summed up over all outcomes of x_r. In the case of binary variables, it is summed up for $x_r = 0$ and $x_r = 1$.

Remark. We call the term $p_Q(x_r|x_{-r})$ the *node conditional* of the r-th variable.

We get the log-pseudo-likelihood function as

$$\ell_p = \log L_p = \log \prod_{i=1}^{d} \prod_{r=1}^{n} p_Q(x_r^{(i)}|x_{-r}^{(i)}) = \sum_{i=1}^{d} \sum_{r=1}^{n} \log p_Q(x_r^{(i)}|x_{-r}^{(i)})$$

and derive the maximum likelihood estimator with negative log-pseudo-likelihood function as follows:

$$\begin{aligned}
\mathbf{p}_{ML} &= \arg\min_Q -\ell(Q) \\
&\approx \arg\min_Q -\ell_p(Q) \\
&= \arg\min_Q -\sum_{i=1}^{d} \sum_{r=1}^{n} \log p_Q(x_r^{(i)}|x_{-r}^{(i)}) \\
&= \arg\min_Q -\sum_{i=1}^{d} \sum_{r=1}^{n} \log\Big(\frac{\exp(2 \sum_{j=1}^{n} q_{rj} x_r^{(i)} x_j^{(i)} - q_{rr} x_r^{(i)} x_r^{(i)})}{1 + \exp(q_{rr} + 2 \sum_{j=1, j\neq r}^{n} q_{rj} x_j^{(i)})} \Big) \\
&= \arg\min_Q -\sum_{i=1}^{d} \sum_{r=1}^{n} \Big(2 \sum_{j=1}^{n} q_{rj} x_r^{(i)} x_j^{(i)} - q_{rr} x_r^{(i)} x_r^{(i)} \\
&\qquad\qquad - \log(1 + \exp(q_{rr} + 2 \sum_{j=1, j\neq r}^{n} q_{rj} x_j^{(i)})) \Big)
\end{aligned}$$

Now that we have derived the optimization problem whose solution estimates the parameters Q, we will look at an example and attempt to fit a multivariate model or pairwise interactions to binary data.

1.3 An Introductory Example

We will look at a concrete example in which we apply a synthetic dataset to a multivariate binary model of pairwise interactions by minimizing the negative log-pseudo-likelihood function to get a better understanding of how the model works.

The generated dataset has three features $(X, Y, Z) \in \{0, 1\}^3$ and 10.000 entries. The frequency of the single outcomes can be seen in Table 1.2:

(X, Y, Z)	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
#	983	2105	4172	1849	11	612	60	208

Table 1.2: Composition of the dataset with frequency of all outcomes.

If we used an entire categorical model with probability distribution p_c, we would have one parameter for each outcome and find its probability through maximum likelihood by

$$p_C(X, Y, Z) = \frac{\#(X, Y, Z)}{10.000}$$

This would yield 8 distinct outcomes, resulting in 7 parameters.

Now, we fit the dataset to a multivariate binary model of pairwise interactions. It has interaction order $K = 2$ and dimension $n = 3$. We minimize the negative log-pseudo-likelihood function

$$\arg\min_Q -\ell_p(Q)$$

and for the parameter vector $\theta = \{q_1, q_2, q_3, q_{12}, q_{13}, q_{23}\}$ we get the following result:

$$\theta = \{-3.6605,\ 1.4626,\ 0.7821,\ -0.4238,\ 1.1982,\ -0.8032\}.$$

We compute the solution of the minimization problem with an optimization package that will be discussed in Section 4.6. The parameter vector θ can be written within a symmetric matrix

$$Q = Q(\theta) = \begin{pmatrix} -3.6605 & -0.4238 & 1.1982 \\ -0.4238 & 1.4626 & -0.8032 \\ 1.1982 & -0.8032 & 0.7821 \end{pmatrix}$$

which yields the probability distribution p_Q of the multivariate binary model of pairwise interactions. as

$$p_Q(X, Y, Z) = \frac{1}{z}\ \exp(x^{\mathrm{T}} Q x)$$

with $x = (X, Y, Z)$ and normalization coefficient z. Now, we calculate the probability of each outcome, according to the model distribution p_Q to verify how close they are to the parameters of the p_c distribution. It is compared in the Table 1.3.

(X, Y, Z)	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
p_C	0.0983	0.2105	0.4172	0.1849	0.0011	0.0612	0.0060	0.0208
p_Q	0.0969	0.2119	0.4185	0.1835	0.0025	0.0599	0.0046	0.0222

Table 1.3: Prediction according to the distribution p_C respectively p_Q for each possible outcome in the dataset.

As we can see, we achieved acceptable results that are close to the real probabilities with one parameter fewer than in the entire categorical model. The idea is to save increasingly more parameters by an increasing number of variables. Also, we will use further techniques such as sparse and low-rank regularization to find the parameters which are expandable to describe the data. We can set them to zero and reduce the number of interaction parameters even further. First, however, we will examine the model class for higher-order interactions.

Chapter 2

Model Class and its Likelihoods

2.1 Multivariate Binary Models of Interaction Order $K = 3$

In the following, we will extend the idea of pairwise interactions that we have seen in the previous chapter. Now, we examine models of interaction order $K = 3$, which means that we also model and specify interactions between three distinct variables. Increasing the number of interactions promises a more accurate description of the data in exchange for a slightly higher number of parameters.

2.1.1 Concept and Definitions

Definition. Let $X = (X_1, \dots, X_n)$ be a vector of random variables with the set of outcomes $\Omega = \{0, 1\}^n$. The distribution $p\colon \Omega \to [0, 1]$ of a *multivariate binary model of interaction order* $K = 3$ has the form

$$p(x) = p_Q(x) \propto \exp\Big(\sum_{i,j,k}^{n} q_{ijk} x_i x_j x_k \Big)$$

with the parameter tensor $Q = \{q_{ijk}\}_{i,j,k=1,\dots,n} \in \mathbb{R}^{n \times n \times n}$ if the normalization condition

$$\sum_{x \in \Omega} p(x) = 1$$

is satisfied.

Remark. The sum $\sum_{i_1=1}^{n} \dots \sum_{i_K=1}^{n}$ can be written as $\sum_{i_1,\dots,i_K}^{n}$. Furthermore, we call Q of the previous definition a *tensor* of order $K = 3$ of the multivariate binary model of interaction order $K = 3$. The number of variables n of the model corresponds to the dimension of the tensor.

In the following, we will informal introduce and clarify the concept of strong symmetry for tensors of order $K = 3$ and show the difference to normal symmetry. This concept will be generally defined in the Section 2.2.1.

As it was pointed out in the previous chapter, a parameter q_{ijk} describes the

interaction between the variables x_i, x_i and x_k. Therefore, we consider parameters $q_{\sigma(i)\sigma(j)\sigma(k)}$ with permuted indices as equal, because they describe the same interaction. It is

$$q_{ijk} = q_{ikj} = q_{jik} = q_{jki} = q_{kij} = q_{kji}.$$

Here, the permutation function σ is defined on the set $\{i, j, k\}$.

It also follows, that the indices of a parameter indicate the affected variables. Therefore, we shorten the parameters q_{ijk} such that no index occurs more than once, because we only need to know whether a variable x_i is affected by a parameter or not. We write

$$q_{iii} := q_i, \quad q_{iij} := q_{ij}.$$

For instance, when we consider a model of interaction order $K = 3$, the parameter q_{ij} is equivalent to the parameters q_{iij} and q_{ijj} but we write it as one, because both describe the same interaction between variable x_i and x_j. Altogether this leads to a strong symmetry concept for the tensor Q, where also

$$q_{iij} = q_{iji} = q_{jii} = q_{ijj} = q_{jij} = q_{jji}.$$

If we only required normal symmetry (equality of permuted indices), it would be

$$q_{iij} \neq q_{ijj}$$

because on the left there are two instances of i, a case which is not in the permutation group of an index with one i (on the right). For shortened notation, however, where only one instance of each index is displayed, they are each other's permutation. Therefore, it is a stronger property. We call the tensor Q of the model p_Q *strongly symmetric* which is assumed in the following sections.

Example. We consider a multivariate binary model of interaction order $K = 3$ and dimension $n = 3$. The associated tensor Q has $3^3 = 27$ parameters which are summarized in 7 distinct parameters, namely

$$\theta = \{q_1,\ q_2,\ q_3,\ q_{12},\ q_{13},\ q_{23},\ q_{123}\},$$

that describe the model.

2.1.2 Representation of the Node Conditional

As we proceeded in the first chapter, we will also look at the pseudo-likelihood function in order to estimate the parameters of the respective model. Therefore, we have to pick up on first the node conditional of the r-th variable in the binary model of interaction order $K = 3$. The node conditional $p(x_r|x_{-r})$ describes slices that we cut out of the tensor Q for the r-th variable. In other words, we are interested in all parameters that contain index r. When we consider strong symmetry of the associated tensor Q, we also find that some parameters in the node conditional have more than one occurrence. These occurrences can be counted, categorized into groups, and summarized as one expression.

Example. We consider the tensor Q for $K = 3$ and $n = 3$. We derive the node conditional for $r = 1$. We list all parameters that contain index $r = 1$ and categorize them by using the shortened parameter notation and strong symmetry:

$$\begin{aligned} 6\, q_{123} &= q_{123} + q_{132} + q_{213} + q_{231} + q_{312} + q_{321} \\ 6\, q_{12} &= q_{112} + q_{121} + q_{211} + q_{122} + q_{212} + q_{221} \\ 6\, q_{13} &= q_{113} + q_{131} + q_{311} + q_{133} + q_{313} + q_{331} \\ q_1 &= q_{111} \end{aligned}$$

For the node conditional, we get

$$p(x_1|x_{-1}) \propto \exp(\, 6\, q_{123}x_1x_2x_3 + 6\, (q_{12}x_1x_2 + q_{13}x_1x_3) + q_1x_1).$$

Definition. ([4], p. 132) An m-partition of a positive integer K is a representation of K as the sum of m positive integers. For $k_1 \geq \ldots \geq k_m, k_i \in \mathbb{N}$, it is

$$K = k_1 + \ldots + k_m.$$

Let us generalize the previous approach with the following lemma.

Lemma 2.1. *For the node conditional of the r-th variable in a multivariate binary model of interaction order $K = 3$, we get*

$$p_Q(x_r|x_{-r}) \propto \exp(\, 6 \sum_{i<j, i\neq r, j\neq r}^{n} q_{rij}x_rx_ix_j + 6\sum_{i\neq r}^{n} q_{ri}x_rx_i + q_rx_r). \tag{2.1}$$

Proof. We will count the occurrences of the parameters q_{ijk} within the node conditional. Therefore, we must count the occurrences for each m-partition of $K = 3$.

At first, we count the combinations for the first term $\sum q_{rij}x_rx_ix_j$ where all indices are distinct. Therefore, we look at the $m = 3$ partition of $K = 3$. We only have one partition:

$$3 = 1 + 1 + 1$$

This means all parameters are distinct. We count the combinations that are represented by q_{rij}:

$$q_{rij} = q_{rji} = q_{irj} = q_{ijr} = q_{jri} = q_{jir}.$$

We get $6 = 3!$ combinations for the $m = 3$ partition of $K = 3$. Therefore, we get 6 as the first coefficient.

The next step is to count combinations for the second term $\sum q_{ri}x_rx_i$ of the $m = 2$ partition of $K = 3$. The term q_{ri} represents the parameters q_{rri} and q_{rii}. That stems from that we have the two partitions

$$3 = 2 + 1, \quad 3 = 1 + 2.$$

We count the combinations that are represented by the parameter q_{rri}:

$$q_{rri} = q_{rir} = q_{irr}$$

We get $3 = \binom{3}{2}$ combinations for q_{rri}, but since we have $m = 2$ partitions of $K = 3$, we get $2 \cdot 3 = 6$ distinct combinations which is the second coefficient.

For the last of the possible partitions, the $m = 1$ partition of $K = 3$, trivially we get only one combination (last coefficient) for q_{rrr} that is represented by q_r. □

Unfortunately, the formula we derived is difficult to use in terms of implementation. Because of its restrictions in the sums it would lay skewed in the memory and therefore high-performance techniques such as vectorization would become complicated to apply. However, we can also think of the node conditional $p(x_r|x_{-r})$ as slices that we cut out of n-dimensional tensor of order K at the r-th position for the r-th variable. That approach leads to an easier formula for the node conditional which we will introduce in the following proposition and lemma.

Remark. In all following sums, the index r is fixed, except it is specified otherwise.

Proposition 2.2. *We have the following decompositions of the sums:*

$$\sum_{i,j}^{n} q_{rij} x_r x_i x_j = 2 \sum_{i<j, i\neq r, j\neq r}^{n} q_{rij} x_r x_i x_j + 3 \sum_{i\neq r}^{n} q_{ri} x_r x_i + q_r x_r \tag{2.2}$$

$$\sum_{i}^{n} q_{ri} x_r x_i = \sum_{i\neq r}^{n} q_{ri} x_r x_i + q_r x_r \tag{2.3}$$

Remark. The findings of Proposition 2.2 are a result of counting the occurrences of the single terms.

Lemma 2.3. *The node conditional of the r-th variable in a multivariate binary model of interaction order $K = 3$ can also be represented as*

$$p_Q(x_r|x_{-r}) \propto \exp\Big(3 \sum_{i,j}^{n} q_{rij} x_r x_i x_j - 3 \sum_{i}^{n} q_{ri} x_r x_i + q_r x_r \Big). \tag{2.4}$$

Proof. We show this by using the terms from Proposition 2.2. We rearrange (2.2) as

$$\begin{aligned} & 2 \sum_{i<j, i\neq r, j\neq r}^{n} q_{rij} x_r x_i x_j = \sum_{i,j}^{n} q_{rij} x_r x_i x_j - 3 \sum_{i\neq r}^{n} q_{ri} x_r x_i - q_r x_r \\ \iff & \sum_{i<j, i\neq r, j\neq r}^{n} q_{rij} x_r x_i x_j = \frac{1}{2} \sum_{i,j}^{n} q_{rij} x_r x_i x_j - \frac{3}{2} \sum_{i\neq r}^{n} q_{ri} x_r x_i - \frac{1}{2} q_r x_r \end{aligned}$$

and (2.3) as

$$\sum_{i\neq r}^{n} q_{ri} x_r x_i = \sum_{i}^{n} q_{ri} x_r x_i - q_r x_r$$

and plug them into the previous formula (2.1) of the node conditional:

$$\begin{aligned} p_Q(x_r|x_{-r}) &\propto \exp\Big(6 \sum_{i<j, i\neq r, j\neq r}^{n} q_{rij} x_r x_i x_j + 6 \sum_{i\neq r}^{n} q_{ri} x_r x_i + q_r x_r \Big) \\ &= \exp\Big(3 \sum_{i,j}^{n} q_{rij} x_r x_i x_j - 3 \sum_{i\neq r}^{n} q_{ri} x_r x_i - 2 q_r x_r \Big) \\ &= \exp\Big(3 \sum_{i,j}^{n} q_{rij} x_r x_i x_j - 3 \sum_{i}^{n} q_{ri} x_r x_i + q_r rr \Big) \end{aligned}$$

□

To illustrate how the derived formula from the previous lemma is composed, we look at an example.

Example. Let $n = 4$ and $K = 3$. We would like to cut out the node conditional for the r-th variable for $r = 2$. We have the tensor Q and want to extract all parameters $q_{2ij}, i, j = 1, \ldots, 4$, that contain the index $r = 2$. It is illustrated in Figure 2.1 and Figure 2.2.

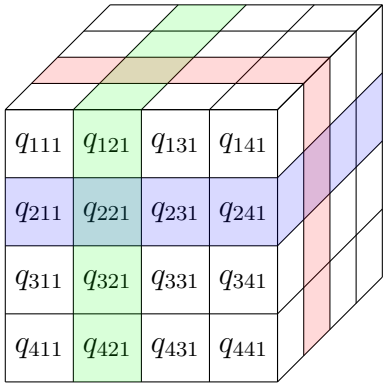

Figure 2.1: A 4-dimensional tensor of order $K = 3$ with marked node conditionals for $r = 2$.

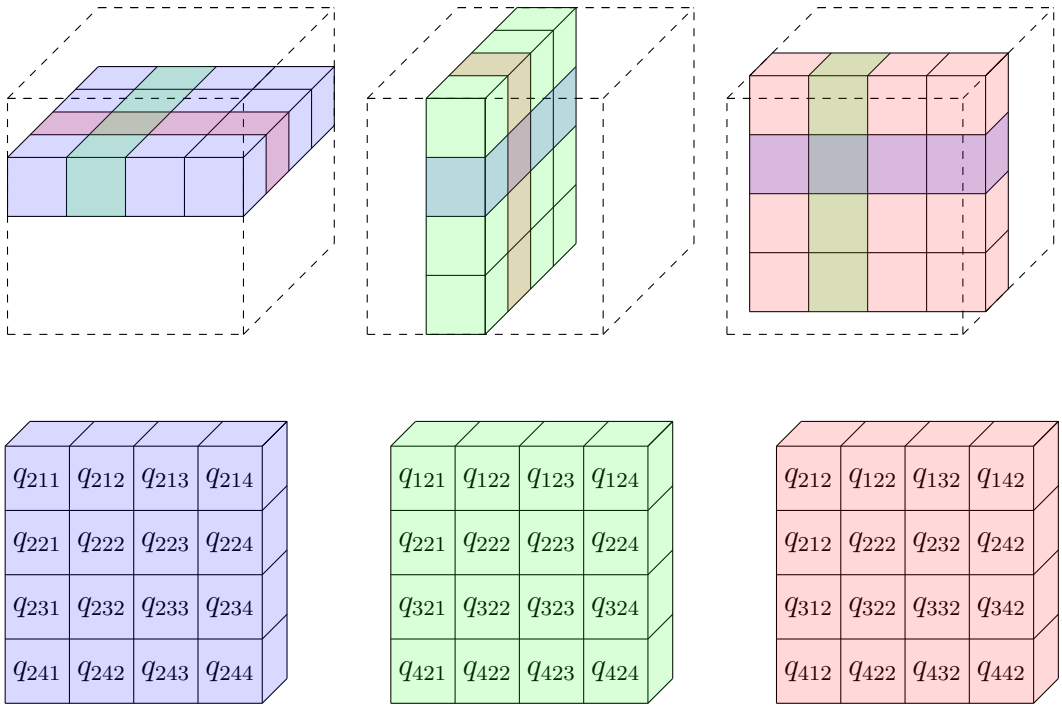

Figure 2.2: Slices that we cut out of the tensor Q that contain index $r = 2$.

After we cut out the slices that all include index $r = 2$ (first sum in (2.4)), we notice that we are left with too many copies of some parameters. As we can see, each of the rows $\{q_{22i}\}_{i=1,\dots,4}$, $\{q_{2i2}\}_{i=1,\dots,4}$ and $\{q_{i22}\}_{i=1,\dots,4}$ appear twice in Figure 2.2. Their sum gives us

$$2q_{22i} + 2q_{2i2} + 2q_{i22} = 2q_{2i} + 2q_{2i} + 2q_{2i} = 6q_{2i}, \quad i = 1, 2, 3, 4$$

even though we need each row only once. Therefore, we have to subtract them three times (second sum in (2.4)). That gives us

$$6q_{2i} - 3q_{2i} = q_{22i} + q_{2i2} + q_{i22} = 3q_{2i}, \quad i = 1, 2, 3, 4$$

which is the correct number of occurrences of q_{2i}. However, we have then subtracted the remaining summand q_{222} one too many times. As a result, we add it another time (third sum in (2.4)). This procedure reminds us of the inclusion-exclusion principle in combinatorics which generalizes the familiar method of obtaining the number of elements in the union of finite sets (compare [4], p. 70). It is clarified by Figure 2.3.

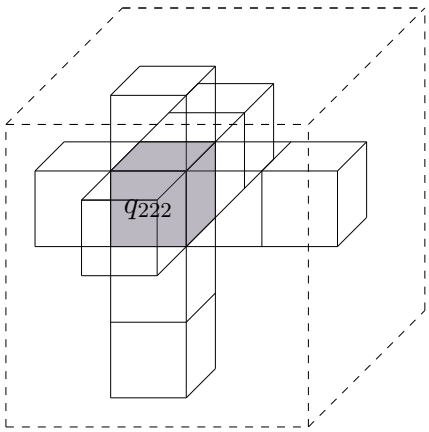

Figure 2.3: Illustration of the parameters that appear more than once when we cut out the slices for the node conditional for $r = 2$.

In the change of representation of the node conditional $p_Q(x_r|x_{-r})$ we went from picking out the single parameters from the tensor Q to using an alternating sum of tensors whose order is getting lower. This is obviously less expensive in terms of computation time since we have to do less jumps within the memory where everything lays linearly.

Now, we will formulate the log-pseudo-likelihood function which brings us closer to fit the model.

2.1.3 Pseudo-Likelihood Approach

Now that we formulated the node conditional for multivariate binary models of interaction order $K = 3$, we can do the next step and derive the pseudo-likelihood function to fit the parameter vector θ to data. First, we define various notations that will be used in the following.

Definition. Let $Q = \{q_{i_1 \dots i_K}\}_{i_1,\dots,i_K=1,\dots,n} \in \mathbb{R}^{n^K}$ be a strongly symmetric n-dimensional tensor of order K. Then, we define the *K-times tensor-vector multiplication* of Q and an arbitrary vector $x \in \mathbb{R}^n$ as

$$Q[x]^K = Q[\underbrace{x, x, \dots, x}_{K-times}] := \sum_{i_1,\dots,i_K}^{n} q_{i_1,\dots,i_K} x_{i_1} \dots x_{i_K}.$$

Definition. Let $Q = \{q_{i_1 \dots i_K}\}_{i_1,\dots,i_K=1,\dots,n} \in \mathbb{R}^{n^K}$ be a strongly symmetric n-dimensional tensor of order K. Then, we call

$$Q_{m,r}, \quad m \in \{1, \dots, K\}, r \in \{1, \dots, n\}$$

the *m-th subtensor* (slice) of Q of order $K - m$ and fixed index r. It is

$$Q_{K,r} = q_{\underbrace{r \ \dots \ r}_{K-times}} = q_r.$$

Example. Let $Q \in \mathbb{R}^{n \times n \times n}$ be a strongly symmetric n-dimensional tensor of order $K = 3$ and $x \in \mathbb{R}^n$. Then, it is

$$Q[x]^3 = Q[x, x, x] = \sum_{i,j,k}^{n} q_{ijk} \ x_i x_j x_k.$$

and

$$Q_{1,r}[x]^2 = Q_{1,r}[x, x] = \sum_{i,j}^{n} q_{rij} \ x_r x_i x_j = x^{\mathsf{T}} Q_{1,r} \ x$$

the first subtensor of Q of index r.

Remark. In Figure 2.2, the first subtensor for $r = 2$, $Q_{1,2}$ of the respective tensor Q is displayed and is equal to the slices we cut out of the tensor. Because of the strong symmetry of Q, all possible subtensors $Q_{m,r}$ for fixed m and r (for example, we have three possible subtensors in Figure 2.2) are equal.

Now, we can write the derived node conditional of Lemma 2.3 for a multivariate binary model of interaction order $K = 3$ as

$$\begin{aligned} p_Q(x_r | x_{-r}) &= \frac{\exp(\ 3 \sum_{i,j}^{n} q_{rij} x_r x_i x_j - 3 \sum_{i}^{n} q_{ri} x_r x_i + q_r x_r)}{1 + \exp(\ 3 \sum_{i,j}^{n} q_{rij} x_i x_j - 3 \sum_{i}^{n} q_{ri} x_i + q_r)} \\ &= \frac{\exp(\ 3 \ Q_{1,r}[x]^2 - 3 \ Q_{2,r}[x] + Q_{3,r} x_r)}{1 + \exp(\ 3 \ Q_{1,r}[x|_{x_r=1}]^2 - 3 \ Q_{2,r}[x|_{x_r=1}] + q_r)}. \end{aligned}$$

Again, we are given d data points $x^{(1)}, \ldots, x^{(d)}$ with each $x^{(i)} \in \{0,1\}, i = 1, \ldots, d$ and $\Omega = \{0,1\}^n$ that are independently drawn from a multivariate binary distribution p on Ω. We start with the maximum likelihood approach and end up in minimizing the negative log-pseudo-likelihood function ℓ_p. We formulate the optimization problem:

$$\begin{aligned}
\mathbf{p}_{ML} &= \arg\max_{Q} \; L(Q) \\
&\approx \arg\min_{Q} -\ell_p(Q) \\
&= \arg\min_{Q} -\sum_{i=1}^{d}\sum_{r=1}^{n} \log p_Q(x_r^{(i)} | x_{-r}^{(i)}) \\
&= \arg\min_{Q} -\sum_{i=1}^{d}\sum_{r=1}^{n} \log \frac{\exp(\; 3\; Q_{1,r}[x^{(i)}]^2 - 3\; Q_{2,r}[x^{(i)}] + Q_{3,r} x_r^{(i)})}{1 + \exp(\; 3\; Q_{1,r}[x^{(i)}|_{x_r=1}]^2 - 3\; Q_{2,r}[x^{(i)}|_{x_r=1}] + q_r)}
\end{aligned}$$

Here, we denote $x^{(i)}|_{x_r=1}$ in the normalization as the data point $x^{(i)}$ where the r-th entry is set to 1. In the first summand, technically it is $x^{(i)}|_{x_r=0}$, therefore the term becomes 0 and we have $\exp(0) = 1$.

The formulated optimization problem solves the parameter estimation problem for a multivariate binary model of interaction order $K = 3$. As we can see, the space of parameters grows, but stays below the complexity of an entire binary model. In the next part, we will generalize the approach for arbitrary interaction orders.

2.2 Multivariate Binary Models of Higher-Order Interactions

Now, we move on to generalize the approaches we discussed in the previous sections. We assume an arbitrary interaction order $1 < K \leq n$, define the corresponding multivariate binary model and derive the node conditional that mainly depends on K. To estimate the parameter within the tensor Q of order K, we will also formulate the optimization problem in the end.

2.2.1 Concept and Definitions

Before we introduce the class of multivariate binary models of higher interaction orders, we formally define the concept of the shortened parameter representation and strong symmetry.

Definition. We define the shortened parameter representation of a parameter of a multivariate binary model of interaction order K as

$$q_{i_1,i_2,\ldots,i_m} := q_{\underbrace{i_1,\ldots,i_1,i_2,\ldots,i_2,\ldots,i_m,\ldots,i_m}_{\#\ K}},$$

where each index appears only once. The absolute number of indices that are covered by the shortened parameter representation is required to be K.

Definition. Let Q be a tensor of order K. Then, we call Q *symmetric* when

$$q_{i_1,\ldots,i_K} = q_{\sigma(i_1),\ldots,\sigma(i_K)}$$

for all permutations σ of the set $\{i_1,\ldots,i_K\}$. That means all entries in Q with permuted indices are equal. Furthermore, we call Q *strongly symmetric* when

$$q_{i_1,\ldots,i_m} = q_{\sigma(i_1),\ldots,\sigma(i_m)}$$

for all $m \in \{1,\ldots,K\}$ for all permutations σ of the set $\{i_1,\ldots,i_m\}$. That means all entries in Q with permuted indices are equal for all shortened parameters.

Hence, normal symmetry and the shortened parameter representation induce strong symmetry of a tensor. Now, we can define the model class for the general case.

Definition. Let $X = (X_1,\ldots,X_n)$ be a vector of random variables with the set of outcomes $\Omega = \{0,1\}^n$. The distribution $p\colon \Omega \to [0,1]$ of a *multivariate binary model of interaction order K* has the form

$$p(x) = p_Q(x) \propto \exp\Big(\sum_{i_1,\ldots,i_K}^{n} q_{i_1\ldots i_K}\, x_{i_1}\ldots\, x_{i_K}\Big) = \exp\big(\, Q[x]^K \,\big)$$

with the strongly symmetric parameter tensor $Q = \{q_{i_1\ldots i_K}\}_{i_1,\ldots,i_K=1,\ldots,n} \in \mathbb{R}^{n^K}$ if the normalization condition

$$\sum_{x\in\Omega} p(x) = 1$$

is satisfied.

2.2.2 Representation of the Node Conditional

Before we arrive to the general node conditional, we define two different ways to represent them. It follows from the findings from the previous section where we already defined them for interaction order $K = 3$. Now, we would like to generalize this approach.

Definition. Let p_Q be the distribution of a multivariate binary model of interaction order K with strongly symmetric tensor Q. We call

$$p_Q(x_r|x_{-r}) \propto \exp\Big(\sum_{m=1}^{K} \tau_{K,m} \sum_{i_1<\ldots<i_m,\ \forall i_l : i_l \neq r}^{n} q_{r i_1 \ldots i_m}\ x_r x_{i_1} \ldots\ x_{i_m} \Big)$$

the *standard representation* for the r-th node conditional of the model p_Q with integer coefficients $\tau_{K,m}, m = 1, \ldots, K$.

Definition. Let p_Q be the distribution of a multivariate binary model of interaction order K with strongly symmetric tensor Q. We call

$$p_Q(x_r|x_{-r}) \propto \exp\Big(\sum_{m=1}^{K} \pi_{K,m} \sum_{i_1,\ldots,i_m}^{n} q_{r i_1 \ldots i_m}\ x_r x_{i_1} \ldots\ x_{i_m} \Big)$$

the *slice representation* for the r-th node conditional of the model p_Q with integer coefficients $\pi_{K,m}, m = 1, \ldots, K$.

The difference between these two types of representations and their coefficients $\tau_{K,m}$ and $\pi_{K,m}$ lies in the second sum. In contrast to the slice representation, we do not have the same index in a single summand twice in the standard representation.

The coefficients $\tau_{K,m}, \pi_{K,m}$ in the previous definitions are the result of counting occurrences of differently categorized parameters, which needs to be done for each K. In the following, we will systematically analyze the behavior of these coefficients and try to find a generalized expression to be able to calculate them for each K and m. The idea is to be able to receive quickly the formula of the node conditional, which will be important for the pseudo-likelihood function, for any given interaction order K. To derive this, we will follow a similar strategy as we did it in the previous section for interaction order $K = 3$.

2.2.2.1 Standard Representation

At first, we will analyze the coefficients $\tau_{K,m}$ of the standard representation. When we consider our observations from the previous section and take a closer look at the proof of Lemma 2.1, we recognize that the coefficients $\tau_{K,m}$ correspond to the m-partitions of K.

Definition. Let

$$P_{K,m} := \{(k_1, \ldots, k_m) \mid K = k_1 + \ldots + k_m,\ k_1 \geq k_2 \geq \ldots \geq k_m \geq 1,\ k_i \in \mathbb{N}\}$$

be the set of all m-partitions of K.

Definition. Let $(k_1, \ldots, k_m) \in P_{K,m}$ be an m-partition of K. Then we define the *counting vector* for a partition $(k_1, \ldots, k_m)$ as

$$a_{(k_1,\ldots,k_m)} := (a_1, \ldots, a_K),\ a_i \text{ is the number of occurrences of } i \text{ in } (k_1, \ldots, k_m),$$

that represents the occurrences of each integer in $(k_1, \ldots, k_m)$.

Proposition 2.4. *Let $a_{(k_1,\ldots,k_m)}$ be a counting vector of an m-partition of K. Then, it is*

$$\sum_{i=1}^{K} a_i = m.$$

Example. Let us consider the 3 partitions of 5 given by the set

$$P_{5,3} = \{(3,1,1),\ (2,2,1)\}$$

and its elements have the counting vectors

$$a_{(3,1,1)} = (2,0,1,0,0),\ a_{(2,2,1)} = (1,2,0,0,0).$$

Before we get to the lemma where we will count the number of permutations of indices for each m-partition of K, we will look at an example to understand how we will count and categorize them. The proof afterward follows the same principle.

Example. Let $K = 5$ and $m = 3$. We count all permutations of indices for the partitions in $P_{5,3} = \{(3,1,1),\ (2,2,1)\}$. Let us first look at $5 = 3 + 1 + 1$. We get 20 elements that are:

$$rrrij,\ rrijr,\ rijrr,\ ijrrr,\ rrirj,\ rirjr,\ irjrr,\ rirrj,\ irrjr,\ irrrj$$

$$rrrji,\ rrjir,\ rjirr,\ jirrr,\ rrjri,\ rjrir,\ jrirr,\ rjrri,\ jrrir,\ jrrri$$

This is the case because we first choose three r's from five, then we choose one i from two and then one j from one:

$$\tbinom{5}{3} \cdot \tbinom{2}{1} \cdot \tbinom{1}{1} = 10 \cdot 2 \cdot 1 = 20$$

This corresponds with the *multinomial coefficient*:

$$\tbinom{5}{3,1,1} = \tfrac{5!}{3!1!1!} = \tbinom{5}{3} \cdot \tbinom{2}{1} \cdot \tbinom{1}{1}$$

However, we also need to consider that each index is different. That means we also count the cases where i occurs three times and j three times. Those multiplicities can be expressed with the multinomial coefficient of the counting vector

$$a_{(3,1,1)} = (2,0,1,0,0),$$

that can be considered as a 2 partition of 3 with $3 = 2+1$. Its multinomial coefficient would be:

$$\tbinom{3}{2,1} = \tfrac{3!}{2!1!} = 3$$

Finally, we find the number of permutations of the indices for the partition $(3,1,1)$ as a product of the multinomial coefficient of the partition and the multinomial coefficient of its the counting vector:

$$\binom{5}{3,1,1} \cdot \binom{3}{2,1} = 20 \cdot 3 = 60$$

The same can be done with the other partition $(2,2,1) \in P_{5,3}$. Together with its counting vector $a_{2,2,1} = (1,2,0,0,0)$, we get

$$\binom{5}{2,2,1} \cdot \binom{3}{2,1} = \frac{5!}{2!2!1!} \cdot \frac{3!}{2!1!} = 30 \cdot 3 = 90$$

permutations of the indices. Overall, for the set $P_{5,3}$, we get

$$\binom{5}{3,1,1} \cdot \binom{3}{2,1} + \binom{5}{2,2,1} \cdot \binom{3}{2,1} = 60 + 90 = 150$$

permutations of indices.

Lemma 2.5. *Let $K \in \mathbb{N}$ be arbitrary and $m \in \{1, \ldots, K\}$. Let $P_{K,m}$ be the set of m-partitions of K and $\tau_{K,m}$ be the coefficients of the standard representation of the r-th node conditional. Then,*

$$\tau_{K,m} = \sum_{(k_1,\ldots,k_m) \in P_{K,m}} \binom{K}{k_1,\ldots,k_m} \binom{m}{a_1,\ldots,a_K},$$

where $a_{(k_1,\ldots,k_m)} = (a_1, \ldots, a_K)$ is the corresponding counting vector of $(k_1, \ldots, k_m)$.

Proof. Let $(k_1, \ldots, k_m)$ be a m-partition of K and $a_{(k_1,\ldots,k_m)} = (a_1, \ldots, a_K)$ its corresponding counting vector. To count the number of permutations of the respective indices for the partition

$$K = k_1 + \ldots + k_m$$

without regarding the order, we need to calculate:

$$\binom{K}{k_1} \cdot \binom{K-k_1}{k_1} \cdot \ldots \cdot \binom{K-k_1-\ldots-k_{m-1}}{k_m} = \frac{K!}{k_1! \ldots k_m!} = \binom{K}{k_1,\ldots,k_m}$$

However, we have different indices, therefore we have to regard the order of the partition as well. For instance, for $k_1 \neq k_2$, the permutations of indices of $(k_1, k_2 \ldots, k_m)$ and $(k_2, k_1, \ldots, k_m)$ must be counted individually. For counting the permutations, we use the counting vector that displays the multiplicities of the single k_i. Because it is $\sum_{i=1}^{K} a_i = m$, the non zero elements of the counting vector can be seen as a partition of m. Therefore, we get the multinomial coefficient to count the permutations in the initial permutation $(k_1, \ldots, k_m)$:

$$\binom{m}{a_1,\ldots,a_K}$$

Together we get the number of the permutations of the indices of a m-partition of K as a product of

$$\binom{K}{k_1,\ldots,k_m} \cdot \binom{m}{a_1,\ldots,a_K}.$$

This needs to be done for all m permutations of K. □

The proof shows how the correct number of permutations of indices can be counted and yields the remaining formula. When we compute the coefficients $\tau_{K,m}$, it generates a triangle that can be seen in Figure 2.4.

K							
1					1		
2				1		2	
3			1		6		6
4		1		14		36	24
5	1		30		150	240	120
6		62		540	1560	1800	720
7	126	1806	8400	16800	15120	5040	

Figure 2.4: Coefficients $\tau_{K,m}$ for $K = 1, \ldots, 7$, in the K-th row and m-th position.

Proposition 2.6. *The coefficients $\tau_{K,m}$ can also be calculated recursively with*

$$\tau_{K,m} = (\tau_{K-1,m-1} + \tau_{K-1,m}) \cdot m,$$

whereby it is $\tau_{K,K} = K!$ and $\tau_{K,1} = 1$.

Proposition 2.7. *For a fixed K (K-th row of the triangle in Figure 2.4), the coefficients $\tau_{K,m}$ can also be used to calculate potencies with any basis n as a sum of binomial coefficients:*

$$n^K = \sum_{i=1}^{K} \tau_{K,i} \cdot \binom{n}{i}$$

Remark. Within the triangle in Figure 2.4, additional structure can be identified. For example, the coefficients $\tau_{K,m}$ correspond with the number of onto functions from an n-element set to a k-element set (see [7], p. 144). Also, together with the results of Proposition 2.6 and Proposition 2.7, one gains direct insights to the problem of the divisibility of any potencies (compare [9]).

Example. Let $K = 4$ and $n = 6$. The parameters of a 6-dimensional tensor of order 4 are distributed as follows:

$$\begin{aligned} 6^4 &= 1 \cdot \binom{6}{1} + 14 \cdot \binom{6}{2} + 36 \cdot \binom{6}{3} + 24 \cdot \binom{6}{4} \\ &= 1 \cdot 6 + 14 \cdot 15 + 36 \cdot 20 + 24 \cdot 15 \\ &= 6 + 210 + 720 + 360 \\ &= 1296 \end{aligned}$$

Thereby we can recognize that there are 6 parameters with one index, 210 parameters with two different indices, 720 parameters with three different indices, 360 parameters with four different indices and all together 1296 indices.

2.2.2.2 Slice Representation

After we find the standard representation and calculated their coefficients $\tau_{K,m}$, we will get to the second type of representation and derive the coefficients $\pi_{K,m}$ for the slice representation. As it was pointed out before, the slice representation is mainly used for a highly efficient implementation, because the occurring sums are simply tensor-vector products. To attain the transition between the two representations, we use the tensor-vector product of the l-th subtensor of Q of order K for all $l \in \{1, \dots, K\}$ and the vector x regarding the r-th variable

$$Q_{l,r}[x]^{K-l} = \sum_{i_1,\dots,i_{K-l}}^{n} q_{ri_1\dots i_{K-l}} x_r x_{i_1} \dots x_{i_{K-l}}$$

as a term of sums from the standard representation. In the same way that we indicated the terms for $K = 2$ and $K = 3$ in Proposition 2.2, we do so for higher K's as well. Again, it is only a matter of counting the different types of parameters and assigning them to the correct group. Before we formulate the transition from both representations for arbitrary K's, we look at an example.

Example. Let $K = 3$, $n = 4$, $r = 1$ and Q be the n-dimensional tensor of order K of the respective model. Then, the tensor-vector product of the first subtensor ($l = 1$) regarding the r-th variable of Q can be expressed as

$$\begin{aligned}
Q_{1,1}[x]^2 &= \sum_{i,j}^{4} q_{1ij} x_1 x_i x_j \\
&= q_{111}x_1x_1x_1 + q_{112}x_1x_1x_2 + q_{113}x_1x_1x_3 + q_{114}x_1x_1x_4 + q_{121}x_1x_2x_1 \\
&\quad + q_{122}x_1x_2x_2 + q_{123}x_1x_2x_3 + q_{124}x_1x_2x_4 + q_{131}x_1x_3x_1 + q_{132}x_1x_3x_2 \\
&\quad + q_{133}x_1x_3x_3 + q_{134}x_1x_3x_4 + q_{141}x_1x_4x_1 + q_{142}x_1x_4x_2 + q_{143}x_1x_4x_3 \\
&\quad + q_{144}x_1x_4x_4 \\
&= q_1x_1 + 3q_{12}x_1x_2 + 3q_{13}x_1x_3 + 3q_{14}x_1x_4 + 2q_{123}x_1x_2x_3 \\
&\quad + 2q_{124}x_1x_2x_4 + 2q_{134}x_1x_3x_4 \\
&= q_1x_1 + 3\sum_{i\neq 1}^{4} q_{1i}x_1x_i + 2\sum_{i<j,\, i,j\neq 1}^{4} q_{1ij}x_1x_ix_j.
\end{aligned}$$

For the tensor-vector product of the second subtensor ($l = 2$) of Q, we get

$$\begin{aligned}
Q_{2,1}[x] &= \sum_{i}^{4} q_{11i}x_1x_1x_i \\
&= q_{111}x_1x_1x_1 + q_{112}x_1x_1x_2 + q_{113}x_1x_1x_3 + q_{114}x_1x_1x_4 \\
&= q_1x_1 + \sum_{i\neq 1}^{4} q_{1i}x_1x_i.
\end{aligned}$$

For the tensor-vector product of the third subtensor ($l = 3$) of Q, we get

$$Q_{3,1}x = q_1x_1.$$

Now, we generalize the approach in the following proposition.

Proposition 2.8. *Let $K \in \mathbb{N}$, $m \in \{1, \dots, K-l+1\}$, $l \in \{1, \dots, K\}$ and Q a tensor of order K. Then, the tensor-vector product of the l-th subtensor of Q can be expressed as*

$$Q_{l,r}[x]^{K-l} = \underbrace{\psi_{K-l+1,\,1}}_{=1}\, q_1 x_1 + \sum_{j=1}^{K-l} \psi_{K-l+1,\,j+1} \sum_{i_1<\dots<i_j,\,\forall i_l : i_l \neq r}^{n} q_{r i_1 \dots i_j} x_r x_{i_1} \dots x_{i_j}$$

with coefficients $\psi_{K-l+1,m} \in \mathbb{N}$ and it is $\psi_{K-l+1,1} = 1$ for all l.

We always assume a determined tensor Q with a fixed order K. From there, we get the coefficients $\psi_{K,m}$ for $m \in \{1, \dots, K\}$ for the tensor-vector product of the first subtensor of Q, the coefficients $\psi_{K-1,m}$ for $m \in \{1, \dots, K-1\}$ for the tensor-vector product of the second subtensor of Q. We continue until we get the coefficient $\psi_{1,1}$ for the tensor-vector product of the K-th subtensor of Q.

Remark. In the example above, we already found that $\psi_{3,1} = 1$, $\psi_{3,2} = 3$, $\psi_{3,3} = 2$ in the first equation, $\psi_{2,1} = 1$, $\psi_{2,2} = 1$ in the second equation, and $\psi_{1,1} = 1$ in the last equation. These are the first three rows in the triangle in Figure 2.5.

Remark. For the next part, let K remain variable, set $l = 1$ and consider the first subtensor. The coefficients $\psi_{K,m}$ apparently are the same, no matter if you vary the order K or the subtensor order $K - l$ through l. This means, for example, that the coefficients $\psi_{3,1}, \psi_{3,2}, \psi_{3,3}$ are needed to represent the first subtensor of a tensor of order 3 , but also to represent the third subtensor of a tensor of order 5.

Proposition 2.9. *Let $K \in \mathbb{N}$, $m \in \{1, \dots, K\}$ and Q a tensor of order K. Let $P_{K,m}$ be the set of m-partitions of K and $\psi_{K,m}$ the coefficients for the representation of the first subtensor of Q. Then,*

$$\psi_{K,m} = \frac{1}{m} \sum_{(k_1,\dots,k_m) \in P_{K,m}} \binom{K}{k_1,\dots,k_m} \binom{m}{a_1,\dots,a_K} = \frac{\tau_{K,m}}{m},$$

where the $\tau_{K,m}$'s stand for the coefficients of the standard representation of the r-th node conditional and $a_{(k_1,\dots,k_m)} = (a_1, \dots, a_K)$ is the corresponding counting vector of $(k_1, \dots, k_m) \in P_{K,m}$.

Remark. Proposition 2.9 mainly claims that the triangle from Figure 2.4 can easily be transformed into the triangle in Figure 2.5. Its proof is similar to the proof of Lemma 2.5. However, because we consider subtensors, we have to divide through m another time in the end to get the correct coefficients.

Proposition 2.10. *The coefficients $\psi_{K,m}$ can also be calculated recursively with*

$$\psi_{K,m} = (\ (m-1)\ \psi_{K,m-1} + m\ \psi_{K,m}\ ),$$

whereby it is $\psi_K = K!$ and $\psi_{K,1} = 1$.

K							
1				1			
2			1		1		
3		1		3		2	
4	1		7		12		6
5		15		50		60		24
6	1		32		180		390		360		120
7		63		602		2100		3360		2520		720

Figure 2.5: Coefficients $\psi_{K,m}$ for $K = 1, \ldots, 7$ in the K-th row and m-th position.

Proposition 2.11. *For a fixed K (K-th row of the triangle in Figure 2.5), the coefficients $\psi_{K,m}$ can also be used to calculate potencies with any basis n as a sum of binomial coefficients:*

$$n^{K-1} = \sum_{i=1}^{K} \psi_{K,i} \cdot \binom{n}{i-1}$$

Remark. The results from Proposition 2.10 and Proposition 2.11 follow directly from the corresponding Proposition 2.6 and Proposition 2.7 about the coefficients $\tau_{K,m}$ in the previous section.

After calculating the coefficients $\psi_{K,m}$, we are going to examine the coefficients $\pi_{K,m}$. As it was pointed out before, we are going from the standard representation of the node conditional to the slice representation. The triangle in Figure 2.5 will support us. First, we will look at an example.

Example. Let $K = 4$. We get the r-th node conditional $p(x_r|x_{-r})$ of the standard representation from the triangle in Figure 2.4 as follows:

$$\begin{aligned}
&\exp\Big(24 \sum_{i<j<k,\, i,j,k\neq r}^{n} q_{rijk} x_r x_i x_j x_k + 36 \sum_{i<j,\, i,j\neq r}^{n} q_{rij} x_r x_i x_j + 14 \sum_{i\neq r}^{n} q_{ri} x_r x_i + q_r x_r \Big) \\
= \; &\exp\Big(4 \sum_{i,j,k}^{n} q_{rijk} x_r x_i x_j x_k - 12 \sum_{i<j,\, i,j\neq r}^{n} q_{rij} x_r x_i x_j - 14 \sum_{i\neq r}^{n} q_{ri} x_r x_i - 3\, q_r x_r \Big) \\
= \; &\exp\Big(4 \sum_{i,j,k}^{n} q_{rijk} x_r x_i x_j x_k - 6 \sum_{i,j}^{n} q_{rij} x_r x_i x_j + 4 \sum_{i\neq r}^{n} q_{ri} x_r x_i + 3\, q_r x_r \Big) \\
= \; &\exp\Big(4 \sum_{i,j,k}^{n} q_{rijk} x_r x_i x_j x_k - 6 \sum_{i,j}^{n} q_{rij} x_r x_i x_j + 4 \sum_{i}^{n} q_{ri} x_r x_i - q_r x_r \Big)
\end{aligned}$$

We can derive the coefficients $\pi_{K,m}$ from the coefficients $\tau_{K,m}$ with the help of the coefficients $\psi_{K,m}$ in the triangle in Figure 2.5. This is done the same way as in

Lemma 2.3, where we derived it for $K = 3$.

We get the slice representation of the r-th node conditional for $K = 4$:

$$p_Q(x_r|x_{-r}) \propto \exp\Big(4\sum_{i,j,k}^{n} q_{rijk}x_r x_i x_j x_k - 6\sum_{i,j}^{n} q_{rij}x_r x_i x_j + 4\sum_{i}^{n} q_{ri}x_r x_i - q_r x_r \Big)$$
$$= \exp\Big(4\, Q_{1,r}[x,x,x] - 6\, Q_{2,r}[x,x] + 4\, Q_{3,r}[x] - Q_{4,r}x_r \Big)$$

After the previous example, we unexpectedly find a significant structure within the coefficients $\pi_{K,m}$, which we formulate in the next proposition.

Proposition 2.12. *Let $K \in \mathbb{N}$ be arbitrary and $m \in \{1, \dots, m\}$. Let $\pi_{K,m}$ be the coefficients of the slice representation of the r-th node conditional of a multivariate binary model of interaction order K, then*

$$\pi_{K,m} = (-1)^{K-m} \binom{K}{m-1}.$$

Remark. Most likely, the result of Proposition 2.12 can be proven by mathematical induction. However, it becomes quite complicated and technical in the field of combinatorics. Also in relation to our problem, we pointed out that a rather small number of interaction orders is sufficient for us to reduce the number of parameters. Therefore, the result of Proposition 2.12 is verified here only by numerical calculations.

Remark. Surprisingly, a well-known structure can be identified. The absolute values of the coefficients $\pi_{K,m}$ within the triangle in Figure 2.6. correspond to the values of the pascal's triangle.

K							
1				1			
2			-1		2		
3		1		-3		3	
4	-1		4		-6		4
5		1		-5		10		-10		5
6	-1		6		-15		20		-15		6
7		1		-7		21		-35		35		-21		7

Figure 2.6: Coefficients $\pi_{K,m}$ for $K = 1, \dots, 7$, in the K-th row and m-th position.

2.2.3 Pseudo-Likelihood Approach

Now that we have derived the coefficients $\pi_{K,m}$ for the slice representation, we can formulate the node conditional with normalization for a generalized multivariate binary model of interaction order K:

$$p_Q(x_r|x_{-r}) = \frac{\exp\Big(\sum\limits_{m=1}^{K} \pi_{K,m}\, Q_{m,r}[x]^{K-m}\Big)}{1+\exp\Big(\sum\limits_{m=1}^{K} \pi_{K,m}\, Q_{m,r}[x|_{x_r=1}]^{K-m}\Big)}$$

As in the chapters before, we can now use the node condition and formulate the log-pseudo-likelihood function for multivariate binary models of higher orders. We are given d data points $x^{(1)}, \ldots, x^{(d)}$ with each $x^{(i)} \in \{0,1\}, i = 1, \ldots, d$ and $\Omega = \{0,1\}^n$ that are independently drawn from a multivariate binary distribution p on Ω. We start with the maximum likelihood approach and end up in minimizing the negative log-pseudo-likelihood function ℓ_p. We formulate the optimization problem:

$$\begin{aligned}
\mathbf{p}_{ML} &= \arg\max_{Q}\ L(Q) \\
&\approx \arg\min_{Q} -\ell_p(Q) \\
&= \arg\min_{Q} -\sum_{i=1}^{d}\sum_{r=1}^{n} \log p_Q(x_r^{(i)}|x_{-r}^{(i)}) \\
&= \arg\min_{Q} -\sum_{i=1}^{d}\sum_{r=1}^{n} \log\left(\frac{\exp\Big(\sum\limits_{m=1}^{K} \pi_{K,m}\, Q_{m,r}[x^{(i)}]^{K-m}\Big)}{1+\exp\Big(\sum\limits_{m=1}^{K} \pi_{K,m}\, Q_{m,r}[x^{(i)}|_{x_r=1}]^{K-m}\Big)}\right)
\end{aligned}$$

Since we have derived the objective function -ℓ_p, we can go to the next chapter and work out the regularized optimization problem. Before that, we will introduce various regularization techniques to obtain certain characteristics as sparse or low-rank on the parameters.

Chapter 3

Model Selection for Interaction Models

In the following chapter, we will first discuss various regularization techniques for tensors that will be used while fitting the model. Then, we will concentrate on the estimation of the optimal parameters, in other words, the formulation of the optimization problem for model selection.

3.1 Regularization Techniques for Tensors

Primarily, regularization is useful to avoid overfitting, i.e., when the parameters correspond too closely to a particular sample of a data set. The additional penalty term causes improved stability within the fitting process and generates a model that is closer to the real distribution. Additionally, the parameters' behavior can be pushed towards some desired characteristics like sparsity, where single parameters converge to zero. We look at an example that illustrates how the concept of regularization works in general.

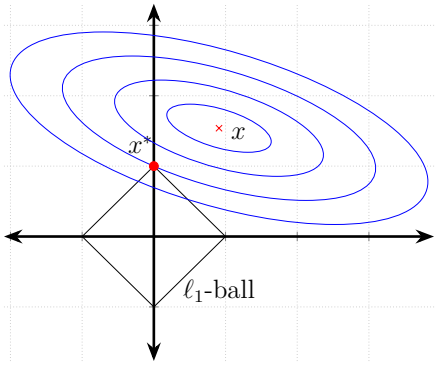

Figure 3.1: Illustration of a ℓ_1-regularized optimization problem. The blue contour plot denotes the objective function with solution x and the square denotes the domain (ℓ_1-ball).

In Figure 3.1, we can recognize how regularization affects an optimization problem. In this case, a solution x^* can only be found within the ℓ_1-ball and here, more likely at its vertices. This causes sparsity on some of the variables. It differs from the actual solution x of the unregularized problem.

Now, we want to apply similar techniques for multivariate binary models of interactions. However, it is not trivial to transfer all existing norms for matrices to tensors of higher orders. Besides sparse and low-rank regularization, we will also examine their associated norms and try to give a short overview over possible techniques for tensors.

3.1.1 Sparse Regularization

As it is also indicated in the previous example, sparsity is induced by ℓ_1-regularization (see [5]). That is because the absolute values of all entries of a solution are compelled to be low, since their sum must fulfill a certain threshold. This leads to a homogeneous shrinkage effect on the entries and thus, sparsity on the solution is induced. We also would like to apply the ℓ_1-norm for tenors of higher orders. Fortunately, this works in the same way as for matrices.

Definition. For a n-dimensional tensor Q of order K, the ℓ_1-norm is defined as

$$|||Q|||_1 = \sum_{i_1,\dots,i_K}^{n} |q_{i_1\dots i_K}|,$$

where $q_{i_1\dots i_K}$, $i_j = 1,\dots,n$, $j = 1,\dots,K$ are the entries of Q.

Besides ℓ_1-regularization, there exist more norms and techniques that induce sparsity. For example, the ℓ_0-regularization, where all non-zero entries are counted, can be used as well. However, it has been demonstrated that solving a ℓ_0-regularized problem is NP-hard, because of its non-convexity (see [5]). Therefore, we will use ℓ_1-regularization which corresponds to a convex relaxation of the ℓ_0-norm.

3.1.2 Low-Rank Regularization

Low-rank regularization aims to reduce the rank of its argument. A tensor with low-rank has more linearly dependent parameters, which leads to a reduction of the number of distinct parameters, because their encoded information is redundant. Before we can apply low-rank regularization, we need to define the rank of a tensor, which requires a different representation. Therefore, we first are going to define how a tensor can be expressed by a rectangular matrix.

Definition. Let Q be a n-dimensional tensor of order K and $m \in \{1,\dots,K\}$. The mode-m unfolding $Q_{(m)} \in \mathbb{R}^{n\times n^{K-1}}$ is a matrix that is obtained by concatenating the mode-m fibers along columns. A mode-m fiber is a n^{K-1}-dimensional vector obtained by fixing all the indices but the m-th index of Q.

Now, using the defined mode-m unfoldings, we can define a concept for the rank of a tensor.

Definition. Let Q be a n-dimensional tensor of order K. We define the *tucker rank* as a tuple

$$(\ \mathrm{rank}(Q_{(1)}),\ \dots\ ,\mathrm{rank}(Q_{(K)})\ ),$$

where $Q_{(m)}$ is the mode-m unfolding of the tensor Q.

The rank of a matrix $M \in \mathbb{R}^{n\times m}$ is defined as the dimension of the vector space that is spanned by the column vectors of M, which corresponds to the maximal number of linearly independent columns of M. It is equal to the maximal number of linearly independent rows of M. The rank of a matrix is lowered by lowering its nuclear norm, because it pelanties large singular values which shrinks the rank. Since the tucker rank is a K-tuple of ranks of matrices, we would like to lower the nuclear norm of its mode-m unfoldings to ultimately lower the tensor's tucker rank. Therefore, we define the Schatten p-norms and get the nuclear norm for matrices (compare the following defintions with [8], p. 3).

Definition. Let $M \in \mathbb{R}^{n\times m}$ a matrix. The Schatten p-norm of a matrix is defined as

$$\|M\|_{S_p} = \Big(\sum_{i=1}^{\min\{n,m\}} \sigma_i(M)^p\Big)^{\frac{1}{p}},$$

where σ_i are the singular values of M. Then, we get the nuclear norm for $p = 1$ as

$$\|M\|_{S_1} = \sum_{i=1}^{\min\{n,m\}} \sigma_i(M).$$

Now, we can define norms on tensors, respectively its unfoldings that affect its tucker rank. First, we look at a convex surrogate for the tucker rank which is the sum of nuclear norms.

Definition. Let Q be a n-dimensional tensor of order K with mode-m unfoldings $Q_{(m)}$, then it is

$$|||Q|||_{\underline{S_1/1}} = \sum_{m=1}^{K} \big\|Q_{(m)}\big\|_{S_1}$$

called the *overlapped Schatten* 1*-norm*.

Minimizing the overlapped Schatten 1-norm means minimizing the sum of nuclear norms of the tensors' unfoldings. Through minimizing the sum of nuclear norms, the nuclear norm of each m-unfolding is lowered, a process by which we cause a rank reduction in each mode. Therefore, when the overlapped Schatten 1-norm is used for regularization, it penalizes all modes to be jointly low-rank.

Furthermore, we look at another norm that comes with less restrictions since it does not require all modes to be low-rank.

Definition. Let Q be a n-dimensional tensor of order K with mode-m unfoldings $Q_{(m)}$, then it is

$$|||Q|||_{\overline{S_1/1}} = \inf_{Q=(Q^{(1)}+\dots+Q^{(K)})} \sum_{m=1}^{K} \Big\|Q^{(m)}_{(m)}\Big\|_{S_1}$$

the *latent Schatten* 1*-norm*.

We see, the latent Schatten 1-norm approach for regularization is based on a decomposition of tensors. Through that, each of the components are regularized to be low-rank in a specific mode. Since one does not have to ensure that all modes are simultaneously low-rank as it is the case for the overlapped Schatten 1-norm, this approach can be seen as a less constrained regularization technique.

However, in our case, we are fortunately dealing with strongly symmetric tensors. In the following lemma, we show how this is affects the structure of its unfoldings.

Lemma 3.1. *Let Q be a strongly symmetric n-dimensional tensor of order K, then its mode-m unfoldings are equal, it is*

$$Q_{(1)} = Q_{(2)} = \ldots = Q_{(K)}.$$

Proof. Since Q is strongly symmetric, all entries of its permutations of shortened indices are equal. It is $Q_{(m)} \in \mathbb{R}^{n \times n^{K-1}}$. Let $Q_{(i)}$ and $Q_{(j)}$ be unfoldings for arbitrary $i, j \in \{1, \ldots, K\}$. Let $l \in \{1, \ldots, n\}$. Then, we get the l-th row of the unfolding $Q_{(i)}$ as

$$Q_{(i)l} = (q_{1,1,\ldots,l,\ldots,1},\ q_{2,1,\ldots,l,\ldots,1}, \ldots\ , q_{n,n,\ldots,l,\ldots,n\text{-}1}\ , q_{n,n,\ldots,l,\ldots,n}),$$

where the index l is at the i-th position. Now, for the unfolding $Q_{(j)}$, we get the l-th row as,

$$Q_{(j)l} = (q_{1,1,\ldots,l,\ldots,1},\ q_{2,1,\ldots,l,\ldots,1}, \ldots\ , q_{n,n,\ldots,l,\ldots,n\text{-}1}\ , q_{n,n,\ldots,l,\ldots,n}),$$

where the index l is at the j-th position. Since the tensor is strongly symmetric, all permuted indices are equal. Therefore it is $Q_{(i)l} = Q_{(j)l}$ for all $l \in \{1, \ldots, n\}$. □

From the previous lemma, it follows that the overlapped Schatten 1-norm of a strongly symmetric tensor Q can be calculated as a K times scaling of the nuclear norm of any mode-m unfolding $Q_{(m)}$. Additionally, we also conclude in the next lemma that the latent Schatten 1-norm that lowers the rank of a specific mode and the overlapped Schatten 1-norm are the same up to a scaling with K.

Lemma 3.2. *Let Q be a strongly symmetric n-dimensional tensor of order K. Then, it is*

$$K \cdot |||Q|||_{\overline{S_1/1}} = |||Q|||_{\underline{S_1/1}}.$$

Proof. We denote any mode-m unfolding of a strongly symmetric n-dimensional tensor Q as $Q_{(*)}$. Then, it is $Q_{(*)} = Q_{(1)} = \ldots = Q_{(K)}$. Furthermore, we implicitly assume that the decomposition of Q in the infimum of the latent Schatten 1-norm (on strongly symmetric tensors) only consists of strongly symmetric tensors $Q^{(m)}$. So, it holds that $Q^{(m)}_{(1)} = \ldots = Q^{(m)}_{(K)}$ for all $m \in \{1, \ldots K\}$. Then, we have

$$\begin{aligned}
|||Q|||_{\overline{S_1/1}} &= \inf_{Q=(Q^{(1)}+\ldots+Q^{(K)})} \sum_{m=1}^{K} \left\| Q^{(m)}_{(m)} \right\|_{S_1} \\
&= \inf_{Q=(Q^{(1)}+\ldots+Q^{(K)})} \sum_{m=1}^{K} \left\| Q^{(m)}_{(*)} \right\|_{S_1} \\
&\geq \inf_{Q=(Q^{(1)}+\ldots+Q^{(K)})} \left\| \sum_{m=1}^{K} Q^{(m)}_{(*)} \right\|_{S_1} \\
&= \left\| Q_{(*)} \right\|_{S_1},
\end{aligned}$$

where the inequality is simply triangle inequality. It follows

$$K \cdot |||Q|||_{\overline{S_1/1}} \geq K \cdot \left\|Q_{(*)}\right\|_{S_1} = |||Q|||_{\underline{S_1/1}}.$$

However, we always find a decomposition such that $Q^{(1)} = Q$, $Q^{(m)} = 0$ for all $m \in \{2, ..., K\}$. Then, we have

$$\sum_{m=1}^{K} \left\|Q_{(*)}^{(m)}\right\|_{S_1} = \left\|Q_{(*)}\right\|_{S_1}$$

and since the latent Schatten 1-norm is defined as the infimum over the decompositions of Q, we have

$$\inf_{Q=(Q^{(1)}+...+Q^{(K)})} \sum_{m=1}^{K} \left\|Q_{(*)}^{(m)}\right\|_{S_1} \leq \left\|Q_{(*)}\right\|_{S_1}.$$

It follows

$$K \cdot |||Q|||_{\overline{S_1/1}} \leq K \cdot \left\|Q_{(*)}\right\|_{S_1} = |||Q|||_{\underline{S_1/1}}.$$

□

3.2 Regularized Optimization Problem

Now that we have introduced the regularization techniques that will be used, we can formulate the regularized optimization problem to fit parameters to a multivariate binary model of interaction order K. Since our goal is to reduce complexity of the model, we think of the high dimensional parameter space as a composition of a lower dimensional, in other words, lower ranked component and a sparse component. The reason lies in the assumption or hope that the real parameters of the distribution are actually located in a lower dimensional subspace, which is is either modeled by a low-rank component through linearly dependent columns, respectively rows or by a sparse component that assumes the irrelevance of certain dimensions. Therefore, we would like to find a solution which is separated in a sparse and low-rank component. Fortunately, we can optimize the negative log-likelihood function with both components. So, the regularized optimization problem has the form

$$\min_{S,L} \text{-}\ell_p(S + L) \tag{3.1}$$

$$\text{s. t. } |||S|||_1 \leq a, \quad |||L|||_{\underline{S_1/1}} \leq b$$

with $S, L \in \mathbb{R}^{n^K}$ strongly symmetric and $a, b > 0$.

It holds that the sum of two strongly symmetric tensors is still strongly symmetric. Technically, the strong symmetry of the optimization variables is encoded as constraints as well. We have to ensure that the optimization variables stay strongly symmetric throughout the whole optimization process. Different algorithms expect

different forms of representations of the optimization problem. Therefore, we additionally formulate the Lagrangian form as

$$\min_{S,L} \text{-}\ell_p(S+L) + \lambda \|\!|S|\!\|_1 + \mu \|\!|L|\!\|_{\underline{S_1/1}} \tag{3.2}$$

with $S, L \in \mathbb{R}^{n^K}$ strongly symmetric and $\lambda, \mu > 0$.

If a solution is known, the optimization problems (3.1) and (3.2) can be transformed into each other through Lagrange duality. More accurately, the constraint parameters a, b of (3.1) are related with the regularization parameters λ, μ of (3.2). So in the end, both formulations can be used to solve the problem, but which formulation is more suitable depends on its requirements and preconditions.

Through substitution, we are also able to normalize the constraint parameters of (3.1) to 1, which can be helpful in some cases too. An example is shown in next chapter. So, let $S' = \frac{1}{a}\, S$ and $L' = \frac{1}{b}\, L$, then we have:

$$\begin{aligned}
& X^* = \min_{S,L} \text{-}\ell_p(S+L), \;\; \text{s. t.} \;\; \|\!|S|\!\|_1 \le a, \;\; \|\!|L|\!\|_{\underline{S_1/1}} \le b \\
\iff & X^* = \min_{S',L'} \text{-}\ell_p(a\, S' + b\, L'), \;\; \text{s. t.} \;\; \|\!|aS'|\!\|_1 \le a, \;\; \|\!|bL'|\!\|_{\underline{S_1/1}} \le b \\
\iff & X^* = \min_{S',L'} \text{-}\ell_p(a\, S' + b\, L'), \;\; \text{s. t.} \;\; a\, \|\!|S'|\!\|_1 \le a, \;\; b\, \|\!|L'|\!\|_{\underline{S_1/1}} \le b \\
\iff & X^* = \min_{S',L'} \text{-}\ell_p(a\, S' + b\, L'), \;\; \text{s. t.} \;\; \|\!|S'|\!\|_1 \le 1, \;\; \|\!|L'|\!\|_{\underline{S_1/1}} \le 1
\end{aligned}$$

Now, after formulating the regularized optimization problem, we can go further and consider optimization algorithms to solve the parameter estimation problem.

Chapter 4

Frank-Wolfe Algorithm for Sparse and Low-Rank Optimization

In this chapter, we will examine an algorithm for convex optimization problems which was published as early as 1956 by Frank, M. and Wolfe, P. and revisited by Jaggi, M. in 2013 (see [2]).

4.1 General Approach

To solve the parameter estimation problem and find solutions for the optimization problems formulated in the previous chapter, we apply the *Frank-Wolfe Algorithm*. At first, we explain the algorithm in general and later, we will adjust it to our problem with different regularization techniques.

The algorithm can be applied for constrained convex optimization problems of the form

$$\min_{X \in \mathcal{D}} \ f(X).$$

The objective function f must be convex and continuously differentiable. In our case, the objective function is the negative log-likelihood function, so we denote $f = \text{-}\ell_p$. The domain $\mathcal{D}$ must be a convex, compact subset of the vector space we are operating on. Then, the basic algorithm of Frank-Wolfe has the following form (compare [2], p. 1):

Algorithm 1 Basic Frank-Wolfe Algorithm

1: Let $X^{(0)} \in \mathcal{D}$
2: **for each** $i = 0, \ldots, N$ **do**
3: $\quad \zeta := \frac{2}{i+2}$
4: $\quad S := \arg\min_{S \in \mathcal{D}} \langle S, \nabla f(X^{(i)}) \rangle$
5: $\quad X^{(i+1)} := (1 - \zeta) X^{(i)} + \zeta S$
return $X^{(N)}$

The direction of the optimization step is computed in Line 4, where the algorithm assumes a linearization of f through its gradient ∇f at the current point X. Then,

the minimizer within the domain $\mathcal{D}$ of that linear function is the direction towards a minimizer of f. The updated point is calculated in Line 5. In the basic form, in Line 3, we reduce the length of the step in each iteration steadily, which ensures a converging algorithm, but also other techniques for optimizing the step size like *line−search* can be applied. The algorithm with step length as in Line 3 guarantees a convergence of $f(x^{(i)}) - f(X^*) < O(\frac{1}{i})$, if X^* is a solution (compare [2], p. 1).

Furthermore, the algorithm requires a convex and compact domain $\mathcal{D}$. The space of all strongly symmetric tensors is not compact, but through regularization that we have introduced in the previous chapter, the domain $\mathcal{D}$ is reduced and becomes compact. That means, adding sparsity or low-rank norm constraints induces an optimization over compact subsets, because feasible points have to be within the corresponding ℓ_p-balls.

4.2 Sparse Optimization

To obtain sparsity, we have to solve the subproblem

$$\inf_{S \in \mathcal{D}} \langle S, \nabla f(X^{(k)}) \rangle \tag{4.1}$$

over the set of strongly symmetric matrices, which are within the ℓ_1-ball. The set is defined as

$$\mathcal{D} := \{X \in \mathbb{R}^{n^K} \mid |||X|||_1 \leq 1 \, , X \text{ strongly symmetric}\}.$$

According to [2], p. 5, here it is enough to optimize over the vertices of the set ℓ_1-ball. We discovered that the algorithm considers a linearization of f at the current point $X^{(k)}$, the gradient $\nabla f(X^{(k)})$. The minimization of the inner product of (4.1) describes a plane that goes through the domain $\mathcal{D}$, which is a polytope. Because the objective function is a linear function, the minimum is taken on at a vertex S of $\mathcal{D}$ and describes the direction of the optimization step. So, in each step the update S pushes the solution towards a vertex of $\mathcal{D}$, which induces sparsity. That means we have to find the vertices of the domain in order to solve the linear subproblem (4.1). So, in fact, the following subproblem has to be solved:

$$\inf_{S \in \mathcal{V}(\mathcal{D})} \langle S, \nabla f(X^{(k)}) \rangle$$

Remark. We call $\mathcal{V}(\mathcal{D})$ the set of vertices of a set $\mathcal{D}$.

Next, we would like to find the vertices of the ℓ_1-ball in the vector space of strongly symmetric tensors. The ℓ_1-ball for n-dimensional tensors of order K has its vertices at the positive and negative unit vectors. Since we are only considering strongly symmetric tensors, the polytope is intersected with the set of strongly symmetric tensors, which is a subset. How the set is truncated is illustrated in the following example.

Example. Let $\mathcal{M} = \{X \in \mathbb{R}^{2\times 2} \mid \|X\|_1 \leq 1\,, X = X^\intercal\}$ be the set of the symmetric 2×2 matrices. We get the set of vertices as

$$V(\mathcal{M}) = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -0.5 \\ -0.5 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \right\}.$$

The set $\mathcal{M}$ can also be interpreted as the set $\mathcal{M}' = \{x \in \mathbb{R}^4 \mid \|x\|_1 \leq 1,\ x_2 = x_3\}$. Then, in the following figure, we can see how the 2-dimensional projection onto the x_2 and x_3 coordinates of the corresponding set $\{x \in \mathbb{R}^4 \mid \|x\|_1 \leq 1\}$ reduces to a truncated 1-dimensional projection of $\mathcal{M}'$, since we set $x_2 = x_3$. The location of the vertices change.

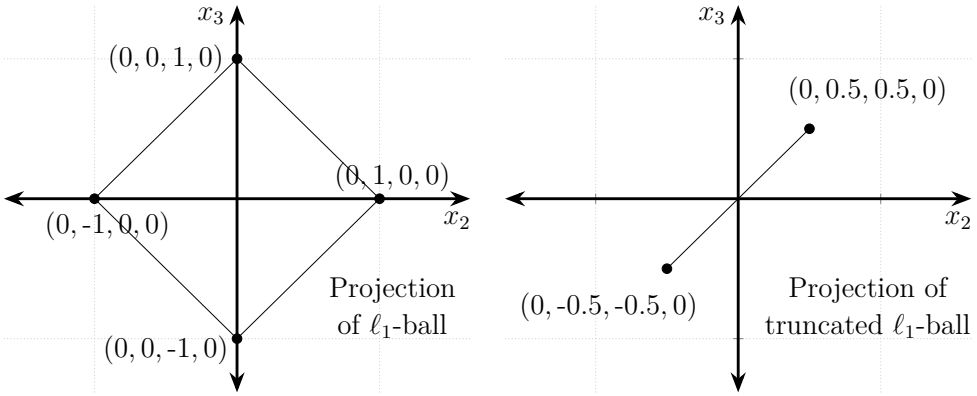

Figure 4.1: On the left, a 2-dimensional projection onto the x_2 and x_3 coordinates of a 4-dimensional polytope is illustrated. After adding symmetry constraints by setting $x_2 = x_3$, it reduces to a 1-dimensional projection on the right.

We see that when we add such constraints, the space gets truncated and the set of vertices of the corresponding ℓ_1-ball change. To define the vertices of $\mathcal{D}$, we need to define the set of shortened indices which directly comes from the shortened parameter representation (see Section 2.2.1).

Definition. We define

$$I_{(K,n)} = \{(i_1, \ldots, i_m) \in \{1, \ldots, n\}^m \mid i_1 < \ldots < i_m,\ \forall m \in \{1, \ldots, K\}\}$$

as the set of the indices from the shortened parameters of any strongly symmetric n-dimensional tensor of order K. An index $(i_1, \ldots, i_K)$ is represented by the shortened index $(i_1, \ldots, i_m)$, when their corresponding parameters $q_{i_1,\ldots,i_K}, q_{i_1,\ldots,i_m}$ are equal by strong symmetry. Then, we write

$$(i_1, \ldots, i_m) \equiv (i_1, \ldots, i_K).$$

Furthermore, we annotate $\#(i_1, \ldots, i_m)$ as the number of indices that are represented by the shortened index $(i_1, \ldots, i_m)$.

Lemma 4.1. *We get the set of vertices of $\mathcal{D}$ as*

$$\mathcal{V}(\mathcal{D}) = \text{-}\mathcal{V}'(\mathcal{D}) \cup \mathcal{V}'(\mathcal{D}),$$

where

$$\mathcal{V}'(\mathcal{D}) = \left\{ \{v_{i_1,\dots,i_K}\}_{i_1,\dots,i_K=1,\dots,n} \in \mathcal{D} \mid j \in I_{(K,n)},\ v_{i_1,\dots,i_K} = \begin{cases} \frac{1}{\#j}, & j \equiv (i_1,\dots,i_K) \\ 0 & ,else \end{cases} \right\}.$$

Proof. Let $V' \in \mathcal{V}(\mathcal{D})$ arbitrary and let $|\mathcal{V}(\mathcal{D})| = l$. Then, it is $V' = \{v'_{i_1,\dots,i_K}\}_{i_1,\dots,i_K=1,\dots,n}$ with

$$v'_{i_1,\dots,i_K} = \begin{cases} \frac{1}{\#j}, & j \equiv (i_1,\dots,i_K) \\ 0 & ,else \end{cases}, \text{ for a } j \in I_{(K,n)}.$$

Then, V' and $\text{-}V'$ are the only tensors in $\mathcal{V}(\mathcal{D})$ that have entries different from 0 at the indices that are represented by the shortened index j. Consequently, there is no $\lambda \in [0,1]^{l-2}$ such that $V' = \sum_{i=1}^{l-2} \lambda_i V^i$ for remaining vertices $V^i \in \mathcal{V}(\mathcal{D}) \setminus \{V', \text{-}V'\}$. Also, there is no $\lambda \in [0,1]$ such that $V' = \text{-}\lambda V'$. So, V' can not be represented as a convex combination of the other vertices.

Let $X \in \mathcal{D}$ arbitrary. By construction, for each $j \in I_{(K,n)}$ there exist exactly two vertices V and $\text{-}V$ with entries different from 0 at the indices represented by the shortened index j. By definition, for each entry x_i within X there exists exactly one $j \in I_{(K,n)}$ such that $j \equiv i$. Consequently, for all entries x_i within X, there exists exactly one $\mu \in [0,1]$ such that $x_i = \mu v_j$ or $x_i = \text{-}\mu v_j$ for entries v_j of a vertex $V \in \mathcal{V}(\mathcal{D})$. Then, $\exists \lambda \in [0,1]^l$ such that $X = \sum_i^l \lambda_i V^i$ for vertices $V^i \in \mathcal{V}(\mathcal{D})$. □

Now that we have discovered where the vertices of $\mathcal{D}$ are located, we can formulate the Frank-Wolfe algorithm for ℓ_1-norm regularization.

Algorithm 2 Frank-Wolfe Algorithm for ℓ_1-norm regularization

1: Let $X^{(0)} \in \mathcal{D} := \{X \in \mathbb{R}^{n^K} \mid |||X|||_1 \leq 1\ , X \text{ strongly symmetric}\}$
2: **for each** $i = 0, \dots, N$ **do**
3: $\quad \zeta := \frac{2}{i+2}$
4: $\quad S := \arg\min_{S \in \mathcal{V}(\mathcal{D})} \langle S, \nabla f(X^{(i)}) \rangle$
5: $\quad X^{(i+1)} := (1-\zeta)X^{(i)} + \zeta S$
return $X^{(N)}$

The sparse variant of Frank-Wolfe keeps its general shape, but in Line 4, we only have to optimize over all vertices instead of the entire domain. That gives us an algorithm which is easy to implement and only requires additional knowledge about the vertices. To apply a different regularization parameter, in other words to optimize over a greater or smaller domain, we only have to scale the domain and its vertices to the desired regularization value.

4.3 Low-Rank Optimization

Next, we would like to apply low-rank regularization to model selection with Frank-Wolfe. The idea behind low-rank regularization is based on the assumption that low-rank tensors have linearly dependent entries and therefore, we get a reduction of parameters. As we have demonstrated in the previous chapter, we are using the overlapped Schatten 1-norm to enforce low-rank regularization. Unfortunately, unlike in the section before, we have to differentiate between the models of order $K = 2$ and higher-order models, because their overlapped Schatten 1-norms change from order $K = 3$ onward, and it is more complicated to assert the strongly symmetry condition throughout the entire optimization process. First, we are going to examine the case of order $K = 2$.

4.3.1 Tensors of Order $K = 2$

Since we are limited to lower order models, the optimization variables are simply symmetric matrices. Note that for $K = 2$ symmetry and strong symmetry are equivalent. Therefore, we only use the basic Schatten 1-norm (nuclear norm) on matrices and we have to solve the subproblem

$$\inf_{L \in \mathcal{D}} \langle L, \nabla f(X^{(k)}) \rangle = \sup_{L \in \mathcal{D}} \langle L, -\nabla f(X^{(k)}) \rangle. \tag{4.2}$$

over the domain

$$\mathcal{D} := \{X \in \mathbb{R}^{n \times n} \mid \|X\|_{S_1} \leq 1,\ X = X^{\mathsf{T}}\}.$$

Fortunately, the case is covered by Frank-Wolfe. According to [2], p. 7, a solution L^* is given through the eigenvalue decomposition of $-\nabla f(X) = U \mathrm{diag}(\sigma) U^{\mathsf{T}}$ by

$$L^* = U \mathrm{diag}(r) U^{\mathsf{T}}$$

for any r that fulfills the following conditions:

$$\bullet\ \|r\|_1 \leq 1 \tag{4.3}$$

$$\bullet\ r^{\mathsf{T}} \sigma = \|\sigma\|_\infty \tag{4.4}$$

We verify the solution from [2]. Since the problem (4.2) is equal to the definition of the dual norm, we get

$$\sup_{L \in \mathcal{D}} \langle L, -\nabla f(X) \rangle = \|-\nabla f(X)\|_{S_\infty} = \sigma_{\max}(-\nabla f(X)).$$

In this case, $\|\cdot\|_{S_\infty}$ denotes the Schatten ∞-norm. Then, since $-\nabla f(X)$ is strongly symmetric, there exists an orthogonal decomposition $-\nabla f(X) = U D U^{\mathsf{T}}$ with unitary matrix U and a diagonal matrix $D = \mathrm{diag}(\sigma)$. Since all Schatten norms are unitarily invariant, it is

$$\begin{aligned}
\|-\nabla f(X)\|_{S_\infty} &= \left\|U^{\mathsf{T}} - \nabla f(X) U\right\|_{S_\infty} \\
&= \|D\|_{S_\infty} \\
&= \sup_{A:\ \|A\|_* \leq 1} \langle A, D \rangle \\
&= \sup_{r:\ \|r\|_1 \leq 1} \langle \mathrm{diag}(r), \mathrm{diag}(\sigma) \rangle
\end{aligned}$$

which is maximized by any $r \in \mathbb{R}^n$ s.t. $\|r\|_1 \leq 1$ and $r^\intercal \sigma = \sigma_{\max}(-\nabla f(X)) = \|\sigma\|_\infty$. Furthermore, for any orthogonal matrices U and V, it holds that $\langle A, B\rangle = \langle UAV^\intercal, UBV^\intercal\rangle$ for some matrices $A, B \in \mathcal{D}$. Therefore, $L^* = U\mathrm{diag}(r)U^\intercal$ is a solution of (4.2).

To find a vector r that fulfills the conditions (4.3) and (4.4), we simply set the i-th entry to ± 1 and the remaining entries to zero, whereby i is the position of a maximum absolute value of the vector σ. Note that this is an ambiguous definition, since a vector can have more entries whose absolute values are equal to its maximum norm.

Definition. We define

$$J(\sigma) = \{(u_1, \ldots, u_n) \in \mathbb{R}^n \mid \exists!\, i\colon u_i = \begin{cases} 1 & , \sigma_i = \|\sigma\|_\infty \\ -1 & , \sigma_i = -\|\sigma\|_\infty \end{cases}, \ \forall j \neq i\colon \ u_j = 0\}$$

as a set of vectors which are filled with zeros except at the position where the argument σ has a maximum absolute value.

Note that the set $J(\sigma)$ always has at least one element. Then, the Frank-Wolfe algorithm for Schatten 1-norm regularization for tensors of order $K = 2$ can be formulated as follows:

Algorithm 3 Frank-Wolfe Algorithm for Schatten 1-norm regularization for tensors of order $K = 2$

1: Let $X^{(0)} \in \{X \in \mathbb{R}^{n\times n} \mid \|X\|_{S_1} \leq 1,\ X = X^\intercal\}$.
2: **for each** $i = 0, \ldots, N$ **do**
3: $\quad \zeta := \frac{2}{i+2}$
4: $\quad -\nabla f(X^{(i)}) = U\mathrm{diag}(\sigma)U^\intercal$
5: $\quad L := \arg\min_{L\in\mathcal{D}}\langle L, \nabla f(X^{(i)})\rangle = U\mathrm{diag}(r)U^\intercal, \ \text{with} \ \ r \in J(\sigma)$
6: $\quad X^{(i+1)} := (1-\zeta)X^{(i)} + \zeta L$
return $X^{(N)}$

The bottleneck of the algorithm is surely the orthogonal decomposition we have to perform in Line 4 in each iteration. However, we received an algorithm where we applied the Schatten 1-norm regularization to our problem for tensors of order $K = 2$. In the following section, we are going to examine the problem for tensors of higher orders.

4.3.2 Tensors of Higher Orders

In this section, we will attempt to apply the Frank-Wolfe algorithm with low-rank regularization for problems of higher orders. This time, since we are operating on n-dimensional tensors of order K, we are going to optimize over the set

$$\mathcal{D} := \{X \in \mathbb{R}^{n^K} \mid |||X|||_{\underline{S_1/1}} \leq 1,\ X \text{ strongly symmetric}\}.$$

As before, we have to solve the subproblem

$$\sup_{L\in\mathcal{D}} \langle L, \text{-}\nabla f(X)\rangle. \tag{4.5}$$

in each iteration step. By definition, the overlapped Schatten 1-norm is calculated on the unfolded tensors. For that, we use one of the mode-m unfolding $\text{-}\nabla f(X)_{(m)}$ which is equal to all other possible unfoldings, since $\text{-}\nabla f(X)$ is strongly symmetric. But this means we actually solve the problem

$$\sup_{L\in\mathcal{D}'} \langle L, \text{-}\nabla f(X)_{(m)}\rangle. \tag{4.6}$$

for the set $\mathcal{D}'$ of unfolded tensors of $\mathcal{D}$. The domain $\mathcal{D}'$ has two restrictions. On the one hand we have to consider the strong symmetry constraints, and on the other hand the nuclear norm of L has to be kept below 1. This is done by its singular values. Consequently, we use singular value decomposition and get $\text{-}\nabla f(X)_{(m)} = U\mathrm{diag}(\sigma)V^{\mathtt{T}}$ with the singular values σ_i of $\text{-}\nabla f(X)_{(m)}$. Then, we have to solve

$$\sup_{r:\, U\mathrm{diag}(r)V^{\mathtt{T}}\in\mathcal{D}'} \langle U\mathrm{diag}(r)V^{\mathtt{T}}, U\mathrm{diag}(\sigma)V^{\mathtt{T}}\rangle. \tag{4.7}$$

for feasible vectors $r \in \mathbb{R}^n$ such that:

- $\|r\|_1 \leq 1$
- $U\mathrm{diag}(r)V^{\mathtt{T}} \in \mathcal{D}'$

In the previous section we did not have to worry about the symmetry for tensors of order $K = 2$, because the decompositions were done on quadratic matrices, where symmetry and strong symmetry are equivalent. Therefore, it kept its strongly symmetric structure for any vector r. Now, however, even though the strongly symmetric structure of $\text{-}\nabla f(x)$ gets forwarded to its unfolding $\text{-}\nabla f(x)_{(m)}$, the result is not necessarily strongly symmetric for any vector r. The unfoldings are rectangular matrices, therefore SVD also returns a rectangular matrix. More precisely, we have

$$U \in \mathbb{R}^{n\times n},\ \mathrm{diag}(r) \in \mathbb{R}^{n\times n},\ V \in \mathbb{R}^{n^{K-1}\times n}.$$

Let us try to find the feasible set of vectors for (4.7). This is given by

$$\begin{aligned}
L_{(m)}(r) &= U\mathrm{diag}(r)V^{\mathtt{T}} \\
&= \begin{pmatrix} u_{11} & \dots & u_{1n} \\ \vdots & & \vdots \\ u_{n1} & \dots & u_{nn} \end{pmatrix} \begin{pmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{1n^{K-1}} \\ \vdots & & \vdots \\ v_{n1} & \dots & v_{nn^{K-1}} \end{pmatrix} \\
&= \begin{pmatrix} r_1u_{11} & \dots & r_nu_{1n} \\ \vdots & & \vdots \\ r_1u_{n1} & \dots & r_nu_{nn} \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{1n^{K-1}} \\ \vdots & & \vdots \\ v_{n1} & \dots & v_{nn^{K-1}} \end{pmatrix} \\
&= \begin{pmatrix} \sum_{i=1}^n r_iu_{1i}v_{i1} & \dots & \sum_{i=1}^n r_iu_{1i}v_{in^{K-1}} \\ \vdots & & \vdots \\ \sum_{i=1}^n r_iu_{ni}v_{i1} & \dots & \sum_{i=1}^n r_iu_{ni}v_{in^{K-1}} \end{pmatrix}
\end{aligned}$$

Since L has to be strongly symmetric, a vector r is only feasible if the corresponding entries of $L_{(m)}$ that are responsible for the strong symmetry of the folded tensor L are equal. $L_{(m)}(r)$ returns a linear system of equations for the vector r that has to be solved to get a feasible set for (4.7). Unfortunately, the linear system of equations already gets quite complicated even for small examples, since all of equalities to obtain strong symmetry have to be considered. Also, there are few relations between the entries of the orthogonal matrices U and V which make it tough to eliminate equations from the system beforehand. Because of the reasons, it is hard to find a general analytical solution. Also, due to numerical inaccuracies, especially when it comes to calculation of the singular value decomposition, it is neither easy nor reliable to compute a solution numerically.

On the other hand, we actually already have a feasible vector for the problem (4.7), the vector σ from the singular value decomposition before. When we use a scaled vector $r = \lambda\sigma$, such that $\|r\|_1 \leq 1$, the tensor $U\mathrm{diag}(r)V^{\mathtt{T}}$ keeps its strong symmetry, since the corresponding above-mentioned linear system of equations is homogeneous. Then, we get a set of feasible vectors. Unfortunately, this only represents a scaling of the negative gradient direction, which means that we do not necessarily get the low-rank characteristics for a the feasible vector.

In the end, we conclude that it is rather complex to find an analytical or numerical solution to the problem (4.6), which makes it difficult to apply overlapped Schatten 1-norm regularization to the Frank-Wolfe algorithm for tensors of higher orders.

4.4 Sparse and Low-Rank Optimization

Since we could not find any analytical solution for the overlapped Schatten 1-norm regularization for the Frank-Wolfe algorithm for higher-order tensors, we can only apply the sparse and low-rank regularization for tensors of order $K = 2$, in other words only for the parameter estimation of multivariate binary models of pairwise interactions. We would like to solve the optimization problem (3.1) we have formulated in the previous chapter, but restricted on symmetric matrices. For $a = b = 1$, we have

$$\begin{aligned} & X^* = \underset{S,L}{\arg\min}\ f(S+L), \ \text{ s. t. } \ \|S\|_1 \leq 1, \ \ \|L\|_{S_1} \leq 1 \\ \iff & X^* = \underset{S,L}{\arg\min}\ f(S+L), \ \text{ s. t. } \ S \in \mathcal{D}_S, \ \ L \in \mathcal{D}_L \\ \iff & X^* = \underset{A}{\arg\min}\ f(A), \ \text{ s. t. } \ A \in \mathcal{D} \end{aligned}$$

for the cartesian product $\mathcal{D} := \mathcal{D}_S \times \mathcal{D}_L = \{(S, L) \mid S \in \mathcal{D}_S, L \in \mathcal{D}_L\}$ and

$$\begin{aligned} \mathcal{D}_S &:= \{X \in \mathbb{R}^{n\times n} \mid \|X\|_1 \leq 1\,, X = X^{\mathtt{T}}\} \\ \mathcal{D}_L &:= \{X \in \mathbb{R}^{n\times n} \mid \|X\|_{S_1} \leq 1,\ X = X^{\mathtt{T}}\}. \end{aligned}$$

Also, f is defined on the cartesian product $\mathcal{D}_S \times \mathcal{D}_L$ by $f(A) = f((S, L)) = f(S+L)$. Then, for Frank-Wolfe, we have to solve the subproblem

$$X^* = \inf_{A \in \mathcal{D}} \langle A, \nabla f(X^{(k)}) \rangle$$

which is equivalent to

$$X^* = \inf_{S \in \mathcal{D}_S, L \in \mathcal{D}_L} \langle S + L, \nabla f(X^{(k)}) \rangle = \inf_{S \in \mathcal{D}_S} \langle S, \nabla f(X^{(k)}) \rangle + \inf_{L \in \mathcal{D}_L} \langle L, \nabla f(X^{(k)}) \rangle.$$

The remaining optimization problem that has to be solved is the sum of two problems that can be solved independently from each other, since the single summands do not share the same optimization variables. That means we can minimize the Sparse and Low-Rank component within the Frank-Wolfe algorithm with the methods we have used in the previous sections. Furthermore, since the update steps for each component can also be done independently from each other, the optimizing variable is the sum of the single components. We see that $X^{(k)} = X_S^{(k)} + X_L^{(k)}$, which is also the position where the gradient has to be calculated. So, we have to solve the subproblems

$$\inf_{S \in \mathcal{D}_\mathcal{S}} \langle S, \nabla f(X_S^{(k)} + X_L^{(k)}) \rangle, \quad \inf_{L \in \mathcal{D}_\mathcal{L}} \langle L, \nabla f(X_S^{(k)} + X_L^{(k)}) \rangle$$

in each iteration exactly as in the previous sections on Sparse and Low-Rank regularization. The entire Frank-Wolfe algorithm for ℓ_1-norm regularization and overlapped Schatten 1-norm regularization can be formulated as a combination of the previous algorithms:

Algorithm 4 Frank-Wolfe Algorithm for ℓ_1-norm regularization and overlapped Schatten 1-norm regularization for tensors of order $K = 2$

1: Let $X_S^{(0)} \in \{X \in \mathbb{R}^{n \times n} \mid \|X\|_1 \leq 1,\ X = X^\intercal\}$.
2: Let $X_L^{(0)} \in \{X \in \mathbb{R}^{n \times n} \mid \|X\|_{S_1} \leq 1,\ X = X^\intercal\}$.
3: **for each** $i = 0, \ldots, N$ **do**
4: $\quad \zeta := \frac{2}{k+2}$
5: $\quad S := \arg\min_{S \in V(\mathcal{D}_\mathcal{S})} \langle S, \nabla f(X_S^{(i)} + X_L^{(i)}) \rangle$
6: $\quad X_S^{(i+1)} := (1 - \zeta) X_S^{(i)} + \zeta S$
7: $\quad -\nabla f(X_S^{(i)} + X_L^{(i)}) = U \mathrm{diag}(\sigma) U^\intercal$
8: $\quad L := \arg\min_{L \in \mathcal{D}} \langle L, \nabla f(X_S^{(i)} + X_L^{(i)}) \rangle = U \mathrm{diag}(r) U^\intercal, \text{ with } r \in J(\sigma)$
9: $\quad X_L^{(i+1)} := (1 - \zeta) X_L^{(i)} + \zeta L$
return $X_S^{(N)} + X_L^{(N)}$

Once we can formulate the Frank-Wolfe algorithm for our problem for tensors of order $K = 2$, we will get into some possible improvements for the algorithm.

4.5 Further Improvements

As it is also described in [2], there are several further improvements available that enhance the basic Frank-Wolfe algorithm. For example, instead of lowering the update step length in each iteration evenly, we can also perform a line-search on ζ to find the optimal step size. Note that the step size has to be in the interval $[0, 1]$, because the updated point has to be a convex combination of the current point and the direction. Otherwise the update might be outside of the domain. That means we have to solve the problem

$$\zeta^* := \underset{\zeta \in [0,1]}{\arg\min}\, f(X_S^{(k)} + X_L^{(k)} - \zeta(X_S^{(k)} + X_L^{(k)} - S - L)),$$

where S and L annotate the optimal step directions for ℓ_1-norm regularization and overlapped Schatten 1-norm regularization. We get the update as

$$X^{(k+1)} := X_S^{(k)} + X_L^{(k)} - \zeta^*(X_S^{(k)} + X_L^{(k)} - S - L).$$

Also, we can use a second step variable to calculate the single step sizes for each regularization independently. Then, we have to solve

$$(\zeta_1^*, \zeta_2^*) := \underset{\zeta_1, \zeta_2 \in [0,1]}{\arg\min}\, f(X_S^{(k)} - \zeta_1(X_S^{(k)} - S) + X_L^{(k)} - \zeta_2(X_L^{(k)} - L)).$$

Here, we get the updates for $X_S^{(k+1)}$ and $X_L^{(k+1)}$ for ζ_1^* and ζ_2^* as described before in algorithm 4. It can be said that the line-search causes a higher complexity because we have to solve an additional optimization problem, but we will end up in the optimal solution in fewer steps. In practice, a naive approach for applying a line−search for the step size would be an equidistant separation $\{a_1, ..., a_n\}$ of the interval $[0, 1]$. Then, the function values can be compared for each $\zeta \in \{a_1, ..., a_n\}$ and the optimal ζ is chosen as the a_i for which f is minimal. Another approach would be using the convexity of f. One can use a separation of $[0, 1]$, but instead of comparing all function values, one stops if the function value becomes better for an a_i. Then one continues deeper into the domain and separates the interval $[a_{i-1}, a_i]$ again. That can be done until some depth threshold is reached.

In the end, it can be said that the Frank-Wolfe algorithm is an effective approach to meet the parameter estimation problem for multivariate models of higher-order interactions. We found an effective derivation for both regularization techniques for tensors of order $K = 2$ and formulated the corresponding algorithms. In the case of higher-order tensors, it still works for ℓ_1-norm regularization, but unfortunately, it is rather difficult to apply the overlapped Schatten 1-norm regularization, because the strong symmetry constraints are hard to process.

Further information about the Frank-Wolfe algorithm, as well as used for other regularization techniques, can be found in [2]. In the next section, we will focus on certain implementation details which are important when we want to fit a multivariate model of higher-order interactions to data.

4.6 Alternative Solvers

Aside from the just presented Frank-Wolfe Algorithm, we would like to suggest other options for solving the regularized parameter estimation problem and give insights to various implementation details. During the work, our tests on multivariate binary models of higher-order interactions were generally based and implemented using *Python 3.7*. More specifically, we mostly used the software distribution *PyTorch*, a Python package that offers wide support for abstract computations on tensors. It includes automatically generated gradients, mathematical operations as singular value decomposition or tensor transpositions, and its own support for solving optimization problems.

Here, we carried out a considerable proportion of our tests using solvers of PyTorch such as the SGD algorithm (stochastic gradient descent) and the Adam algorithm (method for stochastic optimization). Both use stochastic processes for their optimization and work well for first attempts to minimize the objective function. These algorithms are widely used and described in [10] and [3]. We mainly compare our results with the solution of CVX, a Matlab-based modeling system for convex optimization (compare [1]). Also, we implement the adjusted Frank-Wolfe algorithm within the PyTorch package, since it is always able to automatically track the necessary gradient of the objective function. Here, we successfully implement the above-mentioned algorithms for pairwise interactions, and in the case of the just ℓ_1-norm regularized problem, it also works for interaction order $K = 3$. The general outcome is a generative software package that expects the number of variables n and the interaction order $K = 2, 3$, ready to fit the corresponding model parameters to input data.

If one is willing to do further research on this topic, the code can be requested and reused (e-mail: christophsaffer@gmail.com).

Chapter 5

Conclusion

At the end of this work, we have found an interesting approach which allows us to describe complex relationships between binary variables with far fewer parameters through multivariate models of higher-order interactions than the entire categorical model would contain. Here, we have studied the behavior of the model in the case of binary variables which is an evident simplification, as opposed to variables with more than two outcomes. Through regularization, it is even possible to eliminate more parameters and simplify the model further.

The main point of the research on the theoretical side is illuminated in the second chapter, where we concentrate on deriving the node conditional for interaction order $K = 3$ and the generalized approach for higher-order interactions. This is important for formulating the maximum pseudo-likelihood estimation in later chapters to find optimal parameters of the model to a set of data. Regarding the node conditional which is essential for the fitting process, we discover deep structures within the model parameters and specifically two different types of representing the node conditional are found, the standard representation and slice representation. In particular, in terms of implementing the pseudo-likelihood function, it is important to find the slice representation, where the sliced tensor is calculated by an alternating sum of tensor-vector products. In the end, the unregularized optimization problem is given through the negative log-pseudo-likelihood function. The here presented theory can be extended to multivariate models of higher-order interactions of variables with more than two outcomes. The parameters within the tensors would extend to model group interactions. In general, this might be more useful considering real-world scenarios, since there are clearly more problem cases where no variable is restricted to be binary.

Furthermore, we require not only to adapt the parameters of the model to the data, but also to apply regularization to the optimization problem. Therefore, we introduce techniques to enforce sparse and low-rank regularization on our problem through the ℓ_1-norm and the overlapped Schatten 1-norm on tensors. Also, we give further theoretical foundations, in particular about low-rank regularization. Then, we derive the regularized optimization problem for multivariate binary models of higher-order interactions.

In the fourth chapter, we introduce then Frank-Wolfe algorithm for convex optimization problems and adjust it to our problem. Here, the regularization was

encoded through the domain, defined as the ℓ_1-norm, respectively overlapped Schatten 1-norm ball on the space of strongly symmetric tensors. For that, we derive the algorithm for ℓ_1-norm and overlapped Schatten 1-norm regularization on multivariate binary models of pairwise interactions. The sparse case also can be derived for higher interaction orders, but for the low-rank case, deeper problems regarding the strong symmetry of the tensor and ensuring it throughout the optimization process emerge. Here, further research is necessary to find an analytical solution or to be able to control the numerical inaccuracies that occur.

Finally, it can be said that the topic has significant potential and can be extended in many possible directions. On the one hand, the theory on interaction models for general categorical variables can be continued using the approach we suggest in this work. On the other hand, the studies on the Frank-Wolfe algorithm are not finished yet, but express an interesting approach, especially when it comes to regularization on various norm balls. Here, further experiments are necessary for a study to significantly compare the accuracy and performance of the different optimization algorithms such as the optimization algorithms of PyTorch, the Frank-Wolfe algorithm and the CVX package of Matlab. Also, experiments on favorable regularization parameters can be considered as an extension of the here presented work.

Overall, this work presents a promising approach for modeling binary data through multivariate models of higher-order interactions that contain fewer parameters than entire categorical models, and where the number of parameters is additionally optimized through regularization.

Bibliography

[1] Grant, M. ; Boyd, S. : CVX: Matlab Software for Disciplined Convex Programming. (2014), March

[2] Jaggi, M. : Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In: *Proceedings of Machine Learning Research* (2013), S. 427–435

[3] Kingma, D. ; Ba, J. : Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations* (2014), 12

[4] Lint, J. H. ; Wilson, R. M.: *A Course in Combinatorics*. 2. Edition. Cambridge University Press, 2001

[5] Natarajan, B. K.: Sparse Approximate Solutions to Linear Systems. In: *SIAM Journal on Computing* (1995), S. 227–234

[6] Nussbaum, F. ; Giesen, J. : Ising Models with Latent Conditional Gaussian Variables. In: *Proceedings of the 30th International Conference on Algorithmic Learning Theory (ALT)* (2019), S. 669–681

[7] Rosen, K. H.: *Handbook Of Discrete And Combinatorial Mathematics*. 1. Edition. CRC Press, 1999

[8] Tomioka, R. ; Suzuki, T. : Convex Tensor Decomposition via Structured Schatten Norm Regularization. In: *Advances in Neural Information Processing Systems* (2013), S. 1331–1339

[9] Wikipedia: *Pascalsches Dreieck*. `https://de.wikipedia.org/wiki/Pascalsches_Dreieck#Potenzen_mit_beliebiger_Basis`. Version: September 2019

[10] Zhang, T. : Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms. In: *Proceedings of the Twenty-first International Conference on Machine Learning* (2004), S. 116

Danksagung

Am Ende möchte ich noch einige Personen erwähnen, die maßgeblich zum Erfolg dieser Arbeit und meines gesamten Studiums beigetragen haben.

Zuerst möchte ich mich ganz besonders bei Frank Nussbaum, dem Betreuer dieser Masterarbeit bedanken, der mich jederzeit mit hilfreichen Ratschlägen und Hinweisen unterstützen konnte. Zusätzlich gab er mir immer wieder neue Impulse und Denkanstöße und hat somit sicherlich zu einem großen Teil zur Entstehung dieser Arbeit beigetragen.

Auch möchte ich mich natürlich ganz besonders bei Professor Joachim Giesen für die Unterstützung bedanken, nicht nur während dieser Abschlussarbeit, sondern auch für die letzten Jahre, während ich am Institut für Informatik als wissenschaftliche Hilfskraft arbeiten durfte. Durch den Austausch während dieser Zeit habe ich viel lernen können, nicht nur in fachlichen Bereichen, sondern auch darüber hinaus.

Zusätzlich möchte ich nun, zum Ende meines Studiums, natürlich auch meinen Eltern danken, die mir während all der Jahre den nötigen Rückhalt gaben, mir bei allem vertraut und mich bei jeglichen Entscheidungen unterstützt haben.

Im Schlusssatz möchte ich nochmal alle meine Freunde und Weggefährten erwähnen, die die bisherige Zeit in Jena zu etwas ganz Besonderem gemacht haben.

Selbstständigkeitserklärung

Ich, Christoph Saffer, erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Seitens des Verfassers bestehen keine Einwände die vorliegende Masterarbeit für die öffentliche Benutzung im Universitätsarchiv zur Verfügung zu stellen.

Ort, Datum	Unterschrift