REGULATORY NETWORKS STUDIED BY ELLIPSOIDAL CALCULUS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SELİM YAYLA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

SEPTEMBER 2015

Approval of the thesis:

REGULATORY NETWORKS STUDIED BY ELLIPSOIDAL CALCULUS

submitted by **SELİM YAYLA** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _______________

Assoc. Prof. Dr. Ömür Uğur
Head of Department, **Scientific Computing** _______________

Prof. Dr. Gerhard-Wilhelm Weber
Supervisor, **Scientific Computing, METU** _______________

Assoc. Prof. Dr. Vilda Purutçuoğlu
Co-supervisor, **Department of Statistics, METU** _______________

Examining Committee Members:

Prof. Dr. Bülent Karasözen
Scientific Computing, METU _______________

Prof. Dr. Gerhard-Wilhelm Weber
Scientific Computing, METU _______________

Assoc. Prof. Dr. Sevtap Kestel
Financial Mathematics, METU _______________

Assoc. Prof. Dr. Ömür Uğur
Scientific Computing, METU _______________

Assist. Prof. Dr. Fikriye Nuray Yılmaz
Department of Mathematics, Gazi University _______________

Date: _______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: SELİM YAYLA

Signature :

ABSTRACT

REGULATORY NETWORKS STUDIED BY ELLIPSOIDAL CALCULUS

Yayla, Selim
M.S., Department of Scientific Computing
Supervisor : Prof. Dr. Gerhard-Wilhelm Weber
Co-Supervisor : Assoc. Prof. Dr. Vilda Purutçuoğlu

September 2015, 57 pages

The identification of regulatory networks affected by noise and data uncertainty is a serious problem in many Operational Research applications. The fundamental structure of underlying systems can be established by regulatory networks in many sector like ecology, education and finance. After clustering and classification methods gene/target and environmental states can be grouped into functional behaviour. The analysis of complex regulatory systems under uncertainty is a compounded complex by the unknown interactions between the variables which are represented by ellipsoids. Ellipsoidal calculus is used in determination of the explicit representations of the uncertain multivariate states of the system. MATLAB Ellipsoidal Toolbox (ET) provides efficient plotting routines of ellipsoids, hyperplanes and reach sets. In this thesis, several regression models are studied in order to approximate regulatory networks under ellipsoidal uncertainty and Ellipsoidal Toolbox routines are explained for representing a parameter estimation and inverse problem.

Keywords : Regulatory networks, regression, ellipsoid, optimization, uncertainty

ÖZ

DÜZENLEYİCİ AĞLARININ ELİPSOİTSEL ANALİZ İLE ARAŞTIRMASI

Yayla, Selim
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Gerhard-Wilhelm Weber
Ortak Tez Yöneticisi : Doç. Dr. Vilda Purutçuoğlu

Eylül, 57 sayfa

Yöneylem araştırmasındaki; veri belirsizliği ve kirliliğinden etkilenen düzenleyici ağların tanımlanması önemli bir sorundur. Düzenleyici-ağları; çevrebilim, eğitim ve maliye alanlarında genel bir yapı sağlamaktadır. Kümeleme ve sınıflama yöntemlerinden sonra işlevsel olarak ilgili hedef ve çevre çarpan grupları belirlenebilir. Belirsizlik içeren karmaşık düzenleyeci ağların çözümlenmesi; sağlam regresyondaki elipsoitler ile temsil edilen değişkenler arasındaki bilinmeyen etkileşimler yüzünden aşamalı olarak karmaşıklaşmaktadır. Çok değişkenli belirsiz durumların açık gösterimini elde etmek için sistemlere elipsoitsel analiz uygulanabilir. MATLAB Ellipsoidal Toolbox (ET), elipsoitler, hiperdüzlemler ve hedef kümelerin grafiğini çizmek için verimli fonksiyonlar sunar. Bu tezde düzenleyici-ağları, elipsoitsel belirsizlik altında tahminlemek için birçok regresyon modeli ve bir parametre tahminleme ve ters problemi imgelemek için Ellipsoidal Toolbox daki araçlar çalışılmıştır.

Anahtar Kelimeler: Hesaplamalı biyoloji, regresyon, elipsoit, optimizasyon, belirsizlik

To My Family

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisor Prof. Dr. Gerhard-Wilhelm Weber and co-supervisor Assoc. Prof. Dr. Vilda Purutçuoğlu for their precious ideas, patience and encouragement. This thesis would not have been possible without their support.

I also would like to express my sincere thanks to my defence committee members, Prof. Dr. Bülent Karasözen, Assoc. Prof. Dr. Sevtap Kestel, Assoc. Prof. Dr. Ömür Uğur and Assist. Prof. Dr. Fikriye Nuray Yılmaz for their feedback and understanding.

I am especially grateful to my brother Assist. Prof. Dr. Oğuz Yayla for his guidance.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

New interconnected technological systems gather huge amount of data. Here, we have very complex systems with many variables and parameters to be analysed. Such systems are known as complex regulatory systems, appearing in biology, finance and engineering sciences. We have some tools in mathematical analysis to understand these systems. The parameters of the coupling rules and the time-dependent states of clusters define our regulatory networks. Target-Environment (TE) regulatory systems consist of two distinguishable groups of data, although they are strongly connected: the environmental factors and their dependent target variables. So, this class of regulatory systems is suitable for many real-world applications.

In gene-environment networks, to understand the relations and interactions between genes and other objects in a cell,tissue or mixture of some environmental unit, we carry out research into the models of gene-expression and environmental states and their predictions. Here, the gene expressions are called as the target variables, and toxins, transcription factors, radiation, etc., are called as the environmental items. Ecological networks, social networks, metabolic networks and immunological networks are other examples in which TE-regulatory systems are used.

TE-models are generally affected by noise and uncertainty. Spline regression and stochastic differential equations are used in order to handle the errors and uncertainty in regression models. Bounding on the uncertain variable with ellipsoids has been proved to be suitable for case of data corrupted by noise (see [12]). Ellipsoids are very resilient in targets and environmental perspective. Confidence ellipsoids are used with respect to stochastic dependencies of the error variables, e.g., in micro-array experiments.

On the other hand, clustering and classification identify structure of data items which act jointly. The uncertain states of these clusters are represented by ellipsoids and ellipsoidal calculus is applied to model the dynamics of the TE-regulatory system. The unknown parameters of the time-discrete TE-model are also clustered to make uncertain (ellipsoidal) measurement data suitable for the regression. Complex regulatory systems usually consist of a large number of interconnected components, and the TE-regulatory network is highly systematized with multiple interactions between different clusters. For practical reasons of regularity or stability, bounds on the degrees of the nodes are applied to reduce some of branches in the TE-regulatory network [11].

Furthermore, uncertainty of gene-environment networks is analyzed with several regression models, e.g., interval arithmetic, stochastic differential equations and spline regression models. Bounding each variable individually does not reveal correlations in multivariate clusters of data. A representation of ellipsoids is used for uncertainty sets with the help of its variance–covariance matrices and distributions. Ellipsoids are more flexible than intervals and parallel pipes with respect to correlations of the data. Also, ellipsoids are bigger than paraxial sets and their orthogonal projections. Moreover, confidence ellipsoids are obtained from stochastic dependencies among any two of the errors made in the measurements of the expression values of genes and environmental levels, e.g., through microarray procedures and environmental technologies [12]. Ellipsoidal states are representations of uncertain expression values of gene-environment clusters.

Finally, ellipsoidal calculus is applied for estimating time-discrete regulatory model of interactions between target and environmental factors. Shape matrix and distance between centers of the ellipsoids are used in the objective function of the regression model. For example, a regression model for parameter estimation of the regulatory system is based on the trace and determinant. After estimation, their solvability can be analyzed by semidefinite programming or conic programming, with their interior point methods [13].

Accordingly, this thesis is planned in a way as follows: In Chapter 2, we present the analysis of regulatory systems. In Chapter 3, we study the optimization of time-discrete regression models under ellipsoidal uncertainty. Then, we demonstrate an inverse problem mathematically with a graphical problem and implement its MATLAB algorithm in Chapter 4. By a conclusion and an outlook, we conclude the thesis in Chapter 5. In appendices, we also offer both definitions of ellipsoidal calculus related to basic regression theory and ellipsoidal toolbox functions with a compact course on MATLAB.

CHAPTER 2

SYSTEMS UNDER ELLIPSOIDAL UNCERTAINTY

2.1 Ellipsoidal Uncertainty

An *ellipsoid* in $\mathbb{R}^p$ with center $c \in \mathbb{R}^p$ and a symmetric non-negative definite *configuration (shape)* matrix $\Sigma \in \mathbb{R}^{p \times p}$ is given as follows:

$$\mathcal{E}(c, \Sigma) = \{\Sigma u + c \mid \|u\|_2 \leq 1\},$$

where $\| \cdot \|_2$ denotes the Euclidean norm, and u is a point in the full unit ball. Then, under ellipsoidal uncertainty the target variables $X_r^{(\kappa)},\ r = 1, \ldots, R$, and the environmental variables $E_s^{(\kappa)},\ s = 1, \ldots, S$, satisfy

$$\begin{aligned} X_r^{(\kappa)} &\in \mathcal{E}\left(\overline{X}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)}\right) \subset \mathbb{R}^{|C_r|}, \\ E_s^{(\kappa)} &\in \mathcal{E}\left(\overline{E}_r^{(\kappa)}, \overline{\Pi}_s^{(\kappa)}\right) \subset \mathbb{R}^{|D_s|}, \end{aligned}$$

where $\overline{\Sigma}_r^{(\kappa)}$ and $\overline{\Pi}_s^{(\kappa)}$ are the variance-covariance matrices of cluster data, C_r is a target cluster, D_s is an environmental cluster, $\kappa = 1, \ldots, T$, is the sampling index, and R and S are the number of overlapping clusters of target and environmental items, respectively [6].

2.2 Gene-Environment Systems

Gene-environment networks are consists of two type of variables: gene and environmental items, e.g., proteins are gene expression and toxins are environmental factors. The relation between the various clusters of genes and environmental factors can be complex and unstable. This yields to uncertainty. Ellipsoidal set representation is very suitable for representing uncertain states. Also, ellipsoids are very flexible to variations in the measurements [4].

2.2.1 Disjoint Clusters

Let us divide n genes into R disjoint clusters $C_r \subset \{1, \ldots, n\},\ r = 1, \ldots, R$, and m environmental factors into S disjoint clusters $D_s \subset \{1, \ldots, m\},\ s = 1, \ldots, S$. Assume that $C_{r_1} \cap C_{r_2} = \emptyset$ when $\forall r_1 \neq r_2$ and $D_{s_1} \cap D_{s_2} = \emptyset$ when $\forall s_1 \neq s_2$ relations are satisfied for strict subset of the variables [1].

2.2.2 Overlapping Clusters

Assume that the vectors $\mathbb{X} = (\mathbb{X}_1, \ldots, \mathbb{X}_n)'$ and $\mathbb{E} = (\mathbb{E}_1, \ldots, \mathbb{E}_m)'$ establish the expression values of the genes and environmental factors, respectively. We symbolize transposition of a matrix or vector by $(\cdot)'$. R overlapping clusters of gene items $C_r \subset \{1, \ldots, n\},\ r = 1, \ldots, R$, and S overlapping clusters of environmental item $D_s \subset \{1, \ldots, m\},\ s = 1, \ldots, S$, are established such that $C_{r_1} \cap C_{r_2} = \emptyset$ when $\forall r_1 \neq r_2$ and $D_{s_1} \cap D_{s_2} = \emptyset$ when $\forall s_1 \neq s_2$ relations may not hold. Specifically, X_r is a $|C_r|$-subvector of $\mathbb{X}$ matched with the elements of C_r which are put together by a gene cluster. E_s is a $|D_s|$-subvector of $\mathbb{E}$ assigned to each clusters of environmental factors which is given by the elements of D_s. $\mathbb{X}$ and $\mathbb{E}$ subvectors define *crisp states* of the clusters. The vector X_r represents ellipsoidal states of the genes in cluster C_r given by the ellipsoid $\mathcal{E}(\mu_r, \Sigma_r) \subset \mathbb{R}^{|C_r|}$ for r-dimensional vector of mean μ_r and covariance matrix of $(r \times r)$-dimensional Σ_r. Similarly, E_s represents the ellipsoidal states of the environmental items in the cluster D_s given by the ellipsoid $\mathcal{E}(\rho_s, \Pi_s) \subset \mathbb{R}^{|D_s|}$. Therefore, *uncertain states* of the clusters are identified by error ellipsoids. The ellipsoid $\mathcal{E}(\mu_r, \Sigma_r)$ is parametrized by $|C_r| + |C_r|^2$ coefficients and the ellipsoid $\mathcal{E}(\rho_s, \Pi_s)$ is determined by $|D_s| + |D_s|^2$ variables. The number of coefficients can be reduced by assuming symmetric shape matrices what refers to specific correlation of the data variables. When we know exactly enough number of variables, flat ellipsoids $\mathcal{E}(\mu_r, \Sigma_r)$ and $\mathcal{E}(\rho_s, \Pi_s)$ are data sets. This situation can be avoided by decreasing bounds on the semiaxes distances or an artificial part added in the specified coordinate directions of distance $\epsilon > 0$ [11]. Identically, by increasing bounds, degenerate or needleshaped ellipsoids are prevented from happening.

Before clustering, numerous overlapping clusters of genes and environmental factors can be determined. But the noise in the data or the fact of existing outliers and errors yield to possibly overlapping clusters of gene-networks. Discrete-time dynamics are used in order to determine the distinctive attribute or characteristics of the regression. Set-theoretic regression models are suggested in order to estimate the unknown system parameters and their solvability by various methods [8].

The clusters of a gene-environment network can attribute importance by the time-dependent ellipsoidal states. The affine (linear) coupling rules can be used to represent the relations between the clusters.

2.2.3 Linear Model

We note that

$$X_r = \mathcal{E}\left(\mu_r, \Sigma_r\right) \subset \mathbb{R}^{|C_r|},\ E_s = \mathcal{E}\left(\rho_s, \Pi_s\right) \subset \mathbb{R}^{|D_s|},$$

is an *ellipsoidal* representation of the two clusters.

To guarantee representations of ellipsoid in all time states, we define a system by (affine) linear coupling rules [11]:

$$\left.\begin{aligned} X_j^{(\kappa+1)} &= \xi_{j0} + \left(\bigoplus_{r=1}^{R} \mathcal{A}_{jr}^{GG} X_r^{(\kappa)}\right) + \left(\bigoplus_{s=1}^{S} \mathcal{A}_{js}^{EG} E_s^{(\kappa)}\right), \\ E_i^{(\kappa+1)} &= \zeta_{i0} + \left(\bigoplus_{r=1}^{R} \mathcal{A}_{ir}^{GE} X_r^{(\kappa)}\right) + \left(\bigoplus_{s=1}^{S} \mathcal{A}_{is}^{EE} E_s^{(\kappa)}\right), \end{aligned}\right\} \tag{EC}$$

is the linear model of the relations and dynamics between the clusters, where $\kappa \geq 0$ and for $j = 1, 2, \ldots, R,\ i = 1, 2, \ldots, S$, the (affine) linear coupling given by $X_j^{(0)} = \overline{X}_j^{(0)}$ and $E_i^{(0)} = \overline{E}_i^{(0)}$ are initial values.

In time-discrete gene-environment regulatory systems, we have the diagram below:

(GG) gene $\xrightarrow{regulates}$ gene,

(EG) environment $\xrightarrow{regulates}$ gene,

(GE) gene $\xrightarrow{regulates}$ environment,

(EE) environment $\xrightarrow{regulates}$ environment.

This diagram defines cluster interactions and regulating effects. In an ellipsoidal or set theoretic manner,

$$\bigoplus_{r=1}^{R} \mathcal{A}_{jr}^{GG} X_r^{(\kappa)},\ \bigoplus_{s=1}^{S} \mathcal{A}_{js}^{EG} E_s^{(\kappa)},$$

are the *cumulative effects* of all clusters exercise on the elements of cluster C_j. Similarly,

$$\bigoplus_{r=1}^{R} \mathcal{A}_{ir}^{GE} X_r^{(\kappa)},\ \bigoplus_{s=1}^{S} \mathcal{A}_{is}^{EE} E_s^{(\kappa)},$$

are the *additive genetic and environmental factors* on E_i clusters. The unknown interactions matrices:

$$\mathcal{A}_{jr}^{GG} \subset \mathbb{R}^{|C_j|\times|C_r|}, \mathcal{A}_{js}^{EG} \subset \mathbb{R}^{|Cj|\times|D_s|}, \mathcal{A}_{ir}^{GE} \subset \mathbb{R}^{|D_i|\times|C_r|}, \mathcal{A}_{is}^{EE} \subset \mathbb{R}^{|D_i|\times|D_s|},$$

which are sub-matrices of the general form

$$\mathcal{A}^{GG} \in \mathbb{R}^{n\times n}, \mathcal{A}^{EG} \in \mathbb{R}^{n\times m}, \mathcal{A}^{GE} \in \mathbb{R}^{m\times n}, \mathcal{A}^{EE} \in \mathbb{R}^{m\times m},$$

gives the *degree of connectivity* between distinct clusters. The general interactions matrices are the bases for the sub-matrices. The position of the general interactions matrices are defined by the particular cluster elements. The intercept vector $\xi_{j0} \in \mathbb{R}^{|C_j|}$ is a disjoint component of $\xi_0 = (\xi_{10}, \ldots, \xi_{n0})' \in \mathbb{R}^n$ and $\zeta_{i0} \in \mathbb{R}^{|D_i|}$ is a disjoint component of $\zeta_0 = (\zeta_{10}, \ldots, \zeta_{m0})' \in \mathbb{R}^m$, respectively.

2.2.4 Algorithm

Let us determine ellipsoidal time states of the clusters in (EC) linear system by applying ellipsoidal calculus. Then, the centers and configuration matrices of the predictions $X_j^{(\kappa+1)}$ and $E_s^{(\kappa+1)}$, respectively, can be calculated iteratively [13]. Here, κ denotes the number of iteration.

Assume that $\kappa \geq 0$ and an ellipsoid

$$X_j^{(\kappa+1)} = \mathcal{E}\left(\mu_j^{(\kappa+1)}, \Sigma_j^{(\kappa+1)}\right)$$

is given with center

$$\mu_j^{(\kappa+1)} = \xi_{j0} + \sum_{r=1}^{R} A_{jr}^{GG} \mu_r^{(\kappa)} + \sum_{s=1}^{S} A_{js}^{EG} \rho_s^{(\kappa)}$$

and configuration matrix

$$\Sigma_j^{(\kappa+1)} = \left(\sqrt{\operatorname{Tr} \mathcal{G}_j^{(\kappa)}} + \sqrt{\operatorname{Tr} \mathcal{H}_j^{(\kappa)}}\right) \left(\frac{\mathcal{G}_j^{(\kappa)}}{\sqrt{\operatorname{Tr} \mathcal{G}_j^{(\kappa)}}} + \frac{\mathcal{H}_j^{(\kappa)}}{\sqrt{\operatorname{Tr} \mathcal{H}_j^{(\kappa)}}}\right),$$

where

$$\mathcal{G}_j^{(\kappa)} = \left(\sum_{r=1}^{R} \sqrt{\operatorname{Tr}\left(A_{jr}^{GG} \Sigma_r^{(\kappa)} \left(A_{jr}^{GG}\right)^T\right)}\right) \left(\sum_{r=1}^{R} \frac{A_{jr}^{GG} \Sigma_r^{(\kappa)} \left(A_{jr}^{GG}\right)^T}{\sqrt{\operatorname{Tr}\left(A_{jr}^{GG} \Sigma_r^{(\kappa)} \left(A_{jr}^{GG}\right)^T\right)}}\right),$$

$$\mathcal{H}_j^{(\kappa)} = \left(\sum_{s=1}^{S} \sqrt{\operatorname{Tr}\left(A_{js}^{EG} \Pi_s^{(\kappa)} \left(A_{js}^{EG}\right)^T\right)}\right) \left(\sum_{s=1}^{S} \frac{A_{js}^{EG} \Pi_s^{(\kappa)} \left(A_{js}^{EG}\right)^T}{\sqrt{\operatorname{Tr}\left(A_{js}^{EG} \Pi_s^{(\kappa)} \left(A_{js}^{EG}\right)^T\right)}}\right).$$

Then, genetic cluster states $C_j,\ j = 1, 2, \ldots, R$, are defined by the above ellipsoid.

In a similar manner, the states of the environmental cluster $D_i,\ i = 1, 2, \ldots, S$, corresponds to an ellipsoid

$$E_i^{(\kappa+1)} = \mathcal{E}\left(\rho_i^{(\kappa+1)}, \Pi_i^{(\kappa+1)}\right)$$

with center

$$\rho_i^{(\kappa+1)} = \zeta_{i0} + \sum_{r=1}^{R} A_{ir}^{GE} \mu_r^{(\kappa)} + \sum_{s=1}^{S} A_{is}^{EE} \rho_s^{(\kappa)}$$

and configuration matrix

$$\Pi_i^{(\kappa+1)} = \left(\sqrt{\mathrm{Tr}\, \mathcal{M}_i^{(\kappa)}} + \sqrt{\mathrm{Tr}\, \mathcal{N}_i^{(\kappa)}} \right) \left(\frac{\mathcal{M}_i^{(\kappa)}}{\sqrt{\mathrm{Tr}\, \mathcal{M}_i^{(\kappa)}}} + \frac{\mathcal{N}_i^{(\kappa)}}{\sqrt{\mathrm{Tr}\, \mathcal{N}_i^{(\kappa)}}} \right),$$

where

$$\mathcal{M}_i^{(\kappa)} = \left(\sum_{r=1}^{R} \sqrt{\mathrm{Tr} \left(A_{ir}^{GE} \Sigma_r^{(\kappa)} \left(A_{ir}^{GE} \right)^T \right)} \right) \left(\sum_{r=1}^{R} \frac{A_{ir}^{GE} \Sigma_r^{(\kappa)} \left(A_{ir}^{GE} \right)^T}{\sqrt{\mathrm{Tr} \left(A_{ir}^{GE} \Sigma_r^{(\kappa)} \left(A_{ir}^{GE} \right)^T \right)}} \right),$$

$$\mathcal{N}_i^{(\kappa)} = \left(\sum_{s=1}^{S} \sqrt{\mathrm{Tr} \left(A_{is}^{EE} \Pi_s^{(\kappa)} \left(A_{is}^{EE} \right)^T \right)} \right) \left(\sum_{s=1}^{S} \frac{A_{is}^{EE} \Pi_s^{(\kappa)} \left(A_{is}^{EE} \right)^T}{\sqrt{\mathrm{Tr} \left(A_{is}^{EE} \Pi_s^{(\kappa)} \left(A_{is}^{EE} \right)^T \right)}} \right).$$

2.2.5 Regression Analysis

The unknown system parameters of linear model (EC) are estimated by the entries of the interaction matrices $\mathcal{A}_{jr}^{GG}, \mathcal{A}_{js}^{EG}, \mathcal{A}_{ir}^{GE}$ and $\mathcal{A}_{is}^{EE}$ also by the intercepts ξ_{j0} and ζ_{i0}. For regression analysis, we have to compare the predictions of the linear model (EC) and the real data obtained from genetic and environmental measurements [12]. The ellipsoidal measurements

$$\begin{aligned} \overline{X}_r^{(\kappa)} &= \mathcal{E} \left(\overline{\mu}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)} \right) \subset \mathbb{R}^{|C_r|}, \\ \overline{E}_s^{(\kappa)} &= \mathcal{E} \left(\overline{\rho}_s^{(\kappa)}, \overline{\Pi}_s^{(\kappa)} \right) \subset \mathbb{R}^{|D_s|}, \end{aligned}$$

with $r = 1, 2, \ldots, R;\ s = 1, 2, \ldots, S$, and $\kappa = 0, 1, \ldots, T$, are taken at specified time states $t_0 < t_1 < \ldots < t_T$ and the first T predictions of the linear model are identified by the ellipsoids

$$\begin{aligned} \widehat{X}_j^{(\kappa+1)} &= \mathcal{E} \left(\widehat{\mu}_j^{(\kappa+1)}, \widehat{\Sigma}_j^{(\kappa+1)} \right) := \xi_{j0} + \left(\bigoplus_{r=1}^{R} \mathcal{A}_{jr}^{GG} \overline{X}_r^{(\kappa)} \right) + \left(\bigoplus_{s=1}^{S} \mathcal{A}_{js}^{EG} \overline{E}_s^{(\kappa)} \right), \\ \widehat{E}_i^{(\kappa+1)} &= \mathcal{E} \left(\widehat{\rho}_i^{(\kappa+1)}, \widehat{\Pi}_i^{(\kappa+1)} \right) := \zeta_{i0} + \left(\bigoplus_{r=1}^{R} \mathcal{A}_{ir}^{GE} \overline{X}_r^{(\kappa)} \right) + \left(\bigoplus_{s=1}^{S} \mathcal{A}_{is}^{EE} \overline{E}_s^{(\kappa)} \right), \end{aligned}$$

with $j = 1, 2, \ldots, R,\ i = 1, 2, \ldots, S$, and $\kappa = 0, 1, \ldots, T-1$.

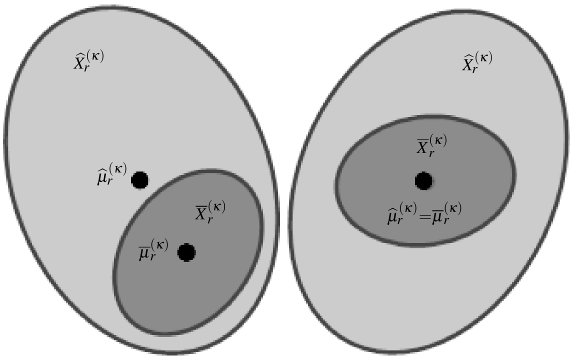
$\widehat{X}_r^{(\kappa)}$
$\widehat{\mu}_r^{(\kappa)}$
$\overline{X}_r^{(\kappa)}$
$\overline{\mu}_r^{(\kappa)}$
$\widehat{X}_r^{(\kappa)}$
$\overline{X}_r^{(\kappa)}$
$\widehat{\mu}_r^{(\kappa)}=\overline{\mu}_r^{(\kappa)}$

Figure 2.1: Overlap of Ellipsoids.

We have fusion (intersection) of ellipsoids ($\cap$ operation), defined by

$$\Delta X_r^{(\kappa)} := \widehat{X}_r^{(\kappa)} \cap \overline{X}_r^{(\kappa)} \text{ and } \Delta E_s^{(\kappa)} := \widehat{E}_s^{(\kappa)} \cap \overline{E}_s^{(\kappa)},$$

with $r = 1, 2, \ldots, R$, $s = 1, 2, \ldots, S$, and $\kappa = 1, \ldots, T$. We aim to maximize the overlap of the ellipsoid representations of predictions and measurement values with a regression problem

$$\text{maximize} \sum_{\kappa=1}^{T} \left\{ \sum_{r=1}^{R} \left\| \Delta X_r^{(\kappa)} \right\|_* - \left\| \widehat{\mu}_r^{(\kappa)} - \overline{\mu}_r^{(\kappa)} \right\|_2^2 \right. \qquad (R)$$
$$\left. + \sum_{s=1}^{S} \left\| \Delta E_s^{(\kappa)} \right\|_* - \left\| \widehat{\rho}_s^{(\kappa)} - \overline{\rho}_s^{(\kappa)} \right\|_2^2 \right\},$$

where $\| \cdot \|_*$ is the size of the intersections, and $\|\Delta X_r^{(\kappa)}\|_* = 0$ if $\Delta X_r^{(\kappa)} = \emptyset$, and $\|\Delta E_s^{(\kappa)}\|_* = 0$ is assumed if $\Delta E_s^{(\kappa)} = \emptyset$. In Figure 2.1, on the left, ellipsoids $\widehat{X}_r^{(\kappa)}$ and $\overline{X}_r^{(\kappa)}$ intersect and the size as well as the measure of the fusions are equal on the figure. The right graphics' $\widehat{\mu}_r^{(\kappa)}$ and $\overline{\mu}_r^{(\kappa)}$ centers are shifted so that the distance from one to another is minimized.

Fusions $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$ are sized by the objective function of (R). In general, $\psi(\mathcal{E}(0, Q))$ nonnegative-valued criteria functions ($\psi(\mathcal{E}_1) \leq \psi(\mathcal{E}_2)$ if $\mathcal{E}_1 \subseteq \mathcal{E}_2$) are defined on the set of all nondegenerate ellipsoids. In order to evaluate the volume of a p-dimensional ellipsoid $\mathcal{E}(0, Q)$, mentioned monotonously increasing functions are put into operation. Let us introduce some of them from [14]:

(a) the *trace* of Q,

$$\psi_T \left(\mathcal{E} \left(0, Q \right) \right) := \operatorname{Tr} Q = \lambda_1 + \ldots + \lambda_p,$$

where λ_i are the eigenvalues of Q;

(b) the *trace of square* of Q,

$$\psi_{TS} \left(\mathcal{E} \left(0, Q \right) \right) := \operatorname{Tr} Q^2,$$

which corresponds to the sum of squares of semiaxes;

(c) the *diameter*,

$$\psi_{Dia}\left(\mathcal{E}\left(0,Q\right)\right) := \operatorname{diam}\left(\mathcal{E}\left(0,Q\right)\right) := d,$$

where

$$\max\left\{\lambda_i \in \mathbb{R} \mid i = 1,\ldots,p\right\} = \left(\frac{d}{2}\right)^2,$$

is the $\mathcal{E}(0,Q)$ parametrization of the p-dimensional sphere with $d/2$ radius;

(d) *the determinant of* Q,

$$\psi_{Det}\left(\mathcal{E}\left(0,Q\right)\right) := \det Q = \lambda_1 \cdot \ldots \cdot \lambda_p,$$

which evaluates the product of the eigenvalues. Therefore, it has a constant ratio of the volume of the following ellipsoid

$$\operatorname{vol}\mathcal{E}\left(0,Q\right) = \pi^{\frac{p}{2}}\left(\det Q\right)^{\frac{1}{2}}\left(\Gamma\left(\frac{p}{2}+1\right)\right)^{-1},$$

where the Gamma-function is written as Γ.

Taking the measurements of the shape of the intersection ellipsoids, with tools defined above, allow us to reformulate the regression model (R) [11, 12].

2.2.5.1 The Trace Criterion

It is based on the intersections of $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$ *traces*. Both of them are shape matrices. The sizes of these ellipsoids are measured by the squared lengths of the semiaxes which are indeed the traces of the shape matrices $\Delta\Sigma_r^{(\kappa)}$ and $\Delta\Pi_s^{(\kappa)}$:

$$\begin{aligned}\text{maximize} \sum_{\kappa=1}^{T}\Bigg\{\sum_{r=1}^{R}&\left[\operatorname{Tr}\left(\Delta\Sigma_r^{(\kappa)}\right) - \sum_{j=1}^{|C_r|}\left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)}\right)^2\right] \\ &+\sum_{s=1}^{S}\left[\operatorname{Tr}\left(\Delta\Pi_s^{(\kappa)}\right) - \sum_{i=1}^{|D_s|}\left(\widehat{p}_{s,i}^{(\kappa)} - \overline{p}_{s,i}^{(\kappa)}\right)^2\right]\Bigg\}.\end{aligned} \tag{R_{Tr}}$$

By substituting trace of the shape matrix of an ellipsoid with the sum of the squares of the semiaxes, the regression problem turns into

$$\begin{aligned}\text{maximize} \sum_{\kappa=1}^{T}\Bigg\{\sum_{r=1}^{R}&\left[\sum_{j=1}^{|C_r|}\lambda_{r,j}^{(\kappa)} - \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)}\right)^2\right] \\ &+\sum_{s=1}^{S}\left[\sum_{i=1}^{|D_s|}\Lambda_{s,i}^{(\kappa)} - \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)}\right)^2\right]\Bigg\},\end{aligned} \tag{R'_{Tr}}$$

where $\lambda_{r,j}^{(\kappa)}$ and $\Lambda_{s,i}^{(\kappa)}$ are the eigenvalues of $\Delta\Sigma_r^{(\kappa)}$ and $\Delta\Pi_s^{(\kappa)}$, respectively.

2.2.5.2 The Trace of the Square Criterion

The volume of an ellipsoid is evaluated with the help of the *traces of the square* of its shape matrix. Now, we achieve the following regression problem:

$$\begin{aligned}\text{maximize} \sum_{\kappa=0}^{T} \Bigg\{ & \sum_{r=1}^{R} \left[\operatorname{Tr}\left(\Delta\Sigma_r^{(\kappa)}\right)^2 - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[\operatorname{Tr}\left(\Delta\Pi_s^{(\kappa)}\right)^2 - \sum_{i=1}^{|D_s|} \left(\widehat{p}_{s,i}^{(\kappa)} - \overline{p}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}. \end{aligned} \qquad (R_{TS})$$

2.2.5.3 The Diameter Criterion

The *diameter* of the ellipsoids $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$ has the form of a sphere which contains all the fusions in least size. By writing them in the objective function, we get:

$$\begin{aligned}\text{maximize} \sum_{\kappa=0}^{T} \Bigg\{ & \sum_{r=1}^{R} \left[\operatorname{diam}\left(\mathcal{E}\left(0, \Delta\Sigma_r^{(\kappa)}\right)\right) - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[\operatorname{diam}\left(\mathcal{E}\left(0, \Delta\Pi_s^{(\kappa)}\right)\right) - \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}. \end{aligned} \qquad (R_{Dia})$$

In eigenvalues terms of $\Delta\Sigma_r^{(\kappa)}$ and $\Delta\Pi_s^{(\kappa)}$, we note:

$$\begin{aligned}\text{maximize} \sum_{\kappa=1}^{T} \Bigg\{ & \sum_{r=1}^{R} \left[2 \cdot \sqrt{\lambda_r^{(\kappa)}} - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[2 \cdot \sqrt{\Lambda_s^{(\kappa)}} - \sum_{i=1}^{|D_s|} \left(\widehat{p}_{s,i}^{(\kappa)} - \overline{p}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}, \end{aligned} \qquad (R'_{Dia})$$

with $\lambda_r^{(\kappa)} := \max\{\lambda_{r,j}^{(\kappa)} \mid j = 1, \ldots, |C_r|\}$ and $\Lambda_s^{(\kappa)} := \max\{\Lambda_{s,i}^{(\kappa)} \mid i = 1, \ldots, |D_s|\}$. So we have a new regression problem:

$$\begin{aligned}\text{maximize} \sum_{\kappa=0}^{T} \Bigg\{ & \sum_{r=1}^{R} \left[\lambda_r^{(\kappa)} - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[\Lambda_s^{(\kappa)} - \sum_{i=1}^{|D_s|} \left(\widehat{p}_{s,i}^{(\kappa)} - \overline{p}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}, \end{aligned} \qquad (R''_{Dia})$$

as an alternative proposal.

2.2.5.4 The Determinant Criterion

This type of objective function of the regression model can be set up on the determinants of the configuration matrices of the ellipsoids $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$ which can be used as a basis for objective function of the regression model:

$$\begin{aligned} \text{maximize} \sum_{\kappa=1}^{T} \Bigg\{ \sum_{r=1}^{R} & \left[\det\left(\Delta\Sigma_r^{(\kappa)}\right) - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[\det\left(\Delta\Pi_s^{(\kappa)}\right) - \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}. \end{aligned} \tag{R_{Det}}$$

The model can be rewritten with respect to the eigenvalues of the configuration matrices:

$$\begin{aligned} \text{maximize} \sum_{\kappa=1}^{T} \Bigg\{ \sum_{r=1}^{R} & \left[\left(\prod_{j=1}^{|C_r|} \lambda_{r,j}^{(\kappa)} \right) - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \\ & + \sum_{s=1}^{S} \left[\left(\prod_{i=1}^{|D_s|} \Lambda_{s,i}^{(\kappa)} \right) - \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}, \end{aligned} \tag{R'_{Det}}$$

and the volumes of the ellipsoids $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$ are addressed explicitly in the following problem:

$$\begin{aligned} \text{maximize} \sum_{\kappa=1}^{T} \Bigg\{ \sum_{r=1}^{R} & \Bigg[\left(\pi^{\frac{2}{|C_r|}} \Gamma\left(\frac{|C_r|}{2} + 1 \right) \mathrm{vol}\left(\Delta X_r^{(\kappa)}\right) \right)^2 \\ & - \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \Bigg] \\ & + \sum_{s=1}^{S} \Bigg[\left(\pi^{\frac{2}{|D_S|}} \Gamma\left(\frac{|D_s|}{2} + 1 \right) \mathrm{vol}\left(\Delta E_s^{(\kappa)}\right) \right)^2 \\ & - \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \Bigg] \Bigg\}. \end{aligned} \tag{R''_{Det}}$$

2.3 Target-Environment Systems

Regulatory systems that are effected by noise and uncertainty can be analyzed by target-environment system. The time-discrete target-environment regulatory systems consist of n targets and m environmental factors and, thus, constitute two-modal systems. The target elements are valued by the following vectors $\mathbb{X} = [\mathbb{X}_1, \ldots, \mathbb{X}_n]'$ and $\mathbb{E} = [\mathbb{E}_1, \ldots, \mathbb{E}_n]'$ which denote the states of the environmental variables.

The set of all targets and environments could be partitioned in $C_r \subset \{1, \dots, n\}$ where $r = 1, \dots, R$, and $D_S \subset \{1, \dots, m\},\ s = 1, \dots, S$, clusters respectively. The clusters might be disjoint or overlapping. The restrictions of partitions are represented by $X_r \in \mathbb{R}^{|C_r|}$ and $E_s \in \mathbb{R}^{|D_s|}$, respectively.

2.3.1 The Time-Discrete Model

A discrete-time model for the states of the targets $\mathbb{X}_j,\ j = 1, \dots, n$, and the environmental factors $\mathbb{E}_i,\ i = 1, \dots, m$.

Four types of interactions and regulating effects exist [10]:

(TT) target cluster → target variable,

(ET) environmental cluster → target vaiable,

(TE) target cluster → environment variable,

(EE) environmental cluster → environment variable.

The target-environment regulatory cluster model:

$$\left.\begin{aligned} \mathbb{X}_j^{(\kappa+1)} &= \zeta_{j0}^T + \sum_{r=1}^{R} \left[X_r^{(\kappa)}\right]' \Theta_{jr}^{TT} + \sum_{s=1}^{S} \left[E_s^{(\kappa)}\right]' \Theta_{js}^{ET}, \\ \mathbb{E}_i^{(\kappa+1)} &= \zeta_{i0}^E + \sum_{r=1}^{R} \left[X_r^{(\kappa)}\right]' \Theta_{ir}^{TE} + \sum_{s=1}^{S} \left[E_s^{(\kappa)}\right]' \Theta_{is}^{EE}, \end{aligned}\right\} \tag{CM}$$

with $\kappa \geq 0$. The additional parameters $\zeta_{j0}^T, \zeta_{i0}^E \in \mathbb{R}$ are intercepts. Here, $\Theta_{jr}^{TT},\ \Theta_{ir}^{TE}$ are $|C_r|$ subvectors of $\Theta_j^{TT} \in \mathbb{R}^n$ and $\Theta_i^{TE} \in \mathbb{R}^n$; Θ_{js}^{ET} and Θ_{is}^{EE} are $|D_s|$ subvectors of $\Theta_j^{ET} \in \mathbb{R}^m$ and $\Theta_i^{EE} \in \mathbb{R}^m$ respective parameter vectors.

2.3.2 The Regression Problem

By substituting measurements $\overline{\mathbb{X}}^{(\kappa)} = \left[\overline{\mathbb{X}}_1^{(\kappa)}, \dots, \overline{\mathbb{X}}_n^{(\kappa)}\right]'$, $\overline{\mathbb{E}}^{(\kappa)} = \left[\overline{\mathbb{X}}_1^{(\kappa)}, \dots, \overline{\mathbb{X}}_m^{(\kappa)}\right]'$ where $\overline{\mathbb{X}}^{(\kappa)}, \overline{\mathbb{E}}^{(\kappa)} \in \mathbb{R}^n$ are obtained at $t_0 < t_1 < \dots < t_T$ sampling times for $\kappa = 0, 1, \dots, T$, into (CM), we have:

$$\begin{aligned} \widehat{\mathbb{X}}_j^{(\kappa+1)} &= \zeta_{j0}^T + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa)}\right]' \Theta_{jr}^{TT} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa)}\right]' \Theta_{js}^{ET}, \\ \widehat{\mathbb{E}}_i^{(\kappa+1)} &= \zeta_{i0}^E + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa)}\right]' \Theta_{ir}^{TE} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa)}\right]' \Theta_{is}^{EE}, \end{aligned}$$

where $\kappa = 0, 1, \dots, T-1;\ i = 1, \dots, n;\ j = 1, \dots, m.$

By comparing measurements and predictions in following regression problem, we obtain:

$$\text{minimize} \sum_{\kappa=1}^{T} \left\{ \sum_{j=1}^{n} \left| \widehat{\mathbb{X}}_j^{\kappa} - \overline{\mathbb{X}}_j^{\kappa} \right| + \sum_{j=1}^{m} \left| \widehat{\mathbb{E}}_i^{\kappa} - \overline{\mathbb{E}}_i^{\kappa} \right| \right\}. \tag{RP}$$

The ellipsoidal states turns the TE-regulatory network in a dynamic graph model instead of a static one. Hereby, network analysis features from discrete mathematics like connectedness, cycles and shortest paths could be applied to assess the importance of ellipsoidal clusters [7].

CHAPTER 3

OPTIMIZATION METHODS

In this chapter, we state optimization methods rely on the shape matrices $\Sigma_r^{(\kappa)}$ and $\Pi_s^{(\kappa)}$, the distance of the centers $\mu_r^{(\kappa)}$ and $\rho_s^{(\kappa)}$ of the fusions $\Delta X_r^{(\kappa)}$ and $\Delta E_s^{(\kappa)}$. Objective functions of these 3-dimensional regression problems mostly depend on the determinant or eigenvalues of symmetric positive semidefinite matrices [5]. However, positive semidefinite representable objective functions are not always available. Therefore, objective functions have to be modified.

3.1 Semidefinite Programming

The problem (R_{Det}) depends heavily on $\det(M)$ in which M is a symmetric positive semidefinite matrix $(M \succ 0)$ has a size of $n \times n$ [4]. If $n \geq 2$, then the function of determinant is neither a convex nor a concave function. However, if $p \in \mathbb{Q}$ with $0 \leq p \leq \dfrac{1}{n}$, then it can be defined as follows:

$$f(M) = \begin{cases} -\det^p(M) & \text{, if } M \succ 0 \text{ (i.e. positive definite),} \\ \infty & \text{, otherwise,} \end{cases}$$

So, the regression model takes the form:

$$\begin{aligned} \text{maximize} \sum_{\kappa=1}^{T} \Bigg\{ & -\sum_{r=1}^{R} \left[\det^p \left(\Delta\Sigma_r^{(\kappa)} \right) + \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \qquad (\widetilde{R}_{Det}) \\ & -\sum_{s=1}^{S} \left[\det^q \left(\Delta\Pi_s^{(\kappa)} \right) + \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \Bigg\}, \end{aligned}$$

where $p, q \in \mathbb{Q}$ with $0 \leq p \leq \dfrac{1}{|C_r|}$ and $0 \leq q \leq \dfrac{1}{|D_s|}$. Since $\det(M) = \prod_{i=1}^{n} \lambda_i(M)$, where $\lambda_i(M)$ are the eigenvalues of M, we can rewrite the determinant criterion (R'_{Det})

$$\text{maximize} \sum_{\kappa=1}^{T} \left\{ -\sum_{r=1}^{R} \left[\left(\prod_{j=1}^{|C_r|} \lambda_{r,j}^{(\kappa)} \right)^{p} + \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \right. \qquad (\widetilde{R}'_{Det})$$
$$\left. -\sum_{s=1}^{S} \left[\left(\prod_{i=1}^{|D_s|} \Lambda_{s,i}^{(\kappa)} \right)^{q} + \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \right\},$$

and instead of (R''_{Det}) we state

$$\text{maximize} \sum_{\kappa=1}^{T} \left\{ \sum_{r=1}^{R} \left[\left(\pi^{\frac{2}{|C_r|}} \Gamma\left(\frac{|C_r|}{2} + 1 \right) \operatorname{vol}\left(\Delta X_r^{(\kappa)} \right) \right)^{2p} \right. \right. \qquad (\widetilde{R}''_{Det})$$
$$\left. -\sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right]$$
$$+\sum_{s=1}^{\widehat{\rho}} \left[\left(\pi^{\frac{2}{|D_s|}} \Gamma\left(\frac{|D_s|}{2} + 1 \right) \operatorname{vol}\left(\Delta E_s^{(\kappa)} \right) \right)^{2q} \right.$$
$$\left. \left. -\sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \right\}.$$

If $\Delta\Sigma_r^{(\kappa)}$ and $\Delta\Pi_s^{(\kappa)}$ are positive definite shape matrices then determinants with negative exponent can be used [13]. The matrix $M \in \mathbb{R}^{n \times n}$ is mapped as a positive semidefinite matrix by

$$f(M) = \begin{cases} \det^{-p}(M) & \text{, if } M \succ 0, \\ \infty & \text{, otherwise,} \end{cases}$$

when $p \in \mathbb{Q}$ and $p > 0$, i.e., it is a positive rational function. As a final note, when $p > 0,\ q > 0$ and $p, q \in \mathbb{Q}$, our model turns into

$$\text{maximize} \sum_{\kappa=1}^{T} \left\{ \sum_{r=1}^{R} \left[\det^{-p}\left(\Delta\Sigma_r^{(\kappa)} \right) + \sum_{j=1}^{|C_r|} \left(\widehat{\mu}_{r,j}^{(\kappa)} - \overline{\mu}_{r,j}^{(\kappa)} \right)^2 \right] \right. \qquad (R'''_{Det})$$
$$\left. +\sum_{s=1}^{S} \left[\det^{-q}\left(\Delta\Pi_s^{(\kappa)} \right) + \sum_{i=1}^{|D_s|} \left(\widehat{\rho}_{s,i}^{(\kappa)} - \overline{\rho}_{s,i}^{(\kappa)} \right)^2 \right] \right\}.$$

In trace criterion (R'_{Tr}), all eigenvalues from configuration matrices are added into $\Delta\Sigma_r^{(\kappa)}$ and $\Delta\Pi_s^{(\kappa)}$ are positive semidefinite representable functions. Also *interior point methods* can be considered but they are more difficult to apply. Alternatively, *gradient methods* from *bilevel programming* can be introduced when calculating sums of eigenvalues in the objective function [16].

3.2 Mixed-Integer Regression Problem

Gene-environment networks have vast amount of highly-interconnected genes and environmental clusters. Some clusters which have a weak influence on the system, can be identified by regression analysis. By deleting negligible degrees of connectivity between clusters, we can reduce complexity [13]. We require *binary constraints* between two clusters for giving a judgement of cluster boundaries.

Given two A, B clusters, $A \leftarrow B$ says that cluster B regulates cluster A.

In contrast, $A \not\leftarrow B$ denotes that A is not regulated by cluster B. Then, we define the Boolean matrices for testing of relations:

$$\chi_{jr}^{GG} = \begin{cases} 1, \text{ if } C_j \leftarrow C_r, \\ 0, \text{ if } C_j \not\leftarrow -C_r, \end{cases} \quad \chi_{ir}^{EG} = \begin{cases} 1, \text{ if } D_i \leftarrow C_r, \\ 0, \text{ if } D_i \not\leftarrow -C_r, \end{cases}$$

$$\chi_{js}^{GE} = \begin{cases} 1, \text{ if } C_j \leftarrow D_s, \\ 0, \text{ if } C_j \not\leftarrow -D_s, \end{cases} \quad \chi_{is}^{EE} = \begin{cases} 1, \text{ if } D_i \leftarrow D_s, \\ 0, \text{ if } D_i \not\leftarrow -D_s. \end{cases}$$

If two clusters do not interact with each other, then not all A^{GG}, A^{GE}, A^{EG}, A^{EE} matrices, but some elements are zero-valued when a disjoint cluster decomposition occurs.

The *indegree* of the genetic cluster C_j in a regulatory network in terms of whole genetic and environmental clusters is expressed with

$$\deg(C_j)_{in}^{GG} := \sum_{r=1}^{R} \chi_{jr}^{GG}, \ \deg(C_j)_{in}^{EG} := \sum_{s=1}^{S} \chi_{js}^{EG},$$

where $j \in \{1, \ldots, R\}$.

The indegrees $\deg(C_j)^{GG}$ and $\deg(C_j)^{EG}$ are the sums of genetic and environmental clusters which regulate C_j cluster, respectively. The *overall indegree* of the genetic C_j cluster is expressed as

$$\deg(C_j)_{in} \ = \deg(C_j)_{in}^{GG} + \deg(C_j)_{in}^{EG}.$$

Correspondingly, for $i \in \{1, \ldots, S\}$ the *indegree* of D_i cluster in connection with the environmental clusters and the genetic cluster is stated as

$$\deg(D_i)_{in}^{GE} := \sum_{r=1}^{R} \chi_{ir}^{GE}, \ \deg(D_i)_{in}^{EE} := \sum_{s=1}^{S} \chi_{is}^{EE}.$$

Likewise, the indegrees $\deg(D_i)^{GE}$ and $\deg(D_i)^{EE}$ are the sums of genetic and environmental clusters which regulate D_i cluster. The *overall indegree* of D_i cluster is stated as

$$\deg(D_i)_{in} := \deg(D_i)_{in}^{GE} + \deg(D_i)_{in}^{EE}.$$

In parallel, bounds on the outdegree, i.e., the sum of *outgoing branches*, are identified. The binary values are created in order to check outgoing connection:

$$\zeta_{jr}^{GG} = \begin{cases} 1, \text{ if } C_r \leftarrow C_j, \\ 0, \text{ if } C_r \nleftarrow C_j, \end{cases} \quad \zeta_{ir}^{GE} = \begin{cases} 1, \text{ if } C_r \leftarrow D_i, \\ 0, \text{ if } C_r \nleftarrow D_i, \end{cases}$$

$$\zeta_{js}^{EG} = \begin{cases} 1, \text{ if } D_s \leftarrow C_j, \\ 0, \text{ if } D_s \nleftarrow C_j, \end{cases} \quad \zeta_{is}^{EE} = \begin{cases} 1, \text{ if } D_s \leftarrow D_i, \\ 0, \text{ if } D_s \nleftarrow D_i. \end{cases}$$

Then, the *outdegree* of genetic C_j cluster in terms of whole genetic and environmental clusters can be stated as

$$\deg(C_j)_{out}^{GG} := \sum_{r=1}^{R} \zeta_{jr}^{GG}, \ \deg(C_j)_{out}^{GE} := \sum_{s=1}^{S} \zeta_{js}^{GE},$$

where $j \in \{1, \ldots, R\}$. The outdegrees $\deg(C_j)_{out}^{GG}$ and $\deg(C_j)_{out}^{EG}$ determine the sum of genetic and environmental clusters regulated by C_j cluster. The *overall outdegree* of C_j genetic cluster is given by

$$\deg(C_j)_{out} := \deg(C_j)_{out}^{GG} + \deg(C_j)_{out}^{EG}.$$

The outdegree of environmental D_i cluster in connection with the environmental clusters and the genetic clusters is defined by

$$\deg(D_i)_{out}^{GE} := \sum_{r=1}^{R} \zeta_{ir}^{GE}, \ \deg(D_i)_{out}^{EE} := \sum_{s=1}^{S} \zeta_{is}^{EE},$$

where $i \in \{1, \ldots, S\}$.

The outdegrees $\deg(D_i)_{out}^{GE}$ and $\deg(D_i)_{out}^{EE}$ determine the sum of genetic and environmental clusters regulated by D_i cluster. The *overall outdegree* of cluster D_i is

$$\deg(D_i)_{out} := \deg(D_i)_{out}^{GE} + \deg(D_i)_{out}^{EE}.$$

Including particular bounds on the indegrees and outdegrees of the genetic and the environmental cluster separately, we obtain:

$$\left.\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{\kappa=1}^{T}\left\{\sum_{r=1}^{R}\left\|\Delta X_r^{(\kappa)}\right\|_* - \left\|\widehat{\mu}_r^{(\kappa)} - \overline{\mu}_r^{(\kappa)}\right\|_2^2\right. \\
& \displaystyle\qquad\left.+\sum_{s=1}^{S}\left\|\Delta E_s^{(\kappa)}\right\|_* - \left\|\widehat{\rho}_s^{(\kappa)} - \overline{\rho}_s^{(\kappa)}\right\|_2^2\right\} \\
\text{subject to} & \\
& \qquad \deg(C_j)_{in}^{GG} \le \alpha_j^{GG},\ j = 1, \ldots, R, \\
& \qquad \deg(C_j)_{in}^{EG} \le \alpha_j^{EG},\ j = 1, \ldots, R, \\
& \qquad \deg(D_i)_{in}^{GE} \le \alpha_i^{GE},\ i = 1, \ldots, S, \\
& \qquad \deg(D_i)_{in}^{EE} \le \alpha_i^{EE},\ i = 1, \ldots, S, \\
& \\
& \qquad \deg(C_j)_{out}^{GG} \le \beta_j^{GG},\ j = 1, \ldots, R, \\
& \qquad \deg(C_j)_{out}^{EG} \le \beta_j^{EG},\ j = 1, \ldots, R, \\
& \qquad \deg(D_i)_{out}^{GE} \le \beta_i^{GE},\ i = 1, \ldots, S, \\
& \qquad \deg(D_i)_{out}^{EE} \le \beta_i^{EE},\ i = 1, \ldots, S.
\end{array}\right\} \qquad \text{(MI1)}$$

Now, joined the bounds and put a restriction on all nodes of individual cluster:

$$\left.\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{\kappa=1}^{T}\left\{\sum_{r=1}^{R}\left\|\Delta X_r^{(\kappa)}\right\|_* - \left\|\widehat{\mu}_r^{(\kappa)} - \overline{\mu}_r^{(\kappa)}\right\|_2^2\right. \\
& \displaystyle\qquad\left.+\sum_{s=1}^{S}\left\|\Delta E_s^{(\kappa)}\right\|_* - \left\|\widehat{\rho}_s^{(\kappa)} - \overline{\rho}_s^{(\kappa)}\right\|_2^2\right\} \\
\text{subject to} & \\
& \qquad \deg(C_j)_{in} \le \gamma_j,\ j = 1, \ldots, R, \\
& \qquad \deg(D_i)_{in} \le \delta_i,\ i = 1, \ldots, S, \\
& \qquad \deg(C_j)_{out} \le \epsilon_j,\ j = 1, \ldots, R, \\
& \qquad \deg(D_i j s_{out} \le \varphi_i,\ i = 1, \ldots, S.
\end{array}\right\} \qquad \text{(MI2)}$$

3.3 Continuous Programming

Real-valued continuous constraints $P_{jr}^{GG},\ P_{js}^{EG},\ P_{ir}^{GE},\ P_{is}^{EE} \in [0,1]$ are less strict than $\chi_{jr}^{GG},\ \chi_{js}^{EG},\ \chi_{ir}^{GE},\ \chi_{is}^{EE}$ binary constraints for relaxing in (MI1) and (MI2) [17]. Furthermore, $\mathcal{A}_{jr}^{GG},\ \mathcal{A}_{js}^{EG},\ \mathcal{A}_{ir}^{GE},\ \mathcal{A}_{is}^{EE}$ are bases of continuous variables.

In terms of the genetic and environmental clusters, C_j gives:

$$\deg(C_j)_{in}^{GG} := \sum_{r=1}^{R} P_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right),\ \deg(C_j)_{in}^{EG} := \sum_{s=1}^{S} P_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right),$$

where C_j is a cluster with real-valued *indegree*. In the same way, D_i cluster with the real-valued *indegree* can be stated as

$$\deg(D_i)_{in}^{GE} := \sum_{r=1}^{R} P_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right),$$

$$\deg(D_i)_{in}^{EE} := \sum_{s=1}^{S} P_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right).$$

Similarly, we can make the outdegrees of clusters suitable for a new use by substituting the binary variables ζ_{jr}^{GG}, ζ_{js}^{EG}, ζ_{ir}^{GE} and ζ_{is}^{EE} with Q_{jr}^{GG}, Q_{js}^{EG}, Q_{ir}^{GE}, $Q_{is}^{EE} \in [0,1]$ real variables which can be expressed in $\mathcal{A}_{jr}^{GG}$, $\mathcal{A}_{js}^{EG}$, $\mathcal{A}_{ir}^{GE}$, $\mathcal{A}_{is}^{EE}$ expressions. Cluster C_j with the real-valued *outdegrees* is given by the help of defined by GE clusters

$$\deg(C_j)_{out}^{GG} := \sum_{r=1}^{R} Q_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right),$$

$$\deg(C_j)_{out}^{EG} := \sum_{s=1}^{S} Q_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right).$$

Also, the cluster D_i with the real-valued *outdegree* can be defined as

$$\deg(D_i)_{out}^{GE} := \sum_{r=1}^{R} P_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right),$$

$$\deg(D_i)_{out}^{EE} := \sum_{s=1}^{S} Q_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right).$$

To protect the significant clusters of the regulatory network to be deleted, we substitute the strict binary constraints in (MI1) with the continuous soft constraints [13], then we acquire following model:

$$
\left.\begin{aligned}
&\text{maximize } \sum_{\kappa=1}^{T}\left\{\sum_{r=1}^{R}\left\|\Delta X_r^{(\kappa)}\right\|_* - \left\|\widehat{\mu}_r^{(\kappa)} - \overline{\mu}_r^{(\kappa)}\right\|_2^2\right.\\
&\qquad\qquad\left. + \sum_{s=1}^{S}\left\|\Delta E_s^{(\kappa)}\right\|_* - \left\|\widehat{\rho}_s^{(\kappa)} - \overline{\rho}_s^{(\kappa)}\right\|_2^2\right\}\\
&\text{subject to}\\
&\qquad\qquad \sum_{r=1}^{R} P_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right) \le \alpha_j^{GG},\ j = 1, \ldots, R,\\
&\qquad\qquad \sum_{s=1}^{R} P_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right) \le \alpha_j^{EG},\ j = 1, \ldots, R,\\
&\qquad\qquad \sum_{r=1}^{R} P_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right) \le \alpha_i^{GE},\ i = 1, \ldots, S,\\
&\qquad\qquad \sum_{s=1}^{R} P_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right) \le \alpha_i^{EE},\ i = 1, \ldots, S,\\
&\\
&\qquad\qquad \sum_{r=1}^{R} Q_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right) \le \beta_j^{GG},\ j = 1, \ldots, R,\\
&\qquad\qquad \sum_{s=1}^{R} Q_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right) \le \beta_j^{EG},\ j = 1, \ldots, R,\\
&\qquad\qquad \sum_{r=1}^{R} Q_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right) \le \beta_i^{GE},\ i = 1, \ldots, S,\\
&\qquad\qquad \sum_{s=1}^{R} Q_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right) \le \beta_i^{EE},\ i = 1, \ldots, S.
\end{aligned}\right\} \tag{C1}
$$

The model (MI2) can be relaxed in the following form by fitting real-valued limits on all cluster nodes separately:

$$\left.\begin{aligned}
&\text{maximize } \sum_{\kappa=1}^{T}\left\{\sum_{r=1}^{R}\left\|\Delta X_r^{(\kappa)}\right\|_* - \left\|\widehat{\mu}_r^{(\kappa)} - \overline{\mu}_r^{(\kappa)}\right\|_2^2 \right.\\
&\qquad\qquad \left. + \sum_{s=1}^{S}\left\|\Delta E_s^{(\kappa)}\right\|_* - \left\|\widehat{\rho}_s^{(\kappa)} - \overline{\rho}_s^{(\kappa)}\right\|_2^2\right\}\\
&\text{subject to}\\
&\qquad\qquad \sum_{r=1}^{R} P_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right) + \sum_{s=1}^{R} P_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right) \leq \gamma_j^{GG},\ j=1,\ldots,R,\\
&\qquad\qquad \sum_{r=1}^{R} P_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right) + \sum_{s=1}^{R} P_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right) \leq \beta_i^{GE},\ i=1,\ldots,S,\\
&\\
&\qquad\qquad \sum_{r=1}^{R} Q_{jr}^{GG}\left(\mathcal{A}_{jr}^{GG}\right) + \sum_{s=1}^{R} Q_{js}^{EG}\left(\mathcal{A}_{js}^{EG}\right) \leq \gamma_j^{EG},\ j=1,\ldots,R,\\
&\qquad\qquad \sum_{r=1}^{R} Q_{ir}^{GE}\left(\mathcal{A}_{ir}^{GE}\right) + \sum_{s=1}^{R} Q_{is}^{EE}\left(\mathcal{A}_{is}^{EE}\right) \leq \delta i^{EE},\ i=1,\ldots,S.
\end{aligned}\right\} \tag{C2}$$

3.4 Robust Regression under Ellipsoidal Uncertainty

We now turn to an estimation of parameters of the linear cluster model (CM) under crisp states. One can write the Regression Problem (RP) model affected by uncertainty as follows:

$$\left.\begin{aligned}
&\text{minimize } \sum_{\kappa=1}^{T}\left\{\sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)}\right\}\\
&\text{such that } \left|\widehat{\mathbb{X}}_j^{(\kappa)} - \overline{\mathbb{X}}_j^{(\kappa)}\right| \leq p_j^{(\kappa)}\\
&\qquad\qquad \left|\widehat{\mathbb{E}}_i^{(\kappa)} - \overline{\mathbb{E}}_i^{(\kappa)}\right| \leq q_i^{(\kappa)},\\
&\qquad\qquad (\kappa=1,\ldots,T;\ i=1,\ldots,m;\ j=1,\ldots,n).
\end{aligned}\right\} \tag{RP}$$

By substituting the equations with corresponding satisfied prediction variables, we can write (RP) as

$$
\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T} \left\{ \sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)} \right\} \\
\text{such that } & \left| \zeta_{j0}^{T} + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa-1)} \right]' \Theta_{jr}^{TT} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa-1)} \right]' \Theta_{js}^{ET} - \overline{\mathbb{X}}_j^{(\kappa)} \right| \leq p_j^{(\kappa)}, \\
& \left| \zeta_{i0}^{E} + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa-1)} \right]' \Theta_{ir}^{TE} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa-1)} \right]' \Theta_{is}^{EE} - \overline{\mathbb{E}}_i^{(\kappa)} \right| \leq q_i^{(\kappa)} \\
& (\kappa = 1, \ldots, T;\ i = 1, \ldots, m;\ j = 1, \ldots, n).
\end{aligned}
$$

Hence, the Robust Regression Problem with uncertain ellipsoidal state reads as follows:

$$
\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T} \left\{ \sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)} \right\} \\
\text{such that } & \left| \zeta_{j0}^{T} + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa-1)} \right]' \Theta_{jr}^{TT} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa-1)} \right]' \Theta_{js}^{ET} - \overline{\mathbb{X}}_j^{(\kappa)} \right| \leq p_j^{(\kappa)}, \\
& \left| \zeta_{i0}^{E} + \sum_{r=1}^{R} \left[\overline{X}_r^{(\kappa-1)} \right]' \Theta_{ir}^{TE} + \sum_{s=1}^{S} \left[\overline{E}_s^{(\kappa-1)} \right]' \Theta_{is}^{EE} - \overline{\mathbb{E}}_i^{(\kappa)} \right| \leq q_i^{(\kappa)} \\
& (\kappa = 1, \ldots, T;\ i = 1, \ldots, n;\ j = 1, \ldots, m) \\
& \forall X_r^{(\kappa)} \in \mathcal{E}\left(\overline{X}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)} \right)\ (\kappa = 0, \ldots, T-1;\ r = 1, \ldots, R), \\
& \forall E_s^{(\kappa)} \in \mathcal{E}\left(\overline{E}_s^{(\kappa)}, \overline{\Pi}_s^{(\kappa)} \right)\ (\kappa = 0, \ldots, T-1;\ s = 1, \ldots, S).
\end{aligned}
$$

The above problem can be rewritten:

$$\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T}\left\{\sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)}\right\} \\
\text{such that } & \zeta_{j0}^{T} + \sum_{r=1}^{R}\left[X_r^{(\kappa-1)}\right]'\Theta_{jr}^{TT} + \sum_{s=1}^{S}\left[E_s^{(\kappa-1)}\right]'\Theta_{js}^{ET} - \overline{\mathbb{X}}_j^{(\kappa)} \leq p_j^{(\kappa)}, \\
& -\zeta_{j0}^{T} + \sum_{r=1}^{R}\left[X_r^{(\kappa-1)}\right]'\Theta_{jr}^{TT} - \sum_{s=1}^{S}\left[E_s^{(\kappa-1)}\right]'\Theta_{js}^{ET} + \overline{\mathbb{X}}_j^{(\kappa)} \leq p_j^{(\kappa)}, \\
& \zeta_{i0}^{E} + \sum_{r=1}^{R}\left[X_r^{(\kappa-1)}\right]'\Theta_{ir}^{TE} + \sum_{s=1}^{S}\left[E_s^{(\kappa-1)}\right]'\Theta_{is}^{EE} - \overline{\mathbb{E}}_i^{(\kappa)} \leq q_i^{(\kappa)}, \\
& -\zeta_{i0}^{E} - \sum_{r=1}^{R}\left[X_r^{(\kappa-1)}\right]'\Theta_{ir}^{TE} - \sum_{s=1}^{S}\left[E_s^{(\kappa-1)}\right]'\Theta_{is}^{EE} + \overline{\mathbb{E}}_i^{(\kappa)} \leq q_i^{(\kappa)} \\
& (\kappa = 1, \ldots, T;\ i = 1, \ldots, m;\ j = 1, \ldots, n) \\
& \forall X_r^{(\kappa)} \in \mathcal{E}\left(\overline{X}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)}\right)\ (\kappa = 0, \ldots, T-1;\ r = 1, \ldots, R), \\
& \forall E_s^{(\kappa)} \in \mathcal{E}\left(\overline{E}_s^{(\kappa)}, \overline{\Pi}_s^{(\kappa)}\right)\ (\kappa = 0, \ldots, T-1;\ s = 1, \ldots, S).
\end{aligned}$$

By another reformulation employing:

$$\begin{aligned}
\mathcal{E}\left(\overline{X}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)}\right) &= \left\{\overline{X}_r^{(\kappa)} + \overline{\Sigma}_r^{(\kappa)} u_r \mid \|u_r\|_2 \leq 1\right\}, \\
\mathcal{E}\left(\overline{E}_s^{(\kappa)}, \overline{\Pi}_s^{(\kappa)}\right) &= \left\{\overline{E}_s^{(\kappa)} + \overline{\Pi}_s^{(\kappa)} v_s \mid \|v_s\|_2 \leq 1\right\},
\end{aligned}$$

with

$$\begin{aligned}
U_r &:= \left\{u_r \in \mathbb{R}^{|C_r|} \mid \|u_r\|_2 \leq 1\right\},\ r = 1, \ldots, R, \\
V_s &:= \left\{v_s \in \mathbb{R}^{|D_s|} \mid \|v_s\|_2 \leq 1\right\},\ s = 1, \ldots, S,
\end{aligned}$$

we then obtain the following equivalent problem:

$$
\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T}\left\{\sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)}\right\} \\
\text{such that } & \zeta_{j0}^{T} + \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{jr}^{TT} + \sum_{r=1}^{R} \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa-1)}\Theta_{jr}^{TT}\right\} \\
& + \sum_{r=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{js}^{ET} + \sum_{s=1}^{S} \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa-1)}\Theta_{js}^{ET}\right\} \\
& - \overline{\mathbb{X}}_j^{(\kappa)} \leq p_j^{(\kappa)} \ (\kappa = 1,\ldots,T;\ j = 1,\ldots,n), \\
& -\zeta_{j0}^{T} - \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{jr}^{TT} - \sum_{r=1}^{R} \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa-1)}\Theta_{jr}^{TT}\right\} \\
& - \sum_{r=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{js}^{ET} - \sum_{s=1}^{S} \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa-1)}\Theta_{js}^{ET}\right\} \\
& + \overline{\mathbb{X}}_j^{(\kappa)} \leq p_j^{(\kappa)} \ (\kappa = 1,\ldots,T;\ j = 1,\ldots,n), \\
& \zeta_{i0}^{E} + \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{ir}^{TE} + \sum_{r=1}^{R} \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa-1)}\Theta_{ir}^{TE}\right\} \\
& + \sum_{r=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{is}^{EE} + \sum_{s=1}^{S} \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa-1)}\Theta_{is}^{EE}\right\} \\
& - \overline{\mathbb{E}}_i^{(\kappa)} \leq q_i^{(\kappa)} \ (\kappa = 1,\ldots,T;\ i = 1,\ldots,m), \\
& -\zeta_{i0}^{E} - \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{ir}^{TE} - \sum_{r=1}^{R} \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa-1)}\Theta_{ir}^{TE}\right\} \\
& - \sum_{r=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{is}^{EE} - \sum_{s=1}^{S} \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa-1)}\Theta_{is}^{EE}\right\} \\
& + \overline{\mathbb{E}}_i^{(\kappa)} \leq q_i^{(\kappa)} \ (\kappa = 1,\ldots,T,\ i = 1,\ldots,m).
\end{aligned}
$$

With the help of the following equalities

$$
\begin{aligned}
& \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa)}\Theta_{jr}^{TT}\right\} = \max_{u_r\in U_r}\left\{-u_r'\overline{\Sigma}_r^{(\kappa)}\Theta_{jr}^{TT}\right\} = \left\|\overline{\Sigma}_r^{(\kappa)}\Theta_{jr}^{TT}\right\|_2, \\
& \max_{u_r\in U_r}\left\{u_r'\overline{\Sigma}_r^{(\kappa)}\Theta_{ir}^{TE}\right\} = \max_{u_r\in U_r}\left\{-u_r'\overline{\Sigma}_r^{(\kappa)}\Theta_{ir}^{TE}\right\} = \left\|\overline{\Sigma}_r^{(\kappa)}\Theta_{ir}^{TE}\right\|_2, \\
& \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa)}\Theta_{js}^{ET}\right\} = \max_{v_s\in V_s}\left\{-v_s'\overline{\Pi}_s^{(\kappa)}\Theta_{js}^{ET}\right\} = \left\|\overline{\Pi}_s^{(\kappa)}\Theta_{js}^{ET}\right\|_2, \\
& \max_{v_s\in V_s}\left\{v_s'\overline{\Pi}_s^{(\kappa)}\Theta_{is}^{EE}\right\} = \max_{v_s\in V_s}\left\{-v_s'\overline{\Pi}_s^{(\kappa)}\Theta_{is}^{EE}\right\} = \left\|\overline{\Pi}_s^{(\kappa)}\Theta_{is}^{EE}\right\|_2, \\
& (\kappa = 1,\ldots,T;\ j = 1,\ldots,n;\ i = 1,\ldots,m),
\end{aligned}
$$

we obtain that

$$\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T}\left\{\sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)}\right\}\\
\text{such that } & \left|\zeta_{j0}^{T} + \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{jr}^{TT} + \sum_{s=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{js}^{ET} - \overline{\mathbb{X}}_j^{(\kappa)}\right|\\
& + \sum_{r=1}^{R}\left\|\overline{\Sigma}_r^{(\kappa-1)}\Theta_{jr}^{ET}\right\|_2 + \sum_{s=1}^{S}\left\|\overline{\Pi}_s^{(\kappa-1)}\Theta_{js}^{ET}\right\|_2 \le p_j^{(\kappa)},\\
& \left|\zeta_{i0}^{E} + \sum_{r=1}^{R}\left[\overline{X}_r^{(\kappa-1)}\right]' \Theta_{ir}^{TE} + \sum_{s=1}^{S}\left[\overline{E}_s^{(\kappa-1)}\right]' \Theta_{is}^{EE} - \overline{\mathbb{E}}_i^{(\kappa)}\right|\\
& + \sum_{r=1}^{R}\left\|\overline{\Sigma}_r^{(\kappa-1)}\Theta_{ir}^{TE}\right\|_2 + \sum_{s=1}^{S}\left\|\overline{\Pi}_s^{(\kappa-1)}\Theta_{is}^{EE}\right\|_2 \le q_i^{(\kappa)},\\
& (\kappa = 1,\ldots,T;\ j = 1,\ldots,n;\ i = 1,\ldots,m).
\end{aligned}$$

For simplification we define the vectors:

$$\Theta_j^T = \left[\zeta_{j0}^T, \Theta_{j1}^{TT}, \ldots, \Theta_{jR}^{TT}, \Theta_{j1}^{ET}, \ldots, \Theta_{jS}^{ET}\right]',$$

$$\Theta_i^E = \left[\zeta_{i0}^E, \Theta_{i1}^{TE}, \ldots, \Theta_{iR}^{TE}, \Theta_{i1}^{EE}, \ldots, \Theta_{iS}^{EE}\right]',$$

$$c^{(\kappa)} = \left[1, \overline{X}_1^{(\kappa)}, \ldots, \overline{X}_R^{(\kappa)}, \overline{E}_1^{(\kappa)}, \ldots, \overline{E}_S^{(\kappa)}\right]'.$$

Then we have the following problem [10]:

$$\left.\begin{aligned}
\text{minimize } & \sum_{\kappa=1}^{T}\left\{\sum_{j=1}^{n} p_j^{(\kappa)} + \sum_{i=1}^{m} q_i^{(\kappa)}\right\}\\
\text{such that } & \left|\left[c^{(\kappa-1)}\right]' \Theta_j^T - \overline{\mathbb{X}}_j^{(\kappa)}\right|\\
& + \sum_{r=1}^{R}\left\|\overline{\Sigma}_r^{(\kappa-1)}\Theta_{jr}^{TT}\right\|_2 + \sum_{s=1}^{S}\left\|\overline{\Pi}_s^{(\kappa-1)}\Theta_{js}^{ET}\right\|_2 \le p_j^{(\kappa)},\\
& \left|\left[c^{(\kappa-1)}\right]' \Theta_i^E - \overline{\mathbb{E}}_i^{(\kappa)}\right|\\
& + \sum_{r=1}^{R}\left\|\overline{\Sigma}_r^{(\kappa-1)}\Theta_{ir}^{TE}\right\|_2 + \sum_{s=1}^{S}\left\|\overline{\Pi}_s^{(\kappa-1)}\Theta_{is}^{EE}\right\|_2 \le q_i^{(\kappa)}\\
& (\kappa = 1,\ldots,T;\ j = 1,\ldots,n;\ i = 1,\ldots,m).
\end{aligned}\right\} \qquad \text{(RCPE)}$$

In this chapter, we examined inverse problems for target-environment networks under ellipsoidal uncertainty methodically and in detail. These systems are especially suited for parameter identification of gene-environment networks in computational biology and genetics along with applications of eco-finance networks [9]. In the subsequent chapter, we shall give a basic application of ellipsoidal techniques in standard regression methodology.

CHAPTER 4

AN APPLICATION OF ELLIPSOIDAL TECHNIQUES IN REGRESSION

Estimating real-world problems from observed data is very popular among scientists from very different disciplines, because it can be used to determine existence, uniqueness and stability of the solutions.

In the following, G is a function which establishes a relation between model (parameter) $\mathbf{m}$, that is an n-element vector, and (response) data are also called observation $\mathbf{d}$, that is an m-element vector. Let us define a mathematical model as

$$G(\mathbf{m}) = \mathbf{d}.$$

When observing physical world, noise η in the data is inevitable. Thus, one can define the following model

$$\begin{aligned} \mathbf{d} &= G(\mathbf{m}_{true}) + \boldsymbol{\eta} \\ &= \mathbf{d}_{true} + \boldsymbol{\eta}, \end{aligned}$$

where $\mathbf{d}_{true}$ and $\mathbf{m}_{true}$ state ideal cases.

Estimating finite vector-wise $\mathbf{m}$ from $\mathbf{d}$ is called a *discrete inverse problem (parameter estimation problem)* [3]. Also, when $\mathbf{d}$ and $\mathbf{m}$ are functions, then, the model can often be turned into a discrete inverse problem by discretizing.

Fitting of a function to a data set via linear regression is a linear parameter estimation problem. In the real-world, linearity is very common among very important inverse problems such as ballistic positioning. Our quadratic mathematical model in ballistic trajectory is

$$y(t) = m_1 + m_2 t - (1/2) m_3 t^2,$$

, where t is observation time, m_1 is initial altitude, m_2 is initial vertical velocity and m_3 is effective gravitational acceleration. Gravitational acceleration is multiplied by -(1/2) because altitude is decreasing. We try to solve for the unknown m model with monitoring altitude of the body y_i at time t_i. Measurement unit of the time is second, and the distance is measured in meter. The data are considered to be independent and Gaussian distributed.

We form a mathematical model based on linear equations of the model parameters:

$$\begin{bmatrix} 1 & t_1 & (-\frac{1}{2})t_1^2 \\ 1 & t_2 & (-\frac{1}{2})t_2^2 \\ . & . & . \\ . & . & . \\ . & . & . \\ 1 & t_m & (-\frac{1}{2})t_m^2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_m \end{bmatrix}.$$

To solve the model by least-squares estimation, we construct the parabolic system matrix $\mathbf{G}$. The rows of $\mathbf{G}$ are given by

$$\mathbf{G}_i = [1, t_i, -(1/2)t_i^2].$$

Then, we build the following parabolic system (or design) matrix

$$\mathbf{G} = \begin{bmatrix} 1.0000 & 1.0000 & -0.5000 \\ 1.0000 & 2.0000 & -2.0000 \\ 1.0000 & 3.0000 & -4.5000 \\ 1.0000 & 4.0000 & -8.0000 \\ 1.0000 & 5.0000 & -12.5000 \\ 1.0000 & 6.0000 & -18.0000 \\ 1.0000 & 7.0000 & -24.5000 \\ 1.0000 & 8.0000 & -32.0000 \\ 1.0000 & 9.0000 & -40.5000 \\ 1.0000 & 10.0000 & -50.0000 \end{bmatrix}.$$

So, $\mathbf{G}$ has full rank.

The precise solution is

$$\mathbf{m}_{true} = [10m, 100m/s, 9.8m/s^2]'.$$

The randomly generated vertical position of a ballistic object from a true solution with mean $\mu = 0$, standard deviation $\sigma = 8$ at a time t is obtained by using the MATLAB code below.

```
clear;
rand('state',0);
randn('state',1);
m_1 = 10;
m_2 = 100;
m_3 = 9.8;
for t=1:10
 sgm(t,1) = 8;
 y(t,1)=m_1+m_2*t-(1/2*m_3)*t^2+sgm(t,1)*randn;
end
```

So, the obtained randomly generated vertical position data are:

t	y
1.0000	112.0152
2.0000	191.1536
3.0000	259.0847
4.0000	338.5880
5.0000	383.9957
6.0000	430.1627
7.0000	461.0782
8.0000	499.5700
9.0000	505.3806
10.0000	521.3476

Number of model parameters are less than data points, so that finding $\mathbf{m}_{true}$ is very unlikely because there usually exist errors in the data and the model function is not exact. Thus, the solution will not satisfy the system. In a geometric perspective, a parabola does not exists that goes all data points through. However, estimated solutions to such problems may be found approximately. The difference between the observations $\mathbf{y}$ and the mathematical model $\mathbf{Gm}$, namely,

$$\mathbf{y} - \mathbf{Gm}$$

is called the *residual (prediction error)*. Data errors are assumed to be independent and to have a normal distribution, so that the least-squares method can be used to find a unique solution [3]. In least-squares estimation, the goal is to find an m that minimizes the 2-norm (Euclidean distance) of the residual:

$$||\mathbf{y} - \mathbf{Gm}||_2 = \sqrt{\sum_{i=1}^{m} (y_i - \mathbf{Gm}_i)^2}.$$

In general form, we note Least-Squares as

$$||\mathbf{d} - \mathbf{Gm}_{L_2}||_2^2 = \min ||\mathbf{d} - \mathbf{Gm}||_2^2 . \tag{LS}$$

The diagonal weighting matrix is represented as given below:

$$\begin{aligned}\mathbf{W} &= \text{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_m)\\ &= \begin{bmatrix} 1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/8 \end{bmatrix}.\end{aligned}$$

Let us apply the weighting matrix:

$$\mathbf{G}_w = \mathbf{W}\mathbf{G} = \begin{bmatrix} 0.1250 & 0.1250 & -0.0625 \\ 0.1250 & 0.2500 & -0.2500 \\ 0.1250 & 0.3750 & -0.5625 \\ 0.1250 & 0.5000 & -1.0000 \\ 0.1250 & 0.6250 & -1.5625 \\ 0.1250 & 0.7500 & -2.2500 \\ 0.1250 & 0.8750 & -3.0625 \\ 0.1250 & 1.0000 & -4.0000 \\ 0.1250 & 1.1250 & -5.0625 \\ 0.1250 & 1.2500 & -6.2500 \end{bmatrix}.$$

Thus, the solution is

$$\mathbf{d}_w = \mathbf{W}\mathbf{d} = \begin{bmatrix} 14.0019 \\ 23.8942 \\ 32.3856 \\ 42.3235 \\ 47.9995 \\ 53.7703 \\ 57.6348 \\ 62.4462 \\ 63.1726 \\ 65.1684 \end{bmatrix}.$$

The maximum-likelihood estimator of model parameters m is found in the following way [3]:

$$\mathbf{G}_w\mathbf{m} = \mathbf{d}_w,$$

where

$$\mathbf{m}_w = (\mathbf{G}_w^{'}\mathbf{G}_w)^{-1}\mathbf{G}_w^{'}\mathbf{d}_w = \mathbf{m}_{L_2} = \begin{bmatrix} 18.4960\, m \\ 96.3291\, m/s \\ 9.2503\, m/s^2 \end{bmatrix}.$$

In Figure 4.1, the data fit of the estimated model could be plotted by the following MATLAB code:

```
% time state vector: 0,0.05,0,10,...,10.90,10.95,11
time=min(t)-1:0.05:max(t)+1
%compute L2 model predictions
model_L2=m_L2(1)+m_L2(2)*time-0.5*m_L2(3)*time.^2
% plot L2 model predictions
plot(time,model_L2,'-');
%plot [t-sigma,t+sigma] and [y-sigma,y+sigma] error bars
%in same figure
hold on;errorbar(t,y,sigma,'+');
```

We have just fitted the model parameters but, we need to test the data fit against uncertainty and estimate error propagation on the model parameters.

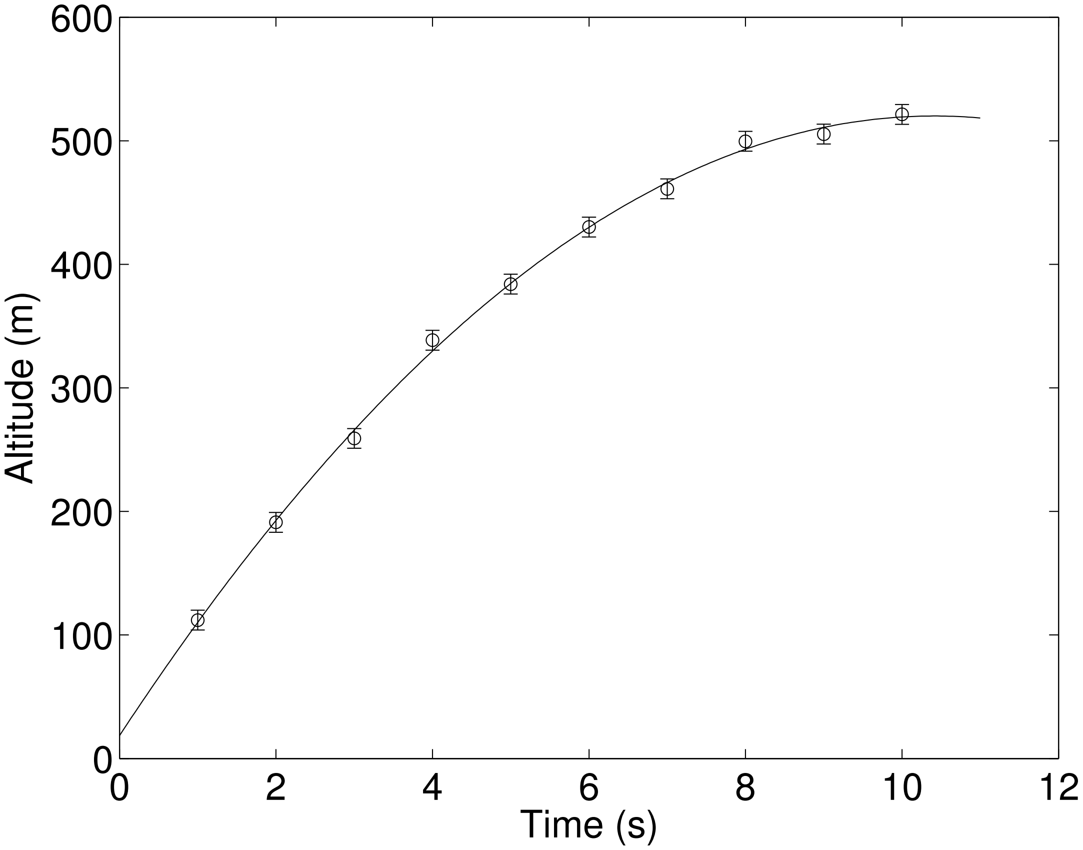

Figure 4.1: Data and L_2 Model Predictions.

A data point that is very much bigger or smaller than the next nearest data point is called an outlier [3]. Let us assume that $d(5) = 683.9957$ is an outlier. Then, the solution turns into:

$$\mathbf{m}_{L_2} = \begin{bmatrix} -41.5040\, m \\ 144.5110\, m/s \\ 18.3412\, m/s^2 \end{bmatrix}.$$

Here, L_2 solution describe a misfit for the set of observables. See Figure 4.2.

Therefore, we define instead the 1-norm:

$$\|\mathbf{y} - \mathbf{Gm}\|_1 = \sum_{i=1}^{m} \|y_i - (\mathbf{Gm})_i\|.$$

The 1-norm solution can be approximated by the MATLAB code below; it is also called *iteratively reweighted least squares strategy* [3].

```
% the size of the system
[m,n]=size(G_w);
tolerance=1.0e-10;
maxiteration=50;
%identity matrix with dimension m
I=eye(m);
%initial m is least square solution
x=G_w\y_w;
%iterate
for i=1:maxiteration
%find residual
 d=G_w*x-y_w;
 for i=1:m
  if (abs(d(i)) < tolerance)
   d(i)=abs(tolerance)^(-1);
  else
   d(i)=abs(d(i))^(-1);
  end
 end
 I=diag(d);
 m_L1=(G_w'*I*G_w)\(G_w'*I*y_w);
 if (norm(m_L1-x)/(1+norm(x)) < tolerance)
 %stop when less than tolerance
  break;
 else
 %solution
  x=m_L1;
 end
end
```

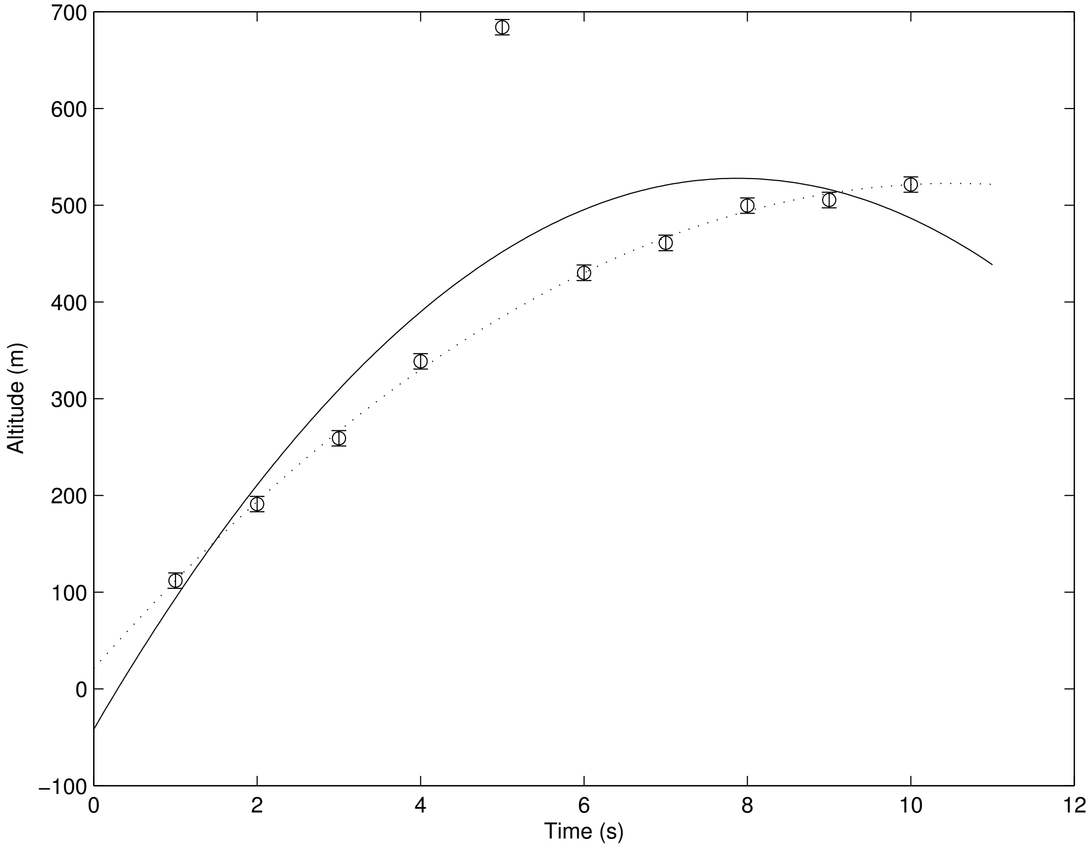

Figure 4.2: L_2 (straight) and L_1 (dashed) Model Predictions with an outlier.

Then, the result is

$$\mathbf{m}_{L_1} = \begin{bmatrix} 21.1601\, m \\ 95.3896\, m/s \\ 9.0742\, m/s^2 \end{bmatrix}.$$

In Figure 4.2, data fit of the L_1 and skewed L_2 models could be plotted similarly by the following MATLAB code:

```
% time state vector: 0,0.05,0,10,...,10.90,10.95,11
time=min(t)-1:0.05:max(t)+1
%compute L1 model predictions
model_L1=m_L1(1)+m_L1(2)*time-0.5*m_L1(3)*time.^2
% plot L1,L2 model predictions
plot(time,model_L1,'k-',time,model_L2,'k:');
%plot [t-sigma,t+sigma] and [y-sigma,y+sigma] error bars
%in same figure
hold on;errorbar(t,y,sigma,'o');
```

Herewith, the 1-norm technique can be more suitable when there are outliers in data but, the calculations are more complex.

We again revert back to $d(5) = 683.9957$. The covariance matrix for fitted parameters shows the dependency between the parameters:

$$\begin{aligned}\mathrm{Cov}(\mathbf{m}_{L_2}) &= (\mathbf{G}_w^{'}\mathbf{G}_w)^{-1}\mathbf{G}_w^{'}((\mathbf{G}_w^{'}\mathbf{G}_w)^{-1}\mathbf{G}_w^{'})^{'} \\ &= \begin{bmatrix} 88.5333 & -33.6000 & -5.3333 \\ -33.6000 & 15.4424 & 2.6667 \\ -5.3333 & 2.6667 & 0.4848 \end{bmatrix}.\end{aligned}$$

Monte-Carlo error propagation is an uncomplicated practice to estimate the covariance matrix of solutions in noisy problems [3]. In this technique, L_1 approximated solution is taken as baseline data for reproducing data errors conjunction with model errors.

$$\mathbf{G}\mathbf{m}_{L_1} = \mathbf{d}_b.$$

Then, the L_1 approximated solution is solved again by the IRLS for q times :

$$\mathbf{G}\mathbf{m}_{L_1,i} = \mathbf{d}_b + \boldsymbol{\eta}_i,$$

where η represents noise and $i = 1, \dots, q$.

Let $A \in \mathbb{R}^{q\times m}$ be defined by $\overline{\mathbf{m}} = \text{mean}(\mathbf{m})$ and with the following row vectors:

$$\mathbf{A}_{i,\cdot} = \mathbf{m}_{L_{1,i}}^{'} - \overline{\mathbf{m}}_{L_{1,\cdot}}^{'},$$

where $i = 1, \dots, q$.

Then, approximated covariance matrix from trial data is defined by:

$$\mathrm{Cov}(\mathbf{m}_{L_1}) := \frac{\mathbf{A}^{'}\mathbf{A}}{q}.$$

We can implement the above procedure in MATLAB:

```
iteration_mc=50000;
%initial y is L1 estimate
y_0 = G*m_L1;
%loop for solving generated data
for i = 1:mc_iteration
 %data generation
 y = y_0+sgm.*randn(N,1);
 %weighting
 y_w=y./sgm;
 %IRLS tecnique mentioned above
 m_mc(i,:)=irls(G_w,y)';
end
%average model
m_avg=[mean(m(:,1))*ones(iteration_mc,1),mean(m(:,2))...
...*ones(iteration_mc,1),mean(m(:,3))*ones(iteration_mc,1)];
%subtract average from irls estimation
cov_L1_est=m_mc-m_avg;
%estimated covariance matrix
covm_L1_est=(covm_L1_est'*covm_L1_est)/mc_iteration;
```

$$\mathrm{Cov}(\mathbf{m}_{L_1}) = \begin{bmatrix} 123.9780 & -46.9560 & -7.4345 \\ -46.9560 & 21.6892 & 3.7473 \\ -7.4345 & 3.7473 & 0.6824 \end{bmatrix}.$$

Then, plot the calculated model points. Figure 4.3 shows the Monte-Carlo error projections.

```
%plot m_1-m_2 estimated data points
plot(m_mc(:,1),m_mc(:,2),'k*')
xlabel('m_1 (m)')
ylabel('m_2 (m/s)')
subplot(1,3,2)
%plot m_1-m_3 estimated data points
plot(m_mc(:,1),m_mc(:,3),'k*')
xlabel('m_1 (m)')
ylabel('m_3 (m/s^2)')
subplot(1,3,3)
%plot m_2-m_3 estimated data points
plot(m_mc(:,2),m_mc(:,3),'k*')
xlabel('m_2 (m/s)')
ylabel('m_3 (m/^2)')
```

The 1.96-sigma confidence interval is a tool to find a range which includes the true parameter with 95% probability.

The 95% parameter confidence intervals $(m-, m_{est}, m+)$ are calculated below, where we used symbolic ± notation to segment internal vectors:

$$\begin{aligned}(m-, m_{est}, m+) &= \mathbf{m}_{L_2} \pm 1.96 \cdot \mathrm{diag}(\mathrm{Cov}(\mathbf{m}_{L_2}))^{1/2} \\ &= [18.4960 \pm 18.4421\, m, 96.3291 \pm 7.7022\, m/s, 9.2503 \pm 1.3648\, m/s^2]' \\ &= \begin{bmatrix} 2.8555 & 18.4960 & 39.7396 \\ 87.3721 & 96.3291 & 102.7764 \\ 7.6491 & 9.2503 & 10.3786 \end{bmatrix}.\end{aligned}$$

We use the chi-square value(and distribution) of sums-of-squares of errors to test the quality of data [3]. There are 3 parameters to estimate, we have 7 (=10-3) degrees of freedom. Chi-square misfit for 7 parameters and random variable d has a chi-square distribution with 7 degrees of freedom. So we calculate the chi-square value:

$$\chi^2 = \|\mathbf{d}_w - \mathbf{G}_w\mathbf{m}_w\|_2^2 = \sum_{i=1}^{m}(d_i - (\mathbf{G}\mathbf{m}_w)_i)^2\sigma_i^2 = 3.5966.$$

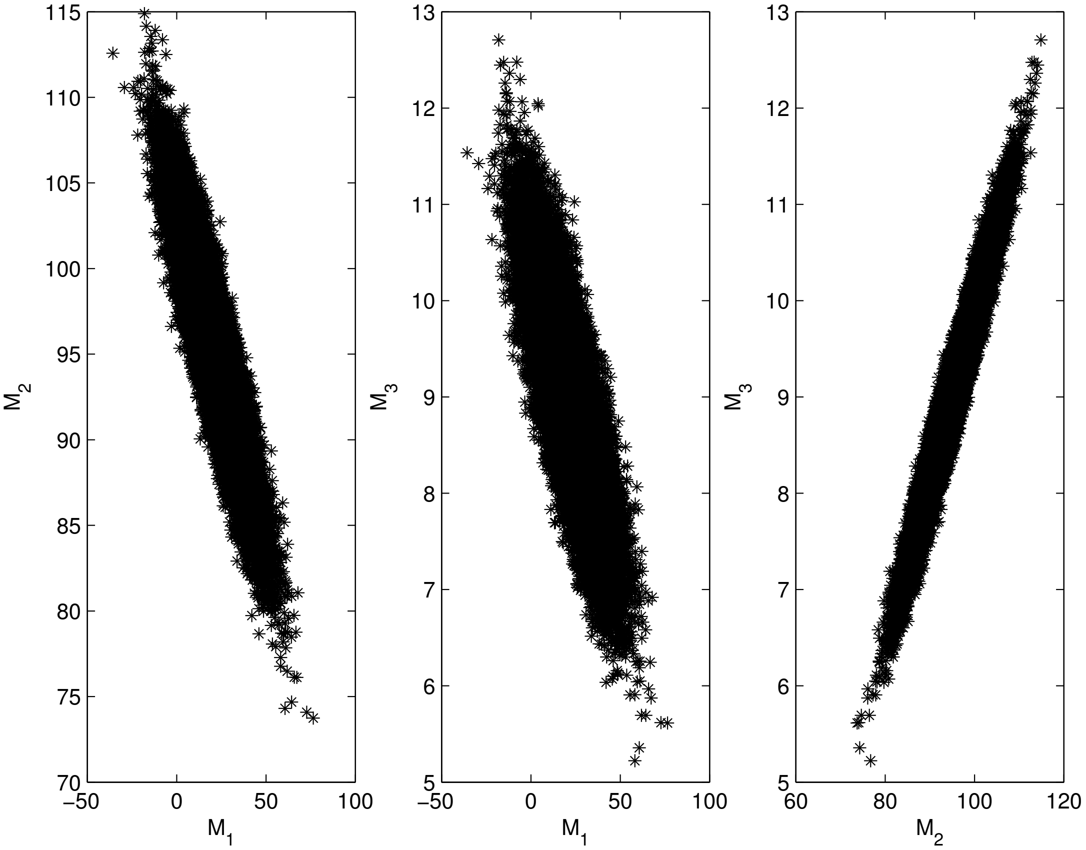

Figure 4.3: Monte-Carlo error projections.

The chi-square p-value for this data set is

$$p = \int_{\chi^2}^{\infty} \frac{1}{2^{7/2}\Gamma(7/2)} x^{\frac{5}{2}} e^{-\frac{x}{2}} dx \approx 0.8249.$$

The p-value is within close range of 1 so the data is very likely to be the real physics. Spread of the expected model parameters obtained by using covariance matrix is given symbolically by

$$\mathrm{Var}(\mathbf{m}_w) = \sqrt{\mathrm{diag}(\mathrm{Cov}(\mathbf{m}_w))} = \begin{bmatrix} 9.4092 & 0 & 0 \\ 0 & 3.9297 & 0 \\ 0 & 0 & 0.6963 \end{bmatrix}.$$

The matrix of parameter correlations obtained by scaling the covariances with variances is

$$\mathrm{Cor}(\mathbf{m}_w) = \frac{\mathrm{Cov}(\mathbf{m}_w)}{\sqrt{\mathrm{Var}(\mathbf{m}_w)\mathrm{Var}(\mathbf{m}_w)'}} = \begin{bmatrix} 1.0000 & -0.9087 & -0.8140 \\ -0.9087 & 1.0000 & 0.9746 \\ -0.8140 & 0.9746 & 1.0000 \end{bmatrix}.$$

The pairwise dependency between $m_1,\ m_2$ and m_3 is strong, and the error ellipsoid is thus inclined and eccentric.

Let us diagonalize the covariance model matrix to find eigenvalues and eigenvectors:

$$\mathrm{Cov}(\mathbf{m}_w)^{-1} = \mathbf{P}^{-1}\mathbf{\Lambda}\mathbf{P}.$$

We note that the eigenvalues correspond to the ellipsoid semiaxis lengths are:

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) \approx \begin{bmatrix} 0.0095 & 0 & 0 \\ 0 & 2.4718 & 0 \\ 0 & 0 & 101.9793 \end{bmatrix}.$$

Orthonormal eigenvectors show the directions of the error ellipsoid principal axes where $\mathbf{P}$ is

$$\mathbf{P} \approx \begin{bmatrix} -0.0299 & 0.3663 & -0.9300 \\ -0.2330 & 0.9022 & 0.3629 \\ 0.9720 & 0.2275 & 0.0584 \end{bmatrix}.$$

The 95% confidence ellipsoid semiaxis lengths are noted in

$$\sqrt{\mathbf{F}^{-1}_{\chi^2,3}(0.95)}[1/\sqrt{\lambda_1}, 1/\sqrt{\lambda_2}, 1/\sqrt{\lambda_3}] \approx \left[28.6062, 1.7781, 0.2768\right],$$

so that the 95% confidence ellipsoid semiaxes are comprised of

$$= \begin{bmatrix} -0.8547 & 0.6514 & -0.2574 \\ -6.6653 & 1.6042 & 0.1005 \\ 27.8057 & 0.4046 & 0.0162 \end{bmatrix}.$$

The 95% confidence ellipsoid with n-dimension:

$$(\mathbf{m} - \mathbf{m}_{L_2})'\mathbf{C}^{-1}(\mathbf{m} - \mathbf{m}_{L_2}) \leq \Delta^2,$$

where $\mathbf{C}$ is the covariance matrix and $\Delta = \sqrt{\mathbf{F}^{-1}_{\chi^2,2}(0.95)}$ mathrmVarpeip.

In ellipsoidal toolbox, the solutions of the equation:

$$\mathbf{E}(\mathbf{c}, \mathbf{C}) = \{\mathbf{x} :< (\mathbf{x} - \mathbf{c}), \mathbf{C}^{-1}(\mathbf{x} - \mathbf{c}) >\leq 1\},$$

defines an ellipsoid with vectors $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$, in fact, with center $\mathbf{c}$ and symmetric positive semidefinite shape matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$ [15]. Here, $< \cdot, \cdot >$ stands for scalar product.

The uncertainty in the model parameters becomes not easy to analyse or understand when all parameters mash-up. We can consider just two parameters at a time more practically. Projection of pairwise confidence ellipsoids onto the coordinate axes can be provided by the following MATLAB code with the help of Ellipsoidal Toolbox (ET) is shown in Figure 4.4.

```
%chi2 reduced to number of parameters
delta=sqrt(chi2inv(0.95,2));
%extract 1st and 2nd row/columns
C=covm((1:2),(1:2));
%find corresponding eigenvalues/eigenvectors
[u,lam]=eig(inv(C));
subplot(1,3,1)
%Create and plot m_1-m_2 ellipsoid
E=ellipsoid([m(1);m(2)],delta^2*C)
hold on;
option.fill=1;
plot(E,'k',options)
axis([-50 50 85 110]);
xlabel('m_1 (m)');
ylabel('m_2 (m/s)');
%extract 1st and 3rd row/columns
C=covm([1,3],[1,3]);
%find corresponding eigenvalues/eigenvectors
[u,lam]=eig(inv(C));
subplot(1,3,2)
%Create and plot m_1- m_3 ellipsoid
E=ellipsoid([m(1);m(3)],delta^2*C)
hold on;
option.fill=1;
plot(E,'k',options)
axis([-50 50 7 12]);
xlabel('m_1 (m)');
ylabel('m_3 (m/s^2)');
%extract 2nd and 3rd row/columns
C=covm([2,3],[2,3]);
%find corresponding eigenvalues/eigenvectors
```

```
[u,lam]=eig(inv(C));
subplot(1,3,3)
%Create and plot m_2-m_3 ellipsoid
E=ellipsoid([m(2);m(3)],delta^2*C)
hold on;
option.fill=1;
plot(E,'k',options)
axis([80 120 7 12]);
xlabel('m_2 (m/s)');
ylabel('m_3 (m/s^2)');
```

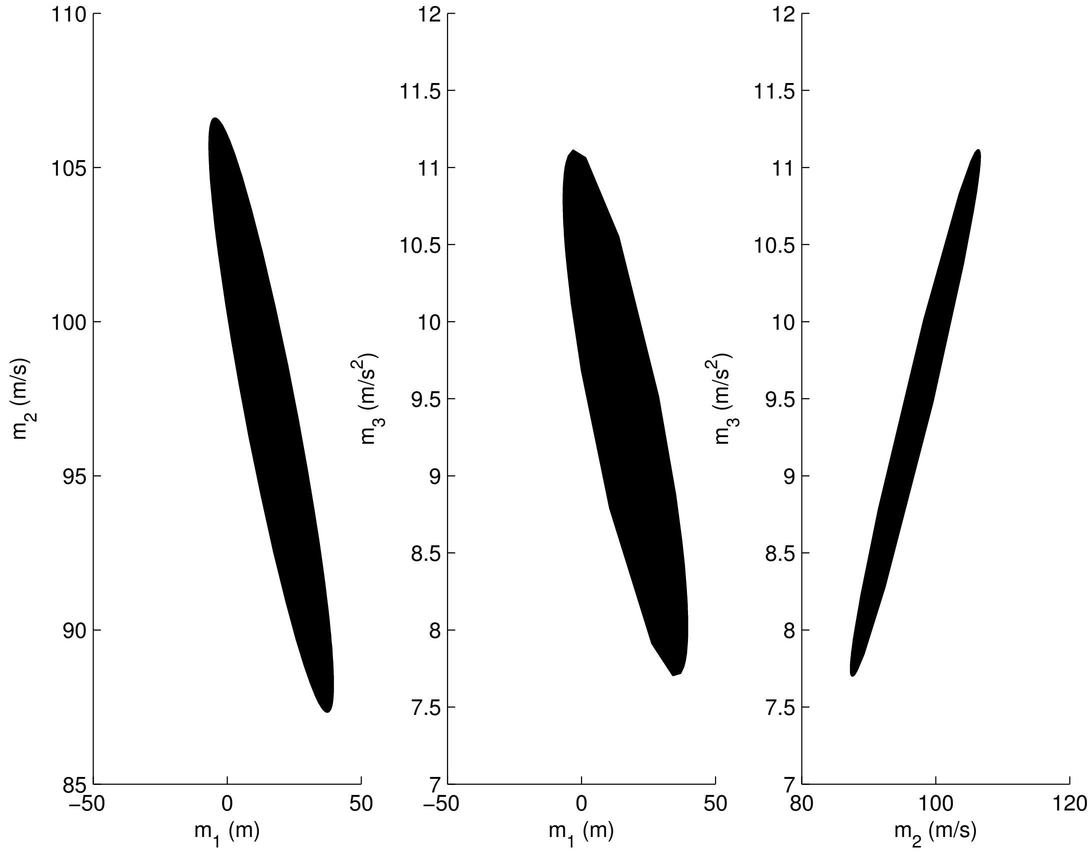

Figure 4.4: 95% Confidence Ellipse Projections.

We obtain 95% confidence intervals which are larger than the 1.96-sigma levels:

$$[m_1, m_2, m_3] = [16.42 \pm 23.03\ m, 96.97 \pm 9.62\ m/s, 9.41 \pm 1.70\ m/s^2].$$

Note that the 95% confidence 3-D ellipsoid simulation in the (m_1, m_2, m_3) coordinate system in Figure 4.5.

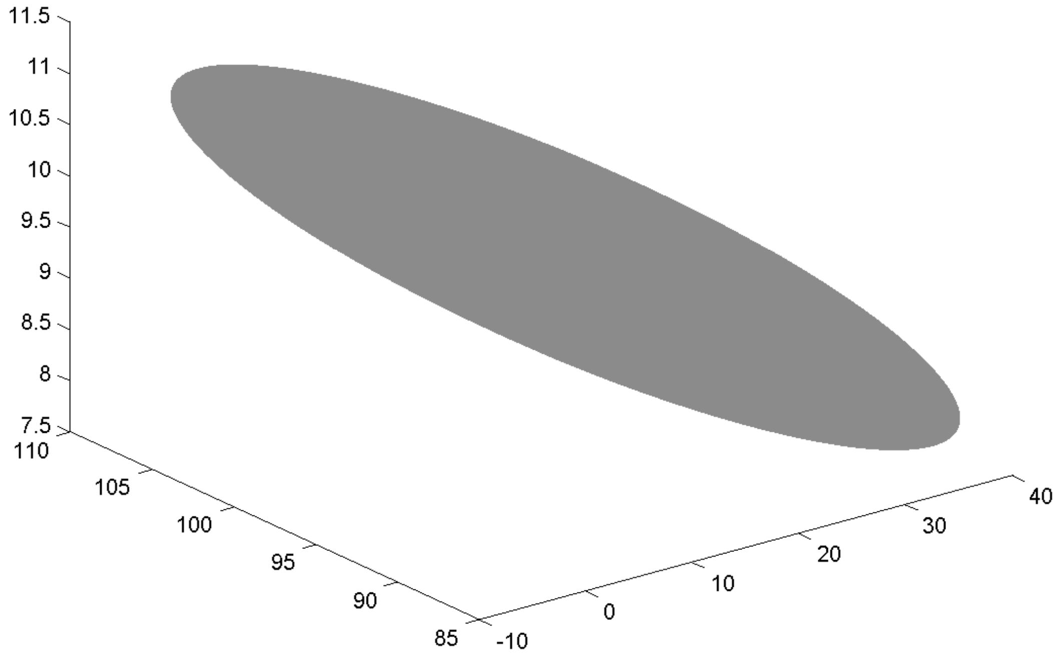

Figure 4.5: 95% Confidence Ellipsoid.

CHAPTER 5

CONCLUSION AND OUTLOOK

Ellipsoids are a more effective representation tool for uncertain states of overlapping and disjoint clusters than interval arithmetics. It is like an additional information layer onto networks when weighting clusters. This layer includes various properties of ellipsoids associated with clusters. After a preprocessing step of clustering and classification, functionally related groups of data are identified. Corresponding cluster parameters determine the dynamics of target and environmental factors as well as their effects on each other. Time-discrete models estimate effects of single variables and their joint effects on clusters. Interior point methods and semidefinite programming are used for solving.

In this study, we aim to summarize the literature in the ellipsoidal optimization field. Firstly, we define suitable systems that produce ellipsoidal uncertainty under some conditions. Then, we introduce optimization methods, especially, regression models, available for those systems. In Chapter 4, we give some basic background information in inverse problems and try to solve an inverse problem step by step using a regression model. Also, we use ellipsoidal toolbox for an easy way to plot ellipsoids in 2D and 3D, using covariance matrix as shape matrix and estimated model parameters as center vector.

In future research, network analysis and concepts from discrete mathematics like connectedness, cycles and shortest paths can be investigated. In mixed-integer programming, using penalty function instead of constraints could be done. More suitable optimization methods for regulatory systems in systems biology and life sciences applications with real-world data can be looked into. Collaborative game theory under interval uncertainty [2] can be studied by ellipsoids. Also, advantages of representing uncertainties with polyhedrals instead of ellipsoids may be investigated.

REFERENCES

[1] D. Akume and G. Weber, Cluster algorithms: theory and methods, Journal of Computational Technologies, 7(1), pp. 15–27, 2002.

[2] S. Alparslan Gök and G.-W. Weber, Cooperative games under ellipsoidal uncertainty, Proceedings of PCO 2010, 3rd Global Conference on Power Control and Optimization, 2010, pp. 2–4.

[3] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems (Second Edition)*, Academic Press, Boston, second edition, 2013, ISBN 978-0-12-385048-5.

[4] A. Ben-Tal, Conic and robust optimization, Lecture Notes, 2002.

[5] A. Ben-Tal and A. Nemirovski, Lectures on modern convex optimization. analysis, algorithms, and engineering applications, Lectures on Modern Convex Optimization, 2001.

[6] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, Wiley, New York, 1999.

[7] E. Kropat, S. Pickl, A. Rössler, and G.-W. Weber, On theoretical and practical relations between discrete optimization and nonlinear optimization, Colloquy Optimization Structure and Stability of Dynamical Systems (at the occasion of the colloquy with the same name, Cologne, October 2000) of Journal of Computational Technologies, 7, pp. 27–62, 2002.

[8] E. Kropat and G.-W. Weber, Robust regression analysis for gene-environment and eco-finance networks under polyhedral and ellipsoidal uncertainty, Preprint 2010-24, Institute of Applied Mathematics, Middle East Technical University, 06800 Ankara, Turkey, November 2010, preprint No. 184.

[9] E. Kropat, G.-W. Weber, and B. Akteke-Öztürk, Eco-finance networks under uncertainty, Proceedings of the International Conference on Engineering Optimization (ISBN 978857650156-5, CD), Engopt, 2008.

[10] E. Kropat, G.-W. Weber, S. Z. Alparslan-Gök, and A. Özmen, Inverse problems in complex multi-modal regulatory networks based on uncertain clustered data, in A. A. Pinto and D. Zilberman, editors, *Modeling, Dynamics, Optimization and Bioeconomics I*, volume 73 of *Springer Proceedings in Mathematics Statistics*, pp. 437–451, Springer International Publishing, 2014, ISBN 978-3-319-04848-2.

[11] E. Kropat, G.-W. Weber, and S. Belen, Dynamical gene-environment networks under ellipsoidal uncertainty: Set-theoretic regression analysis based on ellipsoidal or, in M. M. Peixoto, A. A. Pinto, and D. A. Rand, editors, *Dynamics, Games and Science I*, volume 1 of *Springer Proceedings in Mathematics*, pp. 545–571, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-11455-7.

[12] E. Kropat, G.-W. Weber, and C. Pedamallu, Regulatory networks under ellipsoidal uncertainty – data analysis and prediction by optimization theory and dynamical systems, in D. Holmes and L. Jain, editors, *Data Mining: Foundations and Intelligent Paradigms*, volume 24 of *Intelligent Systems Reference Library*, pp. 27–56, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-23240-4.

[13] E. Kropat, G.-W. Weber, and J.-J. Rückmann, Regression analysis for clusters in gene-environment networks based on ellipsoidal calculus and optimization, volume 17, pp. 639–657, 2010.

[14] A. Kurzhanski and I. Valyi, Ellipsoidal Calculus for Estimation and Control, 1997.

[15] A. A. Kurzhanskiy and P. Varaiya, Ellipsoidal toolbox, Technical Report UCB/EECS-2006-46, EECS Department, University of California, Berkeley, May 2006.

[16] A. Nemirovski, Modern Convex Optimization, Lecture at PASCAL Workshop, 2005.

[17] A. Shapiro, D. Dentcheva, and A. Ruszczynski, Lectures on Stochastic Programming: Modeling and Theory, 2009.

APPENDIX A

Ellipsoidal Calculus

Calculation of approximation at each time step with ellipsoidal calculus can be used in discrete-time systems with singular state *tansition matrices*. Lets review the basic ellipsoidal operations to deal with ellipsoidal uncertainty [15].

A.1 Ellipsoidal Descriptions

$\mathcal{E}(c, \Sigma) \in \mathbb{R}^p$ is an *ellipsoid* as

$$\mathcal{E}(c, \Sigma) = \left\{\Sigma^{1/2}u + c \mid \|u\|_2 \leq 1\right\},$$

wherein $c \in \mathbb{R}^p$ is its center and $\Sigma \in \mathbb{R}^{p\times p}$ is its positive definite *shape matrix*. When $\Sigma^{1/2}$ is positive semi-definite, we get the non-degenerate ellipsoid $\mathcal{E}(c, \Sigma)$

$$\mathcal{E}(c, \Sigma) = \left\{x \in \mathbb{R}^p \mid (x - c)'\Sigma^{-1}(x - c) \leq 1\right\}.$$

where corresponding eigenvector are zero.

Affine transformations are given by

$$A\mathcal{E}(c, \Sigma) + b = \mathcal{E}\left(Ac + b, A\Sigma A'\right),$$

where $\mathcal{E}(c, \Sigma) \subset \mathbb{R}^p$, matrix $A \in \mathbb{R}^{m\times p}$ and vector $b \in \mathbb{R}^m$.

The family of ellipsoids is closed with respect to *affine transformations*. Thus, ellipsoids are preserved under affine transformation. If the rows of A are linearly independent (which implies $m \leq p$), and $b = 0$, the affine transformation is called a *projection*.

Geometric sum of ellipsoids as

$$\mathcal{E}_1 + \ldots + \mathcal{E}_k = \{z_1 + \ldots + z_k \mid z_k \in \mathcal{E}_k\},$$

where $\mathcal{E}_k = \mathcal{E}(c_k, \Sigma_k),\ k = 1, \ldots, K$, are nondegenerate ellipsoids. The sum generally is not an ellipsoid. However, it can be tightly approximated by parametrized families of external ellipsoids.

Geometric difference of ellipsoids as

$$\mathcal{E}_1 - \mathcal{E}_2,$$

where $\mathcal{E}_1 = \mathcal{E}(c_1, \Sigma_1),\ \mathcal{E}_2 = \mathcal{E}(c_2, \Sigma_2)$, are nondegenerate ellipsoids.

Minimal trace ellipsoid could be denoted as

$$\sigma = \sum_{k=1}^{K} c_k$$

and

$$P = \left(\sum_{k=1}^{K} \sqrt{\mathrm{Tr}\Sigma_k} \right) \left(\sum_{k=1}^{K} \frac{\Sigma_k}{\sqrt{\mathrm{Tr}\Sigma_k}} \right),$$

where $\mathcal{E}(\sigma, P) = \bigoplus_{k=1}^{K} \mathcal{E}_k$ is outer ellipsoidal approximation and $S = \sum_{k=1}^{K} \mathcal{E}_k$ is containing the sum ellipsoids [15].

Let us assume that

$$\mathcal{X} := \lambda\Sigma_1^{-1} + (1-\lambda)\Sigma_2^{-1}$$

and

$$\tau := 1 - \lambda(1-\lambda)(c_2 - c_1)'\Sigma_2^{-1}\mathcal{X}^{-1}\Sigma_1^{-1}(c_2 - c_1).$$

Fusion of ellipsoids is defined by

$$\begin{aligned} \mathcal{E}_\lambda(c_0, \Sigma_0) := \{ x \in \mathbb{R}^p \mid\ & \lambda(x - c_1)'\Sigma_1^{-1}(x - c_1) \\ & +(1-\lambda)(x - c_2)'\Sigma_2^{-1}(x - c_2) \leq 1 \}, \end{aligned}$$

where $\lambda \in [0,1],\ \mathcal{E}(c_1, \Sigma_1),\ \mathcal{E}(c_2, \Sigma_2) \subset \mathbb{R}^p$ are non-degenerate ellipsoids when $\mathcal{E}(c_1, \Sigma_1) \cap \mathcal{E}(c_2, \Sigma_2) \neq \emptyset$. The ellipsoid $\mathcal{E}_\lambda(c_0,\ \Sigma_0)$ is given by the center

$$c_0 = \mathcal{X}^{-1}\left(\lambda\Sigma_1^{-1}c_1 + (1-\lambda)\,\Sigma_2^{-1}c_2\right)$$

and the shape matrix

$$\Sigma_0 = \tau\mathcal{X}^{-1}.$$

We introduce two new terms:

$$\mathcal{X} := \lambda\Sigma_1^{-1} + (1-\lambda)\Sigma_2^{-1},$$

and

$$\tau := 1 - \lambda(1-\lambda)(c_2 - c_1)'\Sigma_2^{-1}\mathcal{X}^{-1}\Sigma_1^{-1}(c_2 - c_1).$$

If $\mathcal{E}(c_1, \Sigma_1) \nsubseteq \mathcal{E}(c_2, \Sigma_2)$ or $\mathcal{E}(c_2, \Sigma_2) \nsubseteq \mathcal{E}(c_1, \Sigma_1)$, then approximated outbound ellipsoidal $\mathcal{E}_\lambda(c_0, \Sigma_0)$ could be defined as

$$\begin{aligned} & \tau\,(\det\mathcal{X})\,\mathrm{Tr}\left(\mathrm{co}\,(\mathcal{X})\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right)\right) - p\,(\det\mathcal{X})^2 \\ & \times \left(2c_0'\Sigma_1^{-1}c_1 - 2c_0'\Sigma_2^{-1}c_2 + c_0'\left(\Sigma_2^{-1} - \Sigma_1^{-1}\right)c_0 - c_1'\Sigma_1^{-1}c_1 + c_2'\Sigma_2^{-1}c_2\right) = 0, \end{aligned}$$

where λ is the only root in $(0,1)$ of the following polynomial of degree $2p-1$ and $\mathrm{co}(\mathcal{X})$ denotes the matrix of cofactors of $\mathcal{X}$.

In fact, let us substitute $\mathcal{X}^{-1} = \text{co}\,(\mathcal{X}) / \det \mathcal{X}$ in this polynomial:

$$\begin{aligned} &\tau\,(\det \mathcal{X})^2 \operatorname{Tr}\left(\mathcal{X}^{-1}\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right)\right) - p\,(\det \mathcal{X})^2 \\ &\times \left(2c_0'\Sigma_1^{-1}c_1 - 2c_0'\Sigma_2^{-1}c_2 + c_0'\left(\Sigma_2^{-1} - \Sigma_1^{-1}\right)c_0 - c_1'\Sigma_1^{-1}c_1 + c_2'\Sigma_2^{-1}c_2\right) = 0. \end{aligned}$$

The input is given in following terms:

$$\overline{X}_r^{(\kappa)} = \mathcal{E}\left(\overline{\mu}_r^{(\kappa)}, \overline{\Sigma}_r^{(\kappa)}\right) \subset \mathbb{R}^{|C_r|}, \overline{E}_s^{(\kappa)} \in \mathcal{E}\left(\overline{\rho}_s^{(\kappa)}, \overline{\Pi}_s^{(\kappa)}\right) \subset \mathbb{R}^{|D_s|}.$$

The approximated fusions $\Delta X_r^{(\kappa)} = \widehat{X}_r^{(\kappa)} \cap \overline{X}_{C_r}^{(\kappa)}$ is an ellipsoid $\mathcal{E}\left(\Delta\mu_r^{(\kappa)},\ \Delta\Sigma_r^{(\kappa)}\right)$ with center

$$\Delta\mu_r^{(\kappa)} := \left[\mathcal{X}_r^{(\kappa)}\right]^{-1}\left(\lambda\left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1}\widehat{\mu}_r^{(\kappa)} + (1-\lambda)\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}\overline{\mu}_r^{(\kappa)}\right)$$

and shape matrix

$$\Delta\Sigma_r^{(\kappa)} = \xi_r^{(\kappa)}\left[\mathcal{X}_r^{(\kappa)}\right]^{-1},$$

where

$$\mathcal{X}_r^{(\kappa)} := \lambda\left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1} + (1-\lambda)\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}$$

and

$$\xi_r^{(\kappa)} := 1 - \lambda\,(1-\lambda)\left(\overline{\mu}_r^{(\kappa)} - \widehat{\mu}_r^{(\kappa)}\right)'\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}\left[\mathcal{X}_r^{(\kappa)}\right]^{-1}\left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1}\left(\overline{\mu}_r^{(\kappa)} - \widehat{\mu}_r^{(\kappa)}\right).$$

The parameter λ is the only root in of the following polynomial of degree $2|C_r| - 1$:

$$\begin{aligned} &\xi_r^{(\kappa)}\left(\det \mathcal{X}_r^{(\kappa)}\right)^2 \operatorname{Tr}\left(\left[\mathcal{X}_r^{(\kappa)}\right]^{-1}\left(\left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1} - \left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}\right)\right) - |C_r|\left(\det \mathcal{X}_r^{(\kappa)}\right)^2 \\ &\times \Bigg(2\left[\Delta\mu_r^{(\kappa)}\right]'\left[\Sigma_r^{(\kappa)}\right]^{-1}\widehat{\mu}_r^{(\kappa)} - 2\left[\Delta\mu_r^{(\kappa)}\right]'\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}\overline{\mu}_r^{(\kappa)} \\ &+ \left[\Delta\mu_r^{(\kappa)}\right]'\left(\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1} - \left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1}\right)\Delta\mu_r^{(\kappa)} - \left[\widehat{\mu}_r\right]\left[\widehat{\Sigma}_r^{(\kappa)}\right]^{-1}\widehat{\mu}_r^{(\kappa)} \\ &+ \left[\overline{\mu}_r^{(\kappa)}\right]'\left[\overline{\Sigma}_r^{(\kappa)}\right]^{-1}\overline{\mu}_r^{(\kappa)}\Bigg) = 0. \end{aligned}$$

Similarly, the predicted fusion $\Delta E_s^{(\kappa)} := \widehat{E}_s^{(\kappa)} \cap \overline{E}_s^{(\kappa)}$ is an ellipsoid $\mathcal{E}(\Delta p_s^{(\kappa)}, \Delta\Pi_s^{(\kappa)})$ with center

$$\Delta\rho_s^{(\kappa)} = \left[\mathcal{Y}_s^{(\kappa)}\right]^{-1}\left(\lambda\left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1}\widehat{\rho}_s^{(\kappa)} + (1-\lambda)\left[\overline{\Pi}_s^{(\kappa)}\right]^{-1}\overline{\rho}_s^{(\kappa)}\right)$$

and shape matrix

$$\Delta\Pi_s^{(\kappa)} = \eta_s^{(\kappa)}\left[\mathcal{Y}_s^{(\kappa)}\right]^{-1},$$

where

$$\mathcal{Y}_s^{(\kappa)} := \lambda\left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1} + (1-\lambda)\left[\overline{\Pi}_s^{(\kappa)}\right]^{-1}$$

and

$$\eta_s^{(\kappa)} := 1 - \lambda(1-\lambda)\left(\overline{p}_s^{(\kappa)} - \widehat{p}_s^{(\kappa)}\right)' \left[\overline{\Pi}_s^{(\kappa)}\right]^{-1} \left[\mathcal{Y}_s^{(\kappa)}\right]^{-1} \left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1} \left(\overline{p}_s^{(\kappa)} - \widehat{p}_s^{(\kappa)}\right).$$

Also, λ is the only root of the following polynomial of degree $2|D_s| - 1$:

$$\begin{aligned}
&\eta_s^{(\kappa)} \left(\det \mathcal{Y}_s^{(\kappa)}\right)^2 \operatorname{Tr}\left(\left[\mathcal{Y}_s^{(\kappa)}\right]^{-1}\left(\left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1} - \left[\overline{\Pi}_s^{(\kappa)}\right]^{-1}\right)\right) - |D_s| \left(\det \mathcal{Y}_s^{(\kappa)}\right)^2 \\
&\times \Bigg(2\left[\Delta\rho_s^{(\kappa)}\right]' \left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1} \widehat{\rho}_s^{(\kappa)} - 2\left[\Delta\rho_s^{(\kappa)}\right]' \left[\overline{\Pi}_s^{(\kappa)}\right]^{-1} \overline{\rho}_s^{(\kappa)} \\
&+ \left[\Delta\rho_s^{(\kappa)}\right]' \left(\left[\overline{\Pi}_s^{(\kappa)}\right]^{-1} - \left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1}\right) \Delta\rho_s^{(\kappa)} - \left[\widehat{\rho}_s\right]' \left[\widehat{\Pi}_s^{(\kappa)}\right]^{-1} \widehat{\rho}_s^{(\kappa)} \\
&+ \left[\overline{p}_s^{(\kappa)}\right]' \left[\overline{\Pi}_s^{(\kappa)}\right]^{-1} \overline{p}_s^{(\kappa)}\Bigg) = 0.
\end{aligned}$$

These representations are called *explicit representations of the fusions* [11] and could be used for numerical calculations and an estimation of the parameters in the regression problems (RP).

APPENDIX B

Ellipsoidal Toolbox

Ellipsoidal Toolbox (ET) is a software package that implements the ellipsoidal calculus and its application freely in MATLAB [15].

The ellipsoidal toolbox features:

- The operations of the ellipsoidal calculus: affine transformation, geometric sum, geometric difference, intersections with hyperplane, ellipsoid, halfspace and polytope, calculation of maximum ellipsoid, calculation of minimum ellipsoid.
- Reachability problem and ellipsoidal methods for the reach set approximation.

B.1 Installation

1. Download version 1.1.3 from

   `http://systemanalysisdpt-cmc-msu.github.io/ellipsoids`
2. Extract the downloaded file.
3. Open and run install.m from extracted ellipsoids folder in MATLAB.
4. `File` → `Set Path...` and click `Save`.

B.2 Examples

```
>> q = [2; -1],
>> Q = [9 -5; -5 4];
% nondegenerate ellipsoid in R^2
>> E = ellipsoid(q, Q);
>> E = ellipsoid(Q) + q;
>> E = sqrtm(Q)*ell unitball(size(Q, 1)) + q;
% 2x2 array of ellipsoids
>> EE = [E polar(E); inv(E) [0 1; -2 0] * E + [3; 0]];
% define halfspace x1 + x2 <= 1
```

```
>> H = hyperplane([1; 1], 1);
% define halfspace x1 + x2 >= 1
>> H = hyperplane([-1; -1], -1);
% convert array of hyperplanes to polytope
>> P = hyperplane2polytope(HH);
% covert polytope to array of hyperplanes
>> HP = polytope2hyperplane(P);
>> HP == HH
ans =
1 1 1 1

% Functions hyperplane2polytope and polytope2hyperplane
% require the Multi-Parametric Toolbox to be installed.
```

B.3 Function Reference

B.3.1 ellipsoid Methods

`contains` – checks if one ellipsoid contains the other.

`dimension` - returns the dimension of the space in which the ellipsoid is defined and the rank of its shape matrix.

`display` – displays the details of the ellipsoid object.

`distance` – computes the distance from the given ellipsoid to the specified object – vector, ellipsoid, hyperplane or polytope.

`double` – returns parameters of the ellipsoid, its center and shape matrix.

`ellintersection_ia` – computes maximum volume ellipsoid that is contained in the intersection of given ellipsoids.

`ellipsoid` – constructor for the ellipsoid object.

`ellunion_ea` – computes minimum volume ellipsoid that contains union of given ellipsoids.

`eq` – overloaded operator `'=='`, checks if two ellipsoids are equal.

`ge`, `gt` – checks if the first ellipsoid is bigger than the second one.

`getAbsTol` – gives array the same size as input array with values of `absTol` properties for each ellipsoid in input array.

`getNPlot2dPoints` – gives value of `nPlot2dPoints` property of ellipsoids in input array of ellipsoids.

getNPlot3dPoints – gives value of nPlot3dPoints property of ellipsoids in input array of ellipsoids.

getRelTol – gives array the same size as input array with values of relTol properties for each ellipsoid in input array.

hpintersection – computes the ellipsoid which results from intersection of given ellipsoid with given hyperplane.

intersect – checks if the union or intersection of ellipsoids intersects given ellipsoid, hyperplane or polytope.

intersection_ea – computes the external ellipsoidal approximation of the intersection of the ellipsoid with given ellipsoid, halfspace or polytope.

intersection_ia – computes the internal ellipsoidal approximation of the intersection of the ellipsoid with given ellipsoid, halfspace or polytope.

inv – inverts the shape matrix of the ellipsoid if it is nonsingular.

isbaddirection – checks if ellipsoidal approximations of the geometric difference of two ellipsoids can be computed for given directions.

isbigger – checks if one ellipsoid would contain the other if their centers would coincide.

isdegenerate – checks if given ellipsoid is degenerate.

isempty – checks if given ellipsoid is an empty object.

isinside – checks if the intersection of ellipsoids contains the union or intersection of given ellipsoids or polytopes.

isinternal – checks if the union or intersection of ellipsoids contains given vectors.

le, lt – checks if the second ellipsoid is bigger than the first one.

maxeig – returns the biggest eigenvalue of the ellipsoid.

mineig – returns the smallest eigenvalue of the ellipsoid.

minkdiff – computes and plots the geometric difference of two ellipsoids in 2D and 3D.

minkdiff_ea – computes external ellipsoidal approximations of the geometric difference of two ellipsoids of arbitrary dimension for given directions, if these directions are not bad.

minkdiff_ia – computes internal ellipsoidal approximations of the geometric difference of two ellipsoids of arbitrary dimension for given directions, if these directions are not bad.

`minkmp` – computes and plots geometric
(Minkowski) sum of the geometric difference of two ellipsoids and the geometric sum of n ellipsoids in 2D or 3D.

`minkmp_ea` – computes external ellipsoidal approximations of the geometric (Minkowski) sum of the geometric difference of two ellipsoids and the geometric sum of n ellipsoids.

`minkmp_ia` – computes internal ellipsoidal approximations of the geometric (Minkowski) sum of the geometric difference of two ellipsoids and the geometric sum of n ellipsoids.

`minkpm` – computes and plots geometric (Minkowski) difference of the geometric sum of ellipsoids and a single ellipsoid in 2D or 3D.

`minkpm_ea` – computes external ellipsoidal approximations of the geometric (Minkowski) difference of the geometric sum of ellipsoids and a single ellipsoid.

`minkpm_ia` – computes internal ellipsoidal approximations of the geometric (Minkowski) difference of the geometric sum of ellipsoids and a single ellipsoid.

`minksum` – computes and plots the geometric sum of finite number of ellipsoids in 2D and 3D.

`minksum_ea` – computes external ellipsoidal approximations of the geometric sum of finite number of ellipsoids of arbitrary dimension for given directions.

`minksum_ia` – computes internal ellipsoidal approximations of the geometric sum of finite number of ellipsoids of arbitrary dimension for given directions.

`minus` – overloaded operator `'-'`.

`move2origin` – moves given ellipsoids to the origin.

`mtimes` – overloaded operator `'*'`.

`ne` – overloaded operator `'~='`, checks if two ellipsoids are not equal.

`parameters` – returns parameters of the ellipsoid, its center and shape matrix.

`plot` – plots ellipsoids in 1D, 2D and 3D.

`plus` – overloaded operator `'+'`.

`polar` – computes polars for ellipsoids which contain the origin.

`projection` – computes projection of ellipsoids onto given orthogonal basis.

`rho` – computes the support function of the ellipsoids for given directions and corresponding boundary points.

`shape` – has the same functionality as `mtimes` but modifies only the shape matrix of the ellipsoid leaving its center as is.

`trace` – computes trace of given ellipsoids.

`uminus` – overloaded operation unitary minus.

`volume` – computes volume of given ellipsoids.

B.3.2 hyperplane Methods

`contains` – checks if the hyperplanes contain given vectors.

`dimension` – returns the dimension of the space in which the hyperplane is defined.

`display` – displays the details of the hyperplane object.

`double` – returns parameters of the hyperplane object, its normal and scalar.

`eq` – overloaded operator `'=='`. checks if two hyperplanes are equal.

`getAbsTol` – gives array the same size as input array with values of `absTol` properties for each hyperplane in input array.

`hyperplane` – constructor for the hyperplane object. If called without parameters, returns empty hyperplane.

`isempty` – checks if the hyperplane object is empty.

`isparallel` – checks if the hyperplanes are parallel.

`ne` – overloaded operator `'~='`, checks if two hyperplanes are not equal.

`parameters` – returns parameters of the hyperplane object, its normal and scalar.

`plot` – plots hyperplanes in 2D and 3D.

`uminus` – overloaded operator unitary minus. It does not change the hyperplane, and affects only the halfspace this hyperplane defines.

B.3.3 linsys Methods

`dimension` – returns dimensions of state, input, output and disturbance input spaces.

`display` – displays the details of the linear system object.

`hasdisturbance` – checks if given linear system is the system with disturbance.

`hasnoise` – checks if given linear system has noise at the output.

`isdiscrete` – checks if given linear system is discrete-time.

`isempty` – checks if given linear system is an empty object.

`islti` – checks if given linear system is time invariant.

`linsys` – constructor for the linear system object. If called without parameters, creates an empty object.

B.3.4 reach Methods

`cut` – extracts a segment of the reach tube from the given start time to the given end time.

`dimension` – returns the dimension of the reach set and the dimension of the state space for which the reach set was originally computed.

`display` – displays the details of the reach set object.

`evolve` – computes further evolution in time of the already existing reach set.

`get_center` – returns trajectory of the center of the reach set.

`get_directions` – returns the trajectories of the direction vectors for which the ellipsoidal approximations were computed.

`get_ea` – returns array of ellipsoids that represent the external approximation of the reach set.

`get_goodcurves` – returns the trajectories along which the ellipsoidal approximations are touching the actual reach set.

`get_ia` – returns array of ellipsoids that represent the internal approximation of the reach set.

`get_system` – returns the linear system object for which the reach set was computed.

`intersect` – checks if the external or internal approximation of the reach set intersects with given ellipsoids, hyperplanes or polytopes.

`iscut` – checks if the given reach set object resulted from `cut` operation.

`isempty` – checks if the given reach set object is empty.

`isprojection` – checks if the given reach set is a result of `projection` operation.

`plot_ea` – plots external approximation of the reach set in 2D and 3D.

`plot_ia` – plots internal approximation of the reach set in 2D and 3D.

`projection` – projects the reach set onto the given orthogonal basis.

`reach` – constructor for the reach set object and the main function that computes the reach set.

`refine` – adds new approximations for the specified directions to the given reach set object, thus improving the overall approximation.

B.3.5 Miscellaneous Functions

`ell_simdiag` – computes the orthogonal transformation matrix that simultaneously diagonalizes two symmetric matrices.

`ell_unitball` – creates the ellipsoid object that represents a unit ball of the given dimension.

`ell_valign` – computes the orthogonal matrix that aligns two vectors.

`hyperplane2polytope` – converts array of hyperplanes of the same dimension into the polytope object of the Multi-Parametric Toolbox.

`polytope2hyperplane` – converts the polytope object of the
Multi-Parametric Toolbox into the array of hyperplanes.