Università degli Studi di Pavia
Università degli Studi di Milano Bicocca

Joint PhD program in Mathematics – XXXVII cycle

Mathematical Modelling in Biosciences and Discrete Optimization

Supervisor
Prof. Giuseppe DE NICOLAO

PhD Student
Simone MILANESI

Oct 2021 - Sep 2024

A colei che ha costruito
questi spazi emotivi,
stanze che ora
mi è difficile abitare

Contents

Abstract

This thesis explores innovative methodologies in two distinct areas of Applied Mathematics: Mathematical Modelling in Biosciences and Discrete Optimization. The work is structured into two parts, each addressing critical challenges in its respective field while offering practical applications and advancements.

The first part focuses on *Mathematical Modelling in Biosciences*, presented in two chapters. The first chapter introduces a statistical monitoring approach using funnel plots for the early detection of COVID-19 Variants of Concern (VoCs). This methodology demonstrates remarkable utility in epidemiological surveillance by providing a simple, cost-effective, and real-time tool for identifying anomalous patterns in regional reproduction numbers. Its practical impact lies in its ability to complement genomic sequencing efforts by enabling more targeted and efficient investigations, ultimately supporting timely public health interventions.

The second chapter reconstructs the temporal profile of new COVID-19 cases in Italy during the first wave of 2020, addressing the significant underreporting that hindered accurate epidemiological assessments. By leveraging dynamic system identification and regularized inverse problem-solving techniques, this work not only offers a quantitative correction for underreported data but also provides a robust framework for evaluating the impact of non-pharmaceutical interventions.

The second part focuses on *Discrete Optimization* and consists of two chapters. The first chapter addresses the design and optimization of few-bit neural networks tailored for classification problems under few-shot learning scenarios. A novel voting structure is proposed to extend the framework to multi-class classification, offering practical applications in scenarios where computational efficiency and adaptability are paramount.

The second chapter investigates the unassigned Distance Geometry Problem in the Manhattan norm, applied to the Mobile Positioning Problem in grid-like geometries. This formulation is particularly suited to scenarios such as mobile device positioning in urban environments, where the assignment of distances between devices is unknown.

Research Activities

Awards

- Recipient with the colleague A.M. Bernardelli of a €11k Excellence Scholarship assigned by the EU network TAILOR project aimed at building the capacity to provide the scientific foundations for Trustworthy AI in Europe. The scholarship funded the period spent at TU Delft, April-May 2023.
- First prize winner of the 2020 edition of Premio ETIC for Rotary District 2050 bestowed by AICA and Rotary International for master's thesis with themes relevant to relationships between technology and ethics. The summary of the thesis was published in the 95th edition of the journal *Mondo Digitale*. Prize amount: €2k.

Published papers

(1) *Correction of Italian under-reporting in the first COVID-19 wave via age-specific deconvolution of hospital admissions*
S. Milanesi, and G. De Nicolao.
Plos one 18.12 (2023): e0295079.

(2) *Early detection of variants of concern via funnel plots of regional reproduction numbers*
S. Milanesi, G. De Nicolao, et al.
Scientific reports 13.1 (2023): 1052.

(3) *Multi-Objective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks*
A.M. Bernardelli, S. Milanesi, S. Gualandi, et al.
INFORMS IJOC - Journal of Computing (2024).

Book chapters in press

(4) *The Mobile Positioning Problem as an Unassigned Distance Geometry Problem*
S. Milanesi, L. Liberti, and C. Lavor.
To appear in the Springer Fields Institute Communications series, volume "Geometric Constraint Systems: Problems, Progress, Connections, and Emerging Applications".

Projects in progress

- *Funnel-Surface-based AMR surveillance in the WHO Europe regions*, in collaboration with researchers from the Department of Biomedical Sciences for Health (University of Milan) and the Department of Biomedical and Clinical Sciences (University of Milan and L. Sacco Hospital), we are developing a surveillance system based on Funnel Plots that correlates AMR data reported in WHO European Region countries with their respective Gross Domestic Products. The data is sourced from the WHO GLASS dashboard and World Bank data.
- *Funnel-Plot-based AMR Surveillance in Italy on a regional level*, in collaboration with researchers from the Department of Biomedical Sciences for Health (University of Milan) and the Department of Biomedical and Clinical Sciences (University of Milan and L. Sacco Hospital), we are developing a surveillance system to be integrated with the existing ones from the Italian National Institute of Health (ISS) at both the national and regional levels.

Invited talks in symposia and conferences

- 25th International Symposium on Mathematical Programming (ISMP), Montreal (Canada), 21st July - 26th July, 2024:
  A.M. Bernardelli, S. Gualandi, S. Milanesi (presenter), N. Yorke-Smith, H.C. Lau, *Slimming down and beefing up: MILP Multi-Objective Optimization and Training for Few-Bit Neural Networks.*
- 33rd EUropean conference in Operational Research (EURO), Copenhagen (Denmark), 30st June - 3rd July, 2024:
  A.M. Bernardelli, S. Gualandi, S. Milanesi (presenter), N. Yorke-Smith, H.C. Lau, *Simultaneous Training and Optimization of Few-Bit Neural Networks through a Multi-Objective MILP.*

Talks in conferences

The list contains my works presented at conferences. The talks marked with an * were presented by a co-author.

- 4th Young Applied Mathematicians Conference (YAMC), Rome (Italy), 16th - 20th September 2024, Talk presentation. *A funnel plot approach for monitoring Antibiotic Resistance in EU countries based on the WHO GLASS dashboard.*
- * 32nd Population Approach Group Europe (PAGE) Meeting, Rome (Italy), 26th - 28th June 2024, Poster presentation. *A mixed effect*

multi-funnel model for country surveillance of antibiotic resistance: evidence from the WHO GLASS dashboard.

- * INFORMS 2023 Annual Meeting, Phoenix (USA), 15th - 18th October 2023, Talk Presentation. *The BeMi Stardust: A Structured Ensemble of Binarized Neural Networks.*
- * 17th Learning and Intelligent Optimization (LION), Nice (France), 4th - 8th June 2023, Talk presentation. *The BeMi Stardust: a Structured Ensemble of Binarized Neural Networks.*
- 7th Associazione Italiana Ricerca Operativa (AIRO) Young, Milano (Italy), 15th - 17th February 2023. Talk presentation. *The BeMi Stardust: a Structured Ensemble of Binarized Neural Networks.*
- The mathematics of Machine Learning, Pisa (Italy), 16th - 27th January 2023. Talk presentation. *The BeMi Stardust: a Structured Ensemble of Binarized Neural Networks.*
- Matematica per l'intelligenza artificiale e il Machine Learning: Giovani ricercatori, Torino (Italy), 24th - 25th November 2022. Talk presentation. *A MILP approach to a structured ensemble Binarized Neural Network.*
- 2nd Young Applied Mathematicians Conference (YAMC), Arenzano (Italy), 18th - 22nd September 2022. Talk presentation. *Early detection of variants of concern via funnel plots of regional reproduction numbers.*

Reviews

- Journal of Royal Society Interface, April 2024: Paper in the area of Statistics applied to Epidemiology.

Collaborations with research groups and visiting periods

- EU Horizon 2020 Research and Innovation Program PERISCOPE: Pan European Response to the ImpactS of Covid 19 and future Pandemics and Epidemics', grant agreement no. 101016233. `Paper (1)`.
- H2020-SC1-PHE-CORONAVIRUS-2020-2-RTD, NextGenerationEU Grant Uniud-DM737, 2022. `Paper (1)`.
- NextGenerationEU-MUR PNRR Extended Partnership initiative on Emerging Infectious Diseases (Project no. PE00000007, INF-ACT). `Papers (1),(2)`.

- 2022 PRIN, Grant no. 2022LP77J4, Proliferation, Resistance and Infection Dynamics in Epidemics (PRIDE), Italian Ministry for Research. `Paper (2)`.
- Socio-Technical Algorithmic Research (STAR) Lab in TU Delft (Netherlands). Visiting period: 14th April - 13th May 2023. `Paper (3)`.
- Laboratoire d'Informatique at École Polytechnique de Paris (France). Traineeship and visiting period: 20th August 2023 - 10th February 2024. `Book chapter (4)`.

Thesis co-supervision

- *Identification of Output Error models for COVID-19 modeling.* Bachelor's thesis in Computer Engineering by Samuele Di Cugno, degree awarded on October 26, 2023.
- *Multidrug resistance monitoring through funnel plot analysis.* Bachelor's thesis in Computer Engineering by Paolo Fabrizio, degree awarded on September 27, 2023.
- *Analysis of active cases and deaths for comparing non-pharmaceutical containment strategies.* Bachelor's thesis in Computer Engineering by Luca Furfaro, degree awarded on June 15, 2022.

Teaching activities

- [2022-2023] Substitute for *Algebra and Geometry* lectures for the Industrial Engineering degree program.
- [2022-2023] Coordinator of tutoring for *Algebra and Geometry* course for the Industrial Engineering degree program.
- [2021] Introductory course for first-year students in Mathematics degree program.
- [2020 - 2022] Tutoring for the *Algebra and Geometry* course for the Bachelor's degree program in Industrial Engineering.
- [2020 - 2021] Tutoring for the *Algebra and Geometry* course for the Bachelor's degree program in Electronic Engineering.

Preface

In this Preface, given the multifaceted nature of this paper collection thesis, the content will be limited to the description of the structure of the thesis, leaving the more appropriate and standard introductions to the beginning of each chapter.

This thesis, along with my PhD journey, is divided into two major sections. The first major section is dedicated to *Mathematical modelling in Biosciences* and was supervised by Professor Giuseppe De Nicolao. It consists of two chapters. The first chapter focuses on statistical monitoring and surveillance using funnel plots applied to Early Detection of COVID-19 Variants of Concern (VoCs) in Italy, India, South Africa and United Kingdom. The second chapter concentrates on reconstructing the temporal profile of new positive cases in Italy during the first wave of COVID-19 in 2020. This chapter employs tools for dynamic system identification and the regularized solution of inverse problems.

The second major section is dedicated to *Discrete Optimization* and was supervised by Professor Stefano Gualandi (University of Pavia), Professor Neil Yorke-Smith (TU Delft), Professor Leo Liberti (École Polytechnique de Paris), and Professor Carlile Lavor (University of Campinas). This section comprises two chapters. The first chapter deals with the training and simultaneous optimization of a few-bit neural network designed to solve a classification problem in a few-shot regime. Alongside the training and optimization algorithm, a voting structure is presented, which facilitates extending the setting to the multi-classification case. A significant part of the project concerning Integer Neural Networks was advanced during a one-month visit to the STARLab at TU Delft. This work was carried out in collaboration with another PhD student from the Department of Mathematics, XXXVII cycle, Ambrogio Maria Bernardelli. The second chapter focuses on the unassigned Distance Geometry Problem in the Manhattan norm applied to the Mobile Positioning Problem. The writing of this work was completed during a six-month internship to LIX at École Polytechnique de Paris.

All the material has been previously published or is currently in the process of publication. Specifically, the content is derived from the following works, enumerated in correspondence with their respective chapters:

(1) *Early detection of variants of concern via funnel plots of regional reproduction numbers* [96] is published in `Scientific Reports` and co-authored by a team of engineers and physicians, namely, Francesca Rosset (Department of Mathematics, Computer Science and Physics, University of Udine), Marta Colaneri (Infectious Diseases and Immunopathology, Department of Clinical Sciences, Università di Milano, L. Sacco Hospital; and Centre for Multidisciplinary Research in Health Science,

University of Milano), Giulia Giordano (Department of Industrial Engineering, University of Trento), Kenneth Pesenti (Department of Medicine, University of Trieste), Franco Blanchini (Department of Mathematics, Computer Science and Physics, University of Udine), Paolo Bolzern (Department of Electronics, Information and Bioengineering, Politecnico di Milano), Patrizio Colaneri (Department of Electronics, Information and Bioengineering, Politecnico di Milano; and Institute of Electronics, Information Engineering and Telecommunication of the Italian National Research Council, Turin, Italy), Paolo Sacchi (Division of Infectious Diseases I, Fondazione IRCCS Policlinico San Matteo, Pavia), Giuseppe De Nicolao (Department of Electrical, Computer, and Biomedical Engineering, University of Pavia; and Division of Infectious Diseases I, Fondazione IRCCS Policlinico San Matteo, Pavia), and Raffaele Bruno (Department of Clinical, Surgical, Diagnostic, and Pediatric Sciences, University of Pavia);

(2) *Correction of Italian under-reporting in the first COVID-19 wave via age-specific deconvolution of hospital admissions* [95] is published in `PLOS ONE` and co-authored by Giuseppe De Nicolao (Department of Electrical, Computer, and Biomedical Engineering, University of Pavia; and Division of Infectious Diseases I, Fondazione IRCCS Policlinico San Matteo, Pavia);

(3) *Multi-Objective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks* [19] is published in `INFORMS Journal on Computing` and co-authored by Ambrogio Maria Bernardelli (Department of Mathematics, University of Pavia), Stefano Gualandi (Department of Mathematics, University of Pavia), Neil Yorke-Smith (STAR Lab, Delft University of Technology, Netherlands), and Hoong Chuin Lau (School of Computing and Information Systems, Singapore Management University);

(4) *The Mobile Positioning Problem as an unassigned Distance Geometry Problem* was requested for the book *Geometric Constraint Systems: Problems, Progress, Connections, and Emerging Applications (Springer)* and co-authored by Carlile Lavor (IMECC, University of Campinas, Brazil), and Leo Liberti (LIX, CNRS, École Polytechnique, Palaiseau, France).

Part I

Mathematical Modelling in Biosciences

1

Early detection of variants of concern via funnel plots of regional reproduction numbers

Early detection of the emergence of a new variant of concern (VoC) is essential to develop strategies that contain epidemic outbreaks. For example, knowing in which region a VoC starts spreading enables prompt actions to circumscribe the geographical area where the new variant can spread, by containing it locally. This section presents *funnel plots* as a statistical process control method that, unlike tools whose purpose is to identify rises of the reproduction number (R_t), detects when a regional R_t departs from the national average and thus represents an anomaly. The name of the method refers to the funnel-like shape of the scatter plot that the data take on. Control limits with prescribed false alarm rate are derived from the observation that regional R_t's are normally distributed with variance inversely proportional to the number of infectious cases. The method is validated on public COVID-19 data demonstrating its efficacy in the early detection of SARS-CoV-2 variants in India, South Africa, England, and Italy, as well as of a malfunctioning episode of the diagnostic infrastructure in England, during which the Immensa lab in Wolverhampton gave 43000 incorrect negative tests relative to South West and West Midlands territories.

1.1 Introduction to Statistical Process Control and Funnel Plots

Statistical process control, currently known as quality control, aims to compare the performance of various institutions regarding a reliability parameter, which can be, for instance, the size of the institution itself. Funnel plots replace the traditional ranking methods that order institutions from best to worst. Instead, funnel plots adopt an approach where institutions are divided into just two groups: those exhibiting *natural variability* and those that are *out of statistical control*. David Spiegelhalter, a world-renowned statistician, has significantly contributed to the widespread use of funnel plots for meta-analyses [127]. Funnel plots are not a new concept: they are a standard tool in meta-analysis for graphically checking any relationship between effect estimates and their precision, which might indicate publication bias. They have

also been used for comparing clinical outcomes. Spiegelhalter's work further develops this tool by detailing their construction, assessing the association between volume and outcome, and addressing issues related to overdispersion. Funnel plots are a graphical tool with immediate interpretative power, making it easy to identify areas that require attention. They are grounded in solid statistical foundations, with control limits established based on specific mathematical and statistical assumptions. Furthermore, funnel plots allow for the comparison of units of analysis while accounting for unmeasured risk factors, thereby avoiding misleading rankings.

In a funnel plot, a measured or estimated quantity is plotted against an interpretable measure of its precision. It is composed of four elements [127]:

(i) an indicator Y that represents the quantity to be monitored,

(ii) a reference value θ that specifies the expectation of the indicator,

(iii) a precision parameter ρ that determines the accuracy with which the indicator is measured,

(iv) the control limits y_{lower} and y_{upper} that specify the boundaries of the out-of-control regions.

An example of funnel plot can be seen in Fig. 1.1. The dot (ρ_i, y_i) is associated with the $i-th$ region, where ρ_i is the number of infectious cases in the region and y_i is the region's reproduction number R_t at a given time t. The horizontal line $y = \theta$ shows the national average R_t and the funnel-shaped pair of control limits y_{lower} and y_{upper} shows where we would expect the regions to lie if their R_t's were statistically indistinguishable from one another, see Panel d in Fig. 1.1. In several circumstances, an exact or approximate normal distribution of the indicator Y can be assumed

$$Y|\theta, \rho \sim N(\theta, g(\theta)/\rho) \tag{1.1}$$

where g is a suitable function of θ [127] such that $Var[Y] = g(\theta)/\rho$. Under this null hypothesis, with probability $1 - \alpha$,

$$\theta - z_{\frac{\alpha}{2}}\sqrt{\frac{g(\theta)}{\rho}} \leq Y \leq \theta + z_{\frac{\alpha}{2}}\sqrt{\frac{g(\theta)}{\rho}}$$

where $z_{\alpha/2}$ is such that $P(Z \leq z_{\alpha/2})) = 1-\alpha/2$ for a standard normal variable Z. For instance, $z_{\alpha/2} = 1.96$, when $\alpha = 5\%$, and $z_{\alpha/2} = 3.09$, when $\alpha = 0.2\%$. This means that, in $100(1-\alpha)\%$ of the cases, Y is expected to lie within the lower and upper control limits defined as

$$y_{lower} = \theta - z_{\frac{\alpha}{2}}\sqrt{\frac{g(\theta)}{\rho}},$$

$$y_{upper} = \theta + z_{\frac{\alpha}{2}} \sqrt{\frac{g(\theta)}{\rho}}.$$

By introducing the Z-score $z_i = (y_i - \theta)/\sqrt{g(\theta)/\rho}$ we have that $P(|z_i| \leq z_{\alpha/2}) = 1 - \alpha$. In Statistical Process Control, the common practice is to select a false alarm probability as small as $\alpha = 0.2\%$, corresponding to $z_{\alpha/2} \geq 3.09$. A Z-score whose absolute value is greater than $z_{\alpha/2}$ is said to be out of (statistical) control and deemed worthy of study to identify a special cause of variation that explains its departure from the mean. Note that there is a 0.2% probability of reporting an out-of-control point when no special cause of variation is actually perturbing the process and the outlier arises by pure chance under common causes of variation. When monitoring n units of analysis, e.g., the R_t of n regions within a country, due to the multiple comparison problem, the false positive rate could become unacceptably large. A simple way to address this problem is the Bonferroni correction that replaces α with α/n [100]. When the indicators y_i measure a frequency of occurrence, e.g., the mortality rates in heart surgery units, it is reasonable to assume a binomial model, with θ representing the probability of the event and ρ_i the number of surgeries in the $i - th$ unit. For the binomial model, the variance of y_i is $\theta(1 - \theta)/\rho_i$ so that, given θ, the variance of y_i is completely specified. For a large enough ρ, the binomial converges to a normal random variable that follows distribution (1.1) with $g(\theta) = \theta(1 - \theta)$. An analogous case is when the products $\rho_i y_i$ are Poisson distributed with expectation $\rho_i \theta$. If $\rho_i \theta$ is greater than 30, the indicators y_i are then normally distributed as (1.1) with $g(\theta) = \theta$. Therefore, for both the ideal binomial and Poisson model, estimating the mean of y_i suffices to specify both the centerline and the alarm limits of the funnel plot. However, as discussed in [128], if one lets the variance be specified by the mean, it very often happens that the fraction of units of analysis that lie outside the ideal alarm limits greatly exceeds the theoretical false positive rate. This phenomenon, well known in the statistical literature, goes under the name of overdispersion. This can be dealt with by modifying (1.1) with the introduction of an overdispersion parameter ϕ to be estimated from data:

$$Y|\theta, \rho \sim N(\theta, \phi g(\theta)/\rho) \tag{1.2}$$

The control limits and the Z-scores are redefined accordingly as

$$y_{lower} = \theta - z_{\frac{\alpha}{2}} \sqrt{\frac{g(\theta)}{\rho}},$$

$$y_{upper} = \theta + z_{\frac{\alpha}{2}} \sqrt{\frac{g(\theta)}{\rho}},$$

$$z_i = (y_i - \theta)/\sqrt{\frac{\phi g(\theta)}{\rho_i}}.$$

When the indicators to be monitored are time series depending on a time index t, i.e., $y_i = y_i(t)$, a distinct funnel plot can be drawn for each time instant. For the purpose of statistical monitoring, the relevant information can be summarized in a Bonferroni control chart where the trends of the Z-scores are plotted in time against Bonferroni limits, see for instance panel e in Fig. 1.1. Under (1.2), we have that $z_i \sim N(0,1)$, so that, when the Z-scores are plotted on a control chart with zero centerline and Bonferroni limits equal to $\pm z_{\alpha/(2n)}$, the probability of one or more dots lying outside the limits is equal to α.

1.2 Funnel Plots applied to Early Detection of COVID-19 Variants of Concern

All viruses, including SARS-CoV-2, evolve over time. Mutations happen frequently and, in most cases, have little to no impact on the viral function. However, a group of mutations with similar genetic lineage, denoted by public health organizations as Variants of Concern (VoC), have gained global attention because of their faster spread and evidence for higher transmissibility and possibly higher virulence [34]. Surveillance aimed at the early detection of a new VoC is fundamental. The World Health Organization (WHO) and its international networks of experts closely monitor SARS-CoV-2 variants [108], but a surveillance system at a national and sub-national level is crucial to identify the emergence of new variants with the potential to spread worldwide, as well as the spread of already detected variants. Local authorities are thereby currently encouraged to strengthen surveillance and sequencing capacities, to early detect unusual epidemiological events. However, several countries still have limited capacity, despite the enormous efforts to facilitate the access to existing international networks [146] and the implementation of low-cost whole genome sequencing (WGS) methods [57]. As happened with SARS-CoV outbreaks [156], new SARS-CoV-2 variants with unforeseen mutations continue to emerge [29, 64, 140], also with the potential risk of immune evasion [30, 44]. The Omicron variant (B.1.1.529 lineage), which contains over 30 mutations in the spike protein, including the same mutations of pre-existing VoC, will definitely not be the last, and possibly not the most challenging we will ever face [43]. The important task of designing early warning systems requires a panoplia of tools, ranging from genome sequencing, epidemiological surveillance, and machine learning applied to spike protein mutations [40, 129]. To support monitoring based on epidemiological data, we propose a statistically based methodology that is easy to apply and enables the early detection of anomalous events, consequently triggering further inquiries. With respect to massive genomic sequencing, statistical methods based on epidemiological data are faster and reduce costs and needed resources; of course, they do not replace sequencing, but integrate it and may defer the genomic sequencing

methods to a more targeted and purpose-driven framework, to effectively detect potential VoCs and prevent their spread. The keystone of our approach is the use of statistical quality control to monitor the homogeneity of the time-varying estimated reproduction numbers of the disease in different regions of a country (or, more in general, in different geographical areas). The novelty of the approach consists in statistically comparing the reproduction numbers of different regions in order to detect if some territories behave as outliers. As a key feature of the proposed methodology, a rigorous statistical threshold is derived, which accounts for the different sample sizes, i.e., the number of infectious cases in a region. In the general context of healthcare monitoring, this sample size issue had come under the spotlight in the early 2000, in a series of works [5, 58, 99]. An example was the detection of abnormal mortality rates in cardiac surgery wards [99]: through the characterization of the baseline variability, one could build control charts with statistical limits which, if exceeded, suggested the existence of an abnormal cause explaining the anomalous mortality. When the key performance indicators were affected by the sample size, it was shown that their monitoring could rely on so-called funnel plots [127, 128]. In the case of epidemics, anomalies can be detected by a comparative monitoring of the regional effective reproduction number, R_t, whose variance depends on the number of new infected subjects in the given region. Closely related to R_t is the so-called basic reproduction number R_0 (i.e. R_t at the beginning of the epidemic outbreak) whose expression is obtained from mathematical models. For the analytical and numerical computation of R_0 for general structured population models, see [9, 10, 26]. A large regional R_t may have a special cause, such as the emergence of a new VoC, or may just be the effect of statistical fluctuations due to sampling noise. In this work, to monitor the onset of statistical anomalies in regional R_t's, we derive suitable funnel plots whose control limits can reveal abnormal trends, while keeping false alarms under control. We validate our proposed methodology using publicly available epidemiological data from Italy, England, India and South Africa: we show that the crossing of control limits promptly reveals the emergence of new more transmissible variants or the malfunctioning of the diagnostic infrastructure. In conclusion, we notice that the utility of funnel plots is not limited only to epidemiological setting, but have also meaningful clinical implications. Indeed, several papers [4, 130] show that VOCs have a reduced sensitivity to both antiviral drugs and monoclonal antibodies. The capacity to detect VOCs earlier means the possibility to improve the appropriateness of early therapies and to reduce hospitalizations and deaths.

1.3 Results

We apply the funnel plot methodology to five case studies, corresponding to different stages of the COVID-19 pandemic, chosen because of their relevance

to the spread of VoC's or to flaws of the diagnostic infrastructure. Two case studies refer to England (initial spread of the Omicron variant in December 2021 and large failure of a diagnostic lab in September 2021), and the other three to Italy (initial spread of the Omicron variant in December 2021), India (first emergence of the Delta variant in February 2021), and South Africa (first emergence of the Omicron variant in November 2021). In addition, the nine English regions are monitored over a 18-month period from December 2020 to June 2022. In all cases, we focus on four key dates. The first date corresponds to a situation of statistical homogeneity: when variants are uniformly spread in the country and contact rates do not vary much across regions, differences between estimated R_t's are exclusively due to natural variability and the regional R_t's are expected to lie within the funnel, centered around the national R_t (see 1.5). The second and the third dates refer to the disruption of the natural variability: when a new VoC starts spreading, at first it colonizes in particular a few territories, whose behavior becomes abnormal with respect to the national one. This is highlighted by the fact that the corresponding R_t's first cross the funnel limits and then clearly move outside the limits. The last date corresponds to a new homogeneity, typically established around a higher R_t: the VoC is now uniformly spread in the country, thus restoring the condition of natural variability. Finally, to have a snapshot of the whole period under study, the standardized R_t's with ± 3.09 sigma are plotted on a Bonferroni control chart, which is a standard univariate control chart whose control limits are adjusted according to the Bonferroni correction (see 1.5). Due to its statistical background, the scope of the new control method is not restricted to VoC monitoring, but can detect other kinds of anomalies, such as those related to testing availability or malfunctioning buffer factories: we discuss an example of such an anomaly in our Immensa case study. Hereafter, the infectious cases at day t are the total number of individuals that are infected and infectious at day t, while the new cases at day t are the number of subjects who become infectious at that time.

1.3.1 Spread of the Omicron variant in Italy

We first apply the funnel plot methodology to the Italian regional data in the period from 4 December to 3 January 2022, based on epidemiological indicators released daily by the Civil Protection Department, which provides 21 regional time series (for 19 regions and the 2 autonomous provinces of Trento and Bolzano). The Delta variant was dominant in Italy until December 2021, when the Omicron variant started to spread across the country. The results are summarized in Fig. 1.1. In the Panels a-d, the estimates of Italian regional R_t's are plotted against the infectious cases on four selected dates. On 7 December 2021 (see panel a), differences between estimated R_t's were due to natural variability alone and the 21 points lay within the funnel limits. On 22 December 2021, Lombardy (dark red) crossed the alarm limit (see panel

b) and on 24 December 2021 (see panel c) it was definitely outside the upper alarm limit. In fact, as confirmed by a retrospective survey by the Italian National Institute of Health published on 31 December 2021 [71], Lombardy was the first Italian region to be colonized by the Omicron variant. As other regions became increasingly colonized by the Omicron variant, their R_t's rose as well and, by 2 January 2022, Lombardy was absorbed again within a funnel, now with a higher mean than in early December (see panel d). We can monitor the trend by plotting the standardized R_t's on a Bonferroni control chart with ± 3.09 sigma limits (see panel e), where the arrival of the Omicron variant in Lombardy in mid-December is clearly detectable.

1.3.2 Statistical monitoring of England for a year and a half

Fig. 1.2 displays the Bonferroni control chart of normalized R_t's of the nine English regions during 18 months, from the end of November 2020 to the beginning of June 2022. Under natural variability conditions, irrespective of the current national R_t, all the normalized curves are expected to lie within the limits. Points outside the limits highlight a disruption of the statistical homogeneity across regions, which should be investigated to unveil the root cause of the anomaly. Fig. 1.2 reports seven major events, labelled from A to G, along with plausible conjectured explanations: the emergence or the arrival of the VoCs (Alpha [3, 144], Delta [138], Omicron [47, 141] and Omicron subvariants [43]), the malfunction of swab factories (further analyzed in Figure 4) [50, 139], some incidents of violation of lockdown restrictions [11, 12, 13], and changes in the testing policies [125].

1.3.3 Emergence of the Delta variant in India

We applied our methodology to epidemic data from India in the period 13 February - 5 March 2021, when the Delta variant emerged and started spreading from the state of Maharashtra. Panels a-d of Fig. 1.3 show funnel plots at four selected times, where colour-coded circles represent the R_t's of the 36 Indian states. While on February 13 all circles fell within the funnel, on February 16 the state of Maharashtra (dark red) crossed the alert threshold (in correspondence with the initial spread of the Delta variant), further departing from the mean on February 22. Lastly, on 4 March 2021, the R_t's of all regions but Kerala (orange) shaped a new funnel with a higher mean, which again incorporated Maharashtra. The peculiar dropping of Kerala's R_t below the lower alert threshold, despite the very high number of infectious cases, might be explained by the co-circulation of Alpha and Delta variants during the same period, resulting in a lower R_t than in the areas predominantly hit by the Delta variant. In the Bonferroni control chart (see panel e), the rise of the Delta variant in Maharashtra is clearly visible since mid-February 2021. One month later, on March 17, it was disclosed that a 10-lab research consor-

tium had alerted the Union Health Ministry about a new variant spreading in Maharashtra [124], leading to a press release on the new VoC a week later [69]. This case study suggests that the use of statistical control methods would have enabled an earlier detection of the variant.

1.3.4 Emergence of the Omicron variant in South Africa

From 7 November to 4 December 2021, the Omicron variant colonized South Africa, starting with the province of Gauteng. Panels f-i of Fig. 1.3 show four funnel plots, where colour-coded circles represent the R_t's of the South African provinces. Until the very beginning of November 2021, the Delta variant was prevalent and the differences in R_t across provinces merely resulted from natural fluctuations (see panel f). By mid-November the Gauteng province crossed the upper alert threshold (see panel g) and then further diverged (see panel h). This is precisely the timing when the Omicron variant was first identified, as declared by the WHO [68], and became a threat [109]. By 3 December 2021, Gauteng was reabsorbed within the funnel, now with a much higher mean, following the spread of Omicron in the other provinces and the consequent rise of their R_t's (see panel i). The Bonferroni control chart with ± 3.09 sigma limits (panel j) clearly shows the out-of-control trajectory of the Gauteng province (red).

1.3.5 Spread of the Omicron variant in England

From 4 December 2021 to 1 January 2022, the Omicron variant massively spread in England. Panels k-n of Fig. 1.3 show four funnel plots, with colour-coded circles corresponding to the R_t's of the English regions. On December 4, all the regions were within the alarm limits (panel k). By 10 December 2021, the London region had crossed the funnel limits (panel l), further diverging from the upper limit on 15 December (panel m). This suggests that Omicron was more prevalent in London than in the rest of England and indeed, on 13 December 2021, 20% of the cases in England and over 44% of the cases in London were attributed to Omicron [141]. As the other regions were colonized, the distribution of their R_t's moved upward and, on 23 December 2021, the London region was again inside the funnel (panel n). An earlier detection would have been allowed by the Bonferroni control chart, where London first crossed the alarm limit in early December (panel o).

1.3.6 Immensa scandal in England

Our last case study concerns England in the period from 27 August to 25 September 2021. Panels a-d of Fig. 1.4 display four funnel plots in selected dates, with colour-coded circles corresponding to the R_t's of the English regions. On 5 September 2021, all English regions were within the funnel

(panel a). By 9 September 2021, the South West (red) had crossed the lower alarm limit (panel b) and remained below the lower limit for about two weeks (panel e). The timing of this swing coincides with the period during which the Immensa lab in Wolverhampton gave some 43,000 incorrect negative tests relative to South West and West Midlands territories [50, 139]. While the suspension of lab operations came in mid-October, the Bonferroni control chart indicated an out-of-control condition already in early September and would have allowed a much earlier detection of the anomaly.

1.4 Discussion

We proposed funnel plots and the associated Bonferroni control chart as a valuable framework for the early detection of a new emerging or imported VoC and showed their effectiveness in six real-life scenarios based on epidemic data from Italy, India, South Africa and England. These case studies demonstrate that the proposed methodology, besides being direct and inexpensive, allows the early detection of anomalies due to different root causes, ranging from the emergence of a new VoC, and its colonization of a country, to flaws in the diagnostic system, such as the Immensa COVID-19 testing scandal in England. Once the method identifies anomalous patterns, further inquiries are needed to assess their causes. Funnel plots provide an innovative and statistically rigorous tool for monitoring the statistical homogeneity of the distribution of regional R_t's. Our method can be seen as an extension to epidemiology of the funnel charts advocated by Spiegelhalter in the assessment and comparison of institutional performances in the healthcare sector [127]. Before then, funnel plots were mainly known as a standard tool for investigating biases in meta-analysis studies. As such, they have also been employed in the context of COVID-19 meta-analyses, see e.g., [155]. Prompt identification of a VoC before its large-scale spread, leading to impactful public health implications, is a key goal in the control of the SARS-CoV-2 pandemic and in preventing and controlling future pandemics. However, as the relentless and flashy worldwide dissemination of the Omicron variant has largely proven, some doubts remain about the most effective way to achieve this goal. Although some rRT-PCR–based algorithms and/or NAAT-based screening assays have been proposed for the early identification of VoCs [91, 103] and might be implemented in routine laboratories [106], Whole Genome Sequencing, or at least the complete or partial sequencing of the spike (S) protein-gene, remains the only tool to both effectively identify the different variants and follow the evolution of SARS-CoV-2 [25, 107]. However, WGS is time consuming, expensive, and needs dedicated structures and personnel with technical expertise to be timely implemented. Furthermore, it is challenging to be applied on low viral loads samples [36]. Exactly in this breach, the potential support of surveillance based on funnel plots and Bonferroni charts might accelerate the detection

of a new VoC, without requiring, at least initially, the backup of a specialized microbiology laboratory. The value of WGS is undisputed, but, in a setting with limited resources, easy and inexpensive data-driven statistical methodologies for surveillance may support more targeted and focused genomic sequencing. Therefore, besides being extremely useful where sequencing is lacking due to scarce resources, the funnel plot framework is also precious to inform and suggest where sequencing efforts should be concentrated. It also allows the detection of anomalies that cannot be revealed by sequencing, such as failures of the testing infrastructure, as shown by the Immensa case study. The statistical underpinning of the methodology takes into account the natural variability of the phenomenon, thus preventing false alarms even in the presence of noisy data, e.g., due to late registration of new cases. While polished data may be available with weeks of delay, funnel plots can work in real-time using the latest data, a crucial feature to allow an early detection of anomalies and hence prompt interventions. For instance, the Italian funnel plots of the first case study were fed by daily published unprocessed data. Other authors have proposed the application of statistical process control methods for monitoring the evolution of the COVID-19 pandemic. For instance, [111] proposed hybrid control charts to detect the start and end of exponential growth in reported deaths within a geographic area. An interesting use of hybrid control charts was investigated in [70], keeping under control exponential and non-exponential growth and decline of cases, disaggregated at the regional and subregional level, to inform local mitigation and containment strategies. Conversely, our approach leverages the characterization of the collective distribution of regional R_t's: we do not monitor each region individually, but rather surveil the homogeneity of the distribution. In view of its nature, the proposed method reveals the loss of statistical stability, but cannot of course unravel its cause. Consistently with established quality control practices, it should be used to trigger an inspection. Therefore, the funnel plot is not a VoC-detector, but an anomaly detector: early detection enables focused inquiries aimed at discovering the cause of the anomaly. In funnel plots, a point lying outside the funnel limits is associated with high confidence to some anomaly of the effective reproduction number. This may be due to several special causes of variations, such as VoCs, outbreaks due to violations of containment measures, failures of the diagnostic infrastructure (such as the Immensa scandal). In the absence of special causes, all funnel plots in the chapter are designed so that all points are inside the alarm limits in 99.8% of the cases (or, equivalently, so that the false alarm probability is 0.2%).

1.5 Material and Methods

Data. Data regarding new positive cases were obtained from publicly available sources:

- https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni for Italian data,
- https://data.covid19bharat.org/ for Indian data,
- https://mediahack.co.za/datastories/coronavirus/data/# for South African data,
- https://coronavirus.data.gov.uk/details/download for English data.

Following [61], we assumed a discretized lognormal distribution for the serial interval, with parameters chosen in accordance with [53]. To correct systematic errors in the data, partly due to the weekly periodicity, partly due to delays and other reporting errors, all data were filtered using a double seven-day moving average.

Distribution of regional R_t's. The reproduction number at time t, named R_t, captures the number of secondary infections from a population including both susceptible and immune individuals. For its estimation, a range of model frameworks and estimation procedures have been proposed [1]. Herein we adopt the approach of Cori et al. [38] that makes minimal assumptions about the mathematical model of the epidemic process. Cori's formula uses the time series of the new cases and estimates of the distribution of the generation time, i.e., the time between infections. According to [38] the estimate $\hat{R}_t$ of the instantaneous reproduction number R_t is obtained as

$$\hat{R}_t = \frac{I_t}{\sum_{s=1}^{t} w_s I_{t-s}} \tag{1.3}$$

where I_t denotes the daily number of new infected cases and w_s are the coefficients, adding up to one, of the infectivity profile, often approximated by the distribution of the serial interval. The denominator

$$\Lambda_t = \sum_{s=1}^{t} w_s I_{t-s}$$

can be interpreted as the total infectiousness of individuals that are currently infected at time t. In view of the typical models of the infectivity profile, e.g., lognormal or gamma density functions, Λ_t is a smoothed version of the time series I_t of daily new cases. If seven-day moving averages are used to filter out weekly oscillations, I_t is already smooth, and the resulting $Lambda_t$ is

insensitive to the precise shape of the infectivity profile. This feature may prove helpful when a new VoC arises whose infectivity profile is unknown, or only approximately known. To derive the distribution of $\hat{R}_t$, we only assume that disease transmission follows a Poisson distribution with mean $\Lambda_t R_t$:

$$I_t|R_t, \Lambda_t \sim Pois(R_t\Lambda_t)$$

Typically, $R_t\Lambda_t > 30$, so that a normal approximation can be used:

$$I_t|R_t, \Lambda_t \sim N(R_t\Lambda_t, R_t\Lambda_t)$$

In view of 1.3, it follows that $\hat{R}_t|R_t, \Lambda_t \sim N(R_t, R_t/\Lambda_t)$. For the sake of interpretability, rather than using the notion of total infectiousness Λ_t, it is more intuitive to refer to the total number of infectious individuals. To this aim, we introduce the parameter

$$\gamma := \frac{1}{\sum_{s=1}^{\infty} sw_s}$$

i.e., the inverse of the mean serial interval, which, for the well-known SIR model, corresponds to the removal rate [8]. Then, $\rho_t = \Lambda_t/\gamma$ represents the number of individuals that are infectious at time t. Letting $\theta = R_t$, and $g(\theta) = R_t/\gamma$, it follows that

$$\hat{R}_t|\theta, \rho_t \sim N(\theta, g(\theta)/\rho_t).$$

Comparing the above distribution with 1.1, it follows that, for any given t, the scatter plot of $\hat{R}_t$ against ρ_t is indeed a funnel plot. Also in this case, it is convenient to introduce an overdispersion parameter ϕ, so that, in accordance with 1.2, the final model becomes

$$\hat{R}_t|\theta, \rho_t \sim N(\theta, \phi g(\theta)/\rho_t) \tag{1.4}$$

A useful byproduct of introducing overdispersion is that ϕ takes into account the effect that possible errors or uncertainties in the estimated mean serial interval has on the variance of $\hat{R}_t$. Indeed, the variance of $\hat{R}_t$ is inversely proportional to γ, but the effect of a wrong γ is automatically compensated when estimating ϕ from the data.

Parameter estimation. Under (1.2), the datum is

$$Y_i = \theta + \frac{\epsilon_i}{\sqrt{x_i}} \tag{1.5}$$

where $\epsilon_i \sim N(0, \sigma^2), i = 1, \ldots, n$ are IID, and $\sigma^2 = \phi g(\theta)$.

Letting $v_i = \epsilon_i/\sqrt{x_i}$, the model in matrix form becomes

$$Y = \Phi\theta + v$$

where $Y = (Y_i)_{i=1,\dots,n}, \Phi = (1)_{i=1,\dots,n}$, and $v \sim N(0, \sigma^2\Sigma)$, where

$$\Sigma = \begin{pmatrix} \frac{1}{x_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{x_n} \end{pmatrix}.$$

Then, the generalized least squares technique [14] provides the minimum variance unbiased estimate and the estimated parameters are

$$\hat{\theta} = (\Phi^T\Sigma^{-1}\Phi)^{-1}\Phi^T\Sigma^{-1}Y$$

$$\hat{\sigma^2} = \frac{1}{n}e^T\Sigma^{-1}e \tag{1.6}$$

where $e = Y - \Phi\hat{\theta}$ is the vector of the residuals. Then, the overdispersion parameter ϕ is estimated as $\hat{\phi} = \hat{\sigma^2}/g(\theta)$. Data winsorization can be performed, as detailed in [127], to reduce the effect of possibly spurious outliers. The following lines provide specific details for this chapter. By defining $Y_i = \hat{R_t^i}$, and $x_i = \rho_t^i$, we can add the following. Estimates $\hat{\theta}$ and $\hat{\phi}$ are needed to compute the funnel at time t, as well as the standardized residuals z_i to be plotted in the Bonferroni control chart. The estimate $\hat{\phi}$ is computed from (1.6) using the set of $\hat{R_t^i}$ that were in control at time $t-1$. The centerline $\hat{\theta}$ is obtained by projecting at time t a weighted linear regression estimated from the set of $\hat{R_t^i}$ that were in control at times $t-1, t-2$, and $t-3$. The weights are given by the numbers x_i of infectious individuals at the same times. This procedure yields an estimate of the current θ that tracks the trends of the national R_t, but is still fairly robust thanks to the use of the last three data points. At the beginning (and in the rare cases when all units are out of control), the whole sets $\{\hat{R_t^i}\}, \{\rho_t^i\}$ are fed to the estimator.

1.6 Conclusions

In this study, we introduced an innovative approach based on funnel plots and Bonferroni control charts for the early detection of emerging Variants of Concern (VoCs) during the COVID-19 pandemic. The proposed methodology proved effective across various real-world case studies in Italy, England, India, and South Africa, detecting anomalies in regional reproduction numbers (R_t) that were linked to both the emergence of new variants, such as Alpha, Omicron and Delta, and failures in the diagnostic infrastructure, as illustrated by the Immensa testing scandal in England.

The value of this method lies in its simplicity, cost-effectiveness, and ability to promptly identify anomalous behaviors in real-time, reducing the need for resource-intensive processes such as full genome sequencing. While it does not replace sequencing, it serves as a useful complementary tool, enabling more targeted sequencing efforts and further investigations.

Lastly, the statistical rigor of funnel plots minimizes the risk of false alarms, ensuring reliable detection of anomalies—crucial for timely interventions in public health settings. This approach could also prove valuable in future pandemics, enhancing epidemiological surveillance and optimizing the use of limited resources.

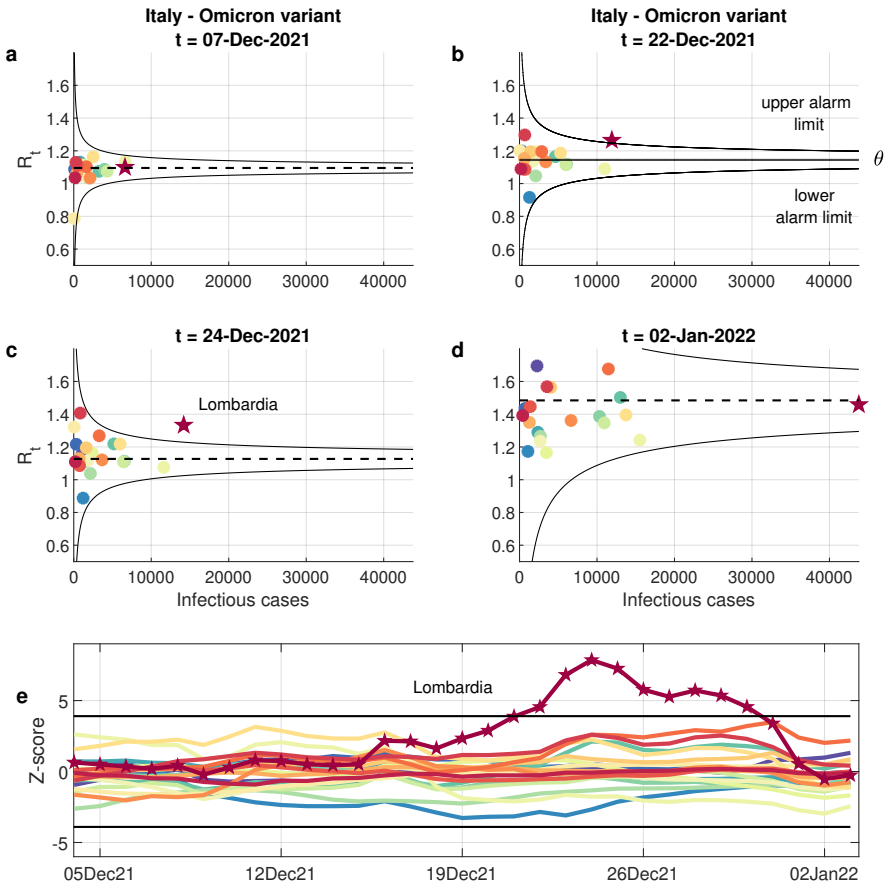


Figure 1.1: **Monitoring regional reproduction numbers (R_t's): funnel plots and control chart.** Panels a-d show the Italian regional R_t's (colour-coded circles), plotted against the infectious cases at four selected times. When the epidemic evolution is homogenous across regions, differences between Rt's are exclusively due to natural statistical variability and the circles are expected to lie inside the black alarm limits in 99.8% of the cases. The alarm limits have the shape of a funnel because the variance of the estimated R_t is inversely proportional to the number of infectious cases. The central dashed line represents the average Rt. A circle is out of statistical control if it lies outside the black funnel. Out-of-control circles might therefore reveal anomalies that disrupt the homogeneity between regions. In Panels a-d, the majority of the points, lying in the funnel, are essentially indistinguishable and therefore not even named. On 22 December 2021, Lombardy (dark red) crossed the alarm limit and on 24 December 2021 it was completely outside the upper alarm limit. As confirmed by a survey by the Italian National Institute of Health, Lombardy was the first Italian region to be colonized by the Omicron variant. As the other regions were colonized too, the distribution of their R_t's moved upward and, on 2 January 2022, Lombardy was again inside the funnel. The trend can be monitored by plotting the standardized R_t's on a Bonferroni control chart with ± 3.09 (Panel e), where the introduction of the Omicron variant in Lombardy in mid-December is clearly visible.

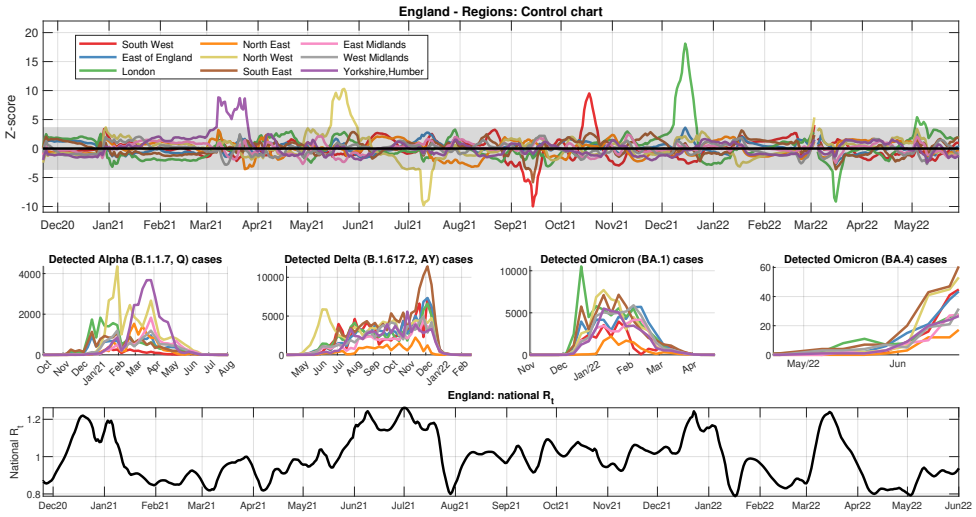


Figure 1.2: **Monitoring regional homogeneity in England from December 2020 to May 2022.** The upper panel shows the Bonferroni control chart of the normalized R_t's of English regions during 18 months. In this period, seven major events, labelled from A to G, are visible, whose plausible explanations are conjectured below. A – Alpha variant: although no crossing is observed, the curves form two clusters, and the upper is formed by the three regions where the alpha variant first became dominant. No crossing occurs because, rather than starting in a unique region, the variant colonized three regions at the same time, hindering out-of-control detection. B – Outbreaks in Yorkshire: during Spring 2021, the number of infection cases in the region was slowly decreasing and then it had a huge peak, due to very high numbers of manufacturing jobs and related high-exposure workplaces, leading to outstanding outbreaks such as the one in a Selby warehouse with more than 700 employees. C – Delta variant first arrival in the North West: in May 2021 the Delta variant, originated in India, started its colonization of England from the North West region. As a characteristic feature of the control chart, the curve that first crosses the upper limit in correspondence of a new VoC, later on is often going to cross also the lower limit if its Rt is the first one to decrease, as happened to the North West. D – Immensa scandal: see Fig. 1.4 for a discussion of the massive lab malfunctioning that perturbed case recording in the South West and South East. Compared to the rise of a new VoC, a specular pattern is observed: when the malfunctioning is fixed, a fake outbreak is observed and the red curve crosses the upper control limit. E – Omicron 1 variant first arrival in the London region: see Fig. 1.3 (panels k-o) for a discussion. F – In mid February, the UK Government announced that on April 1 free tests would be suspended, thus triggering social behaviors that may explain the anomalous trends in the control chart. G – Omicron 4 variant first arrival in the London region. For validation purposes, the four middle panels report the number of weekly detected cases, in each region, of the main VoCs [117], Alpha (B.117, Q) , Delta (B.1.617.2, AY), Omicron 1 (BA.1) and Omicron 4 (BA.4), during selected time windows. In each case, the control chart correctly identifies the region where the spread starts. The number of detected cases for Omicron 4 is far less than for the other VoCs, due to cuts to genomic surveillance: the value of the control chart as a surveillance tool is even more evident. Finally, the bottom panel reports the time profile of the national R_t, to facilitate the connection with the different phases of the pandemic in UK.

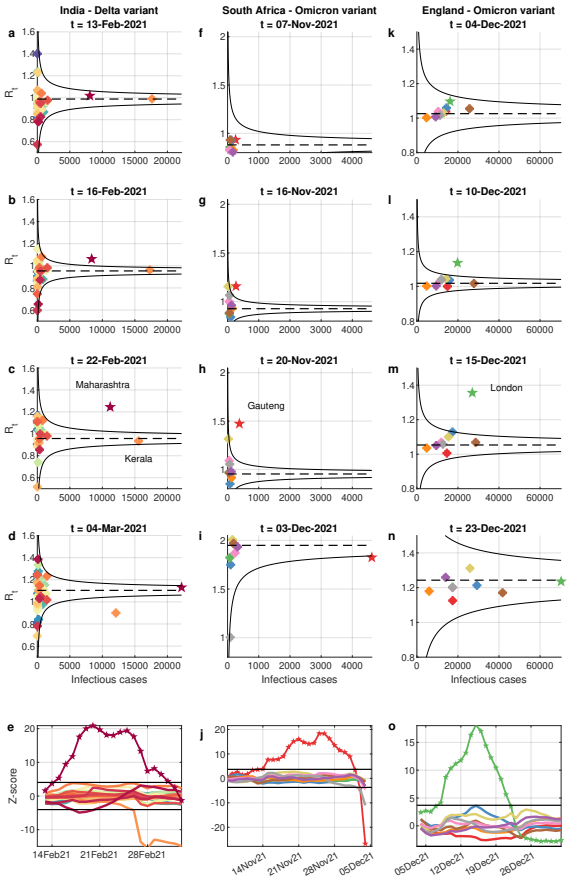


Figure 1.3: **Funnel plots help detect anomalies: spread of the Delta variant in India and of the Omicron variant in South Africa and England.** India: Panels a-d display the funnel plots at four selected times, with colour-coded circles corresponding to the R_t's of the Indian states. On 13 February 2021, all points are within the funnel, but on 16 February 2021, when the Delta variant starts spreading, there is an out-of-control point corresponding to Maharashtra (dark red), which on 22 February 2021 is further apart from the mean. Finally, on 4 March 2021 the Rt's of all regions except Kerala (orange) converge to a new distribution characterized by a higher R_t. The trend can be monitored by plotting the standardized R_t's on a Bonferroni control chart with ± 3.09 sigma limits, see Panel e, where the rise of the Delta variant in Maharashtra is clearly visible. South Africa: Panels f-i display the funnel plots at four selected times, with colour-coded circles corresponding to the Rt's of the South African provinces. The rise of the Omicron variant in the Gauteng province (red) is well visible both in the funnel plots and in the Bonferroni control chart reported in Panel j. England: Panels k-n display the funnel plots at four selected times, with colour-coded circles corresponding to the Rt's of the English regions. The spread of Omicron in England started from the London region (green), whose R_t had already crossed the alarm limit on 10 December, when, as seen in Figure 3E of [47], the daily proportion of Omicron infections did not exceed 25%. As the other regions were colonized, the distribution of their R_t moved upward and, on 23 December, the London region was again inside the funnel, as also seen in the Bonferroni control chart reported in Panel o.

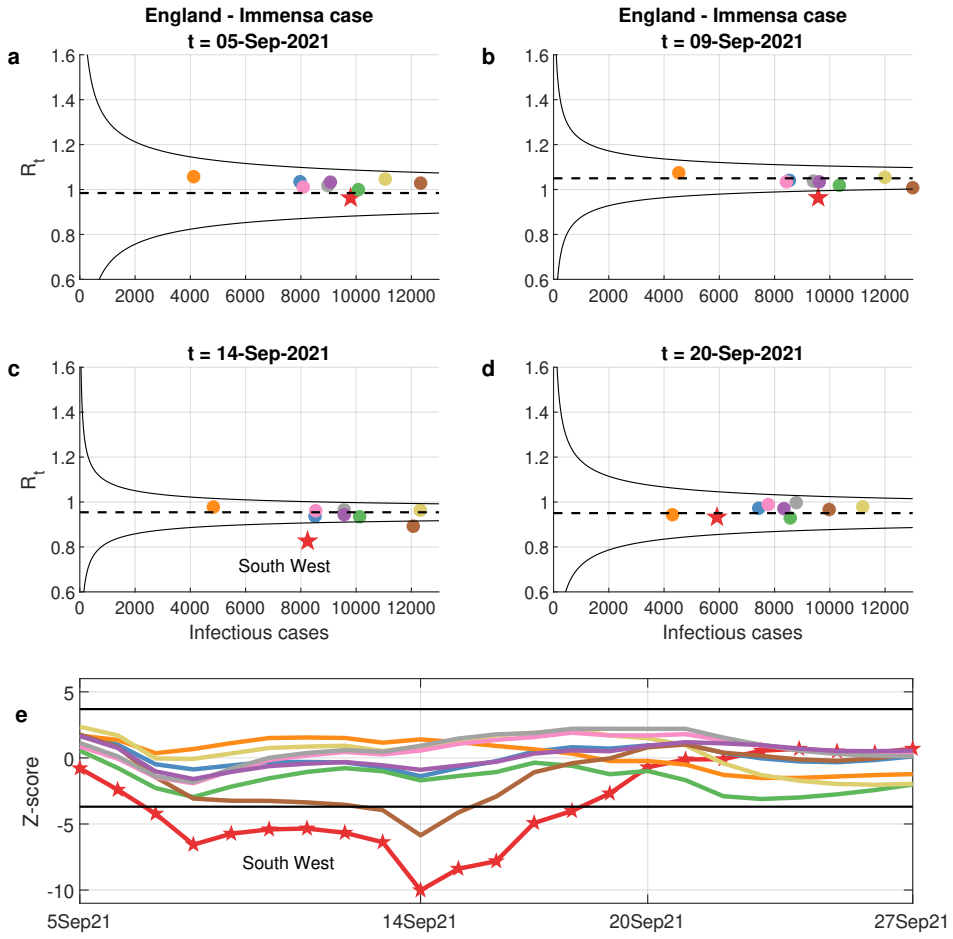


Figure 1.4: **Funnel plots help detect anomalies: the incorrect negative tests of the Immensa lab in England.** Panels a-d display the funnel plots at four selected times, with colour-coded circles corresponding to the R_t's of the England regions. On 5 September 2021, all circles were inside the funnel, but on 9 September 2021 there was an out-of-control point below the lower alarm limit corresponding to South West (red), which was further apart from the mean on 14 September 2021, when also West Midlands (brown) went below the lower limit. Finally, on 20 September 2021 the R_t's of all regions returned within the limits. The anomalous decrease of R_t in the South West corresponds to the period during which the Immensa lab (Wolverhampton) gave some 43000 incorrect negative tests relative to South West and West Midlands. The whole trend can be monitored by plotting the standardized R_t's on a Bonferroni control chart with ± 3.09 sigma limits, see Panel e. Lab operations were suspended in mid-October as a consequence of the malfunction, while the control chart indicated an out-of-control condition as early as late August.

2

Correction of Italian under-reporting in the first COVID-19 wave via age-specific deconvolution of hospital admission

When the COVID-19 pandemic first emerged in early 2020, healthcare and bureaucratic systems worldwide were caught off guard and largely unprepared to deal with the scale and severity of the outbreak. In Italy, this led to a severe underreporting of infections during the first wave of the spread. The lack of accurate data is critical as it hampers the retrospective assessment of nonpharmacological interventions, the comparison with the following waves, and the estimation and validation of epidemiological models. In particular, during the first wave, reported cases of new infections were strikingly low if compared with their effects in terms of deaths, hospitalizations and intensive care admissions. In this chapter, we observe that the hospital admissions during the second wave were very well explained by the convolution of the reported daily infections with an exponential kernel. By formulating the estimation of the actual infections during the first wave as an inverse problem, its solution by a regularization approach is proposed and validated. In this way, it was possible to compute corrected time series of daily infections for each age class. The new estimates are consistent with the serological survey published in June 2020 by the National Institute of Statistics (ISTAT) and can be used to speculate on the total number of infections occurring in Italy during 2020, which appears to be about double the number officially recorded.

2.1 The problem of estimating the size of the pandemic outbreak accurately

The availability of accurate data on a pandemic outbreak is essential, as the data provide key information on the spread of the disease and enable authorities to assess and compare public health policies. Available data on COVID-19 cases can be used, e.g., for forecasting and analyzing deaths, hospitalizations, and occupation of intensive care beds, thereby deciding if and what containment measure to take. Also, estimating the size of an outbreak can predict the outbreak's future trajectory, thus informing planning about resources and

interventions.

In Italy, the Istituto Superiore di Sanità (ISS) [72], was in charge of the official recording of COVID-19 data, based on which government committees were informed about the status of the pandemic phases and and advices were issued on non-pharmaceutical interventions (NPIs). It is widely recognized that, during the first outbreak, these official statistics largely underestimated the true number of cases [81]. The Italian national Institute of Statistics (ISTAT) estimated the under-reporting during the first wave by conducting a seroprevalence survey published by ISTAT [73]. It was found that the number of positive subjects was about six times larger than the number of registered cases. Knowing this ratio, however does not provide a reliable correction for the daily time series, apart from knowing that they suffer from a severe underdetection.

The issue of correcting for the underreporting has been addressed by several authors using a variety of approaches. Giordano and colleagues [55] employed a compartmental model that incorporated the potential presence of undetected symptomatic and asymptomatic COVID-19 cases. The model's parameters were calibrated empirically using data on reported cases. Their simulations predicted that during the first phases as many as 35% of the cases were not reported. Alternatively, also Bayesian methods have been employed [150]. Since they require a number of assumptions on the prior distributions of key parameters, these methods suffer from some subjectivity.

An approach more grounded on evidence is to leverage the health effects of the outbreak in order to trace back to the infections. For instance, whenever COVID-19 related deaths, hospitalization and ICU admissions or occupancy are less prone to underreporting, their recordings can be used to reconstruct the causes, i.e. the infections. Of course, this is possible provided that a reliable cause-effect model is available. For instance, in the delay-adjusted CFR (Case Fatality Ratio) model [142], deaths are modeled as a fraction of new cases delayed by some interval, see [113]. More precisely, letting $u(t)$ denote new cases at time t, the delay adjusted model predicts deaths at time t as $d(t) = \mathrm{CFR}u(t-T)$, where T is the average delay between case recording and deaths. In this case, the CFR must be known in advance or estimated from a dataset where new cases do not suffer from underreporting [132]. Correction of underreporting by the delay-adjusted CFR method is easily achieved letting the estimated time series of infections be equal to the time shift of the time series of deaths divided by the CFR. Of course, the same method can be used replacing deaths with other health effects such as hospital or ICU admissions or also hospital or ICU occupancy. In these cases, the cause-effect model consists of the fraction of severe cases leading either to hospital or ICU admission as well as the typical delay from infection to the admission.

The main shortcoming of these delay adjusted methods is that they do not account for the nondeterministic nature of the delay between infection and the considered outcome, such as death or hospital/ICU admission. In-

deed, the delay changes from patient to patient according to some probability distribution. In fact, due to the randomness of the delay, the time series of the outcome is not just a shifted version of the new cases, as recognized by Noh and Danuser, who devise an ingenious method, based on Expectation-Maximization, in order to improve the estimate of the ascertainment rate of new cases [105]. It is worth noting that the use of a unique Case Fatality Ratio, that makes no distinction for age, widens the confidence intervals of the reconstructed new cases.

A key observation, in order to achieve a better reconstruction of actual infections, is that hospitalizations could be better modeled as the convolution of new cases with a kernel function proportional to the random lag distribution. The proportionality coefficient is equal to the fraction of cases experiencing the given outcome: in the case of death it is the apparent CFR, and in the case of hospitalization it is the fraction of cases admitted.

The implementation of more rigorous correction methods has been hampered by the need of enhanced approaches for both the identification and correction procedures. Specifically, in the identification phase, the estimation of the delay distribution must be taken into consideration in addition to the two parameters of delay and gain. This estimation could be performed utilizing epidemiological data not impacted by underreporting. Furthermore, in the correction step, scaling and shifting techniques need to be replaced by the resolution of an inverse problem, namely a deconvolution problem, which increases the complexity of the correction procedure.

In the present work, we address both steps: a technique based on linear system identification and regularization for inverse problems is proposed as a method to correct underreporting and used to reconstruct the time profile of new positives during the first COVID-19 wave in Italy.

The chapter is organized as follows. In the *Material and Methods* section, we introduce the system identification technique used to learn the cause-effect relationship between new positive cases and hospitalizations. We then describe the deconvolution problem and the algorithm for its solution. Finally, we present a method for assessing the uncertainty of the proposed reconstruction. The *Results* and *Discussion* sections present our main findings and their interpretation, as well strengths and limitations of the proposed approach. The *Conclusions* end the chapter.

2.2 Materials and methods

Data

Data were downloaded from a publicly available platform (see *Data Availability Statement*). A centered seven-day moving average was applied [118] to

filter noise errors in the data, mostly due to weekly oscillations but also caused by random delays and recording errors.

2.2.1 Identification of input-output model of hospital admissions

Letting the t (days) be the integer time, u_t will denote the observed time series of the daily number of new positive subjects (at swab time) and y_t the number of daily hospital admissions. It is assumed that y_t obeys a convolution model:

$$y_t = \sum_{k=0}^{\infty} g_k u_{t-k-D} + \epsilon_t,$$

where D is an integer delay, ϵ_t is a zero-mean white noise and the kernel g_t is the impulse response of the linear system with input u_t and output y_t.

A useful constant related to the impulse response is the gain

$$\mu := \sum_{t=0}^{\infty} g_t$$

which represents the expected fraction of observed positive subjects that will eventually be hospitalized. The normalized function $\tilde{g}_t = g_t/\mu$ can be interpreted as the probability distribution of the random time T_h elapsed from D days after the swab day to the day of hospital admission. A particular case is given by an exponential distribution:

$$g_t = \begin{cases} 0 & \text{if } t < 0 \\ \beta\alpha^t & \text{if } t \geq 0 \end{cases} \tag{2.1}$$

where $\beta > 0$ and $0 \leq \alpha < 1$. The average of T_h is $\bar{T}_h = \mathbb{E}[T_h] = \log(\alpha)$. Note also that

$$\mu = \frac{\beta}{1-\alpha}$$

In order to estimate the parameters α and β of the impulse response, the `oe.m` function of the MATLAB System Identification Toolbox was employed [93]. The MATLAB function `oe.m` relies on prediction error minimization carried out by means of a Levenberg-Marquardt nonlinear least squares algorithm [88].

Reconstruction of new cases via deconvolution

An inverse problem is the process of estimating from a set of observations the unknown causal factors that produced them. In the context of the present chapter, the set of observations are the hospital admissions, while the unknown causal factor is the time series of new cases. When the cause-effect

relationship is described by a linear time-invariant system, the output signal is the convolution of the input with an impulse response and the inverse problem is called deconvolution. Observing that during 2020 new variants were not observed in Italy, it is assumed that the impulse response did not change from the first to the second wave.

In order to formulate the deconvolution problem, it is worth introducing a matrix representation of the convolution model. Let $U = (u_{t-D})_{t=1\dots n} \in \mathbb{R}^n$, $Y = (y_t)_{t=1\dots n} \in \mathbb{R}^n$ be the vectors of new cases and hospital admissions during the n days of the first wave. Moreover, $E = (\epsilon_t)_{t=1\dots n} \in \mathbb{R}^n$ will denote the vector of the errors.

Assuming $u_t = 0, t \leq -D$, if the impulse response g_t is as in (2.1), we have that

$$Y = GU + E \tag{2.2}$$

$$G = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & & \vdots \\ \vdots & & \ddots & 0 \\ g_{n-1} & g_{n-2} & \dots & g_0 \end{bmatrix} \tag{2.3}$$

Then, for a known impulse response g_t, the problem of estimating the new cases from the hospitalization admissions can be written as

$$C = \arg\min_U \{\phi(Y, GU) + \lambda J(U)\} \tag{2.4}$$

where

$C \in \mathbb{R}^n$ is the vector of estimated new cases;

$G \in \mathbb{R}^{n\times n}$ is the convolution matrix obtained by the impulse response g_t identified from the second wave data;

$\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the loss function, whose purpose is to measure the data fitting;

$J : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a regularization penalty function, whose purpose is to measure irregularity of U;

$\lambda \geq 0$ is the hyperparameter that adjusts the relative importance between of ϕ and J.

In order to relax the assumption $u_t = 0, t \leq -D$, we can include as additional unknowns the $L + 1$ input values $u_t, -(D + L) \leq t \leq -D$, so that $U = (u_{t-D})_{t=-L,\dots n} \in \mathbb{R}^{n+L+1}$ In this case, the matrix G in (2.3) has to be updated

as follows

$$G = \begin{bmatrix} g_{L+1} & \cdots & g_0 & 0 & 0 & 0 \\ g_{L+2} & \cdots & \cdots & g_0 & 0 & 0 \\ \vdots & & & & \ddots & 0 \\ g_{L+n} & \cdots & \cdots & \cdots & \cdots & g_0 \end{bmatrix}$$

Although the number of unknowns is greater than the number of observations, by a proper choice of the loss ϕ and the regularization term J one can still ensure that (2.4) is convex so that a unique solution exists.

Concerning the cost function, a classical choice is to use a quadratic loss $\phi(v) = ||v||_2^2 = \sum_{i=1}^{n} v_i^2$ and a regularization penalty J equal to the squared 2-norm of a linear operator applied to the input, i.e. $J = ||Pu||_2^2$. Some possible choices for P are:

- $P = I_n \in \mathbb{R}^{n \times n}$ or ridge penalty;
- $P = \Delta_n \in \mathbb{R}^{n \times n}$ or penalty on first differences;
- $P = \Delta_n^2 \in \mathbb{R}^{n \times n}$ or penalty on second differences.

where I_n is the identity matrix of order n and

$$\Delta_n = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix}$$

Under these assumptions, (2.4) is a quadratic programming problem whose closed form solution is

$$C = (G^T G + \lambda P^T P)^{-1} G^T Y$$

Observe that there is no guarantee that $c_t \geq 0 \ \forall t = 1 \dots n$. Whenever violated, the nonnegativity constraint is dealt with by reformulating the problem as a constrained quadratic programming one, solved via Matlab's Optimization Toolbox [92].

Finally, regardless of the selected approach, the tuning of the regularization penalty has to be properly addressed. Herein, Mallow's C_p criterion is adopted, assuming $\mathbb{E}[E] = 0, \mathrm{Var}[E] = \sigma^2 I_n$ [90]. The variance σ^2 is estimated by fitting an overparametrized model, i.e., a 20-degree polynomial, to the hospital admission data, capturing random noise by overfitting and isolating it from systematic trends.

2.2.2 Assessing uncertainty via boostrap

In order to assess the uncertainty of the estimated impulse response, we resorted to a *wild bootstrap* resampling scheme [149], which allows for heteroschedastic data, like the hospital admissions whose variance is possibly nonuniform.

Concerning the uncertainty of the impulse response identified from the second wave data, the idea is to start from the predicted time series $\hat{y}_t$ of hospital admissions obtained by convoluting the new cases with the identified impulse response $\hat{g}_t$:

$$\hat{y}_t = \sum_{k=0}^{\infty} \hat{g}_k u_{t-k-D}$$

Then, letting $e_t = |y_t - \hat{y}_t|$, the bootstrapped datasets are obtained by generating resampled data as

$$y_t^* = \hat{y}_t + e_t v_t$$

where $\{v_t\}$ are indipendent and identically distributed standard normal variables. Each boostrapped series of hospital admissions is then fed into `oe.m` to obtain a new realization $\hat{g}_t^*$ of the identified impulse response. Given a sufficient number of realizations, for any given time t, it is then possible to compute percentiles for g_t. Moreover percentiles for the gain μ and the time constant $\bar{T}_h$ can also be obtained.

For what concerns the uncertainty of the reconstructed profile of first wave new cases, knowing the uncertainty of the impulse response is not sufficient, because a second source of uncertainty must be accounted for, namely the noise affecting the daily hospital admissions during the first wave. To account for both sources, a double wild boostrap scheme can be used: not only is the deconvolution replicated on the boosted hospital admissions but, in each replication, a different impulse response from the previously described wild boostrap performed on the second wave is used. As a result, several realizations of the reconstructed new cases are obtained, so that percentiles can be computed both pointwise or relative to the total cases.

2.3 Results

2.3.1 Hospitalization model during the second wave

The daily new cases and daily hospital admissions for the different age groups during the second wave in the period 1 October - 15 December 2020 are displayed in Fig 2.1. For all ages, the time series of the new cases exhibit an increase until the beginning of November when the progression slows down and a declining phase starts in correspondence with the containment measures taken by the Italian Government. The time series of the daily hospital

admissions show a similar profile. Depending on the age group, their maxima range from some tens (e.g. age 0-9) to some hundreds (e.g. age 70-79).

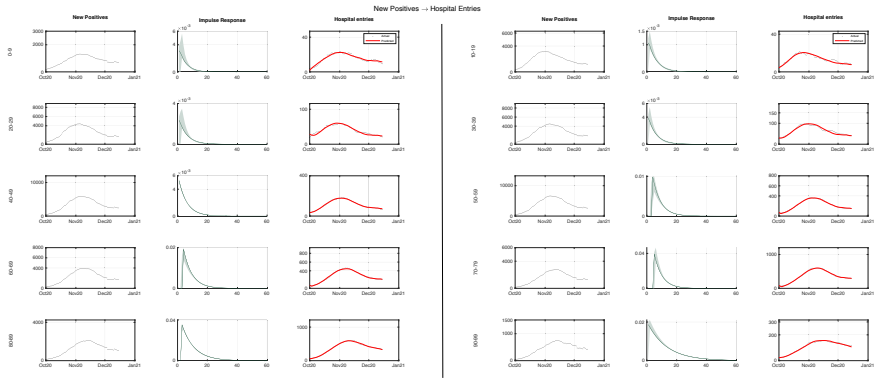


Figure 2.1: **Identification of input-output model of hospital admissions.** In the first and fourth columns, the daily new cases (black dots) during the second wave are plotted for the ten age groups. The new cases for all age groups showed an upward trend until early November, when the growth rate slowed down and a declining phase began in correspondence of the government's NPIs. An exponential impulse response model was identified for all age groups using Matlab's `oe.m` function. The impulse responses (green) are shown in the second and fifth columns, along with pointwise confidence intervals calculated via wild bootstrap. The new cases were convoluted with the estimated impulse response to produce the predicted hospital admissions (red) for all ten age groups that are plotted in the third and sixth columns together with the actual admissions (black dots).

The exponential impulse response models and the delay parameters D were identified for all age groups from the new positives and hospital admissions via Matlab's `oe.m` function. It resulted that the delay parameters D were negative for all age groups and ranged from -10 to -6. This does not violate causality, because D would be strictly positive is new cases were recorded at infection time, while the positive cases were actually recorded at swab time. The obtained impulse responses are displayed in Fig 2.1 together with pointwise confidence bands computed via wild boostrap. The differences between the ten impulse responses can be appreciated from the upper panel of 2.5, where they are displayed together. Each exponential impulse response is characterized by two parameters: its area, i.e. the gain μ, and the time constant $\bar{T}_h$, whose estimated values are reported in Table 2.1 together with their 2.5th and 97.5th percentiles. The gains ranged from 0.68% for the age group $10-19$ to 30.62% for the age group $80-89$, reflecting how illness severity varied with age. Concerning the time constant, it ranged from 4.8 days (age $0-9$) to 12.8 days (age $90-99$). Both parameters were estimated with good precision. If impulse is normalized to have unit area, it can be interpreted as a probability density function of elapsed time. The ten density functions are displayed together in lower panel of 2.5 in the Supplementary Materials. The

age groups until 49 year-old exhibit a very similar distribution with shorter mean compared to the three age groups from 50 to 79 year-old. Finally the last two age groups have the longest time constants.

In order to assess the convolution model, the new cases were convoluted with the estimated impulse response, thus the time series of predicted hospital admissions for all the ten age groups. As seen in Fig 2.1, in all cases there is very good agreement between the predicted and observed items.

2.3.2 Reconstruction of first-wave infections via deconvolution

The new cases during the first wave (7 January - 15 May 2020) were reconstructed via regularized deconvolution applied to the time series of first wave hospital admissions. The estimated new cases per 100,000 subjects are displayed in Fig 2.2, together with pointwise 95% bands. The shapes are similar with steep rise, a peak at the beginning of March and a subsequent decay. In all age groups the time series of the reconstructed new positives is uniformly larger than that of official ones, except for the age group 80-89, where official cases exceed the reconstructed ones in the second half of March, and the age group 90-99 where the phenomenon, occurring since mid March, is even more apparent. In the lower right corner of Fig 2.2, the ten profiles are plotted together to form a surface, giving new cases per 100,000 as a function of date and age.

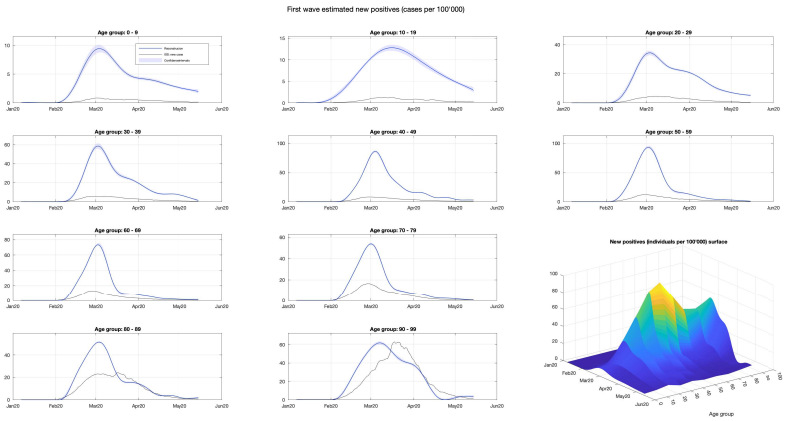


Figure 2.2: **Reconstruction of first-wave infections through regularized deconvolution.** The new cases during the first wave were reconstructed via regularized deconvolution, starting from the daily hospital admissions. The estimated new cases per 100,000 individuals (blues) are shown along with pointwise 95% confidence intervals (light blue). The recorded cases (black) are also plotted for comparison. The patterns are similar across the age groups, featuring a rapid increase, a peak in early March, and a subsequent decline. In the bottom right corner, the ten profiles are combined to form a surface, providing the new cases per 100,000 as a function of date and age.

Fig 2.3 provides a simultaneous look at all the age groups for both absolute number (upper panel) and cases per 100,000 (lower panel). Concerning absolute numbers, the largest numbers of daily cases referred, for both waves, to the age groups from 40 to 59, with higher values reached in the first wave. Concerning the cases per 100,000, the two waves show some differences. In the first one, the first two peak values are again reached by the age groups 40-49 and 50-59, while in the second wave the first two peak values are reached by the 90-99 and 20-29 age groups. In both waves, the 90-99 group is the one exhibiting a slower decay. Another major difference regards the younger age groups. In the first wave the 0-9 and 10-19 groups show the lowest peak values, below 10 and 15 cases per 100,000, respectively. In the second wave, instead, the group 0-9 has still the lowest peak value, but reaches 27 daily cases per 100,000, while the new cases of group 10-19, not only rise earlier but their peak value exceeds 50 daily new cases, more than the peak values of the 60-69 and 70-79 age groups. In Table 2.2, additional information about the underestimation factor is provided.

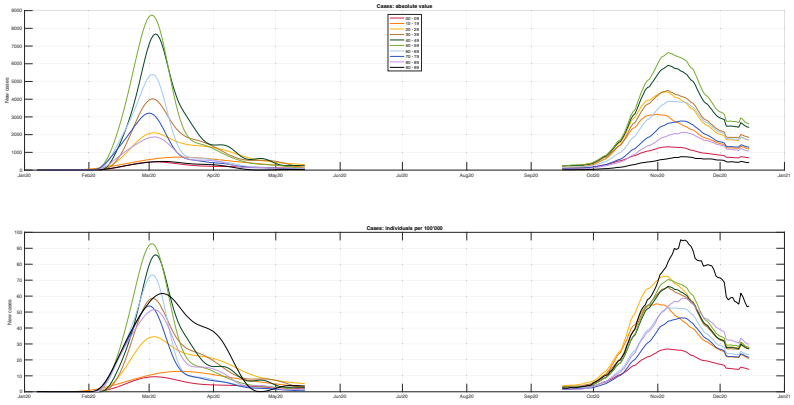


Figure 2.3: **Comparison between the reconstructed first-wave cases and official second-wave ones.** A simultaneous view of all age groups is provided, where the upper panel shows the absolute values and the lower panel displays the cases per 100,000 individuals. With regards to absolute numbers, the highest daily case count for both waves was among the age groups 40-59, with higher values seen in the first wave. When looking at cases per 100,000, some differences between the two waves should be noticed. During the first wave, the 40-49 and 50-59 age groups had the highest peaks, while in the second wave the 90-99 and 20-29 age groups had the highest peaks. The 90-99 age group showed the slowest decline in both waves. The 0-9 and 10-19 age groups had the lowest peaks in the first wave, but in the second wave the 0-9 group had a higher peak of 27 daily cases per 100,000. The peak value for the 10-19 group was early and exceeded 50 daily new cases, surpassing the 60-69 and 70-79 age groups.

The estimated new cases for all age groups can be summed to obtain a corrected version of the global time series of the new cases during the first wave, displayed in Fig 2.4, together with the official recorded new cases from

7 January to 15 December. Thanks to the reconstruction procedure, is appears that the official data severerely underestimated the number of new cases during the first wave, whose peak values appeared much smaller than that observed during the second one. As a matter of fact, the reconstructed peak value reached at the beginning of March is about equal to the 35,000 cases observed in mid October 2020. In spite of the similar peak values, the two waves have different slopes: both the rise and the decay of the new cases during the first wave are steeper. In the lower panel of Fig 2.4, it is shown that the hospital admissions (black dots) are well fitted by the admissions predicted by the convolution (red curve). The prediction of hospital admission for all age groups are displayed in 2.6, where the fit appears good except for youngest age groups (up to 39 year-old) during the late summer period, while for older age groups the fit appears satisfactory even during that period.

It is of some interest to assess the daily values of the underestimation factor, defined as the ratio of the reconstructed number of new cases to the number of official ones. It is seen that the daily underestimation factor was much larger than one, reaching a six-fold underestimation at the beginning of March, followed by a decrease until three and some further wandering around four.

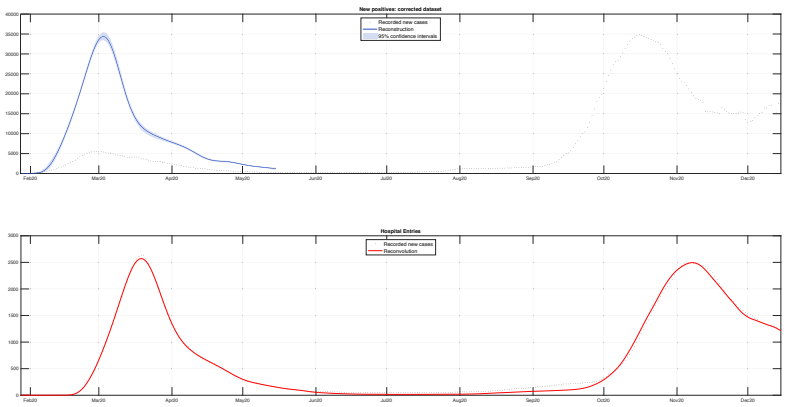


Figure 2.4: **Corrected and reconvoluted dataset.** The estimated new cases for all age groups can be summed to get a corrected version of the overall new cases during the first wave. The upper panel compares the corrected series (blue) with the officially recorded new cases (black dots) from 7 January to 15 December, 2020. The analysis confirms that the official data significantly underestimated the number of new cases during the first wave, whose peak value was much lower than those observed during the second wave. The rise and decay of new cases during the first wave were steeper compared to the second wave, even though the peak values were similar. The lower panel shows that the hospital admissions (black dots) match well with the admissions predicted by the convolution model (red).

2.4 Discussion

In this chapter we addressed the underreporting of Italian COVID-19 cases during the first wave, by exploiting a convolution model of hospital admissions.

2.4.1 Modeling hospital admissions and reconstructing first-wave cases

A first assumption underpinning our study is that the second wave was not subject to a massive underreporting as that occurred during the first one. A second assumption is that in both the first and second wave, for each age-class, the hospital admissions were explained with good approximation by a convolution model involving the time series of the new cases and a suitable kernel function. Under these assumptions, the new cases during the first wave were reconstructed by solving two inverse problem. The first one, formulated on the second wave data, yielded the kernel function of the hospital admission model as the solution of a parameter estimation problem involving hospital admissions and new cases. In the second step, the new cases during the first wave were obtained via deconvolution, using the first-wave hospitalizations and the kernel function estimated in the first step.

Concerning hospital admissions, the results showed that a simple 3-parameter kernel, consisting of a delayed and scaled exponential function, predicted remarkably well the hospital admissions of all age groups, thus validating the convolution model. The kernel parameter estimates shown in Table 2.1 seem meaningful: for instance, the kernel areas, representing the fraction of hospitalizations among the positive subjects, tend to increase with the age, ranging from a minimum of 0.7% (10-19 year-old) to a maximum of 30.7% (80-89 year-old). Also the delay between swab time and hospital admission as well as the time constant tend to be rather homogeneous across age groups. The negative delays, ranging from -6 to -10 days, are likely explained by different delays in the two pipelines of infection and hospitalization registries.

The solution of the second inverse problem yielded the reconstructed time series of daily new cases during the first wave. Differently from the first inverse problem, a simple parametric model describing the unknown signal was not available, so that the solution hinged on a regularization approach. The use of a penalty on the squared second differences proved effective.

As seen in Fig 2.2, it turned out that, for all ages, the reconstructed profiles of daily new cases were definitely higher than the official data, thus confirming the massive underreporting, highlighted by several authors, see e.g. [55], and confirmed by the serological survey published in June 2020 by the National Institute of Statistics (ISTAT) [73]. For all age groups the peak value was reached almost simultaneously in the first half of March 2020. Apart from younger (until 29 year-old) and older (from 80 year-old) subjects, the steep rise of cases was followed by a quick decline during the enforcement of NPIs.

Younger and older groups showed a somehow slower decline. In any case, the shape of the curves changed gradually between adjacent age groups, as seen in the surface plot in the lower right corner of Fig 2.2.

The lower panel of Fig 2.3, displaying new cases per 100,000, makes it possible to compare the impact of the first and second wave on the different age groups. A first observation is that during the first wave the time series of new cases exhibited high and narrow profiles, with an acute angle while the profiles during the second wave appeared definitely smoother. This change of behavior is consistent with the different severity of the containment measures that passed from strict national lockdown established by the Government on March 9, 2020, to a region-based and adjustable set of measures adopted during the second wave. Major differences between the waves were observed for the younger age groups until 29 year-old: during the most severe phase of the first wave they exhibited the lowest values, while in the second wave a different behavior was observed. In fact, the 0-9 age group, while still having the lowest peak, tripled its value. In the second wave, the 10-19 and 20-29 age groups, not only more than doubled their peak values but rised more quickly. These observations are consistent with early closure of schools and universities in the first wave [147], while closures were not so prompt during the second one. Another major difference between the two waves regards the oldest age group, i.e. 90-99. During the second wave the peak value was the highest one among all age groups and more than 1.5 times higher than the reconstructed peak value for the same age group during the first wave. However, our reconstruction is based on hospitalization data, so the profile of the first wave could be distorted if hospital bed eligibility changed from wave to wave for specific age groups, such as older individuals. [35].

2.4.2 Assessing uncertainty in estimated data

A crucial issue regards the assessment of uncertainty affecting our estimates. In order to account for the eteroskedasticity of observed hospital admissions, uncertainty propagation was assessed by means of a wild boostrap scheme. It was found that the 3-parameter kernel was estimated with a very good confidence, see Fig 2.1 for a visual appraisal, and Table 2.1 for the 95% confidence intervals of the kernel parameters, which were narrow for all age groups. As expected, wider ranges were observed for age groups with fewer hospitalizations.

2.4.3 Alternative approaches in selecting the model inputs

The main idea behind our reconstruction scheme is modeling the effect of contagions on hospitalization during the second wave and then inverting this causal model to obtain the contagions during the first wave from the observed effect. As observed effect we used the hospital admissions, but other choices

could be the ICU admissions or the deaths. There are several reasons suggesting that hospital admissions are to be preferred. First of all, the number of deaths and ICU admissions are definitely smaller, which, in view of the Poissonian nature of these occurrences yields noisier signals. Also, deaths may not be a completely reliable metric to compare the impact of the first and second waves of COVID-19 because the assumption of comparability may not be valid. In fact, not all deaths caused by COVID-19 during the first wave were accurately recorded as such [98]. Moreover, the average time elapsed from diagnosis to death is longer than that to hospitalization, so that the kernel function has a slower decay. As a consequence, the deconvolution problem becomes more ill-conditioned [45]. In an analogous way, using hospital occupancy as observed effect, in place of admissions, would entail a slower kernel function, worsening again the ill conditioning. Finally, concerning ICU admissions, it is also possible that during the first wave the availability of ICU beds reached saturation [35, 60, 118], which would again violate the comparability assumption. Of course, some degree of saturation during the first wave may have occurred also for hospital beds, which implies that our estimates are somehow conservative, in the sense that the reconstructed cases, though much greater than official ones, might still underestimate the actual ones.

2.4.4 Main findings

Our analysis led to three main findings. The first one regards the possibility of accurately predicting hospital admissions by means of a simple convolution model whose exponential kernel depends on three meaningful parameters, the scale factor, the delay, and the decay rate. From the scale factor and the decay rate, the percentage of positive subjects, for each age group, that are hospitalized can be derived as a secondary parameter. The prediction of the impact of COVID-19 on the healthcare system has emerged as a key issue since the early phase of the pandemic, see e.g. [114], where the fitting of ICU occupancy by means of an exponential curve was used to obtain a short term prediction of the strain on critical care facilities. Several authors derived simplified causal models that predict healthcare outcomes as delayed and scaled versions of the time series of the new cases [113, 142]. A more realistic modeling approach describes the outcomes as the convolution of the new cases with suitable kernel functions. This approach was pursued in [56] in order to assess the healthcare system costs associated with different vaccination rollout scenarios. Differently from the present chapter, where hospital admissions have been modeled, in [56], besides deaths, hospital and ICU occupancy were predicted. A possible future development may regard the identification of analogous models for the ICU admissions and the COVID-19 related deaths, as well as for hospital and ICU bed occupancy. Due to the convolution, these time series follow with varying delays the time profile of the new cases. Therefore the availability of reliable predictive models may help to take decisions and manage the

healthcare resources, see e.g. the role played by disease severity parameters in COVID-19 Pandemic Planning Scenarios [33].

It should be noted that these predictive models are distinct from and complementary to classical epidemiological models, which are geared toward describing the evolution of the number of susceptible, infected, and recovered people in a population where the virus is spreading. Given the structural uncertainty related to the difficulty of predicting policy decisions and their effects on the spreading of the virus, it is inevitable to propose a range of scenarios with varying degrees of severity. It is precisely this variety of scenarios, which included worst-case catastrophic outcomes, that is at the origin of some controversy in the media about the models developed by Imperial College in England [49] and the Kessler Foundation in Italy [52]. In contrast, as shown in this chapter, the day-to-day cause-and-effect relationship between numbers of infections and hospital admissions can be predicted with a high degree of reliability.

The second main finding regards the assessment of the underreporting ratio during the first wave, defined as the ratio between the actual infected subjects to the official cases. Our methodology makes it possible to equalize underreporting between the first and second wave. In particular, in order to restore comparability, the official cases of the first wave should be multiplied by a factor 4.5. Moreover, as seen in Fig 2.2, such ratio factor did not remain constant during the first wave. This means that, without some form of correction, the data from February to May 2020 are unusable for almost all epidemiological purposes. It is worth observing that, even after equalization, there remains some underdetection, given that a significant fraction of cases were not recorder also during the second wave. Differently from UK, where the ONS monitored a statistical sample throughout the pandemic, the Italian national institute of statistics, ISTAT, carried out only one survey, just after the first wave [73]. According to this survey, the actual cases were 6.2 times larger than the official ones. This means that, even after our reconstruction, a further 1.4 fold underdetection should be accounted for before arriving at the true number of cases. This suggests that also during the second wave at least one third of the cases were not recorded and the total number of cases during 2020 was about 4 million. In other words it is estimated that about 7% of the Italian population had been infected by the end of 2020 against the official 3.5%. This means that by the end of 2020, there were twice as many actual cases as officially registered ones. For the sake of comparison, in early December 2020, the ONS estimated that in England about 7 million people had antibodies, corresponding to around $13\%(10\% - 16\%)$ of the English population [126].

The third finding has to do with the comparison of age groups in the two waves. Indeed, it was found that some age groups exhibited notable differences. In particular, the 0-9 and 10-19 age groups had the lowest peaks in the first wave, below 10 and 15 cases per 100, 000, respectively. In the second

wave, instead, the group 0-9 had still the lowest peak value, but reached 27 daily cases per $100,000$, while the new cases of group 10-19, not only rose early but their peak value exceeded 50 daily new cases, more than the peak values of the 60-69 and 70-79 age groups. These differences may be worth some further investigation, being associated with school reopening in September 2020, compared with early closure in Northern regions during the first wave [147]. Another notable difference between the two waves concerns the time series of positive cases in the age group $90-99$: the peak value of daily cases per $100,000$ passed from 62 in the first wave to 95 in the second one. Before drawing any conclusions on the management of resthomes, it must be kept into account that our estimate may biased by a change in eligibility for hospital beds for infected elderly individuals, motivated by saturation that occurred during the first wave. The issue could be explored further by replicating our deconvolution approach using deaths in place of hospital admissions, under the assumption that deaths related to COVID-19 were recorded with uniform criteria during the two waves.

Additional information are displayed in 2.5 and 2.6. In particular, the comparison between impulse responses from different age and the reconvoluted dataset for each age group are provided.

2.4.5 Strenghts and limitations

This study introduces a method to correct the underreporting of new cases during the first COVID-19 wave, taking into account the differences between age groups, so that the temporal evolution of the underreporting factor can be estimated for each age group. An expensive serological survey was conducted in Italy at the end of the first wave to determine the accurate overall count of new cases, but it could not provide a daily estimate. The present study introduces a cost-effective method to correct underreporting, which not only complements serological tests but yields daily time series. Our findings rely on the assumption that the cause-effect links between new cases and hospital admissions of the first and second waves were comparable. This assumption is quite plausible, given that no variants emerged and no vaccines were introduced during the early months of the pandemic.

2.5 Conclusions

This chapter highlights the critical importance of accurate data in controlling a pandemic outbreak. The COVID-19 pandemic caught the healthcare and bureaucratic systems off-guard, resulting in severe underreporting of infections during the first wave in Italy. The lack of accurate data hampers the retrospective assessment of nonpharmacological interventions and the estimation and validation of epidemiological models. The approach proposed in this chapter, based on system identification and regularization for inverse problem,

Table 2.1: **Gains (percentage) and average time before hospitalization (days) with 2.5th and 97.5th percentiles, by age group.**

	Gain (%)	Delay (days)	Time constant (days)
$00-09$	1.76 (1.73, 1.79)	-10 (-10, -8)	4.3 (2.5, 5.2)
$10-19$	0.68 (0.67, 0.69)	-10 (-10, -9)	5.4 (3.9, 6.3)
$20-29$	1.44 (1.41, 1.45)	-10 (-10, -8)	5.3 (3.1, 6.2)
$30-39$	2.3 (2.26, 2.34)	-10 (-10, -9)	5.2 (3.8, 6.1)
$40-49$	3.19 (3.17, 3.2)	-10 (-10, -10)	5.5 (5.3, 5.8)
$50-59$	5.72 (5.68, 5.75)	-7 (-8, -6.5)	5.3 (4.7, 6.3)
$60-69$	11.98 (11.87, 12.05)	-7 (-8, -7)	5.7 (5.4, 6.9)
$70-79$	22.54 (22.39, 22.66)	-6 (-6.5, -6)	5.2 (4.9, 5.8)
$80-89$	30.62 (30.3, 30.83)	-8 (-9, -8)	8.2 (8, 9.2)
$90-99$	25.06 (24.66, 25.33)	-10 (-10, -9)	12.8 (11.1, 13.5)

The gains, expressed as a percentage, and the average times before hospitalization with their 2.5th and 97.5th percentiles, vary greatly depending on the age group and reflect the differences in illness severity across different age groups. Both parameters appear to be estimated with good precision.

Table 2.2: **Integrals of new positives (thousands) by ISS, our estimate and the respective underestimation factors.**

	ISS	Reconstructed	U.C.
$00-09$	2	23 (22, 24)	11.5
$10-19$	4	47 (45, 48)	11.75
$20-29$	13	102 (100, 105)	7.846
$30-39$	18	139 (136, 142)	7.722
$40-49$	31	221 (219, 223)	7.129
$50-59$	42	236 (233, 238)	5.619
$60-69$	31	132 (131, 133)	4.258
$70-79$	33	89 (88, 90)	2.697
$80-89$	41	63 (63, 64)	1.537
$90-99$	19	21 (21, 22)	1.105
All ages	234	1073 (1058 - 1089)	4.585

The first two columns of the table respectively display the sum of the new cases recorded by ISS and reconstructed by the regularized deconvolution model (with their 2.5-th and 97.5-th percentiles). The last column shows the underestimation factor.

is valuable under two respects. First, it offers a valuable tool for analyzing the human and health costs of COVID-19 over a given period. Moreover, it

enables a quantitative correction of underreporting. Our study shows that cause-and-effect relationships between numbers of infections and hospitalization admissions can be predicted with a high degree of reliability. This feature may prove critical to better respond to future epidemics in terms of safeguarding economic and health systems and implementing vaccine plans. In fact, the nature of the methodology is not strictly linked to the nature of COVID-19 or to one particular of its strains, although Alpha and Omicron variants are expected to require an update of the kernel function. One option is to conduct a piecewise-constant analysis, discontinuously studying each period associated with a particular variant. An alternative solution and a potential future research direction would be to replace the exponential parametric model with a time-varying impulse response. This approach can be modeled using identification techniques such as kernel methods [112], which would allow the model parameters to be time-dependent functions and not constrained to be constant over short periods. Finally, we highlight that correcting the underestimation of Italian cases during the first wave is valuable in several ways. According to the recorded data, Italy was one of the countries with the highest mortality rate, a fact that has prompted commentary and interpretation on the political and health management of epidemics. The availability of a more realistic estimate of actual cases is crucial to arrive at a fair assessment of the Italian response to the epidemic. Another point is the possibility of estimating the number of susceptible individuals at the beginning of the vaccination campaign. Again, without a reliable estimate, it is difficult or even impossible to quantify the impact of vaccines on overall mortality. In particular, we notice that some retrospective analysis were conducted by using official and underestimated data, e.g., [120], which makes problematic the assessment of public health interventions, including lockdowns and vaccination campaigns. For instance, without knowing the fraction of susceptible subjects it is impossible to run counterfactual simulations relative to NPIs and vaccination deployment. Our correction method is believed to prove useful also in connection with studies of the early and cryptic transmission [32] and those employing mobility data to study the spatial spread [6]: in both cases the reliability of the results would be compromised if heavily underreported cases were input in the models.

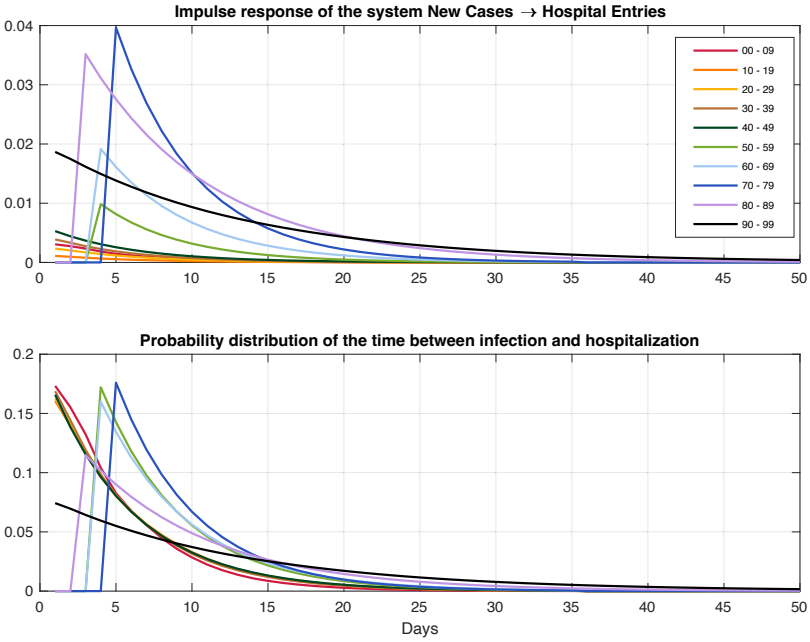


Figure 2.5: **Comparison between impulse responses from different age groups.** The upper panel shows the ten different impulse responses. The lower panel of the figure shows ten different probability distributions of elapsed time from positivity. Among these distributions, the age groups up to 49 years old share a similar pattern, with a shorter mean compared to the three age groups between 50 and 79 years old. The last two age groups exhibit the slowest decay rates.

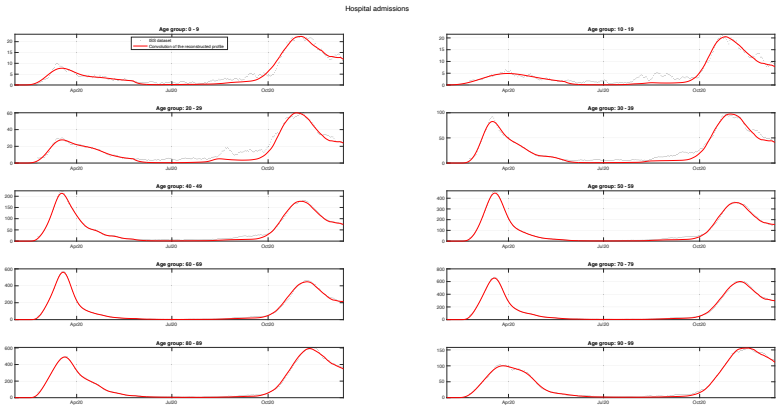


Figure 2.6: **Reconvoluted dataset for each age group.** The hospital entry predictions for all age groups are presented. The model fits well, except for the younger age groups that, however, account for a smaller proportion of hospitalizations. Conversely, the fit for older age groups is satisfactory during the whole period.

Part II

Discrete Optimization

3

Multi-Objective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks

Training neural networks using combinatorial optimization solvers has gained attention in recent years. In low-data settings, the use of state-of-the-art mixed integer linear programming solvers, for instance, has the potential to train exactly a neural network (NN), while avoiding intensive GPU-based training and hyper-parameter tuning, and simultaneously training and sparsifying the network. We study the case of few-bit discrete-valued neural networks, both Binarized Neural Networks (BNNs), whose values are restricted to ± 1, and Integer Neural Networks (INNs), whose values lie in a range $\{-P, \ldots, P\}$. Few-bit NNs receive increasing recognition due to their light-weight architecture and ability to run on low-power devices, for example being implemented using boolean operations. This paper proposes new methods to improve the training of BNNs and INNs. Our contribution is a multi-objective ensemble approach, based on training a single NN for each possible pair of classes and applying a majority voting scheme to predict the final output. Our approach results in the training of robust sparsified networks, whose output is not affected by small perturbations on the input, and whose number of active weights is as small as possible. We empirically compare this *BeMi* approach to the current state-of-the-art in solver-based NN training, and to traditional gradient-based training, focusing on BNN learning in few-shot contexts. We compare the benefits and drawbacks of INNs versus BNNs, bringing new light to the distribution of weights over the $\{-P, \ldots, P\}$ interval. Finally, we compare multi-objective versus single-objective training of INNs, showing that robustness and network simplicity can be acquired simultaneously, thus obtaining better test performances. While the previous state-of-the-art approaches achieve an average accuracy of 51.1% on the MNIST dataset, the *BeMi* ensemble approach achieves an average accuracy of 68.4% when trained with 10 images per class and 81.8% when trained with 40 images per class, whilst having up to 75.3% NN links removed.

3.1 Introduction

State-of-the-art deep neural networks (NNs) contain a huge number of neurons organized in many layers, and they require an immense amount of data

for training [82]. The training process is computationally demanding and is typically performed by stochastic gradient descent algorithms running on large GPU- or even TPU-based clusters. Whenever the trained (deep) neural network contains many neurons, also the network deployment is computationally demanding. However, in some real-life applications, extensive GPU-based training might be infeasible, training data might be scarce with only a few data points per class, or the hardware using the NN at inference time might have a limited computational power, as for instance, whenever the NN is executed on an industrial embedded system [23].

Binarized Neural Networks (BNNs) were introduced in [66] as a response to the challenge of running NNs on low-power devices. BNNs contain only binary weights and binary activation functions, and hence they can be implemented using only efficient bit-wise operations. However, the training of BNNs raises interesting challenges for gradient-based approaches due to their combinatorial structure. In previous works [136], it is shown that the training of a BNN can be performed by using combinatorial optimization solvers: a hybrid constraint programming (CP) and mixed integer programming (MIP) approach outperformed the stochastic gradient approach proposed by [66] if restricted to a few-shot-learning context [143]. Some of these works, e.g., [136], have been furthered by a number of authors more recently, as we survey in the next section.

Indeed, combinatorial approaches (principally MIP) for training neural networks, both discrete and continuous, have been employed in the literature, as demonstrated in subsequent works, e.g., [80, 110]. MIP optimization has been explored in the machine learning community, as in [65, 84, 152] for instance. Various architectures and activation functions have been utilized in these studies. Solver-based training has the advantage that, in principle, the optimal NN weights can be found for the training data and that network optimization (e.g., pruning) or adversarial hardening can be performed.

The main challenge in training a NN by an exact MIP-based approach is the limited amount of training data that can be used since, otherwise, the size of the optimization model explodes. In the recent work of [133], the combinatorial training idea was further extended to Integer-valued Neural Networks (INN). Exploiting the flexibility of MIP solvers, the authors were able to (i) minimize the number of neurons during training and (ii) increase the number of data points used during training by introducing a MIP batch training method.

We remark that training a NN with a MIP-based approach is more challenging than solving a verification problem, as in [2,51], even if the structure of the nonlinear constraints modelling the activation functions is similar. In NNs verification [79], the weights are given as input, while in MIP-based training, the weights are the *decision variables* that must be computed.

We note several lines of work aiming at producing compact and simple NNs that maintain acceptable accuracy, e.g., in terms of parameter pruning

[28,46,121,154], loss function improvement [131], gradient approximation [116] and network topology structure [87].

In the context of MIP-based training and optimization of NNs, this paper proposes new methods to improve the training of BNNs and INNs. In summary, our contributions are (i) the formulation of a MILP model with a multi-objective target that consists of already existing single-objective steps in a lexicorgraphic order, (ii) the implementation of an ensemble of few-bits NNs in which each of them is specialized in a specific classification task, (iii) the proposal of a voting scheme inspired by One-Versus-One (OVO) strategy, tailored specifically for the constructed ensemble of NNs. Our computational results using the MNIST and the Fashion-MNIST dataset show that the *BeMi* ensemble permits to use for training up to 40 data points per class, thanks to the fact that the OVO strategy results in smaller MILPs, reaching an average accuracy of 81.8% for MNIST and 70.7% for Fashion-MNIST. In addition, thanks to the multi-objective function that minimizes the number of links, i.e. the connections between different neurons, up to 75% of weights are set to zero for MNIST, and up to 48% for Fashion-MNIST. We also perform additional experiments on the Heart Disease dataset, reaching an average accuracy of 78.5%. A preliminary report of this work appeared at the LION'23 conference [17]. This paper better motivates and situates our approach in the state of the art, develops our ensemble approach for INNs (not only BNNs), presents more extensive and improved empirical results, and analyses the distribution of INN weights.

Outline. The remainder of this paper is as follows. Section 3.2 situates our work in the literature. Section 3.3 introduces the notation and defines the problem of training a single INN with the existing MIP-based methods. Section 3.4 presents the *BeMi* ensemble, the majority voting scheme, and the improved MILP model to train a single INN. Section 3.5 presents the computational results on the MNIST, Fashion-MNIST and Heart Disease datasets. Finally, Section 3.6 concludes the paper with a perspective on future work.

3.2 Related Works

Recently, there has been a growing research interest in studying the impact of machine learning on improving traditional operations research methods (e.g., see [15] and [31]), in designing integrated predictive and modelling frameworks, as in [16], or in embedding pre-trained machine learning model into an optimization problem (e.g., see [89,97,137]). In this work, we take a different perspective, and we study how an exact MILP solver can be used for training Machine Learning models, more specifically, to train (Binary or Integer) Neural Networks. In the following paragraphs, we first review two recent applications to MILP solvers in the context of neural networks (i.e., weight

pruning and NN verification), and later, we review the few works that tackled the challenge of training an NN using an exact solver.

MIP-based Neural Networks Pruning. Recent works have shown interesting results on pruning a trained neural network using an optimization approach based on the use of MIP solvers [28, 46, 59, 121, 154]. While pruning a trained neural network, the weights are fixed, and the optimization variables represent the decision to keep or remove an existing weight different from zero.

MIP-based Neural Networks Verification. Another successful application of exact solvers in the context of Neural Networks is for tackling the verification problem, that is, to verify under which conditions the accuracy of a given trained neural network does not deteriorate. In other words, in NN verification, the optimization problem consist in finding adversial examples using a minimal distortion of the input data. The use of MIP solvers for this application was pioneered in [135], and later studied in several papers, as for instance, [2, 24, 51, 134, 135]. For a broader discussion of the use of polyhedral approach to verification, see Section 4 in [67]. In NN verification, the weights of the NN are given as input, and the optimization problem consists of assessing how much the input can change without compromising the output of the network. Indeed, also verification is a different optimization problem than the exact training the we discuss in the following sections.

Exact training of Neural Networks. The utilization of MIP approaches for training neural networks has already been explored in the literature, primarily in the context of few-shot learning and NNs with low-bit parameters [133, 136]. One of the main advantages of these approaches is the ability to simultaneously train and optimize the network architecture. While the modelling that defines the structure of the neural network is rigid and heavily relies on the discrete nature of the parameters, the choice of the objective function provides more flexibility and allows for the optimization of various network characteristics. For instance, it enables minimizing the number of connections in a fixed architecture network, thereby promoting lightweight architectures.

From now on, by INN we indicate a general NN whose weights take value in the set $\{-P, \dots, P\}$, where integer $P \geq 1$. Notice this choice include the special case of BNNs ($P = 1$). When referring to an INN with $P > 1$, we will write *non-trivial* INN.

By incorporating the power of both CP and MIP, [136]'s study showcased the effectiveness of leveraging combinatorial optimization methods for training BNNs in a few-shot learning context. Two types of objective functions were selected to drive the optimization process. The first objective function aimed at promoting a lightweight architecture by minimizing the number of non-zero weights. This objective sought to reduce the overall complexity of the neural

network, allowing for efficient computation and resource utilization. Note that with this choice, pruning is not necessary: the possibility of setting a weight to zero is equivalent to removing the corresponding weight (i.e., connection). In contrast, the second objective function focused on enhancing the robustness of the network, particularly in the face of potential noise in the input data. Robustness here refers to the ability of the network to maintain stable and reliable performance even when the input exhibits variations or disturbances. [133]'s research extends [136]'s single-objective approach to the broader class of Integer Neural Networks (INNs). In addition to leveraging the objectives of architectural lightness and robustness, a novel objective aimed at maximizing the number of correctly classified training data instances is introduced. It is worth noting that this type of objective shares some similarities with the goals pursued by gradient descent methods, albeit with a distinct formulation. An interesting observation is that this objective formulation remains feasible in practice, ensuring that a valid solution can be obtained. We remark that both papers propose single-objective models that involve training a single neural network to approximate a multi-classification function.

Ensembles of neural networks are well-known to yield more stable predictions and demonstrate superior generalizability compared to single neural network models [145]. In this paper, we aim to redefine the concept of structured ensemble by composing our ensemble of several networks, each specialized in a distinct task. Given a classification task over k classes, the main idea is to train $\frac{k(k-1)}{2}$ INNs, where every single network learns to discriminate only between a given pair of classes. When a new data point (e.g., a new image) must be classified, it is first fed into the $\frac{k(k-1)}{2}$ trained INNs, and later, using a Condorcet-inspired majority voting scheme [153], the most frequent class is predicted as output. This method is similar to and generalizes the Support Vector Machine - One-Versus-One (SVM-OVO) approach [21], while it has not yet been applied within the context of neural networks, to the best of our knowledge.

For training every single INN, our approach extends the methods introduced in [136] and [133], described above.

3.3 Few-bit Neural Networks

In this section, we formally define a Binarized Neural Network (BNN) and an Integer Neural Network (INN) using the notation as in [136] and [133], while, in the next section, we show how to create a structured ensemble of INNs.

3.3.1 Binarized Neural Networks

The architecture of a BNN is defined by a set of layers $\mathcal{N} = \{N_0, N_1, \ldots, N_L\}$, where $N_l = \{1, \ldots, n_l\}$, and n_l is the number of neurons in the l-th layer.

Let the training set be $\mathcal{X} := \{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^t, y^t)\}$, such that $\boldsymbol{x}^i \in \mathbb{R}^{n_0}$ and $y^i \in \{-1, +1\}^{n_L}$ for every $i \in T = \{1, 2, \ldots, t\}$. The first layer N_0 corresponds to the size of the input data points $\boldsymbol{x}^k$. Regarding n_L, we make the following consideration. For a classification problem with $|\mathcal{I}|$ classes, we set $n_L := \lceil \log[2]|\mathcal{I}| \rceil$. For the case $|\mathcal{I}| = 2$, therefore, n_L will be equal to 1. This is consistent with binary classification problems, as the two classes can be represented as $+1$ and -1. When $|\mathcal{I}| = 4$, then n_L will be equal to 2, and the four classes will be represented by $(+1, +1), (+1, -1), (-1, +1)$, and $(-1, -1)$. When $|\mathcal{I}|$ is a power of 2, the procedure generalizes in an obvious manner. However, when the number of classes is not a power of 2, we still choose n_L as the nearest integer greater than or equal to the base-2 logarithm of that number, with the caveat that a single network may opt not to classify. For example, if $|\mathcal{I}| = 3$, then $n_L = 2$, and $(+1, +1)$ will be associated with the first class, $(+1, -1)$ will be associated with the second class, $(-1, +1)$ will be associated with the third class, while $(-1, -1)$ will be interpreted as 'unclassified'.

The link between neuron i in layer N_{l-1} and neuron j in layer N_l is modelled by weight $w_{ilj} \in \{-1, 0, +1\}$. Note that the binarized nature is encoded in the ± 1 weights, while when a weight is set to zero, the corresponding link is removed from the network. Hence, during training, we are also optimizing the architecture of the BNN.

The activation function is the binary function

$$\rho(x) := 2 \cdot \mathbb{1}(x \geq 0) - 1, \tag{3.1}$$

that is, a sign function reshaped such that it takes ± 1 values. Here the indicator function $\mathbb{1}(p)$ outputs $+1$ if proposition p is verified, and 0 otherwise. This choice for the activation function has been made in line with the literature [66].

In this paper, we aim to build different MILP models for the simultaneous training and optimization of a network architecture. To model the activation function (3.1) of the j-th neuron of layer N_l for data point $\boldsymbol{x}^k$, we introduce a binary variable $u^k_{lj} \in \{0, 1\}$ for the indicator function $\mathbb{1}(p)$. To re-scale the value of u^k_{lj} in $\{-1, +1\}$ and model the activation function value, we introduce the auxiliary variable $z^k_{lj} = (2u^k_{lj} - 1)$. For the first input layer, we set $z^k_{0j} = x^k_j$; for the last layer, we account in the loss function whether z^k_{Lh} is different from y^k_h. The definition of the activation function becomes

$$z^k_{lj} = \rho\left(\sum_{i \in N_{l-1}} z^k_{(l-1)i} w_{ilj}\right) = 2 \cdot \mathbb{1}\left(\sum_{i \in N_{l-1}} z^k_{(l-1)i} w_{ilj} \geq 0\right) - 1 = 2u^k_{lj} - 1.$$

Notice that the activation function at layer N_l gives a nonlinear combination of the output of the neurons in the previous layer N_{l-1} and the weights w_{ilj} between the two layers. Section 3.4.1 shows how to formulate this activation

function in terms of mixed integer linear constraints. We remark that the modelling we proposed has already been presented in the literature by [136].

The choice of a family of parameters $W := \{w_{ilj}\}_{l\in\{1,\dots,L\},i\in N_{l-1},j\in N_l}$ determines the function

$$f_W : \mathbb{R}^{n_0} \to \{\pm 1\}^{n_L}.$$

The training of a neural network is the process of computing the family W such that f_W classifies correctly both the given training data, that is, $f_W(\boldsymbol{x}^i) = y^i$ for $i = 1, \dots, t$, and new unlabelled testing data.

In the training of a BNN, we follow two machine learning principles for generalization: robustness and simplicity [136]. In doing so, we target two objectives: (i) the resulting function f_W should generalize from the input data and be *robust* to noise in the input data; (ii) the resulting network should be *simple*, that is, with the smallest number of non-zero weights that permit to achieve the best accuracy.

Regarding the robustness objective, there is argument that deep neural networks have inherent robustness because mini-batch stochastic gradient-based methods implicitly guide toward robust solutions [77,78,104]. However, as shown in [136], this is false for BNNs in a few-shot learning regime. On the contrary, MIP-based training with an appropriate objective function can generalize very well [133,136], but it does not apply to large training datasets, because the size of the MIP training model is proportional to the size of the training dataset.

One possible way to impose robustness in the context of few-shot learning is to maximize the margins of the neurons, that is, fixing one neuron, we aim at finding an ingoing weights configuration such that for every training input, the entry of the activation function evaluated at that neuron is confidently far away from the discontinuity point. Intuitively, neurons with larger margins require larger changes to their inputs and weights before changing their activation values. This choice is also motivated by recent works showing that margins are good predictors for the generalization of deep convolutional NNs [75].

Regarding the simplicity objective, a significant parameter is the number of connections [101]. The training algorithm should look for a NN fitting the training data while minimizing the number of non-zero weights. This approach can be interpreted as a simultaneous compression during training, and it has been already explored in recent works [115,122].

MIP-based BNN training. In [136], two different MIP models are introduced: the Max-Margin, which aims to train robust BNNs, and the Min-Weight, which aims to train simple BNNs. These two models are combined with a CP model into two hybrid methods `HW` and `HA` in order to obtain a feasible solution within a fixed time limit. [136] employ CP because their MIP models do not scale as the number of training data increases. We remark that in

that work, the two objectives, robustness and simplicity, are never optimized simultaneously.

Gradient-based BNN training. In [66], a gradient descent-based method is proposed, consisting of a local search that changes the weights to minimize a square hinge loss function. Note that a BNN trained with this approach only learns ± 1 weights. An extension of this method that exploits the same loss function but admits zero-value weights, called $\texttt{GD}_t$, is proposed in [136], to facilitate the comparison with the other approaches.

3.3.2 Integer Neural Networks

A more general discrete NN can be obtained when the weights of the network lie in the set $\{-P, -P+1, \ldots, -1, 0, 1, \ldots, P-1, P\}$, where P is a positive integer. The resulting network is called INN, and by letting $P = 1$, we obtain the BNN presented in the previous subsection. The activation function ρ and the binary variables u_{ij}^k are defined as above. The principles leading the training are again simplicity and robustness.

An apparent advantage of using more general integer neural networks lies in the fact that the parameters have increased flexibility while still maintaining their discrete nature. Additionally, by appropriately selecting the parameter P, one can determine the number of bits used for each parameter. For instance, $P = 1$ corresponds to 1-bit, $P = 3$ corresponds to 2-bit, $P = 8$ corresponds to 3-bit, and in general, $P = 2^{n-1}$ corresponds to n-bit.

MIP-based INN training. In [133], three MIP models are proposed in order to train INNs. The first model, Max-Correct, is based on the idea of maximizing the number of corrected predicted images; the second model, Min-Hinge, is inspired by the squared hinge loss (compare [66]); the last model, Sat-Margin, combines aspects of both the first two models. These three models always produce a feasible solution but use the margins only on the neurons of the last level, obtaining, hence, less robust NNs.

Relations to Quantized Neural Networks. By using an exact MIP solver for training Integer NNs, we are dealing directly with the problem of training a quantized neural network, where all the weights are restricted to take values over a small domain, as discussed above. For instance, as reviewed in [54], there is a growing trend in training NNs using floating point numbers in low precision , that is, using only as few as 8 bits per weight (see for example [7]). However, most of the work in the ML literature either focuses on the impact of low-precision arithmetic on the computation of the (stochastic) gradient and in the backpropagation algorithm or focuses on how to *quantize* a trained NN by minimizing the deterioration in the accuracy. In our work, we take a different perspective on quantization methods, since we do not rely

on a gradient-based method to train our INN, but we model and directly solve the problem of training the NN using only a restricted number of integer weights, which is called *Integer-only Quantization* in [54]. Moreover, by using an exact MIP solver, we can directly find the optimal weights of our (small) INN without running the risks to be trapped into a local minima as stochastic gradient-based methods.

3.4 The *BeMi* ensemble

This section first introduces a multi-objective model that allows a simultaneous training and optimization for an INN (Section 3.4.1), and then proposes a method for combining a set of neural networks for classification purposes (Section 3.4.2).

3.4.1 A multi-objective MILP model for training INNs

For ease of notation, we denote with $\mathcal{L} := \{1, \dots, L\}$ the set of layers and with $\mathcal{L}_2 := \{2, \dots, L\}$, $\mathcal{L}^{L-1} := \{1, \dots, L-1\}$ two of its subsets. We also denote with $\mathfrak{b} := \max_{k \in T, j \in N_0}\{|x_j^k|\}$ a bound on the values of the training data.

The multi-objective target. A few MIP models are proposed in the literature to train INNs efficiently. In this work, to train a single INN, we use a lexicographic multi-objective function that results in the sequential solution of three different state-of-the-art MIP models: the Sat-Margin (`SM`) described in [133], the Max-Margin (`MM`), and the Min-Weight (`MW`), both described in [136]. The first model `SM` maximizes the number of confidently correctly predicted data. The other two models, `MM` and `MW`, aim to train a INN following two principles: robustness and simplicity. Our model is based on a lexicographic multi-objective function: first, we train a INN with the model `SM`, which is fast to solve and always gives a feasible solution. Second, we use this solution as a warm start for the `MM` model, training the INN only with the images that `SM` correctly classified. Third, we fix the margins found with `MM`, and minimize the number of active weights with `MW`, finding the simplest INN with the robustness found by `MM`.

Problem variables. The critical part of our model is the formulation of the nonlinear activation function (3.1). We use an integer variable $w_{ilj} \in \{-P, -P+1, \dots, P\}$ to represent the weight of the connection between neuron $i \in N_{l-1}$ and neuron $j \in N_l$. Variable u_{lj}^k models the result of the indicator function $\mathbb{1}(p)$ that appears in the activation function $\rho(\cdot)$ for the training instance $\boldsymbol{x}^k$. The neuron activation is actually defined as $2u_{lj}^k - 1$. We introduce auxiliary variables c_{ilj}^k to represent the products $c_{ilj}^k = (2u_{lj}^k - 1)w_{ilj}$. Note that, while in the first layer, these variables share the same domain of the

inputs, from the second layer on, they take values in $\{-P, -P+1, \ldots, P\}$. Finally, the auxiliary variables $\hat{y}^k$ represent a predicted label for the input $\boldsymbol{x}^k$, and variable q_j^k are used to take into account the data points correctly classified.

The procedure is designed such that the parameter configuration obtained in the first step is used as a warm start for the (MM). Similarly, the solution of the second step is used as a warm start for the solver to solve (MW). In this case, the margins lose their nature as decision variables and become deterministic constants derived from the solution of the previous step.

Sat-Margin (SM) model. We first train our INN using the following SM model.

$$\max \quad \sum_{k\in T}\sum_{j\in N_L} q_j^k \tag{3.2a}$$

$$\text{s.t.} \quad q_j^k = 1 \implies \hat{y}_j^k \cdot y_j^k \geq \frac{1}{2} \qquad \forall j \in N_L, k \in T, \tag{3.2b}$$

$$q_j^k = 0 \implies \hat{y}_j^k \cdot y_j^k \leq \frac{1}{2} - \hat{\epsilon} \qquad \forall j \in N_L, k \in T, \tag{3.2c}$$

$$\hat{y}_j^k = \frac{2}{P \cdot (n_{L-1}+1)} \sum_{i\in N_{L-1}} c_{iLj}^k \qquad \forall j \in N_L, k \in T, \tag{3.2d}$$

$$u_{lj}^k = 1 \implies \sum_{i\in N_{l-1}} c_{ilj}^k \geq 0 \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \tag{3.2e}$$

$$u_{lj}^k = 0 \implies \sum_{i\in N_{l-1}} c_{ilj}^k \leq -\epsilon \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \tag{3.2f}$$

$$c_{i1j}^k = x_i^k \cdot w_{i1j} \qquad \forall i \in N_0, j \in N_1, k \in T, \tag{3.2g}$$

$$c_{ilj}^k = (2u_{(l-1)j}^k - 1)w_{ilj} \qquad \forall l \in \mathcal{L}_2, i \in N_{l-1}, j \in N_l, k \in T, \tag{3.2h}$$

$$q_j^k \in \{0,1\} \qquad \forall j \in N_L, k \in T, \tag{3.2i}$$

$$w_{ilj} \in \{-P, -P+1, \ldots, P\} \qquad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l, \tag{3.2j}$$

$$u_{lj}^k \in \{0,1\} \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in T, \tag{3.2k}$$

$$c^k_{i1j} \in [-P \cdot \mathfrak{b}, P \cdot \mathfrak{b}] \qquad \forall i \in N_0, j \in N_1, k \in T, \tag{3.2l}$$

$$c^k_{ilj} \in \{-P, -P+1, \dots, P\} \qquad \forall l \in \mathcal{L}_2, i \in N_{l-1}, j \in N_l, k \in T, \tag{3.2m}$$

with $\hat{\epsilon} \coloneqq \frac{\epsilon}{2P \cdot (n_{L-1}+1)}$. The objective function (3.2a) maximizes the number of data points that are correctly classified. Note that ϵ is a small quantity standardly used to model strict inequalities. The implication constraints (3.2b) and (3.2c) and constraints (3.2d) are used to link the output $\hat{y}^k_j$ with the corresponding variable q^k_j appearing in the objective function. The implication constraints (3.2e) and (3.2f) model the result of the indicator function for the k-th input data. The constraints (3.2g) and the bilinear constraints (3.2h) propagate the results of the activation functions within the neural network. We linearize all these constraints with standard big-M techniques [148].

The solution of model (3.2a)–(3.2m) gives us the solution vectors $\mathbf{c}_{\texttt{SM}}$, $\mathbf{u}_{\texttt{SM}}$, $\mathbf{w}_{\texttt{SM}}$, $\hat{\mathbf{y}}_{\texttt{SM}}$, $\mathbf{q}_{\texttt{SM}}$. We then define the set

$$\hat{T} = \{k \in T \mid q^k_{j\,\texttt{SM}} = 1,\ \forall j \in N_L\}, \tag{3.3}$$

of confidently correctly predicted images. We use these images as input for the next Max-Margin `MM`, and we use the vector of variables $\mathbf{c}_{\texttt{SM}}, \mathbf{u}_{\texttt{SM}}, \mathbf{w}_{\texttt{SM}}$ to warm start the solution of `MM`.

Max-Margin (MM) model. The second level of our lexicographic multi-objective model maximizes the overall margins of every single neuron activation, with the ultimate goal of training a robust INN. Starting from the model `SM`, we introduce the margin variables m_{lj}, and we introduce the following Max-Margin model.

$$\max \quad \sum_{l \in \mathcal{L}} \sum_{j \in N_l} m_{lj} \tag{3.4a}$$

$$\text{s.t.} \quad \text{(3.2g)–(3.2m)} \qquad \forall k \in \hat{T},$$

$$\sum_{i \in N_{L-1}} y^k_j c^k_{iLj} \geq m_{Lj} \qquad \forall j \in N_L, k \in \hat{T}, \tag{3.4b}$$

$$u^k_{lj} = 1 \implies \sum_{i \in N_{l-1}} c^k_{ilj} \geq m_{lj} \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \tag{3.4c}$$

$$u^k_{lj} = 0 \implies \sum_{i \in N_{l-1}} c^k_{ilj} \leq -m_{lj} \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \tag{3.4d}$$

$$m_{lj} \geq \epsilon \qquad \forall l \in \mathcal{L}, j \in N_l. \tag{3.4e}$$

Again, we can linearize constraints (3.4c) and (3.4d) with standard big-M constraints. This model gives us the solution vectors $\mathbf{c}_{\texttt{MM}}, \mathbf{u}_{\texttt{MM}}, \mathbf{w}_{\texttt{MM}}, \mathbf{m}_{\texttt{MM}}$. We

then evaluate $\mathbf{v}_{\texttt{MM}}$ as

$$v_{ilj_{\texttt{MM}}} = \begin{cases} 0 & \text{when} \quad w_{ilj_{\texttt{MM}}} = 0, \\ 1 & \text{otherwise,} \end{cases} \qquad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l. \tag{3.5}$$

Min-Weight (MW) model. The third level of our multi-objective function minimizes the overall number of non-zero weights, that is, the connections of the trained INN. We introduce the new auxiliary binary variable v_{ilj} to model the presence or absence of the link w_{ilj}. Starting from the solution of model `SM`, we fix $\hat{\mathbf{m}} = \mathbf{m}_{\texttt{MM}}$, and we pass the solution $\mathbf{c}_{\texttt{MM}}, \mathbf{u}_{\texttt{MM}}, \mathbf{w}_{\texttt{MM}}, \mathbf{v}_{\texttt{MM}}$ as a warm start to the following `MW` model:

$$\min \quad \sum_{l \in \mathcal{L}} \sum_{i \in N_{l-1}} \sum_{j \in N_l} v_{ilj} \tag{3.6a}$$

$$\text{s.t.} \quad (3.2g)\text{–}(3.2m) \qquad \forall k \in \hat{T},$$

$$\sum_{i \in N_{L-1}} y_j^k c_{iLj}^k \geq \hat{m}_{Lj} \qquad \forall j \in N_L, k \in \hat{T}, \tag{3.6b}$$

$$u_{lj}^k = 1 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \geq \hat{m}_{lj} \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \tag{3.6c}$$

$$u_{lj}^k = 0 \implies \sum_{i \in N_{l-1}} c_{ilj}^k \leq -\hat{m}_{lj} \qquad \forall l \in \mathcal{L}^{L-1}, j \in N_l, k \in \hat{T}, \tag{3.6d}$$

$$-v_{ilj} \cdot P \leq w_{ilj} \leq v_{ilj} \cdot P \qquad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l, \tag{3.6e}$$

$$v_{ilj} \in \{0, 1\} \qquad \forall l \in \mathcal{L}, i \in N_{l-1}, j \in N_l. \tag{3.6f}$$

Note that whenever v_{ilj} is equal to zero, the corresponding weight w_{ilj} is set to zero due to constraint (3.6e), and, hence, the corresponding link can be removed from the network.

Lexicographic multi-objective. By solving the three models `SM`, `MM`, and `MW`, sequentially, we first maximize the number of input data that is correctly classified, then we maximize the margin of every activation function, and finally, we minimize the number of non-zero weights. The solution of the decision variables w_{ilj} of the last model `MW` defines our classification function $f_W : \mathbb{R}^{n_0} \to \{\pm 1\}^{n_L}$.

3.4.2 The *BeMi* structure

Having explained the various MIP models of INNs, we next introduce our ensemble approach for MIP-based training of INNs.

Ensemble. Define $\mathcal{P} := \{\{i, j\} \text{ s.t. } i \neq j, i, j \in \mathcal{I}\}$ as the set of all the subsets of the set $\mathcal{I}$ that have cardinality 2, where $\mathcal{I}$ is the set of the classes

of the classification problem. Then our structured ensemble is constructed in the following way.

1. We train a INN denoted by $\mathcal{N}_{ij}$ for every $\{i, j\} \in \mathcal{P}$, i.e. for each possible pair of elements of $\mathcal{I}$.

2. When testing a data point, we feed it to our list of trained INNs obtaining a list of predicted labels, namely we obtain the predicted label $\mathfrak{e}_{ij}$ from the network $\mathcal{N}_{ij}$.

3. We then apply a majority voting system.

The idea behind this structured ensemble is that, given an input $\boldsymbol{x}^k$ labelled l $(= y^k)$, the input is fed into $\binom{n}{2}$ networks where $n-1$ of them are trained to recognize an input with label l. If all of the networks correctly classify the input $\boldsymbol{x}^k$ as l, then at most $n-2$ other networks can classify the input with a different label $l' \neq l$, and so the input is correctly labelled with the most occurring label l. With this approach, if we plan to use $r \in \mathbb{N}$ inputs for each label, we are feeding each of our INNs a total of $2 \cdot r$ inputs instead of feeding $n \cdot r$ inputs to a single large INN. Clearly, when training the networks $\mathcal{N}_{ij}$ and the network $\mathcal{N}_{ik}$, the inputs of the class i are the same, so we only need a total of r inputs for each class. When $n \gg 2$, it is much easier to train our structured ensemble of INNs rather than training one large INN because of the fact that the MILP model size depends linearly on the number of input data.

Majority voting system. After the training, we feed one input $\boldsymbol{x}^k$ to our list of INNs, and we need to elaborate on the set of outputs.
First, we define what a *Dominant label* is. For every $b \in \mathcal{I}$, we define

$$C_b = \{\{i, j\} \in \mathcal{P} \mid \mathfrak{e}_{ij} = b\},$$

and we say that a label b is a *dominant label* if $|C_b| \geq |C_l|$ for every $l \in \mathcal{I}$. We then define the set of dominant labels

$$\mathcal{D} \coloneqq \{b \in \mathcal{I} \mid b \text{ is a dominant label}\}.$$

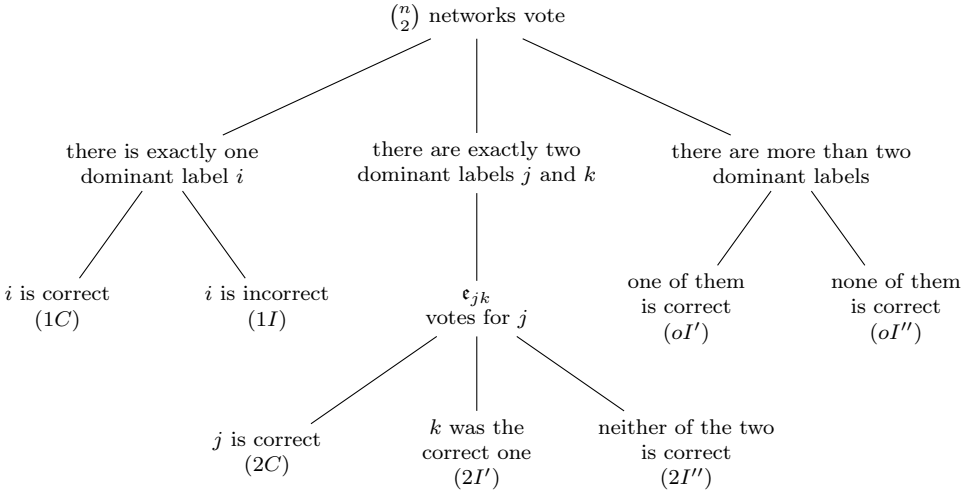
Using this definition, we can have three possible outcomes.

(a) There exists a label $i \in \mathcal{I}$ such that $\mathcal{D} = \{i\} \implies$ our input is labelled as i.

(b) There exist $j, k \in \mathcal{I}$ such that $\mathcal{D} = \{j, k\} \implies$ our input is labelled as $\mathfrak{e}_{jk}$.

(c) There exist more than two dominant labels $\implies$ our input is not classified.

While case (a) is straightforward, we can label our input even when we do not have a clear winner, that is, when we have trained a INN on the set of labels that are the most frequent (i.e., case (b)). Note that the proposed structured ensemble alongside its voting scheme can also be exploited for regular NNs. Secondly, we define what *Label Statuses* are. In our labelling system, when testing an input, seven different cases (herein called *label statuses*) can arise. The statuses names are of the form 'number of the dominant labels + fairness of the prediction'. The first parameter can be $1, 2$, or o, where o means 'other cases'. The fairness of the prediction is C when it is correct, or I when it is incorrect. The superscripts related to I' and I'' only distinguish between different cases. These cases are described through the following tree diagram.



The cases in which the classification algorithm classifies correctly are therefore only $(1C)$ and $(2C)$. Note that every input test will fall into exactly one label status.

Example 1. Let us take $\mathcal{I} = \{bird, cat, dog, frog\}$. Note that, in this case, we have to train $\binom{4}{2} = 6$ networks:

$$\mathcal{N}_{\{bird,\,cat\}},\ \mathcal{N}_{\{bird,\,dog\}},\ \mathcal{N}_{\{bird,\,frog\}},\ \mathcal{N}_{\{cat,\,dog\}},\ \mathcal{N}_{\{cat,\,frog\}},\ \mathcal{N}_{\{dog,\,frog\}},$$

the first one distinguishes between *bird* and *cat*, the second one between *bird* and *dog*, and so on. A first input could have the following predicted labels:

$$\begin{aligned} &\mathfrak{e}_{\{bird,\,cat\}} = bird, && \mathfrak{e}_{\{bird,\,dog\}} = bird, && \mathfrak{e}_{\{bird,\,frog\}} = frog, \\ &\mathfrak{e}_{\{cat,\,dog\}} = cat, && \mathfrak{e}_{\{cat,\,frog\}} = cat, && \mathfrak{e}_{\{dog,\,frog\}} = dog. \end{aligned}$$

We would then have

$$C_{bird} = \{\{bird,\, cat\}, \{bird,\, dog\}\}, \qquad C_{cat} = \{\{cat,\, dog\}, \{cat,\, frog\}\},$$

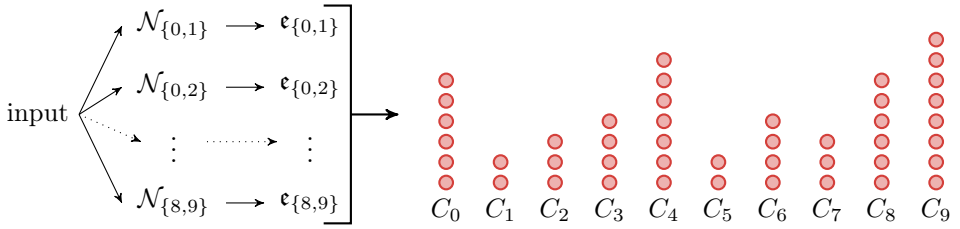
$$C_{dog} = \{\{dog,\, frog\}\}, \qquad C_{frog} = \{\{bird,\, frog\}\}.$$

In this case $\mathcal{D} = \{bird,\, cat\}$ because $|C_{bird}| = |C_{cat}| = 2 > 1 = |C_{dog}| = |C_{frog}|$ and we do not have a clear winner, but since $|\mathcal{D}| = 2$, we have trained a network that distinguishes between the two most voted labels, and so we use its output as our final predicted label, labelling our input as $\mathfrak{e}_{\{bird,\, cat\}} = bird$. If *bird* is the right label we are in label status $(2C)$, if the correct label is *cat*, we are in label status $(2I')$. Else we are in label status $(2I'')$.

Example 2. Let us take $\mathcal{I} = \{0, 1, \dots, 9\}$. Note that, in this case, we have to train $\binom{10}{2} = 45$ networks and that $|C_b| \leq 9$ for all $b \in \mathcal{I}$. Hence, an input could be labelled as follows:

$$\begin{aligned}
&C_0 = (\{0, i\})_{i=1,2,3,5,7,8}\ , && C_1 = (\{1, i\})_{i=5,6}\ , && C_2 = (\{2, i\})_{i=1,5,8}\ ,\\
&C_3 = (\{3, i\})_{i=1,2,4,5}\ , && C_4 = (\{4, i\})_{i=0,1,2,5,6,7,9}\ , && C_5 = (\{5, i\})_{i=6,7}\ ,\\
&C_6 = (\{6, i\})_{i=0,2,3,7}\ , && C_7 = (\{7, i\})_{i=1,2,3}\ , &&\\
&C_8 = (\{8, i\})_{i=1,3,4,5,6,7}\ , && C_9 = (\{9, i\})_{i=0,1,2,3,5,6,7,8}\ . &&
\end{aligned}$$

Visually, we can represent an input being labelled as above with the following scheme:



where we have omitted the name of each element of the set C_i for simplicity: for example, the dots above C_1 represent the sets $\{1, 5\}, \{1, 6\}$. Since $\mathcal{D} = \{9\}$, our input is labelled as 9. If 9 is the right label, we are in label status $(1C)$, if it is the wrong one, we are in label status $(1I)$. If instead $\hat{C}_j = C_j$, $j = 0, \dots, 7$, and

$$\hat{C}_8 = (\{8, i\})_{i=1,3,4,5,6,7,9}\ , \qquad \hat{C}_9 = (\{9, i\})_{i=0,1,2,3,5,6,7}\ ,$$

then $|\hat{\mathcal{D}}| = |\{4, 8, 9\}| = 3$, so that our input was labelled as -1. If the correct label is $4, 8$ or 9, we are in label status (mI'), else we are in label status (mI''). Lastly, if $\bar{C}_j = C_j$, $j \in \{0, 1, 2, 4, 5, 6, 7, 8\}$, and

$$\bar{C}_3 = (\{3, i\})_{i=1,2,4,5,9}\ , \qquad \bar{C}_9 = (\{9, i\})_{i=0,1,2,5,6,7,8}$$

then $|\bar{\mathcal{D}}| = |\{4, 9\}| = 2$ and since $\{4, 9\} \in \bar{C}_4$ our input is labelled as 4. If 4 is the correct label, we are in label status $(2C)$, if 9 is the correct label, we are

in label status ($2I'$), else we are in label status ($2I''$). Note that, for brevity, in this example we used the notation $(\{j,i\})_{i=i_1,\dots,i_n} = \{\{j,i_1\},\dots,\{j,i_n\}\}$, $j,i_1,\dots,i_n \in \{0,\dots,9\}$.

3.5 Empirical Study

Having introduced the *BeMi* approach, we now undertake a series of six experiments in order to explore the following questions:

- **Experiment 1:** What is the impact of a three-fold multi-objective model compared to a two-fold or single objective model? (Recall Section 3.4.)
- **Experiment 2:** How does the *BeMi* ensemble compare with the previous state-of-the-art MIP models for training BNNs in the context of few-shot learning?
- **Experiment 3:** How does the *BeMi* ensemble scale with the number of training images, considering two different types of BNNs?
- **Experiment 4:** How does the *BeMi* ensemble perform on various datasets, comparing the running time, the average gap to the optimal training MILP model, and the percentage of links removed?
- **Experiment 5:** What are the performance differences between a non-trivial INN and a BNN? Do INN exhibit particular weights distribution characteristics? A state-of-the-art comparison is also provided.
- **Experiment 6:** How does the *BeMi* ensemble perform on a continuous, low-dimension dataset, comparing BNNs and non-trivial INNs? Do INN exhibit the same weights distribution characteristics found in Experiment 5?

Datasets. Three datasets are adopted for the experiments. We use first the standard MNIST dataset [83] for a fair comparison with the literature, and second the larger Fashion-MNIST dataset [151]. For these two MNIST datasets, we test our results on 800 images for each class in order to have the same amount of test data for every class. Note that the MNIST dataset has 10 000 test data but they are not uniformly distributed over the 10 classes. For each experiment, we report the average over five different samples of images, i.e., we perform five different trainings and we report the average over them, while testing the same images. The images are sampled uniformly at random in order to avoid overlapping between different experiments. Beyond MNIST, we use the Heart Disease dataset [74] from the UCI repository. Table 3.1 summarizes the datasets.

Dataset	Number of classes	Input Dimension	Data Values	# Training Set	# Test Set
MNIST	10	28×28	Integers	60 000	10 000
Fashion-MNIST	10	28×28	Integers	60 000	10 000
Heart Disease	2	13	Continuous	$920 - x$	x

Table 3.1: Details of the different datasets exploited in the experiments.

Implementation details. As the solver we use Gurobi version 10.0.1 [62] to solve our MILP models. The solver parameters are left to the default values if not specified otherwise. Apart from the first experiment, where we chose to consider every model equally, the fraction of time given to each step of the multi-objective model has been chosen accordingly to the importance of finding a feasible and robust solution. All the MILP experiments were run on an HPC cluster running CentOS, using a single node per experiment. Each node has an Intel CPU with 8 physical cores working at 2.1 GHz, and 16 GB of RAM. In all of our experiments concerning integer-value datasets, we fix the value $\epsilon = 0.1$. Notice that, because of the integer nature of the weights, of the image of the activation function, and of the data, setting ϵ equal to any number smaller than 1 is equivalent. When using continuous-value datasets, we fix the value $\epsilon = 1 \cdot 10^{-6}$ in accordance with the default variable precision tolerance of the Gurobi MILP solver we will use. The source code is available on GitHub [18].

Time limit management. Concerning the time limits for the different optimization models, the following choices have been made. In Experiment 1 and 6, the time limit is equally distributed between the three models to have a fair comparison. In Experiment 2, 3, 4, and 5, the majority of the imposed time limit was given to the first two models. The first model ensures feasibility of the whole pipeline and maximises the number of correctly classified images in the training phase, and it was considered important in the context of few-shot learning, since we do not have lots of images as training inputs. The second model was given a bigger time limit too because preliminary results, also shown in previous works, highlight the fact that the maximization of the margin ensures a better test accuracy with respect to the minimization of the links. In addition to this, the overall time limits have been chosen based on two criteria. Where comparisons with the literature are made, the selection ensures a fair comparison. In the remaining cases, the choice of time limit has been empirical, aiming to highlight the algorithm's quality.

3.5.1 Experiment 1

The goal of the first experiment is to study the impact of the multi-objective model composed by `SM`, `MM` and `MW` with respect to the models composed by `SM` and `MM`, the one composed by by `SM` and `MW`, and only `SM`, respectively.

The results refer to a BNN specialised in distinguishing between digits 4 and 9 of the MNIST dataset. These two digits were chosen because their written form can be quite similar. Indeed, among all ten digits, 4 and 9 are very often mistaken for each other.

The NN architecture consists of two hidden layers and has $[784, 4, 4, 1]$ neurons. The architecture is chosen to mimic the one used in [136], that is $[784, 16, 16, 10]$, but with fewer neurons. The number of training images varies between 2, 6 and 10, while the test images are 800 per digit, so 1600 in total. The imposed time limit is 30 minutes, equally distributed in the steps of each model: 30 minutes for the `SM`, $15 + 15$ for the `SM`+`MW`, $15 + 15$ for the `SM`+`MM`, and $10 + 10 + 10$ for `SM`+`MM`+`MW`.

In Figure 3.1, the plot in the left compares the test accuracy of the four hierarchical models, showing how the `MM` model ensures an increase in accuracy, while the `MW` allows the network to be pruned without performance being affected. The plot in the right displays the percentages of non-zero weights of the three trained models. Note that in this case the training accuracy is always 100% and so we did not add it to the plot. Also, dotted lines represent the average accuracy obtained over 5 instances, while the shaded areas highlight the range between the minimum and maximum values. This will be the case for every other plot if not specified otherwise. The `MW` step allows the number of non-zero weights to drop without accuracy being affected, hence resulting in an effective pruning. This behaviour is also observed for other couples of digits, even the ones that are easier to distinguish, namely, 1 and 8.

Based on these reasons, for the remaining experiments, we will exclusively employ the model that incorporates all three steps.

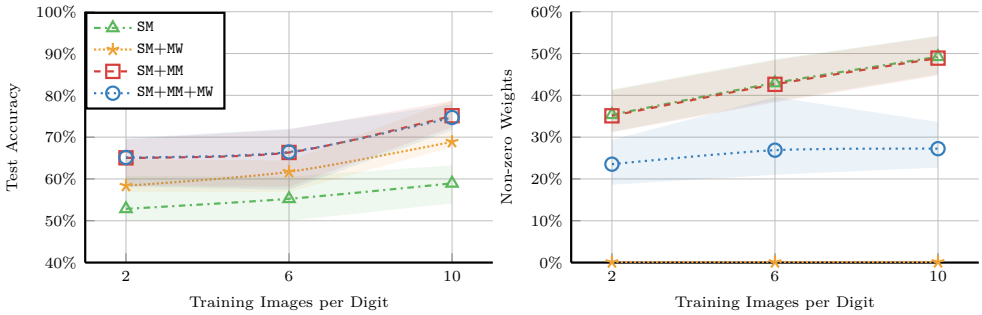


Figure 3.1: The `SM`+`MM`+`MW` achieve the same accuracy of the `SM`+`MM` model, outperforming the `SM` model, while having the smallest percentage of non-zero weights, a part from the `SM`+`MW` model, which has an almost-zero percentage of non-zero weights, but also a lower accuracy of the models that maximize the margins. The dotted lines represent the average accuracy obtained over 5 instances, while the shaded areas highlight the range between the minimum and maximum values.

3.5.2 Experiment 2

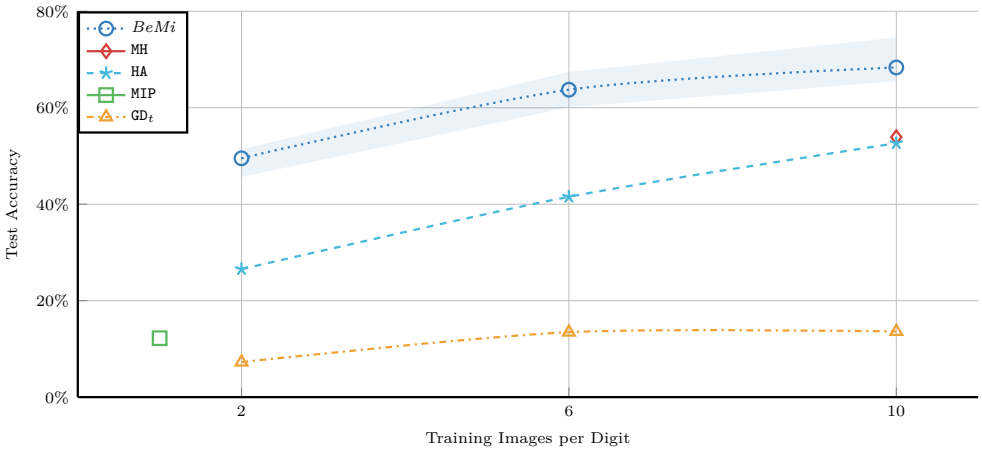


Figure 3.2: Comparison of published approaches vs *BeMi*, in terms of accuracy over the MNIST dataset using few-shot learning with 2, 6, and 10 images per digit.

The goal of the second experiment is to compare the *BeMi* ensemble with the following state-of-the-art methods: the pure MIP model in [136]; the hybrid CP–MIP model based on Max-Margin optimization (HA) [136]; the gradient-based method $\texttt{GD}_t$ introduced in [66] and adapted in [136] to deal with link removal; and the Min-hinge (MH) model proposed in [133]. For the comparison, we fix the setting of [136], which takes from the MNIST up to 10 images for each class, for a total of 100 training data points, and which uses a time limit of 7 200 seconds to solve their MIP training models.

In our experiments, we train the *BeMi* ensemble with 2, 6 and 10 samples for each digit. Since our ensemble has 45 BNNs, we leave for the training of each single BNN a maximum of 160 seconds (since $160 \cdot 45 = 7\,200$). In particular, we give a 75 seconds time limit to the solution of SM, 75 seconds to MM, and 10 seconds to MW. In all of our experiments, whenever the optimum is reached within the time limit, the remaining time is added to the time limit of the subsequent model. We remark that our networks could be trained in parallel, which would highly reduce the wall-clock runtime. For the sake the completeness, we note that we are using $45 \cdot (784 \cdot 4 + 4 \cdot 4 + 4 \cdot 1) = 142\,020$ parameters (all the weights of all the 45 BNNs) instead of the $784 \cdot 16 + 16 \cdot 16 + 16 \cdot 10 = 12\,960$ parameters used in [136] for a single large BNN. Note that, in this case, the dimension of the parameter space is $3^{12\,960} (\cong 10^{6\,183})$, while, in our case, it is $45 \cdot 3^{3\,156} (\cong 10^{1\,507})$. In the first case, the solver has to find an optimal solution between all the $10^{6\,183}$ different parameter configurations, while with the *BeMi* ensemble, the solver has to find 45 optimal solutions, each of which lives in a set of cardinality $10^{1\,507}$. This significantly improves the solver performances. We remark that a parameter configuration is given

by a weight assignment $\hat{W} = (\hat{w}_{ijl})_{ijl}$ since every other variable is uniquely determined by $\hat{W}$.

Figure 3.2 compares the results of our *BeMi* ensemble with the four other methods presented above. Note that the pure MIP model in [136] can handle a single image per class in the given time limit, and so only one point is reported, and note also that for the minimum hinge model `MH` presented in [133] only the experiment with 10 digits per class was performed. We report the best results reported in the original papers for these four methods.

The *BeMi* ensemble obtains an average accuracy of 68%, outperforms all other approaches when 2, 6 or 10 digits per class are used. Note that our method attains 100% accuracy on the training set, that is, the `SM` model correctly classifies all the images. In this case, the first model is not needed to ensure feasibility, but it serves mainly as a warm start for the `MM` model.

3.5.3 Experiment 3

The goal of the third experiment is to study how our approach scales with the number of data points (i.e., images) per class, and how it is affected by the architecture of the small BNNs within the *BeMi* ensemble. For the number of data points per class, we use $10, 20, 30, 40$ training images per digit. We use the layers $\mathcal{N}_a = [784, 4, 4, 1]$ and $\mathcal{N}_b = [784, 10, 3, 1]$ for the two architectures. While the first architecture is chosen as to be consistent with the previous experiments, the second one can be described as an integer approximation of $[N, \log_2 N, \log_2(\log_2 N), \log_2(\log_2(\log_2 N))]$. Herein, we refer to Experiments $3a$ and $3b$ as the two subsets of experiments related to the architectures $\mathcal{N}_a$ and $\mathcal{N}_b$. In both cases, we train each of our 45 BNNs with a time limit of 290s for model `SM`, 290s for `MM`, and 20s for `MW`, for a total of 600s (i.e., 10 minutes for each BNN).

In Figure 3.3, the plot in the left shows the results for Experiments $3a$ and $3b$: the dotted and dashed lines refer to the two average accuracies of the two architectures, while the coloured areas include all the accuracy values obtained as the training instances vary. The two architectures behave similarly and the best average accuracy exceeds 81%.

Table 3.2 reports the results for the *BeMi* ensemble where we distinguish among images classified as correct, wrong, or unclassified. These three conditions refer to different label statuses specified in Definition II: the correct labels are the sum of the statuses $(1C)$ and $(2C)$; the wrong labels of statuses $(2I'), (2I'')$, and $(1I)$; the unclassified labels $(n.l.)$ of (oI') and (oI''). Notice that the vast majority of the test images have only one dominant label, and so falls into statuses $(1C)$ or $(1I)$. The unclassified images are less than 2.31%.

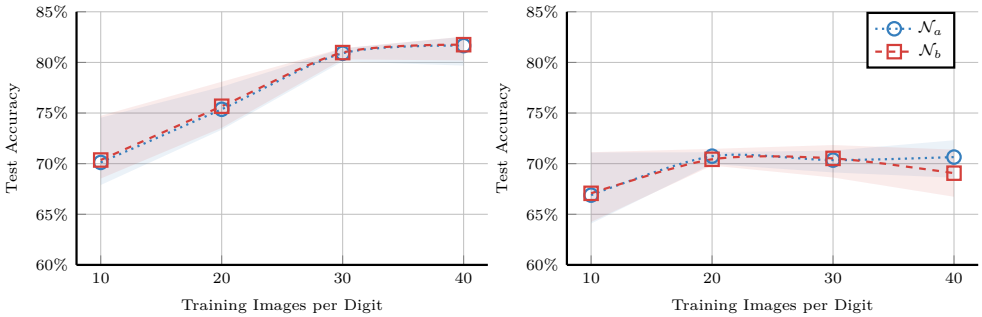


Figure 3.3: Average accuracy for the *BeMi* ensemble tested on two architectures, namely $\mathcal{N}_a = [784, 4, 4, 1]$ and $\mathcal{N}_b = [784, 10, 3, 1]$, using $10, 20, 30, 40$ images per class. The left plot refers to the MNIST dataset, while the right plot refers to the Fashion-MNIST dataset.

Table 3.2: Percentages of MNIST images classified as correct, wrong, or unclassified (*n.l.*), and of label statuses, for the architecture $\mathcal{N}_a = [784, 4, 4, 1]$. The vast majority of the test images have only one dominant label, and so falls into statuses $(1C)$ or $(1I)$. The unclassified images are less than 2.31%.

Images per class	Classification (%) correct	wrong	*n.l.*	Label status (%) $(1C)$	$(1I)$	$(2C)$	$(2I')$	$(2I'')$	(oI')	(oI'')
10	70.12	27.65	2.23	68.43	24.53	1.69	1.21	1.91	1.88	0.35
20	75.37	22.32	2.31	73.79	19.33	1.58	1.39	1.60	2.02	0.29
30	80.90	17.46	1.64	79.64	15.01	1.26	1.25	1.20	1.46	0.18
40	81.66	16.68	1.66	80.34	14.36	1.32	1.09	1.23	1.45	0.21

Table 3.3: Aggregate results for Experiments 2 and 3: the 4-th column reports the runtime to solve the first model `SM`; *Gap (%)* refers to the mean and maximum percentage gap at the second MILP model `MM`; *Links (%)* is the percentage of non-zero weights after the solution of models `MM` and `MW`; *Active links* is the total number of non-zero weights after the solution of model `MW`, as always averaged over five instances.

Dataset	Layers	Total links	Images per class	Model `SM` time (s)	Gap (%) mean	Gap (%) max	Links (%) `(MM)`	Links (%) `(MW)`	Active links
MNIST	N_a	3156	10	3.00	12.06	20.70	49.13	29.21	921.80
			20	6.47	14.81	22.18	54.70	28.41	896.60
			30	10.60	16.04	24.08	56.44	30.46	961.40
			40	15.04	15.98	26.22	57.92	29.27	923.80
	N_b	7873	10	6.04	4.52	7.37	49.28	24.72	1946.20
			20	15.01	5.46	8.40	54.97	24.40	1921.00
			30	22.68	5.92	11.00	56.73	26.96	2122.60
			40	33.97	6.21	20.87	58.29	24.66	1941.40
F-MNIST	N_a	3156	10	4.83	13.34	26.48	87.75	58.76	1854.40
			20	9.77	14.05	28.91	90.73	59.97	1892.60
			30	36.10	19.95	136.33	92.41	58.12	1834.20
			40	72.15	30.36	333.70	93.57	59.46	1876.60
	N_b	7873	10	11.42	4.87	9.14	87.90	51.57	4060.20
			20	21.46	5.11	9.86	91.05	52.29	4116.80
			30	35.12	6.30	40.07	92.78	52.62	4142.80
			40	52.37	8.99	56.14	94.01	53.38	4202.60

3.5.4 Experiment 4

The goal of the fourth experiment is to revisit the questions of Experiments 3*a* and 3*b* with the two architectures $\mathcal{N}_a$ and $\mathcal{N}_b$, using the more challenging Fashion-MNIST dataset.

In Figure 3.3, the plot in the right shows the results of Experiments 3*a* and 3*b*. As in Figure 3.2, the dotted and dashed lines represent the average percentages of correctly classified images, while the coloured areas include all accuracy values obtained as the instances vary. For the Fashion-MNIST, the best average accuracy exceeds 70%.

Table 3.3 reports detailed results for all Experiments 2 and 3. The first two columns give the dataset and the architecture, the third column reports the total number of links, i.e., the total number w variables, for each architecture, and the fourth column specifies the number of images per digit used during training. The 5-th column reports the runtime for solving model `SM`. Note that the time limit is 290 seconds; hence, we solve exactly the first model, consistently achieving a training accuracy of 100%. The remaining five columns give: *Gap (%)* refers to the mean and maximum percentage gap at the second MILP model `(MM)` of our lexicographic multi-objective model, as reported by the Gurobi `MIPgap` attribute; *Links (%)* is the percentage of non-zero weights after the solution of the second model `MM`, and after the solution of the last model `MW`. *Active links* is the total number of non-zero weights after the solution of the last model `MW`. The results show that the runtime and the gap increase with the size of the input set. However, for the percentage of removed links, there is a significant difference between the two datasets: for MNIST, our third model `MW` removes around 70% of the links, while for the Fashion-MNIST, it removes around 50% of the links. Note that in both cases, these significant reductions show how our model is also optimizing the BNN architecture. Furthermore, note that even if the accuracy of the two architectures is comparable, the total number of non-zero weights of $\mathcal{N}_a$ is about half the number of non-zero weights of $\mathcal{N}_b$.

3.5.5 Experiment 5

The goal of the fifth experiment is to compare the performances of BNNs and non-trivial INNs. The results refer to five different runs of an INN of architecture $[784, 4, 4, 1]$ specialised in distinguishing between digits 4 and 9 of the MNIST dataset. The number of training images varies between 2, 6, 10, 20, 30, and 40, while the test images are 800 per digit, so 1600 in total. The imposed time limit is 290s + 290s + 20s.

As Figure 3.4 shows, different INNs are comparable not only in the average accuracy, represented by the dotted lines, but also in the maximum and minimum accuracy, reported by the shaded areas.

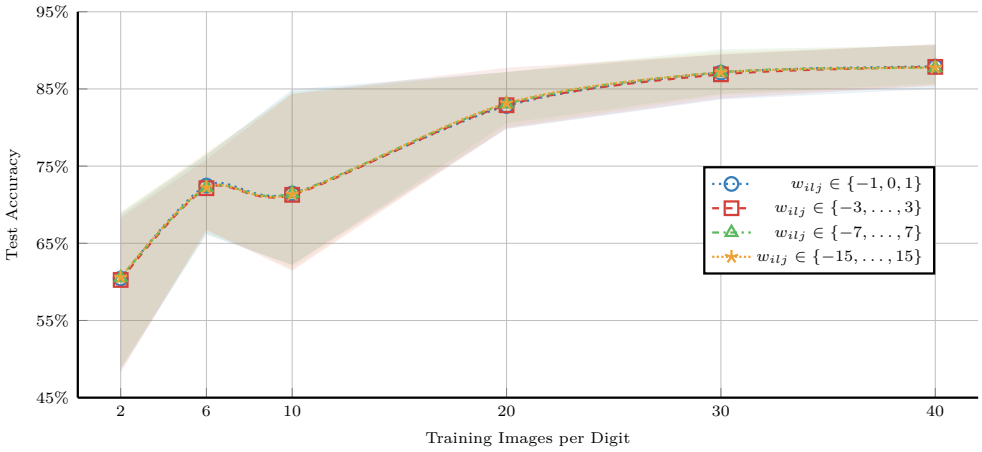


Figure 3.4: Comparison of accuracy for different values of P. Note how using an exponentially larger research space, namely, using values of P greater than 1, does not improve the accuracy.

Table 3.4: Weights distributions of the INNs whose accuracy is depicted in Figure 3.4. The column $w = -P$ indicates the percentage of weights that are equal to $-P$, and so on. The networks have different values but similar extremal weights distributions, where less than 2% of the weights attain a value that is not P, $-P$, or zero, indicated as *others*

value of P	Images per class	$w = -P$	$w = 0$	$w = P$	*others*
1	2	4.06	73.66	22.28	-
	6	4.75	73.70	21.55	-
	10	5.15	74.02	20.83	-
	20	7.12	61.31	31.57	-
	30	6.21	74.89	18.90	-
	40	13.16	64.98	21.86	-
3	2	3.86	75.91	20.15	0.08
	6	4.60	73.46	21.63	0.31
	10	6.25	73.50	19.72	0.53
	20	12.48	69.58	17.19	0.75
	30	13.06	74.00	11.89	1.05
	40	13.99	65.51	19.31	1.19
7	2	3.90	75.89	20.10	0.11
	6	4.46	73.40	21.76	0.38
	10	8.82	68.00	22.60	0.58
	20	12.46	65.56	21.06	0.92
	30	14.99	73.82	9.95	1.24
	40	18.05	70.90	9.40	1.65
15	2	3.69	75.89	20.30	0.12
	6	4.68	73.31	21.64	0.37
	10	5.53	73.28	20.61	0.58
	20	7.43	69.34	22.19	1.04
	30	13.76	67.64	17.23	1.37
	40	17.34	70.86	10.13	1.67

In order to study why different values of P lead to comparable accuracy,

Table 3.5: Average accuracy and weight distribution for the Heart Disease dataset. The column Average accuracy depicts the average percentage of correctly classified test data. The column $w = -P$ indicates the percentage of weights that are equal to $-P$, and so on. For each value of P, the best result in terms of accuracy with respect to the model is written in bold. Notice that in the majority of the cases, the multi-objective function performs better than the single-objective one. Notice also that even if the percentage of the non-zero and non-extremal weights, indicated as *others*, is higher than the one obtained with the MNIST dataset, the distribution is still not uniform.

Models	value of P	Average accuracy	$w = -P$	$w = 0$	$w = P$	*others*
SM	1	74.00	35.43	45.14	19.43	-
	3	75.00	21.43	27.43	16.57	34.57
	7	**77.00**	27.43	14.86	16.86	40.85
	15	77.00	18.86	11.71	12.86	56.57
SM + MM	1	75.00	14.57	18.29	67.14	-
	3	75.50	10.57	13.43	63.14	12.86
	7	73.50	12.29	9.43	52.57	25.71
	15	**78.50**	11.71	5.71	46.86	35.72
SM + MM + MW	1	**75.50**	9.43	27.43	63.14	-
	3	**76.00**	5.43	25.43	57.43	11.71
	7	73.50	7.43	18.57	49.71	24.29
	15	**78.50**	8.57	13.43	45.71	32.29

we report the weights distributions of the INNs whose accuracy is depicted in Figure 3.4. Table 3.4 highlights not only the role of the MW model but also the extreme-valued nature of the distributions. In fact, it can be seen that apart from the percentage of weights set to zero, the vast majority of the remaining weights have a value of either P or $-P$, with less than 1.67% of the weights attaining one of the other intermediate values. We remark that this type of phenomenon is recurrent in the literature under the name of magnitude pruning, see [22, 63, 102]. In our setting, such a phenomenon occurs spontaneously.

3.5.6 Experiment 6

The goal of the sixth experiment is to study the impact of the multi-objective model and the weight distributions over a different dataset, namely the Heart Disease Dataset by [74].

Table 3.5 reports the average accuracy and weights distributions of this experiment. The average was performed over 5 instances, and for each instance 200 data samples were used, where 80% was used for the training and 20% was used for the test. All the networks have the same architecture, namely $[13, 5, 1]$, and each network's imposed time limit is 60 minutes, equally distributed in the steps of each model: 60 minutes for the SM, $30 + 30$ for the SM+MM, and $20 + 20 + 20$ for SM+MM+MW.

3.6 Conclusion and Future Work

This chapter introduced the *BeMi* ensemble, a structured architecture of INNs for classification tasks. Each network specializes in distinguishing between pairs of classes and combines different approaches already existing in the literature to preserve feasibility while being robust and simple. These features and the nature of the parameters are critical to enabling neural networks to run on low-power devices. In particular, binarized NNs can be implemented using Boolean operations and do not need GPUs to run.

The output of the *BeMi* ensemble is chosen by a majority voting system that generalizes the one-versus-one scheme. Notice that the *BeMi* ensemble is a general architecture that could be employed using other types of neural networks.

An interesting conclusion from our computational experiments is a counter-intuitive result: that the greater flexibility in the search space of INNs does not necessarily result in better classification accuracy compared to BNNs. We find it noteworthy that our computational evidence supports the idea that simpler BNNs are either superior or equal to INNs in terms of accuracy.

A current limitation of our approach is the strong dependence on the randomly sampled data used for training. In future work, we plan to improve the training data selection by using a k-medoids approach, dividing all images of the same class into disjoint non-empty subsets and consider their centroids as training data. This approach should mitigate the dependency on the sampled training data points.

Second, we also plan to better investigate the scalability of our method with respect to the number of classes of the classification problem training fewer BNNs, namely, one for every $\mathcal{J} \in \mathcal{Q} \subset \mathcal{P}$, with $|\mathcal{Q}| \ll |\mathcal{P}|$. Besides the datasets we exploited, in future, we intend to investigate datasets more appropriate for the task of few-shot learning [27].

Third, another possible future research direction is to exploit the generalisation of our ensemble, allowing to have networks which distinguishes between m classes instead of 2, where the total number of classes of the problem is $n \gg m$.

Fourth, an interesting future research direction regards stochastic/robust optimization and scenario generation, in the following sense. In the case of the MNIST/FashionMNIST for example, the images can be seen as samples of many unknown probability distributions, one for each class: if one would like to train a neural network with a MIP, using few images, the selection of these samples with which the training is performed is crucial, and so the study of this problem from a stochastic point of view could lead to interesting results.

4

The Mobile Positioning Problem as an unassigned Distance Geometry Problem

The Mobile Positioning Problem is at the intersection of technology and communication networks. It asks for the 2-D geographical position of mobile devices, such as cell phones or GPS units, given a subset of their pairwise distances. In this chapter, we see this problem as a special case of the Manhattan-norm unassigned Distance Geometry Problem, which we formulate by means of mixed-integer linear programming. Computational results within different time-limits are provided, and mean and largest distance errors analyzed. Possible research directions are hence outlined.

4.1 The mobile positioning problem

The Mobile Positioning Problem (MPP) arises in the field of technology and communication networks. Its objective is to determine the position of a set of mobile devices, such as cell phone or GPS, within a specified 2-D geographical area [119], given a subset of their pairwise distances. The positions can be expressed in geographic coordinates (latitude and longitude) or other reference systems. The MPP is fundamental to many applications, including GPS navigation, location-based apps, emergency services, and more. A well studied version of MPP considers the presence of base stations, i.e. stations of known position than can compute the distances between themselves and the mobile devices [37]. In this chapter, we will consider instead the version of the problem where the number of points (i.e., mobile devices) and the distances between them are known, but the specific pairings of points and distances is unknown. This could occur if some distance measurements are incomplete or lack specific associations between landmarks or sensors. In the line-of-sight scenario, where there are no obstacles between both mobile devices, the unobstructed distance is simply the Euclidean distance, i.e. $||x||_2 = \sqrt{x_1^2 + x_2^2}$. However, in non-line-of-sight conditions, the unobstructed distance is greater or equal. More specifically, if the obstacles are rectangular and have longer sides than the widths of the streets, the unobstructed distance can be approximated by the Manhattan distance, i.e. $||x||_1 = |x_1| + |x_2|$. An example of the prototype of the geographical area and the physical meaning of the two distances can be found in Fig 4.1. This chapter will focus on this last case, in

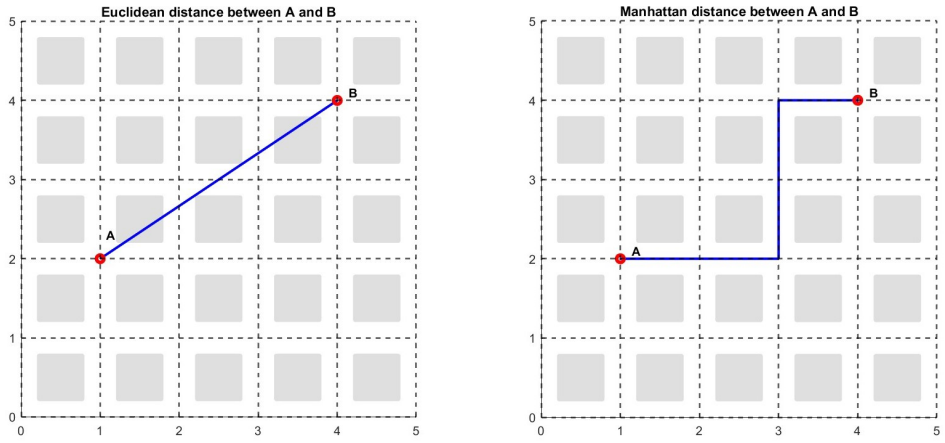


Figure 4.1: **Comparison between Euclidean norm and Manhattan norm.** In a geographical area with grid-based geometry, it is not suitable to compute distances by the Euclidean norm, since mobile devices can move only on the grid. Hence, it is appropriate to compute distances by the Manhattan norm.

which the territory has a grid-based geometry.

A toy problem. In a Manhattan-like area, four mobile devices $\{A, B, C, D\}$ at a fixed instant have the following pairwise distance values

$$\{2, 2, 2, 2, 4, 4\} \tag{4.1}$$

but we are not aware of which distance is associated to which pair of devices. In general, this problem does not warrant solution uniqueness: two possible solutions are indeed depicted in Fig 4.2. In the first one, the four distance values equal to 2 refer to the length of the square edges $\overline{AB}, \overline{BC}, \overline{CD}, \overline{DA}$, while the two distance values equal to 4 refer to the length of the diagonals $\overline{AC}, \overline{BD}$. In the second one, the distance values $\overline{A'C'}, \overline{A'D'}$ are equal to 4, while the remaining ones are equal to 2. The main difference between the two proposed solutions is the assignment of distance values to pairs of mobile devices.

4.2 Assigned and unassigned Distance Geometry Problems

The toy problem described in the previous section is a particular case of the unassigned Distance Geometry Problem (uDGP), a variant of the Distance Geometry Problem (DGP) [86]. In this section, we will summarize some relevant aspects of the DGP and uDGP.

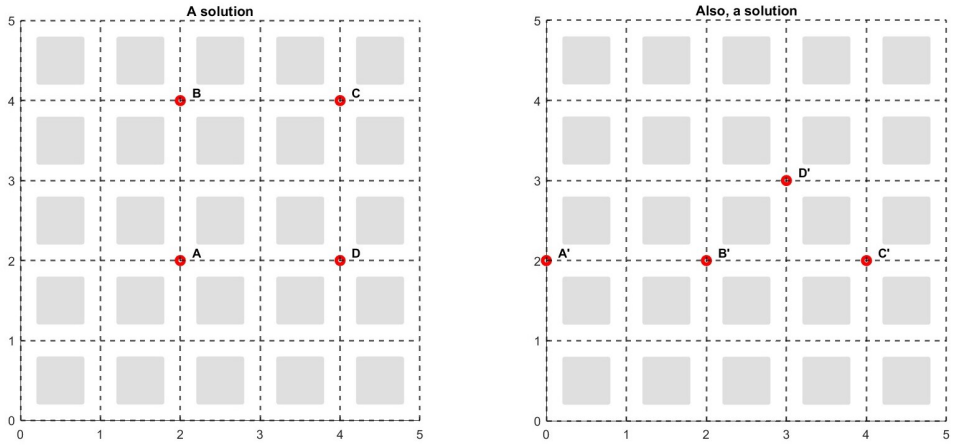


Figure 4.2: **Different solutions are possible.** Different assignments between the distances and couple of points have been made. In both cases, the set of reciprocal distances between the mobile devices is {2,2,2,2,4,4}.

The (assigned) Distance Geometry Problem

Given a positive integer K, and a simple undirected edge-weighted graph $\mathcal{G} = (V, \mathcal{E}, d)$, with $d : \mathcal{E} \to \mathbb{R}^+$, the DGP asks to determine if there are position vectors $\{x_u\}_{u \in V} \subseteq \mathbb{R}^K$ such that $||x_u - x_v|| = d_{uv}$ for all $\{u, v\} \in \mathcal{E}$. From now on, for the sake of notation convenience, we will represent the edge {u, v} as the ordered pair (u, v), subject to the constraint $1 \leq u < v \leq N$.

In most application settings, $K \in \{1, 2, 3\}$. For example,

- in one dimension, it is possible to estimate and exchange the time difference between specific clocks in specific network synchronization protocols. The problem here is the determination of the absolute times and the solution consists in a sequence of scalars [123].
- in wireless networks, devices typically move within a two-dimensional space. Certain pairwise distances, particularly those that are close enough, can be approximated by assessing the battery power consumption during peer-to-peer communication. The network provider's primary concern is determining the precise 2-D locations of these devices [48].
- three-dimensional protein structures and nanostructures can be achieved by using distance data between pairs of atoms [39], which is supplied by experimental methods like nuclear magnetic resonance [86] and the pair distribution function technique [76], respectively.

Concerning the norm, the choice depends on the application of interest. As previously emphasized, the Manhattan norm is an appropriate choice for the MPP. Conversely, for various other engineering and scientific applications, the Euclidean norm is better suited [85].

The unassigned Distance Geometry Problem

Studies on DGP mostly focus on the case where the graph structure is known and the provided distances are uniquely associated to the graph edges. However, this is not the case of some practical applications, such as the molecular reconstructions via the pair-distribution function (PDF) method [20], or the problem presented at the beginning of the chapter, i.e., the MPP. In this case, the data consist of a set of vertices $V = \{1, \dots, N\}$ and a multiset of distances $D = \{d_1, ..., d_M\}$ with no particular instruction on which distance is associated with which edge. In order for the problem to be feasible, it is important to note that the number of weights must be less than or equal to the number of possible edges, i.e., $M \leq N(N-1)/2$. The uDGP problem is two-fold: one has to find an assignment between d_k's and the edges, and to determine a position in $\mathbb{R}^K$ for the vertices [41]. In other words, we want to find an injection $\alpha : D \to \{(u, v) \,|\, 1 \leq u < v \leq N\}$ and $\{(x_u)_{u \in V}\} \subseteq \mathbb{R}^K$ such that

$$||x_u - x_v|| = \alpha^{-1}(u, v) \;\; \forall (u, v) \in Im(\alpha) =: \mathcal{E} \tag{4.2}$$

where $\mathcal{E}$ completes the graph structure. In the case of the MPP, the vertices represent the mobile devices, and the unassociated weights correspond to the distances between them.

In the following subsection we will describe in a more detailed way which is our contribution.

4.3 A MILP formulation for the uDGP

This section will focus on a Mixed-Integer Linear Programming (MILP) formulation for the uDGP with the Manhattan norm. While this approach has already been implemented in literature for the DGP, e.g., in [42], we are unaware of any work in this field covering the uDGP case.

In our formulation, the target is to minimize the error made in embedding points in $\mathbb{R}^K$ by consistent assignment, which is also an unknown of the problem. Constraints, on the other hand, define the structure of the problem or are algebraic manipulations with the aim of linearizing non-linear constraints.

The target is

$$\text{minimize} \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{k=1}^{M} a_{uv}^k \,|\, ||x_u - x_v||_1 - d_k \,|, \tag{4.3}$$

where the decision variable a_{uv}^k is such that

$$a_{uv}^k = 1 \iff d_k \text{ is assigned to } (u, v), \text{ and 0 otherwise.}$$

As one can deduce, the objective function is always non-negative and reaches its minimum at 0. Furthermore, from now on, we will refer to an *assignment* as a family of 0-1 parameters $\{a_{uv}^k\}_{u,v,k}$.

Hence, the objective function (4.3) is the sum of errors associated with a specific assignment. This choice of the target enables the solver to search for the assignment that achieves a faithful embedding of the available weights, i.e., the d_k's.

We introduce the following variables in order to linearize the objects involved. First, we define the Manhattan distance between the embedded nodes x_u and x_v as $m_{uv} := ||x_u - x_v||_1$, by obtaining

$$\min \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{k=1}^{M} a_{uv}^k \, | \, \mathbf{m_{uv}} - d_k \, |.$$

Second, we define $e_{uv}^k := | \, m_{uv} - d_k \, |$, that is the discrepancy bewteen such distance and a weight d_k:

$$\min \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{k=1}^{M} a_{uv}^k \, \mathbf{e_{uv}^k}.$$

Finally, $c_{uv}^k := a_{uv}^k \, e_{uv}^k$ filters out only the errors of interest, $\forall \, 1 \leq u < v \leq N$, $\forall k = 1, \ldots, M$. One hence obtains

$$\min \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{k=1}^{M} \mathbf{c_{uv}^k}. \tag{4.4}$$

The constraints of the MILP formulation with the target (4.4) are listed below:

$$\sum_{u=1}^{N-1} \sum_{v=u+1}^{N} a_{uv}^k \leq 1 \qquad \forall k = 1, \ldots, M \tag{4.5a}$$

$$\sum_{k=1}^{M} a_{uv}^k \leq 1 \qquad \forall \, 1 \leq u < v \leq N \tag{4.5b}$$

$$m_{uv}^i \geq x_u^i - x_v^i \qquad \forall i = 1, \ldots, K \;\; \forall \, 1 \leq u < v \leq N \tag{4.5c}$$

$$m_{uv}^i \geq -x_u^i + x_v^i \qquad \forall i = 1, \ldots, K \;\; \forall \, 1 \leq u < v \leq N \tag{4.5d}$$

$$m_{uv}^i \leq \tilde{M} \theta_{uv}^i + x_u^i - x_v^i \qquad \forall i = 1, \ldots, K \;\; \forall \, 1 \leq u < v \leq N \tag{4.5e}$$

$$m_{uv}^i \leq \tilde{M}(1 - \theta_{uv}^i) - x_u^i + x_v^i \qquad \forall i = 1, \ldots, K \;\; \forall \, 1 \leq u < v \leq N \tag{4.5f}$$

$$m_{uv} := \sum_{i=1}^{d} m_{uv}^i \geq \epsilon \qquad \forall \, 1 \leq u < v \leq N \tag{4.5g}$$

$$e_{uv}^k \geq \sum_{i=1}^{d} m_{uv}^i - d_k \qquad \forall \, 1 \leq u < v \leq N \;\; \forall k = 1, \ldots, M \tag{4.5h}$$

$$e_{uv}^k \geq -\sum_{i=1}^{d} m_{uv}^i + d_k \qquad \forall \, 1 \leq u < v \leq N \;\; \forall k = 1, \ldots, M \tag{4.5i}$$

$$c^k_{uv} \geq 0 \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5j)$$

$$c^k_{uv} - E a^k_{uv} \leq 0 \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5k)$$

$$c^k_{uv} - e^k_{uv} \leq 0 \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5l)$$

$$c^k_{uv} - E a^k_{uv} \geq 0 \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5m)$$

$$x_u \in \mathbb{R}^K \qquad \forall u = 1, \dots, N \qquad (4.5n)$$

$$a^k_{uv} \in \{0, 1\} \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5o)$$

$$m^i_{uv} \in \mathbb{R} \qquad \forall\, 1 \leq u < v \leq N \;\; \forall i = 1, \dots, K \qquad (4.5p)$$

$$\theta^i_{uv} \in \{0, 1\} \qquad \forall\, 1 \leq u < v \leq N \;\; \forall i = 1, \dots, K \qquad (4.5q)$$

$$e^k_{uv} \in \mathbb{R} \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5r)$$

$$c^k_{uv} \in \mathbb{R} \qquad \forall\, 1 \leq u < v \leq N \;\; \forall k = 1, \dots, M \qquad (4.5s)$$

In the following, each constraint is described:

- The constraints (4.5a) and (4.5b) model the injective assignment α in (4.2).
- The contraints (4.5c)-(4.5g) define the modules m_{uv}. Note that ϵ is a small quantity standardly used to model strict inequalities.
- The constraints (4.5h) and (4.5i) compute the errors of a feasible solution. The two inequalities are sufficient because of the optimization structure: if we want to minimize $\sum_{u,v,k} a^k_{uv} e^k_{uv}$ and the a^k_{uv}'s are either 0 or 1, then $e^k_{uv} = |m_{uv} - d_k|$ each time it appears.
- The constraints (4.5j)-(4.5m) linearize the bilinear term $a^k_{uv} e^k_{uv}$. They are known as McCormick cuts [94].
- Finally, the constraints (4.5n)-(4.5s) define the domain of the decision variables.

$\tilde{M}$ and E are big-M constraints and depend on the analyzed data. If one can estimate a parameter S such that $x_u \in [-S, S]^K \;\; \forall u = 1, \dots, N$, geometric considerations lead to set $M := \tilde{2}S + \epsilon$ and $E := \max_k\{\max\{K\tilde{M} + d_k, d_k\}\}$. We notice that the problem has $O(N^4)$ variables and $O(N^4(3+d))$ constraints.

A toy problem. In order to better explain the decision variables we introduced, we refer to the toy problem (4.1). In this case, the four devices are the vertices of the graph $V = \{1, 2, 3, 4\}$, and the reciprocal distances are $d_1 = d_2 = d_3 = d_4 = 1$, and $d_5 = d_6 = 2$. In Fig 4.2, two different assignments have been made. By using the notation of the figure, we highlight that

- in the first case, $x_A = (2, 2), x_B = (2, 4), x_C = (4, 4), x_D = (4, 2)$. Also, $a^1_{AB}, a^2_{BC}, a^3_{CD}, a^4_{DA}, a^5_{AC}, a^6_{BD} = 1$;

- in the second case, $x_{A'} = (0,2), x_{B'} = (2,2), x_{C'} = (4,2), x_{D'} = (3,3)$. Furthermore, $a^1_{A'B'}, a^2_{B'C'}, a^3_{C'D'}, a^4_{B'D'}, a^5_{A'C'}, a^6_{A'D'} = 1$.

In these examples, since the number of distances is equal to the number of possible edges, none of the a^k_{uv}'s is equal to 0. The two assignments create two different weighted graph structures. For example, the first one is such that the edge (A, D) is associated to $d_4 = 1$, while in the second one the edge (A', D') is associated to $d_6 = 2$. As one can easily check, these decision variables values satisfy the constraints of the formulation and realize the minimum for the target function (4.3), i.e. $\sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \sum_{k=1}^{M} a^k_{uv} \, | \, ||x_u - x_v||_1 - d_k \, | = 0$.

4.4 Computational results

The formulation (4.4) belong to the MILP class, and can be solved by several existing MILP solvers. We employ Gurobi Optimizer version 9.1.0 on a 11th Gen Intel Core i7-1185G7 processor running at 3.00 GHz using 16,0 GB of RAM.

The exploited dataset has been randomly generated within an integer grid $[0, 30] \times [0, 30]$. It consists of a set of (feasible) random uDGP instances $\{(N, D)\}$ where $N \in \mathbb{N}$ and $D = \{d_1, \ldots, d_M\}$ is a multiset of real numbers, with $M = \frac{N(N-1)}{2}$. For each $N \in \{5, 10, 15, 20\}$, five different instances have been tested with different time-limits. The errors are evaluated by employing two well known measures: the scaled Mean Distance Error (MDE) and the scaled Largest Distance Error (LDE), i.e.,

$$MDE(x) = \frac{1}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \frac{| \, ||x_u - x_v||_1 - d_{uv}|}{d_{uv}},$$

$$LDE(x) = \max_{(u,v)\in\mathcal{E}} \frac{| \, ||x_u - x_v||_1 - d_{uv}|}{d_{uv}},$$

where $\mathcal{E}$ is given by the assignment function α, or, in other words, by the a^k_{uv}'s, and $|\mathcal{E}| = M$. Hence, we can re-formulate in the following way by using the notation introduced in (4.4):

$$MDE(x) = \frac{1}{M} \sum_{1\le u<v\le N} \sum_{k=1,\ldots,M} \frac{c^k_{uv}}{d_k},$$

$$LDE(x) = \max_{1\le u<v\le N} \sum_{k=1,\ldots,M} \frac{c^k_{uv}}{d_k}.$$

In a intuitive manner, the scaled Mean Discrepancy Estimation (MDE) provides insight into the average percentage discrepancy between the observed realization and the provided data. On the other hand, the scaled Largest Discrepancy Estimation (LDE) offers an indication of the average worst error across all edges. By construction, the LDE is bigger than the MDE.

The computational results are shown in Tables 4.1 and 4.2. Each row of the table reports the instance statistics, followed by the time taken by Gurobi to run, the target value reached within the time-limit, and the scaled MDE and LDE scores.

N	**Time (s)**	**Target**	**MDE**	**LDE**
5	0.72	0	0	0
5	0.22	0	0	0
5	0.59	0	0	0
5	0.22	0	0	0
5	0.24	0	0	0
10	time-limit	54	0.28	4.00
10	time-limit	61	0.06	0.53
10	time-limit	39	0.15	1.50
10	time-limit	45	0.04	0.37
10	time-limit	58	0.10	1.00
15	time-limit	58	0.06	2.00
15	time-limit	84	0.08	1.40
15	time-limit	56	0.03	1.00
15	time-limit	88	0.11	2.00
15	time-limit	98	0.04	0.40
20	time-limit	328	0.14	1.33
20	time-limit	276	0.14	1.70
20	time-limit	201	0.07	0.91
20	time-limit	149	0.04	0.56
20	time-limit	289	0.13	2.11

Table 4.1: **Time limit of 10 minutes.** For each N, each row represents a different instance. In the case of 5 vertices, the problem is solved at the optimum and within the time-limit. In the cases of 10, 15 and 20 vertices, the solver reaches the time-limit of 600s before getting the optimum. MDE and LDE values are reported.

N	Time (s)	Target	MDE	LDE
5	0.72	0	0	0
5	0.22	0	0	0
5	0.59	0	0	0
5	0.22	0	0	0
5	0.24	0	0	0
10	time-limit	53	0.30	4.00
10	time-limit	61	0.06	0.54
10	time-limit	39	0.15	1.50
10	time-limit	45	0.04	0.40
10	time-limit	58	0.10	1.00
15	time-limit	58	0.06	2.00
15	time-limit	84	0.06	0.80
15	time-limit	56	0.03	1.00
15	time-limit	88	0.11	2.00
15	time-limit	94	0.03	0.40
20	time-limit	197	0.08	1.00
20	time-limit	270	0.14	1.60
20	time-limit	139	0.05	0.80
20	time-limit	135	0.04	1.20
20	time-limit	289	0.13	2.10

Table 4.2: **Time limit of 20 minutes.** For each N, each row represents a different instance. In the case of 5 vertices, the problem is solved at the optimum and within the time-limit. In the cases of 10, 15 and 20 vertices, the solver reaches the time-limit of 1200s before getting the optimum. MDE and LDE values are reported.

Moreover, the solver's time requirement for finding a solution exceeds 4 hours, even for the case where $N = 10$. For the GDP version of the problem, where the assignment is known a priori, a similar experiment shows that, for N=20 and within a time frame of 300 seconds, the order of magnitude of the MDE is 10^{-6} (e.g., in [42], Appendix B, Table 1). This suggests that the utilization of heuristics that employ the GDP version could be mandatory, mostly to address larger instance sizes.

4.5 Conclusions

In this work, we have introduced a MILP formulation for the unassigned Distance Geometry Problem that exploits the Manhattan norm. This kind of norm is more suitable for problems whose domains present a gridded-like geometry, e.g., the Mobile Positioning Problem of mobile devices in a Manhattan-like area, in the hypothesis in which it is not known which distance is associated with which pair of devices.

While the literature extensively covers MILP formulations for the assigned DGP, there is a gap in terms of the unassigned version. The purpose of this chapter is to address this gap. A set of computational results is presented and it is emphasized that the model size is a barrier to solving the problem. This fact paves the way for potential research directions, with one of the primary options being the utilization of heuristics that leverage the assigned version of the problem.

Conclusions

This thesis is divided into two main parts, each focusing on a distinct area of applied mathematics: Mathematical Modelling in Biosciences and Discrete Optimization. Each part contains two chapters, each contributing with novel methodologies or already existing methodologies applied to novel contexts, and solutions to real-world problems. While the main conclusions are presented at the end of each chapter, this section provides a summarized overview to highlight the key ideas and results discussed throughout the thesis.

The first part of this thesis focused on the application of mathematical modelling to address challenges in the biosciences, particularly in the context of the COVID-19 pandemic.

The first chapter introduced an innovative approach for early detection of emerging Variants of Concern (VoCs) using funnel plots and Bonferroni control charts. This methodology provided a statistical framework for monitoring regional reproduction numbers, allowing for the identification of anomalous patterns that can signal the emergence of new variants. Through case studies in Italy, England, India, and South Africa, it was demonstrated that the approach is simple, cost-effective, and can complement genomic sequencing efforts, thus facilitating more targeted surveillance and interventions. The ability to detect anomalies in real-time significantly enhances public health responses without the need for resource-intensive genomic sequencing.

The second chapter extended the scope of mathematical modelling by addressing the critical issue of underreporting during the first wave of the COVID-19 pandemic in Italy. Using dynamic system identification and inverse problem-solving techniques, this chapter proposed a method to reconstruct the true temporal profile of new COVID-19 cases, accounting for significant underreporting. This correction provided a more accurate representation of the pandemic's progression, which is crucial for evaluating the effectiveness of non-pharmaceutical interventions and planning future public health responses. The methodology not only aids in understanding the historical impact of the pandemic but also offers a framework that can be adapted to future epidemic outbreaks, enhancing decision-making in terms of health and economic policy.

The second part of the thesis transitioned into the domain of Discrete Optimization, with an emphasis on optimization problems arising in machine learning and mobile positioning applications. The first chapter introduced the BeMi ensemble, an architecture of Binarized Neural Networks (BNNs) designed to solve classification problems under few-shot learning scenarios. The chapter explored how the ensemble method leverages a majority voting system to combine specialized BNNs, each trained to distinguish between pairs of classes. This approach allowed the model to operate efficiently on low-power devices, such as mobile phones, which is crucial in resource-constrained

environments. The computational experiments revealed that simpler BNNs can outperform more flexible and complex networks, offering valuable insights into the trade-offs between model flexibility and computational efficiency. The chapter concluded with a discussion on future directions, including improvements to training data selection and scalability to multi-class classification problems.

The second chapter focused on the unassigned Distance Geometry Problem (DGP) in the context of the Manhattan norm, a problem with applications in mobile device positioning within grid-like environments. Unlike traditional approaches that deal with assigned DGPs, this chapter presented a MILP formulation for the unassigned version of the problem, addressing the challenge of positioning mobile devices when the distances between them are unknown. This formulation filled a significant gap in the existing literature on DGPs, and the computational results highlighted the challenges posed by the problem's size, which limits the applicability of exact solutions. The chapter concluded by proposing potential future research directions, such as the use of heuristics to improve the scalability of the model and adapt it to larger, more complex scenarios.

Bibliography

[1] R. Anderson, C. Donnelly, D. Hollingsworth, M. Keeling, C. Vegvari, R. Baggaley, and R. Maddren. Reproduction number (r) and growth rate (r) of the covid-19 epidemic in the uk: methods of estimation, data sources, causes of heterogeneity, and use as a guide in policy formulation. *The Royal Society*, 2020, 2020.

[2] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183(1):3–39, 2020.

[3] R. Andrew. Preliminary genomic characterisation of an emergent sars-cov-2 lineage in the uk defined by a novel set of spike mutations. *Virological*, 2020.

[4] P. Arora, A. Kempf, I. Nehlmeier, S. R. Schulz, H.-M. Jäck, S. Pöhlmann, and M. Hoffmann. Omicron sublineage bq. 1.1 resistance to monoclonal antibodies. *The Lancet Infectious Diseases*, 23(1):22–23, 2023.

[5] P. Aylin, N. Best, A. Bottle, and C. Marshall. Following shipman: a pilot system for monitoring mortality rates in primary care. *The Lancet*, 362(9382):485–491, 2003.

[6] D. Balcan, B. Gonçalves, H. Hu, J. J. Ramasco, V. Colizza, and A. Vespignani. Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model. *Journal of computational science*, 1(3):132–145, 2010.

[7] R. Banner, I. Hubara, E. Hoffer, and D. Soudry. Scalable methods for 8-bit training of neural networks. *Advances in neural information processing systems*, 31, 2018.

[8] Y. M. Bar-On, R. Sender, A. I. Flamholz, R. Phillips, and R. Milo. A quantitative compendium of covid-19 epidemiology. *arXiv preprint arXiv:2006.01283*, 2020.

[9] C. Barril, À. Calsina, S. Cuadrado, and J. Ripoll. On the basic reproduction number in continuously structured populations. *Mathematical methods in the applied sciences*, 44(1):799–812, 2021.

[10] C. Barril, A. Calsina, and J. Ripoll. A practical approach to r 0 in continuous-time ecological models. *Mathematical Methods in the Applied Sciences*, 41(18):8432–8445, 2018.

[11] BBC. https://www.bbc.com/news/health-56449409.

[12] BBC. https://www.bbc.com/news/uk-england-south-yorkshire-56322001.

[13] BBC. https://www.bbc.com/news/uk-england-york-north-yorkshire-56899444.

[14] J. V. Beck and K. J. Arnold. *Parameter estimation in engineering and science*. James Beck, 1977.

[15] Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[16] D. Bergman, T. Huang, P. Brooks, A. Lodi, and A. U. Raghunathan. Janos: an integrated predictive and prescriptive modeling framework. *INFORMS Journal on Computing*, 34(2):807–816, 2022.

[17] A. M. Bernardelli, S. Gualandi, H. C. Lau, and S. Milanesi. The BeMi stardust: a structured ensemble of binarized neural networks. In *International Conference on Learning and Intelligent Optimization (LION)*, pages 443–458. Springer, 2023.

[18] A. M. Bernardelli, S. Gualandi, S. Milanesi, H. C. Lau, and N. Yorke-Smith. Multi-Objective Linear Ensembles for Robust and Sparse Training of Few-Bit Neural Networks, 2024. Available for download at https://github.com/INFORMSJoC/2023.0281.

[19] A. M. Bernardelli, S. Gualandi, S. Milanesi, H. C. Lau, and N. Yorke-Smith. Multiobjective linear ensembles for robust and sparse training of few-bit neural networks. *INFORMS Journal on Computing*, 2024.

[20] S. J. Billinge, P. M. Duxbury, D. S. Gonçalves, C. Lavor, and A. Mucherino. Recent results on assigned and unassigned distance geometry with applications to protein molecules and nanostructures. *Annals of Operations Research*, 271:161–203, 2018.

[21] C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006.

[22] D. W. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. V. Guttag. What is the state of neural network pruning? In *Machine Learning and Systems (MLSYS)*, volume 2, pages 129–146, 2020.

[23] M. Blott, L. Halder, M. Leeser, and L. Doyle. Qutibench: Benchmarking neural networks on heterogeneous hardware. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(4):1–38, 2019.

[24] E. Botoeva, P. Kouvaros, J. Kronqvist, A. Lomuscio, and R. Misener. Efficient verification of relu-based neural networks via dependency analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3291–3299, 2020.

[25] A. Boudet, R. Stephan, S. Bravo, M. Sasso, and J.-P. Lavigne. Limitation of screening of different variants of sars-cov-2 by rt-pcr. *Diagnostics*, 11(7):1241, 2021.

[26] D. Breda, F. Florian, J. Ripoll, and R. Vermiglio. Efficient numerical computation of the basic reproduction number for structured populations. *Journal of computational and applied mathematics*, 384:113165, 2021.

[27] L. Brigato, B. Barz, L. Iocchi, and J. Denzler. Image classification with small datasets: overview and benchmark. *IEEE Access*, 10:49233–49250, 2022.

[28] J. Cai, K.-N. Nguyen, N. Shrestha, A. Good, R. Tu, X. Yu, S. Zhe, and T. Serra. Getting away with more network pruning: From sparsity to geometry and linear regions. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 200–218. Springer, 2023.

[29] E. Callaway. Beyond omicron: what's next for covid's viral evolution. *Nature*, 600:204–207, 12 2021.

[30] E. Callaway et al. Fast-spreading covid variant can elude immune responses. *Nature*, 589(7843):500–501, 2021.

[31] Q. Cappart, D. Chételat, E. B. Khalil, A. Lodi, C. Morris, and P. Veličković. Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24(130):1–61, 2023.

[32] S. Cauchemez and P. Bosetti. A reconstruction of early cryptic covid spread, 2021.

[33] CDC. https://www.cdc.gov/coronavirus/2019-ncov/hcp/planning-scenarios.html.

[34] CDC. https://www.cdc.gov/coronavirus/2019-ncov/variants/variant-classifications.html.

[35] M. Cesari and M. Proietti. Covid-19 in italy: ageism and decision making in a pandemic. *Journal of the American Medical Directors Association*, 21(5):576, 2020.

[36] C. Charre, C. Ginevra, M. Sabatier, H. Regue, G. Destras, S. Brun, G. Burfin, C. Scholtes, F. Morfin, M. Valette, et al. Evaluation of ngs-based approaches for sars-cov-2 whole genome characterisation. *Virus evolution*, 6(2):veaa075, 2020.

[37] W.-Y. Chiu and B.-S. Chen. Mobile positioning problem in manhattan-like urban areas: Uniqueness of solution, optimal deployment of bss, and fuzzy implementation. *IEEE transactions on signal processing*, 57(12):4918–4929, 2009.

[38] A. Cori, N. M. Ferguson, C. Fraser, and S. Cauchemez. A new framework and software to estimate time-varying reproduction numbers during epidemics. *American journal of epidemiology*, 178(9):1505–1512, 2013.

[39] G. M. Crippen, T. F. Havel, et al. *Distance geometry and molecular conformation*, volume 74. Research Studies Press Taunton, 1988.

[40] A. de Hoffer, S. Vatani, C. Cot, G. Cacciapaglia, M. L. Chiusano, A. Cimarelli, F. Conventi, A. Giannini, S. Hohenegger, and F. Sannino. Variant-driven early warning via unsupervised machine learning analysis of spike protein mutations for covid-19. *Scientific Reports*, 12(1):9275, 2022.

[41] P. M. Duxbury, L. Granlund, S. Gujarathi, P. Juhas, and S. J. Billinge. The unassigned distance geometry problem. *Discrete Applied Mathematics*, 204:117–132, 2016.

[42] C. D'ambrosio and L. Liberti. Distance geometry in linearizable norms. In *International Conference on Geometric Science of Information*, pages 830–837. Springer, 2017.

[43] ECDC. https://www.ecdc.europa.eu/en/news-events/epidemiological-update-sars-cov-2-omicron-sub-lineages-ba4-and-ba5.

[44] R. T. Eguia, K. H. Crawford, T. Stevens-Ayers, L. Kelnhofer-Millevolte, A. L. Greninger, J. A. Englund, M. J. Boeckh, and J. D. Bloom. A human coronavirus evolves antigenically to escape antibody immunity. *PLoS pathogens*, 17(4):e1009453, 2021.

[45] M. Ekstrom. A spectral characterization of the ill-conditioning in numerical deconvolution. *IEEE Transactions on Audio and Electroacoustics*, 21(4):344–348, 1973.

[46] M. ElAraby, G. Wolf, and M. Carvalho. OAMIP: Optimizing ANN Architectures Using Mixed-Integer Programming. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 219–237. Springer, 2023.

[47] P. Elliott, B. Bodinier, O. Eales, H. Wang, D. Haw, J. Elliott, M. Whitaker, J. Jonnerby, D. Tang, C. E. Walters, et al. Rapid increase in omicron infections in england during december 2021: React-1 study. *Science*, 375(6587):1406–1411, 2022.

[48] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization in network localization. In *IEEE INFOCOM 2004*, volume 4, pages 2673–2684. IEEE, 2004.

[49] N. M. Ferguson, D. Laydon, G. Nedjati-Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. Cucunubá, G. Cuomo-Dannenburg, et al. *Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand*, volume 16. Imperial College London London, 2020.

[50] T. Fetzer. Measuring the epidemiological impact of a false negative: evidence from a natural experiment. 2021.

[51] M. Fischetti and J. Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, 2018.

[52] B. K. Foundation. Valutazione di politiche di riapertura utilizzando contatti sociali e rischio di esposizione professionale, https://www.quotidianosanita.it/allegati/allegato1389403.pdf.

[53] C. Geismar, E. Fragaszy, V. Nguyen, W. L. E. Fong, M. Shrotri, S. Beale, A. Rodger, V. Lampos, T. Byrne, J. Kovar, et al. Household serial interval of covid-19 and the effect of variant b. 1.1. 7: analyses from prospective community cohort study (virus watch). *Wellcome open research*, 6, 2021.

[54] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[55] G. Giordano, F. Blanchini, R. Bruno, P. Colaneri, A. Di Filippo, A. Di Matteo, and M. Colaneri. Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. *Nature medicine*, 26(6):855–860, 2020.

[56] G. Giordano, M. Colaneri, A. Di Filippo, F. Blanchini, P. Bolzern, G. De Nicolao, P. Sacchi, P. Colaneri, and R. Bruno. Modeling vaccination rollouts, sars-cov-2 variants and the requirement for non-pharmaceutical interventions in italy. *Nature medicine*, 27(6):993–998, 2021.

[57] D. M. Gohl, J. Garbe, P. Grady, J. Daniel, R. H. Watson, B. Auch, A. Nelson, S. Yohe, and K. B. Beckman. A rapid, cost-effective tailed amplicon method for sequencing sars-cov-2. *BMC genomics*, 21:1–10, 2020.

[58] H. Goldstein and D. J. Spiegelhalter. League tables and their limitations: statistical issues in comparisons of institutional performance. *Journal of the royal statistical society series a: statistics in society*, 159(3):385–409, 1996.

[59] A. Good, J. Lin, X. Yu, H. Sieg, M. Fergurson, S. Zhe, J. Wieczorek, and T. Serra. Recall distortion in neural network pruning and the undecayed pruning algorithm. *Advances in Neural Information Processing Systems*, 35:32762–32776, 2022.

[60] G. Grasselli, A. Pesenti, and M. Cecconi. Critical care utilization for the covid-19 outbreak in lombardy, italy: early experience and forecast during an emergency response. *Jama*, 323(16):1545–1546, 2020.

[61] J. Griffin, M. Casey, Á. Collins, K. Hunt, D. McEvoy, A. Byrne, C. McAloon, A. Barber, E. A. Lane, and S. More. Rapid review of available evidence on the serial interval and generation time of covid-19. *BMJ open*, 10(11):e040263, 2020.

[62] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[63] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[64] W. T. Harvey, A. M. Carabelli, B. Jackson, R. K. Gupta, E. C. Thomson, E. M. Harrison, C. Ludden, R. Reeve, A. Rambaut, S. J. Peacock, et al. Sars-cov-2 variants, spike mutations and immune escape. *Nature Reviews Microbiology*, 19(7):409–424, 2021.

[65] T. Huang, A. M. Ferber, Y. Tian, B. Dilkina, and B. Steiner. Searching large neighborhoods for integer linear programs with contrastive learning. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 13869–13890. PMLR, 2023.

[66] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 29:4107–4115, 2016.

[67] J. Huchette, G. Muñoz, T. Serra, and C. Tsay. When deep learning meets polyhedral theory: A survey. *arXiv preprint arXiv:2305.00241*, 2023.

[68] G. Humphreys. Public health round-up. *Bulletin of the World Health Organization*, 100(1):4–5, Jan. 2022.

[69] INDGov. Genome sequencing by insacog shows variants of concern and a novel variant in india (https://pib.gov.in/pressreleasepage.aspx?prid=1707177).

[70] M. Inkelas, C. Blair, D. Furukawa, V. G. Manuel, J. H. Malenfant, E. Martin, I. Emeruwa, T. Kuo, L. Arangua, B. Robles, et al. Using control charts to understand community variation in covid-19. *PLoS One*, 16(4):e0248500, 2021.

[71] ISS. https://www.iss.it/en/cov19-cosa-fa-iss-varianti.

[72] ISTAT. https://covid19.infn.it.

[73] ISTAT. Tavole di dati, id: 25653, tavola 1 (12 aprile 2021), https://www.istat.it/it/archivio/256536.

[74] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. U. Detrano. Heart disesase dataset, 1988.

[75] Y. Jiang, D. Krishnan, H. Mobahi, and S. Bengio. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations (ICLR)*, 2019.

[76] P. Juhás, D. Cherba, P. Duxbury, W. Punch, and S. Billinge. Ab initio determination of solid-state nanostructure. *Nature*, 440(7084):655–658, 2006.

[77] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 1(8), 2017.

[78] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, volume 5, 2017.

[79] E. B. Khalil, A. Gupta, and B. Dilkina. Combinatorial attacks on binarized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[80] J. Kurtz and B. Bah. Efficient and robust mixed-integer optimization methods for training binarized deep neural networks. *arXiv preprint arXiv:2110.11382*, 2021.

[81] H. Lau, T. Khosrawipour, P. Kocbach, H. Ichii, J. Bania, and V. Khosrawipour. Evaluating the massive underreporting and undertesting of covid-19 cases in multiple global epicenters. *Pulmonology*, 27(2):110–115, 2021.

[82] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553), 2015.

[83] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998.

[84] X. Li, C. Sun, and Y. Ye. Simple and fast algorithm for binary integer and online linear programming. In *Advances in Neural Information Processing Systems*, 2020.

[85] L. Liberti. Distance geometry and data science. *Top*, 28(2):271–339, 2020.

[86] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM review*, 56(1):3–69, 2014.

[87] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[88] L. Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

[89] M. Lombardi and M. Milano. Boosting combinatorial problem modeling with machine learning. *arXiv preprint arXiv:1807.05517*, 2018.

[90] C. L. Mallows. Some comments on cp. *Technometrics*, 42(1):87–94, 2000.

[91] N. Matic, C. F. Lowe, G. Ritchie, A. Stefanovic, T. Lawson, W. Jang, M. Young, W. Dong, Z. L. Brumme, C. J. Brumme, et al. Rapid detection of sars-cov-2 variants of concern, including b. 1.1. 28/p. 1, british columbia, canada. *Emerging Infectious Diseases*, 27(6):1673, 2021.

[92] MATLAB. https://it.mathworks.com/help/optim.

[93] MATLAB. https://it.mathworks.com/products/sysid.html.

[94] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[95] S. Milanesi and G. De Nicolao. Correction of italian under-reporting in the first covid-19 wave via age-specific deconvolution of hospital admissions. *Plos one*, 18(12):e0295079, 2023.

[96] S. Milanesi, F. Rosset, M. Colaneri, G. Giordano, K. Pesenti, F. Blanchini, P. Bolzern, P. Colaneri, P. Sacchi, G. De Nicolao, et al. Early detection of variants of concern via funnel plots of regional reproduction numbers. *Scientific reports*, 13(1):1052, 2023.

[97] M. Mistry, D. Letsios, G. Krennrich, R. M. Lee, and R. Misener. Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. *INFORMS Journal on Computing*, 33(3):1103–1119, 2021.

[98] C. Modi, V. Böhm, S. Ferraro, G. Stein, and U. Seljak. Estimating covid-19 mortality in italy early in the covid-19 pandemic. *Nature communications*, 12(1):2729, 2021.

[99] M. A. Mohammed, K. Cheng, A. Rouse, and T. Marshall. Bristol, shipman, and clinical governance: Shewhart's forgotten lessons. *The Lancet*, 357(9254):463–467, 2001.

[100] D. C. Montgomery. *Introduction to statistical quality control*. John wiley & sons, 2019.

[101] J. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. *Advances in Neural Information Processing Systems (NeurIPS)*, 4:847–854, 1991.

[102] A. Morcos, H. Yu, M. Paganini, and Y. Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[103] P. Neopane, J. Nypaver, R. Shrestha, and S. S. Beqaj. Sars-cov-2 variants detection using taqman sars-cov-2 mutation panel molecular genotyping assays. *Infection and Drug Resistance*, pages 4471–4479, 2021.

[104] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5947–5956, 2017.

[105] J. Noh and G. Danuser. Estimation of the fraction of covid-19 infected people in us states and countries worldwide. *PloS one*, 16(2):e0246772, 2021.

[106] D. S. Ong, J. G. Koeleman, N. Vaessen, S. Breijer, S. Paltansing, and P. de Man. Rapid screening method for the detection of sars-cov-2 variants of concern. *Journal of Clinical Virology*, 141:104903, 2021.

[107] W. H. Organization et al. Genomic sequencing of sars-cov-2: a guide to implementation for maximum impact on public health, 8 january 2021. 2021.

[108] W. H. Organization et al. Guidance for surveillance of sars-cov-2 variants: interim guidance, 9 august 2021. Technical report, World Health Organization, 2021.

[109] W. H. Organization et al. Weekly bulletin on outbreak and other emergencies: Week 48: 22-28 november 2021. 2021.

[110] V. Patil and Y. Mintz. A mixed-integer programming approach to training dense neural networks. *arXiv preprint arXiv:2201.00723*, 2022.

[111] R. J. Perla, S. M. Provost, G. J. Parry, K. Little, and L. P. Provost. Understanding variation in reported covid-19 deaths with a novel shewhart chart application. *International Journal for Quality in Health Care*, 33(1):mzaa069, 2021.

[112] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung. *Regularized system identification: Learning dynamic models from data*. Springer Nature, 2022.

[113] G. Pullano, L. Di Domenico, C. E. Sabbatini, E. Valdano, C. Turbelin, M. Debin, C. Guerrisi, C. Kengne-Kuetche, C. Souty, T. Hanslik, et al. Underdetection of cases of covid-19 in france threatens epidemic control. *Nature*, 590(7844):134–139, 2021.

[114] A. Remuzzi and G. Remuzzi. Covid-19 and italy: what next? *The lancet*, 395(10231):1225–1228, 2020.

[115] C. Roger, R. Molina, C. Rey, T. Serra, E. Puertas, and O. Pujol. Training thinner and deeper neural networks: Jumpstart regularization. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, volume 13292, pages 345–357. Springer, 2022.

[116] C. Sakr, J. Choi, Z. Wang, K. Gopalakrishnan, and N. Shanbhag. True gradient-based training of deep binary activated neural networks via continuous binarization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2346–2350. IEEE, 2018.

[117] SANGER. Covid-19 genomic surveillance, https://covid-surveillance-data.cog.sanger.ac.uk/download/lineages_by_ltla_and_week.tsv.

[118] G. Sartor, M. Del Riccio, I. Dal Poz, P. Bonanni, and G. Bonaccorsi. Covid-19 in italy: Considerations on official data. *International journal of infectious diseases*, 98:188–190, 2020.

[119] A. H. Sayed, A. Tarighat, and N. Khajehnouri. Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. *IEEE signal processing magazine*, 22(4):24–40, 2005.

[120] G. Sebastiani, M. Massa, and E. Riboli. Covid-19 epidemic in italy: evolution, projections and impact of government measures. *European journal of epidemiology*, 35:341–345, 2020.

[121] T. Serra, A. Kumar, and S. Ramalingam. Lossless compression of deep neural networks. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, volume 17, pages 417–430, 2020.

[122] T. Serra, X. Yu, A. Kumar, and S. Ramalingam. Scaling up exact neural network compression by ReLU stability. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27081–27093, 2021.

[123] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011.

[124] J. Singh, S. A. Rahman, N. Z. Ehtesham, S. Hira, and S. E. Hasnain. Sars-cov-2 variants of concern are emerging in india. *Nature medicine*, 27(7):1131–1133, 2021.

[125] SKYnews. https://news.sky.com/story/covid-19-lateral-flow-kits-unavailable-for-delivery-after-pm-announcement-triggers-rush-for-free-tests-12548633.

[126] D. Spiegelhalter and A. Masters. *Covid by numbers: making sense of the pandemic with data*. Penguin UK, 2021.

[127] D. J. Spiegelhalter. Funnel plots for comparing institutional performance. *Statistics in medicine*, 24(8):1185–1202, 2005.

[128] D. J. Spiegelhalter. Handling over-dispersion of performance indicators. *BMJ Quality & Safety*, 14(5):347–351, 2005.

[129] L. Subissi, A. von Gottberg, L. Thukral, N. Worp, B. B. Oude Munnink, S. Rathore, L. J. Abu-Raddad, X. Aguilera, E. Alm, B. N. Archer, et al. An early warning system for emerging sars-cov-2 variants. *Nature medicine*, 28(6):1110–1115, 2022.

[130] E. Takashita, S. Yamayoshi, P. Halfmann, N. Wilson, H. Ries, A. Richardson, M. Bobholz, W. Vuyk, R. Maddox, D. A. Baker, et al. In vitro efficacy of antiviral agents against omicron subvariant ba. 4.6. *New England Journal of Medicine*, 387(22):2094–2097, 2022.

[131] W. Tang, G. Hua, and L. Wang. How to train a compact binary neural network with high accuracy? In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 2625–2631, 2017.

[132] B. S. Thomas and N. A. Marks. Estimating the case fatality ratio for covid-19 using a time-shifted distribution analysis. *Epidemiology & Infection*, 149:e197, 2021.

[133] T. Thorbjarnarson and N. Yorke-Smith. Optimal training of integer-valued neural networks with mixed integer programming. *PLOS ONE*, 18(2):1–17, 2023.

[134] C. Tjandraatmadja, R. Anderson, J. Huchette, W. Ma, K. K. Patel, and J. P. Vielma. The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification. *Advances in Neural Information Processing Systems*, 33:21675–21686, 2020.

[135] V. Tjeng, K. Y. Xiao, and R. Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2018.

[136] R. Toro Icarte, L. Illanes, M. P. Castro, A. A. Cire, S. A. McIlraith, and J. C. Beck. Training binarized neural networks using MIP and CP. In *International Conference on Principles and Practice of Constraint Programming*, volume 11802, pages 401–417. Springer, 2019.

[137] C. Tsay, J. Kronqvist, A. Thebelt, and R. Misener. Partition-based formulations for mixed-integer optimization of trained ReLU neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3068–3080, 2021.

[138] UKGov. Briefing 10 (https://www.gov.uk/government/publications/investigation-of-sars-cov-2-variants-technical-briefings).

[139] UKGov. https://www.gov.uk/government/news/testing-at-private-lab-suspended-following-nhs-test-and-trace-investigation.

[140] UKGov. https://www.gov.uk/government/publications/long-term-evolution-of-sars-cov-2-26-july-2021/long-term-evolution-of-sars-cov-2-26-july-2021.

[141] UKGov. https://www.gov.uk/government/speeches/health-and-social-care-secretary-oral-statement-on-covid-19.

[142] J. Unnikrishnan, S. Mangalathu, and R. V. Kutty. Estimating underreporting of covid-19 cases in indian states: an approach using a delay-adjusted case fatality ratio. *BMJ open*, 11(1):e042584, 2021.

[143] J. Vanschoren. Meta-learning. In *Automated machine learning*, pages 35–61. Springer, 2019.

[144] E. Volz, S. Mishra, M. Chand, J. C. Barrett, R. Johnson, L. Geidelberg, W. R. Hinsley, D. J. Laydon, G. Dabrera, Á. O'Toole, et al. Assessing transmissibility of sars-cov-2 lineage b. 1.1. 7 in england. *Nature*, 593(7858):266–269, 2021.

[145] K. Wang, L. Lozano, C. Cardonha, and D. Bergman. Optimizing over an ensemble of trained neural networks. *INFORMS Journal on Computing*, 35(3):652–674, 2023.

[146] WHO. https://www.who.int/initiatives/global-influenza-surveillance-and-response-system/virus-sharing/shipping-and-logistics-activities.

[147] Wikipedia. https://en.wikipedia.org/wiki/covid-19_pandemic_in_italy.

[148] H. P. Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.

[149] C.-F. J. Wu. Jackknife, bootstrap and other resampling methods in regression analysis. *the Annals of Statistics*, 14(4):1261–1295, 1986.

[150] S. L. Wu, A. N. Mertens, Y. S. Crider, A. Nguyen, N. N. Pokpongkiat, S. Djajadi, A. Seth, M. S. Hsiang, J. M. Colford Jr, A. Reingold, et al. Substantial underestimation of sars-cov-2 infection in the united states. *Nature communications*, 11(1):4507, 2020.

[151] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[152] H. Ye, H. Xu, H. Wang, C. Wang, and Y. Jiang. GNN&GBDT-guided fast optimizing framework for large-scale integer programming. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 39864–39878. PMLR, 2023.

[153] H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.

[154] X. Yu, T. Serra, S. Ramalingam, and S. Zhe. The combinatorial brain surgeon: Pruning weights that cancel one another in neural networks. In *International Conference on Machine Learning (ICML)*, pages 25668–25683, 2022.

[155] A. Zhang, Y. Leng, Y. Zhang, K. Wu, Y. Ji, S. Lei, and Z. Xia. Meta-analysis of coagulation parameters associated with disease severity and poor prognosis of covid-19. *International Journal of Infectious Diseases*, 100:441–448, 2020.

[156] Z. Zhao, H. Li, X. Wu, Y. Zhong, K. Zhang, Y.-P. Zhang, E. Boerwinkle, and Y.-X. Fu. Moderate mutation rate in the sars coronavirus genome and its implications. *BMC evolutionary biology*, 4:1–9, 2004.