TECHNISCHE
UNIVERSITÄT
DARMSTADT

Robust and Distributed Cluster Enumeration and Object Labeling

Dem Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
vorgelegte Dissertation
von

Freweyni Kidane Teklehaymanot, M.Sc.

Erstgutachter: Dr.-Ing. Michael Muma
Zweitgutachter: Prof. Dr.-Ing. Abdelhak M. Zoubir,
Distinguished Prof. Petar M. Djurić, Ph.D.

Darmstadt 2019

Teklehaymanot, Freweyni K.– Robust and Distributed Cluster Enumeration and Object Labeling
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2019
URN: urn:nbn:de:tuda-tuprints-85393
Tag der mündlichen Prüfung: 04. März 2019

To Kidane, Zigbey, Fitsum, and Samrawit.

Acknowledgments

I would like to thank everyone who contributed to the success of this work and supported me through the challenging yet rewarding time of my life.

First of all, I would like to express my sincere gratitude to my supervisor Dr.-Ing. Michael Muma for his guidance and encouragement over the years. Thank you for giving your time so generously and all the valuable suggestions. I would also like to thank my co-supervisor Prof. Dr.-Ing. Abdelhak M. Zoubir for giving me the opportunity to pursue my Ph.D. at the Signal Processing Group. Thank you for your motivation and continuous support throughout my Ph.D. study. In addition, I would like to thank my second co-supervisor Distinguished Prof. Petar M. Djurić, Ph.D. for the enriching discussions. It is your research work that sparked my interest in model selection problems. I would also like to thank my doctoral examiners Prof. Dr. rer. nat. Florian Steinke, Prof. Dr. mont. Mario Kupnik, and Prof. Dr. techn. Heinz Koeppl.

A huge thanks go to the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering (GSC CE) at Technische Universität Darmstadt for financialy supporting my Ph.D. study. I would like to thank my colleagues and all the people at GSC CE, specially Prof. Dr. rer. nat. Michael Schäfer, Dr. rer. pol. Markus Lazanowski, Carina Schuster, Steffi Vass, and Christian Schmitt. I would also like to thank Prof. Dr. Frank Aurzada for his support.

I wish to thank all the current and former members of the Signal Processing Group for the great atmosphere. In particular, thank you to my roommates Dr.-Ing. Sahar Khawatmi, Dr.-Ing Lala Khadidja Hamaidi, Ann-Khatrin Seifert, and Dr.-Ing. Tim Shäck for the interesting discussions and the laughter we shared. Thank you to Mark Ryan Leonard for your insightful suggestions and customizing this beautiful latex template. In addition, it was a great pleasure working with Dr.-Ing. Christian Debes, Dr.-Ing. Michael Fauß, Huiping Huang, Di Jin, Amare Kassaw, Toufik Mouchini, Afief Dias Pambudi, Dominik Reinhard, Sergey Sukhanov, Dr.-Ing. Mouhammad Alhumaidi, Dr.-Ing. Sara Al-Sayed, Patricia Binder, Dr.-Ing. Nevine Demitri, Dr. Stefano Fortunati, Dr.-Ing. Gökhan Gül, Dr.-Ing. Jürgen Hahn, Dr. Roy Howard, Dr.-Ing Michael Leigsnering, Dr.-Ing. Simon Rosenkranz, Dr.-Ing. Adrian Šošić, Dr.-Ing. Wassim Suleiman, Dr.-Ing. Christian Weiss, and Dr.-Ing. Wenjun Zeng.

I am grateful to Renate Koschella and Hauke Fath for the administrative and technical support and their readiness to help at every occasion.

I wish to express my deepest gratitude to my parents Kidane and Zigbey and my sister Samrawit for their unconditional love and support. Thank you for believing in me. I would also like to thank Ulrike and Wolfgang for making me feel at home. Thank you for always being there for me. In addition, thank you to my friend Mulubrhan for your enthusiasm and fun

loving character.

Finally and most importantly, I would like to thank my husband Fitsum. Without your love, patience, and encouragement this work would not have been possible. Thank you for everything!

Darmstadt, 11.03.2019

Robuste und verteilte Cluster-Enumeration und Objektkennzeichnung

Kurzfassung

Diese Dissertation leistet einen Beitrag zum Bereich der Cluster-Analyse durch die Bereitstellung grundsätzlicher Methoden zur Bestimmung der Cluster-Anzahl und -Zugehörigkeiten, die auch in Anwesenheit von Ausreißern zuverlässig funktionieren. Die wichtigsten theoretischen Beiträge sind in zwei Theoremen über die Bayes'sche Cluster-Enumeration zusammengefasst, die auf der Modellierung der Daten als Familie von Gauß- und t_ν-Verteilungen basieren. Die praktische Relevanz wird durch die Anwendung auf fortgeschrittene Probleme der Signalverarbeitung, wie beispielsweise verteilte Kameranetze und radarbasierte Personenidentifikation, demonstriert.

Insbesondere wird ein neues Kriterium zur Cluster-Enumeration, das auf eine breite Klasse von Datenverteilungen anwendbar ist, unter Verwendung des Bayes-Theorems sowie asymptotischer Approximationen hergeleitet. Dies dient als Ausgangspunkt für die Formulierung von Kriterien zur Cluster-Enumeration bei spezifischen Datenverteilungen. In diesem Zusammenhang wird ein Bayes'sches Kriterium zur Cluster-Enumeration hergeleitet, indem die Daten als eine Familie multivariater Gauß-Verteilungen modelliert werden. In der Praxis sind die beobachteten Daten oft starkem Rauschen und Ausreißern ausgesetzt, wodurch die eigentliche Struktur der Daten nur schwer erkennbar ist. Daher ist es schwierig, die Anzahl der Cluster robust zu schätzen. In dieser Arbeit wird ein robustes Kriterium zur Cluster-Enumeration entwickelt, das auf Modellierung der Daten als Familie multivariater t_ν-Verteilungen beruht. Die Familie der t_ν-Verteilungen ist, durch Variation ihres Freiheitsgrads (ν), flexibel und enthält als Sonderfälle die Cauchy-Verteilung mit schweren Rändern für $\nu = 1$ sowie die Gauß-Verteilung für $\nu \to \infty$. Unter der Annahme, dass ν hinreichend klein ist, berücksichtigt das robuste Kriterium Ausreißer, indem es ihnen weniger Gewicht in der Zielfunktion gibt. Ein weiterer Beitrag dieser Dissertation liegt in der Weiterentwicklung der Strafterme sowohl des robusten als auch des Gauß'schen Kriteriums für eine endliche Stichprobengröße. Die hergeleiteten Kriterien zur Cluster-Enumeration erfordern einen Clustering-Algorithmus, der die Daten entsprechend der Anzahl der durch jedes potentielle Modell spezifizierten Cluster aufteilt und eine Schätzung der Cluster-Parameter liefert. Hierbei wird eine modellbasierte, unüberwachte Lernmethode angewendet, um die Daten vor der Berechnung eines Enumerationskriteriums zu partitionieren, was zu einem zweistufigen Algorithmus führt. Der vorgeschlagene Algorithmus stellt ein vereinheitlichtes methodisches Rahmenwerk zur Schätzung der Cluster-Anzahl und -Zugehörigkeiten bereit.

Die entwickelten Algorithmen werden auf zwei anspruchsvolle Probleme der Signalverarbeitung angewendet. Im Speziellen werden die Kriterien zur Cluster-Enumeration für die Anwendung in einem verteilten Sensornetz um zwei verteilte und adaptive Bayes'sche Algorithmen zur Cluster-Enumeration erweitert. Die vorgestellten Algorithmen werden auf

ein Kameranetz-Szenario angewendet, bei dem die Aufgabe darin besteht, die Anzahl der Fußgänger basierend auf eingehenden Datenströmen zu schätzen. Die Datenströme werden von mehreren Kameras, die eine nicht-stationäre Szene aus verschiedenen Blickwinkeln filmen, aufgenommen. Ein weiterer Forschungsschwerpunkt dieser Dissertation ist die Zuordnung einzelner Datenpunkte zu Clustern und der zugehörigen Cluster-Bezeichnungen unter der Voraussetzung, dass die Anzahl der Cluster entweder vom Anwender vorab festgelegt oder durch eines der zuvor beschriebenen Verfahren geschätzt wird. Die Lösung dieser Aufgabe ist bei einer Vielzahl von Anwendungen, wie z.B. verteilten Sensornetzen und radarbasierter Personenidentifikation erforderlich. Zu diesem Zweck wird ein adaptiver Algorithmus zur gemeinsamen Objektkennzeichnung und -verfolgung vorgeschlagen und auf einen realen Datensatz zur Fußgängerkennzeichnung in einer unkalibierten Mehrobjekt-Mehrkamera-Anordnung mit geringer Videoauflösung und häufigen Objektverdeckungen angewendet. Der vorgeschlagene Algorithmus eignet sich gut für Ad-hoc-Netze, da er weder eine Registrierung der Kameraansichten noch ein Fusionszentrum erfordert. Schließlich wird ein Algorithmus zur gemeinsamen Cluster-Enumeration und -Bezeichnung vorgeschlagen, um das kombinierte Problem der gleichzeitigen Schätzung von Cluster-Anzahl und -Zugehörigkeiten zu lösen. In einer Echtdatenanwendung wird der vorgestellte Algorithmus auf die Personenkennzeichnung anhand von Radar-Daten angewendet, ohne vorherige Informationen über die Anzahl der Personen. Er erreicht eine vergleichbare Leistung wie ein überwachter Ansatz, der Kenntnis über die Anzahl der Personen sowie eine beträchtliche Menge an Trainingsdaten mit bekannten Cluster-Bezeichnungen erfordert. Die vorgeschlagene unüberwachte Methode ist bei der betrachteten Anwendung eines intelligenten, betreuten Wohnens von Vorteil, da sie die fehlenden Informationen aus den Daten extrahiert. Basierend auf diesen Beispielen und unter Berücksichtigung der vergleichsweise niedrigen Rechenkosten kann davon ausgegangen werden, dass die vorgeschlagenen Methoden nützliche Werkzeuge für die robuste Cluster-Analyse mit vielen potenziellen Anwendungsbereichen – auch außerhalb des Ingenieurwesens – darstellen.

Robust and Distributed Cluster Enumeration and Object Labeling

Abstract

This dissertation contributes to the area of cluster analysis by providing principled methods to determine the number of data clusters and cluster memberships, even in the presence of outliers. The main theoretical contributions are summarized in two theorems on Bayesian cluster enumeration based on modeling the data as a family of Gaussian and t_ν distributions. Real-world applicability is demonstrated by considering advanced signal processing applications, such as distributed camera networks and radar-based person identification.

In particular, a new cluster enumeration criterion, which is applicable to a broad class of data distributions, is derived by utilizing Bayes' theorem and asymptotic approximations. This serves as a starting point when deriving cluster enumeration criteria for specific data distributions. Along this line, a Bayesian cluster enumeration criterion is derived by modeling the data as a family of multivariate Gaussian distributions. In real-world applications, the observed data is often subject to heavy tailed noise and outliers which obscure the true underlying structure of the data. Consequently, estimating the number of data clusters becomes challenging. To this end, a robust cluster enumeration criterion is derived by modeling the data as a family of multivariate t_ν distributions. The family of t_ν distributions is flexible by variation of its degree of freedom parameter (ν) and it contains, as special cases, the heavy tailed Cauchy for $\nu = 1$, and the Gaussian distribution for $\nu \to \infty$. Given that ν is sufficiently small, the robust criterion accounts for outliers by giving them less weight in the objective function. A further contribution of this dissertation lies in refining the penalty terms of both the robust and Gaussian criterion for the finite sample regime. The derived cluster enumeration criteria require a clustering algorithm that partitions the data according to the number of clusters specified by each candidate model and provides an estimate of cluster parameters. Hence, a model-based unsupervised learning method is applied to partition the data prior to the calculation of an enumeration criterion, resulting in a two-step algorithm. The proposed algorithm provides a unified framework for the estimation of the number of clusters and cluster memberships.

The developed algorithms are applied to two advanced signal processing use cases. Specifically, the cluster enumeration criteria are extended to a distributed sensor network setting by proposing two distributed and adaptive Bayesian cluster enumeration algorithms. The proposed algorithms are applied to a camera network use case, where the task is to estimate the number of pedestrians based on streaming-in data collected by multiple cameras filming a non-stationary scene from different viewpoints. A further research focus of this dissertation is the cluster membership assignment of individual data points and their associated cluster labels given that the number of clusters is either prespecified by the user or estimated by one of the methods described earlier. Solving this task is required in a broad range of appli-

cations, such as distributed sensor networks and radar-based person identification. For this purpose, an adaptive joint object labeling and tracking algorithm is proposed and applied to a real data use case of pedestrian labeling in a calibration-free multi-object multi-camera setup with low video resolution and frequent object occlusions. The proposed algorithm is well suited for ad hoc networks, as it requires neither registration of camera views nor a fusion center. Finally, a joint cluster enumeration and labeling algorithm is proposed to deal with the combined problem of estimating the number of clusters and cluster memberships at the same time. The proposed algorithm is applied to person labeling in a real data application of radar-based person identification without prior information on the number of individuals. It achieves comparable performance to a supervised approach that requires knowledge of the number of persons and a considerable amount of training data with known cluster labels. The proposed unsupervised method is advantageous in the considered application of smart assisted living, as it extracts the missing information from the data. Based on these examples, and, also considering the comparably low computational cost, we conjuncture that the proposed methods provide a useful set of robust cluster analysis tools for data science with many potential application areas, not only in the area of engineering.

Publications

The following publications have been produced during the period of doctoral candidacy.

Internationally Refereed Journal Articles

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Bayesian cluster enumeration criterion for unsupervised learning". In: *IEEE Transactions on Signal Processing* 66.20 (2018), pp. 5392–5406.

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Robust Bayesian cluster enumeration". Under review in: *IEEE Transactions on Signal Processing*. 2018. online: https://arxiv.org/abs/1811.12337.

Internationally Refereed Conference Papers

- M. Faußß, M. Muma, F. K. Teklehaymanot & A. M. Zoubir. "Robust detection for cluster analysis". Accepted for publication in: *The 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, UK, 2019.

- F. K. Teklehaymanot, A.-K. Seifert, M. Muma, M. G. Amin & A. M. Zoubir. "Bayesian target enumeration and labeling using radar data of human gait". In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. Rome, Italy, 2018, pp. 1356–1360.

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Diffusion-based Bayesian cluster enumeration in distributed sensor networks". In: *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP)*. Freiburg, Germany, 2018, pp. 1–5.

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Novel Bayesian cluster enumeration criterion for cluster analysis with finite sample penalty term". In: *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, 2018, pp. 4274–4278.

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Adaptive diffusion-based track assisted multi-object labeling in distributed camera networks". In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece, 2017, pp. 2299–2303.

- F. K. Teklehaymanot, M. Muma, J. Liu & A. M. Zoubir. "In-network adaptive cluster enumeration for distributed classification/labeling". In: *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*. Budapest, Hungary, 2016, pp. 448–452.
- F. K. Teklehaymanot, M. Muma, B. Béjar, P. Binder, A. M. Zoubir & M. Vetterli. "Robust diffusion-based unsupervised object labelling in distributed camera networks". In: *Proceedings of the 12th IEEE AFRICON*. Addis Ababa, Ethiopia, 2015.

MATLAB CODE

- F. K. Teklehaymanot, M. Muma & A. M. Zoubir. *MATLAB toolbox for Bayesian cluster enumeration*. 2018.
  ONLINE: https://github.com/FreTekle/Bayesian-Cluster-Enumeration.

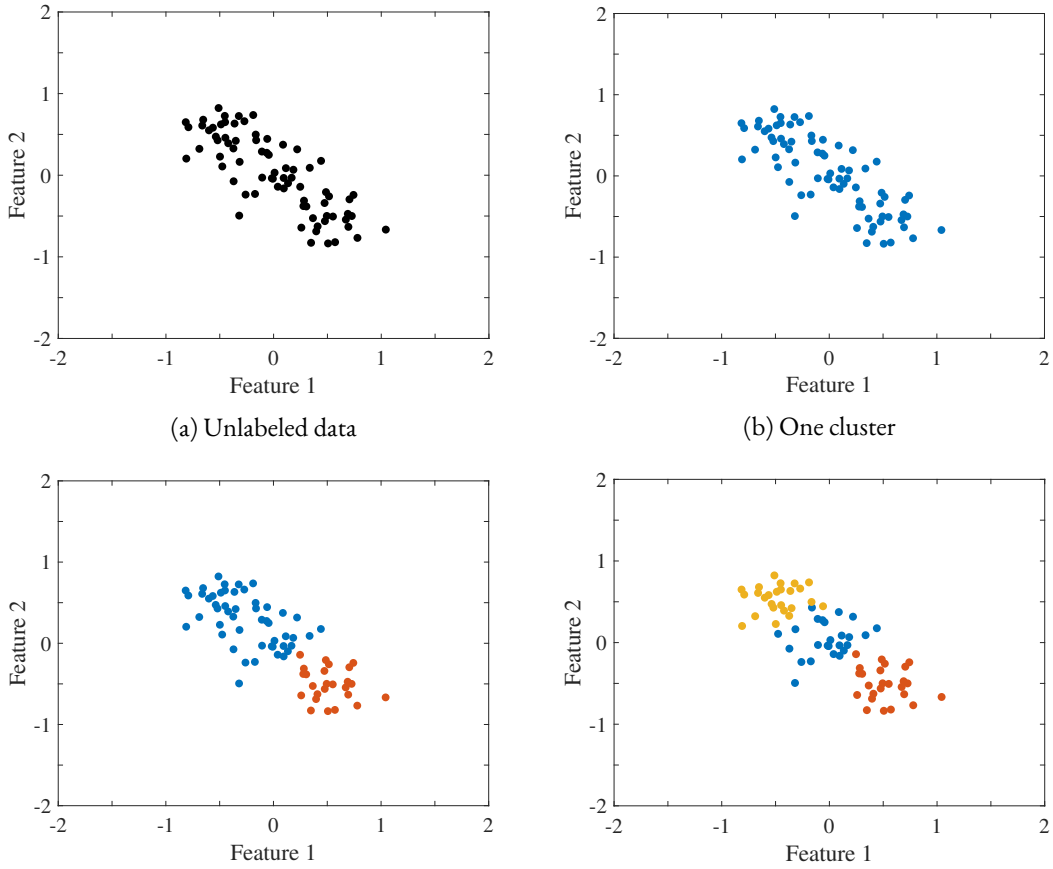
Contents

1 Introduction

1.1 Introduction and Motivation

Cluster analysis is the task of finding the underlying groupings (or clusters) in a set of unlabeled data. It is an unsupervised learning task, where the goal is to create distinct clusters such that data points that belong to the same cluster are more similar to each other compared to those belonging to a different cluster. A major challenge in cluster analysis is that the notion of a cluster is not consistently defined. On a very high level, a cluster is easily defined as a group of similar data points. However, this definition raises many questions regarding which similarity measure to use and how similar data points should be in order to belong to the same cluster. Consequently, among other reasons, the lack of unique definition for a cluster has paved the way for the development of various clustering algorithms which differ in their understanding of what a cluster is and how to find it. Nevertheless, most clustering algorithms have a common strategy, which is, to divide cluster analysis into two subtasks. The first subtask is to estimate the number of clusters (or partitions) that best describe the underlying structure of the data based on some predefined measure. However, this task is non-trivial since there might exist many possible ways of clustering the same data set as shown in the illus-

trative example in Figure 1.1. As a result, without prior knowledge of the underlying structure of the data, different methods can come up with different ways to partition the data. Once the number of clusters is estimated, the next subtask is to provide a common label to data points that are grouped together based on some similarity measure. In case there is an overlap between clusters, the labeling task becomes challenging and different methods might result in different labeling solutions for the same data set. As an example, Figure 1.2 shows the labeling results of a data set which contains two features from the Fisher's Iris data set [Fisher, 1936; Lichman, 2013] using the K-means [Lloyd, 1982; Arthur & Vassilvitskii, 2007] and the expectation maximization (EM) [Dempster et al., 1977] algorithm. Even though two out of three clusters are overlapping and difficult to partition for a human observer, the EM algorithm partitions the data almost perfectly.



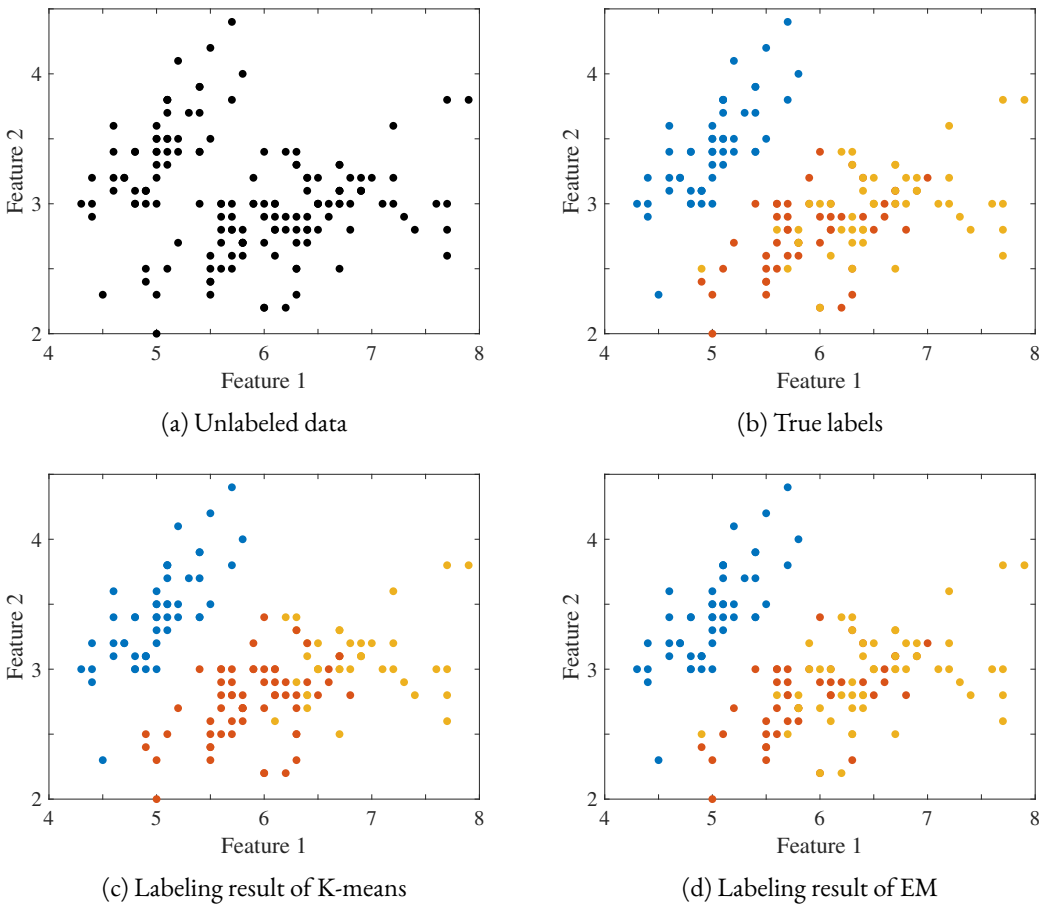
(a) Unlabeled data

(b) One cluster

(c) Two clusters

(d) Three clusters

Figure 1.1: Based on the number of clusters the same data set can be interpreted in different ways.



(a) Unlabeled data

(b) True labels

(c) Labeling result of K-means

(d) Labeling result of EM

Figure 1.2: Labeling results of different clustering algorithms on a data set comprised of two features from the Fisher's Iris data set.

Cluster analysis plays a crucial role in a wide variety of fields of study, such as social sciences, biology, medical sciences, statistics, machine learning, pattern recognition, and computer vision [Kaufman & Rousseeuw, 1990; King, 2015; Davé & Krishnapuram, 1997; Xu & Wunsch, 2005]. In this dissertation, the importance of cluster analysis techniques in real-world applications is demonstrated using two use cases. The first use case is concerned with object labeling in uncalibrated distributed camera networks, which is a common problem, among others, in video surveillance and sports analysis. In such applications, the number of objects is mostly unknown and possibly time-varying. After extracting valuable information from the video recorded by the distributed camera network, we propose to solve the object labeling problem by treating it as a data clustering task. The number of clusters is estimated, and a unique and

consistent label is provided to objects across camera views and time frames. The second use case deals with person labeling using gait measurements recorded by a radar. In applications such as smart homes and assisted living radar is preferable to cameras since it preserves privacy, can penetrate common materials, and is insensitive to lighting conditions. Similar to the first use case, clustering algorithms perform person labeling requiring neither prior information on the number of individuals nor training data with known labels.

In real-world applications, the observed data is often subject to noise and outliers [Davé & Krishnapuram, 1997; Gallegos & Ritter, 2005; Garcá-Escudero et al., 2011; Zoubir et al., 2012; Zoubir et al., 2018] which obscure the true underlying structure of the data. Consequently, cluster analysis becomes even more challenging when either the data are contaminated by a fraction of outliers or there exists deviations from the distributional assumptions. This calls for robust clustering algorithms which can withstand small deviations from distributional assumptions and are insensitive to noise and outliers. However, designing cluster analysis techniques for contaminated data sets raises even more questions, such as, *how much contamination and outliers should a clustering algorithm tolerate before it is forced to open a new cluster that explains the additional data?*, and *how should the clustering algorithm behave when outliers form their own cluster?*.

1.2 Aims of this Doctoral Project

The aim of this doctoral project is to develop, analyze, and improve cluster analysis techniques that enable us to solve advanced statistical signal processing problems. The main research questions raised and addressed in this dissertation are the following.

- Proposing principled methods to estimate the number of clusters and cluster memberships in the presence of cluster overlap and outliers. A particular emphasis is given to robust statistical methods that can deal with heavy tailed noise and outliers. Bayes' theorem and asymptotic approximations are utilized to arrive at closed-form expressions.

- Developing algorithms and demonstrating their applicability in advanced signal processing problems. The algorithms are required to have small computational cost, and, at the same time, estimate the desired parameters with small error.

1.3 Dissertation Overview

The main body of the dissertation is organized into three parts. Part I presents the derivation, numerical analysis, and practical application of novel cluster enumeration methods. It contains Chapter 2 and Chapter 3. In Chapter 2, a Bayesian cluster enumeration criterion, which is applicable to a broad class of data distributions, is derived. A criterion with a closed-form expression is obtained by modeling the data as a family of multivariate Gaussian distributions. The expectation maximization algorithm is applied to partition the data prior to the calculation of an enumeration criterion, resulting in a two-step approach. The penalty term of the new criterion is further refined for the finite sample regime. Finally, the derived criteria are extended to a distributed sensor network setup. A detailed performance evaluation of the proposed algorithms is provided using numerical and real data experiments.

In Chapter 3, the focus lies on deriving robust Bayesian cluster enumeration criteria by modeling the data as a family of multivariate t_ν distributions. Similar to Chapter 2, a two-step algorithm that provides a unified framework for the estimation of the number of clusters and cluster memberships is developed. The performance of the proposed algorithm is evaluated and compared to existing methods on challenging experimental scenarios.

Part II develops object labeling and tracking algorithms in the context of uncalibrated distributed camera networks in the absence of a fusion center that can collect and process the data in one place. Particularly, Chapter 4 presents a robust and distributed object labeling algorithm for a camera network whose nodes are interested in a static scene. Next, the algorithm is extended to the case where the nodes are interested in a time-varying scene. The performance of the proposed algorithms is evaluated using real data use cases on multi-object and multi-camera network application.

Part III fuses the ideas from Part I and Part II. In Chapter 5, the simultaneous estimation of the number of clusters and cluster memberships is discussed and, consequently, a joint cluster enumeration and labeling algorithm is proposed. The performance of the proposed algorithm is analyzed using numerical experiments. In addition, the proposed method is applied to a real data example of person labeling using radar-based human gait measurements.

Finally, the dissertation is summarized and concluding remarks are made in Chapter 6 and, future research directions are briefly discussed in Chapter 7.

PART I

Cluster Enumeration

2

Bayesian Cluster Enumeration

2.1 Introduction

Standard clustering methods, such as the K-means and the expectation maximization (EM) algorithm, can be used to partition data only when they are provided with a value for the number of clusters. In this chapter, we propose an estimator for the number of clusters using newly derived Bayesian cluster enumeration criteria, as detailed in the sequel.

Specifically, the state-of-the-art on cluster enumeration is discussed in Section 2.2 and the main contributions made in this chapter are summarized in Section 2.3. In Section 2.4, the problem of estimating the number of data clusters is formulated. The generic Bayesian cluster enumeration criterion is introduced in Section 2.5. In Section 2.6, first, a new criterion that models the data as a family of multivariate Gaussian distributions is derived. Then, a two-step approach which uses the EM algorithm to partition the data prior to the calculation of the new criterion is presented. The penalty term of the new criterion is further refined in Section 2.7 by replacing an asymptotic approximation with the exact expression. In Section 2.8, the derived cluster enumeration criteria are extended to a sensor network setup where the task is to estimate the number of clusters in a streaming-in data collected by spatially distributed sensors. Finally, the chapter is summarized in Section 2.9.

2.2 State-of-the-art

Statistical model selection is concerned with choosing a model that adequately explains the observed data from a family of candidate models. Over the years, many model selection criteria have been proposed in the literature, see for example [Jeffreys, 1961; Akaike, 1969; Akaike, 1970; Akaike, 1973; Allen, 1974; Stone, 1974; Rissanen, 1978; Schwarz, 1978; Hannan & Quinn, 1979; Shibata, 1980; Rao & Wu, 1989; Breiman, 1992; Kass & Raftery, 1995; Shao, 1996; Djurić, 1998; Cavanaugh & Neath, 1999; Zoubir, 1999; Zoubir & Iskander, 2000; Brcich et al., 2002; Morelande & Zoubir, 2002; Spiegelhalter et al., 2002; Claeskens & Hjort, 2003; Lu & Zoubir, 2013b; Lu & Zoubir, 2013a; Lu & Zoubir, 2015] and the review in [Rao & Wu, 2001]. One of the prominent fields of study where statistical model selection criteria are extensively used is cluster analysis. A major challenge in cluster analysis is that the number of data clusters is mostly unknown and it must be estimated prior to clustering the observed data. The estimation of the number of clusters, also called cluster enumeration, has been intensively researched for decades, see [Kalogeratos & Likas, 2012; Hamerly & Charles, 2003; Pelleg & Moore, 2000; Shahbaba & Beheshti, 2012; Ishioka, 2005; Zhao et al., 2008a; Zhao et al., 2008b; Feng & Hamerly, 2007; Constantinopoulos et al., 2006; Huang et al., 2017; Fraley & Raftery, 1998; Mehrjou et al., 2016; Dasgupta & Raftery, 1998; Campbell et al., 1997; Mukherjee et al., 1998; Krzanowski & Lai, 1988; Tibshirani et al., 2001; Teklehaymanot et al., 2016; Binder et al., 2018; Dolatabadi et al., 2017; Caliński & Harabasz, 1974; Rousseeuw, 1987; Davies & Bouldin, 1979; Dunn, 1973] and the surveys in [Xu & Wunsch, 2005; Arbelaitz et al., 2013; Milligan & Cooper, 1985; Maulik & Bandyopadhyay, 2002; Halkidi et al., 2001]. A popular approach for cluster enumeration is to apply the Bayesian information criterion (BIC) [Ishioka, 2005; Fraley & Raftery, 1998; Zhao et al., 2008a; Zhao et al., 2008b; Pelleg & Moore, 2000; Mehrjou et al., 2016; Dasgupta & Raftery, 1998; Campbell et al., 1997; Mukherjee et al., 1998; Teklehaymanot et al., 2016; Dolatabadi et al., 2017].

The BIC finds the large sample limit of the Bayes' estimator which leads to the selection of a model that is a posteriori most probable. It is consistent if the true data generating model belongs to the family of candidate models under investigation [Schwarz, 1978]. The BIC was originally derived by Schwarz in [Schwarz, 1978] assuming that (a) the observations are independent and identically distributed (iid), (b) they arise from an exponential family of distribu-

tions, and (c) the candidate models are linear in parameters. Ignoring these rather restrictive assumptions, the BIC has been used in a much larger scope of model selection problems. A justification of the widespread applicability of the BIC was provided in [Cavanaugh & Neath, 1999] by generalizing Schwarz's derivation. In [Cavanaugh & Neath, 1999], the authors drop the first two assumptions made by Schwarz given that some regularity conditions are satisfied.

One of the prominent cluster enumeration algorithms that use the BIC is the X-means algorithm [Pelleg & Moore, 2000]. The X-means algorithm attempts to extend K-means with an estimation of the number of clusters by assuming that each cluster contains iid Gaussian data points and all clusters are spherical with an identical variance. These assumptions greatly simplify computation but are far from reality. Extensions of the X-means algorithm that replace the BIC with either the minimum noiseless description length (MNDL) or univariate hypothesis tests are presented in [Shahbaba & Beheshti, 2012], and [Hamerly & Charles, 2003; Feng & Hamerly, 2007], respectively.

The BIC is a generic criterion in the sense that it does not incorporate information regarding the specific model selection problem at hand. As a result, it penalizes two structurally different models the same way if they have the same number of unknown parameters. The work in [Djurić, 1998] has shown that model selection rules that penalize for model complexity have to be examined carefully before they are applied to specific model selection problems. Nevertheless, despite the widespread use of the BIC for cluster enumeration, very little effort has been made to check the appropriateness of the original BIC formulation [Cavanaugh & Neath, 1999] for cluster analysis. One noticeable work towards this direction was made in [Mehrjou et al., 2016] by providing a more accurate approximation to the marginal likelihood for small sample sizes. This derivation was made specifically for mixture models assuming that they are well separated. The resulting expression contains the original BIC term plus some additional terms that are based on the mixing probability and the Fisher information matrix (FIM) of each partition. The method presented in [Mehrjou et al., 2016] requires the calculation of the FIM for each cluster in each candidate model, which is computationally very expensive and impractical in real-world applications with high-dimensional data. This greatly limits its applicability. Other than the above mentioned work, to the best of our knowledge, no one has thoroughly investigated the derivation of the BIC for cluster analysis using large sample approximations.

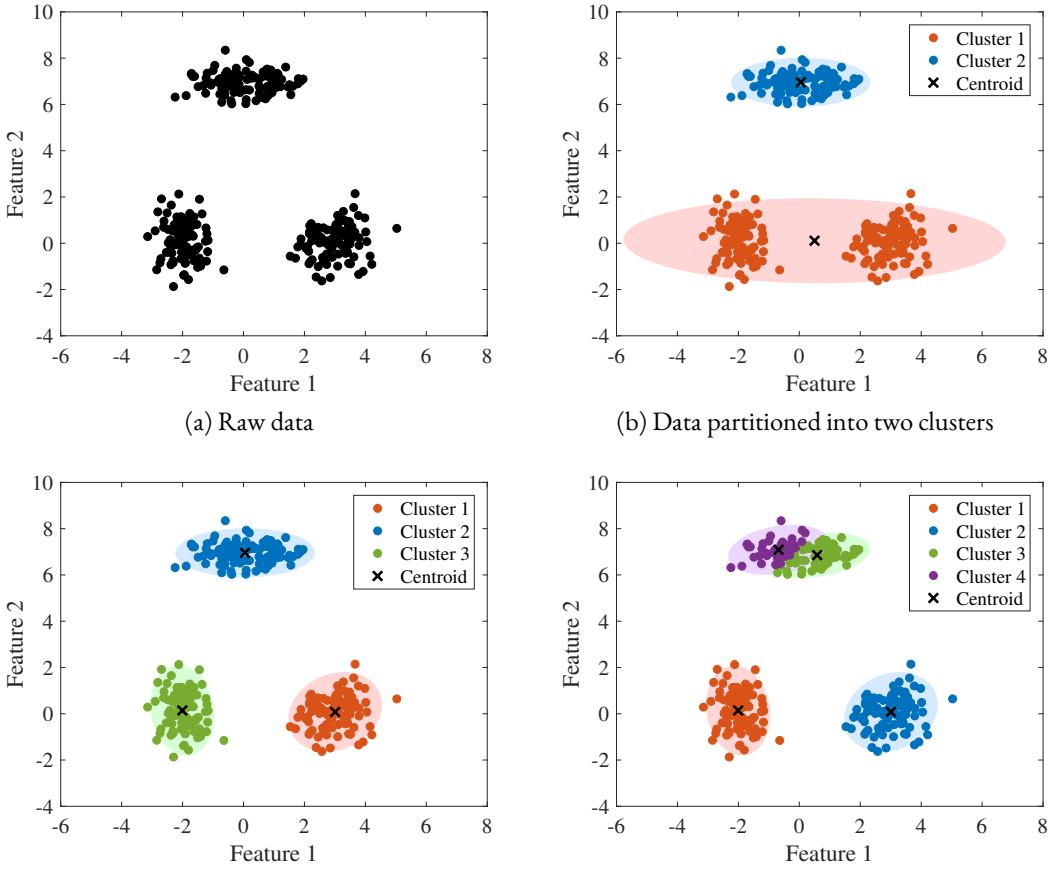
2.3 Contributions in this Chapter

Our first contribution lies in the derivation of a new BIC by formulating the problem of estimating the number of clusters as maximization of the posterior probability of candidate models. Under some mild assumptions, we provide a general expression for the BIC, which is applicable to a broad class of data distributions. This serves as a starting point when deriving the BIC for specific data distributions in cluster analysis. Along this line, we derive a closed-form BIC expression by modeling the data as a family of multivariate Gaussian distributions. Further more, to mitigate the shortcomings of clustering methods that require the number of clusters as an input, we present a two-step cluster enumeration algorithm which provides a principled way of estimating the number of clusters by utilizing existing clustering algorithms. The two-step algorithm uses a model-based unsupervised learning method to partition the data into the number of clusters provided by the candidate model prior to the calculation of the new BIC for that particular model. A major advantage of the new BIC is that it can be used as a wrapper around any clustering algorithm.

The second contribution is the refinement of the penalty term of the new BIC for the finite sample regime, which results in a cluster enumeration criterion with a strong penalty term. Hence, it is able to estimate the correct number of clusters in data sets with small sample sizes. In the asymptotic regime, both criteria behave in the same way.

The third contribution lies in proposing two distributed and adaptive Bayesian cluster enumeration algorithms by extending the newly derived criteria to a distributed sensor network setup where the nodes exchange valuable information via the diffusion principle [Sayed et al., 2013]. The proposed methods are applied to a camera network use case, where multiple users film a non-stationary scene from different angles using their camera equipped portable devices. The number of pedestrians is estimated based on streaming-in feature vectors without assuming prior information, such as known positions of the devices, registration of camera views or the availability of a fusion center.

The first and second contributions have been published in [Teklehaymanot et al., 2018a] and [Teklehaymanot et al., 2018d], respectively. The third contribution has been published in [Teklehaymanot et al., 2018b], while some preliminary work in the area has been published in [Teklehaymanot et al., 2016].



(a) Raw data

(b) Data partitioned into two clusters

(c) Data partitioned into three clusters

(d) Data partitioned into four clusters

Figure 2.1: Partitioning of a data set based on the number of clusters specified by different candidate models.

2.4 Problem Formulation

Given a set of r-dimensional vectors $\mathcal{X} \triangleq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, let $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$ be a partition of $\mathcal{X}$ into K clusters $\mathcal{X}_k \subseteq \mathcal{X}$ for $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$. The subsets (clusters) $\mathcal{X}_k, k \in \mathcal{K}$, are independent, mutually exclusive, and non-empty. Let $\mathcal{M} \triangleq \{M_{L_{\min}}, \ldots, M_{L_{\max}}\}$ be a family of candidate models. Each candidate model $M_l \in \mathcal{M}$ represents the partitioning of $\mathcal{X}$ into $l \in \{L_{\min}, \ldots, L_{\max}\}$ subsets, where $l \in \mathbb{Z}^+$. The parameters of each model $M_l \in \mathcal{M}$ are denoted by $\boldsymbol{\Theta}_l = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_l]$ which lies in a parameter space $\Omega_l \subset \mathbb{R}^{q \times l}$. An example that illustrates the partitioning of a synthetic data set according to the number of clusters specified by different candidate models is displayed in Figure 2.1. Our research goal is to choose the model $M_{\hat{K}} \in \mathcal{M}$, where $\hat{K} \in \{L_{\min}, \ldots, L_{\max}\}$, which is most probable a

posteriori assuming that

(A-2.1) the true number of clusters (K) in the observed data set $\mathcal{X}$ satisfies the constraint $L_{\min} \leq K \leq L_{\max}$.

2.5 Generic Bayesian Cluster Enumeration Criterion

In the considered Bayesian setting, estimating the number of clusters in a given data set corresponds to choosing the model $M_{\hat{K}} \in \mathcal{M}$ which is a posteriori most probable [Teklehaymanot et al., 2018a]. Mathematically, this corresponds to solving

$$M_{\hat{K}} = \arg\max_{\mathcal{M}} p(M_l|\mathcal{X}), \tag{2.1}$$

where $p(M_l|\mathcal{X})$ is the posterior probability of $M_l \in \mathcal{M}$ given the observations $\mathcal{X}$. The selected model $M_{\hat{K}}$ explains how the data set $\mathcal{X}$ should be partitioned.

Solving (2.1) starts with writing $p(M_l|\mathcal{X})$ as

$$p(M_l|\mathcal{X}) = \int_{\Omega_l} f(M_l, \boldsymbol{\Theta}_l|\mathcal{X}) d\boldsymbol{\Theta}_l, \tag{2.2}$$

where $f(M_l, \boldsymbol{\Theta}_l|\mathcal{X})$ is the joint posterior density of M_l and $\boldsymbol{\Theta}_l$ given $\mathcal{X}$. According to Bayes' theorem

$$f(M_l, \boldsymbol{\Theta}_l|\mathcal{X}) = \frac{p(M_l) f(\boldsymbol{\Theta}_l|M_l) f(\mathcal{X}|M_l, \boldsymbol{\Theta}_l)}{f(\mathcal{X})}, \tag{2.3}$$

where $p(M_l)$ is the discrete prior on the model $M_l \in \mathcal{M}$, $f(\boldsymbol{\Theta}_l|M_l)$ is a prior on the parameter vectors in $\boldsymbol{\Theta}_l$ given M_l, $f(\mathcal{X}|M_l, \boldsymbol{\Theta}_l)$ is the probability density function (pdf) of the observation set $\mathcal{X}$ given M_l and $\boldsymbol{\Theta}_l$, and $f(\mathcal{X})$ is the pdf of $\mathcal{X}$. Substituting (2.3) into (2.2), we obtain

$$p(M_l|\mathcal{X}) = f(\mathcal{X})^{-1} p(M_l) \int_{\Omega_l} f(\boldsymbol{\Theta}_l|M_l) \mathcal{L}(\boldsymbol{\Theta}_l|\mathcal{X}) d\boldsymbol{\Theta}_l, \tag{2.4}$$

where $\mathcal{L}(\boldsymbol{\Theta}_l|\mathcal{X}) \triangleq f(\mathcal{X}|M_l, \boldsymbol{\Theta}_l)$ is the likelihood function. Since log is a monotonic func-

tion, $M_{\hat{K}}$ can also be determined via

$$\arg\max_{\mathcal{M}} \ \log p(M_l|\mathcal{X}) \tag{2.5}$$

instead of (2.1). Hence, taking the logarithm of (2.4) results in

$$\log p(M_l|\mathcal{X}) = \log p(M_l) + \log \int_{\Omega_l} f(\boldsymbol{\Theta}_l|M_l)\mathcal{L}(\boldsymbol{\Theta}_l|\mathcal{X})d\boldsymbol{\Theta}_l - \log f(\mathcal{X}). \tag{2.6}$$

Since the partitions (clusters) $\mathcal{X}_m \subseteq \mathcal{X}, m = 1, \ldots, l$, are independent, mutually exclusive, and non-empty, $f(\boldsymbol{\Theta}_l|M_l)$ and $\mathcal{L}(\boldsymbol{\Theta}_l|\mathcal{X})$ can be written as

$$f(\boldsymbol{\Theta}_l|M_l) = \prod_{m=1}^{l} f(\boldsymbol{\theta}_m|M_l) \tag{2.7}$$

$$\mathcal{L}(\boldsymbol{\Theta}_l|\mathcal{X}) = \prod_{m=1}^{l} \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m). \tag{2.8}$$

Substituting (2.7) and (2.8) into (2.6) results in

$$\log p(M_l|\mathcal{X}) = \log p(M_l) + \sum_{m=1}^{l} \log \int_{\mathbb{R}^q} f(\boldsymbol{\theta}_m|M_l)\mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)d\boldsymbol{\theta}_m - \log f(\mathcal{X}). \tag{2.9}$$

Maximizing $\log p(M_l|\mathcal{X})$ over all candidate models $M_l \in \mathcal{M}$ involves the computation of the logarithm of a multidimensional integral. Unfortunately, the solution of the multidimensional integral does not possess a closed analytical form for most practical cases. This problem can be solved using either numerical integration or approximations that allow a closed-form solution. In the context of model selection, closed-form approximations are known to provide more insight into the problem than numerical integration [Djurić, 1998]. Following this line of argument, we use Laplace's method of integration [Djurić, 1998; Stoica & Selen, 2004; Ando, 2010] to simplify the multidimensional integral in (2.9).

Laplace's method of integration makes the following assumptions.

(A-2.2) $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$, for $m = 1, \ldots, l$, has first- and second-order derivatives which are continuous over the parameter space Ω_l.

(A-2.3) $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$, for $m = 1, \ldots, l$, has a global maximum at $\hat{\boldsymbol{\theta}}_m$, where $\hat{\boldsymbol{\theta}}_m$ is an interior point of Ω_l.

(A-2.4) $f(\boldsymbol{\theta}_m|M_l)$, for $m = 1, \ldots, l$, is continuously differentiable and its first-order derivatives are bounded on Ω_l with $f(\hat{\boldsymbol{\theta}}_m|M_l) \neq 0$.

(A-2.5) The negative of the Hessian matrix of $\frac{1}{N_m} \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$

$$\hat{\boldsymbol{H}}_m \triangleq -\frac{1}{N_m} \frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m} \in \mathbb{R}^{q\times q}, \tag{2.10}$$

where N_m is the number of data points in the mth cluster, is positive definite. That is, $\min_{s,m} \lambda_s(\hat{\boldsymbol{H}}_m) > \epsilon$ for $s = 1, \ldots, q$ and $m = 1, \ldots, l$, where $\lambda_s(\hat{\boldsymbol{H}}_m)$ is the sth eigenvalue of $\hat{\boldsymbol{H}}_m$ and ϵ is a small positive constant.

The first step in Laplace's method of integration is to write the Taylor series expansion of $f(\boldsymbol{\theta}_m|M_l)$ and $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ around $\hat{\boldsymbol{\theta}}_m$, for $m = 1, \ldots, l$. We begin by approximating $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ by its second-order Taylor series expansion around $\hat{\boldsymbol{\theta}}_m$ as follows:

$$\begin{aligned}\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m) &\approx \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) + \tilde{\boldsymbol{\theta}}_m^\top \frac{d \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m} \\ &\quad + \frac{1}{2}\tilde{\boldsymbol{\theta}}_m^\top \left[\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m}\right] \tilde{\boldsymbol{\theta}}_m \\ &= \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) - \frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m,\end{aligned} \tag{2.11}$$

where $\tilde{\boldsymbol{\theta}}_m \triangleq \boldsymbol{\theta}_m - \hat{\boldsymbol{\theta}}_m$, for $m = 1, \ldots, l$. The first derivative of $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ evaluated at $\hat{\boldsymbol{\theta}}_m$ vanishes because of assumption (A-2.3). With

$$U \triangleq \int_{\mathbb{R}^q} f(\boldsymbol{\theta}_m|M_l) \exp\left(\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)\right) d\boldsymbol{\theta}_m, \tag{2.12}$$

substituting (2.11) into (2.12) and approximating $f(\boldsymbol{\theta}_m|M_l)$ by its Taylor series expansion yields

$$U \approx \int_{\mathbb{R}^q} \left(f(\hat{\boldsymbol{\theta}}_m|M_l) + \tilde{\boldsymbol{\theta}}_m^\top \frac{df(\boldsymbol{\theta}_m|M_l)}{d\boldsymbol{\theta}_m}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m} + \text{HOT}\right) \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) \exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right) d\boldsymbol{\theta}_m, \tag{2.13}$$

where HOT denotes higher order terms and $\exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right)$ is a Gaussian kernel with mean $\hat{\boldsymbol{\theta}}_m$ and covariance matrix $(N_m \hat{\boldsymbol{H}}_m)^{-1}$. Ignoring the higher order terms, (2.13) can be simplified to

$$\begin{aligned}
U &\approx f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) \int_{\mathbb{R}^q} \exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right) d\boldsymbol{\theta}_m \\
&\quad + \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) \int_{\mathbb{R}^q} \tilde{\boldsymbol{\theta}}_m^\top \frac{df(\boldsymbol{\theta}_m|M_l)}{d\boldsymbol{\theta}_m}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m} \exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right) d\boldsymbol{\theta}_m \\
&= f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) \int_{\mathbb{R}^q} \exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right) d\boldsymbol{\theta}_m \\
&= f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) \int_{\mathbb{R}^q} \frac{(2\pi)^{q/2}\left|N_m^{-1}\hat{\boldsymbol{H}}_m^{-1}\right|^{1/2}}{(2\pi)^{q/2}\left|N_m^{-1}\hat{\boldsymbol{H}}_m^{-1}\right|^{1/2}} \exp\left(-\frac{N_m}{2}\tilde{\boldsymbol{\theta}}_m^\top \hat{\boldsymbol{H}}_m \tilde{\boldsymbol{\theta}}_m\right) d\boldsymbol{\theta}_m \\
&= f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)(2\pi)^{q/2}\left|N_m^{-1}\hat{\boldsymbol{H}}_m^{-1}\right|^{1/2} \qquad (2.14)
\end{aligned}$$

given that $N_m \to \infty$, where $|\cdot|$ stands for the determinant. The term in the second line of (2.14) vanishes because it simplifies to $\kappa\,\mathbb{E}[\boldsymbol{\theta}_m - \hat{\boldsymbol{\theta}}_m] = 0$, where $\kappa < \infty$ is a constant (see [Ando, 2010, p. 53] for more detail). Now, substituting (2.14) into (2.9), we arrive at

$$\begin{aligned}
\log p(M_l|\mathcal{X}) \approx \log p(M_l) + \sum_{m=1}^{!} \log\Big(f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)\Big) + \frac{lq}{2}\log 2\pi \\
- \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{J}}_m| - \log f(\mathcal{X}), \qquad (2.15)
\end{aligned}$$

where

$$\hat{\boldsymbol{J}}_m \triangleq N_m \hat{\boldsymbol{H}}_m = -\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top}\bigg|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m} \in \mathbb{R}^{q\times q} \qquad (2.16)$$

is the Fisher information matrix (FIM) of the data vectors from the mth partition.

In the derivation of $\log p(M_l|\mathcal{X})$, so far, we have made no distributional assumption on the data set $\mathcal{X}$ except that the log-likelihood function $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ and the prior on the parameter vectors $f(\boldsymbol{\theta}_m|M_l)$, for $m = 1, \ldots, l$, should satisfy some mild conditions under each model $M_l \in \mathcal{M}$. Hence, (2.15) is a general expression of the posterior probability of the model M_l given $\mathcal{X}$ for a general class of data distributions that satisfy assumptions (A-2.2)–

(A-2.5). The BIC is concerned with the computation of the posterior probability of candidate models and thus (2.15) can also be written as

$$\begin{aligned}\mathrm{BIC}_{\mathrm{G}}(M_l) &\triangleq \log p(M_l|\mathcal{X}) \\ &\approx \log p(M_l) + \log f(\hat{\boldsymbol{\Theta}}_l|M_l) + \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) + \frac{lq}{2}\log 2\pi \\ &\quad - \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{J}}_m| - \log f(\mathcal{X}). \end{aligned} \tag{2.17}$$

After calculating $\mathrm{BIC}_{\mathrm{G}}(M_l)$ for each candidate model $M_l \in \mathcal{M}$, the number of clusters in $\mathcal{X}$ is estimated as

$$\hat{K}_{\mathrm{BIC_G}} = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \mathrm{BIC}_{\mathrm{G}}(M_l). \tag{2.18}$$

However, calculating $\mathrm{BIC}_{\mathrm{G}}(M_l)$ using (2.17) is a computationally expensive task as it requires the estimation of the FIM, $\hat{\boldsymbol{J}}_m$, for each cluster $m = 1, \ldots, l$ in the candidate model $M_l \in \mathcal{M}$. Our objective is to find an asymptotic approximation for $|\hat{\boldsymbol{J}}_m|$, for $m = 1, \ldots, l$, in order to simplify the computation of $\mathrm{BIC}_{\mathrm{G}}(M_l)$. We solve this problem by imposing specific assumptions on the distribution of the data set $\mathcal{X}$. In the next section, we provide an asymptotic approximation for $|\hat{\boldsymbol{J}}_m|$, for $m = 1, \ldots, l$, assuming that each cluster $\mathcal{X}_m$ contains iid multivariate Gaussian data points.

2.6 Bayesian Cluster Enumeration Algorithm for Multivariate Gaussian Data

In this section, we first derive an asymptotic cluster enumeration criterion by modeling the data as a family of Gaussian distributions. Then, we present a two-step approach that uses the EM algorithm to partition the data according to each candidate model prior to the calculation of an enumeration criterion. In addition, we conduct numerical and real data experiments and demonstrate the performance of the proposed cluster enumeration algorithm in comparison to existing methods.

2.6.1 Bayesian Cluster Enumeration Criterion for Multivariate Gaussian Data

Let $\mathcal{X} \triangleq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ denote the observed data set which can be partitioned into K clusters $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$. Each cluster $\mathcal{X}_k, k \in \mathcal{K}$, contains N_k data vectors that are realizations of iid Gaussian random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k \in \mathbb{R}^{r\times 1}$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{r\times r}$ represent the centroid and the covariance matrix of the kth cluster, respectively. Further, let $\mathcal{M} \triangleq \{M_{L_{\min}}, \ldots, M_{L_{\max}}\}$ denote a set of Gaussian candidate models and let there be a clustering algorithm that partitions $\mathcal{X}$ into l independent, mutually exclusive, and non-empty subsets (clusters) $\mathcal{X}_m$, for $m = 1, \ldots, l$, by providing parameter estimates $\hat{\boldsymbol{\theta}}_m = [\hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m]^\top$ for each candidate model $M_l \in \mathcal{M}$, where $l \in \{L_{\min}, \ldots, L_{\max}\}$ and $l \in \mathbb{Z}^+$. Assume that (A-2.1)–(A-2.6) are satisfied.

Theorem 2.1. *The posterior probability of $M_l \in \mathcal{M}$ given $\mathcal{X}$ can be asymptotically approximated as*

$$\begin{aligned} \mathrm{BIC}_{\mathrm{N}}(M_l) &\triangleq \log p(M_l|\mathcal{X}) \\ &\approx \textstyle\sum_{m=1}^{l} N_m \log N_m - \frac{1}{2}\sum_{m=1}^{l} N_m \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{q}{2}\sum_{m=1}^{l}\log N_m, \end{aligned} \tag{2.19}$$

where $q = \frac{1}{2}r(r+3)$ is the number of estimated parameters per cluster and N_m is the number of data points in the mth cluster, which satisfies the condition $N = \sum_{m=1}^{l} N_m$.

Proof. Proving Theorem 2.1 requires finding an asymptotic approximation to $|\hat{\boldsymbol{J}}_m|$ in (2.17) and, based on this approximation, deriving an expression for $\mathrm{BIC}_{\mathrm{N}}(M_l)$. A detailed proof is given in Appendix B.1. ■

Once $\mathrm{BIC}_{\mathrm{N}}(M_l)$ is computed for each candidate model $M_l \in \mathcal{M}$, the number of partitions (clusters) in $\mathcal{X}$ is estimated as

$$\hat{K}_{\mathrm{BIC}_{\mathrm{N}}} = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \mathrm{BIC}_{\mathrm{N}}(M_l). \tag{2.20}$$

Remark. *The proposed criterion,* $\mathrm{BIC}_{\mathrm{N}}$*, and the original BIC as derived in [Schwarz, 1978; Cavanaugh & Neath, 1999] differ in terms of their penalty terms. A detailed discussion is provided in Section 2.6.4.*

The first step in calculating $\mathrm{BIC}_{\mathrm{N}}(M_l)$ for each model $M_l \in \mathcal{M}$ is the partitioning of the data set $\mathcal{X}$ into l clusters $\mathcal{X}_m$, where $m = 1, \ldots, l$, and the estimation of the associated cluster parameters using an unsupervised learning algorithm. Since the approximations in $\mathrm{BIC}_{\mathrm{N}}(M_l)$ are based on maximizing the likelihood function of Gaussian distributed random variables, we use a clustering algorithm that is based on the maximum likelihood principle. Accordingly, a natural choice is the EM algorithm for Gaussian mixture models.

2.6.2 The Expectation Maximization Algorithm for Gaussian Mixture Models

The EM algorithm finds maximum likelihood solutions for models with latent variables [Dempster et al., 1977; Bishop, 2006]. In our case, the latent variables are the cluster memberships of the data vectors in $\mathcal{X}$, given that the l-component Gaussian mixture distribution of a data vector $\boldsymbol{x}_n$ can be written as

$$f(\boldsymbol{x}_n | M_l, \boldsymbol{\Theta}_l) = \sum_{m=1}^{l} \tau_m g(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \tag{2.21}$$

where $g(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ represents the r-variate Gaussian pdf and τ_m is the mixing coefficient of the mth cluster. The goal of the EM algorithm is to maximize the log-likelihood function of the data set $\mathcal{X}$ with respect to the parameters of interest as follows:

$$\arg\max_{\boldsymbol{\Phi}_l} \log \mathcal{L}(\boldsymbol{\Phi}_l | \mathcal{X}) = \arg\max_{\boldsymbol{\Phi}_l} \sum_{n=1}^{N} \log \sum_{m=1}^{l} \tau_m g(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m), \tag{2.22}$$

where $\boldsymbol{\Phi}_l = [\boldsymbol{\tau}_l, \boldsymbol{\Theta}_l^\top]$ and $\boldsymbol{\tau}_l = [\tau_1, \ldots, \tau_l]^\top$. Maximizing (2.22) with respect to the elements of $\boldsymbol{\Phi}_l$ results in coupled equations. The EM algorithm solves these coupled equations using a two-step iterative procedure. The first step (E step) evaluates $\hat{\upsilon}_{nm}^{(i)}$, which is an estimate of the probability that data vector $\boldsymbol{x}_n$ belongs to the mth cluster at the ith iteration, for $n = 1, \ldots, N$ and $m = 1, \ldots, l$. $\hat{\upsilon}_{nm}^{(i)}$ is calculated as

$$\hat{\upsilon}_{nm}^{(i)} = \frac{\hat{\tau}_m^{(i-1)} g(\boldsymbol{x}_n; \hat{\boldsymbol{\mu}}_m^{(i-1)}, \hat{\boldsymbol{\Sigma}}_m^{(i-1)})}{\sum_{j=1}^{l} \hat{\tau}_j^{(i-1)} g(\boldsymbol{x}_n; \hat{\boldsymbol{\mu}}_j^{(i-1)}, \hat{\boldsymbol{\Sigma}}_j^{(i-1)})}, \tag{2.23}$$

where $\hat{\boldsymbol{\mu}}_m^{(i-1)}$ and $\hat{\boldsymbol{\Sigma}}_m^{(i-1)}$ represent the centroid and covariance matrix estimates, respectively, of the mth cluster at the previous iteration $(i-1)$. The second step (M step) re-estimates the cluster parameters using the current values of $\hat{v}_{nm}$ as follows:

$$\hat{\boldsymbol{\mu}}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)} \boldsymbol{x}_n}{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)}} \tag{2.24}$$

$$\hat{\boldsymbol{\Sigma}}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m^{(i)})(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m^{(i)})^\top}{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)}} \tag{2.25}$$

$$\hat{\tau}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)}}{N} \tag{2.26}$$

The E and M steps are performed iteratively until either the cluster parameter estimates $\hat{\boldsymbol{\Phi}}_l$ or the estimate of the log-likelihood function $\log \mathcal{L}(\hat{\boldsymbol{\Phi}}_l | \mathcal{X})$ converges.

A summary of the estimation of the number of clusters in an observed data set using the two-step approach is provided in Algorithm 2.1. Note that the computational complexity of $\mathrm{BIC}_{\mathrm{N}}(M_l)$ is only $\mathcal{O}(1)$, which can easily be ignored during the run-time analysis of the proposed two-step cluster enumeration algorithm. Hence, since the EM algorithm is run for all candidate models in $\mathcal{M}$, the computational complexity of the proposed algorithm is $\mathcal{O}(Nr^2 \left(L_{\min} + \ldots + L_{\max}\right) i_{\max})$, where $i_{\max}$ is a fixed stopping threshold of the EM algorithm.

2.6.3 Existing BIC-Based Cluster Enumeration Methods

As discussed in Section 2.2, existing cluster enumeration algorithms that are based on the original BIC use the criterion as it is known from parameter estimation tasks without questioning its validity on cluster analysis. Nevertheless, since these criteria have been widely used, we briefly review them to provide a comparison to our criterion $\mathrm{BIC}_{\mathrm{N}}$, which is given by (2.19).

The original BIC, as derived in [Schwarz, 1978; Cavanaugh & Neath, 1999], evaluated at a candidate model $M_l \in \mathcal{M}$ is written as

$$\mathrm{BIC}_{\mathrm{O}}(M_l) = \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l | \mathcal{X}) - \frac{ql}{2} \log N, \tag{2.27}$$

where $\mathcal{L}(\hat{\boldsymbol{\Theta}}_l | \mathcal{X})$ denotes the likelihood function, q is the number of estimated parameters in

Algorithm 2.1 Two-step cluster enumeration algorithm

Inputs: $\mathcal{X}$, $L_{\min}$, and $L_{\max}$
for $l = L_{\min}, \ldots, L_{\max}$ do
  Step 1: model-based clustering
  Step 1.1: the EM algorithm
  for $m = 1, \ldots, l$ do
    Initialize $\boldsymbol{\mu}_m$ using K-means++ [Arthur & Vassilvitskii, 2007]
    $\hat{\boldsymbol{\Sigma}}_m = \frac{1}{N_m} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)^\top$
    $\hat{\tau}_m = \frac{N_m}{N}$
  end for
  for $i = 1, 2, \ldots, i_{\max}$ do
    E step:
    for $n = 1, \ldots, N$ do
      for $m = 1, \ldots, l$ do
        Calculate $\hat{\upsilon}_{nm}^{(i)}$ using (2.23)
      end for
    end for
    M step:
    for $m = 1, \ldots, l$ do
      Determine $\hat{\boldsymbol{\mu}}_m^{(i)}$, $\hat{\boldsymbol{\Sigma}}_m^{(i)}$, and $\hat{\tau}_m^{(i)}$ via (2.24)-(2.26)
    end for
    Check for convergence of either $\hat{\boldsymbol{\Phi}}_l^{(i)}$ or $\log \mathcal{L}(\hat{\boldsymbol{\Phi}}_l^{(i)} | \mathcal{X})$
    if convergence condition is satisfied then
      Exit for loop
    end if
  end for
  Step 1.2: hard clustering
  for $n = 1, \ldots, N$ do
    for $m = 1, \ldots, l$ do

$$\iota_{nm} = \begin{cases} 1, & m = \underset{j=1,\ldots,l}{\arg\max}\ \hat{\upsilon}_{nj}^{(i)} \\ 0, & \text{otherwise} \end{cases}$$

    end for
  end for
  for $m = 1, \ldots, l$ do
    $N_m = \sum_{n=1}^{N} \iota_{nm}$
  end for
  Step 2: calculate $\mathrm{BIC}_{\mathrm{N}}(M_l)$ via (2.19)
end for
Estimate the number of clusters, $\hat{K}_{\mathrm{BIC}_{\mathrm{N}}}$, in $\mathcal{X}$ via (2.20)

Algorithm 2.2 Cluster enumeration using BIC_{os}

Inputs: $\mathcal{X}$, $L_{\min}$, and $L_{\max}$
for $l = L_{\min}, \ldots, L_{\max}$ do
  for $m = 1, \ldots, l$ do
    Estimate N_m and $\boldsymbol{\mu}_m$ using K-means++ [Arthur & Vassilvitskii, 2007]
    Calculate $\hat{\sigma}^2$ using (2.30)
  end for
  Calculate $\text{BIC}_{\text{os}}(M_l)$ via (2.29)
end for
Estimate the number of clusters in $\mathcal{X}$ as
$\hat{K}_{\text{BIC}_{\text{OS}}} = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max} \ \text{BIC}_{\text{os}}(M_l)$

the candidate model M_l, and $N = \#\mathcal{X}$. In (2.27), $\log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X})$ denotes the data fidelity term, while $\frac{ql}{2} \log N$ is the penalty term. Under the assumption that the observed data is Gaussian distributed, the data fidelity terms of BIC_{O} and the ones of our criterion, BIC_{N}, are exactly the same. The only deference between the two is the penalty term. Hence, we use a similar procedure as in Algorithm 2.1 to implement the original BIC as a wrapper around the EM algorithm.

Moreover, the original BIC is commonly used as a wrapper around K-means by assuming that the data points that belong to each cluster are iid as Gaussian and all clusters are spherical with an identical variance, i.e. $\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}_j = \sigma^2 \boldsymbol{I}_r$ for $m \neq j$, where σ^2 is the common variance of the clusters in M_l [Pelleg & Moore, 2000; Zhao et al., 2008a; Zhao et al., 2008b]. Under these assumptions, the original BIC is given by

$$\text{BIC}_{\text{os}}(M_l) = \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) - \frac{\alpha}{2} \log N, \tag{2.28}$$

where $\text{BIC}_{\text{os}}(M_l)$ denotes the original BIC of the candidate model M_l derived under the assumptions stated above and $\alpha = (rl + 1)$ is the number of estimated parameters in $M_l \in \mathcal{M}$. Ignoring the model independent terms, $\text{BIC}_{\text{os}}(M_l)$ can be written as

$$\text{BIC}_{\text{os}}(M_l) = \sum_{m=1}^{l} N_m \log N_m - \frac{rN}{2} \log \hat{\sigma}^2 - \frac{\alpha}{2} \log N, \tag{2.29}$$

where

$$\hat{\sigma}^2 = \frac{1}{rN} \sum_{m=1}^{l} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)^\top (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m) \tag{2.30}$$

is the maximum likelihood estimator of the common variance. Algorithm 2.2 summarizes cluster enumeration using BIC_{os}.

2.6.4 Comparison of the Penalty Terms of Different Bayesian Cluster Enumeration Criteria

Comparing (2.19), (2.27), and (2.28), we notice that they have a common form [Stoica & Selen, 2004; Rao & Wu, 1989], that is,

$$\log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l | \mathcal{X}) - \eta, \tag{2.31}$$

but with different penalty terms, where

$$\text{BIC}_{\text{N}}: \quad \eta = \frac{q}{2} \sum_{m=1}^{l} \log N_m \tag{2.32}$$

$$\text{BIC}_{\text{O}}: \quad \eta = \frac{ql}{2} \log N \tag{2.33}$$

$$\text{BIC}_{\text{OS}}: \quad \eta = \frac{1}{2} (rl + 1) \log N. \tag{2.34}$$

Remark. BIC_{O} *and* BIC_{OS} *carry information about the structure of the data only on their data fidelity term, which is the first term in* (2.31). *On the other hand, as shown in* (2.19), *both the data fidelity and penalty terms of our criterion,* BIC_{N}, *contain information about the structure of the data.*

The penalty terms of BIC_{O} and BIC_{OS} depend linearly on l, while the penalty term of our criterion, BIC_{N}, depends on l in a non-linear manner. Comparing the penalty terms in (2.32)-(2.34), BIC_{OS} has the weakest penalty term. In the asymptotic regime, the penalty terms of BIC_{N} and BIC_{O} coincide. But, in the finite sample regime, for values of $l > 1$, the penalty term of BIC_{O} is stronger than the penalty term of BIC_{N}. Note that the penalty term of BIC_{N} depends on the number of data vectors in each cluster, N_m, for $m = 1, \ldots, l$, of each candidate model $M_l \in \mathcal{M}$, while the penalty term of the original BIC depends only on the total number of

data vectors in the data set. Hence, the penalty term of our criterion might exhibit sensitivities to the initialization of cluster parameters and the associated number of data vectors per cluster.

2.6.5 Experimental Results

We compare the cluster enumeration performance of our criterion, BIC_{N} given by (2.19), with BIC_{O} and BIC_{OS}, which are given by (2.27) and (2.29), respectively, using five synthetic and four real data sets. The considered data sets are diverse in the sense that the number of features ranges from $r = 2$ up to $r = 79$, the number of samples ranges from $N = 150$ up to $N = 16,800$ and the number of clusters ranges from $K = 2$ up to $K = 20$. For all simulations, we set $L_{\min} = 1$ and $L_{\max} = 2K$, where K is the true number of clusters in the data set $\mathcal{X}$. All simulation results are an average of 1000 Monte Carlo experiments unless stated otherwise. The compared cluster enumeration criteria are based on the same initial cluster parameters in each Monte Carlo experiment, which allows for a fair comparison. The MATLAB code that implements the proposed two-step algorithm and the Bayesian cluster enumeration methods discussed in Section 2.6.3 is available in [Teklehaymanot et al., 2018c].

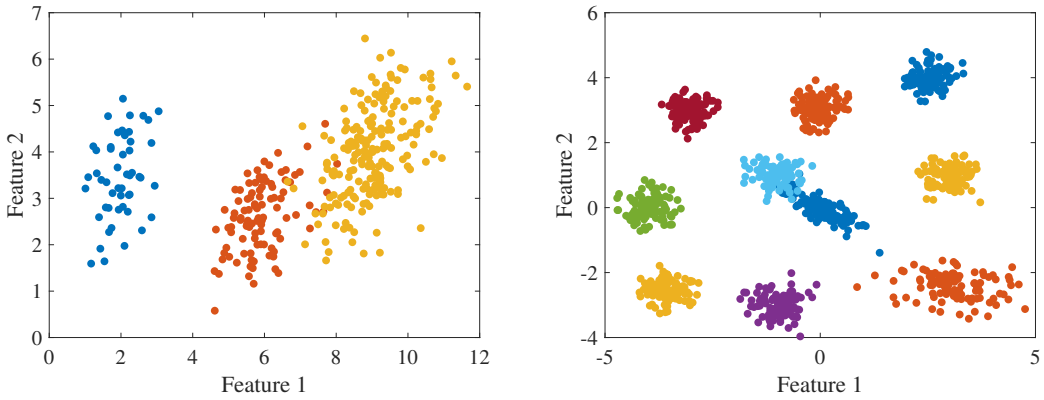
In this section, we first describe the performance measures used to compare the different cluster enumeration criteria. Then, the numerical experiments performed on synthetic data sets and the results obtained from real data sets are discussed in detail.

2.6.5.1 Performance Measures

The main performance measures are the empirical probability of detection (p_{det}), the empirical probability of underestimation (p_{under}), the empirical probability of selection, and the mean absolute error (MAE). The empirical probability of detection is defined as the probability with which the correct number of clusters is selected and it is calculated as

$$p_{\text{det}} = \frac{1}{I} \sum_{i=1}^{I} \mathbb{1}_{\{\hat{K}_i = K\}}, \tag{2.35}$$

where I is the total number of Monte Carlo experiments, $\hat{K}_i$ is the estimated number of clusters in the ith Monte Carlo experiment, and $\mathbb{1}_{\{\hat{K}_i = K\}}$ is the indicator function which is de-



(a) Data-2.1 for $\gamma = 1$ (b) Data-2.2 for $N_k = 100$

Figure 2.2: Synthetic data sets

fined as

$$\mathbb{1}_{\{\hat{K}_i = K\}} \triangleq \begin{cases} 1, & \text{if} \quad \hat{K}_i = K \\ 0, & \text{otherwise} \end{cases} . \tag{2.36}$$

The empirical probability of underestimation (p_{under}) is the probability that $\hat{K} < K$ and the empirical probability of overestimation (p_{over}) can be easily computed as

$$p_{\text{over}} = 1 - p_{\text{det}} - p_{\text{under}}. \tag{2.37}$$

The empirical probability of selection is defined as the probability with which the number of clusters specified by each candidate model $M_l \in \mathcal{M}$ is selected. The last performance measure, which is the mean absolute error (MAE), is computed as

$$\text{MAE} = \frac{1}{I} \sum_{i=1}^{I} \left| K - \hat{K}_i \right| . \tag{2.38}$$

2.6.5.2 Numerical Experiments

The numerical experiments are based on five synthetic data sets out of which three have been already used in the literature for cluster analysis tasks [Fränti & Virmajoki, 2006; Kärkkäinen & Fränti, 2002; Fränti et al., 2016]. We perform three experiments where we study the impact of cluster overlap, cluster unbalance, and initialization of cluster parameters on the overall

Table 2.1: The empirical probability of detection in %, the empirical probability of underestimation in %, and the mean absolute error (MAE) of various Bayesian cluster enumeration criteria as a function of γ for Data-2.1.

γ		1	3	6	12	48
$p_{\text{det}}(\%)$	BIC_{N}	**55.2**	**74.3**	**87.4**	**95.7**	**100**
	BIC_{O}	43.6	69.7	85.1	94.9	**100**
	BIC_{OS}	53.9	50.5	49.4	42.4	31.8
$p_{\text{under}}(\%)$	BIC_{N}	44.5	25.7	12.6	4.3	0
	BIC_{O}	56.4	30.3	14.9	5.1	0
	BIC_{OS}	0	0	0	0	0
MAE	BIC_{N}	**0.449**	**0.257**	**0.126**	**0.043**	**0**
	BIC_{O}	0.564	0.303	0.149	0.051	**0**
	BIC_{OS}	0.469	0.495	0.506	0.576	0.682

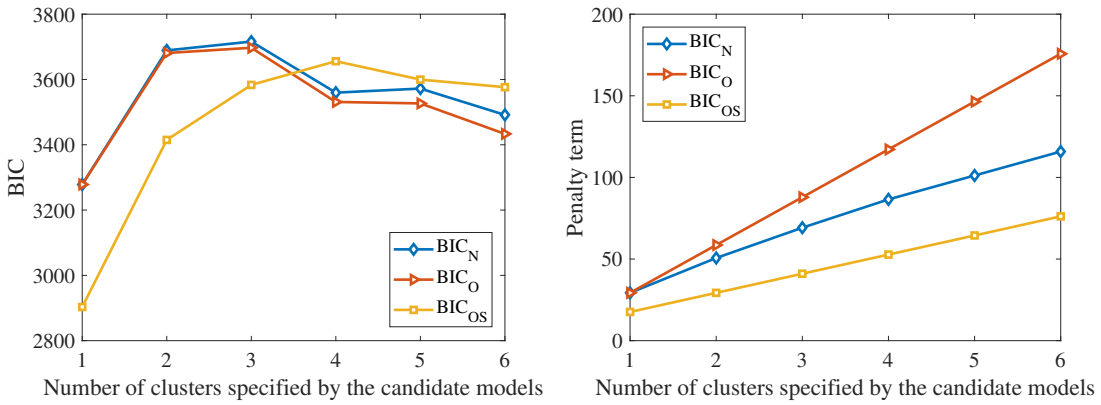
performance of different cluster enumeration methods.

In the first experiment, we consider a data set, referred to as Data-2.1 and depicted in Figure 2.2a, which contains realizations of the random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $k = 1, 2, 3$, with cluster centroids $\boldsymbol{\mu}_1 = [2, 3.5]^\top$, $\boldsymbol{\mu}_2 = [6, 2.7]^\top$, $\boldsymbol{\mu}_3 = [9, 4]^\top$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.75 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

The first cluster is linearly separable from the others, while the remaining clusters overlap. The number of data vectors per cluster is specified as $N_1 = \gamma \times 50$, $N_2 = \gamma \times 100$, and $N_3 = \gamma \times 200$, where γ is a constant. Data-2.1 is particularly challenging for cluster enumeration criteria because it has not only overlapping but also unbalanced clusters. Cluster unbalance refers to the fact that different clusters have a different number of data vectors, which might result in some clusters dominating the others.

The impact of cluster overlap and unbalance on p_{det} and MAE is displayed in Table 2.1. This table shows p_{det} and MAE as a function of γ, where γ is allowed to take values from the set $\{1, 3, 6, 12, 48\}$. The cluster enumeration performance of BIC_{OS} is lower than the other methods because it is designed for spherical clusters with identical variance, while Data-2.1 has one elliptical and two spherical clusters with different covariance matrices. Our criterion, BIC_{N}, performs best in terms of p_{det} and MAE for all values of γ. As γ increases, which corresponds to an increase in the number of data vectors in the data set, the cluster enumeration



(a) Total criteria (b) Penalty terms

Figure 2.3: The total criteria and penalty terms of different Bayesian cluster enumeration criteria for Data-2.1 when $\gamma = 1$.

performance of BIC_N and BIC_O greatly improves, while the performance of BIC_OS deteriorates because of the increase in overestimation. The total criterion (BIC) and penalty term of different Bayesian cluster enumeration criteria as a function of the number of clusters specified by the candidate models for $\gamma = 1$ is shown in Figure 2.3. The BIC plot in Figure 2.3a is the result of one Monte Carlo run. It shows that BIC_N and BIC_O have a maximum at the true number of clusters ($K = 3$), while BIC_OS overestimates the number of clusters to $\hat{K}_{\text{BIC}_\text{OS}} = 4$. As shown in Figure 2.3b, our criterion, BIC_N, has the second strongest penalty term. Note that, the penalty term of our criterion shows a curvature at the true number of clusters, while the penalty terms of BIC_O and BIC_OS are uninformative on their own.

In the second numerical experiment, we consider a data set, referred to as Data-2.2, which contains realizations of the random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $k = 1, \ldots, 10$, with cluster centroids $\boldsymbol{\mu}_1 = [0, 0]^\top$, $\boldsymbol{\mu}_2 = [3, -2.5]$, $\boldsymbol{\mu}_3 = [3, 1]^\top$, $\boldsymbol{\mu}_4 = [-1, -3]^\top$, $\boldsymbol{\mu}_5 = [-4, 0]^\top$, $\boldsymbol{\mu}_6 = [-1, 1]^\top$, $\boldsymbol{\mu}_7 = [-3, 3]^\top$, $\boldsymbol{\mu}_8 = [2.5, 4]^\top$, $\boldsymbol{\mu}_9 = [-3.5, -2.5]$, $\boldsymbol{\mu}_{10} = [0, 3]^\top$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.25 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.15 \end{bmatrix}, \boldsymbol{\Sigma}_i = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix},$$

where $i = 3, \ldots, 10$. As depicted in Figure 2.2b, Data-2.2 contains eight identical and spherical clusters and two elliptical clusters. There exists an overlap between two clusters, while the

Table 2.2: The empirical probability of detection in %, the empirical probability of underestimation in %, and the mean absolute error (MAE) of various Bayesian cluster enumeration criteria as a function of the number of data vectors per cluster (N_k) for Data-2.2.

N_k		100	200	500	1000
$p_{\text{det}}(\%)$	BIC_{N}	**56.1**	**66**	**81**	**85.3**
	BIC_{O}	41	57.1	78	84.9
	BIC_{OS}	2.7	0.9	0.1	0
$p_{\text{under}}(\%)$	BIC_{N}	37.6	30.2	18.2	13.5
	BIC_{O}	58.6	41.7	21.4	14.1
	BIC_{OS}	0	0	0	0
MAE	BIC_{N}	**0.452**	**0.341**	**0.19**	**0.148**
	BIC_{O}	0.59	0.429	0.22	0.151
	BIC_{OS}	1.613	1.659	1.745	1.8

rest of the clusters are well separated. All clusters in this data set have the same number of data vectors. Table 2.2 shows p_{det} and MAE as a function of the number of data vectors per cluster, $N_k, k = 1, \ldots, 10$, where N_k is allowed to take values from the set $\{100, 200, 500, 1000\}$. Our criterion, BIC_{N}, consistently outperforms the cluster enumeration methods that are based on the original BIC for the specified number of data vectors per cluster (N_k). BIC_{O} tends to underestimate the number of clusters to $\hat{K}_{\text{BIC}_{\text{O}}} = 9$ when N_k is small since it merges the two overlapping clusters. Even though majority of the clusters are spherical, BIC_{OS} rarely finds the correct number of clusters.

The overall performance of the two-step approach presented in Algorithm 2.1 depends on how well the clustering algorithm in the first step is able to partition the given data set. Clustering algorithms such as K-means and EM are known to converge to a local optimum and exhibit sensitivity to initialization of cluster parameters. The simplest initialization method is to randomly select cluster centroids from the set of data points. However, unless the random initializations are repeated sufficiently many times, the algorithms tend to converge to a poor local optimum. K-means++ [Arthur & Vassilvitskii, 2007] attempts to solve this problem by providing a systematic initialization to K-means. One can also use a few runs of K-means++ to initialize the EM algorithm. An alternative approach to the initialization problem is to use random swap [Fränti, 2018; Zhao et al., 2012]. Unlike repeated random initializations, random swap creates random perturbations to the solutions of K-means and EM in an attempt

Table 2.3: Summary of synthetic data sets in terms of their number of features (r), number of samples (N), number of samples per cluster (N_k), and number of clusters (K).

Data sets	r	N	N_k	K
S3 [Fränti & Virmajoki, 2006]	2	5000	333	15
A1 [Kärkkäinen & Fränti, 2002]	2	3000	150	20
G2-2-40 [Fränti et al., 2016]	2	2048	1024	2

to move the clustering result away from an inferior local optimum.

In the third experiment, we compare the performance of our criterion and the original BIC as wrappers around the above discussed clustering methods using five synthetic data sets, which include Data-2.1 with $\gamma = 6$, Data-2.2 with $N_k = 500$, and the ones summarized in Table 2.3. The number of random swaps is set to 100 and the results are an average of 100 Monte Carlo experiments. To allow for a fair comparison, the number of replicates required by the clustering methods that use K-means++ initialization is set equal to the number of random swaps. The empirical probability of detection (p_{det}) of our criterion and the original BIC as wrappers around the different clustering methods is depicted in Table 2.4, where RSK-means is the random swap K-means and RSEM is the random swap EM. BIC_{NS} is the implementation of our BIC as a wrapper around the K-means variants and is given by

$$\text{BIC}_{\text{NS}} = \sum_{m=1}^{l} N_m \log N_m - \frac{Nr}{2} \log \hat{\sigma}^2 - \frac{\alpha}{2} \sum_{m=1}^{l} \log N_m, \tag{2.39}$$

where $\alpha = r + 1$ and $\hat{\sigma}^2$ is given by (2.30). For the data sets that are mostly spherical, the K-means variants outperform the ones that are based on EM in terms of the correct estimation of the number of clusters, while, as expected, EM is superior for the elliptical data sets. Among the K-means variants, the gain obtained from using random swap instead of simple K-means++ is almost negligible. On the other hand, for the EM variants, EM significantly outperforms RSEM especially for BIC_{N}.

2.6.5.3 Real Data Results

Here, we study the performance of different cluster enumeration methods using real data sets. Out of the considered data sets, the first three are standard machine learning data sets [Lich-

Table 2.4: Empirical probability of detection in % of various Bayesian cluster enumeration criteria as wrappers around different clustering algorithms.

		Data-2.1	Data-2.2	S3	A1	G2-2-40
K-means++	BIC_{NS}	49	0	**100**	98	**100**
	BIC_{OS}	48	0	**100**	98	**100**
RSK-means	BIC_{NS}	49	0	**100**	**100**	**100**
	BIC_{OS}	48	0	**100**	**100**	**100**
EM	BIC_{N}	**87**	**92**	10	98	**100**
	BIC_{O}	85	89	10	98	**100**
RSEM	BIC_{N}	22	68	11	16	90
	BIC_{O}	85	89	9	28	97

man, 2013] and the last one has already been used in the literature for cluster enumeration [Binder et al., 2018; Teklehaymanot et al., 2016]. Although there is no randomness when repeating the experiments for the real data sets, we still use the empirical probabilities defined in Section 2.6.5.1 as performance measures because the cluster enumeration results vary depending on the initialization of the EM and the K-means++ algorithm.

Iris Data Set

The Fisher's Iris data set [Fisher, 1936] is a 4-dimensional data set collected from three species of the Iris flower. It contains three clusters of 50 instances each, where each cluster corresponds to one species of the Iris flower [Lichman, 2013]. One cluster is linearly separable from the other two, while the remaining ones overlap. We have normalized the data set by dividing the features by their corresponding mean.

Figure 2.4a shows the empirical probability of selection of different cluster enumeration criteria as a function of the number of clusters specified by the candidate models in $\mathcal{M}$. Our criterion, BIC_{N}, is able to estimate the correct number of clusters ($K = 3$) 98.8% of the time, while BIC_{O} always underestimates the number of clusters to $\hat{K}_{\text{BIC}_{\text{O}}} = 2$. BIC_{OS} completely breaks down and, in most cases, goes for the specified maximum number of clusters. Even though two out of three clusters are not linearly separable, our criterion is able to estimate the correct number of clusters with a very high accuracy. Figure 2.5 shows the behavior of the BIC curves of BIC_{N}, given by (2.19), and the original BIC implemented as a wrapper around the EM

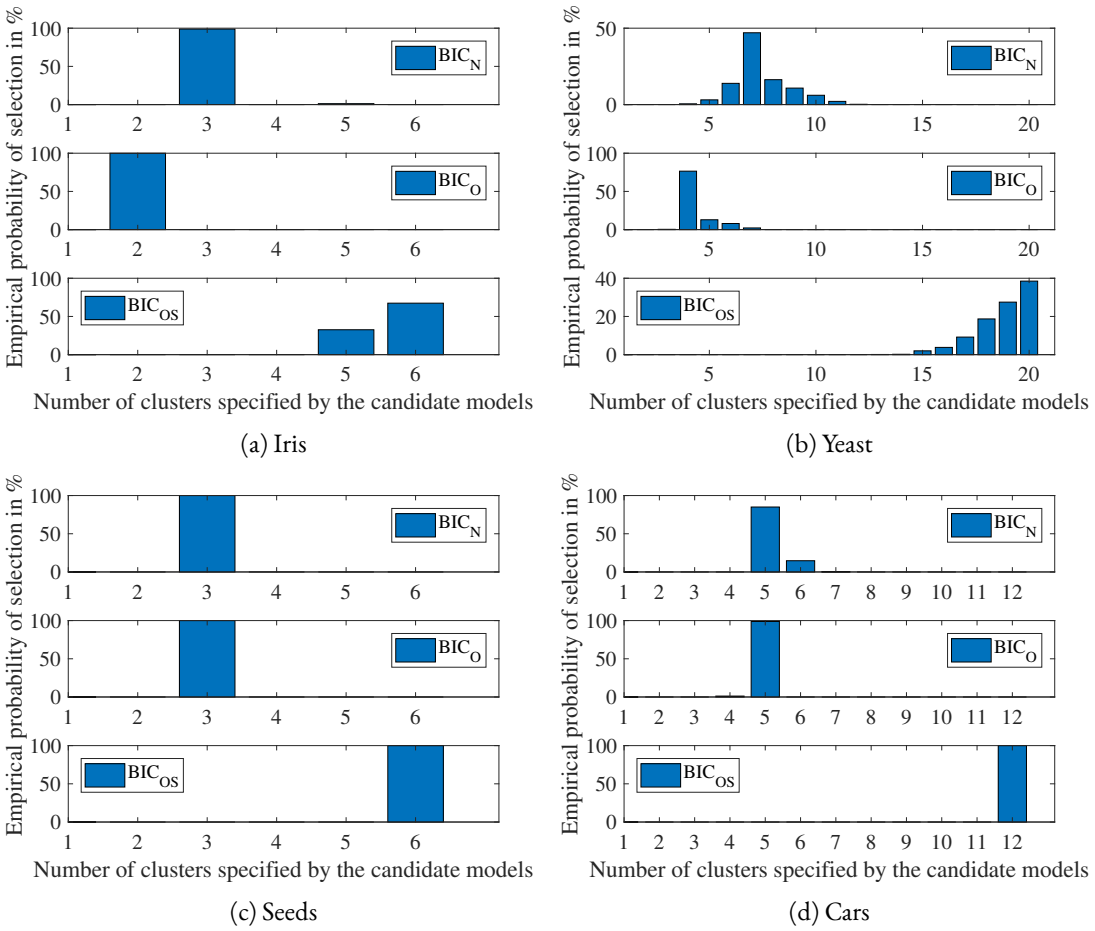
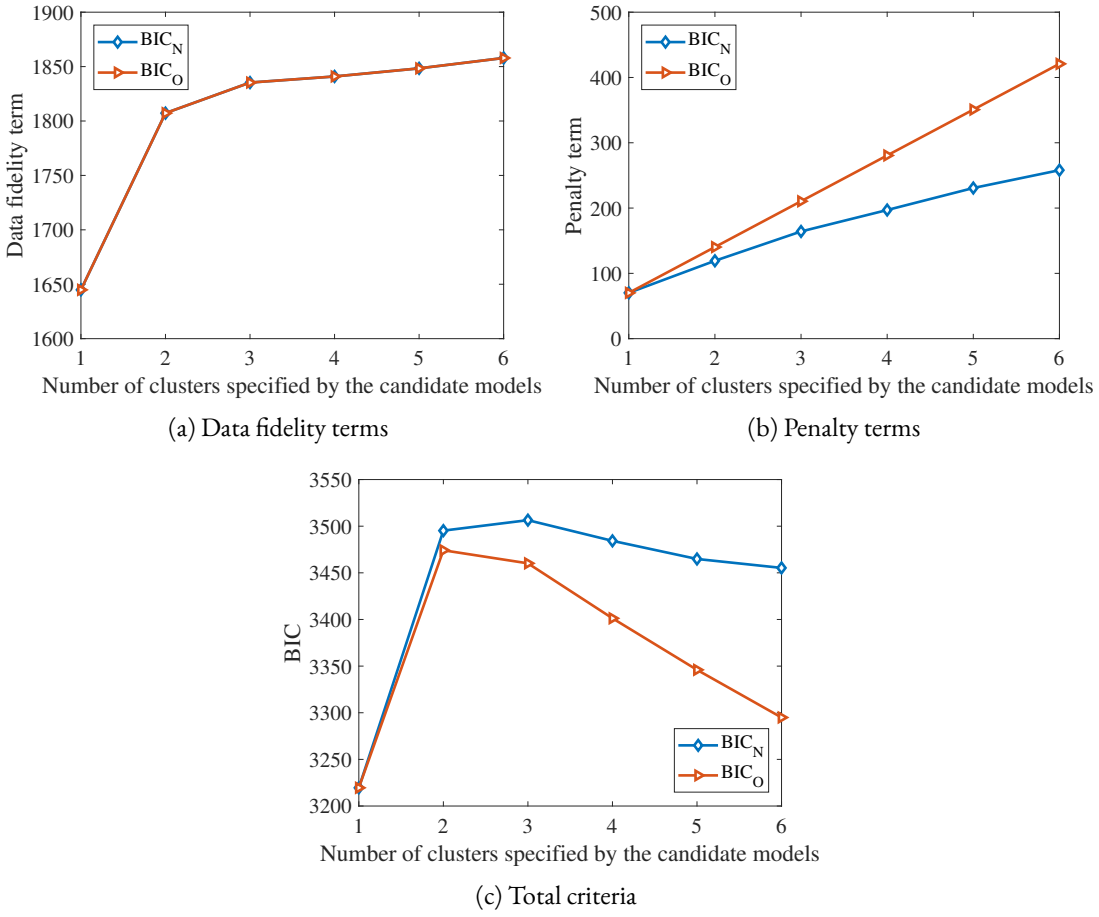


Figure 2.4: Empirical probability of selection of our criterion, $\mathrm{BIC}_{\mathrm{N}}$, and existing Bayesian cluster enumeration criteria for the real data sets.

algorithm, $\mathrm{BIC}_{\mathrm{O}}$ given by (2.27), for one Monte Carlo experiment. From (2.31), we know that the data fidelity terms of both criteria are the same and this can be seen in Figure 2.5a. But, their penalty terms are quite different, see Figure 2.5b. Due to the difference in the penalty terms of $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{O}}$, we observe a different BIC curve in Figure 2.5c. The total criterion (BIC) curve of $\mathrm{BIC}_{\mathrm{N}}$ has a maximum at the true number of clusters ($K = 3$), while $\mathrm{BIC}_{\mathrm{O}}$ has a maximum at $\hat{K}_{\mathrm{BIC}_{\mathrm{O}}} = 2$. Observe that, again, the penalty term of our criterion, $\mathrm{BIC}_{\mathrm{N}}$, has a curvature at the true number of clusters $K = 3$. Just as in the simulated data experiment, the penalty term of $\mathrm{BIC}_{\mathrm{N}}$ gives valuable information about the true number of clusters in the data set while the penalty terms of the other cluster enumeration criteria are uninformative on their own.



(a) Data fidelity terms

(b) Penalty terms

(c) Total criteria

Figure 2.5: The data fidelity terms, penalty terms, and the total criteria of $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{O}}$ for the Iris data set.

Yeast Data Set

The Yeast data set is an 8-dimensional data set with 1484 instances [Lichman, 2013]. It contains ten clusters with the following distribution of data vectors in each cluster: $N_1 = 463$, $N_2 = 429$, $N_3 = 244$, $N_4 = 163$, $N_5 = 51$, $N_6 = 44$, $N_7 = 37$, $N_8 = 30$, $N_9 = 20$, and $N_{10} = 5$. Some clusters have very few data vectors compared to others. Hence, this data set is very challenging for any cluster enumeration method.

Figure 2.4b depicts the empirical probability of selection as a function of the number of clusters specified by the candidate models in $\mathcal{M}$. $\mathrm{BIC}_{\mathrm{os}}$ overestimates the number of clusters 100% of the time and $\mathrm{BIC}_{\mathrm{o}}$ underestimates the number of clusters 100% of the time. Our criterion, $\mathrm{BIC}_{\mathrm{N}}$, estimates the correct number of clusters 6.1% of the time. Comparing the

estimated number of clusters by BIC_{O} and BIC_{N} one notices a very interesting result. BIC_{O} estimates four clusters majority of the time, while BIC_{N} finds seven clusters. Hence, for this real data example, BIC_{N} is able to provide better cluster resolution than BIC_{O}.

Seeds Data Set

The Seeds data set is a 7-dimensional data set which contains measurements of geometric properties of kernels belonging to three different varieties of wheat, where each variety is represented by 70 instances [Lichman, 2013].

As shown in Figure 2.4c, BIC_{N} and BIC_{O} are able to estimate the correct number of clusters 100% of the time, while BIC_{OS} overestimates the number of clusters to $\hat{K}_{\text{BIC}_{\text{OS}}} = 6$. In cases where either the maximum found from the BIC curve is very near to the maximum number of clusters specified by the candidate models or no clear maximum can be found, different post-processing steps that attempt to find a significant curvature in the BIC curve have been proposed in the literature. One such method is the knee point detection strategy [Zhao et al., 2008a; Zhao et al., 2008b]. For the Seeds data set, applying the knee point detection method to the BIC curve generated by BIC_{OS} allows for the correct estimation of the number of clusters 100% of the time.

Multi-Object Multi-Camera Network Application

The multi-object multi-camera network application [Teklehaymanot et al., 2016; Binder et al., 2018] depicted in Figure 2.6 contains seven cameras that actively monitor a common scene of interest from different viewpoints. There are six cars that enter and leave the scene of interest at different time frames. The video captured by each camera in the network is 18 seconds long and 550 frames are captured by each camera. Our objective is to estimate the total number of cars observed by the camera network. This multi-object multi-camera network example is a challenging scenario for cluster enumeration in the sense that each camera monitors the scene from different angles, which can result in differences in the extracted feature vectors (descriptors) of the same object. Furthermore, as shown in Figure 2.6, the video that is captured by the cameras has a low resolution.

We consider a centralized network structure where the spatially distributed cameras send feature vectors to a fusion center for further processing. Hence, each camera $c_i \in \mathcal{C} \triangleq$

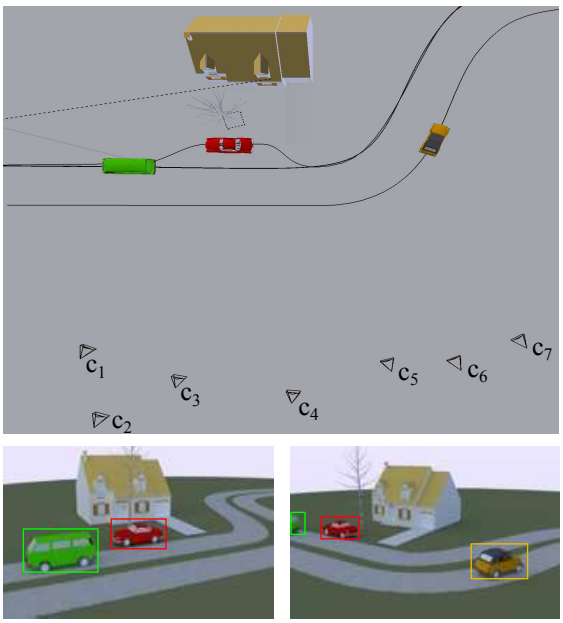


Figure 2.6: A wireless camera network continuously observing a common scene of interest. The top image depicts a camera network with seven spatially distributed cameras that actively monitor the scene from different viewpoints. The bottom left and right images show a frame captured by cameras 1 and 7, respectively, at the same time instant.

$\{c_1, \ldots, c_7\}$ first extracts the objects of interest, cars in this case, from the frames in the video using a Gaussian mixture model-based foreground detector. Then, speeded up robust features (SURF) [Bay et al., 2008] and color features are extracted from the cars. A standard MATLAB implementation of SURF is used to generate a 64-dimensional feature vector for each detected object. Additionally, a 10 bin histogram for each of the RGB color channels is extracted, resulting in a 30-dimensional color feature vector. In our simulations, we apply principal component analysis (PCA) to reduce the dimension of the color features to 15. Each camera $c_i \in \mathcal{C}$ stores its feature vectors in $\mathcal{X}_{c_i}$. Finally, the feature vectors extracted by each camera, $\mathcal{X}_{c_i}$, are sent to the fusion center. At the fusion center, we have the total set of feature vectors $\mathcal{X} \triangleq \{\mathcal{X}_{c_1}, \ldots, \mathcal{X}_{c_7}\} \subset \mathbb{R}^{79\times 5213}$ based on which cluster enumeration is performed.

The empirical probability of selection for different Bayesian cluster enumeration criteria as a function of the number of clusters specified by the candidate models in $\mathcal{M}$ is displayed in Figure 2.4d. Even though there are six cars in the scene of interest, two cars have similar colors. Our criterion, BIC_N, finds six clusters only 14.7% of the time, while the other cluster enumeration criteria are unable to find the correct number of clusters (cars). BIC_N finds five

Table 2.5: Comparison of cluster enumeration performance of different Bayesian criteria for the real data sets. The performance metrics are the empirical probability of detection in %, the empirical probability of underestimation in %, and the mean absolute error (MAE).

		Iris	Yeast	Seeds	Cars
$p_{\text{det}}(\%)$	BIC_{N}	**98.8**	**6.1**	**100**	**14.7**
	BIC_{O}	0	0	**100**	0
	BIC_{OS}	0	0	0	0
$p_{\text{under}}(\%)$	BIC_{N}	0	91.6	0	85
	BIC_{O}	100	100	0	100
	BIC_{OS}	0	0	0	0
MAE	BIC_{N}	**0.024**	**2.61**	**0**	**0.853**
	BIC_{O}	1	5.649	**0**	1.012
	BIC_{OS}	2.674	8.804	3	6

clusters majority of the time. This is very reasonable due to the color similarity of the two cars, which results in the merging of their clusters. The original BIC, BIC_{O}, also finds five clusters majority of the time. But, it also tends to underestimate the number of clusters even more by detecting only four clusters. Hence, our cluster enumeration criterion outperforms existing BIC-based methods in terms of MAE as shown in Table 2.5, which summarizes the performance of different Bayesian cluster enumeration criteria on the real data sets.

2.7 Bayesian Cluster Enumeration Criterion with Finite Sample Penalty Term

Like many model selection criteria in the literature, BIC_{N} is derived under asymptotic assumptions on the size of the observed data. However, in the finite sample regime, the asymptotic assumptions made by BIC_{N} are violated, which results in a weak penalty term. Having a weak penalty term translates into having a BIC curve which has an increasing trend throughout the considered range of number of clusters. In such cases, the criterion becomes prone to overestimation. To alleviate this problem, in this section, we extend the derivation of BIC_{N} by providing an exact expression for its penalty term [Teklehaymanot et al., 2018d].

In the case where the data set $\mathcal{X}$ is composed of data vectors which are realizations of multivariate Gaussian distributed random variables, given that assumptions (A-2.1)-(A-2.6) are

satisfied, the posterior probability of the candidate model $M_l \in \mathcal{M}$ given $\mathcal{X}$ can be written as

$$\log p(M_l|\mathcal{X}) \approx \sum_{m=1}^{l} \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) - \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{J}}_m| - \log f(\mathcal{X}), \tag{2.40}$$

where $\log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)$ is given by (A.1). Ignoring the model independent terms, (2.40) can be simplified to

$$\log p(M_l|\mathcal{X}) \approx \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{J}}_m|. \tag{2.41}$$

In (2.41), the first two terms in the right hand side of the approximation are the data fidelity terms and the last term is the penalty term. In this section, our objective is to provide an exact expression to the penalty term.

The determinant of the FIM, $|\hat{\boldsymbol{J}}_m|$, for $m = 1, \ldots, l$, is given by (B.15). Substituting (B.15) into (2.41) results in

$$\begin{aligned}
\log p(M_l|\mathcal{X}) \approx & \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log|\hat{\boldsymbol{\Sigma}}_m| \\
& - \frac{1}{2}\sum_{m=1}^{l} \log\left(\left|N_m \hat{\boldsymbol{\Sigma}}_m^{-1}\right| \times \left|-\frac{N_m}{2}\boldsymbol{D}^\top \hat{\boldsymbol{F}}_m \boldsymbol{D}\right|\right) \\
= & \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\sum_{m=1}^{l} \log\left|N_m \hat{\boldsymbol{\Sigma}}_m^{-1}\right| \\
& - \frac{1}{2}\sum_{m=1}^{l} \log\left|-\frac{N_m}{2}\boldsymbol{D}^\top \hat{\boldsymbol{F}}_m \boldsymbol{D}\right|,
\end{aligned} \tag{2.42}$$

where $\boldsymbol{D} \in \mathbb{R}^{r^2 \times \frac{1}{2}r(r+1)}$ is the duplication matrix and

$$\hat{\boldsymbol{F}}_m \triangleq \hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \left(\hat{\boldsymbol{\Sigma}}_m^{-1} - \frac{2}{N_m}\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\Delta}}_m\hat{\boldsymbol{\Sigma}}_m^{-1}\right) \tag{2.43}$$

$$\hat{\boldsymbol{\Delta}}_m \triangleq \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)^\top. \tag{2.44}$$

From (A.5), we know that

$$\hat{\boldsymbol{\Delta}}_m = N_m \hat{\boldsymbol{\Sigma}}_m. \tag{2.45}$$

Substituting (2.45) into (2.43) results in

$$\begin{aligned}\hat{\boldsymbol{F}}_m &= \hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \left(\hat{\boldsymbol{\Sigma}}_m^{-1} - 2\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\Sigma}}_m\hat{\boldsymbol{\Sigma}}_m^{-1}\right) \\ &= -\hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \hat{\boldsymbol{\Sigma}}_m^{-1}.\end{aligned} \tag{2.46}$$

Finally, substituting (2.46) into (2.42) results in

$$\begin{aligned}\mathrm{BIC}_{\mathrm{NF}}(M_l) &\triangleq \log p(M_l|\mathcal{X}) \\ &\approx \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\sum_{m=1}^{l} \log\left|N_m \hat{\boldsymbol{\Sigma}}_m^{-1}\right| \\ &- \frac{1}{2}\sum_{m=1}^{l} \log\left|\frac{N_m}{2}\boldsymbol{D}^\top\left(\hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \hat{\boldsymbol{\Sigma}}_m^{-1}\right)\boldsymbol{D}\right| \\ &= \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{4}r(r+3)\sum_{m=1}^{l} \log N_m \\ &+ \frac{1}{4}r(r+1)l\log 2 + \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\sum_{m=1}^{l} \log\left|\boldsymbol{D}^\top\left(\hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \hat{\boldsymbol{\Sigma}}_m^{-1}\right)\boldsymbol{D}\right|.\end{aligned} \tag{2.47}$$

The duplication matrix $\boldsymbol{D}$ is calculated as [Magnus & Neudecker, 1980]

$$\boldsymbol{D}^\top = \sum_{i \geq j} \boldsymbol{v}_{ij} \mathrm{vec}\left(\boldsymbol{Y}_{ij}\right)^\top, \tag{2.48}$$

where $1 \leq j \leq i \leq r$ and $\boldsymbol{v}_{ij} \in \mathbb{R}^{\frac{1}{2}r(r+1)\times 1}$ is a unit vector with one at its $((j-1)r + i - \frac{1}{2}j(j-1))$th entry and zero elsewhere. $\boldsymbol{Y}_{ij} \in \mathbb{R}^{r\times r}$ is given by

$$\boldsymbol{Y}_{ij} = \begin{cases} \boldsymbol{U}_{ii}, & i = j \\ \boldsymbol{U}_{ij} + \boldsymbol{U}_{ji}, & i \neq j \end{cases}, \tag{2.49}$$

where $\boldsymbol{U}_{ij}$ contains one at its i, jth entry and zero elsewhere.

Comparing (2.19) and (2.47) one notice that

$$\mathrm{BIC}_{\mathrm{NF}}(M_l) = \mathrm{BIC}_{\mathrm{N}}(M_l) + \frac{1}{4}r(r+1)l\log 2 + \frac{1}{2}\sum_{m=1}^{l}\log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\sum_{m=1}^{l}\log\left|\boldsymbol{D}^\top\left(\hat{\boldsymbol{\Sigma}}_m^{-1}\otimes\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\boldsymbol{D}\right|. \tag{2.50}$$

Unlike $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{O}}$, the penalty term of $\mathrm{BIC}_{\mathrm{NF}}$ depends on the covariance matrix of the individual clusters in $M_l \in \mathcal{M}$. This allows $\mathrm{BIC}_{\mathrm{NF}}$ to lower the penalty term when the determinant of the covariance matrices are high and penalize more severely when they are low. However, if the observations span a large range of values, then the covariance matrices of individual clusters are very large and their inverses become close to zero. As a result, the penalty term of $\mathrm{BIC}_{\mathrm{NF}}$ might go to infinity. Hence, in such cases, we recommend normalizing the data prior to the estimation of cluster parameters.

Once $\mathrm{BIC}_{\mathrm{NF}}(M_l)$ is computed for each candidate model $M_l \in \mathcal{M}$, the number of partitions (clusters) in $\mathcal{X}$ is estimated as

$$\hat{K}_{\mathrm{BIC}_{\mathrm{NF}}} = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \mathrm{BIC}_{\mathrm{NF}}(M_l). \tag{2.51}$$

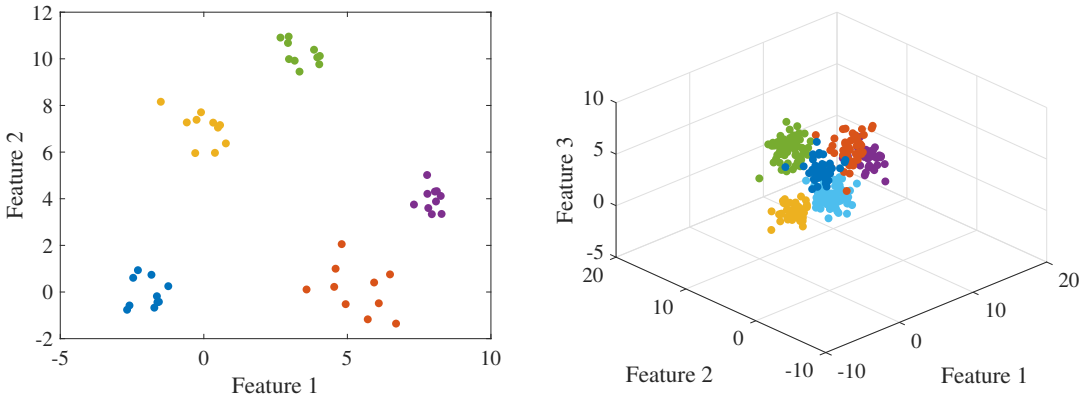
Similar to $\mathrm{BIC}_{\mathrm{N}}$, the cluster parameters required by $\mathrm{BIC}_{\mathrm{NF}}$ are estimated using the EM algorithm. The additional complexity of $\mathrm{BIC}_{\mathrm{NF}}(M_l)$ compared to $\mathrm{BIC}_{\mathrm{N}}(M_l)$ comes from the term

$$\frac{1}{2}\sum_{m=1}^{l}\log\left|\boldsymbol{D}^\top\left(\hat{\boldsymbol{\Sigma}}_m^{-1}\otimes\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\boldsymbol{D}\right|.$$

The duplication matrix $\boldsymbol{D}$ is computed only once, and thus it can be ignored in the complexity analysis. Hence, the excess computational cost is $\mathcal{O}(lr^6)$.

2.7.1 Experimental Results

In all simulations, we set $L_{\min} = 1$ and $L_{\max} = 2K$, where K is the true number of clusters in $\mathcal{X}$. All simulation results are an average of 1000 Monte Carlo experiments. We compare the proposed criterion, $\mathrm{BIC}_{\mathrm{NF}}$, with $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{O}}$ using two synthetic data sets. The compared criteria use the same implementation of the EM algorithm.



(a) Data-2.3 for $N_k = 10$ (b) Data-2.4 for $N_k = 50$

Figure 2.7: Synthetic data sets

The first experiment is based on Data-2.3, depicted in Figure 2.7a, which contains realizations of the random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, for $k = 1, \ldots, 5$, with cluster centroids $\boldsymbol{\mu}_1 = [-2, 0]^\top$, $\boldsymbol{\mu}_2 = [5, 0]^\top$, $\boldsymbol{\mu}_3 = [0, 7]^\top$, $\boldsymbol{\mu}_4 = [8, 4]^\top$, $\boldsymbol{\mu}_5 = [3, 10]^\top$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}, \boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, \boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

Comparison of the three Bayesian cluster enumeration criteria as a function of the number of data vectors per cluster, N_k, for Data-2.3 is given in Table 2.6. The proposed criterion, $\mathrm{BIC}_{\mathrm{NF}}$, outperforms the other criteria when the number of data vectors per cluster is small and it exhibits a very small mean absolute error (MAE). The empirical probability of overestimation, p_{over}, of $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{O}}$ is very high especially when the number of data vectors per cluster is small. As expected the cluster number estimates of all compared criteria converge to the correct number of clusters, $K = 5$, when the number of data vectors per cluster increases. A comparison of the total criteria and penalty terms of the different cluster enumeration criteria for a single Monte Carlo experiment is shown in Figure 2.8a and Figure 2.8b, respectively. $\mathrm{BIC}_{\mathrm{NF}}$ penalizes complex models more severely than the other criteria.

In the second experiment, we consider Data-2.4, shown in Figure 2.7b, which contains realizations of the random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, for $k = 1, \ldots, 6$, with cluster centroids $\boldsymbol{\mu}_1 = [-1, 0, 7]^\top$, $\boldsymbol{\mu}_2 = [3, 0, 8]^\top$, $\boldsymbol{\mu}_3 = [0, 5, 1]^\top$, $\boldsymbol{\mu}_4 = [9, 4, 4]^\top$, $\boldsymbol{\mu}_5 = [3, 9, 5]^\top$,

Table 2.6: The empirical probability of detection in %, the empirical probability of overestimation in %, and the mean absolute error (MAE) of various Bayesian cluster enumeration criteria as a function of the number of data vectors per cluster (N_k) for Data-2.3.

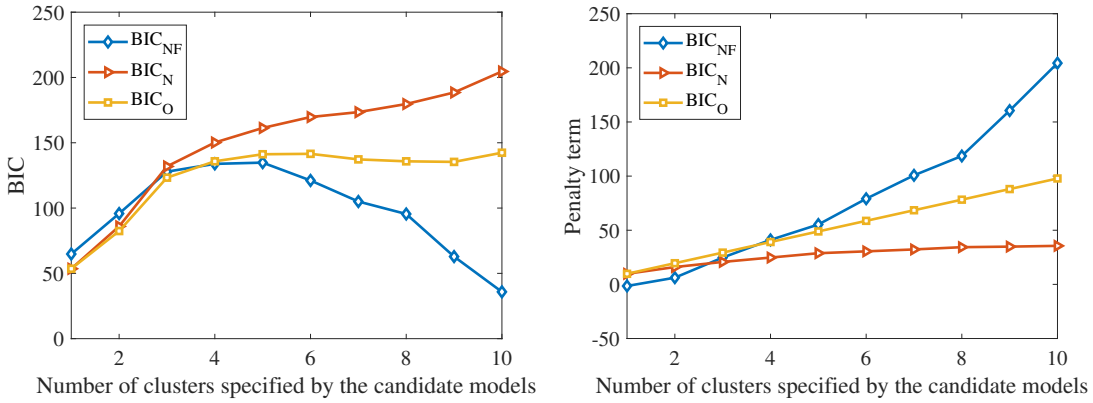
N_k		10	50	100	1000
$p_{\text{det}}(\%)$	BIC_{NF}	**77.6**	**100**	**100**	**100**
	BIC_{N}	0	77.8	96.2	**100**
	BIC_{O}	26.4	99.3	99.7	**100**
$p_{\text{over}}(\%)$	BIC_{NF}	0	0	0	0
	BIC_{N}	100	22.2	3.8	0
	BIC_{O}	73.1	0.7	0.3	0
MAE	BIC_{NF}	**0.228**	**0**	**0**	**0**
	BIC_{N}	4.768	0.483	0.043	**0**
	BIC_{O}	2.461	0.007	0.003	**0**

Table 2.7: The empirical probability of detection in %, the empirical probability of overestimation in %, and the mean absolute error (MAE) of various Bayesian cluster enumeration criteria as a function of the number of data vectors per cluster (N_k) for Data-2.4.

N_k		50	100	250	1000
$p_{\text{det}}(\%)$	BIC_{NF}	**82.1**	**96.7**	**98.7**	**99.3**
	BIC_{N}	64.7	92.9	98.1	**99.3**
	BIC_{O}	51.7	91.1	**98.7**	**99.3**
$p_{\text{over}}(\%)$	BIC_{NF}	0.6	0.6	0.6	0.2
	BIC_{N}	30.9	5.7	1.3	0.2
	BIC_{O}	0	0.2	0.2	0
MAE	BIC_{NF}	**0.19**	**0.033**	**0.013**	**0.007**
	BIC_{N}	0.851	0.079	0.019	**0.007**
	BIC_{O}	0.602	0.089	**0.013**	**0.007**

$\boldsymbol{\mu}_6 = [5, 5, 1.5]^\top$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.6 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.6 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.8 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.3 \end{bmatrix},$$



(a) Total criteria (b) Penalty terms

Figure 2.8: The total criteria and the penalty terms of different Bayesian cluster enumeration criteria for Data-2.3 when $N_k = 10$.

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}, \boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}, \boldsymbol{\Sigma}_6 = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}.$$

We compare the cluster enumeration performance of different criteria for Data-2.4 by setting the number of data vectors per cluster to one of the values in $\{50, 100, 250, 1000\}$. As shown in Table 2.7, BIC_{NF} outperforms the other criteria when N_k is small and it exhibits a small MAE. For small values of N_k, BIC_{N} performs better than BIC_{O}, while BIC_{O} tends to underestimate the number of clusters. Similar to the results of Data-2.3, asymptotically, all cluster enumeration criteria behave satisfactorily.

2.8 Application: Distributed and Adaptive Bayesian Cluster Enumeration

Distributed signal processing and communication networking are advancing rapidly. This has led to new paradigms for signal and parameter estimation. one such paradigm is the so-called multiple devices multiple tasks (MDMT) paradigm, where distributed heterogeneous devices solve different signal processing tasks by cooperating in an ad hoc sensor network without a fusion center [Plata-Chaves et al., 2017; Bogdanovic et al., 2014; Bertrand & Moonen,

2012; Chen et al., 2015; Plata-Chaves et al., 2015; Chouvardas et al., 2015]. A crucial first step towards the successful cooperation of nodes is to answer the question: *Who observes what?* For example, distributed node-specific image/video enhancement requires the common labeling of all objects within a camera network [Teklehaymanot et al., 2015; Teklehaymanot et al., 2017], and distributed node-specific speech enhancement requires the common labeling of all speakers [Chouvardas et al., 2015; Bahari et al., 2016]. Several distributed algorithms have been proposed in the literature, which frame the labeling task in form of a data clustering problem after extracting source-specific features [Teklehaymanot et al., 2015; Teklehaymanot et al., 2017; Chouvardas et al., 2015; Bahari et al., 2016; Binder et al., 2015; Binder et al., 2016]. However, a major drawback of these state-of-the-art methods is that they assume the number of sources/objects, which translates into the number of clusters, to be known a priori. The assumption is rather restrictive, since this information is mostly unavailable in real-world source/object labeling applications. In addition, the number of sources/objects could be time-varying, which calls for adaptive methods. In the literature, only few methods exist that tackle the problem of estimating data clusters from observation collected by distributed heterogeneous devices with node-specific interests [Teklehaymanot et al., 2016; Binder et al., 2018]. The work in [Teklehaymanot et al., 2016] presents diffusion-based X-means and PG-means algorithms to estimate the number of clusters sequentially from streaming-in data collected by a distributed sensor network.

In this section, we present two distributed and adaptive Bayesian cluster enumeration algorithms. The performance of the presented methods is tested using numerical experiments and real data multi-object multi-camera network application. Comparison to the method presented in [Teklehaymanot et al., 2016] is also provided.

2.8.1 Problem Formulation

Consider a wireless sensor network with J nodes whose topology is described by a graph. The neighborhood of node $j \in \mathcal{J} \triangleq \{1, \ldots, J\}$, denoted as $\mathcal{B}_j$, is the set of all nodes, including j, that node j can exchange information with. At time instant t, where $t = 1, 2, \ldots$, each node $j \in \mathcal{J}$ collects r-dimensional data vectors and stores them in $\mathcal{X}_{jt} \triangleq \{\boldsymbol{x}_{j1}, \ldots, \boldsymbol{x}_{jN_t}\} \in \mathbb{R}^{r \times N_t}$, where N_t is the number of data vectors observed by node $j \in \mathcal{J}$ at time instant t. As time progresses, each node $j \in \mathcal{J}$ stores its data in $\mathcal{S}_{jt} \triangleq \{\mathcal{X}_{j1}, \ldots, \mathcal{X}_{jt}\} \in \mathbb{R}^{r \times N_{jt}}$, where

$N_{jt} = \sum_{i=1}^{t} N_i$. $\mathcal{S}_{jt}$ contains K_t independent, mutually exclusive, and non-empty clusters. Assume that a set of candidate models $\mathcal{M}_j \triangleq \{M_{jL_{\min}}, \ldots, M_{jL_{\max}}\}$, is given, where $L_{\min}$ and $L_{\max}$ are the specified minimum and maximum number of clusters, respectively. Each candidate model $M_{jl} \in \mathcal{M}_j$ represents a partitioning of $\mathcal{S}_{jt}$ into $l \in \{L_{\min}, \ldots, L_{\max}\}$ clusters, where $l \in \mathbb{Z}^+$. Each data vector $\boldsymbol{x}_{jn} \in \mathcal{S}_{jt}, n = 1, \ldots, N_{jt}$, has an associated class label $k \in \mathcal{K} \triangleq \{1, \ldots, K_t\}$. Our goal is to enable each node $j \in \mathcal{J}$ to adaptively estimate the number of clusters in the data set $\mathcal{S}_{jt}$ by cooperating with its neighbors in $\mathcal{B}_j$.

2.8.2 Distributed and Adaptive Bayesian Cluster Enumeration Algorithms

We present two distributed and adaptive Bayesian cluster enumeration algorithms, which estimate the time-varying number of clusters K_{jt} from streaming-in data $\mathcal{S}_{jt}$ [Teklehaymanot et al., 2018b]. Based on the Gaussian data assumption, the cluster parameters are estimated via the expectation maximization (EM) algorithm using the number of clusters specified by each candidate model $M_{jl} \in \mathcal{M}_j$, where $l \in \{L_{\min}, \ldots, L_{\max}\}$. Given the parameter estimates for all candidate models, the Bayesian cluster enumeration criteria derived in Section 2.6.1 and Section 2.7 determine the number of clusters in the data set $\mathcal{S}_{jt}$ at time instant t in a distributed and cooperative manner by incorporating the diffusion principle [Sayed et al., 2013]. The working principle of the distributed and adaptive Bayesian cluster enumeration algorithms, depicted in Figure 2.9, is explained as follows.

1. *Collect data:* each node $j \in \mathcal{J}$ collects N_t data vectors at time instant t and stores them in $\mathcal{X}_{jt}$. The accumulated data vectors at node j at time instant t are stored in $\mathcal{S}_{jt} \triangleq \{\mathcal{X}_{j1}, \ldots, \mathcal{X}_{jt}\}$. Optionally, each node $j \in \mathcal{J}$ exchanges $\mathcal{X}_{jt}$ within its neighborhood $\mathcal{B}_j$. This exchange step significantly increases the communication and computation costs, but may also provide a performance gain; see Section 2.8.3.

2. *Estimate parameters:* each node $j \in \mathcal{J}$ estimates cluster parameters for each candidate model $M_{jl} \in \mathcal{M}_j$ using the EM algorithm. The estimated parameters are the cluster centroids $\hat{\boldsymbol{\mu}}_{jml}^0 \in \mathbb{R}^{r \times 1}$, covariance matrices $\hat{\boldsymbol{\Sigma}}_{jml}^0 \in \mathbb{R}^{r \times r}$, and the number of data vectors per cluster $N_{jml} \in \mathbb{Z}^+$ for $m = 1, \ldots, l$ and $l = L_{\min}, \ldots, L_{\max}$, where

$N_{jt} = \sum_{m=1}^{l} N_{jml}$.*

3. *Exchange parameter estimates:* each node $j \in \mathcal{J}$ exchanges $\hat{\boldsymbol{\mu}}_{jml}^0$ and $\hat{\boldsymbol{\Sigma}}_{jml}^0$ for $m = 1, \ldots, l$ and $l = L_{\min}, \ldots, L_{\max}$ within its neighborhood $\mathcal{B}_j$.

4. *Synchronize parameter estimates:* nodes assign labels to each cluster in an arbitrary manner. Hence, each node $j \in \mathcal{J}$ should synchronize the label assigned to the parameter estimates it received from its neighbors with respect to the label assigned to its own parameter estimates. For each cluster $m = 1, \ldots, l$ and $l = L_{\min}, \ldots, L_{\max}$, each node $j \in \mathcal{J}$ uses the Euclidean norm as a metric to measure the distance between its own cluster centroid estimates $\hat{\boldsymbol{\mu}}_{jml}^0$ and the estimates of its neighbors $\hat{\boldsymbol{\mu}}_{bml}^0$, where $b \in \mathcal{B}_j/\{j\}$, as follows:

$$\delta_{jbm} = \left|\left|\hat{\boldsymbol{\mu}}_{jml}^0 - \hat{\boldsymbol{\mu}}_{bml}^0\right|\right|_2 \tag{2.52}$$

This creates a matrix $\boldsymbol{\Delta}_{jbl}$ whose row and column vectors are δ_{jbm} for $m = 1, \ldots, l$. Now, the synchronization problem coincides with an assignment problem that can be efficiently solved using the Hungarian algorithm [Munkres, 1957]. Using the assignment results of the Hungarian algorithm, each node $j \in \mathcal{J}$ corrects the labels of the parameter estimates received from its neighborhood $\mathcal{B}_j$.

5. *Adapt parameter estimates:* the own and received cluster covariance matrix estimates are adapted via

$$\hat{\boldsymbol{\Sigma}}_{jml} = \alpha \hat{\boldsymbol{\Sigma}}_{jml}^0 + (1-\alpha) \sum_{b \in \mathcal{B}_j/\{j\}} a_{bml} \hat{\boldsymbol{\Sigma}}_{bml}^0 \tag{2.53}$$

at each node $j \in \mathcal{J}$ for each candidate model $M_{jl} \in \mathcal{M}_j$. α denotes the tradeoff between the weight given to the own and neighboring node estimates. Here, we use inverse distance norm combination weights [Sayed, 2014b], which are defined as

$$a_{bml} = \frac{1}{||\hat{\boldsymbol{\Sigma}}_{jml}^0 - \hat{\boldsymbol{\Sigma}}_{jmb}^0||_2}. \tag{2.54}$$

*We use the EM algorithm to estimate cluster parameters because of its effectiveness in dealing with spherically as well as elliptically distributed data clusters. However, in principle, the proposed framework also allows for using other parameter estimators.

The combination weights are further normalized such that $\sum_{b\in\mathcal{B}_j/\{j\}} a_{bml} = 1$. This way erroneous nodes are given less weight compared to the good nodes in the neighborhood.

6. *Perform model order selection:* using the adapted parameter estimates, each node $j \in \mathcal{J}$ selects the model $M_{j\hat{K}_{jt}^0} \in \mathcal{M}_j$, with $\hat{K}_{jt}^0 \in \{L_{\min}, \ldots, L_{\max}\}$, that maximizes the posterior probability given $\mathcal{S}_{jt}$. For this purpose, the BIC is calculated using either

$$\text{D-BIC}_{\text{N}}(M_{jl}) = \sum_{m=1}^{l} N_{jml} \log N_{jml} - \sum_{m=1}^{l} \frac{N_{jml}}{2} \log|\hat{\boldsymbol{\Sigma}}_{jml}| - \frac{q}{2}\sum_{m=1}^{l} \log N_{jml}, \text{ or} \tag{2.55}$$

$$\begin{aligned}\text{D-BIC}_{\text{NF}}(M_{jl}) = \text{D-BIC}_{\text{N}}(M_{jl}) &+ \frac{1}{4}r(r+1)l\log 2 + \frac{1}{2}\sum_{m=1}^{l}\log|\hat{\boldsymbol{\Sigma}}_{jml}| \\ &- \frac{1}{2}\sum_{m=1}^{l}\log\left|\boldsymbol{D}^\top\left(\hat{\boldsymbol{\Sigma}}_{jml}^{-1}\otimes\hat{\boldsymbol{\Sigma}}_{jml}^{-1}\right)\boldsymbol{D}\right|,\end{aligned} \tag{2.56}$$

where $\boldsymbol{D}$ denotes the duplication matrix of $\hat{\boldsymbol{\Sigma}}_{jml}$ [Magnus & Neudecker, 1980] and $q = \frac{1}{2}r(r+3)$ represents the number of estimated parameters per cluster. Once each node $j \in \mathcal{J}$ computes either $\text{D-BIC}_{\text{N}}(M_{jl})$ or $\text{D-BIC}_{\text{NF}}(M_{jl})$ for each candidate model $M_{jl} \in \mathcal{M}_j$, the next task is to estimate the number of clusters in $\mathcal{S}_{jt}$ using either

$$\hat{K}_{jt}^0 = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \text{D-BIC}_{\text{N}}(M_{jl}), \quad \text{or} \tag{2.57}$$

$$\hat{K}_{jt}^0 = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \text{D-BIC}_{\text{NF}}(M_{jl}). \tag{2.58}$$

7. *Exchange cluster number estimates:* at this point, each node $j \in \mathcal{J}$ exchanges its preliminary estimate of the number of clusters, $\hat{K}_{jt}^0$, in $\mathcal{S}_{jt}$ at time instant t within its neighborhood $\mathcal{B}_j$.

8. *Adapt cluster number estimates:* finally, each node $j \in \mathcal{J}$ adapts its cluster number estimate using

$$\hat{K}_{jt} = \text{median}\left(\hat{K}_{jt}^0, \hat{K}_{bt}^0\right), \tag{2.59}$$

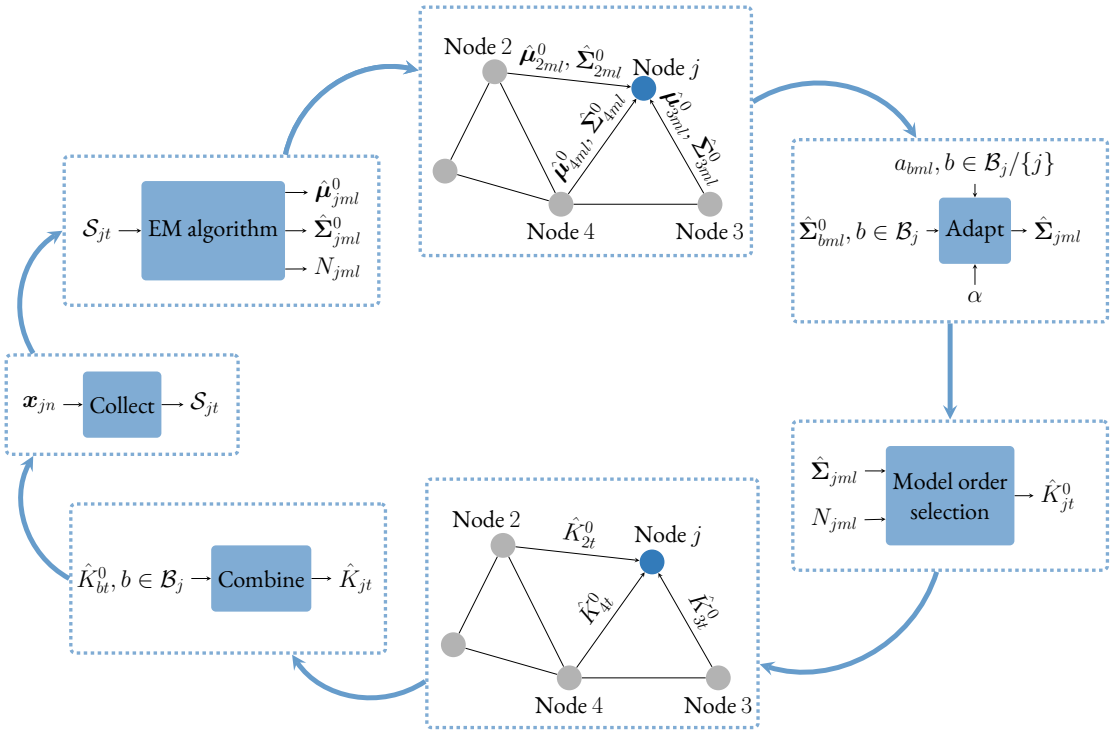


Figure 2.9: Overview of the distributed and adaptive Bayesian cluster enumeration algorithm.

where $\hat{K}^0_{bt}, b \in \mathcal{B}_j/\{j\}$, denotes the cluster number estimates that node j received from its neighbors.

Algorithm 2.3 summarizes the distributed and adaptive Bayesian cluster enumeration algorithms.

2.8.3 Experimental Results

In this section, we evaluate the performance of the distributed and adaptive Bayesian cluster enumeration algorithm using two synthetic and a real data set. A comparison to the DX-means algorithm [Teklehaymanot et al., 2016], is given. All simulation results are an average of 300 Monte Carlo experiments and the minimum and maximum number of clusters in the candidate models is set to $L_{\min} = 1$ and $L_{\max} = 2K$, respectively, where K is the number of clusters in the data set $\mathcal{S}_{jt}$ at the final time instant. Equal weight is given to the own and neighborhood-based estimates by setting $\alpha = 0.5$.

Algorithm 2.3 Distributed and adaptive Bayesian cluster enumeration algorithm

Inputs: $L_{\min}$ and $L_{\max}$
for $t = 1, 2, \ldots$ do
  for $j = 1, \ldots, J$ do
    Collect N_t data vectors
    Store data vectors in $\mathcal{X}_{jt}$
    Update $\mathcal{S}_{jt}$
  end for
  for $j = 1, \ldots, J$ do
    for $l = L_{\min}, \ldots, L_{\max}$ do
      for $m = 1, \ldots, l$ do
        Estimate N_{jml}, $\hat{\boldsymbol{\mu}}^0_{jml}$, and $\hat{\boldsymbol{\Sigma}}^0_{jml}$ using the EM algorithm
        Exchange $\hat{\boldsymbol{\mu}}^0_{jml}$ and $\hat{\boldsymbol{\Sigma}}^0_{jml}$ within $\mathcal{B}_j$
      end for
    end for
  end for
  for $j = 1, \ldots, J$ do
    Synchronize parameter estimates
  end for
  for $j = 1, \ldots, J$ do
    for $l = L_{\min}, \ldots, L_{\max}$ do
      for $m = 1, \ldots, l$ do
        Adapt covariance matrix estimates using (2.53)
      end for
      Calculate BIC either via (2.55) or (2.56)
    end for
    Estimate $\hat{K}^0_{jt}$ using either (2.57) or (2.58)
  end for
  for $j = 1, \ldots, J$ do
    Exchange $\hat{K}^0_{jt}$ within $\mathcal{B}_j$
  end for
  for $j = 1, \ldots, J$ do
    Combine $\hat{K}^0_{jt}$ and $\hat{K}^0_{bt}$, $b \in \mathcal{B}_j / \{j\}$, using (2.59)
  end for
end for

2.8.3.1 Network-Wide Performance Measures

The network-wide empirical probability of detection and mean absolute error, which are defined as

$$p_{\text{det}}^{\text{net}} = \frac{1}{JIT} \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{t=1}^{T} \mathbb{1}_{\{\hat{K}_{jt}^{(i)} = K_t\}} \tag{2.60}$$

$$\text{MAE}^{\text{net}} = \frac{1}{JIT} \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{t=1}^{T} \left| K_t - \hat{K}_{jt}^{(i)} \right|, \tag{2.61}$$

are used as performance measures. I is the total number of Monte Carlo experiments, T is the total number of time instances, $\hat{K}_{jt}^{(i)}$ is the estimated number of clusters by the jth node at time instant t and the ith Monte Carlo experiment, and $\mathbb{1}_{\{\hat{K}_{jt}^{(i)} = K_t\}}$ is the indicator function.

2.8.3.2 Numerical Experiments

We consider a wireless sensor network with $J = 10$ nodes and $\#\mathcal{B}_j = 5$. Two types of cooperative networks, namely coop-1 and coop-2, are distinguished, where coop-1 allows only the exchange of estimates, while coop-2 additionally exchanges data vectors within the neighborhood. Results are also reported for distributed non-cooperative (non-coop) and centralized networks. In a centralized network, the fusion center solves the cluster enumeration task after receiving data vectors from all nodes in the network.

In the first experiment, each node $j \in \mathcal{J}$ contains realizations of Data-2.3, which was defined in Section 2.7.1. Each cluster contains $N_k = 210$ data points and each node $j \in \mathcal{J}$ observes $N_t = 30$ data points at time instant t. The number of clusters is time-varying and cluster unbalance appears, which mimics the real data application described in Section 2.8.3.3. For example, at $t = 8$ one cluster contains 210 data vectors while the other cluster contains only 30 data vectors. Table 2.8 summarizes the cluster enumeration performance of the presented distributed and adaptive algorithms and the DX-means algorithm [Teklehaymanot et al., 2016] for this data set in terms of $p_{\text{det}}^{\text{net}}$ and MAE^{net}. DX-means is consistently outperformed for all modes of cooperation. Perfect results are obtained in all cases with D-BIC$_{\text{NF}}$. Cooperation among neighboring nodes enhances the performance of D-BIC$_{\text{N}}$ compared to the distributed non-cooperative scenario.

Table 2.8: The network-wide empirical probability of detection in % and the network-wide mean absolute error of various distributed and adaptive Bayesian cluster enumeration methods for Data-2.3 in different network setups.

		non-coop	coop-1	coop-2	centralized
$p_{\text{det}}^{\text{net}}(\%)$	D-BIC$_{\text{NF}}$	**100**	**100**	**100**	**100**
	D-BIC$_{\text{N}}$	89.69	97.16	98.7	99.95
	DX-means	83.62	96.10	90.5	94.64
MAE$^{\text{net}}$	D-BIC$_{\text{NF}}$	**0**	**0**	**0**	**0**
	D-BIC$_{\text{N}}$	0.68	0.20	0.0131	0.0004
	DX-means	0.17	0.04	0.124	0.0964

Table 2.9: The network-wide empirical probability of detection in % and the network-wide mean absolute error of various distributed and adaptive Bayesian cluster enumeration methods for Data-2.5 in different network setups.

		non-coop	coop-1	coop-2	centralized
$p_{\text{det}}^{\text{net}}(\%)$	D-BIC$_{\text{NF}}$	**85.25**	84.03	**99.11**	**91.25**
	D-BIC$_{\text{N}}$	60.81	**87.89**	99.08	90.67
	DX-means	0.14	5.67	0	0
MAE$^{\text{net}}$	D-BIC$_{\text{NF}}$	**0.32**	**0.34**	**0.0089**	**0.0894**
	D-BIC$_{\text{N}}$	1.68	0.53	0.0092	0.098
	DX-means	4.80	3.59	5.95	6.64

In the second experiment, we use Data-2.5 [Binder et al., 2016], depicted in Figure 2.10, which contains realizations of $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, for $k = 1, \ldots, 8$ and $j \in \mathcal{J}$, with cluster centroids $\boldsymbol{\mu}_1 = [1, 0, 3]$, $\boldsymbol{\mu}_2 = [1, 4, 3]$, $\boldsymbol{\mu}_3 = [1, 0, 6]$, $\boldsymbol{\mu}_4 = [-1, 3, 3]$, $\boldsymbol{\mu}_5 = [4, 4, 4]$, $\boldsymbol{\mu}_6 = [6, 3, 7]$, $\boldsymbol{\mu}_7 = [4.5, 7, 6]$, $\boldsymbol{\mu}_8 = [2, 4, 7]$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_4 = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 1.6 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}, \boldsymbol{\Sigma}_5 = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \boldsymbol{\Sigma}_6 = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1.5 \end{bmatrix},$$

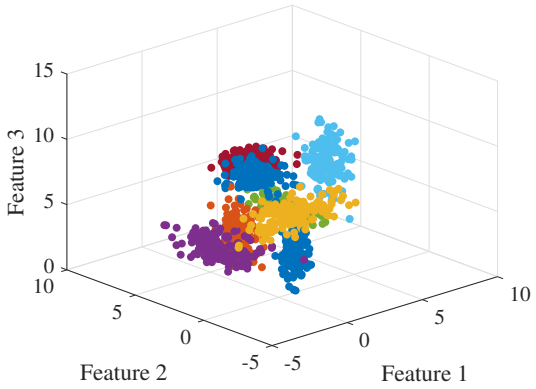


Figure 2.10: Single node realization of Data-2.5

$$\boldsymbol{\Sigma}_7 = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, \boldsymbol{\Sigma}_8 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}.$$

Each cluster k contains $N_k = 150$ data vectors and each node $j \in \mathcal{J}$ observes $N_t = 100$ data vectors, which are randomly drawn from the data set, at time instant t. Table 2.9 summarizes the results for Data-2.5. In all network setups, the proposed distributed and adaptive cluster enumeration algorithms outperform the DX-means algorithm by a large margin. As time progresses, the DX-means algorithm consistently overestimates the number of clusters despite the increase in number of data vectors. Interestingly, coop-2 outperforms the centralized network implementation for our two proposed algorithms. However, this increase in performance comes at an expense of communication and computation cost compared to coop-1.

2.8.3.3 Multi-Object Multi-Camera Network Application

We use an outdoor video sequence that was recorded by $J = 3$ unsynchronized digital video cameras on the campus of École Polytechnique Fédérale de Lausanne in Switzerland [Fleuret et al., 2008; Berclaz et al., 2011] and set the neighborhood size to $\#\mathcal{B}_j = 3$. The cameras were mounted at head level (≈ 1.80m), observing the scene of interest from different angles, and

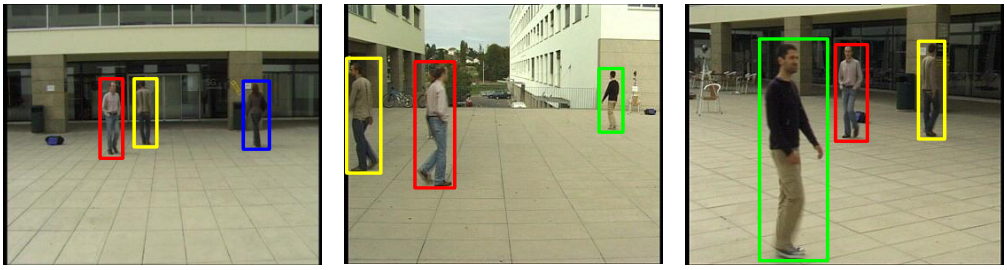


Figure 2.11: Frames captured by cameras 1, 2, and 3, respectively, at the same time instant [Berclaz et al., 2011; Fleuret et al., 2008]. The bounding boxes are associated to the detected pedestrians and the color defines the label, which is identical across all three views.

the captured videos were synchronized by hand. For pedestrian enumeration purposes, we consider the first 550 frames that are captured by the cameras, where up to four people are seen entering and exiting the scene at different time frames. The cameras monitor the scene of interest from different angles which results in different cameras observing different number of pedestrians at the same time instance. To ensure correct estimation of the number of pedestrians in the network we allow the cameras to exchange raw data vectors (features). Each node provides an estimate of the number of clusters every two seconds which corresponds to 60 time frames.

We extract two color features from ground truth detections of pedestrians, see Figure 2.11. The first color feature is obtained by dividing the detected bounding box into three concentric circles and extracting a 10 bin histogram per color channel. The concatenation of the three color channels, which correspond to red, green, and blue (RGB), results in a 90-dimensional feature vector for each detected pedestrian. The second color feature is generated by cutting the detected bounding box horizontally into four equal parts and computing the average of each part for each color channel. This results in a 12-dimensional color feature. Finally, we concatenate the two color features to create a 102-dimensional feature vector. The pedestrians in the scene are dressed in relatively similar colors, which results in a sparse and linearly dependent color feature vector. To solve this problem we reduce the dimension of the color feature to 10 using principal component analysis (PCA).

Figure 2.12 depicts the average estimated number of clusters as a function of time frames for the multi-object multi-camera network setup. The true number of clusters is represented by the black staircase line, which increases every time a person that has not been observed by the camera network enters the scene. On the other hand, if a person re-enters, the number of clus-

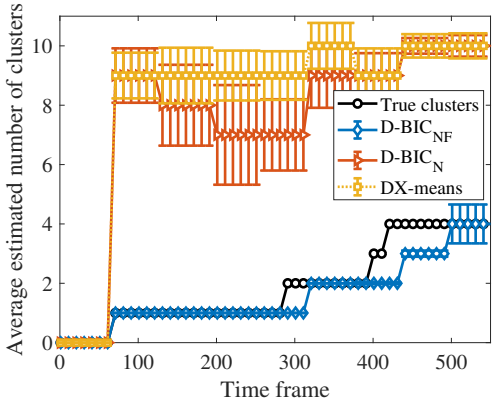


Figure 2.12: Average estimated number of clusters as a function of time frames for the multi-object multi-camera network setup.

ters is not incremented. The error bars show the fluctuations that are traced back to the random initialization of the clustering algorithms. D-BIC$_{\text{NF}}$ is able to continuously estimate the correct number of clusters with an empirical probability of detection $p_{\text{det}}^{\text{net}} = 77.09\%$, while the remaining distributed cluster enumeration methods have $p_{\text{det}}^{\text{net}} = 12.55\%$ and severely overestimate the number of pedestrians in the scene. D-BIC$_{\text{NF}}$ is also the best cluster enumeration method in terms of the mean absolute error with $\text{MAE}^{\text{net}} = 0.26$ followed by D-BIC$_{\text{N}}$ with $\text{MAE}^{\text{net}} = 5.65$ and DX-means with $\text{MAE}^{\text{net}} = 6.32$. Even in this challenging pedestrian enumeration problem, D-BIC$_{\text{NF}}$ is able to estimate the correct number of pedestrians with a high accuracy. In fact, errors can mainly be accounted to a delay in detecting a new cluster due to the lack of feature vectors from this cluster.

2.9 Summary

In this chapter, we presented three main contributions. First, we derived a general expression of the BIC for cluster analysis which is applicable to a broad class of data distributions. By imposing the multivariate Gaussian assumption on the distribution of the observations, we provided a closed-form BIC expression, referred to as BIC$_{\text{N}}$. We showed that the new BIC for cluster analysis has a different penalty term compared to the original BIC [Schwarz, 1978; Cavanaugh & Neath, 1999]. Moreover, BIC$_{\text{N}}$ contains information about the structure of the data in both its data fidelity and penalty terms because it is derived by taking the cluster analy-

sis problem into account. Further, BIC_N is incorporated into a two-step cluster enumeration algorithm which provides a principled way of estimating the number of clusters in a given data set. Numerical and real data experiments demonstrated the superiority of the proposed BIC for cluster analysis over existing Bayesian cluster enumeration methods. Next, we extended the two-step cluster enumeration algorithm by refining the penalty term of the new BIC for the finite sample regime, which results in the criterion BIC_{NF}. Simulation results confirmed the strength of BIC_{NF} for estimating the number of clusters in data sets with small sample sizes. BIC_{NF} achieves good performance results with a small additional computational complexity compared to BIC_N. Finally, we proposed two distributed and adaptive Bayesian cluster enumeration algorithms for an ad hoc sensor network where nodes communicate only with their immediate neighbors. The proposed algorithms adaptively estimate the number of clusters from streaming-in data. Experimental results demonstrated the superiority of the proposed methods over the DX-means algorithm on synthetic data sets and a challenging real data application.

3

Robust Bayesian Cluster Enumeration

3.1 Introduction

In real-world applications, the observed data is often subject to heavy tailed noise and outliers [Davé & Krishnapuram, 1997; Gallegos & Ritter, 2005; Garcá-Escudero et al., 2011; Zoubir et al., 2012; Zoubir et al., 2018] which obscure the true underlying structure of the data. Consequently, cluster enumeration becomes challenging when either the data is contaminated by a fraction of outliers or there exist deviations from the distributional assumptions.

In this chapter, we derive two robust Bayesian cluster enumeration criteria by modeling the data as a family of multivariate t_ν distributions. Specifically, a review of the state-of-the-art on robust cluster enumeration is given in Section 3.2 and the contributions made in this chapter are summarized in Section 3.3. The problem of estimating the number of clusters in a contaminated data set is formulated in Section 3.4 and a robust cluster enumeration algorithm that uses a clustering method to partition the data prior to the calculation of either of the derived robust criteria is presented in Section 3.5. A theoretical comparison of different robust Bayesian cluster enumeration criteria is made in Section 3.6. The performance of the proposed

algorithm is evaluated and compared to state-of-the-art methods using numerical and real data experiments in Section 3.7. Finally, the chapter is summarized in Section 3.8.

3.2 State-of-the-art

The estimation of the number of clusters in contaminated data has attracted interest in the literature, see [Wang et al., 2018; Neykov et al., 2007; Gallegos & Ritter, 2009; Gallegos & Ritter, 2010; Fraley & Raftery, 1998; Dasgupta & Raftery, 1998; Andrews & McNicholas, 2012; McNicholas & Subedi, 2012; Frigui & Krishnapuram, 1996; Hu et al., 2011; Binder et al., 2018; Garcá-Escudero et al., 2011; Wu et al., 2009; Zemene et al., 2016; Ott et al., 2014; García-Escudero et al., 2010] and the references therein. A popular approach in robust cluster analysis is to use the BIC, as derived by Schwarz [Schwarz, 1978; Cavanaugh & Neath, 1999], to estimate the number of data clusters after either removing outliers from the data [Neykov et al., 2007; Gallegos & Ritter, 2009; Gallegos & Ritter, 2010; Wang et al., 2018], modeling noise or outliers using an additional component in a mixture modeling framework [Fraley & Raftery, 1998; Dasgupta & Raftery, 1998], or exploiting the idea that the presence of outliers causes the distribution of the data to be heavy tailed and, subsequently, modeling the data as a mixture of heavy tailed distributions [Andrews & McNicholas, 2012; McNicholas & Subedi, 2012]. For example, modeling the contaminated data using a family of t_ν distributions [McLachlan & Peel, 1998; Peel & McLachlan, 2000; Kotz & Nadarajah, 2004; Lange et al., 1989; Liu & Rubin, 1995; Kibria & Joarder, 2005; Kent et al., 1994] provides a principled way of dealing with outliers by giving them less weight in the objective function. The family of t_ν distributions is flexible as it contains the heavy tailed Cauchy for $\nu = 1$ and the Gaussian distribution for $\nu \to \infty$ as special cases. Consequently, we model the clusters using a family of multivariate t_ν distributions and derive robust cluster enumeration criteria that account for outliers given that the degree of freedom parameter ν is sufficiently small.

3.3 Contributions in this Chapter

A robust Bayesian cluster enumeration criterion, BIC_{t_ν}, is derived by formulating the problem of estimating the number of clusters as maximization of the posterior probability of multivariate t_ν candidate models. We show that BIC_{t_ν} has a different penalty term compared to

the original BIC (BIC_{ot_ν}) [Schwarz, 1978; Cavanaugh & Neath, 1999], given that the candidate models in the original BIC are represented by a family of multivariate t_ν distributions. Interestingly, for BIC_{t_ν} both the data fidelity and the penalty terms depend on the assumed distribution for the data, while for the original BIC changes in the data distribution only affect the data fidelity term. Asymptotically, BIC_{t_ν} converges to BIC_{ot_ν}. As a result, our derivations also provide a justification for the use of the original BIC with multivariate t_ν candidate models from a cluster analysis perspective. Further, we refine the derivation of BIC_{t_ν} by providing an exact expression for its penalty term. This results in a robust criterion, $\mathrm{BIC}_{\mathrm{r}t_\nu}$, which behaves better than BIC_{t_ν} in the finite sample regime and converges to BIC_{t_ν} in the asymptotic regime. The derived robust cluster enumeration criteria require a clustering algorithm that partitions the data according to the number of clusters specified by each candidate model and provides an estimate of cluster parameters. Hence, we apply the expectation maximization (EM) algorithm to partition the data prior to the calculation of an enumeration criterion, resulting in a two-step approach. The proposed algorithm provides a unified framework for the estimation of the number of clusters and cluster memberships.

The main contributions have been submitted for publication in the IEEE Transactions on Signal Processing [Teklehaymanot et al., 2018e].

3.4 Problem Formulation

Let $\mathcal{X} \triangleq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^{r \times N}$ denote the observed data set which can be partitioned into K independent, mutually exclusive, and non-empty clusters $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$. Each cluster $\mathcal{X}_k$, for $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$, contains N_k data vectors that are realizations of independent and identically distributed (iid) multivariate t_ν random variables $\boldsymbol{x}_k \sim t_{\nu_k}(\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)$, where $\boldsymbol{\mu}_k \in \mathbb{R}^{r \times 1}$, $\boldsymbol{\Psi}_k \in \mathbb{R}^{r \times r}$, and $\nu_k \in \mathbb{R}^+$ represent the centroid, the scatter matrix, and the degree of freedom of the kth cluster, respectively. Let $\mathcal{M} \triangleq \{M_{L_{\min}}, \ldots, M_{L_{\max}}\}$ be a family of multivariate t_ν candidate models, where $L_{\min}$ and $L_{\max}$ represent the specified minimum and maximum number of clusters, respectively. Each candidate model $M_l \in \mathcal{M}$, for $l = L_{\min}, \ldots, L_{\max}$ and $l \in \mathbb{Z}^+$, represents a partition of $\mathcal{X}$ into l clusters with associated cluster parameter matrix $\boldsymbol{\Theta}_l = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_l]$, which lies in a parameter space $\Omega_l \subset \mathbb{R}^{q \times l}$. Assuming that

(A-3.1) the degree of freedom parameter ν_m, for $m = 1, \ldots, l$, is fixed at some prespecified value,

the parameters of interest reduce to $\boldsymbol{\theta}_m = [\boldsymbol{\mu}_m, \boldsymbol{\Psi}_m]^\top$ and $q = r(r+1)$. Our research goal is to estimate the number of clusters in $\mathcal{X}$ given $\mathcal{M}$ assuming that (A-2.1) is true.

3.5 Robust Bayesian Cluster Enumeration Algorithm

Given that assumptions (A-2.2)-(A-2.5) are fulfilled, we have derived a general Bayesian cluster enumeration criterion, referred to as BIC_G, in Section 2.5. However, since we assume multivariate t_ν candidate models, some of the assumptions made in the derivation of BIC_G require mathematical justification [Teklehaymanot et al., 2018e].

In this section, first, mathematical justification is provided for some of the assumptions in the case where each candidate model is represented by a multivariate t_ν distribution. Next, robust cluster enumeration criteria are derived. Finally, a robust two-step cluster enumeration algorithm is presented.

3.5.1 Robust Bayesian Cluster Enumeration Criteria for the Multivariate t_ν Distribution

When the candidate models are represented by a family of multivariate t_ν distributions, assumptions (A-2.3) and (A-2.5) require justification in order for BIC_G, given by (2.17), to be valid. For the multivariate t_ν distribution, the log-likelihood function is known to have multiple local maxima [Kent et al., 1994; Liu & Rubin, 1995]. In order for assumption (A-2.3) to hold, we have to show that $\hat{\boldsymbol{\theta}}_m$ is the global maximum of $\log \mathcal{L}(\boldsymbol{\theta}_m | \mathcal{X}_m)$, for $m = 1, \ldots, l$ and $l = L_{\min}, \ldots, L_{\max}$. $\hat{\boldsymbol{\theta}}_m$ is the maximum likelihood estimator of $\boldsymbol{\theta}_m$ and its derivation and the final expressions are given in Appendix A.2. We know that the global maximizer of $\log \mathcal{L}(\boldsymbol{\theta}_m | \mathcal{X}_m)$, is $\boldsymbol{\theta}_m^0$, where $\boldsymbol{\theta}_m^0$ is the true parameter vector. In [Maronna, 1976], it was proven that

$$\lim_{N_m \to \infty} \hat{\boldsymbol{\theta}}_m = \boldsymbol{\theta}_m^0$$

with probability one, where N_m is the number of data points in the mth cluster. As a result, asymptotically, assumption (A-2.3) holds. Assumption (A-2.5) directly follows because $\hat{\boldsymbol{\theta}}_m$ is

a maximizer of $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$. Hence, (2.17) holds for the case where the data is modeled by a family of t_ν distributions.

Now, assume that, for each candidate model $M_l \in \mathcal{M}$, there is a clustering algorithm that partitions $\mathcal{X}$ into l clusters and provides parameter estimates $\hat{\boldsymbol{\theta}}_m = [\hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Psi}}_m]^\top$, for $m = 1, \ldots, l$. Further, let (A-2.1)–(A-2.6) and (A-3.1) be fulfilled.

Theorem 3.1. *The posterior probability of M_l given $\mathcal{X}$ can be asymptotically approximated by*

$$\boxed{\begin{aligned} \mathrm{BIC}_{t_\nu}(M_l) &\triangleq \log p(M_l|\mathcal{X}) \\ &\approx \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) - \tfrac{q}{2}\textstyle\sum_{m=1}^{l} \log \epsilon, \end{aligned}} \tag{3.1}$$

where $q = \frac{1}{2}r(r+3)$ represents the number of estimated parameters per cluster and

$$\epsilon = \max\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \,,\; N_m \right). \tag{3.2}$$

The likelihood function, also called the data fidelity term, is given by

$$\begin{aligned} \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) \approx & \sum_{m=1}^{l} N_m \log N_m - \sum_{m=1}^{l} \frac{N_m}{2} \log |\hat{\boldsymbol{\Psi}}_m| + \sum_{m=1}^{l} N_m \log \frac{\Gamma\left((\nu_m + r)/2\right)}{\Gamma\left(\nu_m/2\right)(\pi\nu_m)^{r/2}} \\ & - \frac{1}{2}\sum_{m=1}^{l} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} (\nu_m + r) \log\left(1 + \frac{\delta_n}{\nu_m}\right), \end{aligned} \tag{3.3}$$

where $N_m = \#\mathcal{X}_m$, $\Gamma(\cdot)$ denotes the gamma function, and $\delta_n = (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)^\top \hat{\boldsymbol{\Psi}}_m^{-1}(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m)$ is the squared Mahalanobis distance. The second term in the second line of (3.1) is referred to as the penalty term.

Proof. Proving that (2.17) reduces to (3.1) for the multivariate t_ν distribution requires approximating $|\hat{\boldsymbol{J}}_m|$ and, consequently, writing a closed-form expression for $\mathrm{BIC}_{t_\nu}(M_l)$. A detailed proof is given in Appendix B.2. ∎

Once $\mathrm{BIC}_{t_\nu}(M_l)$ is computed for each candidate model $M_l \in \mathcal{M}$, the number of clusters in $\mathcal{X}$ is estimated as

$$\hat{K}_{\mathrm{BIC}_{t_\nu}} = \underset{l = L_{\min}, \ldots, L_{\max}}{\arg\max} \; \mathrm{BIC}_{t_\nu}(M_l). \tag{3.4}$$

Corollary 3.1. *When the data size is finite, one can opt to compute* $\log|\hat{\boldsymbol{J}}_m|$*, without asymptotic approximations to obtain a more accurate penalty term. In such cases, the posterior probability of* M_l *given* $\mathcal{X}$ *becomes*

$$\boxed{\mathrm{BIC}_{\mathrm{F}t_\nu}(M_l) \approx \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) - \frac{1}{2}\sum_{m=1}^{l} \log|\hat{\boldsymbol{J}}_m|,} \tag{3.5}$$

where the expression for $|\hat{\boldsymbol{J}}_m|$ *is given in Appendix C. Then, the number of clusters in* $\mathcal{X}$ *is estimated as*

$$\hat{K}_{\mathrm{BIC}_{\mathrm{F}t_\nu}} = \underset{l=L_{\min},\ldots,L_{\max}}{\arg\max}\ \mathrm{BIC}_{\mathrm{F}t_\nu}(M_l). \tag{3.6}$$

Both BIC_{t_ν} and $\mathrm{BIC}_{\mathrm{F}t_\nu}$ should be implemented as wrappers around a clustering algorithm since they require estimates of cluster parameters as an input. In the next section, we discuss the expectation maximization algorithm as a possible alternative to estimate cluster parameters and present the robust two-step cluster enumeration algorithm.

3.5.2 The Expectation Maximization Algorithm for Mixture of t_ν Distributions

The EM algorithm is widely used to estimate the parameters of the l-component mixture of t_ν distributions [Peel & McLachlan, 2000; McLachlan & Peel, 1998; Kotz & Nadarajah, 2004; Nadarajah & Kotz, 2008], which is given by

$$f(\boldsymbol{x}_n|M_l, \boldsymbol{\Phi}_l) = \sum_{m=1}^{l} \tau_m g(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Psi}_m, \nu_m), \tag{3.7}$$

where $g(\boldsymbol{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Psi}_m, \nu_m)$ denotes the r-variate t_ν pdf and $\boldsymbol{\Phi}_l = \left[\boldsymbol{\tau}_l, \boldsymbol{\Theta}_l^\top, \boldsymbol{\nu}_l\right]$. The mixing coefficients, denoted by $\boldsymbol{\tau}_l = [\tau_1, \ldots, \tau_l]^\top$, satisfy the constraints $0 < \tau_m < 1$ for $m = 1, \ldots, l$, and $\sum_{m=1}^{l} \tau_m = 1$. $\boldsymbol{\nu}_l = [\nu_1, \ldots, \nu_l]^\top$ are assumed to be known or estimated, e.g. using [Peel & McLachlan, 2000].

The EM algorithm contains two basic steps, namely the E step and the M step, which are

performed iteratively until a convergence condition is satisfied. The E step computes

$$\hat{v}_{nm}^{(i)} = \frac{\hat{\tau}_m^{(i-1)} g(\boldsymbol{x}_n; \boldsymbol{\mu}_m^{(i-1)}, \boldsymbol{\Psi}_m^{(i-1)}, \nu_m)}{\sum_{j=1}^{l} \hat{\tau}_j^{(i-1)} g(\boldsymbol{x}_n; \boldsymbol{\mu}_j^{(i-1)}, \boldsymbol{\Psi}_j^{(i-1)}, \nu_j)} \tag{3.8}$$

$$\hat{w}_{nm}^{(i)} = \frac{\nu_m + r}{\nu_m + \delta_n^{(i-1)}}, \tag{3.9}$$

where $\hat{v}_{nm}^{(i)}$ is the posterior probability that $\boldsymbol{x}_n$ belongs to the mth cluster at the ith iteration and $\hat{w}_{nm}^{(i)}$ is the weight given to $\boldsymbol{x}_n$ by the mth cluster at the ith iteration. Once $\hat{v}_{nm}^{(i)}$ and $\hat{w}_{nm}^{(i)}$ are calculated, the M step updates cluster parameters as follows:

$$\hat{\tau}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)}}{N} \tag{3.10}$$

$$\hat{\mu}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)} w_{nm}^{(i)} \boldsymbol{x}_n}{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)} w_{nm}^{(i)}} \tag{3.11}$$

$$\hat{\boldsymbol{\Psi}}_m^{(i)} = \frac{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)} w_{nm}^{(i)} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m^{(i)})(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_m^{(i)})^\top}{\sum_{n=1}^{N} \hat{v}_{nm}^{(i)}} \tag{3.12}$$

As the name suggests, the robust two-step cluster enumeration algorithm contains two steps, which are the model-based clustering step and the cluster enumeration step. The model-based clustering step performs iterations of the EM algorithm until convergence followed by hard cluster membership assignments. In the cluster enumeration step, either $\mathrm{BIC}_{t_\nu}(M_l)$ or $\mathrm{BIC}_{\mathrm{F}t_\nu}(M_l)$ is computed for each model $M_l \in \mathcal{M}$. The output of the algorithm is not only an estimate of the number of clusters but also cluster membership assignments. A pseudo-code that describes the working principle of the robust two-step cluster enumeration algorithm is given in Algorithm 3.1.

Given that the degree of freedom parameter ν is fixed at some finite value and the cluster enumeration criterion used is BIC_{t_ν}, the computational complexity of Algorithm 3.1 is the sum of the run times of the two steps. Since the initialization, i.e., the K-medians algorithm is performed only for a few iterations, the computational complexity of the first step is dominated by the EM algorithm and it is given by $\mathcal{O}(Nr^2 l i_{\max})$ for a single candidate model M_l, where $i_{\max}$ is a fixed stopping threshold of the EM algorithm. The computational complexity of $\mathrm{BIC}_{t_\nu}(M_l)$ is $\mathcal{O}(Nr^2)$, which is much smaller than the run-time of the EM algorithm and,

as a result, it can easily be ignored in the run-time analysis of the proposed algorithm. Hence, the total computational complexity of Algorithm 3.1 is $\mathcal{O}(Nr^2(L_{\min} + \ldots + L_{\max})i_{\max})$. Note that if $\text{BIC}_{\text{F}t_\nu}$ is used in Algorithm 3.1 instead of BIC_{t_ν}, the computational complexity of the algorithm increases significantly with the increase in the number of features (r) due to the calculation of the determinant of the Fisher information matrix of each cluster in each candidate model, which is given by (C.1).

3.6 Comparison of Different Robust Bayesian Cluster Enumeration Criteria

Model selection criteria that are derived by maximizing the posterior probability of candidate models given data are known to have a common form [Stoica & Selen, 2004; Rao & Wu, 1989] that is consistent with

$$\log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) - \eta, \tag{3.13}$$

where $\log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X})$ is the data fidelity term and η is the penalty term. The proposed robust cluster enumeration criteria, BIC_{t_ν} and $\text{BIC}_{\text{F}t_\nu}$, and the original BIC with multivariate t_ν candidate models, $\text{BIC}_{\text{o}t_\nu}$, [Andrews & McNicholas, 2012; McNicholas & Subedi, 2012] have an identical data fidelity term. The difference in these criteria lies in their penalty terms, which are given by

$$\text{BIC}_{t_\nu}: \quad \eta = \frac{q}{2}\sum_{m=1}^{l}\log\epsilon \tag{3.14}$$

$$\text{BIC}_{\text{F}t_\nu}: \quad \eta = \frac{1}{2}\sum_{m=1}^{l}\log|\hat{\boldsymbol{J}}_m| \tag{3.15}$$

$$\text{BIC}_{\text{o}t_\nu}: \quad \eta = \frac{ql}{2}\log N, \tag{3.16}$$

where ϵ and $|\hat{\boldsymbol{J}}_m|$ are given by (3.2) and (C.1), respectively. Note that $\text{BIC}_{\text{F}t_\nu}$ calculates an exact value of the penalty term, while BIC_{t_ν} and $\text{BIC}_{\text{o}t_\nu}$ compute its asymptotic approximation. In the finite sample regime the penalty term of $\text{BIC}_{\text{F}t_\nu}$ is stronger than the penalty term of BIC_{t_ν}, while asymptotically all three criteria have an identical penalty term.

Algorithm 3.1 Robust two-step cluster enumeration approach

Inputs: $\mathcal{X}$, $L_{\min}$, $L_{\max}$, and ν
for $l = L_{\min}, \ldots, L_{\max}$ do
  Step 1: model-based clustering
  Step 1.1: the EM algorithm
  for $m = 1, \ldots, l$ do
    Initialize $\hat{\boldsymbol{\mu}}_m^0$ using the K-medians algorithm
    Initialize $\hat{\boldsymbol{\Psi}}_m^0$ using the sample covariance estimator
    $\hat{\tau}_m^0 = \frac{N_m}{N}$
  end for
  for $i = 1, 2, \ldots, i_{\max}$ do
    E step:
    for $n = 1, \ldots, N$ do
      for $m = 1, \ldots, l$ do
        Calculate $\hat{v}_{nm}^{(i)}$ and $\hat{w}_{nm}^{(i)}$ using (3.8) and (3.9), respectively
      end for
    end for
    M step:
    for $m = 1, \ldots, l$ do
      Determine $\hat{\tau}_m^{(i)}$, $\hat{\boldsymbol{\mu}}_m^{(i)}$, and $\hat{\boldsymbol{\Psi}}_m^{(i)}$ via (3.10)-(3.12)
    end for
    Check for the convergence of either $\hat{\boldsymbol{\Phi}}_l^{(i)}$ or $\log \mathcal{L}(\hat{\boldsymbol{\Phi}}_l^{(i)}|\mathcal{X})$
    if convergence condition is satisfied then
      Exit for loop
    end if
  end for
  Step 1.2: hard clustering
  for $n = 1, \ldots, N$ do
    for $m = 1, \ldots, l$ do

$$\iota_{nm} = \begin{cases} 1, & m = \underset{j=1,\ldots,l}{\arg\max}\ \hat{v}_{nj}^{(i)} \\ 0, & \text{otherwise} \end{cases}$$

    end for
  end for
  for $m = 1, \ldots, l$ do
    $N_m = \sum_{n=1}^{N} \iota_{nm}$
  end for
  Step 2: calculate either $\mathrm{BIC}_{t_\nu}(M_l)$ or $\mathrm{BIC}_{\mathrm{F}t_\nu}(M_l)$ using (3.1) or (3.5), respectively
end for
Estimate the number of clusters in $\mathcal{X}$ via (3.4) or (3.6)

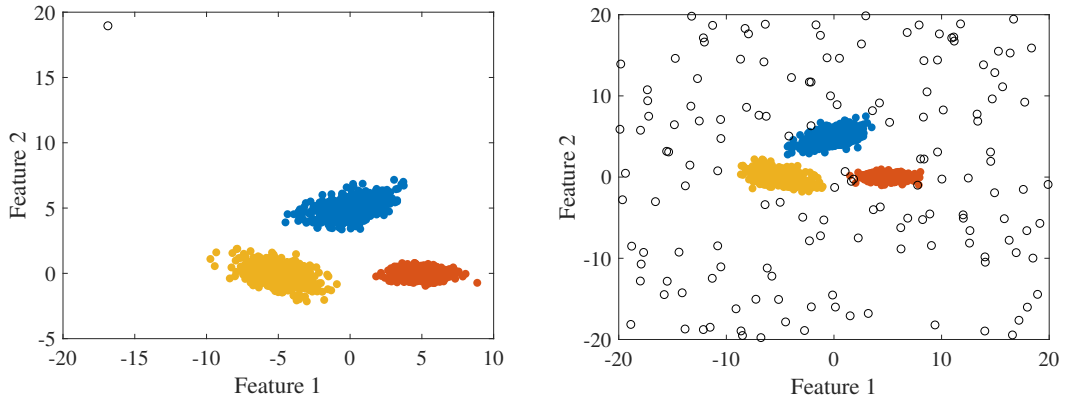
Remark. *A modification of the data distribution of the candidate models only affects the data fidelity term of the original BIC [Schwarz, 1978; Cavanaugh & Neath, 1999]. However, given that the BIC is specifically derived for cluster analysis, we showed that both the data fidelity and penalty terms change as the data distribution of the candidate models changes, see (3.1) and (2.19).*

Remark. *When the degree of freedom parameter $\nu \to \infty$,* BIC_{t_ν} *converges to* $\mathrm{BIC}_{\mathrm{N}}$, *where* $\mathrm{BIC}_{\mathrm{N}}$ *is given by (2.19).*

A related robust cluster enumeration method that uses the original BIC to estimate the number of clusters is the trimmed BIC (TBIC) [Neykov et al., 2007]. The TBIC estimates the number of clusters using the original BIC with Gaussian candidate models after trimming some percentage of the data. In TBIC, the fast trimmed likelihood estimator (FAST-TLE) is used to obtain maximum likelihood estimates of cluster parameters. The FAST-TLE is computationally expensive since it carries out a trial and a refinement step multiple times, see [Neykov et al., 2007] for details.

3.7 Experimental Results

We compare the performance of the proposed robust two-step algorithm with state-of-the-art cluster enumeration methods using numerical and real data experiments. In addition to the methods discussed in Section 3.6, we compare our cluster enumeration algorithm with the gravitational clustering (GC) [Binder et al., 2018] and the X-means [Pelleg & Moore, 2000] algorithm. All experimental results are an average of 300 Monte Carlo runs. The degree of freedom parameter is set to $\nu = 3$ for all methods that have multivariate t_ν candidate models. We use the author's implementation of the gravitational clustering algorithm [Binder et al., 2018]. For the TBIC, we trim 10% of the data and perform 10 iterations of the trial and refinement steps. The minimum and maximum number of clusters is set to $L_{\min} = 1$ and $L_{\max} = 2K$, where K denotes the true number of clusters in the data under consideration, whenever required. The performance measures that were defined in Section 2.6.5.1 are used here.



(a) Data-3.1 (b) Data-3.2 with $\alpha = 10\%$

Figure 3.1: Data-3.1 and Data-3.2 with $\alpha = 10\%$, where filled circles represent clean data and an open circle denotes an outlier.

3.7.1 Numerical Experiments

We study the performance of different cluster enumeration methods as a function of the amount of outliers in the data, the number of features, the amount of overlap between clusters, and cluster heterogeneity.

Analysis of the sensitivity of different cluster enumeration methods to outliers

We generate two data sets which contain realizations of 2-dimensional random variables $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $k = 1, 2, 3$, with cluster centroids $\boldsymbol{\mu}_1 = [0, 5]^\top$, $\boldsymbol{\mu}_2 = [5, 0]^\top$, $\boldsymbol{\mu}_3 = [-5, 0]^\top$, and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}, \boldsymbol{\Sigma}_3 = \begin{bmatrix} 2 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}.$$

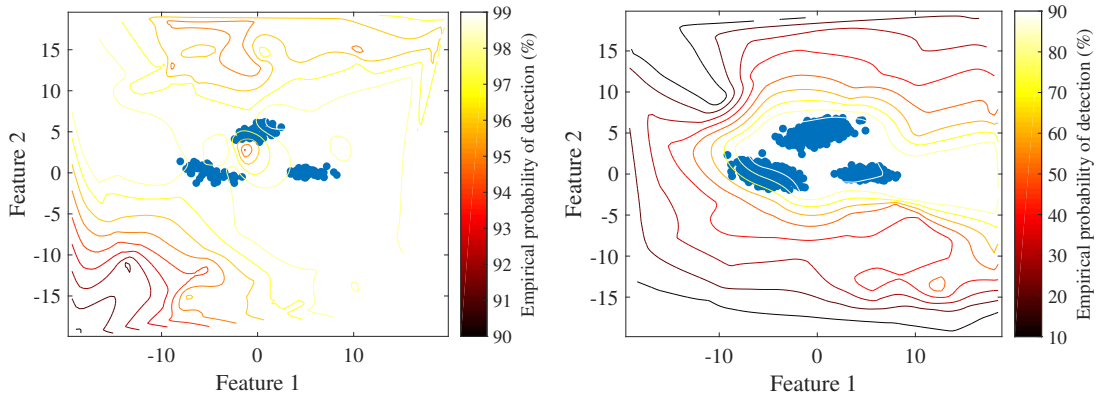
The first data set (Data-3.1), depicted in Figure 3.1a, replaces a randomly selected data point with an outlier that is generated from a uniform distribution over the range $[-20, 20]$ on each variate at each iteration. The sensitivity of different cluster enumeration methods to a single replacement outlier over 100 iterations as a function of the number of data vectors per cluster (N_k) is displayed in Table 3.1. From the compared methods, our robust criterion $\text{BIC}_{\text{FT}3}$

Table 3.1: The sensitivity of different cluster enumeration methods to the presence of a single replacement outlier as a function of the number of data points per cluster.

N_k		50	100	250	500
BIC_{t_3}	p_{det}	43.20	92.18	99.77	**100**
	MAE	1.28	0.11	0.002	**0**
$\text{BIC}_{\text{F}t_3}$	p_{det}	96.89	**100**	**100**	**100**
	MAE	0.03	**0**	**0**	**0**
BIC_{ot_3}	p_{det}	88.13	99.5	99.98	**100**
	MAE	0.18	0.005	0.0002	**0**
TBIC	p_{det}	**98.75**	99.26	98.92	98.8
	MAE	**0.013**	0.008	0.01	0.01
GC	p_{det}	73.07	94.85	99.80	**100**
	MAE	0.29	0.05	0.002	**0**
BIC_{N}	p_{det}	10.92	15.60	33.82	42.20
	MAE	1.25	1.13	0.99	0.83
X-means	p_{det}	1.24	1.17	1.38	0.17
	MAE	2.69	2.67	2.33	2.13

has the best performance in terms of both p_{det} and MAE. Except for $\text{BIC}_{\text{F}t_3}$ and the TBIC, the performance of all methods deteriorates when N_k, for $k = 1, 2, 3$, is small and, notably, BIC_{t_3} performs poorly. This behavior is attributed to the fact that BIC_{t_3} is an asymptotic criterion and in the small sample regime its penalty term becomes weak which results in an increase in the empirical probability of overestimation. BIC_{N} and X-means are very sensitive to the presence of a single outlier because they model individual clusters as multivariate Gaussian. X-means performs even worse than BIC_{N} since it uses the K-means algorithm to cluster the data, which is ineffective in handling elliptical clusters. An illustrative example of the sensitivity of $\text{BIC}_{\text{F}t_3}$ and BIC_{N} to the presence of an outlier is displayed in Figure 3.2. Despite the difference in N_k, when the outlier is either in one of the clusters or very close to one of the clusters, both $\text{BIC}_{\text{F}t_3}$ and BIC_{N} are able to estimate the correct number of clusters reasonably well. The difference between the two methods arises when the outlier is far away from the bulk of data. While $\text{BIC}_{\text{F}t_3}$ is still able to estimate the correct number of clusters, BIC_{N} starts to overestimate the number of clusters.

The second data set (Data-3.2), shown in Figure 3.1b, contains $N_k = 500$ data points in each



(a) $\text{BIC}_{\text{F}t_3}$ for $N_k = 50$ (b) BIC_{N} for $N_k = 500$

Figure 3.2: Sensitivity curves of $\text{BIC}_{\text{F}t_3}$ and BIC_{N} at different values of N_k. The sensitivity curve demonstrates the sensitivity of a method to the presence of an outlier relative to the position of the outlier.

cluster k and replaces a certain percentage of the data set with outliers that are generated from a uniform distribution over the range $[-20, 20]$ on each variate. Data-3.2 is generated in a way that no outlier lies inside one of the data clusters. In this manner, we make sure that outliers are points that do not belong to the bulk of data. Figure 3.3a shows the empirical probability of detection as a function of the percentage of outliers (α). GC is able to correctly estimate the number of clusters for $\alpha > 3\%$ at the cost of increased computation compared to the other methods. The proposed robust criteria, BIC_{t_3} and $\text{BIC}_{\text{F}t_3}$, and the original BIC, BIC_{ot_3}, behave similarly and are able to estimate the correct number of clusters when $\alpha \leq 3\%$. The behavior of these methods is rather intuitive because as the amount of outliers increases, then the methods try to explain the outliers by opening a new cluster. A similar trend is observed for TBIC even though its curve decays slowly. BIC_{N} is able to estimate the correct number of clusters 99% of the time when there are no outliers in the data set. However, even 1% of outliers is enough to drive BIC_{N} into overestimating the number of clusters.

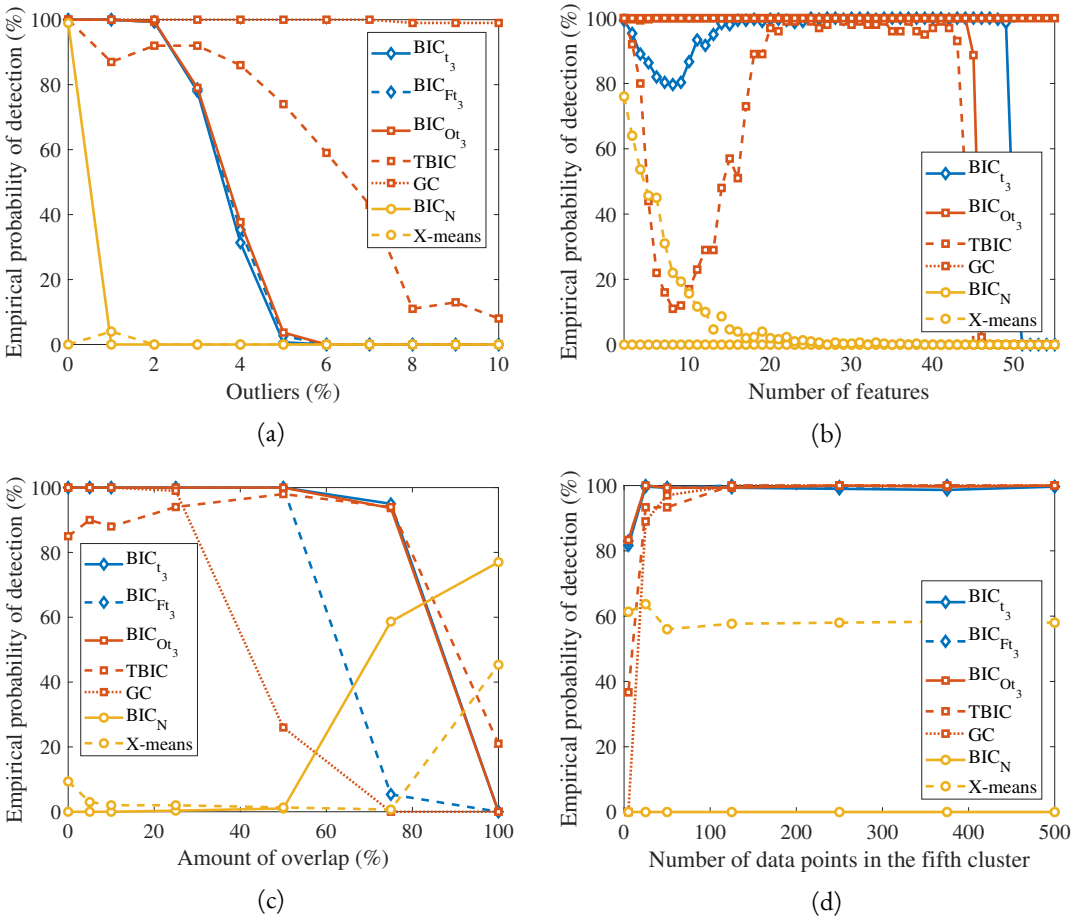


Figure 3.3: The empirical probability of detection in % as a function of different parameters.

Impact of the increase in the number of features on the performance of cluster enumeration methods

We generate realizations of the random variables $\boldsymbol{x}_k \sim t_3\left(\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k\right)$ whose cluster centroids and scatter matrices are given by

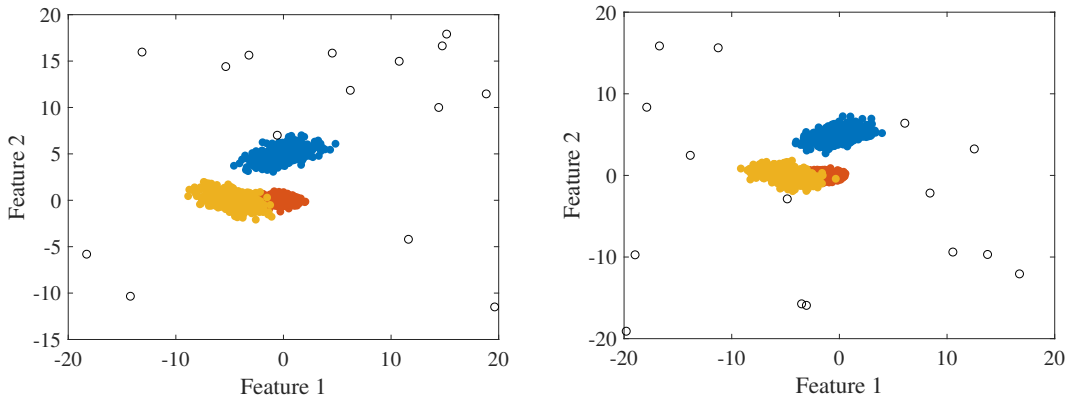
$$\boldsymbol{\mu}_k = c\mathbf{1}_{r\times 1}$$
$$\boldsymbol{\Psi}_k = \boldsymbol{I}_r,$$

with $c \in \{0, 15\}$, $\mathbf{1}_{r\times 1}$ denoting an r-dimensional all one column vector, $\boldsymbol{I}_r$ representing an $r \times r$-dimensional identity matrix, and $k = 1, 2$. For this data set, referred to as Data-3.3, the

number of features r is varied in the range $r = 2, 3, \ldots, 55$ and the number of data points per cluster is set to $N_k = 500$. Because $\nu = 3$, Data-3.3 contains realizations of heavy tailed distributions and, as a result, the clusters contain outliers. The empirical probability of detection as a function of the number of features is displayed in Figure 3.3b. The performance of GC appears to be invariant to the increase in the number of features, while the remaining methods are affected. But, compared to the other cluster enumeration methods, GC is computationally very expensive. BIC_{ot_3} outperforms BIC_{t_3} and the TBIC when the number of features is low, while the proposed criterion BIC_{t_3} outperforms both methods in high dimensions. $\text{BIC}_{\mp t_3}$ is not computed for this data set because it is computationally expensive and it is not beneficial given the large number of samples.

Analysis of the sensitivity of different cluster enumeration methods to cluster overlap

To study the sensitivity to cluster overlap, we consider Data-3.2 with 1% outliers and vary the distance between the second and the third centroid such that the percentage of overlap between the two clusters takes on a value from the set $\{0, 5, 10, 25, 50, 75, 100\}$. As an example, Figure 3.4a and Figure 3.4b show Data-3.2 with 25% and 75% overlap, respectively, between the second and the third cluster. The empirical probability of detection as a function of the amount of overlap is depicted in Figure 3.3c. The best performance is achieved by BIC_{t_3} and BIC_{ot_3} and, remarkably, both cluster enumeration criteria are able to correctly estimate the number of clusters even when there exists 75% overlap between the two clusters. As expected, when the amount of overlap is 100%, most methods underestimate the number of clusters to two. While it may appear that the enumeration performance of BIC_{N} increases for increasing amounts of overlap, in fact BIC_{N} groups the two overlapping clusters into one and attempts to explain the outliers by opening a new cluster. A similar trend is observed for X-means. GC is inferior in performance to the other robust methods, and experiences an increase in the empirical probability of underestimation.

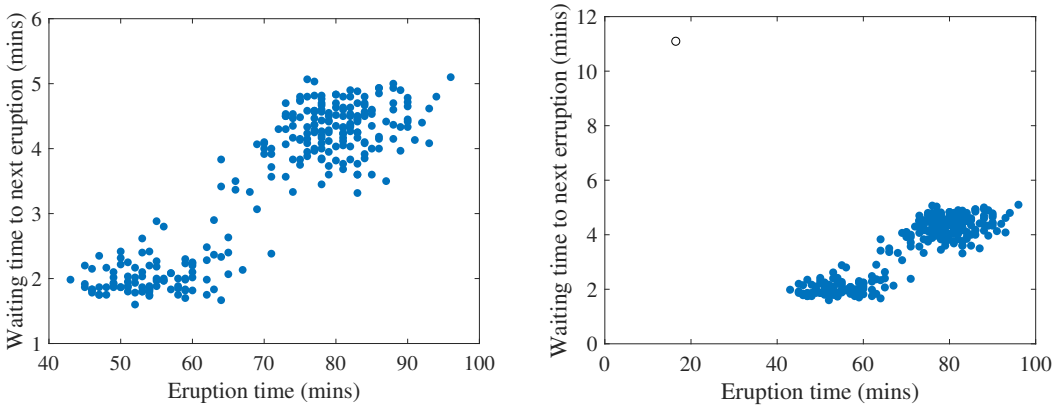


(a) 25% overlap (b) 75% overlap

Figure 3.4: Data-3.2 with 1% outliers and varying percentage of overlap.

Analysis of the sensitivity of cluster enumeration methods to cluster heterogeneity

To analyze the sensitivity to cluster heterogeneity, we generate realizations of 2-dimensional random variables $\boldsymbol{x}_k \sim t_3(\boldsymbol{\mu}_k, \boldsymbol{\Psi}_k)$, where the cluster centroids $\boldsymbol{\mu}_k$ are selected at random from a uniform distribution in the range $[-200, 200]$ in each variate and the scatter matrices are set to $\boldsymbol{\Psi}_k = \boldsymbol{I}_r$ for $k = 1, \ldots, 5$. The data set is generated in a way that there is no overlap between the clusters. The number of data points in the first four clusters is set to $N_k = 500$, while N_5 is allowed to take on values from the set $\{5, 25, 50, 125, 250, 375, 500\}$. This data set (Data-3.4) contains multiple outliers since each cluster contains realizations of heavy tailed t_3 distributed random variables. The empirical probability of detection as a function of the number of data points in the fifth cluster is shown in Figure 3.3d. The proposed cluster enumeration methods, BIC_{t_3} and $\text{BIC}_{\text{F}t_3}$, are able to estimate the correct number of clusters with a high accuracy even when the fifth cluster contains only 1% of the data available in the other clusters. A similar performance is observed for BIC_{ot_3}. TBIC and GC are slightly inferior in performance to the other robust cluster enumeration methods. When the number of data points in the fifth cluster increases, all robust methods perform well in estimating the number of clusters. Interestingly, X-means outperforms BIC_{N} since the considered clusters are all spherical. BIC_{N} overestimates the number of clusters and possesses the largest MAE.



(a) Clean data (b) Contaminated data

Figure 3.5: Clean and contaminated versions of the Old Faithful geyser data set.

3.7.2 Real Data Results

Old Faithful geyser data set

Old Faithful is a geyser located in Yellowstone National Park in Wyoming, United States. This data set, depicted in Figure 3.5a, was used in the literature for density estimation [Izenman, 2008], time series analysis [Azzalini & Bowman, 1990], and cluster analysis [Bishop, 2006; Hennig, 2003]. The performance of different cluster enumeration methods on the clean and contaminated versions of the Old Faithful data set is reported in Table 3.2. The contaminated version, shown in Figure 3.5b, is generated by replacing a randomly selected data point with an outlier similar to the way Data-3.1 was generated. Most methods are able to estimate the correct number of clusters 100% of the time for the clean version of the Old Faithful data set. Our criteria, BIC_{t_3} and $\text{BIC}_{\text{F}t_3}$, and $\text{BIC}_{\text{O}t_3}$ are insensitive to the presence of a single replacement outlier, while TBIC exhibits slight sensitivity. In the presence of an outlier, the performance of BIC_{N} deteriorates due to an increase in the empirical probability of overestimation. In fact, BIC_{N} finds 3 clusters 100% of the time. GC shows the worst performance and possesses the highest MAE.

Next, we replace a certain percentage of the Old Faithful data set with outliers and study the performance of different cluster enumeration methods. The outliers are generated from a uniform distribution over the range $[-20, 20]$ on each variate. The empirical probability of detection as a function of the percentage of replacement outliers is depicted in Figure 3.6. Our

Table 3.2: The performance of different cluster enumeration methods on a clean and a contaminated version of the Old Faithful data set.

		Old Faithful	Old Faithful with a single outlier
BIC_{t_3}	p_{det}	**100**	**100**
	MAE	**0**	**0**
$\text{BIC}_{\text{F}t_3}$	p_{det}	**100**	**100**
	MAE	**0**	**0**
BIC_{ot_3}	p_{det}	**100**	**100**
	MAE	**0**	**0**
TBIC	p_{det}	**100**	92.03
	MAE	**0**	0.09
GC	p_{det}	0	0
	MAE	10.34	10.26
BIC_{N}	p_{det}	**100**	5.08
	MAE	**0**	1.36
X-means	p_{det}	0	0
	MAE	2	2

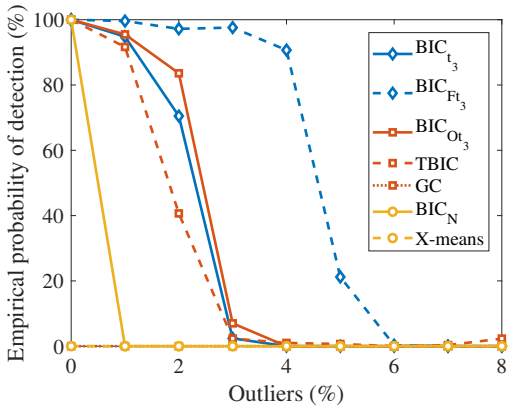


Figure 3.6: Empirical probability of detection in % for the Old Faithful data set as a function of the percentage of replacement outliers.

criterion $\text{BIC}_{\text{F}t_3}$ outperforms the other methods by a considerable margin. Although BIC_{t_3}, BIC_{ot_3}, and TBIC are able to estimate the correct number of clusters reasonably well for clean data, their performance deteriorates quickly as the percentage of outliers increases. X-means and GC overestimate the number of clusters for 100% of the cases.

3.8 Summary

In this chapter, we focused on the challenges in robust cluster analysis and derived a robust cluster enumeration criterion. Further, we refined the penalty term of the robust criterion for the finite sample regime. Since both robust criteria require cluster parameter estimates as an input, we proposed a two-step cluster enumeration algorithm that uses the EM algorithm to partition the data and estimate cluster parameters prior to the calculation of either of the robust criteria. The following two statements can be made with respect to the original BIC: First, the asymptotic criterion derived specifically for cluster analysis has a different penalty term compared to the original BIC based on multivariate t_ν candidate models. Second, since the derived asymptotic criterion converges to the original BIC as data size goes to infinity, we are able to provide a justification for the use of the original BIC with multivariate t_ν candidate models. The performance of the proposed cluster enumeration algorithm is demonstrated using numerical and real data experiments. We showed superiority of the proposed algorithm in estimating the number of clusters in contaminated data sets.

PART II

Object Labeling

4

Object Labeling in Distributed Sensor Networks

4.1 Introduction

Driven by a wide range of applications, distributed and adaptive signal processing has attracted much attention recently, see [Al-Sayed et al., 2017; Schizas et al., 2009; Lopes & Sayed, 2007; Cattivelli et al., 2008; Lorenzo et al., 2017; Szurley et al., 2015; Li & Fang, 2007; Chou et al., 2003] and the references therein. As a result, new paradigms for signal and parameter estimation have been developed. One such paradigm is the multiple devices multiple tasks (MDMT) paradigm where multiple devices communicate with in some neighborhood to solve multiple complex signal processing tasks [Bertrand & Moonen, 2012; Chen et al., 2015; Plata-Chaves et al., 2017; Plata-Chaves et al., 2015; Chouvardas et al., 2015; Bogdanovic et al., 2014]. This is different from other information and communication technology (ICT) paradigms, where stand-alone devices merely focus on individual tasks or multiple devices perform one single joint task, as it is typically assumed in a classical wireless sensor network.

Among the many sensing modalities, our focus lies on camera networks. A distributed camera network containing multiple heterogeneous devices, such as smart phones, tablets and/or

handheld cameras, which neither has a predefined network structure nor a centralized computing unit can make use of the MDMT paradigm. Nodes in a distributed camera network can be interested in, for example, image enhancement, object detection, pose analysis, and object tracking. In most real-world applications, the signal received by these nodes is contaminated by noise, contains frequent object occlusions, and lacks visibility in densely crowded scenes. Under the MDMT paradigm, such nodes can benefit from cooperation with their immediate neighbors to solve their signal processing task of interest given that the nodes share common interests or observations. For cooperation to be successful, it is thus necessary to account for a distributed labeling scheme that allows to uniquely identify every object of interest observed by the nodes in the network. Only in this way, a node can make sure that the information it received from its neighbors refers to its object of interest. Furthermore, from a communication cost perspective, knowledge about which node sees which object allows the formation of interest specific clusters.

In this chapter, we present distributed labeling algorithms in the context of wireless camera networks where no central unit is available to fuse the information collected by the nodes in the network. In Section 4.2, the state-of-the-art on object labeling and tracking in distributed sensor networks is discussed. In Section 4.3, the contributions made in this chapter are summarized. The proposed distributed object labeling algorithm for camera networks whose nodes monitor a stationary scene is presented in Section 4.4. In Section 4.5, the proposed adaptive object labeling and tracking algorithm for camera networks whose nodes monitor a non-stationary scene is discussed and applied to a real data use case. Finally, the chapter is summarized in Section 4.6.

4.2 State-of-the-art

Several methods have been proposed in the last years that frame object labeling in form of a data clustering and classification problem after extracting source-specific features [Brooks et al., 2003; Hai et al., 2012; Kokiopoulou & Frossard, 2011; Malhotra et al., 2008; Nowak, 2003; Forero et al., 2011; Tu & Sayed, 2014; Wang et al., 2009; Binder et al., 2015; Bahari et al., 2016; Binder et al., 2016]. However, some of them still assume the presence of a fusion center [Hai et al., 2012; Malhotra et al., 2008], are hardly real-time capable [Brooks et al., 2003] or require a set of prelabeled training data [Kokiopoulou & Frossard, 2011; Wang et al., 2009].

In terms of distributed signal processing without a fusion center, several adaptive strategies, such as incremental, consensus, and diffusion algorithms have been developed in the last few years, see [Sayed, 2014a] for an overview and a comparison of these methods. In [Forero et al., 2011], a distributed K-means algorithm that uses the consensus strategy was proposed. An alternative hybrid diffusion-based approach which consists of a classification method based on preliminary clustering of the data has been presented in [Binder et al., 2015; Binder et al., 2016].

Object labeling becomes even more challenging when the scene of interest is non-stationary. In such cases, the distributed nodes require to not only label objects of interest but also track them over time. The tracking and labeling of multiple objects in multiple cameras is fundamental, e.g. for applications such as video surveillance, autonomous driving, and sports analysis. Multi-object multi-camera tracking systems must maintain consistent labels of objects of interest across camera views and over time to take advantage of the information available from different camera views in the network. Previous approaches to the labeling of multiple objects across camera views include principal axis-based integration of multi-camera information [Du & Piater, 2007], nonlinear manifold learning and system dynamics identification [Morariu & Camps, 2006], and approaches that either use homography or camera calibration information to register stationary camera views on top of a known ground plane [Kang et al., 2004; Khan & Shah, 2006; Taj & Cavallaro, 2009; Berclaz et al., 2011; Fleuret et al., 2008]. These state-of-the-art methods are centralized approaches in the sense that camera views are aggregated into a ground plane to make sure that unique and consistent labels are assigned to the objects in the scene of interest.

4.3 Contributions in this Chapter

The first contribution concerns two aspects. First, considering that a common planar scene is observed by distributed cameras with different viewpoints, we present the use of histograms of oriented gradients (HOG) [Dalal & Triggs, 2005] and histogram of color descriptors for the task of object labeling. Second, we adapt the hybrid classification scheme that was developed in [Binder et al., 2015; Binder et al., 2016] so that it is capable of labeling objects that are captured by hand held cameras. This is a new application, and we demonstrate using real data that high labeling rates can be achieved. Since many areas of engineering today concern prob-

lems where the distribution of the measurements is far from Gaussian as it contains outliers, which cause the distribution to be heavy tailed [Zoubir et al., 2012; Zoubir et al., 2018], we demonstrate that the use of a robust technique as in [Binder et al., 2015; Binder et al., 2016] (K-medians) provides higher labeling rates as compared to the distributed K-means algorithm. We also compare the distributed strategies to their respective centralized counterparts, where all information is available and computing is done at a fusion center.

The second contribution lies in developing a fully distributed algorithm which does not require camera view registration to ensure that the same object is provided with the same identity in a multi-camera network. This is radically different from state-of-the-art methods [Kang et al., 2004; Khan & Shah, 2006; Taj & Cavallaro, 2009; Berclaz et al., 2011; Fleuret et al., 2008; Du & Piater, 2007; Morariu & Camps, 2006] which require the aggregation of camera views in order to label objects of interest. The information available in a neighborhood of cameras is exploited to provide unique and consistent labels to multiple objects across multiple camera views and time frames. Each node solves a regularized cost function during the assignment of a label to a particular object to exploit both the information obtained from a local (single node) Kalman filter-based tracker and a diffusion-based labeling algorithm. The advantage of such an approach is that it is applicable to ad hoc networks of mobile cameras. Furthermore, the approach is robust against a single node failure and since it is based on the diffusion principle [Sayed, 2014a] it is adaptive and scalable.

The first and second contributions have been published in [Teklehaymanot et al., 2015] and [Teklehaymanot et al., 2017], respectively.

4.4 Object Labeling in a Stationary Scene

In this section, we present distributed labeling strategies in cases where the scene observed by the distributed camera network is stationary over time [Teklehaymanot et al., 2015]. In particular, we consider a common planar scene, which is observed by cameras with different viewpoints, and demonstrate the use of a robust data clustering algorithm in order to provide unique identity to objects of interest.

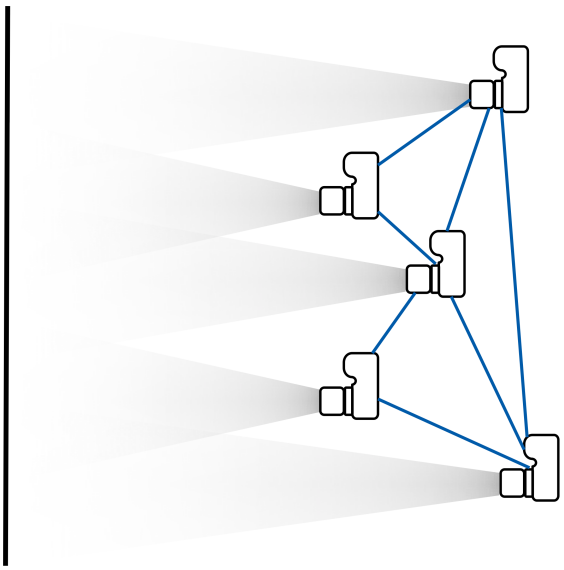


Figure 4.1: Example of a camera network observing a common planar scene.

4.4.1 Problem Formulation

Consider a wireless camera network as the one depicted in Figure 4.1, where the nodes have overlapping observations of some scene. Each node might be interested in a particular part or object in the scene that needs to be enhanced, e.g., in terms of resolution or removal of occlusions. We assume that

(A-4.1) more than one node is interested in every object or region of interest,

(A-4.2) the observed scene can be well approximated by a planar scene (e.g., the captured scene is far from the cameras), and

(A-4.3) every node is only interested in one object or region.

Under this setting, nodes would like to label the objects or regions of interest in order to identify common interests among them. For such a task, each node extracts a window patch that contains the object of interest and computes a descriptor of it. Using these descriptors (or feature vectors), we then seek to reach a global labeling scheme through the local interaction of nodes.

More formally, consider a camera network with J nodes. The neighborhood of node $j \in \mathcal{J} \triangleq \{1, \ldots, J\}$, which is denoted by $\mathcal{B}_j$, contains nodes that are directly connected with

node $j \in \mathcal{J}$. Let $\boldsymbol{x}_j \in \mathbb{R}^{r\times 1}$ denote the r-dimensional descriptor extracted at the jth node which belongs to class $\mathcal{C}_k \in \{1,\ldots,K\}$, where k is the class (cluster) label. K denotes the number of objects of interest in the scene, which is assumed to be known or estimated via the cluster enumeration method discussed in Section 2.6.1, Section 2.7, or Section 3.5.1. Let $\hat{\mathcal{C}}_{jk} = F(\boldsymbol{x}_j)$ be the predicted class of feature vector $\boldsymbol{x}_j$, where $F : \mathbb{R}^{r\times 1} \mapsto \{1,\ldots,K\}, F \in \mathcal{F}$ is some prediction function within the family of functions $\mathcal{F}$. Then, our goal is to solve the following optimization problem

$$\min_{F\in\mathcal{F}} \; \mathbb{1}_{\{F(\boldsymbol{x}_j)-\mathcal{C}_k\}}, \tag{4.1}$$

where $\mathbb{1}_{\{F(\boldsymbol{x}_j)-\mathcal{C}_k\}}$ is the indicator function defined as

$$\mathbb{1}_{\{F(\boldsymbol{x}_j)-\mathcal{C}_k\}} = \begin{cases} 0 & F(\boldsymbol{x}_j) = \mathcal{C}_k \\ 1 & \text{otherwise,} \end{cases} \tag{4.2}$$

which penalizes wrong class assignments. In order to solve the labeling problem in (4.1) in a distributed fashion, we rely on K-means and K-medians based clustering schemes.

4.4.2 Robust Diffusion-Based Image Labeling Methodology

In this section, we present the diffusion K-medians and K-means algorithms which are used to cluster the set of feature vectors into K clusters and provide objects or regions of interest with a unique label.

4.4.2.1 Diffusion K-Medians Algorithm

The diffusion K-medians, depicted in Figure 4.2, is a hybrid classification/labeling method which contains a local clustering phase and a real time distributed classification or labeling phase [Binder et al., 2015; Binder et al., 2016]. In the local clustering phase, the camera at node $j \in \mathcal{J}$ captures a predefined number of images. Then, each node $j \in \mathcal{J}$ extracts N_n feature vectors from the regions of interest in the images and collect them in $\boldsymbol{X}_{jn}$, where n is a discrete time index. Next, each node $j \in \mathcal{J}$ exchanges $\boldsymbol{X}_{jn}$ within its neighborhood $\mathcal{B}_j$. The own and received feature vectors are stored in the matrix $\boldsymbol{S}_{jn} \triangleq \left[\boldsymbol{s}_{j1},\ldots,\boldsymbol{s}_{jN_{jn}}\right] \in \mathbb{R}^{r\times N_{jn}}$, where $N_{jn} = \sum_{b\in\mathcal{B}_j} N_n$. Then, K-medians is performed to minimize the ℓ_1-distance between $\boldsymbol{s}_{ji}$,

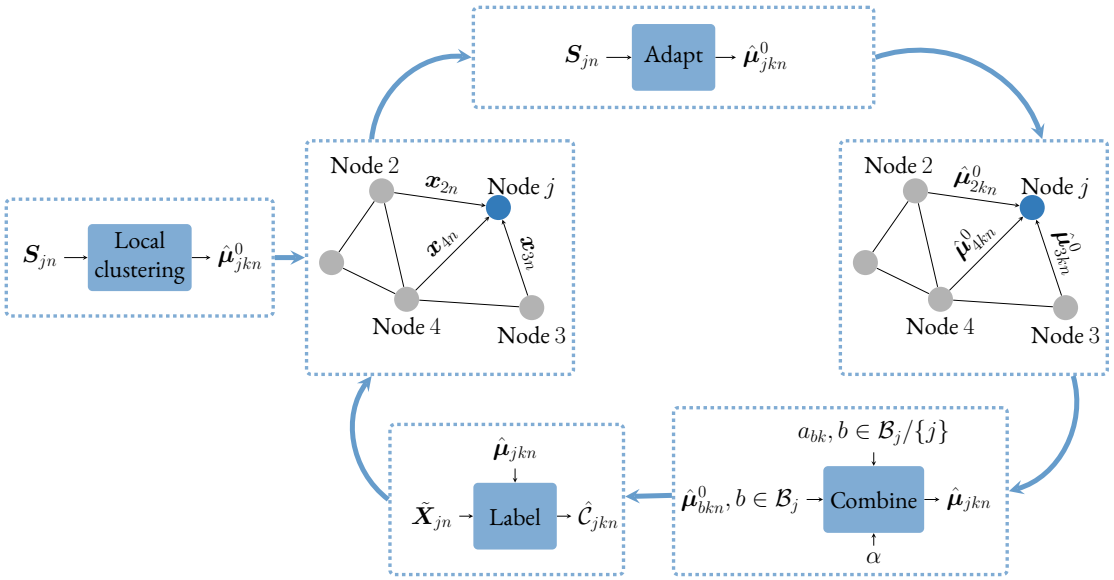


Figure 4.2: An overview of the diffusion K-medians algorithm.

$i = 1, \ldots, N_{jn}$, and the randomly initialized cluster centroids:

$$\arg\min_{\boldsymbol{\mu}^0_{jkn}} \sum_{k=1}^{K} \sum_{i=1}^{N_{jn}} \|\boldsymbol{s}_{ji} - \boldsymbol{\mu}^0_{jkn}\|_1 \tag{4.3}$$

Each feature vector is assigned to class $\hat{\mathcal{C}}^0_{jkn}$ based on the minimal ℓ_1-distance. The labeled feature vectors are stored in the matrix $\boldsymbol{V}_{jkn} \in \mathbb{R}^{r \times N_{jkn}}$, for $k = 1, \ldots, K$, where $\sum_{k=1}^{K} N_{jkn} = N_{jn}$. Next, as a robust local initial estimate of the cluster centroids, the row-wise median of $\boldsymbol{V}_{jkn}$ is computed as

$$\hat{\boldsymbol{\mu}}^0_{jkn} = \text{median}\left(\boldsymbol{V}_{jkn}\right). \tag{4.4}$$

The order of the initial centroid estimates is random at different nodes. Synchronization is therefore necessary when exchanging cluster centroid estimates and the re-sorting is performed by computing the Euclidean distance relative to the cluster centroid estimates of an arbitrary reference node in the neighborhood $\mathcal{B}_j$. This completes the initial clustering phase.

The distributed labeling phase is performed for every feature vector extracted from new images recorded by the cameras and is summarized as follows.

1. *Exchange step:* each node $j \in \mathcal{J}$ broadcasts the new feature vector to its neighbors

and accumulates the own and received feature vectors in the matrix $\tilde{\boldsymbol{X}}_{jn}$. Then, each node $j \in \mathcal{J}$ updates its data matrix $\boldsymbol{S}_{jn}$ by adding $\tilde{\boldsymbol{X}}_{jn}$ to it, such that $\boldsymbol{S}_{jn} \in \mathbb{R}^{r \times N_{jn}}$, where $N_{jn} = N_{j(n-1)} + \#\mathcal{B}_j$, where $\#\mathcal{B}_j$ denotes the size of the neighborhood of node j.

2. *Adaptation step:* at this stage, two important procedures are under taken by each node $j \in \mathcal{J}$. First, K-medians is performed to minimize the ℓ_1-distance between the feature vectors in $\boldsymbol{S}_{jn}$ and the current centroid estimates using (4.3). Then, using the new class labels, $\boldsymbol{V}_{jkn}$ is updated and preliminary cluster centroid estimates $\hat{\boldsymbol{\mu}}^0_{jkn}$ are obtained by solving (4.4).

3. *Exchange step:* each node $j \in \mathcal{J}$ broadcasts the preliminary centroid estimates $\hat{\boldsymbol{\mu}}^0_{jkn}$, for $k = 1, \ldots, K$, within the neighborhood $\mathcal{B}_j$.

4. *Combination step:* each node $j \in \mathcal{J}$ adapts its cluster centroid estimates using

$$\hat{\boldsymbol{\mu}}_{jkn} = \alpha \hat{\boldsymbol{\mu}}^0_{jkn} + (1-\alpha) \sum_{b \in \mathcal{B}_j / \{j\}} a_{bk} \hat{\boldsymbol{\mu}}^0_{bkn}, \tag{4.5}$$

where α denotes the tradeoff between the weight given to the own and the neighborhood estimates. We use uniform combination weights, which are given by

$$a_{bk} = \frac{1}{(\#\mathcal{B}_j - 1)}. \tag{4.6}$$

5. *Labeling step:* using the newly estimated cluster centroids $\hat{\boldsymbol{\mu}}_{jkn}$, each node $j \in \mathcal{J}$ assigns the new feature vectors in $\tilde{\boldsymbol{X}}_{jn}$ by computing the Euclidean distance as follows:

$$\begin{aligned} d(\boldsymbol{x}_{ji}, \hat{\boldsymbol{\mu}}_{jkn}) &= \|\boldsymbol{x}_{ji} - \hat{\boldsymbol{\mu}}_{jkn}\|_2 \\ &= \sqrt{(\boldsymbol{x}_{ji} - \hat{\boldsymbol{\mu}}_{jkn})^\top (\boldsymbol{x}_{ji} - \hat{\boldsymbol{\mu}}_{jkn})} \end{aligned} \tag{4.7}$$

where $\boldsymbol{x}_{ji}$ is the ith feature vector in $\tilde{\boldsymbol{X}}_{jn}$. Then, the new feature vectors are assigned to class $\hat{\mathcal{C}}_{jkn}, k = 1, \ldots, K$ with minimum Euclidean distance. Based on the new

cluster membership of the feature vectors, the final estimate of the cluster centroids is calculated using (4.4).

Algorithm 4.1 summarizes the diffusion K-medians algorithm.

4.4.2.2 Diffusion K-Means Algorithm

The diffusion K-means is computed analogously to the diffusion K-medians, the differences are the following: Whenever a measure of central tendency is required, the diffusion K-means uses the mean instead of the median. Further, the diffusion K-means algorithm minimizes the ℓ_2-distance between $\boldsymbol{s}_{ji}, i = 1, \ldots, N_{jn}$, and the randomly initialized centroids:

$$\arg\min_{\boldsymbol{\mu}^0_{jkn}} \sum_{k=1}^{K} \sum_{i=1}^{N_{jn}} \|\boldsymbol{s}_{ji} - \boldsymbol{\mu}^0_{jkn}\|_2 \tag{4.8}$$

The initial cluster centroids are estimated as

$$\hat{\boldsymbol{\mu}}^0_{jkn} = \text{mean}\left(\boldsymbol{V}_{jkn}\right). \tag{4.9}$$

All other steps of the diffusion K-means are identical to the ones of the diffusion K-medians. Hence, the diffusion K-means can be implemented in a similar manner as Algorithm 4.1 by replacing (4.3) and (4.4) by (4.8) and (4.9), respectively.

4.4.3 Experimental Results

In this section, a comparison of the labeling performance of the diffusion K-medians and K-means algorithm is provided. All experimental results are an average of 1000 Monte Carlo experiments and for each experiment a different random topology of the network is considered. As a benchmark, we compare our results to a centralized implementation, where all nodes forward their feature vectors to a fusion center. The fusion center computes the K-medians or K-means based labeling, having available the data of the entire camera network.

Algorithm 4.1 The diffusion K-medians algorithm

Input: K
Local clustering phase
for $j = 1, \ldots, J$ do
  Record a predefined number of images
  Extract N_n feature vectors
  Store feature vectors in $\boldsymbol{X}_{jn}$
  Broadcast $\boldsymbol{X}_{jn}$ to all nodes in $\mathcal{B}_j$
end for
for $j = 1, \ldots, J$ do
  Store the own and received feature vectors in $\boldsymbol{S}_{jn}$
  Perform K-medians according to (4.3)
  Store labeled data in $\boldsymbol{V}_{jkn}$
  Calculate $\hat{\boldsymbol{\mu}}^0_{jkn}$ via (4.4)
end for
for $j = 1, \ldots, J$ do
  Synchronize cluster centroid estimates
end for
Distributed labeling phase
for every new feature vector do
  for $j = 1, \ldots, J$ do
    Broadcast the new feature vector to all nodes in $\mathcal{B}_j$
  end for
  for $j = 1, \ldots, J$ do
    Accumulate the own and received feature vectors in $\tilde{\boldsymbol{X}}_{jn}$
    Update data matrix $\boldsymbol{S}_{jn}$ by adding $\tilde{\boldsymbol{X}}_{jn}$ to it
    Perform K-medians according to (4.3)
    Update $\boldsymbol{V}_{jkn}$ based on the new class labels
    Calculate $\hat{\boldsymbol{\mu}}^0_{jkn}$ via (4.4)
  end for
  for $j = 1, \ldots, J$ do
    Broadcast $\hat{\boldsymbol{\mu}}^0_{jkn}$ to all nodes in $\mathcal{B}_j$
  end for
  for $j = 1, \ldots, J$ do
    Calculate $\hat{\boldsymbol{\mu}}_{jkn}$ via (4.5)
    Compute distance from new feature vectors to all centroids by evaluating (4.7)
    Assign new feature vectors to class $\hat{\mathcal{C}}_{jkn}$ which minimizes (4.7)
    Re-estimate cluster centroids $\hat{\boldsymbol{\mu}}_{jkn}$ using (4.4)
  end for
end for

4.4.3.1 Network-Wide Performance Measure

The network-wide average labeling rate is used as a performance measure, which is computed as

$$\mathrm{ALR}^{\mathrm{net}} = \frac{1}{JIN_j} \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{n=1}^{N_j} \mathbb{1}_{\{\hat{C}_{jkn}^{(i)} = C_k\}}, \tag{4.10}$$

where I represents the total number of Monte Carlo experiments, N_j denotes the number of feature vectors that were available at node $j \in \mathcal{J}$ during the distributed labeling phase, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function. $\hat{C}_{jkn}^{(i)}$ represents the label that was given to the nth feature vector at the ith Monte Carlo experiment.

4.4.3.2 Feature Extraction

For the purpose of unsupervised labeling of the detected regions of interest across the network, we extract two different descriptors namely, histograms of oriented gradients (HOG) [Dalal & Triggs, 2005] and color histograms. These two types of features are used in the distributed labeling phase in order to identify common interests among the nodes. It is important to mention that for a consistent representation, all extracted regions of interest are scaled into a patch of size 24×24 pixels prior to the feature extraction process. For resizing the patches we used bicubic interpolation preceded by an anti-aliasing filter when shrinking the patches.

For the color histogram, the image patch (region of interest) is subdivided into three concentric rings and a 10-bin histogram per color channel is computed for every region in a cumulative manner (i.e., adding the previous region). The concatenation of these three histograms gives us the descriptor of each color channel. The resulting feature vector for the color is the concatenation of the three color channels resulting in a vector of dimension 90. For the extraction of the HOG descriptor, we use the MATLAB implementation with default parameters which results in a 144-dimensional feature vector. In our experiments, we consider both descriptors separately as well as their concatenation which yields a 234-dimensional feature vector. Note that, unless otherwise stated, the experimental results are generated using the concatenation of both the HOG and color histogram feature vectors.

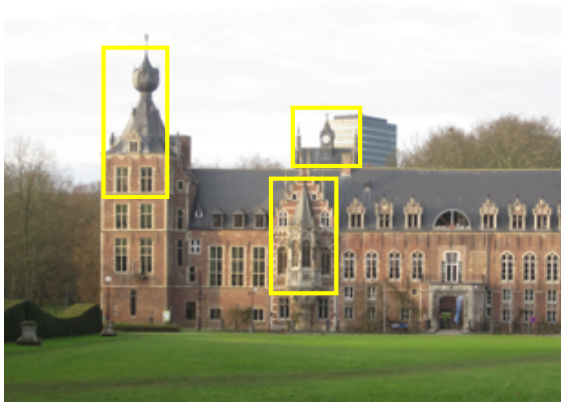


Figure 4.3: The planar scene that was used in the simulations. The yellow rectangles contain the three different regions of interest.

Figure 4.4: Identical region of interest for two different nodes.

4.4.3.3 Real Data Results

As depicted in Figure 4.3, we have considered a planar scene containing $K = 3$ different regions of interest. From this scene, we have randomly generated a total of $J = 20$ different views (each one corresponding to a different node) which are related by affine transformations. That is, if we consider our original scene as a 2-dimensional function $g : \mathbb{R}^{2\times 1} \mapsto \mathbb{R}$ we generate the different views according to

$$g_j(\boldsymbol{x}) = g(\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{t}_j), \tag{4.11}$$

for each node $j \in \mathcal{J}$, where $\boldsymbol{A}_j \in \mathbb{R}^{2\times 2}$ is non-singular and $\boldsymbol{t}_j \in \mathbb{R}^{2\times 1}$ is some translation vector. It is important to realize that, due to the affine warping, the extracted patches between nodes will be different, even when they are interested in the same region of the object. This effect is illustrated in Figure 4.4, where two nodes are interested in the clock of the building,

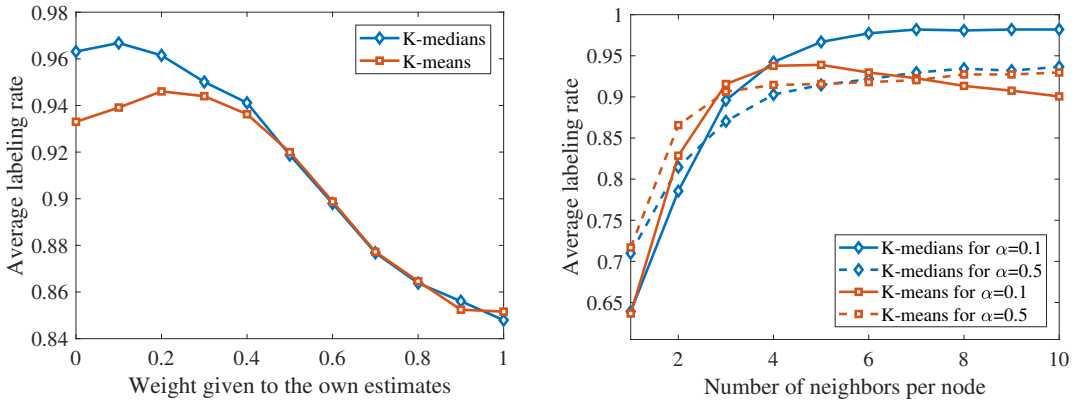
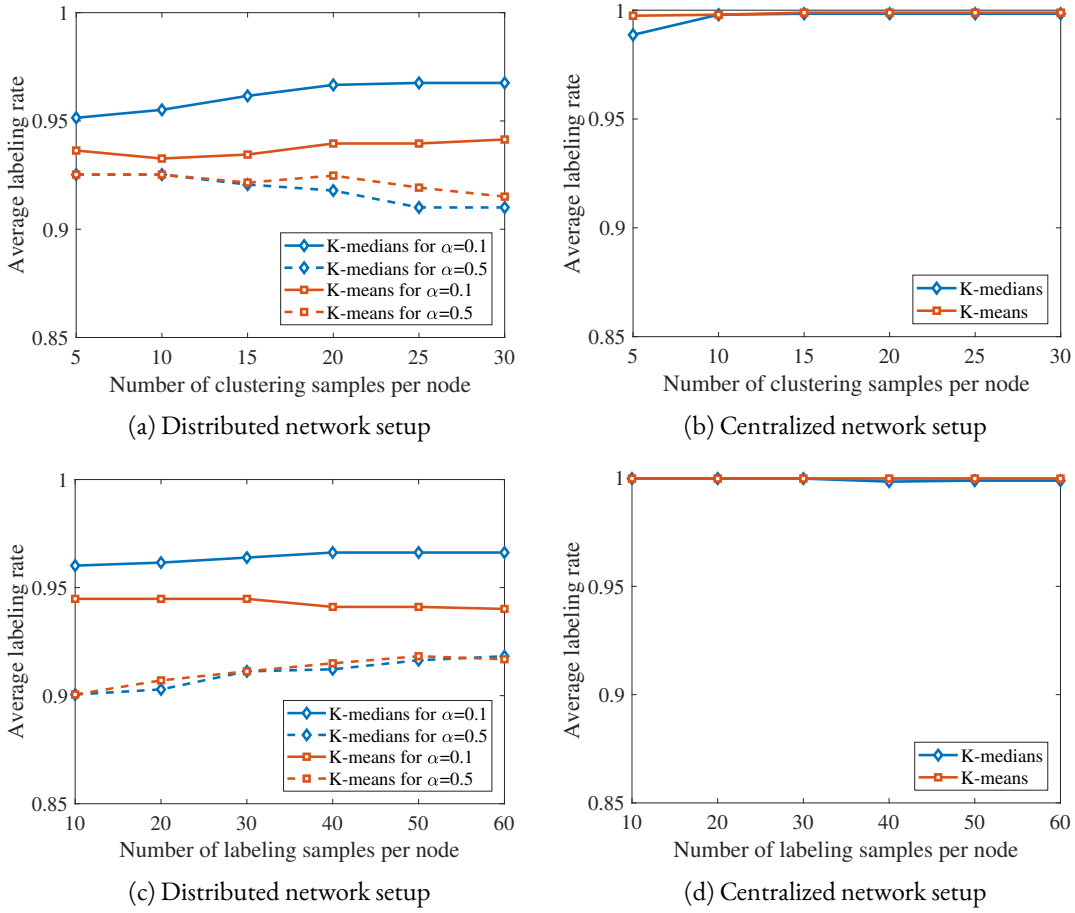


Figure 4.5: Average labeling rate as a function of the weight given to the own estimates (left) and the number of neighbors per node (right).

however the regions that contain this object are different, since they observe the scene from different viewpoints. This means that, even in the noiseless case, feature vectors corresponding to the same region or object of interest will be different for different nodes. For this reason, to some extent, the chosen feature vectors should provide a representation that is robust against scene transformations (affine in this case).

Noise samples drawn from a zero mean Gaussian distribution are added to the feature vectors to make a first step at simulating real time streaming data. Each node $j \in \mathcal{J}$ has a total of 80 feature vectors and the first 20 feature vectors are used for the local clustering phase, unless mentioned otherwise. The remaining 60 feature vectors are used for real time labeling. The neighborhood size is set to $\#\mathcal{B}_j = 5$ and $\alpha = 0.1$.

Figure 4.5a displays the average labeling rate as a function of the weight given to the own estimates (α) by each node for the diffusion K-medians and K-means algorithms. Clearly, cooperation improves the results, and best performance is achieved when a high weight is given to the neighborhoods estimates $(\alpha = 0.1)$. The average labeling rate versus the number of neighbors per node is shown in Figure 4.5b. In general, an increase of the neighborhood size leads to a higher labeling accuracy, at the cost of an increase in node communication. Figure 4.6a depicts the average labeling rate as a function of the number of clustering samples per node for a distributed network setup. This experiment shows that only a small number of clustering samples are required, which makes the method suitable for real time applications. The



(a) Distributed network setup

(b) Centralized network setup

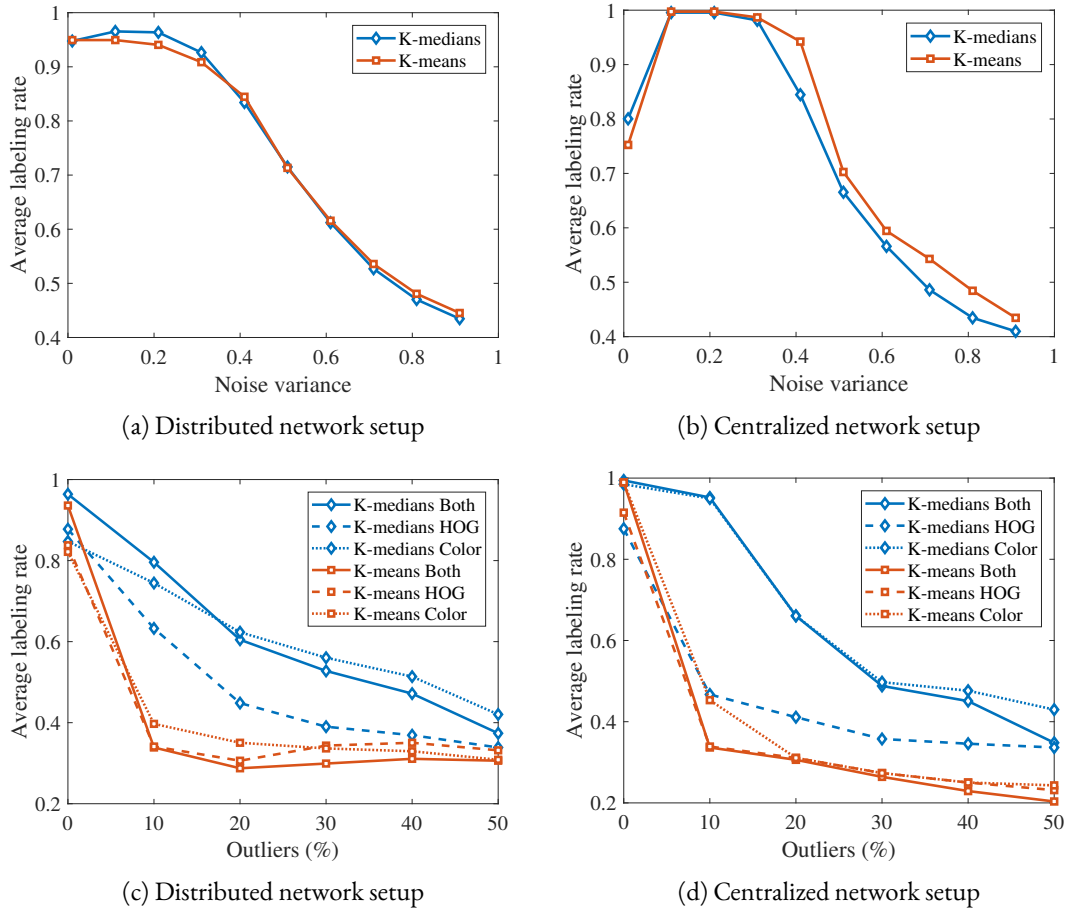
(c) Distributed network setup

(d) Centralized network setup

Figure 4.6: Average labeling rate as a function of the number of clustering and labeling samples for the distributed (left) and the centralized (right) network setups.

diffusion K-medians algorithm outperforms the diffusion K-means algorithm specially when $\alpha = 0.1$. As displayed in Figure 4.6c, the average labeling rate of the diffusion K-Medians is higher than that of the K-Means and it achieves a good performance even for a small number of available labeling samples. Again, a smaller value of α produces better labeling rates for both algorithms. Comparing Figure 4.6a and Figure 4.6c with Figure 4.6b and Figure 4.6d, respectively, one notices that the centralized network setup outperforms its distributed counterpart. However, this improvement in performance comes at the cost of having a single point of failure.

The average labeling rate as a function of the noise variance for the distributed and centralized network setups are shown in Figure 4.7a and Figure 4.7b, respectively. Note that the



(a) Distributed network setup

(b) Centralized network setup

(c) Distributed network setup

(d) Centralized network setup

Figure 4.7: Average labeling rate as a function of noise variance and percentage of outliers for the distributed (left) and the centralized (right) network setups.

noise is added to the feature vectors whose elements take values between zero and one. As expected, for a variance bigger than 0.1, there is a gradual decrease in performance. The lower performance for a variance equal to zero can be contributed to the fact that the algorithms are designed for real time streaming data and not for a static picture. The performance loss is less pronounced for the distributed implementations. Figure 4.7c and Figure 4.7d depict the average labeling rate as a function of the percentage of outliers for distributed and centralized network setups, respectively. For this experiment, additive outliers at random positions were generated from a zero mean Gaussian distribution with a variance that is five times that of the nominal noise. The expressions "K-medians Both" and "K-means Both" in the figures represent the usage of HOG and color features. "HOG" and "color", on the other hand, refer to the

usage of a single feature. The figures clearly show the superiority of the diffusion K-medians over the diffusion K-means in the presence of outliers. Furthermore, using both feature vectors is beneficial, especially for low amounts of outlier contamination. In general, the color outperforms the HOG descriptor, and the distributed solutions approach the performance of the centralized ones.

4.5 Object Labeling in a Non-Stationary Scene

In this section, we study object labeling when the scene observed by a distributed camera network is time-varying. To this end, we develop a distributed and adaptive multi-object labeling algorithm for a multi-camera network without assuming any form of camera calibration or utilizing a centralized computing unit that fuses all information collected from different cameras [Teklehaymanot et al., 2017].

4.5.1 Problem Formulation

Consider a wireless camera network with J nodes distributed over some geographic region as the one shown in Figure 4.8. The set of nodes that communicates directly with node $j \in \mathcal{J} \triangleq \{1, \ldots, J\}$ is called the neighborhood of node j and is denoted by $\mathcal{B}_j \subseteq \mathcal{J}$. Let $\boldsymbol{X}_{jn} \in \mathbb{R}^{r \times m_{jn}}$ represent the r-dimensional feature vectors extracted at the jth node from the m_{jn} objects that are observed by the camera of node j at time instant n. Each feature vector belongs to a certain cluster $\mathcal{C}_k, k \in \{1, \ldots, K_n\}$, where k is the cluster label. The total number of objects (clusters) K_n at time instant n is assumed to be known or estimated via the cluster enumeration method discussed in Section 2.6.1, Section 2.7, or Section 3.5.1. Due to the different viewpoints of the cameras, even at the same time instant, the number of objects observed by different cameras differs. Our research goal is to adaptively estimate cluster centroids and enable cameras with different viewpoints to assign the same identity to the same object in the scene of interest.

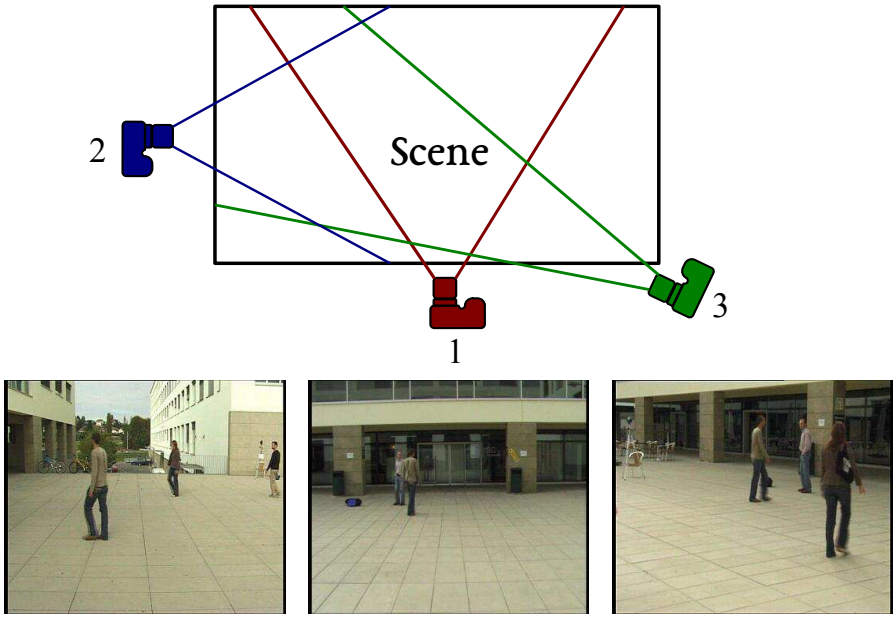


Figure 4.8: A wireless camera network [Berclaz et al., 2011; Fleuret et al., 2008] observing a scene of interest. The top image shows a camera network with $J = 3$ nodes continuously monitoring a scene of interest from different observation angles. The bottom images show frames captured at the same time instant by cameras 2, 1, and 3, respectively.

4.5.2 Adaptive Diffusion-Based Track Assisted Multi-Object Labeling Algorithm

We propose a distributed and adaptive track assisted multi-object labeling algorithm for multi-camera networks, which is based on the adapt then combine (ATC) diffusion principle [Sayed, 2014a]. An overview of the algorithm is shown in Figure 4.9. The general procedure involved in the proposed framework is summarized as follows.

1. *Record:* the camera at node $j \in \mathcal{J}$ captures a frame from the scene of interest at time instant n.

2. *Detect and extract:* if there are objects of interest in the frame, then each node j extracts feature vectors from the detected bounding boxes of the objects and stores them in $\boldsymbol{X}_{jn}$. Otherwise, the camera at node $j \in \mathcal{J}$ continues to record at time instant $n + 1$.

3. *Exchange features:* each node $j \in \mathcal{J}$ exchanges its feature vectors $\boldsymbol{X}_{jn}$ within its neighborhood $\mathcal{B}_j$. The own and received feature vectors are stored in the matrix $\tilde{\boldsymbol{X}}_{jn} \in$

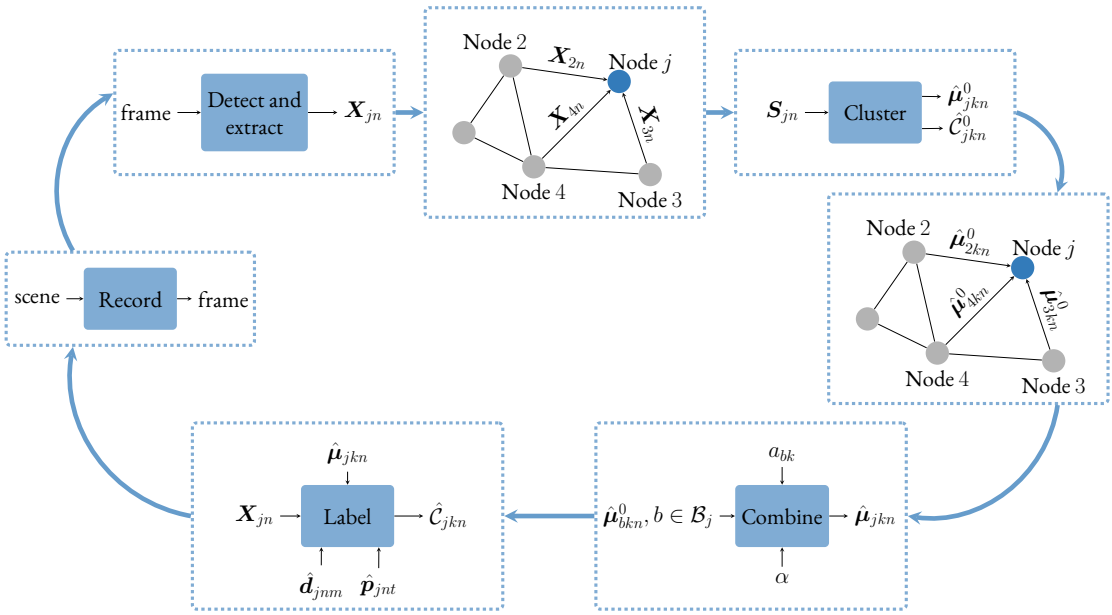


Figure 4.9: An overview of the distributed and adaptive diffusion-based track assisted multi-object labeling algorithm.

$\mathbb{R}^{r \times \sum_{b \in \mathcal{B}_j} m_{bn}}$, where m_{bn} represents the number of objects detected by node $b \in \mathcal{B}_j$ at time instant n. Next, each node j accumulates $\tilde{\boldsymbol{X}}_{ji}, i = 1, \ldots, n$, inside the matrix $\boldsymbol{S}_{jn} \in \mathbb{R}^{r \times N_{jn}}$, where $N_{jn} = N_{j(n-1)} + \sum_{b \in \mathcal{B}_j} m_{bn}$ is the total number of feature vectors at node j at time instant n.

4. *Cluster:* each node $j \in \mathcal{J}$ performs K-means++ [Arthur & Vassilvitskii, 2007] to minimize the ℓ_2-distance between the feature vectors in $\boldsymbol{S}_{jn}$ and the initial cluster centroids $\boldsymbol{\mu}^0_{jkn} \in \mathbb{R}^{r \times 1}$

$$\arg\min_{\boldsymbol{\mu}^0_{jkn}} \sum_{k=1}^{K_n} \sum_{i=1}^{N_{jn}} \|\boldsymbol{s}_{ji} - \boldsymbol{\mu}^0_{jkn}\|_2, \tag{4.12}$$

where $\boldsymbol{s}_{ji}$ denotes the ith column of $\boldsymbol{S}_{jn}$. This results in a unique cluster label $\hat{\mathcal{C}}^0_{jkn}$ for each feature vector in $\boldsymbol{S}_{jn}$. The feature vectors that belong to the same cluster $k \in \{1, \ldots, K_n\}$ are saved in $\boldsymbol{V}_{jkn} \in \mathbb{R}^{r \times N_{jkn}}$, where $\sum_{k=1}^{K_n} N_{jkn} = N_{jn}$. Then, the row-wise mean of $\boldsymbol{V}_{jkn}$ is computed as

$$\hat{\boldsymbol{\mu}}^0_{jkn} = \text{mean}\left(\boldsymbol{V}_{jkn}\right). \tag{4.13}$$

The minimization of the ℓ_2-distance using $\boldsymbol{\mu}^0_{jkn}$ in (4.12) is performed only if $K_n > K_{n-1}$. Otherwise, $\boldsymbol{\mu}^0_{jkn}$ is replaced with $\hat{\boldsymbol{\mu}}_{jk(n-1)}$ in (4.12).

5. *Exchange estimates:* each node $j \in \mathcal{J}$ exchanges its intermediate centroid estimates $\hat{\boldsymbol{\mu}}^0_{jkn}$ within its neighborhood $\mathcal{B}_j$. Synchronization of $\hat{\boldsymbol{\mu}}^0_{bkn}, b \in \mathcal{B}_j$, is necessary because the order of $\hat{\boldsymbol{\mu}}^0_{jkn}$ is random at different nodes. The re-ordering of the intermediate centroid estimates is performed by computing the Euclidean distance relative to an arbitrarily chosen neighborhood head in $\mathcal{B}_j$.

6. *Combine estimates:* each node $j \in \mathcal{J}$ adapts its centroid estimates using

$$\hat{\boldsymbol{\mu}}_{jkn} = \alpha \hat{\boldsymbol{\mu}}^0_{jkn} + (1-\alpha) \sum_{b \in \mathcal{B}_j \setminus \{j\}} a_{bk} \hat{\boldsymbol{\mu}}^0_{bkn}, \tag{4.14}$$

where α controls the tradeoff between the weight given to the own and neighborhood estimates. Here, uniform combination weights, given by (4.6), are used.

7. *Assign:* at this step, each node $j \in \mathcal{J}$ assigns unique labels $\hat{\mathcal{C}}_{jkn}$ to objects of interest in the current frame. We propose a regularized cost function that aggregates the information obtained from a local Kalman filter-based tracker and a diffusion-based labeling algorithm. In particular,

$$\boldsymbol{Z}_{jn}(t,m) = \lambda \|\hat{\boldsymbol{d}}_{jnm} - \hat{\boldsymbol{p}}_{jnt}\|_2 + (1-\lambda)\|\boldsymbol{x}_{jm} - \hat{\boldsymbol{\mu}}_{jtn}\|_2, \tag{4.15}$$

where λ is the regularization parameter, $\hat{\boldsymbol{d}}_{jnm}$ and $\hat{\boldsymbol{p}}_{jnt}$ are detected and predicted bounding box center positions for $m = 1, \ldots, m_{jn}$ and $t = 1, \ldots, t_{jn}$, respectively, and $\boldsymbol{x}_{jm}$ is the mth feature vector in $\boldsymbol{X}_{jn}$. The total number of open tracks in the jth node at time instant n is denoted by t_{jn} and $\hat{\boldsymbol{\mu}}_{jtn}$ represents the cluster centroid that belongs to the tth track. The two ℓ_2-distances in (4.15) are normalized by their respective maximum to make sure that they are comparable. Then, the Hungarian algorithm [Munkres, 1957] is applied on $\boldsymbol{Z}_{jn}$ to assign unique labels $\hat{\mathcal{C}}_{jkn}$ to feature vectors in $\boldsymbol{X}_{jn}$.

Algorithm 4.2 summarizes the adaptive diffusion-based track assisted multi-object labeling algorithm.

Algorithm 4.2 Distributed and adaptive diffusion-based track assisted multi-object labeling algorithm

Input: K
for $n = 1, 2, \ldots$ do
  for $j = 1, 2, \ldots, J$ do
    Record frame
    if objects are detected then
      Extract feature vectors and store them in $\boldsymbol{X}_{jn}$
    else
      Proceed with record step at $n + 1$
    end if
  end for
  for $j = 1, 2, \ldots, J$ do
    Exchange $\boldsymbol{X}_{jn}$ within $\mathcal{B}_j$
    Store own and received feature vectors in $\tilde{\boldsymbol{X}}_{jn}$
    Accumulate $\tilde{\boldsymbol{X}}_{ji}, i = 1, \ldots, n$, in $\boldsymbol{S}_{jn}$
  end for
  for $j = 1, 2, \ldots, J$ do
    Perform K-means++ according to (4.12)
    Calculate $\hat{\boldsymbol{\mu}}^0_{jkn}$ via (4.13)
  end for
  for $j = 1, 2, \ldots, J$ do
    Exchange $\hat{\boldsymbol{\mu}}^0_{jkn}$ within $\mathcal{B}_j$
  end for
  for $j = 1, 2, \ldots, J$ do
    Synchronize $\hat{\boldsymbol{\mu}}^0_{bkn}, b \in \mathcal{B}_j$
    Combine $\hat{\boldsymbol{\mu}}^0_{bkn}, b \in \mathcal{B}_j$, via (4.14)
    Solve (4.15) using the Hungarian algorithm [Munkres, 1957]
    Assign unique labels $\hat{\mathcal{C}}_{jkn}$ to feature vectors in $\boldsymbol{X}_{jn}$
  end for
end for

4.5.3 Experimental Results

In this section, we first describe the network-wide performance measures used to evaluate the labeling performance of Algorithm 4.2. Then, real data results of the diffusion-based track assisted multi-object labeling algorithm are provided.

4.5.3.1 Network-Wide Performance Measures

We define two network-wide performance measures, i.e., the average labeling rate (ALR^{net}) and the average mislabeling rate (AMR^{net}) as follows:

$$\text{ALR}^{\text{net}} = \frac{1}{JN_j} \sum_{j=1}^{J} \sum_{n=1}^{N_j} \mathbb{1}_{\{\hat{\mathcal{C}}_{jkn} = \mathcal{C}_k\}} \tag{4.16}$$

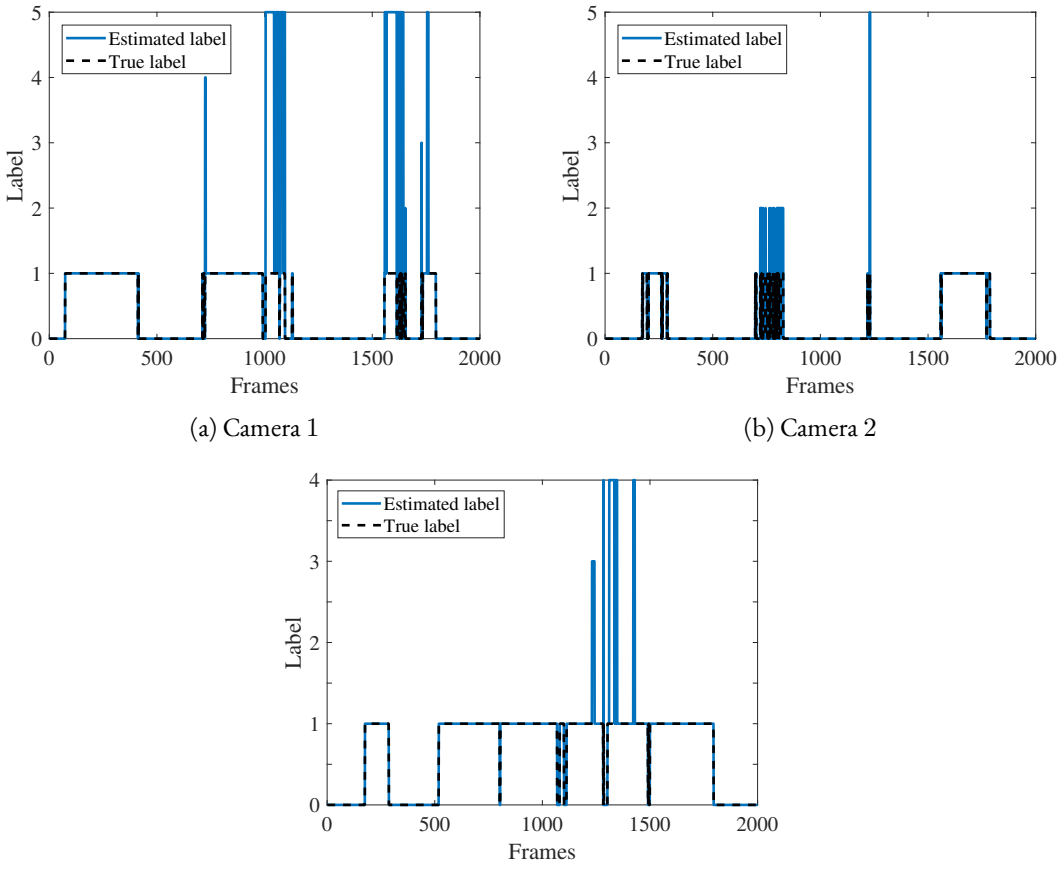
$$\text{AMR}^{\text{net}} = \frac{1}{JN_j} \sum_{j=1}^{J} \sum_{n=1}^{N_j} \mathbb{1}_{\{\hat{\mathcal{C}}_{jkn} \neq \mathcal{C}_k\}}, \tag{4.17}$$

where $\mathcal{C}_k$ is the set of ground truth labels, N_j is the total number of detected objects over the span of the observed video, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function. ALR^{net} indicates if object k is provided with the correct label k and AMR^{net} indicates if object k is provided with a wrong label h, where $h \neq k$. To evaluate the performance of the proposed algorithm, ALR^{net} and AMR^{net} are placed in the diagonal and off-diagonal, respectively, of a confusion matrix. In the confusion matrix, both ALR^{net} and AMR^{net} are given in percentage.

4.5.3.2 Real Data Results

Here, we use the whole video sequence that was described in Section 2.8.3.3. The multi-camera video sequence contains $J = 3$ stationary cameras and a neighborhood size of $\#\mathcal{B}_j = 3$ is considered. Each video sequence contains 2000 frames and up to five people are seen entering and exiting the scene of interest at different times. The multi-camera video sequence is challenging in the sense that the videos have low resolution, the cameras monitor pedestrians from different angles, and there are frequent pedestrian occlusions.

To detect pedestrians in the scene of interest, we use an already trained MATLAB implementation of the aggregate channel features (ACF) pedestrian detector [Dollár et al., 2014]. The ACF pedestrian detector uses boosting to train decision trees over features and a multi-scale sliding window approach to distinguish objects of interest from the background. Two color features, whose concatenation results in a 102-dimensional feature vector, are extracted for each detected object using the method described in Section 2.8.3.3. The weight parameter is set to $\alpha = 0.5$ and the regularization parameter is $\lambda = 0.4$. The number of pedestrians seen until the nth time instant, K_n, is assumed to be known.



(a) Camera 1

(b) Camera 2

(c) Camera 3

Figure 4.10: A comparison of the estimated and true labels for pedestrian 1 in the video sequence captured by three cameras.

The labeling performance of cameras 1, 2, and 3 for pedestrian 1 is depicted in Figure 4.10a, Figure 4.10b, and Figure 4.10c, respectively. A zero label indicates that either the pedestrian is not detected in the current frame or he/she is no longer in the scene of interest. For these multi-camera video sequences, the ACF pedestrian detector has a high misdetection rate and the position of the bounding boxes is unstable. In some frames, multiple bounding boxes are detected for a single pedestrian in the scene, see for example the second raw in Figure 4.11 for camera 3. These problems affect the performance of the Kalman filter-based tracker and the diffusion-based labeling algorithm. The blue spikes in Figure 4.10 indicate mislabels which are partly due to identity (label) switches between two pedestrians and multiple detections for a single pedestrian. However, even under such conditions, the proposed diffusion-based

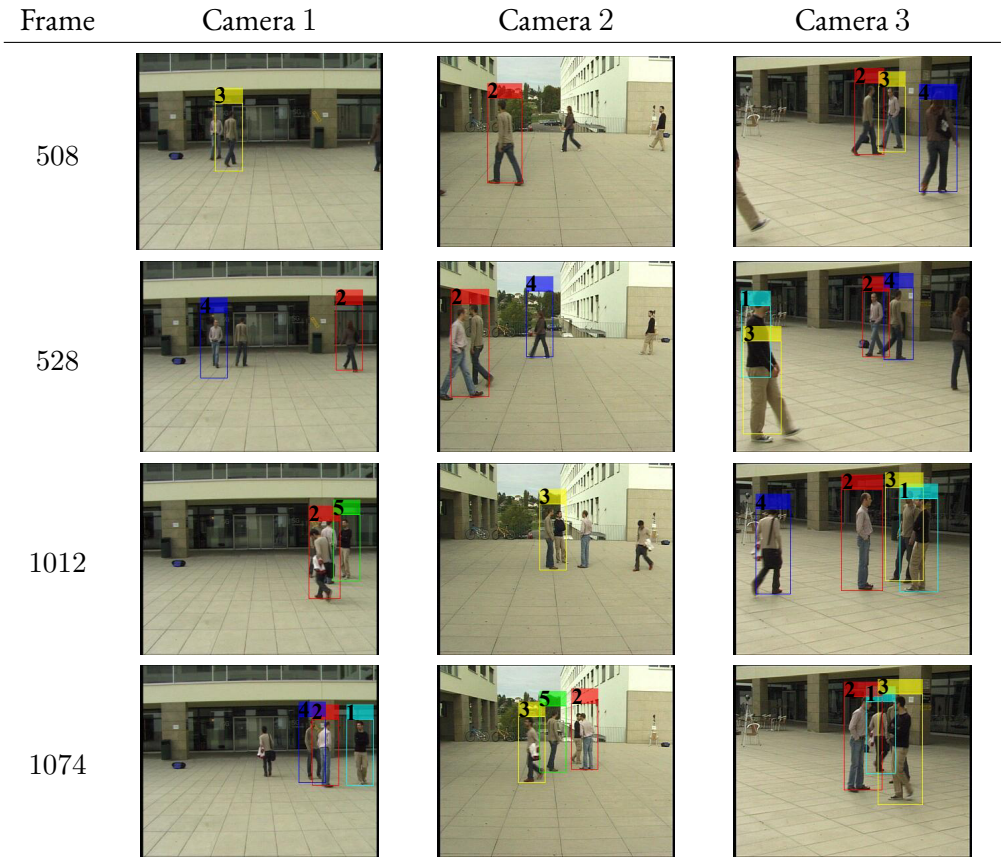


Figure 4.11: An example of the network-wide mislabeling results of the proposed algorithm using ACF pedestrian detector. Each row displays frames captured by different cameras at the same time instant. The color of each bounding box and the number displayed on it show the identity that is given to the particular pedestrian.

multi-object labeling algorithm performs reasonably well.

Figure 4.12 shows an example of the network-wide labeling results of the proposed algorithm using ACF pedestrian detector. The bounding box of each pedestrian is provided with a unique color and number. We define the multi-object labeling algorithm to be performing well if the same pedestrian is provided with the same color of bounding box and number across different camera views and time frames. The proposed algorithm is able to provide unique and consistent labels to pedestrians in the scene even when there are partial occlusions. On the contrary, Figure 4.11 depicts the frames where the proposed multi-object labeling algorithm fails to label the pedestrians correctly.

Table 4.1 shows the confusion matrix of the multi-object labeling algorithm averaged over

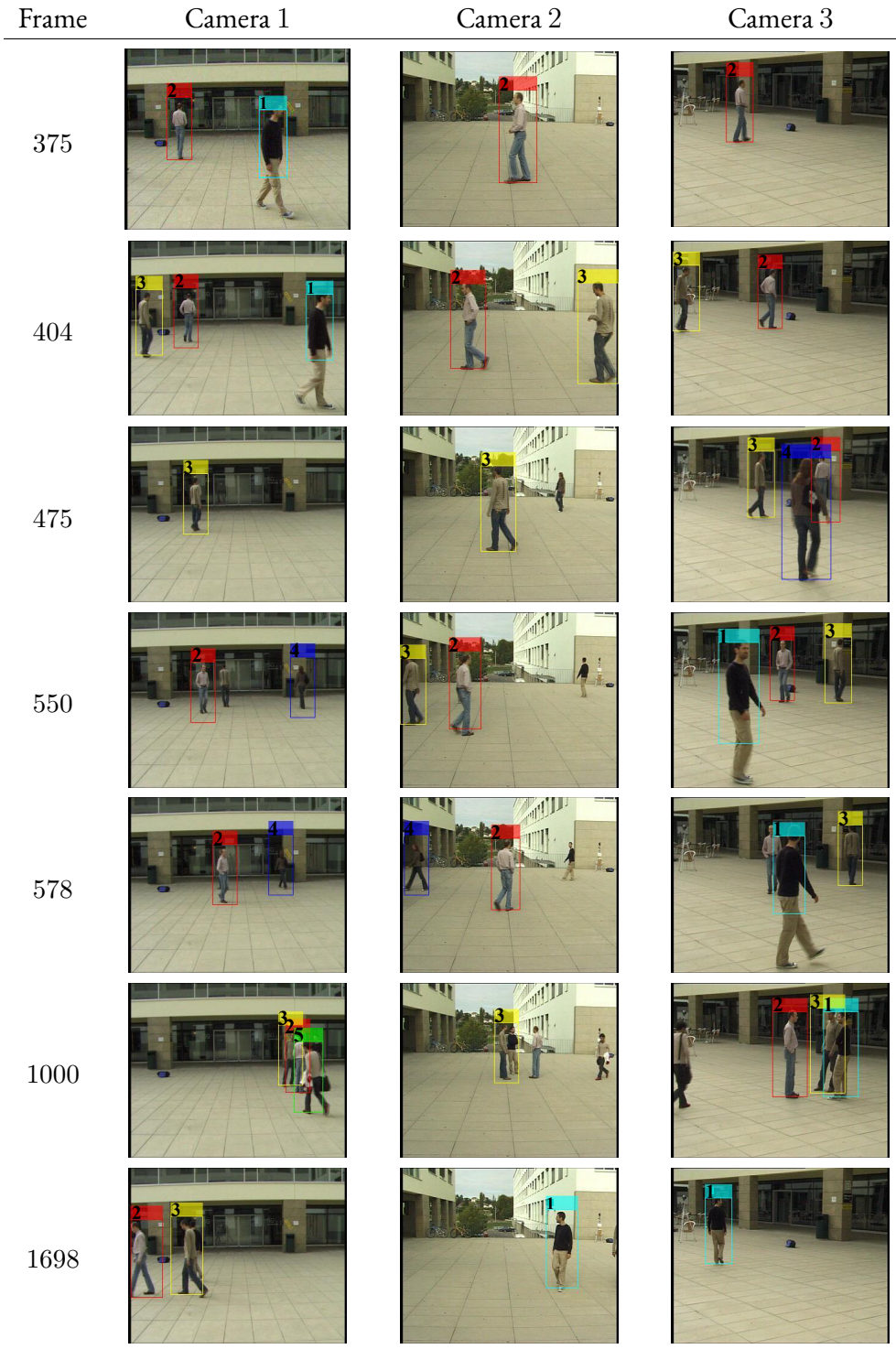


Figure 4.12: An example of the network-wide correct labeling results of the proposed algorithm using ACF pedestrian detector. Each row displays frames captured by different cameras at the same time instant. The color of each bounding box and the number displayed on it show the identity that is given to the particular pedestrian.

Table 4.1: Confusion matrix in percentage with ALR^{net} in the diagonal and AMR^{net} in the off-diagonal using ACF pedestrian detector.

		Estimated Labels				
		1	2	3	4	5
True Labels	1	**88.75**	3.56	0.93	0.96	5.80
	2	0	**95.12**	1.56	0.07	3.18
	3	0.79	3.36	**83.66**	4.57	8.12
	4	0	3.89	5.03	**91.08**	0
	5	4.66	8.65	48.59	18.33	**19.77**

Table 4.2: Confusion matrix in percentage with ALR^{net} in the diagonal and AMR^{net} in the off-diagonal using ground truth pedestrian detections.

		Estimated Labels				
		1	2	3	4	5
True Labels	1	**96.25**	0.79	1.31	0.10	1.55
	2	0.54	**97.26**	1.11	0.85	0.24
	3	1.24	1.80	**89.44**	1.48	6.05
	4	0	0.21	1.88	**97.92**	0
	5	0	0	0.29	1.11	**98.60**

all cameras and all time frames. The confusion matrix shows the network-wide average labeling rate in the diagonal and the network-wide average mislabeling rate in the off-diagonal as defined in Section 4.5.3.1. The proposed algorithm performs well in providing unique and consistent labels to the first four pedestrians in the scene. Lower labeling performance is exhibited for the fifth pedestrian because he is only visible for a short time and is not well detected by the ACF pedestrian detector, see Figure 4.11.

Table 4.2 shows the confusion matrix of the proposed algorithm when we replace the ACF pedestrian detector with the ground truth pedestrian detections which provides the best achievable performance of our algorithm. If a good pedestrian detector is used, the multi-object labeling algorithm achieves a very high labeling performance even when the color features have strong similarities, which is the case when pedestrians are dressed similarly.

4.6 Summary

In this chapter, we discussed object labeling for wireless camera networks in the presence of static and time-varying scenes of interest and presented two main contributions. First, we developed a robust distributed labeling strategy in the context of camera networks where the cameras are interested in a static scene. The robust distributed labeling method achieved high labeling rates and the loss compared to a centralized solution is small. Then, we studied multi-object labeling in multi-camera networks whose cameras monitor a time-varying scene. To this end, we developed a distributed and adaptive diffusion-based track assisted multi-object labeling algorithm. The proposed algorithm is able to provide unique and consistent labels to multiple objects across camera views and time frames requiring neither camera view registration nor a fusion center. The performance of the proposed algorithm was tested on a real multi-camera network use case. In both the static and time-varying scenes, good labeling performance was achieved given that the number of objects in the scene is known a priori.

However, specially when the scene of interest is time-varying, assuming that the number of objects, which could also be time-varying, is known a priori is impractical in real-world applications. In the next chapter, we explore multi-object labeling by automatically estimating the number of objects in the scene.

PART III

Cluster Enumeration and Labeling

5

Joint Cluster Enumeration and Labeling

5.1 Introduction

Joint cluster enumeration and labeling refers to a broad spectrum of methods that automatically estimate the number of clusters in a given data set and, subsequently, provide unique labels to individual clusters. Cluster enumeration and labeling methods are unsupervised in the sense that parameters of interest are learned from the data without requiring training data with known class labels.

In this chapter, we present a joint cluster enumeration and labeling algorithm that is able to determine the intrinsic structure of clustered data when no information other than the observed values is available. The state-of-the-art on cluster enumeration and labeling is discussed in Section 5.2 and a summary of the main contributions in this chapter is provided in Section 5.3. Section 5.4 formulates the cluster enumeration and labeling problem and Section 5.5 details a new joint cluster enumeration and labeling algorithm. Experimental results, including a real data application of person labeling using radar measurements of the human gait, are discussed in Section 5.6. Finally, the chapter is summarized in Section 5.7.

5.2 State-of-the-art

Over the past four decades, abundance of algorithms for the simultaneous determination of the number of clusters and cluster memberships have been developed. The vast interest in the area partly emanates from the lack of a unique definition for a cluster [Kaufman & Rousseeuw, 1990]. On top of this, a variety of applications, such as characterizing customer groups based on purchasing patterns, categorizing web documents, grouping genes and proteins that have similar functionality, and so on [Karypis et al., 1999], have attracted researchers to the area of cluster enumeration and labeling.

Most cluster enumeration and labeling strategies are composed of two separate methods, namely a cluster enumeration method and a clustering algorithm. State-of-the-art clustering algorithms can be roughly divided into methods that follow hierarchical strategy [King, 1967; Karypis et al., 1999; Boley, 1998; Zhao & Karypis, 2005; Murtagh, 1983; Everitt, 2011; Sibson, 1973; Defays, 1977] and those that are based on partitioning (or relocation) of data points into clusters [Lloyd, 1982; Hartigan & Wong, 1979; Dempster et al., 1977; Kaufman & Rousseeuw, 1987; Zadeh, 1965; Bezdek, 1981; Kaufman & Rousseeuw, 1990]. Neither hierarchical nor partitioning-based methods directly address the issue of determining the number of groups (or clusters) in a given data set. As a result, the estimation of the number of clusters in a given data set has been widely researched, see Section 2.2 and Section 3.2 for a review of the state-of-the-art on cluster enumeration.

5.3 Contributions in this Chapter

In this chapter, we propose a joint cluster enumeration and labeling algorithm that automatically estimates the number of clusters in a given data set and labels individual clusters at the same time by incorporating the cluster enumeration criteria derived in Section 2.6.1, Section 2.7, and Section 3.5.1. The proposed algorithm is unsupervised since it requires neither training data nor prior knowledge of the number of clusters. Further more, we apply the proposed method to a real data application of person labeling using radar measurements of the human gait. In this context, person (or target) enumeration and labeling is achieved by exploiting the fact that feature vectors extracted from the gait of the same target create a cluster in feature space. Using radar data of normal human walk, we are able to estimate the cor-

rect number of targets and label them with a high accuracy despite short observation times. To the best of our knowledge, this is the first work towards utilizing unsupervised learning methods to jointly estimate the number of targets and to label them using radar-based gait measurements.

The main contributions in this chapter have been published in [Teklehaymanot et al., 2018f].

5.4 Problem Formulation

Given a set of r-dimensional vectors $\mathcal{X} \triangleq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, let $\{\mathcal{X}_1, \ldots, \mathcal{X}_K\}$ be a partition of $\mathcal{X}$ into K clusters, such that $\mathcal{X}_k \subseteq \mathcal{X}$, for $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$. The clusters $\mathcal{X}_k$, for $k \in \mathcal{K}$, are independent, mutually exclusive, and non-empty. Assume that a family of candidate models $\mathcal{M} \triangleq \{M_{L_{\min}}, \ldots, M_{L_{\max}}\}$ is given, where $L_{\min}$ and $L_{\max}$ are the specified minimum and maximum number of clusters, respectively. Each candidate model $M_l \in \mathcal{M}$, for $l = L_{\min}, \ldots, L_{\max}$ and $l \in \mathbb{Z}^+$, represents a partition of $\mathcal{X}$ into l clusters with associated parameters $\boldsymbol{\Theta}_l = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_l] \in \mathbb{R}^{q \times l}$. Our research goal is to estimate the number of clusters in $\mathcal{X}$ given that assumption (A-2.1) is fulfilled. Once the number of clusters is estimated, we provide unique labels to the clusters. This way, we are able to accomplish joint cluster enumeration and labeling in an unsupervised learning framework.

5.5 Joint Cluster Enumeration and Labeling Algorithm

The joint cluster enumeration and labeling algorithm estimates the number of data clusters and, subsequently, provides individual clusters with unique labels [Teklehaymanot et al., 2018f]. We use the two-step cluster enumeration algorithm presented in Section 2.6 to estimate the number of clusters in $\mathcal{X}$. Since the two-step cluster enumeration algorithm produces a cluster number estimate, $\hat{K}$, as well as an estimate of cluster parameters, we can provide labels to cluster centroid estimates $\hat{\boldsymbol{\mu}}_m$, for $m = 1, \ldots, \hat{K}$. Hence, the data vectors that are associated with a specific centroid receive the label given to that centroid. This way, we are able to estimate the number of clusters and, at the same time, provide unique labels to clusters. The proposed joint cluster enumeration and labeling method, which uses either BIC_{N} or BIC_{NF} for cluster enumeration, is summarized in Algorithm 5.1.

Algorithm 5.1 Joint cluster enumeration and labeling algorithm

Inputs: $\mathcal{X}$, $L_{\min}$, and $L_{\max}$
Cluster enumeration
for $l = L_{\min}, \ldots, L_{\max}$ do
  for $m = 1, \ldots, l$ do
    Estimate $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ using the EM algorithm
    Calculate N_m via hard clustering, see Algorithm 2.1
  end for
  Calculate either $\mathrm{BIC}_{\mathrm{N}}(M_l)$ or $\mathrm{BIC}_{\mathrm{NF}}(M_l)$ via (2.19) or (2.50), respectively
end for
Estimate the number of clusters in $\mathcal{X}$ using either (2.20) or (2.51)
Cluster labeling
for $m = 1, \ldots, \hat{K}$ do
  Assign unique labels to the data vectors that belong to $\hat{\boldsymbol{\mu}}_m$
end for

In cases where robustness against heavy tailed noise and outliers is required, one can replace the Bayesian cluster enumeration criteria which are based on the Gaussian distribution, namely $\mathrm{BIC}_{\mathrm{N}}$ and $\mathrm{BIC}_{\mathrm{NF}}$, by the robust cluster enumeration criteria derived in Section 3.5.1. Hence, Algorithm 5.1 can be robustified by calculating the BIC using either (3.1) or (3.5) and estimating the number of clusters via (3.4) or (3.6), respectively.

5.6 Experimental Results

In this section, we perform numerical and real data experiments to compare the performance of the proposed joint cluster enumeration and labeling algorithm with the X-means algorithm [Pelleg & Moore, 2000] and a robust implementation of the original BIC (BIC_{ot_ν}). In the experiments, we show the performance of the cluster enumeration criteria that were derived in the dissertation, namely $\mathrm{BIC}_{\mathrm{N}}$, $\mathrm{BIC}_{\mathrm{NS}}$, $\mathrm{BIC}_{\mathrm{NF}}$, and BIC_{t_ν} which are given by (2.19), (2.39), (2.47), and (3.1), respectively. Note that $\mathrm{BIC}_{\mathrm{N}}$, $\mathrm{BIC}_{\mathrm{NF}}$, BIC_{t_ν}, and BIC_{ot_ν} are implemented as wrappers around the EM algorithm, while $\mathrm{BIC}_{\mathrm{NS}}$ and X-means are implemented as wrappers around the K-means++ algorithm. All experimental results are an average of 300 Monte Carlo runs and the minimum and maximum number of clusters specified by the candidate models are set to $L_{\min} = 1$ and $L_{\max} = 2K$, where K is the true number of clusters in the considered data set. The degree of freedom parameter for BIC_{t_ν} and BIC_{ot_ν} is set to $\nu = 3$.

5.6.1 Performance Measures

We use the empirical probability of detection (p_{det}) and the mean absolute error (MAE), as defined in Section 2.6.5.1, to compare the cluster enumeration performance of the different methods. The cluster labeling performance of the different methods is compared using the average labeling rate (ALR), which is defined as

$$\text{ALR} = \frac{1}{IN} \sum_{i=1}^{I} \sum_{n=1}^{N} \mathbb{1}_{\{\hat{\mathcal{C}}_{kn}^{(i)} = \mathcal{C}_k\}}, \tag{5.1}$$

where I denotes the total number of Monte Carlo experiments, $\hat{\mathcal{C}}_{kn}^{(i)}$ represents the estimated cluster label of the nth data point at the ith experiment, $\mathcal{C}_k$ denotes the set of ground truth labels, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function. ALR is computed only if the particular method is able to estimate the correct number of clusters.

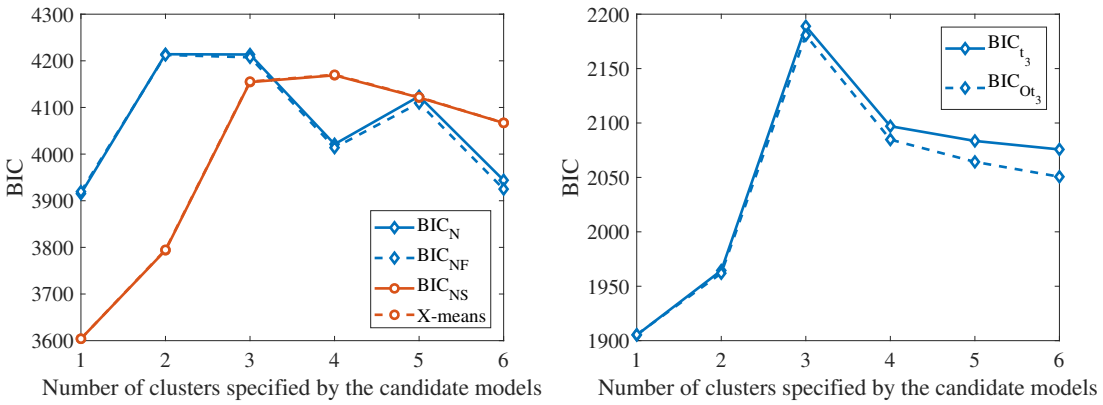
5.6.2 Numerical Experiments

In the first experiment, we use Data-2.1, which is a three cluster data set defined in Section 2.6.5.2, and set the number of data points per cluster as $N_1 = 100$, $N_2 = 200$, and $N_3 = 400$. This data set is particularly challenging for cluster enumeration and labeling algorithms because it contains not only overlapping but also unbalanced clusters. Table 5.1 shows the cluster enumeration and labeling performance of different methods. In terms of the estimation of the correct number of clusters, the robust methods, whose BIC curve is shown in Figure 5.1b, achieve the best performance, while BIC_{NS} and X-means perform the worst. As depicted in Figure 5.1a, the performance loss in BIC_{NS} and X-means emanates from overestimation, while the performance loss in BIC_{N} and BIC_{NF} is attributed to underestimation. The criteria that are implemented as wrappers around the EM algorithm have higher ALR as compared to the once which are implemented as wrappers around the K-means++ algorithm. This result is in line with our expectation since the data set contains elliptical as well as spherical clusters. An example of the labeling result of BIC_{NF} and BIC_{NS} is shown in Figure 5.2. Comparing Figure 5.2c and Figure 5.2e, we notice that the EM algorithm is able to capture the underlying structure of the data, while the K-means++ algorithm simply cuts the two overlapping clusters.

Table 5.1: Comparison of the cluster enumeration and labeling performance of our criteria and two state-of-the-art methods, namely the X-means algorithm and a robust implementation of the original BIC. The empirical probability of detection in %, the mean absolute error (MAE), and the average labeling rate (ALR) are used as performance measures.

		Data-2.1	Data-2.5		Data-3.3	Data-3.2
			$N_k = 150$	varying N_k		
BIC_{N}	p_{det}	63	98	44.67	0.33	0
	MAE	0.3733	0.03	**0.56**	1.30	1.36
	ALR	96.84	98.52	94.79	99.3	–
BIC_{NS}	p_{det}	53	0	0	73.33	7.67
	MAE	0.47	6.76	6.09	0.2933	1.82
	ALR	92.67	–	–	**99.57**	99.18
BIC_{NF}	p_{det}	60	**99.33**	**45**	0.33	0
	MAE	0.40	**0.01**	**0.56**	1.36	1.33
	ALR	**96.87**	**98.54**	**95.12**	99.3	–
X-means	p_{det}	52	0	0	73.33	7.67
	MAE	0.48	6.25	5.65	0.29	1.82
	ALR	92.67	–	–	**99.57**	99.18
BIC_{t_3}	p_{det}	**100**	85.67	14	99.33	**100**
	MAE	**0**	0.17	1.04	0.007	**0**
	ALR	96.55	97.99	85.62	99.56	**99.35**
$\text{BIC}_{\text{ot}_\nu}$	p_{det}	**100**	92.67	12.67	**99.67**	**100**
	MAE	**0**	0.07	1.16	**0.003**	**0**
	ALR	96.55	97.69	84.44	99.56	**99.35**

In the second experiment, we use Data-2.5, which is defined in Section 2.8.3.2. Data-2.5 is a 3-dimensional data set and it contains eight clusters. Here, we generate single node realizations of Data-2.5 by setting $N_k = 150$, for $k = 1, \ldots, 8$. As shown in Table 5.1, BIC_{NF} is the best method in terms of both cluster enumeration and labeling. BIC_{NS} and X-means severely overestimate the number of clusters, which is clear from their respective MAEs. The robust methods are inferior to BIC_{N} and BIC_{NF} due to a slight overestimation. To make this experiment even more challenging we varied the number of data points per cluster as follows: $N_1 = 100$, $N_2 = 200$, $N_3 = 50$, $N_4 = 300$, $N_5 = 200$, $N_6 = 400$, $N_7 = 50$, and $N_8 = 100$. As shown in Table 5.1, all methods perform poorly and tend to underestimate the



(a) Gaussian criteria

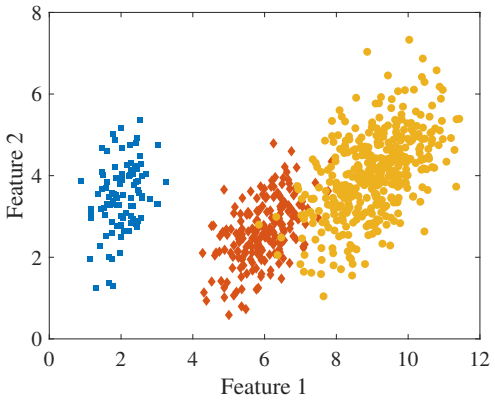
(b) Robust criteria

Figure 5.1: BIC curves of different criteria for Data-2.1.

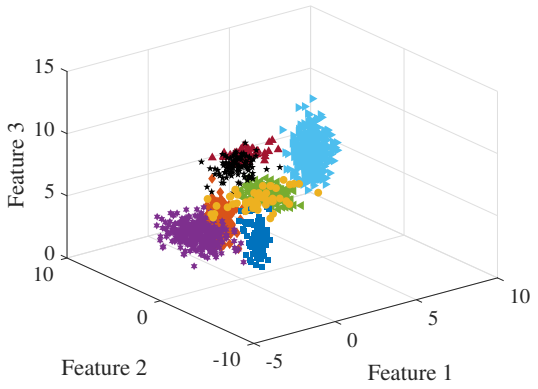
number of clusters. This behavior arises due to the big difference in the number of data points in different clusters. BIC_{N} and BIC_{NF} have higher ALR compared to the robust methods. An example of the labeling performance of BIC_{NF} and BIC_{t_3} in comparison to the ground truth labels is displayed in Figure 5.2.

To test the performance of the proposed joint cluster enumeration and labeling algorithm on a data set generated from a heavy tailed distribution, in our third experiment, we use Data-3.3, which is defined in Section 3.7.1. We set the number of features to $r = 2$ and the number of samples per cluster to $N_k = 500$, for $k = 1, 2$. The cluster enumeration and labeling performance of the different methods for this particular data set is displayed in Table 5.1. As expected, the robust methods outperform the others in terms of cluster enumeration, while all methods perform equally well in terms of the ALR. Interestingly, BIC_{NS} and X-means estimate the correct number of clusters more often than BIC_{N} and BIC_{NF}, see Figure 5.3 for a comparison of the BIC curves of the different criteria.

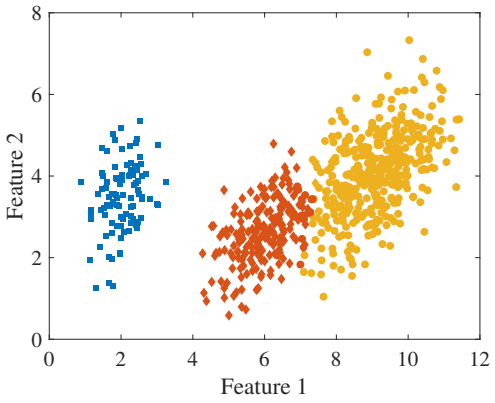
Finally, we test the performance of the different cluster enumeration and labeling methods in the presence of outliers by using Data-3.2, which is defined in Section 3.7.1, with 1% replacement outliers. As reported in Table 5.1, the robust methods are able to estimate the correct number of clusters 100% of the time, while BIC_{N} and BIC_{NF} overestimate the number of clusters 100% of the time. Given that the number of clusters is estimated correctly, all methods perform equally well in terms of the ALR. This is expected since only 1% of the data is contaminated with outliers and the remaining data form three well separated clusters.



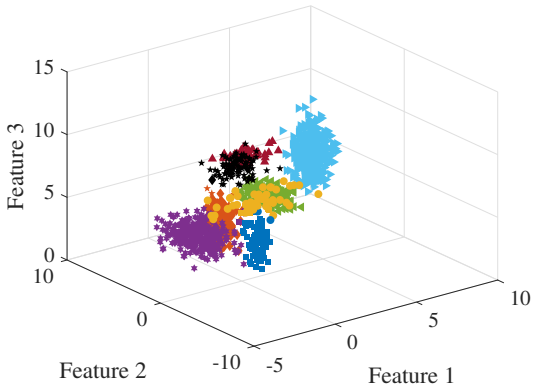
(a) Ground truth labels



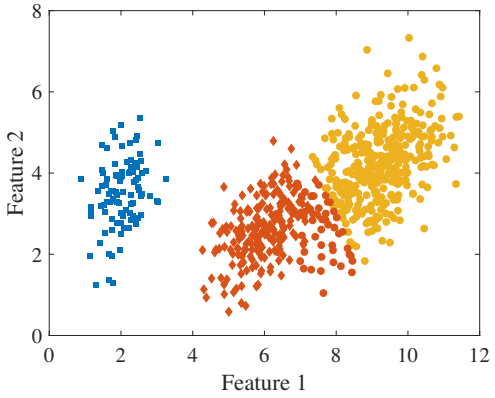
(b) Ground truth labels



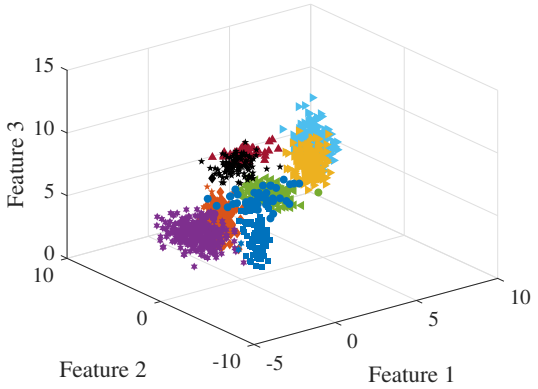
(c) Labeling result of $\mathrm{BIC}_{\mathrm{NF}}$



(d) Labeling result of $\mathrm{BIC}_{\mathrm{NF}}$

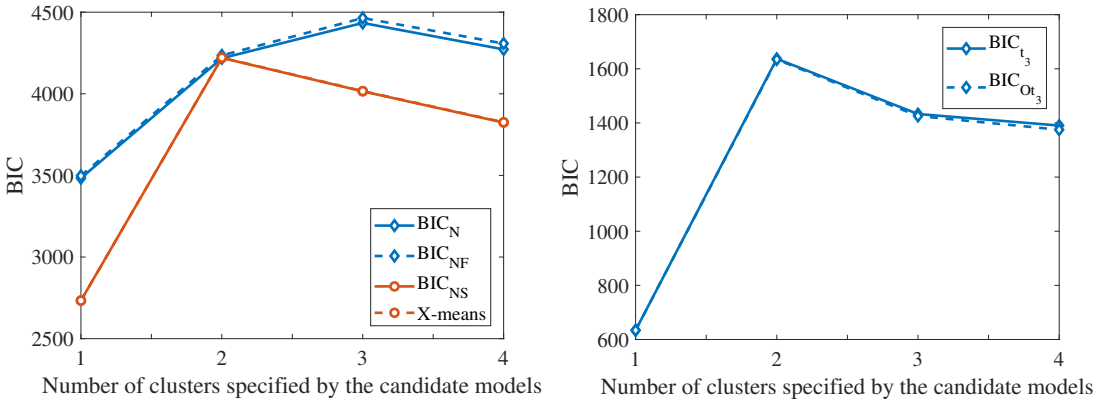


(e) Labeling result of $\mathrm{BIC}_{\mathrm{NS}}$



(f) Labeling result of BIC_{t_3}

Figure 5.2: Comparison of the labeling performance of different methods on Data-2.1 (left) and Data-2.5 (right). The ground truth labels (top plots) are given as references. In the figures shown in the last two rows shapes indicate ground truth labels and colors represent the estimated labels.



(a) Gaussian criteria

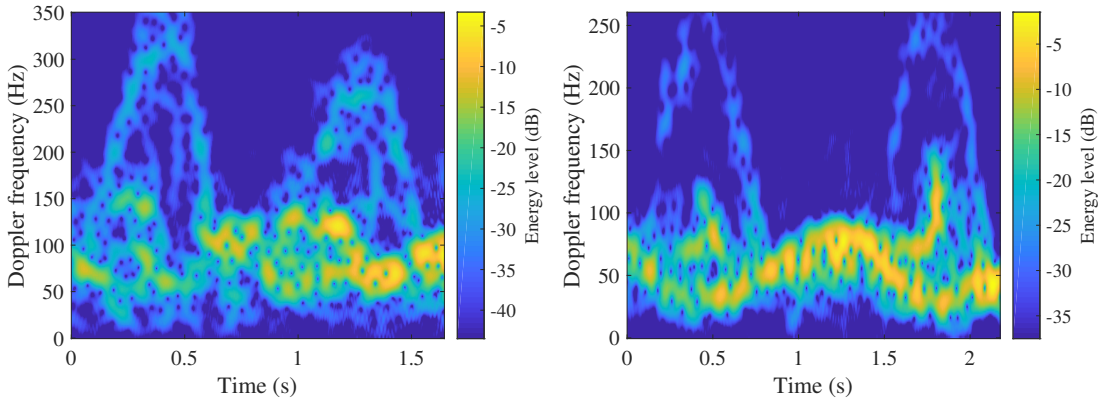
(b) Robust criteria

Figure 5.3: BIC curves of different criteria for Data-3.3.

5.6.3 Real Data Application: Target Enumeration and Labeling Using Radar Data of Human Gait

Previous works on radar-based sensing of humans are mostly concerned with detection or activity recognition, see for example [Amin, 2017; Chen et al., 2014; Amin, 2010]. On the other hand, identification of humans by the use of radar is relatively recent [Mokhtari et al., 2017; Ricci & Balleri, 2015; Garreau et al., 2011; Tahmoush & Silvious, 2009; Vandersmissen et al., 2018], where we note that there are similar works based on sonar data [Zhang & Andreou, 2008; Kaustubh & Bhiksha, 2007]. However, state-of-the-art methods on human identification require knowledge of the number of targets and availability of training data. These requirements are stringent in real-world applications, where the number of observed targets is mostly unknown and possibly time-varying. That is why, amongst other reasons (see [Teixeira et al., 2010] for a survey on human sensing), automatic identification of human subjects remains a challenging task for many ambient intelligent systems with application to surveillance, security, and smart homes.

In this section, we first describe data acquisition using an experimental radar setup and then discuss the feature extraction technique. Next, using two experiments, we demonstrate that joint cluster enumeration and labeling can be a valuable entity for advanced radar technologies that monitor human gait.



(a) Person A (b) Person B

Figure 5.4: Examples of micro-Doppler stride signatures of two individuals.

5.6.3.1 Experimental Radar Setup

Using a 24 GHz radar system [Ancortek Inc, 2018], the experimental data was collected in an office environment at Technische Universität Darmstadt. The antenna feed point was positioned at approximately 1.15 m above the floor. Five test subjects were asked to walk toward the radar system starting at approximately 4.5 m in front of the radar, where only one person was present in front of the radar at a time. Data were collected at a 0° angle relative to the radar line-of-sight and with a non-oblique view on the targets. The volunteers were asked to walk slowly and without swinging their arms. In total, 65 radar measurements of 6 seconds duration are considered. The number of measurements are equal among the test subjects, i.e, the data set contains 13 gait samples per person.

5.6.3.2 Feature Extraction

The recorded radar return signals are processed to obtain the spectrogram, see [Seifert et al., 2017] for more details. In order to detect single strides, the maxima of the envelope signal of the micro-Doppler signatures are utilized. The part of the spectrogram that shows a pair of strides, i.e., a full gait cycle, is extracted and converted to a gray scale image. All images are resized to have the same dimension, i.e., each image $\boldsymbol{S}_n \in \mathbb{R}^{f \times t}$, for $n = 1, \ldots, N$, with $f = 100$ and $t = 128$. Examples of extracted stride pairs for two individuals are shown in Figure 5.4. Each image $\boldsymbol{S}_n$, for $n = 1, \ldots, N$, is vectorized to create a long column vector

$\boldsymbol{s}_n \in \mathbb{R}^{d\times 1}$, where $d = ft$, which is referred to as feature vector. In the set of feature vectors $\mathcal{S} \triangleq \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\} \subset \mathbb{R}^{d\times N}$, $d > N$, which creates a sample scarce scenario.

To mitigate the curse of dimensionality, we reduce the dimension of our set of feature vectors $\mathcal{S}$ from d to r, where $r < d$, using a probabilistic PCA [Tipping & Bishop, 1999] that approximates the likelihood function of $\mathcal{S}$ as

$$p(\mathcal{S}|c) \approx N^{-\frac{1}{2}(z+c)} \left(\frac{\sum_{a=c+1}^{d} \lambda_a}{d-c} \right)^{-\frac{1}{2}N(d-c)} \left(\prod_{a=1}^{c} \lambda_a \right)^{-\frac{1}{2}N}, \tag{5.2}$$

where c denotes the number of principal components, λ_a, for $a = 1, \ldots, d$, are the eigenvalues, and $z = d(d-1)/2 - (d-c)(d-c-1)/2$ [Minka, 2001]. Once the likelihood function is evaluated for each candidate number of principal components $c = C_{\min}, \ldots, C_{\max}$, the correct number of principal components is selected as [Minka, 2001]

$$r = \underset{c=C_{\min},\ldots,C_{\max}}{\arg\max} \log p(\mathcal{S}|c). \tag{5.3}$$

Then, the new set of feature vectors with reduced dimensions is given by

$$\mathcal{X} = \boldsymbol{V}^\top \mathcal{S}, \tag{5.4}$$

where $\mathcal{X} \subset \mathbb{R}^{r\times N}$ and the column vectors of $\boldsymbol{V} \in \mathbb{R}^{d\times r}$ are the eigenvectors of $\mathcal{S}$ corresponding to the first r eigenvalues such that $\lambda_1 > \lambda_2 > \ldots > \lambda_r > 0$. This way, a small number of descriptive features is automatically extracted for each spectrogram.

5.6.3.3 Person Enumeration and Labeling

Scenario-1

Considering the first four persons, $N = 187$ stride pairs are obtained from 52 radar measurements, where person A, B, C, and D are represented by $N_1 = 40$, $N_2 = 38$, $N_3 = 62$, and $N_4 = 47$ samples, respectively. Using (5.3), 5 principal components are selected, such that the original set of vectorized spectrogram images, $\mathcal{S} \subset \mathbb{R}^{12800\times 187}$, is reduced to $\mathcal{X} \subset \mathbb{R}^{5\times 187}$. As an example, Figure 5.5 shows a scatter plot of principal component scores using three principal components. Note that estimating the number of clusters in $\mathcal{X}$ is very challenging because $\mathcal{X}$

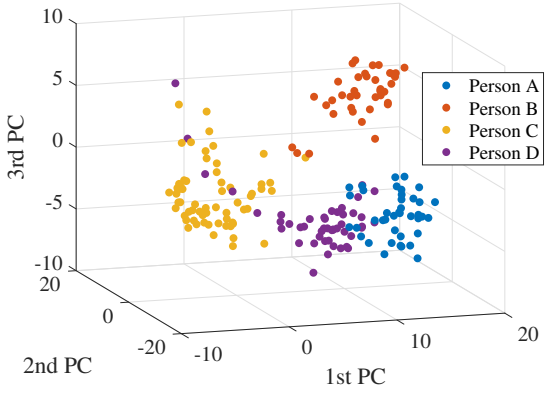
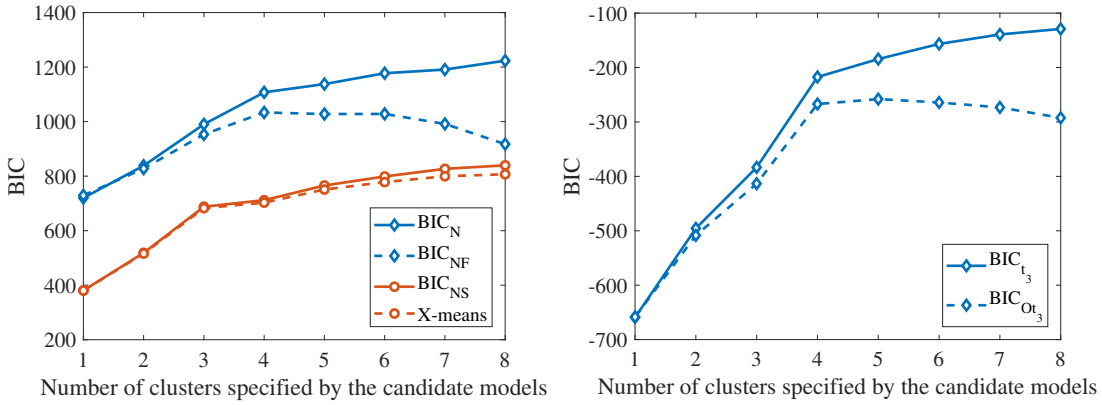


Figure 5.5: Principal component scores for radar-based human gait data of four different persons using the first three principal components (PC).



(a) Gaussian criteria

(b) Robust criteria

Figure 5.6: BIC curves of different criteria for the radar data set.

has few feature vectors which results in even fewer feature vectors per cluster.

Figure 5.6 shows the BIC computed by the different methods as a function of the number of clusters specified by the candidate models. Only BIC_{NF} is able to estimate the correct number of clusters (or persons), which corresponds to $\hat{K}_{\text{BIC}_{\text{NF}}} = 4$, 96% of the time, while the other methods overestimate the number of clusters 100% of the time. The asymptotic methods, BIC_{N}, BIC_{NS}, BIC_{t_3}, BIC_{ot_3}, and X-means, stand at a disadvantage when the number of feature vectors is small because they are derived assuming that the number of feature vectors $N \to \infty$. In such cases, BIC_{NF} is more appropriate because its penalty term is refined for the finite sample regime.

Table 5.2: Confusion matrix in percentage for person labeling using the proposed joint cluster enumeration and labeling algorithm.

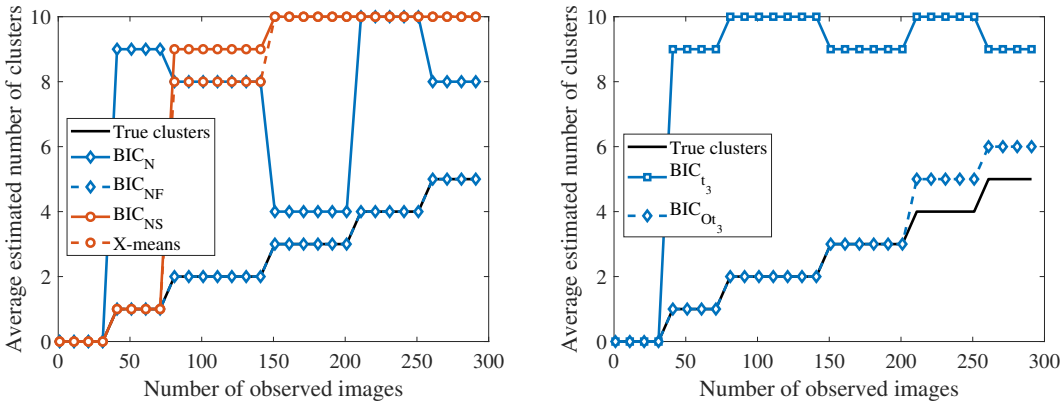
		Estimated labels			
		A	B	C	D
True labels	A	**100**	0	0	0
	B	0	**100**	0	0
	C	0	6.45	**93.55**	0
	D	2.13	6.38	2.13	**89.36**

Table 5.3: Confusion matrix in percentage for person recognition using a NN classifier.

		Estimated labels			
		A	B	C	D
True labels	A	**100**	0	0	0
	B	0	**97.25**	2.75	0
	C	0	1.33	**98**	0.62
	D	0.3	0.1	6.1	**93.50**

Since BIC_{NF} is the only criterion that results in the correct estimate of the number of clusters in $\mathcal{X}$, we show the labeling performance of the proposed joint cluster enumeration and labeling algorithm using the cluster enumeration result of BIC_{NF}. Table 5.2 shows the confusion matrix generated by the proposed method. The first two persons are correctly labeled 100% of the time, while person D is often confused with the remaining targets, but is still recognized in approximately 89% of the cases. Overall, we achieve a high labeling rate using the proposed method. Note that we get an average labeling rate of 95.73% without a prior knowledge of the number of clusters (or targets) and no training data.

In order to underscore the performance of the joint cluster enumeration and labeling algorithm, we also present results obtained using the same set of feature vectors $\mathcal{X}$, but a trained classifier for discriminating the four different persons. Using a simple nearest neighbor (NN) classifier, we obtain the confusion matrix shown in Table 5.3, where 80% of the data was used to train the classifier and the remainder was used for testing. The reported numbers are the average rates over 100 classifications, where training and test data were randomly chosen. The overall accuracy is 97.19%, where person A is correctly classified in all cases and person D shows the lowest labeling rate with 93.5%.



(a) Gaussian criteria (b) Robust criteria

Figure 5.7: Estimated number of clusters (or persons) for the radar data set as a function of the number of observed images.

We note that, the results obtained using a trained classifier and the proposed unsupervised cluster enumeration and labeling algorithm are comparable, despite the fact that 80% of the data was available to the classifier for training. In some real-world applications, however, training data is unavailable. In such cases, joint cluster enumeration and labeling algorithms, such as the one presented in this chapter, can provide a high target labeling rate without training data and prior knowledge of the number of clusters.

Scenario-2

In this scenario, we consider a different measurement setup. First, we observe person A and collect $N_1 = 40$ images. Next, we do the same for person B, where $N_2 = 38$. Every time a new person is observed, in total, $N = \sum_{k=1}^{K} N_k$, where K is the number of persons already observed. We do this for five persons in a sequential manner, resulting in $N = 252$ images. Whenever a set of images is available, we re-estimate the total number of targets observed by the radar so far. For this, the number of principal components is also re-estimated based on the current set of feature vectors.

Figure 5.7 shows the number of estimated clusters (or persons), $\hat{K}$, as a function of the number of observed images n, where $n = 1, \ldots, N$. Due to the setup described above, the true number of clusters forms a staircase. Among the compared Bayesian cluster enumeration criteria BIC_{NF} is the only criterion that is able to correctly estimate the number of persons and

track the change in the number of persons as we observe more images. The second best performance is achieved by BIC_{ot_3}. Due to the small number of available features, BIC_{N} and BIC_{t_3} do not correctly estimate the number of persons even though they respond to the change in the number of observed persons. BIC_{NS} and X-means are able to correctly estimate the number of persons in the beginning but quickly overestimates K and choose the specified maximum number of clusters as we observe more images.

5.7 Summary

In this chapter, we proposed a joint cluster enumeration and labeling algorithm by extending the cluster enumeration criteria that were derived in Part I of the dissertation. The performance of the proposed method in estimating the number of clusters and providing individual clusters with unique labels was demonstrated using five numerical experiments. Further, the proposed method was applied to person labeling using radar measurements of the human gait. Despite short observation times, the persons were labeled with a high accuracy in the absence of training data and knowledge of the number of persons. The proposed method also showed promising result in tracking the change in the number of persons.

PART IV

Conclusions and Outlook

6

Summary and Conclusions

The dissertation contributes to the area of cluster analysis by developing statistical methods that determine the number of clusters and cluster memberships. The developed methods tackled challenging data clustering scenarios, such as the presence of outliers, cluster overlap, or cluster heterogeneity in the observed data set. Use cases in distributed camera networks and radar-based person identification demonstrated the applicability of the developed methods in advanced signal processing problems.

In particular, a new Bayesian cluster enumeration criterion was derived by formulating the problem of estimating the number of clusters as maximization of the posterior probability of candidate models. In a nutshell, this formulation transformed cluster enumeration into a model selection problem where the model with the highest posterior probability is selected among a family of candidate models. The new criterion is applicable to a broad class of data distributions and, consequently, serves as a starting point when deriving cluster enumeration criteria for specific data distributions. Following this line of argument, a robust and a Gaussian criterion were derived by modeling the data as a family of multivariate t_ν and Gaussian distributions, respectively. Further, the penalty terms of both criteria were refined for the finite sample regime. In contrast to Schwarz's BIC [Schwarz, 1978], which is a generic criterion, the new robust and Gaussian criterion can be interpreted as the BIC derived specifically for

cluster analysis. Interestingly, as the data distribution of the candidate models changes, the data fidelity and penalty terms of our criteria also change, while for the original BIC changes in the data distribution only affect the data fidelity term. Our derivation of the BIC for clustering problems supports the statements made by the author in [Djurić, 1998] who came to the conclusion that the original BIC should be carefully analyzed before being applied to specific model selection problems.

The derived cluster enumeration criteria were incorporated into two-step algorithms where the cluster assignment and model parameter estimation tasks were separated from the enumeration. While the EM algorithm was used in this dissertation to provide a unified framework, the proposed enumeration criteria can be used as a wrapper around any clustering algorithm. This allows for selecting the clustering algorithm according to the application's demands without affecting the cluster enumeration strategy. Numerical and real data experiments demonstrated the superiority of the proposed algorithms over existing cluster enumeration methods.

Real-world applicability of the proposed cluster enumeration framework was demonstrated on use cases in the area of distributed sensor networks and radar technologies for assisted living. Specifically, the cluster enumeration criteria were extended to a distributed sensor network setup where the nodes exchange valuable information via the diffusion principle. Two distributed and adaptive Bayesian cluster enumeration algorithms were proposed and applied to a camera network use case, where multiple cameras film a non-stationary scene from different angles. The number of pedestrians was estimated based on streaming-in data requiring neither registration of camera views nor a fusion center. Good performance was achieved compared to an existing distributed cluster enumeration algorithm.

A further research goal of the dissertation was the cluster membership assignment of individual data points and their associated cluster labels assuming that the number of clusters is either prespecified by the user or estimated via the above described methods. Solving this problem is relevant for real-world applications, such as video surveillance and sports analysis, since object labeling can be formulated as a data clustering and labeling task after extracting valuable features from the observed data. To this end, a robust object labeling algorithm was proposed for a distributed camera network whose nodes are interested in a static scene. In addition, an adaptive object labeling and tracking algorithm was developed for the case where nodes in an ad hoc camera network monitor a time-varying scene from different viewpoints

in the absence of a fusion center. Both algorithms were shown to provide good labeling performance using images and videos recorded by uncalibrated distributed camera networks.

Finally, the simultaneous estimation of the number of clusters and cluster memberships was tackled by proposing a joint cluster enumeration and labeling algorithm. The proposed unsupervised method was applied to person enumeration and labeling in a real data application of radar-based person identification. The proposed approach performed as good as a supervised learning method which requires knowledge of the number of individuals and training data.

7

Future Research Directions

Possible extensions of the developed cluster enumeration and labeling framework and open research directions are summarized below.

7.1 Extension of the Robust Bayesian Cluster Enumeration Criteria

In Chapter 3, we have assumed that the degree of freedom parameter ν is fixed at some prespecified value in order to simplify the theoretical derivations. Ideally, one can extend the derivations by treating the degree of freedom of each cluster in each candidate model as an unknown parameter. Such an extension allows the robust cluster enumeration criteria to model the data with a family of t_ν distributions, which ranges from the Cauchy distribution for $\nu = 1$ all the way to the Gaussian distribution for $\nu = \infty$. This means that we can drop the assumption that the distribution of all clusters has the same degree of freedom. Consequently, in the same data set, some clusters may be heavy tailed, while others are Gaussian. However, the extension is not straightforward as it requires the justification of assumption (A-2.3) when the parameter vector of the mth cluster is given by $\boldsymbol{\theta}_m = [\boldsymbol{\mu}_m, \boldsymbol{\Psi}_m, \nu_m]^\top$.

7.2 Theoretical Analysis of the Proposed Bayesian Cluster Enumeration Criteria

In this dissertation, we have tested the performance of the proposed Bayesian cluster enumeration criteria using numerical and real data experiments. Specifically, the performance of the proposed criteria in estimating the correct number of clusters is tested using the empirical probability of detection and the mean absolute error as performance metrics. The original BIC is known to be consistent if the true data generating model belongs to the family of candidate models under investigation [Schwarz, 1978]. Since the proposed criteria asymptotically converge to the original BIC, we empirically conjuncture that our criteria are also consistent. However, a formal proof of consistency for our criteria that carefully checks all the assumptions made in the proof of consistency for the original BIC would provide an interesting future work. In addition, establishing the concepts of qualitative and quantitative robustness for the robust criteria discussed in Chapter 3 is an open problem. In particular, defining the breakdown point of the robust cluster number estimator for increasing percentages of outliers requires a careful analysis that takes into account the cluster memberships. This is because, for example, a consistent overestimation of the number of clusters that simply groups all outliers into a separate cluster should not be considered as a breakdown of the method. However, for this example, the probability of detection of the true number of clusters is zero. This example shows that in the context of robust cluster enumeration care must be taken when defining the error metrics. Also, it is not fully clear what the true number of clusters should be in this case. Finding robustness and performance measures that take such problems into account constitutes an intriguing line of future research, as it raises fundamental questions, such as, *what is a cluster?*, and *what are outliers?*.

7.3 Efficiency of Clustering Algorithms in Partitioning Data

Standard clustering algorithms, such as the K-means and the EM algorithm, solve a non-convex problem and, consequently, chances of the algorithms converging to a local optimum is very high. These algorithms are iterative in nature, and, depending on the starting point,

they can end up with different solutions for the same data set. This problem is known in the literature and different remedies have been proposed [Arthur & Vassilvitskii, 2007; Fränti, 2018; Zhao et al., 2012; Blömer & Bujna, 2016]. Local convergence of the clustering algorithms becomes even more amplified when there are outliers in the data set. As discussed in Section 2.6.5.2, the clustering method is the backbone of the proposed two-step cluster enumeration algorithms. If the clustering method used in the first step fails at partitioning the data set correctly, then the performance of the second step, which is the calculation of one of the proposed criteria, is bound to be bad. Hence, investigating better clustering algorithms or even improving the ones that were used in the dissertation is a valuable extension of our work.

7.4 Cluster analysis in High-Dimensional Spaces

In high-dimensional spaces, the relative distance from any data point to its nearest and farthest neighbor tend to be almost identical [Klawonn et al., 2015; Zimek et al., 2012]. Consequently, cluster analysis becomes a complicated task due to the distance concentration effect and the presence of irrelevant features hiding relevant information. In this dissertation, we focused on the case where the number of features (r) is smaller than the number of samples (N). Whenever, $r > N$, we used dimension reduction techniques prior to estimating the number of clusters and cluster memberships. However, the extension of the proposed methods to high-dimensional spaces where $r \geq N$ is an open problem, and future research in this area is essential. A possible, and very interesting research direction is that of regularizing the clustering method to account for the sparsity of the high-dimensional spaces. On the other hand, a non-asymptotic derivation of a cluster enumeration criterion that considers $\frac{r}{N} \to c$, where $c > 0$ is a constant, would be relevant.

PART V

Appendix

A

Maximum Likelihood Estimators

A.1 The Maximum Likelihood Estimators of the Parameters of Multivariate Gaussian Distributed Random Variables

Given that the data points that belong to the mth cluster ($\mathcal{X}_m$) are realizations of iid Gaussian random variables $\boldsymbol{x}_m \sim \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, the log-likelihood function is written as

$$\begin{aligned}
\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m) &= \log \prod_{\boldsymbol{x}_n \in \mathcal{X}_m} p(\boldsymbol{x}_n \in \mathcal{X}_m) f(\boldsymbol{x}_n|\boldsymbol{\theta}_m) \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \log\left(\frac{N_m}{N} \frac{1}{(2\pi)^{\frac{r}{2}} |\boldsymbol{\Sigma}_m|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Sigma}_m^{-1} \tilde{\boldsymbol{x}}_n\right)\right) \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \left(\log\frac{N_m}{N} - \frac{r}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_m| - \frac{1}{2}\tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Sigma}_m^{-1} \tilde{\boldsymbol{x}}_n\right) \\
&= N_m \log\frac{N_m}{N} - \frac{r N_m}{2}\log 2\pi - \frac{N_m}{2}\log|\boldsymbol{\Sigma}_m| - \frac{1}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Sigma}_m^{-1} \tilde{\boldsymbol{x}}_n,
\end{aligned} \tag{A.1}$$

where $\tilde{\boldsymbol{x}}_n \triangleq \boldsymbol{x}_n - \boldsymbol{\mu}_m$, N_m denotes the number of data points in the mth cluster, and N represents the total number of data points in the data set. As the name implies, the maximum likelihood estimator attempts to maximize the likelihood function. To accomplish this, (A.1) is first derivated with respect to its parameters, which results in

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m} = \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Sigma}_m^{-1} \tag{A.2}$$

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\Sigma}_m} = -\frac{N_m}{2}\boldsymbol{\Sigma}_m^{-1} + \frac{1}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \boldsymbol{\Sigma}_m^{-1}\tilde{\boldsymbol{x}}_n\tilde{\boldsymbol{x}}_n^\top\boldsymbol{\Sigma}_m^{-1}. \tag{A.3}$$

Then, setting (A.2) and (A.3) to zero and solving the resulting expressions result in

$$\hat{\boldsymbol{\mu}}_m = \frac{1}{N_m}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \boldsymbol{x}_n \tag{A.4}$$

$$\hat{\boldsymbol{\Sigma}}_m = \frac{1}{N_m}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \tilde{\boldsymbol{x}}_n\tilde{\boldsymbol{x}}_n^\top \tag{A.5}$$

A.2 The Maximum Likelihood Estimators of the Parameters of Multivariate t_ν Distributed Random Variables

If the data points that belong to the mth cluster ($\mathcal{X}_m$) are realizations of iid multivariate t_{ν_m} distributed random variables $\boldsymbol{x}_m \sim t_{\nu_m}(\boldsymbol{\mu}_m, \boldsymbol{\Psi}_m)$, then the log-likelihood function is given by

$$\begin{aligned}
\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m) &= \log \prod_{\boldsymbol{x}_n \in \mathcal{X}_m} p(\boldsymbol{x}_n \in \mathcal{X}_m) f(\boldsymbol{x}_n|\boldsymbol{\theta}_m) \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \log\left(\frac{N_m}{N}\frac{\Gamma\left((\nu_m + r)/2\right)}{\Gamma\left(\nu_m/2\right)(\pi\nu_m)^{r/2}|\boldsymbol{\Psi}_m|^{1/2}}\left(1 + \frac{\delta_n}{\nu_m}\right)^{-(\nu_m+r)/2}\right) \\
&= N_m \log\frac{N_m}{N} + N_m \log\frac{\Gamma\left((\nu_m + r)/2\right)}{\Gamma\left(\nu_m/2\right)(\pi\nu_m)^{r/2}} - \frac{N_m}{2}\log|\boldsymbol{\Psi}_m| \\
&\quad - \frac{(\nu_m + r)}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \log\left(1 + \frac{\delta_n}{\nu_m}\right),
\end{aligned} \tag{A.6}$$

where $\delta_n = (\boldsymbol{x}_n - \boldsymbol{\mu}_m)^\top \boldsymbol{\Psi}_m^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_m)$ is the squared Mahalanobis distance, $\Gamma(\cdot)$ is the gamma function, N_m denotes the number of data points in the mth cluster, and N represents the total number of data points in the data set. To find the maximum likelihood estimators of the centroid $\boldsymbol{\mu}_m$ and the scatter matrix $\boldsymbol{\Psi}_m$, we first derivate the log-likelihood function with respect to each parameter, which results in

$$\begin{aligned}
\frac{\partial \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m} &= -\frac{(\nu_m + r)}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{d}{d\boldsymbol{\mu}_m} \left(\log \left(1 + \frac{\delta_n}{\nu_m} \right) \right) \\
&= -\frac{1}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{\nu_m + r}{\nu_m + \delta_n} \frac{d\delta_n}{d\boldsymbol{\mu}_m} \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1} \qquad \text{(A.7)} \\
\frac{\partial \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\Psi}_m} &= -\frac{N_m}{2} \operatorname{Tr} \left(\boldsymbol{\Psi}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right) - \frac{(\nu_m + r)}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{d}{d\boldsymbol{\Psi}_m} \left(\log \left(1 + \frac{\delta_n}{\nu_m} \right) \right) \\
&= -\frac{N_m}{2} \operatorname{Tr} \left(\boldsymbol{\Psi}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right) - \frac{1}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{\nu_m + r}{\nu_m + \delta_n} \frac{d\delta_n}{d\boldsymbol{\Psi}_m} \\
&= -\frac{N_m}{2} \operatorname{Tr} \left(\boldsymbol{\Psi}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right) + \frac{1}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \operatorname{Tr} \left(\boldsymbol{\Psi}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right) \\
&= -\frac{N_m}{2} \boldsymbol{\Psi}_m^{-1} + \frac{1}{2} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \boldsymbol{\Psi}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}, \qquad \text{(A.8)}
\end{aligned}$$

where $\tilde{\boldsymbol{x}}_n = \boldsymbol{x}_n - \boldsymbol{\mu}_m$ and

$$w_n = \frac{\nu_m + r}{\nu_m + \delta_n} \qquad \text{(A.9)}$$

is the weight given to $\boldsymbol{x}_n$. Then, setting (A.7) and (A.8) to zero and simplifying the resulting expressions result in

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \boldsymbol{x}_n}{\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n} \qquad \text{(A.10)}$$

$$\hat{\boldsymbol{\Psi}}_m = \frac{1}{N_m} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top. \qquad \text{(A.11)}$$

B

Proofs

B.1 Proof of Theorem 2.1

Proving Theorem 2.1 requires finding an asymptotic approximation of the FIM, $\hat{\boldsymbol{J}}_m$, for $m = 1, \ldots, l$, and, based on this approximation, showing how (2.17) is simplified to come up with the expression for $\mathrm{BIC}_{\mathrm{N}}$ in (2.19). To obtain an asymptotic approximation of $\hat{\boldsymbol{J}}_m$, we first express the log-likelihood function of the data points that belong to the mth cluster by (A.1). The first-order derivative of $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ with respect to $\boldsymbol{\theta}_m$ is given by

$$\begin{aligned}\frac{d \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m} &= -\frac{N_m}{2} \operatorname{Tr}\left(\boldsymbol{\Sigma}_m^{-1} \frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\right) + \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Sigma}_m^{-1} \frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\Delta}_m\right) \\ &\quad + \operatorname{Tr}\left(\boldsymbol{\Sigma}_m^{-1} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m} \tilde{\boldsymbol{x}}_n^\top\right) \\ &= \frac{1}{2} \operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{E}_m \boldsymbol{\Sigma}_m^{-1}\right) + N_m \operatorname{Tr}\left((\bar{\boldsymbol{x}}_m - \boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right),\end{aligned} \tag{B.1}$$

where $\bar{\boldsymbol{x}}_m \triangleq \frac{1}{N_m} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \boldsymbol{x}_n$ is the sample mean of the data points that belong to the mth cluster, $\tilde{\boldsymbol{x}}_n \triangleq \boldsymbol{x}_n - \boldsymbol{\mu}_m$, $\boldsymbol{\Delta}_m \triangleq \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top$, and $\boldsymbol{E}_m \triangleq \boldsymbol{\Delta}_m - N_m \boldsymbol{\Sigma}_m$. To make the

dissertation self contained, we have included the most important vector and matrix differentiation rules in Appendix D (see [Magnus & Neudecker, 2007] for details).

Differentiating (B.1) with respect to $\boldsymbol{\theta}_m^\top$ results in

$$\begin{aligned}
\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top} &= \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\frac{d\boldsymbol{\Sigma}_m^{-1}}{d\boldsymbol{\theta}_m^\top}\boldsymbol{E}_m\boldsymbol{\Sigma}_m^{-1}\right) + \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\boldsymbol{E}_m\frac{d\boldsymbol{\Sigma}_m^{-1}}{d\boldsymbol{\theta}_m^\top}\right) \\
&\quad + \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{E}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) + N_m\operatorname{Tr}\left((\bar{\boldsymbol{x}}_m-\boldsymbol{\mu}_m)^\top\frac{d\boldsymbol{\Sigma}_m^{-1}}{d\boldsymbol{\theta}_m^\top}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) \\
&\quad - N_m\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m^\top} \\
&= -\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\Delta}_m - N_m\boldsymbol{\Sigma}_m)\boldsymbol{\Sigma}_m^{-1}\right) \\
&\quad - N_m\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}(\bar{\boldsymbol{x}}_m-\boldsymbol{\mu}_m)\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) \\
&\quad - \frac{N_m}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) \\
&\quad - N_m\operatorname{Tr}\left((\bar{\boldsymbol{x}}_m-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) - N_m\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m^\top} \\
&= \frac{N_m}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) - \operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\boldsymbol{\Delta}_m\boldsymbol{\Sigma}_m^{-1}\right) \\
&\quad - N_m\operatorname{Tr}\left(\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}(\bar{\boldsymbol{x}}_m-\boldsymbol{\mu}_m)\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) \\
&\quad - N_m\operatorname{Tr}\left(\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}(\bar{\boldsymbol{x}}_m-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\Sigma}_m}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Sigma}_m^{-1}\right) - N_m\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m}\boldsymbol{\Sigma}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m^\top}.
\end{aligned} \tag{B.2}$$

Next, we exploit the symmetry of the covariance matrix $\boldsymbol{\Sigma}_m$ to come up with a final expression for the second-order derivative of $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$. The unique elements of $\boldsymbol{\Sigma}_m$ can be collected into a vector $\boldsymbol{u}_m \in \mathbb{R}^{\frac{1}{2}r(r+1)\times 1}$ as defined in [Magnus & Neudecker, 2007, pp. 56–57]. Hence, incorporating the symmetry of the covariance matrix $\boldsymbol{\Sigma}_m$ and replacing the parameter vector $\boldsymbol{\theta}_m$ by $\check{\boldsymbol{\theta}}_m = [\boldsymbol{\mu}_m, \boldsymbol{u}_m]^\top$ in (B.2) results in the following expression:

$$\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\check{\boldsymbol{\theta}}_m d\check{\boldsymbol{\theta}}_m^\top} = \frac{N_m}{2}\operatorname{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m}\right)^\top\boldsymbol{V}_m\operatorname{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m^\top}\right) - \operatorname{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m^\top}\right)^\top\boldsymbol{W}_m\operatorname{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m}\right)$$

$$
\begin{aligned}
&- N_m \mathrm{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m}\right)^\top \boldsymbol{Z}_m \mathrm{vec}\left(\frac{d\boldsymbol{\mu}_m^\top}{d\check{\boldsymbol{\theta}}_m^\top}\right) - N_m \mathrm{vec}\left(\frac{d\boldsymbol{\mu}_m}{d\check{\boldsymbol{\theta}}_m}\right)^\top \boldsymbol{Z}_m^\top \mathrm{vec}\left(\frac{d\boldsymbol{\Sigma}_m}{d\check{\boldsymbol{\theta}}_m^\top}\right) \\
&- N_m \frac{d\boldsymbol{\mu}_m^\top}{d\check{\boldsymbol{\theta}}_m} \boldsymbol{\Sigma}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\check{\boldsymbol{\theta}}_m^\top},
\end{aligned} \tag{B.3}
$$

where

$$
\boldsymbol{V}_m \triangleq \boldsymbol{\Sigma}_m^{-1} \otimes \boldsymbol{\Sigma}_m^{-1} \in \mathbb{R}^{r^2 \times r^2} \tag{B.4}
$$

$$
\boldsymbol{W}_m \triangleq \boldsymbol{\Sigma}_m^{-1} \otimes \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\Delta}_m \boldsymbol{\Sigma}_m^{-1} \in \mathbb{R}^{r^2 \times r^2} \tag{B.5}
$$

$$
\boldsymbol{Z}_m \triangleq \boldsymbol{\Sigma}_m^{-1} \otimes \boldsymbol{\Sigma}_m^{-1} (\bar{\boldsymbol{x}}_m - \boldsymbol{\mu}_m) \in \mathbb{R}^{r^2 \times r}. \tag{B.6}
$$

For the symmetric matrix $\boldsymbol{\Sigma}_m$, the duplication matrix $\boldsymbol{D} \in \mathbb{R}^{r^2 \times \frac{1}{2}r(r+1)}$ transforms $\boldsymbol{u}_m$ into $\mathrm{vec}(\boldsymbol{\Sigma}_m)$ using the relation $\mathrm{vec}(\boldsymbol{\Sigma}_m) = \boldsymbol{D}\boldsymbol{u}_m$ [Magnus & Neudecker, 2007, pp. 56–57]. Hence, (B.3) can be further simplified into

$$
\begin{aligned}
\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\check{\boldsymbol{\theta}}_m d\check{\boldsymbol{\theta}}_m^\top} = \frac{N_m}{2} &\left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m}\right)^\top \boldsymbol{D}^\top \boldsymbol{V}_m \boldsymbol{D} \frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top} - \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top}\right)^\top \boldsymbol{D}^\top \boldsymbol{W}_m \boldsymbol{D} \frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m} \\
&- N_m \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m}\right)^\top \boldsymbol{D}^\top \boldsymbol{Z}_m \mathrm{vec}\left(\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top}\right) - N_m \mathrm{vec}\left(\frac{d\boldsymbol{\mu}_m}{d\check{\boldsymbol{\theta}}_m}\right)^\top \boldsymbol{Z}_m^\top \boldsymbol{D} \frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top} \\
&- N_m \frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m} \boldsymbol{\Sigma}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m^\top}.
\end{aligned} \tag{B.7}
$$

A compact matrix representation of the second-order derivative of $\log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)$ is given by

$$
\frac{d^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\check{\boldsymbol{\theta}}_m d\check{\boldsymbol{\theta}}_m^\top} = \begin{bmatrix} \frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top} & \frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top} \\ \frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{\mu}_m^\top} & \frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} \end{bmatrix}. \tag{B.8}
$$

The individual elements of the above matrix can be written as

$$
\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top} = -N_m \boldsymbol{\Sigma}_m^{-1} \tag{B.9}
$$

$$
\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top} = -N_m \boldsymbol{Z}_m^\top \boldsymbol{D} \tag{B.10}
$$

$$
\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{\mu}_m^\top} = -N_m \boldsymbol{D}^\top \boldsymbol{Z}_m \tag{B.11}
$$

$$\frac{\partial^2 \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} = \frac{N_m}{2} \boldsymbol{D}^\top \boldsymbol{F}_m \boldsymbol{D}, \tag{B.12}$$

where $\boldsymbol{F}_m \triangleq \boldsymbol{\Sigma}_m^{-1} \otimes \left(\boldsymbol{\Sigma}_m^{-1} - \frac{2}{N_m} \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\Delta}_m \boldsymbol{\Sigma}_m^{-1} \right) \in \mathbb{R}^{r^2 \times r^2}$.

The FIM of the mth cluster is given by

$$\begin{aligned} \hat{\boldsymbol{J}}_m &= \begin{bmatrix} -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top} & -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top} \\ -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{\mu}_m^\top} & -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} \end{bmatrix} \\ &= \begin{bmatrix} N_m \hat{\boldsymbol{\Sigma}}_m^{-1} & N_m \hat{\boldsymbol{Z}}_m^\top \boldsymbol{D} \\ N_m \boldsymbol{D}^\top \hat{\boldsymbol{Z}}_m & -\frac{N_m}{2} \boldsymbol{D}^\top \hat{\boldsymbol{F}}_m \boldsymbol{D} \end{bmatrix}. \end{aligned} \tag{B.13}$$

The maximum likelihood estimators of the mean and covariance matrix of the mth Gaussian cluster are given by (A.4) and (A.5), respectively (see Appendix A.1 for details). Hence, $\hat{\boldsymbol{Z}}_m \triangleq \hat{\boldsymbol{\Sigma}}_m^{-1} \otimes \hat{\boldsymbol{\Sigma}}_m^{-1} (\bar{\boldsymbol{x}}_m - \hat{\boldsymbol{\mu}}_m) = \boldsymbol{0}_{r^2 \times r}$. Consequently, (B.13) can be further simplified to

$$\hat{\boldsymbol{J}}_m = \begin{bmatrix} N_m \hat{\boldsymbol{\Sigma}}_m^{-1} & \boldsymbol{0}_{r \times \frac{1}{2} r(r+1)} \\ \boldsymbol{0}_{\frac{1}{2} r(r+1) \times r} & -\frac{N_m}{2} \boldsymbol{D}^\top \hat{\boldsymbol{F}}_m \boldsymbol{D} \end{bmatrix}. \tag{B.14}$$

The determinant of the FIM, $\hat{\boldsymbol{J}}_m$, can be written as

$$\left| \hat{\boldsymbol{J}}_m \right| = \left| N_m \hat{\boldsymbol{\Sigma}}_m^{-1} \right| \times \left| -\frac{N_m}{2} \boldsymbol{D}^\top \hat{\boldsymbol{F}}_m \boldsymbol{D} \right|. \tag{B.15}$$

As $N \to \infty$, $N_m \to \infty$ given that $l << N$, it follows that

$$\left| \frac{1}{N_m} \hat{\boldsymbol{J}}_m \right| \approx \mathcal{O}(1), \tag{B.16}$$

where $\mathcal{O}(1)$ denotes Landau's term which tends to a constant as $N \to \infty$. Using this result, we provide an asymptotic approximation to (2.17), in the case where $\mathcal{X}$ is composed of Gaussian distributed data vectors, as follows:

$$\begin{aligned} \log p(M_l|\mathcal{X}) \approx \log p(M_l) + \log f(\hat{\boldsymbol{\Theta}}_l|M_l) + \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) + \frac{lq}{2} \log 2\pi \\ - \frac{1}{2} \sum_{m=1}^{l} \log \left| N_m \frac{1}{N_m} \hat{\boldsymbol{J}}_m \right| - \log f(\mathcal{X}) \end{aligned}$$

$$= \log p(M_l) + \log f(\hat{\boldsymbol{\Theta}}_l|M_l) + \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) + \frac{lq}{2}\log 2\pi$$
$$- \frac{q}{2}\sum_{m=1}^{l}\log N_m - \frac{1}{2}\sum_{m=1}^{l}\log\left|\frac{1}{N_m}\hat{\boldsymbol{J}}_m\right| - \log f(\mathcal{X}), \tag{B.17}$$

where $q = \frac{1}{2}r(r+3)$.

Assume that

(A-2.6) $p(M_l)$ and $f(\hat{\boldsymbol{\Theta}}_l|M_l)$ are independent of the data length N.

Then, ignoring the terms in (B.17) that do not grow as $N \to \infty$ results in

$$\mathrm{BIC}_{\mathrm{N}}(M_l) \triangleq \log p(M_l|\mathcal{X}) \approx \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) - \frac{q}{2}\sum_{m=1}^{l}\log N_m - \log f(\mathcal{X}). \tag{B.18}$$

Since $\mathcal{X}$ is composed of multivariate Gaussian distributed data, $\mathrm{BIC}_{\mathrm{N}}(M_l)$ can be further simplified as follows:

$$\begin{aligned}\mathrm{BIC}_{\mathrm{N}}(M_l) &= \log \mathcal{L}(\hat{\boldsymbol{\Theta}}_l|\mathcal{X}) + p_l \\ &= \sum_{m=1}^{l}\left(N_m \log\frac{N_m}{N} - \frac{rN_m}{2}\log 2\pi - \frac{N_m}{2}\log|\hat{\boldsymbol{\Sigma}}_m| - \frac{1}{2}\mathrm{Tr}\left(N_m\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\Sigma}}_m\right)\right) + p_l \\ &= \sum_{m=1}^{l} N_m \log N_m - N\log N - \frac{rN}{2}\log 2\pi - \sum_{m=1}^{l}\frac{N_m}{2}\log|\hat{\boldsymbol{\Sigma}}_m| - \frac{rN}{2} + p_l,\end{aligned} \tag{B.19}$$

where

$$p_l \triangleq -\frac{q}{2}\sum_{m=1}^{l}\log N_m - \log f(\mathcal{X}). \tag{B.20}$$

Finally, ignoring the model independent terms in (B.19) results in (2.19) which concludes the proof.

B.2 Proof of Theorem 3.1

Proving Theorem 3.1 requires finding an asymptotic approximation to $|\hat{\boldsymbol{J}}_m|$ in (2.17) and, consequently, deriving an expression for $\text{BIC}_{t_\nu}(M_l)$. We start the proof by taking the first derivative of the log-likelihood function, given by (A.6), with respect to $\boldsymbol{\theta}_m$, which results in

$$\begin{aligned}
\frac{d \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m} &= -\frac{N_m}{2}\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) - \frac{d}{d\boldsymbol{\theta}_m}\left(\text{Tr}\left(\frac{(\nu_m + r)}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m}\log\left(1 + \frac{\delta_n}{\nu_m}\right)\right)\right) \\
&= -\frac{N_m}{2}\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) - \text{Tr}\left(\frac{(\nu_m + r)}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m}\frac{d}{d\boldsymbol{\theta}_m}\left(\log\left(1 + \frac{\delta_n}{\nu_m}\right)\right)\right) \\
&= -\frac{N_m}{2}\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) - \text{Tr}\left(\frac{(\nu_m + r)}{2}\sum_{\boldsymbol{x}_n \in \mathcal{X}_m}\frac{1}{\nu_m + \delta_n}\frac{d\delta_n}{d\boldsymbol{\theta}_m}\right) \\
&= -\frac{N_m}{2}\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) - \frac{1}{2}\text{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \frac{d\delta_n}{d\boldsymbol{\theta}_m}\right),
\end{aligned} \tag{B.21}$$

where w_n is given by (A.9), $\tilde{\boldsymbol{x}}_n = \boldsymbol{x}_n - \boldsymbol{\mu}_m$, $\delta_n = \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\tilde{\boldsymbol{x}}_n$, and

$$\begin{aligned}
\frac{d\delta_n}{d\boldsymbol{\theta}_m} &= \frac{d}{d\boldsymbol{\theta}_m}\left(\tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\tilde{\boldsymbol{x}}_n\right) \\
&= -2\tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m} - \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}\tilde{\boldsymbol{x}}_n.
\end{aligned} \tag{B.22}$$

Substituting (B.22) into (B.21) results in

$$\begin{aligned}
\frac{d \log \mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m} &= -\frac{N_m}{2}\,\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) + \text{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) \\
&\quad + \frac{1}{2}\,\text{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}\tilde{\boldsymbol{x}}_n\right) \\
&= -\frac{N_m}{2}\,\text{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) + \text{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) \\
&\quad + \frac{1}{2}\,\text{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}\right)
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{N_m}{2}\operatorname{Tr}\left(\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\right) + \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\tilde{\boldsymbol{x}}_n^\top\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right)\\
&+ \frac{1}{2}\operatorname{Tr}\left(\boldsymbol{Z}_m\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}\right)\\
&= \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}(\boldsymbol{Z}_m - N_m\boldsymbol{\Psi}_m)\boldsymbol{\Psi}_m^{-1}\right) + \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\tilde{\boldsymbol{x}}_n^\top\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right),
\end{aligned}
\tag{B.23}
$$

where

$$
\boldsymbol{Z}_m = \sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\tilde{\boldsymbol{x}}_n\tilde{\boldsymbol{x}}_n^\top. \tag{B.24}
$$

Derivating (B.23), once again, with respect to $\boldsymbol{\theta}_m^\top$ results in

$$
\begin{aligned}
\frac{d^2\log\mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top} &= \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\frac{d\boldsymbol{\Psi}_m^{-1}}{d\boldsymbol{\theta}_m^\top}(\boldsymbol{Z}_m - N_m\boldsymbol{\Psi}_m)\boldsymbol{\Psi}_m^{-1}\right)\\
&+ \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}\left(\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\theta}_m^\top} - N_m\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m^\top}\right)\boldsymbol{\Psi}_m^{-1}\right)\\
&+ \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\boldsymbol{\Psi}_m^{-1}(\boldsymbol{Z}_m - N_m\boldsymbol{\Psi}_m)\frac{d\boldsymbol{\Psi}_m^{-1}}{d\boldsymbol{\theta}_m^\top}\right)\\
&+ \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{dw_n}{d\boldsymbol{\theta}_m^\top}\tilde{\boldsymbol{x}}_n^\top\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) - \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m^\top}\boldsymbol{\Psi}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right)\\
&+ \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\tilde{\boldsymbol{x}}_n^\top\frac{d\boldsymbol{\Psi}_m^{-1}}{d\boldsymbol{\theta}_m^\top}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right).
\end{aligned}
\tag{B.25}
$$

From (B.25), the Fisher information matrix of observations from the mth cluster is given by

$$
\begin{aligned}
\hat{\boldsymbol{J}}_m &= -\left.\frac{d^2\log\mathcal{L}(\boldsymbol{\theta}_m|\mathcal{X}_m)}{d\boldsymbol{\theta}_m d\boldsymbol{\theta}_m^\top}\right|_{\boldsymbol{\theta}_m=\hat{\boldsymbol{\theta}}_m}\\
&= -\frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\hat{\boldsymbol{\Psi}}_m^{-1}\left(\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\theta}_m^\top} - N_m\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m^\top}\right)\hat{\boldsymbol{\Psi}}_m^{-1}\right) - \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{dw_n}{d\boldsymbol{\theta}_m^\top}\tilde{\boldsymbol{x}}_n^\top\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right)\\
&+ \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right)
\end{aligned}
$$

$$= -\frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\theta}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\right) + \frac{N_m}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\theta}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\right)$$
$$- \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{dw_n}{d\boldsymbol{\theta}_m^\top}\tilde{\boldsymbol{x}}_n^\top\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right) + \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\theta}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\theta}_m}\right), \tag{B.26}$$

where w_n, $\tilde{\boldsymbol{x}}_n$, $\frac{dw_n}{d\boldsymbol{\theta}_m^\top}$, and $\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\theta}_m^\top}$ are also evaluated at $\hat{\boldsymbol{\theta}}_m$, but the hat is removed for ease of notation. Note that, evaluated at $\hat{\boldsymbol{\theta}}_m$, (B.25) reduces to (B.26) because

$$\hat{\boldsymbol{Z}}_m - N_m\hat{\boldsymbol{\Psi}}_m = 0$$
$$\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\tilde{\boldsymbol{x}}_n^\top = 0.$$

(B.26) can be written in a compact matrix form as

$$\hat{\boldsymbol{J}}_m = \begin{bmatrix} -\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\mu}_m\partial\boldsymbol{\mu}_m^\top} & -\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\mu}_m\partial\boldsymbol{\Psi}_m^\top} \\ -\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\Psi}_m\partial\boldsymbol{\mu}_m^\top} & -\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\Psi}_m\partial\boldsymbol{\Psi}_m^\top} \end{bmatrix}. \tag{B.27}$$

The individual elements of the block matrix in (B.27) are given by

$$\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\mu}_m\partial\boldsymbol{\mu}_m^\top} = \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{dw_n}{d\boldsymbol{\mu}_m^\top}\tilde{\boldsymbol{x}}_n^\top\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m}\right) - \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m} w_n\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m}\right) \tag{B.28}$$

$$\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\mu}_m\partial\boldsymbol{\Psi}_m^\top} = \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{dw_n}{d\boldsymbol{\Psi}_m^\top}\tilde{\boldsymbol{x}}_n^\top\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m}\right) \tag{B.29}$$

$$\frac{\partial^2\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial\boldsymbol{\Psi}_m\partial\boldsymbol{\Psi}_m^\top} = \frac{1}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\Psi}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\right) - \frac{N_m}{2}\operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m}\hat{\boldsymbol{\Psi}}_m^{-1}\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top}\hat{\boldsymbol{\Psi}}_m^{-1}\right), \tag{B.30}$$

where

$$\frac{dw_n}{d\boldsymbol{\mu}_m^\top} = \frac{d}{d\boldsymbol{\mu}_m^\top}\left(\frac{\nu_m + r}{\nu_m + \delta_n}\right)$$
$$= -\frac{(\nu_m + r)}{(\nu_m + \delta_n)^2}\frac{d\delta_n}{d\boldsymbol{\mu}_m^\top}$$

$$= \frac{2w_n^2}{\nu_m + r} \frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \in \mathbb{R}^{r \times 1} \tag{B.31}$$

$$\begin{aligned}
\frac{dw_n}{d\boldsymbol{\Psi}_m^\top} &= \frac{d}{d\boldsymbol{\Psi}_m^\top} \left(\frac{\nu_m + r}{\nu_m + \delta_n} \right) \\
&= -\frac{(\nu_m + r)}{(\nu_m + \delta_n)^2} \frac{d\delta_n}{d\boldsymbol{\Psi}_m^\top} \\
&= \frac{w_n^2}{\nu_m + r} \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \in \mathbb{R}^{r \times r}
\end{aligned} \tag{B.32}$$

$$\begin{aligned}
\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\Psi}_m^\top} &= \frac{d}{d\boldsymbol{\Psi}_m^\top} \left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \right) \\
&= \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{dw_n}{d\boldsymbol{\Psi}_m^\top} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \in \mathbb{R}^{r^2 \times r^2}.
\end{aligned} \tag{B.33}$$

Note that, due to the symmetry of the Fisher information matrix, the following holds:

$$\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m | \mathcal{X}_m)}{\partial \boldsymbol{\psi}_m \partial \boldsymbol{\mu}_m^\top} = \left(\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m | \mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\Psi}_m^\top} \right)^\top \tag{B.34}$$

Using (B.31)-(B.33), (B.28)-(B.30) can be simplified to

$$\begin{aligned}
\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m | \mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top} &= \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{dw_n}{d\boldsymbol{\mu}_m^\top} \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right) - \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right) \\
&= \frac{2}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \operatorname{Tr}\left(\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right) \\
&\quad - \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \operatorname{Tr}\left(\frac{d\boldsymbol{\mu}_m^\top}{d\boldsymbol{\mu}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right) \\
&= \frac{2}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} - \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n \hat{\boldsymbol{\Psi}}_m^{-1} \\
&= \frac{2}{\nu_m + r} \hat{\boldsymbol{\Psi}}_m^{-1} \left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \right) \hat{\boldsymbol{\Psi}}_m^{-1} - \hat{\boldsymbol{\Psi}}_m^{-1} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n
\end{aligned} \tag{B.35}$$

$$\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m | \mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\Psi}_m^\top} = \operatorname{Tr}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{dw_n}{d\boldsymbol{\Psi}_m^\top} \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right)$$

$$= \frac{1}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \operatorname{Tr}\left(\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \right)$$

$$= \frac{1}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \operatorname{vec}\left(\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \otimes \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right) \tag{B.36}$$

$$\begin{aligned}
\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m | \mathcal{X}_m)}{\partial \boldsymbol{\Psi}_m \partial \boldsymbol{\Psi}_m^\top} &= \frac{1}{2} \operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{Z}_m}{d\boldsymbol{\Psi}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \right) - \frac{N_m}{2} \operatorname{Tr}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \right) \\
&= \frac{1}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{Z}_m}{d\boldsymbol{\Psi}_m^\top} \right) \\
&\quad - \frac{N_m}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right) \\
&= \frac{1}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} \frac{dw_n}{d\boldsymbol{\Psi}_m^\top} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \right) \\
&\quad - \frac{N_m}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right) \\
&= \frac{1}{2(\nu_m + r)} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \\
&\quad \times \operatorname{vec}\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \right) \\
&\quad - \frac{N_m}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right) \\
&= \frac{1}{2(\nu_m + r)} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \\
&\quad \times \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \left(\tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right) \\
&\quad - \frac{N_m}{2} \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m} \right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \right) \operatorname{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{\Psi}_m^\top} \right).
\end{aligned} \tag{B.37}$$

The scatter matrix $\boldsymbol{\Psi}_m$, $m = 1, \ldots, l$, is a symmetric and positive definite matrix. Hence, $\operatorname{vec}(\boldsymbol{\Psi}_m) = \boldsymbol{D}\boldsymbol{u}_m$, where $\operatorname{vec}(\boldsymbol{\Psi}_m) \in \mathbb{R}^{r^2 \times 1}$ represents the stacking of the columns of $\boldsymbol{\Psi}_m$ into a long column vector, $\boldsymbol{D} \in \mathbb{R}^{r^2 \times \frac{1}{2}r(r+1)}$ denotes the duplication matrix, and $\boldsymbol{u}_m \in \mathbb{R}^{\frac{1}{2}r(r+1) \times 1}$ contains the unique elements of $\boldsymbol{\Psi}_m$ [Magnus & Neudecker, 2007, pp. 56–

57]. Taking the symmetry of the scatter matrix into account and replacing $\boldsymbol{\theta}_m$ by $\check{\boldsymbol{\theta}}_m = [\boldsymbol{\mu}_m, \boldsymbol{u}_m]^\top$, (B.36) and (B.37) simplify to

$$\begin{aligned}
\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top} &= \frac{1}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \text{vec}\left(\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m}\right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \otimes \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \text{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{u}_m^\top}\right) \\
&= \frac{1}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \text{vec}\left(\frac{d\boldsymbol{\mu}_m}{d\boldsymbol{\mu}_m}\right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \otimes \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D} \frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top} \\
&= \frac{1}{\nu_m + r} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \left(\hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1} \otimes \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D} \qquad \text{(B.38)} \\
\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} &= \frac{1}{2(\nu_m + r)} \text{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{u}_m}\right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \\
&\quad \times \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \left(\tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \text{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{u}_m^\top}\right) \\
&\quad - \frac{N_m}{2} \text{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{u}_m}\right)^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \text{vec}\left(\frac{d\boldsymbol{\Psi}_m}{d\boldsymbol{u}_m^\top}\right) \\
&= \frac{1}{2(\nu_m + r)} \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m}\right)^\top \boldsymbol{D}^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \\
&\quad \times \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \left(\tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D} \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top}\right) \\
&\quad - \frac{N_m}{2} \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m}\right)^\top \boldsymbol{D}^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D} \left(\frac{d\boldsymbol{u}_m}{d\boldsymbol{u}_m^\top}\right) \\
&= \frac{1}{2(\nu_m + r)} \boldsymbol{D}^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2 \left(\tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \otimes \hat{\boldsymbol{\Psi}}_m^{-1} \tilde{\boldsymbol{x}}_n \tilde{\boldsymbol{x}}_n^\top \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D} \\
&\quad - \frac{N_m}{2} \boldsymbol{D}^\top \left(\hat{\boldsymbol{\Psi}}_m^{-1} \otimes \hat{\boldsymbol{\Psi}}_m^{-1}\right) \boldsymbol{D}. \qquad \text{(B.39)}
\end{aligned}$$

In face of (B.35), (B.38), and (B.39) three normalization factors exist, which are $\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2$, $\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n$, and N_m. While the relationship between $\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2$ and $\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n$ is non-trivial, starting from (A.11) and doing straight forward calculations the authors in [Kent et al., 1994] showed that

$$\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n = N_m. \qquad \text{(B.40)}$$

As a result, we end up with only two normalization factors, namely $\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2$ and N_m.

Given that $l \ll N$, $N \to \infty$ indicates that $\epsilon \to \infty$, where $\epsilon = \max\left(\sum_{\boldsymbol{x}_n \in \mathcal{X}_m} w_n^2, N_m\right)$. Hence, as $N \to \infty$

$$\left|\frac{1}{\epsilon}\hat{\boldsymbol{J}}_m\right| \approx \mathcal{O}(1), \tag{B.41}$$

where $\mathcal{O}(1)$ denotes Landau's term which tends to a constant as $N \to \infty$. Using the result in (B.41), (2.17) can be simplified to

$$\begin{aligned}
\log p(M_l|\mathcal{X}) &\approx \log p(M_l) + \sum_{m=1}^{l} \log\left(f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)\right) \\
&\quad + \frac{lq}{2}\log 2\pi - \frac{1}{2}\sum_{m=1}^{l}\log\left|\epsilon\frac{\hat{\boldsymbol{J}}_m}{\epsilon}\right| - \log f(\mathcal{X}) \\
&= \log p(M_l) + \sum_{m=1}^{l} \log\left(f(\hat{\boldsymbol{\theta}}_m|M_l)\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)\right) \\
&\quad + \frac{lq}{2}\log 2\pi - \frac{q}{2}\sum_{m=1}^{l}\log\epsilon - \frac{1}{2}\sum_{m=1}^{l}\log\left|\frac{\hat{\boldsymbol{J}}_m}{\epsilon}\right| - \log f(\mathcal{X}),
\end{aligned} \tag{B.42}$$

where $q = \frac{1}{2}r(r+3)$ is the number of estimated parameters per cluster. Note that the value of q changed from $q = r(r+1)$ to $q = \frac{1}{2}r(r+3)$ because we estimate only the unique elements of the scatter matrix $\hat{\boldsymbol{\Psi}}_m$.

Assuming that (A-2.6) is satisfied and ignoring the terms in (B.42) that do not grow as $N \to \infty$ results in

$$\begin{aligned}
\text{BIC}_{t_\nu}(M_l) &\triangleq \log p(M_l|\mathcal{X}) \\
&\approx \sum_{m=1}^{l}\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m) - \frac{q}{2}\sum_{m=1}^{l}\log\epsilon - \log f(\mathcal{X}).
\end{aligned} \tag{B.43}$$

Substituting the expression of $\log\mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)$, given by (A.6), into (B.43) results in

$$\begin{aligned}
\text{BIC}_{t_\nu}(M_l) = &\sum_{m=1}^{l}\left(N_m\log\frac{N_m}{N} + N_m\log\frac{\Gamma\left((\nu_m+r)/2\right)}{\Gamma\left(\nu_m/2\right)(\pi\nu_m)^{r/2}} - \frac{N_m}{2}\log|\hat{\boldsymbol{\Psi}}_m|\right) \\
&- \sum_{m=1}^{l}\sum_{\boldsymbol{x}_n\in\mathcal{X}_m}\frac{(\nu_m+r)}{2}\log\left(1+\frac{\delta_n}{\nu_m}\right) - \frac{q}{2}\sum_{m=1}^{l}\log\epsilon - \log f(\mathcal{X})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{m=1}^{l} N_m \log N_m - N \log N - \sum_{m=1}^{l} \frac{N_m}{2} \log |\hat{\boldsymbol{\Psi}}_m| \\
&+ \sum_{m=1}^{l} N_m \log \frac{\Gamma\left((\nu_m + r)/2\right)}{\Gamma\left(\nu_m/2\right)(\pi\nu_m)^{r/2}} - \frac{1}{2} \sum_{m=1}^{l} \sum_{\boldsymbol{x}_n \in \mathcal{X}_m} (\nu_m + r) \log\left(1 + \frac{\delta_n}{\nu_m}\right) \\
&- \frac{q}{2} \sum_{m=1}^{l} \log \epsilon - \log f(\mathcal{X}). \qquad \text{(B.44)}
\end{aligned}
$$

Finally, ignoring the model independent terms in (B.44) results in (3.1). This concludes the proof.

C

Calculation of the Determinant of the Fisher Information Matrix

The Fisher information matrix, given by (B.27), is a block matrix and its determinant is calculated as

$$|\hat{\boldsymbol{J}}_m| = \left| -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top} + \frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top} \left(\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} \right)^{-1} \frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{\mu}_m^\top} \right| \times \left| -\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top} \right|, \tag{C.1}$$

where $\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{\mu}_m^\top}$, $\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{\mu}_m \partial \boldsymbol{u}_m^\top}$, and $\frac{\partial^2 \log \mathcal{L}(\hat{\boldsymbol{\theta}}_m|\mathcal{X}_m)}{\partial \boldsymbol{u}_m \partial \boldsymbol{u}_m^\top}$ are given by (B.35), (B.38), and (B.39), respectively.

D

Vector and Matrix Differentiation Rules

Part V uses the numerator layout of derivatives. Given that $y \in \mathbb{R}$, $\boldsymbol{y} \in \mathbb{R}^{p\times 1}$, $\boldsymbol{Y} \in \mathbb{R}^{p\times q}$, $\boldsymbol{x} \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, the numerator layout of derivatives states that

$$\frac{dy}{d\boldsymbol{x}} \in \mathbb{R}^{1\times m} \tag{D.1}$$

$$\frac{dy}{d\boldsymbol{X}} \in \mathbb{R}^{n\times m} \tag{D.2}$$

$$\frac{d\boldsymbol{x}}{dy} \in \mathbb{R}^{m\times 1} \tag{D.3}$$

$$\frac{d\boldsymbol{x}}{d\boldsymbol{y}} \in \mathbb{R}^{m\times p}. \tag{D.4}$$

In addition, we have used the following matrix and vector differentiation rules (see [Magnus & Neudecker, 2007] for details):

$$\frac{d}{d\boldsymbol{y}}\boldsymbol{y}^\top\boldsymbol{y} = 2\boldsymbol{y}^\top \tag{D.5}$$

$$\frac{d}{d\boldsymbol{Y}}\boldsymbol{Y}^{-1} = -\boldsymbol{Y}^{-1}\frac{d\boldsymbol{Y}}{d\boldsymbol{Y}}\boldsymbol{Y}^{-1} \tag{D.6}$$

$$\frac{d}{d\boldsymbol{Y}} \log |\boldsymbol{Y}| = \mathrm{Tr}\left(\boldsymbol{Y}^{-1} \frac{d\boldsymbol{Y}}{d\boldsymbol{Y}}\right) \tag{D.7}$$

We have also exploited properties of the trace (Tr) and vectorization (vec) operators. Given matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ with matching dimensions, the following hold true:

$$\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA}) \tag{D.8}$$

$$\frac{d}{d\boldsymbol{A}} \mathrm{Tr}\left(\boldsymbol{A}\right) = \mathrm{Tr}\left(\frac{d\boldsymbol{A}}{d\boldsymbol{A}}\right) \tag{D.9}$$

$$\frac{d\,\mathrm{Tr}(\boldsymbol{BA})}{d\boldsymbol{B}} = \mathrm{Tr}\left(\frac{d\boldsymbol{B}}{d\boldsymbol{B}}\boldsymbol{A}\right) = \mathrm{Tr}\left(\frac{d\boldsymbol{B}}{d\boldsymbol{B}}\right)\boldsymbol{A} = \boldsymbol{A} \tag{D.10}$$

$$\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{B}) = \mathrm{vec}(\boldsymbol{A})^\top \mathrm{vec}(\boldsymbol{B}) \tag{D.11}$$

$$\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{CDB}^\top) = \mathrm{vec}(\boldsymbol{A})^\top (\boldsymbol{B} \otimes \boldsymbol{C}) \mathrm{vec}(\boldsymbol{D}) \tag{D.12}$$

$$\mathrm{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^\top \otimes \boldsymbol{A}) \mathrm{vec}(\boldsymbol{B}) \tag{D.13}$$

List of Acronyms

ACF	aggregate channel features
ALR	average labeling rate
AMR	average mislabeling rate
ATC	adapt then combine
BIC	Bayesian information criterion
DX-means	distributed X-means
EM	expectation maximization
FAST-TLE	fast trimmed likelihood estimator
FIM	Fisher information matrix
GC	gravitational clustering
HOG	histogram of oriented gradients
HOT	higher order terms
iid	independent and identically distributed
ICT	information and communication technology
MAE	mean absolute error
MDMT	multiple devices multiple tasks
MNDL	minimum noiseless description length
NN	nearest neighbor
PCA	principal component analysis
pdf	probability density function
RGB	red, green, and blue
RSEM	random swap expectation maximization
RSK-means	random swap K-means
TBIC	trimmed Bayesian information criterion

List of Notation and Symbols

The following list contains the most important notation, operators, and symbols. The remaining once are introduced where they are used.

Notation

n	normal-font lowercase letter denotes a scalar
N	normal-font uppercase letter denotes a scalar
$\boldsymbol{x}$	boldface lowercase letter denotes a column vector
$\boldsymbol{X}$	boldface uppercase letter denotes a matrix
$\mathcal{X}$	calligraphic letter denotes a set
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of positive real numbers
$\mathbb{Z}^+$	set of positive integers
$\mathbb{R}^{r\times 1}$	set of column vectors of size r on $\mathbb{R}$
$\mathbb{R}^{r\times r}$	set of matrices of dimension $r \times r$ on $\mathbb{R}$
$\mathbb{1}_{\{\cdot\}}$	the indicator function
$\exp(\cdot)$	the exponential function
$\Gamma(\cdot)$	the gamma function
$f(\cdot)$	probability density function
$p(\cdot)$	probability mass function
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$t_\nu(\boldsymbol{\mu}, \boldsymbol{\Psi})$	t distribution with location parameter $\boldsymbol{\mu}$, scatter matrix $\boldsymbol{\Psi}$, and degree of freedom ν
(A.)	assumption
$\arg\max_x F(x)$	returns the value of x that maximizes $F(x)$
$\arg\min_x F(x)$	returns the value of x that minimizes $F(x)$

$\mathcal{O}(1)$	Landau's term which tends to a constant as the data size goes to infinity
$x \triangleq y$	x is defined as y
$x \equiv y$	x is equivalent to y
$x \sim y$	x is distributed as y

Operators

$(\cdot)^{-1}$	matrix inverse
$(\cdot)^\top$	vector or matrix transpose
$\lvert \cdot \rvert$	determinant of a matrix or the absolute value of a scalar
$\mathrm{Tr}(\cdot)$	trace of a matrix
$\mathrm{vec}(\cdot)$	vectorizes its argument by stacking the columns on top of each other
$\#\mathcal{X}$	cardinality of the set $\mathcal{X}$
$\mathcal{X}/\{x\}$	the set $\mathcal{X}$ without the element x
$\otimes$	Kronecker product
$\lvert\lvert \cdot \rvert\rvert_1$	l_1-norm
$\lvert\lvert \cdot \rvert\rvert_2$	l_2-norm
$\mathbb{E}$	expectation operator
$\log$	natural logarithm
$\lim$	limit

Symbols

$\mathbf{0}_{r\times r}$	all zero matrix of dimension $r \times r$
$\mathbf{1}_{r\times 1}$	all one column vector of size r
$\boldsymbol{I}_r$	identity matrix of dimension $r \times r$
$\mathcal{L}$	the likelihood function
$\hat{\boldsymbol{\theta}}$	estimator (or estimate) of the parameter $\boldsymbol{\theta}$
$\boldsymbol{\mu}_m$	centroid of the mth cluster
$\boldsymbol{\Sigma}_m$	covariance matrix of the mth cluster
$\boldsymbol{\Psi}_m$	scatter matrix of the mth cluster
ν_m	degree of freedom of the mth cluster
$\hat{\boldsymbol{J}}_m$	Fisher information matrix of the data points in the mth cluster

K	true number of clusters in the data set
N_m	number of data points in the mth cluster
N	total number of data points in the data set
$\mathcal{M}$	set of candidate models
M_l	candidate model that clusters the data set into l clusters
$L_{\min}$	specified minimum number of clusters
$L_{\max}$	specified maximum number of clusters
ALR^{net}	network-wide average labeling rate
AMR^{net}	network-wide average mislabeling rate
MAE^{net}	network-wide mean absolute error
BIC_{G}	generic BIC which is applicable to a broad class of data distributions
BIC_{N}	asymptotic BIC derived by modeling the data as a family of Gaussian distributions
BIC_{NF}	extension of BIC_{N} with an exact computation of its penalty term
BIC_{t_ν}	asymptotic BIC derived by modeling the data as a family of t_ν distributions
$\text{BIC}_{\text{F}t_\nu}$	extension of BIC_{t_ν} with an exact computation of its penalty term
BIC_{O}	the original BIC which models the data as a family of Gaussian distributions
BIC_{OS}	the original BIC which models the data as a family of Gaussian distributions with the constraint that the clusters are identical and spherical
$\text{BIC}_{\text{O}t_\nu}$	the original BIC which models the data as a family of t_ν distributions
D-BIC_{N}	distributed implementation of BIC_{N}
D-BIC_{NF}	distributed implementation of BIC_{NF}
$\hat{K}_{\text{BIC}_{\text{G}}}$	number of clusters estimated using BIC_{G}
$\hat{K}_{\text{BIC}_{\text{N}}}$	number of clusters estimated using BIC_{N}
$\hat{K}_{\text{BIC}_{\text{NF}}}$	number of clusters estimated using BIC_{NF}
$\hat{K}_{\text{BIC}_{t_\nu}}$	number of clusters estimated using BIC_{t_ν}
$\hat{K}_{\text{BIC}_{\text{F}t\nu}}$	number of clusters estimated using $\text{BIC}_{\text{F}t_\nu}$
$\hat{K}_{\text{BIC}_{\text{O}}}$	number of clusters estimated using BIC_{O}
$\hat{K}_{\text{BIC}_{\text{OS}}}$	number of clusters estimated using BIC_{OS}

p_{det}	empirical probability of detection
$p_{\text{det}}^{\text{net}}$	network-wide empirical probability of detection
p_{under}	empirical probability of underestimation
p_{over}	empirical probability of overestimation

References

[Akaike, 1969] H. Akaike. "Fitting autoregressive models for prediction". In: *Annals of the Institute of Statistical Mathematics* 21.1 (1969), pp. 243–247 (cited on page 10).

[Akaike, 1970] H. Akaike. "Statistical predictor identification". In: *Annals of the Institute of Statistical Mathematics* 22.1 (1970), pp. 209–217 (cited on page 10).

[Akaike, 1973] H. Akaike. "Information theory and an extension of the maximum likelihood principle". In: *Petrov, B.N. & Csaki, F., Eds., 2nd International Symposium on Information Theory*. Budapest, Hungary, 1973, pp. 267–281 (cited on page 10).

[Allen, 1974] D. M. Allen. "The relationship between variable selection and data augmentation and a method for prediction". In: *Technometrics* 16.1 (1974), pp. 125–127 (cited on page 10).

[Amin, 2010] M. G. Amin, ed. *Through-the-Wall Radar Imaging*. CRC Press, 2010 (cited on page 115).

[Amin, 2017] M. G. Amin, ed. *Radar for Indoor Monitoring: Detection, Classification, and Assessment*. CRC Press, 2017 (cited on page 115).

[Ancortek Inc, 2018] Ancortek Inc. *SDR-KIT 2400AD*. http://ancortek.com/sdr-kit-2400ad, retrieved: 02/20/2018 (cited on page 116).

[Ando, 2010] T. Ando. *Bayesian Model Selection and Statistical Modeling*. Chapman & Hall/CRC, 2010, pp. 51–53 (cited on pages 15, 17).

[Andrews & McNicholas, 2012] J. L. Andrews & P. D. McNicholas. "Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions". In: *Statistics and*

Computing 22.5 (2012), pp. 1021–1029 (cited on pages 56, 62).

[Arbelaitz et al., 2013] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez & I. Perona. "An extensive comparative study of cluster validity indices". In: *Pattern Recognition* 46.1 (2013), pp. 243–256 (cited on page 10).

[Arthur & Vassilvitskii, 2007] D. Arthur & S. Vassilvitskii. "K-means++: the advantages of careful seeding". In: *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*. New Orleans, USA, 2007, pp. 1027–1035 (cited on pages 2, 22, 23, 29, 94, 131).

[Azzalini & Bowman, 1990] A. Azzalini & A. W. Bowman. "A look at some data on the Old Faithful geyser". In: *Applied Statistics* 39.3 (1990), pp. 357–365 (cited on page 71).

[Bahari et al., 2016] M. H. Bahari, J. Plata-Chaves, A. Bertrand & M. Moonen. "Distributed labelling of audio sources in wireless acoustic sensor networks using consensus and matching". In: *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*. Budapest, Hungary, 2016, pp. 2345–2349 (cited on pages 43, 78).

[Bay et al., 2008] H. Bay, A. Ess, T. Tuytelaars & L. Van Gool. "Speeded-Up Robust Features (SURF)". in: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359 (cited on page 35).

[Berclaz et al., 2011] J. Berclaz, F. Fleuret, E. Türetken & P. Fua. "Multiple object tracking using K-shortest paths optimization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.9 (2011), pp. 1806–1819 (cited on pages 51, 52, 79, 80, 93).

[Bertrand & Moonen, 2012] A. Bertrand & M. Moonen. "Distributed signal estimation in sensor networks where nodes have different interests". In: *Signal Processing* 92.7 (2012), pp. 1679–1690 (cited on pages 42, 77).

[Bezdek, 1981] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publish-

ers, 1981 (cited on page 108).

[Binder et al., 2015] P. Binder, M. Muma & A. M. Zoubir. "Robust and computationally efficient diffusion-based classification in distributed networks". In: *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia, 2015, pp. 3432–3436 (cited on pages 43, 78–80, 82).

[Binder et al., 2016] P. Binder, M. Muma & A. M. Zoubir. "Robust and adaptive diffusion-based classification in distributed networks". In: *EURASIP Journal on Advances in Signal Processing* 2016.34 (2016), pp. 1–13 (cited on pages 43, 50, 78–80, 82).

[Binder et al., 2018] P. Binder, M. Muma & A. M. Zoubir. "Gravitational clustering: a simple, robust and adaptive approach for distributed networks". In: *Signal Processing* 149 (2018), pp. 36–48 (cited on pages 10, 31, 34, 43, 56, 64).

[Bishop, 2006] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, 2006 (cited on pages 20, 71).

[Blömer & Bujna, 2016] J. Blömer & K. Bujna. "Adaptive seeding for Gaussian mixture models". In: *Proceedings, Part II, of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume 9652*. Auckland, New Zealand, 2016, pp. 296–308 (cited on page 131).

[Bogdanovic et al., 2014] N. Bogdanovic, J. Plata-Chaves & K. Berberidis. "Distributed incremental-based LMS for node-specific adaptive parameter estimation". In: *IEEE Transactions on Signal Processing* 62.20 (2014), pp. 5382–5397 (cited on pages 42, 77).

[Boley, 1998] D. Boley. "Principal direction division partitioning". In: *Data Mining and Knowledge Discovery* 2.4 (1998), pp. 325–344 (cited on page 108).

[Brcich et al., 2002] R. F. Brcich, A. M. Zoubir & P. Pelin. "Detection of sources using bootstrap techniques". In: *IEEE Trans-*

actions on Signal Processing 50.2 (2002), pp. 206–215 (cited on page 10).

[Breiman, 1992] L. Breiman. "The little bootstrap and other methods for dimensionality selection in regression: x-fixed prediction error". In: *Journal of the American Statistical Association* 87.419 (1992), pp. 738–754 (cited on page 10).

[Brooks et al., 2003] R. R. Brooks, P. Ramanathan & A. M. Sayeed. "Distributed target classification and tracking in sensor networks". In: *Proceedings of the IEEE* 91.8 (2003), pp. 1163–1171 (cited on page 78).

[Caliński & Harabasz, 1974] T. Caliński & J. Harabasz. "A dendrite method for cluster analysis". In: *Communications in Statistics* 3.1 (1974), pp. 1–27 (cited on page 10).

[Campbell et al., 1997] J. G. Campbell, C. Fraley, F. Murtagh & A. E. Raftery. "Linear flaw detection in woven textiles using model-based clustering". In: *Pattern Recognition Letters* 18.14 (1997), pp. 1539–1548 (cited on page 10).

[Cattivelli et al., 2008] F. S. Cattivelli, C. G. Lopes & A. H. Sayed. "Diffusion recursive least-squares for distributed estimation over adaptive networks". In: *IEEE Transactions on Signal Processing* 56.5 (2008), pp. 1865–1877 (cited on page 77).

[Cavanaugh & Neath, 1999] J. E. Cavanaugh & A. A. Neath. "Generalizing the derivation of the Schwarz information criterion". In: *Communication in Statistics - Theory and Methods* 28.1 (1999), pp. 49–66 (cited on pages 10, 11, 19, 21, 53, 56, 57, 64).

[Chen et al., 2015] J. Chen, C. Richard & A. H. Sayed. "Diffusion LMS over multitask networks". In: *IEEE Transactions on Signal Processing* 63.11 (2015), pp. 2733–2748 (cited on pages 43, 77).

[Chen et al., 2014] V. C. Chen, D. Tahmoush & W. J. Miceli. *Radar Micro-Doppler Signature: Processing and Applications*. The Institution of Engineering and Technology, 2014 (cited on page 115).

[Chou et al., 2003] J. Chou, D. Petrovic & K. Ramchandran. "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks". In: *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*. San Francisco, USA, 2003, pp. 1054–1062 (cited on page 77).

[Chouvardas et al., 2015] S. Chouvardas, M. Muma, L. K. Hamaidi, S. Theodoridis & A. M. Zoubir. "Distributed robust labeling of audio sources in heterogeneous wireless sensor networks". In: *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia, 2015, pp. 5783–5787 (cited on pages 43, 77).

[Claeskens & Hjort, 2003] G. Claeskens & N. L. Hjort. "The Focused information criterion". In: *Journal of the American Statistical Association* 98.464 (2003), pp. 900–916 (cited on page 10).

[Constantinopoulos et al., 2006] C. Constantinopoulos, M. K. Titsias & A. Likas. "Bayesian feature and model selection for Gaussian mixture models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.6 (2006), pp. 1013–1018 (cited on page 10).

[Dalal & Triggs, 2005] N. Dalal & B. Triggs. "Histograms of oriented gradients for human detection". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. San Diego, USA, 2005, pp. 886–893 (cited on pages 79, 87).

[Dasgupta & Raftery, 1998] A. Dasgupta & A. E. Raftery. "Detecting features in spatial point processes with clutter via model-based clustering". In: *Journal of the American Statistical Association* 93.441 (1998), pp. 294–302 (cited on pages 10, 56).

[Davé & Krishnapuram, 1997] R. N. Davé & R. Krishnapuram. "Robust clustering methods: a unified view". In: *IEEE Transactions on Fuzzy Systems* 5.2 (1997), pp. 270–293 (cited on pages 3, 4, 55).

[Davies & Bouldin, 1979] D. L. Davies & D. W. Bouldin. "A cluster separation measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227 (cited on page 10).

[Defays, 1977] D. Defays. "An efficient algorithm for a complete link method". In: *The Computer Journal* 20.4 (1977), pp. 364–366 (cited on page 108).

[Dempster et al., 1977] A. P. Dempster, N.-. M. Laird & D. B. Rubin. "Maximum likelihood for incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society Series B* 39.1 (1977), pp. 1–38 (cited on pages 2, 20, 108).

[Djurić, 1998] P. M. Djurić. "Asymptotic MAP criteria for model selection". In: *IEEE Transactions on Signal Processing* 46.10 (1998), pp. 2726–2735 (cited on pages 10, 11, 15, 126).

[Dolatabadi et al., 2017] E. Dolatabadi, A. Mansfield, K. K. Patterson, B. Taati & A. Mihailidis. "Mixture-model clustering of pathological gait patterns". In: *IEEE Journal on Biomedical and Health Informatics* 21.5 (2017), pp. 1297–1305 (cited on page 10).

[Dollár et al., 2014] P. Dollár, R. Appel, S. J. Belongie & P. Perona. "Fast feature pyramids for object detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), pp. 1532–1545 (cited on page 97).

[Du & Piater, 2007] W. Du & J. Piater. "Multi-camera people tracking by collaborative particle filters and principal axis-based integration". In: *Proceedings, Part I, of the 8th Asian Conference on Computer Vision*. Tokyo, Japan, 2007, pp. 365–374 (cited on pages 79, 80).

[Dunn, 1973] J. C. Dunn. "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters". In: *Journal of Cybernetics* 3.3 (1973), pp. 32–57 (cited on page 10).

[Everitt, 2011] B. Everitt. *Cluster Analysis*. Wiley, 2011 (cited on page 108).

[Feng & Hamerly, 2007] Y. Feng & G. Hamerly. "PG-means: learning the number of clusters in data". In: *Advances in Neural Information Processing Systems 19*. 2007, pp. 393–400 (cited on pages 10, 11).

[Fisher, 1936] R. A. Fisher. "The use of multiple measurements in taxonomic problems". In: *The Annals of Eugenics* 7 (1936), pp. 179–188 (cited on pages 2, 31).

[Fleuret et al., 2008] F. Fleuret, J. Berclaz, R. Lengagne & P. Fua. "Multicamera people tracking with a probabilistic occupancy map". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 267–282 (cited on pages 51, 52, 79, 80, 93).

[Forero et al., 2011] P. A. Forero, A. Cano & G. B. Giannakis. "Distributed clustering using wireless sensor networks". In: *IEEE Journal of Selected Topics in Signal Processing* 5.4 (2011), pp. 707–724 (cited on pages 78, 79).

[Fraley & Raftery, 1998] C. Fraley & A. Raftery. "How many clusters? Which clustering method? Answers via model-based cluster analysis". In: *The Computer Journal* 41.8 (1998), pp. 578–588 (cited on pages 10, 56).

[Fränti, 2018] P. Fränti. "Efficiency of random swap clustering". In: *Journal of Big Data* 5.13 (2018), pp. 1–29 (cited on pages 29, 131).

[Fränti et al., 2016] P. Fränti, R. Mariescu-Istodor & C. Zhong. "XNN graph". In: *Joint International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. Mérida, Mexico, 2016, pp. 207–217 (cited on pages 26, 30).

[Fränti & Virmajoki, 2006] P. Fränti & O. Virmajoki. "Iterative shrinking method for clustering problems". In: *Pattern Recognition* 39.5 (2006), pp. 761–765 (cited on pages 26, 30).

[Frigui & Krishnapuram, 1996] H. Frigui & R. Krishnapuram. "A robust algorithm for automatic extraction of an unknown number of clusters from noisy data". In: *Pattern Recognition Letters* 17.12 (1996), pp. 1223–1232 (cited on page 56).

[Gallegos & Ritter, 2005] M. T. Gallegos & G. Ritter. "A robust method for cluster analysis". In: *The Annals of Statistics* 33.1 (2005), pp. 347–380 (cited on pages 4, 55).

[Gallegos & Ritter, 2009] M. T. Gallegos & G. Ritter. "Trimming algorithms for clustering contaminated grouped data and their robustness". In: *Advances in Data Analysis and Classification* 3.2 (2009), pp. 135–167 (cited on page 56).

[Gallegos & Ritter, 2010] M. T. Gallegos & G. Ritter. "Using combinatorial optimization in model-based trimmed clustering with cardinality constraints". In: *Computational Statistics & Data Analysis* 54.3 (2010), pp. 637–654 (cited on page 56).

[Garcá-Escudero et al., 2011] L. A. Garcá-Escudero, A. Gordaliza, C. Martrán & A. Mayo-Iscar. "Exploring the number of groups in robust model-based clustering". In: *Statistics and Computing* 21.4 (2011), pp. 585–599 (cited on pages 4, 55, 56).

[García-Escudero et al., 2010] L. A. García-Escudero, A. Gordaliza, C. Matrán & A. Mayo-Iscar. "A review of robust clustering methods". In: *Advances in Data Analysis and Classification* 4.2–3 (2010), pp. 89–109 (cited on page 56).

[Garreau et al., 2011] G. Garreau, C. M. Andreou, A. G. Andreou, J. Georgiou, S. Dura-Bernal, T. Wennekers & S. Denham. "Gait-based person and gender recognition using micro-Doppler signatures". In: *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS)*. San Diego, USA, 2011, pp. 444–447 (cited on page 115).

[Hai et al., 2012] M. Hai, S. Zhang, L. Zhu & Y. Wang. "A survey of distributed clustering algorithms". In: *Proceedings of the International Conference on Industrial Control and Electronics Engineering (ICICEE)*. Xi'an, China, 2012, pp. 1142–1145 (cited on page 78).

[Halkidi et al., 2001] M. Halkidi, Y. Batistakis & M. Vazirgiannis. "On clustering validation techniques". In: *Journal of Intelligent Information Systems* 17.2/3 (2001), pp. 107–145 (cited on page 10).

[Hamerly & Charles, 2003] G. Hamerly & E. Charles. "Learning the K in K-means". In: *Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS)*. Whistler, Canada, 2003, pp. 281–288 (cited on pages 10, 11).

[Hannan & Quinn, 1979] E. J. Hannan & B. G. Quinn. "The determination of the order of an autoregression". In: *Journal of the Royal Statistical Society Series B* 41.2 (1979), pp. 190–195 (cited on page 10).

[Hartigan & Wong, 1979] J. A. Hartigan & M. A. Wong. "A K-means clustering algorithm". In: *Journal of the Royal Statistical Society Series C* 28.1 (1979), pp. 100–108 (cited on page 108).

[Hennig, 2003] C. Hennig. "Clusters, outliers, and regression: fixed point clusters". In: *Journal of Multivariate Analysis* 86.1 (2003), pp. 183–212 (cited on page 71).

[Hu et al., 2011] Y. Hu, C. Zou, Y. Yang & F. Qu. "A robust cluster validity index for fuzzy c-means clustering". In: *Proceedings of the International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*. Changchun, China, 2011, pp. 448–451 (cited on page 56).

[Huang et al., 2017] T. Huang, H. Peng & K. Zhang. "Model selection for Gaussian mixture models". In: *Statistica Sinica* 27.1 (2017), pp. 147–169 (cited on page 10).

[Ishioka, 2005] T. Ishioka. "An expansion of X-means for automatically determining the optimal number of clusters". In: *Proceedings of the 4th IASTED International Conference on Computational Intelligence*. Calgary, Canada, 2005, pp. 91–96 (cited on page 10).

[Izenman, 2008] A. J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Science+Business Media, LLC, 2008 (cited on page 71).

[Jeffreys, 1961] H. Jeffreys. *The Theory of Probability*. Oxford University Press, 1961 (cited on page 10).

[Kalogeratos & Likas, 2012] A. Kalogeratos & A. Likas. "Dip-means: an incremental clustering method for estimating the number of clusters". In: *Advances in Neural Information Processing Systems 25*. 2012, pp. 2402–2410 (cited on page 10).

[Kang et al., 2004] J. Kang, I. Cohen & G. Medioni. "Tracking people in crowded scenes across multiple cameras". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. Jeju Island, Korea, 2004 (cited on pages 79, 80).

[Kärkkäinen & Fränti, 2002] I. Kärkkäinen & P. Fränti. *Dynamic local search algorithm for the clustering problem*. Tech. rep. A-2002-6. Joensuu, Finland: Department of Computer Science, University of Joensuu, 2002 (cited on pages 26, 30).

[Karypis et al., 1999] G. Karypis, E.-H. Han & V. Kumar. "Chameleon: hierarchical clustering using dynamic modeling". In: *Computer* 32.8 (1999), pp. 68–75 (cited on page 108).

[Kass & Raftery, 1995] R. E. Kass & A. E. Raftery. "Bayes factors". In: *Journal of the American Statistical Association* 90.430 (1995), pp. 773–795 (cited on page 10).

[Kaufman & Rousseeuw, 1990] L. Kaufman & P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc, 1990 (cited on pages 3, 108).

[Kaufman & Rousseeuw, 1987] L. Kaufman & P. Rousseeuw. "Clustering by means of medoids". In: *Statistical Data Analysis Based on the ℓ_1–Norm and Related Methods* (1987), pp. 405–416 (cited on page 108).

[Kaustubh & Bhiksha, 2007] K. Kaustubh & R. Bhiksha. "Acoustic Doppler sonar for gait recognition". In: *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*. London, UK, 2007, pp. 27–32 (cited on page 115).

[Kent et al., 1994] J. T. Kent, D. E. Tyler & Y. Vard. "A curious likelihood identity for the multivariate t-distribution". In: *Communications in Statistics - Simulations and Computation* 23.2 (1994), pp. 441–453 (cited on pages 56, 58, 149).

[Khan & Shah, 2006] S. Khan & M. Shah. "A multiview approach to tracking people in crowded scenes using a planar homography constraint". In: *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*. Graz, Austria, 2006, pp. 133–146 (cited on pages 79, 80).

[Kibria & Joarder, 2005] B. Kibria & A. Joarder. "A short review of multivariate t-distribution". In: *Journal of Statistical Research* 40.1 (2006), pp. 59–72 (cited on page 56).

[King, 1967] B. King. "Step-wise clustering procedures". In: *Journal of the American Statistical Association* 62.317 (1967) (cited on page 108).

[King, 2015] R. S. King. *Cluster Analysis and Data Mining: An Introduction*. Mercury Learning and Information, 2015 (cited on page 3).

[Klawonn et al., 2015] F. Klawonn, F. Höppner & B. Jayaram. "What are clusters in high dimensions and are they difficult to Find?" In: *Revised Selected Papers of the 1st International Workshop on Clustering High–Dimensional Data - Volume 7627*. Springer-Verlag Berlin Heidelberg, 2015, pp. 14–33 (cited on page 131).

[Kokiopoulou & Frossard, 2011] E. Kokiopoulou & P. Frossard. "Distributed classification of multiple observation sets by consensus". In: *IEEE Transactions on Signal Processing* 59.1 (2011), pp. 104–114 (cited on page 78).

[Kotz & Nadarajah, 2004] S. Kotz & S. Nadarajah. *Multivariate t Distributions and Their Applications*. Cambridge university press, 2004 (cited on pages 56, 60).

[Krzanowski & Lai, 1988] W. J. Krzanowski & Y. T. Lai. "A criterion for determining the number of groups in a data set using sum-of-squares clustering". In: *Biometrics* 44.1 (1988), pp. 23–34 (cited on page 10).

[Lange et al., 1989] K. L. Lange, R. J. A. Little & J. M. G. Taylor. "Robust statistical modeling using the t distribution". In: *Journal of the American Statistical Association* 84.408 (1989), pp. 881–896 (cited on page 56).

[Li & Fang, 2007] H. Li & J. Fang. "Distributed adaptive quantization and estimation for wireless sensor networks". In: *IEEE Signal Processing Letters* 14.10 (2007), pp. 669–672 (cited on page 77).

[Lichman, 2013] M. Lichman. *UCI Machine Learning Repository*. 2013. ONLINE: http://archive.ics.uci.edu/ml (cited on pages 2, 30, 31, 33, 34).

[Liu & Rubin, 1995] C. Liu & D. Rubin. "ML estimation of the t distribution using EM and its extensions, ECM and ECME". in: *Statistica Sinica* 5 (1995), pp. 19–39 (cited on pages 56, 58).

[Lloyd, 1982] S. Lloyd. "Least squares quantization in PCM". in: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137 (cited on pages 2, 108).

[Lopes & Sayed, 2007] C. G. Lopes & A. H. Sayed. "Incremental adaptive strategies over distributed networks". In: *IEEE Transactions on Signal Processing* 55.8 (2007), pp. 4064–4077 (cited on page 77).

[Lorenzo et al., 2017] P. D. Lorenzo, E. Isufi, P. Banelli, S. Barbarossa & G. Leus. "Distributed recursive least squares strategies for adaptive reconstruction of graph signals". In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece, 2017, pp. 1–5 (cited on page 77).

[Lu & Zoubir, 2013a] Z. Lu & A. M. Zoubir. "Flexible detection criterion for source enumeration in array processing". In: *IEEE Transactions on Signal Processing* 61.6 (2013), pp. 1303–1314 (cited on page 10).

[Lu & Zoubir, 2013b] Z. Lu & A. M. Zoubir. "Generalized Bayesian information criterion for source enumeration in array processing". In: *IEEE Transactions on Signal Processing* 61.6 (2013), pp. 1470–1480 (cited on page 10).

[Lu & Zoubir, 2015] Z. Lu & A. M. Zoubir. "Source enumeration in array processing using a two-step test". In: *IEEE Transac-*

tions on Signal Processing 63.10 (2015), pp. 2718–2727 (cited on page 10).

[Magnus & Neudecker, 1980] J. R. Magnus & H. Neudecker. "The elimination matrix: some lemmas and applications". In: *SIAM Journal on Algebraic and Discrete Methods* 1.4 (1980), pp. 422–449 (cited on pages 38, 46).

[Magnus & Neudecker, 2007] J. R. Magnus & H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons Ltd, 2007 (cited on pages 140, 141, 148, 155).

[Malhotra et al., 2008] B. Malhotra, I. Nikolaidis & J. Harms. "Distributed classification of acoustic targets in wireless audio-sensor networks". In: *Computer Networks* 52.13 (2008), pp. 2582–2593 (cited on page 78).

[Maronna, 1976] R. A. Maronna. "Robust M-estimators of multivariate location and scatter". In: *The Annals of Statistics* 4.1 (1976), pp. 51–67 (cited on page 58).

[Maulik & Bandyopadhyay, 2002] U. Maulik & S. Bandyopadhyay. "Performance evaluation of some clustering algorithms and validity indices". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.12 (2002), pp. 1650–1654 (cited on page 10).

[McLachlan & Peel, 1998] G. J. McLachlan & D. Peel. "Robust cluster analysis via mixtures of multivariate t-distributions". In: *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*. London, UK: Springer-Verlag, 1998, pp. 658–666 (cited on pages 56, 60).

[McNicholas & Subedi, 2012] P. D. McNicholas & S. Subedi. "Clustering gene expression time course data using mixtures of multivariate t-distributions". In: *Journal of Statistical Planning and Inference* 142.5 (2012), pp. 1114–1127 (cited on pages 56, 62).

[Mehrjou et al., 2016] A. Mehrjou, R. Hosseini & B. N. Araabi. "Improved Bayesian information criterion for mixture model se-

lection". In: *Pattern Recognition Letters* 69 (2016), pp. 22–27 (cited on pages 10, 11).

[Milligan & Cooper, 1985] G. W. Milligan & M. C. Cooper. "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2 (1985), pp. 159–179 (cited on page 10).

[Minka, 2001] T. P. Minka. "Automatic choice of dimensionality for PCA". in: *Advances in Neural Information Processing Systems*. 2001, pp. 598–604 (cited on page 117).

[Mokhtari et al., 2017] G. Mokhtari, Q. Zhang, C. Hargrave & J. C. Ralston. "Non-wearable UWB sensor for human identification in smart home". In: *IEEE Sensors Journal* 17.11 (2017), pp. 3332–3340 (cited on page 115).

[Morariu & Camps, 2006] V. I. Morariu & O. I. Camps. "Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target daynamics". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006, pp. 545–552 (cited on pages 79, 80).

[Morelande & Zoubir, 2002] M. R. Morelande & A. M. Zoubir. "Model selection of random amplitude polynomial phase signals". In: *IEEE Transactions on Signal Processing* 50.3 (2002), pp. 578–589 (cited on page 10).

[Mukherjee et al., 1998] S. Mukherjee, E. D. Feigelson, G. J. Babu, F. Murtagh, C. Fraley & A. Raftery. "Three types of Gamma-ray bursts". In: *Astrophysical Journal* 508.1 (1998), pp. 314–327 (cited on page 10).

[Munkres, 1957] J. Munkres. "Algorithms for the assignment and transportation problems". In: *Journal of the Society for Industrial and Applied Mathematics* 5.1 (1957), pp. 32–38 (cited on pages 45, 95, 96).

[Murtagh, 1983] F. Murtagh. "A survey of recent advances in hierarchical clustering algorithms". In: *The Computer Journal* 26.4 (1983), pp. 354–359 (cited on page 108).

[Nadarajah & Kotz, 2008] S. Nadarajah & S. Kotz. "Estimation methods for the multivariate t distribution". In: *Acta Applicandae Mathematicae* 102.1 (2008), pp. 99–118 (cited on page 60).

[Neykov et al., 2007] N. Neykov, P. Filzmoser, R. Dimova & P. Neytchev. "Robust fitting of mixtures using the trimmed likelihood estimator". In: *Computational Statistics & Data Analysis* 52.1 (2007), pp. 299–308 (cited on pages 56, 64).

[Nowak, 2003] R. D. Nowak. "Distributed EM algorithms for density estimation and clustering in sensor networks". In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2245–2253 (cited on page 78).

[Ott et al., 2014] L. Ott, L. Pang, F. Ramos & S. Chawla. "On integrated clustering and outlier detection". In: *Advances in Neural Information Processing Systems 27*. 2014, pp. 1359–1367 (cited on page 56).

[Peel & McLachlan, 2000] D. Peel & G. J. McLachlan. "Robust mixture modelling using the t distribution". In: *Statistics and Computing* 10 (2000), pp. 339–348 (cited on pages 56, 60).

[Pelleg & Moore, 2000] D. Pelleg & A. Moore. "X-means: extending K-means with efficient estimation of the number of clusters". In: *Proceedings of the 17th International Conference on Machine Learning (ICML)*. Stanford, USA, 2000, pp. 727–734 (cited on pages 10, 11, 23, 64, 110).

[Plata-Chaves et al., 2015] J. Plata-Chaves, A. Bertrand & M. Moonen. "Distributed signal estimation in a wireless sensor network with partially-overlapping node-specific interests or source observability". In: *Proceedings of the 40th IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia, 2015, pp. 5808–5812 (cited on pages 43, 77).

[Plata-Chaves et al., 2017] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis & A. M. Zoubir. "Heterogeneous and multi-task wireless sensor networks - algorithms, applications and

challenges". In: *IEEE Journal on Selected Topics in Signal Processing* 11.3 (2017), pp. 450–465 (cited on pages 42, 77).

[Rao & Wu, 1989] C. R. Rao & Y. Wu. "A strongly consistent procedure for model selection in a regression problem". In: *Biometrika* 76.2 (1989), pp. 369–74 (cited on pages 10, 24, 62).

[Rao & Wu, 2001] C. R. Rao & Y. Wu. *On model selection*. IMS Lecture Notes - Monograph Series. 2001 (cited on page 10).

[Ricci & Balleri, 2015] R. Ricci & A. Balleri. "Recognition of humans based on radar micro-Doppler shape spectrum features". In: *IET Radar, Sonar & Navigation* 9.9 (2015), pp. 1216–1223 (cited on page 115).

[Rissanen, 1978] J. Rissanen. "Modeling by shortest data description". In: *Automatica* 14.5 (1978), pp. 465–471 (cited on page 10).

[Rousseeuw, 1987] P. J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20.1 (1987), pp. 53–65 (cited on page 10).

[Sayed, 2014a] A. H. Sayed. "Adaptive networks". In: *Proceedings of the IEEE* 102.4 (2014), pp. 460–497 (cited on pages 79, 80, 93).

[Sayed, 2014b] A. H. Sayed. "Diffusion adaptation over networks". In: *Academic Press Library in Signal Processing: Array and Statistical Signal Processing*. Ed. by A. M. Zoubir, M. Viberg, R. Chellappa & S. Theodoridis. Vol. 3. ELSEVIER, 2014. Chap. 9, pp. 323–453 (cited on page 45).

[Sayed et al., 2013] A. H. Sayed, S. Tu, J. Chen, X. Zhao & Z. J. Towfic. "Diffusion strategies for adaptation and learning over networks". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 155–171 (cited on pages 12, 44).

[Al-Sayed et al., 2017] S. Al-Sayed, A. M. Zoubir & A. H. Sayed. "Robust distributed estimation by networked agents". In:

IEEE Transactions on Signal Processing 65.15 (2017), pp. 3909–3921 (cited on page 77).

[Schizas et al., 2009] I. D. Schizas, G. Mateos & G. B. Giannakis. "Distributed LMS for consensus-based in-network adaptive processing". In: *IEEE Transactions on Signal Processing* 57.6 (2009), pp. 2365–2382 (cited on page 77).

[Schwarz, 1978] G. Schwarz. "Estimating the dimension of a model". In: *The Annals of Statistics* 6.2 (1978), pp. 461–464 (cited on pages 10, 19, 21, 53, 56, 57, 64, 125, 130).

[Seifert et al., 2017] A.-K. Seifert, A. M. Zoubir & M. G. Amin. "Radar-based human gait recognition in cane-assisted walks". In: *Proceedings of the IEEE Radar Conference*. Seattle, USA, 2017, pp. 1428–1433 (cited on page 116).

[Shahbaba & Beheshti, 2012] M. Shahbaba & S. Beheshti. "Improving X-Means clustering with MNDL". in: *Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. Montreal, Canada, 2012, pp. 1298–1302 (cited on pages 10, 11).

[Shao, 1996] J. Shao. "Bootstrap model selection". In: *Journal of the American Statistical Association* 91.434 (1996), pp. 655–665 (cited on page 10).

[Shibata, 1980] R. Shibata. "Asymptotically efficient selection of the order of the model for estimating parameters of a linear process". In: *The Annals of Statistics* 8.1 (1980), pp. 147–164 (cited on page 10).

[Sibson, 1973] R. Sibson. "SLINK: an optimally efficient algorithm for the single-link cluster method". In: *The Computer Journal* 16.1 (1973), pp. 30–34 (cited on page 108).

[Spiegelhalter et al., 2002] D. J. Spiegelhalter, N. G. Best, B. P. Carlin & A. van der Linde. "Bayesian measures of model complexity and fit". In: *Journal of the Royal Statistical Society Series B* 64.4 (2002), pp. 583–639 (cited on page 10).

[Stoica & Selen, 2004] P. Stoica & Y. Selen. "Model-order selection: a review of information criterion rules". In: *IEEE Signal*

Processing Magazine 21.4 (2004), pp. 36–47 (cited on pages 15, 24, 62).

[Stone, 1974] M. Stone. "Cross-validatory choice and assessment of statistical prediction". In: *Journal of the Royal Statistical Society Series B* 36.2 (1974), pp. 111–133 (cited on page 10).

[Szurley et al., 2015] J. Szurley, A. Bertrand & M. Moonen. "Distributed adaptive node-specific signal estimation in heterogeneous and mixed-topology wireless sensor networks". In: *Signal Processing* 117 (2015), pp. 44–60 (cited on page 77).

[Tahmoush & Silvious, 2009] D. Tahmoush & J. Silvious. "Radar micro-Doppler for long range front-view gait recognition". In: *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. Washington, USA, 2009, pp. 346–351 (cited on page 115).

[Taj & Cavallaro, 2009] M. Taj & A. Cavallaro. "Multi-camera track-before-detect". In: *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras*. Como, Italy, 2009 (cited on pages 79, 80).

[Teixeira et al., 2010] T. Teixeira, G. Dublon & A. Savvides. "A survey of human-sensing: methods for detecting presence, count, location, track, and identity". In: *ACM Computing Surveys* 5.1 (2010), pp. 59–69 (cited on page 115).

[Teklehaymanot et al., 2015] F. K. Teklehaymanot, M. Muma, B. Béjar, P. Binder, A. M. Zoubir & M. Vetterli. "Robust diffusion-based unsupervised object labelling in distributed camera networks". In: *Proceedings of the 12th IEEE AFRICON*. Addis Ababa, Ethiopia, 2015 (cited on pages 43, 80).

[Teklehaymanot et al., 2016] F. K. Teklehaymanot, M. Muma, J. Liu & A. M. Zoubir. "In-network adaptive cluster enumeration for distributed classification/labeling". In: *Proceedings of the 24th European Signal Processing Conference (EU-*

SIPCO). Budapest, Hungary, 2016, pp. 448–452 (cited on pages 10, 12, 31, 34, 43, 47, 49).

[Teklehaymanot et al., 2017] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Adaptive diffusion-based track assisted multi-object labeling in distributed camera networks". In: *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*. Kos, Greece, 2017, pp. 2363–2367 (cited on pages 43, 80, 92).

[Teklehaymanot et al., 2018a] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Bayesian cluster enumeration criterion for unsupervised learning". In: *IEEE Transactions on Signal Processing* 66.20 (2018), pp. 5392–5406 (cited on pages 12, 14).

[Teklehaymanot et al., 2018b] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Diffusion-based Bayesian cluster enumeration in distributed sensor networks". In: *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP)*. Freiburg, Germany, 2018, pp. 1–5 (cited on pages 12, 44).

[Teklehaymanot et al., 2018c] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. *MATLAB toolbox for Bayesian cluster enumeration*. 2018. online: https://github.com/FreTekle/Bayesian-Cluster-Enumeration (cited on page 25).

[Teklehaymanot et al., 2018d] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Novel Bayesian cluster enumeration criterion for cluster analysis with finite sample penalty term". In: *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, 2018, pp. 4274–4278 (cited on pages 12, 36).

[Teklehaymanot et al., 2018e] F. K. Teklehaymanot, M. Muma & A. M. Zoubir. "Robust Bayesian cluster enumeration". Under review in: *IEEE Transactions on Signal Processing*. 2018. online: https://arxiv.org/abs/1811.12337 (cited on pages 57, 58).

[Teklehaymanot et al., 2018f] F. K. Teklehaymanot, A.-K. Seifert, M. Muma, M. G. Amin & A. M. Zoubir. "Bayesian target enumeration and labeling using radar data of human gait". In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. Rome, Italy, 2018, pp. 1356–1360 (cited on page 109).

[Tibshirani et al., 2001] R. Tibshirani, G. Walther & T. Hastie. "Estimating the number of clusters in a dataset via the gap statistic". In: *Journal of the Royal Statistical Society Series B* 63.2 (2001), pp. 411–423 (cited on page 10).

[Tipping & Bishop, 1999] M. E. Tipping & C. M. Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society Series B* 61.3 (1999), pp. 611–622 (cited on page 117).

[Tu & Sayed, 2014] S.-Y. Tu & A. H. Sayed. "Distributed decision-making over adaptive networks". In: *IEEE Transactions on Signal Processing* 62.5 (2014), pp. 1054–1069 (cited on page 78).

[Vandersmissen et al., 2018] B. Vandersmissen, N. Knudde, A. Jalalvand, I. Couckuyt, A. Bourdoux, W. D. Neve & T. Dhaene. "Indoor person identification using a low-power FMCW radar". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.7 (2018) (cited on page 115).

[Wang et al., 2009] D. Wang, J. Li & Y. Zhou. "Support vector machine for distributed classification: a dynamic consensus approach". In: *Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing (SSP)*. Cardiff, UK, 2009, pp. 753–756 (cited on page 78).

[Wang et al., 2018] M. Wang, Z. B. Abrams, S. M. Kornblau & K. R. Coombes. "Thresher: determining the number of clusters while removing outliers". In: *BMC Bioinformatics* 19.9 (2018), pp. 1–15 (cited on page 56).

[Wu et al., 2009] K.-L. Wu, M.-S. Yang & J.-N. Hsieh. "Robust cluster validity indexes". In: *Pattern Recognition* 42.11 (2009), pp. 2541–2550 (cited on page 56).

[Xu & Wunsch, 2005] R. Xu & D. Wunsch. "Survey of clustering algorithms". In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678 (cited on pages 3, 10).

[Zadeh, 1965] L. A. Zadeh. "Fuzzy sets". In: *Information and Control* 8 (1965), pp. 338–353 (cited on page 108).

[Zemene et al., 2016] E. Zemene, Y. T. Tesfaye, A. Prati & M. Pelillo. "Simultaneous clustering and outlier detection using dominant sets". In: *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*. Cancún, Mexico, 2016, pp. 2325–2330 (cited on page 56).

[Zhang & Andreou, 2008] Z. Zhang & A. G. Andreou. "Human identification experiments using acoustic micro-Doppler signatures". In: *Proceedings of the Argentine School of Micro-Nanoelectronics, Technology and Applications (EAMTA)*. Buenos Aires, Argentina, 2008, pp. 81–86 (cited on page 115).

[Zhao et al., 2008a] Q. Zhao, V. Hautamaki & P. Fränti. "Knee point detection in BIC for detecting the number of clusters". In: *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems. (ACIVS)*. Juan-les-Pins, France, 2008, pp. 664–673 (cited on pages 10, 23, 34).

[Zhao et al., 2012] Q. Zhao, V. Hautamäki, I. Kärkkäinen & P. Fränti. "Random swap EM algorithm for Gaussian mixture models". In: *Pattern Recognition Letters* 33 (2012), pp. 2120–2126 (cited on pages 29, 131).

[Zhao et al., 2008b] Q. Zhao, M. Xu & P. Fränti. "Knee point detection on Bayesian information criterion". In: *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*. Dayton, USA, 2008, pp. 431–438 (cited on pages 10, 23, 34).

[Zhao & Karypis, 2005] Y. Zhao & G. Karypis. "Hierarchical clustering algorithms for document datasets". In: *Data Mining and Knowledge Discovery* 10 (2005), pp. 141–168 (cited on page 108).

[Zimek et al., 2012] A. Zimek, E. Schubert & H.-P. Kriegel. "A survey on unsupervised outlier detection in high-dimensional numerical data". In: *Statistical Analysis and Data Mining* 5.5 (2012), pp. 363–387 (cited on page 131).

[Zoubir, 1999] A. M. Zoubir. "Bootstrap methods for model selection". In: *International Journal of Electronics and Communications* 53.6 (1999), pp. 386–392 (cited on page 10).

[Zoubir & Iskander, 2000] A. M. Zoubir & D. R. Iskander. "Bootstrap modeling of a class of nonstationary signals". In: *IEEE Transactions on Signal Processing* 48.2 (2000), pp. 399–408 (cited on page 10).

[Zoubir et al., 2012] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh & M. Muma. "Robust estimation in signal processing". In: *IEEE Signal Processing Magazine* 29.4 (2012), pp. 61–80 (cited on pages 4, 55, 80).

[Zoubir et al., 2018] A. M. Zoubir, V. Koivunen, E. Ollila & M. Muma. *Robust Statistics for Signal Processing*. Cambridge University Press, 2018 (cited on pages 4, 55, 80).

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 04. Dezember 2018

Freweyni Kidane Teklehaymanot

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 12 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at github.com/suchow/Dissertate or from its lead author, Jordan Suchow, at suchow@post.harvard.edu.