DISS. ETH NO. 21628

INFORMATION-THEORETIC VALIDATION OF CLUSTERING ALGORITHMS

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

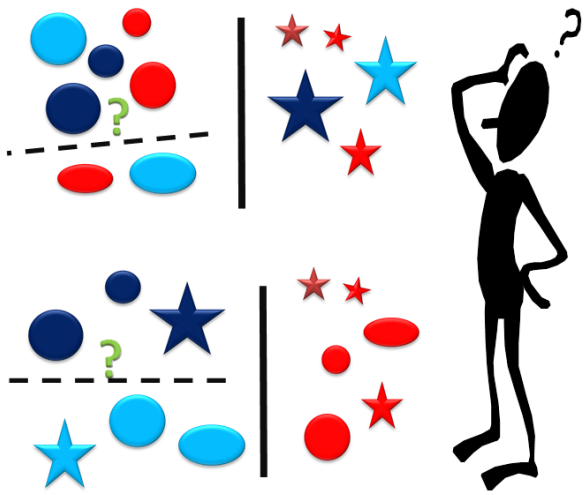
presented by
MORTEZA HAGHIR CHEHREGHANI
Master of Science in Computer Engineering
Sharif University of Technology, Tehran, Iran
born Feb 23th, 1982
citizen of Iran

accepted on the recommendation of
Prof. Dr. Joachim M. Buhmann, examiner
Prof. Dr. Peter Widmayer , co-examiner
Prof. Dr. Marcello Pelillo , co-examiner

2013

INFORMATION-THEORETIC VALIDATION OF CLUSTERING ALGORITHMS

MORTEZA HAGHIR CHEHREGHANI



An Information-Theoretic Approach to Analyzing Clustering Algorithms

2013

Morteza Haghir Chehreghani: *Information-Theoretic Validation of Clustering Algorithms,* An Information-Theoretic Approach to Analyzing Clustering Algorithms.

ABSTRACT

This thesis focuses on an information-theoretic analysis of clustering algorithms. In many real-world applications, because of imprecise or incomplete measurements, the data is contaminated by noise. For instance, gene expression levels might be measured imperfectly due to improper experimental conditions. This renders the empirical output of an algorithm to be unstable. Statistical learning advocates to employ the generalization ability of models as a measure of model quality. Therefore, we introduce the *Minimum Transfer Costs* (MTC) principle for model order selection inside a specific family of models described by a cost function. We employ the principle to compute the number of clusters in several clustering models such as Gaussian Mixture Models, Pairwise Clustering, and Correlation Clustering.

Stability is, however, only one aspect of statistical modeling; the *informativeness* of the solutions is the other side of the modeling trade-off. Maximizing stability without informativeness can yield very simple and useless solutions. An optimal trade-off requires an information-theoretic approach where the uncertainty in the measurements quantizes the solution space and thereby, induces a coarsening in the solution space. Approximation Set Coding (ASC) [Buh10] attempts to address such questions by establishing a conceptual set-based communication scenario. We elaborate *Generalization Capacity* ($\mathcal{GC}$), a context-sensitive principle for model validation based on Approximation Set Coding. An algorithm is assumed as a data processing mechanism that during execution, produces a weight distribution over the solution space. Generalization capacity computes the optimal concentration of the weights, i.e. it measures the maximal rate of reliable information which can be captured by the algorithm.

We establish a principled pipeline to compute and analyze the generalization capacity for model selection and validation in the context of data clustering. This approach provides a framework to address the fundamental learning questions: i) finding the optimal number of clusters, ii) ranking different similarity measures, and iii) validating alternative clustering methods. Efficient approximation schemes such as *mean-field approximation* are utilized to overcome the computational challenges that occur when computing generalization capacity. Furthermore, we propose exploiting a Hamming metric in the solution space to analyze the ad hoc algorithms that do not yield a trajectory of weight distributions over the solution space.

The principle is first exemplified for density estimation in different settings, particularly for learnability phase transitions in the high dimensional limit. Generalization capacity confirms the evolution of the phase transitions detected by order parameters. Moreover, it yields consistent results with other principles such as BIC and MTC.

The principle is then employed to analyze several aspects of well-known graph clustering methods. We particularly investigate the parametrization of the clustering models in an information-theoretic manner. The principle, for example, computes the optimal adaptation of Ratio Cut with respect to different Laplacians, or, determines the optimal termination of Dominant Set clustering. In the same way, we design a prototypical model by augmenting the basic Min Cut model, which is shown to be equivalent to Correlation Clustering, by a *shift* parameter. Its optimal adaptation is obtained through a *context sensitive* search over the space of alternatives, by exploiting the generalization capacity principle. This approach advocates a scientific procedure for validating the optimal model suitable for the specific application at hand, rather than an arbitrary elegant design which might yield bias towards specific types of patterns.

The framework is demonstrated on clustering of experimental gene expression data. For each method and similarity measure, $\mathcal{GC}$ computes the optimal number of clusters. It constitutes a consistent but more general principle than BIC. In particular, we compare different clustering methods and similarity measures and show how properly shifted Correlation Clustering with an appropriate measure extracts the largest amount of reliable information for all algorithms under consideration. In different biological applications, $\mathcal{GC}$ suggests a consistent ranking of similarity measures with respect to the context of the data, e.g. correlation coefficients are preferred for temporal data.

ZUSAMMENFASSUNG

Diese Arbeit behandelt eine Informations-theoretische Analyse von Cluster-Algorithmen. In vielen praktischen Anwendungen sind die Daten aufgrund unpräziser oder unvollständiger Messungen durch Rauschen verunreinigt. So könnten beispielsweise Gen-Expressions-Levels wegen ungeeigneter experimenteller Bedingungen ungenau gemessen werden. Dies führt dazu, dass die empirische Ausgabe eines Algorithmus instabil wird. Statistisches Lernen plädiert dafür, die Generalisierungsfähigkeit von Modellen als Mass für die Modellqualität einzusetzen. Daher führen wir *Minimum Transfer Cost* (MTC) als Prinzip zur Modell-Ordnungsauswahl innerhalb einer speziellen Familie von Modellen ein, die durch eine Kostenfunktion beschrieben werden. Wir wenden das Prinzip an, um die optimale Anzahl an Clustern für mehrere Clustering-Modelle wie Gauss'sche Mixturmodelle, paarweises Clustern und korrelations-basiertes Clustern zu finden.

Stabilität ist jedoch nur ein Aspekt der statistischen Modellierung; die andere Seite im Modellierungs-Tradeoff ist die Informativität von Lösungen. Das Maximieren der Stabilität - ohne die Informativität zu beachten - kann zu sehr einfachen und nutzlosen Lösungen führen. Ein optimaler Tradeoff verlangt nach einem Informations-theoretischen Ansatz, wobei die Ungenauigkeit in den Messungen den Lösungsraum quantisieren und damit eine Vergröberung im Lösungsraum veranlassen. *Approximation Set Coding* (ASC) [Buh10] versucht diese Fragen anzugehen, indem ein konzeptionelles, Mengen-basiertes Kommunikations-Szenario etabliert wird. Wir arbeiten die sog. *Generalisierungskapazität* ($\mathcal{GC}$) aus, ein Kontext-sensitives Prinzip für die Modellvalidierung basierend auf Approximation Set Coding. Ein Algorithmus wird als datenverarbeitender Mechanismus interpretiert, der im Verlauf seiner Ausführung eine gewichtete Verteilung im Ausgaberaum produziert. Die Generalisierungskapazität berechnet die optimale Konzentration der Gewichte, d.h. sie misst die maximale Rate zuverlässiger Information, die vom Algorithmus erfassst werden kann.

Wir etablieren einen prinzipientreuen Rahmen, um die Generalisierungsfähigkeit zur Modellselektion und -validierung im Kontext von Daten-Clustering zu berechnen und analysieren. Der Rahmen bietet eine wohldefinierte Methodik, um die fundamentalen Fragen der Lerntheorie anzugehen: i) die Bestimmung der optimalen Anzahl an Clustern, ii) ein Ranking der unterschiedlichen Ähnlichkeitsmasse und iii) alternative Clustering-Methoden zu validieren. Wir verwenden effiziente Approximations-Schemen wie die Mean-Field-

Approximation, um die berechnungstechnischen Herausforderungen bei der Berechnung der Generalisierungsfähigkeit zu überkommnen. Wir schlagen die Nutzung einer Hamming-Metrik im Lösungsraum vor, um diejenigen ad-hoc Algorithmen analysieren zu können, die keine Trajektorie von Gewichtsverteilungen über dem Lösungsraum bieten.

Wir veranschaulichen das Prinzip zuerst für die Dichteschätzung in unterschiedlichen Situationen, vor allem der Phasenübergang der Maximum-Likelihood-Schätzung im hochdimensionalen Limit. Die Generalisierungskapazität bestätigt die Entwicklung der Phasenübergänge, die von den Ordnungsparametern detektiert werden. Darüber hinaus sind die Ergebnisse mit anderen Prinzipien wie BIC und MTC konsistent.

Dann wenden wir das Prinzip an, um mehrere Aspekte von wohlbekannten Graph-Clustering Methoden zu analysieren. Speziell untersuchen wir die Parametrisierung der Clustering-Modelle in einer Informations-theoretischen Art. Wir benutzen das Prinzip zum Beispiel, um die optimale Anpassung von Ratio Cut inbezug auf unterschiedliche Laplacians zu berechnen, oder um die optimale Beendigung von Domimant Set Clustering zu bestimmen. In der gleichen Art designen wir ein prototypisches Modell, indem wir das grundlegende Min Cut Modell (welches, wie gezeigt wird, äquivalent zu Correlation Clustering ist) mit einem Shift-Parameter erweitern. Seine optimale Anpassung wird durch eine kontext-sensitive Suche über den Raum der Alternativen erzielt, wobei das Prinzip der Generalisierungs-Kapazität ausgenutzt wird. Dieser Ansatz steht für ein wissenschaftliches Vorgehen, um das optimale Modell - passend für die spezifische, vorliegende Anwendung - zu validieren, anstatt ein beliebiges, elegantes Design zu wählen, welches einen Bias für spezifische Typen von Mustern haben könnte.

Wir demonstrieren den Rahmen für das Clustering von experimentellen Gen-Expressionsdaten. Für jede Methode und jedes Ähnlichkeitsmass bestimmt $\mathcal{GC}$ die optimale Zahl an Clustern. Es stellt ein konsistentes, aber generelleres Prinzip als BIC dar. Wir vergleichen speziell verschiedene Clustering-Methoden und Ähnlichkeitsmasse und zeigen, wie ein angemessen verschobenes Correlation Clustering mit einem angemessenen Mass die grösste Menge an verlässlicher Information unter allen betrachteten Algorithmen extrahiert. In jeder biologischen Anwendung schlägt $\mathcal{GC}$ ein konsistentes Ranking von Ähnlichkeitsmassen bezogen auf den Kontext der Daten vor, z.B. Korrelations-Koeffizienten für zeitliche Daten.

ACKNOWLEDGMENTS

Foremost, my sincerest appreciation goes to my advisor, Prof. Joachim Buhmann, for his constant support and encouragement, novel ideas, and constructive feedbacks during the entire period of my PhD. I learned a lot from him, both about information theory and learning, and most importantly, about the scientific way of thinking and analysis. Also, I would like to express my gratitude to Prof. Peter Widmayer and to Prof. Marcello Pelillo for co-refereeing this thesis.

I feel very happy that I did my PhD at the Machine Learning Laboratory of ETH Zurich, where I enjoyed the international environment and many interesting scientific discussions. I specially thank Rita Klute for her sincere administrative helps and for providing a friendly atmosphere inside the group.

It was a great pleasure to work with Ludwig Busse. I very much enjoyed of both scientific and personal interactions with him. We performed several scientific research together, as well as we shared a lot of time in conferences and meetings. Thank you very much Ludwig! I am always proud of having you as a friend and as a collaborator.

I am very much thankful to Mario Frank for many insightful discussions and for his bright ideas about performing research. I benefited a lot from his knowledge, and his experience. I also appreciate working with Alberto Giovanni Busetto from both scientific and personal aspects and for sharing many common interests. I thank Alberto Giovanni Busetto, Ashraf Masood Kibriya, Brian McWilliams, and particularly Ludwig Busse for proof-reading the thesis.

I dedicate the thesis to my family. Whatever I have achieved in my life comes from their support, love, and patience. They are always there for me whenever I need them.

CONTENTS

LIST OF TABLES

Part I

DATA, INFORMATION, AND LEARNING

1 INTRODUCTION

1.1 THE LEARNING PROBLEM

The overwhelming amount of data that today's information society faces requires sophisticated methods to compress, abstract and visualize the data and to, finally, extract knowledge from it. The process of converting the *raw data* to *information* and, ultimately, to *knowledge* (Figure 1) helps the decision makers to understand the nature of the problem at hand and thereby improves decisions by knowing the potential consequences. In order to extract *information*, a usual approach is to analyze the problem at an *adequate* level of detail. This *abstraction* provides the decision makers with important regularities (information) hidden in the data without drowning in minor irrelevant detail. Thereby, the first question is "What is the right model/method to convert the data to information, and then to knowledge?" The second question asks "What is the sufficient level of detail a data analyst should care about?" Machine Learning provides a framework to address such questions.

Figure 1: The process of extracting information from data: The information is distilled to knowledge to support decision makers.

Machine learning methods are evaluated based on different criteria such as speed, scalability and memory usage. However, an important aspect is the ability of a method to capture a suitable amount of regularities which are *stable*. Stability refers to the compatibility of the solutions among different problem instances from the same data source. Many learning methods can output very complicated and overdetailed solutions if they are not properly regularized. Unconstrained complexity might lead to a lack of stability. Machine learning provides not only a set of tools for analyzing data, but also a framework to study different aspects of learning such as stability, complexity, computational bottlenecks, scalability, performance and so on.

This thesis addresses the learning problem in the context of data clustering. *Clustering* is a common learning task that addresses the question of partitioning objects into groups and constitutes a fundamental processing step in exploratory data analysis. This unsupervised task arises in many applications such as web data analysis, im-

age segmentation, data compression, computational biology, network analysis, computer vision, routing and document summarization. Today, there is a zoo of clustering methods developed for different applications and data types, e.g. K-means, Normalized Cut and Link Linkage methods.

Learning, including cluster analysis, can be formulated in three steps:

1. **Modeling**: involves defining a process for interpreting the data.
2. **Inference**: is the process of estimating/computing the free parameters of a model.
3. **Regularization**: is the strategy to be followed to limit the complexity of a model; too complicated models usually lead to overfitting.

1.1.1 *Modeling*

Essentially, a model provides a prototype for abstracting the regularities in data. A model is usually augmented by a so-called *model complexity* parameter which identifies the degree of the detail. A model with a higher complexity refers to extracting more detail from the data as compared to a simpler model. Statistical models constitute an important category of learning models. They assume an underlying probability distribution from which the data is drawn. The model complexity then refers to the number of free parameters of the distribution.

However, models are not limited to statistical distributions. Generally speaking, a clustering model can be expressed by a probabilistic model (e.g. a Gaussian Mixture Model), a cost function (e.g. the K-means cost function) or an algorithm (e.g. Single Linkage clustering). Gaussian Mixture Models (GMMs) are arguably the most widely used statistical models for clustering. These models assume a K-modal Gaussian distribution responsible for generating the data. K, the number of clusters, is the main parameter that controls the model complexity. A very large K might yield many small unstable clusters.

1.1.2 *Inference*

Inference is the procedure of estimating the free parameters and producing the final solution(s). It usually leads to computing the *empirical output* of the model along with the model parameters, e.g. the mean and the variance in the case of a Gaussian model. For the models characterized by a cost function, inference constitutes the minimization procedure.

Maximum likelihood and Bayesian estimation are the two main inference approaches for statistical models. Maximum likelihood inference relies only on the evidence/data without a reference to prior beliefs. Bayesian inference, on the contrary, provides a principled way of combining new evidence with prior knowledge. If a model is specified by an algorithm, the inference step then involves just executing the algorithm. Actually, for algorithms, there is not a clear separation between modeling and inference steps.

1.1.3 *Regularization*

Unconstrained minimization of a cost function might lead to overfitting which causes instability of solutions. For example, the empirical minimizer of the K-means cost function returns an undesirable and unstable solution with $K =$*'number of objects'* singleton clusters. Therefore, the inference procedure should be regularized in order to reduce the risk of overfitting.

Statistical models that assume an explicit parametric model are often controlled by a model complexity penalty, e.g. Akaike Information Criterion (AIC) [Aka73], Bayesian information Criterion (BIC) [Sch78] and Minimum Description Length (MDL) [Ris78], or by stochastic complexity. However, these types of principles have only been applied to statistical models where computing the effective number of free parameters is straightforward.

Stability-based cluster validation [LBRB04] can be employed to identify the optimal number of clusters or other model parameters. For a very general setting of clustering algorithms, stability-based validation procedures detect discrepancies between a predicted and an optimized clustering solution in the spirit of cross-validation. Properly regularized or approximated solutions, i.e. solutions with the "right" model and the "correct" number of clusters, exhibit only insignificant changes and are considered to be stable. The stability analysis has been criticized [BDvP06] based on the observation that clusterings for all cluster numbers become stable in the asymptotic limit. The key to understanding the effectiveness of stability-based validation seems to be hidden in the fast convergence behavior of models with the correct model complexity [ST10b]. However, in practice, usually only a few finite data samples are available from a source.

Maximum entropy inference is a very efficient entropy-based principle where the so-called *free energy* $F = \mathbb{E}[R] - TH$ is minimized instead of the cost function R. The lagrange multiplier T, the so-called *computational temperature*, controls the balance between the empirical minimizer and the entropy H. In the limit $T \to 0$, there are no thermal fluctuations and one can calculate an empirical minimizer of the cost function.

According to [Jay57a, Jay57b], maximizing the entropy provides the least biased inference method that has the minimum commitment to the unknown part of data and leads to the robustness of the inference technique. It has been shown [TTL84] that in a temperature-based sampling procedure, the maximum-entropy probability distribution is maximally stable if the expected costs change proportional to the temperature. The family of Gibbs distributions possesses this property [GG84]. The Gibbs distribution is defined over the set of all possible solutions and assigns a higher probability to solutions with lower costs. A sampling scheme known as Gibbs sampler [GG84] employs the conditional Gibbs distribution to sample from the solution space. High temperatures render stable, but possibly non-informative solutions. A very low temperature solution, improves informativeness but with the price of a potential reduction in stability. Thereby, the main challenge is concerned with computing an appropriate temperature that provides the best trade-off between stability and informativeness.

1.2 MODEL SELECTION AND VALIDATION

Today, a wide range of possibilities exists in the literature for solving almost every learning problem. The collection of clustering models, for example, ranges from centroid-based algorithms like K-means or K-medoids, graph partitioning methods like Normalized Cut, Correlation Clustering or Pairwise Clustering to algorithmic procedures like Single Linkage, Average Linkage or Dominant Set clustering. The various clustering methods and algorithms ask for a unifying principle how to choose the 'right' clustering method. Typically, the data analyst has the option to select a clustering algorithm from a repository of alternatives, while often puzzling over the central question: Which algorithm is most suitable in a certain context? Clustering might even be considered to be an *art* rather than a scientific problem [vLWG12].

Thereby, a learning problem faces three fundamental choice challenges:

(i) Which algorithm should be chosen?

(ii) What kind of preprocessing is required?

(iii) What is the optimal model order?

In the case of clustering algorithms, Question (i) might concern the choice of e.g. K-means or Correlation Clustering? Question (ii) could distinguish between Euclidian distance measurements or correlation coefficients? The model order (iii) addresses the issue what number of clusters for an (clustering) algorithm should be selected. To the

best of our knowledge, there does not exist a general-purpose principle to answer the first two questions. Although the third question has been addressed by several principles (e.g. AIC, BIC, instability index and gap statistics [TWHoo]), however, they rely on very specific assumptions for example on the type of the underlying data distribution.

This thesis advocates a shift of viewpoint away from the problem "What is the *right* clustering model?" to the question "How can we algorithmically validate different clustering models?". This conceptual shift roots in the assumption that ultimately, the data should vote for their preferred model type and model complexity [CHo8]. Therefore, a validation principle is required to maneuver through the space of clustering models and, dependent on the datasets at hand, select a model with maximal information content and optimal robustness. While the design of clustering models based on prior knowledge of the data source might be considered as *art*, the systematic search through the space of clustering models by cluster validation principles defines an algorithmic strategy of a *scientific* program. Thereby, such an analysis not only renders a principled comparison of algorithms possible, but also guides the design of algorithms that provide the maximum reliable information.

1.3 INFORMATION-THEORETIC LEARNING

Learning, in general, resembles communication from a conceptual viewpoint: For *communication*, one demands a high rate (a large amount of information transferred per channel use) together with a decoding rule that is stable under the perturbations of the messages by the noise in the channel. For *learning* patterns in data, the data analyst favors a rich model with high complexity (e.g., a large number of groups in clustering), while the generalization error is expected to remain stable and low. We require that solutions of pattern analysis problems are reliably inferred from noisy data.

Statistical learning advocates to employ the generalization ability of models as a measure of model quality. In the same sprit, stability analysis supports the consistency between the solutions of different instances from the same problem type. *Stability* is, however, only one aspect of statistical modeling, the *informativeness* of the solutions is the other side of the modeling trade-off. Maximizing stability without informativeness might lead to very simple and useless solutions. In a grouping task, putting every object in a single cluster gives a perfect stability but no information is then provided. These criteria often behave in opposite ways: increasing stability reduces informativeness and vice versa. The correct balancing of these two antagonistic goals is very important since a tolerable decrease in model stability of inferred patterns might provide a substantial increase in informa-

tion content [TPB99]. An optimal trade-off requires an information-theoretic approach as described in [Buh10], where the uncertainty in the measurements quantizes the solution space and thereby, induces a coarsening in the solution space.

In the two-datasets scenario of learning theory, training and test datasets usually differ due to the noise in the empirical measurements. This situation can be compared with a communication scenario where the noisy channel renders the message in the receiver side to be different from the message sent by the sender. Approximation Set Coding [Buh10] employs this analogy by transforming the model selection problem into a coding problem for a generic *set-based communication* scenario. A model is considered as a noisy channel and the training dataset is used to generate a codebook for communication. The test dataset is then used to investigate how well this coding scheme works. More detail on this principle will be provided in Chapter 4.

However, there exists a longer history of information-theoretic approaches to model selection, which traces back at least to Akaike's extension of the maximum likelihood inference. AIC aims to provide an unbiased estimation of the Kullback-Leibler (KL) divergence between a candidate model and the underlying true model. The method of *information bottleneck* [TPB99] proposes to select the number of clusters according to a difference of mutual informations. This asymptotic concept is closely related to rate distortion theory with side information (see [CT06]). Finite sample size corrections of the information bottleneck [SB04] yield an optimal temperature with a preferred number of clusters.

Relations between learning theory and information theory have been also developed for defining information-theoretic clustering measures [GP02, SATB05] and for computing minimax lower bounds to estimate the parameters of statistical models [Mer94, YB99]. These validation methods, however, cannot be applied to the validation of general algorithms.

1.4 CONTRIBUTIONS

The following list summarizes the main contributions of this thesis.

- We introduce the Minimum Transfer Costs principle for model order selection by generalizing cross-validation to unsupervised learning problems [FCB11]. We investigate the applicability of this principle to compute the optimal number of clusters in different models such as Gaussian mixture models, K-means, Pairwise Clustering and Correlation Clustering. This is joint work with Mario Frank.

- We elaborate *Generalization Capacity*, a context-sensitive principle for model validation and ranking based on Approximation Set Coding. We prove the consistency of the principle with Shannon capacity for K-ary codes. The Approximation Set Coding framework [Buh10] has been recently developed by Joachim Buhmann.

- We establish an algorithmic procedure to compute and interpret generalization capacity for validating arbitrary methods in the context of data clustering. We use *mean-field approximation* to overcome the computational challenges for graph clustering models [CBB12]. For the ad hoc methods that do not yield an appropriate metric, we propose using Hamming distance in the solution space.

- We examine the principle for density estimation at different settings. Particularly, we study the phase transition of maximum likelihood estimation in the high dimensional limit [BCFS12]. This study explores the first application of Approximation Set Coding.

- We study the parametrization of clustering models in an information-theoretic manner. We use the generalization capacity principle to, for example, analyze the effect of shifting the pairwise similarities, or to compute the optimal termination of Dominant Set clustering in the context of dynamical systems. The study provides a framework to perform validation algorithmically through a *context sensitive* scanning of the space of alternative models and selecting a model with maximal information content and optimal robustness.

- We investigate the framework for analyzing different clustering methods in the domain of gene expression data [CBB12, CBB13]. The framework addresses the fundamental learning questions arising in the context of data clustering: i) finding the optimal number of clusters, ii) ranking different similarity measures, and iii) validating alternative clustering methods.

1.5 THESIS OVERVIEW

The thesis is organized as follows. We start by introducing the notations and the preliminary concepts in Chapter 2. We setup the learning problem and define the basic elements of learning theory. In Chapter 3 we propose the Minimum Transfer Costs principle and its application to model order selection. In Chapter 4 we elaborate generalization capacity and prove its consistency with Shannon capacity. In Chapter 5 we present the procedure to compute and analyze generalization capacity for validation of clustering methods. We then

study density estimation at different low and high dimensional settings. In Chapter 6 we analyze different aspects of well-known graph clustering methods and particularly describe the parametrization of Min Cut as a prototypical model. Finally, in Chapter 7, we investigate the principle for clustering of gene expression data. The thesis is concluded in chapter 8 with an outlook on further developments of the concept of information-theoretic analysis of algorithms.

2 LEARNING SETUP

In this chapter, we precisely define the basic elements of learning theory. We first describe what is the input of a learning task, and then formulate the general learning problem. We finally adapt the formulation to clustering problem.

2.1 THE INPUT OF A LEARNING PROBLEM

An *object* or an *entity* is an item or a concept, i.e. something that is sensible. An object exists by itself and its existence is not dependent on something else. Objects are the fundamental elements of learning analysis that carry the patterns to be discovered. Examples of objects are images in a vision task, documents in a text analysis task, or the set of genes in a biological problem.

Objects are represented by *measurements*. Measurements specify the features of the objects or describe the relation between them, e.g. the vectors for text documents or the pairwise similarities between genes.

Let $\mathbf{O} \in \mathcal{O}^N$ be a set of N objects, respectively associated to their measurements $\mathbf{X}$. Multiple representations of the measurements are admissible.

1. **Vector-based representations**: The i^{th} object is described by a D-dimensional vector $\mathbf{x}_i$, i.e.

$$\mathbf{X} : \mathcal{O}^N \to \mathbb{R}^{N \times D}. \tag{1}$$

   In this representation, there is a bijective map between objects and measurements. Thereby, we can use the terms synonymously, i.e., the objects $\mathbf{O}$ are directly characterized by the measurements $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathcal{X}$.

2. **Relation-based representations**: the only available information about the objects are the pairwise similarities, where the measurement X_{ij} refers to the pairwise similarity between objects i and j. In this representation, a graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ with the set of nodes $\mathbf{O}$ and the edge weight matrix $\mathbf{X} := \{X_{ij}\} \in \mathcal{X}$ characterizes the relations for all pairs of objects (i, j), $1 \leqslant i \leqslant N, 1 \leqslant j \leqslant N$. Then we have

$$\mathbf{X} : \mathcal{O}^N \times \mathcal{O}^N \to \mathbb{R}^{N \times N}. \tag{2}$$

A *dataset* is characterized by a set of objects and the corresponding measurements, i.e. by $\{\mathbf{O}, \mathbf{X}\}$. In a two-instance learning scenario, we need to have access to two datasets $\{\mathbf{O}^{(1)}, \mathbf{X}^{(1)}\}$ and $\{\mathbf{O}^{(2)}, \mathbf{X}^{(2)}\}$ from a unique source. Often, in practical situations, only one such dataset is available. Then we randomly partition it into $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

2.2 FORMULATION OF LEARNING

A **hypothesis**, i.e. a potential solution of a pattern analysis problem, is a mapping that assigns a set of N objects to patterns $\mathbb{K}$:

$$\begin{aligned} c \quad : \quad & \mathcal{O}^N \to \mathbb{K}^N, \\ & \mathbf{O} \mapsto c(\mathbf{O}). \end{aligned} \tag{3}$$

Accordingly, the **hypothesis class** or the **solution space** is the set of all such mappings, i.e.

$$\mathcal{C}(\mathbf{O}) := \{c(\mathbf{O})\}. \tag{4}$$

In parametric models such as K-means clustering, the solution includes a set of model parameters which are learned through an inference procedure. To simplify the notation, model parameters θ are not explicitly listed as arguments of the model. The solution c encodes all such parameters.

A category of models are characterized as an optimization problem with an associated *cost function*. A cost function $R(c, \mathbf{X})$ quantifies how well a particular solution $c \in \mathcal{C}$ explains the measurements $\mathbf{X}$. Thereby, a cost function determines a *partial ranking* over the hypothesis class. A solution c implicitly determines the number of model parameters. In clustering, for instance, c would identify the number of clusters.

The inference procedure computes the outcome of a model denoted by *empirical output* or *empirical minimizer*. We use $c^{\perp}(\mathbf{X})$ to refer to this solution. For instance, for a model described by a cost function the empirical minimizer is defined as

$$c^{\perp}(\mathbf{X}) \in \arg\min_{c} R(c, \mathbf{X}). \tag{5}$$

2.3 LEARNING SETUP FOR CLUSTERING

In clustering, the patterns are the cluster labels, i.e. $\mathbb{K} = \{1, \ldots, K\}$, where K denotes the number of clusters. Thereby the hypothesis class is the set of all object partitionings

$$\mathcal{C}(\mathbf{O}) = \{1, \ldots, K\}^N. \tag{6}$$

Clustering solutions are denoted by c, where $c(i) \in \mathbb{K}$ indicates the cluster label for object i. c encodes all additional parameters, e.g. centroids of K-means clustering.

A clustering solution can be alternatively specified by a $N \times K$ Boolean assignment matrix $\mathbf{M}$ such that

$$\mathbf{M} = \{M_{ik}\} \in \mathcal{M},\ 1 \leqslant i \leqslant N,\ 1 \leqslant k \leqslant K. \tag{7}$$

The solution space $\mathcal{M}$ is then defined as

$$\mathcal{M} = \left\{ \mathbf{M} \in \{0,1\}^{N \times K} : \sum_{k=1}^{K} M_{ik} = 1, \forall i \right\}. \tag{8}$$

$M_{ik} = 1$ indicates that the object i is assigned to cluster k, while $M_{ik} = 0$ otherwise. The constraint $\sum_{k=1}^{K} M_{ik} = 1, \forall i$ guarantees that each object is assigned to only and exactly one cluster[1].

A clustering solution c provides a *co-clustering* matrix $\mathbf{H} \in \{0,1\}^{N \times N}$.

$$H_{ij} = \begin{cases} 1 & \text{iff } c(i) = c(j) \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Thereby, $\mathbf{H}$ satisfies the following conditions:

1. *Reflexivity*: $H_{ii} = 1$.
2. *Symmetry*: $H_{ij} = H_{ji}$.
3. *Transitivity*: if $H_{ij} = 1$ and $H_{jl} = 1$, then $H_{il} = 1$.

Given a graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ and a clustering solution c, the set of edges between two clusters k and k' is defined as

$$\mathcal{E}_{kk'} = \{(i,j) \in \mathcal{E} : c(i) = k \wedge\ c(j) = k'\}, \tag{10}$$

where $\mathcal{E}$ denotes the edge set of the graph, i.e.

$$\mathcal{E} = \{(i,j) : i \in \mathbf{O} \wedge\ j \in \mathbf{O}\}. \tag{11}$$

$\mathcal{E}_{kk'}, k \neq k'$ and $\mathcal{E}_{kk}$ denote respectively *inter-cluster edges* and *intra-cluster edges*. Moreover, $\mathbf{O}_k \subset \mathbf{O}$ contains the members of the k^{th} cluster, i.e.

$$\mathbf{O}_k := \{i \in \mathbf{O}\ :\ c(i) = k\}. \tag{12}$$

For the two clusters $\mathbf{O}_k$ and $\mathbf{O}_{k'}$, we define $links(\mathbf{O}_k, \mathbf{O}_{k'})$ to be the sum of the edge weights between objects in $\mathbf{O}_k$ and $\mathbf{O}_{k'}$, i.e.

$$links(\mathbf{O}_k, \mathbf{O}_{k'}) = \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} X_{ij}. \tag{13}$$

1 In this thesis we always consider single assignment clustering. However, one can discuss a multi-assignment clustering model where an object can be assigned to more than one clusters [FSBB12], i.e. $\exists i : \sum_{k=1}^{K} M_{ik} > 1$.

Furthermore, the degree of $\mathbf{O}_k$ is defined as sum of the edge weights between objects in $\mathbf{O}_k$ and all the objects, i.e.

$$\mathtt{degree}(\mathbf{O}_k) \;=\; \mathtt{links}(\mathbf{O}_k, \mathbf{O}) \;=\; \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}} X_{ij}. \tag{14}$$

Different clustering algorithms are denoted by e.g. $\mathcal{A}^{PC}$ (Pairwise Clustering), $\mathcal{A}^{DS}$ (Dominant Set clustering) or $\mathcal{A}^{NCut}$ (Normalized Cut). Clustering methods can, but need not necessarily, be characterized by distinct cost functions $R(c, \mathbf{X})$, e.g. $R^{NCut}(c, \mathbf{X})$ and $R^{PC}(c, \mathbf{X})$.

2.4 SIMILARITY MEASURES

Given a set of pairwise similarities $\mathbf{X}$, the matrix of pairwise distances $\mathbf{D}$ is computed through negation and shift operations, i.e.

$$\mathbf{D} = \upsilon - \mathbf{X}, \tag{15}$$

where υ is a constant. υ can be selected by the maximum pairwise similarity in matrix $\mathbf{X}$. This choice renders the pairwise distances non-negative, i.e. $\mathbf{D}_{ij} \geqslant 0,\ \forall i, j$.

The pairwise measurements can be computed in different ways. In this thesis we consider three types of pairwise measurements: i) Euclidean measure, ii) Pearson correlation, and iii) Path-based measure.

1. **Euclidean measure**: The Euclidean distance between objects i and j in D-dimensional vector space is defined as the length of the line connecting them, i.e.

$$\mathbf{D}^{euc}_{ij} = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2}\,. \tag{16}$$

   The distance measure is usually used in squared form, i.e.

$$\mathbf{D}^{seuc}_{ij} = \sum_{d=1}^{D} (x_{id} - x_{jd})^2\,. \tag{17}$$

2. **Pearson correlation**: measures the linear correlation (i.e. the dependence) between two variables $\mathbf{A}$ and $\mathbf{B}$. This measure yields a value in $[-1, +1]$, where $+1$ and -1 respectively indicate total positive and total negative correlations. Formally, the Pearson correlation between two variables $\mathbf{A}$ and $\mathbf{B}$ is defined as the covariance of the two variables normalized by the product of their standard deviations.

$$\begin{aligned} \mathrm{cor}(\mathbf{A}, \mathbf{B}) \;&=\; \frac{\mathrm{cov}(\mathbf{A}, \mathbf{B})}{\sigma_{\mathbf{A}}\, \sigma_{\mathbf{B}}} \\ &=\; \frac{\mathbb{E}[(\mathbf{A} - \mu_{\mathbf{A}})(\mathbf{B} - \mu_{\mathbf{B}})]}{\sigma_{\mathbf{A}}\, \sigma_{\mathbf{B}}}, \end{aligned} \tag{18}$$

   where, $\mu_{\mathbf{A}}$ and $\mu_{\mathbf{B}}$ are the means of $\mathbf{A}$ and $\mathbf{B}$, respectively.

3. **Path-based measure**: Given a distance (dissimilarity) matrix $\mathbf{D}$, the path-based distance measure between objects i and j is computed by [FB03]

$$\mathbf{D}_{ij}^{path} = \min_{p \in \mathcal{P}_{ij}(\mathbf{O})} \left\{ \max_{1 \leqslant l \leqslant |p|} \mathbf{D}_{p(l)p(l+1)} \right\}, \tag{19}$$

where $\mathcal{P}_{ij}(\mathbf{O})$ is the set of all paths from i to j. Thereby, the effective distance between i and j is the largest gap of the path $p^{\star}$, where $p^{\star}$ is the path with minimum largest gap among all admissible paths between i and j.

Part II

VALIDATION PRINCIPLES

3

THE MINIMUM TRANSFER COSTS PRINCIPLE FOR MODEL ORDER SELECTION

The goal of model order selection is to select a model variant that generalizes best from training data to unseen test data. In unsupervised learning without any labels, the computation of the generalization error of a solution poses a conceptual problem. We address this question by formulating the principle of "*Minimum Transfer Costs*" (MTC) for model order selection. This principle renders the concept of cross-validation applicable to unsupervised learning problems. As a substitute for labels, we introduce a mapping between objects of the training set to objects of the test set enabling the transfer of training solutions. The method is studied in several clustering models including Gaussian Mixture Models, K-means clustering, Correlation Clustering and Pairwise Clustering. The principle finds the optimal model complexity in controlled and real-world applications.

3.1 GENERALIZABILITY IN LEARNING

Many learning problems require to specify the complexity of solutions in order to provide good performance on test datasets. When partitioning a set of objects into clusters, we must select an appropriate number of clusters. Learning a low-dimensional representation of a set of objects, for example by learning a dictionary, involves choosing the number of atoms or codewords in the dictionary. More generally speaking, learning the parameters of a model given some measurements requires selecting the number of parameters, i.e. one must select the model order.

In order to provide good generalization performance, parametric models are often controlled by a model complexity penalty (a regularizer). Akaike information criterion (AIC) [Aka73] and Bayesian information criterion (BIC) [Sch78] both tradeoff the goodness of fit measured in terms of a likelihood function against the number of model parameters used. Minimum description length (MDL) [Ris78] selects the lowest model order that can explain the data. It essentially minimizes the negative log posterior of the model and is thus formally identical to BIC [HTF08]. These particular principles can be considered as the specific forms of a more general principle

$$c^*(\mathbf{X}) = \arg\min_{c \in \mathcal{C}(\mathbf{X})} \left(R(c, \mathbf{X}) + \lambda \cdot \Upsilon(c, \mathbf{X}, \theta) \right), \tag{20}$$

where $\Upsilon(.)$ is a penalty function and λ trades off the penalty term and the empirical cost. In the special cases of AIC and BIC, λ is fixed to 1 and the penalty term is ϕ and $\frac{1}{2}\phi \ln N$, respectively. ϕ is the number of free parameters. Cross-validation can be employed for the cases where the value of the trade-off parameter λ is not fixed and requires to be determined. However, it is still unclear how to generalize the model-based criteria [Aka73, Sch78, HL96] to non-probabilistic methods such as, for instance, Correlation Clustering, being specified by a cost function instead of a likelihood.

Statistical learning suggests to measure the generalization ability of models and to quantify the prediction error as a measure of model quality. For clustering problems, a very effective heuristic, called *stability analysis* [LBRB04], has been proposed in the spirit of cross-validation. According to this principle, a clustering method should find a "stable" structure on the data source, i.e. it should discover similar solutions on datasets from the same distribution. The clusters are defined by the algorithm and it does not matter how they appear, but the structure that the algorithm finds must be consistent among different instances of the same problem type.

Stability analysis has shown very promising results for model order selection [DF02, LBRB04] but there is also criticism [BDvP06]: the stability measure vanishes for different model order choices in the asymptotic limit. In addition, the stability measure is not defined for the case when the model order is 1. Moreover, stability is only one aspect of statistical modeling, the informativeness of the solutions is also important.

The *Minimum Transfer Costs* principle considers the performance of a model on unseen test data. A model is learned with various model orders from a given dataset $\mathbf{X}^{(1)}$. This model with its respective parameters is then used to interpret a second dataset $\mathbf{X}^{(2)}$, i.e. to compute the transfer costs. The principle selects the model order that achieves lowest transfer costs, i.e. the solution that generalizes best to the second dataset. Too simple models underfit and achieve high costs on both datasets; too complex models overfit to the fluctuations of $\mathbf{X}^{(1)}$ which results in high costs on $\mathbf{X}^{(2)}$ where the fluctuations are different.

An important question is concerned with the transfer of the solution inferred from the first dataset to the second dataset. This transfer requires a mapping function which generalizes the conceptually straightforward assignments in supervised learning. In supervised learning a model is trained on a set of given observations $\mathbf{X}^{(1)}$ and the labels (or output variables) $\mathbf{y}^{(1)}$. Usually, we assume i.i.d. training and test data in classification and, therefore, the transfer problem disappears. We demonstrate how to map two datasets to each other when no labels are given in unsupervised setting. The princi-

ple is then employed in different applications to compute the optimal model orders.

3.2 THE MINIMUM TRANSFER COSTS PRINCIPLE

We develop the principle of *Minimum Transfer Costs* (MTC) for addressing the general issue of model order selection for unsupervised learning problems. This principle generalizes classical cross-validation known from supervised learning. It is applicable to a broad class of model order selection problems even when no labels or target values are given. In essence, MTC can be applied whenever a cost function is defined. Conceptually, MTC advocates the well-known concept of generalizability in learning: A good choice of the model order based on a given dataset should also yield low costs on a second dataset from the same distribution.

3.2.1 *Definitions and Assumptions*

Let $\mathbf{O}$ be a set of N objects with the corresponding measurements $\mathbf{X}$. The measurements can be either the vector-based representations of the objects, i.e. $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, or the pairwise similarities between objects, i.e. $\mathbf{X} := \{X_{ij}\}, 1 \leqslant i \leqslant N, 1 \leqslant j \leqslant N$. Furthermore, let $\{O^{(1)}, \mathbf{X}^{(1)}\}$ and $\{O^{(2)}, \mathbf{X}^{(2)}\}$ be two datasets given from a unique source.

We study the models characterized by a *cost function* $R(c, \mathbf{X})$. The solution c encodes the number of model parameters, e.g. in clustering, the number of clusters is the number of unique labels in c.

3.2.2 *Minimum Transfer Costs*

A cost function imposes a partial order on all possible solutions given the data. Since usually the measurements are contaminated by noise, one aims at finding solutions that are robust against the noise fluctuations and thus generalize well to future data. Learning theory demands that a well-regularized model explains not only the dataset at hand, but also new datasets generated from the same source and thus drawn from the same probability distribution.

Let $c^{(1)}$ be the solution (e.g. model parameters) learned from a given set of objects $\mathbf{O}^{(1)} = \{i : 1 \leqslant i \leqslant N_1\}$ and the corresponding measurements $\mathbf{X}^{(1)}$. Let the set $\mathbf{O}^{(2)} = \{i' : 1 \leqslant i' \leqslant N_2\}$ represent the objects of a second dataset $\mathbf{X}^{(2)}$ drawn from the same distribution as $\mathbf{X}^{(1)}$. In a supervised learning scenario, the given class labels of both datasets guide a natural and straightforward mapping of the trained solution from the first to the second dataset: the model should assign objects of both sets with same labels to the same classes. However, when no labels are available, it is unclear how to transfer a solution.

To enable the use of cross-validation, we propose to compute the costs of a learned solution on a new dataset in the following way. We start with defining a mapping ψ from objects of the second dataset to objects of the first dataset:

$$\begin{aligned}\psi : \mathcal{O} \times \mathcal{X} \times \mathcal{X} &\to \mathcal{O}\,, \\ \left(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right) &\mapsto \psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}).\end{aligned} \tag{21}$$

This mapping function aligns each object from the second dataset to an appropriate object in $\mathbf{O}^{(1)}$. We have to compute such a mapping in order to transfer a solution. Let's assume, for the moment, that the given model is a sum over independent partial costs

$$R(c, \mathbf{X}) = \sum_{i=1}^{N} R_i(c(i), \mathbf{x}_i). \tag{22}$$

$R_i(c(i), \mathbf{x}_i)$ denotes the partial costs of object i and $c(i)$ denotes the structure part of the solution that relates to object i. For a parametric centroid-based clustering model $c(i)$ would also include the centroid object i is assigned to. Using the object-wise mapping function ψ to map objects $i' \in \mathbf{O}^{(2)}$ to objects in $\mathbf{O}^{(1)}$, we define the **transfer costs** $R^T(c^{(1)}, \mathbf{X}^{(2)})$ of a solution c with model-order K as follows:

$$R^T(c^{(1)}, \mathbf{X}^{(2)}) := \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{i=1}^{N_1} R_{i'}(c^{(1)}(i), \mathbf{x}_{i'}^{(2)}, K) \mathbb{I}_{\{\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = i\}}. \tag{23}$$

For each object $i' \in \mathbf{O}^{(2)}$ we compute the costs of i' with respect to the learned solution $c(\mathbf{X}^{(1)})$. The mapping function $\psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ ensures that the cost function treats the measurement $\mathbf{x}_{i'}^{(2)}$ with $i' \in \mathbf{O}^{(2)}$ as if it was the object $i \equiv \psi(i', \mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \in \mathbf{O}^{(1)}$. The minimum transfer cost principle then selects the model order K with lowest transfer costs.

$$K^* = \arg\min_K R^T(c^{(1)}, \mathbf{X}^{(2)}). \tag{24}$$

Please note that the model order K is included in the solution $c^{(1)}$. MTC disqualifies models with a too high complexity that perfectly explain $\mathbf{X}^{(1)}$ but fail to fit $\mathbf{X}^{(2)}$ (overfitting), as well as models with too low complexity which insufficiently explain both of them (underfitting).

3.2.3 *On the Choice of Mapping Function*

In the following, we will describe two mapping variants. In some problems the data source is available such that the measurements can

be sampled/generated multiple times. Given the data source, we sample the measurements twice, once for $\mathbf{X}^{(1)}$ and the other time for $\mathbf{X}^{(2)}$. This so-called *generative* mapping renders an identity mapping between the pairs in $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ and can be used whenever the data source is available.

$$\begin{aligned} \psi^{G} \quad &: \quad \mathcal{O} \to \mathcal{O}\,, \\ &\quad\; i' \mapsto \psi(i') = i\,. \end{aligned} \tag{25}$$

In practice, however, the data is usually generated from an unknown source. One has a single dataset $\mathbf{X}$ and subdivides it (eventually multiple times) into equal-sized random subsets $\{\mathbf{O}^{(1)}, \mathbf{X}^{(1)}\}$ and $\{\mathbf{O}^{(2)}, \mathbf{X}^{(2)}\}$. A mapping is then obtained by assigning each object $i' \in \mathbf{O}^{(2)}$ to a unique object $i \in \mathbf{O}^{(1)}$ using the Hungarian algorithm [Kuh55]. Given the data $\mathbf{X}$, the Hungarian algorithm computes the mapping ψ^{H} by minimizing the sum of the mutual distances.

$$\psi^{H} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{N_1} \sum_{i'=1}^{N_2} d(i, \pi(i')), \tag{26}$$

where Π denotes the set of distinct permutations of the objects in $\mathbf{O}^{(2)}$. Please note that $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ are equal in size, i.e. $N_1 = N_2$.

3.3 MTC ANALYSIS OF FACTORIAL CLUSTERING MODELS

We study the application of MTC to two popular clustering methods, i) Gaussian Mixture Models, and ii) K-means clustering.

3.3.1 *MTC for Gaussian Mixture Models*

We start with Gaussian Mixture Models (GMMs). For this model, the transfer of the learned solution to a second dataset is straightforward and requires no particular mapping function. This case demonstrates that cross-validation for unsupervised learning is a powerful technique that can compete with well known model-selection scores such as BIC and AIC.

A GMM solution consists of the centers $\boldsymbol{\mu}_t$ and the covariances $\boldsymbol{\Sigma}_t$ of the Gaussians, as well as the mixing coefficients π_t. The model order is the number of Gaussians K and the cost function is the negative log likelihood of the model

$$R(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{X}) = -\sum_{i=1}^{N} \ln \left(\sum_{t=1}^{K} \pi_t N(\mathbf{x}_i | \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \right) \tag{27}$$

As all model parameters are independent of the object index i, it is straightforward to compute the transfer costs on a second dataset.

The learned model parameters provide a probability density estimate for the entire measurement space such that the individual likelihood of each new data item can be readily computed. The transfer costs are then

$$R^T(c^{(1)}, X^{(2)}) = R(\mu^{(1)}, \Sigma^{(1)}, X^{(2)}). \tag{28}$$

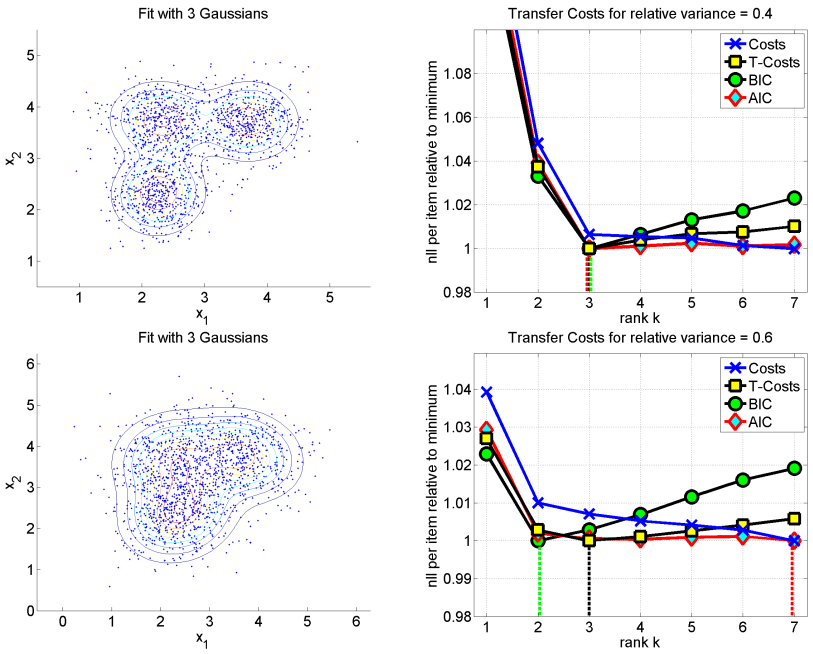


Figure 2: Selecting the number of Gaussians K when the data is generated from 3 Gaussians. The nnl score shows the normalized (w.r.t the context) negative log likelihood. Going from the upper to the lower row, their overlap is increased. For very high overlap, BIC and MTC select $K = 1$. The lower row illustrates the smallest overlap where BIC selects $K < 3$.

We carry out experiments by generating 500 items from three Gaussians. As we increase their variances to increase their overlap, we learn GMMs with varying number of Gaussians K and compute the BIC score, the AIC score, as well as the transfer costs. Two exemplary results are illustrated in Figure 2: an easy setting in the upper row and a difficult setting with high overlap in the lower row. In the easy case, each of the four methods selects the correct number of clusters. For increasing overlap, AIC exhibits a tendency to select a too high number of components. At the variance depicted in the lower plots, BIC starts selecting $K < 3$, while MTC still estimates 3 Gaussians. For very high overlap, we observe that both BIC and MTC select $K = 1$ while AIC selects the maximum number of Gaussians that we offered. The interval of the standard deviation where BIC selects a lower number of Gaussians than MTC ranges from 60% of the distance between the centers (illustrated in Figure 2 bottom) to 85%. The reason for this discrepancy has to be theoretically explored. This gap might be due

to the MTC score being less accurate than the BIC score that is exact in the asymptotic limit of many observations. Maybe, BIC underfits due to non-asymptotic corrections. Visual inspection of the data suggests that this discrepancy regime poses a hard model-order selection problem.

3.3.2 *MTC for K-means Clustering*

Stability analysis of K-means clustering [LBRB04] proposes the mapping of objects from the second dataset to the nearest centroids inferred from the first dataset. We investigate in detail this type of mapping function and compare it to the generative mapping.

A solution c of K-means is an assignment vector $c \in \{1,..,K\}^N$ and K centroids $\mu_t : t \in \{1,..,K\}$. Thereby, $c(i) = t$ means that object i is assigned to cluster t. The model order is the number of centroids K. The cost function of K-means is the sum of distances between each object and its centroid, i.e. $R(c, \mathbf{X}) = \sum_i d\left(\mu_c(i), \mathbf{x}_i\right)$. The distance function $d(.)$ depends on the data type (Hamming, squared Euclidean, ...).

NEAREST-CENTROID MAPPING. Using the nearest centroid mapping, the transfer costs are computed by

$$R^T(c^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{t=1}^{K} d\left(\mu_t^{(1)}, \mathbf{x}_{i'}^{(2)}\right) \mathbb{I}_{\{\psi^{nc}(i', c^{(1)}, \mathbf{X}^{(2)}) = t\}}, \quad (29)$$

where ψ^{nc} is the mapping of objects to the nearest centroid as defined as

$$\begin{aligned} \psi^{nc} \quad : \quad & \mathcal{O} \times \mathcal{C} \times \mathcal{X} \to \mathcal{C}\,, \\ & \left(i', c(\mathbf{X}^{(1)}), \mathbf{X}^{(2)}\right) \mapsto \psi^{nc}\left(i', c(\mathbf{X}^{(1)}), \mathbf{X}^{(2)}\right)\,. \end{aligned} \quad (30)$$

The setup of the experiment is as follows: We sample 200 objects from three bivariate Gaussian distributions (see, for example, Figure 3 top right). The task is to find the appropriate number of clusters. By altering the variances and the pairwise distances of the centers, we control the difficulty of this problem and especially tune it such that selecting the number of clusters is hard. We investigate the selection of K by the nearest-neighbor mapping of the objects from the second dataset to the centroids $\mu^{(1)}$ as well as by the generative mapping where the two data subsets are aligned by construction. We report the statistics over 20 random repetitions of generating the data.

Our findings for three different problem difficulties are illustrated in Figure 3. As expected, the costs on the training dataset monotonically decrease with K. When the mapping is given by the generation process of the data (generative mapping), MTC provides the true number of clusters in all cases. MTC with a nearest-centroid mapping follows almost exactly the same trend as the original costs on

the first dataset and therefore proposes selecting the highest model order that we offer to MTC. The higher the number of clusters is, the closer are the centroids of the nearest neighbors of each object. This reduces the transfer costs of high K. The only difference between original costs and transfer costs stems from the average distance between nearest neighbors (the data granularity). Only when the pairwise centroid distances become smaller than this distance, the transfer costs increase again. Ultimately, the favored solution is a vector quantization at the level of the data granularity. This limit is the natural behavior of K-means, as the cost function, in contrast to GMM, does not include estimating the variances of the clusters. As we have seen in the first experiments with Gaussian mixture models, fitting Gaussian data with MTC imposes no particular difficulties when the appropriate model (here GMM) is used. The K-means behavior is due to a model mismatch.

PROBABILISTIC NEAREST-CENTROID MAPPING. We extend the notion of nearest-centroid mapping to a probabilistic mapping. Let $p_{i't}$ be the probability that ψ^{nc} maps object i' from the second dataset to centroid t inferred from the first dataset, i.e.

$$p_{i't} := \Pr(\psi^{nc}(i', c(\mathbf{X}^{(1)}), \mathbf{X}^{(2)}) = t). \tag{31}$$

We define $p_{i't}$ as

$$\begin{aligned} p_{i'i} &:= Z^{-1} \exp\left(-\beta\ d(\boldsymbol{\mu}_t^{(1)}, \mathbf{x}_{i'}^{(2)})\right), \\ Z &= \sum_{t'} \exp\left(-\beta\ d(\boldsymbol{\mu}_{t'}^{(1)}, \mathbf{x}_{i'}^{(2)})\right). \end{aligned} \tag{32}$$

This mapping distribution is parameterized by the computational temperature β^{-1} and depends on the properties of the problem. A probabilistic mapping is more general than the deterministic function ψ. When β has a finite value, then objects are mapped to more than one centroid. In the case of $\beta \to \infty$, it reduces to a deterministic nearest-centroid mapping. When $\beta = 0$ then object $i' \in O^{(2)}$ is mapped to all K centroids inferred from $\mathbf{X}^{(1)}$ with equal probability, thereby maximizing the entropy of $p_{i't}$.

Using this probabilistic mapping, the nearest-centroid transfer costs $R^T(c^{(1)}, \mathbf{X}^{(2)})$ of a factorial model with model order K are computed as follows:

$$R^T(c^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{N_2} \sum_{i'=1}^{N_2} \sum_{t=1}^{K} d\left(\boldsymbol{\mu}_t^{(1)}, \mathbf{x}_{i'}^{(2)}\right) \frac{e^{-\beta\, d(\boldsymbol{\mu}_t^{(1)}, \boldsymbol{x}_i^{(2)})}}{\sum_{t'=1}^{K} e^{-\beta\, d(\boldsymbol{\mu}_{t'}^{(1)}, \boldsymbol{x}_i^{(2)})}}. \tag{33}$$

A finite choice of the computational temperature β^{-1} extends the dynamic range of mapping objects to centroids and thereby renders

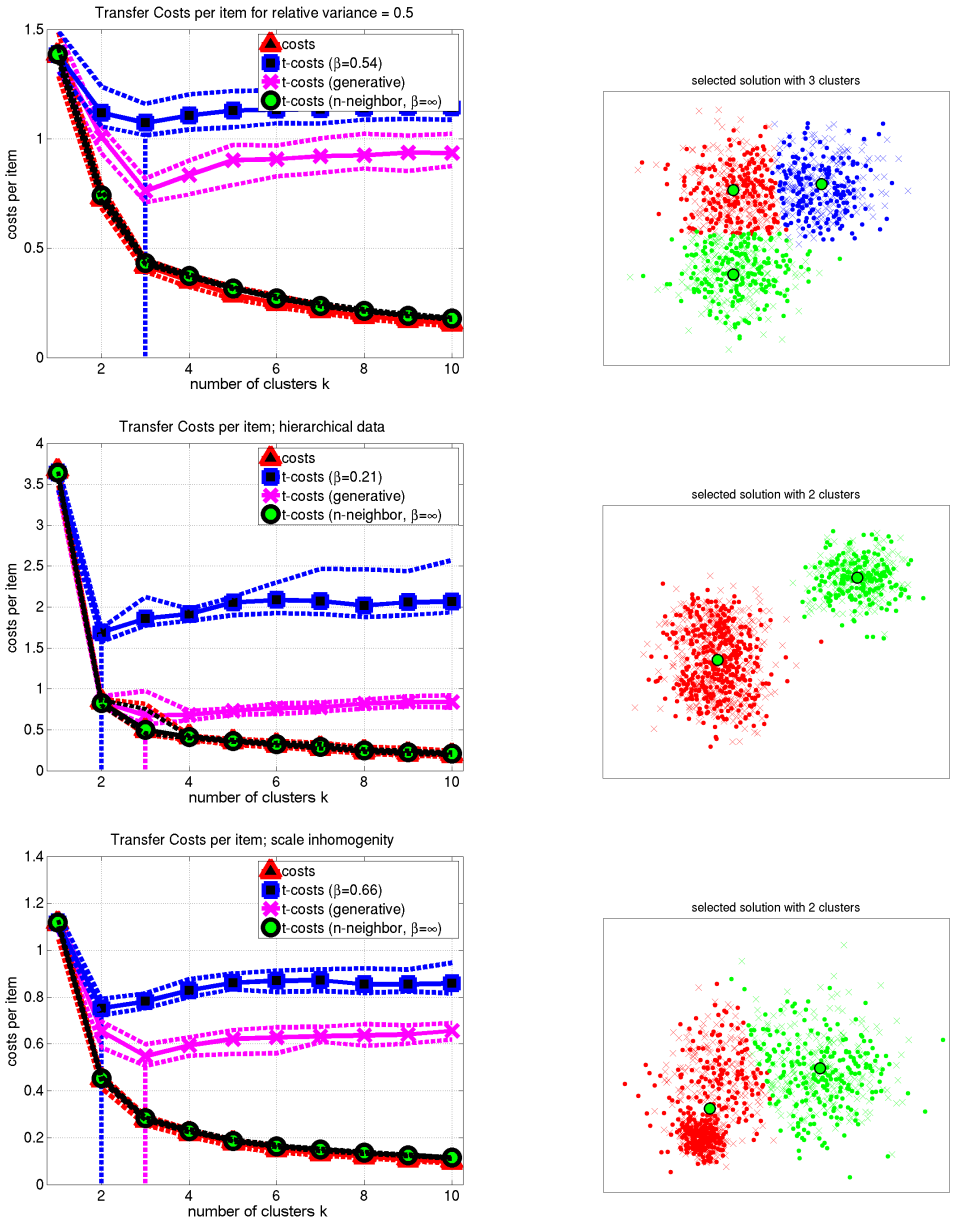


Figure 3: Costs and transfer costs (computed with three mappings: nearest-neighbor, generative, soft) for K-means clustering of three Gaussians. Solid lines indicate the median and dashed lines are the 25% and 75% percentiles. The right panel shows the clustering result selected by soft mapping MTC. Top: equidistant centers and equal variance. Middle: heterogeneous distances between centers (hierarchical). Bottom: heterogeneous distances and variances.

the transfer costs to be high for a too large number of clusters. Essentially, β^{-1} plays the role of the variance in a GMM. Its values is determined by the heuristic of the costs of the data with respect to a single cluster: $\beta = 0.75 * R(c^{(1)} = \mathbf{1}, \mathbf{X}^{(1)})^{-1}$.

EXPERIMENTAL STUDY OF PROBABILISTIC NEAREST-CENTROID MAPPING. The probabilistic mapping finds the true number of clusters when the variances of the Gaussians are roughly the same, even for a substantial overlap of the Gaussians (Figure 3, top row). Please note that although the differences of the transfer costs are within the plotted percentiles, the rank-order of the number of clusters in each single experiment is preserved over the 20 repetitions, i.e. the variance mainly results from the data and not from the selection of K.

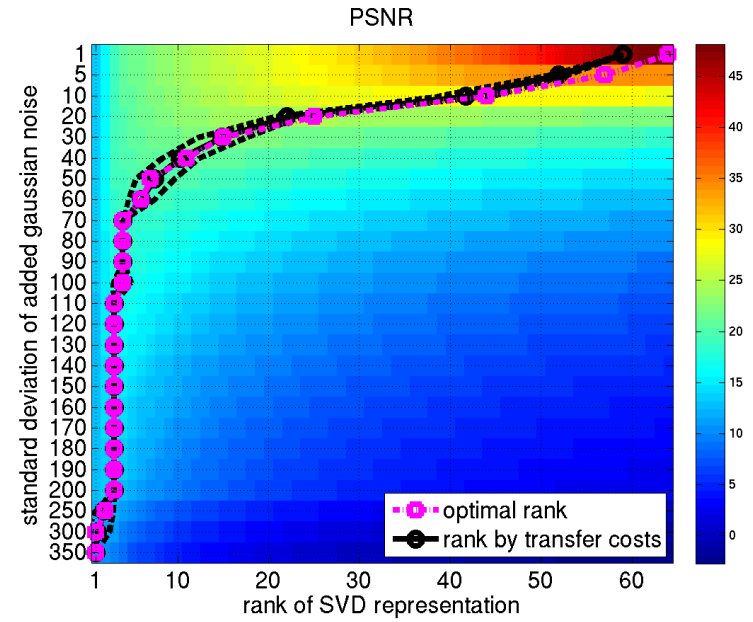
When the problem scale varies on a local level, fixing the temperature at the $K = 1$ solution does not resolve the dynamic range of the costs. We illustrate this by two hard problems: The middle problem in Figure 3 has a hierarchical structure, i.e. the pairwise distances between centers vary a lot. In the bottom problem in Figure 3, both the distances and the individual variances of the Gaussians vary. In both cases the number of clusters is estimated too low. When inspecting the middle plot, this choice seems reasonable, whereas in the bottom plot clearly three clusters would be desirable. The introduction of a computational temperature simulates the role of the variances in Gaussian mixture models. However, as the temperature is the same for all clusters, it fails to mimic situations where the variances of the Gaussians substantially differ. A Gaussian mixture model would be more appropriate than modeling Gaussian data with K-means.

3.4 IMAGE DENOISING WITH RANK-LIMITED SVD

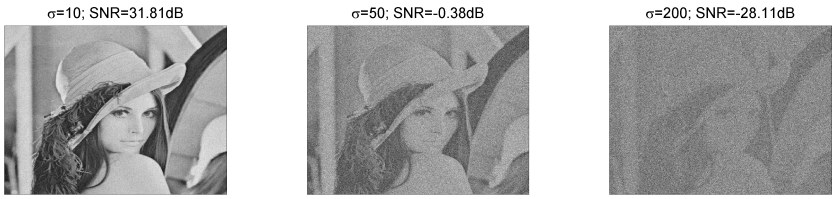
We demonstrate the wide applicability of the MTC principle by investigating rank selection for Singular Value Decomposition (SVD). More applications related to image denoising via SVD and error correction in access-control configurations have been described in [FCB11]. SVD provides a powerful, yet simple method of denoising images. Given a noisy image, one extracts small $N \times M$ patches from the image (where usually $M = N$) and computes a rank-limited SVD on the matrix $\mathbf{X}$ containing the ensemble of all patches, i.e. the pixel values of one patch are one row in $\mathbf{X}$. SVD provides a dictionary that describes the image content on a local level. Restricting the rank of the decomposition, the image content is approximated and, hopefully, denoised.

SVD has been frequently applied to image denoising in the described way or as part of more sophisticated methods (e.g. [EA06]). Thereby, selecting the rank of the decomposition poses a crucial modeling choice. In [EA06], for instance, the rank is selected by experience of the authors and the issue of automatic selection is shifted to further research. Here, we address this specific part of the problem. The task is to select the rank of the SVD decomposition such that the denoised image is closest to the noise-free image. Please note that our goal is not primarily to achieve the very best denoising error given an image (clearly, better image denoising techniques than SVD exist).

Therefore, we do not optimize on other parameters such as the size of the patches. The main goal is to demonstrate that MTC selects the optimal rank for a defined task, such as image denoising, conditioned on a predefined method.



(a) PSNR vs. rank and noise



(b) low noise level (c) middle noise level (d) high noise level

Figure 4: PSNR (logarithmic) of the denoised image as a function of the added noise and the rank of the SVD approximation of the image patches (Figure 4a). The crest of this error marks the optimal rank at a given noise level and is highlighted (dashed magenta). The rank selected by MTC (solid black) is close to this optimum.

We extract $N = 4096$ patches of size $D = 8 \times 8$ from the image and arrange each of them in one row of a matrix $\mathbf{X}$. We randomly split this matrix along the rows into two sub-matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ and select the rank K that minimizes the transfer costs

$$R^T(c, \mathbf{X}) = \frac{1}{N_2} \left\| \psi^H(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \circ \mathbf{X}^{(2)} - \left(\mathbf{U}_K^{(1)} \mathbf{S}_K^{(1)} \mathbf{V}_K^{(1)T} \right) \right\|_2^2 . \quad (34)$$

The mapping $\psi^H(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ reindexes all objects of the test set with the indices of their nearest neighbors in the training set. We illustrate the results for the Lenna image in Figure 4 by color-coding

the peak-SNR of the image reconstruction. As one can see, there is a crest ranging from a low standard deviation of the added Gaussian noise and maximal rank ($K = 64$) down to the region with high noise and low optimal rank ($K = 1$). The top of the crest marks the optimal rank for given noise (dashed magenta line). The rank selected by MTC is highlighted by the solid black line (dashed lines are three times the standard deviation).

The selected rank is always very close to the optimum. At low noise where the crest is rather broad, the deviation from the optimum is maximal. There the selection problem is most difficult. However, in this parameter range the choice of the rank has little influence on the error. For high noise, where a deviation from the optimum has higher influence, our method finds the optimal rank.

3.5 MTC FOR CORRELATION CLUSTERING

The representation of the measurements plays an important role for optimization. In parametric or central clustering, given the parameters of the clusters, the cost function can be written as a sum over independent object-wise costs $R_i(c, \mathbf{x}_i)$ as shown in Eq. 22. When the measurements are characterized by pairwise similarities, instead of explicit coordinates, then such a function form of the costs as in Eqs. 22, 23 does not exist. An example is the cost function for Correlation Clustering [BBC04]. In the following, we explain how to obtain the transfer costs for such models.

Correlation Clustering partitions a graph with positive and negative edge weights. Given a graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ with similarity matrix $\mathbf{X} := \{X_{ij} \in \mathbb{R}\}$ between objects i and j and a clustering solution c, the cost function sums the disagreements, i.e. the sum of negative intra-cluster edge weights plus the sum of positive inter-cluster edge weights:

$$\begin{aligned} R(c, \mathbf{X}) &= \frac{1}{2} \sum_{1 \leqslant u \leqslant K} \sum_{(i,j) \in \mathcal{E}_{uu}} (|X_{ij}| - X_{ij}) \\ &\quad + \frac{1}{2} \sum_{1 \leqslant u \leqslant K} \sum_{1 \leqslant v < u} \sum_{(i,j) \in \mathcal{E}_{uv}} (|X_{ij}| + X_{ij}). \end{aligned} \tag{35}$$

NUMERICAL ANALYSIS OF MTC FOR CORRELATION CLUSTERING

Given the noise parameter η and the complexity parameter ξ, the correlation graph is generated in the following way:

1. Construct a perfect graph with K clusters, i.e. assign the weight $+1$ to all intra-cluster edges and -1 to all inter-cluster edges.
2. Change the weight of each inter-cluster edge in $\mathcal{E}_{uv}, v \neq u$ to $+1$ with probability ξ, increasing structure complexity.

3. With probability η, replace the weight of each edge ($\mathcal{E}_{uv}, v \neq u$ and $\mathcal{E}_{uu}$) by a random weight in $\{-1, +1\}$.

For our experiments, we construct a graph with 900 nodes and 3 clusters. We fix the structure complexity at $\xi = 0.30$ and vary the noise level η from 0.7 to 0.95. We then divide the graph into two smaller graphs of identical cardinality $N_1 = N_2 = 450$. We use the Hungarian method to map the objects of $\mathbf{O}^{(2)}$ to $\mathbf{O}^{(1)}$. The label of object $i' \in \mathbf{O}^{(2)}$ is determined by the cluster index of the corresponding object $i \in \mathbf{O}^{(1)}$. For clustering, we use Gibbs sampling since, according to our experiments, it usually achieves lower costs than approximation algorithms such as CC-Pivot [ACN08]. We run the sampler with a number of clusters varying from 1 to 10 each for 10 different random initializations. We compare the transfer costs with the instability measure proposed in [LBRB04]. The results are summarized in Figure 5.

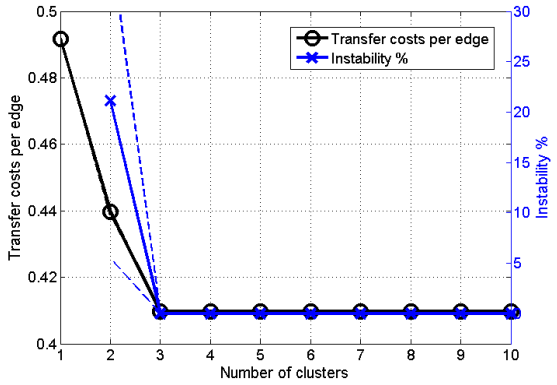
At $\eta = 0.70$ the problem is simple, which means that the Gibbs sampler, even when initialized with a large number of clusters, always selects the correct number of clusters on its own. The extra clusters are simply left empty. As a consequence, the transfer costs are indifferent for a number of clusters larger than or equal to the correct number (Figure 5a). At $\eta = 0.80$ the problem is complicated but still learnable. Here, the inferred clustering and also the transfer costs vary for different choices of the number of clusters. As illustrated in Figure 5b MTC selects the true number of clusters.

For both $\eta = 0.70$ and $\eta = 0.80$ the instability measure is consistent with the transfer costs. At $\eta = 0.95$ the edge weights are almost entirely random, hiding all structure in the data. Therefore, as Figure 5c confirms, the number of learnable clusters is 1. In this regime, instability cannot determine the correct number of clusters as it is not defined for $K = 1$.

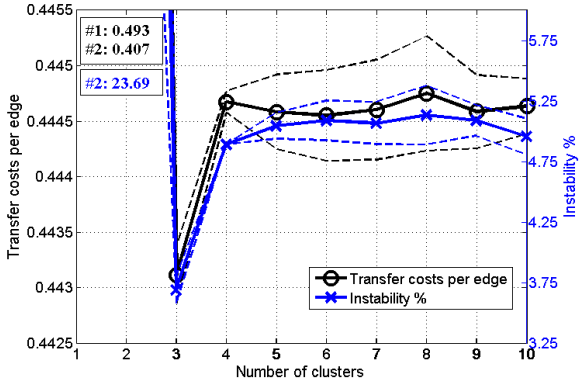
3.6 CLUSTER ANALYSIS OF GENE EXPRESSION DATA

In this section, we employ the MTC principle to compute the optimal number of clusters for experimental gene expression profiles. The data comes from the female digestive gland of *Mytilus galloprovincialis* [BNM+11], an organism studied to assess the impact of environmental pollutants. Time points correspond to relative gene expression measurements for 12 consecutive months. The first sample, which corresponds to January, is defined as reference. Logarithmic values are obtained for the 295 differentially expressed genes. The biological goal is to study the influence of the seasonal environmental changes on physiology across the annual cycle of the organism [BNM+11].

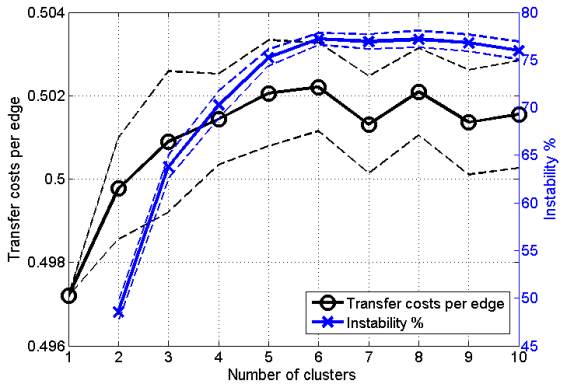
Taking advantage of the temporal structure of the data, the two object sets $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ are constructed by splitting the feature vector. Thereby, the measurements for the months *(Feb, Mar, Apr, May, Jun,*



(a) $\eta = 0.70$



(b) $\eta = 0.80$



(c) $\eta = 0.95$

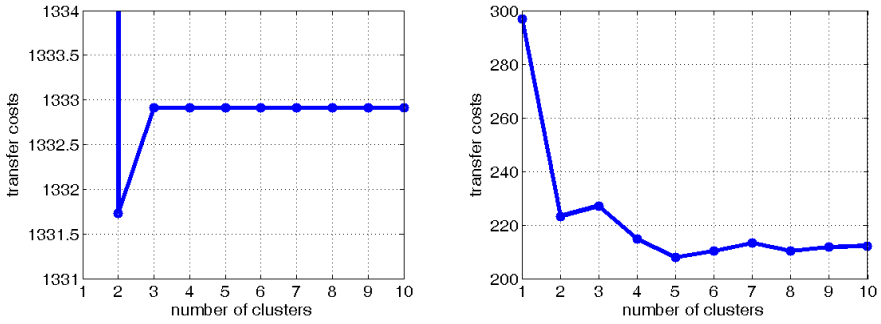
Figure 5: Transfer costs and instability measure for various noises η in Correlation Clustering. The model complexity ξ is kept fixed at 0.30. For both $\eta = 0.70$ and $\eta = 0.80$ the instability measure is consistent with the transfer costs. At $\eta = 0.95$ the edge labels are almost entirely random, hiding all structure in the data. Therefore, the number of learnable clusters is 1. In this regime, instability cannot determine the correct number of clusters as it is not defined for $K = 1$.

Jul, Aug, Sep, Oct, Nov, Dec) are separated into the values for *(Mar, May, Jul, Sep, Nov)* and for *(Apr, Jun, Aug, Oct, Dec)*. This interleaved separation captures the statistical dependence of the samples due to time proximity. Thus it avoids the risk of under sampling small clusters of high biological relevance (as in this study) by having too few genes per cluster. Pearson correlation coefficients are calculated for each pair of genes in each set to construct the similarity matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

We analyze the data by performing Pairwise Clustering (PC) and Correlation Clustering (CC) models. Given a set of pairwise similarities $\mathbf{X}$, the Pairwise Clustering cost function is defined as [HB97]

$$R^{PC}(c, \mathbf{X}) = -\frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} \frac{X_{ij}}{|\mathbf{O}_k|}. \tag{36}$$

This cost function sums the average similarities per cluster (weighted by the respective cluster sizes). If we appropriately convert similarities $\mathbf{X}$ into dissimilarities (e.g. by transformation $\mathbf{D} = \text{constant} - \mathbf{X}$), then Pairwise Clustering equivalently performs K-means clustering in kernel space [RLKB03].

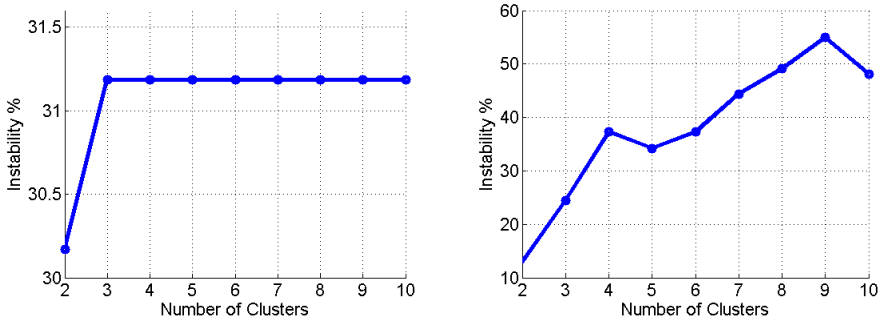


(a) Transfer costs for CC

(b) Transfer costs for PC

Figure 6: Transfer costs for Correlation Clustering and for Pairwise Clustering: The MTC principle validates 2 clusters for Correlation Clustering and 5 clusters for Pairwise Clustering.

For each method we compute the transfer costs for different number of clusters from 1 to 10. Figures 6a and 6b show the transfer costs respectively for Correlation Clustering and Pairwise Clustering. The MTC principle computes two clusters for Correlation Clustering and five clusters for Pairwise Clustering. However, the transfer costs reveal a local minimum at $K = 2$, which might be related to the computation of the two clusters of Correlation Clustering. For this dataset, Correlation Clustering computes maximally three clusters even if the Gibbs sampler is initialized with more clusters. The superfluous clusters are left empty. Figures 7a and 7b show the instability index respectively for Correlation Clustering and Pairwise Clustering. For



(a) Instability index for CC (b) Instability index for PC

Figure 7: Instability index for Correlation Clustering and for Pairwise Clustering: The instability analysis validates 2 clusters for both methods. Instability favors a small number of clusters.

both methods, instability index finds only two clusters. Instability renders a bias towards a smaller number of clusters.

3.7 SCOPE OF APPLICABILITY

We introduced the *Minimum Transfer Costs* (MTC) principle for model order selection which extends the concept of cross-validation to unsupervised learning problems. We studied in detail transferring a learned model from one dataset to another when no labels are provided. We investigated the principle on different problems such as maximum likelihood inference, K-means, Correlation Clustering and Singular Value Decomposition. MTC exhibits a wider applicability than classic principles such BIC. For instance, it can be employed to compute the optimal number of clusters even when the effective dimensionality is unclear. Correlation Clustering is an example of such models.

Despite the broad applicability, however, the MTC principle cannot address all model validation questions. In particular,

1. The applicability of MTC is limited to the models specified by a cost function. For example, it is not clear how MTC can be applied to algorithms which do not explicitly optimize a cost function.

2. The principle can only be employed for selecting the optimal model order inside a limited family of models. MTC, for example, does not discriminate between Pairwise Clustering and Correlation Clustering, the two alternative methods used to analyze the gene expression data.

MTC advocates to use generalization performance in unsupervised learning. It is related to a more fundamental concept in learning theory: *What is the the optimal trade-off between stability and informativeness?*

Increasing stability reduces informativeness and vice versa. To obtain the optimal trade-off, a recent principle, called *Approximation Set Coding* [Buh10], suggests to compute a set of statistically indistinguishable solutions $\mathcal{C}_\gamma(\mathbf{X})$, instead of seeking for the empirical minimizer. This solution set approximates the average of the costs and is defined as:

$$\mathcal{C}_\gamma(\mathbf{X}) := \{c \in \mathcal{C} : R(c, \mathbf{X}) \leqslant R(c^\perp, \mathbf{X}) + \gamma\}. \tag{37}$$

γ determines the resolution of the solution space, i.e. controls the trade-off between stability and informativeness. Optimal trade-off is achieved by establishing a conceptual communication scenario and maximizing the information rate over channel. In the next chapter, we will elaborate this principle.

4

GENERALIZATION CAPACITY: AN INFORMATION-THEORETIC PRINCIPLE FOR OPTIMAL LEARNING

We elaborate *Generalization Capacity* ($\mathcal{GC}$)[1], a new principle for analyzing algorithms. This principle opens a new perspective incorporating the *informativeness* of algorithmic procedures and their *stability* against noise. An algorithm is considered to be a noisy channel which demands a high, but stable information rate. Informative algorithms yield a high generalization capacity, whereas fragile algorithms suffer from a low capacity.

We analytically prove the consistency of the generalization capacity with the Shannon capacity for K-ary codes. However, $\mathcal{GC}$ constitutes a more general principle which is applicable to any arbitrary data processing mechanism. $\mathcal{GC}$ objectively ranks different algorithms for the same data processing task based on the bit rate of their respective capacities.

4.1 INFORMATIVENESS-STABILITY DILEMMA

The first step of a data processing task involves converting the *data* into *information* (Figure 8). Information is then further processed to produce *knowledge*. *Algorithms*[2] are the tools for processing the data

Figure 8: The pipeline of extracting information from data: The information is then converted into knowledge to support decision makers.

and filtering out the information (Figure 9). Information is technically represented by elements of a solution space $\mathcal{C}$. Thereby, an algorithm $\mathcal{A}$ in the classical view accepts some input data $\mathbf{X}$, it performs algo-

1 The derivation of the generalization capacity principle relies very much on the works of my advisor, Joachim M. Buhmann, on information-theoretic model validation, and in particular, on Approximation Set Coding. The conceptual set-based communication scenario and the analysis of decoding error, have been developed and derived in [Buh10]. Our contributions include the generalization to algorithms and the calculations for K-ary codes. The presentation of generalization capacity is adapted from [Bus12].

2 We consider algorithms as arbitrary data processing mechanisms, e.g. optimization of a cost function, a computer routine or a dynamical system. According to the convention of Chapter 1, an algorithm includes both modeling and inference (execution) steps together.

rithmic operations on the data, and after a finite number of steps, it produces an output $c \in \mathcal{C}$, i.e.

$$\boxed{\text{input } \mathbf{X} \rightarrow \mathcal{A} \rightarrow \text{output } c}.$$

This definition encompasses any data processing mechanism, e.g. statistical models, computer algorithms, and heuristic methods.

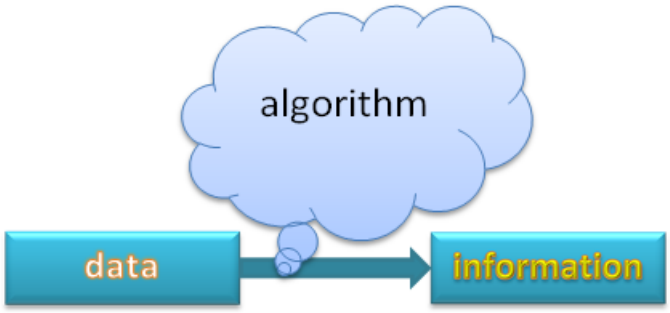


Figure 9: Algorithms are data processing mechanisms that extract the information from structured and possibly noisy data.

The data is usually contaminated by *noise*. This renders the empirical output of an algorithm to vary among different instances drawn from a unique source. We aim to design algorithms that generate consistent solutions, i.e. they provide *stability* of the solution under the data fluctuations. Abundant results demonstrate the power of stability analysis for model order selection [LBRB04]. In the same spirit, PAC-Bayesian generalization bounds have been derived for different clustering models [ST10a]. Appeal to stability in data analysis is related to two fundamental scientific concepts: i) mathematical well-posedness [KT01], and ii) experimental repeatability, i.e. good generalization capability [Vap98]. In principle, controlled experiments are expected to obtain similar outcomes under similar (controlled) conditions.

However, stability is only one aspect of a learning procedure. Unrestricted maximization of the stability might lead to a lack of *informativenss*. As in the case of clustering, producing only one single cluster is maximally stable but it carries no information.

However, overestimating the information increases the overfitting risk. As mentioned, the explanatory power of models comes from their ability to generalize well. Thereby, quantifying the "information content" of a model constitutes a necessary step to obtain the optimal trade-off between stability and informativeness. Reasonable reductions of stability are acceptable, as long as they are compensated by more expressive representations [TPB99]. *Approximation Set Coding* (ASC) [Buh10, Buh11] computes the optimal trade-off by establishing a conceptual set-based communication scenario. ASC suggests selecting a set of statistically indistinguishable solutions (*approximation set*) instead of a single empirical output, in order to provide stability. In

fact, the empirical solution for a given dataset might not coincide exactly with the solution selected by another dataset from the same source. By selecting a large set of solutions the chance of coincidence increases. Sets consisting of many solutions are more stable under data fluctuations, but might not be informative enough. ASC computes the optimal trade-off by maximizing the reliable information transfer between datasets.

Based on the ASC framework, we elaborate a measure of *context-sensitive information* for algorithms, which computes the amount of reliable information that an algorithm can extract from the data. The uncertainty in the data increases the size of the solution set and thereby reduces the resolution of the solutions space. The maximum information rate of the set-based communication scenario, called *Generalization Capacity* ($\mathcal{GC}$), gives the optimal resolution of the solution space.

Generalization capacity can be viewed as a general-purpose validation principle to rank alternative algorithms for solving a learning question. It can be applied not only to statistical models like e.g. log-likelihoods or cost functions, but also to general descriptive formalisms (e.g. in form of algorithms or dynamical systems).

4.2 THE GENERALIZATION CAPACITY PRINCIPLE

New machine learning applications require a *context sensitive* principle that measures the information content of alternative algorithms. In this section, we elaborate generalization capacity as a general-purpose model validation principle.

4.2.1 *Weighted Outputs of Algorithms*

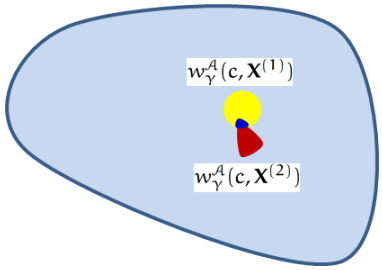
The notion of *approximation set* can be extended to a *weight distribution* over the hypothesis class. Thereby, suitable hypotheses to interpret data, which amounts to condensing data to relevant information, are selected by a weighting function

$$\begin{aligned} w \quad &: \quad \mathcal{C} \times \mathcal{X} \times \mathbb{R} \to [0,1], \\ &\quad\quad (c, \mathbf{X}, \gamma) \mapsto w_\gamma(c, \mathbf{X}). \end{aligned} \tag{38}$$

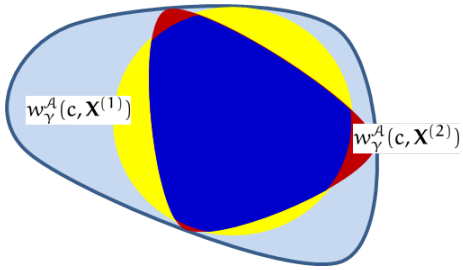
The weights $w_\gamma(c, \mathbf{X})$ might be interpreted to encode the belief that a hypothesis c yields an acceptable solution or sufficient interpretation of the data $\mathbf{X}$. Learning now pursues the goal to concentrate the weights on a small subset of satisfactory hypotheses. This concentration is controlled by the resolution parameter γ:

- A small value of γ encodes for a sharp concentration of solutions (around e.g. the empirical minimizer) – providing *informative*, however possibly unstable solutions (Figure 10a).

- A large value of γ implies a wide weight distribution, thereby yields more *stable* solutions at the price of loosing information about the actual localization of the solution (Figure 10b).



(a) Informative but not stable solution set.



(b) Stable but not informative solution set.

Figure 10: Informativeness and stability dilemma controlled by the weighting parameter γ: The goal is to compute the optimal concentration of the weight distribution in order to provide stable and informative solutions.

The ability to localize the weights (favoring *informative* solutions) is therefore balanced by the constraint that the subset of hypotheses with large weights is *stable* under data fluctuations. In case of error free data, the weights assume the binary values $w_0(c, \mathbf{X}) = 1$ if $c = c^\perp$ and $w_0(c, \mathbf{X}) = 0$ otherwise.

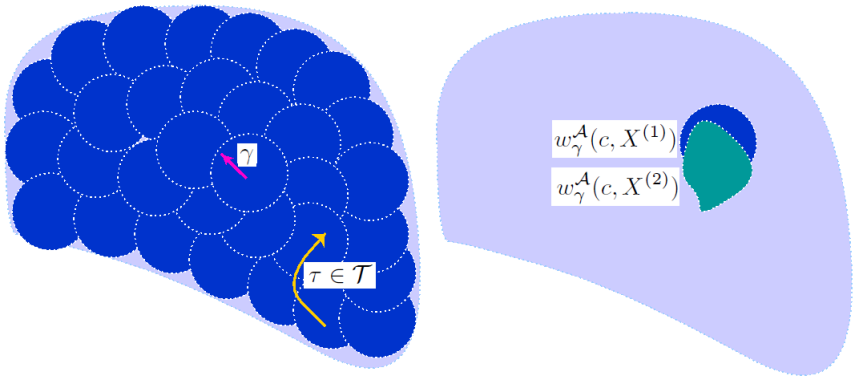
Contrary to the classical view, algorithms return not a unique output hypothesis, but a weight distribution over the hypothesis class. More formally, an **algorithm** $\mathcal{A}$ defines a mapping of the hypothesis class to the space of weight distributions $\mathcal{W}$, i.e.

$$\begin{aligned} \mathcal{A} : \mathcal{C} \times \mathcal{X} \times \mathbb{R} &\to \mathcal{W} \\ (c, \mathbf{X}, \gamma) &\mapsto \mathcal{A}(c, \mathbf{X}, \gamma) =: w_\gamma^{\mathcal{A}}(c, \mathbf{X}). \end{aligned} \tag{39}$$

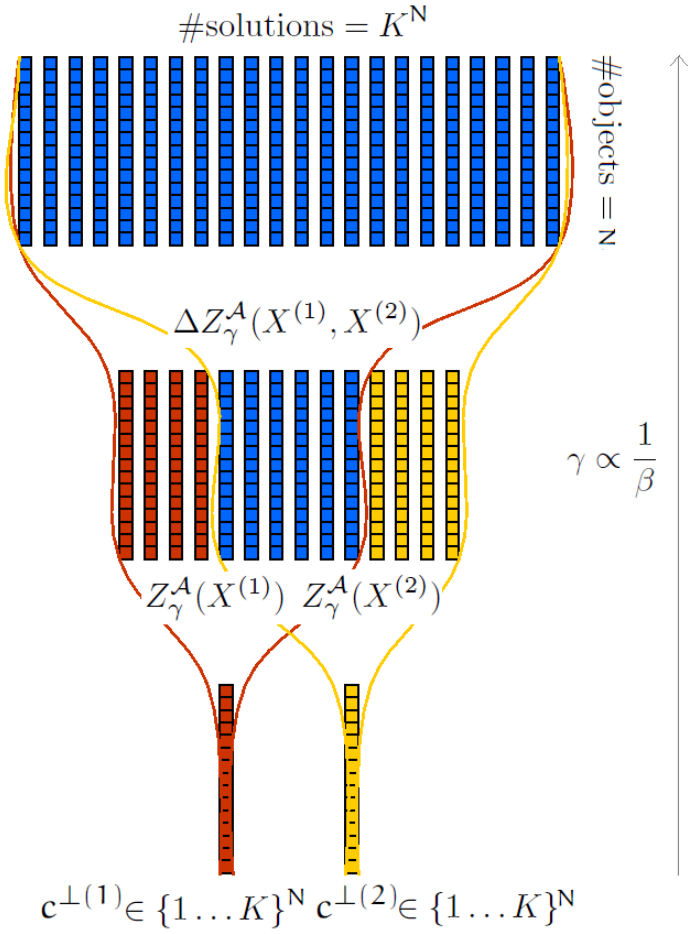
Under this viewpoint, algorithms are mappings between an input and a weight distribution over the solution space:

$$\boxed{\text{input } (\mathbf{X}, \gamma) \to \mathcal{A} \to \text{output } w_\gamma^{\mathcal{A}}(c, \mathbf{X})}.$$

This setup will enable us to quantify the amount of reliable information – measured w.r.t. the hypothesis class $\mathcal{C}(\mathbf{X})$ – that $\mathcal{A}$ is able to



(a)



(b)

Figure 11: Top left: The hypothesis class, and a γ-relaxed localization of sufficient solutions. The set of transformations $\mathcal{T}$ cover the solution space by all distinct solutions. Top right: Two weight distributions for the same problem in $\mathcal{A}$ at resolution γ, but different noise effects in data realizations $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Bottom: Evolution of the distribution of the weights for clustering algorithms steered by γ.

extract from the data $\mathbf{X}$, e.g. the ability of a model to localize a solution in the hypothesis class. The goal is to localize hypotheses finely (to extract a lot of information), but identifiably (for robustness). The empirical output of the algorithm, e.g. the empirical minimizer, exhibits the maximal informativeness, but it might suffer from a lack of stability.

For a clustering method, *informativeness* means identifying the actual cluster labels of objects (i.e., increasing the concentration of weights to more specific solutions; visualization of the principle will be developed in Figure 11b), while *stability* calls for a large overlap between solutions of noise-affected problem instances.

4.2.2 *Optimal Concentration of Weights*

Learning now amounts to optimally coarsening the hypothesis class by weight distributions. Mathematically, the problem rephrases as: *How concentrated can the weight distribution be chosen to still ensure identifiability of* $w_\gamma^{\mathcal{A}}(c, \mathbf{X})$*'s under variation of the data* $\mathbf{X}$? To answer this question, we refer to a generic set-based coding and communication scenario to derive a criterion for the optimal information gain. There exists a conceptual analogy between learning and communication: For communication, one demands a high rate together with a decoding rule that is stable under the perturbations of the message by channel noise.

We adopt the two sample set scenario, that is widely used in statistical learning theory [Vap82] and that is analogous to two-terminal communication systems in information theory [CK11]. Two datasets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are available with the same underlying signal structure (i.e. correspond to the same objects $\mathbf{O}$), but different noise realizations. They are employed to estimate the generalization performance of the quantization induced by γ. The weight distributions $w_\gamma^{\mathcal{A}}(c, \mathbf{X}^{(1)})$, $w_\gamma^{\mathcal{A}}(c, \mathbf{X}^{(2)})$ may differ: some weights may assume similar values for the two datasets, while other weights may differ (Figure 11a right). A large overlap means that the weights of solutions for the instance $\mathbf{X}^{(1)}$ generalize well to the instance $\mathbf{X}^{(2)}$, whereas little agreement between the two weight distributions indicates a lack of generalization.

The weight distributions $w_\gamma^{\mathcal{A}}(c, \mathbf{X}^{(q)}), q = 1, 2$ are used in a conceptual **communication** scenario. A high rate is achieved with a large concentration of weights. On the other hand, messages should be decodable with vanishing error probability, which calls for a large overlap.

The Sender-Receiver scenario [Buh10] exploits the ability to localize hypotheses given the data to transmit a codeword through a channel. The scenario consists of two stages, the coding and the communication phase (see also Figure 12):

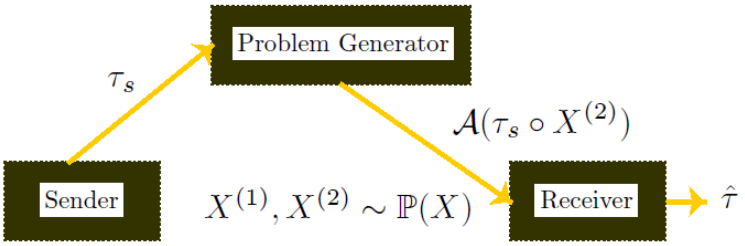


Figure 12: Organization of the communication process

I. CODING is employed to generate the code for the communication process:

1. Sender and receiver know the algorithm $\mathcal{A}$ and share the dataset $\mathbf{X}^{(1)}$.

2. Transformations $\tau : \mathcal{O}^N \to \mathcal{O}^N$ that map sets of N objects to sets of N objects are defined to generate alternative algorithmic problems given the same measurements. The set of transformations $\mathcal{T}$ is then generated.

$$\mathcal{T} = \{\tau_1, \ldots, \tau_{2^{N\rho}}\}, \quad \tau_i \in \mathbb{T}, \tag{40}$$

   where $\mathbb{T}$ denotes the space of transformations[3] and ρ parameterizes the rate of the code.

   By applying $\mathcal{T}$ to the dataset $\mathbf{X}^{(1)}$, all distinct weight distributions $\{w_\gamma^{\mathcal{A}}(c, \tau \circ \mathbf{X}^{(1)}), \tau \in \mathcal{T}\}$ are constructed and are shared between sender and receiver. After this setup procedure, both sender and receiver have a list of weight distributions covering the solution space (Figure 11a left). The weight distributions play the role of codebook vectors in Shannon's theory of communication.

II. TRANSMISSION Under this framework, the weight distributions are indexed by a transformation τ (i.e. $\{w_\gamma^{\mathcal{A}}(c, \tau \circ \mathbf{X}^{(1)}), \tau \in \mathcal{T}\}$) and these indices are used as codewords:

1. The sender selects a transformation τ_s as message, without revealing it to the receiver.

2. The sender applies τ_s to the second dataset $\mathbf{X}^{(2)}$. Then, $\tau_s \circ \mathbf{X}^{(2)}$ is sent to the receiver.

3. The receiver has to reconstruct the transformation τ_s without knowing the two components $\mathbf{X}^{(2)}$ and τ_s from the set of weight distributions $\{w_\gamma^{\mathcal{A}}(c, \tau \circ \mathbf{X}^{(1)}), \tau \in \mathcal{T}\}$.

3 In clustering, $\mathbb{T}$ is the set of permutations that generate new distinct solutions when applied to the cluster indices of the objects.

The receiver estimates the transformation τ_s that has been selected by the sender through the decoding rule

$$\hat{\tau} = \arg\max_{\tau \in \mathcal{T}} \left| \sum_{c \in \mathcal{C}} w_\gamma^{\mathcal{A}}(c, \tau \circ \mathbf{X}^{(1)}) \cdot w_\gamma^{\mathcal{A}}(c, \tau_s \circ \mathbf{X}^{(2)}) \right|. \tag{41}$$

Despite not knowing τ_s and $\mathbf{X}^{(2)}$, the receiver knows $\tau_s \circ \mathbf{X}^{(2)}$ and it can estimate the transformation $\hat{\tau}$ which yields the largest weight overlap between the two weight distributions.

4.2.3 *Generalization Capacity*

Optimal communication is informative and reliable. For a high information throughput, a large set of transformations $\mathcal{T}$ should be used and the weight distributions should be concentrated (fine resolution of the space of hypothesis). However, the receiver has to estimate the weight distribution that was selected by the sender. Thereby the weight distributions need to be sufficiently broad to yield a non-vanishing overlap (see Figure 10). A vanishing error probability for decoding on the receiver side is only achievable for an adequate resolution. The optimal resolution should match the quantization induced by the fluctuations in the noisy data *in the solution space*. As depicted in Figure 11a, the hypothesis space is partitioned (quantized) into equivalence classes of statistically indistinguishable (equivalent) solutions. A low noise level enables a fine resolution, whereas a high noise level requires a coarsening of the hypothesis space. Hypotheses with approximately the same weights cannot be distinguished and have to be considered as statistically equivalent solutions to the inference problem.

The criterion for reliable communication is the ability of the receiver to identify the specific transformation that has been selected by the sender. For large γ, the rate ρ will be low since we resolve the solution space in only a coarse-grained fashion. For too small γ, the error probability does not vanish, which indicates confusions between τ_s and $\tau_j, j \neq s$. Communication errors are caused by erroneous decoding (i.e. the sender selects τ_s and the receiver decodes $\hat{\tau} = \tau_j, j \neq s$). The asymptotic error analysis of such a communication scenario [Buh10] yields a criterion $\mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ to be optimized, i.e.

$$\mathbf{P}(\text{error}) \leqslant \exp\left(-N\left(\mathbb{E}\hat{\jmath}_\gamma^{\mathcal{A}} - \rho\right)\right) \tag{42}$$

$$\hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{N} \log \frac{|\mathbb{T}| \cdot \Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})}{Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}) \cdot Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(2)})}. \tag{43}$$

The expectation in Eq. 42 is calculated for the random variables $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$ and the hat denotes an empirical variable. The rate pa-

rameter ρ should not exceed the function $\mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}}\hat{\mathcal{I}}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)})$ for asymptotically error free communication. Maximizing $\mathbb{E}\hat{\mathcal{I}}_\gamma^{\mathcal{A}}$ w.r.t. γ, therefore, will yield the optimal rate and, equivalently the optimal resolution of the hypothesis class.

$|\mathbb{T}|$ denotes the cardinality of the transformation space and is the term accounting for informativeness. $Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(q)}), q = 1,2$ define the weight sums for the two datasets.

$$Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(q)}) = \sum_{c\in\mathcal{C}} w_\gamma^{\mathcal{A}}(c,\mathbf{X}^{(q)}) \ , \ q = 1,2. \tag{44}$$

$\Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)})$ computes the overlap, i.e. how many large weight solutions of the first dataset also share a large weight in the second dataset (the joint weight sum). Thereby,

$$\Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)}) = \sum_{c\in\mathcal{C}} w_\gamma^{\mathcal{A}}(c,\mathbf{X}^{(1)})\cdot w_\gamma^{\mathcal{A}}(c,\mathbf{X}^{(2)}). \tag{45}$$

For a fixed γ, a large overlap of the weight distributions indicates that the algorithm's evaluation of the first dataset generalizes well to the second dataset, whereas a small or empty overlap denotes lack of generalization. The fraction $\Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)})/Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)})Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(2)})$ measures the stability of the solutions under noise fluctuations. It is in $[0,1]$ and thereby controls the effective useable size of the hypothesis class. The normalization of $\hat{\mathcal{I}}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)})$ with N ensures a bit rate per object.

The maximum of $\hat{\mathcal{I}}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)})$ is called **generalization capacity** and determines the best trade-off between stability and informativeness.

$$\mathcal{GC}^{\mathcal{A}} := \max_\gamma \mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}}\hat{\mathcal{I}}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)},\mathbf{X}^{(2)}). \tag{46}$$

This criterion allows us to rank a collection of algorithms $\{\mathcal{A}^1(\mathbf{X}),\ldots,\mathcal{A}^l(\mathbf{X})\}$ according to their informativeness. The algorithm or model to be selected is

$$\mathcal{A}^\star(\mathbf{X}) = \arg\max_{1\leqslant j\leqslant l} \mathcal{GC}^{\mathcal{A}^j}. \tag{47}$$

The selection rule (47) prefers the algorithm or model which is "expressive" enough to exhibit high information content and, at the same time, robustly resists to noise in the data. The bits which are measured here are context-sensitive, since they refer to a hypothesis class $\mathcal{C}(\mathbf{X})$, i.e. how finely or coarsely functions can be resolved in $\mathcal{C}$.

4.3 GENERALIZATION CAPACITY AND SHANNON CAPACITY

Here, we prove the consistency of generalization capacity with Shannon capacity for K-ary codes, a well studied problem in communication theory.

4.3.1 *Generalization Capacity of* K*-ary Codes*

In the communication scenario for K-ary codes, the sender forwards the codeword $\xi^{(1)} = (\xi^{(1)}(1), \dots, \xi^{(1)}(N)) \in \{1, \dots, K\}^N$ over the noisy channel. The receiver receives the string $\xi^{(2)} = (\xi^{(2)}(1), \dots, \xi^{(2)}(N)) \in \{1, \dots, K\}^N$. There might exist some discrepancies between $\xi^{(1)}$ and $\xi^{(2)}$ due to the noise in the channel.

If the channel is noise-free, then $\xi^{(2)}$ would be identical to $\xi^{(1)}$, and thereby the channel capacity per symbol would be $\log K$. However, in a noisy channel, the probability $\Pr(\xi^{(1)} \neq \xi^{(2)})$ is non-zero, which yields a reduction in the channel capacity. We compute the generalization capacity of such a communication scenario and compare it with Shannon capacity.

The hypothesis class is the set of strings with length N, i.e.

$$\mathcal{C} = \{1, ..., K\}^N. \tag{48}$$

Given the reference codeword $\xi^{(1)}$, the costs of matching the string $\mathbf{s} \in \{1, \dots, K\}^N$ with $\xi^{(1)}$ are defined as the number of differing components, i.e.

$$R(\mathbf{s}, \xi^{(1)}) = \sum_{i=1}^{N} \mathbb{I}_{\{s(i) \neq \xi^{(1)}(i)\}}. \tag{49}$$

The empirical minimizer is $\mathbf{s}^{\perp(1)} = \xi^{(1)}$. However, $\mathbf{s}^{\perp(1)}$ is exposed to channel noise and, it is not stable.

To set up the communication protocol and generate the codebook, the sender applies the set of all transformations $\mathcal{T} = \{\tau_1, \dots, \tau_{2^{N\rho}}\}$ to the string $\xi^{(1)}$ to generate the set of weight distributions

$$w_\gamma(\mathbf{s}, \tau_j \circ \xi^{(1)}),\ 1 \leqslant j \leqslant 2^{N\rho} \tag{50}$$

Each τ_j corresponds to a distinct permutation of the reference string $\xi^{(1)}$. The set of the weight distributions serves as a communication codebook.

Without loss of generality, we assume that the transformation τ_s selected by the sender for communication is the identity transformation, i.e. $\tau_s \circ \xi^{(1)} = \xi^{(1)}$. During the communication step, the receiver then receives the string $\tilde{\xi} = \xi^{(2)}$.

The weight distributions are computed according to the matching costs, i.e.

$$w_\gamma(\mathbf{s}, \xi^{(q)}) = \exp\left(-\beta \sum_{i=1}^{N} \mathbb{I}_{\{\mathbf{s}(i) \neq \xi^{(q)}(i)\}}\right), \quad q = 1, 2. \tag{51}$$

where $\beta = 1/\gamma$ controls the width of the distributions.

Let δ be the noise rate of the channel. The receiver then receives the string $\xi^{(2)}$ with an average distance $\hat{\delta}N$ to the sent codewords $\xi^{(1)}$.

The weight sums $Z_\beta(\xi^{(q)})$, $q = 1, 2$ are given by

$$\begin{aligned} Z_\beta(\xi^{(q)}) &= \sum_{s\in\mathcal{C}} \exp\left(-\beta \sum_{i=1}^{N} \mathbb{I}_{\{s(i)\neq\xi^{(q)}(i)\}}\right) \\ &= \prod_{i=1}^{N} \sum_{s(i)\in\{1\ldots K\}} \exp\left(-\beta\, \mathbb{I}_{\{s(i)\neq\xi^{(q)}(i)\}}\right) \\ &= [(K-1)\exp(-\beta)+1]^N . \end{aligned} \tag{52}$$

Similarly, the joint weight sum $\Delta Z_\beta(\xi^{(1)}, \xi^{(2)})$ is computed as

$$\begin{aligned} &\Delta Z_\beta(\xi^{(1)}, \xi^{(2)}) \\ = &\sum_{s\in\mathcal{C}} \exp\left(-\beta\left(\sum_{i=1}^{N} \mathbb{I}_{\{s(i)\neq\xi^{(1)}(i)\}} + \sum_{i=1}^{N} \mathbb{I}_{\{s(i)\neq\xi^{(2)}(i)\}}\right)\right) \\ = &\prod_{\substack{i\leqslant N \\ \xi^{(1)}(i)=\xi^{(2)}(i)}} ((K-1)\exp(-2\beta)+1) \times \\ &\prod_{\substack{i\leqslant N \\ \xi^{(1)}(i)\neq\xi^{(2)}(i)}} (2\exp(-\beta)+(K-2)\exp(-2\beta)) \\ = &\ [1+(K-1)\exp(-2\beta)]^{N(1-\hat{\delta})} \times \ [2\exp(-\beta)+(K-2)\exp(-2\beta)]^{N\hat{\delta}} . \end{aligned} \tag{53}$$

$\hat{\mathcal{J}}_\beta$ is then written by

$$\begin{aligned} \hat{\mathcal{J}}_\beta &= \log K - \beta\hat{\delta} - 2\log(1+(K-1)\exp(-\beta)) \\ &\quad +(1-\hat{\delta})\log(1+(K-1)\exp(-2\beta)) \\ &\quad +\hat{\delta}\log(2+(K-2)\exp(-\beta)), \end{aligned} \tag{54}$$

where $|\mathbb{T}|$, i.e. the cardinality of the space of transformations, is estimated by K^N, which is equivalent to the uniform input selection in Shannon capacity.

Generalization capacity is defined as the maximum of $\mathbb{E}_{\xi^{(1)},\xi^{(2)}}[\hat{\mathcal{J}}_\beta]$ over all values of β. We compute:

$$\begin{aligned} \mathcal{GC} &:= \max_\beta \mathbb{E}_{\xi^{(1)},\xi^{(2)}}[\hat{\mathcal{J}}_\beta] \\ &= \log K + (1-\hat{\delta})\log(1-\hat{\delta}) + \hat{\delta}\log\frac{\hat{\delta}}{K-1} . \end{aligned} \tag{55}$$

4.3.2 *Comparison with Shannon Capacity*

A communication channel is generally characterized by a *transition matrix* $p(y|x)$ which determines the conditional distribution of the output symbol (i.e. $y \in \{1, \dots, K\}$) given the input symbol (i.e. $x \in \{1, \dots, K\}$). A channel is called *weakly symmetric* if every row of the transition matrix $p(.|x)$ is a permutation of every other row, and all the column sums $\sum_x p(y|x)$ are equal [CT06].

For the weakly symmetric channel ([CT06], Thm. 7.2.1), Shannon's channel theory provides the capacity

$$\mathcal{SC} = \log K - H(p(\cdot|x)), \tag{56}$$

where $H(\cdot)$ is the entropy function and $p(\cdot|x)$ is a row of the transition matrix. The Shannon capacity for the symmetric K-ary code is thus

$$\begin{aligned} \mathcal{SC} &= \log K + (1-\delta)\log(1-\delta) + \sum_{k=1}^{K-1} \frac{\delta}{K-1} \log \frac{\delta}{K-1} \\ &= \log K + (1-\delta)\log(1-\delta) + \delta \log \frac{\delta}{K-1}\,, \end{aligned} \tag{57}$$

which is identical to Eq. 55. Therefore, the generalization capacity of K-ary codes (with error rate δ) yields a consistent result with Shannon capacity.

Given that the principle of generalization capacity is applicable to a broad class of learning problems, we reason that it is a more general theory, containing classical information theory as a special case.

4.4 GENERALIZATION CAPACITY VS. MINIMUM TRANSFER COSTS

Minimum Transfer Costs (MTC) and Generalization Capacity ($\mathcal{GC}$), both advocate the stability of the solution, i.e. the compatibility among different instances from the same source. However, generalization capacity supports a more fundamental objective which corresponds to the *reproducibility* of the solution sets (weight distributions). The way of defining the weight distributions and then computing the optimal resolution of the solution space provides a *generic* way to formulate and address the trade-off dilemma between stability and informativeness. The answer to this question gives the optimal *localization* of the weight distributions in the solution space.

These properties provide a broader applicability to the generalization capacity principle compared to Minimum Transfer Costs:

1. MTC is only applicable to the models described by a cost function, but $\mathcal{GC}$ can be used whenever a data processing mechanism generates a trajectory of the weight distributions over the solution space.

2. MTC can be employed only for selecting the optimal order of the models from the *same* family, e.g. it fails to compare Correlation Clustering and Pairwise Clustering. $\mathcal{GC}$ computes the rate of reliable information extractable by an algorithm as an absolute number. This enables to rank and validate totally different algorithms for solving the same data processing task.

Part III

GENERALIZATION CAPACITY ANALYSIS OF CLUSTERING METHODS

5

CALCULATION AND ANALYSIS OF GENERALIZATION CAPACITY

In this chapter, we use the problem of grouping data to investigate the generalization capacity of clustering algorithms. We describe how the principle can be used to address the fundamental learning questions, i.e. i) computing the optimal number of clusters, ii) ranking alternative clustering methods, and iii) comparing different similarity measures. We establish a concrete pipeline to compute and interpret generalization capacity for different clustering methods. Calculating the individual and joint weight sums and thereby generalization capacity, renders an exponential complexity for many standard clustering models as it requires summation of the weights over the solution space. For parametric clustering methods, e.g. K-means, given the parameters of the clusters, the weights take a factorial form, which simplifies the calculation of the weight sums.

However, in graph clustering models, the assignment of an object to a cluster affects the other assignments. Thereby, the weight distributions no longer take a factorial form. To overcome the computational challenges, we utilize efficient and general-purpose approximation schemes such as *mean-field approximation* to compute the most similar factorial weights. We then extend the framework to analyzing the arbitrary ad hoc algorithms that intrinsically do not yield a trajectory of weight distributions (e.g. dynamical systems such as Dominant Set clustering). We propose to define the weight distributions according to a Hamming metric in the solution space.

We exemplify the generalization capacity principle on the well-known Gaussian Mixture Models (GMMs). We employ the principle to analyze the behavior of maximum likelihood inference for GMMs in different settings:

1. in low dimensional setting when dimensionality $D = 2$, and
2. in the high dimensional limit with $D \to \infty$ and $\alpha := N/D$ is kept finite.

The high dimensional setting reveals several phase transitions depending on the number of objects per dimension, i.e. α. *Order* parameters are usually used in this context to analyze different phase transitions. Generalization capacity consistently confirms the evolution of such phase transitions.

5.1 CALCULATION OF GENERALIZATION CAPACITY FOR CLUSTERING MODELS

Let $\mathbf{X} \in \mathcal{X}$ represent the measurements for a set of N objects $\mathbf{O}$. The measurements can be either vectors in the feature space or the set of pairwise similarities between objects. For clustering, the patterns are object partitionings, i.e. $\mathcal{C} = \{1, \dots, K\}^N$. To simplify the notation, model parameters θ (e.g., centroids of K-means clustering) are not explicitly listed as arguments of the algorithm. The solution c encodes all such parameters.

As the measurements $\mathbf{X}$ are random variables, the empirical minimizer $c^{\perp}(\mathbf{X})$ is a random variable as well. Let $\mathbf{X}^{(q)}, q \in \{1, 2\}$, be two datasets with the same signal structure but different noise realizations. In most cases, their global minima differ, i.e. $c^{\perp}(\mathbf{X}^{(1)}) \neq c^{\perp}(\mathbf{X}^{(2)})$. Thereby, the empirical output of the algorithm suffers from the lack of stability.

We consider clustering methods as data processing mechanisms that evolve the weight distributions $w_\gamma(c, \mathbf{X})$ during execution. At the beginning the weights are uniformly distributed over the solution space $\mathcal{C}$. During execution they become more distinguishable and provide a more precise localization in the solution space, until, at the end, they concentrate only around the empirical output of the algorithm. Stopping the algorithm at an earlier step, which corresponds to a wider weight distribution, effectively determines a large set of solutions and thereby provides *stability* of the solutions with a risk of loosing *informativeness*. Optimal clustering now renders computing the optimal localization of the weights around the empirical output.

Generalization capacity provides a framework to analyze the trajectory of weight distributions. The optimal weight distribution, in an information-theoretic manner, is achieved by maximizing the function $\mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$, where $\hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ is computed by [Buh10]

$$\hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = \frac{1}{N} \log \left(\frac{|\mathbb{T}| \, \Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})}{Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}) \, Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(2)})} \right). \tag{58}$$

The maximum of $\mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ with respect to γ, called Generalization Capacity ($\mathcal{GC}$), gives the optimal bit rate that the clustering algorithm $\mathcal{A}$ can extract from the data.

Let us assume that the weights are given in a factorial way, i.e.

$$w_\gamma(c, \mathbf{X}) = \prod_{i=1}^{N} w_\gamma(c(i), \mathbf{X}, i), \tag{59}$$

where $w_\gamma(k, \mathbf{X}, i)$ indicates the weight of object i w.r.t cluster k, given the data $\mathbf{X}$.

We will explain later how such factorial weights are computed for different clustering methods.

Computing $\mathcal{GC}$ generally requires three steps:

1. Identify the hypothesis space of the model and calculate the cardinality of the space of transformations, i.e. $|\mathbb{T}|$.
2. Compute the weight sums $Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)})$ and $Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(2)})$ and the joint weight sum $\Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$.
3. Compute $\hat{\jmath}_\gamma^{\mathcal{A}}$ and maximize the expectation $\mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_\gamma^{\mathcal{A}}$ with respect to γ to obtain the generalization capacity.

We explain how each of the steps is performed for a clustering algorithm.

CALCULATION OF THE CARDINALITY $|\mathbb{T}|$. In clustering problems, the hypothesis class is spanned by all possible assignments of objects to clusters. Thereby, $\mathbb{T}$ is the set of permutations that generate new distinct solutions when applied to the object indices of $\mathbf{X}^{(1)}$. Although a solution contains the cluster assignments and cluster parameters like centroids, the parameters contribute a much smaller entropy to the solution than the assignments. By assuming $p_k = \frac{N_k}{N}$ and N_k being the size of cluster k, we then have

$$|\mathbb{T}| = \frac{N!}{(p_1 N)!(p_2 N)! \dots (p_K N)!} . \tag{60}$$

Thus,

$$\begin{aligned}
\frac{1}{N} \log |\mathbb{T}| &= \frac{1}{N} \log \frac{N!}{\prod_{k=1}^{K} (p_k N)!} \\
&\approx \frac{1}{N} \log \frac{N^N}{\prod_{k=1}^{K} (p_k N)^{p_k N}} \\
&= \frac{1}{N} \left(N \log N - \sum_{k=1}^{K} p_k N \log (p_k N) \right) \\
&= \frac{1}{N} \left(N \log N - N \log N - \sum_{k=1}^{K} p_k N \log p_k \right) \\
&= - \sum_{k=1}^{K} p_k \log p_k \\
&:= H(\mathcal{A}_K) .
\end{aligned} \tag{61}$$

CALCULATION OF THE WEIGHT SUMS. The factorial form of the weight distributions simplifies the summation of the weights over the solution space. The weight sums $Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(q)}),\ q = 1, 2$ are calculated by

$$\begin{aligned} Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(q)}) &= \sum_{c \in \mathcal{C}} w_\gamma(c, \mathbf{X}^{(q)}) \\ &= \sum_{c \in \mathcal{C}} \prod_{i=1}^{N} w_\gamma(c(i), \mathbf{X}^{(q)}, i) \\ &= \prod_{i=1}^{N} \sum_{k=1}^{K} w_\gamma(k, \mathbf{X}^{(q)}, i). \end{aligned} \tag{62}$$

Similarly we obtain the joint weight sum as

$$\begin{aligned} \Delta Z_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= \sum_{c \in \mathcal{C}} w_\gamma(c, \mathbf{X}^{(1)}) \cdot w_\gamma(c, \mathbf{X}^{(2)}) \\ &= \sum_{c \in \mathcal{C}} \prod_{i=1}^{N} w_\gamma(c(i), \mathbf{X}^{(1)}, i) \cdot w_\gamma(c(i), \mathbf{X}^{(2)}, i) \\ &= \prod_{i=1}^{N} \sum_{k=1}^{K} w_\gamma(k, \mathbf{X}^{(1)}, i) \cdot w_\gamma(k, \mathbf{X}^{(2)}, i). \end{aligned} \tag{63}$$

COMPUTING $\hat{\mathcal{I}}_\gamma^{\mathcal{A}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$. $\hat{\mathcal{I}}_\gamma$ is then obtained by

$$\begin{aligned} \hat{\mathcal{I}}_\gamma(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= H(\mathcal{A}_K) \\ &\quad + \frac{1}{N} \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_\gamma(k, \mathbf{X}^{(1)}, i) \cdot w_\gamma(k, \mathbf{X}^{(2)}, i) \\ &\quad - \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_\gamma(k, \mathbf{X}^{(1)}, i) - \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_\gamma(k, \mathbf{X}^{(2)}, i) \\ &= H(\mathcal{A}_K) \\ &\quad + \frac{1}{N} \sum_{i=1}^{N} \log \sum_{k=1}^{K} \left(\frac{w_\gamma(k, \mathbf{X}^{(1)}, i)}{\sum_{k'} w_\gamma(k', \mathbf{X}^{(1)}, i)} \times \frac{w_\gamma(k, \mathbf{X}^{(2)}, i)}{\sum_{k'} w_\gamma(k', \mathbf{X}^{(2)}, i)} \right). \end{aligned} \tag{64}$$

We note that $\frac{w_\gamma(k, \mathbf{X}, i)}{\sum_{k'} w_\gamma(k', \mathbf{X}, i)} := P_{ik}$ effectively indicates the probability of assigning object i to cluster k. Thereby, $\hat{\mathcal{I}}_\gamma(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ can be written as

$$\hat{\mathcal{I}}_\gamma(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = H(\mathcal{A}_K) + \frac{1}{N} \sum_{i=1}^{N} \log \sum_{k=1}^{K} \left(P_{ik}^{(1)} \times P_{ik}^{(2)} \right). \tag{65}$$

This formulation provides a straightforward interpretation of $\hat{\mathcal{I}}_\gamma$:

- The first term, i.e. $H(\mathcal{A}_K)$, is responsible for informativeness and is maximal for clusters with equal size.
- The second term, i.e. $\frac{1}{N}\sum_{i=1}^{N}\log\sum_{k=1}^{K}\left(P_{ik}^{(1)}\times P_{ik}^{(2)}\right)$, investigates the consistency between the weight distributions of the first and the second datasets, thereby is responsible for the stability of the solutions.

5.2 WEIGHT DISTRIBUTIONS OF CLUSTERING METHODS

Clustering methods are assumed as data processing mechanisms that, at each step, identify a weight distribution $w_\gamma(c, \mathbf{X})$ over the solution space.

For the methods characterized by a cost function, e.g. K-means clustering or Correlation Clustering, one can use the family of Boltzmann weights[1]

$$w_{\beta\propto\frac{1}{\gamma}}(c, \mathbf{X}) \quad := \quad \exp(-\beta R(c, \mathbf{X}))\,, \tag{66}$$

that are parameterized by the inverse computational temperature β.

Once a cost function is identified, in principle, any known optimization technique could be applied to find the optimal solutions. However, different optimization techniques might yield different trajectory of weight distributions over the solutions space. Thus, the respective generalization capacity can differ. Boltzmann weights are consistent with the principle of maximum entropy estimation. The principle leads to the provably most robust data assignment strategy [TTL84] which is the prominent advantage of heuristic algorithms such as simulated or deterministic annealing [Ros98]. According to [Jay57a, Jay57b], maximizing the entropy provides the least biased inference method that has the minimum commitment to the unknown part of data and thereby yields the robustness of the inference procedure. Moreover, this type of inference provides a scalable optimization procedure in terms of the *complexity-quality* trade-off [RGF90].

The inverse computational temperature β, which replaces γ, controls the width of the weight distribution. A large β yields the concentration of the weights around the empirical minimizer, while a small β renders a wide distribution, i.e. many solutions as good approximations of the minimizer $c^{\perp}(\mathbf{X})$ in terms of costs.

This choice of weights fulfills

1. non-negativity, i.e.

$$w_{\beta\propto\frac{1}{\gamma}}(c, \mathbf{X}) \geqslant 0. \tag{67}$$

1 Although the Boltzmann weights are a particular choice, all other weighting schemes can be explained by a monotonic rescaling of the costs.

2. the inverse order, i.e.

$$R(c', \mathbf{X}) \leqslant R(c'', \mathbf{X}) \quad \Longleftrightarrow \quad w_\beta(c', \mathbf{X}) \geqslant w_\beta(c'', \mathbf{X}) \,. \tag{68}$$

In the following, we investigate the computation of the weights for different clustering models: i) parametric models, ii) graph-based models, and iii) arbitrary ad hoc methods.

5.2.1 *Weight Distributions of Parametric Clustering Methods*

A category of clustering models represent each cluster by a set of cluster-dependent parameters. The parameters are sufficient to describe the clusters completely, e.g. the centroids in K-means clustering or the set of means, covariances and the mixing coefficients in a Gaussian mixture model. Such models, so-called *parametric clustering models*, usually operate on object-specific feature vectors, i.e. they assume the special case of a bijective mapping between objects and measurements where the i^{th} object is isomorphic to the vector $\mathbf{x}_i \in \mathbb{R}^D$.

In a parametric clustering model, given the parameters of the clusters, the objects are then *independently* assigned to the clusters since the parameters sufficiently represent the clusters. Thereby, the cost function of a parametric model with K clusters can be written as

$$\begin{aligned} R(c, \mathbf{X}) &= \sum_{i=1}^{N} h_{i,c(i)} \\ \text{with} \qquad & \forall i,\, c(i) \in \{1, .., K\} \,. \end{aligned} \tag{69}$$

The potential h_{ik} indicates the costs of assigning object i to cluster k. In the well-known case of K-means clustering, the potential $h_{i,c(i)}$ corresponds to the squared distance between object i and its centroid, i.e.

$$h_{i,c(i)} = \|\mathbf{x}_i - \boldsymbol{\mu}_{c(i)}\|^2, \tag{70}$$

where $\boldsymbol{\mu}_{c(i)}$ is the centroid object i is assigned to.

In a parametric model, the weights naturally take a factorial form given the cluster parameters, since,

$$\begin{aligned} w_\beta(c, X) &:= \exp(-\beta R(c, \mathbf{X})) \\ &= \exp(-\beta \sum_{i=1}^{N} h_{i,c(i)}) \\ &= \prod_{i=1}^{N} \exp(-\beta h_{i,c(i)}) \\ &= \prod_{i=1}^{N} w_\beta(c(i), \mathbf{X}, i)\,. \end{aligned} \tag{71}$$

5.2.2 *Weight Distributions of Non-Factorial Models*

In general, the (object, measurement) relation might be more complex than an object-specific feature vector. In many applications such as molecular biology, social networks and linguistics only a set of pairwise similarity/dissimilarity measurements is available. This type of data is characterized mathematically by a *graph* where the pairwise similarities constitute the edge weights. Clustering a graph is considered a more complicated problem as the inherent structure is hidden in N^2 pairwise relations [HB97].

However, the pairwise similarities might violate the requirements of a valid metric, i.e. they might be nonsymmetric or negative and the triangle inequality does not necessarily hold. Therefore, in general, a loss-free embedding into a vector space is not possible, so that graph clustering cannot directly be transformed into vectorial clustering by means of classical embedding techniques. In such models, the clusters are not represented by parameters. Thereby, the assignment of an object to a cluster influences the assignments of other objects. As a result, the weights no longer assume a product form. This implies that the corresponding Gibbs distribution $\mathbf{P}^{\mathcal{A}}(c), c \in \mathcal{C}(\mathbf{X})$ cannot factorize as product of individual probabilities $P_1^{\mathcal{A}}(c(1)) \times ... \times P_N^{\mathcal{A}}(c(N))$. However, cost contributions converge to averages in the limit of large datasets. For consistent measurements, the influence of dependencies on individual data assignments becomes small and vanishing.

5.2.2.1 *Mean-Field Approximation for Graph Clustering Models*

For non-factorial models, we compute the *mean-field* potentials h_{ik} by approximating the original Gibbs distribution $P_\beta^{\mathcal{A}}(c) = w_\beta(c, \mathbf{X})/Z_\beta$ by a factorial distribution with the mean-fields as adjustable parameters. Given the mean-fields, the assignments $c(.)$ of objects to clusters are independent, i.e. $c(i)$ is not influencing $c(j), j \neq i$.

The family of *factorial* distributions is defined as

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathcal{P} : \mathbf{Q}(c) = \prod_{i=1}^{N} q_{i,c(i)}, \quad q_{i,c(i)} \in [0,1] \right\}, \tag{72}$$

where $\mathcal{P}$ is the space of all probability distributions defined on the solution space $\mathcal{C}(\mathbf{X})$.

This choice of $\mathcal{Q}$ provides a couple of interesting advantages: i) the computational runtime of computing the probability distribution $\mathbf{Q}(c)$ increases linearly with N rather than being exponential, ii) the approximation scheme is applicable to a broad class of cost functions, iii) efficient alternation algorithms exist for working on this family of distributions, and iv) the approach offers the possibility of tracking the phase transitions (e.g. cluster splits) which yields computing a tree topology of structures [RMG96, Ros98].

The closest factorial distribution (in an information-theoretic sense) can be determined by minimizing the Kullback-Leibler divergence (see [HB97])[2]

$$\begin{aligned}
D_{KL}\left(\mathbf{Q}\|\mathbf{P}^{\mathcal{A}}\right) &= \sum_{c\in\mathcal{C}} \mathbf{Q} \log \frac{\mathbf{Q}}{\mathbf{P}^{\mathcal{A}}} \\
&= \sum_{c\in\mathcal{C}} \mathbf{Q} \log \frac{\mathbf{Q}}{\exp(-\beta(R^{\mathcal{A}} - F^{\mathcal{A}}))} \\
&= \sum_{c\in\mathcal{C}} \mathbf{Q} \left[\sum_{i=1}^{N} \log q_{i,c(i)} + \beta(R^{\mathcal{A}} - F^{\mathcal{A}}) \right] \\
&= \sum_{i=1}^{N} \sum_{c\in\mathcal{C}} \mathbf{Q} \log q_{i,c(i)} + \beta(\mathbb{E}_{\mathbf{Q}}\{R^{\mathcal{A}}\} - F^{\mathcal{A}}) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} q_{ik} \log q_{ik} + \beta \mathbb{E}_{\mathbf{Q}}\{R^{\mathcal{A}}\} - \beta F^{\mathcal{A}}.
\end{aligned} \tag{73}$$

The free energy $F^{\mathcal{A}} := -\frac{1}{\beta} \log Z_\beta(\mathbf{X})$ does not depend on q_{ik}. To find the optimal factorial distribution, we minimize $D_{KL}\left(\mathbf{Q}\|\mathbf{P}^{\mathcal{A}}\right)$ with respect to q_{ik} observing the normalization constraint $\sum_{k=1}^{K} q_{ik} = 1, \forall i$:

$$\begin{aligned}
0 &= \frac{\partial}{\partial q_{ik}} \left[D_{KL}\left(\mathbf{Q}\|\mathbf{P}^{\mathcal{A}}\right) + \sum_{j=1}^{N} \lambda_j \Big(\sum_{k=1}^{K} q_{jk} - 1 \Big) \right] \\
&= \sum_{c\in\mathcal{C}} \prod_{j\leqslant N : j\neq i} q_{j,c(j)} \mathbb{I}_{\{c(i)=k\}} R^{\mathcal{A}} + \frac{1}{\beta}(\log q_{ik} + 1) + \lambda_i.
\end{aligned} \tag{74}$$

2 The probabilities can also be estimated numerically by Markov chain Monte Carlo (MCMC) methods such as Gibbs sampling. Essentially, the mean-fields h_{ik} are the $\mathbf{Q}$-averaged variants of Gibbs transition costs (Gibbs weights) computed during performing Gibbs sampling. Sub-sampling can be useful when working with large datasets.

For the extremum of the bound, the necessary condition determines the mean-field assignments

$$q_{ik} = \frac{\exp(-\beta h_{ik})}{\sum_{k'} \exp(\beta h_{ik'})}, \quad \text{with } h_{ik} = \mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}\}. \tag{75}$$

$\mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}\}$ is the expectation over all configurations subject to the constraint of assigning object i to cluster k. Thereby, to calculate the mean-fields, $R^{\mathcal{A}}$ is decomposed into contributions which depend on object i and on the costs of all other objects. Each q_{ik} is influenced uniquely by the terms which depend on object i. The rest is constituted by an constant, irrelevant to optimization.

One can interpret mean-field approximation as replacing the original cost function by an additive mean-field cost function R^{MF}, defined as

$$\begin{aligned} R^{MF}(c, \mathbf{X}) &= \sum_{i=1}^{N}\sum_{k=1}^{N} M_{ik} h_{ik} \\ &= \sum_{i=1}^{N} h_{i,c(i)}. \end{aligned} \tag{76}$$

Essentially, mean-field approximation yields a lower bound Z^{MF} for the weight sums (Jensen's inequality)

$$Z_\beta \geqslant Z_\beta^{MF} = \sum_{i=1}^{N}\sum_{k=1}^{K} \exp(-\beta h_{ik}). \tag{77}$$

Since the assignment probabilities q_{ik} depend on the mean-fields h_{ik}, this leads to a system of $N \times K$ coupled transcendental equations. Thus, through an annealing scheme, an iterative **EM**-type algorithm at each temperature approximates the mean-fields and the probabilities by mutual conditioning. The t^{th} iteration of the algorithm consists of two main steps. First, $q_{ik}^{(t)}$ is estimated as a function of ${h_{ik}}^{(t-1)}$. Second, ${h_{ik}}^{(t)}$ is calculated for given $q_{ik}^{(t)}$. The corresponding mean-fields h_{ik} are then used to compute the weights, i.e., $w_\beta(k, \mathbf{X}, i) = \exp(-\beta h_{ik})$. Algorithm 1 formulates the procedure.

The concentration parameter β increases either exponentially (i.e. $\beta \leftarrow \eta\beta$) or linearly (i.e. $\beta \leftarrow \beta + \eta$). Linear annealing might yield a superior optimization since the search process is performed by a slower and thereby more accurate temperature reduction [HB97].

5.2.2.2 *Computationally Efficient Calculation of Mean-Field Approximation*

Performing the potentials h_{ik} at every **M**-step of the mean-field annealing algorithm 1 requires computing the expectation $\mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}(c, \mathbf{X})\}$.

Algorithm 1 Calculate $\hat{\mathcal{J}}_{\beta}^{\mathcal{A}_K}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$

for $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ **do**
  Initialize $h_{ik}^{(0)}$ and $q_{ik}^{(0)}$ randomly.
  Initialize the concentration parameter $\beta \leftarrow \beta_0$.
  while $\beta \leqslant \beta_{final}$ **do**
    $t \leftarrow 0$.
    repeat
      E-step: estimate $q_{ik}^{(t+1)}$ as a function of $h_{ik}^{(t)}$.
      M-step: estimate $h_{ik}^{(t+1)}$ as a function of $q_{ik}^{(t+1)}$.
      $t \leftarrow t + 1$.
    until $h_{ik}^{(t)}$ and $q_{ik}^{(t)}$ converge
    Update β.
    $q_{ik}^{(0)} \leftarrow q_{ik}^{(t)}$.
    $h_{ik}^{(0)} \leftarrow h_{ik}^{(t)}$.
  end while
end for
Compute $H(\mathcal{A}_K)$ (for β_{final}).
Compute $\hat{\mathcal{J}}_{\beta}^{\mathcal{A}_K}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ at different β.

The expectation can be efficiently evaluated for quadratic cost functions, e.g. for Correlation Clustering, by directly applying the expectation to the cluster assignments.

$$\begin{aligned} h_{ik} &= \mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}(c, \mathbf{X})\} \\ &= R^{\mathcal{A}}(\mathbb{E}_{\mathbf{Q}_{i\to k}}\{c\}, \mathbf{X}). \end{aligned} \tag{78}$$

However, a main technical difficulty occurs when computing the expectation of the cost functions which normalize the clusters by a cluster-dependent factor. Normalized Cut [SM00] is an example of such models where each cluster is normalized by its degree. Thereby, the equality $\mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}(c, \mathbf{X})\} = R^{\mathcal{A}}(\mathbb{E}_{\mathbf{Q}_{i\to k}}\{c\}, \mathbf{X})$ does not hold anymore. In principle, one can use the polynomial normal forms of such cost functions to eliminate the denominator. However, this transformation yields an exponential order in the number of conjunctions. Therefore, it has been proposed to approximate the expectation by replacing the cluster assignment variables in the cost function by their probabilistic counterparts [HB97, GKR01], i.e. by independently calculating the expectation of the numerator and the denominator. Thus,

$$h_{ik} \approx R^{\mathcal{A}}(\mathbb{E}_{\mathbf{Q}_{i\to k}}\{c\}, \mathbf{X}). \tag{79}$$

This approximation is exact in the zero-temperature limit, i.e. when $\beta \to \infty$, for any N as well as in thermodynamic limit, i.e. $N \to \infty$, for any arbitrary β [GKR01]. Higher order improvements of the

approximation can be obtained e.g. by a Taylor expansion around $\mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}(c,\mathbf{X})\}$. Empirically, one can investigate the quality of approximation via measuring average values of the assignments by Markov Chain Monte Carlo (MCMC) simulations. Based on extensive experimental evidences [HB97, PHB00], the first-order approximation competes with more refined techniques such as the TAP method [TAP77]. This naive approximation is even superior at the low temperature range [HB97]. Moreover, as mentioned earlier, for a large enough N (e.g. $N \geqslant 250$ in our experiments) the approximation in Eq. 79 becomes almost exact.

A straightforward implementation of Algorithm 1 would basically compute the cost function at every **M**-step. However this procedure is very inefficient for normalized cost functions as it would require $O(N^2)$ computational complexity. Therefore, for a complete sweep the complexity would be $O(N^3)$. We utilize *book keeping quantities* [GKR01] in order to improve the efficiency of computing the potentials. The trick is based on an important observation that the potentials h_{ik} only have to be computed up to an additive shift, which may depend on the object i, but not on cluster index k. The additive terms cancel in computing the assignment probabilities q_{ik}. Thereby, they can be neglected for reducing the computations. After changing/updating the assignment of a single object, the update of all book-keeping quantities requires $O(N)$ operations. The computational complexity then would be $O(N^2)$ for a complete sweep. We will elaborate more on this technique when dealing with concrete clustering models in the next chapter.

5.2.2.3 *Quality of Mean-Field Approximation*

At each inverse temperature β, the mean-field algorithm 1 converges to a local minimum of the free energy $F_{\mathbf{Q}}$ defined over Ω [GKR01]. The quality of mean-field approximation can be evaluated by comparing the results with MCMC simulations. In-depth studies in [HPB98, PHB00] confirm the consistency between mean-field annealing and Gibbs sampling, such that in all different case studies the differences are very small. The consistency is observed in different experimental settings, e.g. various annealing schedules. It is known that for logarithmic annealing schedules Gibbs sampling converges to the global minimum in probability [Haj88]. Because of these global optimization properties, one can conclude that mean-field annealing yields near optimal solutions in most cases. On the other hand, the loss in clustering quality for mean-field algorithm by a faster annealing schedule is substantially smaller than for Gibbs sampling [PHB00]. Thus, the mean-field technique provides a possibility for an adequate trade-off between *computational complexity* and *clustering quality*.

5.2.3 *Weight Distributions of General Algorithms Not Guided by Gradient Flow on Costs*

The models characterized by a cost function render a *partial rank* over the solution space, i.e. each pair of the solutions are compared based on the respective costs. This property makes the use of Boltzmann weights feasible to track the trajectory of algorithms.

However, there exist clustering methods that do not explicitly minimize a cost function and thereby do not identify an appropriate trajectory of weight distributions during execution. DBSCAN [EKSX96] and Dominant Set clustering [PP07] are examples of such models. For such cases, we employ a Hamming metric in the solution space to introduce the weight distributions. This choice is consistent with the maximum entropy principle in the data space.

Given two datasets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, we let the clustering algorithm under investigation $\mathcal{A}$ compute the two corresponding output hypotheses $c^{\perp(q)}, q = 1, 2$. The weights are then computed by

$$w_\beta(c, \mathbf{X}^{(q)}) = \exp\left(-\beta d(c, c^{\perp(q)})\right), \tag{80}$$

where the Hamming distance $d(c, c^{\perp(q)})$ is defined as

$$d(c, c^{\perp(q)}) = \sum_{i=1}^{N} \mathbb{I}_{\{c(i) \neq c^{\perp(q)}(i)\}}. \tag{81}$$

This choice of weight distributions is similar to the case of K-ary codes. However, for K-ary codes, the entropy term is estimated by $\log K$ due to uniform input distribution, whereas here it is computed by the entropy of the clusters of the empirical output. Similar to the case of K-ary codes, the weights sums $Z_\beta(\mathbf{X}^{(q)})$, $q = 1, 2$ amount to

$$\begin{aligned} Z_\beta(\mathbf{X}^{(q)}) &= \sum_{c \in \mathcal{C}} \exp\left(-\beta \sum_{i=1}^{N} \mathbb{I}_{\{c(i) \neq c^{\perp(q)}(i)\}}\right) \\ &= [(K-1)\exp(-\beta) + 1]^N . \end{aligned} \tag{82}$$

The joint weight sum $\Delta Z_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ is computed as

$$\begin{aligned} \Delta Z_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= \sum_{c \in \mathcal{C}} \exp\left(-\beta \left(\sum_{i=1}^{N} \mathbb{I}_{\{c(i) \neq c^{\perp(1)}(i)\}} + \sum_{i=1}^{N} \mathbb{I}_{\{c(i) \neq c^{\perp(2)}(i)\}}\right)\right) \\ &= [1 + (K-1)\exp(-2\beta)]^{N(1-\hat{\delta})} \\ &\quad \times [2\exp(-\beta) + (K-2)\exp(-2\beta)]^{N\hat{\delta}} . \end{aligned} \tag{83}$$

Here, $\hat{\delta}$ denotes the fraction of different cluster assignments

$$\begin{aligned}\hat{\delta} &= d(c^{\perp(1)}, c^{\perp(2)}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\{c^{\perp(1)}(i) \neq c^{\perp(2)}(i)\}}\,. \end{aligned} \tag{84}$$

Please note that computing $\hat{\delta}$ requires comparing two sets of labels that are not necessarily in correct correspondence. Two clustering solutions $c^{\perp(1)}$ and $c^{\perp(2)}$ might be structurally equivalent but with different labelings. For example, for $K = 2$, the cluster label '1' in $c^{\perp(1)}$ might correspond to the label '2' in $c^{\perp(2)}$, and vice versa. Therefore, we minimize $\hat{\delta}$ over different labelings of the clusters in $c^{\perp(2)}$, i.e.

$$\hat{\delta} = \min_{\pi \in \Pi_K} d(c^{\perp(1)}, \pi \circ c^{\perp(2)})\,, \tag{85}$$

where Π_K denotes the set of all $K!$ labelings of K clusters.

$\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ is then obtained by

$$\begin{aligned}\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= H(\mathcal{A}_K) - \beta\hat{\delta} - 2\log\left(1 + (K-1)\exp(-\beta)\right) \\ &\quad + (1-\hat{\delta})\log\left(1 + (K-1)\exp(-2\beta)\right) \\ &\quad + \hat{\delta}\log\left(2 + (K-2)\exp(-\beta)\right)\,. \end{aligned} \tag{86}$$

We then compute the generalization capacity by

$$\mathcal{GC} = H(\mathcal{A}_K) + \left(1-\hat{\delta}\right)\log\left(1-\hat{\delta}\right) + \hat{\delta}\log\frac{\hat{\delta}}{K-1}\,. \tag{87}$$

In Eq. 87, the first term, i.e. $H(\mathcal{A}_K)$, is responsible for informativeness and is maximal for balanced clusters. The second term, i.e. $\left(1-\hat{\delta}\right)\log\left(1-\hat{\delta}\right) + \hat{\delta}\log\frac{\hat{\delta}}{K-1}$, accounts for stability of the solutions. $\hat{\delta}$ effectively is the instability measure used in stability analysis of clustering models (e.g. [LBRB04]). The formula above gives the correction term required to include *model complexity*.

5.3 $\mathcal{GC}$-BASED MODEL SELECTION AND VALIDATION

Generalization capacity computes the optimal rate of information, as an *absolute number*, that an algorithm extracts from the data at hand. This suggests an intrinsic inference principle for model validation: *A superior model yields a higher generalization capacity*.

We propose an algorithmic procedure for model order selection and model validation based on the generalization capacity principle.

5.3.1 *Model Order Selection and Model Validation*

Given algorithm $\mathcal{A}$ with a fixed model order K, $\mathcal{GC}_K^{\mathcal{A}}$ computes the optimal information rate for the order K.

$$\mathcal{GC}_K^{\mathcal{A}} := \max_{\gamma} \mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_{\gamma}^{\mathcal{A}_K}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) \,. \tag{88}$$

To identify the optimal model order, e.g. the optimal number of clusters, one can compute $\mathcal{GC}$ for different model orders $2 \leqslant K \leqslant K_{max}$ and choose the order which maximizes the generalization capacity. $\mathcal{GC}$ is usually 0 for $K = 1$ as this case carries no information. Algorithm 2 describes the order selection for a model optimized by an annealing procedure.

Algorithm 2 Calculate_Model_Order

1: **for** $1 \leqslant K \leqslant K_{max}$ **do**
2: Perform either annealed Gibbs sampling or annealed *mean-field approximation* to compute the potentials h_{ik} at different inverse temperatures β.
3: At each inverse temperature β, calculate $\hat{\jmath}_{\beta}^{\mathcal{A}_K}$.
4: Compute $\mathcal{GC}_K^{\mathcal{A}} := \max_{\beta} \mathbb{E}_{\mathbf{X}^{(1)},\mathbf{X}^{(2)}} \hat{\jmath}_{\beta}^{\mathcal{A}_K}$.
5: **end for**
6: $\mathcal{GC}^{\mathcal{A}} := \max\limits_{1 \leqslant K \leqslant K_{max}} \mathcal{GC}_K^{\mathcal{A}}$.
7: **return** $K^{opt} := \arg \max\limits_{1 \leqslant K \leqslant K_{max}} \mathcal{GC}_K^{\mathcal{A}}$.

While annealing, at each computational temperature β^{-1}, the potentials h_{ik} and the other parameters are updated. Thereby, in the case of $K > K^{opt}$, the optimal parameters, e.g. the optimal number of clusters, are expected to be obtained at the optimal temperature.

Another choice to be made in modeling is to select a suitable model $\mathcal{A}$ for the learning problem at hand. Let us assume that a collection of algorithms $\{\mathcal{A}^1(\mathbf{X}), \ldots, \mathcal{A}^l(\mathbf{X})\}$ are considered as candidates. Generalization capacity depends on the algorithm through the weights that the algorithm identifies over the solution space during execution. Robust and informative algorithms yield a higher generalization capacity than simpler or more brittle models. Thereby, the principle allows us to rank the collection of algorithms according to their information content. The algorithm or model to be selected is

$$\mathcal{A}^{\star}(\mathbf{X}) = \arg \max_{1 \leqslant j \leqslant l} \mathcal{GC}^{\mathcal{A}^j}. \tag{89}$$

The selection rule (89) prefers the algorithm or model which is 'rich' enough to exhibit high information content (e.g. many clusters in clustering) and, at the same time, robustly resists to noise in the data. Algorithm 3 describes the model selection procedure.

Algorithm 3 Model_Comparison

1: **for** $1 \leqslant j \leqslant l$ **do**
2: Compute the generalization capacity $\mathcal{GC}^{\mathcal{A}^j}$.
3: **end for**
4: Rank $\{\mathcal{A}^j\}$ according to the respective capacities $\{\mathcal{GC}^{\mathcal{A}^j}\}$.
5: **return** $\mathcal{A}^\star(\mathbf{X}) = \arg\max_{1 \leqslant j \leqslant l} \mathcal{GC}^{\mathcal{A}^j}$.

5.3.2 *Model Validation for Clustering*

Clustering constitutes a fundamental task in exploratory data analysis. Typically, the analyst has the option to select a clustering algorithm from a plethora of alternatives, while often puzzling over the central question: *Which algorithm is most informative for a given data source*? Generalization capacity measures quantitatively the amount of *reliable clustering information* that a clustering algorithm can establish in uncertain settings. Such an analysis not only renders a principled comparison of algorithms possible, but also guides the design of algorithms that provide the maximal information.

As mentioned in Chapter 1, a learning problem faces three fundamental choice challenges:

(i) Which algorithm should be chosen? K-means or Correlation Clustering?

(ii) What kind of preprocessing is required? Euclidian distance measurements or correlation coefficients?

(iii) What is the optimal model order, i.e. the optimal number of clusters?

Algorithm 4 describes the application of $\mathcal{GC}$ to address the questions in the context of data clustering.

Algorithm 4 Model Order Selection and Model Validation for Clustering

1: **for** all algorithms $\mathcal{A}$ and similarly measures $\mathbf{X}$ **do**
2: **for** different number of clusters K **do**
3: Compute $\mathcal{GC}^{\mathcal{A}}_K(\mathbf{X})$.
4: **end for**
5: Compute the maximal capacity achievable from $\mathcal{A}(\mathbf{X})$: $\mathcal{GC}^{\mathcal{A}}(\mathbf{X}) = \max_K \mathcal{GC}^{\mathcal{A}}_K(\mathbf{X})$.
6: Select the optimal order of $\mathcal{A}(\mathbf{X})$.
7: **end for**
8: Evaluate the generalization capacity of all algorithms and similarity measures, and select the setting with the maximum generalization capacity.

5.4 GENERALIZATION CAPACITY FOR MIXTURES OF GAUSSIANS

In this section, we exemplify the calculation of generalization capacity for Gaussian Mixture Models (GMMs). This model plays an important role in clustering analysis of real-world data. We describe the calculation of the generalization capacity and compare the results with other model selection principles such as MTC and BIC.

5.4.1 *Experimental Study of Generalization Capacity for Mixture of Gaussians*

A Gaussian mixture model with K components is defined as

$$\begin{aligned} p(\mathbf{x}) &= \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \\ \text{s.t.} \qquad & \pi_k \geqslant 0 \, , \; \sum_k \pi_k = 1. \end{aligned} \tag{90}$$

For didactical reasons, we do not optimize the covariance matrix $\boldsymbol{\Sigma}$ and simply fix it to $\boldsymbol{\Sigma} = 0.5 \cdot \mathbf{I}$. Then, maximizing the GMM likelihood essentially reduces to centroid-based clustering. Therefore, $h_{ik} := \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$ indicates the costs of assigning object i to cluster k. The weights are then computed as

$$w_{\gamma \propto \frac{1}{\beta}}(k, \mathbf{X}, i) = \exp(-\beta h_{ik}). \tag{91}$$

By substituting the weights to Eq. 64, $\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ amounts to

$$\begin{aligned} \hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= H(\mathcal{A}_K) \\ &+ \frac{1}{N} \sum_{i=1}^{N} \log \sum_{k=1}^{K} \left(\frac{\exp(-\beta h_{ik}^{(1)})}{\sum_{k'} \exp(-\beta h_{ik'}^{(1)})} \times \frac{\exp(-\beta h_{ik}^{(2)})}{\sum_{k'} \exp(-\beta h_{ik'}^{(2)})} \right). \end{aligned} \tag{92}$$

Please note that $\frac{\exp(-\beta h_{ik})}{\sum_{k'} \exp(-\beta h_{ik'})}$ indicates the Gibbs probability of assigning object i to cluster k.

EXPERIMENT SETTINGS. For experimental evaluation, we define $K = 5$ Gaussians with parameters $\pi_k = 1/K$, $\boldsymbol{\mu} \in \{(1,0), (0,1.5), (-2,0), (0,-3), (4.25,-4)\}$, and with covariance $\boldsymbol{\Sigma} = 0.5 \cdot \mathbf{I}$. Let $\mathbf{X}^{(q)}$, $q = 1,2$ be two datasets of identical size $N = 10{,}000$ drawn from these Gaussians. We optimize the assignment variables and the centroid parameters of our GMM model via annealed Gibbs sampling [GG84]. The computational temperature in Gibbs sampling is equivalent to the assumed width of the distributions. Thereby, we provide twice

as many clusters to the model in order to enable overfitting. Starting from a high temperature, we successively cool down while optimizing the model parameters. In Figure 13a, we illustrate the positions of the centroids with respect to the center of mass. At high temperature, all centroids coincide, indicating that the optimizer favors one cluster. As the temperature is decreased, the centroids separate into increasingly many clusters until, finally, the sampler uses all 10 clusters to fit the data.

Figure 13b shows the numerical analysis of the generalization capacity. When the stopping temperature of the Gibbs sampler coincides with the optimal temperature β^{-1}, we expect the best trade-off between robustness and informativeness. And indeed, as illustrated in Figure 13a, the correct model-order $K = 5$ is found at this temperature. At lower stopping temperatures, the clusters split into many instable clusters which increases the decoding error, while at higher temperatures informativeness of the clustering solutions decreases.

5.4.2 *Comparison with Other Principles*

We compare generalization capacity with two other model order selection principles: i) generalization ability, and ii) BIC score.

RELATION TO GENERALIZATION ABILITY. A properly regularized clustering model explains not only the dataset at hand, but also new datasets from the same source. The inferred model parameters and assignment probabilities from the first dataset $\mathbf{X}^{(1)}$ can be used to compute the costs for the second dataset $\mathbf{X}^{(2)}$. The appropriate clustering model yields low costs on $\mathbf{X}^{(2)}$, while very informative but unstable structures and also very stable but little informative structures have high costs due to overfitting or underfitting, respectively.

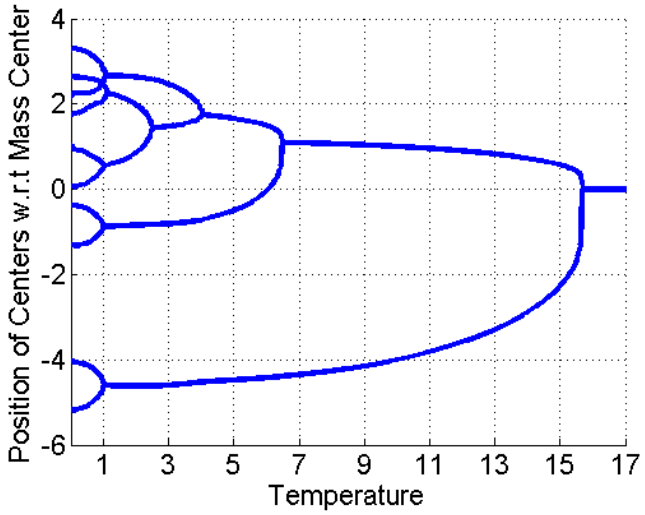
We measure this generalization ability by computing the *transfer costs* $R(c^{(1)}, \mathbf{X}^{(2)})$: At each stopping temperature of the Gibbs sampler, the current parameters $\boldsymbol{\mu}^{(1)}$ and assignment probabilities $\mathbf{P}^{(1)}$ inferred from $\mathbf{X}^{(1)}$ are transferred to $\mathbf{X}^{(2)}$. The assignment probabilities $\mathbf{P}^{(1)}$ assume the form of a Gibbs distribution

$$p(\boldsymbol{\mu}_k^{(1)}|\mathbf{x}_i^{(1)}) \;=\; Z_\beta^{-1} \exp\left(-\beta \|\mathbf{x}_i^{(1)} - \boldsymbol{\mu}_k^{(1)}\|^2\right), \tag{93}$$

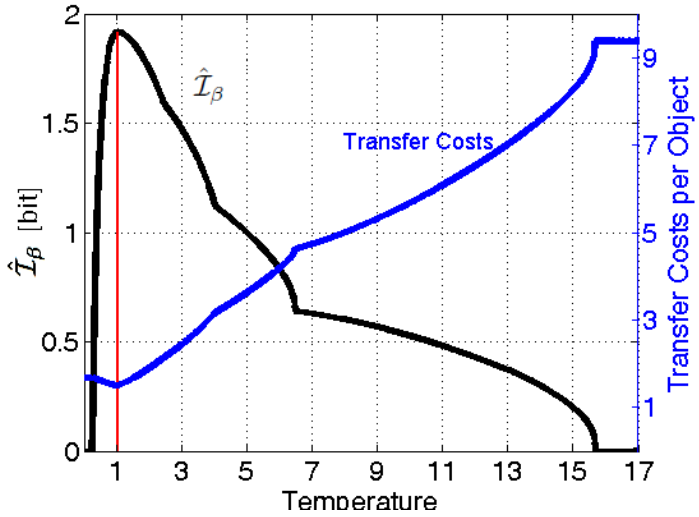
with Z_β as the normalization constant. The expected transfer costs with respect to these probabilities are then

$$\begin{aligned}\left\langle R(c^{(1)}, \mathbf{X}^{(2)})\right\rangle &= \sum_{i=1}^{N}\sum_{k=1}^{K} p(\mathbf{x}_i^{(1)}, \boldsymbol{\mu}_k^{(1)}) \|\mathbf{x}_i^{(2)} - \boldsymbol{\mu}_k^{(1)}\|^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} p(\boldsymbol{\mu}_k^{(1)}|\mathbf{x}_i^{(1)}) \|\mathbf{x}_i^{(2)} - \boldsymbol{\mu}_k^{(1)}\|^2. \end{aligned} \tag{94}$$

Figure 13b illustrates the transfer costs as a function of β and compares it with the generalization capacity. The optimal transfer costs



(a) Clustering hierarchy



(b) $\hat{\jmath}_\beta$ and transfer costs

Figure 13: Annealed Gibbs sampling for GMM: Influence of the stopping temperature for annealed optimization on $\hat{\jmath}_\beta$, on the transfer costs and on the positions of the cluster centroids. The lowest transfer cost is achieved at the temperature with highest $\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$. This is the lowest temperature at which the correct number of clusters $\hat{K} = 5$ is found. The hierarchy in Figure 13a is obtained by projecting the two-dimensional centroids at each stopping temperature to the optimal one-dimensional subspace using multidimensional scaling.

are obtained at the stopping temperature that corresponds to the generalization capacity.

RELATION TO BIC. Arguably the most popular criterion for model order selection is BIC as proposed by [Sch78]. It is, like generalization

capacity, an asymptotic principle, i.e. for sufficiently many observations, the fitted model preferred by BIC ideally corresponds to the candidate which is a posteriori most probable. However, the application of BIC is limited to models where one can determine the number of free parameters as here with GMM. Figure 14 confirms the consistency of generalization capacity with BIC in finding the correct model order in our experiment.

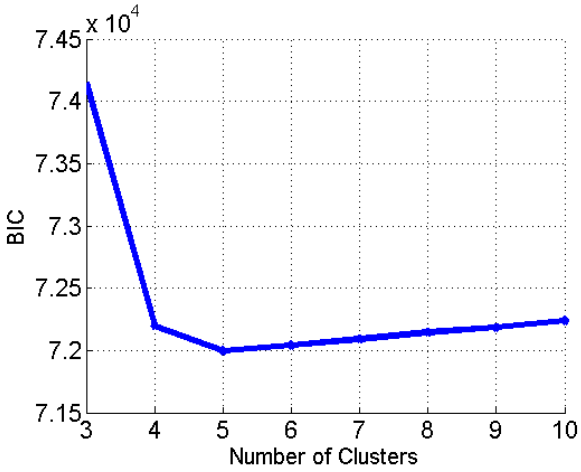


Figure 14: The BIC score for different number of clusters: BIC verifies correctness of $\hat{K} = 5$ computed by generalization capacity.

5.5 PHASE TRANSITION IN INFERENCE

This section discusses phase transitions and learnability limits. We review theoretical results and show how the generalization capacity principle can be employed to verify them.

5.5.1 *Phase Transition of Learnability*

Let the two centroids $\boldsymbol{\mu}_k$, $k = 1, 2$ of a GMM be orthogonal to each other and let them have equal magnitudes: $|\boldsymbol{\mu}_1| = |\boldsymbol{\mu}_2|$. The normalized separation u is defined as

$$u := |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2| / \sqrt{2\sigma_0}\,, \tag{95}$$

where σ_0 indicates the variance of the underlying Gaussian probability distributions with $\boldsymbol{\Sigma} = \sigma_0 \cdot \mathbf{I}$. We consider the asymptotic limit where the dimensionality $D \to \infty$ while $\alpha := N/D$, σ_0 and u are kept finite as described in [BSS93]. In this setting, the complexity of the problem, measured by the Bayes error, is proportional to $\sqrt{1/D}$. Therefore, we decrease the distance between the centroids by a factor of $\sqrt{1/D}$ when going to higher dimensions in order to keep the

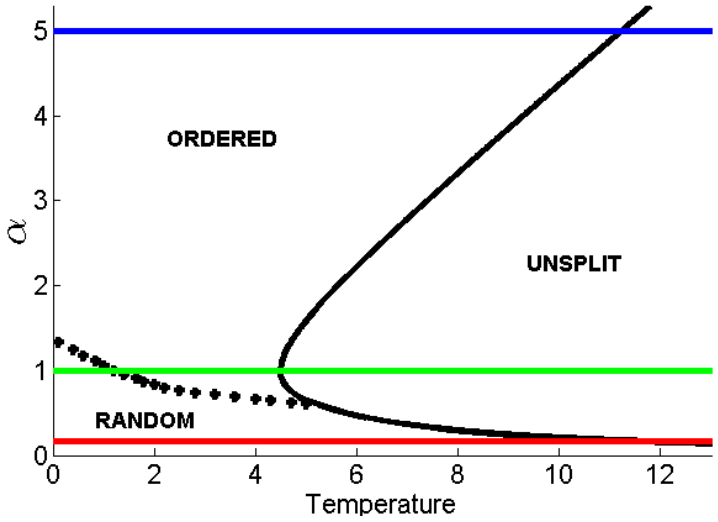


Figure 15: The phase diagram for maximum likelihood estimation at thermodynamic limits. Dependent on the number of objects per dimension (α) and the temperature different phases (Unsplit, Ordered and Random) are observed.

problem complexity constant. Similar to the two dimensional study, we use annealed Gibbs sampling to estimate the centroids μ_1, μ_2 at different temperatures.

The theory of this problem is studied in [BS94] and [WB09]. The study shows the presence of different phases depending on the values of stopping temperature and α. We introduce the same parameters as in [BS94]:

1. The separation vector $\Delta\hat{\mu} = (\hat{\mu}_1 - \hat{\mu}_2)/2$.
2. The order parameter s which computes the separation between the two estimated centers, i.e.
$$s = \sigma_0 |\Delta\hat{\mu}|^2 . \tag{96}$$
3. The order parameter r that computes the projection of the distance vector between the estimated centroids onto the distance vector between the true centroids, i.e.
$$r = \Delta\hat{\mu} \cdot \Delta\mu / u . \tag{97}$$

Computing these order parameters guides to construct the phase diagram.

Thereby, we sample $N = 500$ data items from two Gaussian sources with orthogonal centroids μ_1, μ_2 and equal prior probabilities $\pi_1 = \pi_2 = 1/2$, and fix the variance σ_0 at $1/2$. We vary α by changing the dimensionality D. To keep the Bayes error fixed, we simultaneously adapt the normalized distance. For different values of α we perform

Gibbs sampling and infer the estimated centroids $\hat{\mu}_1$ and $\hat{\mu}_2$ at varying temperature. Then we compute the order parameters and thereby obtain the phase diagram shown in Figure 15 which is consistent with the theoretical and numerical study in [BS94]:

Unsplit phase: $s = r = 0$. For high temperature and large α the estimated cluster centroids coincide, i.e. $\hat{\mu}_1 = \hat{\mu}_2$.

Ordered split phase: $s, r \neq 0$. For values of $\alpha > \alpha_c = 4u^{-4}$, the single cluster obtained in the unsplit phase splits into two clusters such that the projection of the distance vector between the two estimated and the two true sources is nonzero.

Random split phase: $s \neq 0, r = 0$. For $\alpha < \alpha_c$, the direction of the split between the two estimated centers is random. Therefore, r vanishes in the asymptotic limits. The experiments also find such a meta-stability at low temperatures which correspond to the disordered spin-glass phase in statistical physics. This experimental meta-stability is illustrated by a dashed line in Figure 15.

Therefore, as temperature decreases, different types of phase transitions can be observed:

1. $\alpha \gg \alpha_c$: Unsplit $\rightarrow$ Ordered. We investigate this scenario by choosing $D = 100$ and then $\alpha = 5$. The order parameter r in Figure 16a shows the occurrence of such a phase transition.

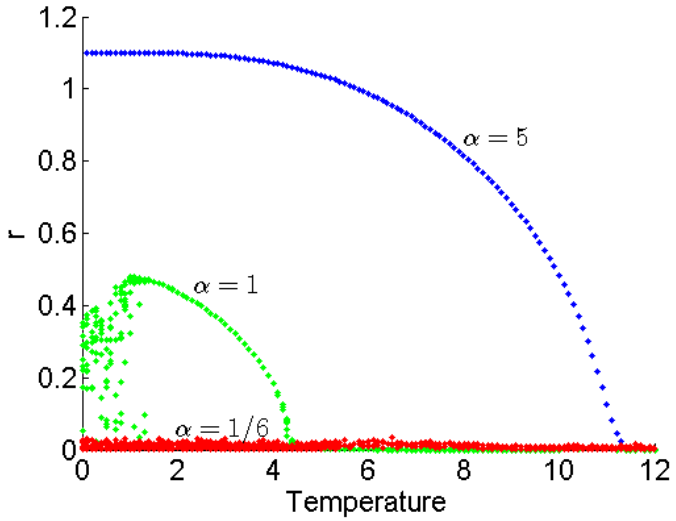
2. $\alpha \gtrapprox \alpha_c$: Unsplit $\rightarrow$ Ordered $\rightarrow$ Random. With $N = D = 500$, we then have $\alpha = 1$. The behavior of the parameter r is consistent with the phase sequence "Unsplit $\rightarrow$ Ordered $\rightarrow$ Random" as the temperature decreases. This result is consistent with the previous study in [BS94].

3. $\alpha \ll \alpha_c$: Random phase. With the choice of $D = 3000, \alpha = 1/6$ then r is always zero. This means there is almost no overlap between the true and the estimated centroids.

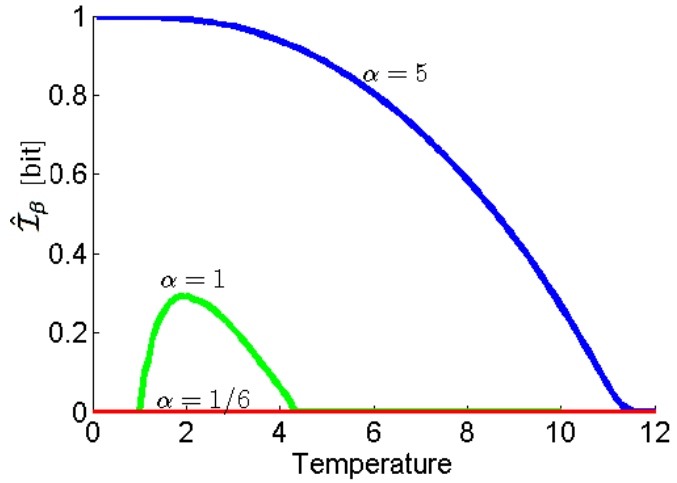
As mentioned before, changing the dimensionality affects the complexity of the problem. Therefore, we adapt the distance between the true centroids to keep the Bayes error fixed. In the following, we study generalization capacity for each of these phase transitions and compare them with the results we obtain in simulations.

5.5.2 *Generalization Capacity Analysis of Phase Transition in Learnability Limits*

Given the two datasets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ drawn from the same source, we numerically compute $\hat{\jmath}_\beta$ for the entire interval of β to obtain the generalization capacity. Figure 16b shows this numerical analysis for the



(a) Overlap



(b) $\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$

Figure 16: Experimental study of the overlap r and $\hat{\jmath}_\beta$ in different learnability limits. The problem complexity is kept constant while varying the number of objects per dimension α.

three different learnability limits. The generalization capacity reflects the difference between the three scenarios described above:

1. **Unsplit** $\rightarrow$ **Ordered**: The centroids are perfectly estimated. The generalization capacity attains the theoretical maximum of 1 bit at low temperature.

2. **Unsplit** $\rightarrow$ **Ordered** $\rightarrow$ **Random**: The strong meta-stability for low temperatures prevents communication. $\hat{\jmath}_\beta$ is maximized at the lowest temperature above this random phase.

3. **Random**: The centroids are randomly split. Therefore, there is no information between the true and the estimated centroids

over the entire temperature range. In this regime $\hat{\mathcal{I}}_\beta$ is always 0 over all values of β.

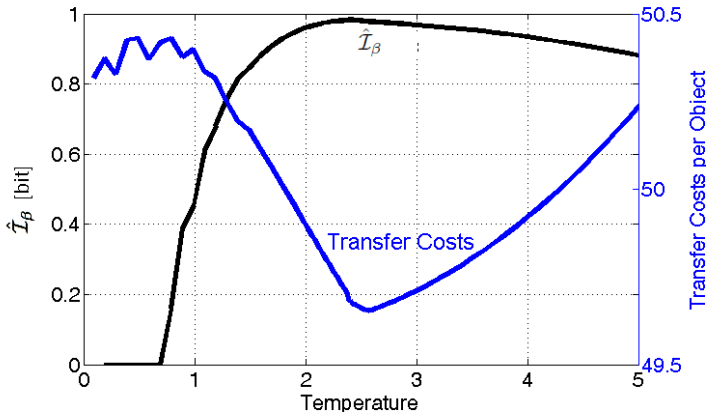


Figure 17: Transfer costs and $\hat{\mathcal{I}}_\beta$ for a mixture of two Gaussians when the number of observations per dimensions is $\alpha = 5$. The Gibbs sampler is initialized with four centroids to enable the sampler to overfit.

We extend the study to a hypothesis class of 4 centroids, thus enabling the sampler to overfit. Using Gibbs sampling on $\mathbf{X}^{(1)}, \mathbf{X}^{(2)} \in \mathbb{R}^{500\times 100}$ (scenario (1)) under an annealing schedule, we compute the clusterings $c(\mathbf{X}^{(1)})$ and $c(\mathbf{X}^{(2)})$. At each temperature, we then compute $\hat{\mathcal{I}}_\beta$ and the transfer costs. In this way, we study the relationship between generalization capacity and minimum transfer costs in the asymptotic limits. Figure 17 illustrates the consistency of the costs of the transferred clustering solution with the generalization capacity. Furthermore, in the annealing procedure, the correct model order, i.e. $\hat{K} = 2$, is attained at the temperature that corresponds to the generalization capacity.

5.6 CONCLUSION

We employed the generalization capacity principle to address the model order selection and model validation questions for clustering algorithms. For an algorithm with a fixed similarity measure and a fixed number of clusters, the optimal information rate is achieved by computing generalization capacity. Then the setting with the highest generalization capacity is selected as superior. In order to overcome the computational limits, we utilized efficient approximation techniques such as *mean-field approximation* to compute the most similar factorial weights in an information-theoretic manner. We then extended the framework to analyzing the ad hoc algorithms that do not produce a trajectory of weight distributions by employing a Hamming metric in the solution space. This study extends the notion of

model validation to the algorithms described *without* an explicit cost function.

We exemplified the framework to study Gaussian mixture models in different settings. Likelihood estimation in high dimensional setting reveals several phase transitions depending on the effective dimensionality of the problem and the computational temperature. Generalization capacity confirms the appearance of such phase transitions consistent to the order parameters. Moreover, it yields consistent results with BIC and MTC.

In the next chapter, we will study the application of generalization capacity to measure the information content in different graph clustering methods.

6 ANALYSIS OF GRAPH CLUSTERING MODELS

Arguably, K-means is one of the preferred choices for clustering in many application domains. This model assumes vector data in an Euclidean space. What if the measurements are characterized by relations, such as e.g. pairwise similarities? Clustering relation data is considered a more complicated problem as the inherent structure is encoded in N^2 pairwise relations [HB97]. This type of data appears frequently in many applications such as molecular biology, social networks, linguistics, web users, etc.

We consider model validation for graph clustering methods. We study in detail the properties of several graph partitioning models: Pairwise Clustering, Normalized Cut, Ratio Cut, Dominant Set clustering and Correlation Clustering. We particularly analyze the influence of shifting the pairwise similarities on the performance of each model. We then propose augmenting the basic Min Cut model by a (negative) *shift* parameter[1]. A negative shift renders Min Cut to produce more balanced clusters. However, this approach is different from standard models which aim at normalizing the Min Cut clusters in an arbitrary way to avoid splitting very small clusters. Our approach provides a prototypical model whose optimal parametrization is attained by performing a search over the space of alternatives. The *context sensitive* adaptation of the prototypical model advocates a scientific procedure for computing and validating the optimal model adapted to the specific application at hand, rather than an elegant design which is supposed to be more *art* rather than being *science* [BDvP06].

6.1 NORMALIZED CLUSTERING MODELS

We study clustering relation data where X_{ij} refers to the pairwise similarity between objects i and j. In this section, we analyze the models that normalize the clusters by a cluster dependent factor such as the size of clusters (Pairwise Clustering and Ratio Cut) or the degree of clusters (Normalized Cut). In the next section, we will study the models developed based on dynamical systems and evolutionary game theory (Dominant Set clustering) and then the quadratic models (Correlation Clustering and Min Cut).

1 Min Cut is the most basic model used to introduce and motivate graph clustering methods. This model splits very small clusters. Thereby, normalized clustering criteria, for instance, propose to normalize the clusters by a function of the size of the clusters (see e.g. [SMoo]).

6.1.1 *Pairwise Clustering*

A relation dataset is often mathematically characterized by a graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ with vertex set $\mathbf{O}$ and edge weights $\mathbf{X}$. Given a graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$, the *Pairwise Clustering* (PC) cost function is defined as[2] [HB97]

$$R^{PC}(c, \mathbf{X}) = -\frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} \frac{X_{ij}}{|\mathbf{O}_k|}. \tag{98}$$

Or, equivalently,

$$R^{PC}(c, \mathbf{X}) = -\frac{1}{2} \sum_{k=1}^{K} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} M_{ik} M_{jk} X_{ij}}{\sum_{l=1}^{N} M_{lk}}. \tag{99}$$

The empirical minimizer solution $c^{\perp}(\mathbf{X})$ and the corresponding assignments $\mathbf{M}$ are obtained by minimizing R^{PC}. The minimization is an $\mathcal{NP}$-hard problem [Bru78], and some approximation heuristics such as mean-field annealing [HB97] have been proposed.

This cost function essentially sums the average similarities per cluster. Therefore, adding a constant to all pairwise similarities shifts the cost value by a constant multiplied by the number of objects. At the same time, it does not modify the order of the clusterings induced by the costs [RLKB03], nor it changes the generalization capacity.

This invariance renders the embedding of the objects into a $N-1$ dimensional kernel space possible. The similarity X_{ij} is then interpreted as a scalar product between two vectors representing objects i and j. If we appropriately convert similarities $\mathbf{X}$ into dissimilarities ($\mathbf{D} = \text{constant} - \mathbf{X}$), then Pairwise Clustering equivalently performs K-means clustering in kernel space. The calculation of the weight sums, which can be performed analytically for K-means clustering, is hence exact for Pairwise Clustering.

Thereby, given the pairwise similarity matrix $\mathbf{X}$, the following procedure is performed to find a vectorial representation of the objects [RLKB03]:

1. Convert $\mathbf{X}$ into a distance matrix: $\mathbf{D} = \text{constant} - \mathbf{X}$.
2. Center the matrix $\mathbf{D}$ by $\mathbf{W} \leftarrow -\frac{1}{2}\mathbf{ADA}$ where $\mathbf{A}$ is defined as: $\mathbf{A} = \mathbf{I}_N - \frac{1}{N}\mathbf{e}_N\mathbf{e}_N^T$.
3. Compute the smallest eigenvalue of $\mathbf{W}$, if it is negative, add this value to the off-diagonal elements of $\mathbf{W}$. This transformation is the minimal shift required to convert $\mathbf{W}$ to squared Euclidean pairwise distances.

2 An equivalent model, called Ratio Assoc [SMoo], minimizes the same criterion multiplied by a constant factor, i.e. $R^{RA}(c, \mathbf{X}) = 2R^{PC}(c, \mathbf{X})$.

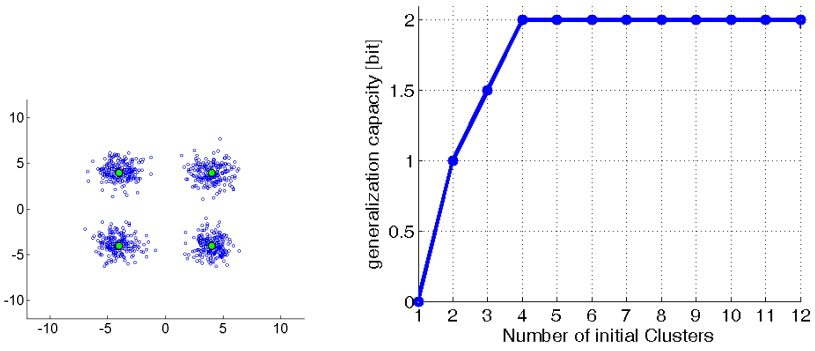
4. Decompose $\mathbf{W}$ to its eigenbasis, i.e. $\mathbf{W} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V} = (v_1, ..., v_N)$ contains the eigenvectors and $\mathbf{\Lambda} = \mathtt{diag}(\lambda_1, ..., \lambda_N)$ is a diagonal matrix of eigenvalues $\lambda_1 \geqslant ... \geqslant \lambda_d \geqslant \lambda_{d+1} = 0 = ... = \lambda_N$.

5. Calculate the $N \times d$ matrix $\mathbf{Y}_d = \mathbf{V}_d(\mathbf{\Lambda}_d)^{1/2}$, with $\mathbf{V}_d = (v_1, ..., v_d)$ and $\mathbf{\Lambda}_d = \mathrm{diag}(\lambda_1, ..., \lambda_d)$.

6. The rows of $\mathbf{Y}_d$ contain the vectors in d dimensional space. We perform Gibbs sampling to obtain the potentials h_{ik}.

NUMERICAL STUDY OF $\mathcal{GC}$ FOR PAIRWISE CLUSTERING. For the purpose of illustration, we consider two sets of objects $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ of identical size, consisting of $N = 800$ objects each. Both datasets are drawn from four isotropic Gaussian sources. For each source, the component parameters are $\pi_k = 1/4$, the means are $\boldsymbol{\mu} = [(4,4); (-4,4); (-4,-4); (4,-4)]$ and the covariances are isotropic $\boldsymbol{\Sigma} \in \{1 \cdot \mathbf{I}, 5 \cdot \mathbf{I}\}$. The pairwise Euclidean distances are then converted to pairwise similarities to give $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

On the basis of this information, the potentials $\{h_{i,c(i)}, 1 \leqslant i \leqslant N\}$ are calculated by performing annealed Gibbs sampling for different numbers of initial clusters (varying from 1 to 10). For different β, $\hat{\jmath}_\beta$ is calculated using the potentials h_{ik}. The generalization capacity is then obtained by maximization.



(a) The data generated by $\boldsymbol{\Sigma} = 1 \cdot \mathbf{I}$ (b) Generalization capacity for $\boldsymbol{\Sigma} = 1 \cdot \mathbf{I}$

Figure 18: Generalization capacity for well-separated clusters as a function of K, the number of initial clusters: Generalization capacity saturates at 2 bits of information per object for the true number of clusters, i.e. $K = 4$. It then stays constant if the Gibbs sampling is initialized even with more clusters than the true number of clusters. However the effective number of clusters at the optimal temperature remains still 4.

Figures 18 and 20 show the datasets (a) and the generalization capacity (b) for different numbers of initial clusters. In this analysis,

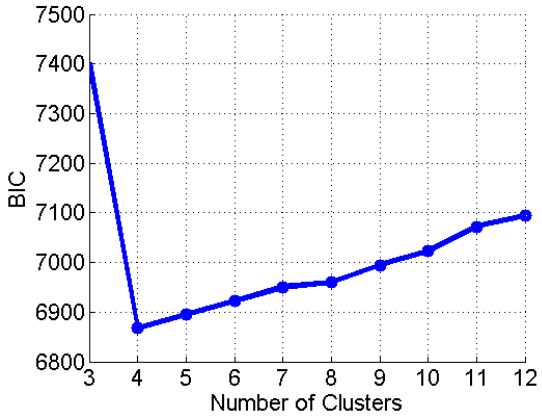
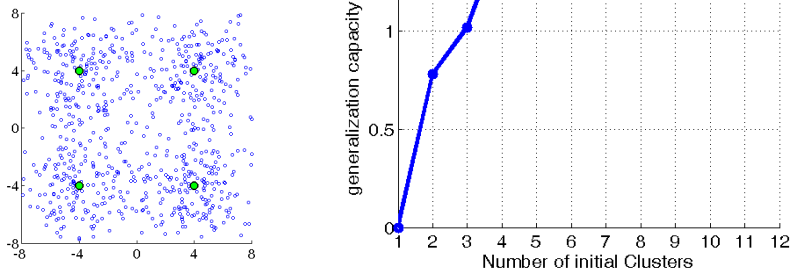


Figure 19: BIC score for $\Sigma = 1 \cdot \mathbf{I}$. Both BIC and $\mathcal{GC}$ principles yield consistent results on the two dimensional data set in Figure 18a.



(a) The data generated with $\Sigma = 5 \cdot \mathbf{I}$ (b) Generalization capacity for $\Sigma = 5 \cdot \mathbf{I}$

Figure 20: Generalization capacity of overlapping clusters for different number of initial clusters: Generalization capacity saturates at almost 1.5 bits of information per object for the true number of clusters, i.e. $K = 4$. By increasing the number of clusters even more, generalization capacity might decrease slightly due to uneven volume estimation effects. This effect disappear when there are the same number of degenerate clusters per each source, e.g. $K = 8$ and $K = 12$. However the effective number of clusters at the optimal temperature remains still 4.

1. $\mathcal{GC}$ saturates respectively at 2 and 1.5 bits per object for the true number of clusters.

2. By performing the Gibbs sampling with an excess of initial clusters, the effective number remains 4 at the optimal temperature. For example, if the number of initial clusters is chosen to be 8, $\mathcal{GC}$ computes an appropriate regularization of the annealing scheme such that at the optimal resolution, i.e. at $\beta^\star$ the effective number of clusters stays 4. The corresponding generalization capacity is the same as the generalization capacity with 4 clusters.

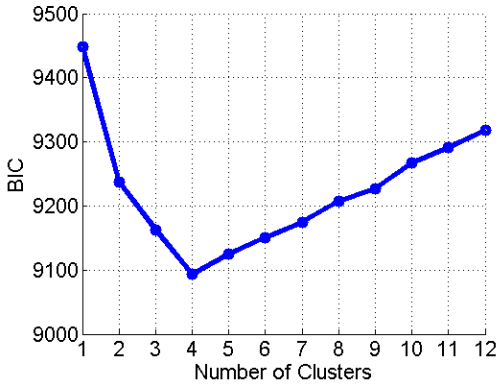
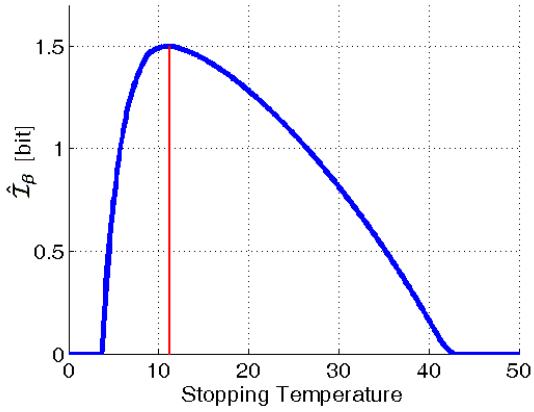


Figure 21: Consistency of $\mathcal{GC}$ and BIC for $\boldsymbol{\Sigma} = 5 \cdot \mathbf{I}$, i.e. the dataset depicted in Figure 20a.

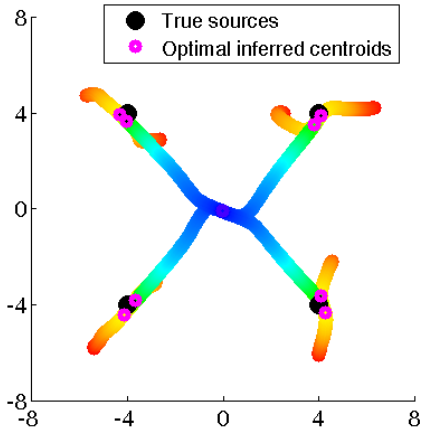
Due to degeneracy in the complicated case $\boldsymbol{\Sigma} = 5 \cdot \mathbf{I}$, the generalization capacity might slightly decrease when the Gibbs sampler is initialized with more clusters than the true number of sources. Note that the effective number of clusters is still equal to the true number of sources. However, the degeneracy effect disappears when each source has the same number of degenerate clusters, i.e. when we initialize the Gibbs sampler with 8 or 12 clusters. As shown in Figures 19 and 21, BIC and $\mathcal{GC}$ selections are consistent for these datasets.

EVOLUTION OF THE CLUSTERS WITH β. We analyze the case that Gibbs sampling is performed with an excess of initial clusters. Figure 22 illustrates in detail the trajectories of the clusters for the case $\boldsymbol{\Sigma} = 5 \cdot \mathbf{I}$ and $K = 8$. As a function of the (inverse) temperature, the positions of the centroids diverge as the system cools down (increase of β) and the model parameters are optimized. Figure 22b shows the positions of the inferred centroids. The colors of the trajectories indicate the value of β. $\beta \approx 0$ is dark blue and $\beta \gg 1$ is red. The transition from green to yellow denotes the simultaneous split of four to eight clusters at $\beta^\star$. At high temperature all centroids coincide, indicating that the optimizer favors a single cluster. At very low to zero temperature, the algorithm estimates 8 clusters with locations strongly determined by fluctuations. Figure 22a depicts $\hat{\mathcal{I}}_\beta$ as a function of β. The optimal temperature corresponding to the generalization capacity is the lowest temperature at which the correct number of clusters is found.

SHIFTED PAIRWISE CLUSTERING. By shifting the pairwise similarities by s, the Pairwise Clustering cost function is written as [RLKB03]



(a) $\hat{\jmath}_\beta$



(b) Cluster trajectories

Figure 22: Annealed Gibbs sampling for Pairwise Clustering. The influence of the stopping temperature on $\hat{\jmath}_\beta$ and on the positions of the cluster centroids are shown. The colors of the trajectories indicate the value of β. $\beta \approx 0$ is dark blue and $\beta \gg 1$ is red. The transition from green to yellow denotes the simultaneous split of four to eight clusters at $\beta^\star$.

$$
\begin{aligned}
R^{PC}(c, \mathbf{X}, s) &= -\frac{1}{2}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{O}_k}\frac{X_{ij}+s}{|\mathbf{O}_k|} \\
&= -\frac{1}{2}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{O}_k}\frac{X_{ij}}{|\mathbf{O}_k|} - \frac{1}{2}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{O}_k}\frac{s}{|\mathbf{O}_k|} \\
&= -\frac{1}{2}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{O}_k}\frac{X_{ij}}{|\mathbf{O}_k|} - \frac{1}{2}\sum_{k=1}^{K}\frac{s|\mathbf{O}_k|^2}{|\mathbf{O}_k|} \\
&= \underbrace{-\frac{1}{2}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{O}_k}\frac{X_{ij}}{|\mathbf{O}_k|}}_{\text{PC without shift}} - \underbrace{\frac{1}{2}sN}_{\text{constant}} . \qquad (100)
\end{aligned}
$$

Therefore, Pairwise Clustering is *invariant* under shifting the pairwise similarities.

6.1.2 *Normalized Cut*

Normalized Cut (NCut) [SMoo] splits a graph by normalizing each cluster by the degree of the cluster. The cost function is defined as

$$\begin{aligned} R^{NCut}(c, \mathbf{X}) &= \sum_{k=1}^{K} \frac{links(\mathbf{O}_k, \mathbf{O} \setminus \mathbf{O}_k)}{degree(\mathbf{O}_k)} \\ &= \sum_{k=1}^{K} \frac{\sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O} \setminus \mathbf{O}_k} X_{ij}}{\sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}} X_{ij}}. \end{aligned} \tag{101}$$

It has been shown that Normalized Cut lifts the dataset to an infinite-dimensional feature space and splits the data by passing a hyperplane through a "gap" in the lifted data [RRo4]. The method then puts objects that occur on the same side of the hyperplane in the same cluster. Thereby, Normalized Cut can be interpreted as maximizing a gap that weights the objects according to their distance from the center of mass: The objects away from the center have a larger weight than those closer to the center.

A connection between $links(.,.)$ and $degree(.)$ can be established in the following way:

$$links(\mathbf{O}_k, \mathbf{O} \setminus \mathbf{O}_k) = degree(\mathbf{O}_k) - links(\mathbf{O}_k, \mathbf{O}_k). \tag{102}$$

Thereby, the Normalized Cut cost function can be written as

$$\begin{aligned} R^{NCut}(c, \mathbf{X}) &= \sum_{k=1}^{K} \frac{degree(\mathbf{O}_k) - links(\mathbf{O}_k, \mathbf{O}_k)}{degree(\mathbf{O}_k)} \\ &= K - \sum_{k=1}^{K} \frac{\sum_{i,j \in \mathbf{O}_k} X_{ij}}{\sum_i \sum_{j \in \mathbf{O}_k} X_{ij}}. \end{aligned} \tag{103}$$

Hence, Pairwise Clustering and Normalized Cut can be assumed as methods for finding the weighted *well-associated* regions of the graph. The prototype cost function is written as

$$R^{wAssoc}(c, \mathbf{X}) = -\sum_{k=1}^{K} \frac{links(\mathbf{O}_k, \mathbf{O}_k)}{\sum_{i \in \mathbf{O}_k} W_i}, \tag{104}$$

where W_i denotes the weight of object i. For Pairwise Clustering and Normalized Cut the weights are $W_i^{PC} = 1$ and $W_i^{NCut} = degree(i)$, respectively. Pairwise Clustering does not distinguish between objects, while Normalized Cut assumes a stronger weight on the objects far away from the center of mass. This formulation provides a framework for comparing the two cost functions.

EFFICIENT MEAN-FIELD CALCULATION FOR NORMALIZED CUT. To compute $\hat{\jmath}_\beta$, we use the Boltzmann weights $\exp(-\beta h_{ik})$, where the potentials h_{ik} are obtained by performing a mean-field annealing. The **M**-step of mean-field annealing requires computing the expectation $\mathbb{E}_{\mathbf{Q}_{i\to k}}\{R^{\mathcal{A}}(c,\mathbf{X})\}$. For computability reasons, as discussed in Chapter 5, we approximate the expectation by replacing the cluster assignment variables in the cost function by their probabilistic counterparts, i.e.

$$h_{ik} \approx R^{NCut}(\mathbb{E}_{\mathbf{Q}_{i\to k}}\{c\},\mathbf{X})\,. \tag{105}$$

The straightforward computation of h_{ik} is still inefficient as it would require a computational time $O(N^2)$ and therefore $O(N^3)$ for one complete sweep of objects.

In order to provide a more efficient implementation, we define two K-dimensional reference vectors $\mathtt{ref1}$ and $\mathtt{ref2}$.

$$\mathtt{ref1}(u) = \sum_{i=1}^{N}\sum_{j=1}^{N} q_{iu}q_{ju}X_{ij}. \tag{106}$$

$$\mathtt{ref2}(u) = \sum_{i=1}^{N}\sum_{j=1}^{N} q_{iu}(1-q_{ju})X_{ij}. \tag{107}$$

The vectors $\mathtt{ref1}$ and $\mathtt{ref2}$ are computed once at the beginning of every sweep, and they are then used as reference for further computations. Given $\mathtt{ref1}$ and $\mathtt{ref2}$, we compute two other K-dimensional vectors $\mathtt{red1}^{-i}$ and $\mathtt{red2}^{-i}$ by eliminating the object i from $\mathtt{ref1}$ and $\mathtt{ref2}$, i.e.

$$\mathtt{red1}^{-i}(u) = \mathtt{ref1}(u) - q_{iu}X_{ii} - 2q_{iu}\sum_{\substack{j=1\\ j\neq i}}^{N} q_{ju}X_{ij}\,. \tag{108}$$

$$\begin{aligned}\mathtt{red2}^{-i}(u) = \mathtt{ref2}(u) &- q_{iu}\sum_{j=1}^{N}(1-q_{ju})X_{ij}\\ &-(1-q_{iu})\sum_{j=1}^{N} q_{ju}X_{ji}\,.\end{aligned} \tag{109}$$

Now, for computing h_{ik}, we only need to add object i to cluster k and update the vectors $\mathtt{red1}^{-i}$ and $\mathtt{red2}^{-i}$ such that they take into account the *hard assignment* of object i to cluster k.

$$\mathrm{upd1}^{i}(u) = \begin{cases} \mathrm{red1}^{-i}(u) + X_{ii} + 2\sum_{\substack{j=1 \\ j\neq i}}^{N} q_{jk} X_{ij} & \text{if } u = k \\ \mathrm{red1}^{-i}(u) & \text{if } u \neq k \end{cases} \tag{110}$$

For the second term $\mathrm{upd2}^{i}$ we then have

$$\mathrm{upd2}^{i}(u) = \begin{cases} \mathrm{red2}^{-i}(u) + \sum_{\substack{j=1 \\ j\neq i}}^{N} (1 - q_{jk}) X_{ij} & \text{if } u = k \\ \mathrm{red2}^{-i}(u) + \sum_{\substack{j=1 \\ j\neq i}}^{N} q_{ju} X_{ij} & \text{if } u \neq k \end{cases} \tag{111}$$

The potential h_{ik} is thereby computed by

$$h_{ik} = K - \sum_{u=1}^{K} \frac{\mathrm{upd1}^{i}(u)}{\mathrm{upd1}^{i}(u) + \mathrm{upd2}^{i}(u)}\,. \tag{112}$$

Algorithm 5 describes the detailed procedure for computing the potentials and $\hat{\jmath}_\beta$ at each inverse temperature β.

SHIFTED NORMALIZED CUT. By shifting the pairwise similarities, the Normalized Cut cost function is written as

$$R^{\mathrm{NCut}}(c, \mathbf{X}, s) \;=\; \sum_{k=1}^{K} \frac{\sum_{i\in\mathbf{O}_k} \sum_{j\in\mathbf{O}\setminus\mathbf{O}_k} X_{ij} + s}{\sum_{i\in\mathbf{O}_k} \sum_{j\in\mathbf{O}} X_{ij+s}}. \tag{113}$$

The Normalized Cut cost function *is not* shift invariant, in contrast to Pairwise Clustering. However, for the special case of balanced clusterings, i.e. $|\mathbf{O}_k| = N/K \;\; \forall 1 \leqslant k \leqslant K$, and similar distribution of intra-cluster similarities among all clusters, all the row-sums of the similarity matrix $\mathbf{X}$ tend to be similar. The objects then share the same degree, i.e. $\sum_{j=1}^{N} X_{ij} = \mathrm{constant}$. In this case, the Normalized Cut cost functions becomes equivalent to the Pairwise Clustering cost function [RLKB03]. Therefore, for clustering problems with similar group structure and balanced clusters, large differences between the models become vanishingly small. Thereby, shifting the pairwise similarities does not significantly change the generalization capacity. This analysis explains the similar performance of graph partitioning models in large-scale comparison studies for image segmentation applications [SS01].

6.1.3 *Adaptive Ratio Cut*

Ratio Cut (RC) clustering [CSZ94] performs a cut where normalizes each cluster by the corresponding size. The cost function is defined as

Algorithm 5 Calculate $\hat{\jmath}_{\beta}^{\mathcal{A}_K}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ for NCut

for $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ **do**
  Initialize $h_{ik}^{(0)}$ and $q_{ik}^{(0)}$ randomly.
  Initialize the concentration parameter $\beta \leftarrow \beta_0$.
  while $\beta \leqslant \beta_{final}$ **do**
    $t \leftarrow 0$.
    repeat
      E-step:
      for all i, k **do**
        Estimate $q_{ik}^{(t+1)}$ as a function of $h_{ik}^{(t)}$.
      end for
      M-step:
      Compute $\mathtt{ref1}$ and $\mathtt{ref1}$ (according to Eqs. 106 and 107).
      for all i **do**
        Compute $\mathtt{red1}^{-i}$ and $\mathtt{red2}^{-i}$ (according to Eqs. 108 and 109).
        for all k **do**
          Compute $\mathtt{upd1}^{i}$ and $\mathtt{upd2}^{i}$ w.r.t cluster k (according to Eqs. 110 and 111).
          Compute $h_{ik}^{(t+1)} = K - \sum_{u=1}^{K} \frac{\mathtt{upd1}^{i}(u)}{\mathtt{upd1}^{i}(u)+\mathtt{upd2}^{i}(u)}$.
        end for
      end for
      $t \leftarrow t + 1$.
    until $h_{ik}^{(t)}$ and $q_{ik}^{(t)}$ converge
    Update β.
    $q_{ik}^{(0)} \leftarrow q_{ik}^{(t)}$.
    $h_{ik}^{(0)} \leftarrow h_{ik}^{(t)}$.
  end while
end for
Compute $H(\mathcal{A}_K)$ (for β_{final}).
Compute $\hat{\jmath}_{\beta}^{\mathcal{A}_K}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ at different β.

$$\begin{aligned} R^{RC}(c, \mathbf{X}) &= \sum_{k=1}^{K} \frac{\text{links}(\mathbf{O}_k, \mathbf{O} \setminus \mathbf{O}_k)}{|\mathbf{O}_k|} \\ &= \sum_{k=1}^{K} \frac{\sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O} \setminus \mathbf{O}_k} X_{ij}}{|\mathbf{O}_k|}. \end{aligned} \tag{114}$$

Ratio Cut has a tendency to separate small sets of isolated objects in the graph. The cut $\text{links}(\mathbf{O}_k, \mathbf{O} \setminus \mathbf{O}_k)$ increases with the number of edges between $\mathbf{O}_k$ and $\mathbf{O} \setminus \mathbf{O}_k$ (i.e. $O(N^2)$ for a fully connected graph), while it is normalized by the size of the cluster $|\mathbf{O}_k|$ (i.e. $O(N)$). For this reason, Normalized Cut proposes to normalize the cut by the degree of the cluster, rather than the size of the cluster [SM00].

To demonstrate such a situation, we generate two clouds of data with fixed centers but varying variances between 0.1 and 0.4. Figure 23a shows the datasets and the clusters produced by Ratio Cut. The clusters are discriminated by blue and red colors. Ratio Cut splits a singleton object for $var = 0.3, 0.4$.

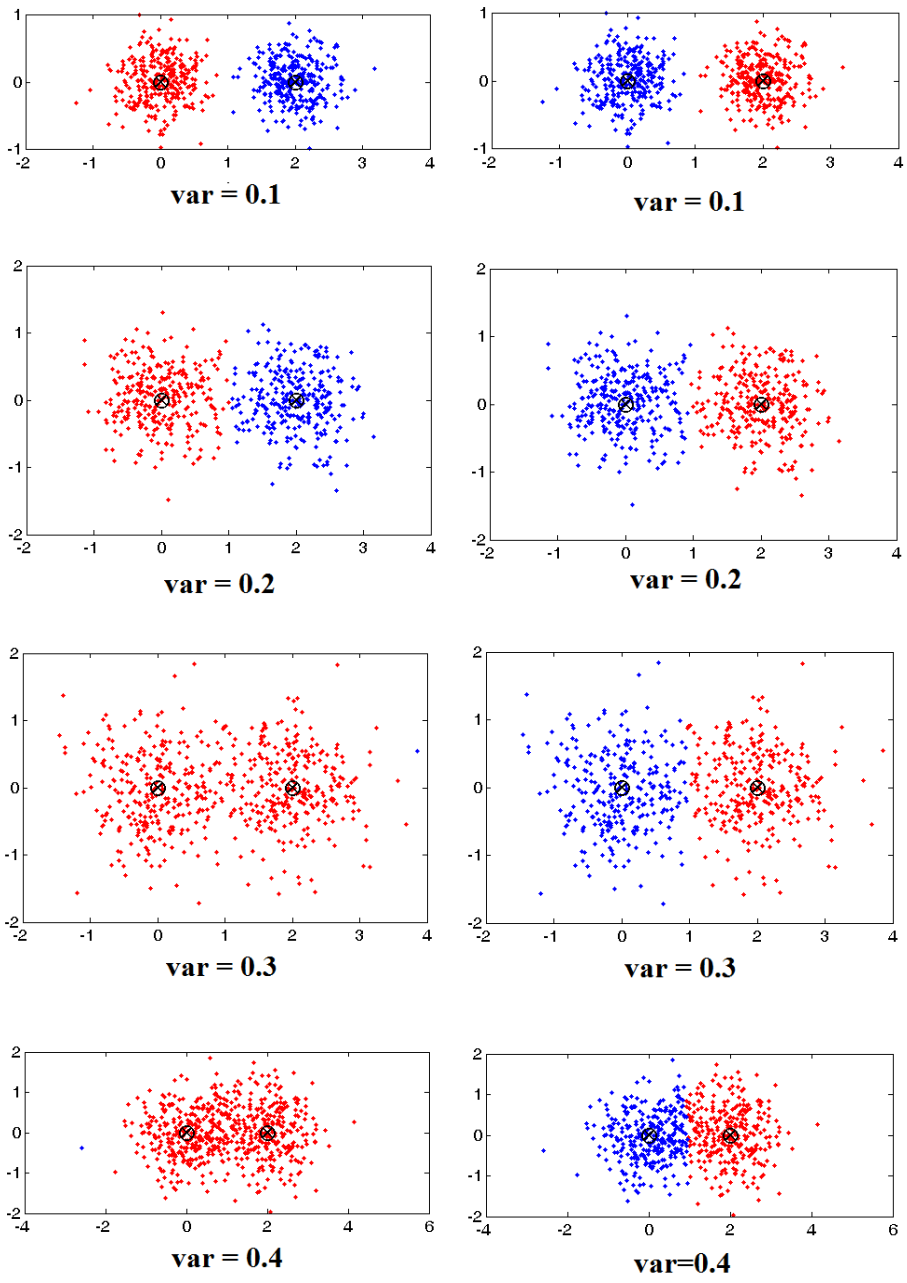
One way to overcome this problem is to apply a stronger constraint on the size of clusters. p-Spectral Clustering [BH09] proposes a non-linear generalization of spectral clustering based on the second eigenvector of the graph p-Laplacian. It has been then shown that it can be interpreted as a generalization of graph clustering models such as Ratio Cut. The standard graph Laplacian Δ is interpreted as an operator which induces a quadratic functional form [HAvL07]. p-Spectral Clustering extends the definition of the operator to the general form (for $p > 1$) [BH09]

$$\langle f, \Delta_p f \rangle = \frac{1}{2} \sum_{i,j=1}^{N} X_{ij} |f_i - f_j|^p. \tag{115}$$

p-Spectral Clustering is an iterative consecutive clustering procedure that at each step performs a bi-partitioning of one of the existing clusters until K clusters are constructed. The underlying cost function for bi-partitioning the graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ into two sets $\mathbf{O}_a$ and $\mathbf{O}_b$ is given by

$$R^{pLap}(c, \mathbf{X}) = \text{links}(\mathbf{O}_a, \mathbf{O}_b) \left(\frac{1}{|\mathbf{O}_a|^{\frac{1}{p-1}}} + \frac{1}{|\mathbf{O}_b|^{\frac{1}{p-1}}} \right)^{p-1}. \tag{116}$$

We introduce Adaptive Ratio Cut (ARC) as a generalization of the p-Laplacian cost function to partition the graph into K parallel clusters:



(a) Ratio Cut clustering (b) Adaptive Ratio Cut

Figure 23: Clusterings performed by standard Ratio Cut (Figure 23a) and Adaptive Ratio Cut (Figure 23b) on datasets with different variances. The clusters are discriminated by blue and red colors. Ratio Cut has a tendency to split isolated singleton objects. Adaptive Ratio Cut overcomes this problem by performing a stronger normalization.

$$R^{ARC}(c, \mathbf{X}) = \sum_{k=1}^{K} \sum_{k'=k+1}^{K} links(\mathbf{O}_k, \mathbf{O}_{k'}) \left(\frac{1}{|\mathbf{O}_k|^{\frac{1}{p-1}}} + \frac{1}{|\mathbf{O}_{k'}|^{\frac{1}{p-1}}} \right)^{p-1} \quad (117)$$

For the special case of $p = 2$, the Adaptive Ratio Cut cost function is equivalent to the standard Ratio Cut model defined in Eq. 114. Moreover, in the limit as $p \to 1$ Adaptive Ratio Cut converges to the multiway Cheeger Cut criterion, defined by

$$R^{Cheeger}(c, \mathbf{X}) \quad = \quad \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \frac{links(\mathbf{O}_k, \mathbf{O}_{k'})}{min(|\mathbf{O}_k|, |\mathbf{O}_{k'}|)}. \tag{118}$$

The value of p renders a constraint on the size of clusters. Figure 23b illustrates the ARC clustering on the datasets with different variances. For $p = 1, 1.1, 1.2, 1.4, 1.6$, ARC produces two balanced clusters for all variances. For $p = 1.8$, ARC computes two balanced clusters for $var = 0.1, 0.2, 0.3$, but it gives only one cluster plus a singleton for $var = 0.4$. For $p = 2$, ARC computes two balanced clusters when $var = 0.1, 0.2$ and produces only one cluster plus a singleton when $var = 0.3, 0.4$. Please note that ARC with $p = 2$ is equivalent to the standard Ratio Cut clustering (Figure 23a).

OPTIMIZED MEAN-FIELD CALCULATION FOR ADAPTIVE RATIO CUT. For this model, similar to Normalized Cut, the potentials h_{ik} are computed by mean-field annealing to obtain the factorial Boltzmann weights $\exp(-\beta h_{ik})$. We follow a similar procedure to compute the potentials efficiently. The reference matrix $ref1$ and the reference vector $ref2$ are computed as:

$$ref1(u, v) = \sum_{i=1}^{N} \sum_{j=1}^{N} q_{iu} q_{jv} X_{ij} \,. \tag{119}$$

$$ref2(u) = \sum_{i=1}^{N} q_{iu} \,. \tag{120}$$

Given $ref1(u, v)$ and $ref2(u)$, then $red1^{-i}(u, v)$ and $red2^{-i}(u)$ are computed by eliminating the object i, i.e.

$$red1^{-i}(u, v) = ref1(u, v) - q_{iu} \left(\sum_{j=1}^{N} q_{jv} X_{ij} \right) - q_{iv} \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} q_{jv} X_{ij} \right) \tag{121}$$

$$red2^{-i}(u) = ref2(u) - q_{iu} \,. \tag{122}$$

To compute h_{ik}, we add the object i to cluster k and update $red1^{-i}$ and $red2^{-i}$.

$$\mathrm{upd1}^{i}(u,v) = \begin{cases} \mathrm{red1}^{-i}(u,v) + X_{ii} + 2\sum_{\substack{j=1\\ j\neq i}}^{N} q_{jv} X_{ij} & \text{if } u = v = k \\ \mathrm{red1}^{-i}(u,v) + \sum_{\substack{j=1\\ j\neq i}}^{N} q_{jv} X_{ij} & \text{if } u = k, v \neq k \\ \mathrm{red1}^{-i}(u,v) + \sum_{\substack{j=1\\ j\neq i}}^{N} q_{ju} X_{ji} & \text{if } v = k, u \neq k \\ \mathrm{red1}^{-i}(u,v) & \text{if } u \neq k, v \neq k \end{cases} \tag{123}$$

Similarly we have,

$$\mathrm{upd2}^{i}(u) = \begin{cases} \mathrm{red2}^{-i}(u) + 1 & \text{if } u = k \\ \mathrm{red2}^{-i}(u) & \text{if } u \neq k \end{cases} \tag{124}$$

Finally the potential h_{ik} is given by

$$h_{ik} = \sum_{u=1}^{K} \sum_{v=u+1}^{K} \mathrm{upd1}^{i}(u,v) \left(\frac{1}{\mathrm{upd2}^{i}(u)^{\frac{1}{p-1}}} + \frac{1}{\mathrm{upd2}^{i}(v)^{\frac{1}{p-1}}} \right)^{p-1} \tag{125}$$

A similar algorithmic procedure to Algorithm 5 is performed to compute $\hat{J}_\beta$ at every inverse temperature β.

SHIFTED ADAPTIVE RATIO CUT. In a similar way to Pairwise Clustering, the shifted Ratio Cut can be written as

$$\begin{aligned} R^{RC}(c, \mathbf{X}, s) &= \sum_{k=1}^{K} \frac{\sum_{i\in O_k} \sum_{j\in O\setminus O_k} X_{ij} + s}{|\mathbf{O}_k|} \\ &= \sum_{k=1}^{K} \frac{\sum_{i\in O_k} \sum_{j\in O\setminus O_k} X_{ij}}{|\mathbf{O}_k|} + \sum_{k=1}^{K} \frac{\sum_{i\in O_k} \sum_{j\in O\setminus O_k} s}{|\mathbf{O}_k|} \\ &= \sum_{k=1}^{K} \frac{\sum_{i\in O_k} \sum_{j\in O\setminus O_k} X_{ij}}{|\mathbf{O}_k|} + \sum_{k=1}^{K} \frac{s|\mathbf{O}_k|(N - |\mathbf{O}_k|)}{|\mathbf{O}_k|} \\ &= \sum_{k=1}^{K} \underbrace{\frac{\sum_{i\in O_k} \sum_{j\in O\setminus O_k} X_{ij}}{|\mathbf{O}_k|}}_{\text{RC without shift}} + \underbrace{sN(K-1)}_{\text{constant}}. \end{aligned} \tag{126}$$

Thereby, both Pairwise Clustering and Ratio Cut, i.e. the models that normalize the clusters by the size of clusters, are *invariant* under shifting the pairwise similarities. However, Adaptive Ratio Cut *is not shift invariant* as the shift s cannot be factored out.

6.2 GAME-THEORETIC CLUSTERING MODELS

A classical approach to clustering relation data is based on partitioning the graph according to a cost function. Correlation Clustering, Pairwise Clustering and Normalized Cut are examples of such models. Minimizing the cost function is usually $\mathcal{NP}$-hard and thereby requires an exponential computational runtime unless $\mathcal{P} = \mathcal{NP}$. Dominant Set clustering [PP07] provides a more efficient approach by extracting maximal weighted cliques through establishing a connection to a replicator dynamics derived from evolutionary game theory.

6.2.1 *Dominant Set Clustering*

Given graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ with eliminated self-loops (i.e. $X_{ii} = 0, \forall i \in \mathbf{O}$), Dominant Set (DS) clustering [PP07, LY10] seeks for the dense subgraphs, i.e. computes the modes of the graph. Basically, the method is a generalization of the mean shift algorithm [CM02] to proximity data[3]. On the other hand, Dominant Set clustering can be interpreted as extension of maximal cliques[4] to weighted graphs [PP03b], since maximal cliques of an unweighed graph do not represent the clusters properly [AM70].

The Motzkin and Straus principle [MS65] provides a continuous formulation for computing maximal cliques. Then, the concept has been generalized to edge-weighted graphs [PP07] which proposes computing the dominant sets by finding the solutions of a quadratic program

$$\begin{aligned} &\text{maximize} && \mathbf{v}^T \mathbf{X} \mathbf{v} \\ &\text{subject to} && \mathbf{v} \in \Delta \,. \end{aligned} \tag{127}$$

Δ is the standard simplex of $\mathbb{R}^N$, i.e.

$$\Delta = \left\{ \mathbf{v} \in \mathbb{R}^N : \mathbf{v} \geqslant \mathbf{0} \text{ and } \sum_{i=1}^{N} \mathbf{v}(i) = 1 \right\} . \tag{128}$$

The N-dimensional vector $\mathbf{v}$ characterizes a cluster, i.e. its components indicate the participation of objects in the cluster. A small value of a component represents a weak association of the corresponding object whereas a large value indicates a strong association. Objects not participating in the cluster have thereby zero association.

3 Mean shift operates directly on the vectorial data.

4 A clique is a complete subgraph, i.e. a subgraph wherein each pair of objects are connected. A maximal clique has the largest number of objects, i.e. it is not subset in any other clique.

The solution of the quadratic program in Eq. 127 is computed by *replicator dynamics*, a class of discrete-time dynamical systems arising in evolutionary game theory [Wei95, PP07].

$$v_i(t+1) = v_i(t)\frac{(\mathbf{X}\,\mathbf{v}(t))_i}{\mathbf{v}^T(t)\,\mathbf{X}\,\mathbf{v}(t)}\,, \quad i = 1..N. \tag{129}$$

For a symmetric similarity matrix $\mathbf{X}$, the cohesion measure $\mathbf{v}^T\mathbf{X}\,\mathbf{v}$ strictly increases during time along any trajectory of the replicator equation [Wei95].

PEELING OFF STRATEGY FOR DOMINANT SET CLUSTERING. Often, the characteristic vector $\mathbf{v}$ converges to the cluster that corresponds to the most dominant mode of the graph even if it is initialized with different random vectors. Thereby, a sequential strategy has been proposed [PP07] which at each iteration, *peels off* a cluster by solving the replicator dynamics in Eq. 129. Algorithm 6 describes the peeling off procedure.

Algorithm 6 Peel_Off_Dominant_Set_Clusters

1: Set $k \leftarrow 1$.
2: **while** $k \leqslant K$ OR $\mathbf{X}$ is not empty **do**
3: Initialize the characteristic vector $\mathbf{v}^{(k)}(0)$ for the k^{th} cluster.
4: Set $t \leftarrow 0$.
5: **repeat**
6: Solve the replicator dynamics

$$v_i^{(k)}(t+1) = v_i^{(k)}(t)\frac{(\mathbf{X}\,\mathbf{v}^{(k)}(t))_i}{\mathbf{v}^{(k)T}(t)\,\mathbf{X}\,\mathbf{v}^{(k)}(t)}\,.$$

7: $t \leftarrow t+1$.
8: **until** $\mathbf{v}^{(k)}(t)$ converges OR $t = t_{max}$
9: Output the k^{th} cluster by selecting the objects whose participations are larger than ϵ, i.e. $\mathbf{O}_k = \{i : v_i^{(k)} \geqslant \epsilon\}$.
10: Remove the selected objects from the similarity matrix $\mathbf{X}$.
11: Set $k \leftarrow k+1$.
12: **end while**
13: Assign the unlabeled objects to the 'nearest' cluster.

The two parameters ϵ and t_{max} control the evolution of the clusters. t_{max} determines the number of steps, i.e. the number of executions of the replicator dynamics until it stops. For a small t_{max}, there are still many nonzero components in the characteristic vector $\mathbf{v}$, since the replicator dynamics has not converged yet. A large t_{max} guides the replicator dynamics to converge to the mode of the graph, thereby only a very compact subset of objects is selected. ϵ determines the

cut-off threshold of the clusters, i.e. controls the size and spread of the clusters.

Dominant Set clustering can automatically find the maximal number of clusters. Therefore, in principle it is not necessary to fix the number of clusters K in advance. However, in real-world applications, which involve large and noisy datasets, there might exist many small and unstable dominant sets that are not suitable to be considered as separate clusters. Thereby, after computing the first K clusters, the algorithm assigns the remaining unlabeled objects to the nearest clusters according to some similarity measure.

SHIFTED DOMINANT SET CLUSTERING. In order to study the influence of shifting the pairwise similarities on Dominant Set clustering, we consider the shifted variant of the quadratic program introduced in Eq. 127, i.e.

$$\begin{aligned} &\text{maximize} && \mathbf{v}^\mathsf{T}(\mathbf{X} + s\,\mathbf{e}\mathbf{e}^\mathsf{T})\mathbf{v} \\ &\text{subject to} && \mathbf{v} \in \Delta\,. \end{aligned} \tag{130}$$

Then, we have

$$\begin{aligned} \mathbf{v}^\mathsf{T}(\mathbf{X} + s\,\mathbf{e}\mathbf{e}^\mathsf{T})\mathbf{v} &= \mathbf{v}^\mathsf{T}\mathbf{X}\mathbf{v} + \mathbf{v}^\mathsf{T} s\,\mathbf{e}\mathbf{e}^\mathsf{T}\mathbf{v} \\ &= \mathbf{v}^\mathsf{T}\mathbf{X}\mathbf{v} + s\,\underbrace{(\mathbf{v}^\mathsf{T}\mathbf{e})}_{=1}\,\underbrace{(\mathbf{e}^\mathsf{T}\mathbf{v})}_{=1} \\ &= \mathbf{v}^\mathsf{T}\mathbf{X}\mathbf{v} + s\,, \end{aligned} \tag{131}$$

where $\mathbf{e} = (1, 1, ...1)^\mathsf{T}$ is a vector of ones. Therefore, Dominant Set clustering is *invariant* under shifting the pairwise similarities.

However, it has been proposed in [PP03a] to shift the diagonal entries of the similarity matrix by a negative value, in order to obtain coarser clusters, which yields computing a hierarchy of clusters. The clusters obtained from the unshifted similarity matrix appear at the lowest level of the hierarchy. The larger the negative shift is the coarser the clusters are, thus the smaller the generalization capacity is. Performing a negative shift is equivalent to adding the same shift but with a positive sign to the off-diagonal pairwise similarities. Thereby, the shifted matrix is still non-negative and has a null diagonal, i.e. satisfies the conditions of Dominant Set clustering.

One can think of performing a negative shift on the off-diagonal pairwise similarities to compute a finer representation of the clusters. However, this type of shift might violate the non-negativity and null diagonal constraints. On the other hand, according to our experiments, a negative shift is effectively equivalent to applying a larger cut-off threshold when peeling off the clusters.

6.2.2 *Generalization Capacity Analysis of Dominant Set Clustering*

The Dominant Set algorithm computes K sequential characteristic vectors $\mathbf{v}^{(1)}(t), ..., \mathbf{v}^{(K)}(t)$ one after the other. For each characteristic vector, after t_{max} number of update steps, the algorithm cuts off the objects whose associations are larger than the threshold ϵ. Thereby, the algorithm performs in total $K \times t_{max}$ steps, where each step proposes a *binary* weighting scheme[5].

$$w_{t\propto\frac{1}{\gamma}}(k, \mathbf{X}, i) = 1 \quad \text{iff} \;\; v_i^{(k)}(t) \geqslant \epsilon, \qquad 0 \quad \text{otherwise.} \tag{132}$$

Or, equivalently,

$$P_{t\propto\frac{1}{\gamma}}(i, k) = 1 \quad \text{iff} \;\; v_i^{(k)}(t) \geqslant \epsilon, \qquad 0 \quad \text{otherwise.} \tag{133}$$

At each step, the weights are then used to compute $\hat{\jmath}_{t\propto\frac{1}{\gamma}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$.

$$\hat{\jmath}_t(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = H(\mathcal{A}_K) + \frac{1}{N}\sum_{i=1}^{N} \log \sum_{k=1}^{K} \left(P_t^{(1)}(i, k) \times P_t^{(2)}(i, k)\right). \tag{134}$$

The *binary* weights render strict clustering assignments, i.e. $P_t(i, k) \in \{0, 1\}$, $\forall t$. Thereby, in contrast to Boltzmann weights, the assignment variables cannot be smoothed to relax the inconsistencies between the solutions of the first and the second datasets. An inconsistency refers to occurring a different clustering assignment between $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ for at least one object i, i.e.

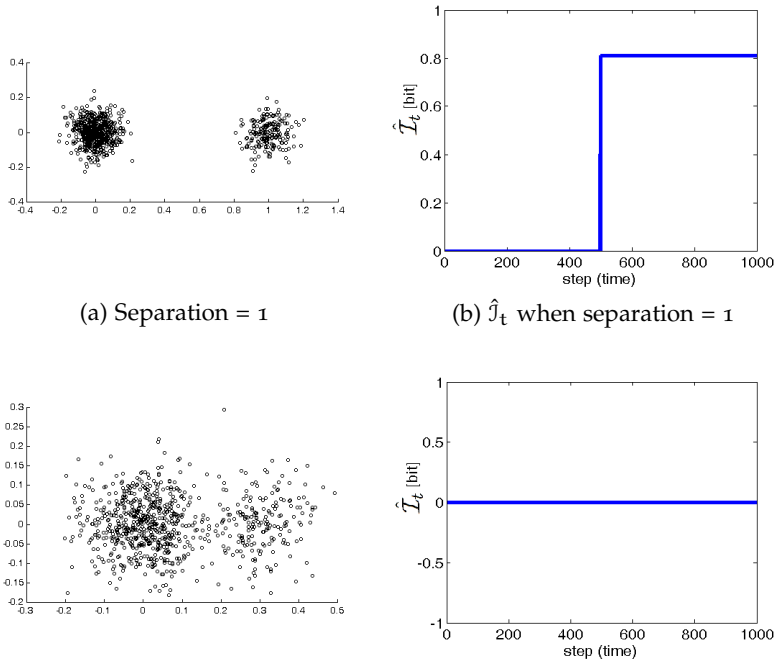
$$\exists k, k' \in \{1, ..., K\} : \; P_{ik}^{(1)} = 1 \bigwedge P_{ik'}^{(2)} = 1 \bigwedge k \neq k. \tag{135}$$

In the case of inconsistency, the sum product $\sum_{k=1}^{K} \left(P_t^{(1)}(i, k) \times P_t^{(2)}(i, k)\right)$ becomes *zero* and thus $\hat{\jmath}_t(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ drops down to zero. When the Boltzmann weights are used, the assignment probabilities become softer as the computational temperature increases, which smoothes the influence of inconsistencies. Thus, in the case of inconsistency, the product and thereby $\hat{\jmath}_\beta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ are still *nonzero*. Algorithm 6 does not provide the possibility to relax the inconsistencies[6].

Figure 24 illustrates two scenarios, i) the sources have no overlap (Figure 24a), and ii) the overlap is non-zero (Figure 24c). In the zero overlap case, the clusters are generated perfectly. Therefore, the solutions are always consistent among different instances and the capacity is maximal. In the case of a non-zero overlap, the possibility of occurring inconsistencies between the solutions of different instances exists, which renders $\mathcal{GC}$ to be zero.

5 The weights of the truncated objects remain fixed after elimination.

6 However, there also exist other issues with the peeling off strategy, e.g. removing a part of the dataset changes the scale of the problem.



(a) Separation = 1

(b) $\hat{\mathcal{I}}_t$ when separation = 1

(c) Separation = 0.3

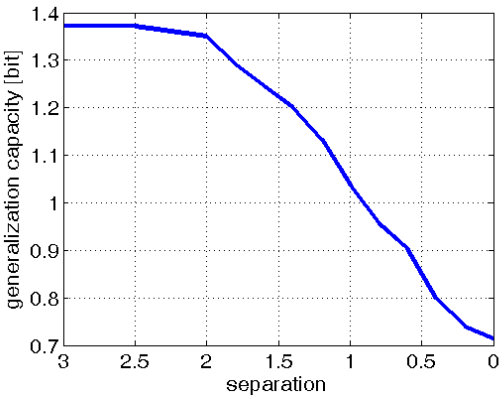
(d) $\hat{\mathcal{I}}_t$ when separation = 0.3

Figure 24: Trajectory of $\hat{\mathcal{I}}_t(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ for Algorithm 6 applied to different types of datasets. When the clusters have an overlap (e.g. Figure 24c), the solutions of different instances might be inconsistent, which renders $\mathcal{GC}$ to be zero. We propose using a Hamming metric in the solution space to overcome such deficiency.

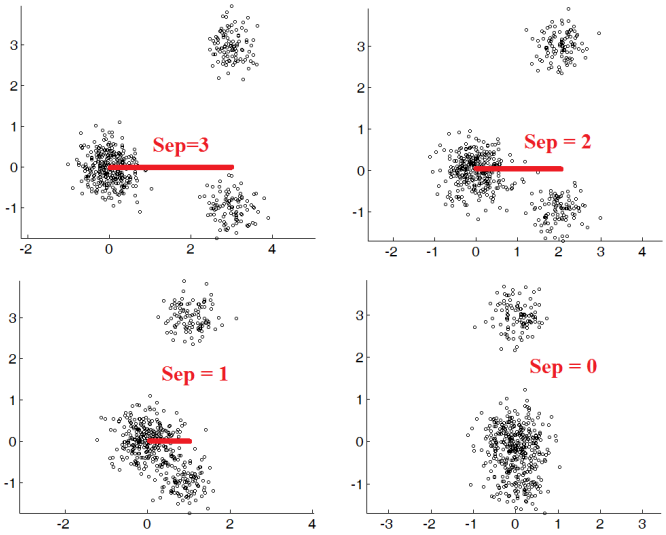
Essentially, the clustering procedure in Algorithm 6 does not identify an appropriate metric in the solution space. Such a metric is necessary to define the sets of approximate solutions instead of the empirical output. For such cases, we have proposed to use a Hamming metric in the solution space. This selection provides a more flexible weighting scheme to relax the inconsistencies among solutions of different instances from the data source. For demonstration purposes, we consider datasets of tree Gaussian sources with different covariances (Figure 25b). The Hamming-based generalization capacity for different separations of the sources has been shown in Figure 25a. For the well-separated case, Dominant Set clustering computes the maximal $\mathcal{GC}$, i.e. 1.37 bits per object. By increasing the complexity, the capacity decreases. However it is still nonzero even if the sources have overlap (e.g. $Sep. = 1$).

6.3 QUADRATIC CLUSTERING MODELS

Arguably, the most basic graph clustering criterion is Min Cut, which minimizes the sum of the crossing weights. This quadratic model, which splits small sets of objects, is usually introduced as the base



(a) generalization capacity



(b) datasets

Figure 25: The Hamming-based generalization capacity for Dominant Set clustering of three Gaussian sources. $\mathcal{GC}$ is still non-zero even if the sources have some overlap (e.g. $Sep. = 1$).

criterion of more advanced models which normalize the clusters with respect to e.g. the degree or the size of the clusters. A more recent quadratic model, called Correlation Clustering, clusters the graphs with positive and negative edge weights. In this section, we study these models and discuss in detail the influence of shifting the pairwise similarities.

6.3.1 *Correlation Clustering*

Correlation Clustering [BBC04] partitions a graph with positive and negative edge weights. The Correlation Clustering cost function sums the disagreements, i.e. the sum of negative intra-cluster edge weights plus the sum of positive inter-cluster edge weights. Figure 26 demon-

strates the model on a small graph with 14 nodes. A green edge refers to $+1$ and a red edge indicates -1. In many clustering models, such as K-means, the number of clusters computed by the empirical minimizer scales with the number of objects N. However, Correlation Clustering might compute a finite number of clusters even if $N \to \infty$. One example is a graph with totally positive edge weights where Correlation Clustering gives only one single cluster. Thereby, Correlation Clustering is sometimes modeled as an integer program performed over the co-clustering matrix $\mathbf{H}$ [DEFI06], i.e.

$$\underset{\mathbf{H}}{\text{minimize}} \sum_{(i,j)\in\mathcal{E}^{(+)}} X_{ij}(1-H_{ij}) - \sum_{(i,j)\in\mathcal{E}^{(-)}} X_{ij}H_{ij}\,, \tag{136}$$

where $\mathcal{E}^{(+)}$ and $\mathcal{E}^{(-)}$ respectively indicate the sets of edges with positive and negative weights. Please note that $\mathbf{H}$ has to fulfill symmetry and transitivity constraints (see the detail in Chapter 2).

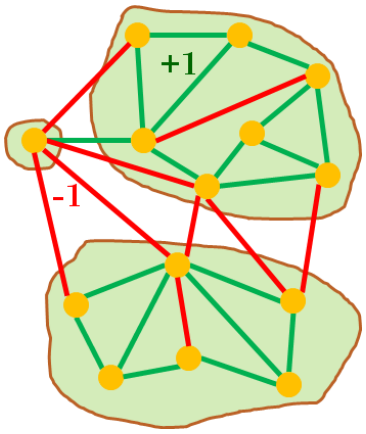


Figure 26: Correlation Clustering acts on graphs with positive (green) and negative (red) edge weights. The criterion computes cuts with minimal sum of negative intra-cluster edge weights plus sum of positive inter-cluster edge weights.

However, unconstrained minimization of the model described in Eq. 136 might lead to generation of small unstable clusters. For example, the appearance of the singleton cluster at the left side of Figure 26 might happen due to randomness in the pairwise measurements. Thereby, in order to better control the model complexity, we redefine Correlation Clustering with an explicitly fixed number of clusters K.

$$\begin{aligned} R^{CC}(c,\mathbf{X}) \;=\; & \frac{1}{2}\sum_{1\leqslant u\leqslant K}\sum_{(i,j)\in\mathcal{E}_{uu}}(|X_{ij}|-X_{ij}) \\ & +\frac{1}{2}\sum_{1\leqslant u\leqslant K}\sum_{1\leqslant v<u}\sum_{(i,j)\in\mathcal{E}_{uv}}(|X_{ij}|+X_{ij}). \end{aligned} \tag{137}$$

MEAN-FIELD APPROXIMATION FOR CORRELATION CLUSTERING. The Correlation Clustering cost function is non-factorial. These dependencies render the computation of weight sums and joint weight sum intractable as it requires summation of the weights over exponentially many solutions [PA88, DTEK06]. We employ mean-field approximation to find the most similar factorial model. For Correlation Clustering, mean-field approximation yields the equations

$$\begin{aligned} h_{ik} &= \frac{1}{2} \sum_{j \leqslant N : j \neq i} (|X_{ij}| + X_{ij})(1 - \mathbb{E}_{Q_{i \to k}}\{\mathbb{I}_{\{c(j)=k\}}\}) \\ &\quad + \frac{1}{2} \sum_{j \leqslant N : j \neq i} (|X_{ij}| - X_{ij}) \mathbb{E}_{Q_{i \to k}}\{\mathbb{I}_{\{c(j)=k\}}\} + \text{constant.} \\ &= \frac{1}{2} \sum_{j \leqslant N : j \neq i} (|X_{ij}| + X_{ij})(1 - q_{jk}) \\ &\quad + \frac{1}{2} \sum_{j \leqslant N : j \neq i} (|X_{ij}| - X_{ij}) q_{jk} + \text{constant.} \end{aligned} \tag{138}$$

The potentials h_{ik} are then used to compute the assignment probabilities q_{ik}. The factorial weights are then given by $\exp(-\beta h_{ik})$.

NUMERICAL STUDY OF $\mathcal{GC}$ FOR CORRELATION CLUSTERING. For the experiments on synthetic data, we generate two correlation graphs $\mathcal{G}(\mathbf{O}^{(1)}, \mathbf{X}^{(1)})$ and $\mathcal{G}(\mathbf{O}^{(2)}, \mathbf{X}^{(2)})$. Given the common noise and complexity parameters η and ξ, both correlation graphs are constructed as follows:

1. First, a perfect graph is constructed. In other words, $+1$ is assigned to intra-cluster edges, while -1 to inter-cluster edges. The perfect graph is planted in both graphs $\mathcal{G}(\mathbf{O}^{(1)}, \mathbf{X}^{(1)})$ and $\mathcal{G}(\mathbf{O}^{(2)}, \mathbf{X}^{(2)})$.

2. Then, each edge in $\mathcal{E}_{uv}, v \neq u$ is flipped to $+1$ with probability ξ. This step aims to increase the complexity of the structure.

3. Finally, each edge ($\mathcal{E}_{uv}, v \neq u$ and $\mathcal{E}_{uu}$) is replaced by a random edge with probability η.

Steps 2. and 3. are performed separately for each graph, which provides independent noise realizations. By construction, each graph consists of $1,500$ nodes and 5 clusters. Identity mapping between objects in $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ is guaranteed by the same order of construction of the graphs. Structure complexity is anchored at $\xi = 0.35$ and noise level η varies from 0.75 to 0.95, thus generating datasets with a broad range of difficulty.

Varying the number of initial clusters from 1 to 10, the mean-field algorithm is executed with 10 random initializations per model order. The best result in terms of cost value is taken at each round, and on

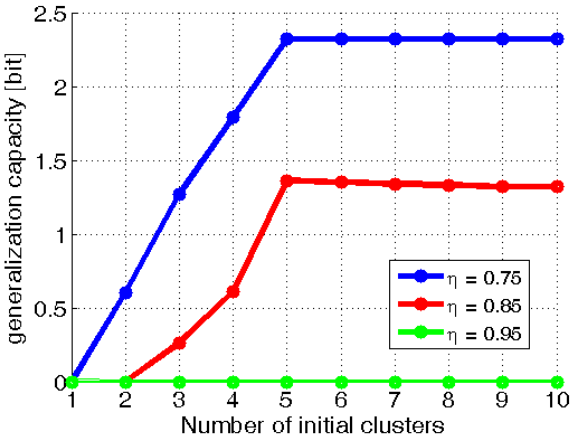


Figure 27: Generalization capacity of Correlation Clustering at three different noise levels η when the complexity parameter ξ is fixed at 0.35. $\mathcal{GC}$ computes the optimal number of clusters correctly for each experiment setting.

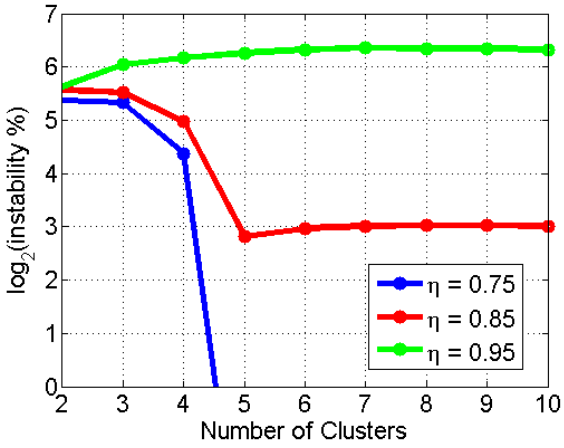


Figure 28: Instability measure in three different settings of Correlation Clustering for $\xi = 0.35$. For $\eta = 0.75$ instability is always zero for 5 and more clusters.

the basis of that $\hat{\mathcal{I}}_\beta$ is maximized over different values of β. Figure 27 illustrates the results of this procedure. The results of mean-field approximation are verified by checking the consistency with Gibbs sampling.

The data analysis problem is easy for $\eta = 0.75$. In this regime, Gibbs samplers or mean-field annealing procedures select the correct number of clusters even when initialized with a large number of clusters. In fact, superfluous clusters are simply left empty as the cost function prefers large clusters for low noise levels. The effective number of clusters remains 5 regardless of the initialization and, hence, the generalization capacity is invariant.

At $\eta = 0.85$ the problem is rather complicated due to noise but still learnable. In this regime, substantial variations are exhibited in the inferred clustering for different choices of the number of clusters. Generalization capacity systematically selects the correct number of clusters. For larger numbers of clusters, the effective number is still 5 and the capacity reduces slightly due to degeneracy. Both for $\eta = 0.75$ and $\eta = 0.85$, the instability measure is consistent with the $\mathcal{GC}$ principle (see Figure 28).

At $\eta = 0.95$ the edge labels are almost entirely random, obfuscating all structure in the data. Therefore, as shown in Figure 27, the number of learnable clusters is just 1. In this regime, instability cannot be used to determine the number of clusters as it remains undefined for $K = 1$.

In the figures, the generalization capacity principle is compared with the instability measure proposed in [LBRB04]. Please note that BIC is not applicable to compute the optimal number of clusters for Correlation Clustering as the effective dimensionality is unknown for this model.

SHIFTED CORRELATION CLUSTERING. In many applications, the pairwise measurements are either all positive or only few measurements are negative. Thereby, Correlation Clustering computes only few clusters and the superfluous clusters are left empty. This might mask a finer representation of the structure in the data. Thereby, we augment the cost function by a parameter which shifts the pairwise similarities by a constant. The cost function is then written by

$$\begin{aligned} R^{CC}(c, \mathbf{X}, s) &= \frac{1}{2} \sum_{1 \leqslant u \leqslant K} \sum_{(i,j) \in \mathcal{E}_{uu}} (|X_{ij} + s| - X_{ij} - s) \\ &\quad + \frac{1}{2} \sum_{1 \leqslant u \leqslant K} \sum_{1 \leqslant v < u} \sum_{(i,j) \in \mathcal{E}_{uv}} (|X_{ij} + s| + X_{ij} + s) \end{aligned} \tag{139}$$

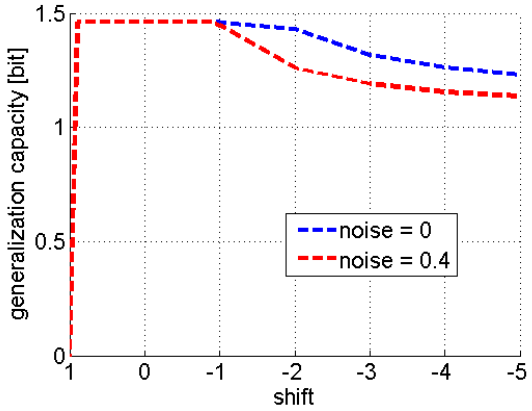
s determines the value of the shift. A negative shift equalizes the size of clusters, since the cost function aims to reduce the negative edge weights which occur inside clusters.

Remember that $\hat{\mathcal{I}}_\gamma(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$ is computed by

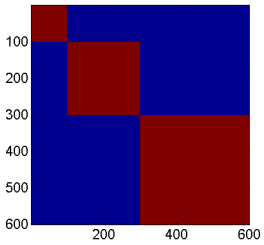
$$\begin{aligned} \hat{\mathcal{I}}_\gamma(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) &= H(\mathcal{A}_K) \\ &\quad + \frac{1}{N} \log \sum_{c \in \mathcal{C}} P_\gamma(c, \mathbf{X}^{(1)}) \times P_\gamma(c, \mathbf{X}^{(2)}). \end{aligned} \tag{140}$$

Shifting the pairwise similarities by a negative value renders a bias towards more balanced clusters, which results in an increment of the informativeness term $H(\mathcal{A}_K)$. Thereby, a question might ask the influence of a very large negative shift on the capacity of inherently

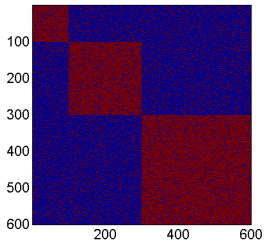
unbalanced clusters: Is it possible to improve the capacity artificially by performing such an unnecessary negative shift such that $H(\mathcal{A}_K)$ increases? Our answer is *no*! Applying an unnecessary shift *increases the entropy of the cluster types*, but at the same time, *decreases the stability term* $\frac{1}{N}\log\sum_{c\in\mathcal{C}} P_\gamma(c,\mathbf{X}^{(1)}) \times P_\gamma(c,\mathbf{X}^{(2)})$, since the shift prevents the algorithm to compute the true underlying structures.



(a) changes in $\mathcal{GC}$ when shifting the pairwise similarities



(b) data when $noise = 0$



(c) data when $noise = 0.4$

Figure 29: Evolution of the generalization capacity when shifting the pairwise similarities. The graph by construction is perfect for Correlation Clustering, therefore, no extra shift is necessary. A positive shift freezes the structures, thereby, the capacity drops down to zero. A large negative shift renders Correlation Clustering to balance the clusters, which yields a reduction in the capacity although the entropy improves. Figures 29b and 29c show the datasets for the noise free and noisy cases. A red edge weight refers to $+1$ and a blue one represents -1.

In Figure 29, we experimentally investigate such a scenario. The data source contains three clusters with 100, 200, and 300 objects. The graph is perfect by construction, i.e. we assign $+1$ to intra-cluster edges and -1 to inter-cluster edges. We study two cases: i) when no noise is applied to the pairwise similarities (Figure 29b), and ii) when the pairwise similarities are replaced by a random value from $\{-1,+1\}$ with probability 0.4 (Figure 29c). A red edge weight refers to $+1$ and a blue one represents -1. Figure 29a illustrates the changes of the generalization capacity for different shifts. A large positive shift

($\geqslant 1$) freezes the structure and thereby the capacity drops down to zero. For the shifts in the range $(-1, +1)$ the underlying structure does not change, thus Correlation Clustering computes the optimal clustering for which the capacity is 1.46 bits. A lager (unnecessary) negative shift renders the method to extract more balanced clusters, which decreases the capacity despite improving the entropy of the clusters. The improvement in the entropy is compensated by a reduction in stability.

6.3.2 *Min Cut, Max Cor and Correlation Clustering*

Traditionally, graph clustering methods seek for a partitioning with minimal similarity between the clusters. Thereby, given a pairwise similarity matrix $\mathbf{X}$, the *cut* of the graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$ into for example two clusters $\mathbf{O}_a$ and $\mathbf{O}_b$ is given by [SM00]:

$$\begin{aligned} \text{cut}(\mathbf{O}_a, \mathbf{O}_b) &= \sum_{i \in \mathbf{O}_a} \sum_{j \in \mathbf{O}_b} X_{ij} \\ &= \text{links}(\mathbf{O}_a, \mathbf{O}_b). \end{aligned} \tag{141}$$

The goal is to find a clustering with minimum cut among an exponential number of alternative splits. However, computing the minimum cut of non-negative graphs (i.e. $X_{ij} \geqslant 0, \forall i, j$) is a well-studied problem for which there exist efficient *polynomial time* algorithms (e.g. $O(N^4)$ [GH88] and $O(N^2 \log^3 N)$ [KS96]). The Min Cut criterion has been used for image segmentation by bisecting the existing segments recursively until K clusters are formed [WL93]. This method works well for only *some* images.

In fact, the criterion defined in Eq. 141 has a tendency to split small sets of objects. The cut increases with the number inter-cluster edge weights, i.e. the edges connecting the two clusters $\mathbf{O}_a$ and $\mathbf{O}_b$. Figure 30 illustrates such a situation. We assume that the edge weights are inversely proportional to the distances between the objects. It is observed that Min Cut favors splitting objects i or j, rather than performing a more balanced partitioning. In fact, any cut that splits one of the objects on the right half will have a smaller cost than the cut that partitions the objects into the left and right halves [SM00].

More elegant clustering models propose normalizing the subsets $\mathbf{O}_a$ and $\mathbf{O}_b$ in order to avoid such a bias towards splitting singleton objects. For example, Normalized Cut proposes to normalize each cluster by the degree of the cluster members.

The Min Cut criterion can be extended to K-way clustering of a graph by

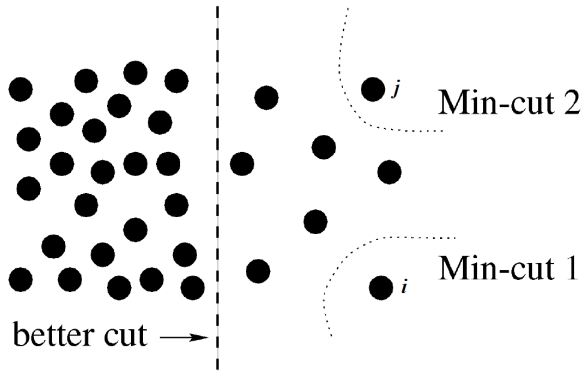


Figure 30: The Min Cut criteria has a tendency to split small (singleton) sets of objects. Any cut that splits one of the objects on the right half will have smaller cost than the cut that partitions the objects into the left and right halves. This issue is particularly problematic when the intra-cluster edge weights are heterogenous among different clusters. The picture has been adapted from [SMoo].

$$\begin{aligned} R^{MinCut}(c, \mathbf{X}) &= \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} X_{ij} \\ &= \sum_{k=1}^{K} \sum_{k'=k+1}^{K} links(\mathbf{O}_k, \mathbf{O}_{k'}). \end{aligned} \tag{142}$$

A similar criterion, the so-called *Max Cor*, would seek for maximizing the intra-cluster similarities. The corresponding cost function is defined as

$$\begin{aligned} R^{MaxCor}(c, \mathbf{X}) &= -\sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij} \\ &= -\sum_{k=1}^{K} links(\mathbf{O}_k, \mathbf{O}_k). \end{aligned} \tag{143}$$

The Min Cut cost function can be rewritten as

$$\begin{aligned} R^{MinCut}(c, \mathbf{X}) &= \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} X_{ij} \\ &= \underbrace{\sum_{k=1}^{K} \sum_{k'=k} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} X_{ij}}_{\text{constant}} - \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij} \\ &= \text{constant} - \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij} \\ &= \text{constant} + R^{MaxCor}(c, \mathbf{X}). \end{aligned} \tag{144}$$

Thereby, Min Cut and Max Cor differ only by a constant. The constant term includes the sum of all edge weights in the graph. Thus, minimizing Min Cut equivalently minimizes the Max Cor cost function and vice versa.

$$\boxed{R^{MinCut}(c, \mathbf{X}) \equiv R^{MaxCor}(c, \mathbf{X}).}$$

The equivalence of $R^{MinCut}(c, \mathbf{X})$ and $R^{MaxCor}(c, \mathbf{X})$ is determined by:

1. The cost functions share the same empirical minimizer, i.e. $\min_c R^{MinCut}(c, \mathbf{X}) = \min_c R^{MaxCor}(c, \mathbf{X})$.
2. $\forall c \in \mathcal{C} : R^{MinCut}(c, \mathbf{X}) - R^{MinCut}(c^{\perp}, \mathbf{X}) = R^{MaxCor}(c, \mathbf{X}) - R^{MaxCor}(c^{\perp}, \mathbf{X})$.

The other *quadratic* cost function is Correlation Clustering. The cost function, invented for partitioning graphs with positive and negative edge wights, can be written as

$$\begin{aligned}
R^{CC}(c, \mathbf{X}) =& \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (|X_{ij}| - X_{ij}) + \frac{1}{2} \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} (|X_{ij}| + X_{ij}) \\
=& \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (|X_{ij}| - X_{ij}) \\
& + \frac{1}{2} \sum_{k=1}^{K} \sum_{k'=k}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} (|X_{ij}| + X_{ij}) - \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (|X_{ij}| + X_{ij}) \\
=& \underbrace{\frac{1}{2} \sum_{k=1}^{K} \sum_{k'=k}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} (|X_{ij}| + X_{ij})}_{\text{constant}} \\
& + \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (|X_{ij}| - X_{ij} - |X_{ij}| - X_{ij}) \\
=& \text{constant} - \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij}. \\
=& \text{constant} + R^{MaxCor}(c, \mathbf{X}). \qquad (145)
\end{aligned}$$

Thereby, Correlation Clustering differs from Max Cor only by a constant shift. Eq. 145 implies the equivalence of Max Cor, Min Cut, and Correlation Clustering.

$$\boxed{R^{MinCut}(c, \mathbf{X}) \equiv R^{MaxCor}(c, \mathbf{X}) \equiv R^{CC}(c, \mathbf{X}).}$$

The equivalence holds for general graphs with arbitrary positive and negative edge weights where all the three models are $\mathcal{NP}$-hard. Restricting the edge weights to be non-negative, i.e. $\forall i, j \in \mathbf{O} : X_{ij} \geqslant 0$, makes the minimization of the cost functions solvable in polynomial time.

6.3.3 *Min Cut vs. Max Cut*

Usually the goal of clustering is to partition a graph into groups with maximal intra-cluster similarities, while the inter-cluster similarities are minimal. Thereby, for instance, Min Cut aims at computing cuts with minimal crossing weights. However, in some applications it is intended to *maximize* the crossing weights rather than minimizing, i.e.

$$R^{MaxCut}(c, \mathbf{X}) \quad = \quad - \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} X_{ij} \,. \tag{146}$$

For example, consider the problem of finding the customers or the sellers of a specific product like digital media in a graph of social interactions. The customers might not interact often with each other, but more possible to interact with several sellers. The same can happen to the sellers. Therefore, the strategy would be to compute a maximal cut.

However, there might exist several types of structures in a graph. Therefore, different models might extract the *same* amount of information which corresponds to *different* types of structures. To illustrate such a case, we perform an experimental study on a graph constructed as follows. Given the noise parameter η:

1. A perfect graph with 4 components and each with 100 objects is constructed ($N = 400$). The objects are assigned to the components in a chronological order, i.e.

$$\mathbf{O}_k = \{i : i \in \mathbf{O} \wedge 100(k-1) + 1 \leqslant i \leqslant 100k\} \,. \tag{147}$$

2. The edge weights of the perfect graph are selected according to the following rule:

$$X_{ij} = \begin{cases} +2 \text{ (dark red)} & \text{if } (i \in \mathbf{O}_1 \wedge j \in \mathbf{O}_2) \bigvee (i \in \mathbf{O}_2 \wedge j \in \mathbf{O}_1) \\ & \bigvee (i \in \mathbf{O}_3 \wedge j \in \mathbf{O}_4) \bigvee (i \in \mathbf{O}_4 \wedge j \in \mathbf{O}_3) \\ +1 \text{ (orange)} & \text{if } i, j \in \mathbf{O}_k \ \forall k \\ -1 \text{ (light blue)} & \text{if } (i \in \mathbf{O}_1 \wedge j \in \mathbf{O}_4) \bigvee (i \in \mathbf{O}_4 \wedge j \in \mathbf{O}_1) \\ & \bigvee (i \in \mathbf{O}_2 \wedge j \in \mathbf{O}_3) \bigvee (i \in \mathbf{O}_3 \wedge j \in \mathbf{O}_2) \\ -2 \text{ (dark blue)} & \text{if } (i \in \mathbf{O}_1 \wedge j \in \mathbf{O}_3) \bigvee (i \in \mathbf{O}_3 \wedge j \in \mathbf{O}_1) \\ & \bigvee (i \in \mathbf{O}_2 \wedge j \in \mathbf{O}_4) \bigvee (i \in \mathbf{O}_4 \wedge j \in \mathbf{O}_2) \end{cases} \tag{148}$$

3. Each edge, intra-cluster or inter-cluster, is replaced by a random edge from $\{-2, -1, +1, +2\}$ with probability η.

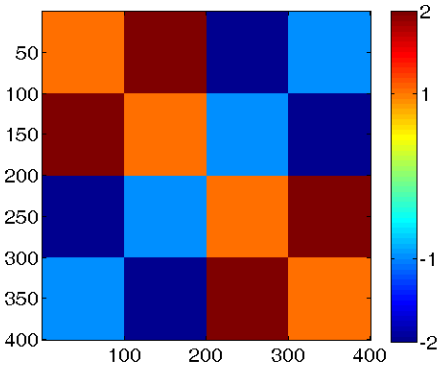
Figure 31a shows an example of a perfect graph (i.e. when $\eta = 0$). This way of constructing the graph provides two types of structures, one appropriate for Min Cut and the other appropriate for Max Cut each with 2 clusters. Figures 31b and 31c show the ground truth co-clustering matrices respectively for Min Cut and Max Cut.

In the following, we investigate the quality of clustering performed by each of the models, while increasing the noise level. Figure 32 illustrates the generalization capacity at different noise levels. At low noise levels both methods produce perfect solutions, thereby the generalization capacity is maximal at 1 bit per object. When we increase the noise level high enough, the generalization capacity drops down since learning becomes difficult. However, Min Cut shows a more robust behavior against this type of noise. Max Cut starts confusing the structure already from $\eta = 0.7$, but Min Cut can still learn the structure and starts confusing from $\eta = 0.9$.

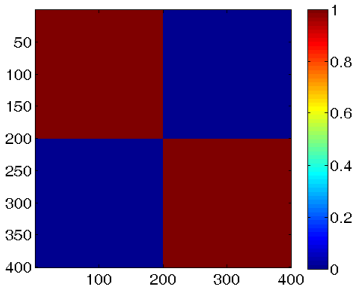
Figure 33 provides an in-depth analysis of the results for $\eta = 0.7$. $\hat{\mathcal{I}}_\beta$ shows the evolution of the information content as the computational temperature changes. Figures 33b and 33d illustrate the optimal clustering computed respectively by Min Cut and Max Cut. Min Cut performs still a perfect clustering while Max Cut confuses some parts of the structure. This explains the reduction in the generalization capacity of Max Cut (Figure 33a vs. Figure 33c).

6.4 MODEL SELECTION: ART OR A SCIENTIFIC APPROACH

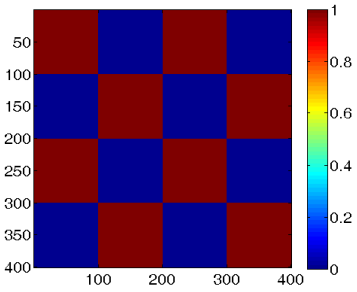
From a modeling point of view, graph clustering methods emphasize on the drawbacks of Min Cut as a base model. Min Cut, as discussed before, yields splitting small sets of singleton objects. Thereby, several clustering methods propose normalizing the clusters by for example the size of clusters (Pairwise Clustering) or the degree of clusters



(a) A perfect graph



(b) The ground truth for Min Cut



(c) The ground truth for Max Cut

Figure 31: A perfect graph constructed according to the weighting scheme in Eq. 148. Figures 31b and 31c show the ground truth co-clustering matrix respectively for Min Cut and Max Cut.

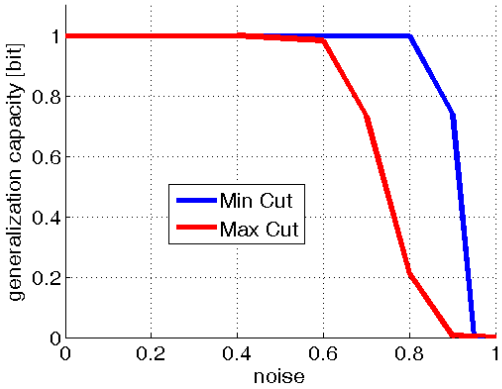
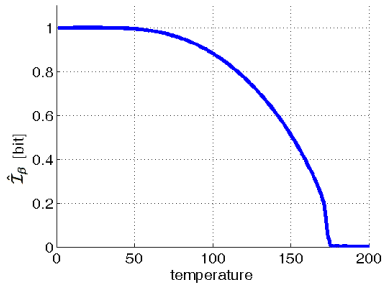


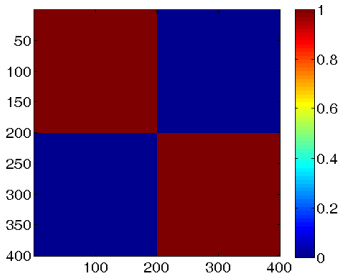
Figure 32: Generalization capacity for Min Cut and Max Cut at different noise levels. For this noise model, Min Cut allows for a more robust clustering than Max Cut.

(Normalized Cut). This design choice, however, introduces a bias towards specific types of structures.

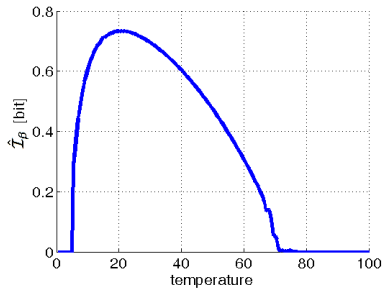
The particular choice of normalization is considered as *art* rather than a scientific approach. These *arbitrary* models might be suitable only for a specific class of problems and then fail to solve the other



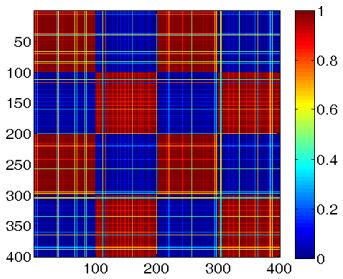
(a) $\hat{\mathcal{I}}_\beta$ for Min Cut



(b) Optimal clustering for Min Cut



(c) $\hat{\mathcal{I}}_\beta$ for Max Cut



(d) Optimal clustering for Max Cut

Figure 33: Comparison of the clustering results computed by Min Cut and Max Cut at the noise level $\eta = 0.7$. Min Cut performs still a perfect clustering (Figure 33b) while Max Cut confuses some parts of the structure (Figure 33d) which yields a reduction in the generalization capacity.

problems. A more sophisticated approach would be developing a framework to compute the optimal model through searching over the space of alternatives, i.e. the *context sensitive adaption* of a prototypical model. This is the ultimate goal of learning: *learning should pursue a scientific approach rather than being art*.

In order to provide a parametric space of alternative models, we augment Min Cut, which is equivalent to Correlation Clustering, with a *shifting parameter* of pairwise similarities. The shift can be negative or positive. However, negative shifts are more interesting as they render the clusters to be more balanced and thereby avoid splitting small singleton sets of objects. More precisely, we introduce shifted Min Cut with shift s as

$$
\begin{aligned}
R^{sMinCut}(c, \mathbf{X}, s) &\equiv R^{sMaxCor}(c, \mathbf{X}, s) \\
&= -\sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (X_{ij} + s) \\
&= -\sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij} - \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} s \\
&= \underbrace{-\sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} X_{ij}}_{\text{Max Cor without shift}} - s \sum_{k=1}^{K} |\mathbf{O}_k|^2 \,. \qquad (149)
\end{aligned}
$$

The first term represents the standard Max Cor (Min Cut) model. The second term, i.e. $-s \sum_{k=1}^{K} |\mathbf{O}_k|^2$, applies a *continuous* constraint on the size of clusters, depending of the value of s.

1. If $s > 0$, then the term $-s \sum_{k=1}^{K} |\mathbf{O}_k|^2$ is minimal when only singleton clusters are separated. Thereby, a positive shift makes the clusters even less balanced.

2. If $s < 0$, then the second term is minimal for balanced clusters, i.e. when $|\mathbf{O}_k| = N/K \,, \forall k \in \{1, ..., K\}$. Thus, a negative shift favors balancing the size of the clusters.

Please note that a negative shift might render the edge weights to be negative, thereby the computational complexity of shifted Min Cut is in general $\mathcal{NP}$-hard.

In our experiments, we use the term *shifted Correlation Clustering*, rather than shifted Min Cut, as Correlation Clustering is a well-known model already developed to partition graphs with negative and positive edge weights, i.e. the case that happens when applying a negative shift. Therefore, we augment Correlation Clustering with a shift parameter to provide a prototypical model. Generalization capacity is then used to compute the optimal model variant, i.e. the optimal shift parameter, in the space of alternative models. This provides a scientific approach to compute the *context sensitive adaptation of the prototypical model* to the *specific application at hand*.

6.5 CONCLUSION

We used the information-theoretic framework to analyze several aspects of well-known graph clustering methods. Computing generalization capacity is more involved for these models. Assignment potentials and resulting weight sums are computed either by embedding into a vector space (e.g. for Pairwise Clustering) or by mean-field approximation (e.g. for Correlation Clustering or Adaptive Ratio Cut).

We particularly investigated the influence of shifting the pairwise similarities on the performance of clustering models. We proposed augmenting the basic Min Cut model by a *shift* parameter, which gives a prototypical model whose optimal parametrization is obtained by scanning the space of alternatives guided by the generalization capacity principle. The *context sensitive* adaptation of the prototypical model advocates a scientific procedure for computing and validating the optimal model adapted to the specific application at hand, rather than an elegant a priori design which might yield a bias towards specific types of patterns. The next chapter will describe in detail the application of such an approach to clustering gene expression data.

7

INFORMATION CONTENT IN CLUSTERING GENE EXPRESSION DATA

We employ the generalization capacity principle to analyze clustering of experimental gene expression data. In the first study, we cluster gene expression profiles from *Mytilus galloprovincialis* female digestive gland [BNM+11], where the temporal logarithmic values of 295 differentially expressed genes are measures for a period of 12 consecutive months. The second application is concerned with clustering gene expression data in the budding yeast *Saccharomyces cerevisiae* [ESBB98], where the expression levels of in total 2,467 genes have been measured in different time points and experimental conditions.

This study provides a comprehensive procedure for ranking and validating different algorithms. We show how to select the number of clusters, how to find the optimal similarity measure and how to evaluate the informativeness of different methods. Figure 34 demonstrates the procedure for model order selection and model validation.

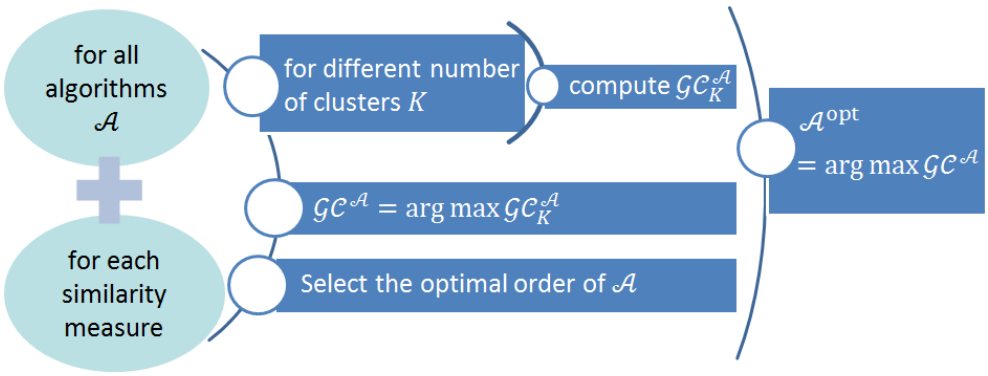


Figure 34: Algorithmic procedure for model order selection and model validation. For a specific model, we use the $\mathcal{GC}$ principle to compute the optimal number of clusters. We then rank the alternative models based on $\mathcal{GC}$ numbers.

In particular, we compare different clustering methods and similarity measures and show how properly shifted Correlation Clustering with an appropriate similarity measure extracts the largest amount of reliable information for all algorithms under consideration.

7.1 CLUSTERING ANALYSIS OF *mytilus galloprovincialis* DATA

In the first experiment, we analyze gene expression profiles from *Mytilus galloprovincialis* female digestive gland [BNM+11], an organism studied to investigate the impact of environmental pollutants.

Mytilus galloprovincialis is an intertidal filter-feeder bivalve whose physiology is affected by seasonal environmental changes. Clustering helps understanding the temporal patterns in gene profiles over the annual cycle, in particular by distinguishing genes associated with thermal variation, food availability, and reproductive functions. The dataset contains expression levels of 295 genes obtained at uniformly spaced time points. The time points correspond to 12 consecutive months, chosen to study how seasonal environmental changes affect physiology across the annual cycle.

7.1.1 *Experiment Setting*

This dataset has been used in Chapter 3 to investigate the Minimum Transfer Costs (MTC) principle. This analysis is set up in a similar way: Taking advantage of the temporal structure of the data, the two object sets $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ are identical and represent the genes. The measurements are then constructed by interleaving the features, i.e. the measurements for the months (Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec) are separated into the values for (Mar, May, Jul, Sep, Nov) and for (Apr, Jun, Aug, Oct, Dec). This interleaved separation captures the statistical dependence of the samples and avoids the risk of under-sampling small clusters of high biological importance.

We examine two different clustering methods combined with two different similarity measures:

Clustering methods

1. Correlation Clustering (CC)
2. Pairwise Clustering (PC)

Similarity measures

1. Pearson correlation (cor)
2. path-based measure (path)

The pairwise measurements (correlation-based or path-based) are calculated for each pair of genes in each set to construct the similarity matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$.

7.1.2 *$\mathcal{GC}$ Analysis of Clustering Methods*

For each method and for each similarity measure, we compute the capacity for a different number of clusters. Figures 35 and 36 illustrate the generalization capacity for the two models, respectively, for correlation-based and path-based measures. Correlation Clustering has been optimized over a range of *shift* parameter. More detail will be provided later. Please note the Pairwise Clustering is *shift invariant*. Comparing the capacities shows the advantage of shifted Correlation

Clustering over Pairwise Clustering. This result means that under identical noise effects, Correlation Clustering is able to discover more structure from the data than Pairwise Clustering.

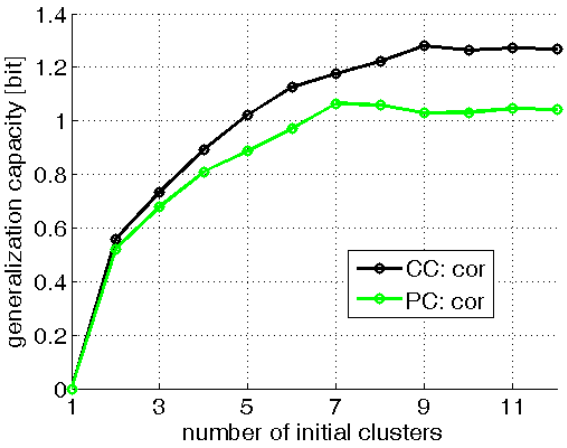


Figure 35: Generalization capacity for the two clustering models with correlation-based measures applied to *Mytilus galloprovincialis* gene expression data. In this dataset, $\mathcal{GC}$ validates shifted Correlation Clustering ($\max_K \mathcal{GC}^{CC} = 1.28$ for $K = 9$) 0.22 bits more informative than Pairwise Clustering ($\max_K \mathcal{GC}^{PC} = 1.06$ for $K = 7$).

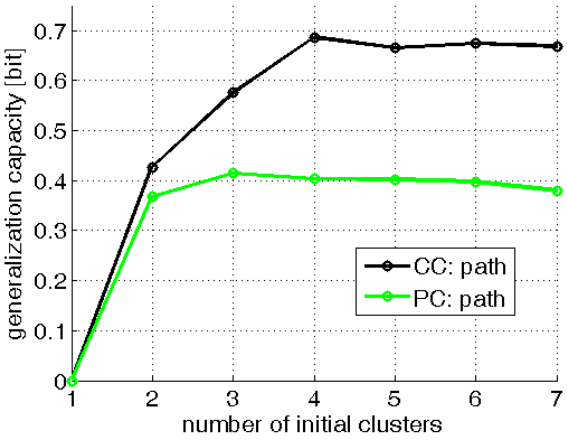


Figure 36: Generalization capacity for the two clustering models with path-based measures applied to *Mytilus galloprovincialis* gene expression data. $\mathcal{GC}$ validates shifted Correlation Clustering ($\max_K \mathcal{GC}^{CC} = 0.69$ for $K = 4$) more informative than Pairwise Clustering ($\max_K \mathcal{GC}^{PC} = 0.41$ for $K = 3$).

For the correlation-based similarity measure, $\mathcal{GC}$ validates shifted Correlation Clustering ($\max_K \mathcal{GC}^{CC} = 1.28$ for $K = 9$) 0.22 bits more informative than Pairwise Clustering ($\max_K \mathcal{GC}^{PC} = 1.06$ for $K = 7$). For the path-based measure, $\mathcal{GC}$ validates shifted Correlation Clustering ($\max_K \mathcal{GC}^{CC} = 0.69$ for $K = 4$) again more informative than

Pairwise Clustering ($\max_K \mathcal{GC}^{PC} = 0.41$ for $K = 3$). For this dataset, a consistent ranking 1. correlation coefficients, and 2. path-based measures is observed for CC and PC clusterings.

ACCURACY OF MEAN-FIELD APPROXIMATION. We have utilized the *mean-field approximation* technique to overcome the computational limitations when computing the individual and joint weight sums. An important question concerns the quality of the approximation. The accuracy of this technique has already been shown in detail by comparing the results with Markov Chain Monte Carlo (MCMC) simulations for different clustering models [HB97, HPB98]. In Figure 37, we investigate the accuracy of mean-field annealing (blue color) by performing Gibbs sampling (red color) for the optimal setting, i.e. for correlation measures and for $\mathtt{shift} = -4$ and $K = 9$. Mean-field annealing and Gibbs sampling yield consistent results for the generalization capacity.

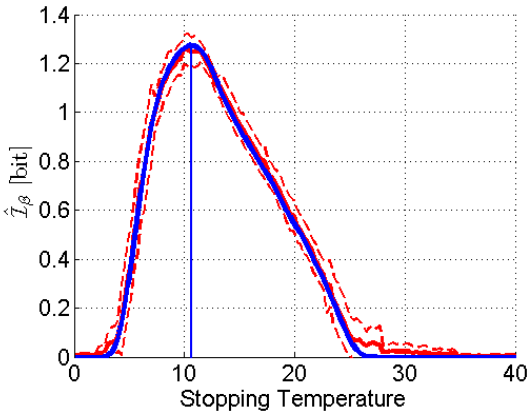
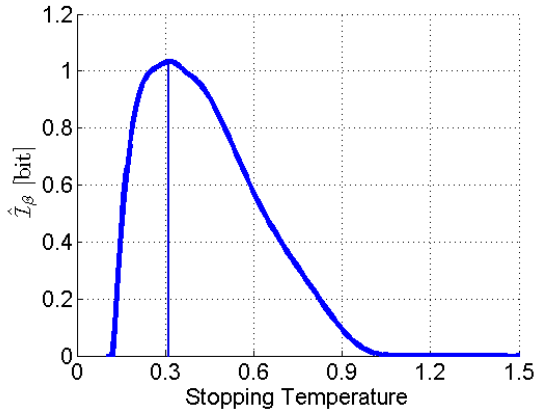


Figure 37: Evolution of $\hat{\mathcal{I}}_\beta$ for CC with correlation measures, for $\mathtt{shift} = -4$ and $K = 9$ as the computational temperature β^{-1} varies. The generalization capacity is computed by mean-field approximation (blue color) and by Gibbs sampling (red color). Both techniques yield consistent results.

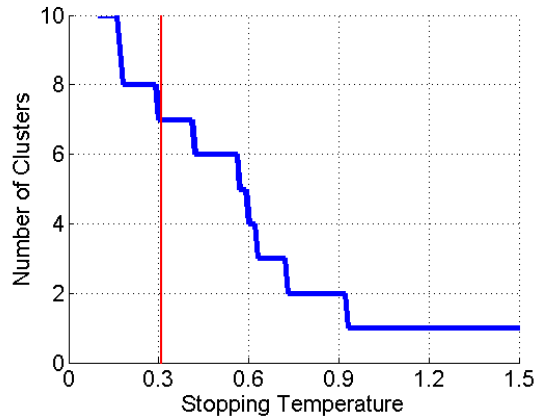
EVOLUTION OF THE CLUSTERS DURING ANNEALING. When computing the generalization capacity for a different number of initial clusters, as we already have seen before, $\mathcal{GC}$ saturates at the optimal number of clusters and stays almost invariant with some small fluctuations if the number of initial clusters increases. Thereby, one can fix the number of initial clusters sufficiently large and then compute $\hat{\mathcal{I}}_\beta$ at different temperatures. At the optimal temperature, which corresponds to $\mathcal{GC}$, the optimal number of clusters is attained.

We investigate the case when Pairwise Clustering with the correlation measure is initialized with 10 clusters. Figure 38 illustrates the evolution of $\hat{\mathcal{I}}_\beta$ as well as the PC clusters as we decrease the computa-

tional temperature β^{-1} from a high temperature to zero. Figure 38b shows the number of clusters generated at different temperatures. At the optimal temperature, i.e. $\beta^{-1} = 0.31$, seven stable clusters are detected. At lower temperatures, more clusters are generated, but they suffer from a lack of stability. This causes a reduction in $\hat{\mathcal{I}}_\beta$.



(a) Evolution of $\hat{\mathcal{I}}_\beta$



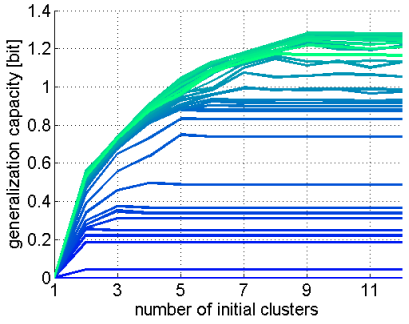
(b) Evolution of the clusters

Figure 38: Evolution of $\hat{\mathcal{I}}_\beta$ and the PC clusters as the computational temperature decreases. At the optimal temperature, i.e. $\beta^{-1} = 0.31$, $\mathcal{GC}$ validates 1.03 bits of information per object. Figure 38b illustrates the evolution of the clusters. At the optimal temperature, 7 stable clusters are computed. At lower temperatures, more clusters, but unstable, are generated.

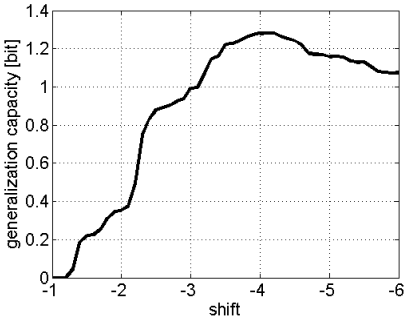
7.1.3 *Shift-Optimized Correlation Clustering*

The maximal number of non-empty clusters produced by Correlation Clustering depends on the proportion of negative and positive edge weights. For example, Correlation Clustering produces only one cluster if all the pairwise similarities are positive. Thereby, for practical purposes, we have proposed a shifted version of Correlation Cluster-

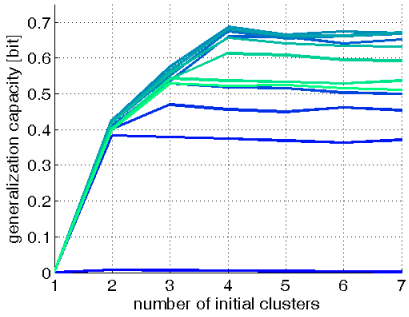
ing, where the pairwise similarities are shifted by a constant. This provides a possibility to perform a *context-sensitive adaptation* of the model. By shifting the similarities by negative values, the clusters start appearing, until, for a very large negative shift, there exist many clusters but some are possibly unstable. Figures 39 illustrates in detail the evolution of the CC clusters as we shift the correlation and path-based similarities. By increasing the negative shift, we obtain more clusters, until, at the optimal shift, the optimal clustering solution is obtained. Then, an even larger negative shift yields reduction in the generalization capacity since the clusters are less stable.



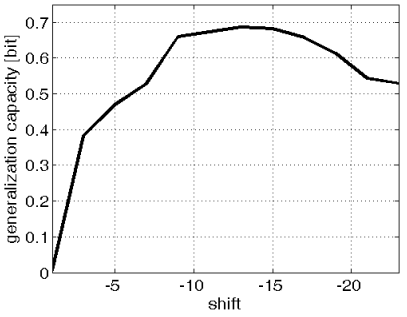
(a) $\mathcal{GC}$ for different shifts



(b) $\mathcal{GC}$ vs. shift for correlation measures



(c) $\mathcal{GC}$ for different shifts



(d) $\mathcal{GC}$ vs. shift for path-based measures

Figure 39: Evolution of the CC clusters while shifting the pairwise similarities: the first row shows the results for correlation-based measure and the second row for path-based measure. $\mathcal{GC}$ computes the *'context sensitive'* adaption of the shift parameter. At the optimal shift, i.e. -4 for correlation-based measure and -13 for path-based measure, $\mathcal{GC}$ computes the highest information rate. Figures 39b and 39d illustrate the maximal capacity as a function of shift.

7.1.4 $\mathcal{GC}$ *and the Other Principles*

For a particular method with a fixed similarity measure, $\mathcal{GC}$ finds the optimal number of clusters. For example, for shifted Correlation Clustering, it computes 9 clusters with correlation-based measures

and 4 clusters with path-based measures. Unlike the other validation principles, $\mathcal{GC}$ does not rely on any specific assumption and can be applied to any arbitrary clustering model or algorithm.

To provide a more detailed analysis, two well-known model order selection criteria are considered for comparison: the BIC score and the instability measure. BIC as a validation principle is difficult to apply in cases where the effective number of free parameters is unclear. Such a situation arises for Pairwise Clustering and it is even less well-defined for Correlation Clustering or Dominant Set clustering. For Pairwise Clustering, we compute the BIC score according to the effective number of dimensions, calculated as the ratio between the trace and the largest eigenvalue [Kiro9], i.e.

$$D_{eff} := \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_{max}} , \tag{150}$$

where λ_i's are the eigenvalues and λ_{max} is the largest eigenvalue.

On the other hand, instability is a heuristic in the spirit of two-instance cross-validation. It is applicable to a wide range of models, but its generality remains confined to alternatives with comparable informativeness.

In Figure 40, we compare $\mathcal{GC}$ with BIC and the instability index for the Pairwise Clustering of correlation-based measures. BIC and instability constitute potentially inconsistent criteria, as they report different numbers. The instability index exhibits by construction a bias that favors a small number of clusters. In particular, this study shows the following properties:

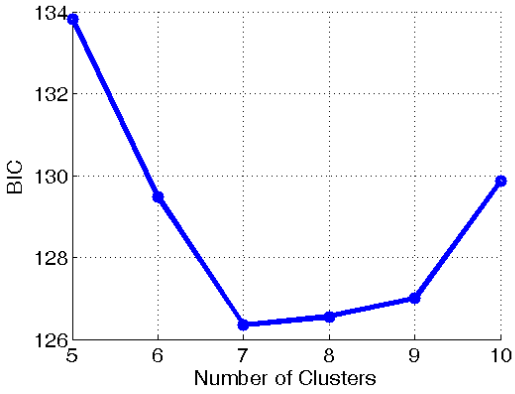
1. the consistency of $\mathcal{GC}$ with BIC (in contrast to the instability index), as both principles compute 7 optimal clusters while the instability index proposes only 2 clusters,
2. greater generality of $\mathcal{GC}$ in comparison to BIC.

7.1.5 *Detailed Analysis of Optimal Clustering Solutions*

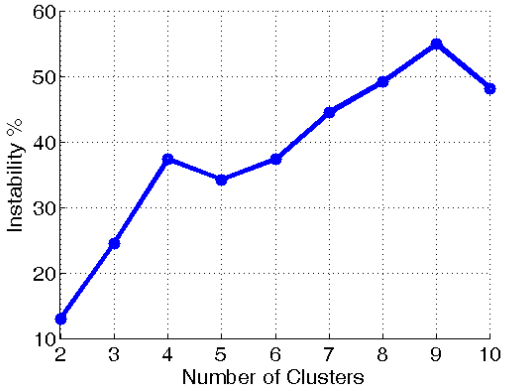
For a fixed similarity measure, $\mathcal{GC}$ validates and ranks the alternative clustering methods. In this application, $\mathcal{GC}$ always first chooses Correlation Clustering as the most informative method. Figure 41 provides an in-depth comparison study of the methods for the correlation similarity measure.

Using the co-clustering matrix $\mathbf{H}$, we compare the optimal clustering solutions computed by different methods. A co-clustering matrix $\mathbf{H}$ is calculated from a clustering solution c such that

$$H_{ij} = 1 \text{ iff } c(i) = c(j), \quad 0 \text{ otherwise.} \tag{151}$$



(a) BIC score



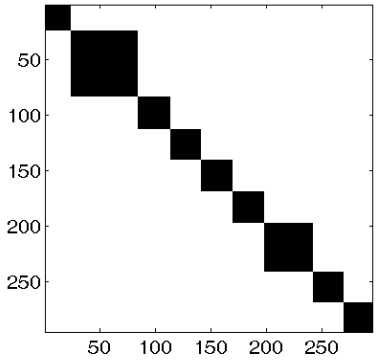
(b) Instability measure

Figure 40: BIC score and instability index computed for the Pairwise Clustering of correlation-based measures. BIC relies upon the calculation of the effective dimensionality. In some cases, such as Pairwise Clustering, the number of free parameters is unclear but heuristics exist. In other cases, such as Correlation Clustering, it is rather problematic. $\mathcal{GC}$ yields a consistent result with BIC in contrast to the instability index.

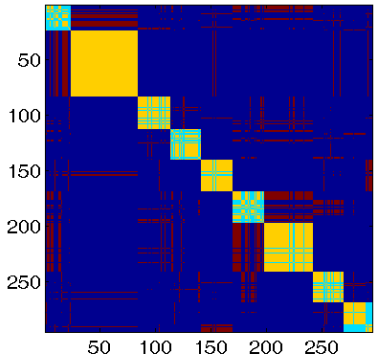
One of the methods is considered as *reference*, with the corresponding co-clustering matrix $\mathbf{H}^{ref}$, and the other as *alternative*, with the corresponding co-clustering matrix $\mathbf{H}^{alt}$. The objects are permuted based on the co-clustering induced by the optimal solution of the reference method, such that the reference clusters appear as diagonal blocks of the co-clustering matrix $\mathbf{H}^{ref}$ (see Figure 41a).

Then, the solution of the alternative method on the permuted objects, e.g. $\mathbf{H}^{alt}$, is compared with $\mathbf{H}^{ref}$. For the pair of objects i and j, the following encoding is used:

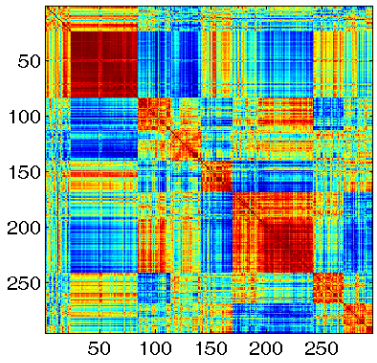
- *'yellow'* if $(H_{ij}^{ref}, H_{ij}^{alt}) = (1,1)$,
- *'red'* if $(H_{ij}^{ref}, H_{ij}^{alt}) = (0,1)$,
- *'light blue'* if $(H_{ij}^{ref}, H_{ij}^{alt}) = (1,0)$, and



(a) Block representation of optimal CC clusters



(b) Correlation Clustering vs. Pairwise Clustering



(c) Correlation matrix

Figure 41: Comparison of optimal Correlation and Pairwise Clustering solutions with correlation measures. The objects are permuted according to the optimal CC solution such that the clusters appear as diagonal blocks (Figure 41a). The optimal CC solution provides a finer, but still reliable representation than the optimal PC solution. Figure 41c shows the pairwise correlations permuted based on the optimal CC ordering.

- *'dark blue'* if $(H_{ij}^{ref}, H_{ij}^{alt}) = (0, 0)$.

In Figure 41, we compare the optimal CC and PC solutions with correlation measures, where the CC solution is used as reference. In this analysis:

1. A high consistency between the clustering results is observed.
2. The superior method, which is used as reference, finds finer representations with more detailed, but still validated structures than the alternative method. This yields a higher generalization capacity. For instance, PC merges the first and the second clusters as well as the sixth and the seventh clusters of the CC solution and thereby reduces the informativeness.

Figure 41c shows the pairwise Pearson correlations between the pairs of genes permuted according to the ordering of Figure 41a.

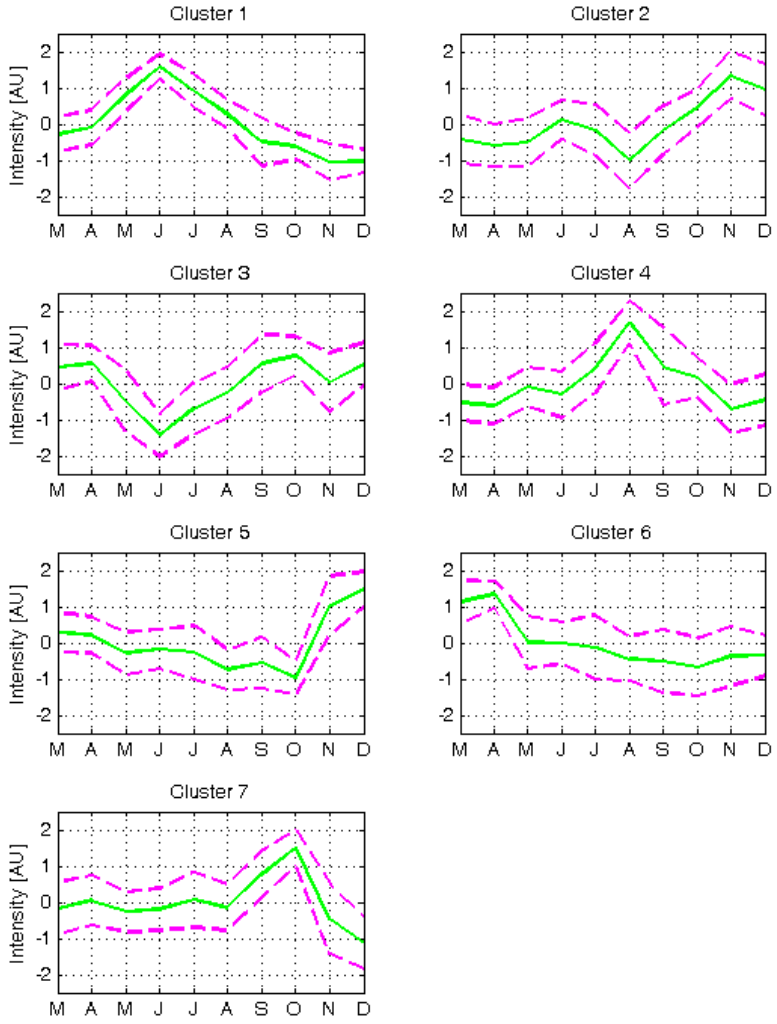


Figure 42: Trajectories of each cluster obtained from the optimal Pairwise Clustering solution with correlation measure. The time frame ranges from March (M) to December (D). Cluster means are plotted in green with normalized standard deviation around.

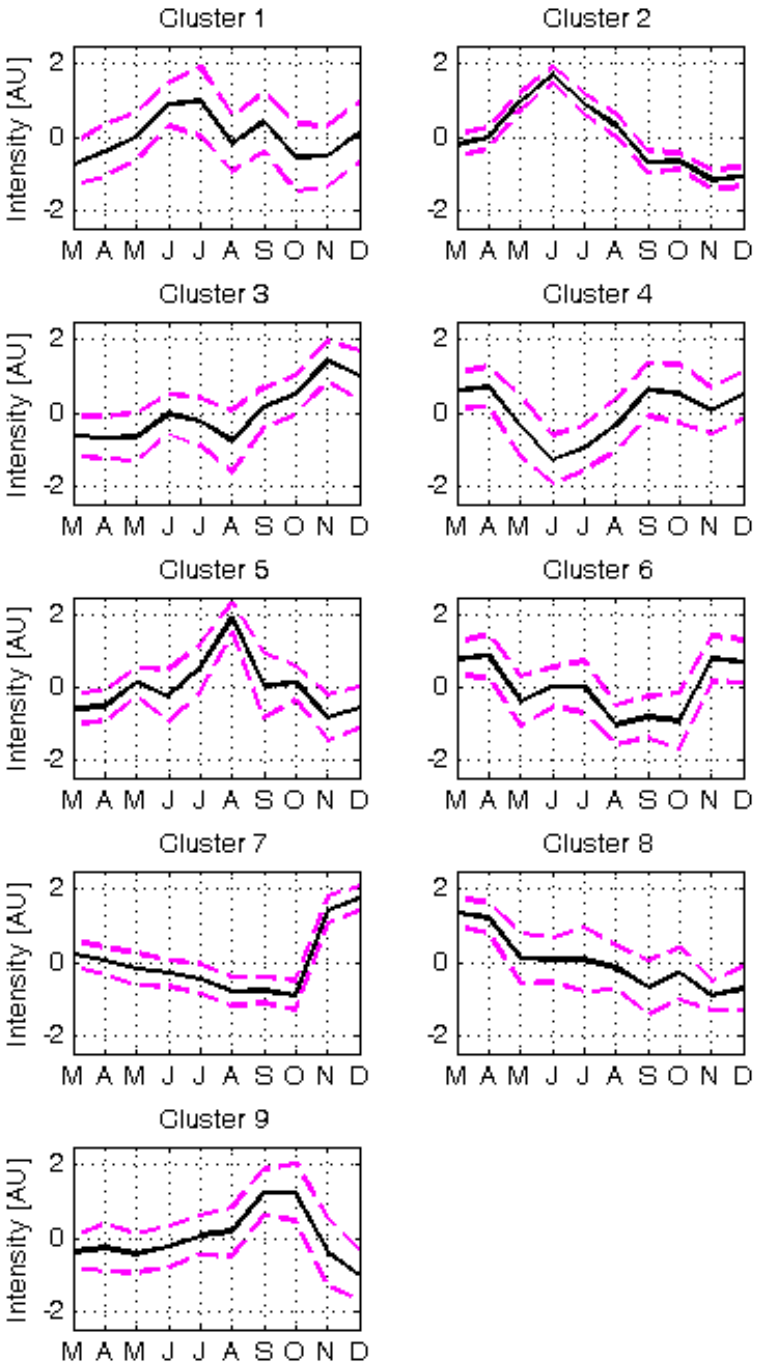


Figure 43: Trajectories of each cluster computed from the optimal Correlation Clustering solution with correlation measure. The time frame ranges from March (M) to December (D). Cluster means are plotted in black with normalized standard deviation at around. The comparison of the trajectories demonstrates the higher resolution of the CC solution compared to the PC solution.

Studying the cluster-specific trajectories provides a detailed analysis of the results. Figures 42 and 43 illustrate the average temporal cluster-specific trajectories. For a better visualization, the trajectory of

each gene has been normalized to have a zero mean and unit variance, so that the gene expression values have the same dynamic range. The trajectories are consistent between the PC and the CC solutions. Table 1 summarizes the correspondences between the cluster indices. However, CC computes a higher resolution for the first and for the fifth PC clusters. Each one is split into two finer clusters by Correlation Clustering. The other trajectories are almost identical. The results are consistent with the analysis performed based on the co-clustering matrices demonstrated in Figure 41.

PC cluster index	1	2	3	4	5	6	7
CC cluster index	1,2	3	4	5	6,7	8	9

Table 1: The correspondence between the optimal PC and CC cluster indices. CC computes a finer resolution for the first and the fifth PC clusters. The other trajectories are almost identical.

7.2 CLUSTERING ANALYSIS OF *saccharomyces cerevisiae* DATA

In the second case study, we analyze gene expression data in the budding yeast *Saccharomyces cerevisiae* [ESBB98]. The expression levels of in total $2,467$ genes have been measured in 79 different time cycles and experimental conditions such as the diauxic shift, the mitotic cell division cycle, sporulation, temperature and so on [ESBB98]. The goal is to find representing groups of genes that share similar expression patterns over multiple conditions. Thereby, clustering constitutes a main biological task.

We, first, describe the experiment settings as well as we present the overview of the $\mathcal{GC}$ analysis of different clustering methods. In the following, we discuss the different aspects of this study. We analyze the influence of shifting the pairwise similarities on evolution of the clusters. For this purpose, we investigate in detail shifted Correlation Clustering and Adaptive Ratio Cut. In a similar way, we investigate the optimal termination of Dominant Set algorithm. This analysis shows how the $\mathcal{GC}$ principle can compute the optimal context-sensitive model in the space of alternatives. We then discuss different model validation aspects, i.e. i) ranking different similarity measures, and ii) validating different clustering methods.

7.2.1 *Experiment Settings*

In this dataset, in contrast to the previous one, the expression levels have not been measured solely at cyclic time points. Thereby, interleaving the features might yield datasets with different underlying structure as the features may differ. Therefore, we randomly split the set of genes and construct $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$. Then, using the Hungarian

method, we align the objects (the genes) of the sets according to a similarity measure. $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are constructed by computing the pairwise similarities in each set.

For this dataset, we examine five clustering methods and three different similarity measures.

Clustering methods

1. Correlation Clustering (CC)
2. Adaptive Ratio Cut (ARC)
3. Pairwise Clustering (PC)
4. Dominant Set algorithm (DS)
5. Normalized Cut (NCut)

Similarity measures

1. Path-based measure (path)
2. Pearson correlation (cor)
3. Euclidean-based measure (euc)

7.2.2 $\mathcal{GC}$ *Analysis of Clustering Methods*

In a similar way to the previous study, for each method and for each similarity measure, we compute the capacity for a different number of clusters from $K = 2$ to $K = 15$. Figures 44, 45 and 46 illustrate the generalization capacity respectively for path-based, correlation-based and Euclidean-based measures. $\mathcal{GC}$ is computed from 25 different random splits of the original data. For a particular method with a fixed similarity measure, $\mathcal{GC}$ finds the optimal number of clusters. For example, it computes 13 clusters for shifted Correlation Clustering with path-based measures.

Computing the capacities, shows the higher information content of shifted Correlation Clustering with path-based measures compared to the other methods. In this dataset, $\mathcal{GC}$ validates shifted Correlation Clustering with path-based measures 0.40 bits per object more informative ($\max_K \mathcal{GC}^{CC} = 2.76 \pm 0.085$ bits for $K = 13$) than Adaptive Ratio Cut with path-based measures ($\max_K \mathcal{GC}^{ARC} = 2.36 \pm 0.071$ bits for $K = 11$). This result convincingly demonstrates that under identical experimental conditions for all algorithms, shifted Correlation Clustering with the path-based similarity measure is able to discover more reliable cluster structure in the data than the other methods.

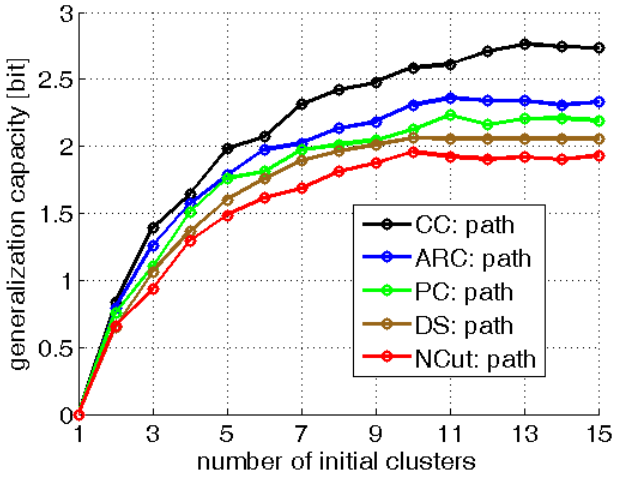


Figure 44: Generalization capacity for different clustering algorithms with path-based measures applied to the second gene expression dataset. Shifted Correlation Clustering obtains the highest capacity for this dataset ($\max_K \mathcal{GC}^{CC}_{path} = 2.76 \pm 0.085$ bits for $K = 13$).

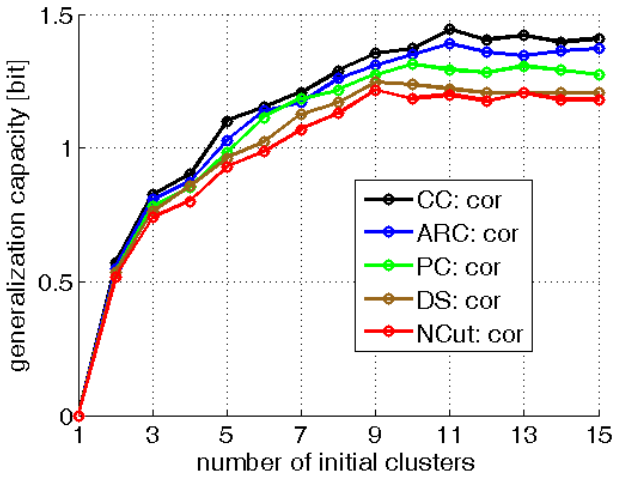


Figure 45: Generalization capacity for different clustering algorithms with correlation-based measures. Shifted Correlation Clustering demonstrates the highest capacity ($\max_K \mathcal{GC}^{CC}_{cor} = 1.44 \pm 0.061$ bits for $K = 11$).

7.2.3 *Gene Expression Clustering by Shifted Correlation Clustering*

In this section we study in detail the evolution of the CC clusters when shifting the pairwise similarities. This analysis proposes a scientific procedure for searching the optimal model in the space of alternatives. The best model, i.e. the optimal value of the shift parameter, is determined by the $\mathcal{GC}$ principle. Figure 47 demonstrates $\mathcal{GC}$ for different number of initial clusters and for different shifts applied to the path-based measures. The range of the depicted shifts is between -90

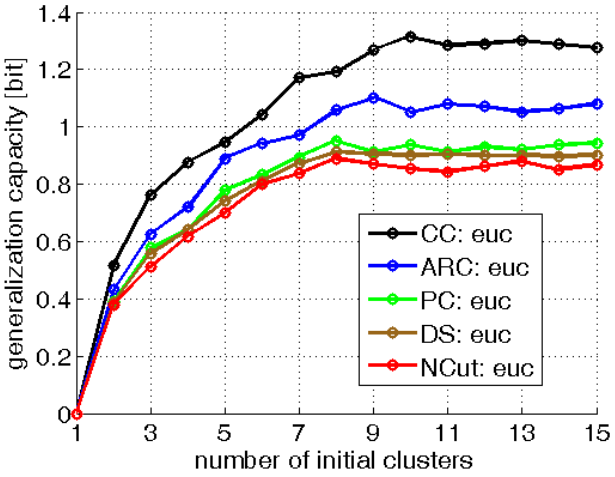
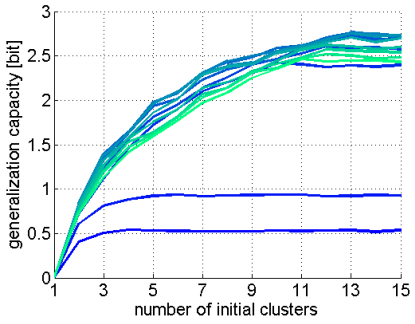
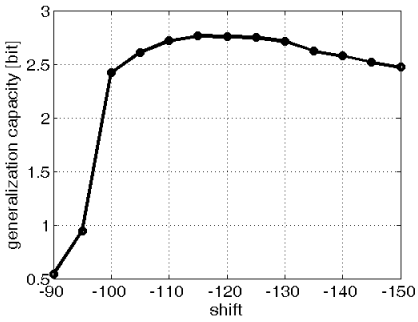


Figure 46: Generalization capacity for different clustering algorithms with Euclidean-based measures. Shifted Correlation Clustering computes the highest capacity ($\max_K \mathcal{GC}^{CC}_{euc} = 1.32 \pm 0.057$ bits for $K = 10$).

(blue plots) and -140 (green plots). The optimal shift is -115, which computes 13 clusters and the corresponding capacity is 2.76 ± 0.085 bits. Figure 47b illustrates the maximum capacity for each shift.



(a) Generalization capacity for different shifts

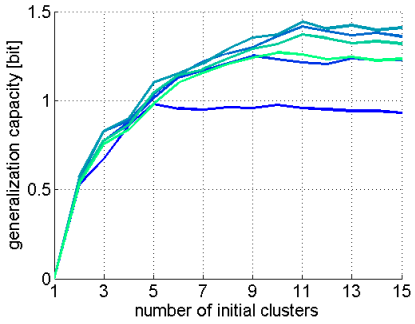


(b) $\mathcal{GC}$ vs. shift

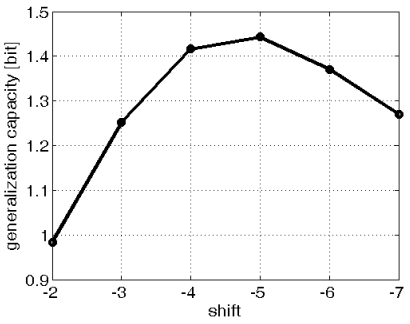
Figure 47: Evolution of the CC clusters while shifting the path-based pairwise similarities. At the optimal shift, i.e. -115, $\mathcal{GC}$ validates 13 clusters with 2.76 ± 0.085 bits of information per object. Figure 47b illustrates the maximal capacity for each shift.

We perform a similar study to correlation-based and Euclidean-based measures. Figure 48 shows the evolution of the clusters (and the capacity) as we shift the correlation-based similarity matrix. The range of the shown shifts is between -2 (blue color) and -7 (green color). The optimal shift is -5, for which $\mathcal{GC}$ computes 11 optimal clusters and 1.44 ± 0.061 bits of information per object. Figure 49 illustrates the $\mathcal{GC}$ results for Correlation Clustering with Euclidean-

based measure. The shifts vary in the range -450 (blue plots) to -600 (green plots). We observe a similar result as before. For this measure, the optimal shift is -525, which computes 10 distinct clusters. The corresponding $\mathcal{GC}$ is 1.32 ± 0.057 bits.

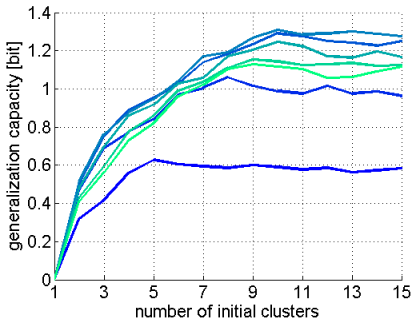


(a) Generalization capacity for different shifts

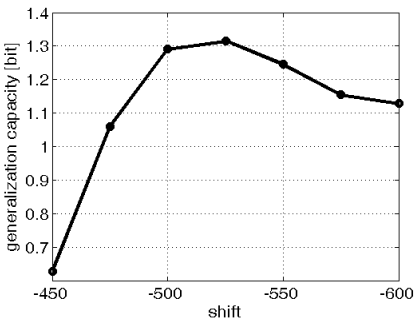


(b) $\mathcal{GC}$ vs. shift

Figure 48: Evolution of the CC clusters while shifting the correlation-based similarity matrix. At the optimal shift, i.e. -5, $\mathcal{GC}$ validates 11 clusters with 1.44 ± 0.061 bits of information per object. Figure 48b illustrates the maximal capacity for each shift.



(a) Generalization capacity for different shifts



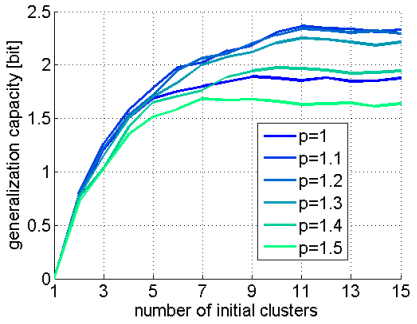
(b) $\mathcal{GC}$ vs. shift

Figure 49: Evolution of the CC clusters while shifting the Euclidean-based measures. At the optimal shift, which is -525, $\mathcal{GC}$ validates 10 clusters with 1.32 ± 0.057 bits of information per object. Figure 49b illustrates the maximal capacity for each shift.

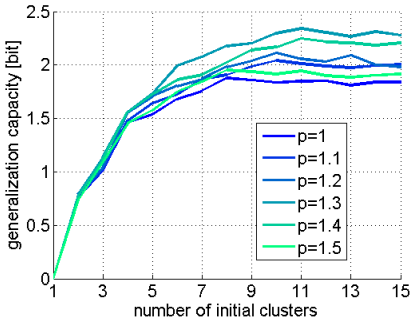
7.2.4 *Gene Expression Clustering by Adaptive Ratio Cut*

We examine in detail the application of Adaptive Ratio Cut to clustering the gene expression data. We investigate the appropriate adaptation of the Laplacian parameter p and study the effect of shifting the pairwise similarities. Essentially, p and shift both control the *shrinkage* or the size of the clusters. Low p equalizes the size of the clusters,

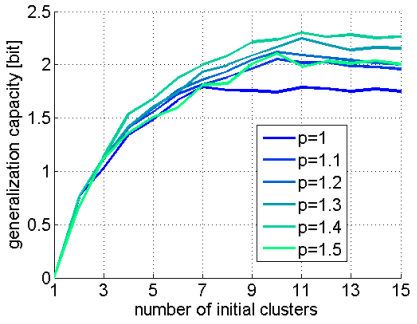
thereby prevents appearance of very extensive (large) and very small (singleton) clusters. Shifting yields the same effect: a large enough positive shift reduces the relative ratios of the pairwise similarities, i.e. makes the ratios of the pairwise similarities closer, therefore yields more balanced clusters.



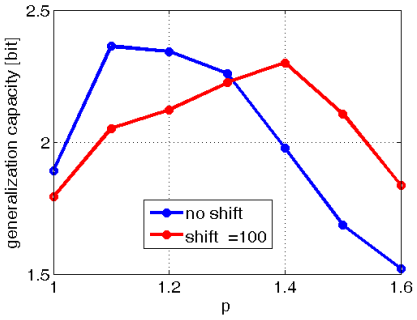
(a) Generalization capacity for different p when *no shift* is applied.



(b) Generalization capacity for different p when $\mathtt{shift} = 50$.



(c) Generalization capacity for different p when $\mathtt{shift} = 100$.



(d) $\mathcal{GC}$ vs. p for *no shift* and for $\mathtt{shift} = 100$.

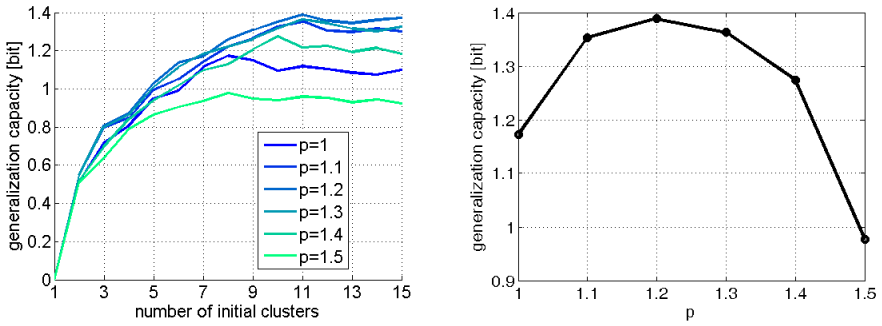
Figure 50: The results of different choices of p when the path-based similarities are shifted by 0 (Figure 50a), by 50 (Figure 50b) and by 100 (Figure 50c). $\mathcal{GC}$ provides a principled way to adapt the optimal p. For example, for the '*no shift*' case, $\mathcal{GC}$ selects $p = 1.1$ and $p = 1.2$ with 11 clusters. Figure 50d shows the generalization capacity as a function of p.

Figures 50 shows the $\mathcal{GC}$ results for three different settings: i) '*no shift*', ii) '$\mathtt{shift} = 50$', and iii) '$\mathtt{shift} = 100$'. It is observed that:

1. For the '*no shift*' case, the optimal clustering is obtained at $p = 1.1$ and $p = 1.2$.

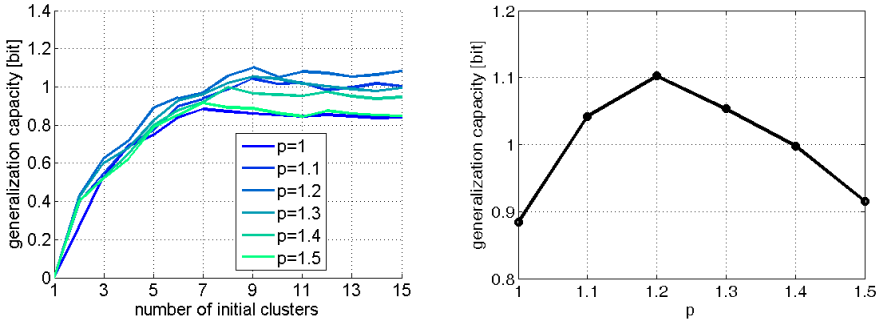
2. By positively shifting the pairwise similarities, the scale (the relative ratios) changes such that the ratios of the pairwise similarities become tighter. This avoids appearance of too big or too small clusters, i.e. equalizes the size of clusters. Thereby, it is effectively equivalent to decreasing p. As a result, the optimal clusters appear at a larger p, i.e. $p = 1.3$ for $\mathtt{shift} = 50$

and $p = 1.4$ for $shift = 100$. Thereby, effectively one can fix one of the parameters (e.g. the shift) and only tune the other parameter (e.g. p).



(a) Generalization capacity for different p. (b) $\mathcal{GC}$ vs. p.

Figure 51: The impact of different choices of the Laplacian p on the generalization capacity of ARC clustering when the correlation-based similarities are used. $\mathcal{GC}$ selects $p = 1.2$ for this measure. The capacity is 1.39 ± 0.054 bits per objects. Figure 51b shows the maximal generalization capacity as a function of p.



(a) Generalization capacity for different p. (b) $\mathcal{GC}$ vs. p.

Figure 52: The impact of different choices of p on generalization capacity when the Euclidean-based similarities are used. $\mathcal{GC}$ selects $p = 1.2$ for this measure. The capacity is 1.10 ± 0.052 bits per objects. Figure 52b shows the maximal generalization capacity as a function of p.

In the following, we study the two other similarity measures, i.e. correlation-based and Euclidean-based measures. Since adapting p and optimizing over shift, both, have a similar effect, therefore, we perform the search only with respect to p. Figures 51 and 52 show the the generalization capacity for different number of clusters and for different p, respectively, for correlation-based and Euclidean-based measures. $\mathcal{GC}$ selects $p = 1.2$ for both similarity measures. For correlation-based measure, $\mathcal{GC}$ validates 11 clusters at the optimal p.

The capacity is 1.39 ± 0.054 bits per objects. For Euclidean-based measure, $\mathcal{GC}$ saturates at 9 clusters with 1.10 ± 0.052 bits of information per object. Figures 51b and 52b illustrate the maximal generalization capacity as a function of p.

7.2.5 *Gene Expression Clustering by Dominant Set Algorithm*

Dominant Set clustering, as it is proposed in Algorithm 6, does not yield an appropriate metric in the solution space. Thereby, we employ the Hamming-based generalization capacity to investigate the information content of the algorithm in this application.

We first study Dominant Set clustering with path-based measures. Figure 53 illustrates the evolution of $\mathcal{GC}$ during execution of the algorithm for different number of initial clusters. The cut-off threshold ν is fixed at three numbers: $\nu = 0.001$ (Figure 53a), $\nu = 0.0001$ (Figure 53b), and $\nu = 0.00001$ (Figure 53c). The blue plots show $\mathcal{GC}$ at the early steps and the green plots correspond to the last steps. For $\nu = 0.001$ and $\nu = 0.0001$ the number of steps varies from 20 (blue) to 1000 (green) and for $\nu = 0.00001$ it varies from 20 to 1200. $\mathcal{GC}$ gives the optimal termination of the algorithm with respect to the context of the data. For instance, $\mathcal{GC}$ computes optimally 2.05 ± 0.08 bits per object for $K = 10$ and $\nu = 0.0001$.

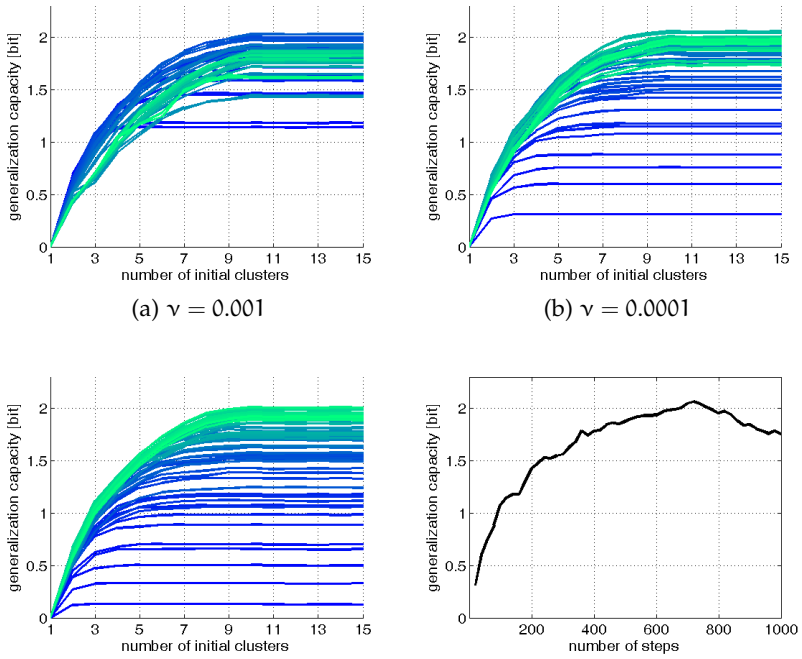
We observe the important roles of the cut-off threshold and the number of steps. In particular, this study shows the following properties:

1. A strong (inverse) dependency between the cut-off threshold ν and the number of steps t_{max} is observed. By setting a small ν we need to run the algorithm longer, and a larger threshold requires less steps.

2. It is not necessary to fix the threshold at very small values (e.g. as proposed in [PP07]), which renders running a very large number of steps if appropriate results are desired. Setting a larger threshold can yield reliably informative solutions, and at the same time, reduces the computational overhead. Generalization capacity provides a framework to analyze algorithms with respect to computational and information complexities and then choose the optimal model variant in terms of *information content* and *computational complexity*.

Figure 53d shows the maximum $\mathcal{GC}$ at different steps for $\nu = 0.0001$. It demonstrates the change of information content during execution.

In the following, we perform a similar study with correlation and Euclidean-based measures. Because of a dependency between ν and t_{max}, we fix ν at 0.0001 and investigate $\mathcal{GC}$ at different steps of the algorithm. Figures 54 and 55 show the evolution of the generalization



(a) $\nu = 0.001$ (b) $\nu = 0.0001$

(c) $\nu = 0.00001$ (d) $\mathcal{GC}$ at different steps for $\nu = 0.0001$

Figure 53: Evolution of $\mathcal{GC}$ during the execution of path-based Dominant Set clustering as the number of steps increases. The cut-off threshold is fixed at three values: $\nu = 0.001$ (Figure 53a), $\nu = 0.0001$ (Figure 53b), and $\nu = 0.00001$ (Figure 53c). The blue plots show $\mathcal{GC}$ at the early steps and the green plots correspond to the last steps. Decreasing ν increases the number of steps the algorithm should take to produce suitable results. Figure 53d shows the maximum $\mathcal{GC}$ at each step for $\nu = 0.0001$.

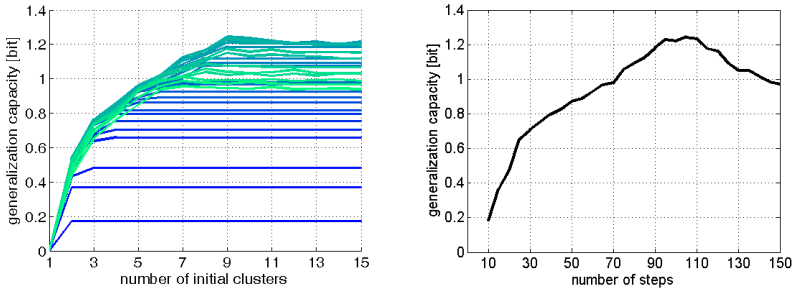
capacity. Figures 54b and 55b illustrate the maximal generalization capacity at different steps, respectively for correlation and Euclidean measures. At the beginning, the information content is low. The optimal clustering results are then obtained at intermediate steps. Running the algorithm even more, decreases the information content.

7.2.6 *Rankings Different Similarity Measures*

For this dataset, $\mathcal{GC}$ always selects the path-based measure as most informative. A consistent ranking of similarity measures is observed for different clustering methods:

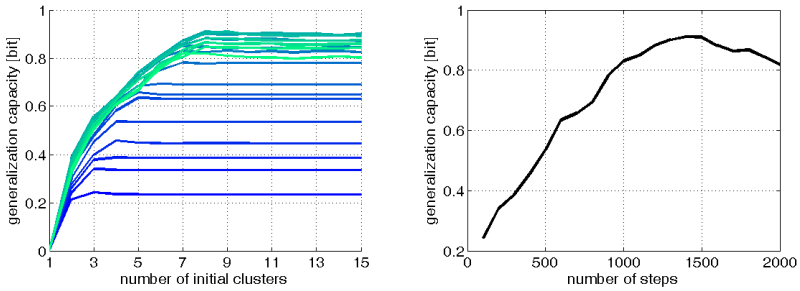
1. path-based measure,
2. correlation coefficients, and
3. Euclidean-based measure.

In particular, $\mathcal{GC}$ validates CC with path-based measure as almost twice as informative ($\max_K \mathcal{GC}^{CC}_{path} = 2.76 \pm 0.085$ bits for $K = 13$) as



(a) correlation measure (b) $\mathcal{GC}$ at different steps

Figure 54: Evolution of $\mathcal{GC}$ during execution of Dominant Set clustering as the number of steps increases. The pairwise similarities are computed according to Pearson correlations and the cut-off parameter ν is fixed at 0.0001.



(a) Euclidean measure (b) $\mathcal{GC}$ at different steps

Figure 55: Evolution of $\mathcal{GC}$ during execution of Dominant Set clustering with Euclidean measures as the number of steps increases. The cut-off parameter ν is fixed at 0.0001.

CC with the correlation-based measure ($\max_K \mathcal{GC}^{CC}_{cor} = 1.44 \pm 0.061$ bits for $K = 11$). Figure 56 compares the corresponding eigenspectra of these two distance matrices, with the second spectrum being rescaled to match the largest eigenvalue. The sharp decay of the eigenvalues for the path-based matrix indicates a tight confinement to a low dimensional subspace, thereby substantially enhancing the cluster structure and increasing generalization capacity.

7.2.7 *Validating Different Clustering Methods*

For a fixed similarity measure, $\mathcal{GC}$ ranks the alternative clustering methods. In this study, $\mathcal{GC}$ always first chooses CC and then selects ARC as the second alternative. Using the co-clustering matrix $\mathbf{H}$, we compare the optimal clustering solutions computed by different methods. In Figure 57, we first compare the optimal CC and PC solutions (CC is used as reference), and then the optimal PC and NCut solutions (PC is used as reference). In this analysis, a high

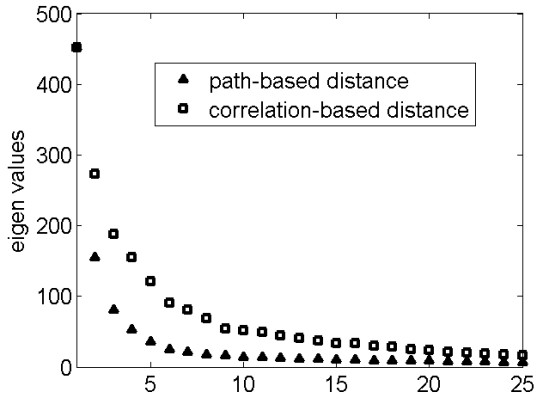


Figure 56: Comparison of the eigenspectra for path-based and correlation-based measures: the more informative measure reveals a sharper eigenspectrum. A consistent ranking of similarity measures is observed for different clustering methods.

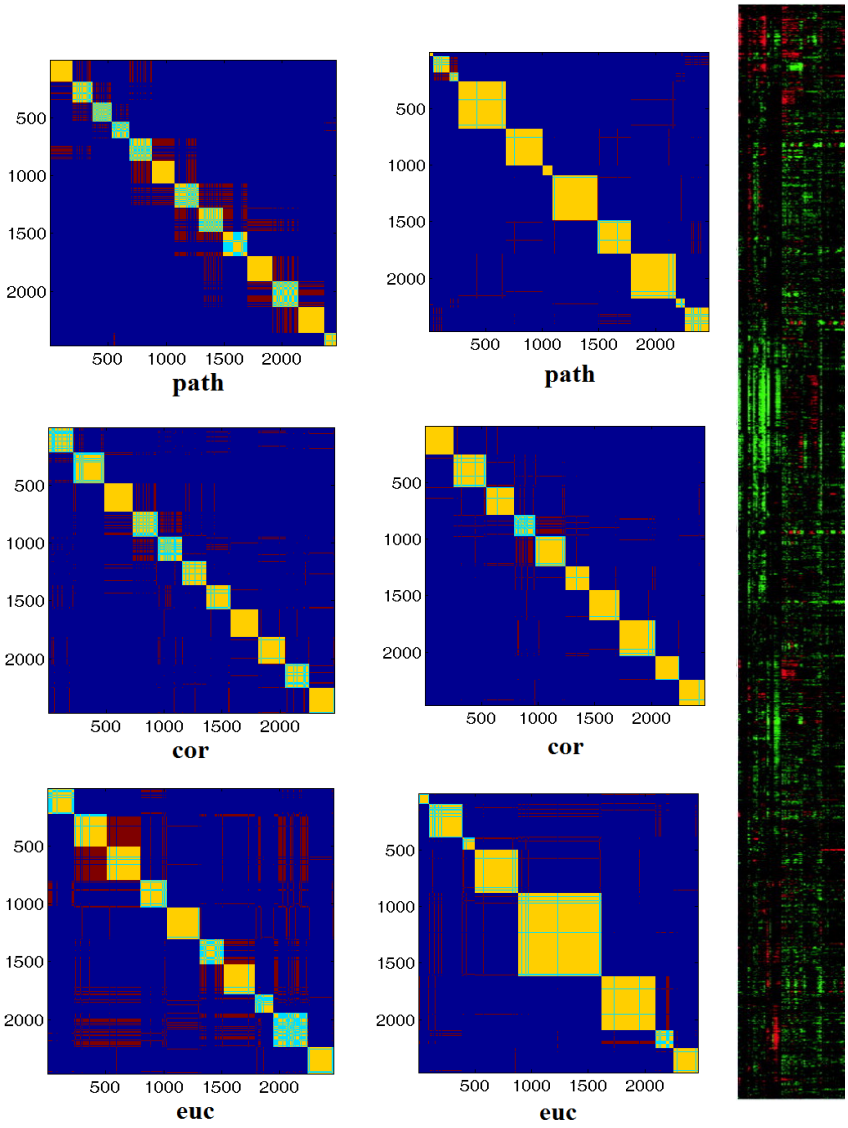
consistency between the clustering results is observed. Similar to the previous study, the superior method extracts the valid structures at a finer resolution compared to the alternative method, which yields an improvement in generalization capacity. For instance, for correlation and path-based measures, for which the number of optimal PC and NCut clusters differ, PC finds finer structures than NCut (Figure 57b). NCut with correlation measure merges the forth and the fifth clusters of the PC solution and thereby reduces the informativeness. Figure 57c shows the original dataset used in this analysis.

Based on the functionalities, we categorize the clustering methods into two classes:

1. Models that partition the *data space* i.e. perform a type of K-means clustering (probably in the kernel space) and partition the data space into equal subspaces. PC and NCut are examples of such methods. A comprehensive list can be found in [DGK07].

2. Models that perform partitioning directly over the *data objects* rather than the data space. CC and ARC belong to this category.

The methods of the second category provide more stable solutions against outliers or singleton clusters. Furthermore, they produce more balanced clusters in terms of the size of the clusters, since they perform partitioning directly on the data objects. This property yields an improved informativeness. The data space partitioning methods are more prone to outliers or to the singleton clusters and they do not necessarily maximize the entropy of the type of clusters. Thus, by a detailed analysis of the results in Figure 57 one observes that:



(a) CC vs. PC (b) PC vs. NCut (c) dataset

Figure 57: Comparison of optimal clustering solutions for different measures: path-based, correlation and Euclidean. In each case, the objects are permuted according to the optimal solution of the reference clustering model. In the color coding, the first bit refers to H_{ij}^{ref} and the second bit corresponds to H_{ij}^{alt}. Optimal clusterings constitute consistent solutions. The consistency is particularly higher for correlation measures. Figure 57c shows the original dataset.

1. The methods from the same category perform consistent clusterings, although the level of the information (e.g. the number of clusters) might be different dependent on the suitability of the method. For example, Figure 57b shows a comparison between optimal PC and NCut clusterings, in order for path-

based, correlation-based and Euclidean-based measures. The PC solutions are used as reference.

2. Depending on the shape and the type of the inherent clusters in the data, a method from the first category might produce either very consistent or partially consistent solutions with a method from the second category. If the inherent structure in the data contains almost balanced clusters, i.e. when *data objects* and *data space* are consistent, then the methods from the two categories produce consistent solutions and as a result, the $\mathcal{GC}$ numbers are very close. This is the case for correlation-based similarity measure as the consistency between CC and PC solutions is higher compared to the other types of measurements. Figure 57a compares optimal CC and PC clusterings (the best of the first and the second categories), in order for path-based, correlation-based and Euclidean-based measures. The CC solutions are used as reference clusterings. For each measure, the solutions are overally consistent between the methods. However, as explained, the consistency is higher for correlation-based similarity measure.

7.3 CONCLUSION

Model order selection and model validation constitute an important aspect in learning. These questions are particularly important in graph clustering, where the effective dimensionality is unknown and thereby the traditional principles such as BIC are not applicable. We employed the generalization capacity principle to study different aspects of clustering gene expression data. This study provides a worked-through procedure for i) finding the optimal number of clusters, ii) ranking different similarity measures, and iii) validating different clustering methods. Moreover, the principle was used to compute the optimal parametrization of a model, e.g. the optimal shift of pairwise similarities in Correlation Clustering. Effectively, the model validation question is addressed in a principled way by searching the optimal adaptation over the (continuous) range of alternatives.

In particular, this study showed the following properties:

1. greater generality of $\mathcal{GC}$ in comparison to classical principles such as BIC,

2. applicability in the biological context of gene expression analysis,

3. better performance of shifted Correlation Clustering (as a base prototypical model) in comparison to more elegant alternative clustering methods in two biological applications,

4. higher importance of selecting an appropriate similarity measure than choosing a superior model.

8

SUMMARY AND OUTLOOK

This thesis aims at providing a principled framework for the analysis and validation of clustering models. Today, there is a zoo of clustering methods with different computational, scalability or performance specifications, which renders choosing the right model very difficult. On the other hand, in real-world applications, the various elegant models might not make a big difference, as it has been shown for example for image segmentation [SS01]. Therefore, we need a principle to rank and validate the alternative models based on a *context sensitive* measure computed according to the ability of the models in extracting reliable patterns from data.

Statistical learning suggests to employ the generalization performance as a measure of model quality. Thereby, at the first attempt, we introduced the Minimum Transfer Costs (MTC) principle, which extends the concept of cross-validation to unsupervised learning: *A good choice of the model variant based on a given dataset should also yield good performance on a second dataset from the same source*. We investigated the principle in several applications such as finding the number of clusters in Gaussian mixture models, Correlation Clustering and Pairwise Clusters, as well as computing the optimal rank of Singular Value Decomposition. Despite the broad applicability, however, the MTC principle does not address all model validation questions. In particular,

1. The applicability of MTC is limited to the models characterized by a cost function.

2. The principle can only be employed for selecting the optimal model order inside a specific family of models. MTC, for example, does not distinguish between Pairwise Clustering and Correlation Clustering, two alternative models which render different bias on the costs.

More generally, model selection criteria indicate a trade-off between two antagonistic goals: *informativeness* and *solution stability*. Maximizing stability without informativeness, e.g. putting every object inside a single cluster, yields an overly simplistic and thereby useless solution. Approximation Set Coding (ASC) [Buh10] is centered around the identification of the optimal trade-offs by establishing a conceptual set-based communication scenario in an information-theoretic approach. We elaborated *Generalization Capacity* ($\mathcal{GC}$), a context-sensitive principle for model validation based on Approximation Set Coding.

We consider algorithms as data processing mechanisms that - during execution - produce a trajectory of weight distributions over the solution space. Generalization capacity computes the optimal concentration of the weights, i.e. the maximal reliable information rate that an algorithm can extract from the data.

We established a principled pipeline to compute generalization capacity for validating different clustering models. Computing the weight sums at each step of the algorithm can be computationally challenging as it might require summation over the entire solution space. Thus,

1. In parametric models, given the cluster parameters, the assignments of objects to clusters are independent, which simplifies calculating the weight sums as they assume a factorial form.

2. In graph clustering models, the assignment of an object to a cluster influences the assignment of other objects. As a result, the weights no longer assume a product form. We utilized efficient approximation schemes such as *mean-field approximation* to compute the closest factorial model in an information-theoretic manner.

3. For the methods that do not yield an appropriate trajectory of weight distributions, e.g. Dominant Set clustering, we proposed to compute the approximate solutions by defining a Hamming metric in the solution space.

Generalization capacity constitutes a provably consistent but more general principle than Shannon capacity. We exemplified the principle first on Gaussian mixture models to analyze the behavior of maximum likelihood inference in different settings:

1. In low dimensional setting when dimensionality $D = 2$. $\mathcal{GC}$ yields consistent results with other principles such as BIC and MTC.

2. In the high dimensional limit with $D \to \infty$ and $\alpha := N/D$ is kept finite. $\mathcal{GC}$ consistently confirms the evolution of several learnability phase transitions detected by order parameters.

We then analyzed several aspects of graph clustering methods, e.g. Pairwise Clustering, Normalized Cut, Ratio Cut, Dominant Set clustering and Correlation Clustering. Particularly, we investigated the evolution of the information content when parameterizing the clustering models. We proposed augmenting the very basic Min Cut model (which is shown to be equivalent to Correlation Clustering), by a *shift* parameter, which provides a prototypical model whose optimal parametrization is obtained through searching over the space of alternatives performed automatically by generalization capacity. The

context sensitive adaptation of the prototypical model advocates a scientific procedure for adapting the optimal model to the specific application at hand, rather than an elegant a priori design which might yield bias towards specific types of patterns, e.g. the normalization of the Min Cut clusters as proposed by Pairwise Clustering or Normalized Cut.

We studied in detail the clustering of two gene expression datasets, one from *Mytilus galloprovincialis* female digestive gland [BNM+11] and the other from the budding yeast *Saccharomyces cerevisiae* [ESBB98]. In particular, we compared different clustering methods and similarity measures and showed how properly shifted Correlation Clustering with an appropriate measure extracts the largest amount of reliable information for all algorithms under consideration. We performed a worked-through procedure to address the fundamental learning questions in the context of clustering gene expression data:

1. **Finding the optimal number of clusters.** Bayesian Information Criterion (BIC) as a validation principle is difficult to apply in cases where the effective number of free parameters is unclear. Such a situation arises for Pairwise Clustering and it is even less well-defined in the case of Correlation Clustering or for algorithms without any log-likelihood formulation (e.g. Dominant Set clustering). $\mathcal{GC}$ does not rely on any specific assumption and can be in principle applied to any arbitrary clustering procedure. For example, it computes 9 clusters for shifted Correlation Clustering with correlation measures in the first biological application and 13 clusters for shifted Correlation Clustering with path-based measures in the second application. On the other hand, instability is a general-purpose heuristic in the spirit of cross-validation, but its generality remains confined due to neglecting informativeness.

2. **Validating alternative clustering methods.** $\mathcal{GC}$ computes the number of stable, task-related bits that an algorithm can optimally extract from the data at hand, which provides a possibility to compare alternative methods w.r.t their respective bit rates. Moreover, $\mathcal{GC}$ allows to compute the optimal context sensitive adaptation of a prototypical model (i.e. shifted Correlation Clustering) which yields superior results compared to more elegant models with arbitrary design choices.

3. **Ranking different similarity measures.** $\mathcal{GC}$ ranks different similarity measures with respect to the nature of the data. In our studies we observed that:

    a) In each biological application, $\mathcal{GC}$ suggests a consistent ranking of similarity measures for different clustering methods, e.g. i) path-based measure, ii) correlation coefficients,

and iii) Euclidean-based measure, for *Saccharomyces cerevisiae* dataset.

b) Choosing an appropriate similarity measure constitutes a more critical task than finding a superior clustering method. Different clustering methods yield (almost) consistent results whereas various similarity measures produce inconsistent solutions, since the representation of patterns changes. Therefore, choosing an appropriate similarity measure plays a more fundamental role, although it suffered from lack of an adequate principle.

This study can be extended in several directions to provide a more insightful analysis of the behavior of data processing mechanisms.

1. **Extension of generalization capacity when several datasets are available.** Approximation Set Coding is established based upon a two-instance scenario. Then, what if we have access to more datasets? How the framework can be adapted to utilize such a possibility?

2. **Analytical connection to the other principles.** For a limited class of clustering models, e.g. statistical models, there exist well-known principles such as BIC. We have studied experimentally the consistency of $\mathcal{GC}$ with such principles. However, an analytical connection between them can be highly informative.

3. **Computational Complexity vs. Statistical Complexity.** Generalization capacity computes the information content at each step of an algorithm as a measure of statistical complexity. A key question concerns the connection of statistical complexity to computational complexity, e.g. what is the minimum computation necessary to attain a reasonable amount of information? Does $\mathcal{NP}$-hardness really matter, as there might exist efficient approximation schemes with the same information content?

4. **Design of energy-saving devices.** The need for design of micro/mobile devices would yield a new generation of energy-saving devices that are prone to unreliable operation execution.

The real power of generalization capacity comes from the fact that it gives the *absolute* reliable information rate that an algorithm can extract from the data at hand. This enables e.g. to compare different algorithms, or to compute the optimal adaptation of a model. Analyzing the computational complexity can add a new dimension to this study, as estimating the computations is usually more straightforward, e.g. by measuring the execution time or the energy consumption. Properly normalized generalization capacity, with respect to computational complexity, then gives the most practical algorithm for the specific application at hand.

BIBLIOGRAPHY

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.

[Aka73] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, 1973.

[AM70] J. G. Augustson and J. Minker. An analysis of some graph theoretical clustering techniques. *J. ACM*, 17(4):571–588, 1970.

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[BCFS12] Joachim M. Buhmann, Morteza Haghir Chehreghani, Mario Frank, and Andreas P. Streich. Information theoretic model selection for pattern analysis. *Journal of Machine Learning Research - Proceedings Track*, 27:51–64, 2012.

[BDvP06] Shai Ben-David, Ulrike von Luxburg, and David Pal. A sober look at clustering stability. In G. Lugosi and H.U. Simon, editors, *COLT'06, Pittsburgh, PA, USA*, pages 5–19, 2006.

[BH09] Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-laplacian. In *ICML*, page 11, 2009.

[BNM+11] Mohamed Banni, Alessandro Negri, Flavio Mignone, Hamadi Boussetta, Aldo Viarengo, and Francesco Dondero. Gene expression rhythms in the mussel Mytilus galloprovincialis (lam.) across an annual cycle. *PLoS ONE*, 6(5):e18904, 05 2011.

[Bru78] P. Brucker. On the complexity of clustering problems. 1978.

[BS94] N. Barkai and H. Sompolinsky. Statistical mechanics of the maximum-likelihood density estimation. *Phys. Rev. E*, 50(3):1766–1769, 1994.

[BSS93] N. Barkai, H. S. Seung, and H. Sompolinsky. Scaling laws in learning of classification tasks. *Phys. Rev. Lett.*, 70(20):3167–3170, 1993.

[Buh10] Joachim M. Buhmann. Information theoretic model validation for clustering. In *International Symposium on Information Theory (ISIT)*, pages 1398 – 1402, 2010.

[Buh11] Joachim M. Buhmann. Context sensitive information: Model validation by information theory. In *Mexican Conference on Pattern Recognition*, pages 12–21, 2011.

[Bus12] Ludwig Maximilian Busse. *Information in orderings. learning to order*. PhD thesis, ETH Zurich, 2012.

[CBB12] Morteza Haghir Chehreghani, Alberto Giovanni Busetto, and Joachim M. Buhmann. Information theoretic model validation for spectral clustering. *Journal of Machine Learning Research - Proceedings Track*, 22:495–503, 2012.

[CBB13] Morteza Haghir Chehreghani, Ludwig M. Busse, and Joachim M. Buhmann. Information content in clustering algorithms. In *DAGSTAT*, 2013.

[CH08] Gerda Claeskens and Nils Lid Hjort. *Model Selection and Model Averaging*. Cambridge University Press, 2008.

[CK11] Imre Csiszár and János Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.

[CM02] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[CSZ94] Pak K. Chan, Martine D. F. Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 13(9):1088–1096, 1994.

[CT06] Thomas M. Cover and Jay A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 2nd edition, 2006.

[DEFI06] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.

[DF02] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002.

[DGK07] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

[DTEK06] John C. Duchi, Daniel Tarlow, Gal Elidan, and Daphne Koller. Using combinatorial optimization within max-product belief propagation. In *Advances in Neural Information Processing Systems, NIPS*, pages 369–376, 2006.

[EA06] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868, December 1998.

[FB03] Bernd Fischer and Joachim M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(4):513–518, 2003.

[FCB11] Mario Frank, Morteza Haghir Chehreghani, and Joachim M. Buhmann. The minimum transfer cost principle for model-order selection. In *European Conference on Machine Learning and Knowledge Discovery in Databases - ECML/PKDD*, pages 423–438, 2011.

[FSBB12] Mario Frank, Andreas P. Streich, David Basin, and Joachim M. Buhmann. Multi-assignment clustering for Boolean data. *Journal of Machine Learning Research*, 13:459–489, 2012.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

[GH88] Olivier Goldschmidt and Dorit S. Hochbaum. Polynomial algorithm for the k-cut problem. In *29th Annual Symposium on Foundations of Computer Science, FOCS*, pages 444–451, 1988.

[GKR01] Martin Grötschel, Sven Krumke, and Jörg Rambau, editors. *Online Optimization of Large Scale Systems*. Springer, 2001.

[GP02] Erhan Gokcay and José Carlos Príncipe. Information theoretic clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(2):158–171, 2002.

[Haj88] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operation Research*, 13:311–329, 1988.

[HAvL07] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1368, 2007.

[HB97] Thomas Hofmann and Joachim M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(1):1–14, 1997.

[HL96] Lars Kai Hansen and Jan Larsen. Unsupervised learning and generalization. In *In Proceedings of the IEEE International Conference on Neural Networks*, pages 25–30, 1996.

[HPB98] Thomas Hofmann, Jan Puzicha, and Joachim M. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):803–818, 1998.

[HTF08] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Verlag, New York, 2008.

[Jay57a] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, 1957.

[Jay57b] Edwin T. Jaynes. Information theory and statistical mechanics. ii. *Physical Review*, 106(2):171–190, 1957.

[Kir09] Mark Kirkpatrick. Patterns of quantitative genetic variation in multiple dimensions. *Genetica*, 136:271–284, 2009.

[KS96] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996.

[KT01] Herbert Kocha and Daniel Tataru. Well-posedness for the Navier-Stokes equations. *Advances in Mathematics*, 157(1):22 –35, 2001.

[Kuh55] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[LBRB04] Tilman Lange, Mikio Braun, Volker Roth, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2004.

[LY10] Hairong Liu and Shuicheng Yan. Robust graph mode seeking by graph shift. In *International Conference on Machine Learning (ICML)*, pages 671–678, 2010.

[Mer94] Neri Merhav. Bounds on achievable convergence rates of parameter estimators via universal coding. *IEEE Transactions on Information Theory*, 40(4):1210–1215, 1994.

[MS65] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of tura'n. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[PA88] Carsten Peterson and James R. Anderson. Neural networks and np-complete optimization problems; a performance study on the graph bisection problem. *Complex Systems*, 2:59–89, 1988.

[PHB00] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. A theory of proximity based clustering: structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 2000.

[PP03a] Massimiliano Pavan and Marcello Pelillo. Dominant sets and hierarchical clustering. In *ICCV*, pages 362–369, 2003.

[PP03b] Massimiliano Pavan and Marcello Pelillo. Graph-theoretic approach to clustering and segmentation. In *CVPR (1)*, pages 145–152, 2003.

[PP07] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, 2007.

[RGF90] Kenneth Rose, Eitan Gurewitz, and Geoffrey Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(9):589–594, 1990.

[Ris78] Jorma Rissanen. Modeling by the shortest data description. *Automation*, 14:465–471, 1978.

[RLKB03] Volker Roth, Julian Laub, Motoaki Kawanabe, and Joachim M. Buhmann. Optimal cluster preserving embedding of non-metric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12), 2003.

[RMG96] Kenneth Rose, David J. Miller, and Allen Gersho. Entropy-constrained tree-structured vector quantizer design. *IEEE Transactions on Image Processing*, 5(2):393–398, 1996.

[Ros98] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 86(11):2210–2239, 1998.

[RR04] Ali Rahimi and Ben Recht. Clustering with normalized cuts is clustering with a hyperplane. In *in Statistical Learning in Computer Vision*, 2004.

[SATB05] Noam Slonim, Gurinder Singh Atwal, Gasper Tracik, and William Bialek. Information-based clustering. *Proceedings of the National Academy of Science (PNAS)*, 102:18297–1830, 2005.

[SB04] Susanne Still and William Bialek. How many clusters? an information-theoretic perspective. *Neural Computation*, 16(12):2483–2506, 2004.

[Sch78] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[SS01] Padmanabhan Soundararajan and Sudeep Sarkar. Investigation of measures for grouping by graph partitioning. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 239–246, 2001.

[ST10a] Yevgeny Seldin and Naftali Tishby. Pac-bayesian analysis of co-clustering and beyond. *J. Mach. Learn. Res.*, 11:3595–3646, 2010.

[ST10b] Ohad Shamir and Naftali Tishby. Stability and model selection in *k*-means clustering. *Machine Learning*, 80(2-3):213–243, 2010.

[TAP77] D.J. Thouless, P.W. Anderson, and R.G. Palmer. A solution to a 'solvable' model of a spin glass. *Philosophical Magazine*, 35:593, 1977.

[TPB99] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[TTL84] Y. Tikochinsky, N. Tishby, and R. D. Levine. Alternative approach to maximum-entropy inference. *Physical Review Letters A*, 30:2638–2644, 1984.

[TWH00] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society, Series B*, 63:411–423, 2000.

[Vap82] Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, Berlin, Heidelberg, 1982.

[Vap98] Vladimir N. Vapnik. *Statistical learning theory*. Wiley, New York, 1998.

[vLWG12] U. von Luxburg, R.C. Williamson, and I. Guyon. Clustering: Science or art? In I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, editors, *Journal for Machine Learning Research, W&CP*, volume 27, pages 65–79. MIT Press, 2012.

[WB09] Aree Witoelar and Michael Biehl. Phase transitions in vector quantization and neural gas. *Neurocomputing*, 72(7-9):1390–1397, 2009.

[Wei95] Jorgen W. Weibull. *Evolutionary game theory*. MIT Press, 1995.

[WL93] Zhenyu Wu and Richard M. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1101–1113, 1993.

[YB99] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, 27(5):1564–1599, 1999.

Curriculum Vitae

Morteza Haghir Chehreghani

Machine Learning Laboratory,
Department of Computer Science,
ETH Zurich, Switzerland.

EDUCATION

- PhD in Computer Science, ETH Zurich, Switzerland, Nov. 2008- Oct. 2013.
- M.Sc. in Computer Engineering, Sharif University of Technology, Iran, Sep. 2005- Dec. 2007.
- B.Sc. in Computer Engineering, Amirkabir University of Technology (Tehran Polytechnic), Iran, Sep. 2000- Jul. 2005.

PUBLICATIONS

[1] Ludwig M. Busse, **Morteza Haghir Chehreghani**, Joachim M. Buhmann, "*Approximate Sorting*", German Conference on Pattern Recognition (GCPR), pp. 142-152, 2013.

[2] **Morteza Haghir Chehreghani**, Ludwig M. Busse, Joachim M. Buhmann, *"Information Content in Clustering Algorithms"*, DAGStat, 2013.

[3] **Moretza Haghir Chehreghani**, Alberto G. Busetto, Joachim M. Buhmann, *"Information Theoretic Model Validation for Spectral Clustering"*, Journal of Machine Learning Research (JMLR), Proc. of AIStats, 22: 495-503, 2012.

[4] Ludwig M. Busse, **Morteza Haghir Chehreghani**, Joachim M. Buhmann, *"The Information Content in Sorting Algorithms"*, International Symposium on Information Theory (ISIT), pp. 2746-2750, 2012.

[5] Joachim M. Buhmann, **Morteza Haghir Chehreghani**, Mario Frank and Andreas P. Streich, *"Information Theoretic Model Selection for Pattern Analysis"* , Journal of Machine Learning Research (JMLR), ICML workshop on Unsupervised and Transfer Learning, 27:51–64, 2012.

[6] **Morteza Haghir Chehreghani**, Mostafa H. Chehreghani, Hassan Abolhassani, "*Probabilistic Heuristics for Hierarchical Web Data Clustering"*. Computational Intelligence, 28(2): 209-233, 2012.

[7] Mostafa H. Chehreghani, **Morteza Haghir Chehreghani**, Caro Lucas, Masoud Rahgozar, *"OInduced: An Efficient Algorithm for Mining Induced Patterns from Rooted Ordered Trees"*, IEEE Transactions on Systems, Man, and Cybernetics, Part A, 41(5): pp. 1013-1025, 2011.

[8] Mario Frank, **Morteza Haghir Chehreghani**, Joachim M. Buhmann, *"The Minimum Transfer Cost Principle for Model-Order Selection"*, European Conference on Machine Learning and Knowledge Discovery in Databases ECML/PKDD (1): 423-438, 2011.

[9] **Moretza Haghir Chehreghani**, Hassan Abolhassani, *"Developing Density and Link Based Methods for Clustering the Web Pages*", Decision Support Systems, 47 (4): 374-382, 2009.

[10] **Moretza Haghir Chehreghani**, Hassan Abolhassani, Mostafa H. Chehreghani, "*Improving Density Based Methods for Hierarchical Clustering of the Web Pages*", Data & Knowledge Engineering, 67 (1): 30-50, 2008.

[11] Mehrdad Mahdavi, **Morteza Haghir Chehreghani**, Hassan Abolhassani, and R. Forsati, "*Novel Meta-Heuristic Algorithms for Clustering Web Documents"*, Applied Mathematics and Computation**,** 201 (1-2): 441–451, 2008.

[12] **Morteza Haghir Chehreghani**, Hassan Abolhassani, *"H-BayesClust: A new Hierarchical Clustering based on Bayesian Networks"*, International Conference on Advanced Data Mining and Applications (ADMA), LNCS, pp. 616–624, Springer, 2007.

[13] **Moretza Haghir Chehreghani**, Mohammad Rahmati, *"Design and Implementation of a 2-D Barcode based on Data Matrix Format"*, 15th Iranian Conference on Electrical Engineering, 2007.