Burgard, Jan Pablo; Moreira Costa, Carina; Schmidt, Martin

Robustification of the k-means clustering problem and tailored decomposition methods: when more conservative means more accurate

Annals of Operations Research

Provided in Cooperation with:
Springer Nature

Suggested Citation: Burgard, Jan Pablo; Moreira Costa, Carina; Schmidt, Martin (2022) : Robustification of the k-means clustering problem and tailored decomposition methods: when more conservative means more accurate, Annals of Operations Research, ISSN 1572-9338, Springer US, New York, NY, Vol. 339, Iss. 3, pp. 1525-1568, https://doi.org/10.1007/s10479-022-04818-w

This Version is available at:
https://hdl.handle.net/10419/318622

Annals of Operations Research (2024) 339:1525–1568
https://doi.org/10.1007/s10479-022-04818-w

ORIGINAL RESEARCH

Robustification of the *k*-means clustering problem and tailored decomposition methods: when more conservative means more accurate

Jan Pablo Burgard[1] · **Carina Moreira Costa**[2] · **Martin Schmidt**[2]

Accepted: 6 June 2022 / Published online: 25 July 2022

Abstract
k-means clustering is a classic method of unsupervised learning with the aim of partitioning a given number of measurements into k clusters. In many modern applications, however, this approach suffers from unstructured measurement errors because the *k*-means clustering result then represents a clustering of the erroneous measurements instead of retrieving the true underlying clustering structure. We resolve this issue by applying techniques from robust optimization to hedge the clustering result against unstructured errors in the observed data. To this end, we derive the strictly and Γ-robust counterparts of the *k*-means clustering problem. Since the nominal problem is already NP-hard, global approaches are often not feasible in practice. As a remedy, we develop tailored alternating direction methods by decomposing the search space of the nominal as well as of the robustified problems to quickly obtain feasible points of good quality. Our numerical results reveal an interesting feature: the less conservative Γ-approach is clearly outperformed by the strictly robust clustering method. In particular, the strictly robustified clustering method is able to recover clusterings of the original data even if only erroneous measurements are observed.

Keywords *k*-means clustering · Alternating direction methods · Robust optimization · Strict robustness · Γ-robustness

Mathematics Subject Classification 90-XX · 90Cxx · 90C11 · 90C90

✉ Martin Schmidt
martin.schmidt@uni-trier.de

Jan Pablo Burgard
burgardj@uni-trier.de

Carina Moreira Costa
Carinamath5@gmail.com

[1] Department of Economic and Social Statistics, Trier University, Universitätsring 15, 54296 Trier, Germany

[2] Department of Mathematics, Trier University, Universitätsring 15, 54296 Trier, Germany

1 Introduction

In statistical analyses, a typical assumption is that the used data is measured without error. However, this assumption is often violated. One possible way to approach this problem is to assume measurement errors in the observed variables. However, these measurement errors often are not specifiable with distributional assumptions as nothing is known about them. Measurement errors with unknown structure appear in many different application areas and will be referred to as *unstructured errors* hereafter. For example, Rocke et al. (2009) state four sources of such errors in the case of micro-array analysis. When using register data, a typical source of errors are over- and undercounts, which are typically not detectable from the register itself; see Burgard and Münnich (2012). Another example is the analysis of the potential use of big data in public health research in Khoury and Ioannidis (2014), where the authors strikingly state "'Big Error' can plague Big Data".

The field of robust optimization treats this problem of unstructured, uncertain data by robustifying optimization problems with uncertain parameters. In this setting, no additional information about these parameters is required besides that they belong to a prescribed uncertainty set. See Soyster (1973) for the first paper on robust optimization and the textbook by Ben-Tal et al. (2009) or the survey article by Bertsimas et al. (2011) for a general overview. For a recent application of these ideas to linear mixed models; see Burgard et al. (2020). However, the most basic variant of robust optimization—usually called strict robustness—was often criticized since it may lead to overly conservative solutions as they are explicitly hedged against the worst-case. As a remedy, less conservative concepts of robustness have been proposed in the last decades; see, e.g., recoverable robustness (Liebchen et al. 2009), light robustness (Fischetti and Monaci 2009), adjustable robustness (Ben-Tal et al. 2004), or Γ-robustness (Bertsimas and Sim 2004).

As motivated above, in statistics, there are several optimization problems, which should be re-engineered to account for the unstructured error setting as it arises in big data sources. By developing new robust optimization methodology for optimization problems from statistics and/or classification, it is widely accepted that a major advancement in the analysis and usability of large data sources can be achieved. For example, a very important and widely used method is k-means clustering (MacQueen 1967; Lloyd 1982). It is used to find clusters in an unsupervised learning setting and is widely applied in the context of big data analysis; see, e.g., Grira et al. (2004) and Celebi and Aydin (2016). However, it does not compensate for possible errors in the data set. Therefore, the clusters that are found are perturbed by the unstructured errors and do not necessarily correspond to the true underlying structure of interest.

The marriage of problems from classification or statistics with modern techniques from mathematical optimization has been a field of very active research during the last years and a comprehensive overview over this literature is out of the scope of this article. Thus, we mainly focus on reviewing the literature at the interface of robust optimization, k-means clustering, and classification problems. One of the most prominent methods for classification are support vector machines (SVMs), which also have been considered from the point of view of robust optimization; see, e.g., Bhattacharyya et al. (2005), Trafalis and Gilbert (2007), Pant et al. (2011) and Bertsimas et al. (2019). The results obtained in these papers often illustrate that robustified solutions can give higher accuracy than nominal ones, leading to improvements in the SVM method. The most recent paper by Bertsimas et al. (2019) from the above list also contains the application of robust optimization to two other major and widely used classification methods (besides SVMs): logistic regression and CART. The presented extensive

computational results show that the robust formulations outperform nominal and regularized models for most of the instances—especially in the case of high-dimensional instances and instances that are hard to separate. Moreover, to the best of our knowledge, Bertsimas et al. (2019) is the only paper that also uses the concept of Γ-robustness for classification. Interestingly, the authors report that Γ-robustness is beneficial for robustifying SVMs. In contrast to that, our results for Γ-robustified k-means clustering are clearly outperformed by the strictly robustified models.

Our contribution is the following. We follow the way paved by Bertsimas et al. (2019) but consider strictly and Γ-robust optimization for the k-means clustering problem. The robustification of the underlying optimization problem allows us to consider erroneous input data. As a consequence, it becomes possible to compute clusterings that are closer to the clusterings of the original, i.e., error-free, data although only erroneous input data can be considered. To the best of our knowledge, this issue has not been studied before in the literature. Moreover, we develop tailored alternating direction methods (ADMs) to quickly obtain solutions of good quality. Here, one of the main insights is that an ADM for the nominal case can be generalized so that it becomes applicable to the robust counterparts while keeping its convergence properties.

Classic ADMs are extensions of augmented Lagrangian methods and have been originally proposed in Gabay and Mercier (1976) and Glowinski and Marroco (1975). A general analysis of the convergence theory of ADMs is presented in Bertsekas and Tsitsiklis (1989). The particular case of ADMs applied to convex objective functions $f(u, v)$ over disjoint constraint sets $u \in U$, $v \in V$, is discussed in Wendell and Hurter (1976). An extension to the case of a biconvex f is given in Gorski et al. (2007). More recently, in Boyd et al. (2011), the authors discuss applications of the alternating direction methods of multipliers (ADMMs), which are closely related to ADMs, in the broad area of machine learning and highlight its possibility of parallelization and distributed computing. ADMs are very broadly applied; for an application of ADMs to solve nonconvex MINLPs in the context of gas transport, we refer to Geißler et al. (2015) and Geißler et al. (2018). For an application in supply chain management, we refer to Schewe et al. (2020). Furthermore, in Geißler et al. (2017) it is shown that idealized feasibility pumps can be seen as an ADM and a penalty-based ADM for MIPs and MINLPs is proposed. A similar method based on block coordinate descent and coordinate descent has been applied recently in the context of optimal imputation in Bertsimas et al. (2017). However, the authors do not consider clustering problems.

In general, these methods rely on a problem-specific decomposition of the variable space of the robust counterparts. Using these decompositions, solutions of good quality can be obtained by alternatingly solving two sub-problems that are significantly easier to solve than the robust counterpart. For the nominal case, the applied ADM exactly mimics the classic k-means clustering algorithm. However, our more abstract view on the decomposition of the problem later allows to carry the idea of alternatingly solving easier subproblems over to the robustified settings. Strictly robust counterparts for the k-means clustering problem are also considered in Li et al. (2016) and Vo et al. (2016). However, there is no investigation of structural properties of data sets that indicate whether a robustification of k-means might be beneficial or not. Additionally, by explicitly framing our method as an ADM, we are also able to provide a convergence result (both for the nominal and the two robustified clustering methods), which is not done in Li et al. (2016) and Vo et al. (2016). We also extend our ADM setting so that partial minima (delivered by the ADM) of bad quality can be detected and improved so that the alternating algorithm can be restarted with a significantly improved starting point. We show in our numerical results that this restart heuristic leads to significantly better clustering results. In Li et al. (2016), the authors focus on clustering of incomplete

data sets. Thus, in their computational study, the robust clustering results are evaluated by comparing them to other existing k-means algorithms for incomplete data sets. In contrast to this, the goal of this paper is to evaluate the quality of robustified solutions obtained and to develop an understanding of structural properties of data sets for which robust clustering is beneficial. It turns out that a robustification of the clustering approach is not required if the underlying data points are already well separated. However, if this is not the case, the robustified method is able to recover the true clustering although only erroneous data is observed. Our analysis especially leads to an interesting result regarding the comparison of strictly and Γ-robust clustering. The Γ-robust approach was developed to overcome the criticism of strictly robust solutions that are often too conservative in practice. For k-means clustering we observe the contrary: The conservatism of strict robustness leads to significantly better clustering results when compared to the Γ-robust solutions.

The remainder of the paper is structured as follows. In Sect. 2 we formally introduce the k-means clustering problem and state it as a mixed-integer nonlinear optimization problem (MINLP). Afterward, Sect. 3 contains the strictly robust counterpart, followed by the Γ-robust counterpart in Sect. 4. We state a generic ADM in Sect. 5 and discuss its basic convergence properties that can also be applied if a tailored version is applied to nominal or robustified clustering problems. By doing so, we can explicitly characterize the obtained clustering solutions as so-called partial minima of the underlying MINLP. Moreover, we state the specific sub-problems that need to be solved in every iteration of the ADM (if applied to the nominal or one of the two robust clustering methods) and show that they can always be solved efficiently. In Sect. 6, we describe the already mentioned restart heuristic and present our numerical results in Sect. 7. The paper closes with some concluding remarks and notes on future research topics in Sect. 8.

2 The k-means clustering problem and an MINLP formulation

Let $X \in \mathbb{R}^{p \times n}$ be the matrix containing the data set for clustering, which consists of n data points in $\mathbb{R}^p$. Here, the data point $x^i \in \mathbb{R}^p, i = 1, \ldots, n$, corresponds to the ith column of X. For given k, the goal of k-means clustering is to find mean vectors $\mu^j \in \mathbb{R}^p$, $j = 1, \ldots, k$, of k clusters that satisfy

$$\mu^* = \arg\min_{\mu} \; h(X, \mu), \quad \mu = (\mu^j)_{j=1,\ldots,k}, \tag{1}$$

where the loss function h is a sum of distances such as the squared Euclidean distance

$$h(X, \mu) = \sum_{j=1}^{k} \sum_{x^i \in S_j} \|x^i - \mu^j\|_2^2, \tag{2}$$

with $S_j \subset \mathbb{R}^p$ being the set of data points that are assigned to cluster j.

By introducing binary variables $b_{i,j} \in \{0, 1\}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$, we can reformulate the function h as

$$h(X, \mu, b) = \sum_{j=1}^{k} \sum_{i=1}^{n} b_{i,j} \|x^i - \mu^j\|_2^2, \quad b = (b_{i,j})_{i=1,\ldots,n}^{j=1,\ldots,k},$$

where the binary variables have the meaning

$$b_{i,j} = \begin{cases} 1, & \text{if point } x^i \text{ is assigned to cluster } S_j, \\ 0, & \text{otherwise.} \end{cases}$$

As $x^i \in \mathbb{R}^p$ should belong to exactly only one cluster, we include the constraint

$$\sum_{j=1}^{k} b_{i,j} = 1 \quad \text{for all} \quad i = 1, \ldots, n.$$

To simplify the presentation in what follows, we introduce the notations

$$N := \{1, \ldots, n\}, \quad P := \{1, \ldots, p\}, \quad K := \{1, \ldots, k\}.$$

Thus, a reformulation of the k-means clustering problem as a mathematical optimization problem is given by

$$\min_{\mu, b} \quad \sum_{j \in K} \sum_{i \in N} b_{i,j} \|x^i - \mu^j\|_2^2 \tag{3a}$$

$$\text{s.t.} \quad \sum_{j \in K} b_{i,j} = 1 \quad \text{for all} \quad i \in N, \tag{3b}$$

$$b_{i,j} \in \{0, 1\} \quad \text{for all} \quad i \in N, \ j \in K, \tag{3c}$$

$$b \in \mathbb{R}^{n \times k}, \tag{3d}$$

$$\mu \in \mathbb{R}^{p \times k}. \tag{3e}$$

This is a mixed-integer nonlinear programming problem (MINLP); all constraints are linear but the objective function is cubic.

3 The strict robust counterpart of the *k*-means clustering MINLP

Up to now, we assumed that the data points in X are known exactly. However, in practice, often only erroneous measurements $\tilde{X} = X + E$ can be observed instead, where $E \in \mathbb{R}^{p \times n}$ describes the respective perturbation of the original data. We assume that there is neither information on the expected value nor on any higher order moment of the additive perturbation E. The error could even be a deterministic error as opposed to a stochastic one.

In statistics, measurement error correction methods typically assume that the errors are random variables that follow a known distribution; see Carroll et al. (2006). For a recent proposal on dealing with clustering in these classic measurement error settings; see Su et al. (2018). By accounting for the error distribution in the estimation process, the influence of the errors is reduced. However, this assumption is often very restrictive. Especially when using big data sources as in Davalos (2017) and Yamada et al. (2018), no meaningful error distribution can be assumed. As it is shown in White (2011) for biomarker measures, errors can be due to specimen collection and storage. Last but not least, in social science and econometric analysis, the indicators used are typically estimates resulting form a long statistical production process that is prone to different kind of errors; see, e.g., Alfons et al. (2013). All these different types of errors can even extend to be systematic and thus are far from being described as a zero-mean distributed error.

In this paper we tackle the following situation: Although we can only observe erroneous measures $\tilde{X}$ in practice, we want to compute a clustering that is as close as possible to the clustering of the original data $X \in \mathbb{R}^{p\times n}$. Obviously, especially if E has a mean vector different from $0 \in \mathbb{R}^p$ over all observations in every cluster for every variable, the optimal solution

$$\tilde{\mu}^* = \arg\min_{\mu} \; h(\tilde{X}, \mu), \quad \mu = (\mu^j)_{j\in K},$$

will differ from the optimal solution of Problem (1). Therefore, the optimization problem has to be reformulated to account for the unobservable and unstructured error E. This can be done by using the min-max problem

$$\min_{\mu} \max_{D\in\mathcal{U}} \; h(\tilde{X} - D, \mu),$$

with $\mathcal{U} \subseteq \mathbb{R}^{p\times n}$ being the uncertainty set, which will be specified right below. In particular, we assume that $\mathcal{U}$ can be chosen so that $E \in \mathcal{U}$ holds. Thus, given an uncertainty set $\mathcal{U}$, we minimize the function h considering the worst-case scenario w.r.t. all possible $D \in \mathcal{U}$.

As noted above, we assume that the error E is unstructured. Thus, we particularly refrain from using distributional information. Instead, we consider the box-uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{p\times n}$ to be given as

$$\mathcal{U}^{\text{box}} := \{D \in \mathbb{R}^{p\times n} : \; -\Delta d^i_l \le d^i_l \le \Delta d^i_l, \; \Delta d^i_l \ge 0, \; (l,i) \in P \times N\}. \tag{4}$$

With this at hand, we can now state the strictly robust counterpart of Problem (3):

$$\min_{\mu,b} \max_{D\in\mathcal{U}^{\text{box}}} \quad \sum_{j\in K}\sum_{i\in N} b_{i,j} \|\tilde{x}^i - d^i - \mu^j\|_2^2 \tag{5a}$$

$$\text{s.t.} \quad \sum_{j\in K} b_{i,j} = 1, \quad i \in N, \tag{5b}$$

$$b_{i,j} \in \{0,1\}, \quad i \in N, \; j \in K, \tag{5c}$$

$$\mu \in \mathbb{R}^{p\times k}. \tag{5d}$$

This robust counterpart is an optimization problem that is not directly tractable as it is stated in (5). The main reason for its hardness is the min-max structure of the objective function. Fortunately, we can reformulate this problem without the inner maximization problem, as shown in the following theorem.

Theorem 3.1 *The robust counterpart* (5) *is equivalent to*

$$\min_{\mu,b,\alpha} \quad \sum_{j\in K}\sum_{i\in N} b_{i,j} \left(\|\tilde{x}^i - \mu^j\|_2^2 + \|\Delta d^i\|_2^2 + \sum_{l\in P} 2\,\Delta d^i_l\,\alpha_{i,j,l} \right) \tag{6a}$$

$$\text{s.t.} \quad -\alpha_{i,j,l} \le \tilde{x}^i_l - \mu^j_l \le \alpha_{i,j,l}, \quad i \in N, \; j \in K, \; l \in P, \tag{6b}$$

$$\sum_{j\in K} b_{i,j} = 1, \quad i \in N, \tag{6c}$$

$$b_{i,j} \in \{0,1\}, \quad i \in N, \; j \in K, \tag{6d}$$

$$\mu \in \mathbb{R}^{p\times k}. \tag{6e}$$

Proof First, we move the uncertainty from the objective function to the constraints by introducing an extra variable $t \in \mathbb{R}$ and by rewriting the robust counterpart (5) equivalently as

$$\min_{\mu,b,t} \quad t \tag{7a}$$

$$\text{s.t.} \quad \max_{D\in\mathcal{U}^{\text{box}}} \left\{ \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - d^i - \mu^j\|_2^2 \right\} \le t, \tag{7b}$$

$$\sum_{j\in K} b_{i,j} = 1, \quad i \in N, \tag{7c}$$

$$b_{i,j} \in \{0,1\}, \quad i \in N,\ j \in K, \tag{7d}$$

$$\mu \in \mathbb{R}^{p\times k}. \tag{7e}$$

Now, the uncertainties only appear in Constraint (7b), which in turn is equivalent to

$$\max_{D\in\mathcal{U}^{\text{box}}} \left\{ \sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|\tilde{x}^i - \mu^j\|_2^2 + \|d^i\|_2^2 - 2(\tilde{x}^i - \mu^j)^\top d^i\right) \right\} \le t$$

$$\iff \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - \mu^j\|_2^2 + \max_{D\in\mathcal{U}^{\text{box}}} \left\{ \sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|d^i\|_2^2 - 2(\tilde{x}^i - \mu^j)^\top d^i\right) \right\} \le t.$$

Using the specific box-structure of the uncertainty set (4), we obtain

$$\begin{aligned}
&\max_{D\in\mathcal{U}^{\text{box}}} \sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|d^i\|_2^2 - 2(\tilde{x}^i - \mu^j)^\top d^i\right) \\
&= \max_{-\Delta d_l^i \le d_l^i \le \Delta d_l^i} \sum_{j\in K}\sum_{i\in N}\sum_{l\in P} b_{i,j}\left((d_l^i)^2 - 2(\tilde{x}_l^i - \mu_l^j)d_l^i\right) \\
&= \sum_{j\in K}\sum_{i\in N}\sum_{l\in P} b_{i,j}\left((\Delta d_l^i)^2 + 2|\tilde{x}_l^i - \mu_l^j|\Delta d_l^i\right).
\end{aligned}$$

Thus, we get the constraint

$$\sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|\tilde{x}^i - \mu^j\|_2^2 + \sum_{l\in P}(\Delta d_l^i)^2 + 2\Delta d_l^i|\tilde{x}_l^i - \mu_l^j|\right) \le t,$$

instead of (7b), which, in turn, can be re-written as

$$\begin{aligned}
\sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|\tilde{x}^i - \mu^j\|_2^2 + \|\Delta d^i\|_2^2 + \sum_{l\in P} 2\Delta d_l^i \alpha_{i,j,l}\right) &\le t, \\
-\alpha_{i,j,l} \le \tilde{x}_l^i - \mu_l^j &\le \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P, \\
\alpha_{i,j,l} &\ge 0, \quad i \in N,\ j \in K,\ l \in P.
\end{aligned}$$

Thus, Problem (5) is equivalent to

$$
\begin{aligned}
\min_{\mu,b,t,\alpha} \quad & t \\
\text{s.t.} \quad & \sum_{j\in K}\sum_{i\in N} b_{i,j}\left(\|\tilde{x}^i-\mu^j\|_2^2+\|\Delta d^i\|_2^2+\sum_{l\in P}2\Delta d_l^i\alpha_{i,j,l}\right)\le t,\\
& -\alpha_{i,j,l}\le \tilde{x}_l^i-\mu_l^j\le\alpha_{i,j,l},\quad i\in N,\ j\in K,\ l\in P,\\
& \alpha_{i,j,l}\ge 0,\quad i\in N,\ j\in K,\ l\in P,\\
& \sum_{j\in K} b_{i,j}=1,\quad i\in N,\\
& b_{i,j}\in\{0,1\},\quad i\in N,\ j\in K,\\
& \mu\in\mathbb{R}^{p\times k},
\end{aligned}
$$

and the result follows by eliminating t again. □

A standard criticism regarding strictly robust solutions (like those of Problem (6)) is that they tend to be too conservative as they are hedged against the worst-case error that can appear for all entries of the matrix $\tilde{X}$. To obtain robust solutions that are practically meaningful but less conservative, in Bertsimas and Sim (2004) the authors propose the so-called Γ-robust approach, which allows to control the degree of conservatism. Thus, in the next section, we apply this approach to the k-means clustering problem as well.

4 The Γ-robust counterpart of the *k*-means clustering MINLP

The Γ-approach proposed in Bertsimas and Sim (2004) does not assume that all parameters will realize in a worst-case way but restricts the number of such parameters by $\Gamma\in\mathbb{N}$, which gives the approach its name. We now apply the technique proposed in Bertsimas and Sim (2004) to the k-means clustering problem. This means that some observations $\tilde{x}_l^i$ are not perturbed and, in this case, $\tilde{x}_l^i=x_l^i$ and $d_l^i=0$ holds. Thus, the matrices $\tilde{X}$ and X only differ in a subset of Γ many elements.

To state the Γ-robust counterpart of Problem (3), we consider the uncertainty set

$$
\mathcal{U}_\Gamma := \left\{ D\in\mathcal{U}^{\text{box}} \colon \left|\left\{ d_l^i : d_l^i\neq 0,\ (l,i)\in P\times N\right\}\right|\le\Gamma\right\}.
$$

Using this uncertainty set, we determine the maximum number of observations that may realize in worst-case way. However, we do not state in advance which will do so. Hence, the Γ-robust counterpart of Problem (3) is given by

$$
\begin{aligned}
\min_{\mu,b,\alpha} \quad & \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i-\mu^j\|_2^2 \\
& \quad +\max_{\{I\subseteq P\times N\colon |I|\le\Gamma\}}\left\{\sum_{j\in K}\sum_{(l,i)\in I} b_{i,j}\left((\Delta d_l^i)^2+2\Delta d_l^i\alpha_{i,j,l}\right)\right\} && \text{(8a)}\\
\text{s.t.} \quad & -\alpha_{i,j,l}\le\tilde{x}_l^i-\mu_l^j\le\alpha_{i,j,l},\quad i\in N,\ j\in K,\ l\in P, && \text{(8b)}\\
& \alpha_{i,j,l}\ge 0,\quad i\in N,\ j\in K,\ l\in P, && \text{(8c)}\\
& \sum_{j\in K} b_{i,j}=1,\quad i\in N, && \text{(8d)}
\end{aligned}
$$

$$b_{i,j} \in \{0,1\}, \quad i \in N,\ j \in K, \tag{8e}$$
$$\mu \in \mathbb{R}^{p\times k}, \tag{8f}$$

where we already replaced the absolute values with additional variables and linear constraints as in Theorem 3.1. This model can be reformulated as an equivalent problem without the inner maximization problem as we show next.

Theorem 4.1 *The* Γ*-robust counterpart* (8) *is equivalent to*

$$\min_{\mu,b,\alpha,\lambda,\beta} \quad \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - \mu^j\|_2^2 + \Gamma\lambda + \sum_{(l,i)\in P\times N} \beta_l^i \tag{9a}$$
$$\text{s.t.} \quad \lambda + \beta_l^i \geq \sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\Delta d_l^i \alpha_{i,j,l}\right), \quad i \in N,\ l \in P, \tag{9b}$$
$$-\alpha_{i,j,l} \leq \tilde{x}_l^i - \mu_l^j \leq \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P, \tag{9c}$$
$$\alpha_{i,j,l} \geq 0, \quad i \in N,\ j \in K,\ l \in P, \tag{9d}$$
$$\sum_{j\in K} b_{i,j} = 1, \quad i \in N, \tag{9e}$$
$$b_{i,j} \in \{0,1\}, \quad i \in N,\ j \in K, \tag{9f}$$
$$\lambda \geq 0, \tag{9g}$$
$$\beta_l^i \geq 0, \quad i \in N,\ l \in P, \tag{9h}$$
$$\mu \in \mathbb{R}^{p\times k}. \tag{9i}$$

Proof First, we rewrite Problem (8) as

$$\min_{\mu,b,\alpha,t} \quad t \tag{10a}$$
$$\text{s.t.} \quad \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - \mu^j\|_2^2 + \max_{\{I\subseteq P\times N:\ |I|\leq\Gamma\}} \left\{\sum_{j\in K}\sum_{(l,i)\in I} b_{i,j}\left((\Delta d_l^i)^2 + 2\Delta d_l^i \alpha_{i,j,l}\right)\right\} \leq t, \tag{10b}$$
$$-\alpha_{i,j,l} \leq \tilde{x}_l^i - \mu_l^j \leq \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P, \tag{10c}$$
$$\alpha_{i,j,l} \geq 0, \quad i \in N,\ j \in K,\ l \in P, \tag{10d}$$
$$\sum_{j\in K} b_{i,j} = 1, \quad i \in N, \tag{10e}$$
$$b_{i,j} \in \{0,1\}, \quad i \in N,\ j \in K, \tag{10f}$$
$$\mu \in \mathbb{R}^{p\times k}. \tag{10g}$$

Note that the inner maximization problem that appears in Constraint (10b), can be re-written by re-arranging the terms in the sums. By doing so, we obtain the following equivalent subset selection problem

$$\max_{\{I\subseteq P\times N:\ |I|\leq\Gamma\}} \left\{\sum_{(l,i)\in I}\sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\,\Delta d_l^i\,\alpha_{i,j,l}\right)\right\}. \tag{11}$$

We now reformulate the latter problem as a linear optimization problem. For details, we refer to Proposition 1 in Bertsimas and Sim (2004). From the cited result, it follows that an equivalent formulation of Problem (11) is given by

$$\max_{z\in\mathbb{R}^{p\times n}} \quad \sum_{(l,i)\in P\times N}\left(\sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\,\Delta d_l^i\,\alpha_{i,j,l}\right)\right) z_l^i \tag{12a}$$

$$\text{s.t.} \quad \sum_{(l,i)\in P\times N} z_l^i \le \Gamma, \tag{12b}$$

$$0 \le z_l^i \le 1, \quad i\in N,\ l\in P. \tag{12c}$$

This is a linear optimization problem in z and its dual problem reads

$$\min_{\lambda,\beta} \quad \Gamma\lambda + \sum_{(l,i)\in P\times N} \beta_l^i \tag{13a}$$

$$\text{s.t.} \quad \lambda + \beta_l^i \ge \sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\,\Delta d_l^i\,\alpha_{i,j,l}\right), \quad i\in N,\ l\in P, \tag{13b}$$

$$\lambda \ge 0, \tag{13c}$$

$$\beta_l^i \ge 0, \quad i\in N,\ l\in P. \tag{13d}$$

Here, λ is the dual variable corresponding to Constraint (12b) and β_l^i are the dual variables corresponding to the constraints in (12c). Since Problem (12) is feasible and bounded, we can apply strong duality, which states that the dual problem (13) is also feasible and bounded and that the primal and dual optimal objective function values coincide. We can thus replace the inner maximization problem in (10b) by its dual minimization problem. In addition, notice that we do not need the minimum here because if

$$\begin{aligned}
t &\ge \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - \mu^j\|_2^2 + \Gamma\lambda + \sum_{(l,i)\in P\times N}\beta_l^i,\\
\lambda + \beta_l^i &\ge \sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\Delta d_l^i\alpha_{i,j,l}\right), \quad i\in N,\ l\in P,\\
\lambda &\ge 0,\\
\beta_l^i &\ge 0, \quad i\in N,\ l\in P,
\end{aligned}$$

is true, it also holds for the minimum value of

$$\Gamma\lambda + \sum_{(l,i)\in P\times N}\beta_l^i$$

over the dual feasible set. Hence, we obtain

$$\begin{aligned}
\min_{\mu,b,t,\lambda,\beta} \quad & t\\
\text{s.t.} \quad & t \ge \sum_{j\in K}\sum_{i\in N} b_{i,j}\|\tilde{x}^i - \mu^j\|_2^2 + \Gamma\lambda + \sum_{(l,i)\in P\times N}\beta_l^i,\\
& \lambda + \beta_l^i \ge \sum_{j\in K} b_{i,j}\left((\Delta d_l^i)^2 + 2\Delta d_l^i\alpha_{i,j,l}\right), \quad i\in N,\ l\in P,\\
& -\alpha_{i,j,l} \le \tilde{x}_l^i - \mu_l^j \le \alpha_{i,j,l}, \quad i\in N,\ j\in K,\ l\in P,
\end{aligned}$$

$$
\begin{aligned}
&\alpha_{i,j,l} \geq 0, \quad i \in N,\ j \in K,\ l \in P,\\
&\sum_{j \in K} b_{i,j} = 1, \quad i \in N,\\
&b_{i,j} \in \{0,1\}, \quad i \in N,\ j \in K,\\
&\lambda \geq 0,\\
&\beta_l^i \geq 0, \quad i \in N,\ l \in P,
\end{aligned}
$$

and the result follows by eliminating t. □

5 Alternating direction methods for nominal and robust *k*-means clustering

In this section, we propose tailored alternating direction methods (ADMs) to compute approximate solutions to Problems (3), (6), and (9). Note that all these problems are MINLPs, which are in general NP-hard to solve to global optimality. Actually, the nominal k-means problem itself is an NP-hard problem even in the particular case of two dimensions; see, e.g., Dasgupta (2007), Aloise et al. (2009) and Mahajan et al. (2012). For such hard problems it is usually considered appropriate to study primal heuristics to quickly compute feasible points of good quality. The presented ADMs can be seen as such primal heuristics with the additional advantage that they come with a formal convergence analysis. The key idea of the ADMs is to decompose the set of variables into two blocks: the set of binary variables and the set of continuous variables. Afterward, we alternatingly solve the problem in one direction while keeping the other one fixed, which is a popular approach for clustering problems; see, e.g., Li et al. (2016) or Ames (2014), Pirinen and Ames (2019), where an alternating direction method of multipliers is used to solve semidefinite programming relaxations for computing a clustering of weighted graphs.

In the following, we first briefly review classic ADMs and afterward propose tailored versions of ADMs to solve Problems (3), (6), and (9).

5.1 General ADM framework and convergence properties

We now consider the general problem

$$
\begin{aligned}
\min_{u,v} \quad & f(u,v) && \text{(15a)}\\
\text{s.t.} \quad & g(u,v) = 0, \quad c(u,v) \geq 0, && \text{(15b)}\\
& u \in U \subseteq \mathbb{R}^{n_u}, \quad v \in V \subseteq \mathbb{R}^{n_v}, && \text{(15c)}
\end{aligned}
$$

for which we make the following assumption.

Assumption 1 The objective function $f : \mathbb{R}^{n_u+n_v} \to \mathbb{R}$ and the constraint functions $g : \mathbb{R}^{n_u+n_v} \to \mathbb{R}^m$, $c : \mathbb{R}^{n_u+n_v} \to \mathbb{R}^q$ are continuous and the sets U and V are non-empty and compact.

The feasible set of Problem (15) is denoted by Ω, i.e.,

$$\Omega = \{(u,v) \in U \times V : g(u,v) = 0,\ c(u,v) \geq 0\} \subseteq U \times V.$$

Alternating direction methods are iterative procedures that solve Problem (15) by alternatingly solving two simpler problems. Given an iterate (u^t, v^t), they solve Problem (15) with v fixed to v^t into the direction of u, yielding a new u-iterate u^{t+1}. Afterward, u is fixed to u^{t+1} and Problem (15) is solved into the direction of v, yielding a new v-iterate v^{t+1}. The algorithm is formally stated in Algorithm 1. The for-loop is repeated until a termination criterion is reached. In order to state convergence results for Algorithm 1, we need the following definition.

Algorithm 1 A standard ADM.

1: Choose initial values $(u^0, v^0) \in U \times V$.
2: **for** $t = 0, 1, \ldots$ **do**
3: Compute
$$u^{t+1} \in \arg\min_u \left\{ f(u, v^t) \colon g(u, v^t) = 0,\ c(u, v^t) \geq 0,\ u \in U \right\}.$$
4: Compute
$$v^{t+1} \in \arg\min_v \left\{ f(u^{t+1}, v) \colon g(u^{t+1}, v) = 0,\ c(u^{t+1}, v) \geq 0,\ v \in V \right\}.$$
5: Set $t \leftarrow t + 1$.
6: **end for**

Definition 5.1 Let $(u^*, v^*) \in \Omega$ be a feasible point of Problem (15). Then, (u^*, v^*) is called a *partial minimum* of Problem (15) if

$$\begin{aligned} f(u^*, v^*) &\leq f(u, v^*) \quad \text{for all} \quad (u, v^*) \in \Omega, \\ f(u^*, v^*) &\leq f(u^*, v) \quad \text{for all} \quad (u^*, v) \in \Omega \end{aligned}$$

holds.

Consider $\Theta(\bar{u}, \bar{v})$ being the set of possible iterates starting from point $(\bar{u}, \bar{v})$, i.e.,

$$\Theta(\bar{u}, \bar{v}) = \left\{ (u^*, v^*) \colon f(u^*, \bar{v}) \leq f(u, \bar{v}),\ u \in U;\ f(u^*, v^*) \leq f(u^*, v),\ v \in V \right\}.$$

The following general convergence result is taken from Gorski et al. (2007).

Theorem 5.1 *Let $\{(u^t, v^t)\}_{t=0}^{\infty}$ be a sequence generated by Algorithm 1 with $(u^{t+1}, v^{t+1}) \in \Theta(u^t, v^t)$. Suppose that the solution of the first optimization problem (in Line 3) is always unique. Then, every convergent subsequence of $\{(u^t, v^t)\}_{t=0}^{\infty}$ converges to a partial minimum. In addition, if w and w' are two limit points of such subsequences, then $f(w) = f(w')$ holds.*

The sequence $\{(u^t, v^t)\}_{t=0}^{\infty}$ generated by Algorithm 1 may not converge, but instead may have several convergent subsequences. In particular, if $\{(u^t, v^t)\}_{t=0}^{\infty}$ is contained in a compact set, then there exists at least one convergent subsequence $\{(u^t, v^t)\}_{t \in \mathcal{T}}$. A partial minimum may not be global minimum. However, as the problems that we need to solve are NP-hard, we are willing to accept sub-optimal solutions. For more details on the convergence theory of classic ADMs; see, e.g., Gorski et al. (2007) and Wendell and Hurter (1976).

5.2 An ADM for the nominal *k*-means clustering problem

In this section, we propose a tailored version of Algorithm 1 to compute partial minimum of Problem (3). First, we define the sets U and V. For each attribute's index $l \in P$, let

$\pi_l^- := \min\{x_l^i : i \in N\}$ and $\pi_l^+ := \max\{x_l^i : i \in N\}$ be the minimum and maximum value for this attribute, respectively, and let $\Pi(X) := [\pi_1^-, \pi_1^+] \times \cdots \times [\pi_p^-, \pi_p^+]$ be the bounding box of all data points. Thus, $x^i \in \Pi(X)$ holds for all $i \in N$.

Now, within the context of Algorithm 1, we consider the following non-empty and compact sets

$$U := \left\{ b \in \mathbb{R}^{n \times k} : b_{i,j} \in \{0,1\},\ \sum_{j \in K} b_{i,j} = 1,\ i \in N,\ j \in K \right\}, \tag{16}$$

$$V := \left\{ \mu \in \mathbb{R}^{p \times k} : \mu^j \in \Pi(X),\ j \in K \right\}, \tag{17}$$

which gives the variable splitting required for the application of the ADM. Note that, in Problem (3), we do not have any coupling constraint such as g or c in the general Problem (15). In what follows, we describe how we compute a partial minimum of Problem (3).

Suppose that initial mean vectors $\mu^0 \in V$ are given. In each iteration t, the problem in the direction of U reads

$$b^{t+1} \in \arg\min_{b \in U} \sum_{j \in K} \sum_{i \in N} b_{i,j} \|x^i - (\mu^j)^t\|_2^2. \tag{18}$$

The optimal solution to this binary problem can be obtained as follows. Consider a fixed $i \in N$. Since $b_{i,j} \in \{0,1\}$ for $j \in K$ and only one $b_{i,j}$ is equal to 1, we set $b_{i,j} = 1$ for j that minimizes the norm. This is exactly the j with μ^j being the center that is the closest to the point x^i. Thus, for each $i \in N$, we compute

$$J_i = \left\{ j : j \in \arg\min_{j \in K} \|x^i - (\mu^j)^t\|_2^2 \right\}. \tag{19}$$

Note that there may exist more than one element in J_i. In this case, we break the tie by choosing the smallest one. With b^{t+1} at hand, we update the mean vectors by solving the problem in the direction of V, which is given by

$$\mu^{t+1} = \arg\min_{\mu \in V} \sum_{j \in K} \sum_{i \in N} b_{i,j}^{t+1} \|x^i - \mu^j\|_2^2.$$

This problem can be solved very effectively by using the formula

$$(\mu^j)^{t+1} = \frac{1}{|C_j^{t+1}|} \sum_{i \in C_j^{t+1}} x^i, \quad j \in K, \tag{20}$$

with

$$C_j^{t+1} = \{i \in N : b_{i,j}^{t+1} = 1\}.$$

In other words, in each iteration, we obtain the new centers $\mu^{t+1} \in V$ by computing the centers for the given assignment b^{t+1}. Now, we can summarize the main steps of our algorithm to compute a partial minimum of Problem (3) in Algorithm 2. For details on how we choose the initial centers we refer to Sect. 7.3.

Since the method in Algorithm 2 is a special case of the general ADM in Algorithm 1, the general convergence result as stated in Theorem 5.1 can be applied, yielding that Algorithm 2 leads to partial minima. Let us finally mention that the presented ADM exactly mimics the classic k-means clustering algorithm. However, this more abstract view on decomposing the

Algorithm 2 ADM applied to the nominal k-means clustering problem (3).

1: Choose $\mu^0 \in V$.
2: **for** $t = 0, 1, \ldots$ **do**
3: **for** $i \in N$ **do**
4: Compute J_i as in (19) and choose $j^* \in J_i$.
5: Set $b_{i,j}^{t+1} = 1$ if $j = j^*$ and $b_{i,j}^{t+1} = 0$ if $j \neq j^*$.
6: **end for**
7: Update μ^{t+1} using (20).
8: Set $t \leftarrow t + 1$.
9: **end for**

problem can be nicely transferred to the robustified settings that we discuss in the following sections.

5.3 An ADM for the strictly robust *k*-means clustering problem

In this section, we propose a tailored ADM to compute partial minima of Problem (6). First, we describe how we choose the set V. Consider the bounding box of the erroneous measures $\Pi(\tilde{X})$ as in Sect. 5.2. Thus, $\tilde{x}^i, \mu^j \in \Pi(\tilde{X})$ holds for all $i \in N$, $j \in K$. With this, for all $i \in N,\ j \in K,\ l \in P$, we can set

$$|\tilde{x}_l^i - \mu_l^j| \leq (\tilde{x}_l^i - \mu_l^j)^2 \leq (\pi_l^+ - \pi_l^-)^2 =: M_l, \quad M := \max\{M_l : l \in P\}. \tag{21}$$

For being a solution of Problem (6), α needs to be minimized while satisfying (6b). Thus, we can conclude that M as defined in (21) is an upper bound for α, i.e., we can set

$$|\tilde{x}_l^i - \mu_l^j| \leq \alpha_{i,j,l} \leq M, \quad i \in N,\ j \in K,\ l \in P.$$

This leads to the compact sets

$$V_\alpha := \left\{\alpha \in \mathbb{R}^{n \times k \times p} : 0 \leq \alpha_{i,j,l} \leq M,\ i \in N,\ j \in K,\ l \in P\right\}, \tag{22}$$

$$V_\mu := \left\{\mu \in \mathbb{R}^{p \times k} : \mu^j \in \Pi(\tilde{X}),\ j \in K\right\}. \tag{23}$$

Finally, let $V := V_\mu \times V_\alpha$ be the Cartesian product of the compacts sets V_μ and V_α. Thus, the variable splitting is given by the non-empty and compact set (16) and V as just defined. Note that, with this variable splitting, we again do not have any coupling constraints. Now, we can apply Algorithm 1 to Problem (6).

Suppose that the continuous variables μ^0 and α^0 are given. In each iteration t, the first subproblem to be solved is the binary optimization problem

$$b^{t+1} \in \arg\min_{b \in U} \sum_{j \in K} \sum_{i \in N} b_{i,j} \|\tilde{x}^i - (\mu^j)^t\|_2^2,$$

in which we already skipped all the constant terms in the objective function. This problem can be solved in analogy to Problem (18). Suppose that we have computed b^{t+1}. The new iterates μ^{t+1} and α^{t+1} are computed by solving

$$\min_{(\mu,\alpha) \in V} \quad \sum_{j \in K} \sum_{i \in N} b_{i,j}^{t+1} \left(\|x^i - \mu^j\|_2^2 + \|\Delta d^i\|_2^2 + \sum_{l \in P} 2\, \Delta d_l^i\, \alpha_{i,j,l} \right) \tag{24a}$$

$$\text{s.t.} \quad -\alpha_{i,j,l} \leq \tilde{x}_l^i - \mu_l^j \leq \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P. \tag{24b}$$

This is a convex quadratic optimization problem and can thus be efficiently solved by standard state-of-the-art solvers; see, e.g., Ben-Tal and Nemirovski (2001) and Boyd and Vandenberghe (2004) for an overview of algorithms and complexity results for convex problems. The overall method to compute a partial minimum of Problem (6) is formally given in Algorithm 3. As in the last section, Theorem 5.1 can be applied again, yielding that Algorithm 3 leads to partial minima of the strictly robust counterpart.

Algorithm 3 ADM applied to the strictly robust counterpart (6).

1: Choose $(\mu^0, \alpha^0) \in V$.
2: **for** $t = 0, 1, \ldots$ **do**
3: Compute b^{t+1} as in Steps 3–6 of Algorithm 2 w.r.t. $\tilde{x}^i$.
4: Update μ^{t+1} and α^{t+1} by solving problem (24).
5: Set $t \leftarrow t + 1$.
6: **end for**

5.4 An ADM for the Γ-robust *k*-means clustering problem

As in Sect. 5.3 we now first discuss how to obtain the set V for applying Algorithm 1 to Problem (9).

Note that in Problem (9), the continuous variables λ and β are always bounded in the direction of optimization since they are non-negative and minimized. Thus, there exist constants M_λ and M_β with

$$0 \leq \lambda \leq M_\lambda, \quad 0 \leq \beta_l^i \leq M_\beta, \quad i \in N,\ l \in P.$$

Now, we consider the following non-empty and compact sets

$$V_\lambda := \{\lambda \in \mathbb{R} \colon 0 \leq \lambda \leq M_\lambda\}, \quad V_\beta := \left\{\beta \in \mathbb{R}^{p \times n} \colon 0 \leq \beta_l^i \leq M_\beta,\ i \in N,\ l \in P\right\}.$$

With these sets and the sets in (22) and (23) at hand, we can define the non-empty and compact set V as

$$V := V_\mu \times V_\alpha \times V_\lambda \times V_\beta. \tag{25}$$

Thus, the variable splitting is represented by the sets (16) and (25). In this case now, we have coupling constraints (9b). In the notation of (15), we define these constraints formally as $c : \mathbb{R}^{n_u + n_v} \to \mathbb{R}^{p \times n}$ with

$$c_{(l,i)}(b, \alpha, \lambda, \beta) = \lambda + \beta_l^i - \sum_{j \in K} b_{i,j} \left((\Delta d_l^i)^2 + 2\, \Delta d_l^i\, \alpha_{i,j,l} \right), \quad (l, i) \in P \times N.$$

In the following, we describe how we solve the ADM subproblems. To solve Problem (9) into the direction of U, we first need to choose μ^0, α^0, λ^0, and β^0. Suppose that μ^0 is given, and we obtain b^0 as in Steps 3–6 of Algorithm 2 w.r.t. $\tilde{x}^i$. Then, we can compute α^0, λ^0, and β^0 by solving the linear optimization problem

$$\min_{(\alpha, \lambda, \beta) \in V} \quad \Gamma\lambda + \sum_{(l,i) \in P \times N} \beta_l^i \tag{26a}$$

$$\text{s.t.} \quad c_{(l,i)}(b^0, \alpha, \lambda, \beta) \geq 0, \quad (l, i) \in P \times N, \tag{26b}$$

$$-\alpha_{i,j,l} \leq \tilde{x}_l^i - (\mu_l^j)^0 \leq \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P. \tag{26c}$$

With the initial values for μ^0, α^0, λ^0, and β^0 at hand, the problem in the direction of U in iteration t reads

$$\min_{b \in U} \quad \sum_{j \in K} \sum_{i \in N} b_{i,j} \|\tilde{x}^i - (\mu^j)^t\|_2^2 \tag{27a}$$

$$\text{s.t.} \quad c_{(l,i)}(b, \alpha^t, \lambda^t, \beta^t) \geq 0, \quad (l,i) \in P \times N, \tag{27b}$$

where we already skipped all constant terms in the objective function.

Note that for $t = 0$, we obtain $b^1 = b^0$ by solving Problem (27). This is because b^0 is feasible for (27) and b^0 leads to the optimal objective value since it has been obtained as in Algorithm 2 w.r.t. $\tilde{x}^i$. Thus, there is no need to compute α^0, λ^0, as well as β^0 and we can directly proceed with the problem into the direction of V using b^0, as described below.

Suppose that μ^0 is given and b^0 is obtained as in Algorithm 2. Thus, in each iteration t, we compute μ^{t+1}, α^{t+1}, λ^{t+1}, and β^{t+1} by solving

$$\min_{(\mu,\alpha,\lambda,\beta) \in V} \quad \sum_{j \in K} \sum_{i \in N} b^t_{i,j} \|\tilde{x}^i - \mu^j\|_2^2 + \Gamma\lambda + \sum_{(l,i) \in P \times N} \beta^i_l \tag{28a}$$

$$\text{s.t.} \quad c_{(l,i)}(b^t, \alpha, \lambda, \beta) \geq 0, \quad (l,i) \in P \times N, \tag{28b}$$

$$-\alpha_{i,j,l} \leq \tilde{x}^i_l - \mu^j_l \leq \alpha_{i,j,l}, \quad i \in N,\ j \in K,\ l \in P, \tag{28c}$$

which is a convex quadratic optimization problem. Having this problem solved, we obtain the next b-iterate by solving Problem (27).

Algorithm 4 summarizes the main steps to compute a partial minimum of Problem (9). Again, Theorem 5.1 ensures that Algorithm 4 computes partial minima of the Γ-robust counterpart.

Algorithm 4 ADM applied to the Γ-robust counterpart (9)

1: Choose $\mu^0 \in V$.
2: **for** $t = 0, 1, \ldots$ **do**
3: **if** $t = 0$ **then**
4: Compute b^0 by following Steps 3–6 of Algorithm 2 w.r.t. $\tilde{x}^i$.
5: **else**
6: Compute b^t by solving Problem (27).
7: **end if**
8: Solve Problem (28) to obtain μ^{t+1}, α^{t+1}, λ^{t+1}, and β^{t+1}.
9: Set $t \leftarrow t + 1$.
10: **end for**

6 Restart heuristic

There are specific clustering instances for which the ADMs described so far obtain visibly bad results; see, e.g., the clustering obtained for the instance Unbalance in Fig. 1. For the details about the benchmark data sets we refer to Sect. 7.2. Observe that Algorithm 2 identified two clusters, S_1 and S_2, where there should be only one, and identified one cluster, S_3, where there should be two. This leads to the situation that the loss within cluster S_1 and within cluster S_2 is much smaller than the one within cluster S_3. Here and in what follows, we compute the weighted value of the loss function h in (2) restricted to cluster S_j by

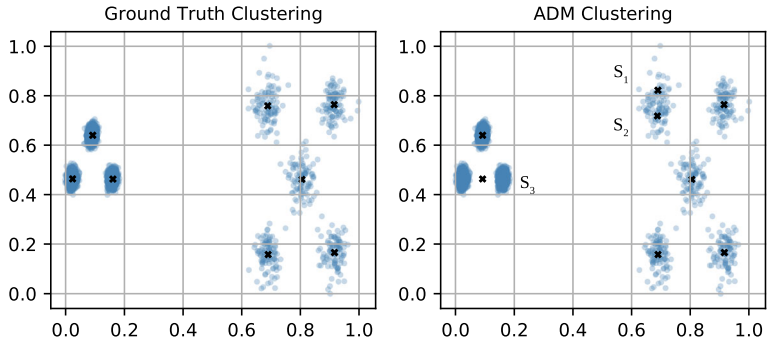

Fig. 1 Comparison between the ground truth clustering (left) and the clustering result of the ADM (right) for the instance Unbalance. Obviously, the partial minimum is of low quality

$$h_j = \frac{1}{|S_j|} \sum_{x^i \in S_j} \|x^i - \mu^j\|_2^2. \tag{29}$$

Suppose now that we join clusters S_1 and S_2 to form a new cluster S_{12}. If the total loss within cluster S_{12} is still smaller than the one within cluster S_3, then our algorithm is actually not minimizing the sum of total losses, because there exists a better point when we split cluster S_3 into two clusters and join clusters S_1 and S_2.

Based on these observations, we propose a heuristic to avoid that the ADMs in Algorithms 2–4 get stuck in a partial minimum of bad quality as shown in Fig. 1; see Fraiman et al. (2013) for a similar heuristic.

Suppose that we have a partial minimum and the h_j values at hand. For each pair of clusters (S_{j_1}, S_{j_2}), we also compute their joint center and the corresponding total loss via

$$\mu^{j_1 j_2} = \frac{1}{|S_{j_1}| + |S_{j_2}|} \sum_{x^i \in S_{j_1} \cup S_{j_2}} x^i \tag{30}$$

and

$$h_{j_1 j_2} = \frac{1}{|S_{j_1}| + |S_{j_2}|} \sum_{x^i \in S_{j_1} \cup S_{j_2}} \|x^i - \mu^{j_1 j_2}\|_2^2. \tag{31}$$

Now, consider the set

$$\Psi := \left\{(S_{j_1}, S_{j_2}, S_{j_3}) \colon h_{j_1 j_2} < h_{j_3}\right\}, \tag{32}$$

which is the set of all possible combinations of three clusters such that the total loss within two joined clusters is smaller than the total loss within a third cluster. Note that the set Ψ can be empty. If so, this means that we cannot obtain a better partial minimum by joining two clusters and splitting another one. On the other hand, i.e., if there exists $(S_{j_1}, S_{j_2}, S_{j_3}) \in \Psi$, then the total loss of the joined clusters S_{j_1} and S_{j_2} is smaller than the total loss within cluster S_{j_3}. Thus, we obtain a better partial minimum by joining S_{j_1} and S_{j_2} and by splitting cluster S_{j_3} into two smaller clusters. To this end, we update the centers in such a way that the clusters S_{j_1} and S_{j_2} are now one cluster with center $\mu^{j_1 j_2}$, cluster S_{j_3} receives two centers which are the two furthest points in S_{j_3}, and the other centers remain the same, i.e.,

$$\hat{\mu}^{j_1} \leftarrow \mu^{j_1 j_2}, \quad \hat{\mu}^{j_2} \leftarrow \bar{x}^i, \quad \hat{\mu}^{j_3} \leftarrow \bar{x}^{i'}, \tag{33}$$

$$\hat{\mu}^j \leftarrow \mu^j \quad \text{for all} \quad j \notin \{j_1, j_2, j_3\}, \tag{34}$$

with

$$(\bar{x}^i, \bar{x}^{i'}) \in \underset{x^i, x^{i'} \in S_{j_3}}{\arg\min} \left\{ \|x^i - x^{i'}\|_2^2 \right\}. \tag{35}$$

Finally, if the set Ψ has more than one element, then we repeat the process starting with the element $(S_{j_1}, S_{j_2}, S_{j_3})$ that gives the minimum ratio $h_{j_1 j_2}/h_{j_3}$. Each time an element $(S_{j_1}, S_{j_2}, S_{j_3})$ is used, we exclude all the elements that contain S_{j_1}, S_{j_2}, or S_{j_3}, because these clusters were already modified.

With $\hat{\mu}$ at hand, we compute a new partial minimum using $\hat{\mu}$ as the initial centers. Given the new partial minimum, we repeat the process until a termination criterion is satisfied. In Algorithm 5, we formally state the proposed heuristic as an additional part of Algorithms 2–4. The termination criterion in our implementation is to stop if the restart heuristic does not provide modified centers anymore and, as a consequence, if two consecutive partial minima are the same (up to numerical tolerances).

Algorithm 5 Improved algorithms 2, 3, and 4.

Given initial centers μ^0, compute a partial minimum with Algorithm 2, 3, or 4.
if termination criterion is not satisfied **then**
  Compute h_j as in (29) for all clusters $j = 1, \ldots, k$.
  Compute $\mu^{j_1 j_2}$ as in (30) and $h_{j_1 j_2}$ as in (31) for all $j_1 \neq j_2 \in \{1, \ldots, k\}$.
  Compute the set Ψ as in (32).
  if $\Psi = \emptyset$ **then**
    return the current partial minimum.
  else
    while $\Psi \neq \emptyset$ **do**
      Compute $(j_1, j_2, j_3) \in \arg\min\{h_{j_1 j_2}/h_{j_3} : (j_1, j_2, j_3) \in \Psi\}$.
      Compute $\hat{\mu}$ as in (33)–(35).
      Update Ψ by deleting all triples that contain either j_1, j_2, or j_3.
    end while
    Set $\mu^0 := \hat{\mu}$ and go to Step 1.
  end if
end if

7 Numerical results

In this section, we present our numerical studies, which have been carried out using the setup described in Sect. 7.1. The goal here is to evaluate the performance of the proposed methods in Algorithms 2–5. To this end, we first describe in Sect. 7.2 the clustering problem instances and the evaluation metrics that we use. For analyzing the performance of the different methods, a ground truth labeling has to be defined. These labels are provided with the test data sets used in the numerical evaluation. For the optimization, these ground truth labels are not used as k-means clustering is an unsupervised learning method, i.e., the input data is not labeled. Since all algorithms assume that initial centers are given, we briefly describe how they are obtained in Sect. 7.3. In Sect. 7.4, we show that the proposed ADM performs very well for typical nominal clustering problems. Moreover, we also evaluate the effectiveness of the heuristic described in Sect. 6. In Sects. 7.5 and 7.6, we discuss the results that we obtain by

robustifying these k-means clustering problems. Finally, in Sects. 7.7 and 7.8, we go further and evaluate the performance of the proposed algorithms on real-world clustering instances.

As we will see, the strictly robust method clearly outperforms the nominal and Γ-robust methods. In the appendix, we also present a comparison of the strictly robust method with the spectral clustering method by Von Luxburg (2007), which is another state-of-the-art method for clustering. By these results, the competitiveness of the strictly robust method is further highlighted.

7.1 Software and hardware setup

We implemented the algorithms in Python 3.8.5 and solved the binary linear as well as the convex quadratic optimization problems with Gurobi 9.1.0. We use the special Python modules sklearn.preprocessing.MinMaxScaler, sklearn.metrics.adjusted_rand_score, sklearn.metrics.silhouette_score, and scipy.stats.wilcoxon to scale the data to the range [0, 1], to compute the ARI (see below), to compute the Silhouette score (see below), and to calculate the p values for the Wilcoxon signed-rank test (see below as well), respectively. All the computations were performed on a computer with a 3.60 GHz Intel(R) Core(TM) i3-8100 processor and 16 GB RAM. For Algorithms 2–4, the termination criterion is $\|\mu^{t+1} - \mu^t\|_\infty < 10^{-4}$.

7.2 Data sets and validation metrics

To comprehensively compare the proposed algorithms on a variety of clustering problems, we use (i) synthetically generated data for which we include some uncertainty by perturbing the data points as well as (ii) real-world data sets, which naturally contain measurement errors.

Since one of the goals of this work is to identify structural properties of clustering problems for which a robustification is beneficial, we use six synthetic clustering benchmark data sets (a, s, Unbalance, dim, g2, birch) proposed by Fränti and Sieranoja (2018). These data sets provide a perfect benchmark since they cover a wide range of typical k-means clustering problems with different degrees of overlap, density, and sparsity. Besides that, most of the data sets are two-dimensional, so we can also visualize and compare the recovered nominal and robust clustering results. For reasons of comparability, we use these data sets as a reference but exclude the data set birch with 100 clusters that leads to extremely large and challenging problems.

It is also standard in the literature to test clustering algorithms on data sets from the UCI Machine Learning Repository (Dua and Graff 2017); see, e.g., Li et al. (2016), Vo et al. (2016) and Aloise et al. (2012). Thus, to also evaluate the quality of robust solutions on high-dimensional and real-world data sets, we select 52 instances from this repository, forming a sample of clustering problems with diverse sizes and difficulties.

For the synthetic data, the ground truth partition of all data sets is publicly available in Fränti and Sieranoja (2018) and for the real-world instances, the true assignments (labels) of each data point are given as well. Thus, we choose as external validity metrics the adjusted Rand index (ARI) and the loss function value. The ARI (Hubert and Arabie 1985; Steinley 2004) measures the similarity between two assignments, ignoring permutations, within the range $[-1, 1]$. Here, 1 means perfect agreement between the two assignments.

However, in some cases, the ground truth partition corresponds to lower quality solutions, which may lead to a subjective validation if only external metrics are used as reference. Therefore, we also use the Silhouette score as an internal validation metric (Rousseeuw

1987). This score is shown to have a good performance for general clustering problems in an extensive study over a variety of validation metrics done in Arbelaitz et al. (2013). The Silhouette score measures the quality of the clustering in terms of cohesion and separation, taking values within the interval $[-1, 1]$, where a value close to 1 means that the data set is well clustered, whereas a result close to -1 means that most of the data points have been misclassified or that there is no natural clustering structure in the data set.

Further, to assess the significance of the obtained results, we apply the Wilcoxon signed-rank test to calculate p values (Wilcoxon 1945). With this, we are able to quantify the observed improvements in cluster quality when comparing the Silhouette score as well as the ARI of nominal and robust models. If the resulting p value is smaller than the α-error of 10 %, we reject the null-hypothesis that the robust model is less or equally good as the nominal one in terms of the Silhouette score or the ARI, respectively. Hence, a rejection supports the hypothesis that we obtain improved clustering results for erroneous data when using robustified models.

7.3 Starting point heuristic

It is well known that a poor initialization of the centers can cause a bad clustering result. In Fränti and Sieranoja (2019), the authors study some popular initialization heuristics and test them on a benchmark library, which is the same synthetic data that we use to test our algorithms. On average, they conclude that the "furthest point heuristic", also known as "Maxmin", reduces the clustering error of k-means. Based on their results, we decide to compute the initial mean points μ^0 with the "Maxmin" heuristic. The idea is to select the first center randomly within the respective bounding box, and then obtain new centers one by one. In each iteration, the next center is the point that is the furthest (max) from its nearest (min) existing center.

7.4 Evaluation of the ADM applied to the synthetic nominal clustering problems

In this section, we discuss the performance of Algorithm 2 and its extension in Algorithm 5 applied to the synthetic data sets. To simplify the presentation here, we refer to Algorithm 2 as ("ADM2") and to Algorithm 5 as ("ADM5"). As already mentioned at the end of Sect. 5.2, the ADM in Algorithm 2 exactly mimics the classic k-means clustering algorithm. Nevertheless, we present the numerical results for the nominal cases here as well to assess the effects of the restart heuristic and to have a proper baseline for a computational comparison of the robustified versions later on. First, we evaluate the quality of the partial minima computed by ADM2 and ADM5 for 12 instances from the synthetic data sets. The results are shown in Table 1. The objective function value is denoted by $h(X, \mu)$. We both state the value of h corresponding to the ground truth ("GT") and the value of h in a partial minimum computed with ADM2 and ADM5. The Silhouette score is computed for the ground truth assignment as well as for the ADM2 and ADM5 assignment results. The two last columns show the ARI, respectively. In Table 2, we state the size of the clustering problem, i.e., the number of clusters k, the number of points n, and the number of attributes p. There we also present the runtime ("Time"; in seconds) and the number of iterations (denoted by "It") that the ADM2 and ADM5 require to compute a partial minimum. Since the initialization heuristic for the centers is a random procedure, we apply the algorithm five times with different initializations and report the average over all five runs.

Table 1 Results of Algorithm 2 and its extended Algorithm 5 on the synthetic clustering data sets

Instance	$h(X, \mu)$			Silhouette			ARI	
	GT	ADM2	ADM5	GT	ADM2	ADM5	ADM2	ADM5
a1	6.945	7.685	6.747	0.566	0.544	0.572	0.890	0.954
a2	7.617	9.853	7.544	0.582	0.530	0.584	0.857	0.977
a3	6.995	8.434	6.992	0.601	0.566	0.601	0.927	0.997
s1	10.552	15.315	10.287	0.708	0.655	0.712	0.915	0.986
s2	16.098	20.725	14.929	0.609	0.559	0.626	0.826	0.938
s3	30.580	22.138	21.656	0.385	0.478	0.485	0.701	0.713
s4	34.544	20.437	20.125	0.321	0.466	0.468	0.602	0.608
Unbalance	4.247	22.913	4.247	0.833	0.667	0.833	0.612	1.000
dim	7.337	7.158	7.158	0.945	0.945	0.945	1.000	1.000
g2-2-30	42.065	41.479	41.479	0.625	0.629	0.629	0.961	0.961
g2-2-50	53.421	46.556	46.556	0.413	0.482	0.482	0.695	0.695
g2-2-70	60.112	49.540	49.540	0.278	0.397	0.397	0.487	0.487

Table 2 Information about the synthetic data sets and the performance of Algorithm 2 and its extended Algorithm 5 on these instances

Instance	Model size			ADM2		ADM5	
	k	n	p	Time	It	Time	It
a1	20	3000	2	2.2	15.8	5.4	31.6
a2	35	5250	2	7.7	18.8	14.4	28.2
a3	50	7500	2	14.8	18.2	31.8	32.2
s1	15	5000	2	1.6	8.6	3.3	12.0
s2	15	5000	2	3.2	17.8	6.0	26.4
s3	15	5000	2	4.5	25.4	7.5	36.6
s4	15	5000	2	7.9	44.8	9.9	34.2
Unbalance	8	6500	2	1.5	10.2	29.2	9.8
dim	16	1024	32	0.1	2.0	0.2	2.0
g2-2-30	2	2048	2	0.1	4.4	0.1	5.2
g2-2-50	2	2048	2	0.2	8.6	0.2	9.0
g2-2-70	2	2048	2	0.3	13.6	0.3	13.8

Looking at the results of ADM2 first, we note that for the instances a1, a2, a3, s1, s2, dim, and g2-2-30, on average, the value of h and the Silhouette score in the partial minimum are close to the ground truth and the ARI is also close to 1. This shows that the clusterings of the partial minima are similar to the ground truth clusterings for these instances. One can also see that the quality of the partial minima measured in terms of the objective function h and in terms of the Silhouette score can both be worse (e.g., instances s1 or s2) or better (e.g., instance g2-2-30) compared to the ground truth.

On the other hand, for the instances s3, s4, Unbalance, g2-2-50, and g2-2-70, on average, the ARI reveals that the corresponding clusterings have low similarity and also the value of h in the partial minima is rather different to the ground truth. However, the Silhouette score is better and the value of h is smaller for all these cases except for the instance Unbalance for

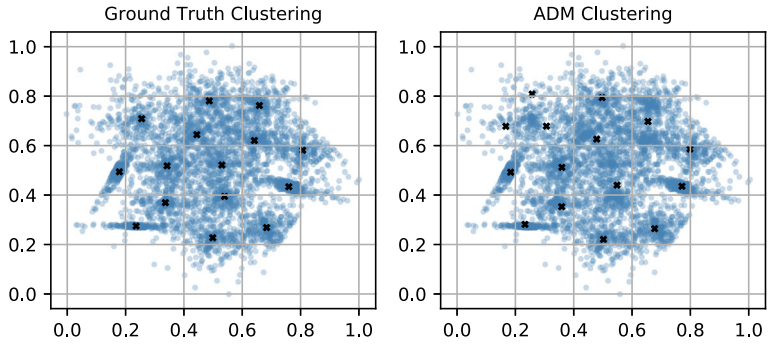

Fig. 2 The ground truth clustering (left) and the ADM5 result (right) for the instance s4

which we obtain a partial minimum of very bad quality; see also Fig. 1. Due to this, in Sect. 6, we proposed a heuristic to improve the quality of the partial minima that are computed by the ADM2; see Algorithm 5.

Regarding the runtime and number of iterations, the ADM2 is very fast in computing partial minima. The median average runtime is 1.9 s (with a maximum of less than 15 s) and the ADM2 never requires more than 45 iterations on average.

Now, looking at the results of ADM5, we note that the extended algorithm is able to find the ground truth clustering of the instance Unbalance. Furthermore, all the results improved in terms of the objective function h and in terms of the Silhouette score. We also observe that the results for all g2 instances (all having $k = 2$) remain the same because the heuristic proposed in Sect. 6 is only applicable for $k > 2$.

Let us briefly discuss the instance s4 for which we get the lowest ARI (0.608). In this case, the objective function value is significantly smaller in the ADM5 result (compared to the ground truth; 20.125 vs. 34.544), and also the Silhouette score is better (compared to the ground truth; 0.468 vs. 0.321). Both results are shown in Fig. 2, where it is easy to see that the clusterings also qualitatively differ from each other. Nevertheless, the low values of the Silhouette score reveal that this is a difficult instance for clustering.

Regarding the runtime and number of iterations that the ADM5, on average, requires to find a partial minimum, both increase due to the restart heuristic. This is expected, because in each "outer" iteration we may have new initial centers and, if this is the case, we apply the ADM2 again leading to additional inner ADM iterations. Thus, the computational costs of getting better partial minima is the increase of median average runtime from 1.9 to 5.7 s (with a maximum of 32 s for the instance a3).

We conclude that the proposed Algorithm 2 combined with the restart heuristic proposed in Algorithm 5, i.e., the combination that we denote here by ADM5, performs very well on the synthetic data sets. The algorithm is able to find partial minima of very good quality with objective function values close to or even better than the ground truth and Silhouette scores always better than or equal to the ground truth.

7.5 Strictly robust clustering results for the synthetic data

Let us start the discussion of the strictly robust clustering results with an informal note. To this end, consider the clustering instance Unbalance in Fig. 1 again. For this instance, it is obvious that the eight clusters are very well separated and not too close to each other. Only for a few

points between the two clusters below and the middle cluster in the right part of the figure it may be not clear to which cluster they belong. However, the specific assignment decision for these data points will not qualitatively change the overall clustering result. In such a setting, it cannot be expected that a slight perturbation of a certain percentage of data points by a certain amount will lead to a qualitatively different result. Even if all data points are perturbed by, e.g., 1 %,[1] the clustering will stay the same although the clusters may be blurred a bit. This general observation is also supported by our preliminary numerical experiments: Clustering of well separated sets of data points does not benefit from robustification. Moreover, in such a situation there is also no need for robustified clustering if compared with the nominal results.

Due to this and to the fact that most of the instances from the twelve synthetic data sets provided in Fränti and Sieranoja (2018) are practically well separated, we induce some uncertainty in the data points by perturbing them. More precisely, we consider an amount of perturbation of 10 % and different quantities of perturbed data points: 5 %, 30 %, and 50 %. Assuming now that the perturbed data set is the "new nominal" or given one, we apply the nominal and the two robust models. Each data set is perturbed 10 times and every time the three approaches are applied to an instance, they start with the same given centers. The average results for all data sets over the 10 runs each are presented in Table 3, where we show the Silhouette score as well as the ARI regarding the clustering results on the perturbed data compared to the nominal ADM solution on the unperturbed set (notice that we do not compare to the ground truth assignment here). For each instance and each validation metric, the best average result is printed in bold. Moreover, since in some cases the differences in terms of Silhouette and ARI among methods seem to be modest, we underline the best robust average results for which the corresponding p values (calculated for each metric and each two methods) are smaller than the α-error of 10 % w.r.t. the nominal results. Therefore, the underlined average results are statistically significantly better on a 10 % level. All the calculated p values are shown in the appendix in Table 10.

It can be directly seen that the advantage of the strictly robust model over the nominal and Γ-robust models increases with the amount of perturbed data points. This is the case, e.g., for the instance Unbalance. As we induce a stronger perturbation, the clusters significantly mix with each other and the performance of the strictly robust model becomes the best in both evaluation metrics.

Let us now discuss the quality of the robustified clusterings for the instances s3, s4, a2, and a3 for which the data points are not well separated or for which the clusters are close to each other; see Fig. 3 (all other data sets are given in Fig. 8 in the appendix). To this end, we consider Fig. 4 in which we display box-plots to analyze the quality of the robustified clustering results. On the y-axes we plot the ARI regarding the strictly robustified clustering results on the perturbed data compared to the nominal ADM solution on the unperturbed set divided by the ARI regarding the nominal ADM clustering results on the perturbed data compared to the nominal ADM solution on the unperturbed set. All ADMs always have been applied using the restart heuristic described in Sect. 6.

For all four instances we see that the quality of the strictly robustified clustering is improved if the errors (in terms of the amount of perturbed points) get larger. The detailed behavior also depends on the specific instance. Thus, we discuss two exemplary ones. For a3 we see that the robustified clustering is advantageous only for the case of 50 % perturbed data points. Taking a look at Fig. 3 again we see that the data points in instance a3 are rather well separated but the clusters are close to each other. Since all points are perturbed by 10 % if they are perturbed

[1] Note that before the computations the data is scaled to $[0, 1]^p$ and the amount of perturbation will be stated in percent for better readability.

Table 3 Nominal and robust results on the perturbed synthetic data sets

Instance	Silhouette			ARI		
	Nominal	Strictly	Γ	Nominal	Strictly	Γ
a1	**0.5592**	0.5567	0.5511	**0.9609**	0.9520	0.9256
a2	**0.5684**	0.5683	0.5665	0.9583	**0.9597**	0.9491
a3	0.5776	0.5726	**0.5791**	0.9364	0.9261	**0.9375**
s1	0.6934	0.6934	0.6918	**0.9905**	0.9899	0.9866
s2	0.6119	**0.6120**	0.6096	**0.9785**	0.9779	0.9612
s3	**0.4662**	0.4645	0.4432	**0.8956**	0.8813	0.8059
s4	0.4556	**0.4612**	0.4326	**0.7898**	0.7734	0.6875
Unbalance	0.8026	0.8026	0.8002	0.9741	**0.9744**	0.9723
dim	0.9332	0.9332	0.9332	1.000	1.000	1.000
g2-2-30	0.6266	0.6266	0.6147	0.9926	0.9926	0.9116
g2-2-50	0.4810	0.4810	0.4710	**0.9794**	0.9746	0.8699
g2-2-70	0.3968	**0.3971**	0.3892	**0.9627**	0.9398	0.7993
a1	0.4964	**0.4977**	0.4584	0.7937	**0.7999**	0.7141
a2	0.4846	**0.4874**	0.4741	0.7379	**0.7538**	0.7180
a3	**0.4894**	0.4882	0.4867	**0.6928**	0.6914	0.6910
s1	0.6075	0.6075	0.5734	0.9407	**0.9425**	0.8783
s2	0.5371	**0.5403**	0.5163	0.8583	**0.8684**	0.8105
s3	**0.4267**	0.4264	0.3830	0.7514	**0.7541**	0.6204
s4	0.4218	**0.4225**	0.3477	0.6362	**0.6465**	0.5070
Unbalance	0.5815	**0.5822**	0.5455	0.6063	**0.6123**	0.4885
dim	0.8831	0.8831	0.8831	1.000	1.000	1.000
g2-2-30	0.6117	0.6117	0.5566	**0.9617**	0.9613	0.7324
g2-2-50	0.4732	0.4732	0.3625	**0.8777**	0.8758	0.3362
g2-2-70	0.4002	0.4002	0.3506	**0.8199**	0.8128	0.4087
a1	0.4435	**0.4461**	0.4091	0.6621	**0.6709**	0.5993
a2	**0.4367**	0.4360	0.4106	0.6147	**0.6150**	0.5665
a3	0.4386	0.4386	0.4237	0.5492	**0.5526**	0.5362
s1	0.5469	0.5469	0.5110	0.9020	**0.9059**	0.8311
s2	0.4795	**0.4868**	0.4429	0.7504	**0.7745**	0.6832
s3	0.3977	**0.4009**	0.3575	0.6337	**0.6546**	0.5333
s4	0.3958	**0.3984**	0.3676	0.5650	**0.5798**	0.5111
Unbalance	0.5082	**0.5352**	0.4599	0.5056	**0.6041**	0.4870
dim	0.8479	0.8479	0.8479	1.000	1.000	1.000
g2-2-30	0.6013	0.6013	0.5519	**0.9353**	0.9351	0.7221
g2-2-50	0.4695	0.4695	0.3596	**0.8095**	0.8084	0.2724
g2-2-70	0.4035	**0.4036**	0.2797	**0.7179**	0.7138	0.1496

The perturbation amount is 10 %, and the quantity of perturbed data points is different in each block of the table: the top block corresponds to 5 %, the middle block to 30 %, and the bottom block to 50 %. The Γ values are exactly the number of perturbed data points for each instance. For each instance and each validation metric, the best average result is printed in bold. The best average result that is significantly better than the nominal one, according to the calculated p value, is underlined

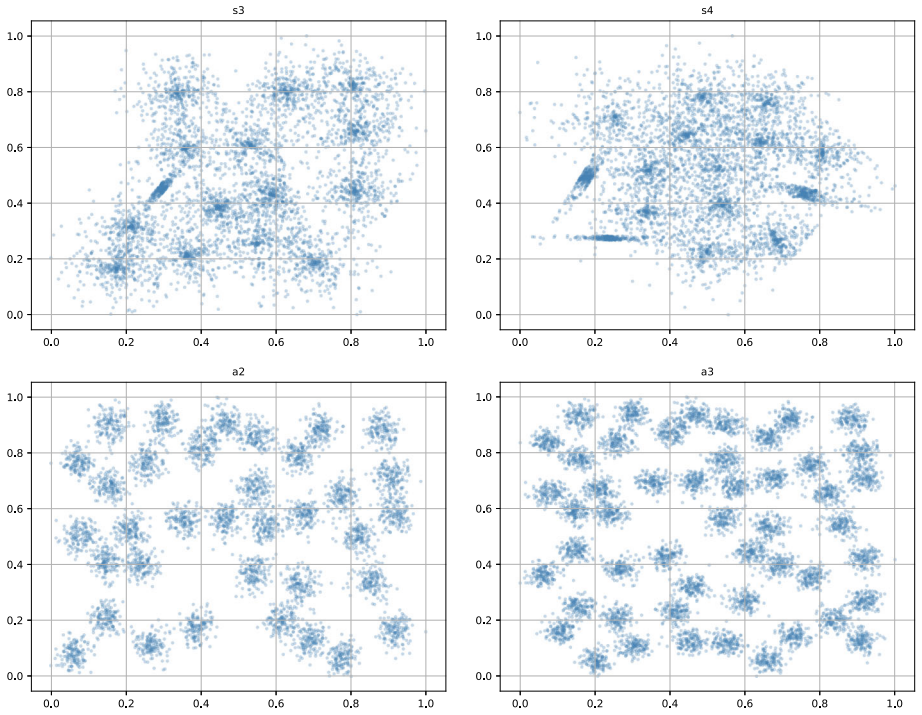

Fig. 3 The four test instances for the strictly robustified clustering method

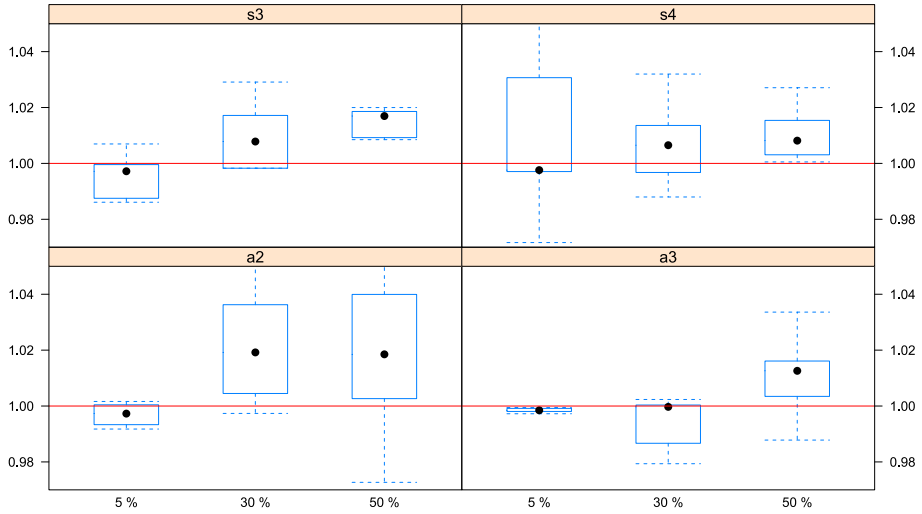

Fig. 4 Statistical performance of the strictly robust vs. the nominal k-means clustering methods measured as the ratio of the respective ARIs. Values over 1 indicate higher ARI for the strictly robust than for the nominal k-means clustering

Table 4 Average runtime and iteration counts for the corresponding ADMs with restart heuristic for the nominal, the strictly, and the Γ robust models on the perturbed (10 %) synthetic data sets

Instance	Time (s)			Iterations		
	Nominal	Strictly	Γ	Nominal	Strictly	Γ
a2	25.2	121.0	289.6	54.6	50.1	51.5
a3	56.7	216.7	575.0	59.2	47.9	54.8
s3	8.2	50.3	85.5	43.2	29.7	23.5
s4	7.8	94.4	89.4	31.9	46.9	22.6
Unbalance	23.2	47.0	62.4	9.3	8.7	8.6
a2	26.2	130.6	278.3	54.7	53.9	50.3
a3	71.0	296.3	733.3	72.5	69.8	74.3
s3	7.9	51.7	89.5	40.1	30.0	26.1
s4	12.4	95.4	103.1	52.4	57.2	30.9
Unbalance	8.0	36.3	44.0	13.5	12.4	10.1
a2	25.1	124.2	207.6	56.3	50.2	37.0
a3	53.8	239.4	551.0	58.3	55.3	55.6
s3	6.8	53.4	86.8	35.6	32.5	25.7
s4	9.3	78.0	91.1	48.5	53.7	27.2
Unbalance	8.7	51.2	60.2	14.2	17.3	9.3

The blocks are separated according to different amounts (5, 30, 50 %) of randomly perturbed data points, respectively

at all, a robustification is getting beneficial in the case that enough clusters mix with each other. This does not seem to be the case for 30 % of perturbed points but is the case for 50 %. We observed a similar behavior in our computations if the number of perturbed points stays the same and if the points are perturbed by a varying amount like 10, 20, or 30 %.

As a second example let us consider instance s4; see also Fig. 3. Here, all clusters are not well separated. In such a situation, already a smaller amount of perturbed points leads to a beneficial robustification but for larger amounts of perturbed points no significant further improvements can be seen. If we take a look at Table 3 again, we see that this is also the case for the Silhouette score. The reason most likely is that for 30 % of perturbed points the clusters significantly mix with each other and that this mix is not qualitatively increased for more perturbed points.

Now we discuss the computational performance of the robust clustering methods for the exemplary instances previously discussed, see Table 4 (all the other results are presented in Table 9 in the appendix), where we report the average (over ten runs with different random perturbations) runtimes and iteration counts for the ADM with restart heuristic applied to the nominal and the two robustified clustering models. (The Γ columns will be discussed in the next section.) It is clearly visible that the solution of the strictly robust clustering problems takes more time: the average factor is 5.9 with minimum 2.0 and maximum 12.1. Moreover, the percentage of perturbed points does not have a clear impact on the runtime. In contrast to runtimes, the number of required iterations does not change significantly and there are almost as many instances with decreased average iteration counts as there are increased iteration counts. Consequently, the main additional computational work is due to harder subproblems that need to be solved in every iteration of the ADM.

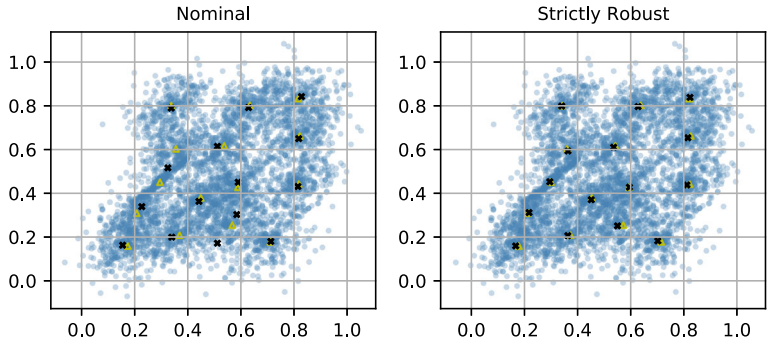

Fig. 5 Nominal versus strictly robustified clustering results for the instance s3

Thus, the bottom line is as follows. The strictly robustified clustering method is harder to compute but the additional computational effort is reasonable since we roughly stay in the same order of magnitude compared to the runtime of the nominal method. Regarding the outcome of the robustified clustering, the results always have to be analyzed with the structure of the measured data points in mind, since different structures in the observed data may lead to different robustified results as the two exemplary discussed instances show.

Let us close this section on the strictly robustified clustering method with a final numerical result for the instance s3. In Fig. 5, the green triangles are the same in the left and the right plot of the figure and represent the clustering obtained by the ADM including the restart heuristic when applied to the unperturbed data set. Then, we perturb 50 % of all points by 10 % and again solve the ADM with restart heuristic considering these perturbed data points as the given ones. This results in the black (crosses) centers in the left plot of Fig. 5. It is obvious, that by considering the perturbed points as "new nominal" ones, one is not able to detect the original centers of the clusters. In contrast, the strictly robustified clustering result (right plot in Fig. 5) almost perfectly recovers the original centers.

7.6 Γ-robust clustering results for the synthetic data

We start with the analysis of the computational performance of the Γ-approach; see Table 4 again. First, one can observe that the number of iterations is again in the same order of magnitude as the number of iterations required to solve the nominal as well as the strictly robust clustering problems. Interestingly, the Γ-approach requires fewer iterations for 11 out of 15 instances. However, the computation times are, on average, a factor of 9.4 longer then for the nominal problems with a minimum factor of 2.7 and a maximum factor of 12.8. This means that the computation times for the Γ-robustified models are roughly two times longer than the ones for the strictly robust versions.

The motivation for Γ-robustness in Bertsimas and Sim (2004) was to introduce a concept that leads to a less conservative notion of robustness—especially compared to strict robustness. This less conservative concept had many applications and was also shown to be useful for classification in Bertsimas et al. (2019), which is, to the best of our knowledge, the first and only application of Γ-robustness in classification so far.

The question now is whether a Γ-robustification is also beneficial in the context of clustering problems. Unfortunately, an analysis comparable to the one given in Fig. 4 does not show a benefit in general; see Table 3 again. Our analysis of the results revealed two major explanations for this. First, the aim of the Γ-approach is to hedge against worst-case realizations

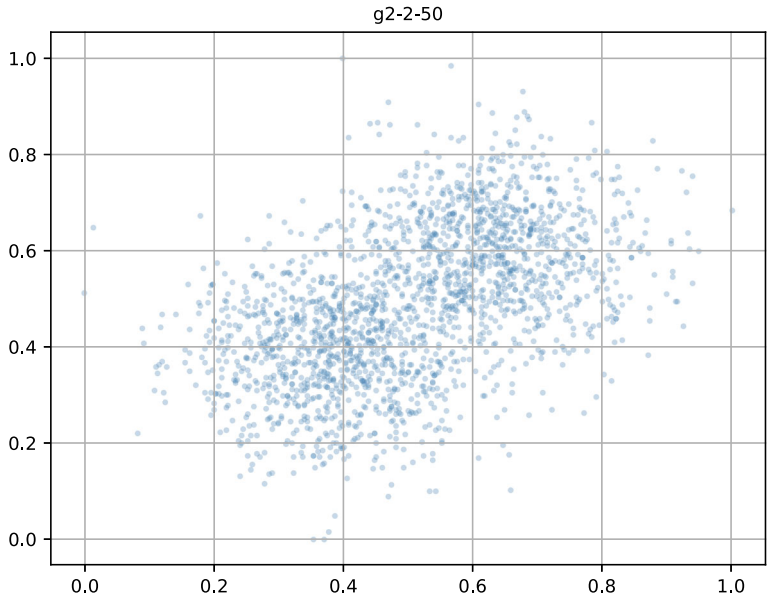

Fig. 6 The instance g2-2-50

in Γ many parameters with the specific parameters being unknown beforehand. In particular, these Γ many parameters do not need to correspond to those, e.g., 30 % of the parameters that are randomly chosen and perturbed in our simulations. Thus, it is likely to be the case that the randomly perturbed ones are not the ones that the Γ-approach hedges against. Let us discuss this on the example of instance g2-2-50, which is shown in Fig. 6. Here, the two clusters are not well separated. The interesting data points for the Γ-approach most likely are the ones near the diagonal of the plot. Thus, only a small percentage of points can be expected to be the points for which a perturbation might lead to a beneficial behavior of the Γ-approach. In Fig. 7, we analyze the performance of the Γ-approach. It can be seen that the Γ-approach can lead to an improvement compared to the nominal and to the strictly robust approach—but only if the parameter Γ is chosen appropriately; see, e.g., the case in which Γ corresponds to 0.1 % of the data points. Unfortunately, it is not clear on how to choose this parameter appropriately in advance.

Let us also briefly comment on the results reported on beneficial Γ-robustifications in case of SVMs in Bertsimas et al. (2019). There, the authors apply this approach in the case in which the labels are subject to uncertainty, which is not the situation here since no labels are considered. In the case that Bertsimas et al. (2019) call "feature-robust", which corresponds to our setting here, they also use strictly robustified models.

Figure 7 also makes another issue visible, namely that the Γ-robustified results do not tend to the strictly robust solutions if Γ is getting larger. In theory, however, this should be the case. Our experiments revealed that this is due to non-optimal partial minima computed by the ADM in Algorithm 4—although we present here the results for which the restart heuristic is used as well. This problem is not visible for the strictly robustified problems that we also solve using a tailored ADM, which might be simply explained by the overall increased complexity of the Γ-robustified models compared to the nominal or strictly robust models.

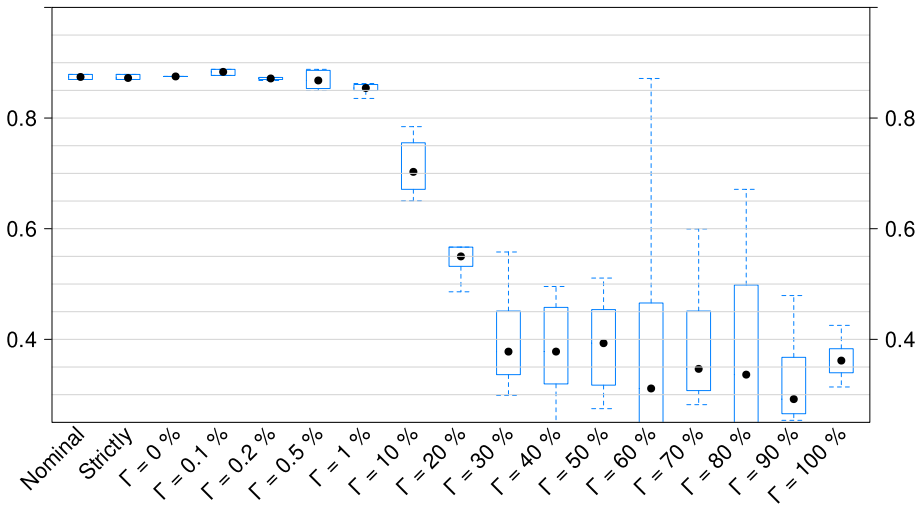

Fig. 7 Performance of the Γ-robustified vs. the nominal k-means clustering method measured as the ratio of their respective ARIs

To sum up, we found an interesting case in which less conservatism does not lead to better results—as it is often expected in robust optimization in general. Moreover, the advantageous clustering results based on the strictly robust models are even more pronounced because these models are also significantly easier to solve; see Table 4 again. Thus, if no specific value of Γ is available, we strongly suggest to use the strictly robustified clustering method for data sets with unstructured errors. Therefore, in Sect. 7.8, we only consider the strictly robustified model for the real-world data sets.

7.7 Evaluation of the ADM applied to nominal real-world clustering problems

In this section, we evaluate the performance of Algorithm 2 with the restart heuristic applied to a sample of 52 real-world data sets taken from Dua and Graff (2017). Considering the starting point procedure described in Sect. 7.3, the ADM is applied five times considering different initial centers. The average (over the five runs) runtimes and iterations counts are present in Table 5, where we also give information on the size of the instances. Again, as seen for the synthetic data, the ADM is very fast in computing partial minima. The median average runtime is 0.1 s and the maximum number of required iterations is, on average, less than 57. Interestingly, the ADM is significantly faster for these real-world instances compared to the synthetic data sets. Thus, the proposed method seems to scale very well.

Let us now discuss the quality of the partial minima. To this end, we consider again the ARI as external validation metric since the true labels are available and we also report the Silhouette score to evaluate how good the clustering is in terms of separation and cohesion. The average results over the five runs are shown in Table 6, where we additionally compute the Silhouette score for the true labels. First, we can see that for some instances the clustering result is very bad. Negative values for the ARI mean that the ADM assignment result has little similarity with the true assignment. However, taking a closer look at the Silhouette score in the true labels, the negative (or very small) values of the ARI happen exactly when this score is also very small. The most likely explanation for this is that the data set simply does

Table 5 Computational performance of Algorithm 2 using the restart heuristic on the real-world clustering data sets

Data set information				Time (s)	Iterations
Instance	n	p	k		
balance-scale	625	4	3	0.1	13.4
banknote-authentication	1372	4	2	0.2	18.2
blood-transfusion	748	4	2	0.1	12.0
breast-cancer	683	9	2	0.0	6.2
breast-cancer-diagnostic	569	30	2	0.1	11.4
breast-cancer-prognostic	194	32	2	0.0	13.2
climate-model-crashes	540	18	2	0.1	15.6
connectionist-bench	990	10	11	1.2	29.8
connectionist-bench-sonar	208	60	2	0.0	5.4
contraceptive-method-choice	1473	9	3	0.8	28.4
dermatology	358	34	6	0.2	18.2
ecoli	336	7	8	0.1	14.0
fertility	100	9	2	0.0	4.2
glass-identification	214	9	6	0.0	6.2
haberman	306	3	2	0.0	8.2
heart-disease-cleveland	297	13	5	0.1	8.4
hepatitis	80	19	2	0.0	6.0
hill-valley	606	100	2	0.0	5.0
hill-valley-noise	606	100	2	0.0	5.8
image-segmentation	210	19	7	0.1	8.8
ionosphere	351	34	2	0.0	5.2
iris	150	4	3	0.0	7.8
libras-movement	360	90	15	0.2	11.2
magic-gamma-telescope	19020	10	2	4.5	25.0
mammographic-mass	830	5	2	0.0	5.0
monks-1	124	6	2	0.0	7.6
monks-2	169	6	2	0.0	10.2
monks-3	122	6	2	0.0	6.8
optical-recognition	3823	64	10	9.2	56.2
ozone-level-eight	1847	72	2	0.3	17.8
ozone-level-one	1848	72	2	0.3	17.0
parkinsons	195	21	2	0.0	18.2
pen-based-recognition	7494	16	10	9.1	35.0
planning-relax	182	12	2	0.0	12.2
qsar-biodegradation	1055	41	2	0.1	7.0
seeds	210	7	3	0.0	6.4
seismic-bumps	2584	14	2	0.1	3.0
soybean-large	266	35	15	0.1	5.8
soybean-small	47	35	4	0.0	4.8
spambase	4601	57	2	0.2	4.0

Table 5 continued

Data set information				Time (s)	Iterations
Instance	n	p	k		
spect-heart	80	22	2	0.0	4.4
spectf-heart	80	44	2	0.0	2.4
statlog-german-credit	1000	24	2	0.1	6.8
statlog-landsat-satellite	4435	36	6	6.4	48.6
teaching-assistant-evaluation	151	5	3	0.0	5.4
thyroid-disease-ann	3772	21	3	0.6	7.0
thyroid-disease-new	215	5	3	0.0	5.8
wall-following-robot-2	5456	2	4	2.9	16.2
wall-following-robot-4	5456	4	4	0.8	10.0
wine	178	13	3	0.0	6.6
yeast	1484	8	10	1.2	27.6
zoo	101	16	7	0.0	3.6

not have an underlying clustering structure and, consequently, does not fit well the k-means clustering idea.

In contrast to this, when the Silhouette score in the true labels is larger than, e.g., 0.2, then the ARI is also closer to 1 in general. Thus, the ADM assignment result is more similar to the true labeling. In other words, if the real-world data set possesses an underlying clustering structure, then the proposed ADM combined with the restart heuristic performs very well and is able to find partial minima of good quality.

7.8 Strictly robust clustering results on real-world data sets

As seen in the last section, several of the real-world data sets among the 52 instances turned out to be instances that are not suitable for the k-means clustering method because they do not possess an underlying clustering structure. Moreover, our experiments reveal that applying the strictly robustified k-means model to these instances does not significantly improve an already bad nominal clustering result. Therefore, from now on, we only focus on those instances for which the Silhouette score for the true labels is larger than 0.2 and we also focus on the strictly robustified model here due to our previous results on synthetic data sets.

In order to compare the nominal method against its strictly robust counterpart, we apply each method five times with different seeds for the initialization procedure and both methods start with the same given initial centers. The average results over the five runs are presented in Table 7, where the best ARI result for the data set is highlighted in bold. As for the synthetic data sets, we calculated p values to assess the significance of the obtained results. Therefore, we underline the average results for which the difference in the ARI of the nominal vs. the strictly robust method is reasonable (p value smaller than an α-error of 10 %). All calculated p values are shown in Table 11 in the appendix.

Note that for the Silhouette score there is no big difference between the results. On the other hand, the ARI is more telling here since we compare how similar the two assignment results are to the true labeling. We can see that only for the instance mammographic-mass the nominal method outperforms the strictly robust method. For 10 out of the 15 instances, the

Table 6 Results of Algorithm 2 using the restart heuristic on the real world clustering data sets

Instance	Silhouette		ARI
	True labels	ADM	
balance-scale	0.0888	0.1656	0.1584
banknote-authentication	0.2103	0.3308	0.0219
blood-transfusion	0.0445	0.4935	−0.0062
breast-cancer	0.5715	0.5968	0.8465
breast-cancer-diagnostic	0.3389	0.3845	0.7302
breast-cancer-prognostic	0.0041	0.1960	0.0318
climate-model-crashes	0.0264	0.0388	0.0158
connectionist-bench	− 0.0024	0.1816	0.1709
connectionist-bench-sonar	0.0341	0.1986	0.0001
contraceptive-method-choice	− 0.0218	0.2840	0.0144
dermatology	0.2373	0.2438	0.5024
ecoli	0.2403	0.3040	0.5389
fertility	0.0121	0.1731	0.0128
glass-identification	− 0.0553	0.5078	0.2505
haberman	0.0324	0.3869	− 0.0040
heart-disease-cleveland	0.0231	0.2066	0.1492
hepatitis	0.0697	0.1860	0.2302
hill-valley	0.0033	0.8847	0.0006
hill-valley-noise	− 0.0021	0.8505	− 0.0006
image-segmentation	0.2068	0.3685	0.4746
ionosphere	0.1614	0.3301	0.1073
iris	0.4570	0.5043	0.7163
libras-movement	0.0215	0.2493	0.2986
magic-gamma-telescope	0.1388	0.3185	0.0318
mammographic-mass	0.2775	0.5803	0.3421
monks-1	0.0388	0.1981	− 0.0018
monks-2	0.0013	0.2060	0.0129
monks-3	0.0594	0.1978	− 0.0043
optical-recognition	0.1728	0.1876	0.6181
ozone-level-eight	− 0.0458	0.3209	− 0.0222
ozone-level-one	− 0.0534	0.3210	− 0.0124
parkinsons	0.1041	0.3030	− 0.0588
pen-based-recognition	0.1928	0.3192	0.5196
planning-relax	− 0.0101	0.1588	− 0.0029
qsar-biodegradation	0.0167	0.3640	− 0.0519
seeds	0.3823	0.4221	0.6980
seismic-bumps	0.3153	0.7691	0.0022
soybean-large	0.1415	0.2533	0.3475
soybean-small	0.3498	0.3498	1.0000
spambase	0.0441	0.6858	− 0.0045
spect-heart	0.0975	0.2973	0.1117

Table 6 continued

Instance	Silhouette		ARI
	True labels	ADM	
spectf-heart	0.0651	0.5671	0.0015
statlog-german-credit	0.0237	0.1456	0.0401
statlog-landsat-satellite	0.1977	0.3571	0.5309
teaching-assistant-evaluation	− 0.0273	0.4817	0.0477
thyroid-disease-ann	− 0.0459	0.4170	− 0.0276
thyroid-disease-new	0.4577	0.5624	0.6283
wall-following-robot-2	0.2334	0.6285	0.1275
wall-following-robot-4	− 0.0067	0.3948	0.0839
wine	0.2923	0.2998	0.8431
yeast	0.0040	0.2182	0.1905
zoo	0.3689	0.3727	0.8037

Table 7 Performance of the strictly robust versus the nominal k-means clustering method on real-world clustering data sets

Instance	Silhouette		ARI	
	Nominal	Strictly	Nominal	Strictly
banknote-authentication	0.3307	0.3314	0.0223	**0.0264**
breast-cancer	0.5968	0.5961	0.8465	**0.8630**
breast-cancer-diagnostic	0.3845	0.3787	0.7302	**0.7354**
dermatology	0.2481	0.2480	0.6211	**0.7113**
ecoli	0.2963	0.2984	0.5401	**0.5458**
image-segmentation	0.3467	0.3570	0.4276	**0.4377**
iris	0.4999	0.4995	0.7132	**0.7188**
mammographic-mass	0.5803	0.5795	**0.3421**	0.3336
seeds	0.4221	0.4221	0.6980	0.6980
seismic-bumps	0.7691	0.7691	0.0022	0.0022
soybean-small	0.3498	0.3498	1	1
thyroid-disease-new	0.5504	0.5451	0.5799	**0.5932**
wall-following-robot-2	0.6117	0.5447	0.1681	**0.3567**
wine	0.2995	0.3007	0.8409	**0.8600**
zoo	0.3735	0.3735	0.8167	0.8167

For each instance, the best average ARI result is printed in bold. The average result that is significantly better than the nominal one according to the calculated p value is underlined

strictly robust model performs the best, yielding clustering results that have more similarity to the true classes than the nominal clustering results do. Moreover, the underlined average results reveal that the improvement in clustering quality attained with the strictly robust method is significant. Thus, in the presence of potential measurement errors, the strictly robustified k-means clustering method is able to give more accurate results than its nominal counterpart.

Table 8 Computational performance of the nominal and strictly robust models on real-world clustering data sets

Instance	Time (s)		Iterations	
	Nominal	Strictly	Nominal	Strictly
banknote-authentication	0.25	14.99	19.8	28.0
breast-cancer	0.04	3.78	5.4	5.0
breast-cancer-diagnostic	0.05	15.28	9.8	7.4
dermatology	0.19	38.15	15.0	16.6
ecoli	0.16	5.34	16.8	11.2
image-segmentation	0.04	7.11	6.2	7.0
iris	0.02	0.46	7.6	5.0
mammographic-mass	0.04	2.70	4.8	5.2
seeds	0.02	1.19	6.2	5.6
seismic-bumps	0.08	16.82	3.0	3.0
soybean-small	0.00	1.27	5.2	4.6
thyroid-disease-new	0.03	1.10	5.6	5.2
wall-following-robot-2	2.39	38.16	15.8	15.4
wine	0.01	1.95	5.8	6.0
zoo	0.01	1.91	3.8	3.4

The running times and iterations counts are listed in Table 8. Clearly, on average the strictly robust method needs more time to find a partial minimum. However, it is still rather fast, never requiring more than 39 s on average. In contrast to runtimes, the number of iterations does not change significantly.

We conclude that the strictly robustified k-means clustering model is also beneficial for real-world clustering instances, which likely contain uncertainties due to measurement errors. Specifically, for almost all of the instances, the strictly robust model leads to improvements regarding the recovery of the true assignment over the nominal model.

8 Conclusion

In this paper, we derived the strictly and Γ-robust counterpart of the k-means clustering problem to hedge clustering results against errors without known structure in the observed data. The robustification makes it possible to obtain clusterings that are closer to clusterings of the original, i.e., error-free, data although only erroneous input data can be considered. Since already the nominal problem is NP-hard, we develop tailored alternating direction methods to quickly compute (robust) feasible solutions of good quality. Since these methods only converge to so-called partial minima that may correspond to clusterings of bad quality, we develop a restart heuristic for which we experimentally show that it significantly improves the quality of the computed clusterings. Our comparison of the strictly and Γ-robust clustering method reveals the interesting aspect that the less conservative Γ-approach is outperformed by the classic concept of strict robustness. Most importantly, the strictly robustified clustering method is able to recover clusterings of the original data even if only erroneous measurements are observed. Thus, we see that the studied robustifications can lead to more reliable models.

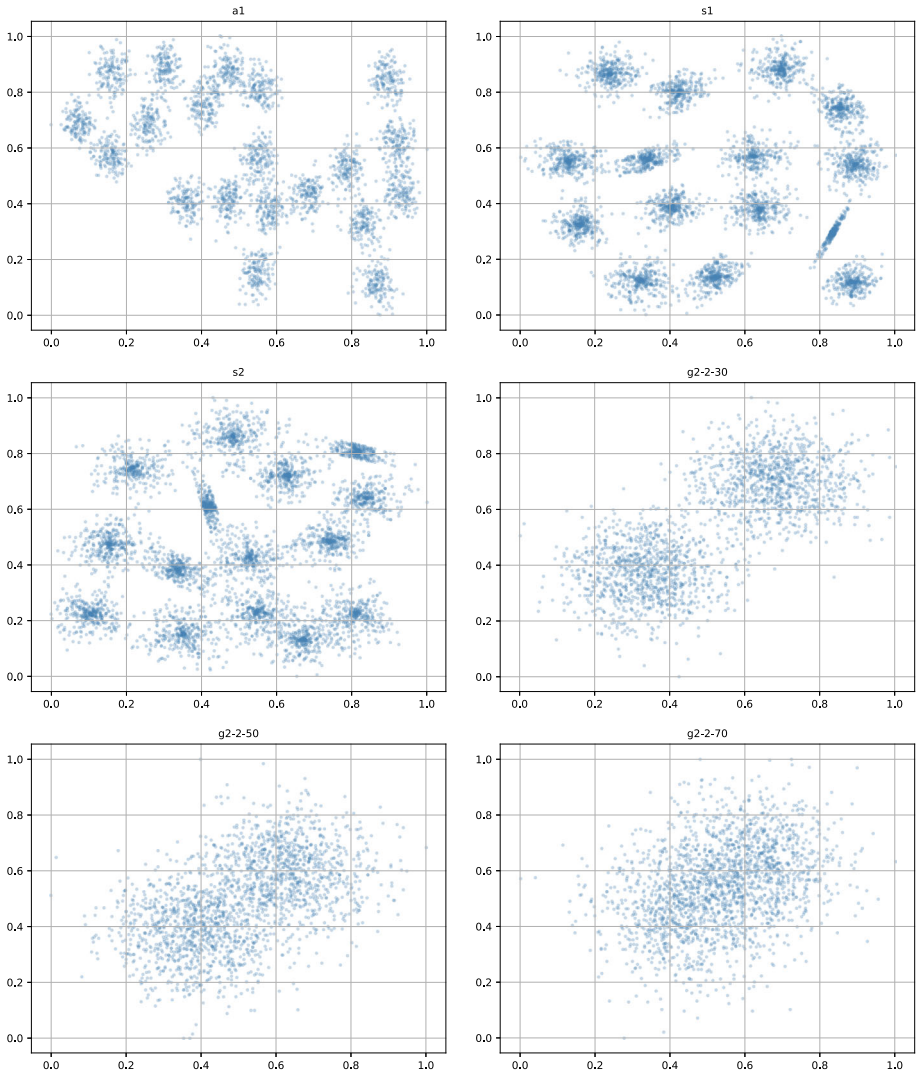

Fig. 8 Some of the synthetic clustering instances used for the experiments; originally proposed in Fränti and Sieranoja (2018)

Let us finally sketch some open problems that we do not resolve in this paper. One issue with the Γ-approach is that partial minima are obtained that do not correspond to globally optimal clusterings. Consequently, the development of methods to compute global optima of robustified clustering problems may be a worthwhile topic of future work. Moreover, we do not answer the question on how to derive practically meaningful uncertainty sets for the robustification, which may be done for practical applications by means of cross-validation. Further, other uncertainty sets such as polyhedral and ellipsoidal sets should be considered in future works as well.

Finally, one could study whether other clustering approaches such as spectral or hierarchical clustering can also be robustified so that they can cope with uncertain data. However,

these approaches are not based on a clear-cut optimization model so that new robust concepts need to be developed that are maybe not directly based on robust optimization.

Acknowledgements The authors thank the DFG for their support within RTG 2126 "Algorithmic Optimization".

Funding Open Access funding enabled and organized by Projekt DEAL.

Appendix

In this section, we present additional information that complement those given in Sect. 7.5 and Sect. 7.8. Specifically, in Fig. 8 we present the plots of the remaining instances, in Table 9 we present the computational performance for the instances not discussed in Sect. 7.5, and

Table 9 Average runtime and iteration counts for the corresponding ADMs with restart heuristic for the nominal, the strictly, and the Γ-robust models on the perturbed (10 %) synthetic data sets

Instance	Time (s)			Iterations		
	Nominal	Strictly	Γ	Nominal	Strictly	Γ
a1	4.5	37.5	61.1	29.4	33.7	24.4
s1	4.0	39.5	70.8	16.9	19.3	18.5
s2	6.1	43.4	93.2	28.1	22.9	26.0
dim	0.1	58.9	70.0	2.0	2.0	2.0
g2-2-30	0.1	2.7	4.3	5.5	5.0	4.7
g2-2-50	0.2	4.1	8.2	9.9	8.6	9.3
g2-2-70	0.3	6.6	11.2	13.4	15.3	13.1
a1	6.7	42.8	68.0	42.2	39.0	28.9
s1	4.7	41.8	77.7	20.4	21.0	21.9
s2	6.4	48.0	93.6	29.0	26.0	27.6
dim	0.1	60.0	68.8	2.0	2.0	2.0
g2-2-30	0.1	3.1	5.7	5.7	5.7	6.9
g2-2-50	0.2	4.6	6.6	10.1	9.4	8.1
g2-2-70	0.3	5.3	6.7	12.0	11.0	8.2
a1	6.3	42.6	55.2	40.9	40.0	23.0
s1	5.9	41.6	89.3	27.1	21.5	26.7
s2	5.4	47.1	75.1	27.5	25.6	21.5
dim	0.1	58.8	67.6	2.0	2.0	2.0
g2-2-30	0.1	3.1	5.9	6.4	5.9	7.6
g2-2-50	0.2	4.4	7.0	10.1	9.5	9.2
g2-2-70	0.3	5.8	7.1	13.1	12.6	9.3

The blocks are separated according to different amounts (5, 30, 50 %) of randomly perturbed data points, respectively

Table 10 Calculated p values using the Wilcoxon signed-rank test for the synthetic data sets w.r.t. each validation metric and each pair of methods

Instance	p values Silhouette			p values ARI		
	Nom. versus Str.	Nom. versus Γ	Str. versus Γ	Nom. versus Str.	Nom. versus Γ	Str. versus Γ
a1	0.105	0.002	0.064	0.010	0.002	0.014
a2	0.004	0.002	0.002	0.064	0.002	0.002
a3	0.002	0.432	1.000	0.322	0.432	1.000
s1	–	0.004	0.004	0.002	0.010	0.027
s2	0.002	0.002	0.002	0.275	0.002	0.002
s3	0.375	0.027	0.037	0.375	0.020	0.037
s4	0.049	0.002	0.002	0.322	0.004	0.002
unbalance	–	0.002	0.002	0.046	0.193	0.131
dim	–	–	–	–	–	–
g2-2-30	–	0.002	0.002	–	0.002	0.002
g2-2-50	–	0.002	0.002	0.002	0.002	0.002
g2-2-70	0.064	0.002	0.002	0.002	0.002	0.001
a1	0.695	0.002	0.004	0.064	0.002	0.004
a2	1.000	0.131	0.014	0.064	0.131	0.004
a3	0.846	0.492	0.625	0.922	0.770	0.922
s1	0.846	0.002	0.002	0.004	0.002	0.002
s2	0.557	0.131	0.027	0.002	0.131	0.027
s3	0.625	0.002	0.002	0.625	0.002	0.002
s4	1.000	0.002	0.002	0.625	0.020	0.002
unbalance	0.002	0.002	0.002	0.002	0.002	0.002
dim	–	–	–	–	–	–
g2-2-30	–	0.002	0.002	0.180	0.002	0.002
g2-2-50	0.176	0.002	0.002	0.237	0.002	0.002
g2-2-70	0.922	0.002	0.002	0.010	0.002	0.002
a1	0.375	0.002	0.002	0.064	0.002	0.002
a2	1.000	0.004	0.002	1.000	0.002	0.002
a3	0.770	0.010	0.006	0.625	0.049	0.020
s1	0.432	0.002	0.002	0.004	0.002	0.002
s2	0.105	0.010	0.004	0.064	0.010	0.004
s3	0.322	0.002	0.002	0.002	0.002	0.002
s4	0.625	0.002	0.002	0.020	0.002	0.002
unbalance	0.002	0.232	0.131	0.002	1.000	0.232
dim	–	–	–	–	–	–
g2-2-30	–	0.002	0.002	0.893	0.002	0.002
g2-2-50	0.735	0.002	0.002	0.463	0.002	0.002
g2-2-70	0.492	0.002	0.002	0.086	0.002	0.002

The ("–") means that the average results are the same for both methods and so the p value is not calculated

Table 11 Calculated p values using the Wilcoxon signed-rank test for the real-world data sets w.r.t. each validation metric and each pair of methods

Instance	Nominal versus strictly	
	p value Silhouette	p value ARI
banknote-authentication	0.063	0.063
breast-cancer	0.063	0.063
breast-cancer-diagnostic	0.063	0.063
dermatology	0.625	0.813
ecoli	0.813	1.000
image-segmentation	0.285	0.285
iris	0.317	0.317
mammographic-mass	0.063	0.063
seeds	–	–
seismic-bumps	–	–
soybean-small	–	–
thyroid-disease-new	0.046	0.046
wall-following-robot-2	0.063	0.063
wine	0.083	0.083
zoo	–	–

The ("–") means that the average results are the same and so the p value is not calculated

in Tables 10 and 11 we present the calculated p values for each validation metric and each pair of methods.

Additionally, we also present a comparison of our nominal as well as strictly robust clustering method with a spectral clustering method. To this end, we further apply spectral clustering to the test sets that we consider in this paper. The setup for the experiment is the following. We apply each method to the same instance and assess the quality of the clustering by computing the Silhouette index and the ARI. Here, we always compare the clustering result to the ground truth labels instead of comparing to the ADM results. To assess the significance of the obtained results, we apply the Wilcoxon signed-rank test to calculate p values. If the resulting p value is smaller than the α-error of 10%, we reject the null-hypothesis that the strictly robust result is less or equally good as the spectral one in terms of the Silhouette score or the ARI, respectively. To compute the spectral clustering, we use the Python module sklearn.cluster.SpectralClustering. The results w.r.t. the synthetic instances are presented in Table 12 and the results w.r.t. the real-world instances are shown in Table 13; the respective computed p values are presented in Tables 14 and 15.

Table 12 Performance of the nominal, strictly robust, and spectral clustering methods on the perturbed synthetic data sets

Instance	Silhouette			ARI			Time (s)		
	Nom.	Str.	Spec.	Nom.	Str.	Spec.	Nom.	Str.	Spec.
a1	0.50	**0.55**	0.33	0.81	**0.91**	0.50	5.2	92.4	66.9
a2	0.51	**0.57**	0.22	0.83	**0.94**	0.31	16.1	194.0	120.2
a3	0.53	**0.57**	0.25	0.83	**0.91**	0.30	49.4	334.8	152.8
s1	0.66	**0.69**	0.45	0.94	**0.98**	0.72	5.3	53.1	24.8
s2	0.58	**0.61**	0.34	0.86	**0.92**	0.59	9.5	90.0	151.9
s3	0.46	**0.48**	0.23	0.66	**0.70**	0.45	14.2	137.5	88.2
s4	0.46	**0.46**	0.22	0.58	**0.60**	0.41	17.3	196.5	240.5
Unbalance	0.66	0.77	**0.80**	0.64	0.90	**0.98**	2.5	101.7	12.7
dim	0.93	0.93	0.93	1.00	1.00	1.00	0.2	140.3	1.7
g2-2-30	0.63	0.63	0.63	0.96	**0.96**	0.96	0.2	4.0	2.4
g2-2-50	0.48	0.48	0.48	**0.69**	0.69	0.69	0.6	9.5	4.4
g2-2-70	0.40	0.40	0.40	0.48	**0.48**	0.48	0.5	9.3	2.3
a1	0.47	**0.49**	0.29	0.72	**0.76**	0.41	6.0	84.3	60.1
a2	0.46	**0.49**	0.20	0.68	**0.74**	0.29	20.3	233.4	134.4
a3	0.47	**0.49**	0.22	0.66	**0.70**	0.27	45.7	438.0	168.2
s1	0.54	**0.61**	0.39	0.82	**0.93**	0.67	8.9	103.6	64.9
s2	0.52	**0.54**	0.30	0.78	**0.83**	0.54	5.0	95.6	116.0
s3	0.42	**0.43**	0.21	0.59	**0.60**	0.40	10.9	109.9	111.3
s4	0.42	**0.42**	0.18	0.51	**0.52**	0.35	19.3	171.7	195.5
Unbalance	0.56	0.62	**0.66**	0.55	0.70	**0.87**	1.8	73.7	10.8
dim	0.88	0.88	0.88	1.00	1.00	1.00	0.1	95.9	1.7
g2-2-30	0.61	0.61	0.61	0.93	0.93	0.93	0.2	5.2	2.2
g2-2-50	0.47	0.47	0.47	0.64	0.64	0.64	0.3	7.5	2.4
g2-2-70	0.40	**0.40**	0.40	0.44	**0.45**	0.44	0.4	8.2	2.1
a1	0.44	**0.45**	0.25	0.63	**0.66**	0.37	5.6	62.1	54.5
a2	0.43	**0.44**	0.19	0.61	**0.63**	0.28	22.9	222.6	106.2
a3	0.43	**0.44**	0.20	0.54	**0.55**	0.24	37.0	447.9	153.9
s1	0.53	**0.55**	0.31	0.85	**0.89**	0.60	4.4	81.9	126.1
s2	0.47	**0.49**	0.31	0.72	**0.76**	0.56	11.2	143.8	134.0
s3	0.40	**0.40**	0.21	0.53	**0.54**	0.39	15.7	125.8	177.0
s4	0.39	**0.40**	0.18	0.46	**0.46**	0.33	10.1	191.3	211.4
Unbalance	0.50	0.51	**0.54**	0.48	0.53	**0.77**	2.1	46.0	9.3
dim	0.85	0.85	0.85	1.00	1.00	1.00	0.1	78.7	1.1
g2-2-30	0.60	0.60	0.60	**0.92**	0.92	0.92	0.2	4.5	2.0
g2-2-50	0.47	0.47	0.47	0.62	0.62	0.61	0.3	6.4	2.1
g2-2-70	0.40	**0.40**	0.40	0.41	0.41	**0.41**	0.3	7.0	2.1

The perturbation amount is 10%, and the quantity of perturbed data points is different in each block of the table: the top block corresponds to 5%, the middle block to 30%, and the bottom block to 50%. For each instance and each validation metric, the best average result is printed in bold. According to the calculated p value (presented in Table 14), the average result of the strictly robust method that is significantly better than the spectral one is underlined. The average results are based on 10 runs

Table 13 Performance of the nominal, strictly robust, and spectral clustering methods on the real-world clustering data sets

Instance	Silhouette			ARI			Time (s)		
	Nom.	Str.	Spec.	Nom.	Str.	Spec.	Nom.	Str.	Spec.
banknote-authentication	0.33	**0.33**	0.32	0.02	0.03	**0.06**	0.3	15.1	1.6
breast-cancer	**0.60**	0.60	0.58	0.85	**0.86**	0.72	0.1	6.1	0.6
breast-cancer-diagnostic	0.38	0.38	**0.40**	0.73	**0.74**	0.50	0.1	31.2	0.7
dermatology	0.23	**0.24**	0.24	**0.61**	0.61	0.56	0.1	86.5	0.4
ecoli	**0.33**	0.33	0.25	**0.61**	0.60	0.43	0.2	12.1	0.4
image-segmentation	0.33	**0.35**	0.28	0.38	**0.41**	0.39	0.1	23.3	0.3
iris	0.49	**0.50**	0.49	0.60	**0.73**	0.62	0.1	2.9	0.4
mammographic-mass	**0.58**	0.58	0.58	0.34	0.33	**0.34**	0.1	7.7	2.1
seeds	0.42	0.42	0.35	0.70	0.70	0.67	0.1	2.8	0.3
seismic-bumps	0.77	0.77	0.59	0.00	0.00	**0.19**	0.2	40.3	3.5
soybean-small	0.35	0.35	0.35	1.00	1.00	1.00	0.0	4.3	0.1
thyroid-disease-new	0.56	0.56	**0.59**	0.63	**0.64**	0.39	0.1	7.1	0.3
wall-following-robot-2	**0.63**	0.54	0.51	0.12	0.36	**0.37**	2.5	85.8	8.4
wine	0.30	**0.30**	0.30	0.86	0.86	**0.93**	0.1	9.0	0.5
zoo	0.37	0.36	**0.40**	0.80	0.75	**0.92**	0.0	10.4	0.3

For each instance and each validation metric, the best average result is printed in bold. According to the calculated p value shown in Table 15, the best average result of the strictly robust method that is significantly better than the spectral one is underlined. The average results are based on five runs

Table 14 Calculated p values using the Wilcoxon signed-rank test for the synthetic data sets

Instance	Nom. vs. Str.	Nom. vs. Spec.	Str. vs. Spec.
a1	0.01/ 0.01	0.01/ 0.01	0.01/0.01
a2	0.01/ 0.01	0.01/ 0.01	0.01/0.01
a3	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s1	0.11/ 0.04	0.01/ 0.01	0.01/0.01
s2	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s3	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s4	0.57/ 0.14	0.01/ 0.01	0.01/0.01
unbalance	0.02/ 0.01	0.01/ 0.01	0.39/ 0.39
dim	–/–	–/–	–/–
g2-2-30	–/ 0.32	–/–	–/ 0.32
g2-2-50	–/ 0.86	–/ 0.46	–/ 0.40
g2-2-70	0.51/ 0.72	0.17 / 0.72	0.96 / 0.44
a1	0.01/ 0.01	0.01/ 0.01	0.01/ 0.01
a2	0.01/ 0.01	0.01/ 0.01	0.01/0.01
a3	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s1	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s2	0.01/ 0.01	0.01/ 0.01	0.01/0.01
s3	0.20/ 0.03	0.01/ 0.01	0.01/0.01
s4	0.51/ 0.07	0.01/ 0.01	0.01/0.01
unbalance	0.00/ 0.01	0.01/ 0.01	0.21/ 0.01
dim	–/–	–/–	–/–
g2-2-30	–/ 0.47	–/ 1.0	–/ 0.72
g2-2-50	0.60/ 0.17	0.26/ 0.76	0.48/ 0.02
g2-2-70	0.28/ 0.02	0.51/ 0.21	0.14/ 0.02
a1	0.01/ 0.01	0.00/ 0.00	0.01/0.01
a2	0.01/ 0.01	0.00/ 0.00	0.01/0.01
a3	0.01/ 0.01	0.00/ 0.00	0.01/0.01
s1	0.04/ 0.01	0.00/ 0.00	0.01/0.01
s2	0.07/ 0.01	0.00/ 0.00	0.01/0.01
s3	0.09/ 0.01	0.00/ 0.00	0.01/0.01
s4	0.39/ 0.14	0.00/ 0.00	0.01/0.01
unbalance	0.01/ 0.01	0.07/ 0.00	0.09/ 0.01
dim	–/–	–/–	–/–
g2-2-30	–/ 0.07	0.04/ 0.02	0.22/0.69
g2-2-50	–/ 0.61	0.01/ 0.73	0.06/0.59
g2-2-70	0.01/ 0.11	0.04/ 0.64	0.01/0.96

For each instance and each pair of methods, the first number represents the p value w.r.t. the Silhouette index and the second number represents the p value w.r.t. the ARI. The ("–") means that the average results are the same for both methods and so the p value is not calculated

Table 15 Calculated p values using the Wilcoxon signed-rank test for the real-world data sets

Instance	Nom. versus Str.	Nom. versus Spec.	Str. versus Spec.
banknote-authentication	0.06/ 0.06	0.06/ 0.06	0.06/0.06
breast-cancer	0.06/ 0.06	0.06/ 0.06	0.06/0.06
breast-cancer-diagnostic	0.06/ 0.06	0.06/0.06	0.06/0.06
dermatology	0.31/ 0.63	0.31/ 0.31	0.06/0.31
ecoli	1.00/ 1.00	0.06/ 0.06	0.06/0.13
image-segmentation	0.19/ 0.81	0.31/ 1.00	0.06/0.44
iris	0.14/ 0.07	0.31/ 0.81	0.31/0.06
mammographic-mass	0.06/ 0.06	0.06/ 0.06	0.06/0.06
seeds	–/–	0.06/ 0.06	0.06/0.06
seismic-bumps	–/–	0.06/ 0.06	0.06/0.06
soybean-small	–/–	–/–	–/–
thyroid-disease-new	0.06/ 0.06	0.06/ 0.06	0.06/0.06
wall-following-robot-2	0.06/ 0.06	0.06/ 0.06	0.06/0.31
wine	0.46/ 0.46	0.81/ 0.06	0.31/0.06
zoo	0.65/ 0.65	0.19/ 0.19	0.06/0.06

For each instance and each pair of methods, the first number represents the p value w.r.t. the Silhouette index and the second number represents the p value w.r.t. the ARI. The ("–") means that the average results are the same and so the p value is not calculated

References

Alfons, A., Templ, M., & Filzmoser, P. (2013). Robust estimation of economic indicators from survey samples based on Pareto tail modelling. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 62*(2), 271–286. https://doi.org/10.1111/j.1467-9876.2012.01063.x.

Aloise, D., Deshpande, A., Hansen, P., & Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning, 75*, 245–248. https://doi.org/10.1007/s10994-009-5103-0

Aloise, D., Hansen, P., & Liberti, L. (2012). An improved column generation algorithm for minimum sum-of-squares clustering. *Mathematical Programming, 131*, 195–220. https://doi.org/10.1007/s10107-010-0349-7

Ames, B. P. W. (2014). Guaranteed clustering and biclustering via semidefinite programming. *Mathematical Programming, 147*(1), 429–465. https://doi.org/10.1007/s10107-013-0729-x

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition, 46*(1), 243–256. https://doi.org/10.1016/j.patcog.2012.07.021

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization* (Vol. 28). Princeton University Press.

Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming, 99*(2), 351–376. https://doi.org/10.1007/s10107-003-0454-y

Ben-Tal, A., & Nemirovski, A. (2001). Lectures on modern convex optimization. *Society for Industrial and Applied Mathematics*. https://doi.org/10.1137/1.9780898718829

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: Numerical methods*. Prentice-Hall Inc.

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review, 53*(3), 464–501. https://doi.org/10.1137/080734510

Bertsimas, D., Dunn, J., Pawlowski, C., & Zhuo, Y. D. (2019). Robust classification. *INFORMS Journal on Optimization, 1*(1), 2–34. https://doi.org/10.1287/ijoo.2018.0001

Bertsimas, D., Pawlowski, C., & Zhuo, Y. D. (2017). From predictive methods to missing data imputation: An optimization approach. *The Journal of Machine Learning Research, 18*(1), 7133–7171.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53. https://doi.org/10.1287/opre.1030.0065

Bhattacharyya, C., Pannagadatta, K. S., & Smola, A. J. (2005) A second order cone programming formulation for classifying missing data. In *Proceedings of the 17th international conference on neural information processing systems* (pp. 153–160). MIT Press.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning, 3*(1), 1–122. https://doi.org/10.1561/2200000016

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Burgard, J. P., Krause, J., Kreber, D., & Morales, D. (2020). The generalized equivalence of regularization and min–max robustification in linear mixed models. *Statistical Papers*. https://doi.org/10.1007/s00362-020-01214-z

Burgard, J. P., & Münnich, R. T. (2012). Modelling over and undercounts for design-based Monte Carlo studies in small area estimation: An application to the German register-assisted census. *Computational Statistics & Data Analysis, 56*(10), 2856–2863. https://doi.org/10.1016/j.csda.2010.11.002

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). *Measurement error in nonlinear models: A modern perspective*. CRC Press.

Celebi, M. E., & Aydin, K. (2016). *Unsupervised learning algorithms.* Springer. https://doi.org/10.1007/978-3-319-24211-8

Dasgupta, S. (2007) The hardness of k-means clustering. Technical Report CS2008-0916. University of California, Department of Computer Science and Engineering. http://cseweb.ucsd.edu/~dasgupta/papers/kmeans.pdf

Davalos, S. (2017). Big data has a big role in biostatistics with big challenges and big expectations. *Biostatistics and Biometrics Open Access Journal, 1*(3), 1–2. https://doi.org/10.19080/BBOAJ.2017.01.555563

Dua, D., & Graff, C. (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml

Fischetti, M., & Monaci, M. (2009) Light robustness. In Ahuja, R. K., Möhring, R. H., & Zaroliagis, C. D. (Eds.) *Robust and online large-scale optimization: Models and techniques for transportation systems* (pp. 61–84). Springer. https://doi.org/10.1007/978-3-642-05465-5_3

Fraiman, R., Ghattas, B., & Svarc, M. (2013). *Interpretable clustering using unsupervised binary trees.* https://doi.org/10.1007/s11634-013-0129-3

Fränti, P., & Sieranoja, S. (2018). k-means properties on six clustering benchmark datasets. *Applied Intelligence, 48*(12), 4743–4759. https://doi.org/10.1007/s10489-018-1238-7

Fränti, P., & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition, 93*, 95–112. https://doi.org/10.1016/j.patcog.2019.04.014

Gabay, D., & Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications, 2*(1), 17–40. https://doi.org/10.1016/0898-1221(76)90003-1

Geißler, B., Morsi, A., Schewe, L., & Schmidt, M. (2015). Solving power-constrained gas transportation problems using an MIP-based alternating direction method. *Computers & Chemical Engineering, 82*, 303–317. https://doi.org/10.1016/j.compchemeng.2015.07.005

Geißler, B., Morsi, A., Schewe, L., & Schmidt, M. (2017). Penalty alternating direction methods for mixed-integer optimization: A new view on feasibility pumps. *SIAM Journal on Optimization*. https://doi.org/10.1137/16M1069687

Geißler, B., Morsi, A., Schewe, L., & Schmidt, M. (2018). Solving highly detailed gas transport MINLPs: Block separability and penalty alternating direction methods. *INFORMS Journal on Computing, 30*(2), 309–323. https://doi.org/10.1287/ijoc.2017.0780

Glowinski, R., & Marroco, A. (1975) Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. In *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 9.R2* (pp. 41–76). http://eudml.org/doc/193269

Gorski, J., Pfeuffer, F., & Klamroth, K. (2007). Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research, 66*(3), 373–407. https://doi.org/10.1007/s00186-007-0161-1

Grira, N., Crucianu, M., & Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: A brief survey. *A Review of Machine Learning Techniques for Processing Multimedia Content, 1*, 9–16.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218. https://doi.org/10.1007/BF01908075

Khoury, M. J., & Ioannidis, J. P. A. (2014). Big data meets public health. *Science, 346*(6213), 1054–1055. https://doi.org/10.1126/science.aaa2709

Li, J., Song, S., Zhang, Y., & Zhou, Z. (2016). Robust k-median and k-means clustering algorithms for incomplete data. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2016/4321928

Liebchen, C., Lübbecke, M., Möhring, R., & Stiller, S. (2009) Robust and online large-scale optimization: Models and techniques for transportation systems. In Ahuja, R. K., Möhring, R. H., & Zaroliagis, C. D. (Eds.) *Chap. The concept of recoverable robustness, linear programming recovery, and railway applications* (pp. 1–27). Springer.https://doi.org/10.1007/978-3-642-05465-5_1.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

MacQueen, J. (1967)"Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Volume 1: Statistics* (pp. 281–297). University of California Press. https://projecteuclid.org/euclid.bsmsp/1200512992

Mahajan, M., Nimbhorkar, P., & Varadarajan, K. (2012) The planar k-means problem is NP-hard. In *Theoretical computer science 442. Special issue on the workshop on algorithms and computation (WALCOM 2009)* (pp. 13–21). https://doi.org/10.1016/j.tcs.2010.05.034

Pant, R., Trafalis, T. B., & Barker, K. (2011) Support vector machine classification of uncertain and imbalanced data using robust optimization. In *Proceedings of the 15th WSEAS international conference on computers* (pp. 369–374). World Scientific, Engineering Academy, and Society (WSEAS).

Pirinen, A., & Ames, B. (2019). Exact clustering of weighted graphs via semidefinite programming. *The Journal of Machine Learning Research, 20*(1), 1007–1040.

Rocke, D. M., Ideker, T., Troyanskaya, O., Quackenbush, J., & Dopazo, J. (2009). Papers on normalization, variable selection, classification or clustering of microarray data. *Bioinformatics, 25*(6), 701–702. https://doi.org/10.1093/bioinformatics/btp038

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Schewe, L., Schmidt, M., & Weninger, D. (2020). A decomposition heuristic for mixed-integer supply chain problems. *Operations Research Letters, 48*(3), 225–232. https://doi.org/10.1016/j.orl.2020.02.006

Soyster, A. L. (1973). Technical note-convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research, 21*(5), 1154–1157. https://doi.org/10.1287/opre.21.5.1154

Steinley, D. (2004). Properties of the Hubert–Arable adjusted rand index. *Psychological Methods, 9*(3), 386–396. https://doi.org/10.1037/1082-989X.9.3.386

Su, Y., Reedy, J., & Carroll, R. J. (2018). Clustering in general measurement error models. *Statistica Sinica, 28*(4), 2337.

Trafalis, T. B., & Gilbert, R. C. (2007). Robust support vector machines for classification and computational issues. *Optimization Methods and Software, 22*(1), 187–198. https://doi.org/10.1080/10556780600883791

Vo, X. T., Le Thi, H. A., & Pham Dinh, T. (2016) Robust optimization for clustering. In *Intelligent information and database systems* (pp. 671–680). Springer. https://doi.org/10.1007/978-3-662-49390-8_65

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z

Wendell, R. E., & Hurter, A. P. (1976). Minimization of a non-separable objective function subject to disjoint constraints. *Operations Research, 24*(4), 643–657. https://doi.org/10.1287/opre.24.4.643

White, E. (2011). Measurement error in biomarkers: Sources, assessment, and impact on studies. *IARC Scientific Publications, 163*, 143–161.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83. https://doi.org/10.2307/3001968

Yamada, K., Takayasu, H., & Takayasu, M. (2018). Estimation of economic indicator announced by government from social big data. *Entropy, 20*(11), 852–864. https://doi.org/10.3390/e20110852