Strong bounds for large-scale Minimum Sum-of-Squares Clustering

Veronica Piccialli

Sapienza University of Rome

MIP Europe 2025 Clermont Ferrand, July 1st 2025

FAIR project

This research has been funded by the PNRR funded project Future Artificial Intelligence Research (FAIR) :

Sapienza: Spoke 5 High Quality AI:

Task 5.7.3 – High Quality AI by means of Optimization

Funded by the European Union
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Future Artificial Intelligence Research

Minimum Sum of Squares Clustering

It starts from a visit of Angelika Wiegele[a] in 2019:
V. Piccialli, A. M. Sudoso and A. Wiegele.
SOS-SDP: An Exact Solver for Minimum Sum-of-Squares Clustering, **INFORMS Journal on Computing**, 34 (4), 2144-2162 (2022).

[a]Alpen-Adria-Universitat Klagenfurt

Minimum Sum of Squares Clustering

It starts from a visit of Angelika Wiegele[a] in 2019:
V. Piccialli, A. M. Sudoso and A. Wiegele. SOS-SDP: An Exact Solver for Minimum Sum-of-Squares Clustering, **INFORMS Journal on Computing**, 34 (4), 2144-2162 (2022).

V. Piccialli, A. Russo Russo and A. M. Sudoso. An exact algorithm for semi-supervised minimum sum-of-squares clustering, **Computers & Operations Research** (2022).

[a] Alpen-Adria-Universitat Klagenfurt

SAPIENZA
UNIVERSITÀ DI ROMA

Minimum Sum of Squares Clustering

It starts from a visit of Angelika Wiegele[a] in 2019:
V. Piccialli, A. M. Sudoso and A. Wiegele. SOS-SDP: An Exact Solver for Minimum Sum-of-Squares Clustering, **INFORMS Journal on Computing**, 34 (4), 2144-2162 (2022).

V. Piccialli, A. Russo Russo and A. M. Sudoso. An exact algorithm for semi-supervised minimum sum-of-squares clustering, **Computers & Operations Research** (2022).

V. Piccialli and A. M. Sudoso. Global optimization for cardinality-constrained minimum sum-of-squares clustering via semidefinite programming, **Mathematical Programming** (2023)

[a] Alpen-Adria-Universitat Klagenfurt

SAPIENZA UNIVERSITÀ DI ROMA

Large scale MSSC

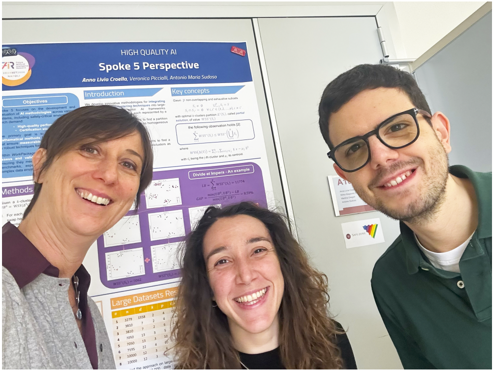
Idea

What can we do to exploit our tools for small-medium size instances and find some optimality guarantee for heuristic solutions?

Large scale MSSC

Idea

What can we do to exploit our tools for small-medium size instances and find some optimality guarantee for heuristic solutions?

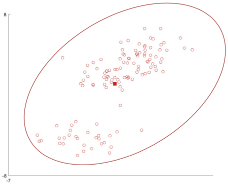


Joint work with Anna Livia Croella Assistant Professor at Mercatorum and Antonio Maria Sudoso my colleague at DIAG-Sapienza

SAPIENZA UNIVERSITÀ DI ROMA

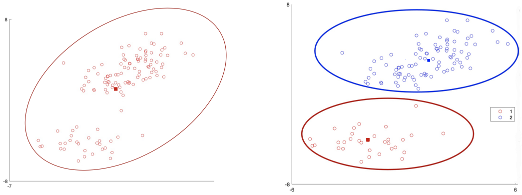
MSSC Clustering

We focus on the **Minimum Sum of Squares Clustering** problem.
Given n data points $x_1, \ldots, x_n$ in $\mathbb{R}^d$, partition them into k clusters by minimizing the total squared distance between each point and the cluster center.



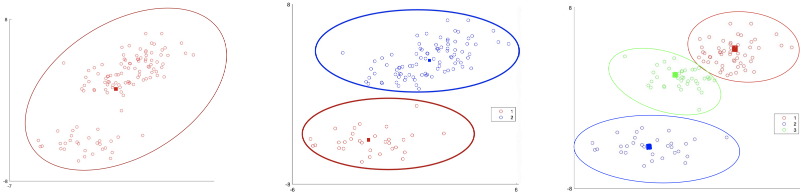
MSSC Clustering

We focus on the **Minimum Sum of Squares Clustering** problem.
Given n data points $x_1, \ldots, x_n$ in $\mathbb{R}^d$, partition them into k clusters by minimizing the total squared distance between each point and the cluster center.



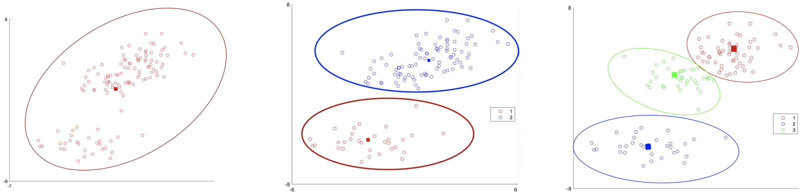
MSSC Clustering

We focus on the **Minimum Sum of Squares Clustering** problem.
Given n data points $x_1, \ldots, x_n$ in $\mathbb{R}^d$, partition them into k clusters by minimizing the total squared distance between each point and the cluster center.



MSSC Clustering

We focus on the **Minimum Sum of Squares Clustering** problem.
Given n data points $x_1, \ldots, x_n$ in $\mathbb{R}^d$, partition them into k clusters by minimizing the total squared distance between each point and the cluster center.



Given a cluster assignment, the optimal cluster center is the average of the points in the cluster.

SAPIENZA
UNIVERSITÀ DI ROMA

Mathematical formulation

The mathematical formulation is

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}\sum_{j=1}^{k} \delta_{ij}||x_i - m_j||_2^2 \\ & \sum_{j=1}^{k} \delta_{ij} = 1 \quad i = 1, \ldots, n \\ & \delta_{ij} \in \{0.1\},\; m_j \in \mathbb{R}^d \quad i = 1, \ldots, n,\; j = 1, \ldots, k \end{aligned} \tag{MSSC}$$

where δ_{ij} are the cluster indicator variables, i.e. $\delta_{ij} = 1$ if point i is assigned to the cluster j and 0 otherwise, and m_j are the cluster centers.

Mathematical formulation

The mathematical formulation is

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{k}\delta_{ij}||x_i - m_j||_2^2 \\
& \sum_{j=1}^{k}\delta_{ij} = 1 \quad i = 1,\dots,n \\
& \delta_{ij} \in \{0.1\},\ m_j \in \mathbb{R}^d \quad i = 1,\dots,n,\ j = 1,\dots,k
\end{aligned}
\tag{MSSC}
$$

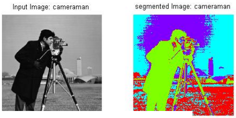
where δ_{ij} are the cluster indicator variables, i.e. $\delta_{ij} = 1$ if point i is assigned to the cluster j and 0 otherwise, and m_j are the cluster centers.

(MSSC) is NP-hard even for $k = 2$ or $d = 2$ (Aloise, Deshpande, Hansen, and Popat 2009)

SAPIENZA UNIVERSITÀ DI ROMA

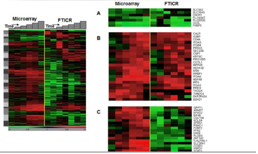
Applications

1. Image segmentation

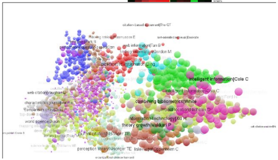


2. credit risk evaluation

3. biology



4. document clustering



SAPIENZA UNIVERSITÀ DI ROMA

Literature Review

Heuristic methods

✓ *k*-means (MacQueen 1967; Lloyd 1982) heuristic ($\approx$ 3M references in Google Scholar 2024) is the most popular heuristic for solving MSSC.

SAPIENZA UNIVERSITÀ DI ROMA

Literature Review

Heuristic methods

- ✓ k-means (MacQueen 1967; Lloyd 1982) heuristic ($\approx$ 3M references in Google Scholar 2024) is the most popular heuristic for solving MSSC.
- ✓ a lot of research has been dedicated to finding efficient initialization for k-means (Arthur and Vassilvitskii 2006; Yu, Chu, Wang, Chan, and Chang 2018; Franti and Sieranoja 2019)

SAPIENZA
UNIVERSITÀ DI ROMA

Literature Review

Heuristic methods

- ✓ k-means (MacQueen 1967; Lloyd 1982) heuristic ($\approx$ 3M references in Google Scholar 2024) is the most popular heuristic for solving MSSC.
- ✓ a lot of research has been dedicated to finding efficient initialization for k-means (Arthur and Vassilvitskii 2006; Yu, Chu, Wang, Chan, and Chang 2018; Franti and Sieranoja 2019)
- ✓ Standard metaheuristic algorithms:simulated annealing (Lee and Perkins 2021), tabu search (Al-Sultan 1995), variable neighborhood search (Hansen and Mladenovic 2001; Orlov, Kazakovtsev, Rozhnov, Popov, and Fedosov 2018), iterated local search (Likas, Vlassis, and Verbeek 2003), evolutionary algorithms (Maulik and Bandyopadhyay 2000; Karmitsa, Bagirov, and Taheri 1997).

Literature Review

Heuristic methods

- ✓ *k*-means (MacQueen 1967; Lloyd 1982) heuristic ($\approx$ 3M references in Google Scholar 2024) is the most popular heuristic for solving MSSC.
- ✓ a lot of research has been dedicated to finding efficient initialization for *k*-means (Arthur and Vassilvitskii 2006; Yu, Chu, Wang, Chan, and Chang 2018; Franti and Sieranoja 2019)
- ✓ Standard metaheuristic algorithms:simulated annealing (Lee and Perkins 2021), tabu search (Al-Sultan 1995), variable neighborhood search (Hansen and Mladenovic 2001; Orlov, Kazakovtsev, Rozhnov, Popov, and Fedosov 2018), iterated local search (Likas, Vlassis, and Verbeek 2003), evolutionary algorithms (Maulik and Bandyopadhyay 2000; Karmitsa, Bagirov, and Taheri 1997).
- ✓ DC (Difference of Convex functions) programming for clustering large datasets (Tao et al. 2014; Bagirov, Taheri, and Ugon 2016; Karmitsa, Bagirov, and Taheri 2017; Karmitsa, Bagirov, and Taheri 2018).

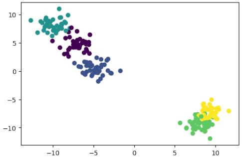
Literature Review

Heuristic methods

- ✓ k-means (MacQueen 1967; Lloyd 1982) heuristic (≈ 3M references in Google Scholar 2024) is the most popular heuristic for solving MSSC.
- ✓ a lot of research has been dedicated to finding efficient initialization for k-means (Arthur and Vassilvitskii 2006; Yu, Chu, Wang, Chan, and Chang 2018; Franti and Sieranoja 2019)
- ✓ Standard metaheuristic algorithms:simulated annealing (Lee and Perkins 2021), tabu search (Al-Sultan 1995), variable neighborhood search (Hansen and Mladenovic 2001; Orlov, Kazakovtsev, Rozhnov, Popov, and Fedosov 2018), iterated local search (Likas, Vlassis, and Verbeek 2003), evolutionary algorithms (Maulik and Bandyopadhyay 2000; Karmitsa, Bagirov, and Taheri 1997).
- ✓ DC (Difference of Convex functions) programming for clustering large datasets (Tao et al. 2014; Bagirov, Taheri, and Ugon 2016; Karmitsa, Bagirov, and Taheri 2017; Karmitsa, Bagirov, and Taheri 2018).
- ✓ The k-means algorithm is also used as a local search subroutine in various algorithms, such as population-based metaheuristics (Mansueto and Schoen 2021)

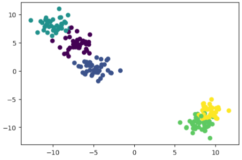
The importance of the global minimum

200 points in the plane $k = 5$

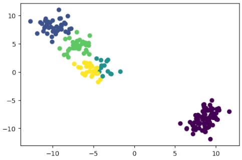


The importance of the global minimum

200 points in the plane $k = 5$

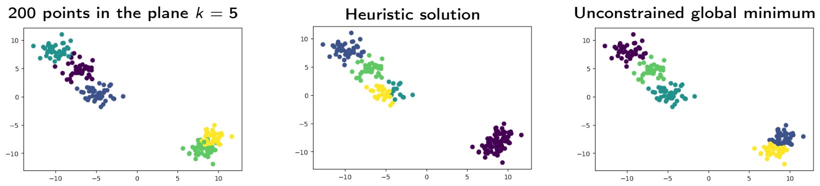


Heuristic solution



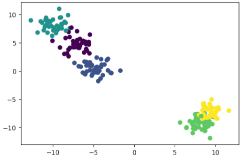
SAPIENZA
UNIVERSITÀ DI ROMA

The importance of the global minimum

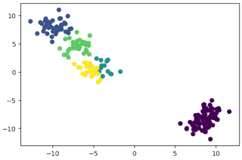


The importance of the global minimum

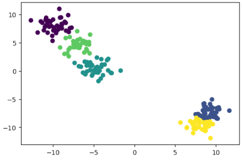
200 points in the plane $k = 5$



Heuristic solution



Unconstrained global minimum



Interpretation derived by the heuristic solution can be completely wrong!

SAPIENZA
UNIVERSITÀ DI ROMA

Literature Review - Exact

Three main approaches:

B& B (Koontz, Narendra, and Fukunaga 1975), (Diehr 1985), and RBBA (Brusco 2006) are B& B algorithms where the lower bound is computed by solving smaller instances and exploiting the properties of MSSC. (Sherali and Desai 2005) employs the reformulation-linearization technique (RLT) to derive lower bounds by transforming the non-linear problem into a 0-1 mixed-integer program. (Burgard, Moreira Costa, Hojny, Kleinert, and Schmidt 2023) focus on mixed-integer programming techniques to improve solver performance.

Literature Review - Exact

Three main approaches:

B& B (Koontz, Narendra, and Fukunaga 1975), (Diehr 1985), and RBBA (Brusco 2006) are B& B algorithms where the lower bound is computed by solving smaller instances and exploiting the properties of MSSC. (Sherali and Desai 2005) employs the reformulation-linearization technique (RLT) to derive lower bounds by transforming the non-linear problem into a 0-1 mixed-integer program. (Burgard, Moreira Costa, Hojny, Kleinert, and Schmidt 2023) focus on mixed-integer programming techniques to improve solver performance.

Col Gen (Du Merle, Hansen, Jaumard, and Mladenovic 1999) propose a column generation approach where the restricted master problem is solved using an interior point method and the auxiliary problem using a hyperbolic program with binary variables to find a column with negative reduced cost. The approach has been improved in (Aloise, Hansen, and Liberti 2012a), and recently in (Sudoso and Aloise 2024), where they can solve problems **in the plane** up to 6000 points.

Literature Review - Exact

Three main approaches:

B& B (Koontz, Narendra, and Fukunaga 1975), (Diehr 1985), and RBBA (Brusco 2006) are B& B algorithms where the lower bound is computed by solving smaller instances and exploiting the properties of MSSC. (Sherali and Desai 2005) employs the reformulation-linearization technique (RLT) to derive lower bounds by transforming the non-linear problem into a 0-1 mixed-integer program. (Burgard, Moreira Costa, Hojny, Kleinert, and Schmidt 2023) focus on mixed-integer programming techniques to improve solver performance.

Col Gen (Du Merle, Hansen, Jaumard, and Mladenovic 1999) propose a column generation approach where the restricted master problem is solved using an interior point method and the auxiliary problem using a hyperbolic program with binary variables to find a column with negative reduced cost. The approach has been improved in (Aloise, Hansen, and Liberti 2012a), and recently in (Sudoso and Aloise 2024), where they can solve problems **in the plane** up to 6000 points.

SDP Peng (Peng and Xia 2005; Peng and Wei 2007) showed the equivalence between MSSC and a 0-1 SDP reformulation. Aloise and Hansen 2009 developed a branch-and-cut algorithm based on the linear programming relaxation of the 0-1 SDP model. Piccialli, Sudoso, and Wiegele 2022 proposed SOS-SDP, a branch-and-bound algorithm using SDP relaxation and polyhedral cuts, capable of solving real-world instances with up to 4.000 data points.

SOS-SDP

SOS-SDP is a branch and bound approach for solving MSSC problem with the following ingredients:

SOS-SDP

SOS-SDP is a branch and bound approach for solving MSSC problem with the following ingredients:

1. an SDP relaxation for computing the lower bound at each node solved by means of a cutting plane algorithm

SOS-SDP

SOS-SDP is a branch and bound approach for solving MSSC problem with the following ingredients:

1. an SDP relaxation for computing the lower bound at each node solved by means of a cutting plane algorithm
2. a primal heuristic to compute an upper bound that heavily relies on the solution of the SDP relaxation

SAPIENZA UNIVERSITÀ DI ROMA

SOS-SDP

SOS-SDP is a branch and bound approach for solving MSSC problem with the following ingredients:

1. an SDP relaxation for computing the lower bound at each node solved by means of a cutting plane algorithm
2. a primal heuristic to compute an upper bound that heavily relies on the solution of the SDP relaxation
3. a branching rule based on the problem

SOS-SDP

SOS-SDP is a branch and bound approach for solving MSSC problem with the following ingredients:

1. an SDP relaxation for computing the lower bound at each node solved by means of a cutting plane algorithm
2. a primal heuristic to compute an upper bound that heavily relies on the solution of the SDP relaxation
3. a branching rule based on the problem
4. We manage to exploit the must link constraints to reduce the size of the SDPs for computing the lower bound

SAPIENZA
UNIVERSITÀ DI ROMA

SDP reformulation

Unconstrained Problem (MSSC) can be reformulated as a **nonlinear** SDP problem [Peng & Wei 2007]:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}\sum_{j=1}^{k} \delta_{ij} ||x_i - m_j||_2^2 \\ & \sum_{j=1}^{k} \delta_{ij} = 1 \quad i = 1, \dots, n \\ & \delta_{ij} \in \{0.1\} \\ & m_j \in \mathbb{R}^d \quad i = 1, \dots, n, \; j = 1, \dots, k \end{aligned} \qquad \text{(MSSC)}$$

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \text{trace}(Z) = k \\ & Z \geq 0 \\ & Z = Z^T \\ & Z^2 = Z \end{aligned} \qquad \text{(SDP} - \text{MSSC)}$$

where $W \in \mathbb{R}^{n \times d}$ is the data matrix obtained by stacking the data points x_i for all i, $e \in \mathbb{R}^n$ is the vector of all ones and $I \in \mathbb{R}^{n \times n}$ is the identity matrix.

SAPIENZA UNIVERSITÀ DI ROMA

SDP reformulation

Unconstrained Problem (MSSC) can be reformulated as a **nonlinear** SDP problem [Peng & Wei 2007]:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{k} \delta_{ij}||x_i - m_j||_2^2 \\
& \sum_{j=1}^{k} \delta_{ij} = 1 \quad i = 1,\dots,n \\
& \delta_{ij} \in \{0.1\} \\
& m_j \in \mathbb{R}^d \quad i = 1,\dots,n,\ j = 1,\dots,k
\end{aligned} \qquad \text{(MSSC)}
$$

$$
\begin{aligned}
\min \quad & \langle WW^T, I - Z\rangle \\
& Ze = e \\
& \text{trace}\,(Z) = k \\
& Z \geq 0 \\
& Z = Z^T \\
& Z^2 = Z
\end{aligned} \qquad \text{(SDP} - \text{MSSC)}
$$

where $W \in \mathbb{R}^{n\times d}$ is the data matrix obtained by stacking the data points x_i for all i, $e \in \mathbb{R}^n$ is the vector of all ones and $I \in \mathbb{R}^{n\times n}$ is the identity matrix.

Problem (MSSC) and (SDP-MSSC) are equivalent.

SAPIENZA UNIVERSITÀ DI ROMA

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

- ✓ If i and j are in the same cluster C:
  - rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

✓ If i and j are in the same cluster C:

- rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
- non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.

SAPIENZA UNIVERSITÀ DI ROMA

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

- ✓ If i and j are in the same cluster C:
  - rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
  - non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.
- ✓ If i and j are not in the same cluster:

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

- ✓ If i and j are in the same cluster C:
  - rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
  - non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.
- ✓ If i and j are not in the same cluster:
  - $Z_{ij} = 0$

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

✓ If i and j are in the same cluster C:

- rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
- non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.

✓ If i and j are not in the same cluster:

- $Z_{ij} = 0$

Remark

Instance-level constraints can be translated into linear constraints on Z

SAPIENZA UNIVERSITÀ DI ROMA

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

- ✓ If i and j are in the same cluster C:
  - rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
  - non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.
- ✓ If i and j are not in the same cluster:
  - $Z_{ij} = 0$

Remark

Instance-level constraints can be translated into linear constraints on Z

- As an example, we consider 6 points in 3 clusters :

SAPIENZA UNIVERSITÀ DI ROMA

Clustering matrix

Any feasible $Z \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with a special structure.

- ✓ If i and j are in the same cluster C:
  - rows i and j of Z are equal, i.e. $Z_{i\cdot} = Z_{j\cdot}$
  - non-zero entries of rows i and j are equal to $\frac{1}{|C|}$, where $|C|$ is the cardinality of the cluster C.
- ✓ If i and j are not in the same cluster:
  - $Z_{ij} = 0$

Remark

Instance-level constraints can be translated into linear constraints on Z

- As an example, we consider 6 points in 3 clusters :
  - $x_1, x_2, x_3, x_4, x_5, x_6,$ $\quad Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$

SAPIENZA UNIVERSITÀ DI ROMA

SDP relaxation

Problem (SDP-MSSC) can be rewritten as an SDP with an additional rank constraint:

$$
\begin{aligned}
\min \quad & \langle WW^T, I - Z \rangle \\
& Ze = e \\
& \mathrm{trace}\,(Z) = k \\
& Z \geq 0 \\
& Z = Z^T \\
& Z^2 = Z
\end{aligned}
\qquad \text{(SDP} - \text{MSSC)}
$$

SAPIENZA UNIVERSITÀ DI ROMA

SDP relaxation

Problem (SDP-MSSC) can be rewritten as an SDP with an additional rank constraint:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & Z \geq 0 \\ & Z = Z^T \\ & Z^2 = Z \end{aligned} \qquad (\mathrm{SDP-MSSC})$$

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & \mathrm{rank}\,(Z) = k \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad (\mathrm{SDP-MSSC})$$

SAPIENZA UNIVERSITÀ DI ROMA

SDP relaxation

Problem (SDP-MSSC) can be rewritten as an SDP with an additional rank constraint:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \operatorname{trace}(Z) = k \\ & Z \geq 0 \\ & Z = Z^T \\ & Z^2 = Z \end{aligned} \qquad \text{(SDP} - \text{MSSC)}$$

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \operatorname{trace}(Z) = k \\ & \operatorname{rank}(Z) = k \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad \text{(SDP} - \text{MSSC)}$$

By dropping the non-convex constraint $rank(Z) = k$, we obtain the Semidefinite Programming (SDP) relaxation [Peng, Wei 2007]

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \operatorname{trace}(Z) = k \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad \text{(SDP} - \text{REL)}$$

SDP relaxation

Problem (SDP-MSSC) can be rewritten as an SDP with an additional rank constraint:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z\rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & Z \geq 0 \\ & Z = Z^T \\ & Z^2 = Z \end{aligned} \qquad \text{(SDP} - \text{MSSC)}$$

$$\begin{aligned} \min \quad & \langle WW^T, I - Z\rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & \mathrm{rank}\,(Z) = k \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad \text{(SDP} - \text{MSSC)}$$

By dropping the non-convex constraint $rank(Z) = k$, we obtain the Semidefinite Programming (SDP) relaxation [Peng, Wei 2007]

$$\begin{aligned} \min \quad & \langle WW^T, I - Z\rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad \text{(SDP} - \text{REL)}$$

In order to tighten the bound, we add valid inequalities of different classes with a cutting plane procedure.

SAPIENZA UNIVERSITÀ DI ROMA

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Pair we know that in any clustering $Z_{ij} \leq Z_{ii} \quad \forall i, j$.

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Pair we know that in any clustering $Z_{ij} \leq Z_{ii} \quad \forall i, j$.

Triangle if i and j are in the same cluster and i and h are in the same cluster, then j and h must be in the same cluster [Peng, Wei 2007], that translates into

$$Z_{ij} + Z_{ih} \leq Z_{ii} + Z_{jh} \quad \forall i, j, h.$$

SAPIENZA UNIVERSITÀ DI ROMA

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Pair we know that in any clustering $Z_{ij} \leq Z_{ii} \quad \forall i, j$.

Triangle if i and j are in the same cluster and i and h are in the same cluster, then j and h must be in the same cluster [Peng, Wei 2007], that translates into

$$Z_{ij} + Z_{ih} \leq Z_{ii} + Z_{jh} \quad \forall i, j, h.$$

Clique if the number of clusters is k, given any subset Q of $k + 1$ points, it must hold that at least two points have to be in the same cluster:

$$\sum_{(i,j) \in Q, i<j} Z_{ij} \geq \frac{1}{n - k + 1} \quad \forall Q \subset \{1, \dots, n\}, |Q| = k + 1$$

SAPIENZA
UNIVERSITÀ DI ROMA

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Pair we know that in any clustering $Z_{ij} \leq Z_{ii} \quad \forall i, j$.

Triangle if i and j are in the same cluster and i and h are in the same cluster, then j and h must be in the same cluster [Peng, Wei 2007], that translates into

$$Z_{ij} + Z_{ih} \leq Z_{ii} + Z_{jh} \quad \forall i, j, h.$$

Clique if the number of clusters is k, given any subset Q of $k+1$ points, it must hold that at least two points have to be in the same cluster:

$$\sum_{(i,j) \in Q, i<j} Z_{ij} \geq \frac{1}{n-k+1} \quad \forall Q \subset \{1, \ldots, n\}, |Q| = k+1$$

We search for violated inequalities and keep adding them until there is a bound improvement.

Cutting plane

In order to tighten the bound, it is possible to add valid inequalities of the following classes:

Pair we know that in any clustering $Z_{ij} \leq Z_{ii} \quad \forall i, j$.

Triangle if i and j are in the same cluster and i and h are in the same cluster, then j and h must be in the same cluster [Peng, Wei 2007], that translates into

$$Z_{ij} + Z_{ih} \leq Z_{ii} + Z_{jh} \quad \forall i, j, h.$$

Clique if the number of clusters is k, given any subset Q of $k+1$ points, it must hold that at least two points have to be in the same cluster:

$$\sum_{(i,j)\in Q, i<j} Z_{ij} \geq \frac{1}{n-k+1} \quad \forall Q \subset \{1, \dots, n\}, |Q| = k+1$$

We search for violated inequalities and keep adding them until there is a bound improvement.

We remove at each iteration the added inequalities that are not active to keep the size of the SDP tractable

Branching Decision

Cannot Link Node Points i and j in different clusters:

SAPIENZA
UNIVERSITÀ DI ROMA

Branching Decision

Cannot Link Node Points i and j in different clusters:

- we add the constraint $Z_{ij} = 0$ to the SDP relaxation

Branching Decision

Cannot Link Node Points i and j in different clusters:

- we add the constraint $Z_{ij} = 0$ to the SDP relaxation

Must Link Node Points i and j in the same cluster:

Branching Decision

Cannot Link Node Points i and j in different clusters:

- we add the constraint $Z_{ij} = 0$ to the SDP relaxation

Must Link Node Points i and j in the same cluster:

- we add the constraint that row $Z_{i\cdot}$ and row $Z_{j\cdot}$ are equal to the SDP relaxation:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & Z_{i\cdot} = Z_{j\cdot} \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad (\mathrm{ML_{ij}})$$

How do we choose the branching pair i and j?

SAPIENZA UNIVERSITÀ DI ROMA

Branching Decision

Cannot Link Node Points i and j in different clusters:

- we add the constraint $Z_{ij} = 0$ to the SDP relaxation

Must Link Node Points i and j in the same cluster:

- we add the constraint that row $Z_{i\cdot}$ and row $Z_{j\cdot}$ are equal to the SDP relaxation:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z \rangle \\ & Ze = e \\ & \mathrm{trace}(Z) = k \\ & Z_{i\cdot} = Z_{j\cdot} \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad (\mathrm{ML_{ij}})$$

How do we choose the branching pair i and j?

In a matrix Z corresponding to a clustering, for each pair (i, j) either $Z_{ij} = 0$ or $Z_{i\cdot} = Z_{j\cdot}$. Select a pair of data points to branch on:

$$\max_{i,j}\{\min\{Z_{ij}, \|Z_{i\cdot} - Z_{j\cdot}\|^2\}\}$$

SAPIENZA Università di Roma

Branching Decision

Cannot Link Node Points i and j in different clusters:

- we add the constraint $Z_{ij} = 0$ to the SDP relaxation

Must Link Node Points i and j in the same cluster:

- we add the constraint that row $Z_{i\cdot}$ and row $Z_{j\cdot}$ are equal to the SDP relaxation:

$$\begin{aligned} \min \quad & \langle WW^T, I - Z\rangle \\ & Ze = e \\ & \mathrm{trace}\,(Z) = k \\ & Z_{i\cdot} = Z_{j\cdot} \\ & Z \geq 0 \\ & Z \succeq 0 \end{aligned} \qquad (\mathrm{ML_{ij}})$$

How do we choose the branching pair i and j?

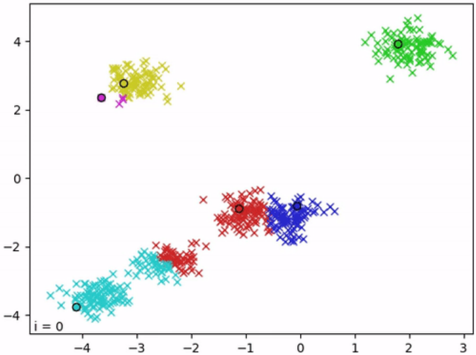
In a matrix Z corresponding to a clustering, for each pair (i, j) either $Z_{ij} = 0$ or $Z_{i\cdot} = Z_{j\cdot}$. Select a pair of data points to branch on:

$$\max_{i,j}\{\min\{Z_{ij}, \|Z_{i\cdot} - Z_{j\cdot}\|^2\}\}$$

Note:

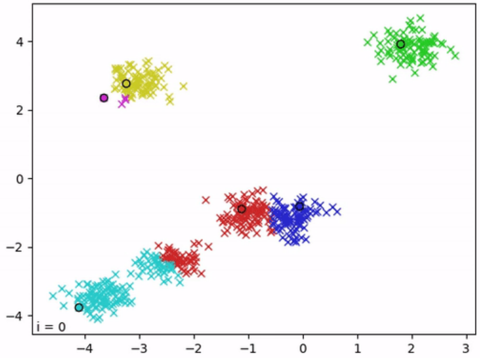
In the B&B nodes we add **instance level constraints**!

K-means Algorithm



Kmeans alternates two phases:

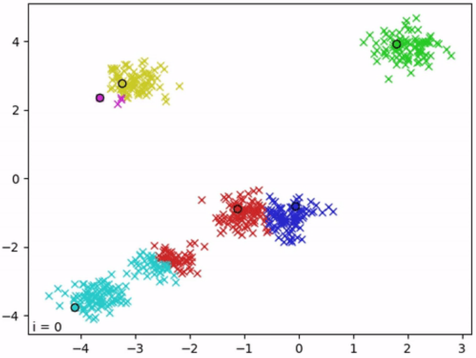
K-means Algorithm



Kmeans alternates two phases:

1. Optimal assignment of the points to the clusters given the current centers

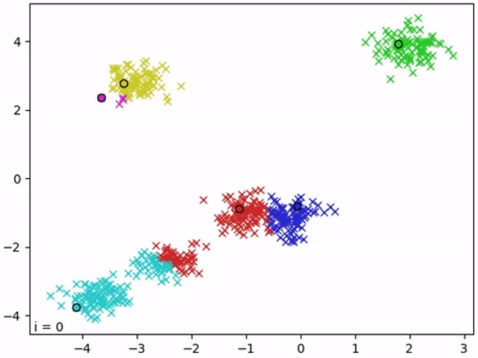
K-means Algorithm



Kmeans alternates two phases:

1. Optimal assignment of the points to the clusters given the current centers
2. Optimal choice of the centers given the points currently in the cluster

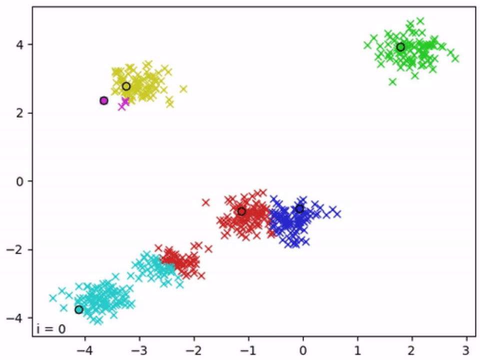
K-means Algorithm



Kmeans alternates two phases:

1. Optimal assignment of the points to the clusters given the current centers
2. Optimal choice of the centers given the points currently in the cluster
3. The algorithm stops when the solution does not change

K-means Algorithm



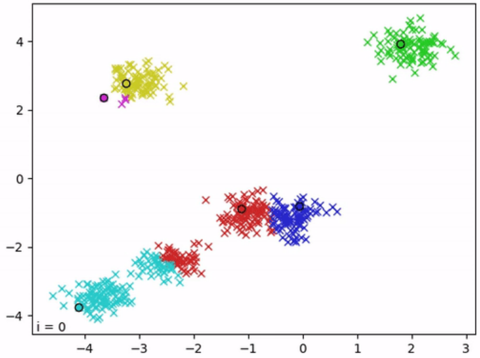
Kmeans alternates two phases:

1. Optimal assignment of the points to the clusters given the current centers
2. Optimal choice of the centers given the points currently in the cluster
3. The algorithm stops when the solution does not change

The produced solution is suboptimal and **heavily depends on the choice of the centers**

SAPIENZA UNIVERSITÀ DI ROMA

K-means Algorithm



Kmeans alternates two phases:

1. Optimal assignment of the points to the clusters given the current centers
2. Optimal choice of the centers given the points currently in the cluster
3. The algorithm stops when the solution does not change

The produced solution is suboptimal and **heavily depends on the choice of the centers**

At each node, we have additional constraints (must link and cannot link), we need a **constrained version**!

IPC-k-means (Baumann 2020)

Input Data points $p_1, \ldots, p_n$, initial cluster centers $m_1, \ldots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

IPC-k-means (Baumann 2020)

Input Data points $p_1, \ldots, p_n$, initial cluster centers $m_1, \ldots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

Until Convergence

IPC-k-means (Baumann 2020)

Input Data points $p_1, \ldots, p_n$, initial cluster centers $m_1, \ldots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

1. Compute the optimal cluster assignments $x_{ij}^\star$ by solving:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \|p_i - m_j\|_2^2 \\
& \textstyle\sum_{j=1}^{k} x_{ij} = 1 \quad \forall i \in \mathcal{N} \\
& \textstyle\sum_{i=1}^{k} x_{ij} \geq 1 \quad \forall j \in \mathcal{K} \\
& x_{ih} = x_{jh} \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{ML} \\
& x_{ih} + x_{jh} \leq 1 \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{CL} \\
& x_{ij} \in \{0.1\} \quad \forall i \in \mathcal{N},\ \forall j \in \mathcal{K}
\end{aligned}
$$

Until Convergence

IPC-k-means (Baumann 2020)

Input Data points $p_1, \ldots, p_n$, initial cluster centers $m_1, \ldots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

1. Compute the optimal cluster assignments $x^\star_{ij}$ by solving:

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \|p_i - m_j\|_2^2 \\ & \textstyle\sum_{j=1}^{k} x_{ij} = 1 \quad \forall i \in \mathcal{N} \\ & \textstyle\sum_{i=1}^{k} x_{ij} \geq 1 \quad \forall j \in \mathcal{K} \\ & x_{ih} = x_{jh} \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{ML} \\ & x_{ih} + x_{jh} \leq 1 \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{CL} \\ & x_{ij} \in \{0.1\} \quad \forall i \in \mathcal{N},\ \forall j \in \mathcal{K} \end{aligned}$$

2. Set $C_j \leftarrow \{p_i : x^\star_{ij} = 1\}$ for each $j = 1, \ldots, k$.

Until Convergence

IPC-k-means (Baumann 2020)

Input Data points $p_1, \dots, p_n$, initial cluster centers $m_1, \dots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

1. Compute the optimal cluster assignments $x_{ij}^\star$ by solving:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \|p_i - m_j\|_2^2 \\
& \sum_{j=1}^{k} x_{ij} = 1 \quad \forall i \in \mathcal{N} \\
& \sum_{i=1}^{k} x_{ij} \geq 1 \quad \forall j \in \mathcal{K} \\
& x_{ih} = x_{jh} \quad \forall h \in \mathcal{K}, \ \forall (i,j) \in \mathcal{ML} \\
& x_{ih} + x_{jh} \leq 1 \quad \forall h \in \mathcal{K}, \ \forall (i,j) \in \mathcal{CL} \\
& x_{ij} \in \{0.1\} \quad \forall i \in \mathcal{N}, \ \forall j \in \mathcal{K}
\end{aligned}
$$

2. Set $C_j \leftarrow \{p_i : x_{ij}^\star = 1\}$ for each $j = 1, \dots, k$.
3. Update the cluster centers $m_1, \dots, m_k$ by taking the mean of the data points assigned to each cluster $C_1, \dots, C_k$.

Until Convergence

SAPIENZA UNIVERSITÀ DI ROMA

IPC-k-means (Baumann 2020)

Input Data points $p_1, \dots, p_n$, initial cluster centers $m_1, \dots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

1. Compute the optimal cluster assignments $x^\star_{ij}$ by solving:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^n \sum_{j=1}^k x_{ij} \|p_i - m_j\|_2^2 \\
& \textstyle\sum_{j=1}^k x_{ij} = 1 \quad \forall i \in \mathcal{N} \\
& \textstyle\sum_{i=1}^k x_{ij} \geq 1 \quad \forall j \in \mathcal{K} \\
& x_{ih} = x_{jh} \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{ML} \\
& x_{ih} + x_{jh} \leq 1 \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{CL} \\
& x_{ij} \in \{0.1\} \quad \forall i \in \mathcal{N},\ \forall j \in \mathcal{K}
\end{aligned}
$$

2. Set $C_j \leftarrow \{p_i : x^\star_{ij} = 1\}$ for each $j = 1, \dots, k$.
3. Update the cluster centers $m_1, \dots, m_k$ by taking the mean of the data points assigned to each cluster $C_1, \dots, C_k$.

Until Convergence

Return Clusters $C_1, \dots, C_k$.

IPC-k-means (Baumann 2020)

Input Data points $p_1, \dots, p_n$, initial cluster centers $m_1, \dots, m_k$, must-link $\mathcal{ML}$ and cannot-link $\mathcal{CL}$ constraints

Repeat

1. Compute the optimal cluster assignments $x_{ij}^\star$ by solving:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} \|p_i - m_j\|_2^2 \\
& \textstyle\sum_{j=1}^{k} x_{ij} = 1 \quad \forall i \in \mathcal{N} \\
& \textstyle\sum_{i=1}^{k} x_{ij} \geq 1 \quad \forall j \in \mathcal{K} \\
& x_{ih} = x_{jh} \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{ML} \\
& x_{ih} + x_{jh} \leq 1 \quad \forall h \in \mathcal{K},\ \forall (i,j) \in \mathcal{CL} \\
& x_{ij} \in \{0.1\} \quad \forall i \in \mathcal{N},\ \forall j \in \mathcal{K}
\end{aligned}
$$

2. Set $C_j \leftarrow \{p_i : x_{ij}^\star = 1\}$ for each $j = 1, \dots, k$.
3. Update the cluster centers $m_1, \dots, m_k$ by taking the mean of the data points assigned to each cluster $C_1, \dots, C_k$.

Until Convergence

Return Clusters $C_1, \dots, C_k$.

Center Initialization

The center initialization is based on the SDP solution

Results

SOS-SDP solves instances up to **4000** points, before up to 1000 only in the plane or 2300 but for $k \geq 230$ (n/k small) (Aloise, Hansen, and Liberti 2012b). Recent paper by Aloise and Sudoso up to 6000 datapoints on the plane. Our code: `https://github.com/antoniosudoso/sos-sdp`

SAPIENZA
UNIVERSITÀ DI ROMA

Results

SOS-SDP solves instances up to **4000** points, before up to 1000 only in the plane or 2300 but for $k \geq 230$ (n/k small) (Aloise, Hansen, and Liberti 2012b). Recent paper by Aloise and Sudoso up to 6000 datapoints on the plane. Our code: `https://github.com/antoniosudoso/sos-sdp`

Interesting point: the bound is so strong, that in general **the gap at the root node is small**

Results

SOS-SDP solves instances up to **4000** points, before up to 1000 only in the plane or 2300 but for $k \geq 230$ (n/k small) (Aloise, Hansen, and Liberti 2012b). Recent paper by Aloise and Sudoso up to 6000 datapoints on the plane. Our code: `https://github.com/antoniosudoso/sos-sdp`

Interesting point: the bound is so strong, that in general **the gap at the root node is small**

Can we exploit SOS-SDP to give some optimality guarantees for large scale (10000 points) instances?

Lower bound

Let the dataset $O = \{p_1, \ldots, p_N\}$ be partitioned into T subsets $\{S_1, \ldots, S_T\}$ such that $\cup_{t=1}^{T} S_t = O$ and $S_i \cap S_j = \emptyset$

Lower bound

Let the dataset $O = \{p_1, \ldots, p_N\}$ be partitioned into T subsets $\{S_1, \ldots, S_T\}$ such that $\cup_{t=1}^{T} S_t = O$ and $S_i \cap S_j = \emptyset$

Assume also that the optimal value of the MSSC problem on each subset is available, i.e. let

$$MSSC(S_t, k) = \min_{\delta_{ij}^t} \sum_{j=1}^{K} \sum_{i \in S_t} \delta_{ij}^t \|p_i - \mu_j^t\|^2 \tag{1a}$$

$$\text{s.t.} \sum_{j=1}^{K} \delta_{ij}^t = 1, \quad \forall i \in S_t \tag{1b}$$

$$\sum_{i \in S_t} \delta_{ij}^t \geq 1, \quad \forall j \in \{1, \ldots, K\} \tag{1c}$$

$$\delta_{ij}^t \in \{0, 1\}, \quad \forall i \in S_t \; \forall j \in \{1, \ldots, K\}. \tag{1d}$$

Lower bound

Let the dataset $O = \{p_1, \dots, p_N\}$ be partitioned into T subsets $\{S_1, \dots, S_T\}$ such that $\cup_{t=1}^{T} S_t = O$ and $S_i \cap S_j = \emptyset$

Assume also that the optimal value of the MSSC problem on each subset is available, i.e. let

$$MSSC(S_t, k) = \min_{\delta_{ij}^t} \sum_{j=1}^{K} \sum_{i \in S_t} \delta_{ij}^t \|p_i - \mu_j^t\|^2 \qquad (1a)$$

$$\text{s.t.} \sum_{j=1}^{K} \delta_{ij}^t = 1, \quad \forall i \in S_t \qquad (1b)$$

$$\sum_{i \in S_t} \delta_{ij}^t \geq 1, \quad \forall j \in \{1, \dots, K\} \qquad (1c)$$

$$\delta_{ij}^t \in \{0, 1\}, \quad \forall i \in S_t \; \forall j \in \{1, \dots, K\}. \qquad (1d)$$

Lower Bound

$$MSSC(O, k) \geq \sum_{t=1}^{T} MSSC(S_t, k) \geq \sum_{t=1}^{T} LB(S_t, k). \qquad LB$$

for any valid lower bound $LB(S_t, k)$ on the objective of Problem (1).

Literature

This lower bound has been exploited in (Koontz, Narendra, and Fukunaga 1975) and (Diehr 1985):

Literature

This lower bound has been exploited in (Koontz, Narendra, and Fukunaga 1975) and (Diehr 1985):

They solve smaller subproblems by enumerations, and use a similar version of the lower bound. For well-separated clusters in two-dimensional space, (Diehr 1985) obtained optimal partitions for problems with $ns = 120$ objects and $k = 4$ clusters. However, for randomly generated data in the two-dimensional plane, the largest problem instance solved to optimality consisted of only $n = 30$ objects grouped into $k = 4$ clusters

SAPIENZA UNIVERSITÀ DI ROMA

Literature

This lower bound has been exploited in (Koontz, Narendra, and Fukunaga 1975) and (Diehr 1985):

They solve smaller subproblems by enumerations, and use a similar version of the lower bound. For well-separated clusters in two-dimensional space, (Diehr 1985) obtained optimal partitions for problems with $ns = 120$ objects and $k = 4$ clusters. However, for randomly generated data in the two-dimensional plane, the largest problem instance solved to optimality consisted of only $n = 30$ objects grouped into $k = 4$ clusters

The bound is used again in (Brusco 2006), where the algorithm begins with the application of branch-and-bound for $k + 1$ objects and subsequently adds objects, one at a time, until all n objects are included. Each time a new object is added, the branch-and-bound algorithm is repeated (or reapplied). The algorithm is improved by a smart ordering of the data points

Idea: decomposition in smaller instances

Lower Bound

$$MSSC(O,k) \geq \sum_{t=1}^{T} MSSC(S_t,k) \geq \sum_{t=1}^{T} LB(S_t,k). \qquad LB$$

for any valid lower bound $LB(S_t, k)$ on the objective of Problem (1).

Idea: decomposition in smaller instances

Lower Bound

$$MSSC(O, k) \geq \sum_{t=1}^{T} MSSC(S_t, k) \geq \sum_{t=1}^{T} LB(S_t, k). \qquad LB$$

for any valid lower bound $LB(S_t, k)$ on the objective of Problem (1).

Good news: We can use SOS-SDP to compute $MSSC(S_t, k)$ or $LB(S_t, k)$ (few nodes of the B&B tree) **as long as the size of S_t is not too large**

Less good news: The quality of the bound heavily depends on the partition!

Idea: decomposition in smaller instances

Lower Bound

$$MSSC(O, k) \geq \sum_{t=1}^{T} MSSC(S_t, k) \geq \sum_{t=1}^{T} LB(S_t, k). \qquad LB$$

for any valid lower bound $LB(S_t, k)$ on the objective of Problem (1).

Good news: We can use SOS-SDP to compute $MSSC(S_t, k)$ or $LB(S_t, k)$ (few nodes of the B&B tree) **as long as the size of S_t is not too large**

Less good news: The quality of the bound heavily depends on the partition!

Research question

How to choose the partitions S_t in order to have a very good bound (possibly optimal)??

SAPIENZA
UNIVERSITÀ DI ROMA

Anticlustering

Ideally, we would like to **find the partition of O in subsets of equal size providing the best bound**, that is solving the following problem:

$$\max_{\xi_{it}} \sum_{t=1}^{T} MSSC(S_t, k) \tag{2a}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in \{1, \ldots, N\} \tag{2b}$$

$$\sum_{i=1}^{N} \xi_{it} = \frac{N}{T}, \quad \forall t \in \{1, \ldots, T\} \tag{2c}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in \{1, \ldots, N\}\ \forall t \in \{1, \ldots, T\}. \tag{2d}$$

SAPIENZA
UNIVERSITÀ DI ROMA

Anticlustering

Ideally, we would like to **find the partition of O in subsets of equal size providing the best bound**, that is solving the following problem:

$$\max_{\xi_{it}} \sum_{t=1}^{T} MSSC(S_t, k) \tag{2a}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in \{1, \dots, N\} \tag{2b}$$

$$\sum_{i=1}^{N} \xi_{it} = \frac{N}{T}, \quad \forall t \in \{1, \dots, T\} \tag{2c}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in \{1, \dots, N\} \; \forall t \in \{1, \dots, T\}. \tag{2d}$$

Too hard, the lower level problem is NP-hard already

SAPIENZA
UNIVERSITÀ DI ROMA

Using the lower bound for validation

In general, the computation of the lower bound is needed to prove the validity of a certain heuristic solution.

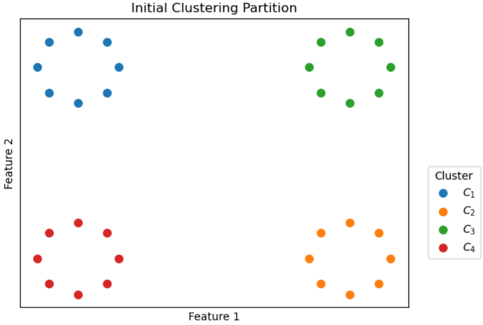


Figure: $MSSC(O, 4) = 287.82$

Using the lower bound for validation

In general, the computation of the lower bound is needed to prove the validity of a certain heuristic solution.

Assume we have the optimal solution (δ_{ij}^*) of the original MSSC problem, and a partition of O.

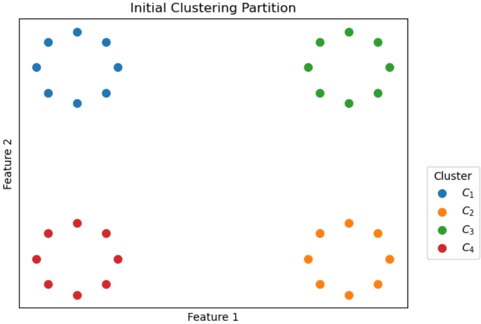
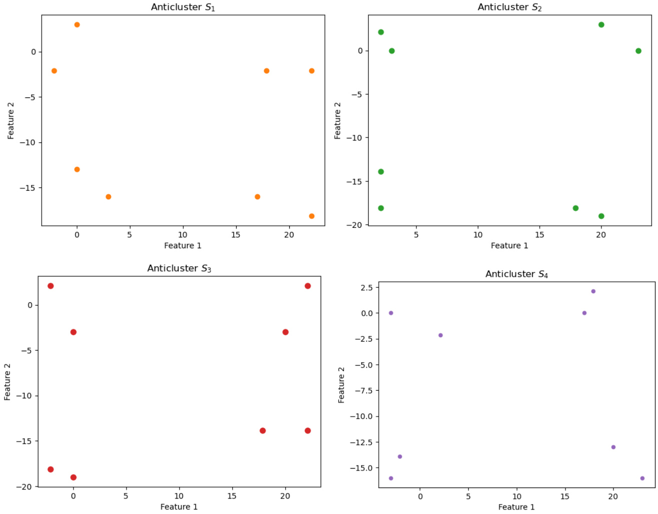
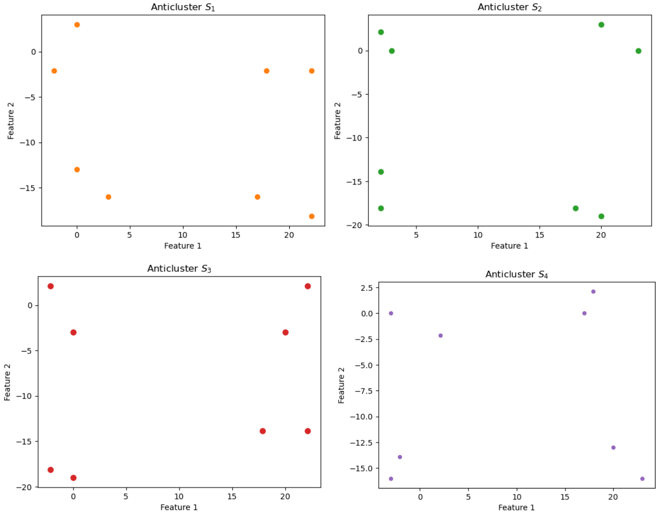


Figure: $MSSC(O, 4) = 287.82$

Partition


Anticluster S_1
Anticluster S_2
Anticluster S_3
Anticluster S_4
Feature 1
Feature 2

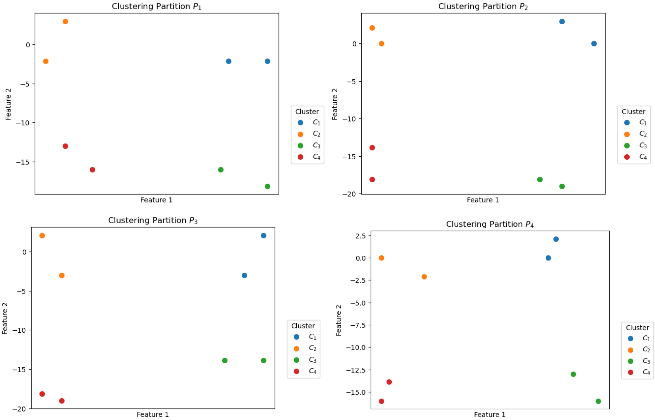
Partition



We can define the projection of the optimal solution on the single partition.

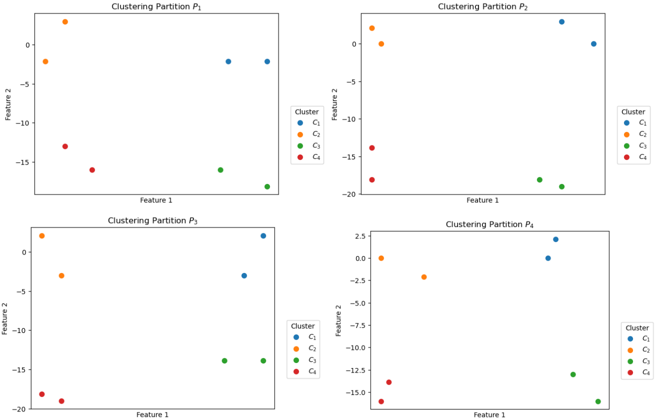
Projection

We can define the projection of δ_{ij}^*:



Projection

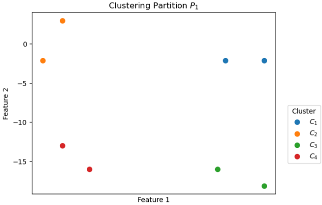
We can define the projection of δ^*_{ij}:



If the clusters are well separated, the projection is optimal for each subset (as in this example)

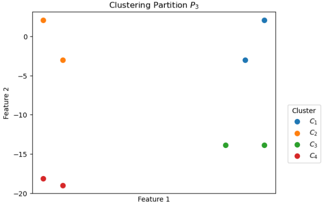
Quality of the bound

What is the quality of this partition?:

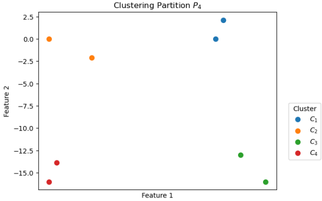


(a) $MSSC(S_{\mathbf{1}}, 4) = 48.69695$

(b) $MSSC(S_{\mathbf{2}}.4) = 23.25395$



(c) $MSSC(S_{\mathbf{3}}.4) = 42.33187$

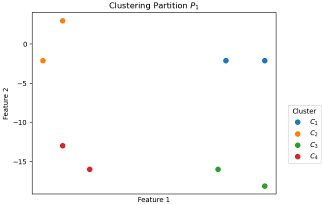


(d) $MSSC(S_{\mathbf{3}}.4) = 29.62268$

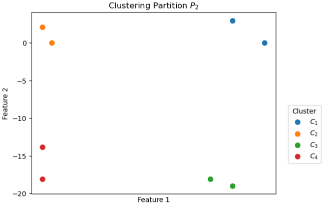
SAPIENZA UNIVERSITÀ DI ROMA

Quality of the bound

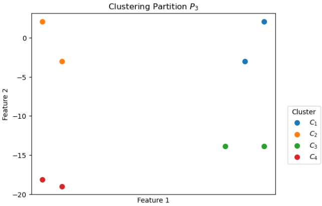
What is the quality of this partition?:



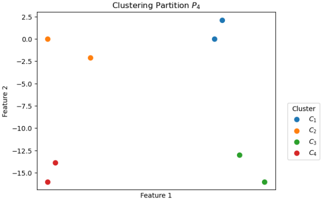
(a) $MSSC(S_{\mathbf{1}}, 4) = 48.69695$



(b) $MSSC(S_{\mathbf{2}}.4) = 23.25395$



(c) $MSSC(S_{\mathbf{3}}.4) = 42.33187$



(d) $MSSC(S_{\mathbf{3}}.4) = 29.62268$

Summing up the different contribution of each subset we get a lower bound $LB = 143.90546$ with a gap of around 50%!!!

Our problem

What makes a partition "good"?

Our problem

What makes a partition "good"?

Let's look at the objective function of MSSC:

$$\sum_{j=1}^{K} \sum_{i \in S_t} \delta_{ij}^t \|p_i - \mu_j^t\|^2$$

with

$$\mu_j^t = \frac{\sum_{i=1}^{N} p_i \delta_{ij}^t}{\sum_{i=1}^{N} \delta_{ij}^t}$$

Our problem

What makes a partition “good”?

Let’s look at the objective function of MSSC:

$$\sum_{j=1}^{K} \sum_{i \in S_t} \delta_{ij}^t \|p_i - \mu_j^t\|^2$$

with

$$\mu_j^t = \frac{\sum_{i=1}^{N} p_i \delta_{ij}^t}{\sum_{i=1}^{N} \delta_{ij}^t}$$

It can be rewritten as

$$\sum_{j=1}^{K} \frac{\sum_{i=1}^{N} \sum_{i'=1}^{N} \delta_{ij}^t \delta_{i'j}^t ||p_i - p_{i'}||^2}{\sum_{i=1}^{N} \delta_{ij}^t}$$

Our problem

What makes a partition "good"?

Let's look at the objective function of MSSC:

$$\sum_{j=1}^{K} \sum_{i \in S_t} \delta_{ij}^t \|p_i - \mu_j^t\|^2$$

with

$$\mu_j^t = \frac{\sum_{i=1}^{N} p_i \delta_{ij}^t}{\sum_{i=1}^{N} \delta_{ij}^t}$$

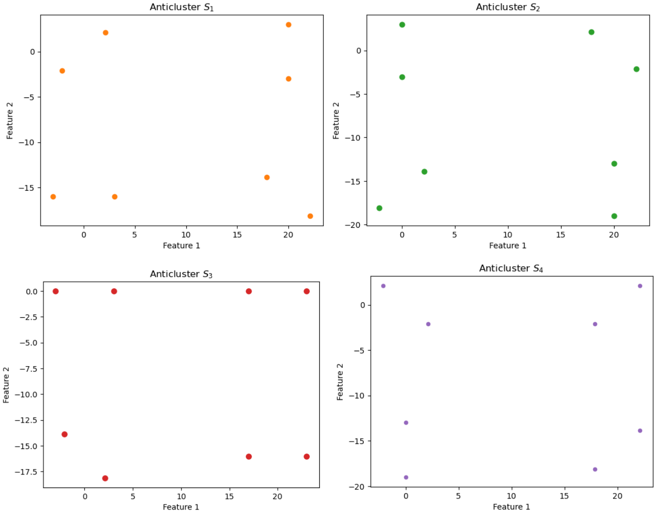
It can be rewritten as

$$\sum_{j=1}^{K} \frac{\sum_{i=1}^{N} \sum_{i'=1}^{N} \delta_{ij}^t \delta_{i'j}^t ||p_i - p_{i'}||^2}{\sum_{i=1}^{N} \delta_{ij}^t}$$

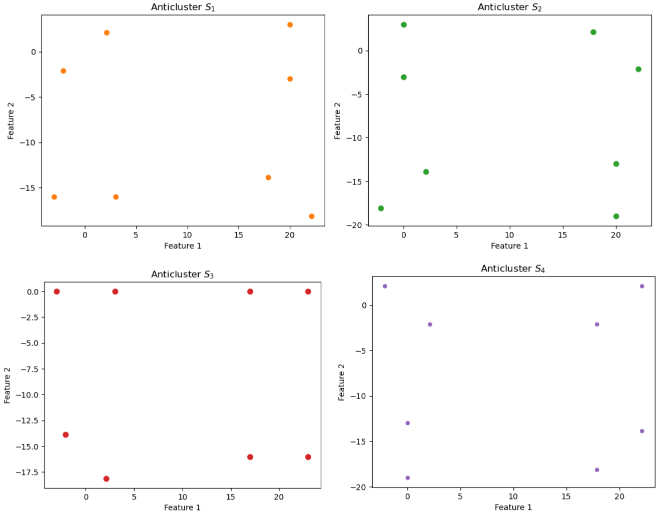
The contribution of the single anticluster is larger for larger distances among points in the same cluster (in that anticluster)

SAPIENZA
UNIVERSITÀ DI ROMA

A different partition



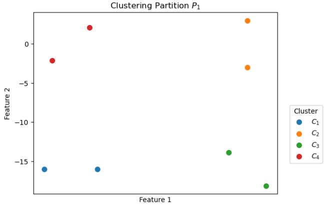
A different partition



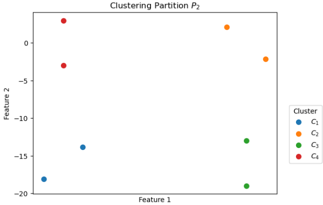
The single anticluster better "represents" the original dataset

Quality of the bound

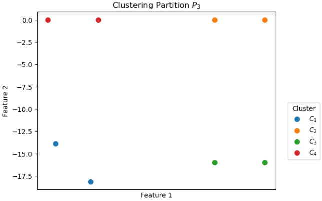
What is the quality of this partition?:



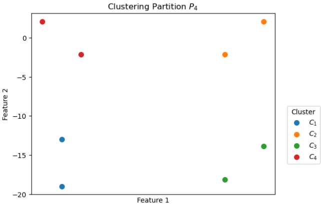
(a) $MSSC(S_{\mathbf{1}}, 4) = 71.97703$



(b) $MSSC(S_{\mathbf{2}}.4) = 71.93196$



(c) $MSSC(S_{\mathbf{3}}.4) = 71.97695$

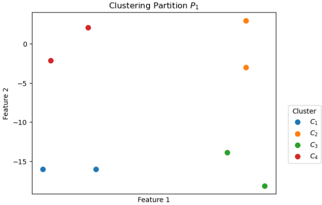


(d) $MSSC(S_{\mathbf{4}}.4) = 71.93248$

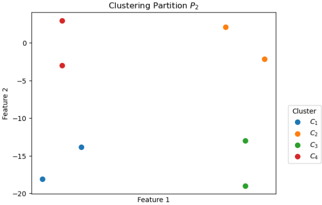
SAPIENZA
UNIVERSITÀ DI ROMA

Quality of the bound

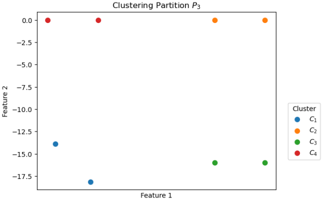
What is the quality of this partition?:



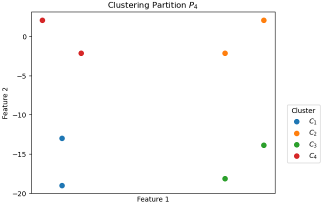
(a) $MSSC(S_{\mathbf{1}}, 4) = 71.97703$



(b) $MSSC(S_{\mathbf{2}}.4) = 71.93196$



(c) $MSSC(S_{\mathbf{3}}.4) = 71.97695$



(d) $MSSC(S_{\mathbf{4}}.4) = 71.93248$

Summing up the different contribution of each subset we get a lower bound $LB = 287.81842$ with a gap of around 0%!!!

Assumption

Assume that we have a feasible (possibly optimal) solution (δ^*_{ij}) of the original MSSC problem, and that for any partition of the dataset the projection of δ^* on the partition is still optimal. This implies that the optimal solution of the MSSC(S_t) is still δ^* restricted to S_t.

SAPIENZA UNIVERSITÀ DI ROMA

Assumption

Assumption

Assume that we have a feasible (possibly optimal) solution (δ^*_{ij}) of the original MSSC problem, and that for any partition of the dataset the projection of δ^* on the partition is still optimal. This implies that the optimal solution of the MSSC(S_t) is still δ^* restricted to S_t.

Under this assumption, problem 2 can be written as follow:

$$\max_{\xi_{it}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{i \in C(k)} \sum_{j \in C(k): j \neq i} \frac{\|p_i - p_j\|^2}{N_k / T} \cdot \xi_{it} \cdot \xi_{jt} \tag{3a}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in \{1, \dots, N\} \tag{3b}$$

$$\sum_{i \in C(k)} \xi_{it} = \frac{|C(k)|}{T}, \quad \forall t \in \{1, \dots, T\} \, \forall k \in \{1, \dots, K\} \tag{3c}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in \{1, \dots, N\} \, \forall t \in \{1, \dots, T\}. \tag{3d}$$

SAPIENZA UNIVERSITÀ DI ROMA

Problem 3 can be decomposed in K independent subproblems, one for each cluster $C(k)$ with $k \in \{1, \cdots, K\}$. We have that:

$$\max_{\xi_{it}} \sum_{t=1}^{T} \sum_{i \in C(k)} \sum_{j \in C(k): j \neq i} \frac{\|p_i - p_j\|^2}{N_k / T} \cdot \xi_{it} \cdot \xi_{jt} \tag{4a}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in C(k) \tag{4b}$$

$$\sum_{i \in C(k)} \xi_{it} = \frac{|C(k)|}{T}, \quad \forall t \in \{1, \ldots, T\} \tag{4c}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in C(k) \; \forall t \in \{1, \ldots, T\}. \tag{4d}$$

SAPIENZA
UNIVERSITÀ DI ROMA

Problem 3 can be decomposed in K independent subproblems, one for each cluster $C(k)$ with $k \in \{1, \cdots, K\}$. We have that:

$$\max_{\xi_{it}} \sum_{t=1}^{T} \sum_{i \in C(k)} \sum_{j \in C(k): j \neq i} \frac{\|p_i - p_j\|^2}{N_k / T} \cdot \xi_{it} \cdot \xi_{jt} \tag{4a}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in C(k) \tag{4b}$$

$$\sum_{i \in C(k)} \xi_{it} = \frac{|C(k)|}{T}, \quad \forall t \in \{1, \dots, T\} \tag{4c}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in C(k)\ \forall t \in \{1, \dots, T\}. \tag{4d}$$

Each subproblem (4) can be linearized by introducing a binary variable for every pair of points and for every anticluster $t \in \{1, \cdots, T\}$.

Related literature

Problem (4) is related to:

Related literature

Problem (4) is related to:

1. the **Anticlustering problem** (Späth 1986; Papenberg and Klau 2021; Brusco, Cradit, and Steinley 2020; Papenberg 2024) in the psycology literature

Related literature

Problem (4) is related to:

1. the **Anticlustering problem** (Späth 1986; Papenberg and Klau 2021; Brusco, Cradit, and Steinley 2020; Papenberg 2024) in the psycology literature
2. **Maximally Diverse Grouping problem** (Lai and Hao 2016; Schulz 2023a; Schulz 2023b; Schulz 2021) in the OR community

SAPIENZA
UNIVERSITÀ DI ROMA

Related literature

Problem (4) is related to:

1. the **Anticlustering problem** (Späth 1986; Papenberg and Klau 2021; Brusco, Cradit, and Steinley 2020; Papenberg 2024) in the psycology literature
2. **Maximally Diverse Grouping problem** (Lai and Hao 2016; Schulz 2023a; Schulz 2023b; Schulz 2021) in the OR community
3. **clique partitioning problem** that can be formulated as a Maximally Diverse Grouping problem (Brimberg, Janićijević, Mladenović, and Urošević 2017)

SAPIENZA
UNIVERSITÀ DI ROMA

Related literature

Problem (4) is related to:

1. the **Anticlustering problem** (Späth 1986; Papenberg and Klau 2021; Brusco, Cradit, and Steinley 2020; Papenberg 2024) in the psycology literature
2. **Maximally Diverse Grouping problem** (Lai and Hao 2016; Schulz 2023a; Schulz 2023b; Schulz 2021) in the OR community
3. **clique partitioning problem** that can be formulated as a Maximally Diverse Grouping problem (Brimberg, Janićijević, Mladenović, and Urošević 2017)

SAPIENZA UNIVERSITÀ DI ROMA

Related literature

Problem (4) is related to:

1. the **Anticlustering problem** (Späth 1986; Papenberg and Klau 2021; Brusco, Cradit, and Steinley 2020; Papenberg 2024) in the psycology literature
2. **Maximally Diverse Grouping problem** (Lai and Hao 2016; Schulz 2023a; Schulz 2023b; Schulz 2021) in the OR community
3. **clique partitioning problem** that can be formulated as a Maximally Diverse Grouping problem (Brimberg, Janićijević, Mladenović, and Urošević 2017)

The idea is to partition elements into disjoint groups with the goal of obtaining **high between-group similarity** and **high within-group heterogeneity**.

SAPIENZA
UNIVERSITÀ DI ROMA

Anticlustering

One of the most common objective functions is the sum of the (squared) distances of the points in the anticluster (group) that is exactly our objective function

Anticlustering

One of the most common objective functions is the sum of the (squared) distances of the points in the anticluster (group) that is exactly our objective function

The problem is equivalent to the **m-equipartition problem** on a **complete graph**

Anticlustering

One of the most common objective functions is the sum of the (squared) distances of the points in the anticluster (group) that is exactly our objective function

The problem is equivalent to the **m-equipartition problem** on a **complete graph**

There are different formulations, but exact methods can be used only for very small size datasets, so heuristic approaches are proposed

Anticlustering

One of the most common objective functions is the sum of the (squared) distances of the points in the anticluster (group) that is exactly our objective function

The problem is equivalent to the **m-equipartition problem** on a **complete graph**

There are different formulations, but exact methods can be used only for very small size datasets, so heuristic approaches are proposed

Our idea

We define our heuristic for our version of the anticlustering problem

SAPIENZA
UNIVERSITÀ DI ROMA

Our evaluation algorithm

S1. Compute a heuristic solution by k-means with objective value $UB(\delta^*_{ij})$

Our evaluation algorithm

S1. Compute a heuristic solution by k-means with objective value $UB(\delta^*_{ij})$

S2. Compute a partition of the dataset $\{S_1, \ldots, S_T\}$ by heuristically solving problems (4)

Our evaluation algorithm

S1. Compute a heuristic solution by k-means with objective value $UB(\delta^*_{ij})$

S2. Compute a partition of the dataset $\{S_1, \ldots, S_T\}$ by heuristically solving problems (4)

S3. Compute the lower bound $(LB(\delta^*_{ij}))$

SAPIENZA UNIVERSITÀ DI ROMA

Our evaluation algorithm

S1. Compute a heuristic solution by k-means with objective value $UB(\delta^*_{ij})$

S2. Compute a partition of the dataset $\{S_1, \ldots, S_T\}$ by heuristically solving problems (4)

S3. Compute the lower bound $(LB(\delta^*_{ij}))$

S4. Return the optimality gap

$$\gamma_{\mathrm{LB}} = \frac{UB(\delta^*_{ij}) - LB(\delta^*_{ij})}{UB(\delta^*_{ij})}$$

Anticlustering heuristic

We aim to find a good feasible solution for the following problems (one for each $C(k)$ induced by the solution δ^*_{ij}, they can be solved in parallel)

$$\max_{\xi_{it}} \sum_{t=1}^{T} \sum_{i \in C(k)} \sum_{j \in C(k): j \neq i} \frac{\|p_i - p_j\|^2}{N_k / T} \cdot \xi_{it} \cdot \xi_{jt}$$

$$\text{s.t.} \sum_{t=1}^{T} \xi_{it} = 1, \quad \forall i \in C(k)$$

$$\sum_{i \in C(k)} \xi_{it} = \frac{|C(k)|}{T}, \quad \forall t \in \{1, \dots, T\}$$

$$\xi_{it} \in \{0, 1\}, \quad \forall i \in C(k)\ \forall t \in \{1, \dots, T\}. \tag{5a}$$

Random Generate a random balanced partition for each $C(k)$.

Mounting Solve a MILP for finding the optimal "mounting" of the generated anticlusters

Improve Try to improve the current partition by a swap procedure: we try to swap points exchanging points close to the centroid of the cluster in the anticluster with points far away from the center in the same cluster but in a different anticluster (larger contribution to the objective)

Swap evaluation

How to decide whether a swap improves my lower bound?

Swap evaluation

How to decide whether a swap improves my lower bound?

We cannot afford to compute the lower bound, neither we can trust that the projecton of δ_{ij}^* is optimal

Swap evaluation

How to decide whether a swap improves my lower bound?

We cannot afford to compute the lower bound, neither we can trust that the projecton of δ^*_{ij} is optimal

We rely on k-means and compute a proxy of the lower bound: we apply k-means with a smart initialization on each anticluster (related to the starting solution) and use the obtained value $\tilde{UB}_t$ to approximate the lower bound:

$$\tilde{LB} = \sum_{t=1}^{T} \tilde{UB}_t$$

SAPIENZA UNIVERSITÀ DI ROMA

Swap evaluation

How to decide whether a swap improves my lower bound?

We cannot afford to compute the lower bound, neither we can trust that the projecton of δ^*_{ij} is optimal

We rely on k-means and compute a proxy of the lower bound: we apply k-means with a smart initialization on each anticluster (related to the starting solution) and use the obtained value $\tilde{UB}_t$ to approximate the lower bound:

$$\tilde{LB} = \sum_{t=1}^{T} \tilde{UB}_t$$

If the bound improves, the swap is implemented and the procedure keeps going until no improvement is achieved (or a time limit is reached)

SAPIENZA
UNIVERSITÀ DI ROMA

Actual lower bound computation

Once we have the final anticlusters,with corresponding estimated gap $\gamma^+ = \frac{UB - LB^+}{UB}$, we compute the real lower bound by applying SOS-SDP on each anticluster (in parallel)

Actual lower bound computation

Once we have the final anticlusters,with corresponding estimated gap $\gamma^+ = \frac{UB - LB^+}{UB}$, we compute the real lower bound by applying SOS-SDP on each anticluster (in parallel)

We need to choose whether to run it only at the root node, or allowing for some branching to improve the lower bound, there is a trade off related also to the size of each subproblem (number of anticlusters)

Actual lower bound computation

Once we have the final anticlusters,with corresponding estimated gap $\gamma^{+} = \frac{UB - LB^{+}}{UB}$, we compute the real lower bound by applying SOS-SDP on each anticluster (in parallel)

We need to choose whether to run it only at the root node, or allowing for some branching to improve the lower bound, there is a trade off related also to the size of each subproblem (number of anticlusters)

We need to choose the number of anticlusters, the quality of the bound can oscillate

SAPIENZA UNIVERSITÀ DI ROMA

Experimental setup

We run SOS-SDP on each anticluster only at the root node, allowing the default cutting plane procedure for computing the bound:

1. As for the pair and triangle inequalities, we randomly separate at most 100000 valid cuts, we sort them in decreasing order with respect to the violation, and we select the first 10% of violated ones, yielding at most 10000 pairs and at most 10000 triangles added in each cutting-plane iteration.

SAPIENZA
UNIVERSITÀ DI ROMA

Experimental setup

We run SOS-SDP on each anticluster only at the root node, allowing the default cutting plane procedure for computing the bound:

1. As for the pair and triangle inequalities, we randomly separate at most 100000 valid cuts, we sort them in decreasing order with respect to the violation, and we select the first 10% of violated ones, yielding at most 10000 pairs and at most 10000 triangles added in each cutting-plane iteration.
2. The tolerance for checking the violation of the cuts is set to $\varepsilon_{\text{viol}} = 10^{-4}$, whereas the tolerance for identifying the active inequalities is set to $\varepsilon_{\text{act}} = 10^{-6}$.

Experimental setup

We run SOS-SDP on each anticluster only at the root node, allowing the default cutting plane procedure for computing the bound:

1. As for the pair and triangle inequalities, we randomly separate at most 100000 valid cuts, we sort them in decreasing order with respect to the violation, and we select the first 10% of violated ones, yielding at most 10000 pairs and at most 10000 triangles added in each cutting-plane iteration.
2. The tolerance for checking the violation of the cuts is set to $\varepsilon_{\text{viol}} = 10^{-4}$, whereas the tolerance for identifying the active inequalities is set to $\varepsilon_{\text{act}} = 10^{-6}$.
3. Finally, we set the accuracy tolerance of SDPNAL+ to 10^{-4}

SAPIENZA
UNIVERSITÀ DI ROMA

Experimental setup

We run SOS-SDP on each anticluster only at the root node, allowing the default cutting plane procedure for computing the bound:

1. As for the pair and triangle inequalities, we randomly separate at most 100000 valid cuts, we sort them in decreasing order with respect to the violation, and we select the first 10% of violated ones, yielding at most 10000 pairs and at most 10000 triangles added in each cutting-plane iteration.
2. The tolerance for checking the violation of the cuts is set to $\varepsilon_{\text{viol}} = 10^{-4}$, whereas the tolerance for identifying the active inequalities is set to $\varepsilon_{\text{act}} = 10^{-6}$.
3. Finally, we set the accuracy tolerance of SDPNAL+ to 10^{-4}
4. The lower bound is valid since we postprocess the output of the SDP solver (to be improved)

SAPIENZA
UNIVERSITÀ DI ROMA

Two toy examples

Table: Toy datasets

Dataset	N	D	K
pr1002	1002	2	4
Synthetic	900	2	9

Sapienza Università di Roma

Two toy examples

Table: Toy datasets

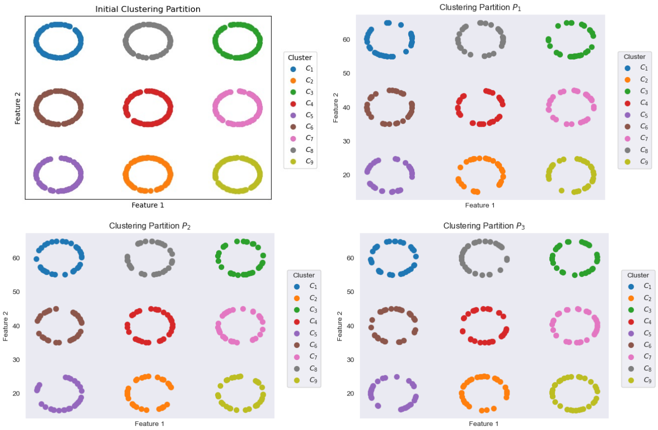
Dataset	N	D	K
`pr1002`	1002	2	4
`Synthetic`	900	2	9

The two instances are on the plane. We use them to visualize what happens in two very different cases: one where the clusters are well separated and one where they are not well separated

SAPIENZA
UNIVERSITÀ DI ROMA

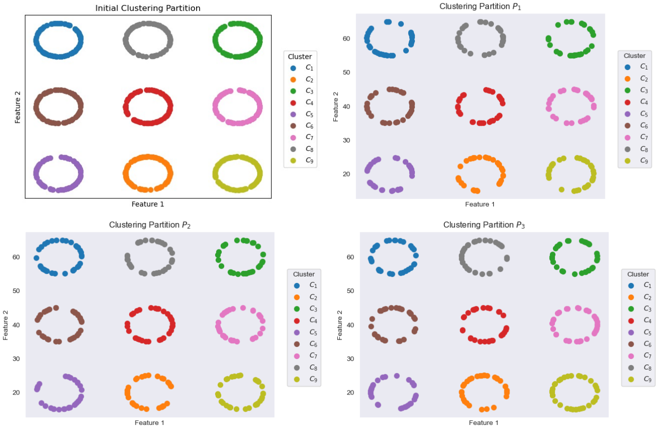
Synthetic- Random Initial Partition

900 points, 9 cluster each with 100 points, 3 anticlusters



Synthetic- Random Initial Partition

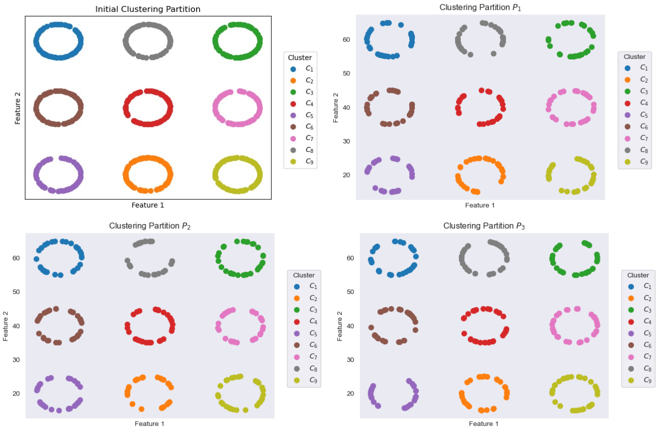
900 points, 9 cluster each with 100 points, 3 anticlusters



Gap (estimated with k-means) $= 1.65$ %

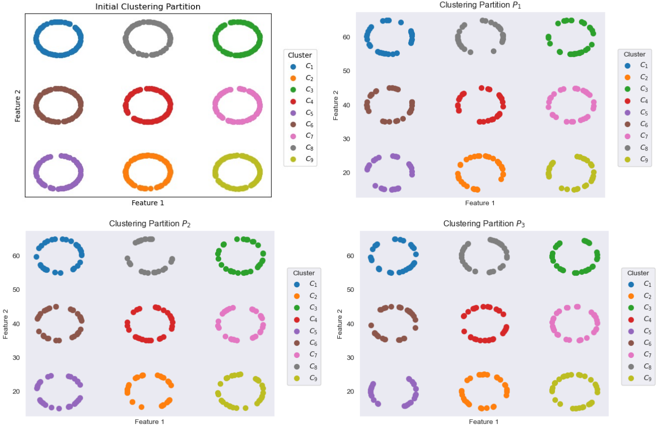
Synthetic- Final Partition

900 points, 9 cluster each with 100 points, 3 anticlusters



Synthetic- Final Partition

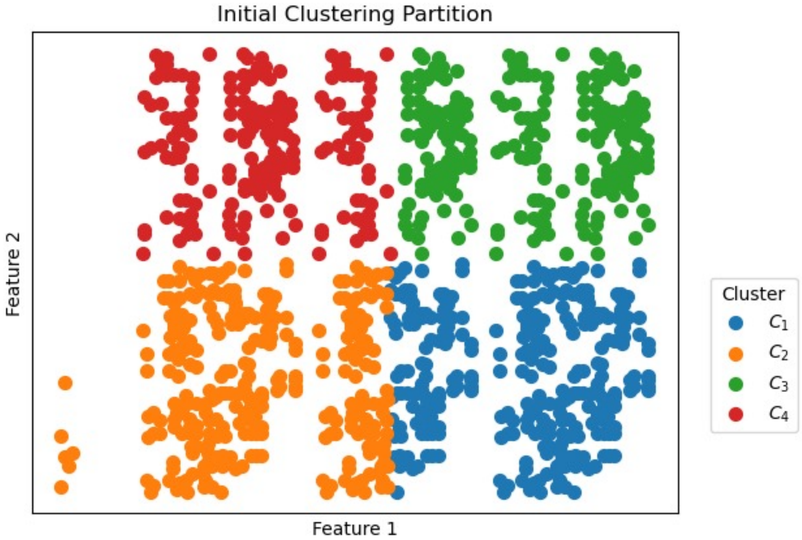
900 points, 9 cluster each with 100 points, 3 anticlusters



Gap (with respect to the lower bound computed by SOS-SDP) = 0.29 %

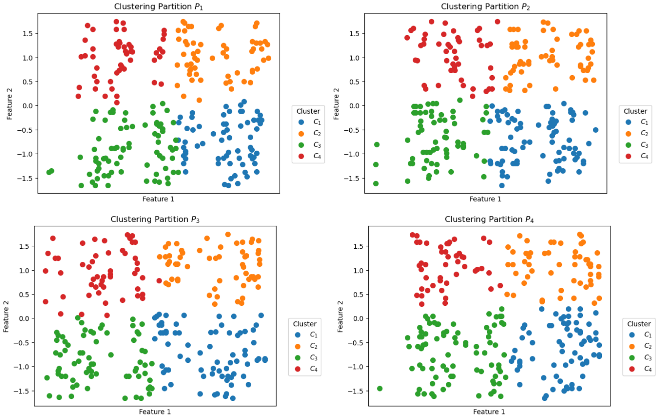
pr1002- Random Initial Partition

1002 points, 4 clusters, 4 anticlusters



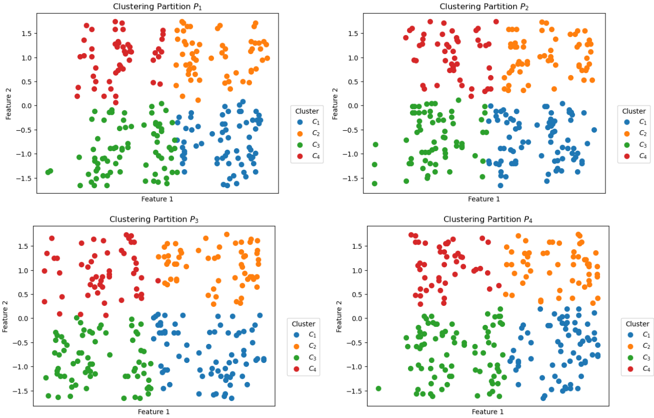
pr1002- Random Initial Partition

1002 points, 4 clusters, 4 anticlusters



pr1002- Random Initial Partition

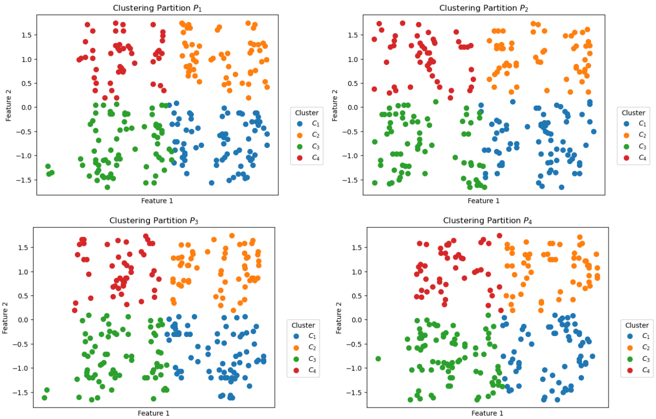
1002 points, 4 clusters, 4 anticlusters



Gap (estimated with k-means) =2.516 %

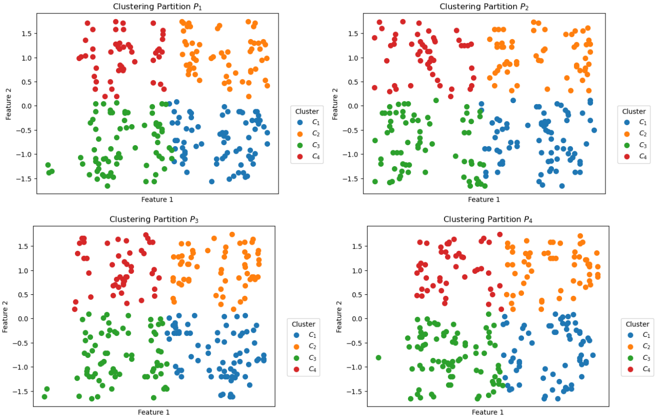
pr1002- Final Partition

1002 points, 4 clusters, 4 partitions



pr1002- Final Partition

1002 points, 4 clusters, 4 partitions



Gap (with respect to the lower bound computed by SOS-SDP) = 1.49%, with the UB = 0.06%

Artificial Instances

- We generate large-scale Gaussian datasets comprising $N = 10.000$ data points in a two-dimensional space ($D = 2$).

SAPIENZA
UNIVERSITÀ DI ROMA

Artificial Instances

- We generate large-scale Gaussian datasets comprising $N = 10.000$ data points in a two-dimensional space ($D = 2$).
- These datasets vary in the number of clusters ($K \in \{2, 3.4\}$) and noise levels.

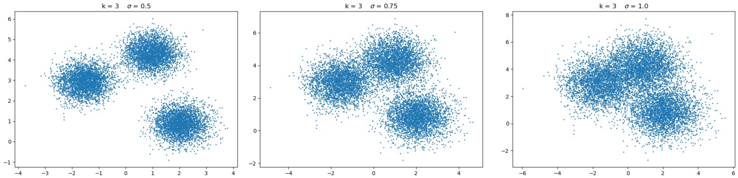
Artificial Instances

- We generate large-scale Gaussian datasets comprising $N = 10.000$ data points in a two-dimensional space ($D = 2$).
- These datasets vary in the number of clusters ($K \in \{2, 3.4\}$) and noise levels.
- Data points are sampled from a mixture of K Gaussian distributions $\mathcal{N}(\mu_j, \Sigma_j)$ for $j \in \{1, \ldots, K\}$, with equal mixing proportions, where each distribution has a mean μ_j and a shared spherical covariance matrix $\Sigma_j = \sigma I$.

SAPIENZA UNIVERSITÀ DI ROMA

Artificial Instances

- We generate large-scale Gaussian datasets comprising $N = 10.000$ data points in a two-dimensional space ($D = 2$).
- These datasets vary in the number of clusters ($K \in \{2, 3.4\}$) and noise levels.
- Data points are sampled from a mixture of K Gaussian distributions $\mathcal{N}(\mu_j, \Sigma_j)$ for $j \in \{1, \ldots, K\}$, with equal mixing proportions, where each distribution has a mean μ_j and a shared spherical covariance matrix $\Sigma_j = \sigma I$.
- The standard deviation σ varies among $\{0.50, 0.75, 1.00\}$, representing different noise levels. Cluster centers μ_j are drawn from a uniform distribution within the interval $[-10.10]$.



Results on artificial datasets : 2 clusters

Noise	T	GAP(LB)	$\tilde{GAP}$(UB)	Time(min)
0.5	10	0.18%	0.18%	43.27
0.5	12	0.29%	0.29%	61,9
0.5	15	0.36%	0.36%	37.58
0.5	17	0.33%	0.33%	38,92
0.5	20	0.4%	0.4%	37.05
0.75	10	0.28%	0.26%	127.68
0.75	12	0.25%	0.22%	102.4
0.75	15	0.32%	0.3%	73.18
0.75	17	0.38%	0.36%	92.07
0.75	20	0.4%	0.38%	65.47
1	10	0.47%	0.26%	198,25
1	12	0.41%	0.23%	130.32
1	15	0.41%	0.32%	146.32
1	17	0.75%	0.65%	126.68
1	20	0.56%	0.5%	105.87

Results on artificial datasets : 3 clusters

Noise	T	GAP(LB)	$\tilde{GAP}$(UB)	Time(min)
0.5	10	0.31%	0.3%	95.62
0.5	12	0.38%	0.38%	88.85
0.5	15	0.58%	0.58%	67.47
0.5	17	0.53%	0.53%	50.8
0.5	20	0.62%	0.61%	57.3
0.75	10	0.43%	0.37%	138.12
0.75	12	0.56%	0.48%	178.25
0.75	15	0.61%	0.56%	94.38
0.75	17	0.65%	0.62%	122.68
0.75	20	0.8%	0.77%	63.98
1	10	1.43%	0.43%	184.53
1	12	1.44%	0.52%	217.72
1	15	1.24%	0.58%	89.18
1	17	0.93%	0.4%	115.2
1	20	1.29%	0.78%	114.52

Results on artificial datasets : 4 clusters

Noise	T	GAP(LB)	$\tilde{GAP}$(UB)	Time(min)
0.5	10	0.51%	0.51%	110.57
0.5	12	0.68%	0.68%	75.18
0.5	15	0.83%	0.83%	70.7
0.5	17	0.99%	0.98%	53.35
0.5	20	0.87%	0.87%	60.83
0.75	10	0.54%	0.47%	155.62
0.75	12	0.58%	0.53%	100.4
0.75	15	0.61%	0.58%	104.32
0.75	17	0.83%	0.79%	76.22
0.75	20	0.91%	0.88%	72.5
1	10	1.58%	0.73%	201.72
1	12	1.25%	0.57%	120.13
1	15	1.31%	0.86%	105.27
1	17	1.24%	0.92%	121.32
1	20	1.16%	0.87%	83.85

Datasets

We select some large scale datasets that cannot be solved directly by SOS-SDP:

Dataset	N	D	K	$\lvert C_1\rvert \dots \lvert C_K\rvert$			
Abalone	4,177	10	3	1,308	1,341	1,528	
Facebook	7,050	13	3	218	2,558	4,274	
Frogs	7,195	22	4	605	670	2,367	3,553
Electric	10,000	12	3	2,886	3,537	3,577	
Pulsar	17,898	8	2	2,057	15,841		

Table: Characteristics of real-world datasets.

- Each dataset has been tested for 5 different values of the number of anticlusters T, depending on the number of data points N and on the size of the clusters of the initial solution.
- The choice of T is influenced by two key requirements: (i) the size of each anticluster must be tractable, i.e., less than 1,000 data points; (ii) each cluster must be adequately represented in each anticluster
- The smallest instance Abalone was solved exactly in 2.6 hours
- Solving an instance of around 1,000 data points to global optimality requires several hours of computational time

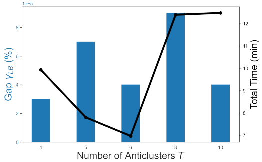
SAPIENZA
UNIVERSITÀ DI ROMA

Results on real world datasets

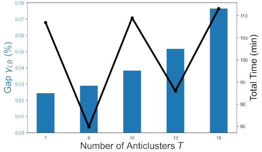
Inst (K)	T	γ_{LB} (%)	γ_{UB} (%)	γ^{+} (%)	MILP (s)	Heur (s)	SOS (s)	Time (min)
Ab (3)	4	**0.003**	0.001	0.001	0	172	424	10
	5	0.007	0.001	0.001	0	154	314	8
	6	0.004	0.001	0.001	0	205	213	7
	8	0.009	0.001	0.001	0	546	198	12
	10	0.004	0.001	0.002	0	591	158	12
El (3)	10	2.880	0.460	0.001	2	1,384	5,467	114
	15	**2.198**	0.757	0.001	4	1,759	6,417	136
	20	2.329	0.944	0.001	9	5,118	3,915	151
	25	2.482	1.270	0.002	21	5,062	3,218	138
	30	2.837	1.393	0.003	45	6,856	2,248	152
FB (3)	7	**2.428**	0.321	0.014	0	1,694	4,813	108
	8	2.881	0.923	0.029	1	1,937	3,155	85
	10	3.820	2.107	0.034	1	2,439	4,130	110
	13	5.157	3.306	0.093	1	3,155	2,423	93
	18	7.639	6.373	0.285	5	4,343	2,349	112
Frogs (4)	8	5.147	2.008	1.824	1	2,032	5,558	127
	10	4.824	2.252	1.807	1	2,443	2,639	85
	13	**4.121**	1.881	1.795	4	3,202	2,217	90
	15	4.339	2.397	1.788	9	3,714	1,885	93
	16	4.131	2.323	1.780	10	3,849	1,758	94
Pulsar (2)	18	2.625	0.165	0.001	7	4,059	19,012	385
	20	2.727	0.206	0.002	7	4,884	19,502	407
	25	2.562	0.020	0.002	7	6,031	11,727	296
	30	2.390	0.159	0.002	8	7,275	10,435	295
	35	**2.274**	0.524	0.003	7	8,523	7,873	273

SAPIENZA UNIVERSITÀ DI ROMA

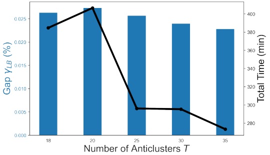
Results



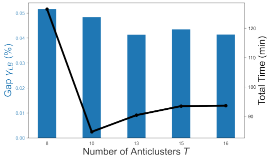
(a) Abalone



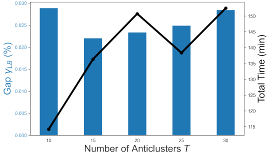
(b) Facebook



(c) Pulsar



(d) Frogs



(e) Electric

Figure: Performance comparison for different numbers of anticlusters. The bar chart represents the lower bound gap (γ_{LB}), while the black line with markers indicates the total computation time in minutes (Time (min)).

Conclusions and future work

1. The gap is in general pretty good, and would be amazing if we could certify the optimality of the solutions of the single anticlusters

Conclusions and future work

1. The gap is in general pretty good, and would be amazing if we could certify the optimality of the solutions of the single anticlusters
2. When the number of anticlusters changes the lower bound can fluctuate, on some instances it tends to increase when the number of partitions increases..should we optimize more? The answer seems to be NO

Conclusions and future work

1. The gap is in general pretty good, and would be amazing if we could certify the optimality of the solutions of the single anticlusters
2. When the number of anticlusters changes the lower bound can fluctuate, on some instances it tends to increase when the number of partitions increases..should we optimize more? The answer seems to be NO
3. The good news is that now we can tackle much larger instances, proving gaps lower than 5%

SAPIENZA UNIVERSITÀ DI ROMA

Conclusions and future work

1. The gap is in general pretty good, and would be amazing if we could certify the optimality of the solutions of the single anticlusters
2. When the number of anticlusters changes the lower bound can fluctuate, on some instances it tends to increase when the number of partitions increases..should we optimize more? The answer seems to be NO
3. The good news is that now we can tackle much larger instances, proving gaps lower than 5%
4. Can we do something exact for the anticlustering problem? Work in progress...

Conclusions and future work

1. The gap is in general pretty good, and would be amazing if we could certify the optimality of the solutions of the single anticlusters
2. When the number of anticlusters changes the lower bound can fluctuate, on some instances it tends to increase when the number of partitions increases..should we optimize more? The answer seems to be NO
3. The good news is that now we can tackle much larger instances, proving gaps lower than 5%
4. Can we do something exact for the anticlustering problem? Work in progress...

SAPIENZA
UNIVERSITÀ DI ROMA

Aloise, Daniel, Amit Deshpande, Pierre Hansen, and Preyas Popat (2009). "NP-hardness of Euclidean sum-of-squares clustering". In: *Machine learning* 75, pp. 245–248.

Aloise, Daniel and Pierre Hansen (2009). "A branch-and-cut SDP-based algorithm for minimum sum-of-squares clustering". In: *Pesquisa Operacional* 29.3, pp. 503–516.

Aloise, Daniel, Pierre Hansen, and Leo Liberti (2012a). "An improved column generation algorithm for minimum sum-of-squares clustering". In: *Mathematical Programming* 131.1, pp. 195–220.

— (2012b). "An improved column generation algorithm for minimum sum-of-squares clustering". In: 131.1, pp. 195–220.

Arthur, David and Sergei Vassilvitskii (2006). *k-means++: The advantages of careful seeding*. Tech. rep. Stanford.

Bagirov, Adil M., Sona Taheri, and Julien Ugon (2016). "Nonsmooth DC programming approach to the minimum sum-of-squares clustering problems". In: *Pattern Recognition* 53, pp. 12–24. issn: 0031-3203. doi: https://doi.org/10.1016/j.patcog.2015.11.011.

Baumann, Philipp (2020). "A binary linear programming-based k-means algorithm for clustering with must-link and cannot-link constraints". In: *2020 IEEE international conference on industrial engineering and engineering management (IEEM)*. IEEE, pp. 324–328.

Brimberg, Jack, Stefana Janićijević, Nenad Mladenović, and Dragan Urošević (2017). "Solving the clique partitioning problem as a maximally diverse grouping problem". In: *Optimization Letters* 11, pp. 1123–1135.

Brusco, Michael J (2006). "A repetitive branch-and-bound procedure for minimum within-cluster sums of squares partitioning". In: *Psychometrika* 71.2, pp. 347–363.

Brusco, Michael J, J Dennis Cradit, and Douglas Steinley (2020). "Combining diversity and dispersion criteria for anticlustering: A bicriterion approach". In: *British Journal of Mathematical and Statistical Psychology* 73.3, pp. 375–396.

Burgard, Jan Pablo, Carina Moreira Costa, Christopher Hojny, Thomas Kleinert, and Martin Schmidt (2023). "Mixed-integer programming techniques for the minimum sum-of-squares clustering problem". In: *Journal of Global Optimization*, pp. 1–57.

Diehr, George (1985). "Evaluation of a branch and bound algorithm for clustering". In: *SIAM Journal on Scientific and Statistical Computing* 6.2, pp. 268–284.

Du Merle, Olivier, Pierre Hansen, Brigitte Jaumard, and Nenad Mladenovic (1999). "An interior point algorithm for minimum sum-of-squares clustering". In: *SIAM Journal on Scientific Computing* 21.4, pp. 1485–1505.

Franti, Pasi and Sami Sieranoja (2019). "How much can k-means be improved by using better initialization and repeats?" In: *Pattern Recognition* 93, pp. 95–112.

Hansen, Pierre and Nenad Mladenovic (2001). "J-Means: a new local search heuristic for minimum sum of squares clustering". In: *Pattern Recognition* 34.2, pp. 405–413. issn: 0031-3203. doi: https://doi.org/10.1016/S0031-3203(99)00216-2.

Karmitsa, Napsu, Adil M. Bagirov, and Sona Taheri (1997). "A clustering algorithm using an evolutionary programming-based approach". In: *Pattern Recognition Letters* 18.10, pp. 975–986. issn: 0167-8655. doi: https://doi.org/10.1016/S0167-8655(97)00122-0.

— (2017). "New diagonal bundle method for clustering problems in large data sets". In: *European Journal of Operational Research* 263.2, pp. 367–379. issn: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2017.06.010.

— (2018). "Clustering in large data sets with the limited memory bundle method". In: *Pattern Recognition* 83, pp. 245–259. issn: 0031-3203. doi: https://doi.org/10.1016/j.patcog.2018.05.028.

Koontz, Warren L. G., Patrenahalli M. Narendra, and Keinosuke Fukunaga (1975). "A branch and bound clustering algorithm". In: *IEEE Transactions on Computers* 100.9, pp. 908–915.

Lai, Xiangjing and Jin-Kao Hao (2016). "Iterated maxima search for the maximally diverse grouping problem". In: *European Journal of Operational Research* 254.3, pp. 780–800. issn: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2016.05.018. url: https://www.sciencedirect.com/science/article/pii/S0377221716303381.

Lee, Julian and David Perkins (2021). "A simulated annealing algorithm with a dual perturbation method for clustering". In: *Pattern Recognition* 112, p. 107713.

Likas, Aristidis, Nikos Vlassis, and Jakob J Verbeek (2003). "The global k-means clustering algorithm". In: *Pattern recognition* 36.2, pp. 451–461.

Lloyd, Stuart (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66)*. Univ. California Press, Berkeley, Calif., Vol. I: Statistics, pp. 281–297.

Mansueto, Pierluigi and Fabio Schoen (2021). "Memetic differential evolution methods for clustering problems". In: *Pattern Recognition* 114, p. 107849.

Maulik, Ujjwal and Sanghamitra Bandyopadhyay (2000). "Genetic algorithm-based clustering technique". In: *Pattern Recognition* 33.9, pp. 1455–1465. issn: 0031-3203. doi: `https://doi.org/10.1016/S0031-3203(99)00137-5`.

Orlov, V I, L A Kazakovtsev, I P Rozhnov, N A Popov, and V V Fedosov (2018). "Variable neighborhood search algorithm for k-means clustering". In: *IOP Conference Series: Materials Science and Engineering* 450, p. 022035. doi: `10.1088/1757-899x/450/2/022035`. url: `https://doi.org/10.1088/1757-899x/450/2/022035`.

Papenberg, Martin (2024). “K-Plus anticlustering: An improved k-means criterion for maximizing between-group similarity”. In: *British Journal of Mathematical and Statistical Psychology* 77.1, pp. 80–102.

Papenberg, Martin and Gunnar W Klau (2021). “Using anticlustering to partition data sets into equivalent parts.”. In: *Psychological Methods* 26.2, p. 161.

Peng, Jiming and Yu Wei (2007). “Approximating k-means-type clustering via semidefinite programming”. In: *SIAM Journal on Optimization* 18.1, pp. 186–205.

Peng, Jiming and Yu Xia (2005). “A new theoretical framework for k-means-type clustering”. In: *Foundations and Advances in Data Mining*. Springer, pp. 79–96.

Piccialli, Veronica, Antonio M Sudoso, and Angelika Wiegele (2022). “SOS-SDP: an exact solver for minimum sum-of-squares clustering”. In: *INFORMS Journal on Computing* 34.4, pp. 2144–2162.

Schulz, Arne (2021). “The balanced maximally diverse grouping problem with block constraints”. In: *European Journal of Operational Research* 294.1, pp. 42–53. issn: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2021.01.029. url: https://www.sciencedirect.com/science/article/pii/S037722172100059X.

— (2023a). “The balanced maximally diverse grouping problem with attribute values”. In: *Discrete Applied Mathematics* 335, pp. 82–103.

— (2023b). “The balanced maximally diverse grouping problem with integer attribute values”. In: *Journal of Combinatorial Optimization* 45.5, p. 135.

Sherali, Hanif D and Jitamitra Desai (2005). “A global optimization RLT-based approach for solving the hard clustering problem”. In: *Journal of Global Optimization* 32.2, pp. 281–306.

Späth, H (1986). “Anticlustering: Maximizing the variance criterion”. In: *Control and Cybernetics* 15.2, pp. 213–218.

Sudoso, Antonio M and Daniel Aloise (2024). “A column generation algorithm with dynamic constraint aggregation for minimum sum-of-squares clustering”. In: *arXiv preprint arXiv:2410.06187*.

Al-Sultan, Khaled S. (1995). “A Tabu search approach to the clustering problem”. In: *Pattern Recognition* 28.9, pp. 1443–1451. issn: 0031-3203. doi: https://doi.org/10.1016/0031-3203(95)00022-R.

Tao, Pham Dinh et al. (2014). “New and efficient DCA based algorithms for minimum sum-of-squares clustering”. In: *Pattern Recognition* 47.1, pp. 388–401.

Yu, Shyr-Shen, Shao-Wei Chu, Chuin-Mu Wang, Yung-Kuan Chan, and Ting-Cheng Chang (2018). “Two improved k-means algorithms”. In: *Applied Soft Computing* 68, pp. 747–755. issn: 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.08.032.