Central European Journal of Operations Research
https://doi.org/10.1007/s10100-023-00887-9

ORIGINAL PAPER

A novel auto-pruned ensemble clustering via SOCP

Duygu Üçüncü[1,2] · **Süreyya Akyüz**[1] · **Erdal Gül**[2]

Accepted: 26 September 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Operations Research (OR) plays a crucial role in strategic decision-making in today's business world; it uses complex algorithms and data analytic to provide decision-makers with the necessary information. The proposed study presents a novel ensemble clustering for decision making in various disciplines including OR problems which introduce a second-order conic optimization model. This method provides a significant advantage over the traditional difference of convex programming by continuously and convexly solving integer programming. The model optimizes the balance between accuracy and diversity, resulting in the selection of the best candidates for prediction. The study's remarkable contribution lies in the automatic sub-ensemble selection while optimizing for accuracy and diversity. The model has been verified using real data and achieves competitive prediction performance. Furthermore, this approach illustrates how OR can be utilized to enhance ensemble clustering and decision-making.

Keywords Strategic decision support · Ensemble clustering · Auto-pruning · Machine learning · Second order conic programming

✉ Duygu Üçüncü
duygu.ucuncu@bau.edu.tr

Süreyya Akyüz
sureyya.akyuz@bau.edu.tr

Erdal Gül
gul@yildiz.edu.tr

[1] Department of Mathematics, Faculty of Engineering and Natural Sciences, Bahçeşehir University, Beşiktaş, İstanbul 34353, Türkiye

[2] Department of Mathematics, Faculty of Arts and Sciences, Yildiz Technical University, Esenler, İstanbul 34220, Türkiye

Published online: 29 October 2023

1 Introduction

Operations research (OR) is a field of mathematics that employs mathematical models and optimization techniques to solve complex problems and make informed decisions (Roy et al. 2017; Perišić and Pahor 2023). Computational tools for strategic decision support have grown in importance in the business sector in the age of big data and complexity. These tools leverage advanced algorithms and data analytics techniques to give decision-makers the insights they need to make strategic decisions (Gupta et al. 2007). These tools can be used to construct prediction models by identifying patterns, trends, and linkages in large data sets while also providing interactive visualization and simulation capabilities that allow decision-makers to experience different situations and understand the potential consequences of their decisions (Garcia-Vicuña et al. 2022; Gütmen et al. 2023; Ewertowski et al. 2023). Studies in the literature demonstrated the growing importance of ensemble learning techniques in computational tools for strategic decision support (Polikar 2006; Delen 2010). Ensemble methods improve the accuracy and robustness of decision support systems, making them more effective tools for strategic decision-making, and are widely regarded as the cutting-edge solution to many machine learning problems. Ensemble clustering is a popular unsupervised learning technique that has received a lot of attention (Fern and Lin 2008; Singh et al. 2010; Huang et al. 2016; Alqurashi and Wang 2019; Du and Wang 2022). This technique aims to combine the results of several basic clustering algorithms to produce better clustering. In recent years, a variety of techniques have been used to develop this point of view, making it not only possible but also developed. The goal of ensemble clustering is to enhance final clusters by combining the partitioning data from all individual clustering algorithms. Ensemble clustering can reduce the drawbacks of individual clustering algorithms and make them more suitable for larger datasets.

Traditional ensemble clustering methods aim to use all clustering solutions to reach a joint decision. However, recent researches have shown that diverse and accurate clustering solutions within the ensemble improve the performance of ensemble learning. Therefore, ensemble selection procedures are developed in the literature (Azimi and Fern 2006; Fern and Lin 2008; Jia et al. 2011; Yu et al. 2014) and have been instrumental in selecting the most accurate and diverse subset from the ensemble library. The two key factors which are accuracy and diversity need to be considered when choosing the ensemble. A concession of accuracy (diversity) is made, while increasing diversity (accuracy) leads to a trade-off between the accuracy and diversity of clustering solutions. In addition, the optimal number of clusters in the clustering ensemble influences the decision of the clustering ensemble. Eliminating unnecessary cluster solutions from the ensemble to find the best ensemble clustering candidates has emerged as a prominent issue in recent studies (Azimi and Fern 2006; Fern and Lin 2008; Yu et al. 2014, 2016). In this study, we propose a better solution to pick the best ensemble for clustering problems referred to ensemble pruning with a novel SOCP model while optimizing accuracy and diversity trade off simultaneously.

Ensemble pruning has been reported to reduce the size of the ensembles while maintaining accuracy. Ensemble pruning methods are also referred to as ensemble selection methods to reduce the complexity of the ensemble. Over the years, numerous ensemble pruning methods have been developed. Ranking and search-based methods are the two veriest popular approaches for selecting an ensemble subset. Ranking methods (Dai 2013; Partalas et al. 2010) order the individual members based on a specified measure and elect the top-ranked base models based on a threshold. Search-based methods (Jiang et al. 2017; Zhang et al. 2006) execute a heuristic search in the classifier's space and assess the joint performance of the classifier's subsets.

Most of the studies are focused on increasing the ensemble accuracy disregarding the diversity or vice a versa (Singh et al. 2010; Zhong et al. 2019). There are few ones considering both with ordered search based methods without using an automatic selection by an optimization model (Fern and Lin 2008). Recently, optimization based search methods have been studied for clustering problems (Özöğür-Akyüz et al. 2020). In these studies, the best subset selection is searched through transforming max-cut optimization model which results with solving non-convex optimization via transforming the non-convex model by difference of convex functions. Here, we propose a new second order conic model which transforms max-cut problem into a a convex model so that the need of approximating a non-convex problem is disappeared.

In optimization-based pruning, the ensemble pruning problem is identified as an optimization problem and the goal is to solve the optimization problem in order to discover the best sub-ensemble that maximizes accuracy and diversity. Using a semi-definite programming (SDP) technique, Zhang et al. (2006) defined the ensemble pruning problem as a quadratic integer programming problem that tries to discover a subset with the optimal accuracy-diversity trade-off. Moreover, in the study (Özöğür-Akyüz et al. 2020), Akyüz et al. created an ensemble selection model using a constrained optimization model where the objective function, which is modeled as the difference of the convex function, minimizes the total error and maximizes diversity by assessing the pairwise errors of the candidate models with a predefined matrix, and they used disciplined convex-concave programming (DCCP) to solve the model. In this study, the max-cut problem is modeled as SOCP so that integer programming is solved continuously and convexly, which is an advantage to the difference of convex programming. Therefore, the second-order conic optimization model is proposed in this paper for clustering to simultaneously optimize the trade-off between accuracy and diversity such that the best candidates of the ensemble are selected for the prediction phase. The proposed model was tested using real-world data, and prediction accuracies were compared with ensemble clustering methods in the literature. Therefore, the main contribution and motivation of this study are as follows:

- Optimizing the trade-off between accuracy and diversity simultaneously while obtaining the sub-ensemble automatically,
- Cluster selection through max-cut problem using SOCP without the need of non-convex transformation.

1.1 Survey on related work

Multiple classifier systems, also known as ensemble systems, have gained popularity in the computational intelligence and machine learning communities over the previous few decades (Hayden 1998; Polikar 2006). This focus is well-deserved, as ensemble systems have proven to be incredibly effective and versatile in a wide range of problem domains and real-world applications. Operations research is used to solve various problems, such as scheduling and time management, urban planning, supply chain management, optimization and engineering (Torjai and Kruzslicz 2016; Dawid et al. 2017; Csehi and Farkas 2017; Gür et al. 2022; Esteso et al. 2022; Heine and Thraves 2023), which uses mathematical models and optimization techniques to make informed decisions and can also benefit from ensemble systems. Ensemble learning predicts with a group of learners and aggregates their results. It has been empirically demonstrated that it is more dependable due to collaboration with several learners. The benefit of ensembles over single models has been reported in terms of greater robustness and accuracy (García-Pedrajas et al. 2005). Using ensemble learning methods, the grouping of independent classifiers yields more accurate results. Bagging, which is short for Bootstrap Aggregation developed by Breiman in 1996, and boosting, a classification approach, yet it may also be used for regression to improve the performance of a learning system, are the most well-known approaches in the literature.

Ensemble pruning attempts to pick a subset of individual learners (sub-ensemble) to compose the ensemble rather than merging all of the trained individual learners (classifiers) in an ensemble (Zhou 2012). In addition, ensemble pruning methods are used to reduce the complexity of an ensemble. Machine learning and data mining make extensive use of ensemble pruning algorithms. However, existing techniques generate ensembles that are often too big, resulting in high memory costs, computational expenditures, and occasional declines in efficacy (Xu et al. 2012; Zhou 2012). There-withal, ensemble pruning can be thought of as looking for the best solution in the space of the ensemble's subsets. However, identifying the best sub-ensemble using a comprehensive search with cross-validation takes time and effort. Therefore, several approximate searching strategies, such as ordered aggregation (Martínez-Muñoz and Suárez 2004; Martinez-Munoz et al. 2009; Lu et al. 2010), optimization based models via genetic algorithms (Zhou et al. 2002), SDP (Zhang et al. 2006), quadratic programming (Li and Zhou 2009) and linear programming (Zhang and Zhou 2011), have been proposed to locate near-optimal sub-ensembles. According to Tsoumakas et al. (2009), the ensemble pruning approach can be broadly classified into four types: ranking-based, clustering-based, optimization-based, and others.

Ranking-based approaches attempt to order the classifiers in the ensemble once according to some evaluation metric and select the top-ordered classifiers for the final ensemble. In numerous studies, many ordering-based ensemble pruning algorithms, such as kappa pruning, orientation pruning, and unsupervised margin-based ordering ensemble pruning, have been presented.

Clustering-based pruning methods attempt to identify several representative individual classifiers to construct an ultimate ensemble. These approaches, in general, work in two stages. In the first stage, a clustering technique is used to aggregate

individual classifiers with low prediction diversities into a number of clusters based on their predictions. One of the popular clustering algorithm is k-means (Forgy 1965) which is a centroid-based clustering. Sub-ensembles from the clusters are joined to create a final ensemble in the second stage (Lazarevic and Obradovic 2001; Bakker and Heskes 2003).

Optimization-based methods portray ensemble pruning as an optimization problem to discover the subset of the original ensemble that optimizes the generalization performance (e.g., accuracy on a separate validation set). Various optimization strategies have been developed, such as heuristic optimization, mathematical programming, and probabilistic methods. Ensemble pruning can be formulated as a mathematical optimization problem using mathematical programming approaches. For instance, the study (Zhang et al. 2006), considered ensemble pruning to be a quadratic integer programming problem and used SDP to get an approximate solution, whereas in the study (as Li and Zhou 2009), the ensemble pruning are regularized as a quadratic programming (QP) problem. To solve the pruning model, studies in (Özöğür-Akyüz et al. 2020; Ali et al. 2020) used disciplined convex-concave programming (DCCP) and difference of a convex algorithm (DCA), respectively. We surveyed the literature review in Table 1 and discovered a gap for clustering-based models (Fern and Lin 2008) since they consider only accuracy or diversity. Although the studies (Özöğür-Akyüz et al. 2020; Ali et al. 2020) consider both accuracy and diversity simultaneously, they use the difference of convex functions to convert the objective to a convex function. However, because we modeled the problem as SOCP in this study, that requirement has vanished.

1.2 Problem description

In this research, by taking into account pairwise errors of the candidate models with a predetermined matrix, we constructed an ensemble selection model utilizing a constrained optimization model, where the objective function minimizes the overall error and maximizes diversity similar to the study (Zhang et al. 2006) in which for the **classification problem**. We updated the accuracy-diversity matrix defined by classification problem using equations for Normalized Mutual Information (*NMI*)

Table 1 Survey of Related Works

References	Optimization Type	Type of Search
Jia et al. (2011)	–	Ranking based
Yu et al. (2014)	–	Ranking based
Fern and Lin (2008)	–	Clustering based
Fern and Lin (2008)	–	Clustering based
Zhang et al. (2006)	SDP	Optimization based
Özöğür-Akyüz et al. (2020)	DCCP	Optimization based
Zhong et al. (2019)	–	Clustering based
Ali et al. (2020)	DCA	Optimization based
This study (*ESC_SOCP*)	SOCP	Optimization based

and the Sum of Normalized Mutual Information (*SNMI*) formulations for **clustering problem**. Consequently, the predetermined matrix is defined as the non-diagonal elements of the matrix measure the diversity between two solutions while the diagonal elements measure the quality of the i-th solution.

To describe the mathematical model, let us consider the matrix D as the predetermined matrix

$$D_{ij} = \begin{cases} D_{ii} = SNMI(C, C_i) \ (i = 1, \ldots, n), \\ D_{ij} = 1 - NMI(C_i, C_j) \ (j = 1, \ldots, n), \end{cases} \tag{1}$$

that leads to the proposed model (2), which will soon be explained and will be referred to as *ESC_SOCP*, as follows:

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H}^+ v + \sum_{j=l+1}^{n+1} \lambda_j z_j \leq \theta, \\ & v^T G v = 4k + z_j, \ j = 1, \ldots, n+1, \\ & z_j \leq \sqrt{n}, \ j = l+1, \ldots, n+1. \end{aligned} \tag{2}$$

The matrix D is normalized in the following manner to scale up all elements to the same level as $\widetilde{D}_{ij} = \dfrac{D_{ij}}{N}$, where N represents the number of clustering solutions in the ensemble and $\widetilde{D}_{ij,i\neq j} = \dfrac{1}{2}\left(\dfrac{D_{ij}}{D_{ii}} + \dfrac{D_{ij}}{D_{jj}}\right)$. Finally, the matrices $\widetilde{H}$ and G in the equation 2 are derivations from the matrix $\widetilde{D}$, which can be seen in the Appendix A. Furthermore, the physical meaning of $\widetilde{D}$ corresponds to the accuracy-diversity trade off matrix as shown in the Appendix A.

The remaining of this article is organized as follows: Sect. 2 introduces the description of symbols used in this article given in Table 2 and compared methods for ensemble cluster selection. Section 3 introduces detailed information of an ensemble selection problem and our formulation of the objective function as a SOCP model. In Sect. 4, we presented our implementations on different datasets from *UCI Machine Learning Repository* (Dua and Graff 2017). A summary of the experiments and results are given in Sect. 4. Finally, discussion is given in Sect. 5.

2 Ensemble pruning methods

2.1 Joint criterion method

Below, we have the *Normalized Mutual Information* (*NMI*) function's definition (Cover and Thomas 1991) which measures the similarities between each clustering solution by

Table 2 Description of symbols in this article

Symbol	Description	Symbol	Description
$\mathbb{R}$	Real numbers	N	Number of clustering solutions in the ensemble
$\|.\|$	Euclidean Norm	N_v	Number of vertices
$\mathcal{K}$	Second-order cone	$w_{i,j}$	Vertex weight between i and j
$\succeq$	Matrix inequality	k	Ensemble size
X, Y	Clusters	D	A matrix
$p(x)$, $p(y)$	Marginal distribution functions	x	A vector
$H(x)$, $H(y)$	Mutual information function between clusters X and Y	y	A vector
$NMI(X, Y)$	Normalized mutual information of clusters X and Y	v	A vector
$SNMI$	Sum of normalized mutual information	H	A matrix
C	A clustering solution	$\widetilde{H}D$	A matrix
L	Large library of an ensemble	G	A matrix
α	A parameter	I	The Identity matrix
E	An ensemble	e	The column vector
CM	Coassociation matrix	θ	A parameter
MM	Newly defined internal validty index	λ_j	Eigenvalues
D	A matrix	q_j	Eigenvectors
$\widetilde{D}$	Normalized matrix D	δ	Threshold

$$I(X;Y) = \sum_{y\epsilon Y}\sum_{x\epsilon X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}, \tag{3}$$

$$NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}, \tag{4}$$

where $I(X, Y)$ is the mutual information function between clusters X and Y, $p(x)$ and $p(y)$ are marginal distribution functions and $H(X)$, $H(Y)$ are entropy functions corresponding clusters X and Y. Let $E = \{C_1, C_2, \cdots, C_n\}$ be the library of different clustering solutions where C_i denotes the *ith* clustering result, $NMI(C, C_i)$ be the function given in equation (4) Strehl and Ghosh (2002). The Sum of Normalized Mutual Information (*SNMI*) defined as

$$SNMI(C,E) = \sum_{i=1}^{n} NMI(C,C_i), \tag{5}$$

where $SNMI(C, E)$ be a function that measures accuracy (Fern and Lin 2008). The joint criterion method merges quality and diversity terms in the same objective function given in Eq. (6). The following objective function introduced in Eq. (6), known as the "Aggregated Objective Function (AOF)" is maximized for a given ensemble size of K to determine the indices of the best candidates within the ensemble, where

the first term assesses quality (accuracy), the second term assesses diversity, and the parameter α regulates the weighting of each term (Fern and Lin 2008) as

$$\alpha \sum_{i=1,\ldots,K} SNMI(C_i, L) + (1-\alpha) \sum_{i \neq j} (1 - NMI(C_i, C_j)). \tag{6}$$

Here, $SNMI(C_i, L)$ calculates the quality of a clustering solution, C_i, where L represents a large library of clustering solutions and the diversity is measured by $(1 - NMI(C_i, C_j))$.

2.2 Cluster and select method

The cluster and select method, developed by Fern and Lin (Fern and Lin 2008), considers each clustering solution in the ensemble library as an entity and analyzes the relation to one another. It is stated that in order to reduce redundancy if C_1 and C_2 are similar clustering solutions and C_1 is not chosen in the ensemble, C_2 is also not chosen despite its high quality. Only one clustering solution is chosen from each group of clustering solutions, which are grouped according to their similarities. The library of clustering solutions is divided into k groups, with each group consisting of a set of related solutions for a specific ensemble of size k. The clustering solution with the highest quality within each group is chosen to obtain the subset of the ensemble using a quality measure denoted by *SNMI*, after clustering solutions have been divided into k groups. The pairwise *NMI* matrix is subjected to spectral clustering (Ng et al. 2001) to divide the clustering solutions into k groups.

2.3 The evidence accumulation method (EAM)

The evidence accumulation model, which can be thought of as a kernel transformation from the original data space to a co-association matrix, gathers data from base partitions in a clustering ensemble method. However, because this transformation may result in the loss of cluster structure information, some methods suggested in the literature make an effort to retrieve the data and add it back to the ensemble process. An intriguing phenomenon is introduced in the study (Zhong et al. 2019) which is removing some evidence from the co-association matrix can result in more accurate clustering results. The instinctive explanation that comes to mind is that some of the evidence in the original co-association matrix could be noise, which would have a negative impact on the final clustering. However, such evidence is difficult to detect, let alone remove from the matrix. To address this issue, Zhong et al. (2019) remove multiple-level evidence with low occurrence frequencies, as negative evidence only sometimes occurs on a regular basis in the base partitions. Then, they use normalized cut to obtain multiple clustering results.

Figure 1 gives an overview of this approach where CM represents co-association matrix and MM represents newly defined internal validity index. The clustering solution with the highest quality within each group is chosen to determine the ensemble subset using a quality metric *NMI*.

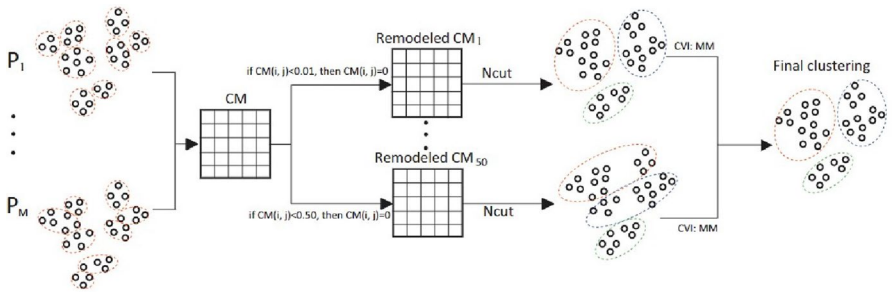


Fig. 1 The overview of EAM (Zhong et al. 2019)

2.4 PrunedOPT

The model that optimizes the accuracy and diversity trade-off given below is the minimization of DC functions over a convex set:

$$\underset{x}{\text{minimize}} \quad \{\tau\|x\|_2^2 - [x^T(\widetilde{G} + \tau I)x - \rho\|x\|_0]\}, \tag{7}$$

where $\|.\|_0$ refers to zero norm and $\|.\|_2$ refers to the Euclidean norm (Özöğür-Akyüz et al. 2020). Here, zero norm approximated as following

$$\|x\|_0 = \lim_{\epsilon \to 0} \sum_{i=1}^{n} \frac{\log(1+ \mid x_i \mid /\epsilon)}{\log(1 + 1/\epsilon)}$$

and then, the limit is neglected by choosing $\epsilon > 0$. By penalization with a regularization constant, the model is independent of the ensemble size k but if the best k is selected through an exhaustive search, the algorithm's performance becomes unacceptably sluggish in terms of simple speed.Therefore, (Özöğür-Akyüz et al. 2020) proposed a formulation reduces the search space by introducing a new parameter ρ that can be defined within some interval of real values.

3 The proposed pruning model

In this section, we demonstrate the generation of the ensemble, ensemble pruning with proposed model, and finally, the aggregation of clusters, as it is illustrated in the flowchart given by Fig. 2.

The proposed model (2), which will soon be demonstrated, will be referred to as *ESC_SOCP*. Three fundamental steps are used to construct ensemble cluster selection. The first step involves the generation of the ensemble using k-means as a base clustering algorithm. The second step is formed by solving *ECS_SOCP* to obtain a subset of the ensemble. Finally, in the last step, we aggregate the clustering decisions of the ensemble found in Step 1 using Hyper-Graph Partitioning Algorithm (HPGA) Strehl and Ghosh (2002).

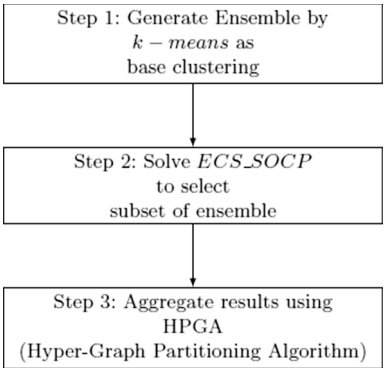


Fig. 2 Flowchart of the proposed *ECS_SOCP*

3.1 Step 1: Generation of ensemble

Generation step is built on obtaining different clustering solutions. For generating numerous base cluster solutions, different clustering algorithms created with various parameters of the same clustering algorithm, or different clustering algorithms, various objects representations, different subsets of the data collection, and projections of objects to different spaces can be used (Sarumathi et al. 2013).

The generation step of the ensemble library, is composed of three steps in this study: (i) Random Initialization (ii) Random Feature Selection (iii) Random Linear Projection. Herein, k-means is used as a base learner, where diverse clustering solutions are obtained by running the k-means algorithm with random parameters which corresponds to step (i). Then, in step (ii), the clustering solutions for each randomly selected features are added to the ensemble's library. Lastly, step (iii) was used to lower the dimensionality of the data while preserving its geometrical structure by projecting the original features into lower dimensions using a random projection matrix.

3.2 Step 2: Ensemble pruning with *ECS_SOCP*

Let us consider the below model (8)

$$\begin{aligned} \underset{v}{\text{maximize}} \quad & v^T \widetilde{H} v \\ \text{subject to} \quad & v^T G v = 4k, \\ & v_0 = 1, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{8}$$

All of the necessary transformations for the matrices $\widetilde{H}$, G and the cardinality constraint can be seen in Appendix A.

At this juncture, model (8), which is the MC-k, can be transformed into to following non-convex quadratic form where should be note that, the constraint $v_0 = 1$ can be relaxed to $v_i \in \{-1, 1\}$ without changing the problem due to $-v$ is being feasible for

the remaining constraints if and only if v is, so, the model (8) can be express as the following:

$$\begin{aligned} &\text{minimize} && \theta \\ &\text{subject to} && v^T\widetilde{H}v \le \theta, \\ &&& v^T G v = 4k, \\ &&& v_i \in \{-1, 1\},\ \forall i \ne 0. \end{aligned} \tag{9}$$

Consider the inequality constraint

$$v^T\widetilde{H}v \le \theta \tag{10}$$

satisfied by (v, V) and use spectral decomposition of $\widetilde{H}$ as $\tilde{H} = \sum_{j=1}^{n+1} \lambda_j q_j q_j^T$. Assume that $\lambda_1 \ge \ldots \ge \lambda_l \ge 0 > \lambda_{l+1} \ge \ldots \ge \lambda_{n+1}$, without loss of generality, where λ_j represents the eigenvalues and q_j is the unit eigenvectors corresponding to these eigenvalues. Let $\tilde{H}^+ = \sum_{j=1}^{l} \lambda_j q_j q_j^T$. By choosing $\widetilde{H}^+$ and $q_j q_j^T$, $j = l+1, \ldots, n$ we obtain:

$$v^T\widetilde{H}^+v - \widetilde{H}^+V \le \theta, \tag{11}$$

$$v^T q_j q_j^T v - q_j q_j^T V \le \theta, \tag{12}$$

where $V = v^T v$. Using equations (10) and (11), we produce a new inequality as follows:

$$v^T\widetilde{H}^+v + \sum_{j=l+1}^{n+1} \lambda_j q_j q_j^T \le \theta, \tag{13}$$

where (v, V) satisfies Eqs. (10) and (11). Thus, this leads that (v, V) also satisfies the Eq. (13). We transform MC-k in Eq. (9) into SOCP relaxation using parameter-reducing technique as follows

$$\begin{aligned} &\text{minimize} && \theta \\ &\text{subject to} && v^T\widetilde{H}^+v + \sum_{j=l+1}^{n+1} \lambda_j z_j \le \theta, \\ &&& v^T G v = 4k + z_j,\ j = 1, \ldots, n+1, \\ &&& z_j \le \sqrt{n},\ j = l+1, \ldots, n+1. \end{aligned} \tag{2}$$

Here, the bound for $z_j = q_j^T V q_j$ arrives considering the fact that V_{ij} takes on either $+1$ or -1 and $\|q_j\| = 1$. As we solve SOCP model (2) (*ECS_SOCP*), we choose a threshold value for v_i when transforming solution vector v to the binary solution. Therefore, the binary solution of Algorithm 1 is determined by a threshold value of 0.1 experimentally using cross-validation as

$$\delta = \begin{cases} 1 \text{ if } v \geq 0.1, \\ 0 \text{ if } v \leq 0.1. \end{cases} \tag{14}$$

This choice gives a sparse solution which leads to a decrease in ensemble subset size. The ensemble size, or k, determines the quality of the solution. It should be acknowledged that k is the specific parameter of the model, which affects the accuracy of the method. The selection of high-value k may indicate a deficiency of meaningful candidates. Hence, in spite of the possibility that there may not be significant candidates, it is essential to take a well-calibrated value k. It should be noted that deciding the value of the k is important while solving the problem. Nonetheless, it should be pointed out that we chose the k as $n/2$ while searching for the best solutions for all n. IPMs, i.e., the Barrier Method, is a particular class of algorithms used to solve linear or non-linear convex optimization problems. It augments the objective function with a barrier term to avoid a breach of the inequality constraints, and that causes the unconstrained optimal value to stay in the feasible space. In this article, Matlab function `fmincon` and IPMs is used to solve the model (2). `CVX` library developed in Grant and Boyd (2014) is used to identify the constraints. `CVX` is a software tool that enables users to create models for convex optimization using `MATLAB` programming language. It essentially transforms `MATLAB` into an optimization modeling language. By doing so, it is possible to specify constraints and objectives using the standard expression syntax of `MATLAB`. The algorithm of the model (2) is given by Algorithm 1 below which is implemented in `CVX` library in `MATLAB`.

Algorithm 1 *ECS SOCP*

Input : Dataset, Base learners, n # n: size of ensemble
Parameters : k, δ # k: pruning size
δ: threshold value

Output : NMI values

1: Generate ensemble library $E = \{E_1, E_2, E_3, \ldots, E_n\}$
2: Solve the below model

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H}^+ v + \sum_{j=l+1}^{n+1} \lambda_j z_j \leq \theta, \\ & v^T G v = 4k + z_j, \ j = 1, \ldots, n+1, \\ & z_j \leq \sqrt{n}, \ j = l+1, \ldots, n+1. \end{aligned}$$

3: Find all of the v_i such that $v_i \geq \delta$ for each i-th clustering solution
4: Aggregate the clustering decisions of the subset in 3 using HPGA

3.3 Step 3: Aggregation

Consensus function, a process where all the partitions generated in the generation step are combined to form a new partition can also be split into two categories. These are the co-occurrences of the objects in the data collection and the median partitions. The first of these is heuristic, and the methods in the second category are optimization-based methods based on minimizing the sum of distances. In the literature, Strehl and Ghosh (2002) proposed the consensus function that maximized the *NMI* function, which measures the proximity between partitions or diversity of clustering solutions to each other.

The pruning step should be followed by aggregating the solutions in the new sub-ensemble to make a final clustering. In this study, we employed the HGPA (Strehl and Ghosh 2002), one of the approaches for clustering problems most frequently recommended in the literature. More information on the HGPA approach, along with the offered `MATLAB` code can be found in the literature (Hein et al. 2013).

4 Data sets and experiments

In this study, data sets are chosen from *UCI Machine Learning Repository* (Dua and Graff 2017) which is illustrated with the explanation of the dataset characteristics, attribute characteristics, numbers of instances, number of attributes and number of clusters, respectively, in Table 3. As it explained in the generation step in Sect. 3, generation of the ensemble library is composed of three steps. At each of these three steps, fifty clustering solutions are constructed. Thus, 150 diverse clustering solutions in the ensemble library are obtained. The best candidates for the sub-ensemble among these 150 clustering solutions are selected through the Algorithm 1.

We implemented the joint criteria, the cluster and select, EAM and PrunedOPT methods to compare the ensemble pruning model's performance with the *ECS_SOCP* in Table 4. The Joint Criteria Method combines quality and diversity components into a single criterion function. In contrast, the Cluster and Select Method considers each clustering solution in the ensemble as an entity and analyzes the relationship between them. The EAM, a kernel transformation from the original data space to a co-association matrix, collects data from base partitions. Then, it removes some evidence from the co-association matrix to get more accurate clustering results. Finally, DCCP is used to solve the model PrunedOPT which is an optimization based model selects the optimal subset of the ensemble that has both accurate and diverse models. Our study stands out from above approaches as we take into account both accuracy and diversity. Although PrunedOPT also considers this trade-off, we model the max-cut problem as SOCP, enabling solving integer programming continuously and convexly. This gives us an edge over difference of convex programming.

The α value in Eq. (6) was choosen as same value as in the study (Fern and Lin 2008), 0.5, for the joint criterion algorithm. The paremeter ρ for the model PrunedOPT is selected among the values $\rho = [10^{-1}, 10^{-2}, 10^{-3}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}, 10^{6}]$ by

Table 3 Detailed Information of Datasets

Data	DataSet Char.	Attribute Char.	#of Instance	#of Attribute	#of Cluster
Non-Verbal Tourist	Multivariate	Integer, Real	73	22	6
E-shop Clothing	Multivariate, Sequential	Integer, Real	165474	14	5
Wine	Multivariate	Integer, Real	178	13	3
Synthetic control	Time-Series	Real	600	N/A	6
Segmentation	Multivariate	Real	2310	19	7
Glass	Multivariate	Real	214	10	6
Bupa	Multivariate	Categorical, Integer, Real	345	7	2
Frog_genus	Multivariate	Real	7195	22	8
Iris	Multivariate	Real	150	4	3
Diabets	Multivariate	Time Series	768	20	2
Wdbc	Multivariate	Real	569	32	2
Wpbc	Multivariate	Real	198	34	2
Ionosphere	Multivariate	Categorical, Integer, Real	351	34	2
Zoo	Multivariate	Categorical, Integer	101	17	7

cross validation. Furthermore, the best ρ value differs for each data sets with cross validation. It should be noted that the proposed approach **ECS_SOCP** achieves *NMI* values comparable to or better than the mean value results of the other methods. Furthermore, apart from the PrunedOPT, as the other methods require the pruning rate to be a hyper-parameter, the best size information found in our algorithm is given as input for these methods, which yields an unfair gain for their performances and should be considered.

Because the problem under consideration is unsupervised, the value of mutual information among clusters represents the quality of the clustering decision. In other words, as the *NMI* values get closer to 1, we get better-quality clustering results. The performances of the models are presented utilizing the mean values of *NMI* with bold font for better result and for each result, confidence intervals are given with square parenthesis in Table 4. If we look closer to the results in Table 4, the *ECS_SOCP* method stands out as the superior approach when analyzing business data for Non-verbal tourist and E-shop clothing, as well as for Glass data. On the other hand, Joint Criteria method for Wpbc data and Cluster and Select method for Wine and Segmentation data gives better results. For Bupa, Frog_genus, Iris, and Wdbc data, EAM method is more effective. As for Synthetic Control and Diabetes data, PrunedOPT is the preferred option. It is worth mentioning that there is no universally accepted standard in machine learning, and the outcomes can vary depending on the data sets employed. Nonetheless, our approach appears to yield superior results based on the mean value, as indicated in Table 4.

Table 4 *NMI* results, 95% confidence intervals

Data	Joint Criteria	Cluster &Select	EAM	Pruned OPT	*ECS_SOCP*	*ECS_SOCP* Best Size
Non-verbal tourist	0.32562 [0.204,0.381]	0.32562 [0.002,0.226]	0.18408 [0.184,0.184]	0.32565 [0.022,0.4262]	**0.35281** [0.230,0.404]	78
E-shop clothing	0.01487 [0.009,0.019]	0.01487 [0.008,0.191]	NAN	0.01622 [0.014,0.017]	**0.01654** [0.004,0.021]	96
Wine	0.70693 [0.700,0.713]	**0.71415** [0.709,0.719]	0.43149 [0.421,0.441]	0.65148 [0.643,0.659]	0.71077 [0.708,0.713]	81
Synthetic Control	0.77803 [0.716,0.839]	0.78284 [0.733,0.832]	0.79617 [0.757,0.835]	**0.87547** [0.871,0.879]	0.81947 [0.778,0.860]	93
Segmentation	0.49268 [0.443,0.541]	**0.53918** [0.537,0.550]	0.52128 [0.572,0.583]	0.46216 [0.405,0.518]	0.52198 [0.501,0.541]	90
Glass	0.67486 [0.301,0.552]	0.67877 [0.210,0.466]	0.36799 [0.359,0.376]	0.62821 [0.625,0.630]	**0.73110** [0.709,0.753]	113
Bupa	0.002369 [0.001,0.003]	0.003378 [0.001,0.005]	**0.012035** [0.012,0.012]	0.00095 [0.0008,0.001]	0.002639 [0.0001,0.003]	86
Frog_genus	0.41006 [0.049,0.360]	0.44999 [0.159,0.470]	**0.54386** [0.541,0.546]	0.44388 [0.299,0.499]	0.45705 [0.055,0.401]	106
Iris	0.69156 [0.294,0.455]	0.67373 [0.621,0.687]	**0.80422** [0.799,0.808]	0.70014 [0.699,0.7007]	0.68359 [0.312,0.477]	98
Diabetes	0.042443 [0.039,0.045]	0.048117 [0.041,0.054]	0.000802 [0.0008,0.0008]	**0.06699** [0.064,0.069]	0.042891 [0.390,0.046]	95
Wdbc	0.,39996 [0.011,0.308]	0.39016 [0.089,0.378]	**0.41431** [0.414,0.414]	0.37307 [0.251,0.420]	0.39476 [0.263,0.447]	106
Wpbc	**0.050578** [0.023,0.057]	0.04007 [0.024,0.055]	0.020618 [0.020,0.020]	0.02514 [0.023,0.026]	0.040235 [0.026,0.054]	102
Ionosphere	0.12498 [0.118,0.131]	**0.13268** [0.122,0.143]	0.11619 [0.107,0.127]	0.1272 [0.124,0.129]	0.12432 [0.116,0.132]	87
Zoo	0.70669 [0.247,0.741]	0.72641 [0.256,0.760]	0.76314 [0.763,0.763]	**0.77688** [0.750,0.803]	0.74824 [0.448,0.750]	97
Mean	0.387259	0.391421	0.382781	0.39096	**0.403313**	**95**

5 Conclusion

This paper introduces a novel ensemble selection model for clustering problem using optimization for the accuracy and diversity trade-off. Furthermore, the proposed optimization model automatically determines the optimal pruning size, unlike most of the existing methods in the literature. The ensemble cluster selection is constructed using three fundamental steps. The first step is to create the ensemble using k-means as the basic clustering technique. The second step is solving *ECS_SOCP* to

acquire a subset of the ensemble. Finally, HPGA is used in the last step to combine the clustering conclusions of the ensemble identified in step one. We evaluated its performance in comparison with the joint criteria method, cluster and select method, EAM and PrunedOPT. Experimental results on 14 *UCI* real datasets where two of them are the business data, indicate that the proposed pruning algorithm approximates the optimal solution in a unified framework using the optimization model, and *ECS_SOCP* usually achives better or comparable *NMI* results. This approach is an example of how OR can be used to optimize ensemble clustering and improve decision-making in complex business environments. In conclusion, we can summarize the results by

- We tested the joint criteria, cluster selection, EAM, PrunedOPT methods, and our method *ECS_SOCP* in real data sets. Overall, *ECS_SOCP* achieved desirable results,
- One advantage of ensemble pruning using *ECS_SOCP* is finding the subset automatically without the need of the pruning size, which sets it apart from many other methods discussed in the literature,
- According to Table 4, it can be observed that using *ECS_SOCP* provides better outcomes for business data, making it a beneficial aspect for OR applications.

The innovative use of operations research can optimize clustering and improve decision-making in complex business environments. This methodology allows for efficient data analysis, providing key insights and trends to gain a competitive edge. It's a game-changer for businesses looking to stay ahead and achieve long-term success.

It is worth noting that determination of the threshold value δ introduced in equation (14) can be considered as a limitation of the proposed model which can be integrated into the model as an additional variable in the future study.

In future work, we will explore using other quality and diversity measures instead of *NMI*. Another future direction is to apply the proposed method to the classification problem, future selection, deep learning and, image processing. To be more precise, the proposed optimization model will be adapted to clustering selection algorithms using feature selection methods similar to the works (Jia et al. 2011; Yu et al. 2016).

Appendices

A. Derivations

Consider the below model, which optimizes the accuracy and diversity trade-off,

$$\begin{aligned} \text{maximize} \quad & x^T \widetilde{D} x \\ \text{subject to} \quad & \sum_i x_i = k, \\ & x_i \in \{0, 1\}, \end{aligned} \tag{15}$$

where $\widetilde{D}$ is the normalization of the accuracy-diversity matrix D defined in equation (1). This model can be treated as an NP-hard integer problem. Fortunately, this formulation is similar to the Maximum Cut optimization problem with the size k (MC-k) in the literature (Muramatsu and Suzuki 2003) which partitions the vertices of an edge-weighted graph into two vertex subsets in such a manner that the sum of the weights of the edges that have an endpoint in each of the subsets is maximum. Even though finding a polynomial-time algorithm for the maximum cut problem appears to be nearly impossible, various heuristics/combinations optimization and heuristic methods have been developed to solve this problem. The MC-k problem is formulated as follows:

$$\begin{aligned} \underset{y}{\text{maximize}} \quad & \frac{1}{2}\sum_{i<j} w_{ij}(1-y_i y_j) \\ \text{subject to} \quad & \sum_i y_i = N_v - 2k, \\ & y_i \in \{-1,1\}, \end{aligned} \tag{16}$$

where N_v is the number of vertices in the graph, and $w_{i,j}$ is the vertex weight between i and j. If the ensemble pruning method is formulated to resemble the Maximum Cut (MC) problem, a relaxation can be made for the SOCP which gives a better solution. Let us apply same methodology in Zhang et al. (2006) with our matrix $\widetilde{D}$. First of all, the MC-k formulation (16) can be expressed as follows where the edge weight matrix ($w_{i,i} = 0$) is defined as W:

$$\begin{aligned} \underset{y}{\text{minimum}} \quad & y^T W y \\ \text{subject to} \quad & \sum_i y_i = N_v - 2k, \\ & y_i \in \{-1,1\}. \end{aligned} \tag{17}$$

Since the key difference between Model (15) and Model (17) is $x_i \in \{0,1\}$ and $y_i \in \{-1,1\}$, a variable conversion is required. Let $v_i = \{-1,1\}$ and $x_i = \dfrac{v_i+1}{2}$. Hence, the new objective function in Model (15) is transformed into $\frac{1}{4}(v+e)^T\widetilde{D}(v+e)$ by taking e as the column vector of all 1's. Therefore, $\sum_{i=1}^{n} x_i = k$ which is the cardinality constraint, can be expressed in quadratic form as $x^T I x = k$, where I is the identity matrix. After variable conversion, we obtain $(v+e)^T I(v+e) = 4k$. Since the objective function and the cardinality constraint are changed, the relaxation can be made from $v = (v_1, v_2, v_3, \ldots, v_n)$, $v_0 = 1$ to $v = (v_0, v_1, v_2, v_3, \ldots, v_n)$ to put it back into quadratic form. Then the newly constructed matrix is $H_{(n+1)\times(n+1)} = \begin{pmatrix} e^T\widetilde{D}e & e^T\widetilde{D} \\ \widetilde{D}e & \widetilde{D} \end{pmatrix}$ and hence, the objective function can be expressed as $v^T H v$. Similarly, the same method can be used to obtain the matrix G as $G_{(n+1)\times(n+1)} = \begin{pmatrix} n & e^T \\ e & I \end{pmatrix}$ and the cardinality constraint can be expressed

as $v^T Gv = 4k$. Thus, the problem becomes the below model (8) by considering $-H = \widetilde{H}$;

$$\begin{aligned} \underset{v}{\text{maximize}} \quad & v^T \widetilde{H} v \\ \text{subject to} \quad & v^T G v = 4k, \\ & v_0 = 1, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{8}$$

In the study (Zhang et al. 2006), SDP is used to solve the model (8). At this juncture, model (8), which is the MC-k, can be transformed into to following non-convex quadratic form

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H} v \leq \theta, \\ & v^T G v = 4k, \\ & v_0 = 1, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{18}$$

Note that, the constraint $v_0 = 1$ can be relaxed to $v_i \in \{-1, 1\}$ without changing the problem due to $-v$ is being feasible for the remaining constraints if and only if v is, so, the model (18) becomes the following:

$$\begin{aligned} \text{minimize} \quad & \theta \\ \text{subject to} \quad & v^T \widetilde{H} v \leq \theta, \\ & v^T G v = 4k, \\ & v_i \in \{-1, 1\},\ \forall i \neq 0. \end{aligned} \tag{9}$$

B. Background

Second order conic programming (SOCP)

In this section, basic definitions and formulations for SOCP will be introduced.

SOCP problems are convex optimization problems that involve minimizing a linear function over the intersection of an affine linear manifold and the Cartesian product of second-order (Lorentz) cones. Linear programs, convex quadratic programs, and second-order constrained convex quadratic programs can all be described as SOCP problems, as well as numerous others that do not fall within the stated categories. Moreover, SDP, which is the optimization problem over the intersection of an affine set and the cone of positive semi-definite matrices, includes SOCP as a special case. As a result, SOCP lies between LP and QP and SDP. SOCP problems, like LP, QP, and SDP problems, can be solved using Interior Point Methods (IPMs) in polynomial time. IPMs require more computational effort per iteration to solve SOCP problems than LP and

QP problems but less than SDPs of comparable size and complexity (Alizadeh and Goldfarb 2003). Consider the below SOCP

$$\begin{aligned} &\text{minimize} \quad f^T x \\ &\text{subject to} \quad \|A_i x + b_i\| \leq c_i^T x + d_i, \ i = 1, \ldots, N, \end{aligned} \tag{19}$$

where $x \in \mathbb{R}^n$ is the variable and parameters are $f \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{(n_i-1)\times n}$, $b_i \in \mathbb{R}^{n_i-1}$, $c_i \in \mathbb{R}^n$, and $d_i \in \mathbb{R}$. The norm emerging in the constraint refers to the standard Euclidean norm i.e., $\|z\| = (z^T z)^{1/2}$. The constraint $\|A_i x + b_i\| \leq c_i x + d_i$ is called a SOC, specifically, Lorentz Cone, constraint of dimension n_i since the standard or unit SOC of dimension k is defined as $\mathcal{K}_m = \left\{ \begin{bmatrix} u \\ t \end{bmatrix} \;\middle|\; u \in \mathbb{R}^{m_i-1}, t \in \mathbb{R}, \|u\| \leq t \right\}$. For $m = 1$, the unit second-order cone is defined as $\mathcal{K}_1 = \left\{ t \;\middle|\; t \in \mathbb{R}, 0 \leq t \right\}$. The set of points satisfying a constraint of the quadratic cone is the inverse image of the unit quadratic cone according to an affine mapping:

$$\|A_i x + b_i\| \leq c_i^T x + d_i \iff \begin{bmatrix} A_i \\ c_i{}^T \end{bmatrix} x + \begin{bmatrix} b_i \\ d_i \end{bmatrix} \in \mathcal{K}_{n_i}$$

and thus is convex. Therefore, model (19) is a convex programming problem since the objective is a convex function and the constraints define a convex set. Since $\|u\| \leq t \iff \begin{bmatrix} tI & u \\ u^T & t \end{bmatrix} \succeq 0$, the second-order cone can be embedded into cone of positive semi-definite matrices, i.e., a second-order cone constraint is equivalent to a linear matrix inequality ($\succeq$ denotes matrix inequality). Using this property, model (19) can be expressed as a SDP qua follows:

$$\begin{aligned} &\text{minimize} \quad f^T x \\ &\text{subject to} \quad \begin{bmatrix} (c_i^T x + d_i) I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0; i = 1, \ldots, N. \end{aligned} \tag{20}$$

IPMs that directly solve the SOCP have substantially better worst-case complexity than SDP methods applied to Problem (20): the number of iterations required to reduce the duality gap to a constant fraction of itself is bounded from above by $O(\sqrt{n})$ for the SOCP algorithm and $O(\sum_i n_i)$ for the SDP approach (Nesterov and Nemirovskii 1994). Besides, most notably in practice, each iteration is considerably faster: the amount of work per iteration in the SOCP algorithm is $O(n^2 \sum_i n_i)$ and for the SDP is $O(n^2 \sum_i n_i^2)$. The difference between these numbers is quite noteworthy when the dimension n_i of the second-order constraints is large. To simplify notations in equation (19), let $u_i = A_i x + b_i, t_i = c_i^T x + d_i; i = 1, \ldots, N$.

$$\begin{aligned} &\text{minimize} \quad f^T x \\ &\text{subject to} \quad \|u_i\| \leq t_i, \ i = 1, \ldots, N, \\ &\qquad\qquad u_i = A_i x + b_i, t_i = c_i^T x + d_i \ i = 1, \ldots, N. \end{aligned} \tag{21}$$

Dual SOCP problem

The dual of equation (19) is given by

$$\begin{aligned} \text{maximize} \quad & -\sum_{i=1}^{N}(b_i^T z_i + d_i w_i) \\ \text{subject to} \quad & \sum_{i=1}^{N}(A_i t + z_i + c_i w_i) = f, \\ & \|z_i\| \le w_i \; i = 1, \ldots, N, \end{aligned} \tag{22}$$

where the variables are the vectors $z_i \in \mathbb{R}^{n_i - 1}$, and $w \in \mathbb{R}^N$. Here, a set of z_i, $i = 1, \ldots, N$ is denoted by z. Equation (22) (dual SOCP) is also a convex programming problem since the objective function is concave, and the constraints are convex. As a matter of fact, i24as the same form as the equation (21). Alternately, we can reform the dual SOCP in the same form as primal SOCP (equation (19)) by eliminating the equality constraints.

If there is an x that satisfies all of the constraints in the primal SOCP problem, then equation (19) is said to be feasible. If there is an x that satisfies the constraints with strict inequality, it is said to be strictly feasible. The vectors z and w are called dual feasible if they satisfy the constraints in equation (22) and strictly dual feasible if they also satisfy $\|z_i\| \le w_i \; i = 1, \ldots, N$. It is said that (dual SOCP) the equation (22) is (strictly) feasible if there exist (strictly) feasible z_i, w. Let us denote the optimal value of the (primal SOCP) equation (19) as $\mathfrak{p}^*$, with the agreement that $\mathfrak{p}^* = \infty$ if the problem is infeasible and the optimal value of the (dual SOCP) equation (22) as $\mathfrak{d}^*$, with the agreement that $\mathfrak{d}^* = -\infty$ if the problem is infeasible. The basic facts about the dual problem are given in three cases:

1. (*Week Duality*) $\mathfrak{p}^* \ge \mathfrak{d}^*$,
2. (*Strong Duality*) If the primal or dual problem is strictly feasible, then $\mathfrak{p}^* = \mathfrak{d}^*$,
3. If the primal and dual problems are strictly feasible, then there exist primal and dual feasible points that attain the (equal) optimal values.

The difference between the primal and dual objectives is called the *duality gap*, and will be denoted by $\eta(x, z, w)$:

$$\eta(x, z, w) = f^T x + \sum_{i=1}^{N}(b_i^t z_i + d_i w_i). \tag{23}$$

Weak duality stands for the fact that the duality gap is not always negative for any feasible x, z, w. To see this, observe the duality gap being expressed as a sum of nonnegative numbers as follows:

$$\eta(x, z, w) = \sum_{i=1}^{N}(z_i^T(A_i x + b_i) + w_i(c_i^T x + d_i)) = \sum_{i=1}^{N}(z_i^T u_i + w_i t_i). \tag{24}$$

Here, each term on the right hand side is non-negative: $z_i^T u_i + w_i t_i \geq -\|z_i\|\|u_i\| + w_i t_i \geq 0$. It is also possible to the case 3 of the duality result as follow: If the problem is strictly primal and dual feasible, then primal and dual feasible points with a zero duality gap exist. By investigating the equation (24), we can see the duality gap is zero if and only if the following conditions are satisfied:

$$\|u_i\| \leq t_i \implies w_i = \|z_i\| = 0, \tag{25}$$

$$\|z_i\| \leq w_i \implies t_i = \|u_i\| = 0, \tag{26}$$

$$\|z_i\| = w_i, \|u_i\| = t_i \implies w_i u_i = -t_i z_i. \tag{27}$$

The complimentary slackness conditions between the optimum primal and dual solutions in LP are generalized by these three conditions. They also provide a sufficient condition for optimality since a primal feasible point x is optimal if there exists z, w such that equations (25–27) hold for $u_i = A_i x + b_i$ and $t_i = c_i^T x + d_i$. Thereby, the conditions are also necessary if the primal and dual problems are strictly feasible.

Acknowledgements These study is a part of PhD Dissertation of Doctoral Programme of Yildiz Technical University. We would also like to express our gratitude for the Scientific and Technological Research Council of Turkey (TÜBİTAK) Project numbered as 119E100 which provided funding for our research.

Author Contributions Software, Validation, Methodology, Writing- Original draft preparation, DÜ. Conceptualization, Methodology, Reviewing and Editing, SA. Conceptualization, Methodology, EG. All authors have no conflicts of interest.

Data Availibility The experimental data results that support the findings of this study are available in *UCI machine learning repository* with the identifier http://archive.ics.uci.edu/ml.

References

Ali MA, Ucuncu D, Atas PK, Ozogur-Akyuz S (2020) Classification of motor imagery task by using novel ensemble pruning approach. IEEE Trans Fuzzy Syst 28:85–91

Alizadeh F, Goldfarb D (2003) Second-order cone programming. Math Program 95:3–51

Alqurashi T, Wang W (2019) Clustering ensemble method. Int J Mach Learn Cybern 10:1227–1246

Azimi J, Fern X (2006) Adaptive cluster ensemble selection, pp 992–997

Bakker B, Heskes T (2003) Clustering ensembles of neural network models. Neural Netw 16:261–269

Cover TM, Thomas JA (1991) Elements of Information Theory, 2nd edn. Wiley, New York

Csehi CG, Farkas M (2017) Truck routing and scheduling. CEJOR 25(4):791–807

Dai Q (2013) A competitive ensemble pruning approach based on cross-validation technique. Knowl-Based Syst 37:394–414

Dawid H, Decker R, Hermann T, Jahnke H, Klat W, König R, Stummer C (2017) Management science in the era of smart consumer products: challenges and research perspectives. CEJOR 25:203–230

Delen D (2010) A comparative analysis of machine learning techniques for student retention management. Decis Support Syst 49(4):498–506

Du H-Y, Wang W-J (2022) A clustering ensemble framework with integration of data characteristics and structure information: a graph neural networks approach. Mathematics 10(11):1834. https://doi.org/10.3390/math10111834

Dua D, Graff C (2017) UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

Esteso A, Alemany M, Ortiz A, Liu S (2022) Optimization model to support sustainable crop planning for reducing unfairness among farmers. CEJOR 30(3):1101–1127
Ewertowski T, Güldoğuş BÇ, Kuter S, Akyüz S, Weber G-W, Sadłowska-Wrzesińska J, Racek E (2023) The use of machine learning techniques for assessing the potential of organizational resilience. Cent Eur J Oper Res 1–26. https://doi.org/10.1007/s10100-023-00875-z
Fern XZ, Lin W (2008) Cluster ensemble selection. Stat Anal Data Min: The ASA Data Sci J 1:128–141
Forgy EW (1965) Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics 21:768–769
García-Pedrajas N, Hervás-Martínez C, Ortiz-Boyer D (2005) Cooperative coevolution of artificial neural network ensembles for pattern classification. IEEE Trans Evol Comput 9:271–302
Garcia-Vicuña D, Esparza L, Mallor F (2022) Hospital preparedness during epidemics using simulation: the case of covid-19. Cent Eur J Oper Res 30(1):1–37
Grant M, Boyd S (2014) CVX: Matlab Software for Disciplined Convex Programming, version 2.1. http://cvxr.com/cvx
Gupta JN, Forgionne GA, Mora M (2007) Intelligent decision-making support systems: foundations, applications and challenges. Springer Science & Business Media
Gür Ş, Pınarbaşı M, Alakaş HM, Eren T (2022) Operating room scheduling with surgical team: a new approach with constraint programming and goal programming. Cent Eur J Oper Res 31:1–25
Gütmen S, Roy SK, Weber G-W (2023) An overview of weighted goal programming: a multi-objective transportation problem with some fresh viewpoints. Cent Eur J Oper Res 1–12. https://doi.org/10.1007/s10100-023-00861-5
Hayden MG (1998) The Ensemble System. Cornell University
Heine OFC, Thraves C (2023) On the optimization of pit stop strategies via dynamic programming. CEJOR 31(1):239–268
Hein M, Setzer S, Jost L, Rangapuram SS (2013) The total variation on hypergraphs - learning on hypergraphs revisited. In: Advances in neural information processing systems, vol 26
Huang D, Lai J-H, Wang C-D, Yuen PC (2016) Ensembling over-segmentations: from weak evidence to strong segmentation. Neurocomputing 207:416–427
Jia J, Xiao X, Liu B, Jiao L (2011) Bagging-based spectral clustering ensemble selection. Pattern Recogn Lett 32:1456–1467
Jiang X, Wu C-A, Guo H (2017) Forest pruning based on branch importance. Comput Intell Neurosci 2017:1–11. https://doi.org/10.1155/2017/3162571
Lazarevic A, Obradovic Z (2001) Effective pruning of neural network classifier ensembles. In: International joint conference on neural networks, vol 2, pp 796–801. https://doi.org/10.1109/ijcnn.2001.939461
Li N, Zhou Z-H (2009) Selective ensemble under regularization framework. In: Multiple classifier systems, vol 5519. https://doi.org/10.1007/978-3-642-02326-2_30
Lu Z, Wu X, Zhu X, Bongard J (2010) Ensemble pruning via individual contribution ordering, p. 871. ACM Press, New York, New York, USA. https://doi.org/10.1145/1835804.1835914. http://dl.acm.org/citation.cfm?doid=1835804.1835914
Martinez-Munoz G, Hernandez-Lobato D, Suarez A (2009) An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell 31:245–259. https://doi.org/10.1109/TPAMI.2008.78
Martínez-Muñoz G, Suárez A (2004) Aggregation ordering in bagging. In: Hamza MH (ed) pp 258–263. International conference on artificial intelligence and applications
Muramatsu M, Suzuki T (2003) A new second-order cone programming relaxation for max-cut problems. J Op Res Soc Jpn 46:164–177. https://doi.org/10.15807/jorsj.46.164
Nesterov Y, Nemirovskii A (1994) Interior-point polynomial algorithms in convex programming. Soc Ind Appl Math 10(1137/1):9781611970791
Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. In: Advances in neural information processing systems, vol 14. https://proceedings.neurips.cc/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf
Özöğür-Akyüz S, Çisil Otar B, Atas PK (2020) Ensemble cluster pruning via convex-concave programming. Comput Intell 36:297–319. https://doi.org/10.1111/coin.12267
Partalas I, Tsoumakas G, Vlahavas I (2010) An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. Mach Learn 81:257–282. https://doi.org/10.1007/s10994-010-5172-0

Perišić A, Pahor M (2023) Clustering mixed-type player behavior data for churn prediction in mobile games. CEJOR 31(1):165–190
Polikar R (2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3):21–45
Roy SK, Maity G, Weber G-W (2017) Multi-objective two-stage grey transportation problem using utility function with goals. CEJOR 25:417–439
Sarumathi S, Shanthi N, Sharmila M (2013) A comparative analysis of different categorical data clustering ensemble methods in data mining. Int J Comput Appl 81:46–55
Singh V, Mukherjee L, Peng J, Xu J (2010) Ensemble clustering using semidefinite programming with applications. Mach Learn 79:177–200
Strehl A, Ghosh J (2002) Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617. https://doi.org/10.1162/153244303321897735
Torjai L, Kruzslicz F (2016) Mixed integer programming formulations for the biomass truck scheduling problem. CEJOR 24:731–745
Tsoumakas G, Partalas I, Vlahavas I (2009) An ensemble pruning primer. In: Applications of supervised and unsupervised ensemble methods, vol 245, pp 1–13. https://doi.org/10.1007/978-3-642-03999-7_1
Xu L, Li B, Chen E (2012) Ensemble pruning via constrained eigen-optimization. In: 2012 IEEE 12th international conference on data mining, Brussels, pp 715–724. https://doi.org/10.1109/ICDM.2012.97
Yu Z, Li L, Gao Y, You J, Liu J, Wong HS, Han G (2014) Hybrid clustering solution selection strategy. Pattern Recogn 47:3362–3375. https://doi.org/10.1016/j.patcog.2014.04.005
Yu Z, Luo P, You J, Wong HS, Leung H, Wu S, Zhang J, Han G (2016) Incremental semi-supervised clustering ensemble for high dimensional data clustering. IEEE Trans Knowl Data Eng 28:701–714. https://doi.org/10.1109/TKDE.2015.2499200
Zhang L, Zhou W-D (2011) Sparse ensembles using weighted combination methods based on linear programming. Pattern Recogn 44:97–106. https://doi.org/10.1016/j.patcog.2010.07.021
Zhang Y, Burer S, Street WN (2006) Ensemble pruning via semi-definite programming. J Mach Learn Res 7:1315–1338
Zhong C, Hu L, Yue X, Luo T, Fu Q, Xu H (2019) Ensemble clustering based on evidence extracted from the co-association matrix. Pattern Recogn 92:93–106. https://doi.org/10.1016/j.patcog.2019.03.020
Zhou Z-H (2012) Ensemble Methods Foundations and Algorithms. Chapman and Hall
Zhou Z-H, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137:239–263. https://doi.org/10.1016/S0004-3702(02)00190-X

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.