A New Framework for Convex Clustering in Kernel Spaces: Finite Sample Bounds, Consistency and Performance Insights

Shubhayan Pan[1] , **Kushal Bose**[2] , **Debolina Paul**[3] , **Saptarshi Chakraborty**[4] and **Swagatam Das**[2]
[1]Indian Statistical Institute, Kolkata
[2]Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata
[3]Department of Statistics, University of Oxford
[4]Department of Statistics, University of Michigan
shubhayanpan@gmail.com, kushalbose92@gmail.com, debolina.paul@stats.ox.ac.uk,
saptarsc@umich.edu, swagatam.das@isical.ac.in

Abstract

Convex clustering is a well-regarded clustering method, resembling the similar centroid-based approach of Lloyd's k-means, without requiring a predefined cluster count. It starts with each data point as its centroid and iteratively merges them. Despite its advantages, this method can fail when dealing with data exhibiting linearly non-separable or non-convex structures. To mitigate the limitations, we propose a kernelized extension of the convex clustering method. This approach projects the data points into a Reproducing Kernel Hilbert Space (RKHS) using a feature map, enabling convex clustering in this transformed space. This kernelization not only allows for better handling of complex data distributions but also produces an embedding in a finite-dimensional vector space. We provide a comprehensive theoretical underpinning for our kernelized approach, proving algorithmic convergence and establishing finite sample bounds for our estimates. The effectiveness of our method is demonstrated through extensive experiments on both synthetic and real-world datasets, showing superior performance compared to state-of-the-art clustering techniques. This work marks a significant advancement in the field, offering an effective solution for clustering in non-linear and non-convex data scenarios.

Github repo: https://tinyurl.com/KernelizedCC
Extended version: https://arxiv.org/pdf/2511.05159

1 Introduction

Convex clustering is one of the modern frameworks for performing a clustering task, formulating it as a convex optimisation problem, thus ensuring a unique and globally optimal solution. It leverages a fusion penalty to enhance the grouping of the data, helping us to uncover hidden structures in the data. It garnered widespread attention as an alternative avenue that offers relaxations of traditionally non-convex problems [Tropp, 2006]. Given n data points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, convex clustering initially assumes n distinct centroids $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^d$ for each of the n points, and minimises the objective function given by

$$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n} \frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{a}_i\|_2^2 + \gamma \sum_{i<j} w_{ij}\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_q \qquad (1)$$

Here $\|\cdot\|_q$ denotes the ℓ_q norm in $\mathbb{R}^d$, for some $q \geq 1$. The first term measures the fit between $\boldsymbol{x}_i$'s and $\boldsymbol{a}_i$'s, while the latter is a fusion term that penalizes the number of unique $\boldsymbol{a}_i$'s by way of an ℓ_q norm penalty with tuning parameter γ. The weights w_{ij} can be chosen heuristically to accelerate computation and improve empirical performance. It is noteworthy that, for $q \geq 1$, the objective is convex in $\boldsymbol{a}_i$'s, and thus has a minimizer. This convex nature of the objective is attractive from a theoretical viewpoint: works by [Tan and Witten, 2015], [Radchenko and Mukherjee, 2017] provide centroid recovery guarantees, and [Chi and Steinerberger, 2018] establish conditions under which the solution path recovers a tree. Apart from this, it has many other attractive theoretical properties, that has garnered growing interest in it [Hocking *et al.*, 2011; Lindsten *et al.*, 2011; Zhu *et al.*, 2014].

In convex clustering, the number of clusters can be chosen automatically, equating it to the number of distinct $\boldsymbol{u}_i$'s. Indeed, the solution of convex clustering offers a continuous path based on the parameter γ, where a larger γ increases the fusion penalty's influence, leading to fewer unique centres or clusters [Chi and Lange, 2015].

Over the years, different variants of convex clustering have been proposed by different researchers. Some of the recent advances include SpaCC [Nagorski and Allen, 2016] for detecting genomic regions, ACC [Chu *et al.*, 2021] for convex clustering in generalized linear models, TROUT [Weylandt and Michailidis, 2021] for clustering of time series. Most of these variants are data/application specific, reducing their general effectiveness. The reader is advised to refer to [Feng *et al.*, 2024] for furthering their knowledge about the different variants of convex clustering.

On the other hand, kernel methods emerge as a relevant preprocessing step in clustering, as they can identify non-linear data patterns, which conventional clustering techniques overlook. By employing the kernel trick, kernel clustering methods map the data into a higher-dimensional feature

space, where clusters are linearly separable. Kernel k-means [Schölkopf *et al.*, 1998; Girolami, 2002] extends the classical k-means algorithm by incorporating kernel functions such as the Gaussian or polynomial kernels, allowing the algorithm to identify complex, non-linear cluster boundaries [Schölkopf *et al.*, 1998]. This method has proven particularly effective in applications like image segmentation and bioinformatics, where the data often has several intricate structures that are not well identified by linear methods [Girolami, 2002]. Kernel power k means [Paul *et al.*, 2023] is one of the many recent applications of kernel methods in the field of clustering. Other applications in the clustering regime mostly include multi-view clustering like [Park and Park, 2025; Wang *et al.*, 2024; Wu *et al.*, 2024; Li *et al.*, 2024].

Furthermore, [Zhu *et al.*, 2014] studied convex clustering from a theoretical perspective, providing crucial details on a perfect cluster recoveries and other related properties. Additionally, they tried to kernelize convex clustering and formulated it as a second-order cone optimization problem, but did not mention any details regarding its implementation or any other theoretical analyses.

Contribution. Our contribution are succinctly outlined as:

1. We address the underlying fallacies of Kernelized Convex Clustering (KCC), where data points are projected to a Hilbert space $\mathcal{H}$, and subsequently, convex clustering is performed on the projected data points. We propose an alternate algorithm that leverages vanilla convex clustering itself to solve the problem effectively. The convexity property of the optimization leads to a unique minimizer, which we approximate after several iterations of our Alternating Direction Method of Multipliers (ADMM) based algorithm [Parikh and Boyd, 2014]. As an interesting consequence, this method naturally leads to an embedding in a finite lower-dimensional vector space, whose convex clustering turns out to be equivalent to the kernel convex clustering of the original data.
2. We study KCC from a theoretical aspect, establishing its convergence and providing finite sample bounds on the iterates and the ground truths. Further, the statistical properties of the finite-dimensional embedding are vividly discussed. This analysis provides certain interesting insights into its underlying structure and its relationship with the projected data points. This aids in identifying patterns that can enhance both the performance and interpretability of the model. We discuss some theoretical details in Section 3 and provide extensive derivations in Section C of the Supplementary document. Finally, we compare our method with state-of-the-art clustering algorithms and achieve impressive performance on benchmark datasets.

2 Proposed Method

The existing clustering algorithms, like k-means or convex clustering, are inefficient for clustering data points that are not linearly separable and contain non-convex patterns. The shortcomings can be alleviated by pursuing kernel methods that project the data points into a higher-dimensional Hilbert space, where data points are linearly separable. This fact motivates us to design a kernelized clustering algorithm to cluster intricately complex datasets.

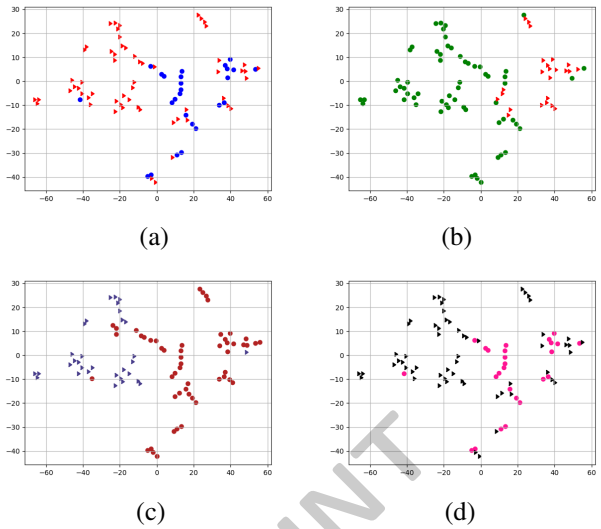


Figure 1: t-SNE plots of GLI85 dataset for (a) ground truth labels, (b) k-means clustering, (c) convex clustering, and (d) KCC are presented. Applying kernels improves performance over the Euclidean similarity measure.

2.1 A Motivating Example

We demonstrate our approach using the biological dataset, GLI85, which comprises 85 samples and 22283 continuous features. Initially, the dataset is pre-processed by standardizing the features. Refer to Figure 1a to observe the actual clusters present in GLI85. Figures 1b and 1c aptly demonstrate that the inefficiencies of k-means and convex clustering are due to their reliance on Euclidean-based similarity measures. Furthermore, we respectively obtain 0.051 and 0.206 as the NMI values, signifying the distortion of the cluster structure. In contrast, kernelized convex clustering captures the accurate cluster structures as evident in Figure 1d. In this context, we employed a Gaussian kernel in our implementation as $k(\boldsymbol{x}, \boldsymbol{y}) = e^{-\|\boldsymbol{x}-\boldsymbol{y}\|^2/2\sigma^2}$ where σ was chosen to be 0.001. The NMI [Vinh *et al.*, 2010] score was found to be 1 in this case, highlighting the utility of the kernels in the paradigm of convex clustering.

2.2 Problem Formulation

Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$ be n data points to be clustered. Let $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map that maps every data point $\boldsymbol{x}_i$ to $\phi(\boldsymbol{x}_i)$ in the Reproducing Kernel Hilbert Space, $\mathcal{H}$. Let $\boldsymbol{u}_i \in \mathcal{H}$ be the centroid corresponding to $\phi(\boldsymbol{x}_i)$. We propose to solve the following optimisation problem

$$\min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_n} \frac{1}{2}\sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i) - \boldsymbol{u}_i\|^2 + \gamma \sum_{i<j} w_{ij}\|\boldsymbol{u}_i - \boldsymbol{u}_j\| \quad (2)$$

Equation (2) has two separate summand terms. The first summation is a measure of the fit of the model: the smaller this term is, the closer the $\phi(\boldsymbol{x}_i)$'s are to their corresponding

centroids, $\boldsymbol{u}_i$'s, indicating a good fit of the model. The second term is a penalisation term, to keep the number of distinct centroids in check. The smaller this penalty term is, the fewer the number of distinct cluster centroids. Here γ is the tuning parameter for the fusion penalty term $\sum w_{ij}\|\boldsymbol{u}_i - \boldsymbol{u}_j\|$, while w_{ij}'s are non-negative weights for every pair of data points i and j. γ serves as a tradeoff between the model fit and the model complexity. The larger γ is, the more probable it is that the cluster centroids fuse to make the fusion penalty small, and thus minimise the entire objective. It is a good choice to select the weights in a way that depends on the proximity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Associated with the map ϕ is an inner product $\langle \cdot, \cdot \rangle$ of the Hilbert space $\mathcal{H}$, which satisfies all three properties of an inner product: symmetry, linearity, and positive-definiteness. Accordingly, we also have the kernel function, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ such that $k(x_i, x_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$, and the kernel matrix $\boldsymbol{K}$, whose $(i,j)^{\text{th}}$ entry is $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Define $\boldsymbol{\phi} = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n)]^\top$. Note that $\boldsymbol{K} = \boldsymbol{\phi}\boldsymbol{\phi}^\top$.

2.3 Towards Optimization

Fix $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in \mathcal{H}$. Decompose each $\boldsymbol{u}_i$ into the linear space, $V = \operatorname{span}\{\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n)\} \subseteq \mathcal{H}$ and its complement, $V^\perp$. Thus, for all $i = 1, \ldots, n$, $\exists\, \boldsymbol{\alpha}_i \in \mathbb{R}^n$ and $\boldsymbol{v}_i \in \boldsymbol{V}^\perp$, such that

$$\boldsymbol{u}_i = \boldsymbol{\phi}^\top \boldsymbol{\alpha}_i + \boldsymbol{v}_i$$

Now, observe that

$$\begin{aligned}\|\phi(\boldsymbol{x}_i) - \boldsymbol{u}_i\|^2 &= \|\phi(\boldsymbol{x}_i) - \boldsymbol{\phi}^\top \boldsymbol{\alpha_i} - \boldsymbol{v}_i\|^2 \\ &= \|\phi(\boldsymbol{x}_i) - \boldsymbol{\phi}^\top \boldsymbol{\alpha_i}\|^2 + \|\boldsymbol{v}_i\|^2 \\ &\geq \|\phi(\boldsymbol{x}_i) - \boldsymbol{\phi}^\top \boldsymbol{\alpha_i}\|^2\end{aligned}$$

In the second equality, there is no term of inner product because $\phi(\boldsymbol{x}_i) - \boldsymbol{\phi}^\top \boldsymbol{\alpha}_i$ and $\boldsymbol{v}_i$ are orthogonal. The inequality becomes an equality if and only if $\boldsymbol{v}_i = 0$. Similarly, for each of the different terms in the second summation,

$$\begin{aligned}\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 &= \|\boldsymbol{\phi}^\top(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j) + \boldsymbol{v}_i - \boldsymbol{v}_j\|^2 \\ &= \|\boldsymbol{\phi}^\top(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)\|^2 + \|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 \\ &\geq \|\boldsymbol{\phi}^\top(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)\|^2\end{aligned}$$

Thus,

$$\begin{aligned}&\|\boldsymbol{u}_i - \boldsymbol{u}_j\| \geq \|\boldsymbol{\phi}^\top(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)\| \\ \implies &\sum_{i<j} \|\boldsymbol{u}_i - \boldsymbol{u}_j\| \geq \sum_{i<j} \|\boldsymbol{\phi}^\top(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)\|\end{aligned}$$

Combining all these, we get the value of 2 at $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ is greater than or equal to at $\boldsymbol{\phi}^\top\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\phi}^\top\boldsymbol{\alpha}_n$ and equality holds if and only if $\boldsymbol{v}_1 = \cdots = \boldsymbol{v}_n = 0$. Hence, if $\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_n^*$'s are the minimisers in 2, their projections $\boldsymbol{v}_i^* \in V^\top$ must all equal the zero vector. Thus $\boldsymbol{u}_i^* = \boldsymbol{\phi}^\top\boldsymbol{\alpha}_i^*$ for some $\boldsymbol{\alpha}_i^* \in \mathbb{R}^n$. This observation turns out to be helpful, as we can just substitute $\boldsymbol{\phi}^\top\boldsymbol{\alpha}_i$ for every $\boldsymbol{u}_i$ in Equation 2 and try to minimise it with respect to $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$.

Remark 1. Although the above theorem is similar to the familiar representer theorem of Hilbert Spaces [Shalev-Shwartz and Ben-David, 2014], it is quite different from the problem statement in 2, and how it is derived. Firstly, representer theorem seeks to find minimiser of a single variable in $\mathcal{H}$; in this case we have n such unknowns, i.e. $\boldsymbol{u}_i$'s. More importantly, the representer theorem involves a penalty term which is strictly increasing in the norm of the unknown variable, contrary to our case where the penalty term involves the norm of the pairwise differences, $\|\boldsymbol{u}_i - \boldsymbol{u}_j\|$. Clearly, the penalty is not increasing in the norm of the unknown variables, $\|\boldsymbol{u}_i\|$'s.

2.4 A Perspective from Convex Clustering

Substituting $\boldsymbol{u}_i = \boldsymbol{\phi}^\top\boldsymbol{\alpha}_i$, and recalling that $\boldsymbol{K} = \boldsymbol{\phi}\boldsymbol{\phi}^\top, \phi(\boldsymbol{x}_i) = \boldsymbol{\phi}^\top\boldsymbol{e}_i$, we see

$$\begin{aligned}\|\phi(\boldsymbol{x}_i) - \boldsymbol{\phi}^\top\boldsymbol{\alpha}_i\|^2 &= (\boldsymbol{\alpha}_i - \boldsymbol{e}_i)^\top \boldsymbol{K}(\boldsymbol{\alpha}_i - \boldsymbol{e}_i) \\ \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 &= (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)^\top \boldsymbol{K}(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)\end{aligned}$$

The reformulated problem, being convex in $\boldsymbol{\alpha}_i$'s, can be solved by computational techniques like ADMM or AMA [Parikh and Boyd, 2014; Chi and Lange, 2015]. However, to avoid computational challenges, we resort to a different approach. If we decompose $\boldsymbol{K} = \boldsymbol{Z}^\top\boldsymbol{Z}$ using Cholesky decomposition, and make the following transformations:

$$\boldsymbol{z}_i = \boldsymbol{Z}\boldsymbol{e}_i, \quad \boldsymbol{a}_i = \boldsymbol{Z}\boldsymbol{\alpha}_i \tag{3}$$

We get a transformed objective function:

$$\min_{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n} \frac{1}{2}\sum_{i=1}^n \|\boldsymbol{z}_i - \boldsymbol{a}_i\|^2 + \gamma \sum_{i<j} w_{ij}\|\boldsymbol{a}_i - \boldsymbol{a}_j\| \tag{4}$$

which is the objective for the convex clustering of the n points, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ (1). Cholesky decomposition of the kernel matrix $\boldsymbol{K} = \boldsymbol{Z}^\top\boldsymbol{Z}$ aids us in reducing KCC to the well-known convex clustering problem. So, solving the kernel convex clustering problem in Equation 2 simultaneously leads to an embedding of the n points in $\mathbb{R}^n$, whose convex clustering is equivalent to KCC in 2.

2.5 Algorithms for Kernel Convex Clustering

[Chi and Lange, 2015] proposed two splitting methods for convex clustering, one using the Alternating Direction Method of Multipliers (ADMM) [Parikh and Boyd, 2014] and the other one using the Alternating Minimization Algorithm (AMA). Since ADMM converges under broader conditions than AMA (Section 4 of [Chi and Lange, 2015]), we have used the former one to get updates of the $\boldsymbol{a}_i$'s; then we revert the transformations in Equation (3) to get the solution of the $\boldsymbol{u}_i$'s. In ADMM, we introduce auxiliary variables, $\boldsymbol{v}_{ij} = \boldsymbol{a}_i - \boldsymbol{a}_j$, which act as constraints, when we rewrite 4 by replacing $\boldsymbol{a}_i - \boldsymbol{a}_j$ with $\boldsymbol{v}_{ij}$, and optimise it with respect to the $\boldsymbol{a}_i$'s and $\boldsymbol{v}_{ij}$'s. Additionally, we also introduce Lagrange multipliers $\boldsymbol{\eta}_{ij}$ corresponding to $\boldsymbol{v}_{ij}$, and a hyperparameter $\rho > 0$, which controls the effect of the quadratic penalty term $\sum_{i<j}\|\boldsymbol{v}_{ij} - \boldsymbol{a}_i + \boldsymbol{a}_j\|^2$ in the ADMM objective.

$$\boldsymbol{a}_i = \frac{\boldsymbol{z}_i + \sum_{j=1}^n(\boldsymbol{\eta}_{ij} + \rho\boldsymbol{v}_{ij}) - \sum_{j=1}^n(\boldsymbol{\eta}_{ji} + \rho\boldsymbol{v}_{ji})}{1 + n\rho} + \frac{\rho\sum \boldsymbol{z}_i}{1 + n\rho}$$

$$\boldsymbol{\eta}_{ij} = \boldsymbol{\eta}_{ij} + \rho(\boldsymbol{v}_{ij} - \boldsymbol{a}_i + \boldsymbol{a}_j)$$

$$\boldsymbol{v}_{ij} = \left(1 - \frac{\sigma_{ij}}{\|\boldsymbol{a}_i - \boldsymbol{a}_j - \eta_{ij}/\rho\|}\right)_+ (\boldsymbol{a}_i - \boldsymbol{a}_j - \boldsymbol{\eta}_{ij}/\rho)$$

Algorithm 1 KERNEL CONVEX CLUSTERING (KCC)

Require: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, $k(\cdot,\cdot)$, w_{ij}, $\rho, \gamma > 0$
Initialise $\boldsymbol{\alpha}_i = \boldsymbol{e}_i$ for all $i = 1, \ldots, n$
Initialise $\boldsymbol{\beta}_{ij} = \boldsymbol{e}_i - \boldsymbol{e}_j$ and $\boldsymbol{\lambda}_{ij}$ for all $i < j$ such that $w_{ij} > 0$
while does not converge **do**

$$\boldsymbol{\alpha}_i^{(m)} = \frac{\boldsymbol{e}_i + \boldsymbol{\tau}_f^{(m-1)} - \boldsymbol{\tau}_b^{(m-1)}}{1+n\rho} + \frac{\rho \sum_{i=1}^n \boldsymbol{e}_i}{1+n\rho}$$

$$\boldsymbol{\lambda}_{ij}^{(m)} = \boldsymbol{\lambda}_{ij}^{(m-1)} + \rho \boldsymbol{K}(\beta_{ij}^{(m-1)} - \boldsymbol{\alpha}_i^{(m)} + \boldsymbol{\alpha}_j^{(m)})$$

$$\boldsymbol{\beta}_{ij}^{(m)} = \left(1 - \frac{\sigma_{ij}}{\sqrt{\boldsymbol{t}_{ij}^{(m)\top} \boldsymbol{K} \boldsymbol{t}_{ij}^{(m)}}}\right)_+ (\boldsymbol{\alpha}_i^{(m)} - \boldsymbol{\alpha}_j^{(m)} - \boldsymbol{K}^{-1}\boldsymbol{\lambda}_{ij}^{(m)}/\rho)$$

$$\boldsymbol{\tau}_f^{(m)} = \sum_{j=1}^n (\boldsymbol{K}^{-1}\boldsymbol{\lambda}_{ij}^{(m)} + \rho\boldsymbol{\beta}_{ij}^{(m)})$$

$$\boldsymbol{\tau}_b^{(m)} = \sum_{j=1}^n (\boldsymbol{K}^{-1}\boldsymbol{\lambda}_{ji}^{(m)} + \rho\boldsymbol{\beta}_{ji}^{(m)})$$

end while

In the last equation, $\sigma_{ij} = \frac{\gamma w_{ij}}{\rho}$. Now note that $\boldsymbol{K} = \boldsymbol{Z}^\top \boldsymbol{Z}$ and $\boldsymbol{K}^{-1} = \boldsymbol{Z}^{-1}\boldsymbol{Z}^{\top -1}$. We could invert $\boldsymbol{Z}$, because almost surely the data to be clustered will come from a continuous distribution, making $\boldsymbol{K}$ non-singular. Letting $\boldsymbol{\lambda}_{ij} = \boldsymbol{Z}^\top \boldsymbol{\eta}_{ij}$, $\boldsymbol{v}_{ij} = \boldsymbol{Z}\boldsymbol{\beta}_{ij}$ and recalling that $\boldsymbol{a}_i = \boldsymbol{Z}\boldsymbol{\alpha}_i$, we can rewrite the updates for $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_{ij}$, $\boldsymbol{\lambda}_{ij}$. The required updates can be found in Algorithm 1.

Remark 2. We see that KCC of a dataset with kernel matrix $\boldsymbol{K}$, is equivalent to convex clustering of the embedded matrix $\boldsymbol{Z}$, which satisfies $\boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{K}$. We can choose $\boldsymbol{Z}$ in any way possible as long as it satisfies the above condition. Using Cholesky decomposition makes $\boldsymbol{Z}$ upper triangular, giving the embedding a redundant structure. However, not all embeddings may have a redundant structure. To see this, suppose $\boldsymbol{Z}$ is a suitable embedding. We select an orthogonal matrix $\boldsymbol{Q}$, so that $\boldsymbol{Q}\boldsymbol{Z}$ is neither upper nor lower triangular. Since $(\boldsymbol{Q}\boldsymbol{Z})^\top \boldsymbol{Q}\boldsymbol{Z} = \boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{K}$, $\boldsymbol{Q}\boldsymbol{Z}$ is also an embedding. This further demonstrates that the embedding is not unique. The number of embeddings is infinite, because of the possible infinite choices of the orthogonal matrix $\boldsymbol{Q}$.

After getting the embedding $\boldsymbol{Z}$, one can use any convex clustering method to get the $\boldsymbol{a}_i$'s. A similar AMA algorithm can be derived in a fashion similar to Algorithm 1, using the steps mentioned in [Chi and Lange, 2015]. Other notable methods to convex cluster $\boldsymbol{Z}$ include Cluster-path as mentioned in [Hocking *et al.*, 2011]. Since ADMM-based convex clustering converges, it also guarantees the convergence of KCC.

2.6 Getting the Optimal Number of Clusters

The final step involves determining the optimal number of clusters and the corresponding cluster assignments of the data points. This is carried out, first by applying agglomerative clustering on the centroids, followed by constructing a dendrogram. Now, for a given number of clusters k, the dendrogram is cut at a suitable height to obtain k clusters and get the respective labels. For this k, we compute the fit of the data using the standard k-means sum of squares formula: $SSE_k = \sum_{t=1}^k \sum_{i \in C_k} \left\| \hat{\boldsymbol{u}}_i - \frac{\sum_{j \in C_t} \hat{\boldsymbol{u}}_j}{|C_t|} \right\|^2$. After computing $\mathrm{SSE_k}$ for every k, we construct the elbow plot of SSE_k vs. k. We identify the elbow point as the point after which the change in $\mathrm{SSE_k}$ becomes small with respect to previous changes, thereafter. In other words, the graph continues to be approximately linear afterwards with the same slope for a long range of values. We also expect this slope not to be quite big. The value of k, corresponding to this elbow point, denotes the optimal number of clusters for the dataset.

2.7 Complexity Analysis

In KCC, the storage complexity is $O(n^2)$ for first storing the kernel matrix $\boldsymbol{K}$, and an additional $O(n^2)$ for storing the vectors $\boldsymbol{\alpha}_i$. So the total storage complexity in this case is $O(n^2)$. In comparison, kernel power k means (KPKM) [Paul *et al.*, 2023] has storage complexity $O(n^2)$, and that of biconvex clustering (BCC) [Chakraborty and Xu, 2021] is $O(np)$. In case of high-dimensional data, with $p \gg n$, KCC turns out to be better than biconvex clustering in terms of memory requirements. In terms of computational complexity, KCC takes $O(n^3)$ number of operations, KPKM takes $O(n^2k + npk)$, while BCC takes $O(n^2p)$. When comparing with KPKM, there is a tradeoff between cluster number and dimensionality, since we need to give the number of clusters k as input. Also, in both cases, the dimensionality plays a crucial role in the complexity. For high-dimensional datasets again with $p \gg n$, KCC overpowers BCC. For KPKM, although the complexity is lower than KCC, KCC predicts the actual number of clusters using the elbow plot. Thus in arbitrarily shaped datasets, KPKM may not give a proper clustering with a given k, but KCC automatically predicts the actual number of clusters.

3 Theoretical Properties

In this section, we will offer insights on the finite sample properties of the estimates and the consistency of the algorithm. Let $\boldsymbol{u}_i$ be the ground truth corresponding to the centroid of the i^{th} data point, and $\hat{\boldsymbol{u}}_i$ be the centroid estimates by minimising Equation 2. The square loss term, $\sum_{i=1}^n \|\phi(\boldsymbol{x}_i) - \boldsymbol{u}_i\|^2$, in 2, measures the fit of the data with the ground truths. The more close $\hat{\boldsymbol{u}}_i$ and $\boldsymbol{u}_i$ are, the more close the two quantities $\sum_{i=1}^n \|\phi(\boldsymbol{x}_i) - \boldsymbol{u}_i\|^2$ and $\sum_{i=1}^n \|\phi(\boldsymbol{x}_i) - \hat{\boldsymbol{u}}_i\|^2$ become, and the better is the fit of the data. So it makes sense to bound the norm $\sum_{i=1}^n \|\hat{\boldsymbol{u}}_i - \boldsymbol{u}_i\|^2$. We assume the following data generating process:

$$\phi(\boldsymbol{x}_i) = \boldsymbol{u}_i + \boldsymbol{\epsilon}_i$$

for all data points $i = 1, \ldots, n$. Here $\phi(\boldsymbol{x}_i)$, $\boldsymbol{u}_i$ and $\boldsymbol{\epsilon}_i$ are all assumed to lie in the Hilbert Space $\mathcal{H}$. $\boldsymbol{\epsilon}_i$ are mean zero sub-Gaussian random variables in $\mathcal{H}$ with respect to an operator Γ [Chen and Yang, 2020]. Further, assume that $\boldsymbol{\epsilon}_i$'s are independent, thus implying that $\mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j \rangle] = 0$ for distinct i and j. Also, suppose $\mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_i \rangle] = \sigma^2$. The sub-Gaussian model assumptions are fairly standard and are quite

analogous to the ones mentioned in [Tan and Witten, 2015] and [Wang *et al.*, 2018]. For practical purposes, we can assume that the error terms $\boldsymbol{\epsilon}_i$'s are bounded in $\mathcal{H}$, that is $\|\boldsymbol{\epsilon_i}\| \leq M$ almost surely, for some $M > 0$ and $i = 1, \ldots, n$. The bounded support assumption is standard in the theoretical analysis for centroid-based clustering [Paul *et al.*, 2021a; Pollard, 1981], [Paul *et al.*, 2021b]. This is surely the case if we take our kernel function $k(\boldsymbol{x}, \boldsymbol{y})$ to be the radial basis function, $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\big(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/2\sigma^2\big)$. Observe that,

$$\begin{aligned}&\|\phi(\boldsymbol{x}_i) - \boldsymbol{u}_i\|^2 - \|\phi(\boldsymbol{x}_i) - \hat{\boldsymbol{u}}_i\|^2\\ =& - \|\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i\|^2 - 2\langle \boldsymbol{\epsilon}_i, \boldsymbol{u}_i - \hat{\boldsymbol{u}}_i\rangle\end{aligned}$$

Further, (2) is minimized by $\hat{\boldsymbol{u}}_i$, and so the first quantity in the above string of equalities can be lower bounded by a difference of the penalty terms at $\hat{\boldsymbol{u}}_i$ and $\boldsymbol{u}_i$. Thus, it remains to bound only $\langle \boldsymbol{\epsilon}_i, \boldsymbol{u}_i - \hat{\boldsymbol{u}}_i\rangle$. We accomplish this by decomposing $\mathcal{H}$ into two mutually orthogonal spaces in $\mathcal{H}$ and analyzing the projections of the estimates in the orthogonal subspaces, separately. The details of our calculations are mentioned explicitly in the supplementary material. We summarise the statistical analysis in the following theorem.

Theorem 1. *Let $\phi(\boldsymbol{x}_i) = \boldsymbol{u}_i + \boldsymbol{\epsilon}_i$ for al $i = 1, \ldots, n$, where $\boldsymbol{\epsilon}_i$ are i.i.d. mean zero sub-Gaussian random variables in the RKHS $\mathcal{H}$, with respect to the operator Γ. Let $\hat{\boldsymbol{u}}_i$ be the solutions of 2. If $\gamma \geq \frac{2z_0}{w_{\min}}$, then*

$$\frac{1}{2n}\sum_{i=1}^{n}\|\hat{\boldsymbol{u}}_i - \boldsymbol{u}_i\|^2 \leq \frac{3\gamma}{2n}\sum_{i<j} w_{ij}\|\boldsymbol{u}_i - \boldsymbol{u}_j\| + \sigma^2\left[\frac{1}{n} + \sqrt{\frac{\log(n)}{n^2}}\right]$$

with probability at least $1 - \frac{2}{\binom{n}{2}} - 2\exp\left[-C\min\left(\frac{\sigma^4\log(n)}{L^4\|\Gamma\|_{\text{HS}}^2}, \frac{\sigma^2\sqrt{\log(n)}}{L^2\|\Gamma\|_{\text{op}}^2}\right)\right]$ *for some constant C.*

The above theorem is proved by using Hanson-Wright's Inequality [Chen and Yang, 2020] on each of the mutually orthogonal projections. The term $\boldsymbol{z}_0$ is explicitly described in the supplementary material; it is dependent on the data through σ and Σ, and is $O(\log n)$. Thus, the sub-Gaussianity assumption greatly reduces the data dependency of γ to just σ and Σ, to get the above finite sample bounds.

Remark 3. In Theorem 1, the fusion parameter is dependent on the value of $\boldsymbol{z}_0$, to attain this upper bound. Now, if $\|\boldsymbol{u}_i - \boldsymbol{u}_j\|$'s are uniformly bounded for all pairs $i \neq j$, and $\frac{\gamma}{n}\sum_{i<j} w_{ij} = o_p(1)$ as $n \to \infty$, then the right hand side of the inequality goes to zero, with the probability of the event going to 1. Thus, the average fit of the centroids goes to zero in such circumstances. However, $\gamma \geq 2z_0/w_{\min}$ as stated in Theorem 1. Thus, a necessary condition for $\frac{\gamma}{n}\sum_{i<j} w_{ij} = o_p(1)$ to hold is to have $z_0\sum_{i<j} w_{ij}/nw_{\min} \to 0$ as $n \to \infty$. From the definition of z_0, we see that it is at most of order $O(\log n)$. Notice that there are exactly $\binom{n}{2}$ possible w_{ij}'s. Suppose for some $0 < \alpha < \frac{1}{2}$, at most n^α of these w_{ij}'s are positive and the remaining ones are zero. Rigorously stating, the number of elements in the set $\{w_{ij} : w_{ij} > 0,\ i < j\}$ is less than or equal to n^α, $0 < \alpha < \frac{1}{2}$. Recall that $w_{\min} = \min\{w_{ij} : w_{ij} > 0,\ i < j\}$. Further, suppose $\frac{1}{\sqrt{n}} \leq w_{ij} \leq 1$ for all the weights lying in the aforementioned set. Then, $z_0\sum_{i<j} w_{ij}/nw_{\min} \leq z_0 n^\alpha/\sqrt{n} \leq 2\,O(\log n)\,/\,n^{\frac{1}{2}-\alpha} \to 0$. Thus Algorithm 1 is consistent if the above-mentioned conditions hold simultaneously.

Remark 3 gives us certain conditions on the weights w_{ij}, so that our estimated centroids converge to the ground truth in the case of a large number of samples. Specifically, we need to ensure that out of all the $\binom{n}{2}$ w_{ij}'s possible, at most $\sqrt{n}$ of them must be positive and the remaining should be set to zero. Further, the weights should be within the range of $\frac{1}{\sqrt{n}}$ and 1. This is practically possible, since w_{ij}'s are hyperparameters and can be tuned accordingly by the practitioner.

Remark 4. For the bounds stated in Theorem 1 to hold, the tuning parameter γ must be greater than $2z_0/w_{\min}$, where z_0 itself is a quantity dependent on $\mathcal{H}$. So the choice of the kernel space indeed does affect the quality of clustering. Compared to Lemma 7 of [Tan and Witten, 2015] , where $\gamma = \Omega\Big(\sqrt{\log pn^2/np}\Big)$, in our case, $\gamma = \Omega(\sqrt{\log n})$ for similar kinds of bound to hold.

4 Experiments

In this section, we establish KCC's superiority by applying it to several synthetic and real-world datasets and comparing it with various state-of-the-art methods. We use Normalized Mutual Information (NMI) [Vinh *et al.*, 2010] to evaluate the quality of clustering. The additional experimental results are demonstrated in Sections D and E of the Supplementary document.

4.1 Results on Synthetic Dataset

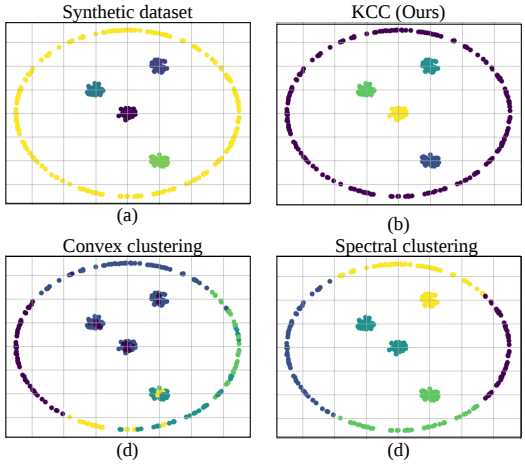


Figure 2: Scatter plots of the synthetic dataset for (a) ground truth labels, (b) KCC (Ours), (c) convex clustering, and (d) spectral clustering are illustrated.

We generate a simulated dataset of 400 data points in $\mathbb{R}^2$, as shown in Figure 2. The four central blobs each consist of 50

Datasets	KCC (Ours)	Convex	k-Means	Spectral	KPKM	BCC	TGCC	C-PAINT	SC-NR	#clusters
Lymphoma	**0.778**	0.718	0.654	0.179	0.633	0.450	0.727	0.689	0.429	7
Orlraws10P	**0.851**	0.821	0.798	0.209	0.810	0.720	0.720	0.759	0.645	11
Yale	**0.657**	0.293	0.480	0.601	0.568	0.288	0.549	0.617	0.477	14
Lung	**0.804**	0.729	0.594	0.018	0.433	0.328	0.769	0.682	0.321	4
Zoo	**0.736**	0.324	0.690	0.637	0.459	0.695	0.629	0.699	0.645	4
Housevotes	**0.573**	0.004	0.536	0.542	0.518	0.489	0.525	0.429	0.538	2
Glass	**0.439**	0.255	0.357	0.367	0.347	0.308	0.398	0.402	0.384	9
New Thyroid	**0.706**	0.491	0.553	0.491	0.376	0.407	0.568	0.467	0.303	5
Glioma	**0.529**	0.506	0.490	0.031	0.411	0.453	0.451	0.279	0.091	3
MNIST	**0.614**	0.062	0.553	0.047	0.486	0.421	0.329	0.496	0.425	10

Table 1: NMI scores of KCC and other clustering methods applied on different datasets.

Method	NMI
Kernel Convex Clustering	0.999
Convex Clustering	0.259
Biconvex clustering	0.721
Kernel Power k means	0.448
TGCC	0.784
C-PAINT	0.582
SC-NR	0.671
Spectral Clustering	0.598
k-means	0.457

Table 2: NMI values after applying different methods on the synthetic dataset.

points, while the outer circle comprises 200 points. For simulating each of the blobs, first we generate $\theta_i \overset{\text{i.i.d.}}{\sim} U(0, 2\pi)$, $R_i \overset{\text{i.i.d.}}{\sim} U(0, 0.45)$. Then we set $x_i = R_i \cos(\theta_i)$ and $y_i = R_i \sin(\theta_i)$. We accordingly shift the points to finally get 4 such blobs of size 50 each. Next, we again generate $\theta_i \overset{\text{i.i.d.}}{\sim} U(0, 2\pi)$, and set $x_i = 3\cos(\theta_i) + \epsilon_{i1}$ and $y_i = 3\sin(\theta_i) + \epsilon_{i1}$, where $\epsilon_{i2}, \epsilon_{i2} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.01)$.

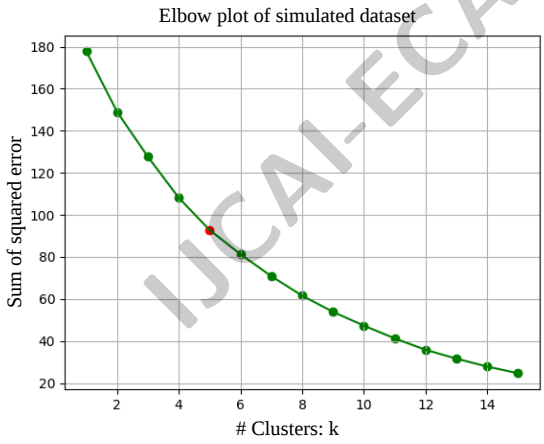


Figure 3: Elbow plot of synthetic dataset with $\sigma_1 = 1$, $\sigma_2 = 100$, $\gamma = 1$, $\rho = 0.001$. This set of values gives the optimal clustering with NMI of 1.

In this way, we get the outer circle. We use the Gaussian kernel as the feature map to project the 400 points in an RKHS $\mathcal{H}$, which is a popular choice for the feature map. The kernel function associated with it is $k(\boldsymbol{x}, \boldsymbol{y}) = e^{-\|\boldsymbol{x}-\boldsymbol{y}\|/2\sigma_1^2}$. The weights were chosen as follows: for every pair $i \neq j$, $w_{ij} = e^{-\|x_i - x_j\|^2/2\sigma_2^2}\, \mathbb{I}[x_j \text{ is one of the 6 nearest neighbours of } x_i]$. To make the weights symmetric, we finally chose $w^*_{ij} = (w_{ij} + w_{ji})/2$. ρ and γ were also chosen after proper tuning.

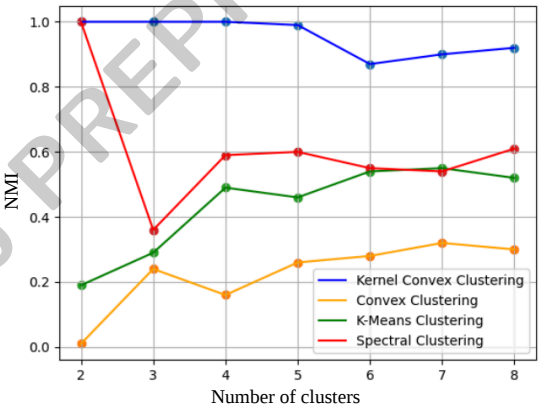
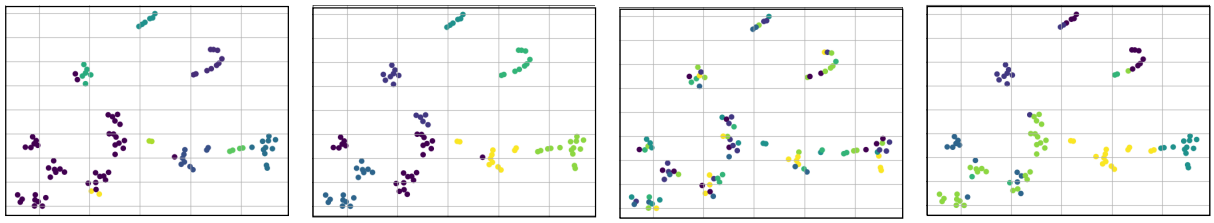


Figure 4: The impact on NMI with varying numbers of clusters is presented. Our method, KCC, performs consistently compared to other methods.

We further demonstrate the result of other competing methods, like convex clustering, spectral clustering, kernel k-means, kernel power k means [Paul *et al.*, 2023], biconvex clustering [Chakraborty and Xu, 2021], C-PAINT [Zhang *et al.*, 2021], SC-NR [Zhou *et al.*, 2023] and TGCC [Zhang and Terada, 2025]. The scatter plots corresponding to ground truths, KCC, convex clustering and spectral clustering are elucidated in Figure 2. The corresponding NMI values are also reported in Table 2, and the corresponding elbow plot is demonstrated in Figure 3.

4.2 Effect of Increasing Number of Clusters

We now check the efficacy of our method on an increasing number of clusters. The number of clusters varies from two to eight. We use the same kind of synthetic dataset as used in the previous experiment. For $k = 2$ clusters, we have the outer circle of 200 points and the central blob of 50 points. As k increases, blobs of 50 points are added one by one inside the interior of the outer circle. The blobs and the outer



(a) (b) (c) (d)

Figure 5: t-SNE plots of Lymphoma dataset for (a) ground truth labels, (b) KCC (Ours), (c) spectral clustering, and (d) k-means clustering are presented.

circle were generated in the manner described in subsection 4.1. On each of the datasets, we apply KCC and different clustering methods, and compare the results using the NMI score. We graphically summarise the effect of increasing the number of clusters on the NMI score in Figure 4. KCC turns out to be the best choice for clustering as k increases. The cluster predictions also turn out to be mostly true for KCC in comparison to other methods.

4.3 Case Study on Lymphoma Dataset

We assess our algorithm's performance by applying KCC on the Lymphoma microarray dataset [Li *et al.*, 2018]. It comprises 96 instances and 4026 features, all of which are discrete. In total, there were 9 classes, two or three of which had very few instances belonging to them. Since the variables were all discrete, we did not standardise the dataset. We used

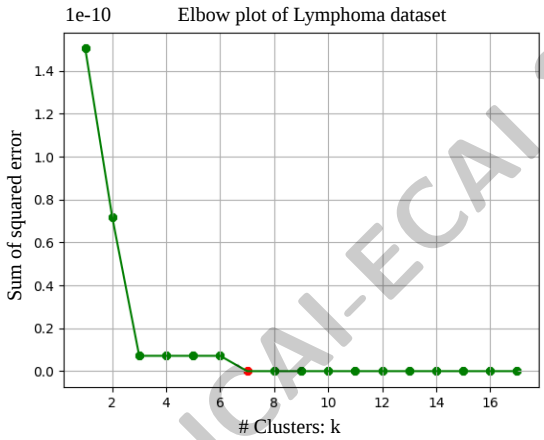


Figure 6: Elbow plot for the Lymphoma dataset. The optimal number of clusters is estimated to be 7. The dataset contains 9 annotated clusters, although several contain very few samples and are therefore merged by KCC.

the Gaussian kernel as the feature map. The weights were chosen similarly to what was done for the synthetic dataset. The Kernel bandwidth σ_1, ADMM convergence controlling variable ρ, fusion penalty γ, and σ_2, all were chosen appropriately after proper tuning. Kernel Convex Clustering was then applied on the datasets. Using agglomerative clustering and an elbow plot, we get that the optimal number of clusters in this case is 7, as shown in Figure 6. Originally, there were 9 clusters, but here we get 7, which does not seem to be a problem, as one or two clusters had just 3 to 4 points in it. So the clusters were merged to minimise the entire fit of the data. After getting the number of clusters, we get the corresponding cluster identities for all points, and then compute the NMI values for the Lymphoma by comparing the original and the experimental cluster identities. The NMI value reported in this case is 0.778. The NMI values for the other methods are given in Table 2. The comparative study of the t-SNE plots for the Lymphoma dataset is demonstrated in Figure 5.

4.4 Performance on Real Benchmarks

To further demonstrate the efficacy of KCC, we compare with the SOTA algorithms on 9 benchmark datasets. The datasets are curated from Keel repository [Alcala-Fdez *et al.*, 2010] and ASU feature selection repository [Li *et al.*, 2018]. Preprocessing is first performed on each dataset before running the KCC algorithm. For datasets with continuous covariates, we scale the data by centering each of the datasets and dividing by the corresponding variance. However, for datasets with categorical variables, centering is not performed, and the original dataset itself is used. We take the Gaussian kernel, which is a popular choice for machine learning practitioners. For MNIST, we randomly select 50 images from 10 classes and apply KCC on the overall 500 data points. There are four key hyperparameters, $\sigma_1, w_{ij}, \rho, \gamma$, which are tuned to get the u_i's corresponding to each point by running KCC. We construct the elbow plots and obtain the optimal number of clusters, K. Then, using cluster identities for all points, we compute the NMI values for the dataset by comparing the original and the experimental cluster identities. The performance of each algorithm is reported in Table 1, indicating the superior performance of KCC on real benchmarks.

5 Conclusion

We propose KCC, a convex clustering algorithm in kernelized Hilbert spaces for linearly inseparable data, guaranteeing convergence to a unique global optimum. A key insight is that solving KCC reduces to convex clustering of a finite-dimensional embedding, which we analyze theoretically through its large-sample properties. Empirical results confirm competitive performance over standard and recent methods. Future work includes multi-kernel extensions, feature weighting for interpretability, and deeper theoretical connections between infinite- and finite-dimensional embeddings.

References

[Alcala-Fdez *et al.*, 2010] Jesus Alcala-Fdez, Alberto Fernández, Julián Luengo, J. Derrac, S Garc'ia, Luciano Sanchez, and Francisco Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 01 2010.

[Chakraborty and Xu, 2021] Saptarshi Chakraborty and Jason Xu. Biconvex clustering, 2021.

[Chen and Yang, 2020] Xiaohui Chen and Yun Yang. Hanson-wright inequality in hilbert spaces with application to k-means clustering for non-euclidean data, 2020.

[Chi and Lange, 2015] Eric C. Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, October 2015.

[Chi and Steinerberger, 2018] Eric C. Chi and Stefan Steinerberger. Recovering trees with convex clustering, 2018.

[Chu *et al.*, 2021] Shuyu Chu, Huijing Jiang, Zhengliang Xue, and Xinwei Deng. Adaptive convex clustering of generalized linear models with application in purchase likelihood prediction. *Technometrics*, 63(2):171–183, 2021.

[Feng *et al.*, 2024] Qiying Feng, C. L. Philip Chen, and Licheng Liu. A review of convex clustering from multiple perspectives: Models, optimizations, statistical properties, applications, and connections. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13122–13142, 2024.

[Girolami, 2002] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[Hocking *et al.*, 2011] Toby Hocking, Jean-Philippe Vert, and Armand Joulin. Clusterpath an algorithm for clustering using convex fusion penalties. pages 745–752, 06 2011.

[Li *et al.*, 2018] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

[Li *et al.*, 2024] Ao Li, Cong Feng, Yuan Cheng, Yingtao Zhang, and Hailu Yang. Incomplete multiview subspace clustering based on multiple kernel low-redundant representation learning. *Information Fusion*, 103:102086, 2024.

[Lindsten *et al.*, 2011] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, 2011.

[Nagorski and Allen, 2016] John Nagorski and Genevera I. Allen. Genomic region detection via spatial convex clustering. *PLoS ONE*, 13, 2016.

[Parikh and Boyd, 2014] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 01 2014.

[Park and Park, 2025] Beomjin Park and Daun Park. Sparse kernel k-means clustering. *Journal of Applied Statistics*, 52(1):158–182, 2025.

[Paul *et al.*, 2021a] Debolina Paul, Saptarshi Chakraborty, and Swagatam Das. On the uniform concentration bounds and large sample properties of clustering with bregman divergences. *Stat*, 10(1):e360, 2021.

[Paul *et al.*, 2021b] Debolina Paul, Saptarshi Chakraborty, Swagatam Das, and Jason Xu. Uniform concentration bounds toward a unified framework for robust clustering. *Advances in Neural Information Processing Systems*, 34:8307–8319, 2021.

[Paul *et al.*, 2023] Debolina Paul, Saptarshi Chakraborty, Swagatam Das, and Jason Xu. Implicit annealing in kernel spaces: A strongly consistent clustering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5862–5871, 2023.

[Pollard, 1981] David Pollard. Strong consistency of k-means clustering. *The annals of statistics*, pages 135–140, 1981.

[Radchenko and Mukherjee, 2017] Peter Radchenko and Gourab Mukherjee. Convex clustering via ℓ_1 fusion penalization. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(5):1527–1546, 2017.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Tan and Witten, 2015] Kean Ming Tan and Daniela Witten. Statistical properties of convex clustering, 2015.

[Tropp, 2006] Joel Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52, 03 2006.

[Vinh *et al.*, 2010] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

[Wang *et al.*, 2018] Binhuan Wang, Yilong Zhang, Will Wei Sun, and Yixin Fang. Sparse convex clustering. *Journal of Computational and Graphical Statistics*, 27(2):393–403, April 2018.

[Wang *et al.*, 2024] Jun Wang, Zhenglai Li, Chang Tang, Suyuan Liu, Xinhang Wan, and Xinwang Liu. Multiple kernel clustering with adaptive multi-scale partition selection. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6641–6652, 2024.

[Weylandt and Michailidis, 2021] Michael Weylandt and George Michailidis. Automatic registration and clustering of time series. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5609–5613, 2021.

[Wu *et al.*, 2024] Tingting Wu, Songhe Feng, and Jiazheng Yuan. Low-rank kernel tensor learning for incomplete multi-view clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15952–15960, Mar. 2024.

[Zhang and Terada, 2025] Bingyuan Zhang and Yoshikazu Terada. Tree-guided l_1-convex clustering, 2025.

[Zhang *et al.*, 2021] Bingyuan Zhang, Jie Chen, and Yoshikazu Terada. Dynamic visualization for l1 fusion convex clustering in near-linear time. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 515–524. PMLR, 27–30 Jul 2021.

[Zhou *et al.*, 2023] Hao Zhou, Zekun Wang, Hongjia Chen, and Xiang Wang. A novel spectral clustering algorithm based on neighbor relation and gaussian kernel function with only one parameter, 03 2023.

[Zhu *et al.*, 2014] Changbo Zhu, Huan Xu, Chenlei Leng, and Shuicheng Yan. Convex optimization procedure for clustering: Theoretical revisit. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.