Information Theoretic Clustering via Divergence Maximization among Clusters

Sahil Garg[1,*] Mina Dalirrooyfard[1] Anderson Schneider[1] Yeshaya Adler[1] Yuriy Nevmyvaka[1] Yu Chen[1]
Fengpei Li[1] Guillermo Cecchi[2]

[1]Dept. of Machine Learning Research, Morgan Stanley, New York, New York, USA
[2]IBM T. J. Watson Research Center, Yorktown Heights, New York, USA,
[*] Corresponding Author: sahil.garg@morganstanley.com, sahil.garg.cs@gmail.com

Abstract

Information-theoretic clustering is one of the most promising and principled approaches to finding clusters with minimal apriori assumptions. The key criterion therein is to maximize the mutual information between the data points and their cluster labels. Such an approach, however, does not explicitly promote any type of inter-cluster behavior. We instead propose to maximize the Kullback–Leibler divergence between the underlying data distributions associated to clusters (referred to as *cluster distributions*). We show it to entail the mutual information criterion along with maximizing cross entropy between the cluster distributions. For practical efficiency, we propose to empirically estimate the objective of KL-D between clusters in its dual form leveraging deep neural nets as a dual function approximator. Remarkably, our theoretical analysis establishes that estimating the divergence measure in its dual form simplifies the problem of clustering to one of optimally finding $k-1$ *cut points for* k *clusters in the 1-D dual functional space*. Overall, our approach enables linear-time clustering algorithms with theoretical guarantees of near-optimality, owing to the submodularity of the objective. We show the empirical superiority of our approach w.r.t. current state-of-the-art methods on the challenging task of clustering noisy timeseries as observed in domains such as neuroscience, healthcare, financial markets, spatio-temporal environmental dynamics, etc.

1 INTRODUCTION

Clustering [Jain, 2010] is one of the most fundamental problems in machine learning. It is particularly challenging in domains such as neuroscience, healthcare, and finance, where data often consists of noisy timeseries of significant length. In neuroscience, the functional structure of neuronal ensemble activity is key to understanding how brain regions interact with each other [Siegle et al., 2021, de Vries et al., 2020]. In finance, systemically grouping assets that change in value together plays a critical role in portfolio optimization [Tola et al., 2008]. Clustering algorithms are typically employed for exploratory analysis. Therefore, they should ideally be sufficiently flexible and make minimal assumptions on the data [Ver Steeg et al., 2014]. They should not require that "prototypical" clusters be specified [Böhm et al., 2006], nor explicitly define notions of similarity between data points [Slonim et al., 2005]. With these considerations, one of the most promising and principled approaches is *information theoretic clustering* introduced by Gokcay and Principe [2002]. Owing to its theoretical appeal, it has been studied extensively [Sugar and James, 2003, Still and Bialek, 2004, Banerjee et al., 2005, Ver Steeg et al., 2014, Cicalese et al., 2019], primarily in the form where the objective is to maximize mutual information (**MI**) between data points and their cluster labels. Despite the popularity of this approach, it is noteworthy that such an objective characterizes only intra-cluster properties, i.e. minimizing entropy (variance) within each cluster, while inter-cluster properties are only implied from the former.

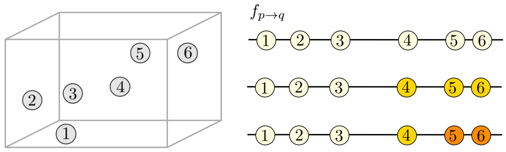


Figure 1: A high level toy example of our approach. On the l.h.s., we show 6 points in the original 3-dimensional space. These data points are represented in the 1-D dual space such that KL-divergence can be computed between any two subsets of data points (clusters) as a deterministic function of the representations. With the objective of maximizing the KL-divergence between clusters, we find clusters simply by greedy search of cut points in the dual space.

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:624–634.

As our *first contribution*, we argue that a fundamental criterion for clustering is that the distributions implied from any two clusters (of the same support) should have minimal overlap with each other, as quantified by the Kullback–Leibler Divergence (*KL-D*) [Cover, 1999]. Such a criterion is general enough to satisfy the desiderata of a "good" clustering solution, while simultaneously stating, explicitly, required inter-cluster behavior. Additional constraints, such as properties of data distributions within a cluster (e.g. low entropy of distribution) or continuity of manifold, are problem-specific, potentially impractical, and can be implied from the primary criterion itself.

While information theoretic clustering is theoretically appealing, it is nontrivial to estimate the required functions (including mutual information), and often intractable to optimize them w.r.t. cluster labels. Various ITC models have been explored in practice, including those based on k-nearest neighbors [Faivishevsky and Goldberger, 2010], minimal spanning trees [Müller et al., 2012], kernel functions, eigen-decomposition, max-k-cut, etc. [Davis and Dhillon, 2006, He et al., 2015, Böhm et al., 2006, Singh and Hooi, 2015, Wang and Sha, 2011, Sugiyama et al., 2014]. Yet, assuming a specific model of the data counteracts the theoretical appeal of the framework of being *model agnostic*. Further, these prior works rely on traditional (non-parametric) models which may be unsuitable for noisy, high-dimensional data which often arise in practice.

In light of the above, we *propose to estimate the divergence measure in its dual form* by Donsker and Varadhan [1975], employing deep neural nets as the dual function approximators [Belghazi et al., 2018]. Estimating divergence in its dual form circumvents the need to learn or characterize the cluster distributions and it suffices to have samples from those cluster distributions - i.e. data points belonging to their respective clusters. Moreover, neural networks lend expressiveness as universal approximators, and capacity to operate in high dimensional (noisy) settings.

Through *theoretical analysis* of the dual form of the proposed objective, we establish that clusters are *optimally contiguous in 1-D dual functional space*. This theoretical result is highly valuable, not only for its interpretability, but also because it simplifies the combinatorial problem of searching for optimal cluster labels to finding cut points in the 1-D dual space. Consequently, k clusters can be near-optimally identified by a *greedy search of* $k-1$ *cut points in the dual space*.

Contributions We make the following contributions to the information theoretic clustering literature: (i) While MI is known as the most principled objective for clustering from an information theoretic perspective, we show that the objective of KL-D is superior, entailing the former, and we advocate for it as the new fundamental criterion for optimization of cluster labels, as well as evaluation. (ii) Our theoretical analysis establishes that clusters are optimally contiguous in the dual function space of the objective thus simplifying the combinatorial optimization of cluster labels to one of finding cut points in the 1-D dual space. (iii) Owing to the submodularity of the proposed objective, we propose nearly-optimal greedy algorithms for finding $k-1$ cut points to obtain k clusters. (iv) We evaluate our approach for clustering noisy timeseries observed in domains like healthcare, finance, environmental dynamics, etc., along with a synthetic dataset, and demonstrate its competitiveness to the other information theoretic, traditional, and advanced deep learning methods for clustering. (v) Codebase at github.com/morganstanley/MSML/tree/main/papers.

Other Related Works Deep learning has been extensively applied to the problem of clustering [Min et al., 2018]. A common theme is to apply traditional clustering algorithms (e.g. K-Means, spectral, Gaussian mixture, subspace, nearest neighbors matching, etc.) in the latent representation space of an autoencoder [Ji et al., 2017, Law et al., 2017, Madiraju, 2018, Ma, 2019, Yang et al., 2019a, Bo et al., 2020, Dang et al., 2021]. Another paradigm for deep clustering is to minimize the KL-D between an auxiliary target distribution and the posterior distribution of the data represented by their cluster labels [Xie et al., 2016]. Aside from information theoretic clustering, information theory has been explored for representation learning including for the task of deep clustering such as based on variational autoencoders [Chen et al., 2016, Hu et al., 2017, Yang et al., 2019b, 2020, 2022, Ntelemis et al., 2021, Ahmadi et al., 2022]. However, these works differ from the niche field of so called "information theoretic clustering" for their primary criteria to cluster are not information theoretic. A KL-Divergence objective was previously considered with the highly restrictive assumption of clusters being Gaussian distributed [Das Gupta et al., 2015]; while Dhillon et al. [2003], used the KL-D between two words for the problem of clustering words, but not as a measure of divergence between clusters.

2 OUR APPROACH

Next, we discuss our approach of information theoretic clustering by maximizing KL-D between cluster distributions.

The standard objective in the information theoretic clustering literature is to maximize mutual information (*MI*) between data points ($\mathbf{X}$) and cluster labels (Y).

$$I(\mathbf{X}:Y) = \mathbb{E}_{(\mathbf{x},y)\sim(\boldsymbol{\mathcal{X}},\mathcal{Y})}\left[\log\frac{P(\mathbf{x},y)}{P(\mathbf{x})P(y)}\right]$$

Here, $\mathcal{X}$ is the high dimensional data distribution, and $\mathcal{Y}$ is the distribution of cluster labels; $\mathbf{X}$ and Y are the respective random variables. $I(\mathbf{X}:Y)$ is the MI function, a fundamental measure of dependence. Typically, the MI function is expressed in terms of the conditional entropy function:

$$\operatorname*{argmax}_{Y} I(\mathbf{X}:Y) = \operatorname*{argmax}_{Y} H(\mathbf{X}) - H(\mathbf{X}|\,Y) \qquad (1)$$

Since entropy of $\mathbf{X}$, $H(\mathbf{X})$, is a constant, the problem of maximizing MI is tantamount to minimizing conditional entropy of data points $\mathbf{X}$ given the clusters Y. This objective accounts for intra-cluster characteristics explicitly and *inter-cluster characteristics are only implied* from the former. To further elucidate, we consider a two-cluster problem with cluster labels $Y=0$ and $Y=1$. Let $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$ denote conditional random variables given the cluster labels, and corresponding cluster distributions, $\mathcal{X}_{y0}$ and $\mathcal{X}_{y1}$, both with support $\mathcal{X}$. We note that the conditional entropy explicitly decomposes into intra-cluster entropy terms.

$$\operatorname*{argmin}_{Y} H(\mathbf{X}|\,Y) = \operatorname*{argmin}_{Y} P_{y0}H(\mathbf{X}_{y0}) + P_{y1}H(\mathbf{X}_{y1})$$

Here, P_{y0} is shorthand for $P(Y=0)$. Similarly, P_{y1} denotes $P(Y=1)$. Thus, maximizing the mutual information is equivalent to minimizing entropy of both the cluster distributions. Considering the limitations of MI criterion in capturing inter-cluster characteristics, we instead propose an objective of maximizing KL-D between the cluster distributions, $\mathcal{X}_{|Y=0}$ and $\mathcal{X}_{|Y=1}$, as below.

$$\begin{aligned} &\operatorname*{argmax}_{Y} D(\mathcal{X}_{y0}\|\mathcal{X}_{y1}) + D(\mathcal{X}_{y1}\|\mathcal{X}_{y0}) \qquad (2)\\ &= \operatorname*{argmin}_{Y} \underbrace{H(\mathbf{X}_{y0}) + H(\mathbf{X}_{y1})}_{\text{intra-cluster}} - \underbrace{H_{\mathbf{X}_{y0}}(\mathbf{X}_{y1}) - H_{\mathbf{X}_{y1}}(\mathbf{X}_{y0})}_{\text{inter-cluster}} \end{aligned}$$

Clearly, the proposed objective minimizes entropy of cluster distributions, while maximizing cross entropy between the distributions, i.e. minimizing their overlap. For the problem of k-clusters, a variety of simple extensions are applicable, which we discuss later in this section. It is interesting to note that cross entropy is a pure and fundamental (directed) measure of non-overlap between two distributions whereas KL-D also accounts for entropy of one of the distributions itself. This insight is important in establishing that the proposed KL-D objective *entails* the MI function.

Theorem 1 *Let $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$ be the conditional random variables associated with the conditional distributions of data points, $\mathcal{X}_{y0}$ and $\mathcal{X}_{y1}$, given cluster labels $Y=0$ and $Y=1$ respectively. Optimizing the two clusters such that KL-Divergence between the two distributions is maximized,*

$$\text{argmax}_Y\; P_{y0}D(\mathcal{X}_{y0}\|\mathcal{X}_{y1}) + P_{y1}D(\mathcal{X}_{y1}\|\mathcal{X}_{y0}),$$

is equivalent to,

$$\text{argmax}_Y I(\mathbf{X}:Y)\; + P_{y1}H_{\mathbf{X}_{y1}}(\mathbf{X}_{y0}) + P_{y0}H_{\mathbf{X}_{y0}}(\mathbf{X}_{y1}),$$

where, $I(\mathbf{X}:Y)$ is mutual information function, and $H_{\mathbf{X}_{y1}}(\mathbf{X}_{y0})$ and $H_{\mathbf{X}_{y0}}(\mathbf{X}_{y1})$ are cross entropy functions.

Note, the above theoretic result depends upon P_{y0} and P_{y1} for establishing the equivalence. In practice, assuming a prior of clusters of equal sizes, we propose to simply maximize the objective, $D(\mathcal{X}_{y0}\|\mathcal{X}_{y1}) + D(\mathcal{X}_{y1}\|\mathcal{X}_{y0})$.

Although the objective of maximizing KL-D between cluster distributions is fundamental and intuitive, doing so w.r.t. cluster labels and input samples is not straightforward. To understand this challenge, we present KL-D between the two cluster distributions in the standard expression below.

$$D(\mathcal{X}_{y0}\|\mathcal{X}_{y1}) = \mathbb{E}_{\mathbf{X}\sim\mathcal{X}_{y0}} \log \frac{P(\mathbf{X}|Y=0)}{P(\mathbf{X}|Y=1)} \qquad (3)$$

To estimate the KL-D objective from the above expression, one needs to obtain the conditional densities, $P(\mathbf{X}|Y=0)$ and $P(\mathbf{X}|Y=1)$, for the respective (unknown) cluster distributions $\mathcal{X}_{y0}$ and $\mathcal{X}_{y1}$, even if the expectation is computed empirically from the data in cluster $Y=0$ as the empirical realization of $\mathcal{X}_{y0}$. We do not have these densities available, and it may be impossible to learn the densities from samples coming from the clusters, due to limited data, small clusters, high dimensionality, or noise that is prevalent in neural or financial timeseries data, etc. There are also practical challenges in estimating the KL-Divergence objective above, such as the function being unbounded in its value, variance of the empirical estimate, compute cost, vulnerability of nonparameteric kNN based KL-D estimators to noise, etc. One practical solution, which we propose next, is to empirically estimate the KL-D objective in its dual form by Donsker and Varadhan [1975], leveraging deep learning as the dual function approximator.

We argue that estimating the KL-D function in its dual form, as shown below, is particularly suitable for its use as the clustering objective.

$$D(\mathcal{X}_{y0}\|\mathcal{X}_{y1}) = \max_{f(.)\in L^{\infty}(\mathcal{X})} \mathbb{E}_{\mathcal{X}_{y0}} f(\mathbf{x}) - \log \mathbb{E}_{\mathcal{X}_{y1}} e^{f(\mathbf{x})}$$

Here, $f:\mathcal{X}\rightarrow\mathbb{R}$, is any function from the space of locally ∞-integrable functions such that expectations in the expression are finite, referred as the *dual function*. To estimate the dual form of KL-D, we only need samples from cluster distributions and not actual density functions. A cluster distributions' existence is only implied by data points in clusters of the same support. Thus, a cluster is the optimal empirical realization of the cluster distribution, and both expectations in the dual form are empirically computable from the respective clusters only.

$$\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1}) = \max_{\hat{f}(.)\in\mathcal{H}} \sum_{\mathbf{x}_{y0}\in\mathbf{X}_{y0}} \frac{\hat{f}(\mathbf{x})}{n_{y0}} - \log \sum_{\mathbf{x}_{y1}\in\mathbf{X}_{y1}} \frac{e^{\hat{f}(\mathbf{x})}}{n_{y1}}$$

Here, $\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})$ is an empirical estimate of $D(\mathcal{X}_{y0}\|\mathcal{X}_{y1})$ from clusters, $\mathbf{X}_{y0} = \{\mathbf{x}_i : y_i = 0\}_{i=1}^{n}$ and $\mathbf{X}_{y1} = \{\mathbf{x}_i : y_i = 1\}_{i=1}^{n}$; n_{y0} and n_{y1} are the respective cluster sizes. As mentioned above, since the cluster distributions, $\mathcal{X}_{y0}$ and $\mathcal{X}_{y1}$, are themselves defined from their respective clusters, $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$, it is only

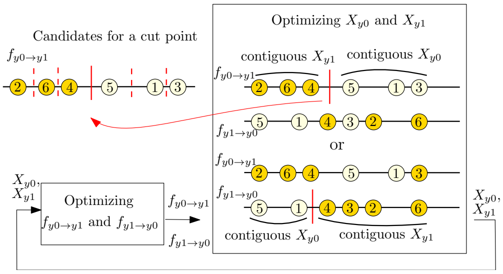


Figure 2: 2-clustering algorithm. First we optimize $f_{y0\to y1}$ and $f_{y1\to y0}$, next we optimize clusters X_{y0} and X_{y1}. Note that in the second step we only consider clusters X_{y0} and X_{y1} such that their representations are contiguous in one of $f_{y0\to y1}$ or $f_{y1\to y0}$. One example is shown for each case.

$\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})$ that is of interest as the clustering objective whereas $D(\mathcal{X}_{y0}\|\mathcal{X}_{y1})$ is notional. Correspondingly, $\hat{f}(.)$ is the dual function for estimating $\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})$, where the maximization is over a fixed class of functions $\mathcal{H}$.

As proposed in Belghazi et al. [2018], $f(.)$ can be a deep neural net function; for instance, neural timeseries models like LSTMs, RNNs, Transformers, TCNs, NBeats, etc. [Bai et al., 2018, Oreshkin et al., 2019, Kitaev et al., 2019, Benidis et al., 2020, Zeng et al., 2021, Fan et al., 2021, Gu et al., 2021, Challu et al., 2022], are all relevant for clustering of time series in this framework. To learn a stable neural dual function and avoid high variance in estimating the objective [Song and Ermon, 2019], practical tricks, such as early stopping, large batch size, low learning rate, etc., are well known. Furthermore, our goal is not to estimate the divergence measure exactly, but to find assignments that maximize divergence across clusters. It is the relative estimate of divergence across different sets of cluster assignments that matters. Besides, to avoid potential numerical instability from $\log$ *sum* $\exp$ function (smooth max), we propose a practical trick of using $\max$ function as a known approximation of the former Boyd et al. [2004].

$$\begin{aligned}&\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})\\ &\approx \max_{\hat{f}(.)} \sum_{\mathbf{x}_{y0}\in\mathbf{X}_{y0}} \hat{f}(\mathbf{x}_{y0}) - \max_{\mathbf{x}_{y1}\in\mathbf{X}_{y1}} \hat{f}(\mathbf{x}_{y1}) + \log|n_{y1}|\end{aligned}$$

Here, note that the $\max$ function is not sensitive to outliers in cluster $\mathbf{X}_{y1}$ since that term is being minimized. This not only stabilizes the optimization, but gives a nice interpretation for the expression in the context of clustering. From a deep learning perspective, the $\max$ function is essentially a $\max$ *pooling* operation, over the dual function outputs of the data points in $\mathbf{X}_{y1}$.

The overall expression for optimizing cluster labels is:

$$\begin{aligned}&\operatorname*{argmax}_{\mathbf{y}} \log(n_{y0}n_{y1}) \qquad (4)\\ &+ \max_{f_{y0\to y1}} \sum_{\mathbf{x}_{y0}\in\mathbf{X}_{y0}} \frac{f_{y0\to y1}(\mathbf{x}_{y0})}{n_{y0}} - \log \sum_{\mathbf{x}_{y1}\in\mathbf{X}_{y1}} e^{f_{y0\to y1}(\mathbf{x}_{y1})}\\ &+ \max_{f_{y1\to y0}} \sum_{\mathbf{x}_{y1}\in\mathbf{X}_{y1}} \frac{\hat{f}_{y1\to y0}(\mathbf{x}_{y1})}{n_{y1}} - \log \sum_{\mathbf{x}_{y0}\in\mathbf{X}_{y0}} e^{f_{y1\to y0}(\mathbf{x}_{y0})}\end{aligned}$$

Here, $f_{y0\to y1}(.)$ and $f_{y1\to y0}(.)$ are the dual functions corresponding to estimating KL-D in both directions, $\hat{D}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})$ and $\hat{D}(\mathbf{X}_{y1}\|\mathbf{X}_{y0})$. Note, $log|n_{y0}n_{y1}|$ naturally encourages balanced clusters. Next, we establish that the optimization in Eq. 4 has a solution which uniquely recovers the two clusters.

Theorem 2 *The optimal solution for the objective in Eq. 4 exists when f (for both $f_{y0\to y1}$ and $f_{y1\to y0}$) is continuous and bounded between $[a,b]$ for some $a \le b$. Moreover, if f is also L-Lipschitz, then for two clusters where the distance between cluster (defined as the minimum between points in separate cluster) is more than the distance within cluster (defined as the maximum distance between points in the same cluster), there exists some Lipschitz constant L where the optimal solution in Eq. 4 uniquely recovers the clusters.*

Clusters are optimally contiguous in the dual space Our *key observation* about the optimization problem (Eq. 4) which enables highly efficient and near-optimal algorithms for clustering, is that clusters are *optimally contiguous* in the space of dual functions ($f_{y0\to y1}$ and $f_{y1\to y0}$) i.e. *dual space*, as we theoretically prove in the following. This simplifies the combinatorial-optimization of finding clusters to that of finding a cut point in the dual space. Theorem 3 proves the contiguity of two clusters in the dual space.

Theorem 3 *Consider a dual function $\hat{f}(.)$, and the associated representation of data points in the dual space, $\{\hat{f}(\mathbf{x}_i)\}_{i=1}^n$, and the KL-D estimate between clusters $\mathbf{X}_{y0} = \{\mathbf{x}_i : y_i = 0\}_{i=1}^n$, $\mathbf{X}_{y1} = \{\mathbf{x}_i : y_i = 1\}_{i=1}^n$ is*

$$\hat{D}_{\hat{f}}(\mathbf{X}_{y0}\|\mathbf{X}_{y1}) = \sum_{\mathbf{x}_{y0}\in\mathbf{X}_{y0}} \frac{\hat{f}(\mathbf{x})}{n_{y0}} - \log \sum_{\mathbf{x}_{y1}\in\mathbf{X}_{y1}} e^{\hat{f}(\mathbf{x})} + \log(n_{y1}).$$

Then the clusters that maximize $\hat{D}_{\hat{f}}(\mathbf{X}_{y0}\|\mathbf{X}_{y1})$, i.e.

$$\operatorname*{argmax}_{\mathbf{y}} \hat{D}_{\hat{f}}(\mathbf{X}_{y0}\|\mathbf{X}_{y1}),$$

are contiguous in the dual space: there is a cut point c such that $\mathbf{x}_i \in \mathbf{X}_{y0}$ if $\hat{f}(\mathbf{x}_i) \ge c$ and $\mathbf{x}_i \in \mathbf{X}_{y1}$ if $\hat{f}(\mathbf{x}_i) < c$.

Cut Point Algorithm for Clustering This theoretical result on the contiguity of clusters in the dual space naturally leads to a cut-point based clustering algorithm as

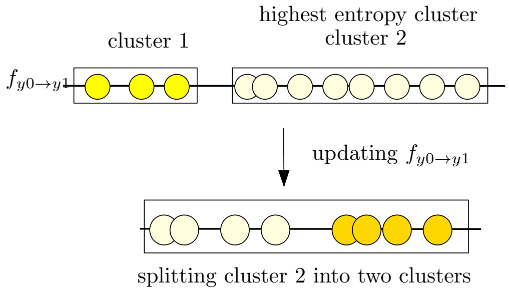


Figure 3: *Greedy bisection of the highest entropy cluster.* First we find two clusters using our 2 cluster algorithm, then we take the cluster with the highest entropy which is cluster 2, we update $f_{y0\to y1}$ and $f_{y1\to y0}$ and then split cluster 2 using our 2 clustering algorithm. Note that we are not showing cluster 1 in the updated $f_{y0\to y1}$ representation.

illustrated in Fig. 2. Here, we consider KL-D in both directions. We start by random clusters $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$ and optimize $f_{y0\to y1}(.)$ and $f_{y1\to y0}(.)$ with respect to these clusters. Then, in order to optimize clusters $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$, we only consider cluster pairs $(\mathbf{X}_{y0}, \mathbf{X}_{y1})$ such that both $\mathbf{X}_{y0}$ and $\mathbf{X}_{y1}$ are "contiguous" in one of the representations defined by dual functions, $f_{y0\to y1}(.)$ and $f_{y1\to y0}(.)$.

More formally, we first consider the one dimensional space defined by $f_{y0\to y1}(.)$. We sort the indices of the corresponding data points with respect to the values of the function $f_{y0\to y1}(.)$. For each pair of clusters defined by a cut point i, we evaluate the divergence objective. For a given cut point, it is computed from mean and (smooth) max statistics, values for all the cut points can be computed iteratively in $f_{y0\to y1}$, with linear time compute complexity. We do the same for the one dimensional space defined by $f_{y1\to y0}$ and output the cluster pair that maximizes the divergence from either of the two dimensions. We continue optimizing for a fixed number of iterations (100) or until convergence of the cluster labels.

When employing DNN as a dual function, the step of optimizing the dual functions given cluster labels as shown in Fig. 2 is a single iteration of updating weights of the corresponding two DNNs via backpropagation, rather than retraining from scratch for a change in cluster labels. Both updating the dual function for a change in cluster labels, and optimizing cluster labels by finding cut points in the re-optimized dual functional spaces, are highly efficient. Moreover, for the first few *warmup* iterations (10), we only update weights of the neural estimators and not the cluster labels. In our experiments, it takes only a few seconds to run the entire procedure to obtain cluster in a dataset of few thousand timeseries.

For the *k-clusters* problem, we want to maximize divergence between each pair of clusters, or maximize divergence of each cluster w.r.t. the rest. While in theory, there should be a different estimator for each pair of clusters, it suffices in practice to produce two estimates $\hat{D}(\mathbf{X}_{yi}\|\mathbf{X}_{yj})$ and $\hat{D}(\mathbf{X}_{yj}\|\mathbf{X}_{yi})$ respectively from $\hat{f}_{y0\to y1}$ and $\hat{f}_{y1\to y0}$, as in the two-cluster problem. For training any of the two estimators, in each batch update, two out of k clusters are randomly sampled, for which the estimator learns to maximize the estimate of KL-D. To optimize cluster labels, we propose a greedy search for $k-1$ cut points in the dual space, in which various variants of the KL-D objective are applicable. Next, we establish contiguity of k clusters in the dual space for one such objective.

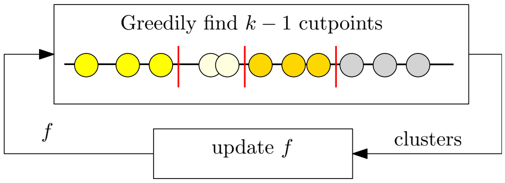


Figure 4: *Greedy cuts.* For a randomly initialized dual function, $f(.)$, data points are represented in the dual space, $\{f(\mathbf{x}_i)\}_{i=1}^n$, and sorted accordingly. The $k-1$ cut points are searched greedily to obtain k clusters, with each data point being a candidate as a cut point. Cut points and the dual function are updated iteratively.

Theorem 4 *Consider a dual function $\hat{f}(.)$ and the associated representation of input points in the dual space, $\{\hat{f}(\mathbf{x}_i)\}_{i=1}^n$. Let $\mathbf{X}^*_{y1} = \{\mathbf{x}_i : y^*_i = 1\}_{i=1}^n, \dots, \mathbf{X}^*_{yk} = \{\mathbf{x}_i : y^*_i = k\}_{i=1}^n$ be the clusters that maximize the objective,*

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{k-1} \hat{D}_{\hat{f}}(\mathbf{X}_{y>i}\|\mathbf{X}_{yi});$$

$$\hat{D}_{\hat{f}}(\mathbf{X}_{y>i}\|\mathbf{X}_{yi}) = \mathbb{E}_{\mathbf{x}\in\mathbf{X}_{y>i}}\hat{f}(\mathbf{x}) - \log \mathbb{E}_{\mathbf{x}_{yi}\in\mathbf{X}_{yi}} e^{\hat{f}(\mathbf{x}_{yi})},$$

where $\mathbf{X}_{y>i}$ is the set of all data points which lie on the r.h.s. of $\mathbf{X}_{yi}$ in the dual space. Then $\mathbf{X}^*_{y1}, \dots, \mathbf{X}^*_{yk}$ form contiguous clusters in the dual space.

We now propose two intuitive, greedy algorithms for finding $k-1$ cut points in dual space.

Greedy bisection of the entropy cluster First, as illustrated in Fig. 3, we use a recursive bisection approach. In each greedy iteration we pick a cluster with the highest entropy (cluster size being a good proxy for it if cluster sizes are non-uniform), and bisect it using the two-clusters algorithm (Fig. 2). As described previously, the two dual functions, $\hat{f}_{y0\to y1}$ and $\hat{f}_{y1\to y0}$, are updated considering all the clusters learned so far, maximizing the estimate of KL-D for each pair of clusters (old or new) in a batch update. We evaluate this algorithm extensively in our experiments.

Greedy cuts Another approach, as illustrated in Fig. 4, is to find $k-1$ greedy cuts in the dual space. Updating the dual function and the cuts is done iteratively. While the algorithm is generally applicable for many possible objectives based on KL-D between clusters, we prove theoretical guarantees for one such objective owing to its submodularity.

Theorem 5 *Consider a dual function $\hat{f}(.)$ and the associated representation of input data points in the dual space, $\{\hat{f}(\mathbf{x}_i)\}_{i=1}^{n}$. Let OPT be the optimal value of the objective,*

$$\operatorname*{argmax}_{\mathbf{y}} \sum_{i=1}^{k-1} \hat{D}_{\hat{f}}(\mathbf{X}_{y>i} \| \mathbf{X}_{yi});$$

$$\hat{D}_{\hat{f}}(\mathbf{X}_{y>i} \| \mathbf{X}_{yi}) = \mathbb{E}_{\mathbf{x} \in \mathbf{X}_{y>i}} \hat{f}(\mathbf{x}) - \log \mathbb{E}_{\mathbf{x}_{yi} \in \mathbf{X}_{yi}} e^{\hat{f}(\mathbf{x}_{yi})},$$

where $\mathbf{X}_{y>i}$ is the set of all data points which lie on the r.h.s. of $\mathbf{X}_{yi}$ in the dual space. Optimizing k-1 cuts greedily in the dual space finds clusters $\mathbf{X}^*_{y1}, \ldots, \mathbf{X}^*_{yk}$ such that,

$$\sum_{i=1}^{k-1} \hat{D}_{\hat{f}}(\mathbf{X}^*_{y>i} \| \mathbf{X}^*_{yi}) \geq \frac{e-1}{e} OPT.$$

In practice, there is space to explore various algorithms for finding the cut points while greedy algorithms as proposed above enjoy theoretical guarantees.

3 EMPIRICAL EVALUATION

One of the best-motivated application of (especially information theoretic) clustering algorithms, is clustering (noisy) timeseries in domains such as neuroscience, healthcare, finance, environmental dynamics, etc. For instance, in neuroscience, it is of substantial interest to find a subset of neurons in which neural activity exhibits high dependence (MI) w.r.t. each other.

Datasets We evaluate our approach on the following timeseries datasets: (i) electrophysiological Neuropixels, (ii) US stock returns, (iii) EEG, (iv) ECG, (v) Rain, (vi) Wind, (vii) Pollution, and four representative UCR datasets, (viii) UCR-Mallat, (ix) UCR-Trace, (x) UCR-Small Kitchen Appliances, (xi) UCR-ECG-Torso, and (xii) a synthetic timeseries dataset. See the supplement for more details.

Competitive Methods We compare our information theoretic clustering approach of divergence maximization (referred as "**ITC-DM***") w.r.t. the traditional baseline models, "KMeans", "Spectral" clustering, "kShape" clustering [Paparrizos and Gravano, 2015]. We use two important baseline estimators of MI based ITC: (i) a kNN based nonparameteric estimator, referred to as "ITC-kNN" [Faivishevsky and Goldberger, 2010], and (ii) a minimum spanning trees estimator, referred as "ITC-MST" [Müller et al., 2012]. We also evaluate various deep learning baselines: DEC [Xie et al., 2016], NNM [Dang et al., 2021], include temporal clustering models, DTC [Sai Madiraju et al., 2018], and DTCR [Ma et al., 2019].

Hyperparameters Selection Our task is to obtain the best possible clusters within an input dataset in an unsupervised setting. The deep learning optimization of estimating and maximizing KL-D w.r.t. cluster labels is unique to every single input of a dataset. Hyperparameters can be chosen independently for a given input of dataset by maximizing the proposed objective itself. We consider it valuable if some hyper-parameter choices perform well across all the datasets, to avoid the overhead of tuning as discussed next. Across all 12 datasets, we use the entire input dataset for rather than batch sampling. This is aligned with previous works on dual divergence estimation Belghazi et al. [2018], Song and Ermon [2019] which suggest to use a large batch size to avoid high variance. We chose LSTMs with one hidden layer of 32 units with a learning rate of 1e-1, weight parameters initialized with std of 0.1. We perform 100 iterations in the greedy algorihtm, with 10 warmup iterations, to update the dual function and not optimize the cut points (clusters labels). We use the greedy bisection algorithm for the primary analysis (Fig. 3). These choices were made via preliminary clustering analysis on a stock price dataset independently of the datasets in this paper. In Sec. 3.1.3, we present an extensive ablation study on the Neuropixels dataset, varying each hyperparameter from the above defaults. For the baseline clustering methods, we follow the respective strategies for selecting the hyperparameters as described in their papers or codebases.

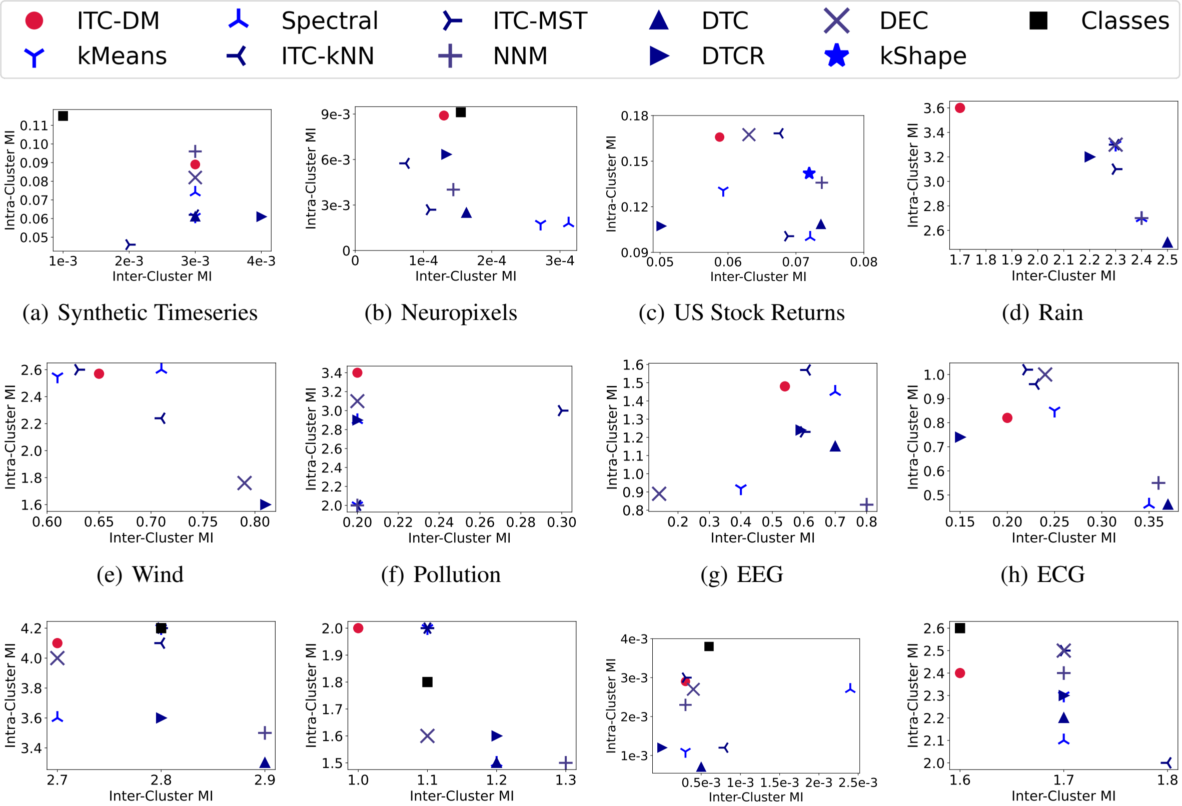
3.1 EVALUATION RESULTS

Next, we present our extensive empirical results on many real world datasets along with a synthetic timeseries dataset.

3.1.1 How to evaluate clusters of timeseries?

As the science of clustering objectives and algorithms advance in consideration of challenges presented by high dimensional noisy datasets in the modern times, we must further the science of evaluation metrics as well. Next, we discuss two evaluation metrics which we deem to be the most appropriate in terms of being fundamental, robust to noise, and can be estimated reliably.

Pairwise mutual information between timeseries We propose to evaluate clusters in terms of pairwise MI between timeseries within- and across- clusters. This is independent of the clustering objective and simple to compute. We treat timeseries observations as I.I.D. samples from a univariate random variable ignoring the temporal correlations; we employ a kNN based estimator ($k = 3$). Cluster level statistics of intra- and inter-cluster MI are obtained from the pairwise MI function by taking averages and normalizing it using the respective cluster sizes. Note, this MI function is *not to be confused* with the clustering criterion of mutual information between high dimensional data points (timeseries) and



(a) Synthetic Timeseries (b) Neuropixels (c) US Stock Returns (d) Rain

(e) Wind (f) Pollution (g) EEG (h) ECG

(i) UCR-Mallat (j) UCR-Trace (k) UCR-Small Kitchen Appl. (l) UCR-ECG-Torso

Figure 5: Evaluating clusters in terms of pairwise mutual information between timeseries within clusters (intra-cluster ↑) and across clusters (inter-cluster ↓). The proposed method is ITC-DM shown in solid red circles.

cluster labels.

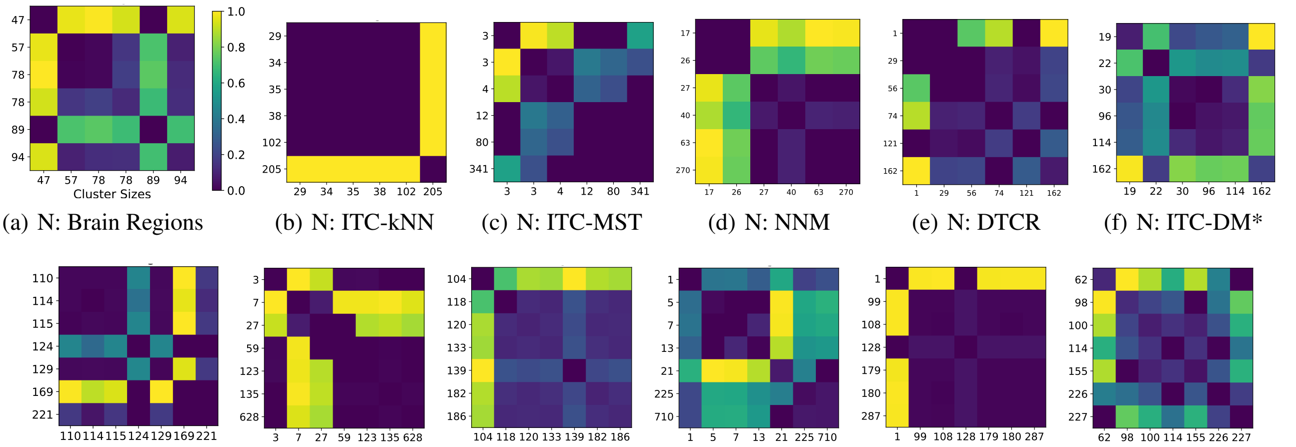
KL-divergence metric In addition, the KL-Divergence objective can itself serve as an evaluation metric. For every pair of clusters, we evaluate empirical KL-D between their respective cluster distributions. We are only interested in the relative values of KL-D between and across clusters obtained from all the methods. For estimating the KL-D criterion as a metric, we ensure that it is estimated independently of its estimation as the objective. We employ Transformers with 10 attention heads and feedforward dimension of 32 to estimate the KL-D metric; we use learning rate of 3e-4, dropout rate of 0.2, and 200 iterations of weight updates or until convergence.

On the role of domain knowledge in evaluating clusters We argue that, since clustering is purely an unsupervised problem for exploratory analysis, it is not apt to consider class labels or any domain knowledge gleaned from supervised tasks as a proxy for cluster labels. Class labels can only serve as yet another human annotation of cluster labels, and not as the ground truth.

3.1.2 Comparative Analysis

First, we focus on the metric of inter- and intra-cluster MI, and present the comparative analysis of our approach and others in Fig. 5. We use the same number of clusters as the classes for the datasets where domain specific class labels are available, like the brain regions in the neuropixel dataset. This is designed to present a fair comparison concerning the class labels (yet not to be considered as ground truth). For the remaining datasets (Rain, Wind, Pollution, Stock Prices), we tune the number of clusters across all the methods and then present results on the same number of clusters. We note that some comparison methods fail to obtain clusters despite many trials with random seeds; results for such cases are missing in the plots.

In terms of achieving high intra-cluster MI but low inter-cluster MI, our method ITC-DM* performs competitively across all the datasets. In contrast, ITC-kNN which uses the mutual information objective (1) achieves high intra-cluster MI for some of the datasets (US stock returns, EEG, ECG, UCR-Mallat) but at the expense of higher inter-cluster MI. For some of the other datasets (Rain, Wind, Pollution, UCR-Trace, UCR-Small Kitchen Appliances), ITC-kNN finds



(a) N: Brain Regions (b) N: ITC-kNN (c) N: ITC-MST (d) N: NNM (e) N: DTCR (f) N: ITC-DM*

(g) SR: ITC-kNN (h) SR: ITC-MST (i) SR: NNM (j) SR: DTCR (k) SR: DEC (l) SR: ITC-DM*

Figure 6: For the Neuropixels and Stock Returns datasets, corresponding to the prefixes "N:" and "SR:", KL-D values (normalized by the maximum value) for intra-cluster (diagonal entries) and inter-cluster (off-diagonal entries) are shown across all the competitive methods while excluding those methods (due to space constraints) which produce clusters of highly poor quality. The clusters are sorted in ascending order of the cluster size.

clusters which are poor at both the metrics, intra- and inter-cluster MI, in comparison to ITC-DM. ITC-MST, which is also based on the objective of mutual information performs poorly across many datasets (Synthetic, Neuropixels, US Stock Returns, Pollution, EEG), partly due to its reliance upon minimal spanning trees for estimating MI.

Traditional methods like kMeans and Spectral clustering are ineffective due to sensitivity to noise and degrade when clusters are high imbalanced; see the results for datasets: Synthetic, Neuropixels, US Stock Returns, Rain, Pollution, EEG, ECG, UCR-Small Kitchen Appliances. Even deep learning based approaches such as NNM, i.e. clustering via nearest neighbor matching of input data points in their deep neural representations, are vulnerable to the noisy timeseries datas including, Neuropixels, US Stock Returns, Rain, Wind, Pollution, EEG, ECG, UCR-Mallat, UCR-Trace. Similarly, the deep clustering method DEC exhibits unreliable performance, finding poor choices of clusters for some of the datasets, Neuropixels, Rain, Wind, EEG, and UCR-Trace. While DTCR is consistently superior to DTC (except for the synthetic data), it also finds clusters of poor quality for Neuropixels, US Stock Returns, Wind, and UCR-Mallat.

In Fig. 5, it is also interesting to see that, for some datasets, ITC-DM* outperforms w.r.t. class labels as well; see the results for Neuropixels, UCR-Mallat and UCR-Trace. In Fig. 5(b), when comparing ITC-DM* w.r.t. brain regions. ITC-DM* finds clusters of neurons with lower inter-cluster MI. This empirical result conforms to knowledge of human brains where strong dependence between neurons across brain regions imply information flow in the visual system. As a matter of fact, all the clustering methods, except for kMeans, spectral, and DTC, find clusters with inter-cluster MI lower than the brain regions. Among those, ITC-DM* find the ones with the highest intra-cluster MI.

kMeans	Spectral	ITC-kNN	ITC-MST	DEC	DTCR	NNM	**ITC-DM***
418	5902	2	15	289	545	231	168

Table 1: Average compute time (in seconds) for all the clustering methods on Neuropixels dataset.

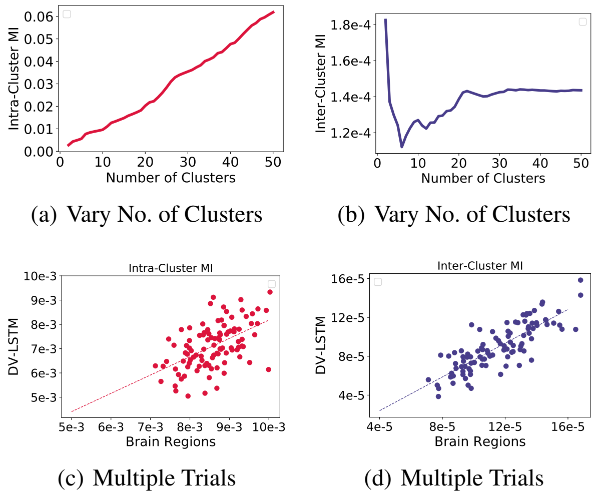
Comparisons with KL-D metric Next, in Fig. 6, we compare the most competitive methods above in terms of the KL-D metric. As desired, we observe higher KL-D (↑) between clusters from our method ITC-DM* vs the other methods. For neuropixels dataset, we observe high divergence between brain regions as well.

Compute time In Table 1, we observe that compute time is competitively lower w.r.t. the neural baselines. On the other hand, ITC-kNN and ITC-MST which rely upon nearest neighbor distances instead of neural representations exhibit comparably negligible compute cost. As for Spectral clustering, compute time can vary as per the Eigen spectrum.

3.1.3 Ablation Study

We present a detailed analysis for our approach using the Neuropixels dataset.

In Fig. 7(a) and 7(b), we analyze intra- and inter-cluster MI as we increase the number of clusters from 2 to 50. We observe that the inter-cluster MI initially declines and attains a minimum at 6 clusters, and then continues to increase. Interestingly, the optimal number of clusters (6) as indicated by the lowest inter-cluster MI is also the number of brain



(a) Vary No. of Clusters

(b) Vary No. of Clusters

(c) Multiple Trials

(d) Multiple Trials

Figure 7: Detailed analysis of our ITC-DM model (also referred as "DV-LSTM") for Neuropixels dataset.

regions. In Fig. 7(c) and 7(d), we analyze clusters obtained from 100 different trials of neural activity, and correlate intra- and inter-cluster MI of the clusters w.r.t. the brain regions. Note that MI metrics vary even for fixed brain regions, since neural activity varies across all the trials. We observe that the intra- and inter- cluster MI of clusters learned by our algorithm remain close to the corresponding brain regions, and exhibit high correlations with the latter.

In Fig. 8, we present an ablation study varying all the hyperparameters in our model one by one. To provide relative measures, we show intra- and inter-cluster MI for all the competitive methods and ITC-DM with its default hyperparameters, as in 5(b). Starting from the default configuration, we vary each hyperparameters to observe the corresponding change in the two metrics. We vary the number of layers in the default LSTM model (*1*, 2, 3, 4, 5, 8, 10) and observe only marginal changes in the metrics. However, the number of hidden units is a sensitive hyperparameter (8, 16, *32*, 64, 128, 256, 512, 1024); large or small numbers of units degrade the performance. Std for the initialization of weight parameters is an important parameter (0.01, 0.03, 0.05, *0.1*, 0.2, 0.3, 0.5) with high variability in the results; however, the default value of 0.1 works consistently across all the experiments. Perhaps surprising, the learning rate (LR) is only mildly sensitive (1e-5 to *1e-1*) with no clear pattern for whether lower or higher LR is better.

From varing the number of iterations in the greedy bisection algorithm (10 to 300), we find that a minimum of 30 iterations is necessary to ensure good performance. We vary warm up iterations between two extremes from 0 to 50 (default is 10) and notice that extremly low or high values are detrimental. We also find that FNNs perform well, and their performance varies by learning rate. We find Transformers suited for estimating KL-D as a metric for their stability in learning, but their performance for clustering underperforms FNNs and LSTMs. Lastly, we evaluate the greedy cut point algorithm (4) and observe its performance comparable but not superior to greedy bisection algorithm. Overall, it is noteworthy that even changing any hyperparameter to extreme values, ITC-DM* remain highly competitive.

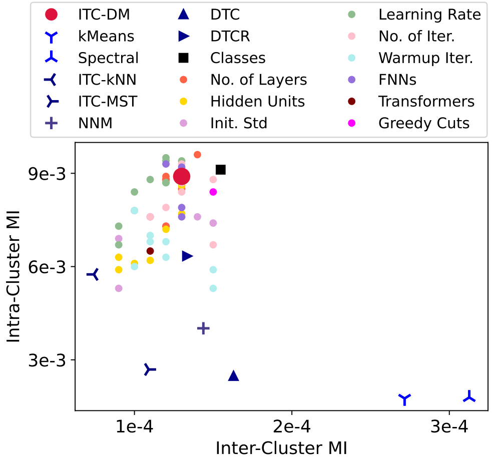


Figure 8: Ablation study for our approach using Neuropixels dataset. ITC-DM refers to our model with default configuration used for the primary analysis. We analyze the performance of our model w.r.t. change in all the hyperparametes, such as the number of layers, learning rate, etc.

4 CONCLUSIONS

To the best of our knowledge, this paper presents the first deep learning based information theoretic approach for clustering, together with a novel KL-Divergence criterion for optimization with no assumptions underlying the true data distribution. This new criterion subsumes the objective of mutual information. We propose to estimate KL-D in its dual form which gives us a highly efficient framework for optimization along with theoretical guarantees. Our experimental results on 12 real world timeseries datasets demonstrate that our approach is highly competitive w.r.t. other information theoretic clustering methods as well as advanced deep learning methods in ensuring two desirable properties: high KL-divergence among cluster distributions, and low inter-cluster pairwise mutual information.

Acknowledgements

The authors would like to thank, Syamantak Datta Gupta, Sanghamitra Dutta, Majid Behbahani, Satish Kumar Thittamaranahalli, Irina Rish, Frederik Wenkel, Fernando Gama, Umang Gupta, Volodymyr Volchenko, Pablo Vicente Juan, Antonio Musumeci, for providing their valuable feedback.

References

Maedeh Ahmadi, Mehran Safayani, and Abdolreza Mirzaei. Deep graph clustering via mutual information maximization and mixture model. *arXiv preprint arXiv:2205.05168*, 2022.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. Clustering with bregman divergences. *Journal of Machine Learning Research*, 2005.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of International Conference on Machine Learning*, 2018.

Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, et al. Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240*, 2020.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of The Web Conference 2020*, 2020.

Christian Böhm, Christos Faloutsos, Jia-Yu Pan, and Claudia Plant. Robust information-theoretic clustering. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Cristian Challu, Kin Olivares, Boris Oreshkin, et al. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 2016.

Ferdinando Cicalese, Eduardo Laber, and Lucas Murtinho. New results on information theoretic clustering. In *Proceedings of International Conference on Machine Learning*, 2019.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Mithun Das Gupta, Srinidhi Srinivasa, Meryl Antony, et al. Kl divergence based agglomerative clustering for automated vitiligo grading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Jason Davis and Inderjit Dhillon. Differential entropic clustering of multivariate gaussians. *Advances in Neural Information Processing Systems*, 2006.

Saskia EJ de Vries, Jerome A Lecoq, Michael A Buice, Peter A Groblewski, Gabriel K Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 2020.

Inderjit S Dhillon, Subramanyam Mallela, and Rahul Kumar. A divisive information theoretic feature clustering algorithm for text classification. *The Journal of Machine Learning Research*, 2003.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. i. *Communications on pure and applied mathematics*, 1975.

Lev Faivishevsky and Jacob Goldberger. Nonparametric information theoretic clustering algorithm. In *Proceedings of International Conference on Machine Learning*, 2010.

Wei Fan, Shun Zheng, Xiaohan Yi, et al. Depts: Deep expansion learning for periodic time series forecasting. In *Proceedings of International Conference on Learning Representations*, 2021.

Erhan Gokcay and Jose C. Principe. Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *Proceedings of International Conference on Learning Representations*, 2021.

Ran He, Liang Wang, Zhenan Sun, Yingya Zhang, and Bo Li. Information theoretic subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2015.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of International Conference on Machine Learning*, 2017.

Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 2010.

Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *Advances in Neural Information Processing Systems*, 2017.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *Proceedings of International Conference on Learning Representations*, 2019.

Marc T Law, Raquel Urtasun, and Richard S Zemel. Deep spectral clustering learning. In *Proceedings of International Conference on Machine Learning*, 2017.

Jian Ma. Estimating transfer entropy via copula entropy. *arXiv preprint arXiv:1910.04375*, 2019.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. *Advances in Neural Information Processing Systems*, 2019.

Naveen Sai Madiraju. *Deep temporal clustering: Fully unsupervised learning of time-domain features*. PhD thesis, Arizona State University, 2018.

Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 2018.

Andreas C Müller, Sebastian Nowozin, and Christoph H Lampert. Information theoretic clustering using minimum spanning trees. In *Joint DAGM and OAGM Symposium*, 2012.

Foivos Ntelemis, Yaochu Jin, and Spencer A Thomas. Information maximization clustering via multi-view self-labelling. *arXiv preprint arXiv:2103.07368*, 2021.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2019.

John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. In *Proceedings of SIGMOD International Conference on Management of Data*, 2015.

Naveen Sai Madiraju, Seid M Sadat, Dimitry Fisher, and Homa Karimabadi. Deep temporal clustering: Fully unsupervised learning of time-domain features. *arXiv e-prints*, 2018.

Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 2021.

Shashank Singh and Bryan Hooi. Information theoretic clustering using kernel density estimation. *CMU*, 2015.

Noam Slonim, Gurinder Singh Atwal, Gašper Tkačik, and William Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences*, 2005.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

Susanne Still and William Bialek. How many clusters? an information-theoretic perspective. *Neural computation*, 2004.

Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 2003.

Masashi Sugiyama, Gang Niu, Makoto Yamada, Manabu Kimura, and Hirotaka Hachiya. Information-maximization clustering based on squared-loss mutual information. *Neural Computation*, 2014.

Vincenzo Tola, Fabrizio Lillo, Mauro Gallegati, and Rosario N. Mantegna. Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control*, 2008.

Greg Ver Steeg, Aram Galstyan, Fei Sha, and Simon DeDeo. Demystifying information-theoretic clustering. In *Proceedings of International Conference on Machine Learning*, 2014.

Meihong Wang and Fei Sha. Information theoretical clustering via semidefinite programming. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2011.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of International Conference on Machine Learning*, 2016.

Lin Yang, Wentao Fan, and Nizar Bouguila. Clustering analysis via deep generative models with mixture models. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019a.

Xiaojiang Yang, Junchi Yan, Yu Cheng, and Yizhe Zhang. Learning deep generative clustering via mutual information maximization. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019b.

Sebastian Zeng, Florian Graf, et al. Topological attention for time series forecasting. *Advances in Neural Information Processing Systems*, 2021.