CONVEX CLUSTERING FOR AUTOCORRELATED TIME SERIES

*Max Revay, and Victor Solo**

School of Electrical Engineering and Telecommunications, UNSW, Sydney, AUSTRALIA

ABSTRACT

While clustering in general is a heavily worked area, clustering of auto-correlated time series (CATS) has received relatively little attention. Here, we develop a convex clustering algorithm suited to auto-correlated time series and compare it with a state of the art method. We find the proposed algorithm is able to more accurately identify the true clusters.

Index Terms— Time series clustering, convex clustering, data mining, time series analysis, unsupervised learning

1. INTRODUCTION

Clustering has a long history of development with applications in many areas [1, 2]. Most prior work, however, has ignored auto-correlated time series; see for instance, the following two surveys [3, 4]. By ignoring the 'auto-correlation feature', these methods assume that the signals are (vector) white noise time series; an assumption that does not hold for many dynamical systems. For instance, financial time series [5], biomedical signals [6] and speech [7] all exhibit autocorrelation.

Time series data is challenging to cluster for a number of reasons: Firstly the datasets are often large. Secondly, the datasets often contain many time series of different lengths. Thirdly, it is not always clear which similarity measure is appropriate [3]. Finally, the autocorrelation feature is often missed. This is true of shape and feature based similarity measures, e.g. dynamic time warping [8], euclidean distance [9] and sequence weighted alignment [10]. These approaches are inappropriate for clustering autoregressions as different realizations of an autoregression can have very different shapes with similar statistical properties.

There has been some development of CATS algorithms in the statistics and data mining literatures [11, 12, 13]. Recent work from our lab has developed a mixture model approach as well as a better performing limiting version of it [14]. These non-convex formulations can work well but sometimes suffer from the classic sensitivity to initialization problem.

The sensitivity of many clustering algorithms to initialization is well known and has motivated a promising new approach: convex clustering [15, 16, 17, 18, 19, 20]. So far, convex clustering has not been applied to auto-correlated time series. To the the author's knowledge, this is the first work to do convex CATS.

The remainder of the paper is organized as follows: In section 2, we formulate the convex CATS problem and then introduce an equivalent but computationally simpler formulation. In section 3, we introduce a specialized algorithm based on sequentially merging clusters. In section 4, we introduce an ad-hoc model selection criteria. We conduct a series of numerical experiments in Section 5 and finally in Section 6 we present our conclusions.

1.1. Preliminaries

Consider a zero mean time series $z_0, .., z_T$, the order p autoregression (AR) with parameters $\phi \in \mathbb{R}^p$ is given by,

$$z_t = \sum_{j=1}^{p} \phi_j z_{t-j} + \epsilon_t, \quad t = 1, .., T. \tag{1}$$

where T is the number of observations and ϵ_t is a sequence of zero mean, independently and identically distributed Gaussian white noise with variance σ^2. Let $z_{a:b} = [z_a, z_{a+1}, ..., z_b]^\top$ where $a < b$. A least squares estimator for the autoregression is then given by

$$\hat{\phi} \ = \ \text{LSE}(z) := \left(X^\top X\right)^{-1} X^\top z_{p:T}, \tag{2}$$

where $X = \left[z_{0:t-p}, z_{1:t-p+1}, \cdots, z_{p-1:T-1}\right]$. The estimates $\hat{\phi}$ are normally distributed with mean ϕ and variance $\left(X^\top X\right)^{-1} \sigma^2$.

2. PROBLEM FORMULATION

Consider the problem of clustering N time series $\{y_1, ...y_N\}$, each of which was generated by one of K autoregressions where $K < N$. We assume that the data was generated by an autoregressive process with parameters $\{\phi_k \mid k = 1, ..., K\}$ and the true cluster assignments are denoted by $\{\mathcal{I}_k \mid k = 1, ..., K\}$ where $i \in \mathcal{I}_k$ if y_i was generated by the autoregression ϕ_k and

$$\bigcup_{k=1}^{K} \mathcal{I}_k = \{1, ..., N\}, \quad \mathcal{I}_\ell \cap \mathcal{I}_k = \emptyset, \quad \ell \neq k. \tag{3}$$

*Corresponding author (email: v.solo@unsw.edu.au). This work was partly supported by an Australian Research Council grant.

Condition (3) ensures that $\mathcal{I}_1, ..., \mathcal{I}_K$ partitions the indices $\{1, ..., N\}$. For each time series, we generate autoregression features by $\hat{\phi}_i = \text{LSE}(y_i),\ i = 1, ..., N$.

2.1. Convex Clustering

We formulate the clustering p roblem as a convex optimization problem: $\min_{\mu_{1:N}}\ J(\mu_{1:N})$, where

$$J(\mu_{1:N}) = \frac{1}{2}\sum_{i=1}^{N} \|\hat{\phi}_i - \mu_i\|^2 + \gamma \sum_{j=1}^{N} w_{ij}\|\mu_i - \mu_j\|_2 \quad (4)$$

where $\mu_i \in \mathbb{R}^p$ indicates the cluster center associated with the AR $\hat{\phi}_i$. The second term in (4) acts as a penalty term which encourages cluster centers to coalesce.

If $\mu_i = \mu_j$ for some i and j, then ARs $\hat{\phi}_i$ and $\hat{\phi}_j$ belong to the same cluster. We also store index sets $\hat{\mathcal{I}}_k$ where $i \in \hat{\mathcal{I}}_k$ if i belongs to cluster k and (3) holds. The total number of clusters estimated is denoted $\hat{K}$ and the size of each cluster is denoted $n_k = \text{Card}(\hat{\mathcal{I}}_k)$.

Denoting $J(\mu_{1:N}) = J$, we can now rewrite (4) as

$$J = \sum_{k=1}^{\hat{K}} \sum_{i \in \hat{\mathcal{I}}_k} \frac{1}{2}\|\hat{\phi}_i - \mu_i\|^2 + \gamma \sum_{\ell=1}^{\hat{K}} \sum_{j \in \hat{\mathcal{I}}_\ell} w_{ij}\|\mu_i - \mu_j\|. \quad (5)$$

A number of specialized algorithms for minimizing (4) have been proposed based on, e.g., splitting [18], subgradient methods [17] or a semi-smooth Newton method [19, 20]. These methods optimize over N variables and then recover the cluster sets in a post-processing step. In this work, we construct an equivalent, reduced version of (4) that depends only on $\hat{K}$ variables.

2.2. Merging Clusters

As previously mentioned, two autoregressions $\hat{\phi}_i$ and $\hat{\phi}_j$ are in the same cluster if $\mu_i = \mu_j$. We introduce a variable for the common cluster center, denoted $\hat{\mu}_k$, so that

$$\mu_i = \hat{\mu}_k, \quad \forall i \in \hat{\mathcal{I}}_k. \quad (6)$$

Substituting $\hat{\mu}_k$ for μ_i , $i \in \mathcal{I}_k$, we can rewrite (5) as

$$J = \sum_{k=1}^{\hat{K}} \sum_{i \in \hat{\mathcal{I}}_k} \frac{1}{2}\|\hat{\phi}_i - \hat{\mu}_k\|^2 + \gamma \sum_{\ell=1}^{\hat{K}} \sum_{j \in \hat{\mathcal{I}}_\ell} w_{ij}\|\hat{\mu}_k - \hat{\mu}_\ell\|. \quad (7)$$

Also note that $\sum_{i \in \mathcal{I}_k} \|\hat{\phi}_j - \hat{\mu}_k\|^2 = n_k\|\bar{\phi}_k - \hat{\mu}_k\|^2 + \mathcal{O}[1]$, where $\bar{\phi}_k = \frac{1}{n_k}\sum_{j \in \hat{\mathcal{I}}_k} \hat{\phi}_j$ where the constant terms are a result of completing the square. Minimizing (7) is therefore equivalent to minimizing

$$\tilde{J} = \sum_{k=1}^{\hat{K}} \frac{n_k}{2}\|\bar{\phi}_k - \hat{\mu}_k\|^2 + \gamma \sum_{\ell=1}^{\hat{K}} \alpha_{\ell k}\|\hat{\mu}_\ell - \hat{\mu}_k\|, \quad (8)$$

where $\alpha_{\ell k} = \sum_{i \in \mathcal{I}_\ell} \sum_{j \in \mathcal{I}_k} w_{ij}$. Note that the (8) depends on decision variables $\hat{\mu}_1, ..., \hat{\mu}_{\hat{K}}$. When $\hat{K} \ll N$, the evaluation of (8) is much more efficient than the original objective (4).

3. ALGORITHMIC DETAILS

We apply cyclic descent to minimize (8). This family of algorithms successively minimizes with respect to a subset of the coordinates at a time. In this case, we consider updating with respect to a cluster center $\hat{\mu}_c$:

$$\begin{aligned}
\hat{\mu}_c^* &= \arg\min_{\hat{\mu}_c} \tilde{J}(\hat{\mu}_{1:\hat{K}}), \\
&= \arg\min_{\hat{\mu}_c} \sum_{k=1}^{\hat{K}} \frac{n_k}{2}\|\bar{\phi}_k - \hat{\mu}_k\|^2 + \gamma \sum_{\ell=1}^{\hat{K}} \alpha_{\ell k}\|\hat{\mu}_\ell - \hat{\mu}_k\|, \\
&= \arg\min_{\hat{\mu}_c} \frac{n_c}{2}\|\bar{\phi}_c - \hat{\mu}_c\|^2 + \gamma \sum_{k=1}^{\hat{K}} \sum_{\ell=1}^{\hat{K}} \alpha_{\ell k}\|\hat{\mu}_\ell - \hat{\mu}_k\|, \\
&= \arg\min_{\hat{\mu}_c} \tilde{J}_c(\hat{\mu}_{1:\hat{K}}), \qquad (9)
\end{aligned}$$

where,

$$\begin{aligned}
\tilde{J}_c := &\frac{n_c}{2}\|\bar{\phi}_c - \hat{\mu}_c\|^2 + \\
&\gamma \sum_{k \neq c} \alpha_{ck}\|\hat{\mu}_c - \hat{\mu}_k\| + \gamma \sum_{\ell \neq c} \alpha_{\ell c}\|\hat{\mu}_\ell - \hat{\mu}_c\|. \quad (10)
\end{aligned}$$

When W is symmetric, α is symmetric and we can write (10) as

$$\tilde{J}_c = \frac{n_c}{2}\|\bar{\phi}_c - \hat{\mu}_c\|^2 + 2\gamma \sum_{\ell \neq c} \alpha_{c\ell}\|\hat{\mu}_\ell - \hat{\mu}_c\|. \quad (11)$$

For the remainder of this work we take α as symmetric but we note that it is easy to extend to the non-symmetric case.

We refer to the problem $\min_{\hat{\mu}_c} \tilde{J}_c$ as the quadratic least absolute deviations (QLAD) problem. QLAD is convex and smooth whenever the second is smooth. That is, $\hat{\mu}_c \neq \hat{\mu}_a$ for some $a \neq c$. Consequently, there are two cases that must be considered: when the optimal $\hat{\mu}_c^* = \hat{\mu}_a$ for some $a \neq c$, and when the $\hat{\mu}_c^* \neq \hat{\mu}_a$ for all a.

3.1. Condition for Merging Clusters

At each iteration, we first check if the minimizer of (11) occurs at one of the other cluster centers μ_a where $a \neq c$. We use the following condition to check if $\hat{\mu}_c = \hat{\mu}_a$ is an optima of J_c:

$$2\gamma\alpha_{ac} \geq \left\| n_c\left(\bar{\phi}_c - \hat{\mu}_a\right) + 2\gamma \sum_{\ell \neq a, c} \alpha_{c\ell} \frac{\hat{\mu}_\ell - \hat{\mu}_a}{\|\hat{\mu}_\ell - \hat{\mu}_a\|} \right\|_2 \quad (12)$$

Theorem 3.1. *Suppose that* (12) *holds for some* $a \neq c$*, then* $\arg\min_{\hat{\mu}_c} \tilde{J}_c = \hat{\mu}_a$.

Proof. We will show that (12) implies $\tilde{J}_c(\hat{\mu}_a + \epsilon) \geq \tilde{J}_c(\hat{\mu}_a)$ for all $\epsilon \in \mathbb{R}^p$. First, note that

$$\begin{aligned} & \tilde{J}_c(\hat{\mu}_a + \epsilon) \geq \tilde{J}_c(\hat{\mu}_a), \qquad (13) \\ \iff & \frac{n_c}{2}\|\bar{\phi}_c - \hat{\mu}_a - \epsilon\|^2 + \\ & 2\gamma\alpha_{ac}\|\epsilon\| + 2\gamma \sum_{\ell \neq a,c} \alpha_{c\ell}\|\hat{\mu}_\ell - \hat{\mu}_a - \epsilon\| \\ & \geq \frac{n_c}{2}\|\bar{\phi}_c - \hat{\mu}_a\|^2 + 2\gamma \sum_{\ell \neq a,c} \alpha_{c\ell}\|\hat{\mu}_\ell - \hat{\mu}_a\|. \end{aligned}$$

Rearranging, dividing through by $\|\epsilon\|$ and taking the limit as $\|\epsilon\| \to 0$, we obtain the directional derivatives of $\|\bar{\phi}_k - \hat{\mu}_c\|^2$ and $\sum_{\ell \neq k,c} \alpha_{\ell k}\|\bar{\phi}_k - \hat{\mu}_c\|$, in the direction ϵ and evaluated at $\hat{\mu}_a$. Evaluating the derivatives gives the following

$$2\gamma\alpha_{ac} \geq \left(n_c(\bar{\phi}_c - \hat{\mu}_a) + 2\gamma \sum_{\ell \neq c,a} \alpha_{\ell c} \frac{\hat{\mu}_\ell - \hat{\mu}_a}{\|\hat{\mu}_\ell - \hat{\mu}_a\|} \right)^\top \frac{\epsilon}{\|\epsilon\|} \qquad (14)$$

Equation (12) is equivalent to (14) which implies (13) for small ϵ. This shows that $\hat{\mu}_a$ is a local optima. Since J_c is convex, $\hat{\mu}_a$ is a global optima. □

3.2. Implementation Details

This section details some of the implementation details of the proposed algorithm:

1. We initialize the algorithm by setting $\hat{\mu}_i = \hat{\phi}_i$, $\hat{\mathcal{I}}_i = \{i\}$ for $i = 1, ..., N$ and $\hat{K} = N$.

2. If condition (12) holds for some $a, c \in [1, ..., \hat{K}]$, then we merge clusters a and c and set $\hat{K} \leftarrow \hat{K} - 1$.

3. If the condition (12) fails for all $a, c \in [1, ..., \hat{K}]$, then we optimize (8) directly using gradient descent with the following linesearch algorithm [21].

4. Following the recommendation of [18], we select the weights as $w_{ij} = \mathbb{1}_n(\hat{\phi}_i, \hat{\phi}_j) \cdot \exp(-\beta\|\hat{\phi}_i - \hat{\phi}_j\|^2)$ where $\beta \geq 0$ and $\mathbb{1}_n(\hat{\phi}_i, \hat{\phi}_j)$ is an indicator function that returns 1 if $\hat{\phi}_i$ is in $\hat{\phi}_j$'s n nearest neighbors, else 0.

5. The set of weights w_{ij} typically contains many zeros. Exploiting sparsity allows evaluation of (8) and it's gradient to have computational complexity proportional to the number of nonzero weights. Condition (12) only needs to be checked for nonzero α_{ac}.

6. The algorithm is terminated when $\|\nabla J\|_\infty \leq \epsilon$.

4. CLUSTER SELECTION

The problem of selecting the correct number of clusters is known to be extremely difficult and usually approached using heuristic methods [22, 23, 24, 25]. Motivated by [14], we propose a limiting version of the Bayesian information criterion (BIC) for mixture models:

$$\begin{aligned} RSS &= \sum_{k=1}^{\hat{K}} \sum_{i \in \hat{\mathcal{I}}_k} \|\hat{\phi}_i - \bar{\phi}_k\|^2, \\ \text{pseudo-BIC} &= RSS + \hat{K} \log N. \qquad (15) \end{aligned}$$

While this criterion is ad-hoc, we find in the next section that it works quite well.

5. NUMERICAL RESULTS

We demonstrate the proposed algorithm using simulated data from randomly sampled stable auto-regressions. We use the following procedure to generate data:

1. Sample random stable auto-regressions $\phi_1, ..., \phi_K$ each of order p with variance $\sigma_k^2 = 1$.

2. For $\phi_1, ..., \phi_K$ simulate N_c time-series of length T.

This generates $N = K \times N_c$ time series of length T.

5.1. Small Scale Example

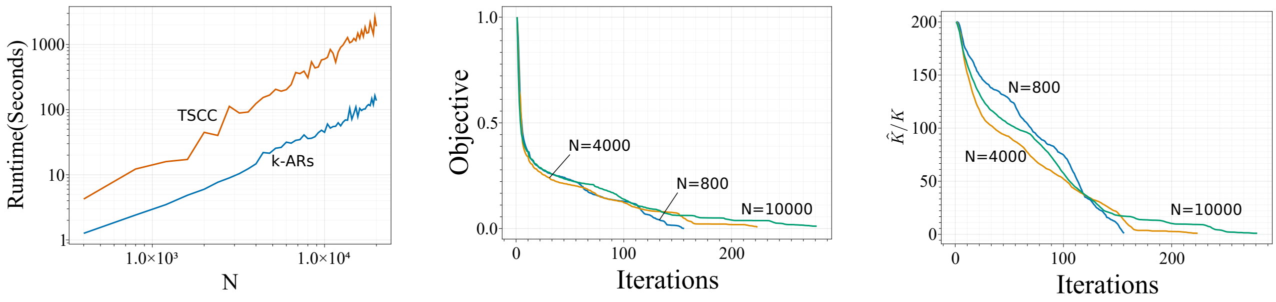
We first demonstrate the performance of the algorithm on a small scale example with $K = 4$, $N_c = 50$, $T = 200$ and $p = 2$. The small scale allows us to plot the parameters directly as the autoregressions only have two free parameters. The weights are chosen as discussed in Section 3.2 with 10 neighbors and $\beta = 1$.

In Fig. 2, we have plotted the coefficients of each autoregression $\hat{\phi}_1, ..., \hat{\phi}_N$. The assigned cluster for each autoregression is indicated by color and marker type. We see that the algorithm appropriately clusters each time series; in fact, in this case, the algorithm exactly recovers the true clusters.

The estimated number of clusters versus γ is plotted in Fig. 3. There is a long plateau for $0.7 \leq \gamma \leq 15$ where the algorithm returns the correct number of clusters. We have also plotted the pseudo-BIC (15) versus γ and observe a minimum around the correct number of clusters. In all of our experiments, we find that the pseudo-BIC usually obtains a minima at or close to the correct number of clusters.

5.2. Comparison to k-ARs

This section studies a larger-scale example and provides a set of comparisons to the k-ARs algorithm [14]. We generate a dataset with $p = 6$, $T = 500$, $N_c = 200$ and vary the number of clusters from $K = 2, 4, \cdots, 100$. For fair comparison to



(a) Runtime versus the number of data points.

(b) Cost versus iterations. The objective function plotted is (4), normalized by its initial value at iteration 0.

(c) Estimated clusters $\hat{K}$ versus iterations for varying N, normalized by the true number of clusters.

Fig. 1: Large Scale Clustering Results

N	400	800	1200	1600	2000	2400	2800	3200	3600	4000
TSCC	**100**	**100.0**	**99.6**	87.4	**100**	**99.6**	**85.7**	**93.8**	**88.2**	**94.9**
k-ARs	**100**	75.0	83.3	**87.5**	**100**	91.7	85.5	87.5	76.8	75.0

Table 1: Comparison of clustering precision between k-ARs and TSCC. When properly tuned, TSCC significantly outperforms k-ARs. We observed a similar performance gap in recall, F1 and adjusted rand index.

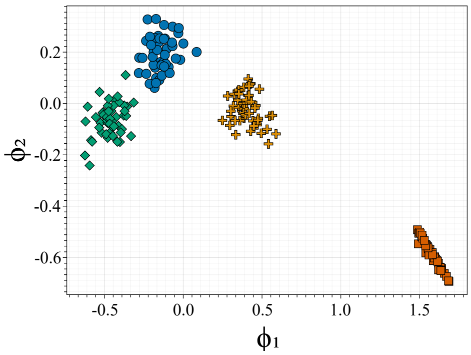


Fig. 2: Plot of the two autoregressive coefficients for each $\hat{\phi}_1, ..., \hat{\phi}_N \in \mathbb{R}^2$. The marker type and color indicates the assigned clusters for each time series.

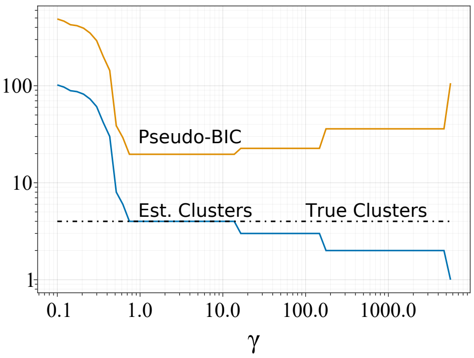


Fig. 3: The blue line shows the estimated number of clusters $\hat{K}$ versus the regularization parameter γ. The black line shows the true number of clusters K.

k-ARs which specifies the number of clusters a-priori, we set $\gamma = 500$ and then terminate TSCC when the correct number of clusters is obtained. The weights were selected using the method discussed in Section 3.2 with 5 neighbors and $\beta = 1$.

Table 1 shows the clustering precision for varying numbers of points. We find that when properly tuned, convex clustering significantly outperforms k-ARs in terms of clustering precision. We also found a similar difference in performance when using the recall, F1 and adjust rand index performance measures, however these results were omitted due to space restrictions. The improved performance of TSCC, however, comes at a cost to runtime, as shown in Figure 1a.

In figures 1b and 1c, we have plotted the normalized objective function and the number of clusters versus the number of iterations, respectively for $N_c = 4, 20, 50$.

6. CONCLUSION

We have demonstrated the first application of convex clustering to autoregressive time series and developed a novel algorithm for solving convex clustering problems. Numerical experiments show that the convex clustering formulation compares favorably to a state of the art method, obtaining significantly higher precision. We have also proposed a heuristic for model selection that we have found empirically effective.

In future work, we plan to refine the algorithm to account for the covariance of the autoregression estimates, investigate alternative similarity measures and further study the empirical and theoretical properties of the pseudo-BIC

7. REFERENCES

[1] Peter E Hart, David G Stork, and Richard O Duda, *Pattern classification*, Wiley Hoboken, 2000.

[2] Christopher M Bishop, "Pattern recognition," *Machine learning*, vol. 128, no. 9, 2006.

[3] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[4] T Warren Liao, "Clustering of time series data—a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[5] Massimiliano Marcellino, James H Stock, and Mark W Watson, "A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series," *Journal of econometrics*, vol. 135, no. 1-2, pp. 499–526, 2006.

[6] Wiktor Olszowy, John Aston, Catarina Rua, and Guy B Williams, "Accurate autocorrelation modeling substantially improves fmri reliability," *Nature communications*, vol. 10, no. 1, pp. 1–11, 2019.

[7] Sriram Ganapathy, "Multivariate autoregressive spectrogram modeling for noisy speech recognition," *IEEE signal processing letters*, vol. 24, no. 9, pp. 1373–1377, 2017.

[8] Meinard Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[9] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos, "Fast subsequence matching in time-series databases," *ACM Sigmod Record*, vol. 23, no. 2, pp. 419–429, 1994.

[10] Michael D Morse and Jignesh M Patel, "An efficient and accurate method for evaluating time series similarity," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 569–580.

[11] Marcella Corduas and Domenico Piccolo, "Time series clustering and classification by the autoregressive metric," *Computational statistics & data analysis*, vol. 52, no. 4, pp. 1860–1872, 2008.

[12] Robert H Shumway, "Time-frequency clustering and discriminant analysis," *Statistics & probability letters*, vol. 63, no. 3, pp. 307–314, 2003.

[13] Yimin Xiong and Dit-Yan Yeung, "Mixtures of arma models for model-based time series clustering," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 2002, pp. 717–720.

[14] Zuogong Yue and Victor Solo, "Large-scale time series clustering with k-ars," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6044–6048.

[15] Kristiaan Pelckmans, Joseph De Brabanter, Johan AK Suykens, and Bart De Moor, "Convex clustering shrinkage," in *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

[16] F Lindsten, H Ohlsson, and L Ljung, "Just relax and come clustering! a convexication of k-means clustering technical report," *Linköping University, Department of Electrical Engineering, Automatic Control*, 2011.

[17] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert, "Clusterpath an algorithm for clustering using convex fusion penalties," in *28th international conference on machine learning*, 2011, p. 1.

[18] Eric C Chi and Kenneth Lange, "Splitting methods for convex clustering," *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, 2015.

[19] Yancheng Yuan, Defeng Sun, and Kim-Chuan Toh, "An efficient semismooth newton based algorithm for convex clustering," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5718–5726.

[20] Defeng Sun, Kim-Chuan Toh, and Yancheng Yuan, "Convex clustering: Model, theoretical guarantee and efficient algorithm.," *J. Mach. Learn. Res.*, vol. 22, pp. 9–1, 2021.

[21] Jorge J Moré and David J Thuente, "Line search algorithms with guaranteed sufficient decrease," *ACM Transactions on Mathematical Software (TOMS)*, vol. 20, no. 3, pp. 286–307, 1994.

[22] 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*, vol. 98, no. 463, pp. 750–763, 2003.

[23] Junhui Wang, "Consistent selection of the number of clusters via crossvalidation," *Biometrika*, vol. 97, no. 4, pp. 893–904, 2010.

[24] Robert Tibshirani, Guenther Walther, and Trevor Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[25] Jeffrey D Banfield and Adrian E Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.