# Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data

**David Hallac, Sagar Vare, Stephen Boyd, Jure Leskovec**
Stanford University
{hallac, svare, boyd, jure}@stanford.edu

## Abstract

Subsequence clustering of multivariate time series is a useful tool for discovering repeated patterns in temporal data. Once these patterns have been discovered, seemingly complicated datasets can be interpreted as a temporal sequence of only a small number of states, or *clusters*. However, discovering these patterns is challenging because it requires simultaneous segmentation and clustering of the time series. Here we propose a new method of model-based clustering, which we call *Toeplitz Inverse Covariance-based Clustering* (TICC). Each cluster in the TICC method is defined by a correlation network, or Markov random field (MRF), characterizing the interdependencies between different observations in a typical subsequence of that cluster. Based on this graphical representation, TICC simultaneously segments and clusters the time series data. We solve the TICC problem through a scalable algorithm that is able to efficiently solve for tens of millions of observations. We validate our approach by comparing TICC to several state-of-the-art baselines in a series of synthetic experiments, and we then demonstrate on an automobile dataset how TICC can be used to learn interpretable clusters in real-world scenarios.

## 1 Introduction

Many applications, ranging from automobiles [Miyajima *et al.*, 2007] to financial markets [Namaki *et al.*, 2011] and wearable sensors [Mörchen *et al.*, 2005], generate large amounts of time series data. In most cases, this data is multivariate, where each timestamped observation consists of readings from multiple entities, or *sensors*. These long time series can often be broken down into a sequence of states, each defined by a simple "pattern", where the states can reoccur many times. For example, using automobile sensor data, a single driving session can be expressed as a sequential timeline of a few key states: turning, speeding up, slowing down, going straight, stopping at a red light, etc. This representation can be used to discover repeated patterns, understand trends, detect anomalies and more generally, better interpret large and high-dimensional datasets.

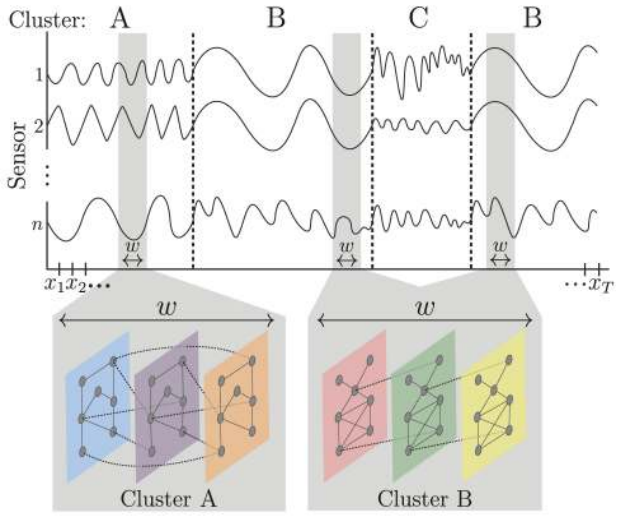



Figure 1: TICC segments a time series into a sequence of states (*i.e.*, A, B, or C). Each cluster is characterized by a correlation network, or MRF, defined over a short window of size $w$. This MRF governs the (time-invariant) partial correlation structure of *any* window inside a segment belonging to that cluster. Here, TICC learns both the cluster MRFs and the time series segmentation.

To achieve this representation, it is necessary to simultaneously segment and cluster the time series. This problem is more difficult than standard time series segmentation [Hallac *et al.*, 2016; Himberg *et al.*, 2001], since multiple segments can belong to the same cluster. However, it is also harder than subsequence clustering [Begum *et al.*, 2015; Smyth, 1997] because each data point cannot be clustered independently (since neighboring points are encouraged to belong to the same cluster). Additionally, even if one is able to simultaneously segment and cluster the data, the question still arises as to how to interpret the different clusters. These clusters are rarely known a priori, and thus are best learned through data. However, without prior knowledge, it is difficult to understand what each of the clusters refers to. Traditional clustering methods are not particularly well-suited to discover interpretable structure in the data. This is because they typically rely on distance-based metrics, such as dynamic time warping [Berndt and Clifford, 1994]. These

methods focus on matching the raw values, rather than looking for more nuanced structural similarities in the data.

In this paper, we propose a new method for multivariate time series clustering, which we call *Toeplitz inverse covariance-based clustering* (TICC). In our method, we define each cluster as a dependency network showing the relationships between the different sensors in a short (time-invariant) subsequence (Figure 1). In these networks, known as Markov random fields (MRFs), an edge represents a partial correlation between two variables [Koller and Friedman, 2009; Rue and Held, 2005; Wytock and Kolter, 2013]. Partial correlations are used to control for the effect of other confounding variables, so the existence of an edge in an MRF implies that there is a *direct* dependency between two variables. Therefore, an MRF provides interpretable insights as to precisely what the key factors and relationships are that characterize each cluster. In TICC, we discover this structure by solving a constrained sparse inverse covariance estimation problem [Friedman *et al.*, 2008; Yuan and Lin, 2006], which we call the *Toeplitz graphical lasso*, to learn the adjacency matrix of the MRF dependency network [Banerjee *et al.*, 2008; Wainwright and Jordan, 2006].

To solve the TICC problem, we use an expectation maximization (EM)-like approach, based on alternating minimization, where we iteratively cluster the data and then update the cluster parameters. Even though TICC involves solving a highly non-convex maximum likelihood problem, our method is able to find a (locally) optimal solution very efficiently in practice. We then implement our TICC method and apply it to both real and synthetic datasets. We start by evaluating performance on several synthetic examples, where there are known ground truth clusters. We compare TICC with several state-of-the-art time series clustering methods, outperforming them all by at least $41\%$ in terms of cluster assignment accuracy. We also quantify the amount of data needed for accurate cluster recovery for each method, and we see that TICC requires 3x fewer observations than the next best method to achieve similar performance. We then analyze an automobile sensor dataset to see an example of how TICC can be used to learn interpretable insights from real-world data. Applying our method, we discover that the automobile dataset has five true clusters, each corresponding to a "state" that cars are frequently in. We then validate our results by examining the latitude/longitude locations of the driving session, along with the resulting clustering assignments, to show how TICC can be a useful tool for unsupervised learning from multivariate time series.

## 2 Problem Setup

Consider a time series of $T$ sequential observations, where $x_i \in \mathbf{R}^n$ is the $i$-th multivariate observation. Our goal is to cluster these $T$ observations into $K$ clusters. However, instead of clustering each observation in isolation, we treat each point in the context of its predecessors in the time series. Thus, rather than just looking at $x_t$, we instead cluster a short subsequence of size $w \ll T$ that ends at $t$. This consists of observations $x_{t-w+1}, \ldots, x_t$, which we concatenate into an $nw$-dimensional vector $X_t$. Rather than clustering the observations directly, our approach consists of clustering these subsequences $X_1, \ldots, X_T$. We do so in such a way that encourages adjacent subsequences to belong to the same cluster, a goal called *temporal consistency*.

**Toeplitz Inverse Covariance-Based Clustering.** We define each cluster by a Gaussian inverse covariance $\Theta_i \in \mathbf{R}^{nw \times nw}$. Recall that inverse covariances show the conditional independency structure between the variables [Koller and Friedman, 2009], so $\Theta_i$ defines a Markov random field encoding the structural representation of cluster $i$. In addition to providing interpretable results, sparse graphical representations are a useful way to prevent overfitting [Lauritzen, 1996]. Our objective is to solve for these $K$ inverse covariances $\mathbf{\Theta} = \{\Theta_1, \ldots, \Theta_K\}$, one per cluster, and the resulting assignment sets $\mathbf{P} = \{P_1, \ldots, P_K\}$, where $P_i \subset \{1, 2, \ldots, T\}$, and each point is assigned to exactly one cluster. Our overall optimization problem is

$$\underset{\mathbf{\Theta} \in \mathcal{T}, \mathbf{P}}{\operatorname{argmin}} \sum_{i=1}^{K} \left[ \overbrace{\|\lambda \circ \Theta_i\|_1}^{\text{sparsity}} + \sum_{X_t \in P_i} \left( \overbrace{-\ell\ell(X_t, \Theta_i)}^{\text{log likelihood}} + \overbrace{\beta \mathbb{1}\{X_{t-1} \notin P_i\}}^{\text{temporal consistency}} \right) \right]. \quad (1)$$

We call this the *Toeplitz inverse covariance-based clustering* (TICC) problem. Here, $\mathcal{T}$ is the set of symmetric block Toeplitz matrices and $\|\lambda \circ \Theta_i\|_1$ is an $\ell_1$-norm penalty of the Hadamard (element-wise) product to incentivize a sparse inverse covariance (where $\lambda$ is a regularization parameter). Additionally, $\ell\ell(X_t, \Theta_i)$ is the log likelihood that $X_t$ came from cluster $i$,

$$\ell\ell(X_t, \Theta_i) = -\frac{1}{2}(X_t - \mu_i)^T \Theta_i (X_t - \mu_i) + \frac{1}{2} \log\det \Theta_i - \frac{n}{2} \log(2\pi), \quad (2)$$

where $\mu_i$ is the empirical mean of cluster $i$. In Problem (1), $\beta$ is a parameter that enforces temporal consistency, and $\mathbb{1}\{X_{t-1} \notin P_i\}$ is an indicator function checking whether neighboring points are assigned to the same cluster. We constrain the $\Theta_i$'s to be block Toeplitz, to ensure time-invariance within each cluster.

## 3 TICC Algorithm

Problem (1) is a mixed combinatorial and continuous optimization problem with two sets of variables, the cluster assignments $P$ and inverse covariances $\Theta$, coupled together to make the problem highly non-convex. As such, there is no tractable way to solve for the globally optimal solution. Instead, we use a variation of expectation maximization (EM) to alternate between assigning points to clusters and updating the cluster parameters.

**Cluster Assignment.** Given the model parameters (*i.e.*, inverse covariances) for each of the $K$ clusters, we assign each of the $T$ subsequences, $X_1, \ldots, X_T$, to these $K$ clusters in such a way that maximizes the likelihood of the data while also minimizing the number of times that the cluster assignment changes across the time series. This combinatorial optimization problem has $K^T$ possible assignments of points to clusters. However, we are able to solve for the globally optimal solution in only $O(KT)$ operations. We do so through dynamic programming, since this is equivalent to finding the

**Algorithm 1** EM Algorithm to Solve TICC

1: **initialize** Cluster MRFs $\boldsymbol{\Theta}$; point assignments $\mathbf{P}$.
2: **repeat**
3: *E-step:* Assign points to clusters $\rightarrow \mathbf{P}$.
4: *M-step:* Update cluster parameters $\rightarrow \boldsymbol{\Theta}$.
5: **until** Stationarity.
**return** $(\boldsymbol{\Theta}, \mathbf{P})$.

minimum cost Viterbi path [Viterbi, 1967] for the length-$T$ sequence.

**Solving the Toeplitz Graphical Lasso.** Once we have the clustering assignments, we then update the inverse covariances, given the points assignments. Here, we solve for each $\Theta_i$ in parallel, as

$$\begin{aligned} &\text{minimize} && -\log\det\Theta_i + \mathrm{tr}(S_i\Theta_i) + \frac{1}{|P_i|}\|\lambda \circ \Theta_i\|_1 \\ &\text{subject to} && \Theta_i \in \mathcal{T}. \end{aligned} \tag{3}$$

where $|P_i|$ is the number of points assigned to cluster $i$ and $S_i$ is the empirical covariance of these points. We call this problem the *Toeplitz graphical lasso*, since it is a variation on the well-known graphical lasso problem [Friedman *et al.*, 2008] where we add a block Toeplitz constraint on the inverse covariance. To solve it, we develop a scalable algorithm based on a distributed convex optimization approach know as the alternating direction method of multipliers (ADMM) [Boyd *et al.*, 2011; Parikh and Boyd, 2014].

**TICC.** Our overall TICC algorithm iterates between assigning points to different clusters (E-step, dynamic programming) and updating the cluster parameters (M-step, ADMM), as outlined in Algorithm (1).

## 4 Experiments

We built a custom Python solver to run the TICC algorithm[1]. Our solver takes as inputs the original multivariate time series and the problem parameters. It then returns the clustering assignments of each point in the time series, along with the structural MRF representation of each cluster. We test TICC on several synthetic examples. We do so because there are known "ground truth" clusters to evaluate the accuracy of our method.

**Generating the Datasets.** We randomly generate synthetic multivariate data in $\mathbf{R}^5$. The overall time series is generated by constructing a temporal sequence of cluster segments (for example, the sequence "$1, 2, 1$" with 200 samples in each of the three segments, coming from two inverse covariances $\Theta_1$ and $\Theta_2$). The data is then drawn one sample at a time, conditioned on the values of the previous $w-1$ samples. We run our experiments on four different temporal sequences. Each segment in each of the examples has $100K$ observations in $\mathbf{R}^5$, where $K$ is the number of clusters in that experiment. These examples were selected to convey various types of temporal sequences over various lengths of time.

[1]Available at http://snap.stanford.edu/ticc/.

| | Clustering Method | Temporal Sequence 1,2,1 | 1,2,3,2,1 | 1,2,3,4,1,2,3,4 | 1,2,2,1,3,3,3,1 |
|---|---|---|---|---|---|
| | **TICC** | **0.92** | **0.90** | **0.98** | **0.98** |
| Model-Based | TICC, $\beta = 0$ | 0.88 | 0.89 | 0.86 | 0.89 |
| | GMM [Banfield and Raftery, 1993] | 0.68 | 0.55 | 0.83 | 0.62 |
| | EEV [Fraley and Raftery, 2006] | 0.59 | 0.66 | 0.37 | 0.88 |
| Distance-Based | DTW, GAK [Cuturi, 2011; Sarda, 2016] | 0.64 | 0.33 | 0.26 | 0.27 |
| | DTW, Euclidean [Sarda, 2016] | 0.50 | 0.24 | 0.17 | 0.25 |
| | Neural Gass [Dimtriadou, 2009] | 0.52 | 0.35 | 0.27 | 0.34 |
| | K-means | 0.59 | 0.34 | 0.24 | 0.34 |

Table 1: $F_1$ score of clustering accuracy for four different temporal sequences, comparing TICC with several alternative model and distance-based methods.

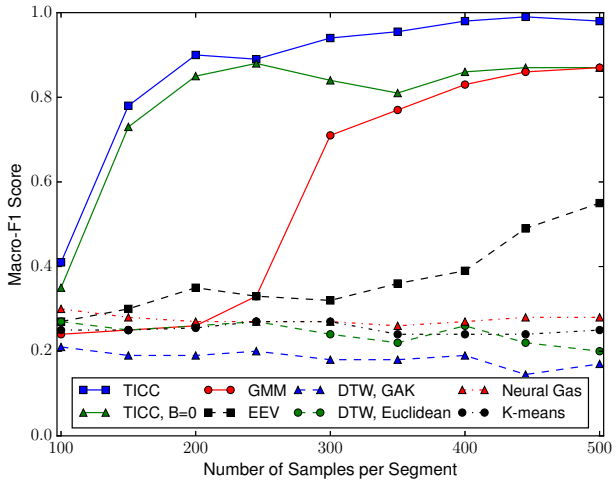



Figure 2: $F_1$ score vs. number of samples for TICC and several baselines. TICC needs significantly fewer samples than the other model-based methods to achieve similar performance, while the distance-based measures are unable to capture the true structure.

**Clustering Accuracy.** Since both TICC and the baseline approaches use very similar methods for selecting the appropriate number of clusters, we fix $K$ to be the "true" number of clusters, for both TICC and for all the baselines. We measure the macro-$F_1$ score for the four different temporal sequences in Table 1. Here, all eight methods are using the exact same synthetic data, to isolate each approach's effect on performance. As shown, TICC significantly outperforms the baselines. Our method achieves a $F_1$ score between $0.90$ and $0.98$, averaging $0.95$ across the four examples. This is $41\%$ higher than the second best method (not counting TICC, $\beta = 0$).

**Effect of the Total Number of Samples.** We next focus on how many samples are required for each method to accurately cluster the time series. We take the "1,2,3,4,1,2,3,4" example from Table 1 and vary the number of samples. We plot the $F_1$ score vs. number of samples per segment in Figure 2. As shown, when there are 100 samples, none of the methods are able to accurately cluster the data. However, as we observe more samples, both TICC and TICC, $\beta = 0$ improve rapidly. By the time there are 200 samples, TICC already has an $F_1$ score above 0.9. Even when there is a limited amount of data, TICC is still able to accurately cluster the data. As the number of samples increases, TICC's $F_1$ score goes to 1.0, but no other method (including TICC, $\beta = 0$) tops 0.9. We note that

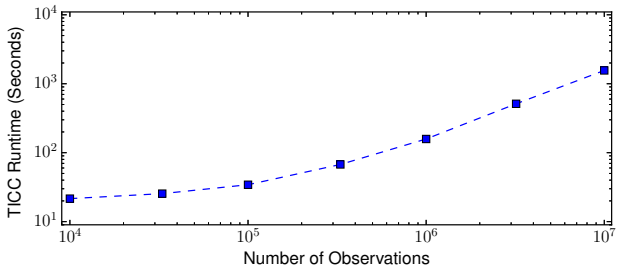



Figure 3: Per-iteration runtime of the TICC algorithm (both the ADMM and dynamic programming steps).

even with 500 samples, the distance-based methods struggle to identify the clusters.

**Scalability of TICC.** One iteration of the TICC algorithm consists of running the dynamic programming algorithm and then solving the Toeplitz graphical lasso problem for each cluster. These steps are repeated until convergence. The total number of iterations depends on the data, but typically is no more than a few tens of iterations. To evaluate the scalability of our algorithm, we vary the number of timestamps and compute the runtime of the algorithm over one iteration. We observe samples in $\mathbf{R}^{50}$, estimate 5 clusters with a window size of 3, and vary $T$ over several orders of magnitude. We plot the results in Figure 3. Note that our ADMM solver is independent of $T$, so this contributes to the constant offset in the plot. However, for large values of $T$, our algorithm scales linearly with the number of points. Our TICC solver can cluster 10 millions points, each in $\mathbf{R}^{50}$, with a per-iteration runtime of approximately 25 minutes.

## 5 Case Study

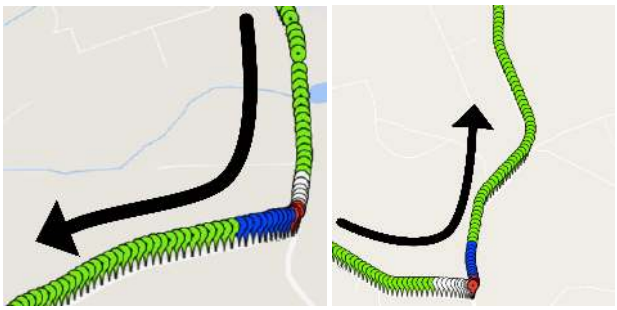
Here, we apply our TICC method to a real-world example to demonstrate how this approach can be used to find meaningful insights from time series data in an unsupervised way. We analyze a dataset, provided by a large automobile company, containing sensor data from a real driving session. This session lasts for exactly 1 hour and occurs on real roads in the suburbs of a large European city. We observe 7 sensors every 0.1 seconds:

- Brake Pedal Position
- Forward (X-)Acceleration
- Lateral (Y-)Acceleration
- Steering Wheel Angle
- Vehicle Velocity
- Engine RPM
- Gas Pedal Position

We run TICC and segment the time series into 5 clusters. We then analyze the clusters to understand and interpret what "driving state" they each refer to. Each cluster has a multilayer MRF network defining its structure. To analyze the result, we use network analytics to determine the relative "importance" of each node in the cluster's network. We plot the betweenness centrality score [Brandes, 2001] of each node in Table 2. We see that each of the 5 clusters has a unique "signature", and that different sensors have different betweenness scores in each cluster. For example, the Y-Acceleration sensor has a non-zero score in only two of the five clusters: #2 and #5. Therefore, we expect these two clusters to refer to

| | Interpretation | Brake | X-Acc | Y-Acc | SW Angle | Vel | RPM | Gas |
|---|---|---|---|---|---|---|---|---|
| #1 | Slowing Down | 25.64 | 0 | 0 | 0 | 27.16 | 0 | 0 |
| #2 | Turning | 0 | 4.24 | 66.01 | 17.56 | 0 | 5.13 | 135.1 |
| #3 | Speeding Up | 0 | 0 | 0 | 0 | 16.00 | 0 | 4.50 |
| #4 | Driving Straight | 0 | 0 | 0 | 0 | 32.2 | 0 | 26.8 |
| #5 | Curvy Road | 4.52 | 0 | 4.81 | 0 | 0 | 0 | 94.8 |

Table 2: Betweenness centrality for each sensor in each of the five clusters. This score can be used as a proxy to show how "important" each sensor is, and more specifically how much it directly affects the other sensor values.



(a) (b)

Figure 4: Two real-world turns in the driving session. The pin color represents cluster assignment from our TICC algorithm (Green = Going Straight, White = Slowing Down, Red = Turning, Blue = Speeding up).

states in which the car is turning, and the other three to refer to intervals where the car is going straight. As such, we can use these betweenness scores to interpret these clusters in a meaningful way. For example, from Table 2, a reasonable hypothesis might be that the clusters refer to 1) slowing down, 2) turning, 3) speeding up, 4) cruising straight, 5) driving on a curvy road segment.

**Plotting the Resulting Clusters.** To validate our hypotheses, we can plot the latitude/longitude locations of the drive, along with the resulting cluster assignments. Analyzing this data, we empirically discover that each of the five clusters has a clear real-world interpretation that aligns very closely with our estimates based on the betweenness scores in Table 2. Furthermore, we notice that many consistent and repeated patterns emerge in this one hour session. For example, whenever the driver is approaching a turn, he or she follows the same sequence of clusters: going straight, slowing down, turning, speeding up, then going straight again. We plot two typical turns in the dataset, coloring the timestamps according to their cluster assignments, in Figure 4.

## 6 Conclusion and Future Work

In this paper, we have defined a method of clustering multivariate time series subsequences. Our method, Toeplitz Inverse Covariance-based Clustering (TICC), simultaneously segments and clusters the data, breaking down high-dimensional time series into a clear sequential timeline. TICC's promising results on both synthetic and real-world data lead to many potential directions for future research. For example, our method could be extended to learn dependen-

cies parameterized by *any* heterogeneous exponential family MRF. This would allow for a much broader class of datasets (such as boolean or categorical readings) to be incorporated into TICC.

## Note

This is an abridged version of the full paper, originally published in *ACM SIGKDD*, 2017.

## References


[Banerjee *et al.*, 2008] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR*, 2008.

[Banfield and Raftery, 1993] Jeffrey D Banfield and Adrian E Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 1993.

[Begum *et al.*, 2015] Nurjahan Begum, Liudmila Ulanova, Jun Wang, and Eamonn Keogh. Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In *KDD*, 2015.

[Berndt and Clifford, 1994] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *AAAI Workshop on Knowledge Disovery in Databases*, 1994.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2011.

[Brandes, 2001] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 2001.

[Cuturi, 2011] Marco Cuturi. Fast global alignment kernels. In *ICML*, 2011.

[Dimtriadou, 2009] Evgenia Dimtriadou. Cclust: Convex clustering methods and clustering indexes. https://CRAN.R-project.org/package=cclust, 2009.

[Fraley and Raftery, 2006] Chris Fraley and Adrian E Raftery. MCLUST version 3: an R package for normal mixture modeling and model-based clustering. Technical report, DTIC Document, 2006.

[Friedman *et al.*, 2008] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2008.

[Hallac *et al.*, 2016] David Hallac, Peter Nystrup, and Stephen Boyd. Greedy Gaussian segmentation of multivariate time series. *arXiv preprint arXiv:1610.07435*, 2016.

[Himberg *et al.*, 2001] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikänmaki, and Hannu TT Toivonen. Time series segmentation for context recognition in mobile devices. In *ICDM*, 2001.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[Lauritzen, 1996] Steffen L Lauritzen. *Graphical models*. Clarendon Press, 1996.

[Miyajima *et al.*, 2007] Chiyomi Miyajima, Yoshihiro Nishiwaki, Koji Ozawa, Toshihiro Wakita, Katsunobu Itou, Kazuya Takeda, and Fumitada Itakura. Driver modeling based on driving behavior and its evaluation in driver identification. *Proceedings of the IEEE*, 2007.

[Mörchen *et al.*, 2005] Fabian Mörchen, Alfred Ultsch, and Olaf Hoos. Extracting interpretable muscle activation patterns with time series knowledge mining. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 2005.

[Namaki *et al.*, 2011] A Namaki, AH Shirazi, R Raei, and GR Jafari. Network analysis of a financial market based on genuine correlation and threshold method. *Physica A: Stat. Mech. Apps.*, 2011.

[Parikh and Boyd, 2014] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 2014.

[Rue and Held, 2005] Havard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. CRC Press, 2005.

[Sarda, 2016] Alexis Sarda. Dtwclust. https://cran.r-project.org/web/packages/dtwclust/index.html, 2016.

[Smyth, 1997] Padhraic Smyth. Clustering sequences with hidden Markov models. *NIPS*, 1997.

[Viterbi, 1967] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 1967.

[Wainwright and Jordan, 2006] Martin J Wainwright and Michael I Jordan. Log-determinant relaxation for approximate inference in discrete Markov random fields. *IEEE Tr. on Signal Processing*, 2006.

[Wytock and Kolter, 2013] Matt Wytock and J Zico Kolter. Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. *ICML*, 2013.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, 2006.