Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data (TICC)

David Miller

CIS5930: Social Network Mining
Florida State University
February 28, 2018

Motivation

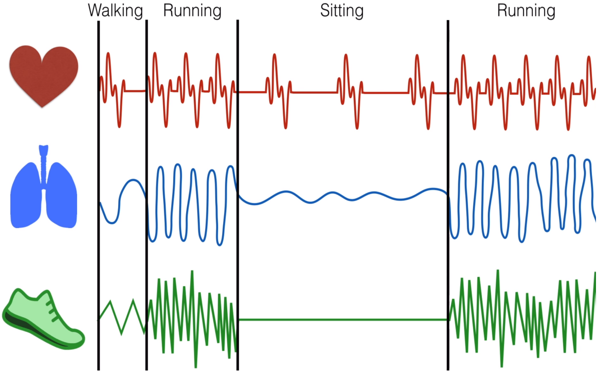
Many things generate large amounts of time series data, most of which is multivariate

- Financial markets
- Wearable sensors
- Automobiles

Long term series can be broken down into a sequence of states, each defined by a simple "pattern", where the states can occur multiple times

- Buy, sell, hold, high volume trading, ...
- Resting, walking, running, ...
- Turning, accelerating, speeding up , ...

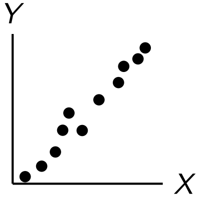
Motivation: Example



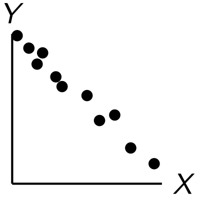
Temporal data of heart rate, oxygen usage, and speed with their respective activity (sitting, walking, running) clustering

Background: Covariance

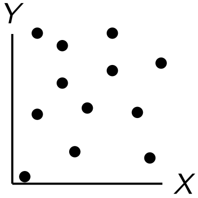
The **covariance** between two variables is positive when they tend to move in the same direction and negative if they tend to move in opposite directions



Positive covariance



Negative covariance



Zero covariance

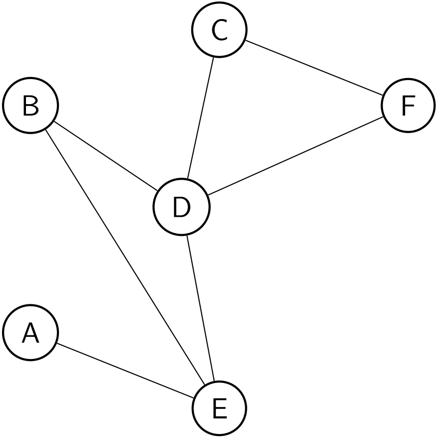
Background: Markov Random Field (MRF)

A **Markov Random Field** is an undirected graph $G = (V, E)$ such that V are random variables and

- If $(v_i, v_j) \notin E$ then random variables i and j are conditionally independent given $V \setminus \{v_i, v_j\}$
- Random variable i is conditionally independent of random variable j if $d(v_i, v_j) > 1$ given all v_j s.t. $d(v_i, v_j) = 1$
- $A = \{v_1, \ldots, v_n\}$ is conditionally independent of $B = \{v_1, \ldots, v_m\}$ given some set S such that every path from a node in A to a node in B passes through S

Background: Markov Random Field (MRF)

A depends on E
B depends on D and E
C depends on D and F
D depends on B, C, E, and F
E depends on A, B, and D
F depends on C and D



Background: Toeplitz Matrix

A **Toeplitz Matrix** is a matrix such that each descending diagonal from left to right is constant. Let A be a $n \times n$ Toeplitz matrix, then A takes on the form

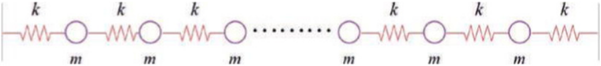
$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_2 & a_1 & a_0 \end{bmatrix}$$

where the i, j element $A_{i,j} = A_{i+1,j+1} = a_{i-j}$

Background: Inverse Covariance Matrix

The **inverse covariance matrix** essentially models the dependency, or relation, of variables with their neighbors. Take for example the multiple mass-spring problem

$$m\ddot{x}_i = -k(x_i - x_{i-1}) - k(x_{i+1} - x_i) + w_i$$



where x_i is displacement of m_i and w_i is external force acting on m_1. This can be written in matrix form

$$m \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_n \end{bmatrix} = k \underbrace{\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \\ 0 & \dots & \dots & 1 & -2 \end{bmatrix}}_{\text{Inverse covariance matrix}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

Problem Statement

Given a time series of T sequential observations

$$x_{orig} = \begin{bmatrix} | & | & | & & | \\ x_1 & x_2 & x_3 & \dots & x_T \\ | & | & | & & | \end{bmatrix}$$

where $x_i \in \mathbb{R}^n$ is the i-th multivariate observation, **the goal is to cluster these $\boldsymbol{T}$ observations into $\boldsymbol{K}$ clusters**. The dimension n is typically the number of sensors of the data.

Problem Statement

Instead of clustering each observation x_i in isolation, each point is treated in context of its previous $w-1$ predecessors, where $w \ll T$. We then have

$$X_t := \begin{bmatrix} x_{t-w+1} \\ x_{t-w+2} \\ \vdots \\ x_{t-1} \\ x_t \end{bmatrix}$$

where $X_t \in \mathbb{R}^{nw}$. Let the sequence $X_1, \ldots, X_T$ be referred to as X. Now **the goal is to cluster these subsequences $\boldsymbol{X_1}$, ..., $\boldsymbol{X_T}$**. The mapping $f : x_{orig} \to X$ is a bijection.

Toeplitz Inverse Covariance-Based Clustering (TICC)

Each cluster is defined by a Gaussian inverse covariance $\Theta_i \in \mathbb{R}^{nw \times nw}$. These inverse covariances show the conditional independency structure between the variables that define a MRF encoding the structural representation of each cluster [2].

The objective is to solve for these K inverse covariances $\mathbf{\Theta} = \{\Theta_1, \ldots, \Theta_K\}$ and assignment sets $\boldsymbol{P} = \{P_1, \ldots, P_k\}$ via the optimization problem, or the TICC problem.

Toeplitz Inverse Covariance-Based Clustering (TICC)

The TICC problem is

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

where $\mathcal{T}$ is the set of symmetric block Toeplitz $nw \times nw$ matrices, $\|\lambda \circ \Theta_i\|_1$ is an ℓ_1-norm penalty to incentivize a sparse inverse covariance, and $\mathbb{1}\{X_t - 1 \notin P_i\}$ is an indicator function checking whether neighboring points are assigned to the same cluster. Additionally, $\ell\ell(X_t, \Theta_i)$ is the log likelihood that X_i came from cluster i,

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

where μ_i is the empirical mean of cluster i.

Regularization Parameters

The TICC optimization problem has two regularization parameters

- β: Smoothness penalty that encourages adjacent subsequences to be assigned to the same cluster.
- λ: Determines the sparsity of level in the MRFs characterizing each cluster. Although it is a $nw \times nw$ matrix, all its values are typically set to a single value to reduce the search space to one parameter.

Parameters can be manually set if given prior knowledge or determined through some method, such as Bayesian information criterion (BIC).

A Note on the Inverse Covariances

The inverse covariances Θ_i's are constrained to be block Toeplitz, thus can be expressed in the form

$$\Theta_i = \begin{bmatrix} A^{(0)} & (A^{(1)})^T & (A^{(2)})^T & \cdots & \cdots & (A^{(w-1)})^T \\ A^{(1)} & A^{(0)} & (A^{(1)})^T & \ddots & & \vdots \\ A^{(2)} & A^{(1)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & (A^{(1)})^T & (A^{(2)})^T \\ \vdots & & \ddots & A^{(1)} & A^{(0)} & (A^{(1)})^T \\ A^{(w-1)} & \ddots & \ddots & A^{(2)} & A^{(1)} & A^{(0)} \end{bmatrix}$$

where $A^{(0)}, A^{(1)}, \ldots, A^{(w-1)} \in \mathbb{R}^{n \times n}$.

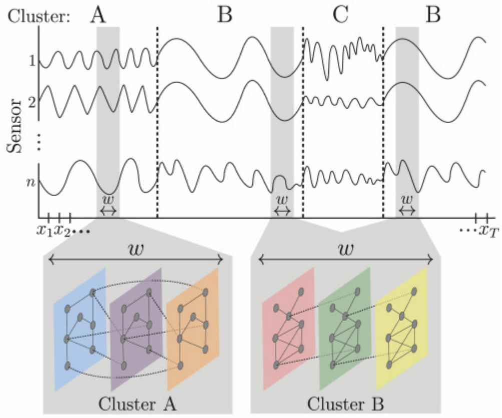
A Note on the Inverse Covariances

Diagonal blocks $A^{(0)}$ represent the intra-time partial correlations.

Off diagonal block $A_{jk}^{(n)} \in \Theta_i$ shows how sensor i at some time t is correlated to sensor j at time $t + n$.

The block Toeplitz structure of the inverse covariance means that we are making a time-invariance assumption over the length-w window. That is, if edges between layer m and layer n, where $m < n$, also exists in layers $n + (n - m), n + 2(n - m), \ldots$.

Problem: Example



Edges across layers within the same time window are time invariant and need not be a distance of 1 apart.

Expectation-Maximization (EM)

The TICC problem is a mixed combinatorial and continuous optimization problem. The cluster assignments P and inverse covariences Θ coupled together make the problem highly non-convex. As such, there is no tractable way to solve for the globally optimal solution.

A variation of the EM algorithm is used to alternate between assigning points to clusters and then updating the cluster parameters.

Assigning Points to Clusters

Points are assigned to clusters by fixing Θ and solving the following optimization problem for $\boldsymbol{P} = \{P_1, \ldots, P_K\}$

$$\text{minimize} \sum_{i=1}^{K} \sum_{X_t \in P_i} -\ell\ell(X_t, \Theta_i) + \beta \mathbb{1}\{X_{t-1} \notin P_i\} \tag{2}$$

This problem assigns each X_t subsequence to one of the K clusters to jointly maximize the log likelihood and the temporal consistency, with the tradeoff between the two objectives being regulated by β.

If $\beta = 0$ then $X_1, \ldots, X_t$ can be assigned independently since there is no penalty to encourage neighboring subsequences to belong to the same cluster. As $\beta \to \infty$, switching penalty becomes so large that all X_t are grouped into just one cluster.

Toeplitz Graphical Lasso

Given $\boldsymbol{P}$ from EM, the cluster parameters $\Theta_1, \ldots, \Theta_K$ are updated by solving the TICC problem while holding $\boldsymbol{P}$ constant. After some recasting the cluster parameters can be solved in parallel via

$$\begin{aligned} \text{minimize} \quad & -\log\det\Theta_i + \operatorname{tr}(S_i\Theta_i) + \frac{1}{|P_i|}\|\lambda \circ \Theta_i\|_1 && (3) \\ \text{subject to} \quad & \Theta_i \in \mathcal{T} && (3.1) \end{aligned}$$

where $|P_i|$ is the number of points assigned to cluster i, and S_i is the empirical covariance of these points. This optimization problem is the Toeplitz graphical lasso.

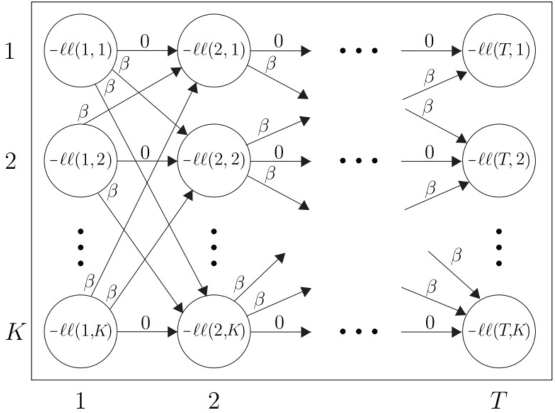
Cluster Assignment

Given K potential cluster assignments of the T points, this combinatorial optimization problem has K^T possible assignments of points to clusters. Cluster assignment are solved in $\mathcal{O}(KT)$ using dynamic programming approach that is equivalent to finding the minimum cost Viterbi path.

Algorithm 1 Assign Points to Clusters

given $\beta > 0$, $-\ell\ell(i, j)$ = negative log likelihood of point i when it is assigned to cluster j.
initialize PrevCost = list of K zeros.
  CurrCost = list of K zeros.
  PrevPath = list of K empty lists.
  CurrPath = list of K empty lists.
for $i = 1, \ldots, T$ **do**
  for $j = 1, \ldots, K$ **do**
    MinIndex = index of minimum value of PrevCost.
    if PrevCost[MinIndex] + β > PrevCost[j] **then**
      CurrCost[j] = PrevCost[j] $-\ell\ell(i, j)$.
      CurrPath[j] = PrevPath[j].append[j].
    else
      CurrCost[j] = PrevCost[minIndex] + $\beta - \ell\ell(i, j)$.
      CurrPath[j] = PrevPath[minIndex].append[j].
  PrevCost = CurrCost.
  PrevPath = CurrPath.
FinalMinIndex = index of minimum value of CurrCost.
FinalPath = CurrPath[FinalMinIndex].
return FinalPath.

Cluster Assignment



Problem 3 is equivalent to finding the minimum cost path from timestamp 1 to T, where the node cost is the negative log likelihood of that point being assigned to a cluster, and the edge cost is β whenever the cluster assignment switches.

Solving the Toeplitz Graphical Lasso

Alternating direction method of multipliers (ADMM) and augmented Lagrangian are used to efficiently solve the Toeplitz graphical lasso. First problem 3 is recast into ADMM form

$$\begin{array}{ll} \text{minimize} & -\log\det\Theta_i + \mathrm{tr}(S_i\Theta_i) + \dfrac{1}{|P_i|}\|\lambda \circ Z\|_1 \\ \text{subject to} & \Theta_i = Z, Z \in \mathcal{T} \end{array}$$

The augmented Lagrangian can then be expressed as

$$\mathcal{L}_\rho(\Theta, Z, U) = -\log\det(\Theta) + \mathbf{Tr}(S\Theta) + \|\lambda \circ Z\|_1 + \frac{\rho}{2}\|\Theta - Z + U\|_F^2$$

where $\rho > 0$ is the ADMM penalty parameter and $U \in \mathbb{R}^{nw \times nw}$ is the scalable dual variable.

The TICC Algorithm

The algorithm consists of the following three steps repeated until convergence

$$
\begin{aligned}
(a) \quad & \Theta^{k+1} := \arg\min_{\Theta} \mathcal{L}_\rho(\Theta, Z^k, U^k) \\
(b) \quad & Z^{k+1} := \arg\min_{Z \in \mathcal{T}} \mathcal{L}_\rho(\Theta^{k+1}, Z, U^k) \\
(c) \quad & U^{k+1} := U^k + (\Theta^{k+1} - Z^{k+1})
\end{aligned}
$$

where k is the iteration number. The stopping criterion is for the algorithm is Residual$(\Theta, Z, U) < \epsilon$ for some arbitrarily small ϵ. Solving (b) requires solving $(w-1)n^2 + \frac{n(n-1)}{2}$ subproblems, which can all be done in parallel. Problem (a) has known analytic solution, see [1] for solution to (b).

The TICC Algorithm

Algorithm 2 Toeplitz Inverse Covariance-Based Clustering

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

The TICC algorithm can be broken down into two key parts: *i*) solving the cluster assignment problem and then *ii*) solving for cluster inverse covariances until convergence.

Implementation

A custom Python solver is built to run the TICC algorithm.

- Input: Original multivariate time series and problem parameters
- Output: The clustering assignments of each point in the time series, along with the structural MRF representation of each cluster

TICC is tested on several synthetic examples because there are "ground truth" clusters to evaluate the accuracy of the method.

Generating Datasets

Synthetic multivariate data in $\mathbb{R}^5$ is randomly generated. Each of the K clusters has a mean $\vec{0}$ so that the clustering result is based entirely on the structure of the data. For each cluster, random ground truth Toeplitz iverse covariance is generated as

- Set $A^{(0)}, \ldots, A^{(4)}$ equal to adjacency matrices of 5 independent Erdős-Rényi where each edge has a 20% of being selected
- For every selected edge in $A^{(m)}$ set $A_{jk}^{(m)} = v_{jk,m}$ a random weight centered at 0
- Construct a $5w \times 5w$ block Toeplitz matrix G, where window size $w = 5$, using the blocks $A^{(0)}, \ldots, A^{(4)}$
- Let c be the smallest eigenvalue of G and set $\Theta_i = G + 0.1 + |c|I$. This ensures that Θ_i is invertible

Generating Datasets

The overall time series is then generated by constructing a temporal sequence of cluster segments (for example, the sequence "1,2,1" with 200 samples in each of the segments, coming from two inverse covariances Θ_1 and Θ_2).

Experiments are run on four different temporal sequences: "1,2,1", "1,2,3,2,1", "1,2,3,4,1,2,3,4", "1,2,2,1,3,3,3,1". Each segment in each of the examples has $100K$ observations in $\mathbb{R}^5$, where K is the number of clusters in that experiment (2, 3, 4, and 3, respectively). K is fixed to be the "true" number of clusters for both TICC and the baseline methods.

Baseline Methods

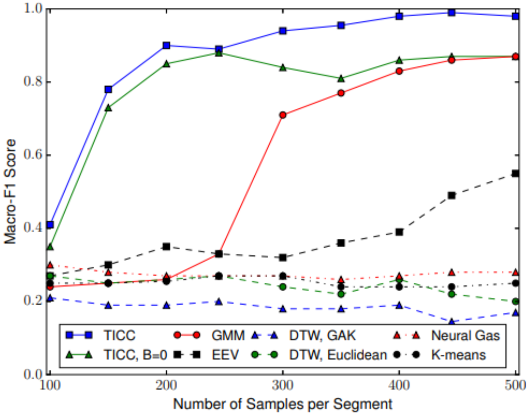
- TICC, $\beta = 0$: TICC and TICC without temporal consistency constraint
- GMM: Clustering using a Gaussian Mixture Model
- EEV: Regularized GMM with shape and volume constraints on the Gaussian covariance matrix
- DTW, GAK: Dynamic time warping-based clustering using a global alignment kernel
- DTW, Euclidean: DTW using a Euclidean distance metric
- Nerual Gas: Artificial neural network clustering method based on self-organizing maps
- K-means: Standard K-means clustering algorithm using Euclidean distance

Results

Method	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**
TICC, $\beta = 0$	0.88	0.89	0.86	0.89
GMM	0.68	0.55	0.83	0.62
EEV	0.59	0.66	0.37	0.88
DTW, GAK	0.64	0.33	0.26	0.27
DTW, Euclid	0.50	0.24	0.17	0.25
Neural Gas	0.52	0.35	0.27	0.34
K-means	0.59	0.34	0.24	0.34

Macro-F_1 scores of clustering accuracy for four different temporal sequences.

Results



Clustering accuracy macro-F_1 score vs number of samples. TICC needs significantly fewer samples than other methods to achieve similar performance.

Case Study

TICC is applied to a large dataset from a one-hour driving session where 7 sensors are observed every 0.1 seconds

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

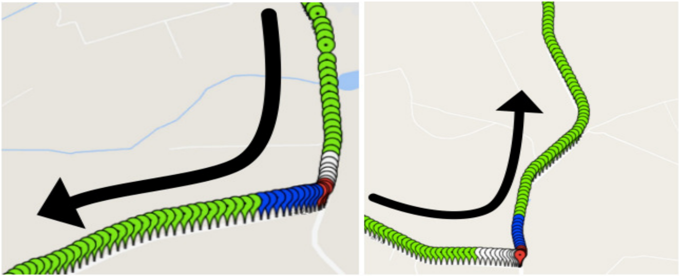
TICC is applied with $w = 10$ (1 second window size). Number of clusters picked using BIC, then discovered that $K = 5$ is optimal.

Case Study

	Brake	X-Acc	Y-Acc	SW Angle	Vel	RPM	Gas
Slow Down	25.64	0	0	0	27.16	0	0
Turning	0	4.24	66.01	17.56	0	5.13	135.1
Speed Up	0	0	0	0	16.00	0	4.50
Drive Straight	0	0	0	0	32.2	0	26.8
Curvy Road	4.52	0	4.81	0	0	0	94.8

Betweenness centrality for each sensor in each of the five clusters. This score can be used as a proxy to show the "importance" of each sensor is and how it directly affects other sensor values

Case Study



Pins represent cluster assingments. The color clusters are Green = Going Straight White = Slowing Down, Red = Turning, Blue = Speeding Up.

References

- David Hallac, Sagar Vare, Stephen Boyd, Jure Leskovec. *Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data.* KDDâĂŹ17, August 13âĂŞ17, 2017, Halifax, NS, Canada
- D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT press, 2009.