



2021-2022 秋季学期创新实践I 第二次报告

一种动态网络推断方法的实现

吕昊泽 11912814

张子研 11912324

指导教师：唐珂教授

2021.12.2

1 进展描述

1.1 前提描述

在开题报告中，我们简单介绍了TVGL方法的研究背景：即基于协方差矩阵的时序网络推导。所谓网络推导，即是通过数学等手段对于自然界或者人类社会中存在的复杂网络进行结构分析，尝试找出复杂网络各个节点之间的内在联系，用于进一步的分析和研究。而所谓时序网络，则是考虑到了现实生活中网络的结构网随着时间的推进发生变换。时序网络，就是在传统网络的分析中，将时间尺度纳入了考量，通过假定时间尺度下前后网络的相似性，对于一个复杂网络内部结构随时间的变化进行了推导。而我们所研究的TVGL方法，则是通过推导出复杂网络各个节点之间协方差矩阵的逆随时间的变换，得到复杂网络内部结构随时间的变换。

1.2 项目进展

在第二阶段的学习中，通过对论文进行更详细的阅读和研究，我们基本了解了TVGL所采用的 Loss Function 背后所蕴含的数学原理和这个问题近四十年的发展历程。随后，我们对开题报告时跑出的数据结果进行了更深层次的分析，与论文的结果进行更详细的对比，得到了更让人信服结论。

1.3 报告结构

本篇报告，关于TVGL问题定义会放在第二章进行讲解。对于TVGL的原型静态网络推断背后的理论将会放在第三阶段讲解。对于TVGL对于时间维度的拓展和时间部分用到的惩罚函数将会在第四部分进行讲解，关于TVGL方法在ADMM上的应用在第五部分讲解。关于更细致的论文结果实验将会在第六部分呈现。

2 问题定义

2.1 过往研究

考虑一个有 p 个节点组成的系统，我们在每一时刻 t_i 都有对于该系统的观察值 $x_i \in R^p$ 。通过这些观察数据，我们想得到系统各个节点之间的内在联系。在考虑这个问题之前，我们先考虑这个问题的简单形式，即我们有关于一个 p 个节点的系统的 n 个观察数据，我们想通过这个 n 个观察数据推断出网络之间的内在结构。首先，我们定义这个问题的观察变量都是遵循分布 $x \sim N(0, \Sigma(t))$ 的，这样这个问题就变成了一个多元统计分析问题，即网络中每个节点便是一个元，我们有他们的观察变量，我们就可以通过推断网络的协方差矩阵来估计网络。而A.P.Dempster在1972年提出，推断多元正态分布的协方差矩阵的问题可以简化成对协方差矩阵的逆矩阵的推断和其中0的分析[1]。作者在论文中通过数学证明了推断协方差矩阵的过多参数会严重影响推断效果，并证明了协方差矩阵关系为0，不比证明协方差矩阵的逆为0更可靠，但是后者的理论优势让它在缺少先验经验的时候更应该被考虑。因此，对于一个服从正态分布的 p 个节点的系统内部结构的推导，变转换成了对其协方差矩阵的逆进行推导的过程。而推导的具体方法将会在第三部分进行讲解。

2.2 TVGL定义

TVGL方法将该问题重新定义为，从分布 $x \sim N(0, \Sigma(t))$ 采样的 R^p 中一个多元观测序列，其中在 t_i $i \in 1 \dots m$ 时刻，都有 $n_i \geq 1$ 个不同的观测向量 $x_i \in R^p$ 作为节点的观察数据输入网络，即 t_i 时刻网络的输入 $p * n$ $n = t_i - t_{i-1}$ ，通过这个输入构建各个 t_i 时刻的协方差矩阵的逆矩阵，通过协方差矩阵的逆矩阵是否为零得到变量之间的相关性。注意到，这里TVGL方法将问题的观察值假定为满足正态分布的观察序列，并且将原本的时刻 t_i 的意义从单个时间点，变为 n 个单个时间点构成的时间节点，也就是将整个时间序列切片成 m 个相互相关的时间节点，从而得到 n 个 x_i $i \in 1 \dots n$ 的观察值组成矩阵作为方法的输入。方法以静态网络推断作为基础，首先对于 m 个时间节点的协方差矩阵进行单独考虑，得到其独立的推断值 θ_i ，用来反映当前时间节点的系统潜在关系。但是为了将系统在时间上的前后依赖性纳入模型的考虑范围，TVGL方法加入了惩罚函数 ψ ，通过对于 θ 变化模式的惩罚，从而将时序网络时间维度的相关性纳入模型考虑，并可根据不同的惩罚函数适应不同的系统。

3 静态网络推断

3.1 协方差矩阵的逆

在具体分析TVGL方法对于时序网络的分析之前，我们现在详细分析一下对于单个时间节点，常规方法如何通过观察数据得到当前时间节点的协方差矩阵的逆的。

首先，我们考虑分布 $G(x_t|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \exp(-\frac{1}{2}(x_t - \mu)^T \Sigma^{-1}(x_t - \mu))$ 的分布，根据极大似然估计定理，我们对其似然函数求导，可得 μ 和 Σ 的极大似然估计 $\mu = \frac{1}{m} \sum_{t=1}^m x_t$ 和 $\mu = \frac{1}{m} \sum_{t=1}^m (x_t - \mu)(x_t - \mu)^T$ 。而对于我们所研究的问题，由于观察数据被认为是中心化的，所以 μ 被省略，可以得到 $\mu = \frac{1}{m} \sum_{t=1}^m x_t x_t^T$ ，被称为经验协方差。此时，如果我们有 $n > p$ ，那么通常我们推断出来的经验协方差便是可逆的，我们可以对其求逆得到我们所要求的 Σ^{-1} 。

3.2 协方差矩阵的逆的估计

但是，在大多数实际的观察问题中，我们所得到的观察量 n 是远远小于网络中的节点数目 p 的，这时我们得到的经验协方差便是不可逆的，因此，我们需要别的方法对 Σ^{-1} 进行估计。许多论文通过最大化对数似然函数的方法对 Σ^{-1} 进行了估计[2, 3]，TVGL论文中静态网络部分的方法也与其相同，但是更改了算式的符号。

TVGL论文中给出的公式为 $\text{minimize} -l(\Theta) + \lambda \|\Theta\|_{od,1}$ 。其中 $\Theta_i = \Sigma(a)^{-1}$
 $l(\Theta) = n(\log \det \Theta - \text{Tr}(S\Theta))$

在Ming Yuan的论文中，我们找到了该方法的推导原理和特性[2]。

首先对于分布的对数似然函数的原型如下 $\frac{n}{2} \log |C| - \frac{1}{2} \sum_{i=1}^n (x_i) C (x_i)^T$ (由于我们的问题分布是中心化的，所以我们省略了算式中 μ 的部分)

对于原公式，我们可以通过数学变形将其重写为 $-\log |C| + \text{tr}(CS)$ $\text{subject to } \sum_{i=1}^n c_{ij} \leq t$ 。可以注意到，算式中 C 为我们所求项， S 为前文提到的经验协方差。论文中提到，由于该算式是凸的，所以我们可以将该算式进一步变形为其拉格朗日形式 $-\log |C| + \text{tr}(CS) + \mu \sum_{i=1}^n |c_{ij}|$ 。在此式中， $C = \Sigma^{-1}$ 。自此，我们得到了与TVGL论文中相同的公式。后面根据极大似然估计的定理，我们只要求 C ，让 C 使得该公式最小化，就可以在 $n < p$ 的条件下得到对于 Σ^{-1} 的最大似然估计。

论文中还用数学方法证明了求得得 Σ^{-1} 一定是正定的，同时论文还指出了该方法估计出来的 Σ^{-1} 是稀疏的，这与其拉格朗日形式引入的 $l1$ 范数有关

至此，我们大概了解了TVGL方法依据的对于静态网络推断的方法，接下来，我们将介绍TVGL方法是如何将其拓展到时序范围内的。

4 时序网络和其惩罚函数

4.1 推广到时间维度的公式

为了考虑系统时间范围内的相关性，TVGL方法将构成静态网络的各个时间节点连接起来，将模型推广到整个 T 时间内。TVGL方法的作者认为，时序网络前后时间节点的网路是存在相似性的。因此，TVGL方法在静态网络推断公式的基础上，加入了新的惩罚函数 ψ 来引入前后时间节点 Θ 相似性的考量

公式： $\text{minimize } \Theta \in S_{++}^P - l(\Theta) + \lambda \|\Theta\|_{od,1} + \beta \sum_{i=2}^T \psi(\Theta_i - \Theta_{i-1})$ ，其中 $l(\Theta) = n(\log \det \Theta - \text{Tr}(S\Theta))$

加入的惩罚函数 ψ 在 0 处取得极值，这使得 Θ_i 和 Θ_{i-1} 相似，因此加入的惩罚函数鼓励了前后时间节点网络的相似性。

在超参数的考量中：

- λ 决定了网络的稀疏程度 越小越稠密
- β 决定相邻协方差估计的强相关程度 越大越平稳

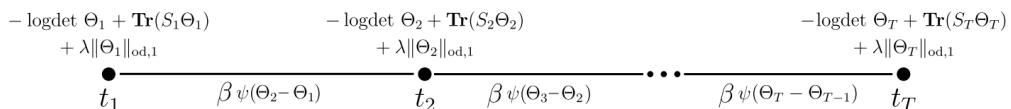


Figure 2: Dynamic network inference can be thought of as an optimization problem on a chain graph, where each node objective solves for a network slice at each timestamp, and edge objectives define the penalties that enforce temporal consistency.

4.2 不同的惩罚函数及其作用分析

论文定义了五种不同的惩罚函数，分别适配了关系网络不同的五种变化类型

- 1 每次改变几条边 $\psi(X) = \sum i, j |X_{i,j}|$
- 2 一次行大范围改动 $\psi(X) = \sum j \|X_j\|_2$
- 3 随着时间平滑变化 $\psi(X) = \sum i, j X_{i,j}^2$

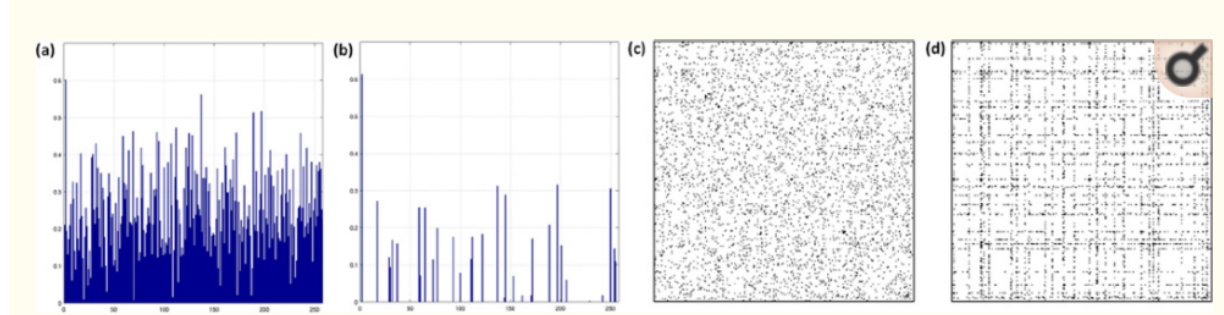
4 基于一个块的重组 $\psi(X) = \sum_j \max_i |X_{i,j}|$

5 扰动节点 $\psi(X) = \min_{V: V+V^T=X} \sum_j \|X_j\|_2$

在对论文的研究中我发现，这五种不同的惩罚函数实际上都来源于之前的研究。我在这里以惩罚函数5作为举例，展示出该惩罚函数的作用和TVGL方法对其的巧妙应用。

惩罚函数5，最早由K. Mohan, M. Chung, S. 等人在2014年提出[4]。其引入该惩罚项的初衷是在对癌症基因序列的探究中，为了探究肺癌患者和脑癌患者背后基因调控网络的特性而引入的。K,M等人认为，肺癌和脑癌患者背后的基因调控网络一定是相近的，只有少数基因发生了改变导致了不同的结果。而当时的方法只能做到使用两组数据估计一个网络，或者是分别估计两个不同的网络。这都不能很好的满足作者的需求，因此作者希望找到一种新的方法来联合两组观察数据，生成两个不同的网络，让他们在彼此相似的同时，允许某些结构化差异。

而作者引入的惩罚函数便是 $\psi(X) = \min_{V: V+V^T=X} \sum_j \|X_j\|_2$ 。作者在论文中提到，此种惩罚函数鼓励 $\theta_1 - \theta_2$ 是一些行和其对应列的集合，即网络中某些行和其对应的列满足非零而其他位置均为0。这也正符合扰动节点对网络代来的影响。



[Open in a separate window](#)

Figure 4

PNJGL with $q = 2$ and FGL were performed on the brain cancer data set corresponding to 258 genes in patients with Proneural and Mesenchymal subtypes. (a)–(b): $N P_j$ is plotted for each gene, based on (a) the FGL estimates and (b) the PNJGL estimates. (c)–(d): A heatmap of $\hat{\theta}^1 - \hat{\theta}^2$ is shown for (c) FGL and (d) PNJGL; zero values are in white, and non-zero values are in black.

而TVGL方法，巧妙地将其运用到了时间序列上，并将[4]中使用的两个网络扩展到了时间范围内的任意数量的网络中，其巧妙的将临近时间节点的网络用此惩罚函数相关联，实现对于某种特定网络变化的分析。

4.3 拓展到时间间隔不同的情况

之前的动态网络推断建立在对于时间序列的m分割是均匀的，其任何时刻的n均相同，但如果观察结果不能满足均匀切片，那么在各个时间节点内的n不同时

通过公式 $\min_{\Theta} \Theta \in S_{++}^P - l(\Theta) + \lambda \|\Theta\|_{od,1} + \beta \sum_{i=2}^T \psi(\frac{\Theta_i - \Theta_{i-1}}{h_i}) h_i = t_i - t_{i-1}$ ，用 h_i 消除不同时间长度对于模型的影响

5 以ADMM为基础的计算方法

由于TVGL网络致力于研究时序网络的内在结构，所以其面临同时求解多个协方差矩阵的逆的难题。作者指出，常规的内点法等方法以不适用于该问题，所以作者基于ADMM算法实现了问题的分块化，使得问题可以在计算机上并行计算，让问题的求解成为了可能。

5.1 近端算子及其应用

在阐述该问题如何转化成ADMM形式进行求解前，我们首先了解近端算子的定义。

近端算子是凸优化问题中求解非平滑，有约束问题的一种工具。Boyd等人给出了近端算子的定义[5]，

$$\text{prox}_{\eta f}(A) = \argmin_{X \in \mathbb{R}^{m \times n}} (f(X) + \frac{1}{2\eta} \|X - A\|_F^2)$$

并且对各种近端算子涉及的各种子问题的求解方法进行了系统性的说明。TVGL的作者巧妙地将近端算子这一工具运用到TVGL问题ADMM形式迭代公式的推导中。作者通过将TVGL问题ADMM形式不同变量对应的近端算子形式，最终得到了完整的迭代公式。

5.2 ADMM方法及其TVGL应用

ADMM算法是一种用来解决凸优化问题的算法，其具有优秀的收敛性和对偶上升法的可分解性，保证了计算机上并行计算的可能，所以是在2011年由boyd重新论证后广泛使用的一种算法。作者在对TVGL网络的求解过程中，同样将问题分解成ADMM形式，利用计算机并行计算来求解。

对于问题的求解，联想到优化函数由 $l(\theta_i)$, $\psi(\theta_i - \theta_{i-1})$ 两个函数组成，所以尝试对变量进行重新定义以构造ADMM形式

重新定义问题为

$$\begin{aligned} & \text{minimize } \sum_{i=1}^T -l_i(\Theta_i) + \lambda \|Z_{i,0}\|_{od,1} + \beta \sum_{i=2}^T \psi(Z_{i,2} - Z_{i-1,1}) \\ & \text{subject to } Z_{i,0} = \Theta_i, \Theta_i \in S_{++}^P \text{ for } i = 1, \dots, T \\ & (Z_{i-1,1}, Z_{i,2}) = (\Theta_{i-1}, \Theta_i) \text{ for } i = 2, \dots, T \end{aligned}$$

ADMM的一般形式为

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \implies L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

所以我们要指出，对于TVGL方法对应的ADMM形式中，一般形式中的 A, B, C 等矩阵分别为 $A = I, B = -I, c = 0$

对于该形式，根据ADMM形式，我们可以写出三个变量的迭代公式

$$\begin{aligned} (a) \quad & \Theta^{k+1} := \underset{\Theta \in S_{++}^P}{\operatorname{argmin}} \mathcal{L}_\rho(\Theta, Z^k, U^k) \\ (b) \quad & Z^{k+1} = \begin{bmatrix} Z_0^{k+1} \\ Z_1^{k+1} \\ Z_2^{k+1} \end{bmatrix} := \underset{Z_0, Z_1, Z_2}{\operatorname{argmin}} \mathcal{L}_\rho(\Theta^{k+1}, Z, U^k) \\ (c) \quad & U^{k+1} = \begin{bmatrix} U_0^{k+1} \\ U_1^{k+1} \\ U_2^{k+1} \end{bmatrix} := \begin{bmatrix} U_0^k \\ U_1^k \\ U_2^k \end{bmatrix} + \begin{bmatrix} \Theta^{k+1} - Z_0^{k+1} \\ (\Theta_1^{k+1}, \dots, \Theta_{T-1}^{k+1}) - Z_1^{k+1} \\ (\Theta_2^{k+1}, \dots, \Theta_T^{k+1}) - Z_2^{k+1} \end{bmatrix}. \end{aligned}$$

之后，作者分别对三个变量的迭代公式进行了进一步推导，将其转化成近端算子形式，最终得出

5.2.1 θ 迭代

根据定义，我们可以重写 θ 迭代公式为

$$\theta^{k+1} = \underset{\theta \in S_{++}^P}{\operatorname{argmin}} -\log \det(\theta_i) + \operatorname{Tr}(S_i \theta_i) + \frac{1}{2\eta} \|\theta_i - A\|_F^2 \quad \text{subject to } A = \frac{Z_{i,0}^k + Z_{i,1}^k + Z_{i,2}^k - U_{i,0}^k - U_{i,1}^k - U_{i,2}^k}{3} \text{ and } \eta = \frac{n_i}{3p}$$

根据近端算子定义，该迭代公式可以重写为

$$\theta^{k+1} = \operatorname{prox}_{\eta(-\log \det(\cdot) + \operatorname{Tr}(S_i \cdot))} \left(\frac{A + A^T}{2} \right)$$

作者根据近端算子的求解方法，解出迭代公式为

$$\theta_i^{k+1} := \frac{1}{2\eta^{-1}} Q(D + \sqrt{D^2 + 4\eta^{-1}I}) Q^T$$

根据该公式可以通过上一次迭代产生的结果得到本代 θ

该公式的求解方法实际上可以看成直接求导一阶导为0，移项后对对称矩阵做特征值分解，即可得到上述答案

5.2.2 Z 迭代

Z 的更新分为 Z_0 和 (Z_1, Z_2) 两个部分

Z_0 部分的更新，由于 Z_0 实际上是范数中的一个变量，所以根据近端算子的定义， Z_0 的迭代公式可以被写成 $l_{od,1} - \text{norm}$ 形式的近端算子，后使用软阈值法进行求解迭代

(Z_1, Z_2) 部分的更新，由于 (Z_1, Z_2) 属于的惩罚函数共有五种形式，所以作者也分别对这五种形式的惩罚函数进行了推导。由于过程过于烦杂，这里简略。但过程大多离不开利用等价变形将迭代公式写成近端算子形式，后利用近端算子的求解方法进行标准化的求解

5.2.3 停止准则

ADMM算法的停止准则由初始残差和对偶残差决定

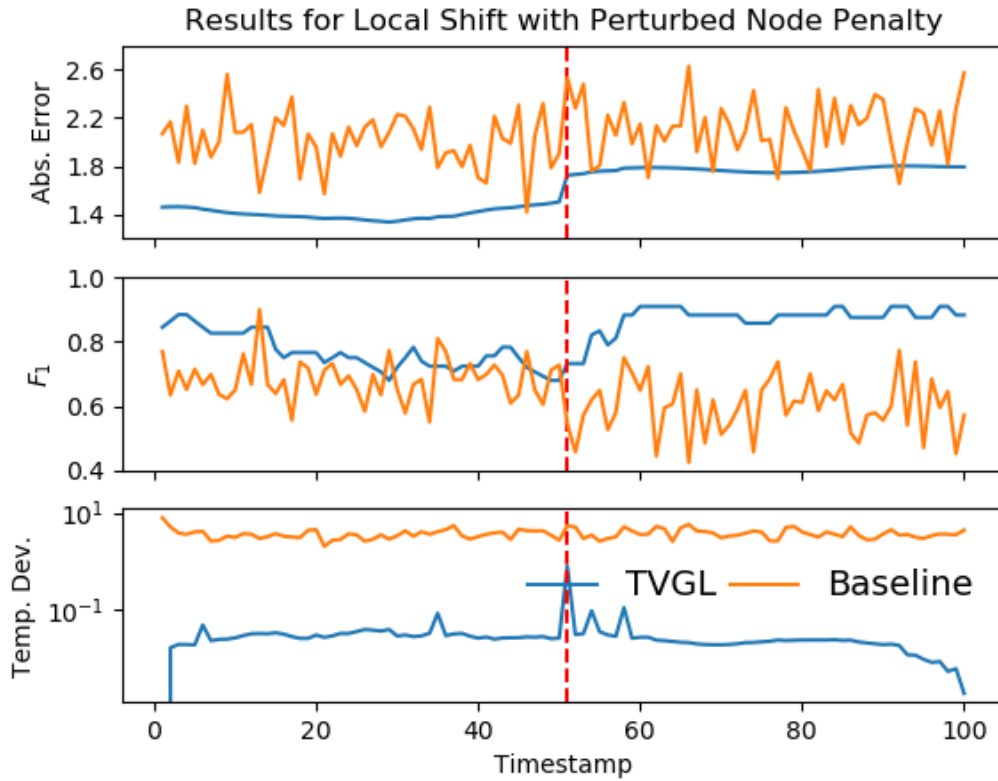
$$\begin{aligned} s^{k+1} &= \rho A^T B(z^{k+1} - z^k) \quad (\text{dual residuals}) \\ r^{k+1} &= Ax^{k+1} + Bz^{k+1} - c \quad (\text{primal residuals}) \end{aligned}$$

算法在对偶残差和初始残差小于预设值时停止

$$\begin{aligned} \|s^k\|_2 &\leq \epsilon^{\text{dual}} = \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \|A^T y^k\|_2 \\ \|r^k\|_2 &\leq \epsilon^{\text{pri}} = \sqrt{p}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\{\|Ax^k\|_2, \|Bz^k\|, \|c\|_2\} \end{aligned}$$

6 实验

6.1 TVGL方法与Baseline对比



本图中测试了TVGL与Baseline在一个突变的多元时间序列中的表现，其中生成该多元时间序列的参数如下：

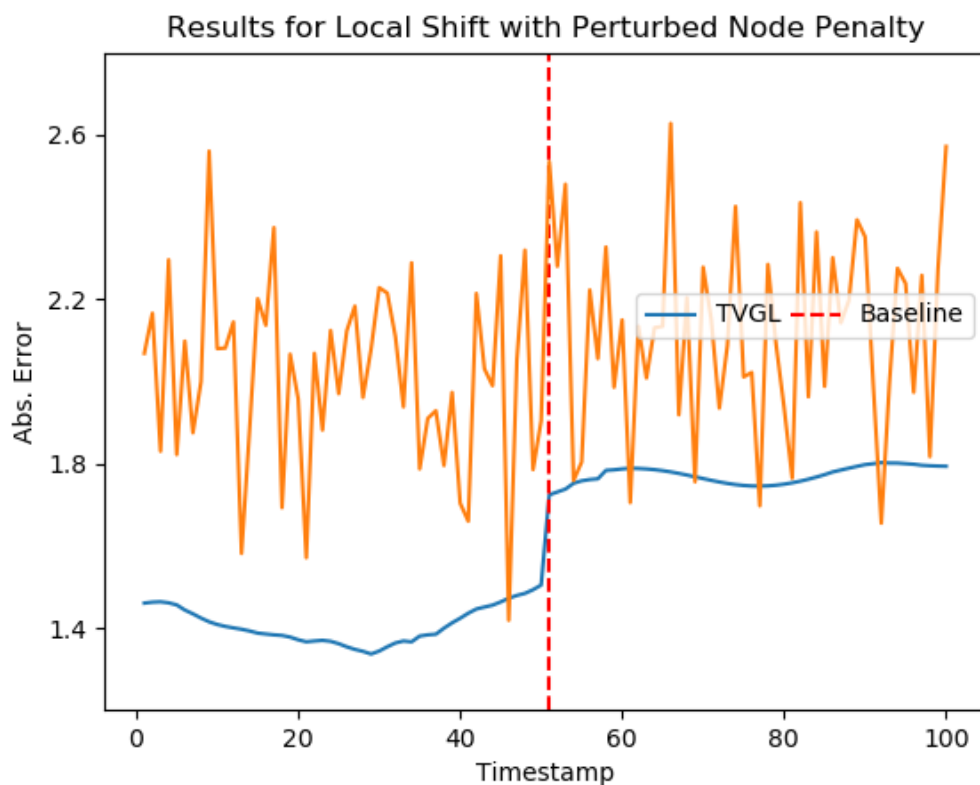
```
size = 10
timesteps = 100
samplesPerStep = 10
numberOfCov = 2
timeShift = int(np.ceil(float(timesteps) / numberOfCov))
index_penalty = 5
```

首先注意timeShift，即为用于生成数据的协方差矩阵改变的时间，可使生成的数据产生变化。在本次实验中协方差矩阵的改变仅限于改变一个随机维度（Local shift）。由于在本次实验中在100个时间点中使用两个不同的协方差矩阵，故协方差矩阵改变时间为t=50。

由以上代码可知，测试数据为10维，共100个时间点，其中每个时间点有10组观测数据。index_penalty即为本次实验利用的是TVGL的5号惩罚函数，为Perturbed Node，可以很好很好地检测某一个观察点的突变。

本图共利用了三种评价方法衡量TVGL与Baseline在Local shift情况下的表现对比

6.1.1 Absolute Error

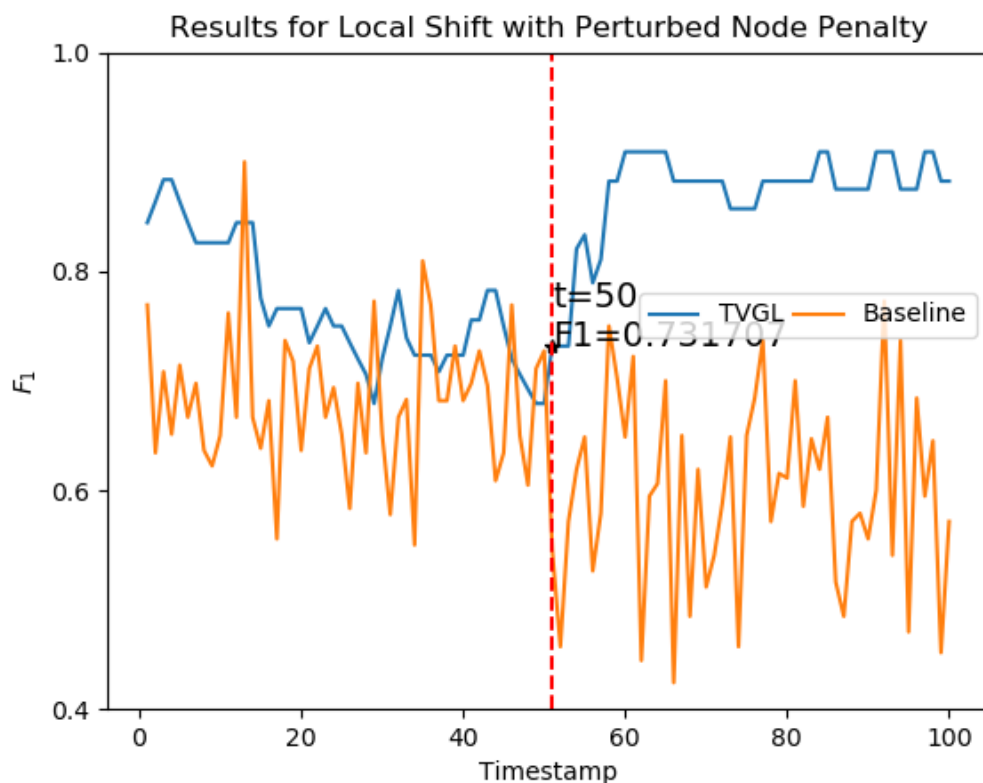


实验利用了在某一时间节点上实际的协方差逆矩阵与模型计算出来的协方差逆矩阵之差的Frobenius范数来衡量绝对误差，即

$$\|\Theta_{i(actual)} - \Theta_{i(estimate)}\|_F$$

可以看出，在t=51的时间节点上TVGL模型因为某一维度的较大改变而产生了较大误差，但误差仍显著低于Baseline。

6.1.2 F1 score



根据F1 score的定义,

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

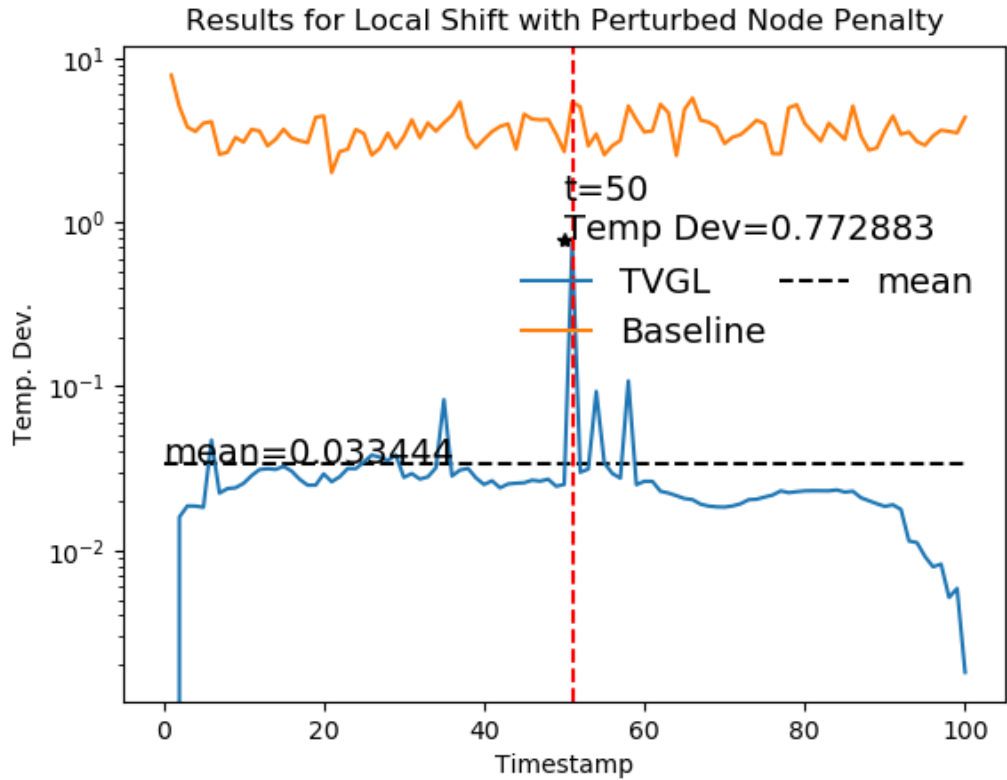
可以用来衡量二分类模型的效果。在本实验中，定义TP (True Positive) 为估计矩阵与实际矩阵相同（实际利用 np.logical_and实现，当两协方差逆矩阵某位置均不为0时即判断为相同）

故Precision（精准度）为：估计矩阵与实际矩阵相同的元素数量/估计矩阵不为0的元素数量

Recall（召回率）为：估计矩阵与实际矩阵相同的元素数量/实际矩阵不为0的元素数量

从图中可以看出，在t=51时，TVGL模型的F1 score为0.73，远高于Baseline，但仍与论文中所提到的0.853有差距。

6.1.3 Temporal deviation (TD)



TD值代表了两次时间节点之间协方差逆矩阵之差的Frobenius范数：

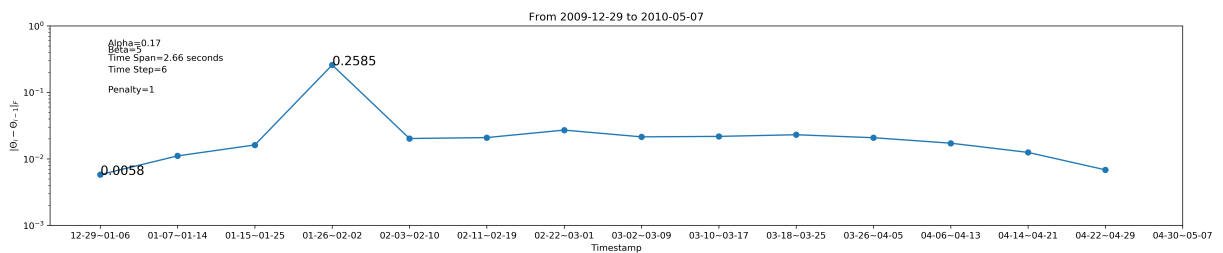
$$\|\Theta_i - \Theta_{i-1}\|_F$$

可以看到在t=50时，TVGL模型的表现十分突出，达到了0.7729，而其在整个时间内平均值为0.0334。相除可得出TD ratio：

$$TD\ Ratio = \frac{0.7729}{0.0334} = 23.14$$

可见TVGL模型在判断某点突变效果显著，但仍与论文中提到的55.5有所差距。

6.2 TD股票实验数据



重新利用，对比论文中图片可以看出，虽然1月中旬时间节点有明显升高，且与论文中值相似约为 2×10^{-1} ，但其余时间并不像论文中一样均处于 5×10^{-2} 之下，而是处于在 1×10^{-1} 的状态。

引用

- [1] DEMPSTER, A. P. (1972). Covariance selection. *Biometrika* 32, 95–108.
- [2] Yuan M, Lin Y. Model selection and estimation in the Gaussian graphical model. *Biometrika*. 2007;94(10):19–35.
- [3] Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*. 2007;9:432–441.

[4] K. Mohan, M. Chung, S. Han, D. Witten, S.-I. Lee, and M. Fazel. Structured learning of Gaussian graphical models. In NIPS, 2012

[5] N. Parikh and S. Boyd. Proximal algorithms. Foundations and Trends in Optimization, 2014.