



The First Report of
2021-2022 Fall Semester Group Projects

Network Inference via the Time-Varying Graphical Lasso (TVGL)

11912814 吕昊泽
11912324 张子研

2021.10.12

1 项目描述

1.1 背景描述

1.1.1 网络与网络推断

自然界有很多自然和人造的网络，例如航线网络，计算机网络，神经网络等。我们通常用一个点和边的集合去定义一个网络，其中点表示网络的参与者，而边则代表网络参与者之间的关系。网络推断就是从已经获得的数据中，通过数学等方法，推断出系统所对应的图模型，从而对系统内部关系，工作原理进行更详细的研究。网络推断是一种方法论，可以被用于许多复杂系统中，对于复杂系统的认知、预测、检测有着重要作用。

譬如在某市构建一个疫情传播扩散的模型，我们可以获得该市各个区的疫情数据，以此构建一个由不同区（实体）构成的系统，更好地探究城区之间疫情传播的关系，预测整个城市的疫情传播情况。更细化地，还可以根据各个街道的数据，以街道为实体构建系统。

现如今，对于网络结构的推断，学界已有显式相似度量，Fuzzy Cognitive Map Graphical Lasso Visibility graph Deep Learning 等方法。它们大多通过自己不同的衡量标准对先有系统的网络进行分析和建模，将建立的模型用于具体应用中去。

现实中，正如上文提到的疫情模型一样，我们所研究的系统中每个实体所记录的数据和测量结果，往往是依托时间变化的。因此，根据这些数据所形成的图模型结构也随着时间的变化而发生着或多或少的变化。所谓动态网络推断，就是在网络推断的基础上，将时间纳入推断的考量中，从而推断出图模型随着时间的变化是如何变化的。动态网络推断可以用于发掘趋势，检测异常，预测未来行为，了解不同实体之间随时间变化的关系。

我们这学期的主要目的，就是探索基于 Graphical Lasso 技术的TVGL方法，探索其背后的原理和应用效果

1.1.2 动态网络中的TVGL方法

为了解决动态网络推断问题，斯坦福大学David Hallac等人提出TVGL方法，将动态时序网络定义为：一个含有 p 个节点的系统，在一个时间序列上有 $m \times n$ 个 R^p 维度的观察数据，通过这些观察数据，TVGL方法尝试对于网络协方差矩阵 θ 和其随着时间的变化进行推断，后用协方差矩阵的数学意义得到各个时刻网络内在的联系。其从静态网络的思路出发，先将这 $m \times n$ 个观察数据分割成 m 个时间节点，每个时间节点的输入数据可视为 $n \times p$ 的一个观察矩阵。TVGL方法先根据协方差的最大似然估计法，由观察矩阵推断出每一时间节点网络的经验协方差 S ，后根据约束函数让所求变量 θ 贴近 S^{-1} 。在得到了每个时间节点 θ 的估计后，TVGL方法通过一个惩罚函数 ψ 将时间纳入网络的考量。通过惩罚函数 ψ 网络保证了每个单独时间节点的推断值 θ_i 都能与前后时间节点的推断值 θ_{i-1} 和 θ_{i+1} 产生相关性，从而得到动态网络协方差矩阵关于时间的变化

1.1.3 动态网络中的其他方法

1. Visibility graph 算法由 Lacasa L 于2008年首次提出，原理大致为根据公式

$y_c < y_b + (y_a - y_b) \frac{t_b - t_c}{t_b - t_a}$ 将两个时刻的y连接起来，形成 visibility graph，然后可以用复杂网络理论来分析时间序列

文章例子：

Shen F, Liu J, Wu K. A Preference-Based Evolutionary Biobjective Approach for Learning Large-Scale Fuzzy Cognitive Maps: An Application to Gene Regulatory Network Reconstruction[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(6): 1035-1049.

2. Fuzzy Cognitive Map 方法 核心是使用FCM算法，每个节点代表一个变量，节点之间的连接代表变量之间的关系，每个节点在t时刻由一个状态值A(t)，节点之间便形成了权值矩阵。

文章例子：

Lacasa L, Luque B, Ballesteros F, et al. From time series to complex networks: The visibility graph[J]. Proceedings of the National Academy of Sciences, 2008, 105(13): 4972-4975.

3. Deep Learning` 捕捉时序信息的模型有很多，比如GRU, LSTM等，但是他们都不能生成对应的图形。因此这方面的研究主要集中于使用GCN, GAT等网络和GRU等网络的结合。例如用GAT等网络提取特征，交给GRU数据进一步处理，从而生成数据。

文章例子：

Zhao H, Wang Y, Duan J, et al. Multivariate time-series anomaly detection via graph attention network[C]//2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020: 841-850.

Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(5): 4027-4035.

1.2 项目目标

1. 读懂论文 *Network Inference via the Time-Varying Graphical Lasso* 并通过网络上代码对论文的结果进行复现和验证，确认其方法有效性。
2. 利用Python3重构该项目算法，提供完整的测试方法。并在项目构建完成后尝试与其他方法进行的对比，寻找可以应用在现实问题的模型。

2 背景与问题定义

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。

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 $i \in 1 - m$ 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，通过协方差矩阵的数学意义得到系统的内在联系。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 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 $i \in 1 - n$ 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 论文方法详解

3.1 方法简介

论文从static graphical lasso问题出发，由静态网络推到到动态网络，最终完成对于不同时间间隔的处理和对于任意时刻协方差矩阵的推导。

3.1.1 静态网络

考虑上述问题定义中的单个时间节点。

节点由输入矩阵 $A = p * n$ $n = t_i - t_{i-1}$,其中A由向量 $x_i \in R^p$ $i = 1 - n$ 组成

通过梯度下降法对公式： $minimize -l(\Theta) + \lambda ||\Theta||_{od,1}$ ，进行求解，得到当前时间节点协方差矩阵逆的估计 θ

条件 $\Theta_i = \sum (a)^{-1}$ $l(\Theta) = n(\logdet\Theta - Tr(S\Theta))$

$||\Theta||_{od,1}$ 是 Θ 的一阶范数，用来保证 θ 的稀疏性，使得其不过拟合

其中 $l(\Theta)$ 鼓励 Θ 接近经验协方差 $S = \frac{1}{n} \sum_{n=1}^n x_i x_i^T$ 的逆

由TVGL的定义可知， $x \sim N(0, \Sigma(t))$,其概率密度函数为

$$f_x(x_1, \dots, x_k) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp(-\frac{1}{2} (x)^T \Sigma^{-1} (x))$$

由最大似然估计法可以得到 $S = \frac{1}{n} \sum_{n=1}^n x_i x_i^T$

对上述概率密度函数取log可得 $-\log |\Sigma| + \frac{1}{n} \sum_{i=1}^n (x_i \Sigma^{-1} x_i^T)$

通过公式变形可以得到 $-\log |\Sigma| + tr(S \Sigma^{-1})$

根据最大似然估计原理，我们可知 $l(\Theta) = n(\logdet\Theta - Tr(S\Theta))$ 取最小值时， Θ 接近经验协方差 $S = \frac{1}{n} \sum_{n=1}^n x_i x_i^T$ 的逆为 Σ^{-1} 即为我们所求

3.1.2 动态网络

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

通过公式： $minimize \Theta \in S_{++}^P - l(\Theta) + \lambda ||\Theta||_{od,1} + \beta \sum_{i=2}^T \psi(\Theta_i - \Theta_{i-1})$ ，加入的惩罚函数 ψ 保证了前后时间节点的 θ 产生相关性，从而得到各个时刻 θ 更为精确的估计

条件 $l(\Theta) = n(\logdet\Theta - Tr(S\Theta))$ 与静态网络相同

ψ 是一个凸惩罚函数，在 $\psi(0)$ 处最小，这使得 Θ_i 和 Θ_{i-1} 相似，也是这个惩罚函数将系统时间尺度的联系纳入模型考虑范围

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

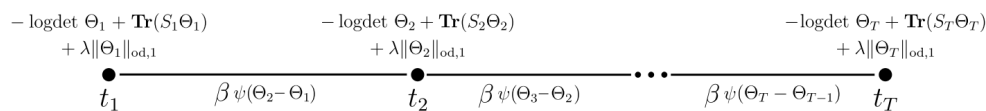


Figure 2: Dynamic network inference can be thought of as an optimization problem on a chain graph, where each node objective solves for a *network slice* at each timestamp, and edge objectives define the penalties that enforce temporal consistency.

3.1.3 不同的惩罚函数

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 $\psi(X) = \min_{V: V+V^T=X} \sum j ||X_j||_2$

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

3.1.5 推断中间网络

我们在获得了分割出来的各个时间节点的 θ 后，如果想要得到某一时刻（其不恰好为分割出的时间节点的最后时刻），

可以通过 $\min_{\Theta} \omega(s - t_{j-1})\psi(\Theta_s - \Theta_{j-1}) + \omega(t_j - s)\psi(\Theta_j - \Theta_s)$ 进行求解

我们在 s （我们要推测的点）创建一个虚拟节点，并将它与最近的输入 j 和 $j-1$ 联系起来

3.1.6 ADMM算法的应用

对于问题的求解，主要到函数由 $l(\theta_i)$ ， $\psi(\theta_i - \theta_{i-1})$ 两个函数，想到增广拉格朗日函数中的 $f(x)$ ， $g(x)$ 故尝试使用ADMM方法对问题进行求解

重新定义问题为

$$\min_{\Theta} \sum_{i=1}^T -l_i(\Theta_i) + \lambda ||Z_{i,0}||_{od,1} + \beta \sum_{i=2}^T \psi(Z_{i,2} - Z_{i-1,1})$$

$$\text{subject to } Z_{i,0} = \Theta_i, \Theta_i \in S_{++}^P \text{ for } i = 1, \dots, T$$

$$(Z_{i-1,1}, Z_{i,2}) = (\Theta_{i-1}, \Theta_i) \text{ for } i = 2, \dots, T$$

推导出迭代方程为，获得协方差矩阵随着时间变化的形式

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

3.2 论文中实验

3.2.1 正确性检验

论文通过自己Mohan et al.提出的方法生成了协方差数据和后续数据，该网络在 $t = 50$ 的时候发生了巨变（巨变分为单一节点的变化和整个网络的变化）。将生成的数据交由网络进行正确性检验，并以其他基本方法，如static graphical lasso, 作为基准线进行对比

对比的性能指标设置为

$F1 \text{ score}$:用来衡量网络是否得到了正确的协方差矩阵

$TD = ||\Theta_i - \Theta_{i-1}||$ 用来衡量每个时间戳中发生了多少变化

结果如下

True Shift	Score	Static GL	Kernel	TVGL (ℓ_1)	TVGL (ℓ_2)	TVGL (Perturbed Node)
Local	F_1	0.646	0.768	0.819	0.817	0.853
	TD ratio	2.02	2.41	27.9	23.3	55.5
Global	F_1	0.496	0.688	0.939	0.952	0.943
	TD ratio	1.06	1.80	47.6	38.6	36.2

Table 1: F_1 score and Temporal Deviation (TD) ratio for a local and global shift, using the two baselines and three different TVGL evolutionary penalties.

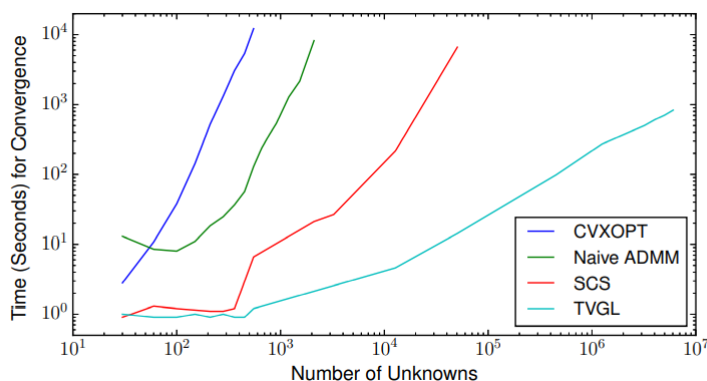
这其中的 $TD \text{ ratio}$ 为 $t = 50$ 时的TD值和TD平均值的比值

从文章的结果可以看出TVGL网络方法在正确率上远高于其他网络，且在 $t = 50$ 网络发生巨变的真实时间戳上，其给出的估计协方差网络的变化也远大于其他基准方法的变化

此外，该实验同时说明了不同的惩罚函数对于不同的网络变化表现出的效果有所差别，由此可见选择正确的惩罚函数也是重要因素

3.2.2 时间检验

文章指出，对于 m 个时间戳， n 个节点，需要计算 $\frac{n(n+1)}{2}$ 个未知变量。文章指出其时间复杂度大概为 $O(n^3)$ 。对比其他方法大概是 $O(n^6)$ 的时间复杂度，可见TVGL方法在时间复杂度上胜于其他方法。



3.2.3 金融数据实验

文章探究了六只股票（苹果，微软，英特尔，亚马逊，联邦快递和波音）在2010年初三个月的1月。苹果发布初代ipad前后的数据，并对其结构进行了推倒。

通过TD数据能清晰地看到该网络在2010年1月左右出现波动，且网络前后结构发生变化。

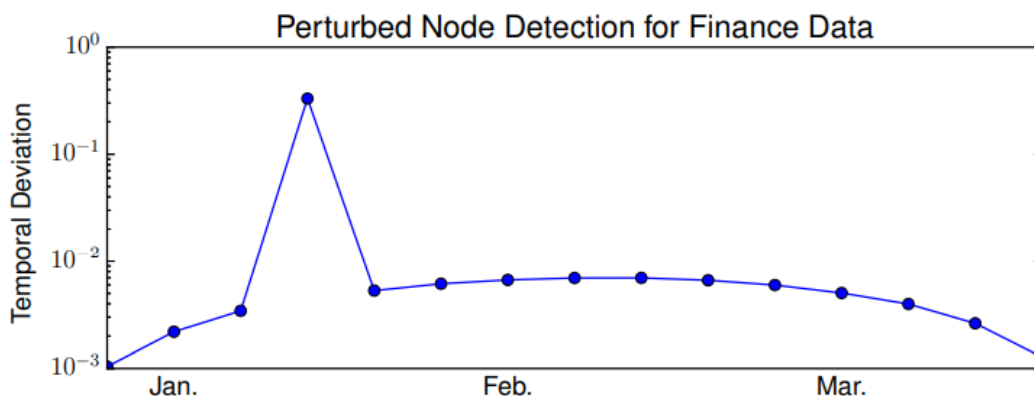
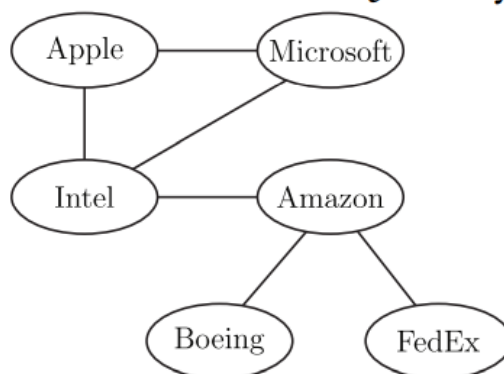


Figure 5: Plot of the temporal deviation of our estimated stock network, which detects a local shift in late January.

	1	2	3	4	5	6
Apple: 1	■	■	■	■	■	■
Microsoft: 2	■	■	■	■	■	■
Amazon: 3	■	■	■	■	■	■
Intel: 4	■	■	■	■	■	■
Boeing: 5	■	■	■	■	■	■
FedEx: 6	■	■	■	■	■	■



4 当前进展

4.1 第一次答辩工作介绍

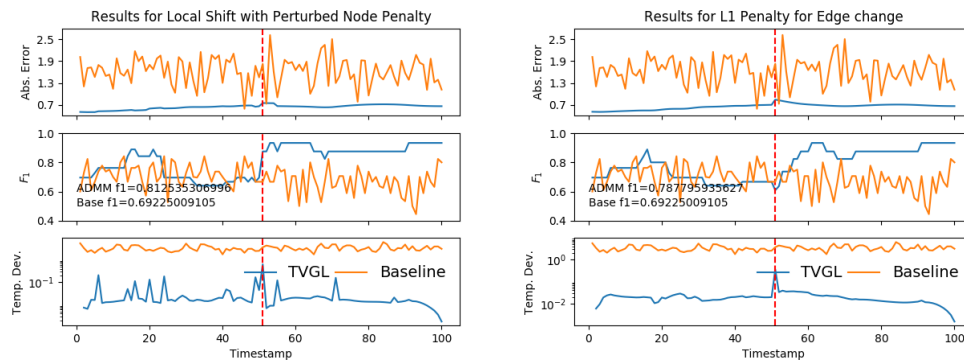
4.1.1 对于网络正确性的验证

对于文章描述的方法，根据网上获取的代码进行适当修正后，进行对于网络正确性的验证，获取TD指标和F1指标

由于电脑性能，本实验采用的检测方法为6个节点，100个时间戳，每个时间戳的输入数据为6（为了验证第二个实验方法的可靠性）

试验了惩罚函数1和5即少数边改动和单一节点扰动，网络变化类型为节点扰动

得到的结果如下：

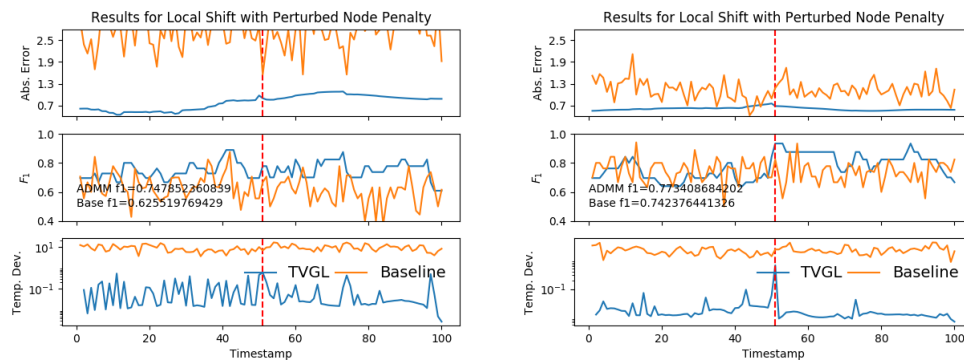


可以看到两种方法在协方差矩阵真正变化的 $t=50$ 时刻都表现出了最高的TD值，且基于5惩罚函数的第一张图F1表现更佳

同时从TD值变化趋势也能看出不同惩罚函数对于方法输出所带来的约束

文章同样获取了在每个时间戳输入数据为10和3时网络的表现 惩罚函数为5即节点扰动

结果如下：



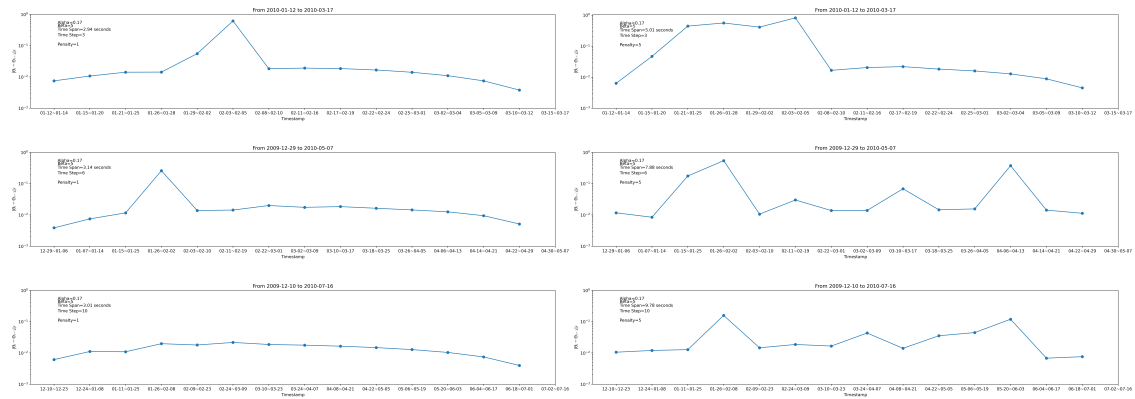
可以看到表现都不如输入数据为6时的表现，由此可见每个时间戳输入的数据个数并不是越多越好，也要有适当的选择

4.1.2 对于股票数据的研究

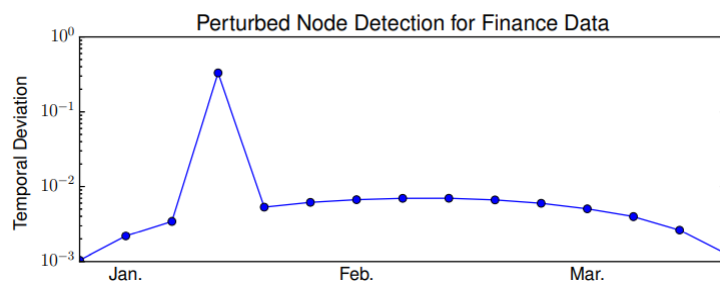
为了对论文股票研究方面进行复现，从网上找到了相关的股票数据，对论文提到的时间节点前后进行了协方差的分析。

4.1.2.1 TD变化图

分别使用1和5两种惩罚函数对数据进行了分析，时间戳之间的间隔从3-7均进行了研究，结果如下。



上面列出了两种惩罚函数 3 6 10 时间步数的结果，与论文中所示图片



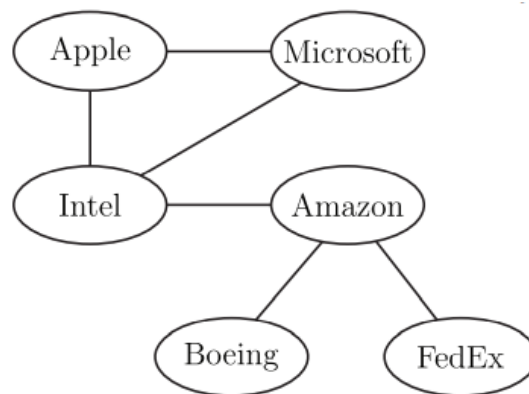
进行对比可知：

1. 步数较少时，网络波动极大，推测与协方差矩阵的定义相关，期望方差的采样均需要一定的样本量
2. 步数太大时，网络基本没有波动，推测是时间步数过长导致网络敏感性降低，导致无法有效探测网络发生的波动。
3. 从结果中可以得知 采用惩罚函数1 step 6 的实验与论文结果最为吻合。这也符合了之前一个实验对不同步长F1的探究

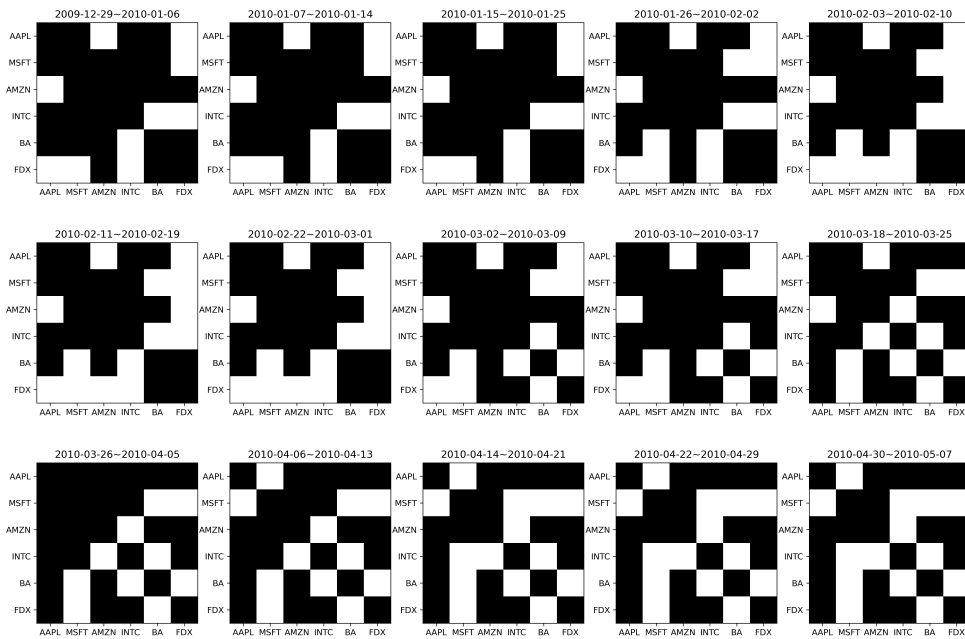
4.1.2.2 协方差矩阵

本块状图均基于惩罚函数1 step 6实验

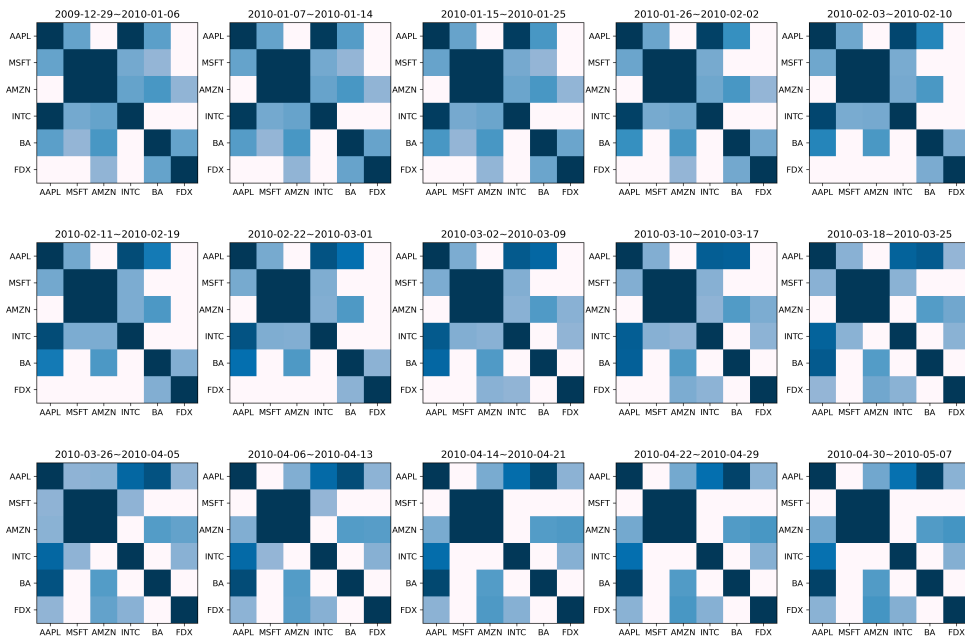
	1	2	3	4	5	6
Apple: 1	1	1	0	1	1	0
Microsoft: 2	1	1	0	0	0	0
Amazon: 3	0	0	1	0	0	0
Intel: 4	1	0	0	1	0	0
Boeing: 5	1	0	0	0	1	0
FedEx: 6	0	0	0	0	0	1



由于协方差矩阵定义是不为0则为相关，所以首先将矩阵0和不为0的部分进行区分，得到的结果如下



可以看到五只股票的关系确实是随着时间的变换不断变换的，但是在1月中旬的矩阵中并没有看出论文中给出的图片结果。由于论文中并没有指出此图的处理方法，在想到协方差矩阵的数值绝对值表示关系强弱后，选择对其进行绝对值数值化。（值得注意的是，对角线被重新分配成除对角线外最大值，0被分配成-1）



这样能更清晰看出一月中旬网络结构确实发生了改变，APPLE公司与微软，英特尔和波音的关系密切。这符合论文的结果，但是并没有看出文中所指的结构变化

4.1.3 复现结果总结

目前，对于论文中所提到的结果，除了对500支股票的整体研究没有复现，我们基本上已经复现完成。

在对于论文方法正确性的研究中，我们成功复现论文对于其方法优越性的证明。从 F1 和 TD 分数两方面全面领先于基于静态图形套索的标准方法。且在网络没有真实变化时，对于输入的正态分布数据，TVGL 方法的网络变化很小，可以证明其抗干扰能力很强。除此之外，我们还探究了时间不长的选取对于网络工作性能的影响。发现在 $n=6$ 的条件下，过长或过短的时间补偿都没有很好的表现，所以我们最终选取了 `step=6`，并将其作为第二部分研究的基础。

在对于几只股票网络变化研究中，基于之前探究的 $\text{step}=6$ 的设置上，我们成功复现论文中对于 TD 值突变的结果。但是对于后续论文中提到的通过网格图的形式表现股票之间的关系，我们仅仅复现出其中的一部分，另一部分结果并不相同。

对于尚未复现的对于500值股票的研究，我们面临的困难主要有以下两点：

1：论文中对于这部分研究只有文字描述，其只提到2010年5月网络整体结构发生了突变，并没有指出这个突变是根据何种指标观测出来的。

2：论文中对于500只股票的研究，是在一年时间长度上研究的。由于 $n=500$ 我们假定时间步长为 6 那么我们需要计算 $\frac{500(500+1)}{2} * 60 = 7515000$ 个未知变量。

其中算法中耗时最长的是对于矩阵的特征分解。对于 $n=500$ 的情况，单次运算的时间复杂度达到 625000000 ，且收敛需要多次运算，且有60个矩阵要分解，时间复杂度过大。

论文中给出的数据显示在未知变量在这个数量级时它们的服务器大概需要数小时时间算出结果，换算到个人机器就是几天，实在没有条件进行计算，且数据收集也面临问题

5 项目面临的困难

本项目的初衷是通过对TVGL方法的了解，对时序模型分析有所了解，并通过对论文结果的复现和验证提高对于网络的理解并熟悉各种机器学习基本方法。看似简单，但实际上，由于此方法是17年的方法，导致网上能找到的代码均是基于Python2所构建，且大多数模型并没有提供完整的数据集，文档，接口等共我们进行实验。可以说在网上寻找代码的过程让我们对网络的认识和理解更深一层。

而找到代码，进行实验，还需要对其给出的数据进行了解，对其方法进行正确性验证。在找到论文所使用的Python2代码后，我们同时采取了直接使用Python2代码与转换为Python3代码再使用两条线路，最后修改过的Python3代码可以更好工作，故使用修改过的Python3代码作为实验用代码。

同时，为了对可能的模型对比进行准备，我们还阅读了其他有关的时序网络，并对其进行了少量的实现。

在进行结构可视化的时候，为了得到论文中的结构进行了大量实验和尝试，最终才得出与论文相近的结果，尤其是在协方差矩阵中，大部分股票之间的关系都不明显，需要加强结果后再画图，才能得到更好的可视化结果。

6 下一步计划

在结束第一次汇报前，我们已经熟悉了本篇论文的思路并且了解了论文所使用的方法论。

第二次答辩前：

- 利用Python3重构该项目算法

最终答辩前：

- 提供测试方法
- 尝试与其他方法进行的对比，并尝试运用在显示问题中