进展

1. 论文创新点

1. 现有的工具可以方便获取微服务间关系，做影响图有没有意义

- 微服务很少时，人工经验结合背景知识即可解决问题。
- 我们的工具要解决服务相对多，且工程师不能通过***背景知识*** 明确知道微服务间的影响关系情况下的服务间相互关联影响。
- 现有工具可以获得微服务间的调用关系，但是这种调用链可能很庞杂，理不出头绪，信息过多反倒不起作用。
- 业务逻辑上不相关的服务，由于共享系统关键资源，实际上产生相互影响。我们的工具可以挖掘潜在隐含关系。

2. 横向扩展：针对不同类型业务、不同类型KPI数据，从算法选择（基于***分数*** 的算法和基于***约束*** 的算法在异常检测中的性能分析）和独立性检验两方面进行对比分析。
3. 纵向：解决特定的一种业务情景（殷博提到的服务跨层调用）下，服务间影响分析。

2. 算法

2.1 独立性检验

2.1.1 Fisher Z test

从对相关性的检验来，

相关系数计算：

$$r = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}}.$$

为了对相关系数进行显著性检验，使用fisher z transformation

$$z = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) = \operatorname{arctanh}(r),$$

相关系数本身是对线性关系的探究，因此猜测这种独立性检验检测的也是序列间的线形关系。

2.1.2 correlation T test

http://www.real-statistics.com/correlation/one-sample-hypothesis-testing-correlation/correlation-testing-via-t-test/

也是基于相关系数的显著性检验

用于样本含量较小的情况。

$$r^2 = \frac{t^2}{t^2+df}$$

例子：

Cigarettes	5	23	25	48	17	8	4	26	11	19	14	35	29	4	23
Longevity	80	78	60	53	85	84	73	79	81	75	68	72	58	92	65

香烟与长寿

$$r = \text{CORREL}(\text{R1}, \text{R2}) = -713$$

p-value = TDIST(ABS(-3.67), 13, 2) = .00282 < .05 = α (two-tail)

t_{crit} = TINV(.05, 13) = 2.16 < 3.67 = $|t_{obs}|$

2.1.3 Exact Test

$$F(r) = \begin{cases} \frac{1}{\pi}[\cos^{-1}(\rho) + c_3 s_1 - c_4 s_2], & n \text{ is even} \\ \frac{1}{\pi}[\cos^{-1}(-r) + c_3 s_3 - c_4 s_4], & n \text{ is odd} \end{cases}$$

2.1.4 总结

这些检验都是基于统计学中相关性的概念，从不同的维度进行显著性检验。

这就不难解释为什么Fisher Z test与correlation T test的结果相同。

2.2 python 源码剖析

利用上次会议中提到的文献《Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm》中给出的方法，进行独立性检验

$$\rho_{i,j|\mathbf{k}} = \frac{\rho_{i,j|\mathbf{k}\setminus h} - \rho_{i,h|\mathbf{k}\setminus h}\rho_{j,h|\mathbf{k}\setminus h}}{\sqrt{(1-\rho^2_{i,h|\mathbf{k}\setminus h})(1-\rho^2_{j,h|\mathbf{k}\setminus h})}}.$$

实验得到结果。

2.3 Tetrad源码分离

对Tetrad源码的项目结构，数据流进行了分析，进一步研究了代码分离的方法。