Effective Temporal Dependence Dicover in Time Series Data

问题描述

在很多情况下，时间序列数据中都有很强的短期依赖关系（temporal dependence），即在时间序列数据中某个时间点前后的数据有着很强的相关性。合理运用这些短期相关性有着很多的应用场景。如分析商场中的不同因素对用户购买情况的影响，或者帮助用户分析投资组合情况，再者，合理的运用还能够对用户的行为进行预测。为了说明这个问题，可以举下面这样的例子。

例一. *对于一个投资组合，找到每周的交易量和接下来几周的平均股价变化的关系。*

例一的需求其实就可以看作探究在当前交易量和平均股价变化的短期依赖关系，其中当前交易量是致使行为（causative behavior）,平均股价的变化是依赖行为（causative behavior）。在短期依赖关系的定义中，致使行为就是指影响用户未来行为的那些因素，依赖行为就是值由致使行为引起的那些行为。

对于用户的行为分析，有一种已经存在的表现的也算比较好的算法叫做同期群分析法（Cohort Analysis）。这种方法首先被应用在社交科学中，最近在网络用户行为模式分析中也有比较好的结果。同期群分析实现起来也比较简单，主要通过两方面来衡量用户的行为：社会行为的变化（social change）和时效（aging），在这个模型中认为这两方面合主要影响了用户的行为。为了捕获社会行为的变化，这个方法将具有相同特征的人放在同一个群（cohort）中，如做一个特定的事件的时间和地点。然后在每一个群中衡量不同时间下用户的行为从而反映出对时效的影响。

但是，对于发现短息依赖关系而言，同期群分析也有很多的缺陷。首先，时效的定义知识通过简单的出生时间来定义，这就大大降低了定义致使行为的灵活性；而且，同期群分析一般定义用户的出生就是指用户的生日，这是不会变化的。但是实际上，在不同的时间，用户很有可能会产生不同日期连续的致使行为，这些是都需要被考虑在内的。如在例一中提到的，在股市交易中，每个股票的价格都是持续变化的，如果只考虑第一周的股票的价格在很多情况下是不具有参考性的。

所以，作者提出了递归的同期群分析（recurrent cohort analysis），它解决了同期群分析的一部分缺陷。这个方法通过定义如何进行分片，以及定义滑动的时间窗口，来将行为分析定义在窗口中。然后通过一系列的群操作（cohort operators）来对初始的eventT模型进行转换，得到最终可以分析短期依赖的模型。最后，再对模型中可挖掘的并行充分利用，并给出了适配于分析的并行架构。

时间窗口的实现

重构的同期群分析的核心就是如何来衡量致使行为和依赖行为。为了衡量这两个行为，首先定义如何进行时间分片，并在这个基础上定义滑动时间窗口，即包含连续的时间分片的集合，并且可以随着分析进行滑动。建立时间窗口的过程中，首先使用一个叫eventT的表，其中包含了每个用户的每个行为。它主要有两列构成：用户和时间，记录了谁在什么时候做了什么动作。其中动作可以单独为一列，也可以不包含动作。

时间分片

为了能随时间动态地衡量致使行为和依赖行为，我们需要将时间序列数据分片，每片代表分析的最小单位。简单的分片方法可以是将数据按时间分为等长的数据片，但是显然这样满足不了大多数情况下的要求。因此定义

了下面的方法来进行分片。

定义1. *时间的划分，如P所示*

$$P=\left\{\begin{array}{cc}(A,C), & if\ A=event\\(A,U), & if\ A=time\\(A), & otherwise\end{array}\right.$$

其中A表示eventT的一个分布，C是eventT的一个可选的分布公式，表示A的一个按event的划分，U是一个时间集合，表示对时间的划分。

定义2 *对于一个给定的划分P，可以将用户u的时间划分为如下的分片：*

$$S_u = [t_i, t_{i+1}) | i \geq 0 \vee t_i \in D \vee t_{i+1} \leftarrow mint | t \in D \vee t > t_i$$

$$D=\left\{\begin{array}{cc}{d_{u,i}[time]|i=1\wedge P.C(d_{u,i}=true)}, & if\ P.A=event\\{d_{u,i}[time]|i=1\wedge d_{u,i}[P.A]\neq d_{u,i-1}[P.A]}, & if\ P.A\notin \{event,time\}\end{array}\right.$$

其中$d_{u,i}$表示u用户按时间顺序表现出的第i个状态，$d_{u,i}[A]$表示状态对应的A列的值。对于这个表示，S_u的第i个时间分片，就是对应的$[t_{i-1}, t_i)$，用$S_{u,i}$来表示。

简言之，对于这个时间分片的划分，如果P是时间，那么对于分片的分割就是P.U对应的时间的划分；如果P是event，那么分割的界限有P.C对应的对于event的划分得到；如果二者都不是，就在时间序列数据中对应的数据变化的部分设置分割得到分片。

时间窗口

通过时间分片，我们得到了一系列的片段。基于这些片段进行动态分析，我们其实已经能够进行简单的短期依赖关系分析了。而且，我们还能够通过分片来推断出这个片段属于致使行为还是依赖行为，并分别对其进行分析能够得到一部分结果。但是，这样就限制了我们只能够每次值衡量一个时间分片的内容。但恰好在很多情况下这种限制是很致命的。比如说例一，如果我们想知道近五周的股票交易量如何影响股票未来的价格，我们就需要对五个时间分片进行分析，这样只用一个分片的情况就无法满足我们的要求了。

对于比较自然的想法，我们可以定义时间窗口为一个可配置长度的包含这连续时间片的窗口，这样我们就可以让这个窗口进行滑动，来满足我们的要求。如当窗口长度为l滑动到分片$S_{u,i}$时，这个窗口就包含分片$(S_{u.i-l-1}, \cdots, S_{u,i})$。虽然这样的定义已经能满足我们的大多数要求，但是仍存在无法解决的问题。仍拿例一举例，如果我们想了解那些比上一周股票价格高的周对交易量的影响，这样的模型就无法满足我们的要求了。因此，我们按照如下的方式来定义时间窗口。

定义3. *时间窗口W的定义如下所示：*

$$W = (P, W^l, W^h)$$

其中P在定义1中有定义，对于这个窗口，它会滑动到$S_{u,i}$位置。w^l和w^h是用于确定窗口区间的两个整数。这个窗口中的分片定义为$T_{u,i}$，如下所示

$$T_{u,i} = S_{u,j} | f(i, w^l) \leq j \leq f(i, w^h)$$

其中方法f如下所示

对于$T_{u,i}$，我们定义$S_{u,i}$为这个时间窗口的位置。

定义3中对于时间窗口的定义就很好地解决了我们前面提到的那个限制。这样定义不仅允许我们设置窗口包含的分片的多少，还能够定位窗口包含的分片的位置。根据等式2，如果定义的w^l为正数，那么窗口中最开始的分片即为S_{u,w^l}，这样就和位置无关了。如果w^l为0或负数，w^l就会随着窗口的滑动，不断改变，这种情况也就能处理我们先前举的那个例子了。

时间窗口的属性

定义3中的时间窗口已经能够满足我们动态选择时间分片的需求了。那么对于滑动过程中时间窗口选择的那些时间分片，我们还需要定义我们要对这些分片做的动作，这个过程可以定义为时间窗口属性的衡量，定义如下。

定义4. 一个时间窗口属性A定义如下

$$A^t = (F, A, W)$$

其中f是一个合计方法，即对这个窗口中的所有分片进行计算的操作；A和定义1中出现的A一致，是表示用户活动的eventT表；W表示被计算属性的窗口。

这样的定义中，属性是定义在一个窗口上的，这样就限制我们只能在一个窗口上来计算相关的属性。仍拿例一举例，如果我们要计算股票交易量的变化，我们就需要计算当前周的窗口计算的贸易量的值和前一周的贸易量的值作差，这就要求我们定义两个窗口属性A_1^t和A_2^t，其中$A_1^t.W.w^l = A_1^t.W.w^h = 0$表示当前窗口，$A_1^t.W.w^l = A_1^t.W.w^h = 0$表示当前窗口前的一个窗口。这样我们就能通过计算$A_1^t - A_2^t$得到我们需要的结果。但是，如果我们的需求是对每周的贸易变化量求平均的话，那么我们仍然单独对每个窗口做操作定义的复杂程度就会很高，因此，可以定义综合时间窗口属性（Composite time window attribute）和复合窗口属性（Compound time window attribute）。

定义5. 综合时间属性是值在一个或多个具有共享的底层划分的窗口上做计算得到的一种属性。

定义6. 复合时间属性的定义和定义4中定义的等式相同，但是其中的A不再是一个eventT表，而是和W具有相同窗口划分的时间窗口属性集合。

通过这两个定义，我们就能够很好的处理刚才提到的那个例子。自此，我们就完成了我们对时间窗口的定义。时间窗口在这个算法中有着中心的地位，这是动态地衡量用户行为的基础。

递归同期群分析（Recurrent Cohort Analysis）

用户行为的衡量

前面在定义窗口的过程中其实我们已经可以随着窗口的滑动对每一个窗口来进行相应的计算了。但是这样的定义仍不完全，因为我们没关注我们可能对哪些行为是感兴趣的哪些是不感兴趣的，而是对于所有的窗口都是统一的定义。这样其实在很多情况下是无法满足需求的。仍拿例一举例，如果我们希望衡量某个时间前一个用户的所有行为的话，我们就无法通过之前的模型来实现了。因此，我们在衡量用户行为，滑动窗口的过程中就需要对我们不感兴趣的窗口进行过滤。

定义7. 用户行为衡量M如下定义

$$M = (C^e, C^w, A^t)$$

和前面对分片的定义相同，C^e是是eventT的一个可选的分布公式；C^w是一个衡量公式，能以一个或多个窗口作为输入，判断其是否满足公式的需求；A^t是表示用户行为的时间窗口分布。

这样，我们就定义了如何对用户的行为进行衡量。在窗口滑动的过程中，如果C^w衡量这个窗口为真，那么$A.f$对应的行为就会施加到这个窗口上，否则就继续滑动。

递归同期群分析

定义8. 递归同期群分析的的定义如下

$$(M^c, M^d, [A^{tg}])$$

其中M^c, M^d是用户行为的衡量，分别表示对致使行为和依赖行为的衡量。A^{tg}是一个可选的时间窗口属性，它用来和给定致使行为相关的每个依赖行为的年龄（age）

在这样的定义中，M^c决定了每一个用户数据是否被加入到被分析的队列中。在每个$M^c.C^w$为真的窗口处，这个用户数据就被加入到分析队列中表示这些数据要被当作致使行为来衡量。

递归同期群分析的操作符

窗口滑动操作符γ

窗口聚合操作符ϕ

Age-by操作符ψ

递归聚合操作符ω

查询计划（Query Plan）

操作符估计

基于SQL的策略

侵入策略（Intrusive Strategy）

一些讨论

分布式架构

例子

相关研究

已有算法的不足

解决方案