**Wang, Yanhao , et al. "Real-time influence maximization on dynamic  
social streams." Proceedings of the VLDB Endowment10.7(2017):805-816.**

**Abstract**

现有的IM解决方案没有考虑到社会影响的高度动态性，导致种子质量较差，或者当网络进化时处理时间较长。为了解决这个问题，作者定义了一个新的IM查询，名为stream influence maximization(SIM) on social streams。在技术上，SIM采用滑动窗模型，窗口内保留了一组当前影响值最大的k个结点。然后作者又提出了影响检查点（influential checkpoints）框架去推进连续的SIM查询过程。IC框架为每个窗口移位创建一个检查点，并确保“近似解”。

**1.Introduction：**

社交媒体广告已经成为许多公司在网上推广业务不可或缺的工具。影响力最大化(IM)是社交媒体病毒式营销背后的一个关键算法问题。

现有的IM解决方案大多依赖于用户之间的影响概率来获取种子。影响概率通常来源于在线社交网络中的社交行为。在现实生活中，社会影响是高度动态的，突发新闻和热门话题可以极大地改变用户之间的传播趋势。因此，假设静态社会影响的IM方法所选择的种子可能很快就会过时。近年来，人们对动态社会影响下的IM进行了一些研究。然而，现有的动态IM解决方案也不能为所选种子的质量提供理论保证(例如，[2,38])以更新的高处理开销为代价提供保证)。

为了解决这种问题，作者提出了SIM，SIM利用广泛存在的社会行为来估计社会影响，并持续维护种子集。

由于SIM的np –hard，我们主要用理论边界对其进行近似处理。

贡献点：

1. 作者解决了现有IM解决方案在支持快速发展的社交网络方面的局限性，并提出了一种新的基于滑动窗口的SIM查询。

2. 作者为了SIM查询提出了一个新的影响检查点框架

3. 控制了时间复杂度

4. 实验验证了作者提出的框架的有效性。

**2.related work**

**2.1 IM**

**在动态网络中的IM问题：**然而，这些方法大多不能提供种子质量的理论保证，并可能返回很差的解。Chen提出了一个上界交换的方法，不过，UBI对所选用户的数量很敏感。当种子集的大小增加时，其性能和质量都会急剧下降。这就阻碍了UBI的实际应用。最近，在[30]中提出了一种新的具有理论界的动态IM方法。它根据图的变化动态地维护一个基于ris的[7]索引，但由于维护成本较高，每秒只能处理几百次影响图更新，无法满足现实社交流的需求。

**2.2 流子模块优化（SSO）**

与SIM密切相关的另一个领域是流子模块优化。SSO采用了只追加的流模型，其中元素一个接一个地到达，目标是动态维护一组最多k个元素，以最大化子模块函数。Saha等人，[31]和Ausiello等人针对SSO的特殊情况开发了两种方法(i.e., the online Maximum k-Coverage problem)。最先进的SSO的解决方案是SieveStreaming和ThresholdStream。不幸的是，SSO算法不能直接应用于滑动窗口模型，因为它们不能处理元素的连续失效，然而，我们将在第4.2节中说明，现有的SSO算法可以作为IC和SIC框架中的检查点预测。

**2.3 滑动窗口的函数估计**

已有文献[8,12]研究了如何在滑动窗模型中连续估计函数。它们利用目标函数的特殊性质来实现次线性性能和合理的质量。设g为目标函数，A,B,C为流上的三个序列，B是a的尾子序列，C是B的邻接子序列。

**3 问题描述**

我们考虑一个用户集为U的社交网络上的社交流。社交流包括由用户活动生成的无界时间均衡的社交活动。作者用表示在t时刻的活动，代表用户u由于之前的活动at’在时间t更喜欢活动at。一些典型的活动为“回复”“评论”。如果一个action at对之前的任何action都没有响应，例如，一个用户u发布了一条原始tweet，我们将其称为root action并表示为

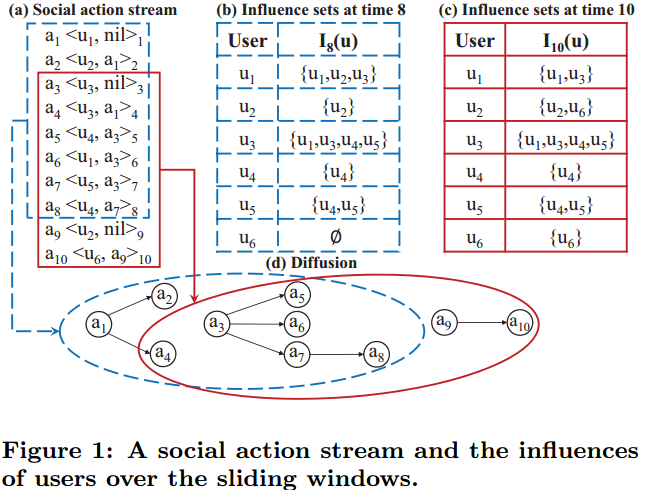
和许多数据流一样，社交流对时间很敏感：最近的行动比过去的行动更有价值，我们采用公认的基于序列的滑动窗口[12]模型来捕捉这种本质。设N为窗口大小，基于序列的滑动窗口Wt保持最新的N个动作，。作者使用Wt[i]表示在Wt中的第i个活动。随后使用At表示表示在Wt中执行至少一个活动的活跃用户集。。

作者通过用户执行的操作来定义用户之间的影响。作者使用表示u在Wt中影响到了v，如果有一个活动a被v执行了的话，并且a是被u的活动a’激活的。作者定义用户的影响集合。

**定义1:**在时间t时一个用户u的影响集合表示为，表示被用户u影响的用户集合。

直观上，u的影响集合表示最近在u的影响下执行操作的用户集。影响集的概念可以自然地扩展到一组用户。特别的，作者使用代表k个用户的集合，Wt是所有成员的影响力集合的集合，然后集合S的影响价值可以计算为

例子1描述了作者定义的在滑动窗口之上的影响力。



随着新操作以高速到达，而旧操作以相同的速度过期，具有最大影响力值的用户不断发展。为了实时跟踪社交流上有影响力的用户，我们提出了一个流影响力最大化的方法(SIM)查询，

其**正式定义**如下：作者使用Wt表示在时间t的滑动窗口，SIM是一个在社交流上的连续的查询，并且一直返回k个影响值最大的用户的集合



注意，本文提出的解决方案也支持滑动窗口移动多个动作的情况。为了简单起见，我们将重点介绍处理滑动窗口的解决方案，该解决方案只需在a处移动一个操作。

**4. influential checkpoint**

由于SIM是np困难的，在多项式时间内保持每个滑动窗口的最优种子集是不可行的。因此，我们的目标是保持一个近似的解决方案，以达到一个有界比的最优有效。最简单的方法是在没一个窗口滑动的时候运行一次贪心算法。贪心算法的低效使得它无法处理窗口大小很大时，新动作的快速到达。

在滑动窗口上有效支持SIM的一个关键挑战是同时处理旧操作的过期和新操作的到来。在本节的剩余部分中，作者提出了一个新的影响力检查点(IC)框架，它由一系列检查点预言组成，可以同时有效地处理操作的过期和到达。

**4.1 influential checkpoint framework**

IC框架的高级思想是避免在窗口移动时处理旧操作的过期。为了实现这一目标，框架保持了对每个窗口移位递增部分结果(即一个影响力检查点)。当旧操作过期时，将删除过期的结果。通过这种方法，将滑动窗口模型转换为每个检查点的更简单的只加模型.

准确的说，用一个影响力节点代表一个checkpoint oracle。一个简单的去处理窗口从Wt-1到Wt的过程如算法1所示。当一个新的动作到达时，Wt - 1中最古老的检查点到期(Λt−1[1])和一个新的检查点Λt [N]添加到Wt。在将Wt−1中的剩余检查点添加到Wt之后(第3-4行)，Wt中的每个检查点都作为一个附加操作来更新其部分解决方案(第5-6行)。回答Wt的SIM查询,我们只是返回Λt[1]的结果。

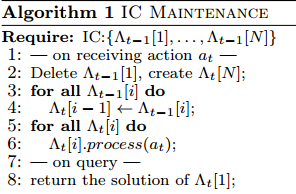
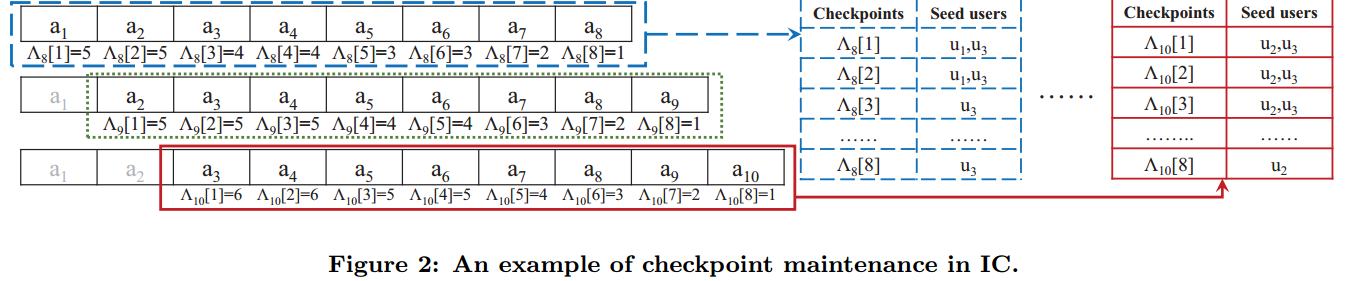


图2演示了下面示例1中的IC框架中检查点的维护。



**4.2 checkpoint**

为了弥补两种流模型（IC，SIM）之间的差距，并利用现有的基于集流模型的算法，我们提出了一种通用集流映射(SSM)接口，该接口对set-stream算法进行了两次调整，以充当检查点oracle。首先，候选解决方案CX用于存储k个用户。

**5. sparse influential checkpoint**

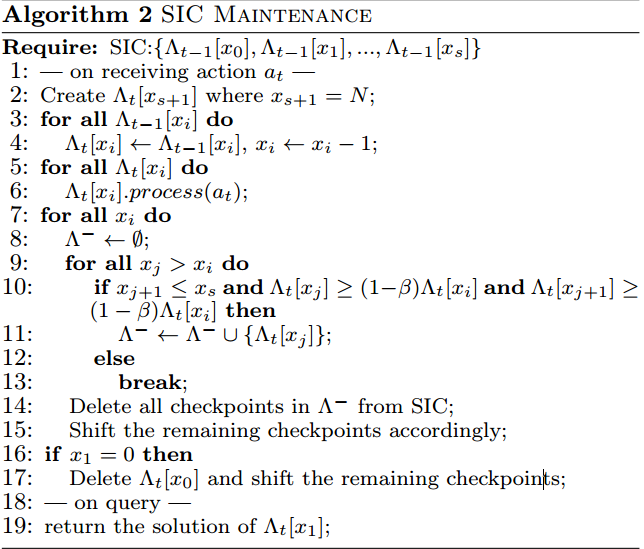
在IC框架中，应该保持N个检查点，以保证“e-近似比”。这需要很多检查点。然而，现实世界中的应用程序常常需要在一个窗口中执行数百万个操作。因此，在实践中维护所有检查点会产生高昂的成本。为了减少维护的检查点数量，从而提高更新效率，作者设计了一个影响稀疏的检查点检查点(SIC)框架，可以有选择地维护检查点的子集，而不会随着窗口的移动而丢失太多的精度。

**5.1 SIC Framework**

SIC的思想是利用检查点的子集来近似其他检查点。一方面，为了降低更新成本，维护的检查点数量应该尽可能少;另一方面，近似比应该保持较小。为了实现这两个目标，作者提出了一种策略来安全地删除当前窗口中的一些检查点，同时确保剩余的检查点能够以有界的比率逼近任何窗口。

我们考虑一系列检查点这是由SIC在时间t保持的。直观地说，给定任意三个连续的检查点，只要小于 那么就可以删掉。

算法2给出了如何在SIC框架中有效地维护滑动窗口上的检查点。

  
与IC的维护类似，在收到一个新的操作at时，我们为at(第2行)创建一个新的检查点，将Wt - 1中的所有检查点添加到Wt中，并使用at更新Wt中的所有检查点(第3-6行)。