相似性距离的合并促进交互时间序列分析

# 摘要：

在时间序列数据中寻找相似性趋势在从财务规划到政策制定等领域发挥重要作用。这些多面关系的检测，特别是不同长度和对齐的时间序列的时间扭曲匹配，是非常昂贵的计算。为了在大时间序列数据集上实现实时响应，我们提出了一种称为时间序列在线探索（ONEX）的新范例，采用强大的一次性预处理步骤，对关键的相似关系进行编码，以支持后续的快速数据探索。由于对于所有可变长度时间序列片段的大量成对相似性关系的编码是不可行的，因此我们的工作依赖于以下重要的见解：利用便宜的点到点距离的聚类，例如欧几里德距离可以支持随后的时间扭曲匹配。我们的ONEX框架克服了与更坚固的弹性距离相关的禁止的计算成本，即DTW通过在令人惊讶的紧凑知识库而不是原始数据上应用它。我们的比较研究表明，ONEX的精度提高了19％，比最先进的技术快几倍。除了是一个高度准确，快速的领域独立的解决方案，ONEX提供了一个真正的互动探索体验支持新的时间序列操作。

目录

[相似性距离的合并促进交互时间序列分析 1](#_Toc477808390)

[摘要： 1](#_Toc477808391)

[1. 介绍 2](#_Toc477808392)

[1.1 动机 2](#_Toc477808393)

[1.2 当前面临的研究挑战和局限性 3](#_Toc477808394)

[1.3 我们的ONEX方法 4](#_Toc477808395)

[2. 在ONEX使用的关键概念 5](#_Toc477808396)

[3. ONEX框架的理论基础 6](#_Toc477808397)

[3.1 ONEX相似性组 7](#_Toc477808398)

[3.2 基于ED-DTW三角不等式的时间扭曲检索 7](#_Toc477808399)

[4. ONEX基础 9](#_Toc477808400)

[4.1 ONEX基础建立的策略 9](#_Toc477808401)

[4.2 处理多个相似性参数 10](#_Toc477808402)

[4.3 ONEX Base的存储和索引 11](#_Toc477808403)

[5. ONEX在线查询处理器 12](#_Toc477808404)

[5.1 ONEX查询类 12](#_Toc477808405)

[5.2 基于ONEX的查询处理 13](#_Toc477808406)

# 介绍

## 动机

在金融，商业，医学到气象，时间序列数据是普遍存在的，表现为股票波动，心电图，降雨量等。

动态实例，让我们来看一个现实生活中的例子：揭示在寻找和利用数据相似性的决策中涉及的挑战。2013年，马萨诸塞州政府废除计算机软件服务的销售使用税，被认为会对本州的经济产生负面影响。数据驱动证据分析显示各种税率随时间的变化以及从大量政府公共网站获得的社会和经济因素的波动，所有这些都是以时间序列为模型。期间遇到了许多困难，主要的是找到和解释以时间序列表示的经济指标之间的相关性。

（1）来自不同领域的数据在特定时间间隔内的存在需要比较不同长度和序列的时间序列，因为税收变化的影响可能需要不同的持续时间才能变得明显。这种时间感知比较必须使用鲁棒距离来执行，例如动态时间规整（DTW）。这些措施的“权力”，在精度方面是有益的，被它们的计算复杂度所遮蔽，因此即使对于中等大小的数据集的时间响应也很慢。

（2）在这个过程中，分析师使用具体指标，如增长率，来评估引入新税的潜在影响。例如，他们“设计”了表示税收积极影响的样本增长率时间线，并在所有州中搜索匹配。在这种情况下，序列可能存在也可能不存在于数据集中。 如果发现完美匹配，则反映那些特定状态下的类似影响，而紧密匹配表示特定状态的影响略有不同。因此，分析人员需要能够通过使用可能存在或可能不存在于这些数据集中的样本序列来探索大时间序列数据。

（3）分析人员必须解决如何找到最佳匹配序列的复杂问题。例如，他们搜索重复的相似性模式，例如几年内一个国家的增长率以及类似的增长率和不同国家在特定时间长度上的其他经济指标，表明类似的“短期”影响。

（4）来自不同域的数据需要使用不同的参数设置，例如相似性阈值，导致每个参数设置的重复和冗余计算。 例如，用于研究人口统计数据的相似性的最合适的阈值不同于用于生长速率的阈值。

## 当前面临的研究挑战和局限性

1. **高数据基数导致响应性降低**

诸如财务记录或ECG数据收集的时间序列数据集往往是巨大的。更糟的是，适应不同时间粒度的需要要求我们考虑不同长度的时间序列。对于包含N个时间序列的数据集，每个长度n，要考虑的子序列的总数是Nn(n-1)/2。例如，一个基准数据集，如来自UCR时间序列数据挖掘Archive2的StarLightCurves，具有9236个时间序列，每个长度为1024，由4.83e9个子序列组成。现实世界的数据集往往比这个数量级大3。对所有这些子序列进行相似性比较显然是不切实际的。

许多最先进的技术试图解决这种缺乏瞬时响应性。滞后的响应性在勘探过程中可能失去分析师的注意。最先进的技术面临着准确性和时间响应之间的权衡，特别是在探索非常大的数据集时。一些系统提供精确或高度准确的解决方案，而牺牲及时的响应。这可能对在医学和财务中的应用是有害的，这取决于对行动的即时回答。其他使用预处理步骤来提高及时响应。然而，它们对设置许多不同参数的要求限制了它们的效率。

1. **支持不同长度和比对的序列的比较**

为了比较不同长度和比对的序列，必须使用鲁棒距离，例如DTW。使用DTW预先计算，存储和探索序列之间的巨大数量的成对相似性比较是不可行的，即使对于不是非常大的数据集。许多系统诉诸于接受增加的时间响应或者以降低的准确度为代价来使用更快的计算距离。

现有技术一直在抉择相似距离和时间响应性之间的折衷。许多应用依赖于使用快速计算距离，如欧几里德距离来实现快速响应时间。因此，它们不能处理未对齐或具有不同长度的序列。然而，使用时间扭曲的距离像DTW被它们的计算复杂性所掩盖，导致响应速度慢和随着数据增长而缩放。因此，应用程序必须在处理高效比较能力和由此产生的时间延迟和存储需求之间做出困难的选择。

1. **提供丰富的探索性查询**

正如我们的动态实例，洞察数据不仅涉及找到序列的最佳匹配，而且发现相似性模式。 因此，这种要求化合了先前的挑战。

由于这个原因，当前大多数系统都缺乏探索相似性的灵活性，只关注找到序列的最佳匹配。分析人员将从有机会回答与相似性相关的其他复杂问题（包括使用相同系统查找模式）获益。这将有助于他们更好地了解数据集。

1. **支持指导相似性探索和使用不同的相似性阈值**

不同域中的分析者不同地解释相似性，并且需要使用不同的参数设置，例如相似性阈值。保持每个可能的相似性参数的结果集是低效的，导致大的存储和重复计算。许多现有技术的系统不能克服支持变化的相似性参数的困难。了解使用不同参数值时时间线之间的相似性变化的方式对于分析人员至关重要。不幸的是，大多数现有系统不支持灵活的相似性见解。

大多数最先进的系统都缺乏参数建议。分析人员可能不知道将导致最有趣的相似性结果的精确参数设置。它可能需要许多连续的试错交互，使用不同的参数值从数据中获得洞察。现有的探索系统往往是“黑盒子”，没有提供参数设置的建议。一个为参数调整提供建议的系统将帮助分析人员使用更少的试错法尝试找到相似之处。包括众所周知的欧几里得距离（ED），动态时间扭曲（DTW）和基于最长公共子序列（LCSS）的距离。它们都涉及在与执行时间扭曲的能力和时间响应相关的距离的选择之间的折衷。

总之，构建用于有效探索时间序列的交互系统是非常困难的。这样的系统可能在选择相似性距离和时间响应性之间折衷。即使最有前途的解决方案专注于特定类型的查询，仍然在回答更丰富的查询类的领域中有改进的空间。

## 我们的ONEX方法

ONEX提供了一个可行的答案，在需要实时响应性和高计算复杂性之间进行权衡，由于使用非度量距离比较不同长度和排列的序列。因为使用DTW代价非常高，我们通过使用两个距离来解决这个问题：虽然我们使用计算廉价的欧几里德距离来构建特定长度的紧凑相似组，我们能够支持使用鲁棒的时间-扭曲法：DTW。这种独特的组合以非常低的响应时间速率产生非常精确的结果，因为DTW成功地应用于紧凑的ONEX基础而不是原始数据。

贡献：

1. 我们正式证明了ED和DTW之间的三角不等式，构建了ONEX基础的离线构建与其在线探索之间的概念桥梁。我们的解决方案为使用复杂时间扭曲距离和及时响应之间的权衡困境提供了一个可行的答案。 （第3.2节）。
2. 我们预先计算关键相似子序列之间的关系，并以“相似组”及其代表的形式保存（第3.1节）。所得到的ONEX基因在实现交互式反应中发挥关键作用。
3. 我们的查询策略使分析人员能够通过探索数据集的相似性及其重现模式来了解数据集。 ONEX还为分析人员提供参数调整指导，以更好地直观地反映相似性的主要变化。（第5.1节）

我们的实验性能表明，ONEX比最快的已知方法[22]快几倍，并且比[19]和[26]快许多个数量级。 此外，ONEX比传统相似性查询的[22]更准确19％（第6节）。

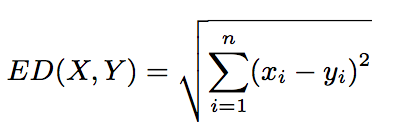
# 在ONEX使用的关键概念

时间序列X = (x1, x2, ..., xn) 是一个实数序列，数据集D ={X1, X2, … }是一个 N个时间序列的集合。

定义1: 时间序列的子序列，表示为，是从位置j开始的长度为i的时间序列，其中1≤i≤n和0≤j≤n-1。

我们在本文中讨论欧氏距离（ED）和动态时间规整（DTW）。

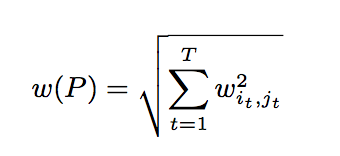
定义2: 给定两个相等长度的时间序列X =(x1 , ..., xn ) and Y = (y1 , ...yn ), 他们的欧氏距离(ED)定义为



不幸的是，两个时间序列可以是相似的，当它们不对齐时具有大的ED。 因此，已经开发了解决这个问题的距离测量动态时间扭曲（DTW）。DTW允许来自两个待比较时间序列的点之间的一对多对齐，以支持时间移位以及不同长度的序列。 ED，其对应于构成时间序列的点对的一对一映射，可以被视为DTW的特殊情况。

动态时间规整DTW:假设我们有两个时间序列X =（x1，x2，...，xn）和Y =（y1，y2，...，ym）。 为了使用DTW对准这些序列，构造n×m矩阵M（X，Y），其中矩阵的第（i，j）元素是xi和yj之间的欧几里德距离，即wi，j = ED xi，yj）。 然后，变形路径P是在从（1,1）到（n，m）的矩阵中形成路径的一组元素。 被表示为pt =（it，jt）的P的第t个元素指的是路径中的该矩阵元素的索引it，jt（xit，yjt）。 因此，路径P是P =（p1，p2，...，pt，...，pT），其中n≤T≤2n-1，p1 =（1,1）和pT =（n，m）。

定义3：给定两个时间序列X =（x1，...，xn）和Y =（y1，...，ym），规整路径P的权重定义为：

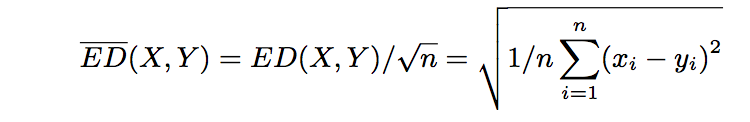


X和Y之间的DTW距离或DTW（X，Y）被定义为在所有可能的变形路径中具有最小权重的路径P的权重。(minP(w(P))。这个路径可以使用动态规划计算。更多的细节可以在[20,22]中找到。

定义4.如果它们之间的选择的距离（例如ED或DTW）在给定的相似性阈值ST内，或者Dist（X，Y）≤ST其中Dist∈{ED，DTW}，则两个时间序列X和Y是相似的。

以下归一化距离对于建立我们的时间扭曲检索的理论基础是至关重要的。

定义5.给定两个时间序列X和Y，我们将它们的归一化欧几里德距离定义为：



其中ED（X，Y）在定义2中。

定义6.给定两个时间序列X和Y，其中m≤n，它们的归一化DTW距离DTW被定义为



其中DTW（X，Y）在定义3中。

不失一般性，我们使用m≤n，但m和n是可互换的。

# ONEX框架的理论基础

这里我们为我们的ONEX框架建立一个正式的基础。 作为核心，我们证明了ED和DTW之间的三角不等式，其在概念上类似于ED的众所周知的三角不等式。 这允许我们基于ED距离来构建代表性序列的紧凑空间（称为ONEX基础），然后基于该紧凑空间而不是原始时间序列来处理基于DTW的查询，同时仍然保证检索过程的质量 。

## ONEX相似性组

我们的ONEX基础紧密地编码D中的时间序列的所有子序列之间的相似性关系。作为基础，我们将相同长度的子序列分组，根据定义4，使用普遍存在且廉价的ED距离成为所谓的“ONEX相似性组”，然后通过仔细选择的代表来概括这些组，如下文进一步描述的。

定义7.给定等长度i的时间序列的组S，然后将S的代表Rki定义为集合S [10]中的序列的逐点平均。也就是说，Rki = avg（（Xp）ij），对于所有（Xp）ij∈S.

我们现在介绍ONEX相似组的概念，通过强加几个关键要求编码子序列之间的相似性关系，正如我们在3.2证明的那样，确保这些组而不是原始数据可以通过其代表安全地探索。

定义8.给定数据集D的时间序列的所有可能子序列（X p）ij的集合T，假设这些子序列（X p）ij∈T被分组成具有它们各自代表R k1的相似性组，使得所有子序列 ij∈T在一个并且只有一个组Gik。 这些相似性组被定义为ONEX相似性组，由Gik表示，如果以下三个属性成立：

（1）组Gik中的所有子序列（Xp）ij必须具有相同长度i，

（2）Gik中任何（Xp）ij和代表之间的ED该组Gik的Rki小于相似度的一半系统使用的阈值ST，即ED（（Xp）ij，Rki））≤ST / 2，∀i，j∈[1，n]∀p∈[1，N]。

（3）（Xp）i和组Gk的代表性R k1之间的ED与（X p）i的ED相比是最小的，并且相同长度i的所有其它代表R i在D或ED（（X p） ij，Rki））≤DD（（Xp）ij，Rli））（∀i，j∈[1，n]）（∀p∈[1， g表示长度i的代表数。

将序列置于相同ONEX相似性组中的关键要求是双重的。 首先，与任何其他代表相比，ED对组的代表的序号必须小于ED，其次，它也小于ST / 2。

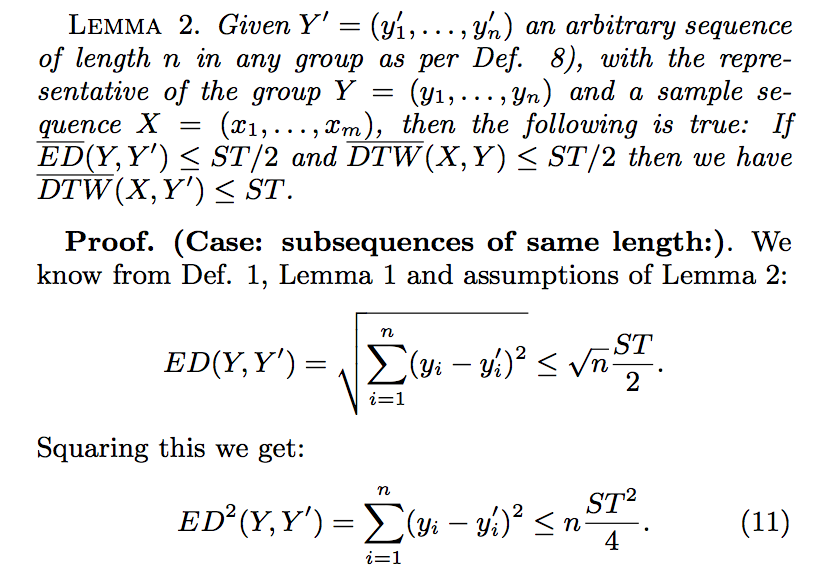
定义10.给定D中的代表集合，即R空间，则相同长度的空间的两个代表Rki和Rli之间的代表间距离Dc由Dkkl = ED（Rki，Rli）定义。

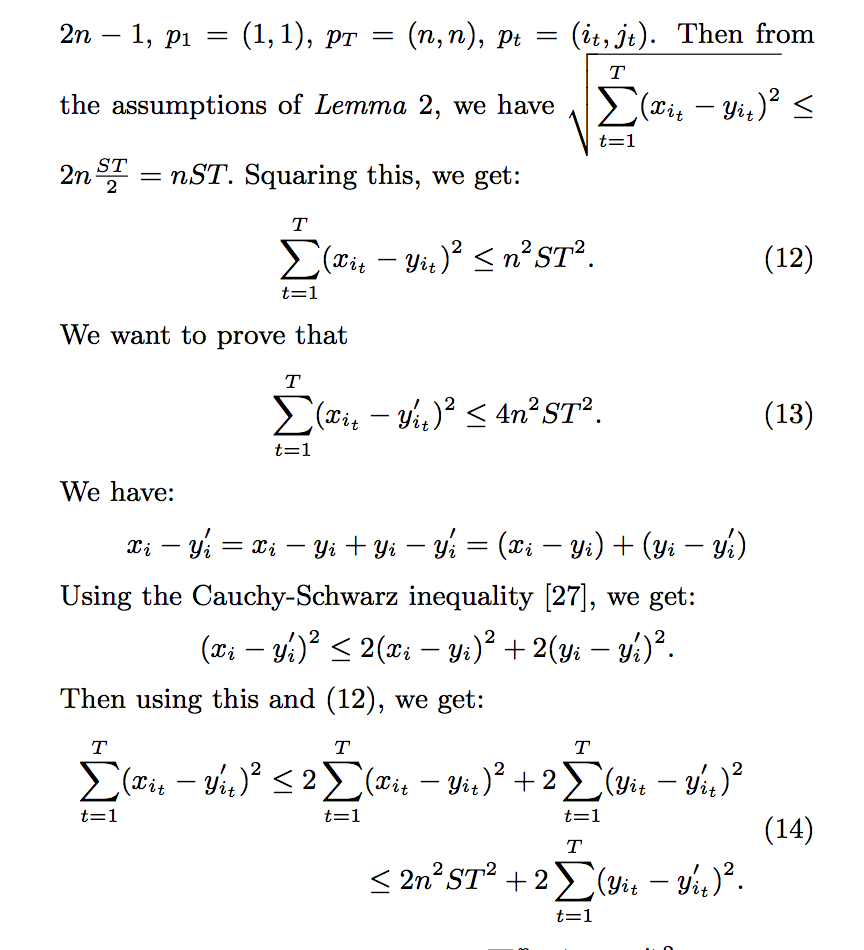
这个距离在处理可变相似性阈值中起着核心作用，如4.2章节所示。

## 基于ED-DTW三角不等式的时间扭曲检索

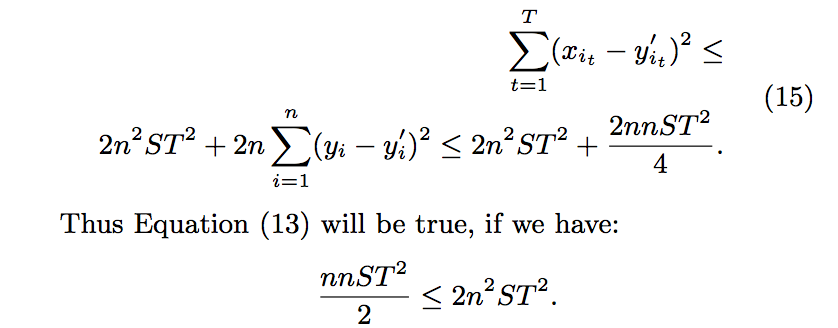
我们的ONEX时间扭曲检索框架的基石是这种独特的概念解决方案基于证明ED和DTW之间的三角不等式。 我们证明由用户提供的样本序列seq与在Def。中定义的ONEX相似性组的代表的相似性。 8“延伸”到该组中的所有子序列。 这使ONEX能够在压缩的R空间而不是整个数据集上执行样本序列的时间扭曲比较。

更具体地，如果seq和代表Rki之间的DTW小于ST / 2，则我们可以保证该组Gik中的所有序列与该序列seq相似，并且seq和这些序列中的任何序列之间的DTW在 相似度阈值ST。



我们像节一样定义矩阵M（X，Y）和M（X，Y'）鉴于与这种情况有关的假设，我们知道在从（1,1）到（n，n）的M（X，Y）中存在变形路径P，其中DTW权重最多为2n ST = nST。 我们现在有2以表示在M（X，Y'）中存在从（1,1）到（n，n）的具有至多2nST的权重的扭曲路径。 实际上，我们将示出来自M（X，Y）的相同的变形路径P将是好的。 令P为P =（p1，p2，...，pt，...，pT），其中n≤T≤2n-1，p1 =（1,1），pT =（n，n），pt =（it，jt）。 然后从T引理2的假设，我们有（xit-yit）2≤t = 12nST = nST。 平方这个，我们得到：

我们估计第二项为ni = 1（yi-yi'）2，其中一些项重复。 由于翘曲路径的长度至多为2n，因此重复的总数至多为n。 每个固定项重复最多n-1次。 因此，从等式（11）和（14），我们有：

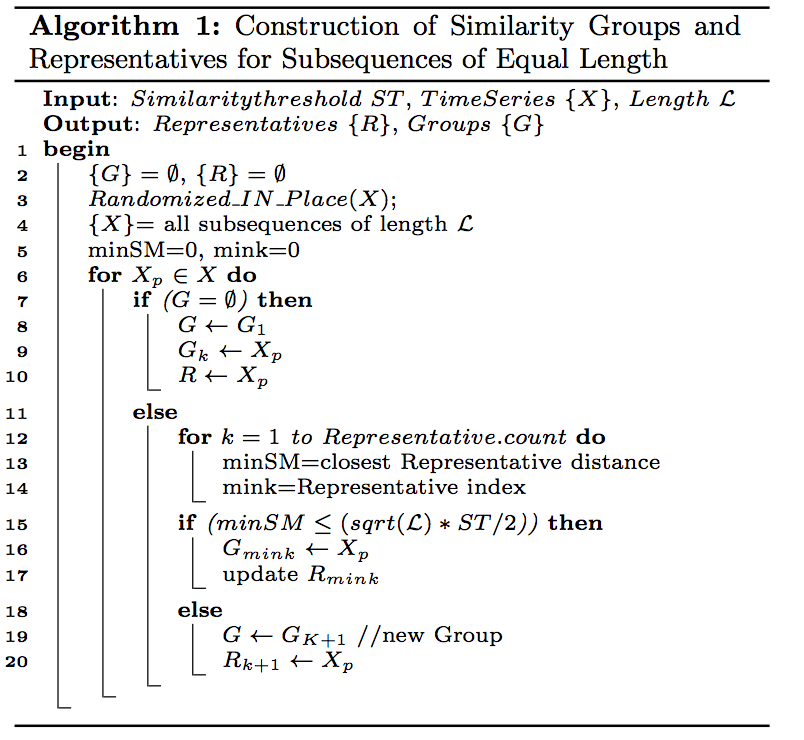


如果n2≤4n2，这也是真的，这总是真的。 证明草图（情况：不同长度的子序列）对于长度为n的Y和Y'子序列，其中Y是组的代表，Y'是组中的任意序列，X是长度为m的查询序列，m≤ n，在不失一般性的情况下，我们在这里考虑m≤n的情况，但是对于n≤m，证明非常相似。 在定义6中定义的DTW中，我们除以2n，因为变形路径可以具有高达m +n≤2n的长度。 然后矩阵M（X，Y）是m×n矩阵，并且变形路径连接（1,1）到（m，n）。 除此之外，不同长度的序列的证明和相同长度的序列的证明是相同的。

# ONEX基础

## ONEX基础建立的策略

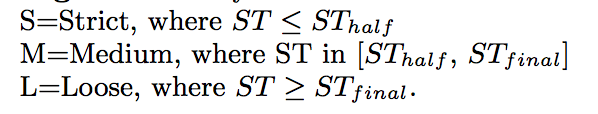
在Algo中说明了用于构建ONEX Base的算法，即，用于找到特定长度子序列的组，然后计算它们的代表。作为第一步，我们将数据集D中的现有时间序列分解为所有可能长度的子序列。然后我们使用众所周知的RANDOMIZE-IN-PLACE方法[8]随机化每个长度的子序列的顺序，以删除数据相关的偏差（1-5行）。在第7-10行中，我们通过随机选择子序列并将其指定为第一组的代表来构建第一组。选择第一个代表性随机确保组不受提供子序列的顺序偏倚[4]。在第12-20行中，将相同长度的新随机选择的子序列与先前的代表进行比较。在其ED小于ST / 2的所有代表中，选择具有最小距离ED的一个组。然后将子序列置于此组中。否则，它将被放置在新组中并被指定为新组的代表。我们重复这一过程，直到每个特定长度的所有子序列都被置于相似性组中。最后，确定每个组的代表。最终的结果是所有组Gik和他们的代表Rki的所有可能的长度的全景。 R空间构造的复杂性是O（nl2g），其中l是每个时间序列被分解成的不同长度的数量，g是组的总数，n是数据集中的时间序列的数量。典型地，n比l大得多。当然，l不倾向于走向无穷，因为我们的初始时间序列是固定长度，我们进一步将它们分成更小的数字。这与趋向于随数据集的大小而增长的n形成对比。因此，我们可以将l视为与n有关的常数。现在的复杂度可以概括为O（ng）。关于g，让我们检查下面的概率论。假设我们在某个点有k个组，并且出现了新的时间序列X.存在可以发生的（k + 1）个事件，X或者属于k个组中的一个，或者X开始新的组。让我们假设这些事件都是同样可能的。则X开始新组的概率为1 /（k + 1）。我们需要多少预期的试验，直到我们得到一个新的群体？这是具有期望值1 / p的几何分布，因此1 /（1 /（k + 1））= k + 1 [23]。因此，g k = 1（k + 1）= n;预期的组数为O（√n）。然后复杂度变为O（n3 / 2），这比O（n3）好得多。有关ONEX基础的其他讨论可以在我们的技术报告中找到。



## 处理多个相似性参数

如上所述，我们的R空间是使用特定的用户提供的ST构造的。 然而，分析人员可能希望使用不同的阈值ST'来基于应用来定制他们的相似性的概念。 因此，我们扩展了ONEX基础，以根据分析师的需要合并ST的不同值。 更精确地，在组的构建期间，我们识别对应于特定长度的相似性阈值，对于该特定长度，该长度的组显着地改变并且将它们并入相似性参数空间（SP空间）中。定义11.SP空间是具有长度L和相似性阈值ST作为其维度的概念空间。

我们基于建立相似性阈值的相关值，为每个特定长度构建SP空间。这些可以是SThalf和STfinal，指示特定长度的子序列的一半和相应地所有预计算组合并 - 导致结果集中的主要差异。一般来说，如果ST'≥ST+ Dc，则两个组合并新的相似性阈值ST'，其中Dc在定义10中定义。在图1中，我们示出三个特定长度以及该长度的一半和所有组的阈值合并的阈值。例如，对于长度i，我们有SThalf = 0.5和STfinal = 0.78。这意味着对于该特定长度，如果分析者选择新的阈值ST'= 0.5并且如果ST'= 0.78或更高则所有组合并，则组的一半合并。可以组合对于每个长度特定的这些“局部”临界阈值，以找到全部相似性阈值，对于所有长度的一半和所有组，分别合并。我们通过选择每个长度的局部值的最大值来计算这些“全局”临界阈值SThalf和STfalsal，如图2中的虚线所示。 1.将所有“本地”STfinal值的最大值选择为“全局”STfinal值确保该长度的所有组合为阈值ST'>= STfinal。我们使用局部和全局阈值来提供探索相似性的指导，如第二节所示。 5.1。我们现在介绍我们的系统的相似度的概念：



对于图1中所示的示例， 如图1所示，如果分析师要求针对“严格”相似性的相似性阈值范围的推荐，则推荐值在[0,0.6]范围内。 分析师在此间隔中选择的任何值都会返回具有“严格相似性”的结果。 在该间隔中选择的较低值导致类似序列之间的较小距离中的“更严格”相似性转换。

## ONEX Base的存储和索引

ONEX使用在线性或恒定时间实现索引的内存中结构。 我们首先构建一个全局时间索引（GTI）作为一个长度索引的数组，以便快速检索每个特定长度的组集合。 GTI中的每个条目包括：

* 包含用于特定长度i的组Gik的标识符k的向量Vi（k）。 它们用于检索与该长度相关联的组。
* 包含长度为i的组Gik和Gij之间的成对的代表间距离（Dc）的二维阵列Di（k，j）。 它们帮助细化不同用户提供的相似性阈值的结果。 （第5.2节）
* 包含代表识别符及其对于长度i的成对的代间距离Dc的相关和的数组S i（k，sum k）。 我们基于距离的总和来排序这个数组，以基于给定的样本序列来优化代表的在线搜索（第5.3节）。
* 局部参数推荐的长度i（Sec.4.2）的相似性阈值SThalf和STfinal。

对于每个组G k，我们维持本地序列索引（LSI），具有以下元件：

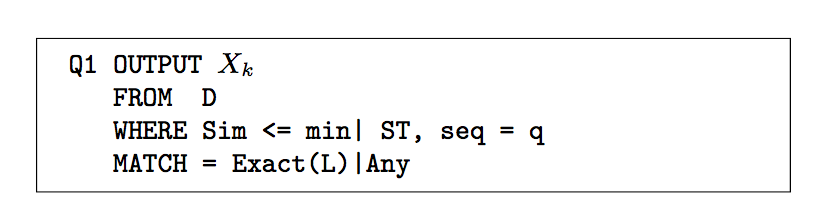
* 包含组中的子序列标识符m和它们的ED的二维数组E Dk（m，E Dm）给该组的代表。 组中的每个子序列m是由时间序列标识符p及其起始位置j进一步标识的子序列（Xp）ij。 我们基于ED将该数组排序到代表以优化在“选择”组内的序列的在线检索。
* 矢量包含组Gik的代表Rki
* 数组，其使用LB（Keogh）[22]包围每个代表性Rki周围的包络，这是用于修剪未提取候选者的公知下界。

# ONEX在线查询处理器

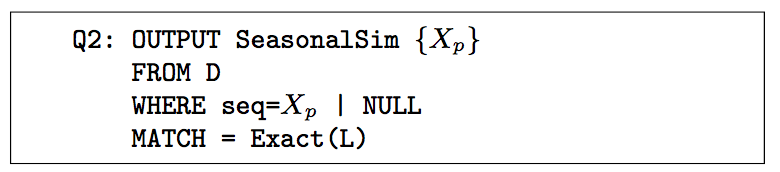
## ONEX查询类

ONEX中的用户驱动分析允许分析人员通过提供目标样本序列seq来控制探索。 数据驱动的分析提供对数据集的洞察，而用户不将样本序列作为目标匹配。

第一类：相似性查询属于用户驱动的查询类。 它们返回与用户提供的样本序列seq最相似的时间序列。 如果“Match = Any”，则搜索每个可能长度的序列，否则只考虑由样本指示的长度为l的序列。

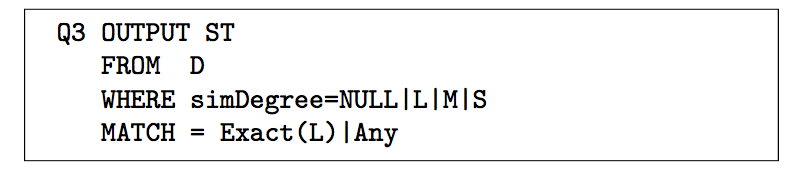


第二类：季节性相似性查询允许分析者通过识别相似性模式来获得对数据集的洞察。 在用户驱动类中，例如，给定样本时间序列的诸如Q2的查询通过检索属于该样本时间序列的特定长度的所有类似子序列来返回“重复”相似性。 在数据驱动类中，分析人员通过提供特定长度来探索数据集而不进行样本查询。 结果由具有指定长度的相似序列的组组成。



用例：对于用户驱动的类，在股市应用程序中，分析师可以找到苹果股票的所有30天长的子序列具有类似的价格。 在数据驱动类中，分析人员可以检索在任何30天内价格彼此相似的所有股票。

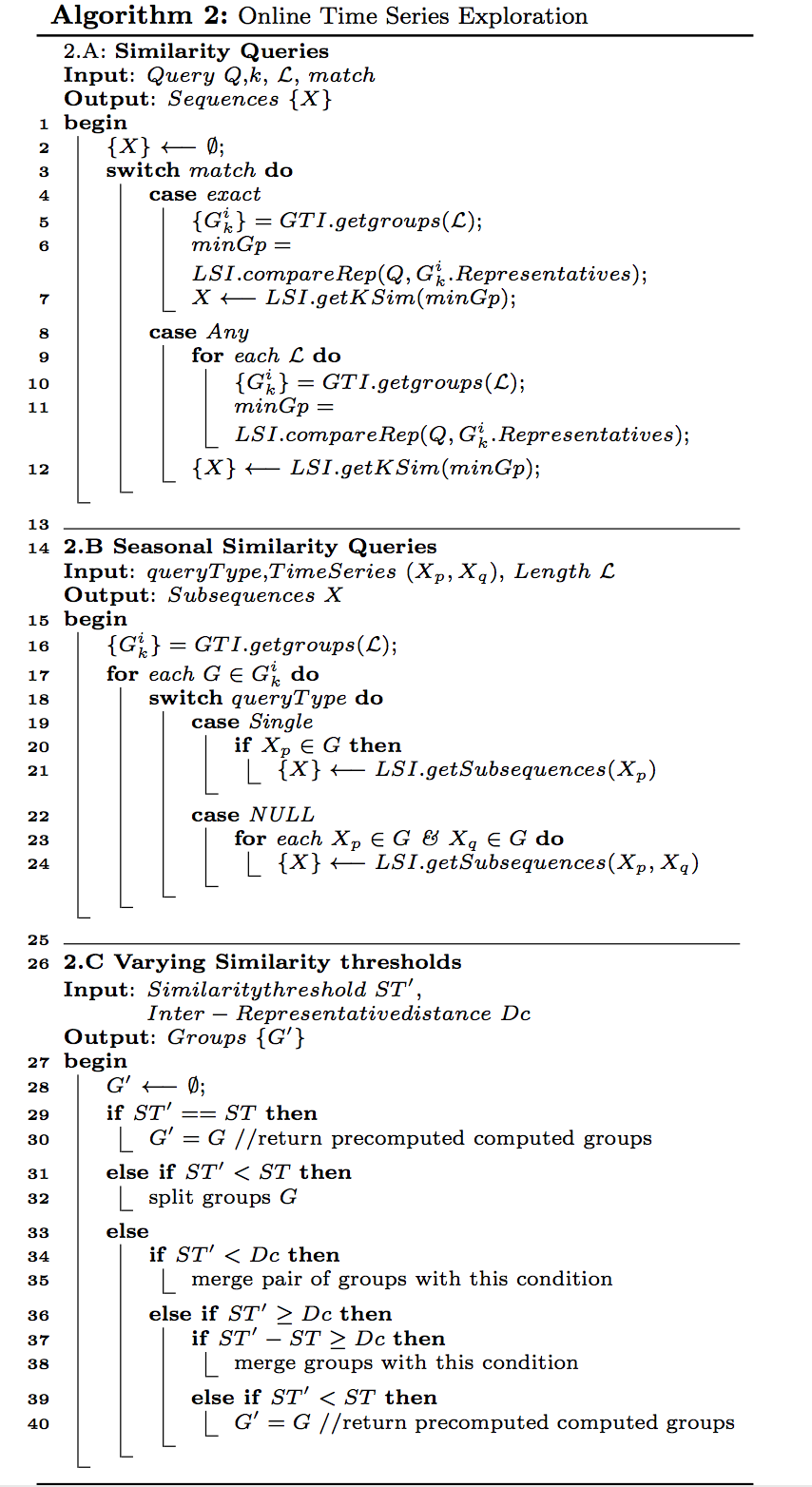
第三类：相似性阈值建议通过将“松散，中等或严格相似性”（在4.2节中介绍）的直觉转换为实际参数值，帮助用户更好地理解数据集。这些术语与基于用户和应用程序域的相似性的解释相关。



用例：有时用户使用不同的相似性阈值提交挖掘请求，但最终接收到相同或非常相似的结果。 在这种情况下，Q3通过允许用户利用将导致输出差异的相似性阈值来节省时间和精力。 例如，如果用户不确定对于相似性阈值使用什么值，则当提示他们感兴趣的相似性强度时，他们可以输入“S”用于严格相似性。系统将返回一系列相似性阈值，其中返回的子序列具有非常小的相似性距离。我们稍后在章节5.2解释如何在不重新构建整个知识库的情况下管理变化的相似性阈值。

## 基于ONEX的查询处理

基于我们扎实的形式基础（3.2节），我们的查询处理器（算法2）现在在紧凑的基于ED的ONEX基础上应用时间扭曲的策略。ONEX通过首先使用三步过程探索R空间来处理相似性查询。在第4-7行中，我们使用GTI（第4.3节）来检索特定长度的组。然后，处理器使用LSI（第4.3节）找到具有到采样序列的最小DT W的代表，即所谓的最佳匹配代表。第三，我们使用LSI来找到所选组内的最佳匹配序列。



此类的处理成本包括：Cost(getGroups), Cost(compareRep ) and Cost(getKSim ). Cost（getGroups）的复杂性对于每个特定长度是不变的。Cost(compareRep)是O(g)其中g表示组的数量。找到最佳匹配代表的成本是Cost（getKSim），其是O（m），其中m是最佳匹配组中的子序列的数量。 因此，总复杂度为O(g)+ O(m)。

对于季节相似性查询，我们区分两种情况。 在第一种情况（线19-21）中，分析者提供了样本时间序列seq和特定长度。 ONEX首先使用GTI（第4.3节）探索R空间以检索特定长度的组。 在检索组之后，基于样本时间序列的标识符使用LSI（第4.3节）来探索组。 ONEX仅返回每个组中具有与样本序列相同的序列标识符的序列。 在第二种情况（第22-24行）中，如果没有提供样本序列，ONEX将检索分析人员指定长度的类似序列组。

此类的处理成本包括：Cost(getGroups)和Cost(getSubsequence)。如4.3节所示，我们维护每个组的序列标识符。因此，我们可以检索O(1)中的所有子序列。Cost(getGroups)的复杂性对于每个特定长度是不变的。查找序列的成本Cost(getSubsequencess)是O(n)，其中n是组中子序列的数量。场景二的复杂度是O(ng)，其中n是每个组中的时间序列的数量，g是组的数量。方案一，使用样本时间序列的季节性相似性的复杂度为O(g)。在这两种情况下，没有额外的存储开销，因为所有需要的信息已经包含在索引结构中。

相似性阈值推荐的复杂度高度依赖于数据集，因为推荐是基于局部和全局相似性阈值的计算。我们通过重新使用特定ST的预计算结果来减少这种计算，并使它们适应新提供的ST'阈值。对于相似组的构建（3.1节），我们使用特定的阈值ST。在实践中，最佳阈值随数据集而变化。这种阈值提供了结果的精确度和时间响应之间的最佳折衷。我们在第二章中讨论。 6如何凭经验找到这些阈值，我们将它们用于我们的实验。然而，分析师可能有兴趣在查询中使用不同的域特定ST。在这种情况下，R空间不必从头开始重建。相反，我们提供了一个有效的策略，以“精化”相似性组，基于预先计算的信息。我们区分以下情况：

1. ST'= ST。 在这种情况下，ST匹配分析师提供的ST。 因此，预先计算的组是“按原样”使用以找到最佳匹配。
2. ST'<ST。 在这种情况下，预先计算的组包含类似的序列，但必须精炼。 细化背后的直觉是对于阈值ST类似的序列对于较小的阈值ST'保持类似。 组被分成“较小”的相似组，所以没有可能的答案。 假设不失一般性ST / ST'= k其中k∈N，每个预先计算的组现在被分成k组。 我们使用相同的方法来构建这些较小的组，如我们最初用于构造特定长度的组（3.1节）。
3. ST'>ST。 这是更复杂的情况，因为一些组对可以合并，这取决于在段落中定义的代表间距离（Dc）的值，在章节3.1中定义。
   1. ST'<Dc。在这种情况下，使用该条件为真的所有预先计算的组对来找到最佳匹配序列。直觉是，其代表彼此相距大于给定相似性阈值S T'的群组不能合并。其余的组将处理如下：
   2. ST'≥Dc。我们区分两种情况：

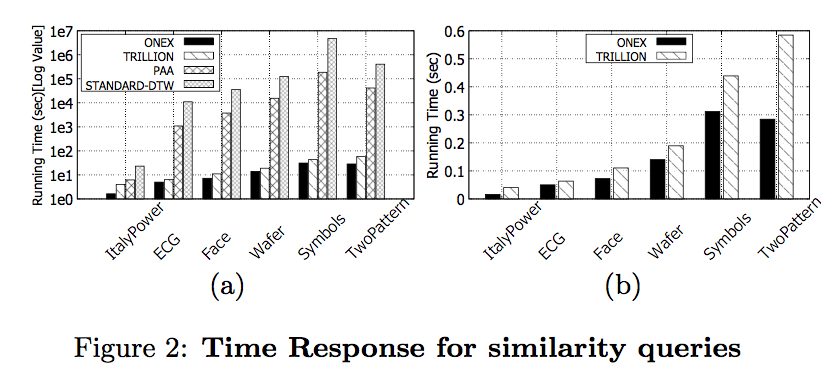
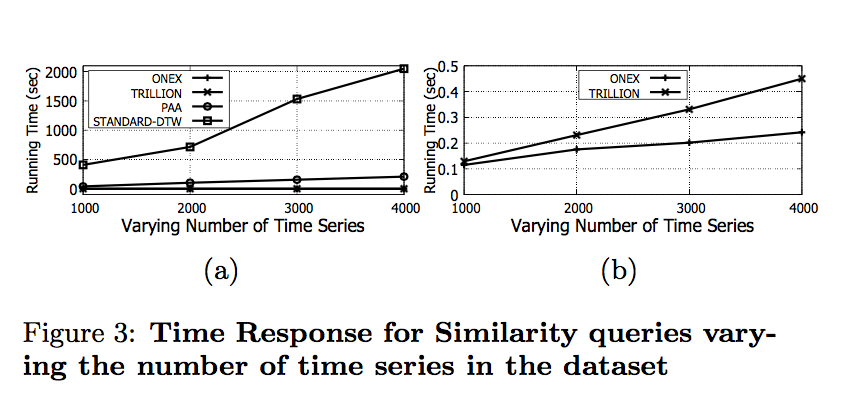
3.2a. ST'-ST≥Dc。将该条件为真的预先计算组对合并并用于找到最佳匹配seq。直觉是更高的相似性阈值允许更多的序列是相似的。将返回其余组，而不执行任何操作。如果出现相同组可以与多于一个组合并的情况，我们随机选择一对合格组并执行合并。然后，我们计算新形成的组与其余组之间的代表间距离，以确定是否需要任何额外的合并。只要上述条件成立，该过程就以级联方式重复。

3.2b. ST'-ST <Dc。在这种情况下，预先计算的组“按原样”使用以找到最佳匹配序列。

## 处理查询的优化

为了有效地检索给定样本序列seq的最佳匹配，我们采用优化来检索该组内的最佳群组代表和最佳匹配序列。 优化包括早期放弃DTW [22]，如果下限超过最佳距离，则停止计算，并修剪代表。 我们将这些与早期遗弃的重新排序相结合。 最后，我们利用级联的下限来修剪不知情的候选人，这种方法也很受DTW [11]的欢迎。 此外，我们开发了ONEX特有的以下优化策略：

* 对于长度为L的给定样本序列，我们开始搜索与查询长度相同的最佳匹配代表。 如果我们没有找到这个长度的“最佳匹配”代表（在ST / 2内具有DTW到seq的代码），我们继续通过以它们的长度的降序搜索代表，直到达到最小长度，接着是 按照长度的增加顺序搜索。
* 我们的策略找到“最佳匹配”代表一个特定的长度，同时，我们使用GTI（第4.3节）中的有序数组Si（k，sumk），其包含成对的代表间距离的和。我们找到“中位数代表”或总和是“在中间”的列表，并开始我们的搜索与这个代表。我们通过交替检查最接近的代表左边，然后右边在索引，直到我们到达代表与最小和相应的最大和。
* 具有最佳匹配代表的特定组内的搜索可以如下优化：使用包含组中的序列的ED的有序数组EDk（p，EDp）给代表，我们搜索“最佳匹配”序列 seq。 这是其代表的ED具有与seq和代表之间的DTW最接近的值的组中的序列。 如果需要，我们跟随，通过交替地检查具有较小和较大距离的相邻序列，直到找到最佳匹配序列。

# 实验评价

## 实验装置

实验在具有3.35GHz Intel Core i5处理器和64GB GB RAM的Windows机器上进行。替代技术的最先进技术。我们将ONEX与三种方法进行比较：标准DTW，它是一种计算所有成对距离并保证最佳匹配的精确强力方法，PAA（分段聚合近似）[19]，它通过减少使用平均近似的数据的维数，Trillion [22]使用下界和早期放弃步骤来检索类似序列。 Trillion展示了迄今为止最快的响应时间。他们的实验显示比EBSM [2]更快的时间相似性查询[22]和许多其他竞争对手。对于相关工作部分详细描述的这个和其他原因，我们不与EBSM比较，而是与Trillion比较。 ONEX系统和替代方法PAA [19]和标准DTW在C ++中使用带有Clang 64位编译器的Qt Creator实现。万亿码从UCRsuite4下载。

数据集：我们从UCR时间序列中选择数据集：系列收集5，公共数据集的最大集合。我们的数据集的统计可以在我们的技术报告中找到。

要在时间序列之间进行有意义的比较，他们需要归一化[22]。我们规范每个序列基于最大（最大）和最小（最小）值在每个数据集。对于任何序列X =（x1 ... xn），我们将每个点xi的归一化值作为xi-min。 max-min性能统计。我们的实验评估旨在说明ONEX是一个交互式系统，在测试的所有数据集中提供少于一秒的结果，同时产生高度精确的结果。在多次运行中查询的平均查询报告在线查询处理时间和精度（Sec.6.2）。当改变数据集和相似性阈值范围（Sec.6.3）时，我们通过测量预生成信息的大小和预处理时间来评估ONEX基础。

## 查询处理时间和精度的评估

* + 1. 相似性查询

此类检索最接近给定查询样本seq的序列。标准DTW将查询样本与数据集中的所有序列进行比较。 PAA将查询样本与降维数据集中的序列进行比较。 Trillion首先计算查询样本的下限，然后使用早期放弃优化查找与查询长度相同的最佳匹配。

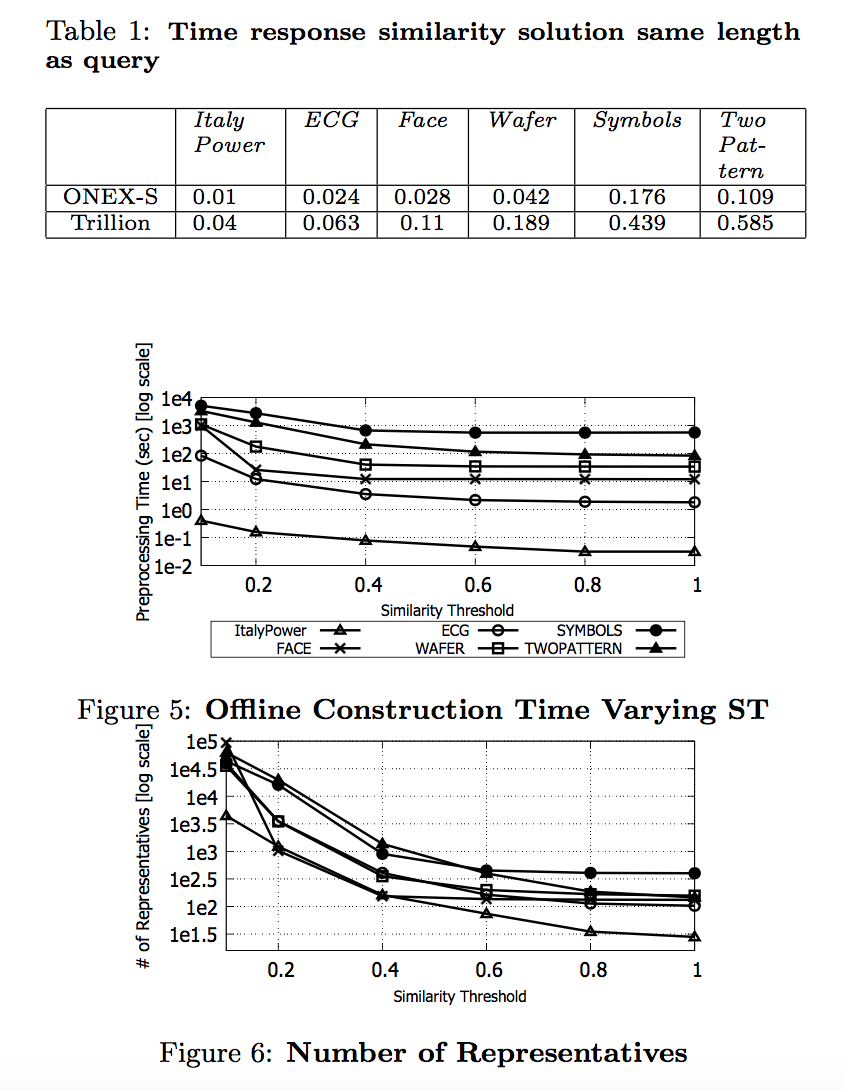
在我们的实验中，我们改变查询序列的长度以覆盖宽范围的长度。我们计算每个系统的平均响应时间，使用20个不同长度的查询，选择覆盖从最小到最大长度的宽范围。此外，我们的目的是对存在的查询序列和可能不存在于数据集中的序列进行实验。随着我们的使用案例激励，找到查询的最佳匹配很重要，无论序列是否存在于数据集中。在这两种情况下，我们寻找最佳匹配，即具有与查询序列最接近的DTW的解。首先，我们从每个数据集中随机选择10个不同长度的子序列，并“促进”它们成为查询序列。这是我们的查询“在数据集”中的实验部分。然后我们采用[13]中提出的方法，其中从每个数据集中选择随机子序列作为查询，并从该数据集中取出。我们这样做10个不同的子序列。这是我们的查询“在数据集之外”的实验部分。对于每个查询，我们在特定数据集中运行每个查询5次，并平均每个查询的时间响应。然后，我们计算每个数据集的20个查询的平均时间响应。

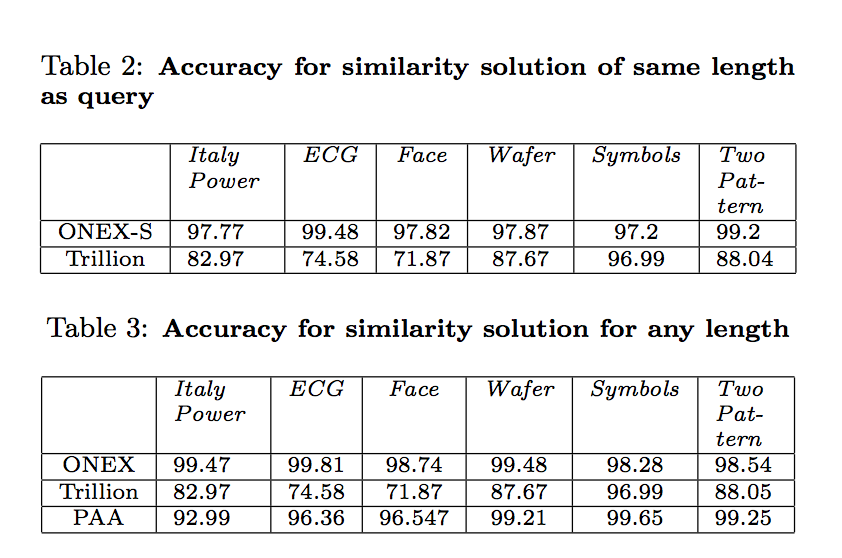
如图2所示，ONEX始终优于标准DTW和PAA几个数量级，并表现出比万亿更好的运行时间。图中的图表。图2a显示在对数标度上乘以100的值， 2b放大ONEX和Trillion的比较。虽然对于小数据集ONEX和Trillion具有相当接近的响应时间，随着数据集大小的增加，ONEX平均比Trillion快1.8倍。

此外，ONEX是高度准确的返回查询的最佳匹配，而不管匹配的长度。由于Trillion只返回与查询序列长度相同的最佳匹配，所以我们通过调整ONEX来仅搜索与查询长度相同的最佳匹配，从而设计了一个附加实验。我们发现在这种受限制的情况下，ONEX时间响应进一步提高，平均比Trillion快3.8倍。该特定实验的结果显示在表1中，其中ONEX-S表示我们的ONEX返回与类似于Trillion的查询具有相同长度的解。

接下来，我们评估可扩展性，即增加时间序列数对StarLightCurves数据集的运行时间的影响。我们随机获取包含长度为100的时间序列的子集。时间序列的数量在1000和5000之间变化，增量为1000.我们采用与上述相同的方法使用数据集中存在的和不存在的查询序列。 ONEX和Trillion的运行时间在数量上接近，因此叠加在图1中。 3a。如图3a所示，标准DTW和PAA的运行时间随着时间序列数量的增加而急剧增加，而ONEX和Trillion的时间增量“似乎”几乎是恒定的。实际上， 3b，缩放的细节显示Trillion时间响应比ONEX慢4倍。解决方案精度。我们测量解决方案的准确性如下：我们检索解决方案和查询序列之间的DTW ONEX，PAA和Trillion。我们计算检索每个系统的每个单独解的误差，作为由该系统计算的距离和由强力标准DTW提供的精确解之间的差。我们取每个单独系统的误差的平均值，并将精度计算为（1-平均（误差））\* 100。我们使用与上述实验中相同的20个查询来进行相似性查询。每个系统的平均精度显示在表3中。由于强力总是检索可能的最佳匹配

用作“准确”，我们从表中省略其结果。如表3所示，ONEX精度优于平均19.5％至万亿和2％的PAA。虽然PAA与ONEX的精度更接近，但时间响应的差异使得PAA不切实际，也就是说，ONEX比PAA快3到4个数量级。有趣的是，如果查询序列是“在数据集中”，则Trillion具有高精度，如使用前10个查询的情况。这是因为Trillion，正如其作者所描述的，执行“精确搜索”。但是，对于不存在于数据集中的查询，即使用的最后10个查询，精度显着下降。如前所述，我们还进行一个单独的实验，限制ONEX检索与查询序列长度相同的解。在表2所示的该实验中，与Trilli相比，ONEX显示增加的精度平均为12.6％





* + 1. 季节性相似性查询

此类返回属于采样时间系列的特定长度的类似子序列。标准DTW或PAA都没有被设计为搜索这种相似性，因此我们从这个实验中省略它们。 Trillion被省略，因为它的优化解决了找到与样本相同长度的仅一个最接近的序列的问题，并且不能应用于找到其相似性复现的子序列。由于长度由分析者指定，我们涵盖从最小到最大的宽范围的长度。对于用户驱动的情况，我们从每个数据集中随机选择5个时间序列，我们对每个样本使用5个不同的长度。对于每个样本时间序列，我们对每个选定的特定长度运行5次季节性相似性查询，并计算每个长度的平均响应时间。然后我们平均每个样本时间序列的那些响应时间。我们对每个数据集重复实验，并报告图2中的平均运行时间。对于数据驱动的情况，我们为每个数据集选择5个随机长度，并检索具有该长度的相似序列的组。我们对每个长度运行5次，并计算检索特定长度的组的平均时间。平均响应时间报告于图 4。我们在我们的技术报告中显示其他可视化结果。

* 1. 预处理性能评价

由于标准DTW，PAA和Trillion不涉及预处理阶段，因此我们显示所有数据集的ONEX基数的预处理时间和大小，以及不同的相似性阈值。图。 5在改变相似性阈值ST的同时显示我们的数据集的ONEX离线构建时间。如所预期的，对于低相似性阈值，构建时间较高，因为创建了许多组。随着相似性阈值增加，创建更少的组，并且因此更多的子序列被分组在一起。在一定阈值之后，施工时间保持不变。 。我们在ONEX中以不同相似性阈值的代表数量来显示预先生成的信息的大小。表4显示了代表数，突出显示所有数据集数据基数减少的子序列总数和索引大小。索引大小（以MB为单位）对应于具有每个代表所需的平均空间的代表数。相似度阈值越大，代表空间中存储的代表的数量越小（如图6所示）。例如在ItalyPower数据集中，对于ST = 0.2，代表的总数为1228。 ItalyPower数据集中的索引的平均空间为1.14 MB。 GTI使用0.83MB（组标识符向量为0.11 MB，包含代表间距离的阵列为0.718 MB，S Tfalsal和S Thalf为0.0004 MB）。 LSI需要0.305MB（对于序列标识符为0.28MB，对于代表矢量为0.0046MB，对于每个代表为上下包络为0.018MB）。如在图7和图8中的四个图中所观察到的，每个数据集具有特定的相似性阈值，其中预先生成的信息的大小和构建时间被最佳地“平衡”。我们使用这个结合观察精度和时间之间的最佳“权衡”，为特定数据集选择最合适的ST。例如，对于我们的大多数数据集，这些相似性阈值大约为0.2。我们确实使用这些阈值为我们在本文报告的实验。由于空间限制，我们只显示四个数据集的结果。

# 相关工作

许多数据表示技术已经用于时间序列挖掘。虽然已知欧氏距离对沿时间轴的失真敏感[7]，[19]，它仍然是最常用的距离[12]，[16]，[31]。它的普遍存在是由于其易于实现和时间和空间效率。时间失真的问题由包括动态时间扭曲（DTW）[5]，最长公共子序列（LCSS）[15]，具有真实惩罚的编辑距离（ERP）[6]等其他距离来解决。

为了减少DTW时间响应，索引技术[12]，[18]，[31]和其他优化，如早期废弃DTW [22]，级联下界到剪枝不成熟候选[11]，以及逆转查询/通过在查询序列周围创建一个包络而不是数据[22]。降维技术包括离散傅里叶变换（DFT）[1]，单值分解（SVD）[30]和片断聚合近似（PAA）[17]，这些技术与我们的工作是正交的， 。这种技术通常侧重于保证没有错误的解雇和精确，而不是作为他们的主要目标处理效率。一些[1]是可索引的，而另一些[17]不是，使得它们对于大型数据集不可扩展。

像ONEX，[2]使用一个预处理阶段，用于将子序列匹配转换为使用基于DTW的嵌入的向量匹配。然而，必须提供许多参数，例如参考序列的数目和分裂点的数目。

范围搜索和最近邻搜索[1]保留时间序列数据的目标查询。通过寻找对象组的代表来回答这样的查询。像最近质心分类器和k均值聚类的方法通过用它们的质心替换邻居集合来推广最近邻分类器。概念上类似，和通过分组相似的时间序列来减少数据基数。 [14]表示时间序列作为多维空间中的轨迹，并使用多尺度比较其结构相似性。最近的翘曲不变平均凝结框架[21]创建了一个基于DTW的分类器，从而提高了精度并减少了运行时的计算。这种方法使用基于序列的“DTW平均”的聚类中心，而ONEX使用ED用于聚类和代表的逐点平均序列。 [21]数据编辑使其分类器更快更准确，而我们必须保持原始数据以支持精确的相似性搜索并返回实际的序列结果。 [3]介绍了使用DTW和修改的密度峰算法构建的聚类，以提高聚类性能。使用DTW的上限和下限，TADPole可以修剪不必要的计算。然而它的目标是聚类，而不是相似性搜索，这是我们手头的任务。

虽然ONEX采用从[14]和[9]中寻找对象组的代表的一般思想，其探索相似性组的解决方案是独特的，组合两个已知的距离，并且显示出与状态 - 最先进的技术。

由于空间限制，我们提供了额外的一般背景信息和符号，以及关于替代聚类方法，ONEX基础的维护，一般查询语法和季节性相似性实验的讨论：

<https://github.com/onexAdditional/ONEX_Additional>

# 结论

ONEX作为一个真正的互动时间序列探索系统。基于两个相似距离的组合的我们的独特方法导致与最快的已知的最先进的方法相比，精确度提高高达19％和高达3.8倍的较短时间响应。 ONEX使大型时间序列数据集的探索更加实用，并通过支持新的操作类别帮助分析人员更好地理解相似性。

**致谢:** 我们感谢UC Riverside的Eamonn Keogh博士提供的宝贵的，始终及时的反馈，本文的详细建议，以及共享数据集和代码。我们感谢来自WPI的M. Andrews和S. Swartz帮助实施，以及来自阿灵顿分校材料的Athitsos博士。我们感谢Fulbright和CS部门在WPI的资金支持，NSF的补助IIS-1018443和CRI-1305258。