

通过 Latent 进行多模态金融时间序列检索 太空投影

Tom Bamford*
摩根大通人工智能研究

安德里亚·科莱塔 (Andrea
Coletta)*摩根大通人工智能研究部

Elizabeth Fons*
摩根大通人工智能研究

Sriram Gopalakrishnan*摩
根大通人工智能研究

Svitlana Vyetrenko摩
根大通人工智能研究

Tucker Balch
摩根大通人工智能研究

Manuela Veloso
摩根大通人工智能研究

抽象的

金融公司通常会处理和存储数十亿个连续且高频率生成的时间序列数据。为了支持高效的数据存储和检索,出现了专门的时间序列数据库和系统。这些数据库支持通过类似结构化查询语言 (SQL)的格式对时间序列进行索引和查询,以实现诸如“每月价格回报率大于 5% 的股票”之类的查询,并以严格的格式表示。然而,此类查询无法捕获高维时间序列数据的内在复杂性,而高维时间序列数据通常可以通过图像或语言更好地描述(例如,“处于低波动性状态的股票”)。此外,在时间序列空间中搜索所需的存储、计算时间和检索复杂性通常是不小的。在本文中,我们提出并演示了一个框架来存储金融时间序列的多模式数据

使用深度编码器的低维潜在空间,使得潜在空间投影不仅捕获时间序列趋势,还捕获金融时间序列数据的其他所需信息或属性(例如价格波动性)。此外,我们的方法允许用户友好的查询界面,支持自然语言文本或时间序列草图,为此我们开发了直观的界面。我们展示了我们的方法在真实历史数据以及合成数据的计算效率和准确性方面的优势,并强调了潜在空间投影在通过直观的查询方式存储和检索金融时间序列数据中的效用。

关键词

时间序列、数据集、神经网络、文本标记

ACM参考格式:

汤姆·班福德、安德里亚·科莱塔、伊丽莎白·丰斯、斯里拉姆·戈帕拉克里希南、斯维特拉娜·维特连科、塔克·巴尔奇和曼努埃拉·维罗索。 2023. 多式联运

*所有作者对本研究的贡献均等。

允许免费制作本作品全部或部分内容的数字或硬拷贝以供个人或课堂使用,前提是制作或分发副本不是为了盈利或商业利益,并且副本在首页上附有此通知和完整引用。必须尊重 ACM 以外的其他人拥有的本作品组件的版权。允许以信用方式提取。要以其他方式复制、重新发布、发布到服务器上或重新分发到列表,需要事先获得特定许可和/或付费。从permissions@acm.org 请求权限。

ICAIF 23,2023 年 11 月 27-29 日,纽约州纽约 ©
2023 计算机协会。
ACM ISBN 978-1-4503-XXXX-X/18/06。 。 15.00
美元<https://doi.org/XXXXXXX.XXXXXXX>

通过潜在空间投影进行金融时间序列检索。在会议记录中,请确保从您的权利确认电子邮件 (ICAIF 23) 中输入正确的会议标题。ACM,美国纽约州纽约市,9 页。 <https://doi.org/XXXXXXX.XXXXXXX>

1 简介

包括 AI/ML 算法在内的数据密集型应用程序的使用量不断增加,为适应不断增长的数据量、速度、数据种类和查询所需的存储系统带来了重大变化。特别是,金融公司需要处理和存储数十亿的时间序列 (TS)。存储这些 TS 是为了满足合规性要求,为业务和客户提供历史数据,并支持数据驱动的算法。此类 TS 经常根据不同的要求和相应的查询进行访问。为了支持高效的数据存储和检索,出现了专门的 TS 数据库和系统[15,18,19,22,32,40]。

虽然这些数据库支持对 TS 进行高效的索引和查询,但它们并没有直接针对金融 TS 的两个重要方面:高频金融数据的内在复杂性;以及最终用户所需的检索方式和难度。事实上,此类数据库通常提供基于某些固定属性集并使用 SQL 样式查询的查询;例如,“每日价格回报率≤ 5 % 的股票”。然而,实际价格序列的统计属性对于用户以这种严格的格式表达来说可能并不简单并且具有挑战性[6]。因此,金融用户需要一系列快速、简单的查询模式,以促进金融 TS 特征的输入,称为程式化事实 (例如,波动性或相关性) [35]。

一些学术工作提出了以下方法: 分析和检索大量异构数据集;管理来自物联网 (IoT) 设备的 TS 数据流;或者在历史数据库中找到不常见且有趣的模式[5,21,41]。然而,据我们所知,这些工作侧重于其特定的应用或问题。它们不针对或适用于金融 TS,金融 TS 有其自己特定的统计属性和要求 [6]。

由于缺乏针对金融领域的专用解决方案,并利用机器学习技术的最新进展,我们提出了一个使用金融 TS 的潜在空间投影进行高效且简单的多模态 TS 检索的框架。特别是,我们的主要贡献是使用深度编码器网络跨不同模式存储和检索金融 TS 数据,并在潜在空间中保留与金融相关的信息的框架。我们的方法使人们能够学习

多模态数据（例如文本到 TS）之间共享嵌入空间,并允许使用不同的查询模式（自然语言、TS 草图、图像等）进行检索。我们使用这样一个学习到的嵌入空间来索引历史 TS 数据（即,我们创建一个<编码,历史数据>的数据库）。然后,数据库可以用于使用不同的输入方式检索 TS,并且我们使用真实的历史数据和合成数据来评估我们的方法。

我们在深度编码器的一般方法的两个实例中展示了我们的框架;一种用于通过自然语言查询（基于文本的检索）,一种用于通过 TS 草图查询（基于草图的检索）。这些模式的界面如图 1 所示。用户可以通过文本查询来搜索历史 TS:“波动性高且价格不断上涨的股票”（图 1a,我们的编码器经过训练,可以关注相关的财务描述符,例如“高波动性且价格上涨”）。或者,人们可以绘制 TS 的总体草图作为搜索输入,因为它通常最容易可视化感兴趣的 TS 的趋势（例如图 1b）。

我们基于文本的检索方法利用预训练的语言编码器和预训练的视觉编码器来组合视觉和文本特征,其潜在空间在微调阶段使用余弦相似度进行对齐。所提出的系统源自 OpenAI CLIP 网络[28],该网络在基于内容的图像检索方面表现出了最先进的性能。为了合并基于草图的功能,我们的第二个模型使用自动编码器将 TS 草图和草图中看到的波动趋势编码到潜在空间中,因此后续查询不仅可以与趋势信息相匹配。这不仅限于波动性,还可以扩展到价格 TS 数据的不同属性。

我们通过实验评估我们的方法,并展示其在计算时间和准确性方面的优势。我们将其与多个基线进行比较,包括标准 TS 检索方法和使用其他降维方法（特别是统一流形逼近和投影（UMAP）[25]）的潜在空间检索。我们表明,我们的方法可以减少检索时间,并提供比基线更好的竞争性准确性,同时支持用户友好的查询方式。最后,值得一提的是我们方法的灵活性,它可以支持其他模式。

总而言之,我们的工作贡献如下: 我们引入了一个针对金融交易系统的多模态检索深度编码框架。我们的框架通过明确编码金融 TS 的程序化事实来适应金融领域。 我们演示了基于文本和基于草图的检索模式框架的功能

我们评估并展示我们工作的好处: 通过将它们与现有基线和方法进行比较来确定推断时间和准确性。

2 相关工作

在本节中,我们简要概述了 TS 检索方面的现有学术和商业工作。

商业和开源 TS 数据库。在最著名的面向 TS 的数据库中,我们发现了 InfluxDB [19],这是一种流行的开源数据库,具有高性能存储和检索功能

对于 TS,提供连续的类似 SQL 的查询和额外的特定 TS 功能,如下采样。 TimescaleDB [18]是一个构建在 PostgreSQL 之上的开源数据库,具有可靠性和健壮性,同时为 TS 提供新的存储优化。

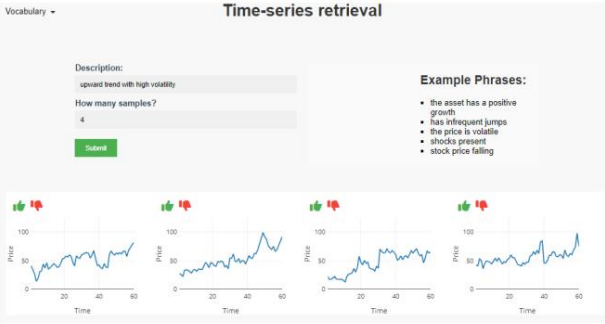
最著名的商业 TS 数据库之一是 KDB+ [22],它通过将数据组织到分层内存中来支持高频数据分析,从而实现最先进的性能和更低的硬件成本[4]。最后,亚马逊最近推出了 Timestream [40],这是一种云无服务器 TS 数据库服务,用于优化内存和历史数据,同时还为用户提供包括平滑、近似和插值在内的 TS 分析工具。这些数据库大多数都专注于 TS 数据的高效索引和存储。它们将大多数较旧的 TS 存储在更具弹性和高容量的内存中,而仅在内存中保留较新的部分数据。然而,与它们不同的是,它们并不专注于提供多模式查询来方便最终用户检索复杂的金融 TS 数据。事实上,现有的数据库通常只支持复杂且缓慢的查询,而我们的工作旨在为金融 TS 检索提供直观的模式,例如自然语言输入和趋势草图。

研究文献。多模态搜索的现有工作考虑如何结合文本和图像等不同模式来查询数据库[34] [37]。然而,据我们所知,我们的工作第一个针对金融 TS 数据的多模态 TS 检索的工作。我们在方法中采用的一般方法是深度编码器,经过修改以编码金融 TS 的相关信息,允许人们使用自然语言文本或 TS 草图进行查询。现有的文献工作大多只关注特定问题（例如,多变量二进制编码和检索 [33, 43],或树索引结构[3]）或目标特定领域相关问题[2, 23]。 L.Bitincka 等人。 [5]提出了一个可用于索引、搜索和分析海量异构数据集的数据库; Y.杨等人。 [41]关注来自物联网（IoT）设备的 TS 数据流及其在边缘节点的管理; 和 E. Keogh 等人。 [21]讨论对 TS 的分析,以发现历史数据库中不常见或有趣的模式。

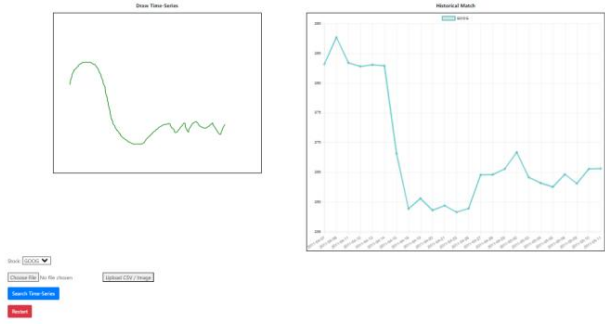
然而,据我们所知,以前的工作并不针对或适用于多模式金融 TS 检索,多模式金融 TS 检索具有特定的要求,例如纳入金融数据 [6] 的相关统计属性（例如价格波动性）。

对于基于文本的 TS 检索,该领域之前的工作很少。然而,已经做出了一些解决这个问题的相关尝试[1, 17]。在这两种情况下,都引入了与 TS 的特定数字特征相关联的有限的预定义词汇表。然后可以使用这样的预定义字典将文本查询转换为数字搜索以检索相关的 TS。所使用词汇的有限性、笨重性和相对不直观性使得这些方法在所涵盖的查询范围内与我们的深度编码方法相比明显缺乏,特别是在样本外或临时查询方面。此外,所提出的方法与领域无关;我们的方法既可以以与领域无关的方式应用,也可以专门针对金融 TS 进行定制。

关于基于草图的 TS 检索,之前有针对此问题的工作[38] [24]。然而,在这些方法中,这些方法仅匹配趋势和人工制品（形状）



(a) 基于文本的 TS 检索的原型接口;允许用户描述搜索历史数据时的TS属性



(b) 基于草图的 TS 检索的原型接口;允许用户绘制假设 TS 的草图以搜索历史数据

图 1:为自然语言和基于草图的查询开发的界面。

并且不考虑任何其他属性,例如波动性 TS (我们这样做)。我们的基线方法涵盖了以下方法在上述工作中使用,我们修改基线来尝试并将波动性纳入实验中的这些方法中。

3 问题描述

我们首先将 TS 空间定义为 \mathcal{R} , 其中 \mathcal{R} 是 每个 TS。我们的问题集中在从大量数据中检索 TS 输入数据集 $\mathcal{D} = \{x_i\}_{i=1}^n$ 时间序列 ($x_i \in \mathbb{R}^{[1..T]}$), 使用不同的方式。形式上,我们的目标是学习一个函数 $f(\epsilon, D, x) \rightarrow \mathcal{C}$ 给定输入查询 x , 检索 来自 \mathcal{D} 的最佳匹配 TS。例如,基于文本的输入 “A 波动性高且价格上涨的股票”应该检索 TS 具有此类描述统计特性。特别是金融 TS 具有特定的统计特性;价格序列数据 (我们关注的 on) 特别具有从 市场微观中个人行为互动的机制 结构。这种统计特性通常被称为程式化 事实 [6, 35]。用现有的表达方式可能具有挑战性 TS 数据的查询框架。我们的目标是支持高效、 直观的查询方式使用用户能够检索匹配的 TS 不仅是趋势,还有与财务相关的属性,例如 价格波动。最重要的是,我们解决了多模态检索的问题,使用户能够使用不同的 方式检索 TS 输入类型 (例如文本、草图、图像等)。

在下一节中,我们将描述我们采取的一般方法 启用多模式检索。为了简单起见,我们只考虑 单变量 TS, 尽管该方法可以推广到多变量 TS。同样,我们将检索方式限制为 使用自然语言的基于文本的搜索和基于 TS 草图的搜索,

尽管我们在这里采用的相同方法也可以应用于 其他形式,例如图像。

4 数据集构建

4.1 综合股价时间序列数据集

没有与金融 TS 配对的现有数据集 相应的文字描述。此类数据集对于训练基于文本的检索模型至关重要,该模型可以有效地提取有意义的信息并在之间建立联系

这两种数据类型。因此,我们可以生成合成 TS 模拟股票及其相应的文字描述 价格 [6, 10, 12], 例如使用深度生成模型 [11, 42]。 具体来说,为了生成合成股票数据,我们利用离散 均值回归 TS 经常用于模拟金融市场 [36] 以及生物过程 [27], 并由

下面的等式:

$$x_t = \max\{0, \bar{x} + (1 - \alpha) (x_{t-1} - \bar{x}) + \sigma N(0, 1), 0\}$$

其中 \bar{x} 是 TS 的平均值, α 是均值回归参数 $N(0, 1)$ 是每次添加到 TS 的随机噪声 步。我们带来股票方向性和大的可能性 通过引入上述生成过程的冲击发生 趋势和大冲击的概念。趋势添加到 每个时间步长指示股票价值的上涨或下跌。 如 [9] 中所示, 特大冲击旨在代表外源性 不经常发生且可能产生重大影响的事件 生成过程。从数学上讲, 巨大的冲击可能会到来 任意时刻发生的概率并被绘制 来自 $N(0, \sigma^2)$, 其中 $\sigma^2 \gg 1$ 。一旦时间序列是 生成后, 它们通过在 a 上绘制而转换为 224x224 图像 预定义域。我们测试了三种不同的自动生成字幕的方法。

未经过滤。我们将每个参数的数值关联起来 $\bar{x}, \alpha, \sigma^2$, 带有描述它的情感。例如, 接近于零的趋势值可以描述为 “中性、水平、不增加、平坦、稳定、不变”; 较大的正值

可以描述为 “向上、成长、积极、增加、 上升、攀爬、前进”; 较小的负值可以是 形容为 “下降、坠落、滑动、下沉、直线下降、向下”。同样, 对于高价值的股票 价格 TS 可以是 , 被描述为 “具有很强的可变性”, “具有显著的变化”, “具有剧烈的变化”、“不稳定”、“具有高波动性”、“是 嘈杂”, “是可变的”; 而对于生成的低值 TS 可以被贴上 “波动性较小”、“股票表现出轻微波动”的标签。 波动性”、“股票的波动性可以忽略不计”、“波动性较低”、“ 价格保持稳定”。对于给定的图像, 我们从列表中采样 由 3-5 个语义相似的短语来生成特定短语

对于每个相应的地面实况政权;然后将每个选定的特征短语连接起来,给出最终的真实标题。

过滤。由于我们的合成模型中参数之间的随机性和相互依赖性,在某些情况下基于参数值自动生成字幕会导致相对于基于简单“眼睛测试”的预期的令人惊讶的字幕结果。

我们发现这一点在波动性较高的环境中尤其突出。为了确保字幕的一致性,我们执行后处理过滤步骤,其中生成的 TS 根据数学上预先定义的条件在趋势和冲击概率范围中重新标记。对于趋势,我们将线性曲线拟合到该序列,并根据相对于阈值的拟合梯度评估适当的机制。对于冲击状态过滤,我们评估每个点处生成的 TS 的梯度,并根据高于阈值幅度的任何梯度值的存在来分配冲击状态。

对于历史数据集,我们扩展了过滤过程,还包括波动性后处理检查。在这种情况下,波动率机制是根据相对于运行 TS 值的平均偏差来分配的。人们发现,与使用回报标准差的传统金融波动性定义相比,这提供了更一致的描述。

过滤+。鉴于每个参数配置的手工文本描述数量有限,我们使用 ChatGPT 增强了文本描述。为此,我们为 Chat- GPT 提供了一些针对每个特征机制的预定义句子,并要求它提出大量语义相似的短语。

然后将它们保存到文本文件中,从中可以绘制替代标题。总的来说,每个特征体系的数据集增加了大约 60-80 个短语,短语总数从 36 个增加到 500 多个。这使我们能够相应地增加 TS 数据集的大小并减少过度拟合的可能性。由于数据集创建的综合性质,我们能够轻松扩展到更大的数据集大小,从 4000 个样本到 16000 个样本进行训练。

众所周知,低流动性股票的波动性较高,而高流动性股票的波动性较低 [6]。因此,我们可以扩展上面生成的数据集的描述,以根据波动率概况区分“高流动性”和“低流动性”股票。最后请注意,附加到每个图像的标题的格式是关于每个感兴趣的特征的逗号分隔的声明。

4.2 历史价格时间序列数据集对于我们基于文本的检索方法,我们使用雅虎财经的开源 Python 包收集了历史股票数据。

选择的股票代码为 GOOG、AMD、INBX 和 DAWN,分别代表高波动性、中波动性和低波动性(后两种)制度。我们使用指定窗口大小的重叠采样构建了具有 60 个时间步长的 TS 数据集。对于较小的数据集,我们仅使用 GOOG、AMD 和 INBX 股票来总共提供 1500 个 TS,而对于较大的数据集,我们包含用于低波动性设置的附加 DAWN 数据,并修改窗口大小以使 TS 数量加倍到 3000。与综合数据集一样,分配的标题是逗号分隔的语句

关于每个感兴趣的特征,并使用固定格式通过绘图将 TS 转换为 224x224 图像。

对于我们基于草图的检索方法,我们类似地收集了 GOOG、AMD 和 INBX 的股票数据,每个轨迹的长度为 30(对应于 1 个月),并且以 5 个时间步长的增量获得连续轨迹。该数据集有 1516 个迹线,我们发现这些迹线足以显示基线方法和使用自动编码器(AE)之间的相对性能成本。

5 方法论

在本节中,我们将介绍基于文本的 TS 检索和基于草图的 TS 检索方法。图 2 显示了提议的总体框架。特别是,我们建议使用深度编码器网络 [14],该网络经过训练可以学习多模态共享潜在空间。学习这种潜在空间投影可以实现不同模态之间的转换,从而从不同查询模态中检索 TS(图 2 - 训练)。我们可以创建一个专用数据库,使用此类潜在空间投影进行索引,稍后我们将使用该数据库来启用用户查询(图 2 - 数据库)。

通过训练编码器将不同的输入模态投影到存储目标数据库 TS 的同一潜在空间,我们可以支持不同的查询模式(如文本、图像、草图)并检索匹配的 TS(图 2 - 检索)。

5.1 基于文本的检索

在我们基于文本的检索方法中,我们的模型允许用户通过所需属性的自然语言描述来执行 TS 检索。我们现在将描述我们的方法。

5.1.1 架构。为了支持不同数据模式的检索功能,我们需要一种方法来学习跨模式的一致表示。为此,我们实现了 [28] 提出的 CLIP 模型作为深度编码器,它利用了一组单模态神经网络编码器,在大型通用数据集上进行了预训练。编码器输出其输入数据的可变大小的特征表示,在这项工作中对应于相关模态中给定 TS 的描述。然后,这些特征表示被输入到单层投影头中,该投影头将表示转换为一致的维度,并允许在训练期间通过参数更新进行微调。综上所述,我们可以使用对比损失来学习不同模式的共享空间。

请注意,我们模型的优点之一是,对于数据库中的给定条目,编码器输出的嵌入保持固定。这意味着这些嵌入可以在第一次评估后保存到查找表中。在检索时,在单个矢量化操作中快速计算单个查询嵌入并快速计算余弦相似度,这显着加快了检索计算速度。

对于这种基于文本的检索方法,我们需要将 TS 数据和自然语言文本投影到同一空间中。我们通过将 TS 数据编码为图像并将其嵌入与文本相同的空间来实现此目的。现在我们将回顾执行此操作的两个编码器。

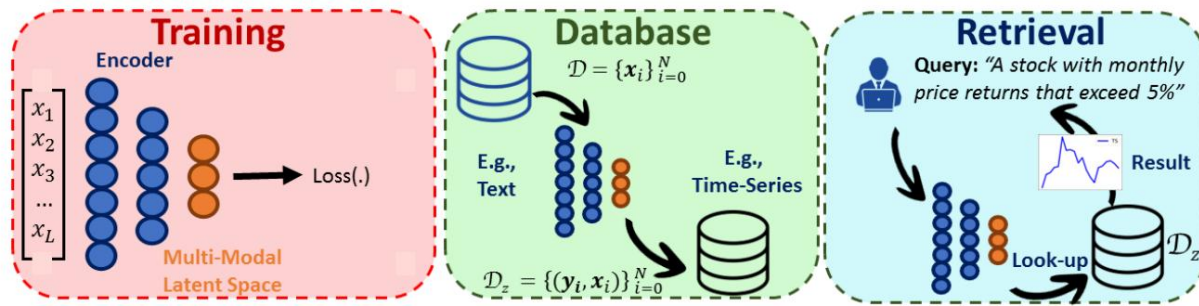


图 2:两种检索方法的通用模型架构。深度编码器使用不同模式的数据进行训练,以防止损失,以创建共享的潜在空间。创建包含 TS 和潜在空间索引的数据库。最后,用户查询通过经过训练的编码器转换到潜在空间中以查询数据库,并检索匹配的TS。

图像编码器。通过将我们的 TS 数据表示为图像,我们可以利用预训练的深度卷积网络 (CNN) 的功能,该网络在基于图像的学习任务上表现出了卓越的性能。在这项工作中,我们使用 2015 年推出的现成的 ResNet 模型 [16]。其架构由一系列残差块组成,其中每个块对应一个小型 CNN,后跟输入的逐元素相加 (从而学习残差函数),通过跳跃连接实现。

这有助于优化过程的稳定性 - 残差函数比未引用的函数更容易学习 - 并且允许计算机视觉中以前未见过的网络深度。

ResNet 存在多种变体,具体取决于其深度;这里我们使用ResNet50模型,它由50层组成,在计算效率和模型性能之间取得了折衷。该模型在 ImageNet 分类数据集[30]上进行训练,该数据集由 128 万张训练图像组成,分为 1000 个类别。它以 224×224 像素图像作为输入,并输出相应的 2048 维向量表示。

文本编码器,对于文本编码,我们利用 Sentence-BERT ([29]), 其旨在通过其句子转换器模型来学习语义上有意义的句子嵌入,该模型使用组合将单词(句子和段落)集合映射到单个嵌入。连体网络[8]和三元组损失[31],该模型是更广为人知的BERT(来自 Transformers 的双向编码器表示,[13])的替代方案,在句子级别而不是基于原始单词/子词的方法进行标记化。通过这种架构,该模型能够学习文本嵌入,其中语义相似的句子靠近在一起。我们使用预训练的 SBERT 模型,该模型已在超过 2500M 单词的英语维基百科数据集上初始化(通过底层预训练的 BERT 模型),并通过 SNLI 进一步微调以实现有意义的句子级嵌入[7]和多流派 NLI [39]数据集。

5.1.2 培训。接下来我们讨论编码器是如何训练的。

对于给定批次的图像、标题对及其相应的嵌入 v 和 t , 我们首先计算所有图像文本对之间的缩放余弦相似度

$$S \neq S(t, v^{\text{时间}} v) = \frac{T t}{\dots},$$

评估嵌入空间中数据各自模态的接近度S。然后计算每个元素各自的单模态相似度

$$S(t, t^{\text{时间}} \cdot t) = \frac{Tt}{\text{电}} \quad \text{和} \quad S(v, v^{\text{时间}}) = \frac{\text{视} \cdot v}{\text{电}},$$

对其进行平均以给出目标相似度

$$S_{\text{目标}} = \text{softmax} \frac{S(t, tT) + S(v, vT)}{2}$$

最终损失L是对交叉熵进行平均计算的
每个单模态嵌入与该目标相似度之间

$$L = \frac{LCE(S, \text{Starget}) + LCE(S \overset{\text{时间}}{T}, S \overset{\text{目标}}{\text{目标}})}{2}。$$

其作用是将每种模态的嵌入空间驱动到一致的共享空间。注意上述计算中使用的缩放参数（温度）；这用于衡量相似性，以便在验证过程中调整预测的清晰度。另请注意，我们在训练期间保留每个数据集的 10% 进行验证。

5.2 基于草图的检索

现在我们讨论如何为基于草图的检索进行编码。

我们基于草图的检索使用户能够绘制一个 TS,并根据绘制的趋势及其中的属性检索类似的 TS (如图1.b 所示)。虽然基于草图的方法可以将趋势中的任何统计属性 (x)纳入搜索中,但为了讨论简单起见,除了匹配价格趋势之外,我们仅关注包含草图中的价格波动性的搜索。我们使用价格的标准差来衡量波动性。我们通过计算与所需属性相关的附加 TS 将波动性信息合并到潜在空间中;我们计算输入 TS 的固定窗口中的波动性,然后在原始 TS 上滑动窗口以生成新的“波动性TS”。通过这样做,我们可以使用与原始 TS 相同的编码架构。这种显示的想法派生 TS 中的信息在金融分析中具有移动平均线和相对强弱指数 (RSI) 等属性。我们应用它来编码和存储此类信息。通过这种方式,我们可以将其显式地保存在潜在空间中,通知后续数据库检索。

这为我们提供了一种存储可以从 TS 计算的其他此类属性 (x) 的通用方法。

5.2.1 培训。使用最小-最大缩放器将所有输入历史 TS x 标准化为 [0, 1] 之间。此外, 给定一个 TS x, 我们根据新的 TS $\epsilon \in [0, 1]$ 定义其价格波动性, 其中每个数据点 ϵ 是相邻 2 个点的波动性 (我们使用 $\epsilon = 4$) : $\epsilon = (x - x_{i-1}) / x_{i-1}$ 。我们将价格趋势 (原始) TS 简称为 “TS”, 并将由此计算出的价格波动率 TS 称为波动率-TS。

为了基于趋势和波动性进行搜索, 我们为 TS 和波动性 TS 训练两个相同 (且独立) 的全连接自动编码器。为了合并趋势信息, 我们训练了一个经典的全连接自动编码器 (AE) (x), 它将 TS x 编码到潜在空间中, 具有以下重建损失:

$$\min_{\epsilon \in D} \|x - x_{\epsilon}\|_2^2$$

其中 x_{ϵ} 是重建的 TS。为了将波动性信息纳入搜索中, 我们训练另一个深度 AE (ϵ) 来编码到潜在空间中, 具有类似的重建损失:

$$\min_{\epsilon \in D} \|\epsilon - \epsilon_{\epsilon}\|_2^2$$

其中 ϵ_{ϵ} 是重构的波动率 TS。

基于草图的检索的整体编码器简单地连接和标准化来自 (x) 和 (ϵ) 的嵌入。

对于 AE 的网络结构, 输入尺寸为 30, 编码器中有 3 个隐藏层, 尺寸分别为 512、256、16; 最后一层是大小为 16 的潜在空间。我们在 AE 中解码之前使用与编码器相同的结构 (尽管相反) 对潜在空间进行归一化。所有层都包含可训练的偏置权重, 并使用 ReLU (整流线性单元) 作为非线性函数, 除了解码器的最后一层不包含 ReLU 计算。

5.2.2 数据库。在数据库构建中, 我们计算每个 TS $x \in D$ 的 AE 输出 (潜在空间)。特别是, 对于每个 TS, 我们可以计算其嵌入, 即 (x) 和 (ϵ) 的串联, 并将其存储到数据库中结构。

数据库维护嵌入和原始 TS 以供后续高效搜索。特别是, 我们使用 x_{ϵ} 到用 Facebook AI 相似性搜索 (FAISS) [20], 这是一个在向量空间上进行快速搜索的库, 用于索引和查找。

5.2.3 检索。为了使用输入 TS 中的趋势和波动性信息检索 TS, 我们使用 AE 生成查询向量来搜索 FAISS 索引。给定包含趋势 x 的用户草图, 我们计算波动性 TS, 然后使用编码器计算查询的嵌入 ϵ 。然后, 我们可以使用这种嵌入来有效地检索 TS $x \in D$ 的子集, 其余弦相似度在数据库中最高。

在实验部分, 我们展示了这些编码器如何保存趋势和波动性信息。我们还展示了这种方法如何比使用相同模式 (TS 草图) 进行可比 TS 检索的现有基线方法快得多。

最重要的是, 虽然现有的消费者 TS 数据库要求用户为匹配属性 (例如趋势近似 (例如, 二次多项式)) 指定严格的定义, 但我们发现我们的 AE 自然地近似并匹配趋势中的任何趋势。

输入草图 TS。这不仅提供了更快的检索模式, 还为最终用户提供了更大的灵活性。

6 实证分析

6.1 评价措施

6.1.1 基于文本的检索措施。我们使用三个指标来

终端性能, 对样本内 (训练期间使用) 和样本外查询的未见过的 TS 图像的验证集进行评估。每个查询都是关于特定功能的单个语句, 每个机制的查询数量大致相等。我们使用 Rank@9 评估检索准确性。这是针对样本内和样本外设置进行记录的。此外, 我们定义了一个多样性指标, 它测量所有查询中返回的不同 TS 图像的数量相对于返回图像总数的比例。因此, 更高的多样性度量意味着数据库中的更大比例的图像能够通过即时变化来检索。对于此指标, 我们记录两个查询设置的聚合值。

6.1.2 基于草图的检索措施。评估草图-

基于检索方法, 我们使用测试集中的 302 个 TS 查询 (在 AE 训练期间未给出) 自动生成 TS 查询 (作为草图 TS 的替代品)。为此, 我们还添加了噪声和向右循环移动 5 步, 以生成与训练数据不太相似的 TS, 并查询最佳匹配。为了评估每种方法的检索性能, 我们考虑将查询草图与返回的 TS 相匹配的度量。我们计算的用于比较基线和 AE 之间性能的度量如下: (1) MAPE: 是趋势和检索到的 TS 之间的平均绝对百分比误差 (MAPE)。这计算逐点误差与真实值之间的平均比率。请注意, 我们将其显示为比率, 而不是百分比。对于每种方法, 我们对所有测试查询 (每个查询返回的所有前 k 个结果) 计算该值的平均值。(2) CORR: 是在 [-1, 1] 之间的皮尔逊相关系数, 在查询 TS 和检索到的 TS 之间测量。我们对所有查询的值进行平均。

(3) 计算时间 (平均值和标准差): 这是对所有查询进行计算的。如果该方法需要嵌入, 则时间包括计算每个查询的嵌入所花费的时间。

6.2 基线

6.2.1 基于文本的检索基线。鉴于数量有限

在此领域之前的工作中, 尚未建立基准。因此, 这项工作的一個关键贡献是为未来研究中的比较提供适当的基准。鉴于这里缺乏此类方法, 我们实现了两个简单的基线, 它们利用传统的 NLP 方法来计算文本的嵌入。

(1) 神经网络分类器 (Classifier): 这里我们将检索问题重新表述为分类问题。我们首先通过使用传统的 NLP 编码算法 - Word2Vec [26] 计算词表中所有单词的 100 维嵌入, 该算法是在来自维基百科的 1 亿字节纯文本的 Text8 数据集上进行预训练的。然后将地面实况图像标题转换为数字向量表示

方法	排名@9 样本内	排名@9 样本外	多样性
分类器	0.59	0.43	0.40
Word2Vec-UMAP	0.37	0.35	0.05
我们的 (没有过滤)	0.65	0.51	0.18
我们的 (已过滤)	0.92	0.71	0.50
我们的 (过滤+)	0.96	0.89	0.30

表 1:以下方法在合成数据集上的检索结果:神经分类器,输出检索概率

给定文本嵌入的数据库中的所有 TS;对比学习与传统的表示学习方法 (UMAP 和 Word2Vec);具有三种过滤变体的深度编码网络。Filtered+ 表示与字幕的 GPT 增强一起进行的后处理过滤步骤。

对各个词嵌入进行平均,其中余弦计算每个基于文本的查询嵌入的相似度训练集。给定输入查询向量,网络输出数据库中每个样本的预测余弦相似度。我们针对训练期间的 MSE 损失优化网络。然后可以将输出相似度按降序排序并进行检索。由于网络架构是固定的使得每个输出节点对应于中的特定样本数据库仅适用样本内检索;因此我们携带在验证集上进行训练和检索。
(2) Word2Vec-UMAP:作为替代基线,我们替换我们的采用传统表示学习方法的 SOTA 深度学习编码器。相应的单模态编码器被替换具有以下内容:文本编码器→ Word2Vec 和图像编码器→ 乌玛普。
与我们基于编码器的模型一样,我们使用对齐嵌入每个的对比损失和单层 MLP 投影头情态。最终对齐的空间维度为 64,其中文本和图像嵌入分别为 100 和 2048 维分别。

6.2.2 基于草图的检索基线。我们根据以下 3 种基线方法评估基于草图的检索:

(1) 暴力搜索 (BF):该方法采用查询 TS,并且将其与数据集中的所有条目进行比较。它选择top-k 查询与每个之间的欧几里得距离的最佳匹配数据库的TS;这是误差的 L2 范数。我们忽略了该基线的波动性信息。
(2) 对 TS 和 Volatility-TS (BF_avg) 进行暴力搜索:方法中,我们重复暴力搜索,除了我们计算原始 TS 和后续 TS 的误差的欧氏距离波动率-TS,并平均距离。这个平均距离是用于从数据库中查找前 k 个最接近的匹配项。
(3) UMAP:我们计算 TS 和 Volatility-TS 的 UMAP 嵌入数据集并将其存储在 FAISS 索引中,以便通过连接和归一化两个嵌入的向量进行查找。这嵌入空间的维数与 AE (16) 的维数相匹配。建立索引后,每个查询都转换为 umap 嵌入它的 TS 和波动性-TS,用于在 FAISS 中搜索指数。在这项工作中,所有 TS 的长度相同,因此我们不使用动态时间扭曲 (DTW),因为它可以简化为欧几里德相同长度 TS 的距离,这就是 BF 方法所使用的。

方法	数据集 尺寸	排名@9 样本内	排名@9 样本外
未过滤	1500	0.56	0.59
未过滤-再训练	1500	0.69	0.60
已过滤	1500	0.59	0.51
过滤保留	1500	0.48	0.48
未过滤-再训练	3000	0.75	0.62
过滤保留	3000	0.55	0.51

表 2:我们的 Deep 对历史数据集的检索结果
编码网络。注意这里的过滤/未过滤是指GPT-趋势、波动性和冲击概率的增强过滤。

6.3 结果

我们展示两种查询方式的结果 (自然语言,和草图 TS)并将它们与基线结果进行比较。

6.3.1 基于文本的检索结果。

综合数据。合成数据集的检索结果为如表 1 所示。我们评估了模型的三种变体基于文本的检索基线 (分类器、Word2Vec-UMAP) ;全部我们的深度编码网络 TS 检索方法的三个变体都优于基线。此外,两个基线模型

比我们提出的方法灵活得多。对于分类器网络来说,缺乏灵活性是由于需要固定网络架构,每个输出节点对应检索数据库中的单个条目。因此,任何新条目数据库只能通过完整的重新培训来合并与增强的数据。另一方面,Word2Vec-UMAP 基线不受任何结构性约束的限制,而是由于 UMAP 方法在极高维空间中进行表示学习的计算费用。这样,我们就可以

仅在 4000 张图像上训练嵌入,每个扁平图像由 50,176 维向量表示。我们多种多样每个最近的邻居,最小距离和数量 UMAP 模型的组件参数,但都几乎没有对最终性能的影响。

历史数据。表 2 显示了历史数据集的结果。这些都是针对综合数据研究中的预训练模型以及在新数据集上重新训练时进行评估的。每个库存与由流动性决定的波动机制相关[6]。为了进行比较,构建了替代数据集

其中波动率的基本事实是由第 4 节中讨论的过滤方法决定的,而不是由流动性决定的。我们发现
在评估之前对数据集进行波动性过滤不是必需的,而且常常会妨碍准确性,即使在评估时也是如此
在合成数据集上训练的原始模型上。虽然性能与预训练模型相比具有竞争力,重新训练发现新数据可以产生更好的结果,特别是当数据集大小增加。鉴于性能改进当增加数据集大小时,我们没有理由认为模型在历史股票数据上的表现无法与该模型相媲美在数据集缩放时的综合情况中可以看到。

噪声、位移、k 测量		BF	BF_平均值	乌玛普	AE
0.05, 0, 1	TS-(MAPE,CORR)	(8.41E-02, 9.83E-01)	(8.41E-02, 9.83E-01)	(2.28E-01, 8.43E-01)	(9.98E-02,9.71E-01)
	体积-(MAPE,CORR)	(2.20E-01,8.02E-01)	(2.20E-01,8.02E-01)	(2.17E-01,7.34E-01)	(2.09E-01,8.04E-01)
	时间- (平均值,标准差)	(4.27E-02,1.48E-02)	(5.08E-02,1.61E-02)	(7.71E+00, 6.60E-01)	(1.90E-02, 8.16E-03)
0.05, 5, 1	TS-(MAPE,CORR)	(2.11E-01, 8.96E-01)	(2.17E-01, 8.95E-01)	(3.61E-01,7.49E-01)	(2.76E-01,8.24E-01)
	体积-(MAPE,CORR)	(3.77E-01,5.73E-01)	(3.37E-01, 5.97E-01)	(3.30E-01, 5.30E-01)	(3.20E-01,5.44E-01)
	时间- (平均值,标准差)	(1.97E-01,1.33E-02)	(2.32E-01,1.35E-02)	(7.82E+00, 8.29E-01)	(1.87E-02, 7.66E-03)
0.05, 0, 3	TS-(MAPE,CORR)	(1.94E-01, 8.92E-01)	(1.99E-01, 8.86E-01)	(3.07E-01, 7.46E-01)	(2.33E-01,8.33E-01)
	体积-(MAPE,CORR)	(3.57E-01,5.37E-01)	(3.00E-01, 6.00E-01)	(2.64E-01, 6.24E-01)	(2.66E-01, 6.25E-01)
	时间- (平均值,标准差)	(1.97E-01,1.38E-02)	(2.34E-01,1.55E-02)	(7.64E+00,1.13E+00)	(1.81E-02,8.52E-03)
0.05,5,3	TS-(MAPE,CORR)	(2.60E-01, 8.53E-01)	(2.66E-01, 8.50E-01)	(3.79E-01,7.18E-01)	(3.04E-01,7.93E-01)
	体积-(MAPE,CORR)	(4.21E-01,4.64E-01)	(3.78E-01,5.00E-01)	(3.36E-01, 5.37E-01)	(3.41E-01, 5.16E-01)
	时间- (平均值,标准差)	(1.87E-01,8.90E-03)	(2.22E-01,9.51E-03)	(7.66E+00, 1.19E+00)	(1.86E-02, 8.80E-03)

表 3:使用高斯噪声(N (= 0, = 0.05))和循环移位的测试集查询的一组 TS 匹配度量的平均值
基线方法 (BF,BF_avg,UMAP)和AE方法。

6.3.2 基于草图的检索结果。关于我们基于草图的检索方法,我们展示了基线的结果

方法 (BF,BF_avg,UMAP)和自动编码器 (AE)方法

表 3。对于 MAPE 测量,越低越好,而对于 CORR

(相关性)越高越好。第一列描述测试

由添加的噪声、循环移位步骤 (移位)定义的场景

这是 TS 旋转的步数,并且

返回的最佳匹配数 (k)。

从数据中,我们可以看到,匹配 TS 趋势数据 (“TS-(MAPE,CORR)”度量)的最佳拟合来自于

BF 方法,这并不奇怪,因为它煞费苦心地计算查询 TS 与中每个 TS 之间的欧几里得距离

数据库之前采取距离最近 (最少错误)的一个。然而,TS 之间的距离并不能转化为匹配

波动性信息,这可以在结果中看到;其他考虑搜索过程中波动性的方法在匹配方面做得更好

波动率-TS 度量 (“Vol-(MAPE,CORR)”)。我们的 AE 通常在匹配波动性方面比基线方法更好

TS (用于 MAPE 和 CORR 测量),当它不是最佳时,它非常接近 UMAP 的最佳值。重要的是,AE

方法比所有其他方法花费的时间要少得多,这

UMAP 过程花费最多时间,尽管 UMAP 和 AE 都

使用具有相同大小潜在空间的 FAISS。这个时间差

UMAP和AE之间的区别是由于UMAP编码时间造成的,时间

将新查询投影到与数据库相同的潜在空间中。

7 结论

总之,我们的论文提出了一个多模式框架

存储和检索金融 TS 数据,以便保留与金融相关的信息。特别是,我们开发了

接受自然语言文本的用户友好查询界面

或 TS 草图以供检索。我们的方法使用深度编码器网络将多模态数据映射到较低维的潜在数据中。

空间,并保留基本的 TS 属性。实验结果

支持该方法的计算效率和检索精度。因此,我们证明了潜在空间投影的实用性

用于检索金融时间序列数据,同时支持更多直观的查询方式。

免责声明

本文由摩根大通人工智能研究小组编写,仅供参考。

其附属公司 (“摩根大通”),并且不是该研究的产品

摩根大通部门。摩根大通不做任何陈述,并且

对本文所含信息的完整性、准确性或可靠性做出任何保证,并且不承担任何责任。

本文件无意作为投资或研究或投资

购买建议、推荐、要约或招揽

或服务任何证券、金融工具、金融产品或

服务,或以任何方式用于评估参与任何交易的优点,并且不得构成招揽

在任何管辖范围内或向任何人,如果此类招揽

此类管辖权或此类人将是非法的。

参考

[1] Rakesh Agrawal,Giuseppe Psaila,Edward L. Wimmers 和 Mohamed Zait. 1995年。查询历史的形状。第21届国际会议论文集
超大型数据库 (VLDB 95)。 502-514。
[2]穆罕默德·M·阿尔卡迪,乔尔·T·约翰逊,安德鲁·J·奥布莱恩,安娜·巴伦扎诺,和弗朗西斯科·马蒂亚。 2019.利用CYGNSS时间序列反演土壤湿度。
IEEE 地球科学与遥感学报 57, 7 (2019), 4322-4331。
[3] Ira Assent,Ralph Krieger,Farzad Afschari 和 Thomas Seidl. 2008. TS 树: 高效的时间序列搜索和检索。第 11 届国际会议论文集
扩展数据库技术会议:数据库技术的进步。
[4]法兹尔·巴雷斯 (Fazl Barez),保罗·比洛孔 (Paul Bilokon),熊瑞杰。 2023. 基准测试专业化
高频数据数据库。 arXiv 预印本 arXiv:2301.12561 (2023)。
[5] Ledion Bitincka,Archana Ganapathi,Stephen Sorkin 和 Steve 张。 2010年。使用半结构化时间序列数据库优化数据分析。车间内
通过日志分析和机器学习技术管理系统。
[6] 让·菲利普·布肖,朱利叶斯·博纳特,乔纳森·多尼尔和马丁·古尔德。 2018. 交易、报价和价格:显微镜下的金融市场。剑桥
大学出版社。
[7] Samuel R. Bowman,Gabor Angeli,Christopher Potts 和 Christopher D. Manning 。 2015. 用于学习自然语言推理的大型注释语料库。
2015 年自然语言经验方法会议论文集
加工。计算语言学协会,葡萄牙里斯本,632-642。
https://doi.org/10.18653/v1/D15-1075
[8] Jane Bromley,Isabelle Guyon,Yann LeCun,Eduard Säckinger 和 Roopak Shah. 1993. 使用 “连体”时延神经网络进行签名验证。在
NeurIPS (NIPS 93)。 737-744。
[9]大卫·伯德。 2019.解释基于代理的金融市场模拟。
arXiv:1909.11650 [cs.MA]
[10]大卫·伯德,玛丽亚·海比内特和塔克·海比内特·巴尔奇。 2020. ABIDES:迈向高保真多主体市场模拟。在 2020 年会议记录中
ACM SIGSIM 高级离散仿真原理会议。 11-22。

[11] Andrea Coletta,Sriram Gopalakrishnan,Daniel Borrajo 和 Svitlana Vyetrenko。2023.关于约束时间序列生成问题。arXiv 预印本arXiv:2307.01717 (2023)。

[12]安德里亚·科莱塔、艾默里克·穆兰、斯维特拉娜·维特连科和塔克·巴尔奇。2022 年。作为世界代理,学习从数据模拟现实的限价订单簿市场。第三届 ACM 国际金融人工智能会议论文集。428–436。

[13] Jacob Devlin,Ming-Wei Chang,Kenton Lee 和 Kristina Toutanova。2019.BERT :用于语言理解的深度双向变压器的预训练。arXiv:1810.04805 [cs.CL]

[14] 伊恩·古德费洛、约舒亚·本吉奥和亚伦·库维尔。2016.深度学习。

[15]郝远哲,秦雄派,陈跃国,李亚茹,孙晓光,涛宇,张晓,杜晓勇。2021.Ts-benchmark :时间序列数据库的基准。2021 年 IEEE ICDE。IEEE,588–599。

[16]何凯明,张翔宇,任少清,孙健。2015.用于图像识别的深度残差学习。arXiv:1512.03385 [cs.CV]

[17] Shima Imani,Sara Alaee 和 Eamonn J. Keogh。2019.将人类置于时间序列分析循环中。2019 年万维网会议配套论文集 (2019)。https://api.semanticscholar.org/CorpusID:153314304 [18] Timescale Inc. 2022.简化的时间序列数据 时间尺度。https://www. timescale.com/

[19]涌入数据。2022. influxdb :开源时间序列数据库。https://www. influxdata.com/products/influxdb-overview/

[20]杰夫·约翰逊、马蒂斯·杜兹和埃尔韦·杰古。2019.使用 GPU 进行十亿级相似性搜索。IEEE 大数据汇刊 7, 3 (2019), 535–547。

[21] Eamonn Keogh,Stefano Lonardi 和 Bill Yuan-chi Chiu。2002 年.在线性时间和空间的时间序列数据库中发现令人惊讶的模式。第八届ACM SIGKDD 知识发现和数据挖掘国际会议论文集。550–556。

[22] KX. 2022. 使用 kdb+ 和 q 语言进行开发 - Kdb+ 和 q 文档。https://code.kx.com/q/ [23]刘佳,薛勇,任凯军,宋俊强, Christopher Windmill 和 Patrick Merritt。2019. GPU 集群上卫星图像的高性能时间序列定量检索。IEEE 应用地球观测和遥感专题杂志 12, 8 (2019), 2810–2821。

[24]米罗·曼尼诺和阿扎·阿布齐德。2018.使用手绘无比例草图进行富有表现力的时间序列查询。2018 年 CHI计算系统人为因素会议论文集。1–13。

[25]利兰·麦金尼斯、约翰·希利和詹姆斯·梅尔维尔。2018.Umap :统一流形近似和降维投影。arXiv 预印本arXiv:1802.03426 (2018)。

[26]托马斯·米科洛夫、陈凯、格雷戈里·S·科拉多和杰弗里·迪恩。2013.向量空间中词表示的有效估计。在国际学习代表会议上。

[27]伯恩特·奥克森达尔。1998. 随机微分方程,应用简介。施普林格。

[28]亚历克·雷德福、金钟郁、克里斯·哈拉西、阿迪亚·拉梅什、加布里埃尔·吴、桑迪尼·阿加瓦尔、吉里什·萨斯特里、阿曼达·阿斯科尔、帕梅拉·米什金、杰克·克拉克、

格雷琴·克鲁格和伊利亚·苏茨克弗。2021.从自然语言监督学习中可迁移的视觉模型。第 38 届国际机器学习会议论文集,卷。139.PMLR,8748–8763。

[29]尼尔斯·赖默斯和伊琳娜·古列维奇。2019.Sentence-BERT :使用 Siamese BERT 网络的句子嵌入。arXiv:1908.10084 [cs.CL]

[30]奥尔加·鲁萨科夫斯基、邓佳、苏浩、乔纳森·克劳森、桑吉夫·萨西什、马肖恩、黄志恒、安德烈·卡帕西、阿迪亚·科斯拉、迈克尔·伯恩斯坦、亚历山大·C. 伯格和李飞飞。2015. ImageNet 大规模视觉识别挑战赛。arXiv:1409.0575 [cs.CV]

[31]弗洛里安·施罗夫、德米特里·卡莱尼琴科和詹姆斯·菲尔宾。2015. FaceNet :用于人脸识别和聚类的统一嵌入。2015 IEEE CVPR (2015) , 815–823。

[32] Bonil Shah,PM Jat 和 Kalyan Sashidhar。2022.时间的表现研究 系列数据库。arXiv 预印本 arXiv:2208.13982 (2022)。

[33]宋东进,夏宁,程伟,陈海峰,陶大成。2018. 用于多变量时间序列检索的秩监督联合二元嵌入的深 r 根。第 24 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集。2229–2238。

[34] Ivona Tautkute,Tomasz Trzcinski,Aleksander P Skorupa、ukasz Brocki 和 Krzysztof Marasek。2019.Deepstyle :时尚和室内设计的多模式搜索引擎。IEEE Access 7 (2019), 84613–84628。

[35] Svitlana Vyetrenko,David Byrd,Nick Petosa,Mahmoud Mahfouz,Danial Dervovic、Manuela Veloso 和 Tucker Balch。2020 年.变得真实 :稳健的限价订单簿市场模拟的现实指标。首届 ACM国际金融人工智能会议论文集。1–8。

[36]伊莱恩·华 (Elaine Wah)、梅森·赖特 (Mason Wright) 和迈克尔·P·威尔曼 (Michael P. Wellman)。2017.福利影响连续双重拍卖中的做市。(2017)。

[37]王伟,杨晓燕,黄明清,张东翔,庄月婷。2016.基于深度学习的有效多模态检索。VLDB 期刊 25 (2016), 79–101。

[38]马丁·瓦滕伯格。2001. 绘制一个图表来查询时间序列数据库。在CHI 01 计算系统中人为因素的扩展摘要中。381–382。

[39]阿迪娜·威廉姆斯、尼基塔·南吉亚和塞缪尔·鲍曼。2018.通过推理进行句子理解的广泛覆盖挑战语料库。计算语言学会北美分会 2018 年会议记录 :人类语言技术,第 1 卷 (长论文)。1112–1122。https://doi.org/10.18653/v1/N18-1101 [40]菲利普·温斯顿。2022 年.时间序列数据库和 Amazon Timestream。IEEE软件 39, 03 (2022), 126–128。

[41]杨阳,曹强,姜洪。2019. EdgeDB :用于边缘计算的高效时序数据库。IEEE Access 7 (2019), 142295–142307。

[42] Jinsung Yoon,Daniel Jarrett 和 Mihaela Van der Schaar。2019.时间序列生成对抗网络。神经信息处理系统的进展32 (2019)。

[43]朱迪贤,宋东进,陈云从,克里斯蒂安·卢梅扎努,程伟,宗波,倪景超,沟口武彦,杨天宝,陈海峰。2020. 用于多元时间序列检索的深度无监督二进制编码网络。AAAI 人工智能会议论文集,卷。34. 1403–1411。