

基于神经网络的动态图表征学习：综述

摘要

近年来，动态图（DG）表示由于能够将拓扑结构和时间信息有效地集成在紧密的嵌入表示中，因此越来越多地被用于动态系统的建模任务中。此外，动态图可以有效地处理社交网络预测、推荐系统、流量预测或脑电图分析等应用程序，这些应用程序无法使用一般的嵌入表示来解决。

而作为动态图表示出现的直接结果，“动态图学习”已成为一个新的机器学习问题，它结合了“顺序/时态数据处理”和“静态图学习”所面临的挑战。而在这一研究领域，**动态图神经网络（DGNN）**已成为最先进的方法，近年来也已提出了大量相关的模型，**而本文旨在回顾与动态图学习相关的问题和模型。**

简而言之，本文分析并讨论了各种动态图监督学习设置，并比较了现有模型在时间信息建模方式上的异同。最后，本文还为 DGNN 设计者提供了在面对动态图学习问题时的一般指导原则。

相关介绍

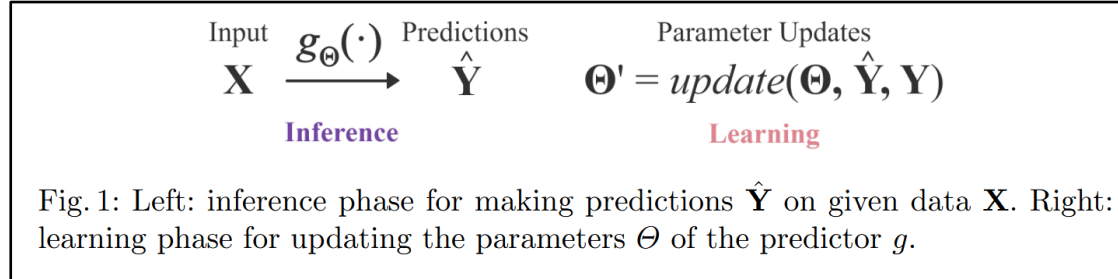
在模式识别（Pattern Recognition）任务中，大多数现有研究都集中在静态图（static graphs）的问题上，其节点集（node set）、边集（edge set）以及节点/边的属性不会随着时间的推移而改变。然而，真实世界的数据往往是动态分布的，需要我们构建动态系统来处理解决。因此，将动态图学习应用于此类问题之上顺理成章也成为了解决该问题更好地选择。不过，它也更加具有挑战性，因为除了静态图学习问题中所面临的挑战，我们还需要考虑时序数据处理方面所带来的挑战。

静态图学习，例如基于节点的消息传递机制所形成的第一代图神经网络（MPNN），虽然取得了一系列成功的结果，但其表达能力仍然不够强大，无法很好地适用于更加现实的问题。

动态图神经网络 DGNN 的创始模型分别来自 2018 年[Structured sequence modeling with graph convolutional recurrent networks.]和 2019 年[Dyrep: Learning representations over dynamic graphs.]，二者分别适用于离散和连续的情况。

本综述的主要目的是扩展现有研究，重点关注使用神经网络的动态图监督学习，主要面向具有神经网络和静态图学习基础知识的受众。

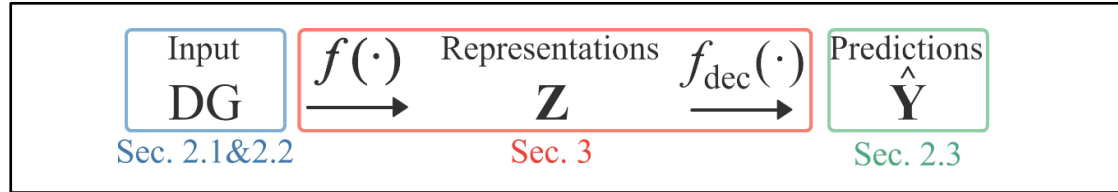
监督学习的一般过程



- 1) 对给定数据 X 进行预测 \hat{Y} 的推理阶段。（前向传播）
- 2) 利用 \hat{Y} 和 Y 更新模型 g 的参数 θ 的学习阶段。（反向传播）

动态图表征学习的一般过程

鉴于 DG 的动态程度高于序列数据，因此编码器/解码器框架为学习 DG 嵌入表示提供了一个非常合适的框架。



应用于动态图的编码器/解码器模型：

- 1) 编码器：计算 $Z = f(DG)$ ，其中 DG 是动态图（包括拓扑和属性）， $f(\cdot)$ 是参数化统计模型（通常是具有可学习参数的图神经网络）， Z 是 DG 的编码张量。
- 2) 解码器：计算 $\hat{Y} = f_{dec}(Z)$ ，即将 DG 的嵌入表示 Z 作为输入以获得预测 \hat{Y} 。

静态图模型

静态图 $G = (V, E)$ ，其中 V 代表 G 的节点集， E 代表 G 的边集。而 G 的连通信息通常由其邻接矩阵 $A \in \mathbb{R}^{|V| \times |V|}$ 表示。如果节点 u 和 v 之间存在边，即 $(u, v) \in E$ ，则 $A(u, v) = 1$ ，否则 $A(u, v) = 0$ 。

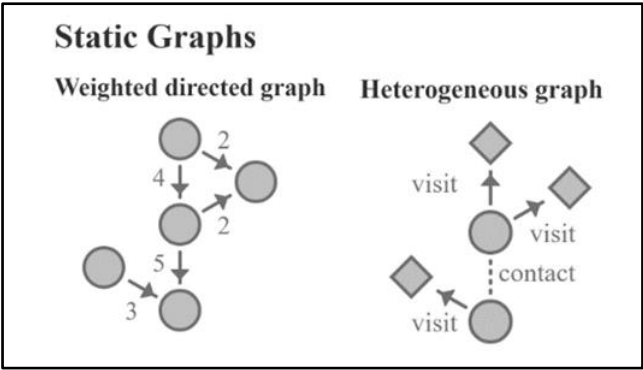
有向图与无向图——在无向图中，邻接矩阵 A 是对称的，而在有向图中，邻

接矩阵 A 不一定是对称的。

加权图——在加权图的情况下，其邻接矩阵 A 中的值是每条边的权重，而不是1，表示为：

$$A(u, v) = \begin{cases} w_{u,v}, & \text{if } (u, v) \in E \\ 0, & \text{otherwise} \end{cases}$$

异构图——异构图的节点有多种类型，其节点类型映射函数 $\phi: V \rightarrow O$ ；它的边也有多种类型，其边类型映射函数 $\psi: E \rightarrow R$ 。其中， O 和 R 表示预定义的节点类型和边类型的集合，且 $|O| + |R| > 2$ 。



动态图模型

动态图是拓扑和(或)属性随时间变化的图，即动态图中的结构和属性都可能随时间而变化——添加或删除边和(或)节点，其属性也可能因时间而更改。

而本文根据动态图的动态程度将 DG 划分为了四种可能的情况，如下表所示：

Table 1: Degree of Dynamism discussed in the article.

	Node Set		
	Constant Variant		
	Constant	$fix_{V,E}$	N/A
Edge Set	Variant	fix_V	$vary$

- (1) 节点集 V 和边集 E 都不变，表示为 $fix_{V,E}$ 。该情况对应于文献中称为“时空图”（STG）的 DG。
- (2) 节点集 V 是不变的，但边集 E 会发生变化，表示为 fix_V 。
- (3) 节点集 V 和边集 E 都在变化，表示为 $vary$ 。
- (4) 边集 E 不变，但节点集 V 会发生变化。但由于 E 是基于节点元组 V 存在的，因此这种情况毫无意义。

连续时间动态图（Continuous Time Dynamic Graphs, CTDG）

使用一组事件来表示的动态图。以下是三种典型的基于事件的表示方法：

（1）Contact-Sequence:

$$Contact - Sequence = \{(u_i, v_i, t_i)\}$$

主要用来表示 t 时刻两个节点 (u, v) 之间的瞬时相互作用。

（2）Event-Based:

$$Event - Based = \{(u_i, v_i, t_i, \Delta_i)\}$$

将两个节点 (u, v) 之间的边表示为起始时间为 t_i ，持续时间为 Δ_i 的一个事件。

（3）Graph Stream:

$$Graph Stream = \{(u_i, v_i, t_i, \delta_i)\}$$

主要用于大规模图，在 t_i 时刻，两个节点 (u, v) 之间的边的添加($\delta_i = 1$)或删除($\delta_i = -1$)。

离散时间动态图（Discrete Time Dynamic Graphs, DTDG）

可以看作由 T 个静态图组成的序列，如下式所示：

$$DTDG = (G^1, G^2, \dots, G^T)$$

它们是动态图在不同时刻或时间窗口的快照，即 DTDGs 可以通过在时间轴上周期性地拍摄 CTDGs 的快照来获得。

等效静态图（Equivalent Static Graphs, ESG）

通过构建单个静态图来用于表示上述的动态图 DG。近年来，人们主要提出了几种构建 ESG 的方法。而本文将它们分为两类：面向边（Edge-oriented）和面向节点（Node-oriented）的 ESG。

（1）Edge-oriented:

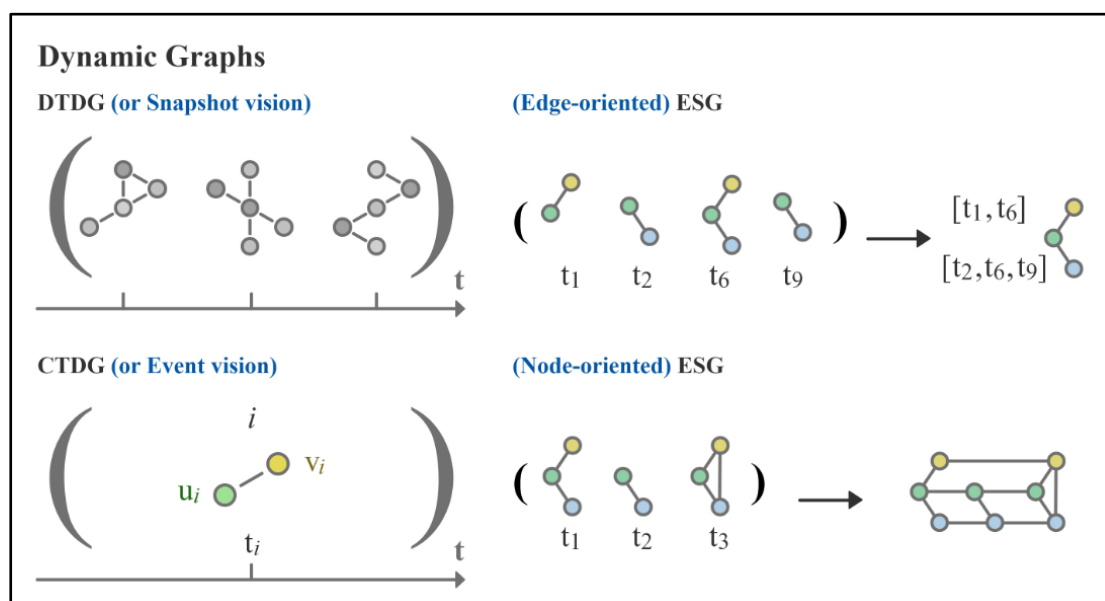
面向边的 ESG 将图序列聚合为静态图，并将时间信息编码为属性序列。这样的表示也被称为时间图表示。

（2）Node-oriented:

面向节点的 ESG 在节点出现的每一时刻建立节点的副本，并定义节点在时间戳/节点出现之间是如何连接的。

【ESG 表示的一个重要作用是它们能使静态图算法应用于动态图的学习。】

连续时间动态图、离散时间动态图和等效静态图



动态图上的直推/归纳

直推式学习 (Transductive learning) 是指：模型在学习阶段除了能够看到训练数据（样本+标签）外，还能够接触到测试数据（样本），以期能够利用测试数据的内在结构，把测试数据作为一个整体来预测其标签，而不是孤立地、one-by-one 地预测每一个测试样本。

归纳式学习 (Inductive learning) 是指：从特定任务到一般任务的学习，模型在学习阶段仅基于训练集，而后将其应用于测试集的预测任务中，训练集与测试集之间是相斥的，即测试集中的任何信息都是没有在训练集中出现过的。因此，模型本身具备一定的通用性和泛化能力。【传统的 supervised learning 都可以理解为是 Inductive learning 的范畴】

而本文将动态图上的学习分为 5 种情况，分别是 3 种直推式学习和 2 种归纳式学习：

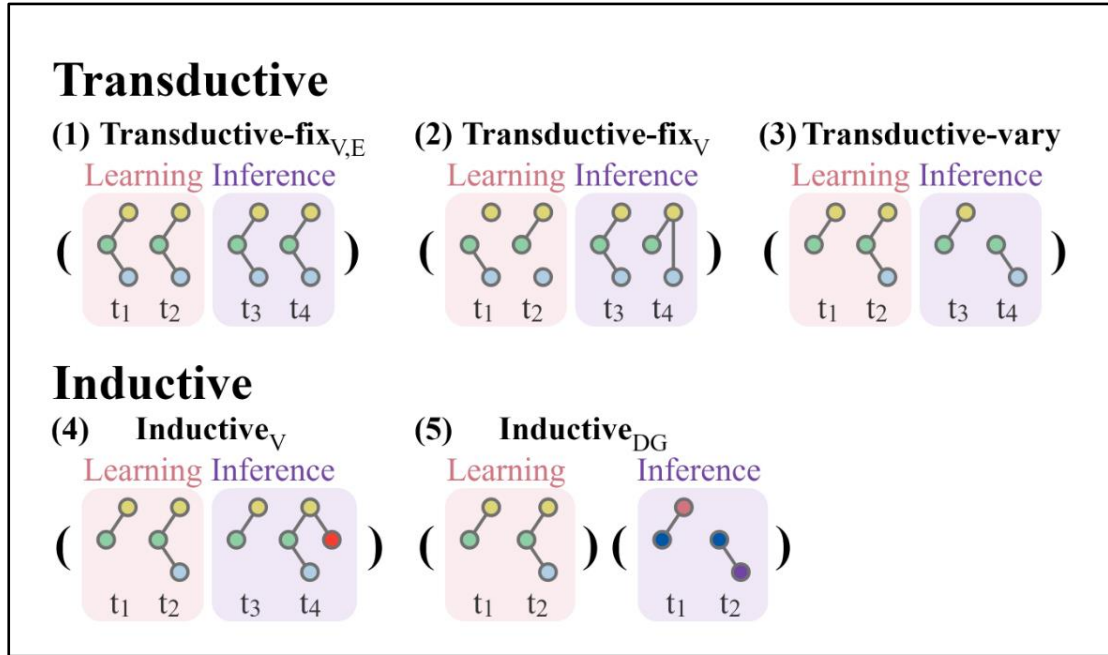
- 1) $Trans - fix_{V,E}$: 在这种情况下，DG 的拓扑结构是固定的，因此 V 和 E 在训练集和测试集上是相同的。
- 2) $Trans - fix_V$: 在这种情况下，训练集和测试集的患者集 V 是固定的，但边集 E 是随时间改变的，而这就需要考虑图上节点之间不断演化的连通性。
- 3) $Trans - vary$: 在这种情况下，训练集和测试集的 V 和 E 都可能随时间而改变，

但测试集中每个待预测节点的存在性已经在训练阶段确定了，即不新增其他未出现过的节点。

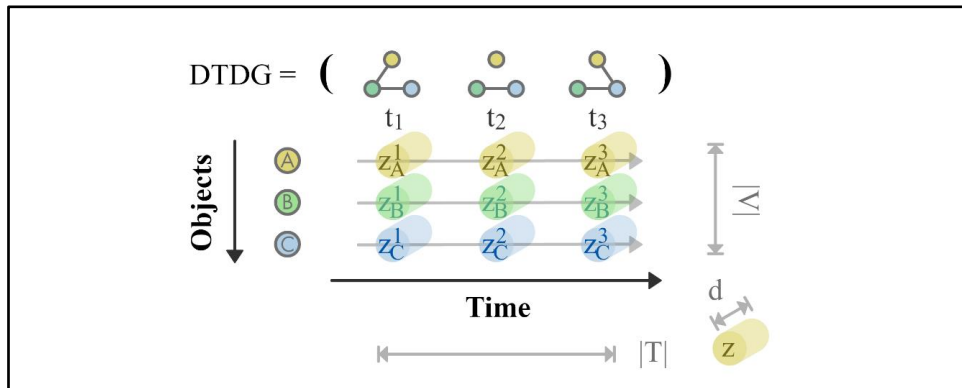
- 4) Ind_V : 这种情况指的是节点级的归纳任务，其中训练阶段和测试阶段都是在同一个 DG 上进行的，但也假设测试集中的节点并未在训练集中出现过。
- 5) Ind_{DG} : 这种情况指的是图级的归纳任务，即对不同的 DG 进行学习和推理，统计模型需要处理完全不可见的动态图。

上述 5 种情况如下图所示：

(离散时间下动态图学习中的直推式和归纳式情形)

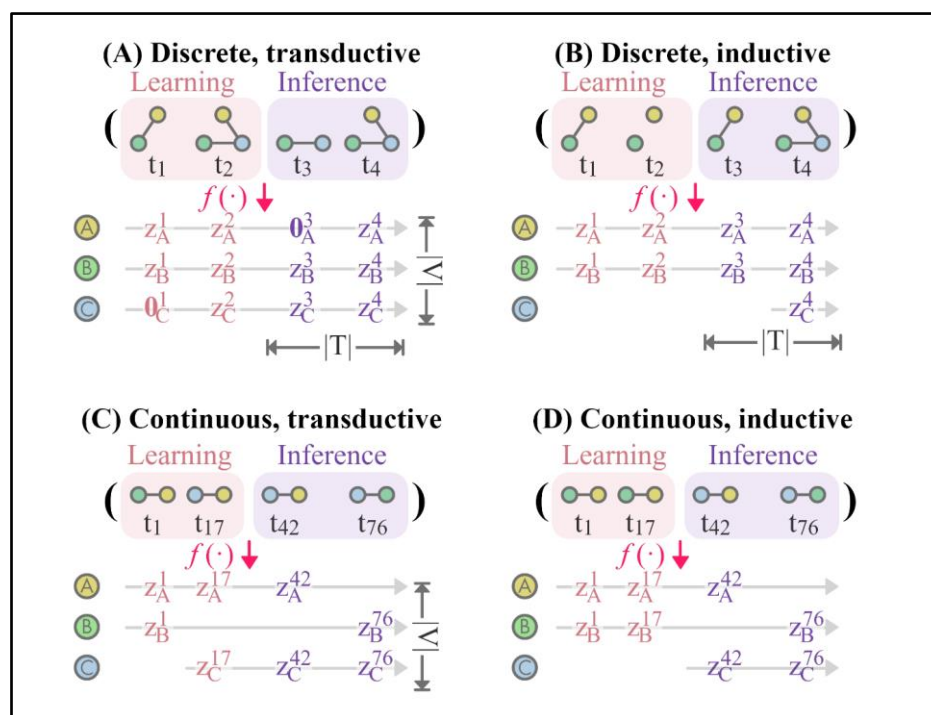


基于神经网络的动态图嵌入表示



如图所示，动态图 $DTDG$ 的嵌入表示 $Z \in \mathbb{R}^{|V| \times |T| \times d}$ ，其中 $|V|$ 代表节点的个数， $|T|$ 代表时间步的个数， d 代表单个节点 v 在单个时间步 t 内的嵌入向量 z_v^t 的维度。

不同动态图情况下可用的嵌入表示



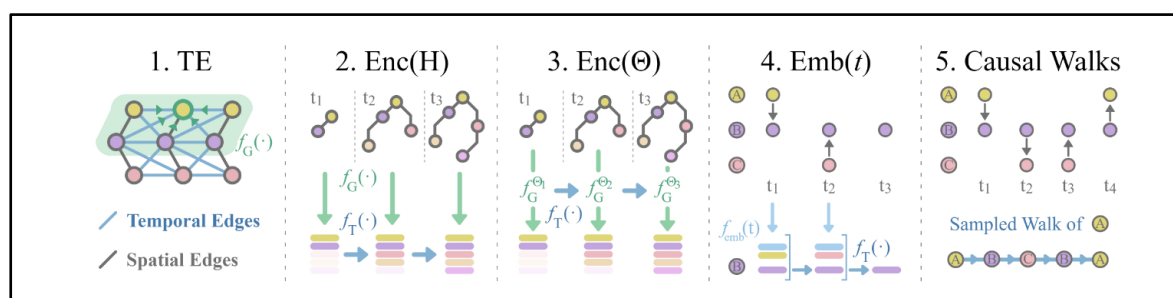
A)：使用0向量填充缺失值；

B)：使用列表来存储每个时间步的嵌入表示大小；

C) 和 D)：当发生事件时，再利用 $f(\cdot)$ 来更新关联节点的嵌入，或增加不可见节点的嵌入。

DGNN 模型的分类

最后，本文还给出了 DGNN 模型的一个分类，共五种，如下图所示：



1. Temporal Edge Modeling: 基于时序边进行建模，将 STG 转化为的等效静态图；

2. Sequentially Encoding Hidden States H: 使用时序编码器 $f_T(\cdot)$ 对每个快照的隐藏状态 H 进行跨时间编码。

3. **Sequentially Encoding Parameters θ** : 对图编码器 $f_G(\cdot)$ 的参数 θ 进行跨时间编码, 时间编码器为 $f_T(\cdot)$ 。
4. **Embedding Time t** : 在对节点进行编码时, 先将时间值转换为向量, 然后再与属性向量拼接或直接相加。
5. **Causal Walks**: 通过因果关系来限制在动态图上的随机游走。

上述各方法适应的动态图类型:

TE stands for "Temporal Edge Modeling", **Enc(H)** for "Sequentially Encoding Hidden States", **Enc(Θ)** for "Sequentially Encoding Model Parameters", **Emb(t)** for "Embedding time", **Causal RW** for "Causal Random Walks". ✓ means "applicable", * means "applicable with output restrictions", and no symbol indicates that it is not yet used in the literature.

Approach	DT Trans.	DT Ind.	CT Trans.&Ind.
① TE	✓		
② Enc(H)	✓	*	
③ Enc(Θ)	✓	✓	
④ Emb(t)	✓	✓	✓
⑤ Causal RW	✓	✓	✓

DGNN 的工作流程

对于动态图, DGNN 的设计不得不考虑更多的因素。因此, 本文将设计 DGNN 的工作流程概括如下:

1. 根据前文所述的动态图分类法, 明确任务的输入、输出和性质;
2. 根据学习任务的需求选择兼容的时间编码方法;
3. 设计对应的图神经网络结构;
4. 优化所设计的 DGNN 模型。