**DynGEM: Deep Embedding Method for Dynamic Graphs 2018**

**Abstract**

近年来，在低维空间中嵌入大型图因其在图形可视化、链接预测和节点分类等方面的广泛应用引起了人们的广泛关注。现有的方法侧重于计算静态图的嵌入。然而，在实际应用中，许多图都是动态的，并随着时间不断发展。将现有的嵌入算法简单地应用于动态图的每个快照通常会导致在稳定性、灵活性和效率方面的性能不佳。在这项工作中，我们提出了一个有效的算法DynGEM基于最近的进展深自动编码器的图形嵌入，以解决这个问题。DynGEM的主要优势包括: (1)随着时间的推移，嵌入是稳定的，(2)它可以处理不断增长的动态图，(3)它比在动态图的每个快照上使用静态嵌入方法有更好的运行时间

**Introduction**

网络分析中的许多重要任务都涉及到对图中的节点和/或边进行预测，这就需要有效的算法来提取有意义的模式和构造预测特征。在实现这一目标的许多尝试中，图嵌入，对图中每个节点的低维表示方法的研究引起了广泛关注，这种方法可以准确地捕捉到节点与其他节点之间的关系。研究表明，在节点分类、链路预测和图重建等监督学习任务中，图的嵌入优于替代方法

通常，我们将动态图表示为图在不同时间步长的快照集合。

本文提出了一种高效的动态图嵌入算法DynGEM，用于生成动态图的稳定嵌入。DynGEM在其核心使用了一个深度自动编码器，并利用深度学习的最新进展来生成高度非线性的嵌入。DynGEM不是从头开始学习每个快照的嵌入，而是从t - 1时刻的快照逐步构建t时刻的快照的嵌入，具体地，我们从之前的时间步长初始化嵌入，然后进行梯度训练。这种方法不仅保证了嵌入的稳定性，而且由于在第一个时间步之后的所有嵌入都需要很少的迭代来收敛，因此也带来了有效的培训。为了处理节点数量不断增加的动态图，我们使用启发式方法(PropSize)逐步增加神经网络的大小，以动态确定每个快照所需的隐藏单元的数量，除了提出的模型外，我们还介绍了动态图嵌入的严格稳定性度量。

**2. definition and preliminaries**

用G=(V,E)表示一个加权图，权重矩阵S，用sij表示两个节点之间的权重，si表示邻接矩阵中的第i行。

给定一个图G，图嵌入就是一个如下的映射：直观地说，如果图G中的u和v两个节点是“相似的”，那么它们的嵌入yu和yv在嵌入空间中应该是相互接近的，作者用

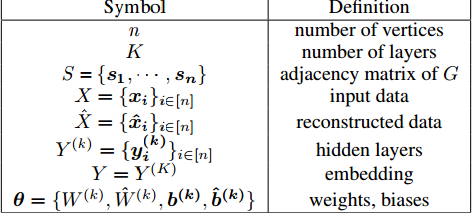
本文主要研究的是动态图的嵌入问题，我们将动态图G表示为一系列快照，。我们考虑具有生长图的环境，也就是说，新节点可以连接动态图并创建到现有节点的链接，我们将删除的节点视为图的一部分，它与其余节点的权值为零。我们假设Et和Et+1之间没有关系，快照之间可以形成新的边，而现有的边可以消失。

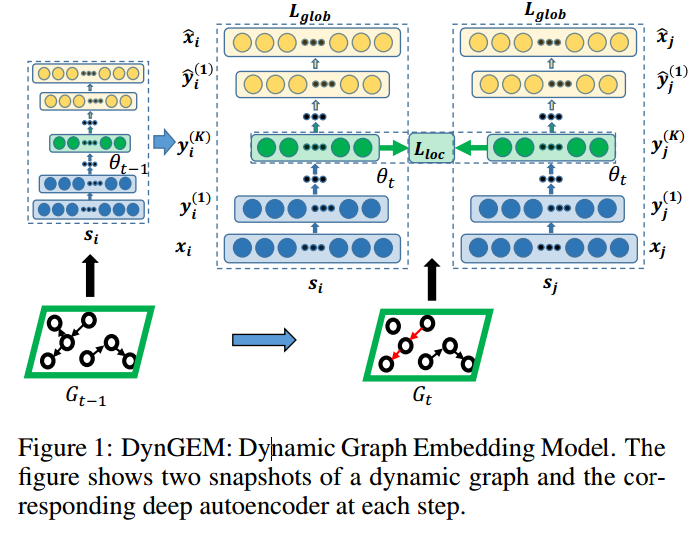
动态图嵌入将嵌入的概念扩展到动态图，给定一个动态图动态图嵌入是一系列时间序列的映射，映射ft是Gt的图嵌入，所有映射都保留各自图的邻近测度。

一个成功的动态图嵌入算法应该随着时间的推移创建稳定的嵌入。直观地说，稳定的动态嵌入是这样一种嵌入，即如果底层图稍有变化，则连续嵌入的差异也很小

**3. DynGEM: Dynamic Graph Embedding Model**

最近在深度无监督学习方面的进展表明，自动编码器可以成功地学习用于各种任务的非常复杂的数据低维表示。DynGEM使用深度自动编码器将输入数据映射到高度非线性的潜在空间，以捕获图快照中任意时间步长的连通性趋势。该模型是半监督的，最小化了一阶近似和二阶近似对应的两个目标函数的组合。图1展示了autoencoder模型，表1展示了用语。顶部有帽子的符号是用于解码器的。





**3.1 Handling growing graphs**

处理不断增长的动态图需要一种良好的机制来扩展自动编码器模型，同时保留以前训练步骤的权重。一个关键组件是决定隐藏层的数量和隐藏单元的数量应该如何随着更多的节点添加到图中而增长。我们提出一种启发式的方法，即PropSize，来计算所有层的新层大小，以确保连续层的大小在彼此一定的因数之内。

**PropSize**：我们提出了这种启发式算法来计算神经网络层在每个时间步长的新尺寸，并根据需要插入新的层。对于编码器，计算每对连续层的层宽

在确定了层数和各层的隐藏单元数后，作者采用Net2WiderNet和Net2DeeperNet [Chen et al.， 2015]提出了扩展深度自动编码器的方法。将自动编码器与PropSize、Net2WiderNet和Net2DeeperNet在每个时间步长上的扩展和深化相结合，使我们能够处理随时间增长的节点数不断增加的动态图，并通过实验得到了显著的性能。

**3.2 loss function and tranining**

为了学习模型参数，在每一步中最小化三个目标的加权组合:



都是超参数

**3.3 Stability by reusing previous step embedding**

对于动态图G，我们在G1上使用随机初始化参数cita1对它进行自编码器训练。对于所有后续时间步骤，我们初始化模型参数θt从之前的时间步长参数θt−1,扩大和加深之前模型。这导致结构的知识直接从ft - 1转移到ft，因此模型只需要了解Gt – 1和Gt之间的变化

**3.4 Techniques for scalabilit**

以前的静态图嵌入深度自编码模型[Wang et al.， 2016]使用sigmoid激活函数，并使用随机梯度下降(SGD)进行训练。

我们在所有的自动编码器层中使用ReLU来支持加权图，因为ReLU可以构造任意的si正分量。它也加速了训练，因为ReLU的导数很容易计算，而sigmoid的导数需要计算指数

我们还使用nesterov动量[Sutskever et al.， 2013]与适当调整的超参数，这比使用纯SGD收敛得更快。最后，我们观察到L1-范数和l2 -范数正则化组合在所有任务上的表现都更好

算法1给出了动态图单快照学习DynGEM模型的伪伪代码。可以对动态图中的每个快照重复调用pse十二位码来生成动态嵌入。