24 知识智能

UCLA 孙怡舟：连接附加符号知识到知识图谱嵌入

整理：智源社区 熊宇轩

在深度学习从感知迈向认知的过程中，知识图谱是表征知识、进行深度推理的一大利器。来自加州大学洛杉矶分校（UCLA）的数据挖掘领域著名学者孙怡舟教授在本届智源大会上带来了主题为《Bridging additional symbolic knowledge to knowledge graph embedding》的演讲，深入浅出地说明了如何将不确定性、一阶逻辑、多视图嵌入技术引入知识图谱嵌入领域。下面是演讲全文：

一、知识图谱嵌入研究现状

在我们眼中，知识图谱可以被表征为一个图（即由节点和边组成的异构网络）。该网络包含各种各样不同类型的节点（例如，「埃菲尔铁塔」是一个著名的景点，而「巴黎」是一个城市）。在知识图谱中，我们一般还会把节点之间的各种关系标注出来。知识图谱旨在把世界上各种各样的知识以三元组的形式存储起来。例如，在三元组（埃菲尔铁塔，位于，巴黎）中，「埃菲尔铁塔」为头实体，「位于」为关系，而「巴黎」则是尾实体。单独从事实的角度来看，知识图谱是由这样的三元组构成的；而如果从图的角度来看，由于在知识图谱中这些实体节点被关系边所连接，我们认为知识图谱可以被表征为图结构。

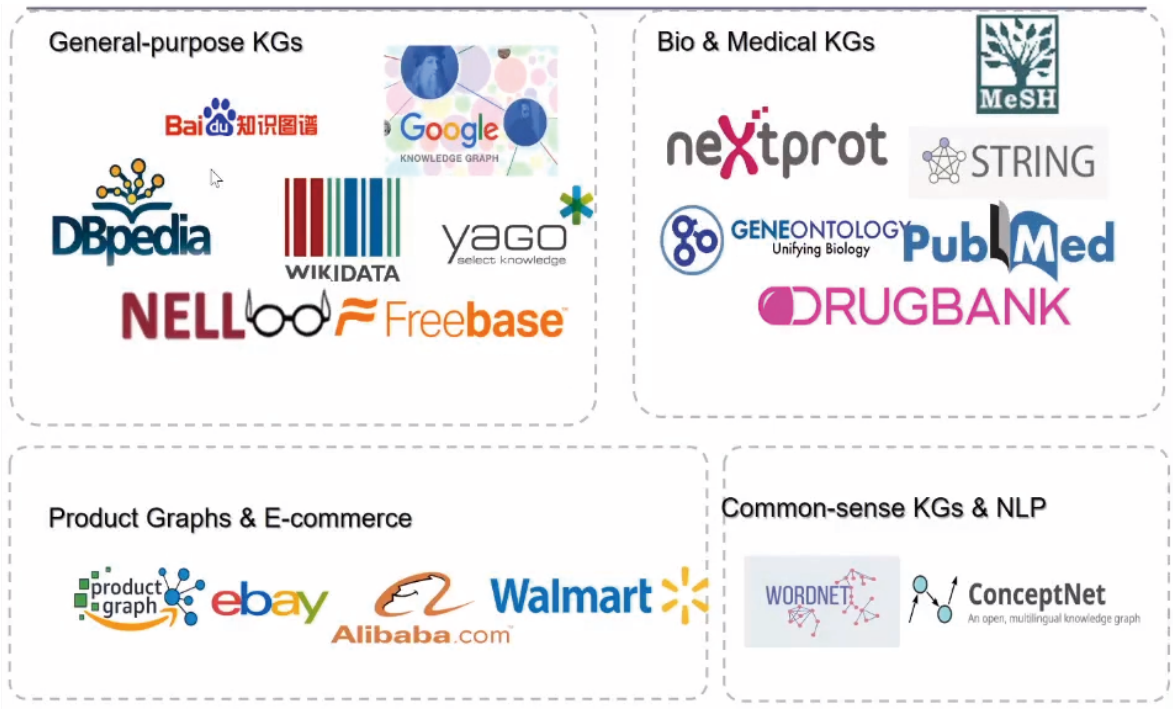


图 1：知识图谱示例

如今，知识图谱在各行各业中都非常流行。我们往往对通用知识图谱（例如，谷歌、百度的知识图谱，以及学术界制作的知识图谱 YaGO 等）更熟悉，但除此之外，还有很多行业知识图谱。例如，我们要做生物医学方面的研究，就需要具体使用生物医学领域的知识图谱（如 PubMed）；面对当下严重的新冠疫情，一些研究者们试图

从此类知识图谱中抽取出相应的实体和关系组成知识图谱，帮助我们进行新冠的预测和推理；对于一些电子商务网站来说（如阿里巴巴淘宝、亚马逊），它们也会构建针对产品的知识图谱；此外，WORDNET、ConceptNet 这类知识图谱则旨在捕获常识信息。

知识图谱的应用场景也十分广泛。例如，当我们拥有高级语义知识时，它可以帮助我们更深入地解自然语言；在问答（QA）系统和对话系统中，知识图谱可以为我们提供背景知识，使对话系统不仅仅是「无意识」地与人交流下去。

针对以上应用，为了更好地利用知识图谱中的数据，「图嵌入」技术就是一种能够很好地利用知识图谱数据的方法。知识图谱嵌入旨在用低维向量对知识图谱中存储的实体和关系进行表征，从而适用于很多的下游任务。在有的模型中，我们将关系也建模为一个低维向量；而在更多的情况下，我们将关系编码为一种代数运算。

Model	Score Function	
SE (Bordes et al., 2011)	$-\|\boldsymbol{W}_{r,1}\mathbf{h}-\boldsymbol{W}_{r,2}\mathbf{t}\|$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^k,\boldsymbol{W}_{r,\cdot}\in\mathbb{R}^{k\times k}$
TransE (Bordes et al., 2013)	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
TransX	$-\|g_{r,1}(\mathbf{h})+\mathbf{r}-g_{r,2}(\mathbf{t})\|$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
DistMult (Yang et al., 2014)	$\langle\mathbf{r},\mathbf{h},\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle\mathbf{r},\mathbf{h},\bar{\mathbf{t}}\rangle)$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^k$
HolE (Nickel et al., 2016)	$\langle\mathbf{r},\mathbf{h}\otimes\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
ConvE (Dettmers et al., 2017)	$\langle\sigma(\mathrm{vec}(\sigma([\bar{\mathbf{r}},\bar{\mathbf{h}}]*\boldsymbol{\Omega}))\boldsymbol{W}),\mathbf{t}\rangle$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^k$
RotatE	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|^2$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^k,\|r_i\|=1$

Source: Sun et al., RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space (ICLR'19)

图 2：现有的知识图谱嵌入方法

在大多数现有的知识图谱嵌入方法中，假设头实体、尾实体已知，我们将试图利用二者之间的关系构建一个评分函数（score function），该函数旨在计算三元组为真的概率（概率越大，三元组越可能为真）。接着，我们将利用该嵌入的评分函数来定义一个引导训练过程的损失函数。在定义损失函数的过程中，我们往往将观测到的三元组作为正样本，并随机采样得到一些三元组作为负样本，我们期望正样本的得分比负样本得分高。

如图 2 所示，现有的各种知识图谱嵌入方法之间最大的区别在于设计评分函数的方式。以经典的 TransE 图嵌入为例，该模型将实体和关系都表征为嵌入向量，我们将度量预测尾实体嵌入「h+r」（其中 h 为头实体的嵌入，r 为关系的嵌入）与真实尾实体嵌入 t 之间的距离。其它著名的知识图谱嵌入方法大多都沿用了这样的思路，进行了相关的改进。本领域目前性能最佳的方法是唐建教授研究组提出的 RotatE。

现有知识图谱嵌入方法的不足之处主要在于：（1）封闭世界假设（close-world assumption）：我们认为知识图谱中可以观测到的三元组都为真，所有知识图谱中没有看到的三元组都为假，即所有的知识已经被观测到了。显然，该假设并不成立，我们无法保证将实际上所有为真的三元组都纳入到了知识图谱中，也无法保证知识图谱中所有的三元组都为真。（2）扁平结构假设（flat structure assumption）：大部分现有的知识图谱嵌入算法认为知识图谱是一个扁平化的结构，即所有的实体和关系仅仅处于同一层上。这意味着我们在定义评分函数时，会

把所有的关系看作是同一个层次上的，因此使用同种类型的评分函数去定义这些关系。然而，在真实世界当中有一些具体的实体（例如，中国（国家）、北京（城市））；而对于一些抽象的概念（本体）来说，比如城市、国家，这些高级概念并不都处在同一个层次上。那么，我们应该如何处理这种层次化的结构呢?

针对以上知识图谱嵌入方法的不足之处，我们希望提出新的知识图谱嵌入的方法，或者将额外的知识引入到知识图谱嵌入方法中，从而提升算法性能。

首先，在封闭世界假设中，我们认为一般的知识图谱是确定性的，即在知识图谱中可以被观测到的三元组为真，否则就为假。为了解决这一假设的不足之处，我们将这种确定性的知识图谱变换为不确定性的知识图谱（例如，NELL，ConceptNet，它们赋予三元组为真的概率）。对于在知识图谱中观测到的三元组，我们认为它们有一定的概率为真，有一定的概率为假，因此我们将评分函数的得分从「0/1」改变为一个具有不确定性的概率值。此外，由于人对世界的认知不仅仅停留在实例的层面上，还应该具有更高级的认知过程，所以我们还应将逻辑推理引入到知识图谱中，使基于知识图谱的推理结果更加准确。

另一方面，我们认为所有的知识图谱是一个扁平化的结构，但对于一些额外的高级概念和本体而言，我们应该将它们也引入到知识图谱中，从而进行更好的推理。

二、将一阶逻辑引入不确定性知识图谱嵌入

在知识图谱中，有两类常见的错误：（1）假正例错误：观测到的三元组是错误的（2）假负例错误：遗漏掉了一些正确的事实。

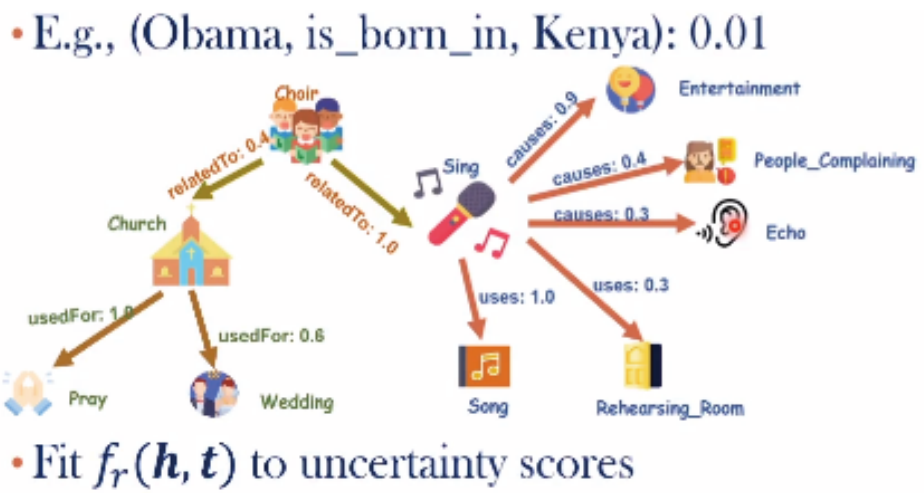


图 3：通过引入不确定性减轻假正例错误

在论文「Embedding Uncertain Knowledge Graph」(AAAI 2019) 中，我们为每个三元组赋予了一个置信度得分 (confidence score)。通过多数投票的对比方式，我们可以为真实度较低的三元组赋予较低的概率。例如，在原始数据中，声称（奥巴马，出生于，肯尼亚）的数据条目很少，而声称（奥巴马，出生于，夏威夷）的条目很多，那么我们为后者赋予高概率，为前者赋予低概率。

这样一来，我们就可以利用知识图谱中的置信度得分，缓解假正例错误，为观测到的错误三元组赋予较低概率。

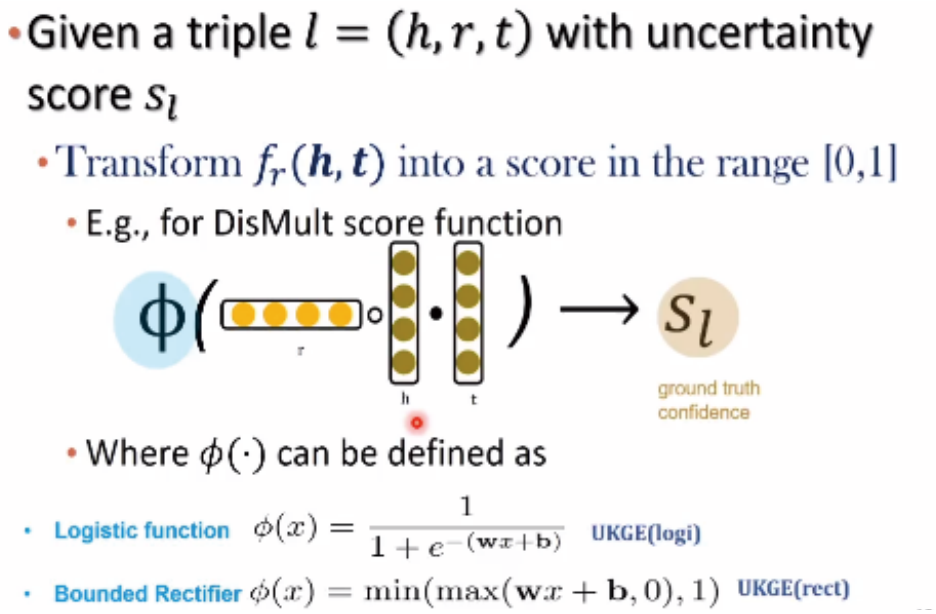


图 4：从评分函数到不确定性得分

之前，在定义评分函数时，我们希望在知识图谱中被观测到的三元组的得分较高。但是这种假设并不成立。实际上，对于知识图谱中的三元组来说，我们现在希望利用嵌入预测出来的得分与其对应的不确定性得分是一致的，这就是我们与普通的知识图谱嵌入的差异。

为了将评分函数的输出变为可以拟合不确定性得分的置信度得分（0 到 1 之间的值），我们可以用 Sigmoid 等函数将任何评分函数的输出变换到 0 到 1 之间。在本文中，我们也测试了一种名为「bounded rectifier」的变换方法：ϕ (x) = min(max(wx+b,0),1)，将评分函数限制在 0 到 1 之间。

根据我们的实验，我们发现仍然需要使用原始知识图谱遗漏的事实。但是，我们不能简单地把没有看到过的三元组的置信度得分处理成 0，我们要通过其它的推理方式推断其概率，而逻辑规则可以帮助我们实现这个目的。

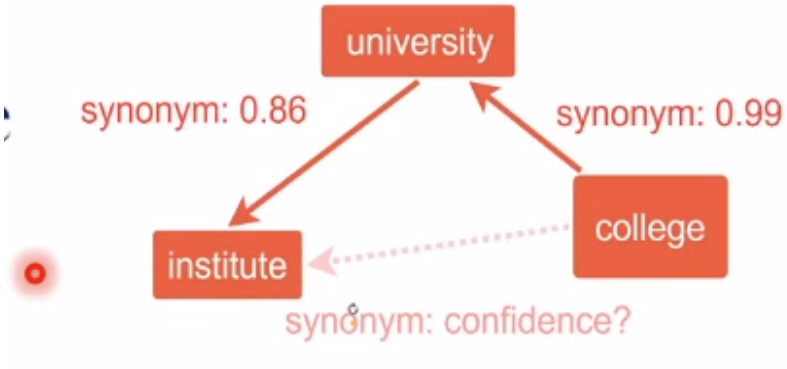


图 5：桥接逻辑规则

假设我们有一些逻辑文字和谓词，我们可以通过「逻辑体 (body) →逻辑头 (head)」的形式来描述逻辑规则，当逻辑体为真时，则逻辑头也为真。如果我们将知识图谱中三元组的关系与逻辑规则中的谓词相对应，我们就可以通过这些三元组来构建相应的逻辑规则。

例如，对于逻辑规则模板 $(\underline{A}, \text{synonym}, \underline{B}) \wedge (\underline{B}, \text{synonym}, \underline{C}) \rightarrow (\underline{A}, \text{synonym}, \underline{C})$，我们可以将占位符 A、B、C 替换成具体的知识图谱中的实体，就可以将其变为一个具体的基本规则 (ground rule) $(\text{college}, \text{synonym}, \text{university}) \wedge (\text{university}, \text{synonym}, \text{institute}) \rightarrow (\text{college}, \text{synonym}, \text{institute})$。由于知识图谱中有各种各样的谓词，我们可以利用高级规则模板来生成具体规则，再利用具体的规则进行推理。在上面的具体规则中，如果我们知道三元组 (college，是 ... 的同义词，university) 的置信度得分为 0.99，而且 (university，是 ... 的同义词，institute) 的置信度得分为 0.86，我们试图根据上面的规则推断出没有见到的 (college，是 ... 的同义词，institute) 为真的概率。

此外，传统的逻辑规则是一种布尔代数，只能处理 0/1 的运算。但是，不确定性知识图谱中的三元组存在一定为真的概率，此时我们就需要把上文中提到的布尔逻辑发展为模糊逻辑。在这里，我们通过概率软逻辑 (PSL) 来实现这种模糊逻辑，PSL 中合取 (与)、析取 (或)、取反 (非) 操作的计算方法如下:

$$
\begin{aligned}
l_1 \wedge l_2 &= \max\{0, I(l_1) + I(l_2) - 1\} \\
l_1 \vee l_2 &= \min\{1, I(l_1) + I(l_2)\} \\
\neg l_1 &= 1 - I(l_1)
\end{aligned}
$$

我们可以根据以上的原子定义，我们可以计算具体规则的概率。

在实现过程中，由于我们可以将「A → B」转化为「¬A ∨ B」，所以对于$\gamma \equiv \gamma_{body} \rightarrow \gamma_{head}$这样的规则，我们可以将其转化为 $\neg\gamma_{body} \vee \gamma_{head}$，从而根据 $\mathrm{p}_\gamma = I\left(\neg\gamma_{body} \vee \gamma_{head}\right) = \min\left\{1, 1 - I\left(\gamma_{body}\right) + I\left(\gamma_{head}\right)\right\}$ 来计算具体规则的概率。

接着，我们就可以根据具体规则的概率计算与理想状况的距离，将其作为损失函数:

$$
d_\gamma = 1 - p_\gamma = \max\{0, I(\gamma_{body}) - I(\gamma_{head})\}
$$

在优化过程中，我们的目标是最小化与理想状态的距离，即最大化所有具体规则为真的概率。在上图所示的例子中，我们首先将规则模板实例化为一个具体规则。假设 l_1 的置信度得分为 0.99，l_2 的置信度得分为 0.86，我们试图根据逻辑规则推断出 l_3 的置信度得分。

根据之前的公式，我们有

$$
d_\gamma = 1 - p_\gamma = \max\{0, I(\gamma_{body}) - I(\gamma_{head})\}
$$

其中，γ _body 为规则尾的置信度得分，γ _head 为规则头的置信度得分。将数据带入该公式得:

$$
\begin{aligned}
d_\gamma &= \max\{0, I(l_1 \wedge l_2) - I(l_3)\} \\
&= \max\{0, \overset{0.99}{s_{l_1}} + \overset{0.86}{s_{l_2}} - 1 - f(l_3)\} \\
&= \max\{0, 0.85 - f(l_3)\}
\end{aligned}
$$

其中，$f(l_3)$ 是未知的 l_3 的置信度得分，我们可以根据 college、synonym、institute 的嵌入将 $f(l_3)$ 计算出来，因此在这里需要将一些嵌入作为计算损失函数的参数。当 $f(l_3)$ 大于等于 0.85 时，损失为 0，说明嵌入很理想；当 $f(l_3)$ 小于 0.85 时，损失函数则大于 0，且 $f(l_3)$ 越小损失越大，因此我们需要调整嵌入，使损失函数接近于 0。这就是逻辑规则影响知识图谱嵌入的机制。

$$
\mathcal{J} = \sum_{l \in \mathcal{L}^+} |f(l) - s_l|^2 + \sum_{l \in \mathcal{L}^-} \sum_{\gamma \in \Gamma_l} |\psi_\gamma(f(l))|^2
$$

Embedding-based confidence function

Distance to satisfaction for a ground rule γ, where *triple* l is involved in

图 6：新型嵌入模型

因此，新型的嵌入模型包含两个部分。首先，对于知识图谱中已观测到的三元组，我们期望通过嵌入定义的置信度与观测到的置信度一致；此外，我们还应该利用未观测到的三元组。如果逻辑规则覆盖到某条未观测到的三元组，我们需要计算该三元组所在的具体规则的 d_γ (损失)。在如图 6 所示的公式中，等号右侧的前后两项都包含知识图谱嵌入，这说明三元组的嵌入既受到知识图谱中其它三元组的影响，也受到高级逻辑规则的影响。

- **Datasets**

Dataset	#Ent.	#Rel.	#Rel. Facts	Avg(s)	Std(s)
CN15k	15,000	36	241,158	0.629	0.232
NL27k	27,221	404	175,412	0.797	0.242
PPI5k	5,000	7	271,666	0.415	0.213

- **Logic Rules**

(A, relatedto, B)∧(B,relatedto,C)→(A,relatedto,C)
(A, causes, B)∧(B,causes,C)→(A,causes,C)

(A, competeswith, B)∧(B,competeswith,C)→(A,competeswith,C)
(A, atheletePlaysForTeam, B)∧(B, teamPlaysSports, C)→(A, atheletePlaysSports, C)

(A, binding, B)∧(B,binding,C)→(A,binding,C)

图 7：实验结果

我们在常用的「CN15k」、「NL27k」、「PPI5K」三个数据集上进行了实验，我们在这篇论文中针对每个数据集手动设计了一些逻辑规则，接下来我们还将研究如何自动发掘逻辑规则。

在实验中，我们三类对比基线进行了比较:

(1) 处理确定性知识图谱嵌入的模型。例如，TransE。
(2) 只提供节点嵌入的不确定性图嵌入方法 (如 URGE)，但是这些方法并没有区分不同种类的节点和关系。
(3) 本模型的两种简化版本:

- 不进行负采样，在训练时不考虑知识图谱中未观测到的三元组。
- 不使用 PSL，将知识图谱中为观测到的三元组的置信度看做 0。

- Metrics: MSE and MAE ($\times 10^{-2}$)

Dataset	CN15k		NL27k		PPI5k	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
URGE	10.32	22.72	7.48	11.35	1.44	6.00
UKGE_{n-}	23.96	30.38	24.86	36.67	7.46	19.32
UKGE_{p-}	9.02	20.05	2.67	7.03	0.96	4.09
UKGE_{rect}	**8.61**	**19.90**	**2.36**	**6.90**	**0.95**	**3.79**
UKGE_{logi}	9.86	20.74	3.43	7.93	0.96	4.07

图 8：对比实验结果

如上图所示，在关系事实置信度得分预测实验 (Relation Fact Confidence Score Prediction) 中，给定一个三元组，我们旨在预测其置信度得分。两种简化版本的模型性能相较于我们提出的模型有很大下降，因此需要使用在知识图谱中未观测到的三元组，也不能直接将它们的置信度当做 0。最下面两行为本文提出的方法，其性能相对对比基线具有很大的优势。

metrics	CN15K		NL27k		PPI5k	
Dataset	linear	exp.	linear	exp.	linear	exp.
TransE	0.601	0.591	0.730	0.722	0.710	0.700
DistMult	0.689	0.677	0.911	0.897	0.894	0.880
ComplEx	0.723	0.712	0.921	0.913	0.896	0.881
URGE	0.572	0.570	0.593	0.593	0.726	0.723
UKGE_{n-}	0.236	0.232	0.245	0.245	0.514	0.517
UKGE_{p-}	0.769	0.768	0.933	0.929	0.940	0.944
UKGE_{rect}	0.773	0.775	0.939	0.942	0.946	0.946
UKGE_{logi}	**0.789**	**0.788**	**0.955**	**0.956**	**0.970**	**0.969**

图 9：在关系事实排序任务中，我们的模型也展现出了最佳的效果

图 10：关系事实排序的案例研究

在上图中的案例中，我们希望通过我们的模型完成一些知识图谱补全的工作。有趣的是，一些原本知识图谱的真实值（Ground Truth）中没有涵盖到的一些客观事实可以被我们的模型预测出来。

三、将本体概念与元关系引入知识图谱嵌入

在前面的工作中，我们介绍了如何利用逻辑规则和不确定性来处理封闭世界假设的缺陷。下面，我们将研究如何将层次化的概念和关系引入到知识图谱嵌入中。

在很多情况下，我们并不需要知道具体的事实。例如，如果我们想对科学家这个职业想有所了解，我们知道科学家们往往在研究所或者大学工作，他们可能毕业于某所大学，我们不需要具体到某一个科学家的情况，就可以对科学家的职业有大致的了解，这种了解其实是被称作为元级别（meta-level）的推理，而不是实例（instance）级别的推理。

如果我们知道一个具体的人 Anna 是一名科学家，我们会认为她的嵌入与其他科学家的嵌入应该是相近的。一旦我们拥有这样的本体知识，我们就可以对实例进行更好的推理。尤其是，对于某些处于长尾分布尾端的实体而言，它们在知识图谱中十分稀疏。如果我们拥有相关的本体知识，就会大大提升推理的准确率。

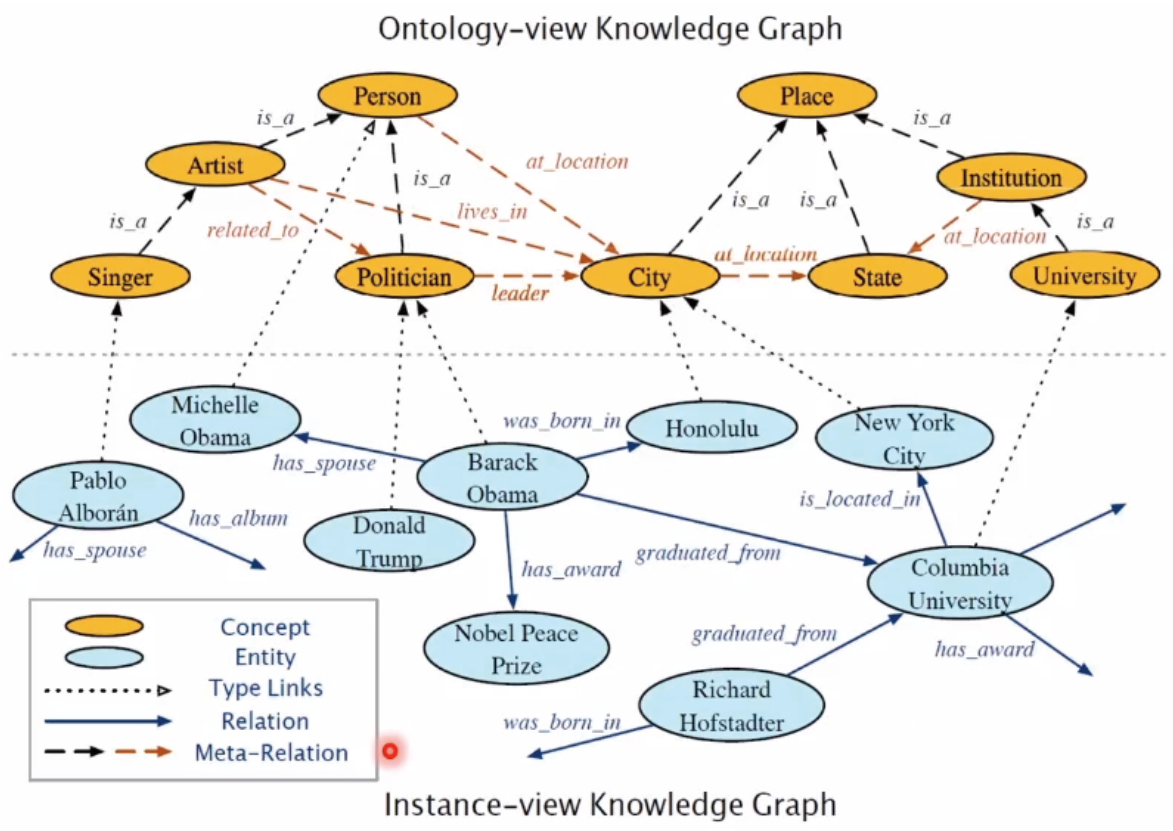


图 11：实例视图和本体视图的知识图谱

实际上，更加全面的知识图谱应该是一种包含底层的实例视图知识图谱（包含实体及其关系），以及上层的本体视图知识图谱（其节点为抽象的高级概念，边为元关系）的二视图知识图谱。在这两种视图之间，我们使用「跨视图链接」（cross-view link）将它们连接起来。

结合本体和实例视图知识图谱的思路与我们曾经在研究异质网络时的一些想法很相似。在研究异质网络时，我们会使用网络模式（schema）作为实体对及其之间关系的模板。本体视图知识图谱相较于网络模式更进一步，将概念之间的层次也考虑了进来。

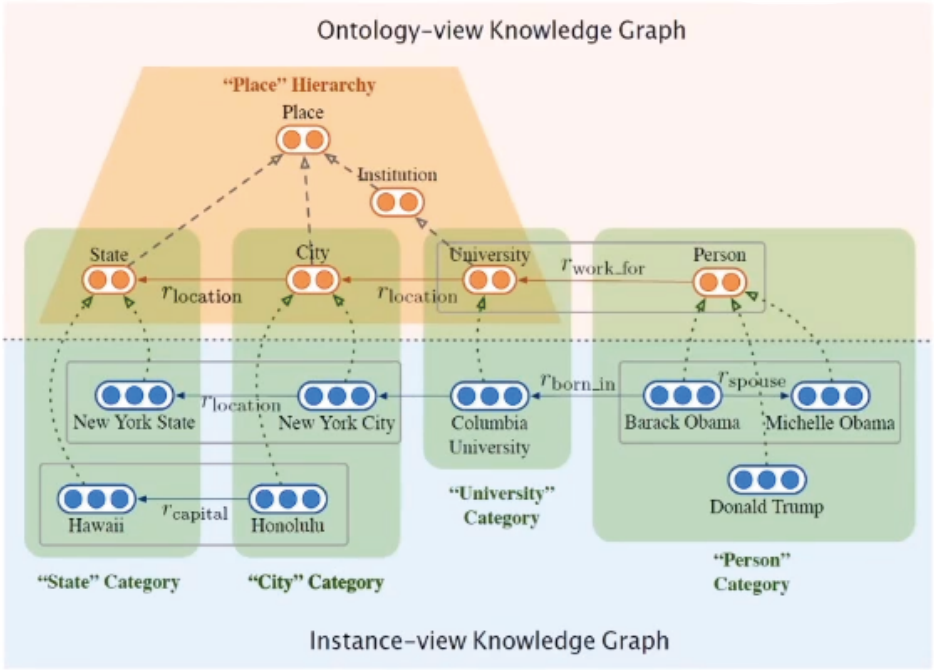


图 12：二视图联合嵌入

假设上层的本体视图及其与下层实例视图之间的「跨视图链接」已知，我们很有必要进行二视图的联合嵌入。首先，如果每个实例的嵌入性能已经很好，那么它们为构建本体概念提供了有效的高级引导。此外，如果我们已经拥有了本体概念的嵌入，那么我们可以大致推测出实体的嵌入，这两个过程是互相促进的。

我们基于跨视图链接将实例嵌入空间和本体嵌入空间联系起来，主要的实现方法有两种:

跨视图聚合（CG)： 直接将本体概念 c 与实例实体 e 放入同一个嵌入空间，若 c 与 e 有跨视图链接，那么我们直接迫使它们在嵌入空间中的距离较为接近。

$$J_{\text{Cross}}^{\text{CG}} = \frac{1}{|\mathcal{S}|} \sum_{(e,c)\in\mathcal{S}} \left[||\mathbf{c} - \mathbf{e}||_2 - \gamma^{\text{CG}} \right]_+$$

跨视图变换（CT)： 并不强迫本体概念与实例实体在同一个嵌入空间中，我们对实体嵌入进行变换，将其变换到本体概念所在的空间中，再将本体概念的嵌入与变换后的实体嵌入进行比较，从而设计损失函数。

$$J_{\text{Cross}}^{\text{CT}} = \frac{1}{|\mathcal{S}|} \sum_{\substack{(e,c)\in\mathcal{S} \\ \wedge(e,c')\notin\mathcal{S}}} \left[\gamma^{\text{CT}} + ||\mathbf{c} - f_{\text{CT}}(\mathbf{e})||_2 - \left\|\mathbf{c}' - f_{\text{CT}}(\mathbf{e})\right\|_2 \right]_+$$

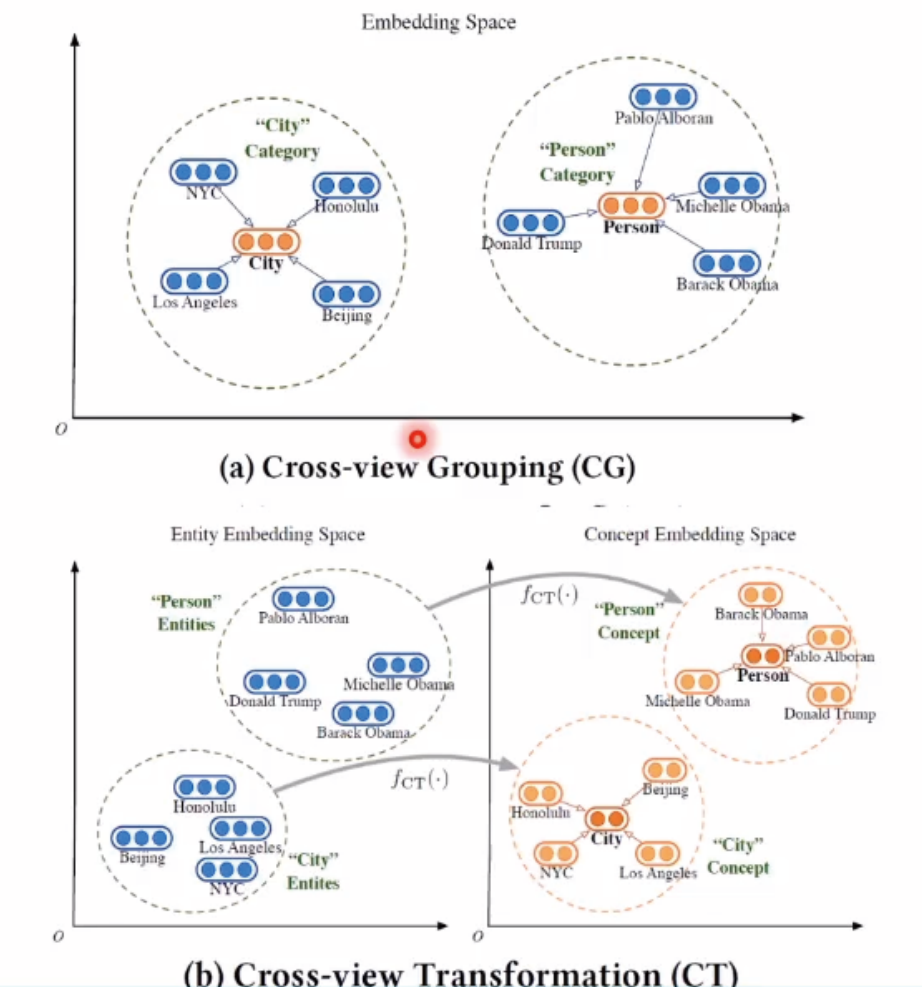


图 13：跨视图聚合与跨视图变换的示意图

- Base models could be any existing KG embedding models
  - Examples: $f_{\text{TransE}}(\mathbf{h},\mathbf{r},\mathbf{t}) = -||\mathbf{h}+\mathbf{r}-\mathbf{t}||_2$
    $f_{\text{Mult}}(\mathbf{h},\mathbf{r},\mathbf{t}) = (\mathbf{h}\circ\mathbf{t})\cdot\mathbf{r}$
    $f_{\text{HolE}}(\mathbf{h},\mathbf{r},\mathbf{t}) = (\mathbf{h}\star\mathbf{t})\cdot\mathbf{r}$
- Hierarchy-aware embedding
  - Similar to CT, transform lower-level concepts into higher-level concepts

$$g_{\text{HA}}(\mathbf{c}_h) = \sigma(\mathbf{W}_{\text{HA}}\cdot\mathbf{c}_l + \mathbf{b}_{\text{HA}})$$

图 14：视图内的层次化建模

对于视图内（Intra-view）的嵌入来说，我们可以用任何现有的知识图谱嵌入技术作为基础模型。在本体视图中，也存在一些概念上的上下位层次关系，我们也可以将位于下层的概念变换到为上层概念，对其进行比较，从而定义损失函数。

$$J = J_{\text{Intra}} + \omega \cdot J_{\text{Cross}}$$

$$\bullet\ \text{Where}\quad J_{\text{Intra}} = J_{\text{Intra}}^{\mathcal{G}_I} + \alpha_1 \cdot J_{\text{Intra}}^{\mathcal{G}_O \setminus \mathcal{T}} + \alpha_2 \cdot J_{\text{Intra}}^{\text{HA}}$$

综上所述，我们提出的联合分布模型将跨视图（Cross-view）模型与视图内（Intra-view）模型整合到一起。跨视图模型的实现方式包含跨视图聚合（CG）与跨视图变换（CT）；对于视图内模型而言，实例视图内的嵌入可以通过任意的基础模型（如 TransE、RotatE）实现，本体视图内的嵌入可以基于视图内上下位概念的变换设计新的损失函数。

- **Datasets**
  - Constructed two new datasets from YAGO and DBpedia

Dataset	Instance Graph $\mathcal{G}_I$			Ontology Graph $\mathcal{G}_O$			Type Links $\mathcal{S}$
	#Entities	#Relations	#Triples	#Concepts	#Meta-relations	#Triples	
YAGO26K-906	26,078	34	390,738	906	30	8,962	9,962
DB111K-174	111,762	305	863,643	174	20	763	99,748

- **Tasks**
  - Triple completion
  - Entity typing
  - Ontology population
- **Baselines: treat all links equally**

图 15：实验设定

在实验中，由于缺乏直接的二视图知识图谱嵌入对比基线，我们基于 YAGO 和 DBpedia 数据集手动构建了实验数据集。实际上在生物学领域中，这种二视图知识图谱是更为普遍的。

我们在三类任务上测试了我们的模型：（1）三元组补全：在传统的实体知识图谱上进行补全任务；（2）实体分类：相当于预测跨视图链接的类别；（3）本体填充：在本体概念知识图谱中，进行类似于知识图谱补全的抽象推理。

Datasets	YAGO26K-906						DB111K-174					
Graphs	$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion			$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion		
Metrics	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE (base)	0.195	14.09	34.51	0.145	12.29	20.59	0.327	22.26	49.01	0.313	23.22	46.91
TransE (all)	0.187	13.73	35.05	0.189	14.72	24.36	0.318	22.70	48.12	0.539	47.90	61.84
TransC	0.252	15.71	37.79	–	–	–	0.359	24.83	49.31	–	–	–
JOIE-TransE-CG	0.264	16.38	35.45	0.189	11.16	29.44	0.394	27.75	51.20	0.598	53.84	71.79
JOIE-TransE-CT	0.292	**18.72**	44.14	0.240	14.49	33.47	0.443	32.10	67.89	**0.622**	**58.10**	72.97
JOIE-HATransE-CT	**0.306**	18.62	**51.72**	**0.263**	**16.72**	**38.46**	**0.473**	**33.79**	**71.37**	0.591	52.07	**79.65**
DistMult (base)	0.253	22.91	28.76	0.197	**17.72**	25.08	0.265	25.95	27.63	0.235	15.18	29.11
DistMult (all)	0.288	**24.06**	31.24	0.156	14.32	16.54	0.280	27.24	29.70	0.501	45.52	64.73
JOIE-Mult-CG	0.274	18.80	37.45	0.198	11.16	27.91	0.320	23.44	49.49	0.532	46.15	68.91
JOIE-Mult-CT	**0.309**	20.40	**46.15**	**0.207**	14.71	30.43	**0.404**	**26.55**	**60.86**	**0.563**	**50.50**	71.62
JOIE-HAMult-CT	0.296	19.39	45.48	0.202	13.72	**31.10**	0.369	24.82	55.86	0.521	38.46	**77.25**
HolE (base)	0.265	**25.90**	28.31	0.192	18.70	20.29	0.301	29.24	31.51	0.227	18.91	32.83
HolE (all)	0.252	24.22	26.56	0.138	11.29	14.43	0.295	28.70	30.32	0.432	38.80	56.05
JOIE-HolE-CG	0.253	18.75	34.11	0.167	13.04	22.33	0.361	24.13	46.15	0.469	41.89	62.16
JOIE-HolE-CT	0.313	20.40	47.80	0.229	**20.85**	28.42	0.425	29.09	66.88	**0.514**	**43.24**	69.23
JOIE-HAHolE-CT	**0.327**	22.42	**52.41**	**0.236**	16.72	**30.96**	**0.464**	**33.11**	**69.56**	0.503	40.80	**71.03**

图 16：三元组补全实验结果

在实验中，我们首先忽略实例的类别，将 TransE 等基础模型作为对比基线；接着，我们测试了了不同的跨视图联合嵌入方法，以及层次化变换对于模型性能的影响。实验结果表明，在联合嵌入时使用跨视图变化的性能比使用跨视图聚合时更好；在本体试图内部，使用概念之间的上下位层次化信息是十分有必要的。

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
TransE	0.144	7.32	35.26	0.503	43.67	60.78
MTransE	0.689	60.87	77.64	0.672	59.87	81.32
JOIE-TransE-CG	0.829	72.63	93.35	0.828	70.58	95.11
JOIE-TransE-CT	0.843	75.31	93.18	0.846	74.41	94.53
JOIE-HATransE-CT	**0.897**	**85.60**	**95.91**	**0.857**	**75.55**	**95.91**
DistMult	0.411	36.07	55.32	0.551	49.83	68.01
JOIE-Mult-CG	0.762	62.62	87.82	0.764	60.83	91.80
JOIE-Mult-CT	0.805	70.83	89.25	**0.791**	65.30	**93.47**
JOIE-HAMult-CT	**0.865**	**81.63**	**91.83**	0.778	**69.38**	85.71
HolE	0.395	34.83	54.79	0.504	44.75	65.38
JOIE-HolE-CG	0.777	65.30	87.89	0.784	66.75	89.37
JOIE-HolE-CT	0.813	72.27	88.71	0.805	68.84	**91.22**
JOIE-HAHolE-CT	**0.888**	**83.67**	**93.87**	**0.808**	**72.51**	89.79

图 17：实体分类实验结果

实体分类实验的结果也说明利用视图间跨模态变换以及视图内层次化信息有助于提升模型性能。

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
DistMult	0.156	10.89	25.33	0.219	16.48	33.71
MTransE	0.526	46.45	67.25	0.505	46.67	64.36
JOIE-TransE-CG	0.708	59.97	79.80	0.741	64.45	83.05
JOIE-TransE-CT	0.737	62.05	82.60	0.758	66.35	83.80
JOIE-HATransE-CT	**0.802**	**69.66**	**87.75**	**0.760**	**67.34**	**89.79**

图 18：对推理长尾分布尾端实体推理的帮助

我们认为本文提出的模型有助于对处于长尾分布尾端实体的推理。例如，一个人与其它实体之间的连接十分稀疏，但是如果我们知道 他是一名科学家，就可以获知很多关于他的信息。

Query	Top 5 Populated Triples with distances
(scientist,?r, university)	scientist, ***graduated from***, university (0.499) scientist, ***isLeaderOf***, university (1.082) scientist, *isKnownFor*, university (1.098) scientist, *created*, university (1.119) scientist, ***livesIn***, university (1.141)
(boxer, ?r, club)	boxer, ***playsFor***, club (1.467) boxer, ***isAffiliatedTo***, club (1.474) boxer, ***worksAt***, club (1.479) boxer, *graduatedFrom*, club (1.497) boxer, *isConnectedTo*, club (1.552)

$$f_{\mathrm{CT}}^{\mathrm{inv}}(\mathbf{c}_{\mathrm{country}}) - f_{\mathrm{CT}}^{\mathrm{inv}}(\mathbf{c}_{\mathrm{office}}).$$

图 19：本体填充实验结果

本体填充指的是，在给定两个本体概念时，预测它们之间的关系。本文提出的模型在该任务上也取得了不错的效果。

四、结语

知识图谱本身已经是符号化的数据表示方式，而知识图谱嵌入实际上也具有一定将符号化表示与连续表示桥接起来的功能。在本文中，我们讨论了如何额外的信息加入到知识图谱嵌入中，使其更加丰富。

首先，我们可以利用人总结出来的逻辑规则，将高阶关系引入知识图谱。因为知识图谱嵌入实际上是引入了一阶关系，而逻辑规则引入了更多的谓词，所以构建了具有高阶依赖的模型。此外，我们不应该将知识图谱看做一种扁平化的结构，我们应该将本体、概念这种额外信息加入到知识图谱中。

在未来，我们试图进一步研究如何更好地将逻辑规则与知识图谱结合起来，如何自动地挖掘这些逻辑规则。同时，我们也考虑研究如何利用网络模式（schema）级别、本体级别的信息改进知识图谱，进行更好的多跳推理。

微软雷德蒙德研究院东昱晓：图表示学习

整理：智源社区 熊宇轩

从网络嵌入、异构网络挖掘、知识图谱到最近年来持续火热的图神经网络，基于图的学习一直是人工智能领域中不可忽视的研究方向。在本届智源大会上，来自微软雷德蒙德研究院的高级应用科学家东昱晓博士带来了以《图表示学习： 嵌入、图神经网络、预训练》为题的主题演讲，从图表示学习的基本概念、嵌入方法、与图神经网络的结合等方面为我们提供了对该领域的一个全景式讲解。

演讲全文如下:

在本次演讲中，我们将首先简要介绍浅层的图嵌入，然后重点讨论图神经网络以及图表示学习的预训练。

在现实世界中，物体往往并不是单独存在的，物体与物体之间总是存在各种各样的关系。因此，我们可以用图结构（网络）对这些关系进行抽象建模。

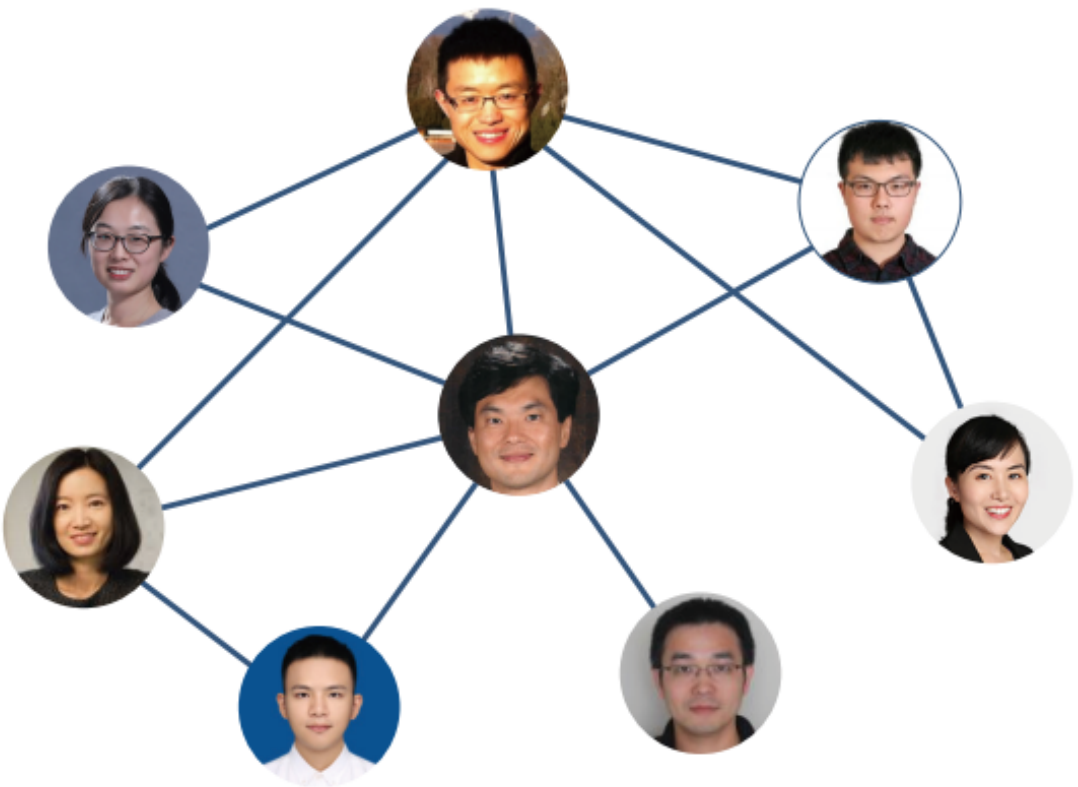


图 1：学术合作网络图

如图 1 所示，在学术合作网络中，如果根据研究者之间的关系构建一个网络结构，它能够表达传统结构不能传递的数据信息（如哪些学者之间的合作关系比较紧密）。

实际上，学术网络是一个相对比较复杂的网络。除了上文中介绍的学者节点之外，在学术网络中经常会有学者发表的期刊会议论文节点，以及学者所属的学术机构等节点。除了学术网络，现实世界中也还有各种各样其它的网络。例如，社交网络、职业网络、生物神经网络、知识图谱、因特网、交通路网。因此，我们可以用网络结构方便地描述现实世界中复杂的关系和现象。

给定这些图数据或者关系数据，我们会关心哪些问题呢？传统意义上来说，对于输入的图结构，我们可能比较关心图上的节点的类别（即节点分类、链接预测等问题）。一般来说，我们首先会人为定义一些结构化的特征，进行非常复杂的结构化特征抽取，从而形成结构化矩阵。该矩阵会成为机器学习和数据挖掘算法的输入，用于解决下游任务（例如，节点分类、链接预测等）。

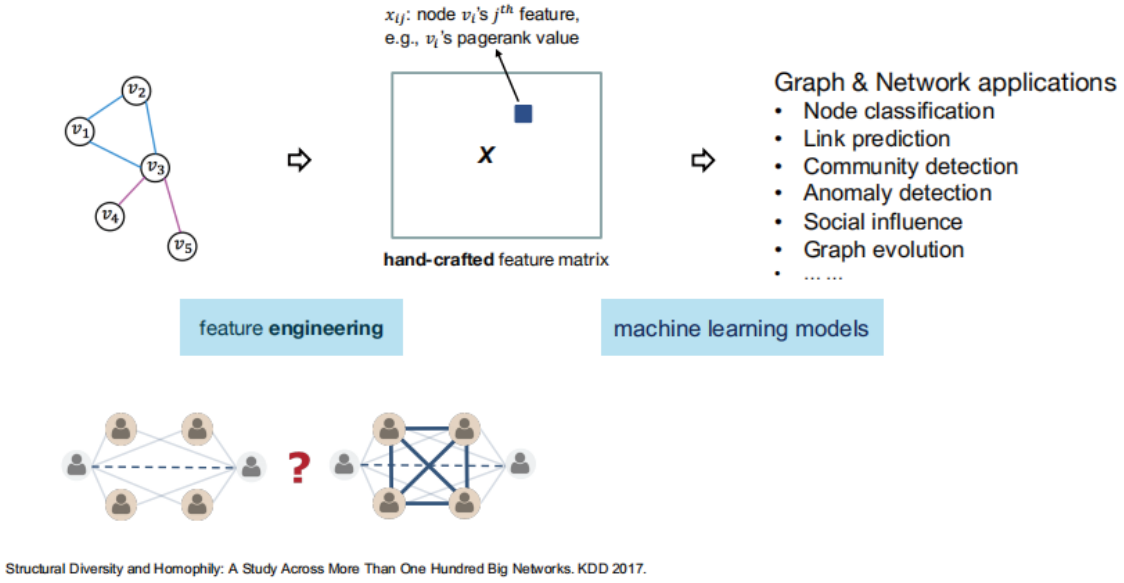


图 2：图挖掘范式

特征工程是非常复杂的，需要我们具备相应的专业知识，并且对于数据有充分的了解。如图 2 左下角所示，如果我们想知道两个用户是否有联系或者在未来是否会成为朋友，我们可以观察他们的共同邻居之间是否相互认识。根据传统的做法，我们会把结构化的特征抽出来，对两个用户之间成为朋友的概率进行建模。显然，这个过程需要较多的人为设计与计算。

到了图表式学习时代，对于给定的图结构输入，我们希望借助深度学习、表式学习的最新技术，从图结构中自动抽取结构化特征。在大多数情况下，这种结构化的特征位于隐空间上，我们并不能明确地知道特征的每个维度代表什么意思。与传统方法相类似，在得到特征矩阵后，我们可以用它来解决一些下游任务。

图表示学习的应用场景非常广泛。以微软学术数据为例，我们可以把它检索到的包含两亿多篇文章、两亿多个学者、四千多个学术会议、接近五万份期刊的异构学术网络输入到表式学习算法当中，从而生成每一个节点的节点表示。

在拥有了节点表示后，我们就可以基于该表示进行开展许多下游任务（例如，相似性搜索与推荐）。比如，当我

们关心“哪些期刊与《自然》期刊的距离最近或最相似”时，根据图表式学习，我们得到的结果是《Nature》、《PNAS》、《Nature Communication》，这与我们人类的主观认识十分接近。当我们关心”哪些学校或机构与哈佛大学在学术上很相似”时，我们根据图表示学习技术得到的排在前五的答案是斯坦福大学、耶鲁大学、哥伦比亚大学、芝加哥大学和约翰霍普金斯大学。

此外，我们还可以根据针对学术网络使用图表式学习所得到的嵌入进行一些推理。当我们关心糖尿病时，可以根据所有与糖尿病相关的文章中推理出糖尿病的起因、症状、治疗方法。

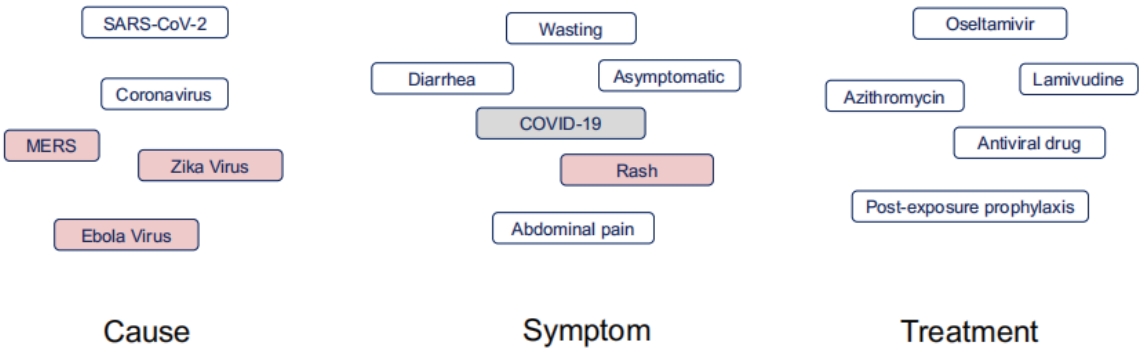


图 3：利用 MAG 学术网络对 COVID–19 进行推理

我们最近也在尝试利用最近发表的关于新冠肺炎的学术数据进行推理，试图找出新冠肺炎的症状与起因，并推理出相应的治疗方案。在图 3 中，白色与红色的方框分别代表推理正确与错误的条目。

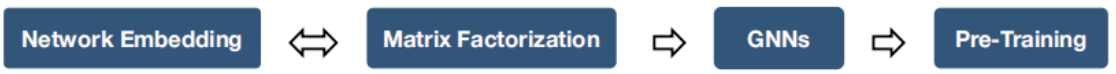


图 4：图表示学习流程

接下来，我将从网络嵌入、矩阵分解、图神经网络、预训练这四方面介绍我们在图表示学习领域的相关工作。

一、网络嵌入学习

网络嵌学习的起源可以追溯到几十年前。起初，许多学者曾经研究基于图的拉普拉斯矩阵的谱聚类、SVD 分解，并生成每个节点的特征向量。近年来，DeepWalk 模型引发了人们对于网络嵌入学习的新一轮研究。DeepWalk 模型受到了自然语言处理领域中词嵌入技术的启发，将文本数据中的每一个句子当做单词的序列，然后将该序列输入到两层的神经网络当中，从而学习到每个单词的嵌入。

类似地，DeepWalk 重点研究如何将网络结构转化为序列化的结构，然后将这个序列化的结构输入到 Skip–Gram 模型中，从而学习到每个节点的网络表示。具体而言，作者通过一种较为直接的方式实现了上述目标的思路： 他们把在网络上进行随机游走经过的节点记录下来，生成节点链； 再将生成的节点链作为 NLP 中的句子输入到 Skip–Gram 模型中，从而得到每个节点的节点表示。

DeepWalk 模型是建立在上世界 50 年代初 Harris 提出的分布式假设之上的，该假设指出：在自然语言中，两个在相似的上下文中出现的单词具有相似的意义。引申到网络表示学习中后，其隐含的假设是：如果两个节点经常在相似的结构中出现，我们认为这两个节点具有相似的意义。具体而言，在 DeepWalk 模型中，我们通过随机游走生成的节点链捕捉这种相似的结构。

在 DeepWalk 提出之后，许多研究者们试图对其进行各种改进。例如，唐健教授研究组提出了 Line 模型，将随机游走的步长设为 1，从而能够快速地学习到网络中的节点表示。Node2vec 的作者利用了网络中蕴含一定网络属性的独特的三角形结构，设定了基于三角形的有偏随机游走，即对随机游走进行一定的引导。针对包含不同类型节点和关系的异构网络，Metapath2vec 通过预先定义的 metapath（由孙怡舟老师团队提出）引导随机游走过程。上述模型在完成随机游走得到序列化结构之后，会将序列化的结构统一输入到 Skip-gram 模型中学习网络节点的表示。

至此，我们回顾了比较流行的几种学习网络嵌入的方法，它们都以随机游走和 Skip-Gram 为基础。那么我们该如何理解这些方法呢?

二、矩阵分解

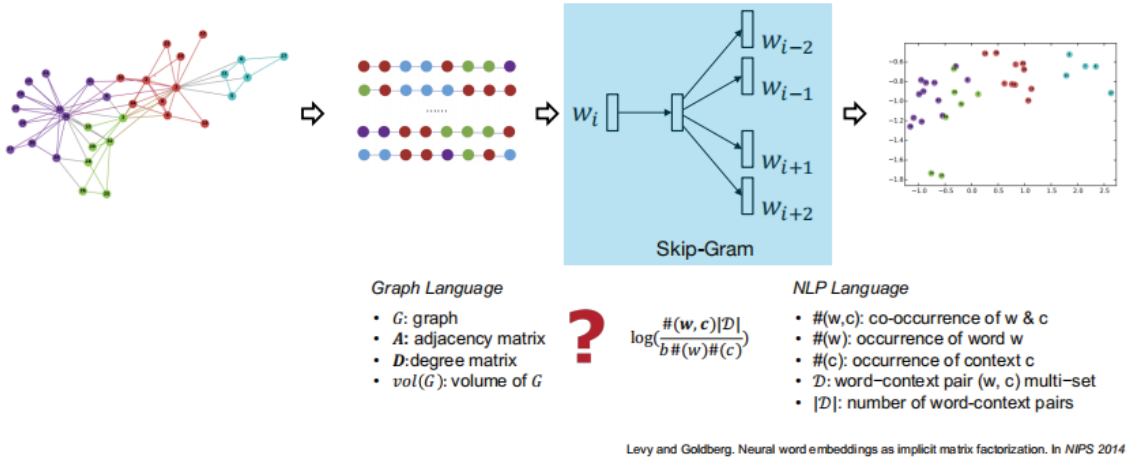


图 5：理解随机游走 +Skip Gram

在网络上做随机游走得到序列化结构，并将随机游走的结果输入到一个两层的神经网络中，这对于网络操作意味着什么?

在论文《Neural word embeddings as implicit matrix factorization》中，作者指出的上述操作等价于隐式的矩阵分解。这份工作也受到了 NLP 领域研究成果的启发，即 Skip-Gram 模型等价于分解 PMI 矩阵，该 PMI 矩阵中的五个变量对应于 NLP 领域中的一些概念。例如，图 14 中右下角的 D 代表所有“单词 – 上下文”对的多重集。类比到图表示学习领域，我们也试图将“随机游走 +Skip Gram”的图表示学习范式归纳为一种矩阵分解。

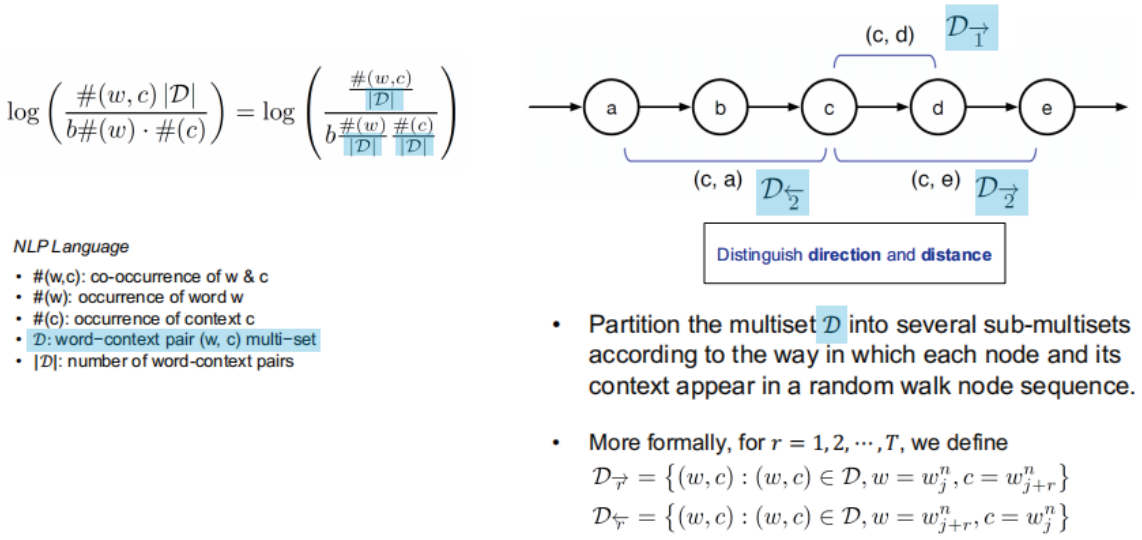


图 6：图表示学习的矩阵分解视角

如图 6 所示，PMI 矩阵中的每一项都是用 NLP 领域中的概念表示的，若令分子分母中带“#”的项同时除以|D|，我们可以得到 $\frac{\frac{\#(w,c)}{|D|}}{b\frac{\#(w)}{|D|}\frac{\#(c)}{|D|}}$。在网络中进行随机游走时，我们通过以下方式构建 D：对于类似于图中的 a、b、c、d、e 形成的随机游走链，我们首先考虑随机游走的方向。以 c 为中心节点，其右侧有 d 和 e，左侧有 a 和 b。同时，我们可以按照方向和距离将以 c 为中心的“单词 – 上下文”对分解成不同的子多重集。具体来说，以 c 为中心点，d 与 c 构成了从 c 往右走一步以内的“单词 – 上下文”对，a 与 c 构成了从 c 往左走两步以内的“单词 – 上下文”对（对应到网络中即为“节点 – 上下文”对）。我们可以通过上述方式表示 D，进一步表示我们所关心的 $\frac{\frac{\#(w,c)}{|D|}}{b\frac{\#(w)}{|D|}\frac{\#(c)}{|D|}}$。

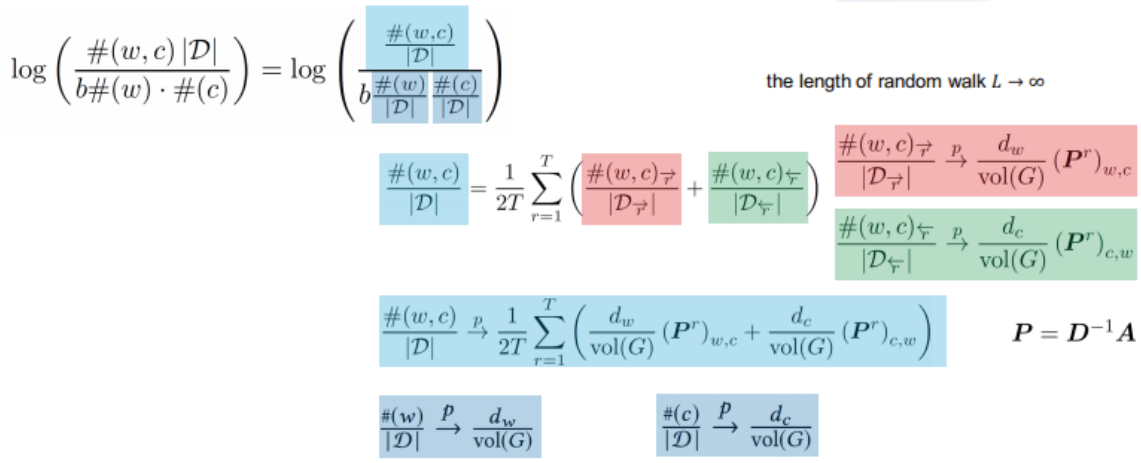


图 7：PMI 矩阵分解过程

以分子为例，我们根据个节点与中心节点的距离将分子中的 |D| 拆分成了括号内红色和绿色的两项。由大数定理可知，当我们在图上进行无限步随机游走时，红色和绿色的两项会依概率收敛于括号后面的形式（其中的每一项都与图表示学习领域的概念相对应）。经过简单的整理，我们可以得到分子等于:

$$\frac{\#(w,c)}{|\mathcal{D}|} \xrightarrow{p} \frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_{\omega}}{vol(G)}\left(P^{r}\right)_{\omega,c}+\frac{d_{\omega}}{vol(G)}\left(P^{r}\right)_{c,\omega}\right)$$

通过同样的分析手段，我们可以将分布中的两项整理成如图 7 中深蓝色方框中的形式。将以上三项带回到 PMI 矩阵中，我们得到最终的矩阵形式为:

$$\frac{vol(\mathrm{G})}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(D^{-1}A\right)^{r}\right)D^{-1}$$

因此，在图上进行随机游走，再将随机游走结果输入给 Skip-Gram 模型的过程，等价于隐式的矩阵分解，其形式为 $\log\left(\frac{vol(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(D^{-1}A\right)^{r}\right)\mathrm{D}^{-1}\right)$。矩阵当中每一项都可以与图表示学习领域的概念相对应，其中 A 是邻接矩阵，D 是度矩阵。

与 DeepWalk 相类似，LINE、PTE、node2vec 等模型也有等价的矩阵分解形式。

- DeepWalk $\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$
- LINE $\log\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\right)$
- PTE $\log\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{ww}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{ww}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{dw}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{dw}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{lw}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{lw}})^{-1}\end{bmatrix}\right)-\log b$
- node2vec $\log\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}^{r}_{c,w,u}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}^{r}_{w,c,u}\right)}{b\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)$

因此，我们提出了名为 NetMF 的算法直接对 DeepWalk 进行显式的矩阵分解。NetMF 包含两个步骤：(1) 构建 DeepWalk 的隐式分解矩阵 (2) 对前一步构建出来的矩阵直接进行分解，从而得到对每个节点的表示。

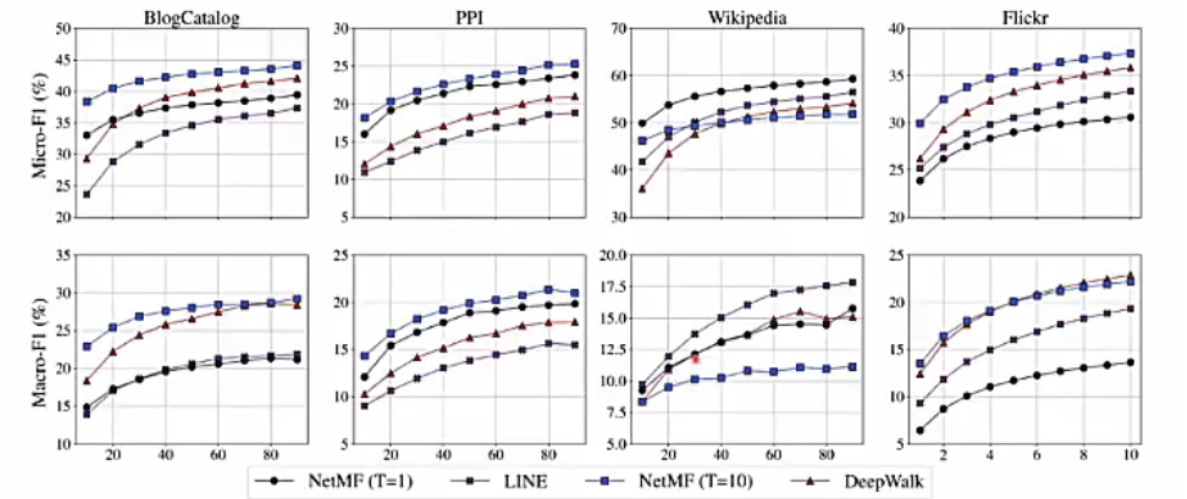


图 8：NetMF 实验结果

NetMF 的实验结果如图 8 所示，我们发现相较于使用隐式矩阵分解，使用显式的矩阵分解可以获得性能上的提升。

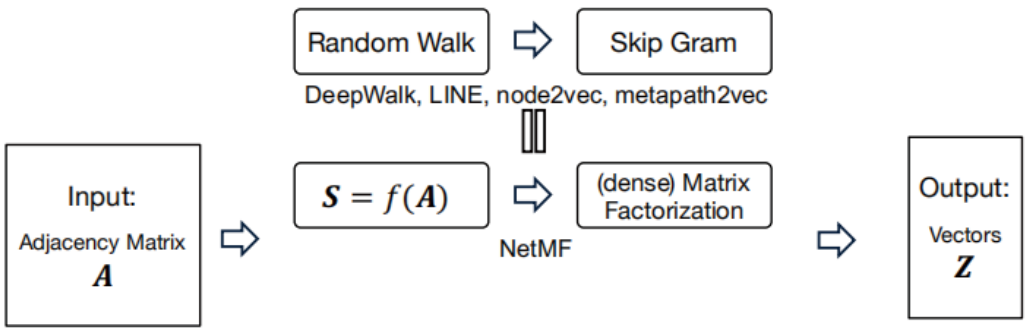


图 9：网络嵌入小结

对于网络表示学习而言，随机游走 +Skip–Gram 的范式等价于矩阵分解。

然而，现实生活中的大部分网络都遵循小世界模型，即在很少的步数内可以将网络中任意两个点连接起来（例如著名的“六度分隔”理论」）。如图 9 所示，对于 DeepWalk 需要分解的矩阵 S 而言，$D^{-1}A$ 为归一化后的邻接矩阵，它代表随机游走在矩阵上的转移概率。若 T 被设置成 5 或者 6，则矩阵为包含 n^2 个非零元素的非稀疏矩阵，构建或分解该矩阵的复杂度为 $O(n^3)$，计算开销十分巨大。

由于我们的算法需要先构建矩阵 S，再对其进行分解，但是由于 S 过于稠密，造成了很大的计算开销。为了提高计算效率，我们试图构建一个稀疏化的矩阵，再对这个稀疏的矩阵进行分解。

我们借鉴了论文《Efficient Sampling for Gaussian Graphical Models via Spectral Sparsification》中的思想，在 O(mlog^n) 阶的时间内对矩阵进行稀疏化处理，得到只包含 O(nlog^n) 个非零元素的稀疏矩阵。当 $\alpha_1 = \cdots = \alpha_T = \frac{1}{T}$ 时，S 可以用矩阵 $\tilde{L}$ 表示。

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$

where $\sum_{r=1}^{T} \alpha_r = 1$ and α_r non-negative

One can construct a $(1+\epsilon)$-spectral sparsifier $\tilde{\boldsymbol{L}}$ with $O(n \log n \epsilon^{-2})$ non-zeros

in time $O(T^2 m \epsilon^{-2} \log n)$ for undirected graphs

$$\alpha_1 = \cdots = \alpha_T = \frac{1}{T} \quad \Rightarrow \quad \begin{aligned} \boldsymbol{S} &= \log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r\right)\boldsymbol{D}^{-1}\right) \\ &= \log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{L})\boldsymbol{D}^{-1}\right) \\ &\approx \log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\tilde{\boldsymbol{L}})\boldsymbol{D}^{-1}\right) \end{aligned}$$

图 10：对矩阵 S 进行稀疏化

根据以上分析，我们可以构建一个稀疏化版的 S，对该矩阵进行稀疏矩阵分解，从而得到每个节点的表示向量。

可以证明，稀疏化矩阵与原始矩阵的误差存在理论上界：

$$trunc\_\log^{\circ}\left(\frac{vol(\mathrm{G})}{b}\tilde{M}\right) - trunc\_\log^{\circ}\left(\frac{vol(\mathrm{G})}{b}\tilde{M}\right)_F \le \frac{4\grave{o}\, vol(G)}{b\sqrt{d_{min}}}\sqrt{\sum_{i=1}^{n}\frac{1}{d_i}}$$

实验结果表明：（1）使用稀疏化矩阵分解得到的模型性能与显式的稠密矩阵分解结果差距不大，它们的性能都比隐式矩阵分解（如 Deepwalk、LINE）的更好。（2）稀疏矩阵分解可以处理亿级网络（包含数亿节点，数十亿条边的网络）。

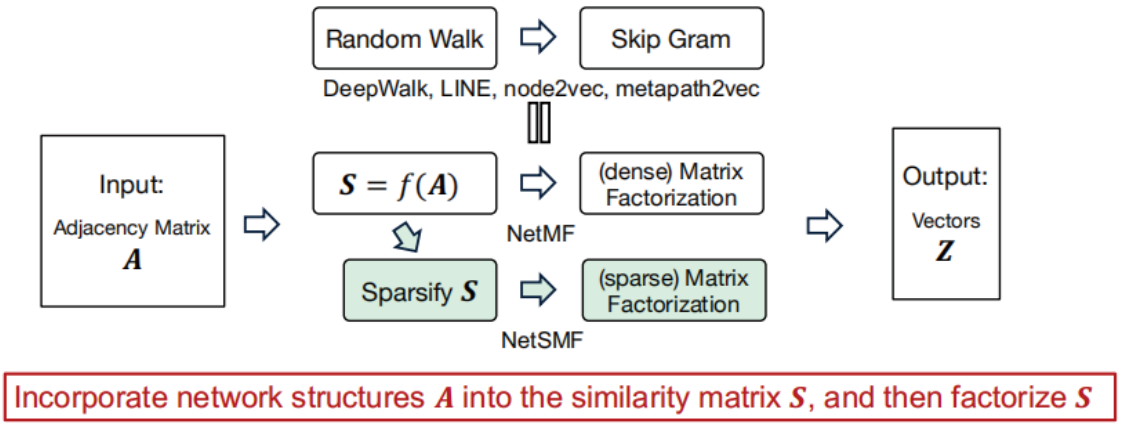


图 11：矩阵稀疏化及其分解

给定网络结构的邻接矩阵 A，我们通过某种操作把关心的结构属性融合到了矩阵 S 当中，从而对矩阵 S 进行分解。

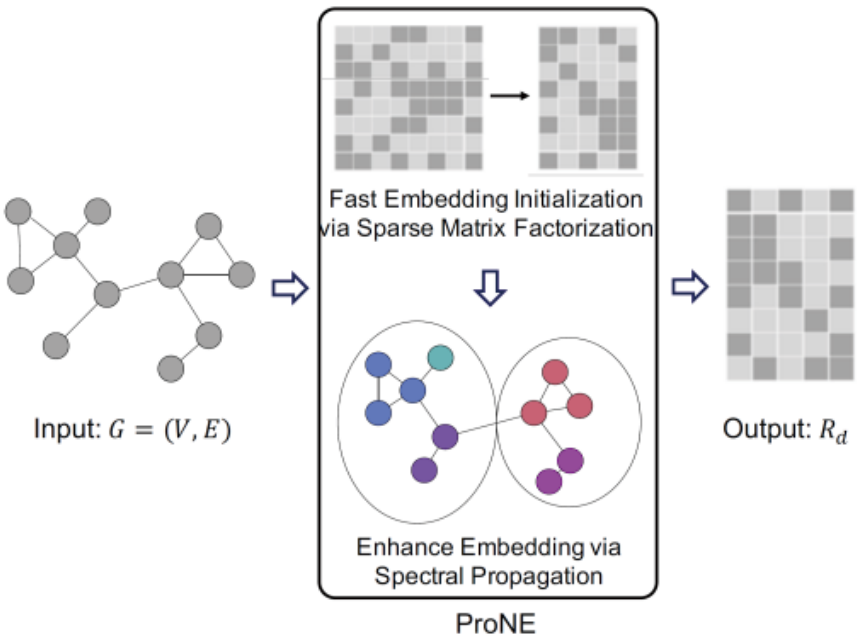


图 12：基于网络嵌入的信息传播 ProNE

反过来，给定一个输入网络（一般情况下是稀疏的），我们对其进行稀疏矩阵分解可以得到一个初始化的节点向量表示。接着，我们通过在谱空间上对前面得到的向量表示进行传播操作得到每个节点更好的向量表示。

$$R_d \leftarrow D^{-1}A(I_n - \tilde{L})\,R_d$$

The idea of **Graph Neural Networks**

$D^{-1}A(I_n - \tilde{L})$ is $D^{-1}A$ modulated by the filter in the spectrum

$\tilde{L} = Ug(\Lambda)U^T$ is the spectral filter of $L = I_n - D^{-1}A$

图 13：谱传播（spectral Propagation）

在通过稀疏分解得到向量表示之后，在谱空间进行传播操作，得到更新后的向量表示，这与一些图神经网络的思想也不谋而合。

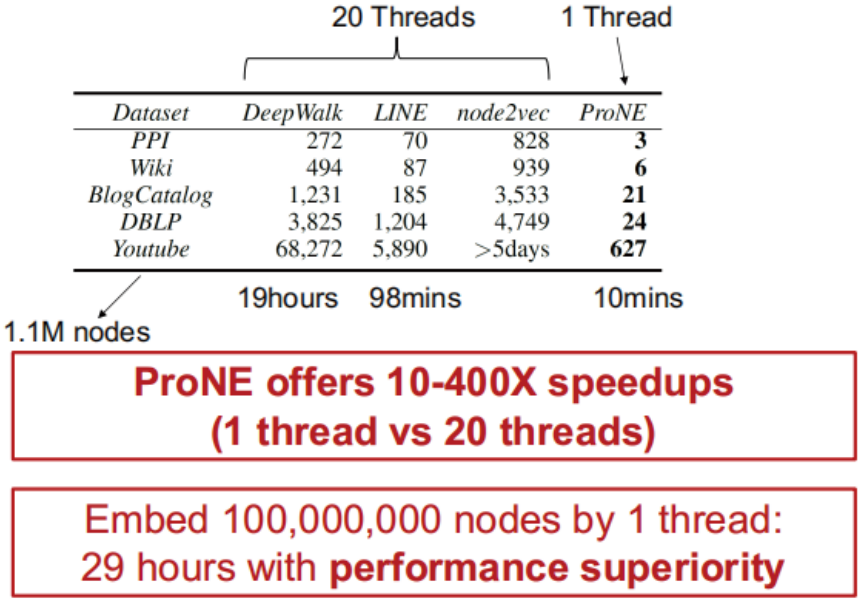


图 14：ProNE 的性能提升

ProNE 带来了两方面的性能提升：(1) 只需要使用一个 CPU 线程，就可以获得比使用 20 个线程快 10–400 倍的运算速度。(2) 在 29 小时内可以学习到亿级网络的表示，并且将该表示用于下游任务时的效果要优于使用传统的 DeepWalk、LINE 等模型得到的表示所得到的结果。

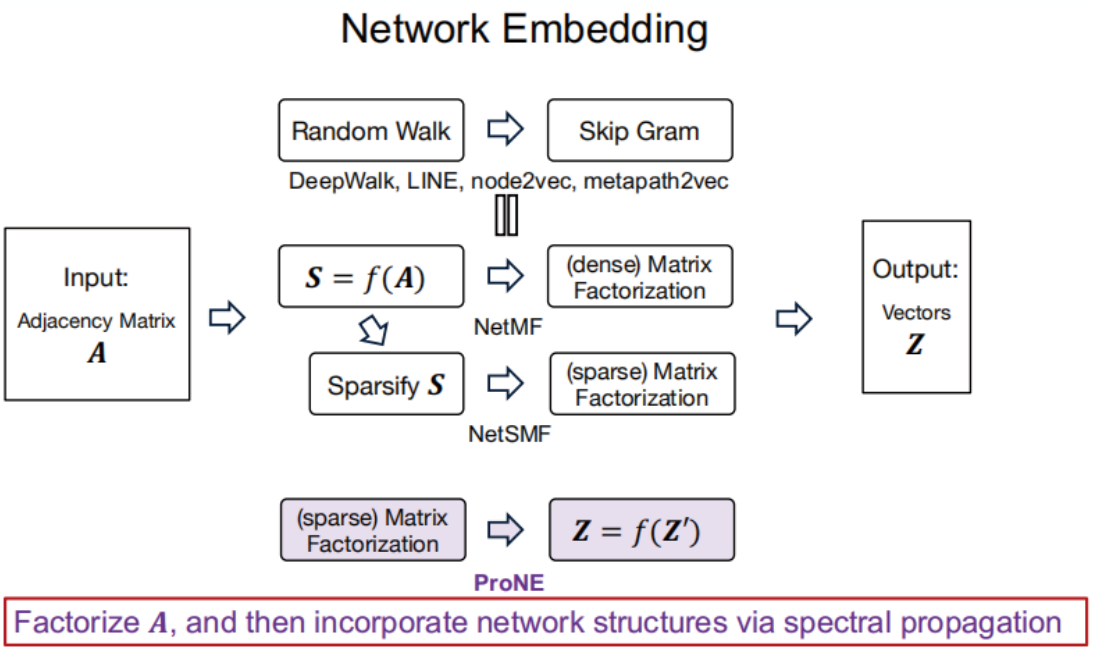


图 15：网络嵌入学习之 ProNE

在 ProNE 中，我们反过来先进行稀疏矩阵分解，再通过传播的思想把网络结构属性融入稀疏矩阵分解学习到的向量表示当中，从而得到最终更新后的节点。

三、图神经网络

图神经网络中也体现了传播（即邻居聚合）的思想。对于网络中的每一个节点，我们对其邻居节点上的信息或信号进行传播与聚合。例如，对于节点 V，我们可以向它传递周围所有邻居节点的信息，该节点通过某种方式选择并整合邻居的信息，从而更新自身的信息，这一过程可以不断迭代。同时，V 的信息也可以传递给其邻居节点。

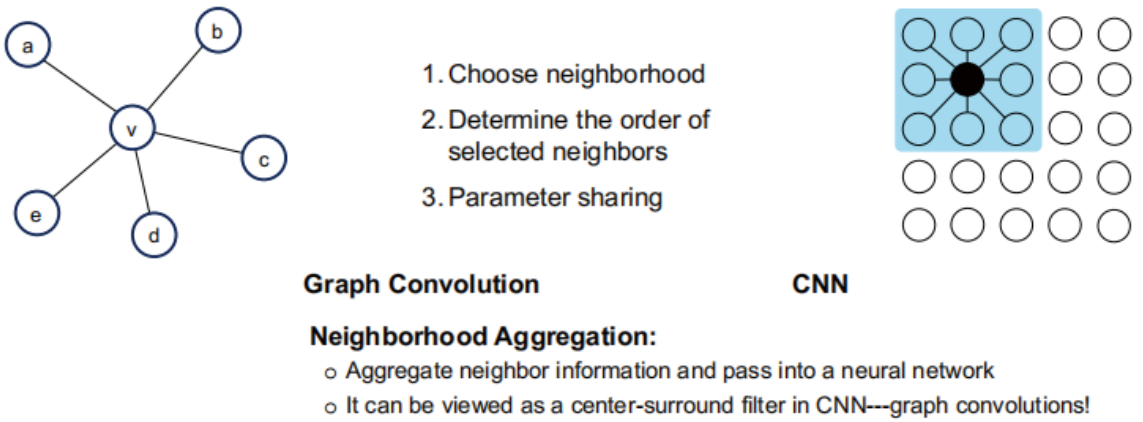


图 16：图神经网络

这种图神经网络的思想来源于传统的卷积神经网络（CNN）。在 CNN 中，卷积核包含三部分操作：（1）定义一个邻域的大小。在如图 16 所示的例子中，卷积核定义了包含九个像素的邻域。（2）更重要的是，CNN 中的卷积核隐式地定义了周围每一个像素的顺序。在欧式空间下的网格数据结构中，无论我们怎么平移卷积核，卷积核内部点的相对位置永远不会变化，卷积核隐式地记录了某像素周围所有像素的顺序。（3）这种平移不变性使卷积核能够进行参数共享，我们只需要使用为数不多的参数就可以定义卷积核。当我们将卷积神经网络的思想迁移到图神经网络中时，我们首先定义网络结构的邻域（节点的邻居），接着定义邻居的顺序，最后进行参数分享。

图卷积网络是当下最流行的图神经网络，邻居的聚合函数可以表示为：

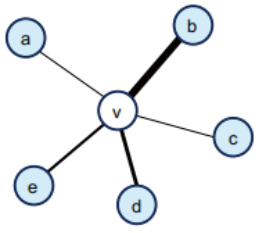
$$h_v^k = \sigma(W^k \sum_{u \in N(v) \cup v} \frac{h_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

其中，h_v^k 是节点 v 在第 k 层的节点表示，我们通过通过聚合节点 v 周围邻居节点的信息获得其节点表示。接着，我们通过参数 W^k 对邻居节点的信息聚合结果进行特征变换。最终，将上述特征变换结果输入给非线性激活函数，用该函数的输出更新节点 v 的表示。

在矩阵形式下，GCN 的聚合函数为：

$$H^k = \sigma(\hat{A}H^{(k-1)}W^{(k)})$$

请注意，邻居节点的信息是通过归一化的拉普拉斯算子求得的，并没有对每个邻居节点的重要性进行具体的参数化，即不会根据数据调整每个邻居节点的重要性。



GCN

$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}^k \sum_{u \in N(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

Aggregate info from neighborhood via the normalized Laplacian matrix

Graph Attention

$$\boldsymbol{h}_v^k = \sigma(\sum_{u \in N(v) \cup v} \alpha_{v,u} \boldsymbol{W}^k \boldsymbol{h}_u^{k-1})$$

Aggregate info from neighborhood via the learned attention

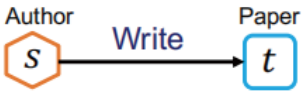
图 17：图注意力网络

为了在聚合信息时，从数据中学习到每个邻居节点对当前节点影响的重要性，研究人员基于注意力机制提出了图注意力网络（GAT）。在 GAT 的原始论文中，作者使用的注意力机制定义如下：

$$\acute{a}_{v,\mu}=\frac{\exp\left(\text{LeakyRuLU}\left(a^{\top}\left[Qh_{v},Qh_{u}\right]\right)\right)}{\sum_{u'\in N(v)\bigcup\{v\}}\exp\left(\text{LeakyRuLU}\left(a^{\top}\left[Qh_{v},Qh_{u'}\right]\right)\right)}$$

实际上，我们可以根据需求自己设计注意力机制。

为了在异构网络中定义注意力机制，我们提出了 HGT 模型，该模型用特定的参数设置异构网络中不同类型的关系。例如，在学术网络中，作者 s 写了一篇论文 t。我们首先找出 s 与 t 之间的元关系 e=(s,t)，该三元组可以表示为 $\tau(s),\phi(e),\tau(t)$。因此，对于异构网络中任意的边，我们都能以三元组形式表示这种关系。HGT 的核心思想为：基于元关系对每条边进行参数化定义。



- meta relation of an edge $e=(s,t)$

$$\langle\tau(s),\phi(e),\tau(t)\rangle$$

- heterogeneous mutual attention

$$\textbf{Attention}_{HGT}(s,e,t)=\underset{\forall s\in N(t)}{\text{Softmax}}\left(\underset{i\in[1,h]}{\Vert}ATT\text{-}head^{i}(s,e,t)\right) \quad (3)$$

$$ATT\text{-}head^{i}(s,e,t)=\left(K^{i}(s)\,W^{ATT}_{\phi(e)}\,Q^{i}(t)^{T}\right)\cdot\frac{\mu_{\langle\tau(s),\phi(e),\tau(t)\rangle}}{\sqrt{d}}$$

$$K^{i}(s)=\text{K-Linear}^{i}_{\tau(s)}\left(H^{(l-1)}[s]\right)$$

$$Q^{i}(t)=\text{Q-Linear}^{i}_{\tau(t)}\left(H^{(l-1)}[t]\right)$$

图 18：HGT 中边关系的参数化定义

具体而言，为了表示 s 和 t 之间的关系，我们在如图 18 所示的公式 (3) 中使用了多头注意力。对于每一个注意力头而言，其注意力与 s 和 t 之间的边的关系相关。我们需要针对 S 和 T 之间的具体关系定义 W。应用于 s 和 t 的生成查询向量的操作 Q 和生成键向量的操作 K 也需要根据其的节点类型τ (s)，τ (t) 的函数单独求得。对于异构网络中任何类型的关系，HGT 会针对它定制一套独特的参数，从而捕捉不同网络当中不同类型节点和边的独特属性。

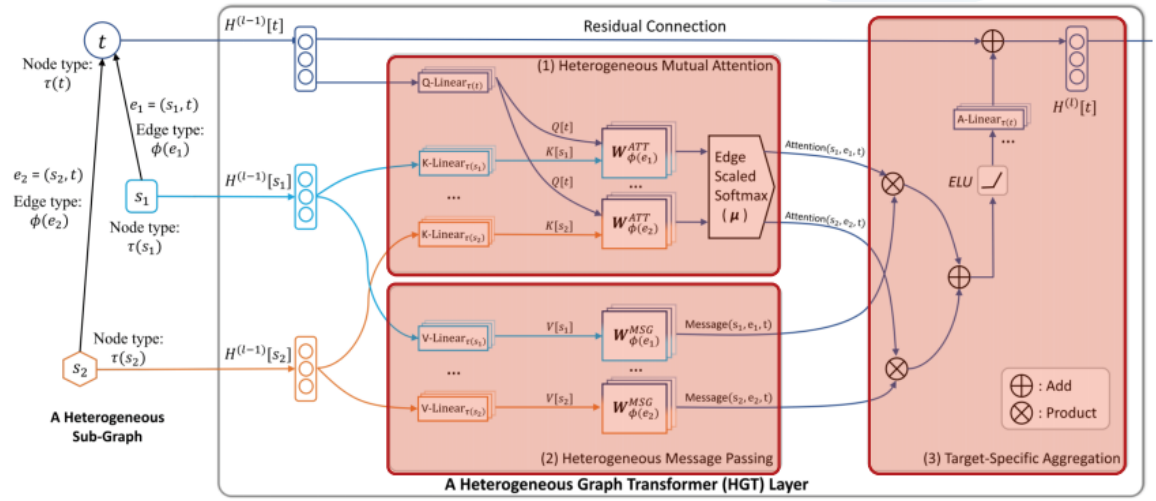


图 19：HGT 架构示意图

HGT 的模型架构如图 19 所示。最左侧为异构子图的输入网络，得到图的初始化嵌入向量后，我们首先针对网络中每条边两端的节点计算它们的异构注意力。同时，我们还需要根据异构网络的元关系计算任意一条边传播的信息。在计算出异构注意力和消息之后，我们根据目标节点的类型对信息进行聚合，最终得到本层中目标节点更新后的节点表示。

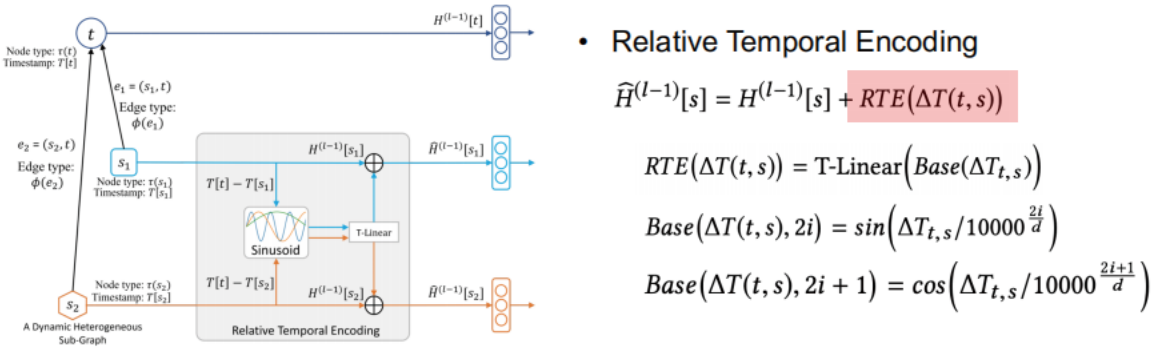


图 20：HGT 的相对时间编码

除了异构性的挑战，现实中的网络也包含时间信息。为了对时间信息进行建模，我们收到 BERT 中位置编码的启发，提出了一种名为“相对时间编码”方法。如果我们想计算出某节点内在之前某个时间点上的表示，那么我们就将这个节点之前该时刻与当前时刻的相对编码与当前的节点表示相加，从而得到二十年前的节点表示。

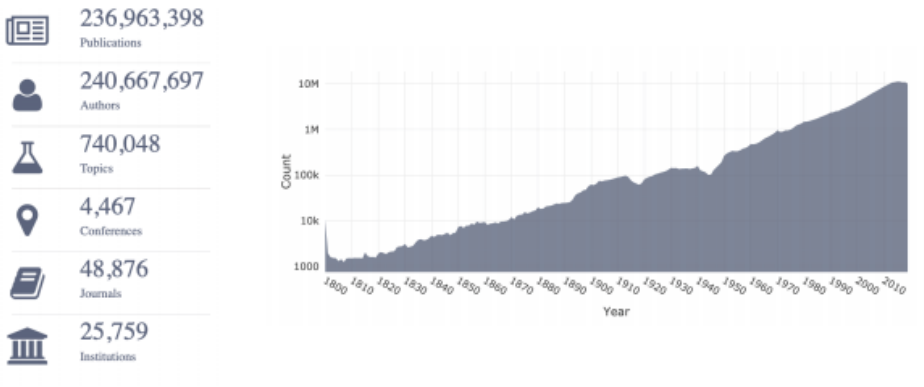


图 21：HGT 在微软学术网络上的实验设定

我们以微软学术网络作为输入开展了一系列实验，将该网络到 HGT 模型中，学习出每一种节点的节点表示，进而利用得到的节点表示完成下游的任务。

GNN Models		GCN [7]	RGCN [12]	GAT [21]	HetGNN [25]	HAN [22]	$HGT_{noHeter}$	HGT_{noTime}	HGT
# of Parameters		1.69M	8.80M	1.69M	8.41M	9.45M	3.12M	7.44M	8.20M
Paper–Field (L1)	NDCG	.558±.141	.563±.128	.601±.103	.615±.084	.617±.096	.674±.086	.702±.089	**.735±.084**
	MRR	.513±.136	.526±.105	.587±.096	.595±.076	.604±.092	.652±.078	.676±.082	**.713±.081**
Paper–Field (L2)	NDCG	.241±.074	.258±.046	.276±.049	.271±.062	.281±.051	.301±.046	.307±.052	**.332±.048**
	MRR	.192±.067	.206±.052	.228±.045	.231±.053	.242±.049	.257±.058	.260±.064	**.276±.071**
Paper–Venue	NDCG	.303±.066	.354±.051	.461±.057	.447±.071	.478±.062	.515±.059	.538±.064	**.551±.062**
	MRR	.114±.070	.198±.047	.244±.052	.226±.059	.269±.067	.295±.060	.322±.048	**.334±.061**
Author Disambiguation	NDCG	.730±.064	.742±.057	.785±.063	.792±.052	.810±.049	.834±.058	.849±.066	**.857±.054**
	MRR	.762±.042	.786±.048	.843±.044	.852±.058	.876±.056	.903±.041	.911±.043	**.918±.048**

HGT offers ~9–21% improvements over existing (heterogeneous) GNNs

图 22：HGT 在微软学术网络上的实验结果

如图所示，实验结果表明通过对异构网络中不同边设置独特的参数化注意力，相较于传统的异构图神经网络，HGT 可以获得 10 到 20 个百分点的性能提升。

Venue	Time	Top−5 Most Similar Venues
WWW	2000	SIGMOD, VLDB, NSDI, GLOBECOM, SIGIR
	2010	GLOBECOM, KDD, CIKM, SIGIR, SIGMOD
	2020	KDD, GLOBECOM, SIGIR, WSDM, SIGMOD
KDD	2000	SIGMOD, ICDE, ICDM, CIKM, VLDB
	2010	ICDE, WWW, NeurIPS, SIGMOD, ICML
	2020	NeurIPS, SIGMOD, WWW, AAAI, EMNLP
NeurIPS	2000	ICCV, ICML, ECCV, AAAI, CVPR
	2010	ICML, CVPR, ACL, KDD, AAAI
	2020	ICML, CVPR, ICLR, ICCV, ACL

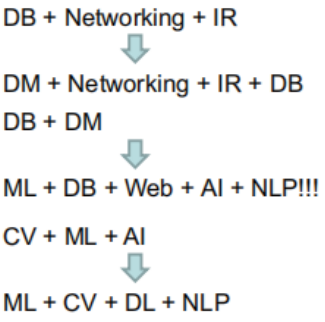


图 23：案例分析

图 23 展示了我们与 2019 年做的一项案例分析，在我们进行实验的时候并没有 2020 年的数据。我们使用了相对时间编码，从而在 2019 年就可以预测 2020 年的学术网络的情况。以 WWW 会议为例，我们预测 2020 年的 WWW 大会将与数据挖掘、计算机网络、信息检索和数据库相关，但是二十年前的 WWW 2000 与数据库、网络、信息检索的相关更大。同样，对于数据挖掘顶会 KDD 而言，在 2019 年的案例分析中，我们认为它在 2000 年时数据库和数据挖掘更相关；而到了 2020 年，我们认为 KDD 2020 与机器学习、数据库、web、人工智能、自然语言处理更相关。对于 NeurIPS 来说，我们也发现它在过去二十年中的演化，我们预测 NeurIPS 2020 除了与机器学习相关，还与计算机视觉和自然语言处理，以及深度学习有关。

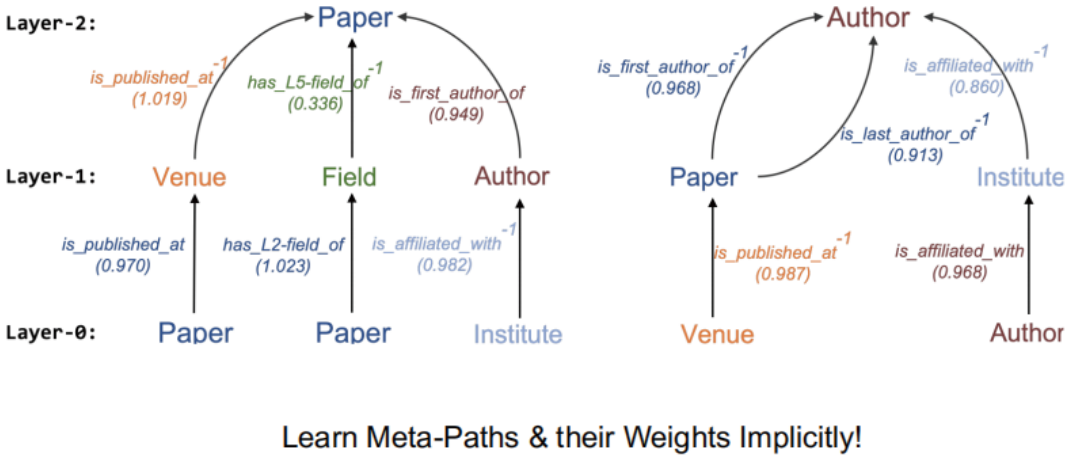


图 24：HGT 的重要意义

在我看来，HGT 模型最大的贡献并不是仅仅是性能上的提升，最重要是，它提供了解决异构网络表示学习（或异构网络挖掘）的方式。在异构网络挖掘和异构表示学习领域中，通常情况下，我们需要对某一个输入的网络有足够的了解。遵循孙怡舟老师提出的范式，当我们对网络有了一定了解之后，针对其设计独特的元路径（Meta-Paths），从而进行后期的数据挖掘和表示学习。我们经常考虑如何设计元路径，以及哪种元路径是更重要的。HGT 模型遵循了图神经网络的基础架构，对于任何一个中心节点，每当我们堆叠一层图神经网络，我们就可以得到该节点与图结构中一跳之外的节点的关系。更重要的是，对于每一跳的过程，我们可以学习到相邻两个节点类型的权重。通过堆叠多层的图神经网络，我们能够自动地学习出元路径。对于每一种元路径，我们还可以

自动求出它的位置。基于图神经网络的异构图表示学习的优势在于：沿着这个研究方向，如果我们未来面临其它输入的异构网络，并不需要对该网络有很深的了解，也不需要人工定义元路径及权重，可以用 HGT 或者其它的图神经网络学习数据中隐含的元路径及其重要性。

四、图神经网络的预训练

对图神经网络的预训练受到了自然语言处理和计算机视觉领域预训练研究的启发。在自然语言处理领域，以 BERT 为例，其预训练包含两个模块：(1) 捕捉数据中信息的强大模型 (2) 预训练任务。

一般情况下，我们希望预训练任务是无监督的 (不需要标注数据)。BERT 从数据本身出发设计了两个预训练的学习任务：(1) 掩码语言建模——masked language modeling；(2) 下一句预测——next sentence prediction。

在图神经网络预训练中，我们考虑两种设定：(1) 给定一个输入网络，我们基于该网络预训练一个模型。在得到预训练的模型之后，我们可以利用预训练的模型，帮助我们完成相同网络数据上的下游任务 (节点分类、链接预测等)，即进行调优 (fine-tune)。在这种情况下，我们在相同的图数据上进行预训练和调优。为了做到这一点，我们仍然需要考虑两方面的问题：(1) 我们应该选用什么样的模型？是否可以默认选用任意的图神经网络？(2) 如何设计自监督的预训练任务？

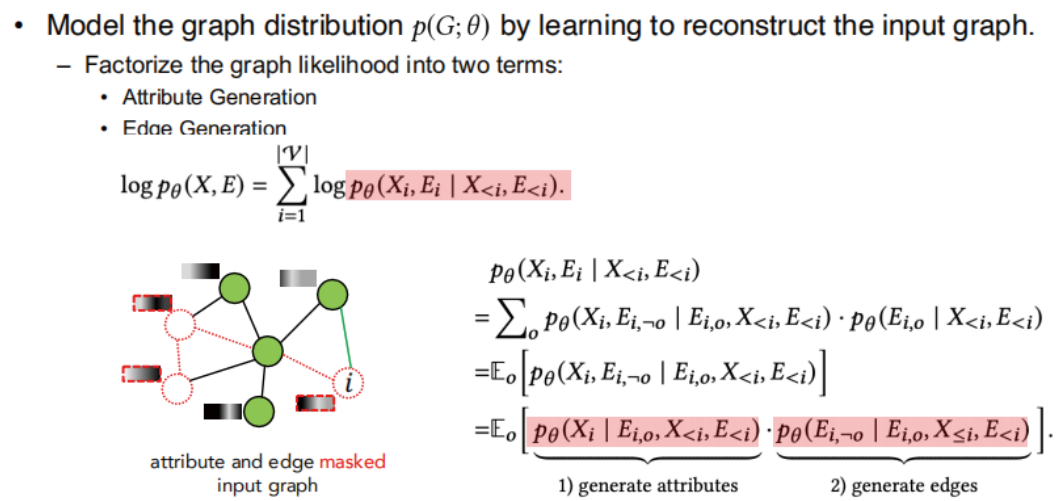


图 25：GPT-GNN 中的预训练任务

在 GPT-GNN 中，我们提出使用生成式的方法进行预训练。对于任何输入的网络，我们试图根据部分观测到的网络数据生成未观测到的数据。以图 25 中左下角的图为例，给定一个输入的网络，我们人为地随机屏蔽掉一部分节点的属性和连边，最后观测到的节点为绿色的节点。

我们的预训练任务如下：(1) 根据观测到的部分生成下一个未观测节点 i 的属性。(2) 预测节点 i 与其它节点的连边。我们可以简单地独立看待 i 的节点生成和连边生成任务，即先直接根据已经观测到的属性 X 和网络结构 E 来预测节点 i 的属性，再单独预测 i 的连边。

然而，在图神经网络当中，我们基于网络结构对每个节点的属性进行传播，因此我们认为这个节点属性和网络

结构之间存在某种耦合关系。如果用上面提到的简单方式进行预训练，则于图神经网络最基本的宗旨相违背。所以，在 GPT-GNN 中，我们认为节点 i 的属性和它的连边并不是独立的。在生成 i 的属性时，我们利用 i 的部分连边（即图 25 中的绿色连边）。当生成 i 的属性之后，我们把 Xi 加入到观测到的数据当中，根据之前观测到的数据和生成的 i 的属性来生成与 i 相连的未观测到的连边（红色虚线）。

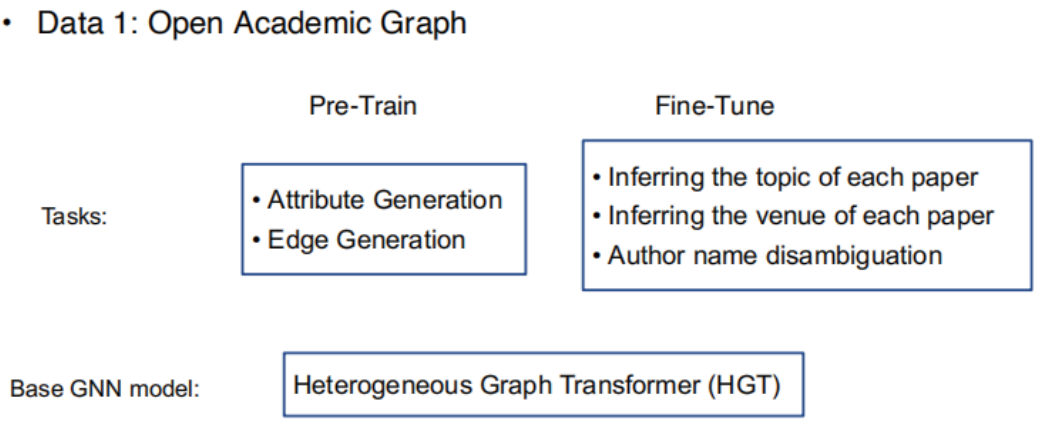
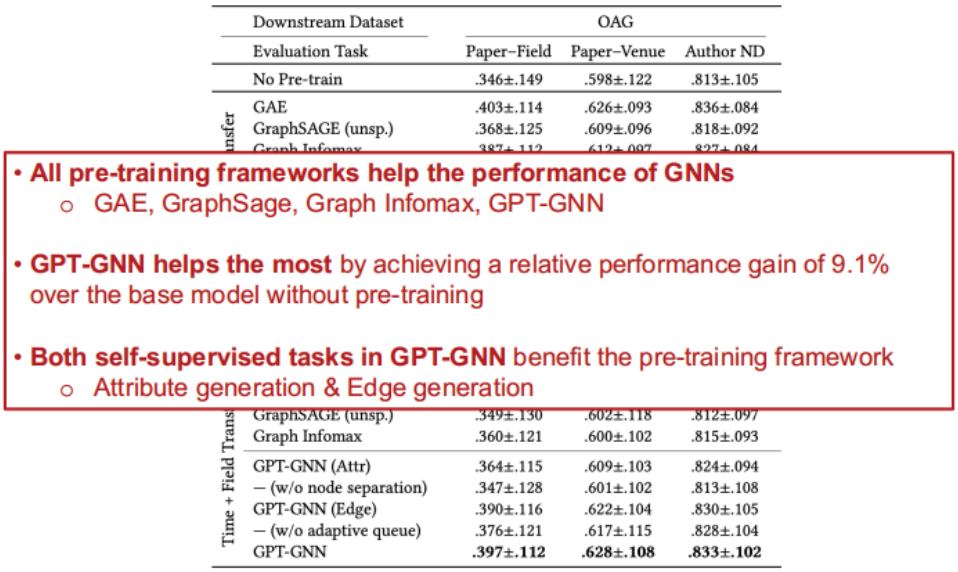


图 26：GPT-GNN 实验设定

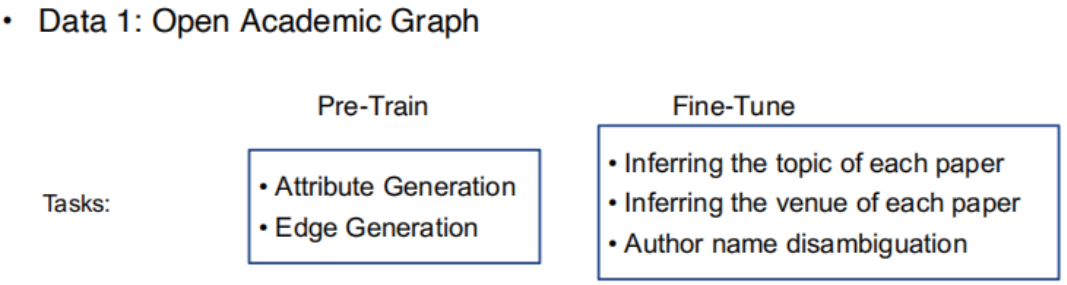
综上所述，在 GPT-GNN 中，我们设计了两个预训练任务，即属性生成和连边生成，并且在相同的图数据上进行预训练和调优。在实验中，针对输入的学术网络，我们采用 HGT 模型作为基础的 GNN 模型，考虑三种下游任务：(1) 预测每篇论文的主题；(2) 预测每篇论文发表的会议；(3) 作者姓名消歧。



	Downstream Dataset	OAG		
	Evaluation Task	Paper-Field	Paper-Venue	Author ND
	No Pre-train	.346±.149	.598±.122	.813±.105
nsfer	GAE	.403±.114	.626±.093	.836±.084
	GraphSAGE (unsp.)	.368±.125	.609±.096	.818±.092
	Graph Infomax	.387±.112	.612±.097	.827±.084
Time + Field Trans	GraphSAGE (unsp.)	.349±.130	.602±.118	.812±.097
	Graph Infomax	.360±.121	.600±.102	.815±.093
	GPT-GNN (Attr)	.364±.115	.609±.103	.824±.094
	− (w/o node separation)	.347±.128	.601±.102	.813±.108
	GPT-GNN (Edge)	.390±.116	.622±.104	.830±.105
	− (w/o adaptive queue)	.376±.121	.617±.115	.828±.104
	GPT-GNN	**.397±.112**	**.628±.108**	**.833±.102**

图 27：GPT-GNN 实验结果

我们主要观测到个重要的实验结果：(1) 所有的预训练模型都能够对图神经网络在性能上有所提升；(2) 我们提出的 GPT-GNN 模型能取得最大的性能提升；(3) 我们设计两种训练任务都能够有助于提升 GPT-GNN 模型的性能。



Model	HGT	GCN	GAT	RGCN	HAN
No Pre-train	**.346**	.327	.318	.296	.332
GPT-GNN	**.420**	.359	.382	.351	.406
Relative Gain	21.4%	9.8%	20.1%	18.9%	22.3%

图 28：更换 GPT–GNN 中基础 GNN 模型的实验结果

通过更换基础 GNN 模型，我们发现 HGT 取得了最好的性能。无论使用哪种图神经网络作为基础模型，我们提出的 GPT–GNN 预训练模型相对于不使用预训练技术时都获得了一定的性能提升。

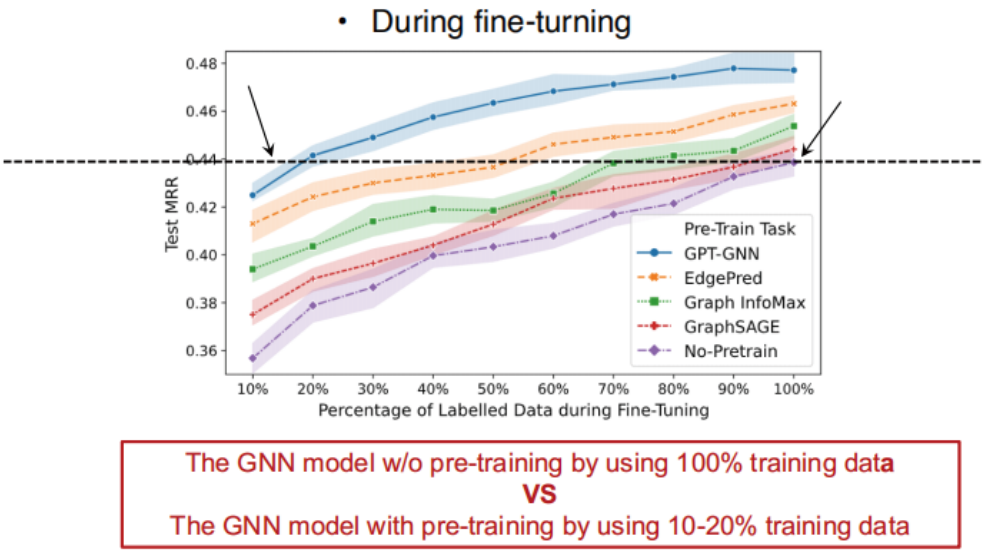


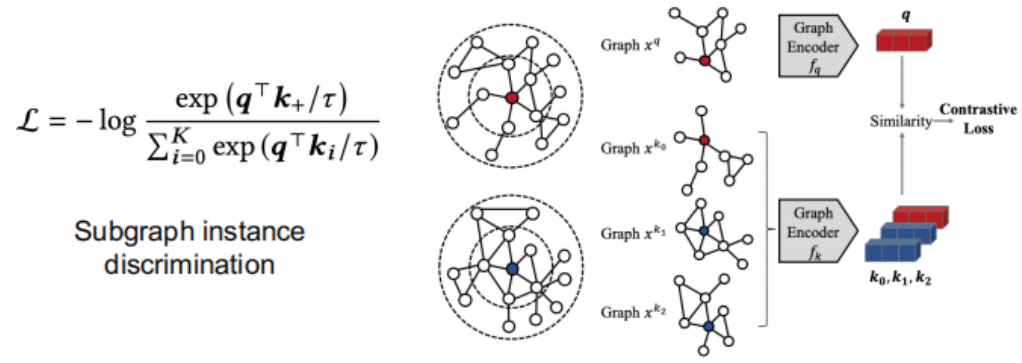
图 29：使用 / 不使用预训练的性能对比

如图 29 所示，给定预训练好的模型，我们在调优阶段使用不同比例的原始数据进行训练。紫色的折线代表没有经过预训练的性能，蓝色折线代表使用 GPT–GNN 预训练之后的性能。我们发现，在不经过预训练的情况下用所有的百分百的数据所取得的性能，与预训练之后只使用 10% 到 20% 的数据进行训练取得的性能是差不多的。实验结果说明预训练能够在数据比较少的情况下提升图表示学习及下游任务的性能。

在 GPT–GNN 的工作中，我们在相同的图上进行预训练和调优。但是生活中有很多不同的图数据和网络结构，

我们希望利用各种各样的网络数据预训练一个图表示模型，使该图表示模型（或图神经网络）可以学习到所有不同图数据中普适性的特征和模式。从而帮助我们通过调优，在之前预训练时未见过的图上更好地完成下游任务。此时，在跨多个网络的情况下，设计恰当的预训练仍然是最重要的问题。

我们借鉴了对比学习（contrasive learning）的思想，将实例区分（instance discrimination）作为预训练任务，判断两个实例是否相似，从而为每个实例生成一个表示。为了将对比学习思想应用到图上的预训练中，我们需要思考三个问题：（1）如何定义图中的实例？（2）如何度量不同实例之间的相似度？（3）如何学习到每个实例的表示？



1.Jiezhong Qiu et al. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. **KDD** 2020.

图 30：图对比编码

为此，我们提出了图对比编码算法（GCC）。为了定义图中的实例，我们以每个节点为中心随机游走形成一个子图，并且将该子图作为一个实例。为了定义两个子图实例之间的相似度，我们认为从同一个节点出发进行随机游走生成的子图是相似的，从不同节点出发进行随机游走生成的子图是不相似的。最后，为了将该子图实例映射到表示空间中，我们在这里也采用了当下最流行的图神经网络。

Dataset	Academia	DBLP (SNAP)	DBLP (NetRep)	IMDB	Facebook	LiveJournal
$\|V\|$	137,969	317,080	540,486	896,305	3,097,165	4,843,953
$\|E\|$	739,384	2,099,732	30,491,458	7,564,894	47,334,788	85,691,368

图 31：GCC 实验设定

在实验中，我们使用了六个数据集进行预训练。而在调优阶段，我们使用与预训练时不同的数据，在不同的任务中进行验证，比如在 US-Airport 数据集上进行节点分类，在 COLLAB、RDT-B 上进行图分类，在 Aminer 学术网络上进行节点相似度搜索。

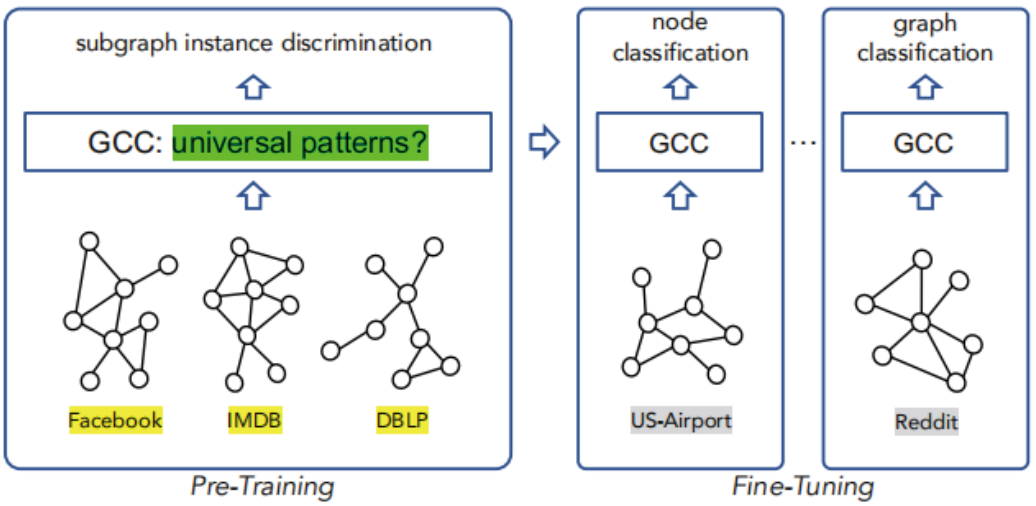


图 32：GCC 实验结果

通过上述实验，我们验证了： 通过对多个图进行表示学习可以学习到图内在的普适性结构化模式和特征，这种特征有助于在其它为观测到的图上的表示学习任务和具体的下游任务。

在介绍完网络嵌入、矩阵分解、图神经网络、图神经网络的预训练中等基于图的研究之后，接下来我们将介绍这类问题使用的一些常见数据集。

OGB 是由斯坦福大学发布的开放的图数据对比基准测试平台（类似于计算机视觉领域的 ImageNet、自然语言处理领域的 SQuAD、GLUE）。OGB 包含十个左右的数据子集，并且针对每个数据子集维护了一个算法性能排行榜，大家可以将自己设计的新型图表示学习算法、基于图的机器学习算法提交到该平台上，用 OGB 平台上的公开数据及测试算法的而性能。

我们微软学术也公开了微软学术图谱，该图谱包含两亿多篇文章，两亿多个作者，以及成千上万的其它不同类型的实体。更重要的是，这些数据具有时间信息，覆盖了我们学术界过去两百年发表的文章。我们还和 Aminer 合作，共同发布了开放学术图谱 OAG。

阿里巴巴达摩院杨红霞：人工智能从感知走向认知——认知推荐

整理：智源社区 熊宇轩

对于阿里巴巴等电商巨头来说，通过认知推荐准确识别用户的意图，为用户生成个性化的推荐结果是当前推荐系统的研究重点，可以为企业带来巨大的收益。在本届智源大会上，阿里巴巴达摩院智能计算实验室资深算法专家杨红霞博士带来了题为「人工智能从感知走向认知——认知推荐」的主题报告。基于大量实例以及多篇高水平论文，从基础数据层、推理引擎层、用户交互层三个层面上深入剖析了阿里巴巴在认知推荐领域的研究进展，揭秘了阿里巴巴强大的推荐系统背后的技术原理。

以下为演讲内容:

在今天的演讲中，我想介绍一下我们团队在认知推荐方面做的一些工作。今天，推荐系统已经覆盖到了许多方面，但仍然面临着诸多挑战。

一、阿里巴巴认知计算平台

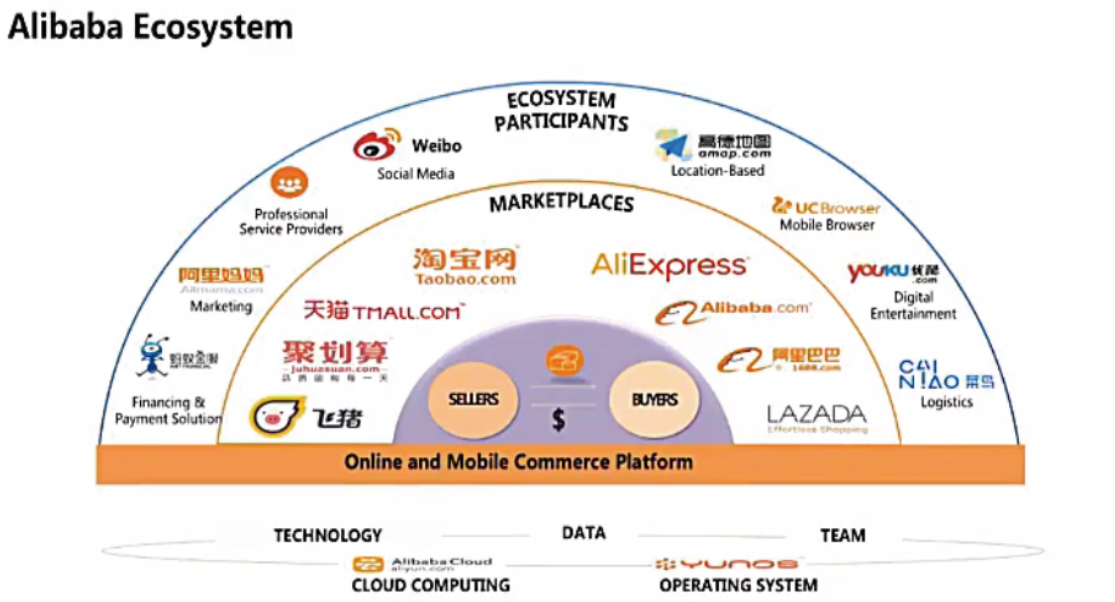


图 1：阿里巴巴的生态系统

图 1 展示了阿里巴巴的数据生态系统。大家在购物时，或多或少都会使用到阿里巴巴的一系列产品（如淘宝、天猫、聚划算）。在阿里巴巴的诸多电商平台中，淘宝是面向消费者用户（2C）的平台，天猫是面向商家（2B）的平台，聚划算是类似于「Groupon」的团购平台，飞猪的主要业务是旅行，AliExpress 针对的是欧洲市场，Alibaba.com 针对的是美国市场，LAZADA 是东南亚最大的电商平台……上述产品构成了阿里巴巴覆盖全球 200 多个国家的电商平台网络。

我们的团队目前正负责手机淘宝和天猫，这也许是全球每天流量最大的电商推荐系统。除了电商平台（marketplace）之外，整个阿里巴巴集团旗下丰富的生态系统实际上还包括其它的组成部分（如新浪微博、高德地图、蚂蚁金服、UC 浏览器、优酷娱乐、材料物流、阿里妈妈等），它们也有助于我们更好地了解消费者、服务消费者。我们可以利用以上各方面的数据全面地了解一个消费者。当然，在具体使用这些数据的过程中仍然存在很多挑战。举例而言，我们需要考虑数据对齐的问题（例如，如何将用户在优酷上表达的兴趣与电商相联系）。

图 2：阿里巴巴大数据的类型

图 2 从另外一个角度对阿里巴巴拥有的大数据进行了细分。具体而言，阿里巴巴的大数据体系包括如上文中介绍的电商数据，市场数据、娱乐数据、健康与幸福感（2H）数据等。上述数据构成了阿里巴巴丰富的大数据体系，也有助于我们的推荐系统更加整体地了解消费者。

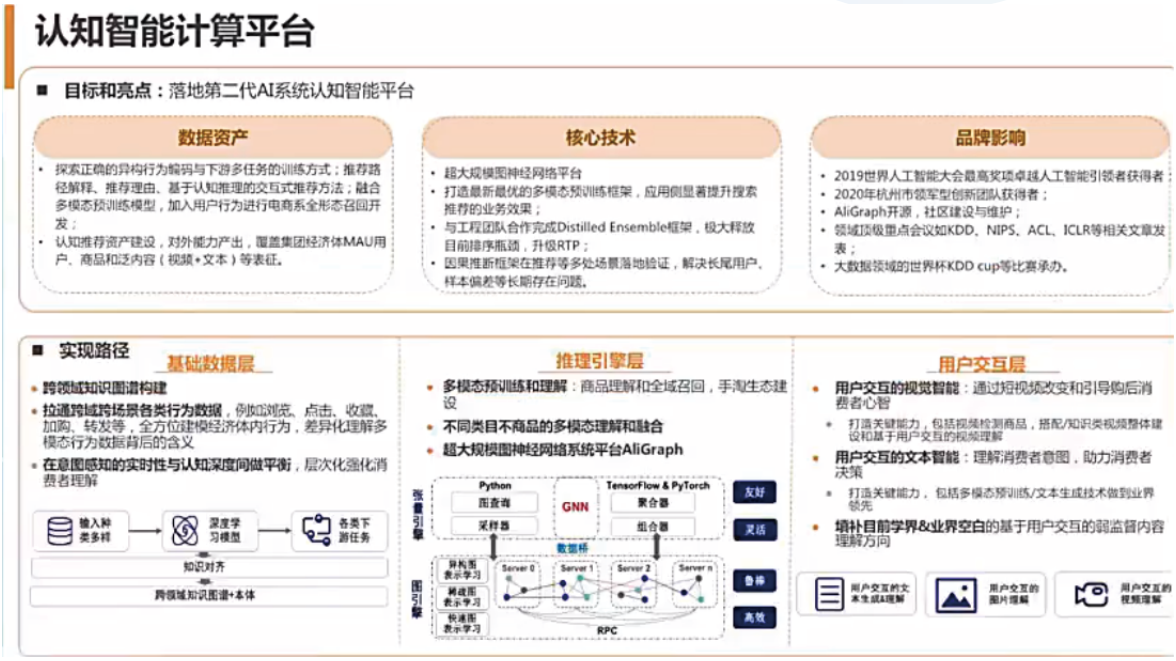


图 3：阿里巴巴认知智能计算平台

下面，我将简单介绍认知推荐系统的相关概念，以及我们对这一课题的理解。在考虑深度学习的千人千面之前，我们可以把整个手机淘宝当做一个大的超级市场，这个市场中有琳琅满目的商品。而在考虑千人千面之后，我们可以认为每个人的超级市场有一个个性化的货架，这个货架上上面摆放着该用户曾经浏览或买过的商品，或者跟该用户一样的人群喜欢的商品。由于当前大多数的推荐系统是各种深度模型的延展，我们很难跳出这种框架针对每个用户接下来的需求进行个性化的推理和认知，这也是深度学习模型普遍面临的挑战（即深度学习黑盒模型无法进行归纳推理）。

我们的团队希望能真正推理出消费者接下来要做什么，而不单单是对已经发生的行为进行拟合。这个任务面临着许多挑战，例如：（1）模型本身是否有足够强大的归纳推理能力；（2）如何与消费者进行有效的沟通与表达。对大多数人来说，推荐系统如果仅仅局限于推荐商品，但实际上推荐系统的业务范畴已经远远超过推荐商品了。

今天，每当我们打开手机淘宝，在我们看到的推荐系统的内容中，除了商品之外有各种模态实体的物料（例如，短视频、直播、达人的图文经验分享、基于用户的兴趣点自动生成的不同主题的物料）。如今，推荐系统每天都会处理包含高达几百亿节点、超万亿条边的网络。除了商品图之外，推荐系统还需要处理短视频图，以及各种各样其它的物料图。面对如此复杂的数据集，我们需要思考如何有效提高召回率，进行全页面优化（whole page optimization），以及对下一步行为进行推理（next step inference）。我们还应该考虑如何与消费者进行沟通与交互，从而使消费者接受这种推荐。此外，如今手机淘宝上有量级是为几十亿的商品与短视频，我们需要思考如何曝光应该曝光的商品。

不可避免地，手机淘宝上一定存在大量的符合长尾分布的商品。但是众所周知，在基于采样的深度学习领域中，公平性（fairness）如今是一个非常火热的话题。尤其是对于大数据集而言，公平性问题尤为重要，如果存在严重的选择和曝光偏置，可能大量的流量只集中于头部的商家和商品，如何曝光几十亿的商品是一个非常棘手的问题。

当然，并非所有的长尾的商品都需要曝光。根据我们的理解，公平性意味着把流量给那些高效的商品。但是，对于很多处于长尾处的商品（尤其是在冷启动的时），如果我们没有给它赋予流量并对其进行曝光，我们也很难判断它们是否会成为很受欢迎的商品。在没有任何「用户 – 物料」的序列化交互数据时，如何公平地将流量分配给这些商品是非常值得思考的问题。

此外，当我们推荐出了这某些物料之后，如何与消费者进行沟通也是需要研究的重要课题。仅仅将物料推荐出来还是不够的，我们能不能自动生成一些推荐理由，或者对内容进行搭配呢?

因此，真正在超大规模系统的复杂的环境下，进行认知推荐以及与消费者的沟通和互动，对于技术层面和平台层面来说都是前所未有的重大挑战。经过两年多的探索，我们团队也正在推进一些解决方案，其在线效果非常显著。

我们的实现路径分为三个层次。第一个层次是基础数据层，我们拥有十分丰富的数据，但是如果仅仅基于在电商平台收集到的数据来训练模型，也很难推理出消费者接下来的意图。

如今，深度学习技术仍然停留在观测周围环境（感知）的初级阶段。正如人需要经历小学、中学、大学的教育过程一样，除了观察世界，人还需要接受课程教育，不断的学习。因此，我们必须构造一个专家系统，沉淀知识和常识，构建跨领域的知识图谱。这种跨领域的知识图谱相当于一个庞大的专家领域，除了积累观测数据之外，它可以使深度学习变得更鲁棒、更高效。如果缺失了这种跨领域知识图谱，我们非常容易落到推荐系统的马太效应当中，即系统只推荐我们曾经观察过的物料，推荐的范围就会越来越窄。对于消费者、商家、以及整个平台的生态系统来说，构建跨领域的知识图谱是十分重要的。

第二个层面是推理引擎层。我们在第一个层面上已经收集到了大量包含「用户 – 物料」交互的数据。这里的物料不仅包括商品，还包括多模态的数据（如短视频）。众所周知，短视频直播的出现极大地改变了消费者的购物习惯，我们的认知智能计算平台需要能针对这些多模态数据进行推理认知和搭配。也就是说，我们的模型必须能够进行归纳推理。

我们的团队针对各种模型做了非常多的探索和研究，并进行了落地的实践。事实证明，基于图神经网络的模型可以较好地进行归纳推理。我们建立了世界范围内首个超大规模的企业级图神经网络开源平台，该平台并非停留于学术研究层面，在实现过程中涉及许多分布式计算的技术，从而使用户可以在超大规模的数据集上验证算法。

针对文本、短视频、商品、用户序列等多模态数据，如何在同样的维度上进行价值统一的召回、排序、推荐、打分是值得深入研究的课题。例如，短视频的点击率和商品的点击率差距是十分巨大的。在这种情况下，为了更好地进行推理和认知，进行多模态的预训练也是非常重要的课题。我们必须把包含短视频、商品在内的所有的物料表征在相同的嵌入空间上，才能进行更好的召回和排序。

第三个层次是用户的交互层。当我们对需要推荐的物料完成排序之后，还需要与用户进行高效、友好的交互与沟通。如果我们只是把物料排列在推荐系统的页面上，也许不能为消费者带来非常良好的感知体验。针对以上问题，我们的团队正在探索两个学术上还鲜为人所研究的方向：（1）基于用户交互的视觉智能。在这里，视觉智能包括对图像和视频的理解。近年来，研究人员在对于图像和视频的理解方面开展了大量的工作，但这些工作

本质上仍然是检索与召回。但是在我们的场景下，我们的目的并不是进行召回。我们想通过将视频融入到推荐系统中，从而引导消费者做出决策，即有用户交互和反馈的视觉智能。(2) 基于用户交互的文本智能。我们试图通过生成某段推荐理由，从而帮助消费者做出决策。例如，在推荐某个电器产品的过程中，如果我们要为某位家庭主妇生成一段推荐理由，我要生成的推荐理由需要与她的背景相关，比如说性价比比较高；如果我们要向一名 IT 工作者生成对于该电器的推荐理由，他可能更关心这个电器产品的品质如何。由此可见，文本智能一定需要考虑与的用户交互。

然而很不幸的是，目前绝大多数该领域中的工作仍然是在做检索或基于相似性的工作。例如，在自然语言生成 (NLG) 领域中，诸如 BLEU 等度量标准旨在评价生成的文本与原始的真实数据足够相似。然而，这种相似性与用户的决策并没有直接关系，我们并没有引入与用户反馈相关的损失项。从当前来看，该领域的研究仍然处于空白状态。举例而言，在进行多模态预训练时，目前研究人员会利用预训练的结果帮助我们召回视频和文本，当前的主流的方法还是基于 BERT 进行各种各样的扩展 (Single-Stream、Double Stream 等)。以图片为例，我们会召回拥有相同颜色或者相同形状的图片，但是这种颜色或形状相似的图片并不一定会帮助消费者做出决策。同样，基于视频的表征进行召回的结果也并不一定能够帮助消费者做决策。针对这一问题，我们进行了大量探索，结合用户的反馈信息序列得到「因果嵌入」表征。

接下来，我分别针对前文提到的数据层、推理层、交互层展开叙述。

二、跨领域知识图谱

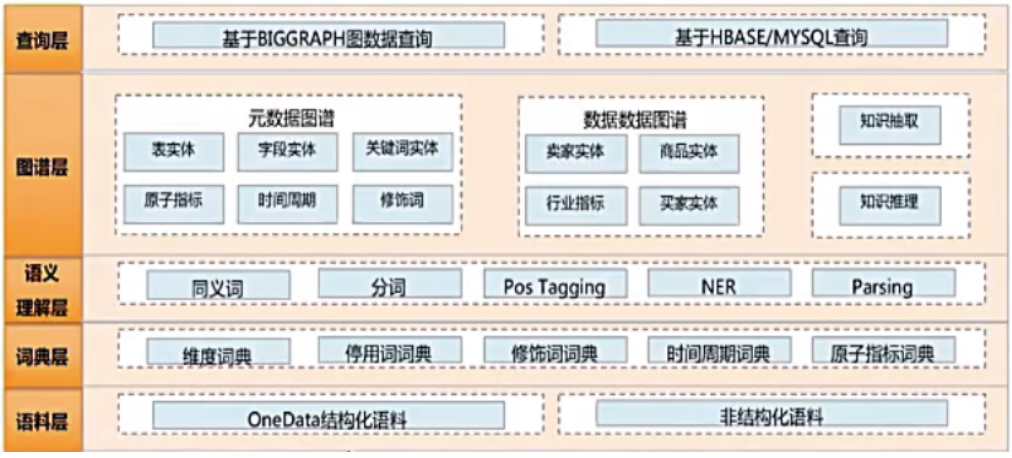


图 4：阿里知识图谱 Hybrid-KG

由于我们需要构建跨领域的知识图谱，这种图谱中实体的跨领域属性有很大的差异，我们需要建立核心节点并打通各个不同的图谱。在我们的工作中，我们以人、商品、商家作为核心的节点打通各个图谱。具体而言，我们在如图 4 所示的语料层中设有一个「oneData 结构化语料」。在得到核心节点之后，对于在不同域中的消费者，我们需要对属于相同消费者的行为进行聚合和知识提炼。

三、图神经网络推理引擎

我们基于图神经网络 (GNN) 构建了自己的推理引擎。简单来说，图神经网络就是结合了深度学习的图计算，既

体现了深度学习在特征工程、处理高阶非线性关系上的强大能力，同时又和图计算相结合，通过在图上的游走，进行推理和认知，具有较高的可解释性。例如，我们可以通过图上的游走路径解释将某种把商品推荐给用户的原因，或者将短视频与某商品搭配的原因。

尽管我们可以通过图神经网络进行归纳推理，但这也意味着我们需要进行高阶采样，这样会引入更多的风险。例如，在 CTR/CVR 预测任务中，目标函数中的真实值 y 是用户和物料之间直接相连的边。但是在图环境下，我们可能会进行二度或三度的游走。例如，在二度游走中，我们从一个用户出发，经过物料 A 之后再游走到物料 B。此时，我们可以根据各种理由建立 A 到 B 之间的路径。购买了 A 的人很可能购买 B，也有可能 B 能够帮助用户更好地使用 A，从而使购买了商品 A 的人喜欢观看视频 B。但是，对于特定的用户而言，是否存在 A 和 B 之间的路径仍然是一个很大的问题，并不一定在大多数人喜欢某件商品的情况下，某位特定的用户就一定喜欢该商品。因此，在这种场景下存在许多的风险与噪声。

如今，推荐系统和欺诈检测是 GNN 重要的应用领域。如果某篇论文的作者声称他们在推荐系统任务中游走了三度或者三度以上的邻居节点，那么这篇文章提出的模型一定没有在真实场景下应用过。因为在三度或三度以上的邻居节点的游走过程中，推荐系统的风险和噪声会十分巨大。但是，对于欺诈检测任务来说，我们必须使用高阶（high-order）邻居节点的信息，因为团伙作案往往会非常的小心，会尽可能的不去留下团的痕迹。所以，我们必须要在图上游走更多的邻居，才能检测出可疑的实体。因此，在不同的领域中，GNN 应该使用的邻居数量差别非常大。

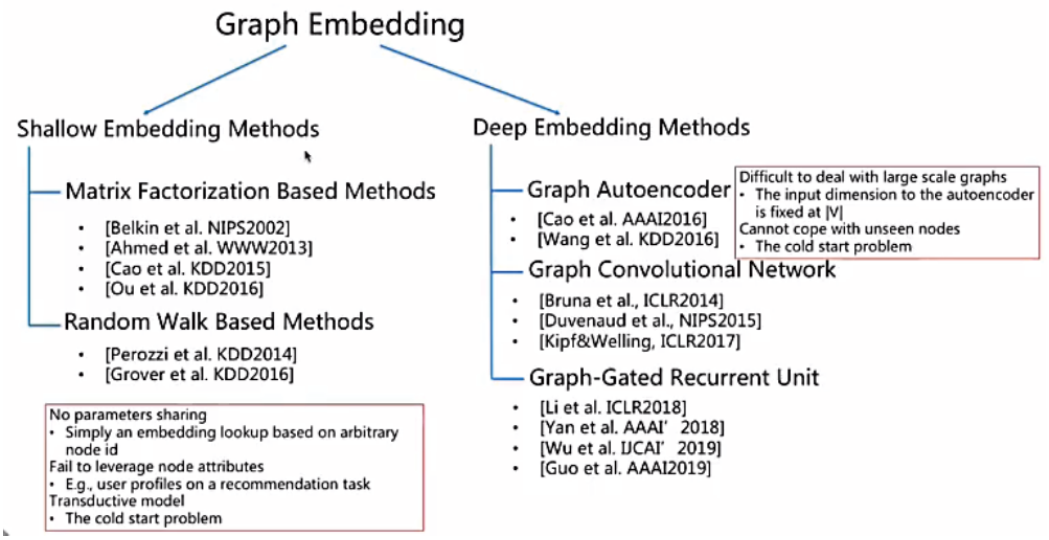


图 5：网络嵌入分类图

从推荐系统的角度来看，网络嵌入方法可以分为两大类。首先，浅层图嵌入方法可以分为基于矩阵分解的方法和基于随机游走的方法。对于基于矩阵分解的方法来说，其最大的问题就是很难进行全局优化，大多数当前的方法仍然是基于周围邻居的推荐。此外，每一个节点也不能被表征为良好的嵌入。

当下许多深度学习算法都受到了通过 NLP 或者视频处理算法的启发。例如，在图表征学习领域第一个具有里程碑意义的工作 Deepwalk 正是借鉴了 word2vec 的思想。DeepWalk 将每个节点的邻居作为其上下文，然后对

相应的目标函数进行优化。但是 DeepWalk 没有进行参数共享，不能有效利用节点属性。然而，在现实问题中，属性信息是非常重要的。对于表现出的行为相同的用户来说，如果他们拥有不同的年龄，来自不同的地域，从事不同的职业，但是最后他们做出的决策可能非常不一样。由于在某一时刻观测到的用户行为一定是不完全的，因此用户的属性十分重要。在没有用户的属性信息的情况下，冷启动的任务也是难以进行下去的。

深度嵌入方法则包含图自编码器、图卷积网络（GCN）、图的门控循环单元（Graph-GRU）等技术。例如，图自编码器指的是将自编码器技术应用到图数据上。然而，图自编码器仍然面临着可扩展性和冷启动的问题，因为自编码器接收的输入必须是固定的结构。我们的团队重点关注基于 GCN 的方法，在真实场景下，这类方法的性能较为优秀。

推荐系统包含丰富的序列化、动态的信息，我们试图通过深度学习中的动态神经网络（例如，GRU）与图计算相结合，从而更好地利用这些信息。由于图具有动态的结构，算法也需要足够高效并进行动态更新，所以这类工作对平台和算法都提出了更高的要求。我们团队的工作主要还是建立在 GCN 的基础之上。

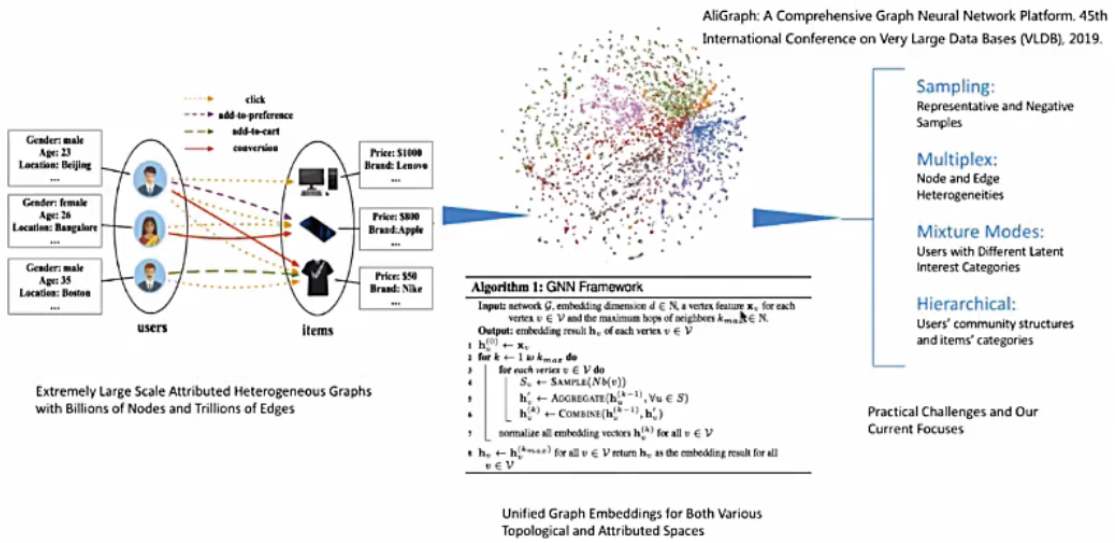


图 6：基于图嵌入的推荐

图 6 直观地展示了推荐系统的概念，可以看出它是一个带有属性的异构图。图 10 中的物料主要还是指商品，而实际上现在的真实场景下还包含各种各样其它模态的数据（如短视频、经验分享文本等），我们可以利用这些数据进行基于主题的推荐。图 10 中的边类型十分丰富（包括点击商品、添加到购物车等操作），我们需要设计有效的机制对图中节点进行表征。对于用户来说，其嵌入代表对用户当前的购物喜好的表征；而对于物料来说，其嵌入代表消费者对其喜爱成都以及它和其它的物料之间的相关性。我们希望基于得到的嵌入表征进行在线的召回。

上面介绍的嵌入主要被用于推荐的召回阶段，接下我们还需要对推荐结果进行排序，最后还需要进行端计算（edge computing）。值得一提的是，我们现在非常看好端计算，它是下一代云计算的发展方向。对于现在的云计算来说，虽然消费者是千人千面，但实际上我们的系统仍然是千人一模。尽管这个模型是非常庞大的集成模型，但是所有用户都共用同一个模型，并没有做到针对每位用户开发定制化的模型，而通过段计算就可以做到千人千模。

举例而言，强化学习在自动玩游戏的工作中取得了突飞猛进的进展，但是在工业界的落地效果并不尽如人意，这主要是云计算环境下的延迟造成的。对许多大数据公司而言，搜索、推荐和广告部门是其核心部门。如果我们使用云计算框架进行推荐，那么我们会预先保存好几十个将会在当前场景下推荐给用户的物料。接着，我们要将这些物料传到用户的客户端和手机端上。以手机淘宝为例，每个页面上可以展示 6 个物料，如果我们每次向用户传 30 个物料，那么我们需要等到用户翻阅完 5 个页面之后才能再传 30 个物料。但是，用户在翻阅第一个页面时就会产生点击，停留，手机滑动等动作。在通过云计算框架进行基于强化学习的推荐时，我们需要等到向用户传递下一组（30 个）物料才能改变策略，但是其实用户在翻阅第一个页面时就已经发生了行为上的变化，如果此时就将数据回传至强化学习模型，那么强化学习模型中的状态（state）和奖励（reward）其实已经完全改变了。因此，在云计算框架下，进行及与强化学习的推荐是不可行的。但是，在端上是有希望实现这一目标的，我们的团队正在探索结合网络嵌入实现这一目标。如果我们能在手机端做到千人千模，这不仅仅会提升推荐效率，这种端计算框架也非常适合应用联邦学习等算法，从而使我们可以很好地保护用户的隐私。基于端计算框架的强化学习也可能带来巨大的推荐效果的提升，我们对这种方法寄予厚望。

现在，我们继续讨论在云计算框架下通过 GNN 进行对物料的召回。实际上，召回结果决定了整个推荐系统性能的天花板。这是我们在进行召回的过程中，会从几十亿个商品、短视频等物料中召回用户当下最喜欢的物料。我们会通过排序对这些物料不断地进行筛选。如果用户当下最喜欢的物料不在我们的召回结果中，那么推荐系统最终的效果一定不会很好。

在运行 GNN 算法时，我们会输入一个异构属性图，希望得到每个节点的嵌入表征。值得一提的是，图中的节点包含一些特征，这些特征对于冷启动任务十分重要。首先，我们以每个节点自身的特征作为嵌入表征的初始状态。在进行信息聚合时，我们需要确定最大的跳数（hop），推荐系统的最大跳数为 2 或 3。因为当跳数太大时，会引入非常大的风险和噪声。

无论 GNN 模型如何变化，在游走经过所有节点的过程中，对某节点进行操作的三个主要步骤为：（1）对该节点的邻居节点进行采样。其中，邻居节点包含积极邻居和消极邻居，挑拣出消极邻居对于最终的嵌入质量非常大。（2）聚合邻居节点的信息。在选择完邻居节点后，我们将邻居节点的嵌入聚合到目标嵌入中。（3）将信息聚合结果与当前结果上一轮的嵌入进行融合。我们遍历所有节点，对它们都进行上述三个步骤的操作，直到最终收敛。

然而，要在超大规模的系统上执行上述 GNN 的算法步骤是相当具有挑战的。例如，在分布式计算场景下，我们无法把所有的节点放到一个进程上。此时，某节点与其邻居节点可能位于不同的进程上，如何设计采样与通信的机制是一个重要的课题。当图上的节点数为百万级时，上述问题还不够明显。但如果面对数十亿、数百亿的节点，上述问题就非常严重了。此外，我们还需要数个小时之内训练完这种超大图，这对于计算平台与算法之间的交互是极具挑战的。对这类平台的建设方法感兴趣的读者可以参阅本团队在 VLDB 2019 上发表的论文「AliGraph：A Comprehensive Graph Neural Network Platform」。

在算法层面上，我们总结出了以下几项重要的挑战：（1）对代表性样本和负样本的采样（2）异构图上多路复用（multiplex）。对于节点类型和边的类型大于等于 2 的异构图，使用同一套嵌入对异构图进行表征。例如，由某一种关系组织起来的图就构成了一个拓扑结构，每一个拓扑结构都会对应于一种嵌入空间中的表征，因此将多个拓扑结构变换为一致的嵌入表征是极具挑战性的。（3）混合模式。对于每个实体，如果我们只通过一种嵌入向量作为其表征，往往难以在下游任务中得到很好的效果。这是因为推荐系统中包含数十亿的各种各样物料（商

品、短视频、直播、图文等），每一位用户都会涉及不同的类目体系。如果我们只为用户建立一个嵌入表征（代表其喜好），那么该用户对服装类的喜好和电器类的喜好就对应于同一个表征。在没有任何约束条件的情况下，嵌入的效率会非常差，对于物料的的区分度会很低。这也正是我们在实际场景下经常遇到的问题，我们往往会召回非常流行的物料，但是对于每个用户来说，这种推荐的个性化程度十分有限。因此我们需要为物料建立多个嵌入表征，或者对嵌入的维度进行正交化等特殊的处理。(4) 层次化结构。由于人群和物料都存在类别结构，我们需要讲这种结构用于嵌入学习。

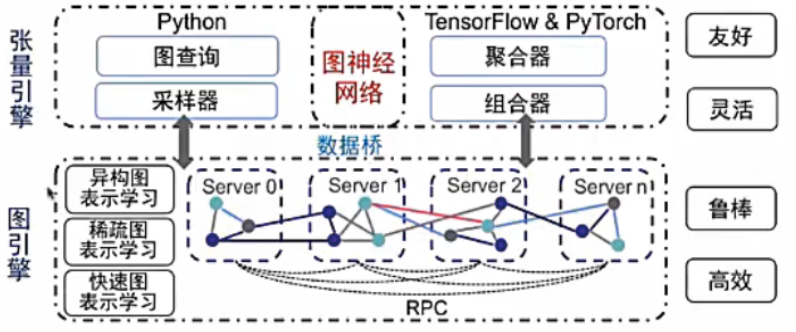


图 7：计算平台示意图

我们建立了如图 7 所示的图神经网络计算平台，平台主要包含图引擎和张量引擎两部分。该平台支持 TensorFlow 和 PyTorch 编程框架，可以在各个服务器之间进行高效的采样。我们在包含百亿级节点的图数据上进行了测试，测试结果表明，如果使用 MapReduce 等朴素的方法，仅仅完成采样工作就需数十个小时，而我们的计算平台在若干分钟内就可以高效地完成这项工作。

四、图神经网络模型

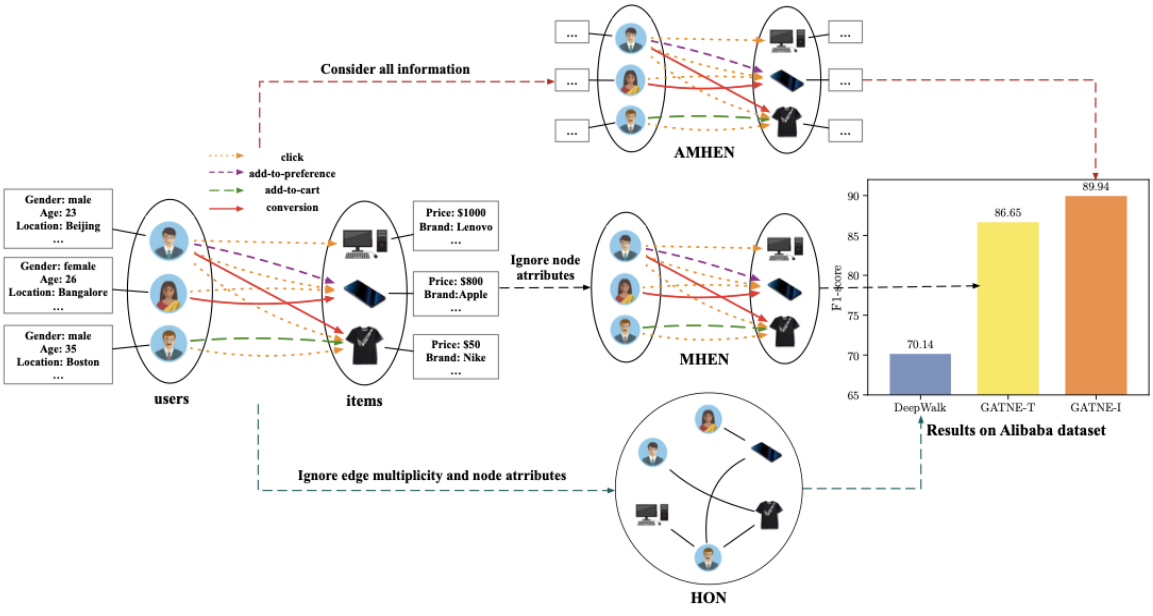


图 8：面向带属性的多路复用异构网络的表征学习

如图 8 所示，我们在 KDD 2019 上发表的论文「Representatoin Learning for Attributed Multiplex Heterogeneous Network」[1] 针对上文提到的带属性的异质网络展开了研究。针对这一问题，我们可以有多重解决方案。例如，我们可以并不考虑各种边的类型和属性，只考虑边的连接，从而将其转化为一个同质图。此外，我们还可以通过 Deepwalk 等目前最先进的方法来解决这一问题。在真实的阿里数据数据集得到的 F1-score 只有 70% 左右。当我们考虑异质的边之后，F1-score 则会上升至 86%。接着，当我们考虑节点的属性之后，最后的 F1-score 会上升至接近 90%，可见我们的方法获得的性能提升是非常显著的。

	Amazon			YouTube			Twitter			Alibaba-S		
	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1
DeepWalk	94.20	94.03	87.38	71.11	70.04	65.52	69.42	72.58	62.68	59.39	60.62	56.10
node2vec	94.47	94.30	87.88	71.21	70.32	65.36	69.90	73.04	63.12	62.26	63.40	58.49
LINE	81.45	74.97	76.35	64.24	63.25	62.35	62.29	60.88	58.18	53.97	54.65	52.85
metapath2vec	94.15	94.01	87.48	70.98	70.02	65.34	69.35	72.61	62.70	60.94	61.40	58.25
ANRL	71.68	70.30	67.72	75.93	73.21	70.65	70.04	67.16	64.69	58.17	55.94	56.22
PMNE(n)	95.59	95.48	89.37	65.06	63.59	60.85	69.48	72.66	62.88	62.23	63.35	58.74
PMNE(r)	88.38	88.56	79.67	70.61	69.82	65.39	62.91	67.85	56.13	55.29	57.49	53.65
PMNE(c)	93.55	93.46	86.42	68.63	68.22	63.54	67.04	70.23	60.84	51.57	51.78	51.44
MVE	92.98	93.05	87.80	70.39	70.10	65.10	72.62	73.47	67.04	60.24	60.51	57.08
MNE	90.28	91.74	83.25	82.30	82.18	75.03	91.37	91.65	84.32	62.79	63.82	58.74
GATNE-T	**97.44**	**97.05**	**92.87**	**84.61**	81.93	**76.83**	**92.30**	91.77	**84.96**	66.71	67.55	62.48
GATNE-I	96.25	94.77	91.36	84.47	**82.32**	**76.83**	92.04	**91.95**	84.38	**70.87**	**71.65**	**65.54**

图 9：实验结果

当我们考虑不同的异质拓扑结构时，由于我们无法对未观测的样本进行推理，因此这里建立的是一种转导推理模型。在推理过程中，我们只需将属性信息输入给模型。在这种转导推理模型中，我们需要考虑不同边的信息。与 R 条边相连的节点 i 有 R 个表征:

$$\mathbf{v}_{i,r} = \mathbf{b}_i + \alpha_r \mathbf{M}_r^T \mathbf{U}_i \mathbf{a}_{i,r} = \mathbf{b}_i + \alpha_r \mathbf{M}_r^T \sum_{p=1}^{m} \lambda_p \mathbf{u}_{i,p}$$

其中，b 为节点的基础嵌入，α为系数，$\alpha_{i,r}$ 是不同的边对于该节点的注意力，M_r^T 为分解出的矩阵。这种简单的模型取得了不错的效果，也非常适用于并行计算。

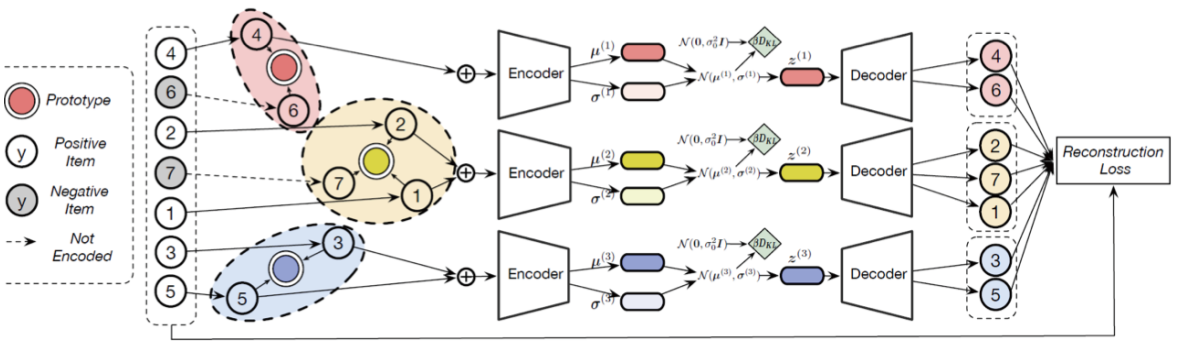


图 10：混合嵌入表征

我们团队关于混合嵌入表征的工作「Disentabled Representation Learning for Recommendation」[2] 发表在了 NIPS 2019 与 KDD 2020 上。在环境极其复杂的情况下，为实体仅仅建立一个表征是不够的。在这份工作中，我们通过变分自编码器（VAE）的方式向编码器中加入了混合高斯，从而进行推理。我们可以为每一个物料建立不同维度（例如，大小、颜色、种类）上的嵌入。当推荐系统发现用户有买包的意向时，推荐系统根据用户的点击和浏览序列会发现该用户对黑色的包比较感兴趣，此时推荐系统可以调节表征包的大小的嵌入，从而为用户推荐大小不同的黑色包。我们可以将其视为一种可控的推荐系统。

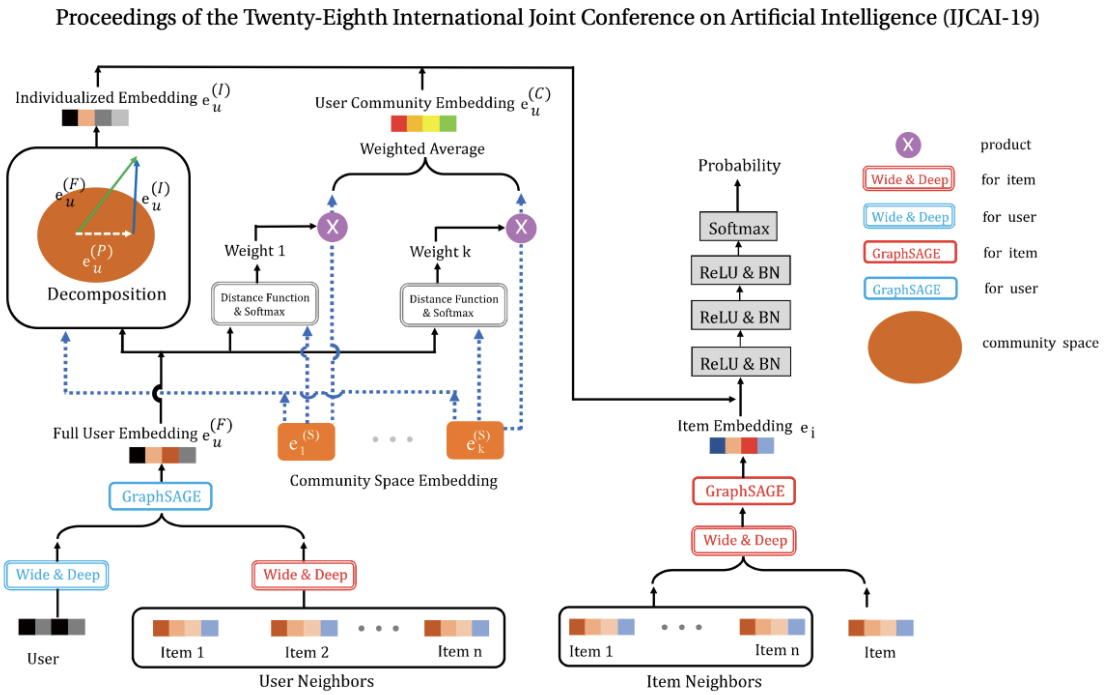


图 11：层次化的图嵌入

由于数据中涉及很多层次化的结构（例如，不同的人群，商品、短视频的不同的类目体系），我们需要更好地利用这种图中的社区（community）的嵌入。我们与 IJCAI 2019 上发表的论文「Hierarchical Representation Learning for Bipartite Graphs」[3] 对该问题进行了探索。

首先，我们使用 GraphSage 得到一个完整的用户嵌入。接着，我们假设用户从属于不同的社区。因此，我们将完整的用户嵌入通过投影分解为社区嵌入和个体嵌入。在社区嵌入方面，我们会对用户在社区空间中的嵌入进行加权平均。对于每一个用户来说，我们会将其个体嵌入与社区嵌入连接在一起。实验结果证明，我们的工作得到的性能提升较为显著。

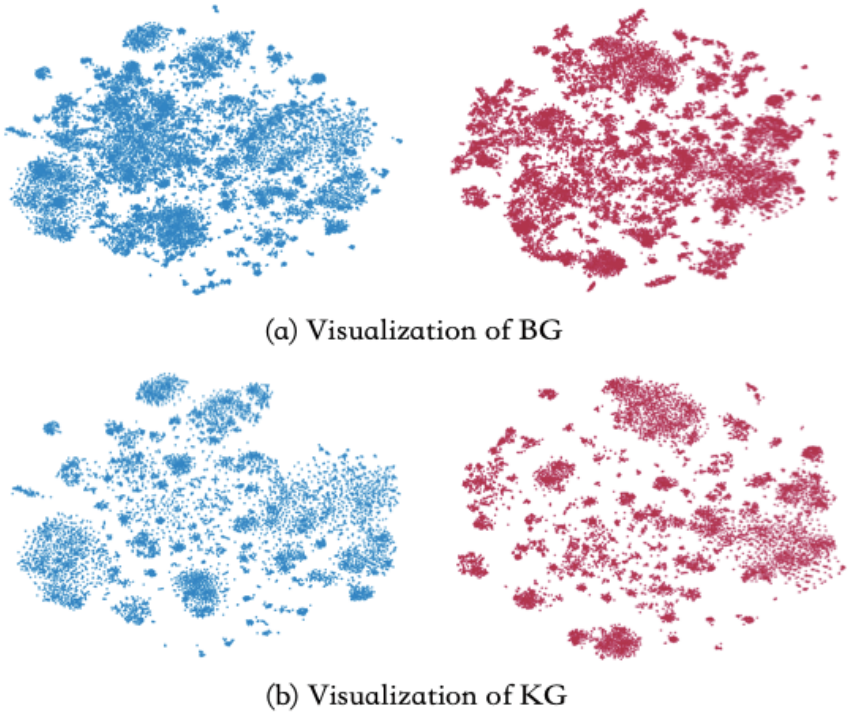


图 12：贝叶斯图神经网络 [4]

如今，推荐系统会使用到大量的知识图谱，也会使用基于用户行为的图神经网络嵌入。而将知识图谱的嵌入与基于用户行为的嵌入进行结合也是一个具有挑战性的问题，困难主要体现在两个方面：(1) 覆盖率。由于对知识图谱的挖掘已经是较为费时费力的，其量级为数百万节点。然而，用户行为图谱通常包含超过数百亿节点，其计算开销更加高昂。(2) 知识图谱是一种语义图，而推荐系统图是用户行为图，它们处于两个不同的空间中，进行一致性嵌入的挑战是十分巨大的。基于贝叶斯先验的思想，在 GNN 的初始化阶段，我们可以不再使用节点的特征作为初始表征，转而将其替换为预训练好的知识图谱嵌入。此外，在信息聚合的阶段，我们可以通过注意力机制将知识图谱嵌入用于调整用户行为图。如图 12 所示 [4]，在结合了知识图谱之后，除了在下游任务中的 AUC 变得更高，图的结构也被划分得更加明显。

五、稳定图结构

由于我们的推荐系统所处的环境十分复杂，所以我们希望学到的图较为鲁棒。

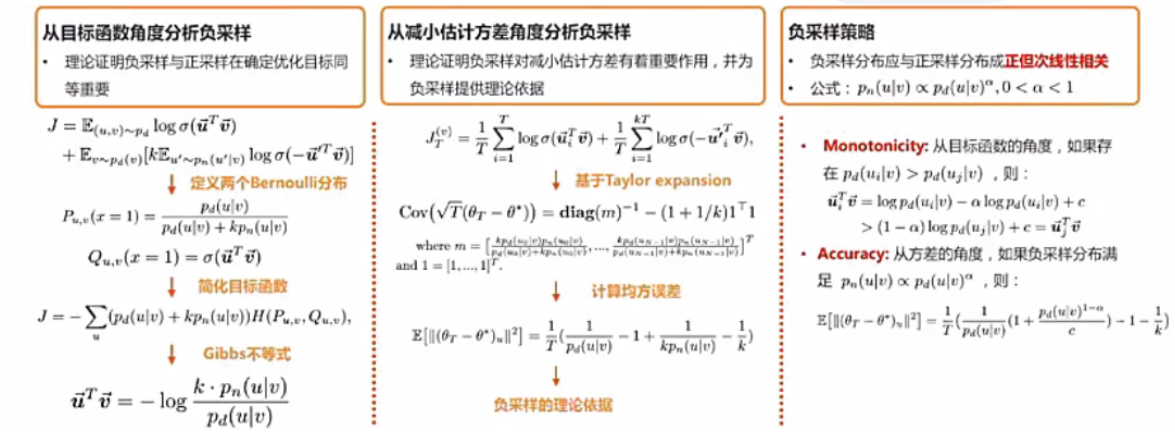


图 13：负采样策略

在今年的 KDD 上，我们团队发表了一系列的工作。首先，我们需要进行有效的负采样。当前 GNN 领域中的许多工作主要关注于如何进对正样本进行采样。而我们从理论上证明负样本的采样对于目标函数和减小方差都十分重要。

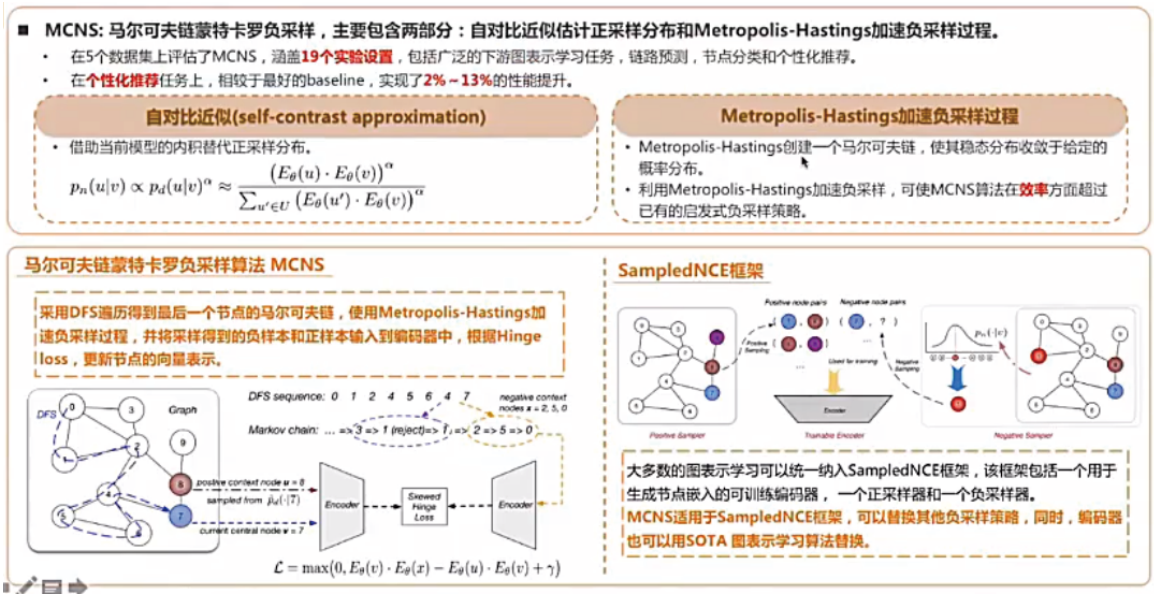


图 14：MCNS 负采样与 SampledNCE 框架

此外，我们还推出了基于 Metropolis–Hastings 的加速负采样的过程，并将其与图表示学习算法统一到了 SampledNCE 框架中。除了性能上的提升之外，模型的方差也得到了降低。

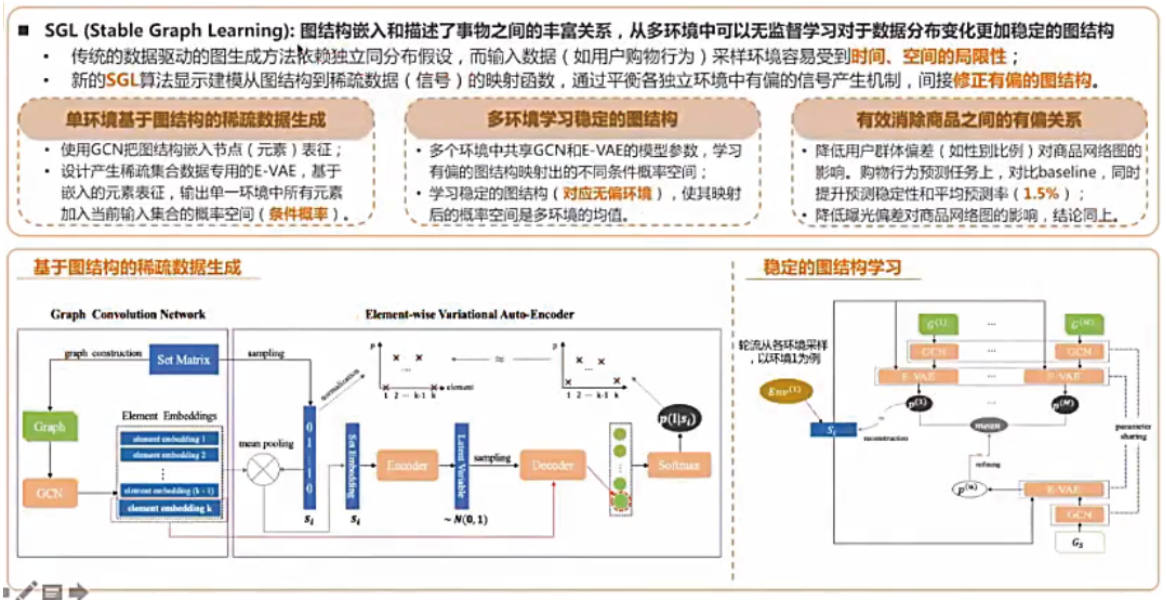


图 15：稳定的图结构学习算法

我们还结合了因果图的思想增强图结构学习的鲁棒性。众所周知，实际数据中存在选择偏差、曝光偏差等各种偏差（bias）。我们试图学习到一种更加稳定的图结构。例如，对于同一个人或者物料而言，在不同的日期学习对其进行表征学习，如果不加以控制，学习到的嵌入差别会很大。我们基于变分自编码器的思想，首先利用图数据训练一个 GCN，然后在不断的仿真过程中得到各实体的不同的表征，对这些表征进行平均池化后，最终通过编码器、解码器的变换得到鲁棒的图结构。

六、推荐系统多兴趣

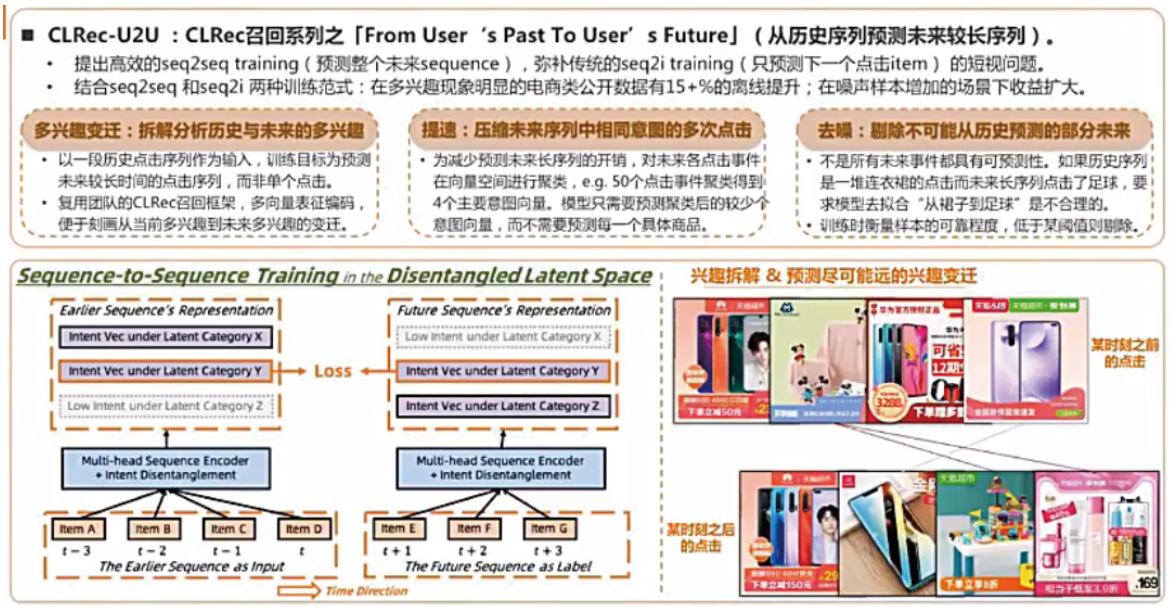


图 16：CLRec–U2U 示意图

每个实体对应于多个嵌入向量的表征。在「CLRec-U2U」中，我们加入了序列到序列的信号，从而学习到更多关于用户的嵌入，同时引入希望学习到一些长期兴趣。目前，性能最好的推荐系统还是建立在用户序列化建模的基础之上。然而，受平台的限制，这种建模方法主要学习到的是用户中短期的行为。在推荐系统当中，有一个尚未被很好地解决的开放性的问题：如何学习到以年为周期的用户兴趣。尽管用户序列建模的效果很好，但是目前我们主要使用的仍然是固定长度的用户序列。这是因为如今用户量过大（接近 10 亿），对于每个用户来说，即使我们只考虑过去 500 个点击的序列，仍然会产生十分巨大的数据量。实际上，对于活跃的用户来说，500 次点击可能只需要一两天就可以完成，但是对于一些不活跃的用户来说，可能是一年或一个月的点击量，我们无法对每个用户都做到以年为周期的多兴趣的向量召回与学习。我们的解决方案是，为每一位用户建立一个周期性的记忆网络，将目标注意力机制用于用户的短期兴趣，不断地激活以年为周期的记忆网络。

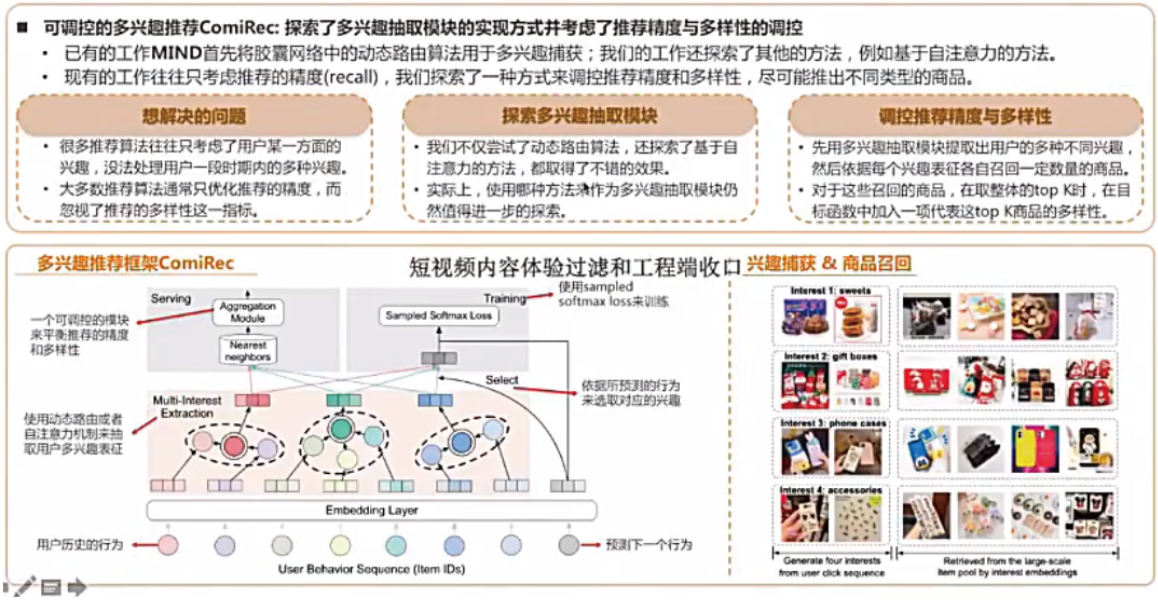


图 17：可调控的多兴趣推荐框架

在我们于 KDD 2020 发表的论文「Controllable Multi-Interest Framework for Recomendation」[5] 中，我们将物料嵌入的信息进行聚合，在信息提取层中会得到兴趣嵌入向量。在进行在线召回时，我们并不直接召回物料，而是先召回兴趣提取层，然后再通过这一层召回对应的物料。通过上述操作，我们可以让增加兴趣的维度。如果直接召回物料的嵌入，可能用户的兴趣某一个类别支配，但是实际上该用户可能还有其它不易透出的兴趣。

七、基于用户交互的内容理解

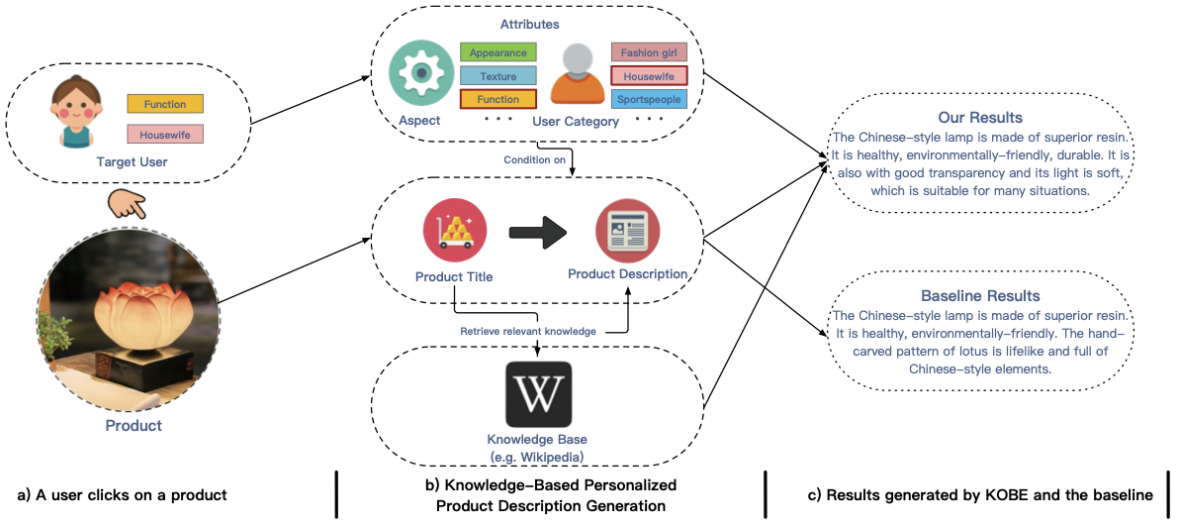


图 18：用户交互之意图生成

在与用户交互的过程中，我们主要需要做两件事情：(1) 基于用户交互的文本理解与生成；(2) 基于用户交互的短视频理解。在我们于 KDD 2019 上发表的论文「Towards Knowledge-Based Personalized Product Decription Generation in E-commerce」[6] 中，我们沿用了 Transformer 的结构，我们对于不同维度上的嵌入应用了不同的 Transformer，同时引入了一个外部的知识图谱作为词典。基于这份工作，我们在手机淘宝中可以通过用户不同的个人兴趣自动生成推荐理由。但是，考虑用户反馈的自然语言生成仍然是一个具有很大探索空间的课题。

在我看来，视频理解仍然是一个存在很大探索空间的课题。由于视频是一个故事的序列，仅仅使用一个嵌入来表征一段视频是远远不够的。另外，在与用户交互的过程中，某段视频对用户做决策的影响也是急需研究的问题。我们通过视频本身的内容，以及用户其它的属性生成了一些标题，帮助用户更好的去做出决策，根据故事线中的一些因素凸显出个性化的推荐，这份工作的性能提升也较为显著。

八、结语

在本次演讲中，我主要向大家介绍了认知推荐中三个主要的模块：(1) 跨领域知识图谱 (2) 推理引擎。我们现在主要依靠 GNN 做了很多研究，并真正的落地 (3) 用户交互。在通过 GNN 推理出用户意图之后，必须和用户进行交互。我们的研究方向主要是基于用户交互的文本生成和视频的理解。

参考资料：

[1] Representatoin Learning for Attributed Multiplex Heterogeneous Network, https://arxiv.org/pdf/1905.01669v2.pdf

[2] Disentabled Representation Learning for Recommendation, https://deepai.org/publication/learning-disentangled-representations-for-recommendation

[3] Hierarchical Representation Learning for Bipartite Graphs, https://www.ijcai.org/Proceedings/2019/398

[4] Bayes EMbedding (BEM): Refining Representation by Integrating Knowledge Graphs and Behavior-specific Networks, https://dl.acm.org/doi/pdf/10.1145/3357384.3358014

[5] Controllable Multi-Interest Framework for Recomendation, https://arxiv.org/abs/2005.09347

[6] Towards Knowledge-Based Personalized Product Decription Generation in E-commerce, http://keg.cs.tsinghua.edu.cn/jietang/publications/KDD19-Chen-et-al-KOBE.pdf

清华大学唐杰：CogDL - 基于图的深度学习开源工具包

整理：学术头条

在第二届智源大会“知识智能”专题论坛中，清华大学唐杰教授介绍了基于图的深度学习开源工具 CogDL。CogDL 是由清华大学知识工程实验室（KEG）联合北京智源人工智能研究院（BAAI）所开发的基于图的深度学习的开源工具包，底层架构 PyTorch，编程语言使用了 Python。

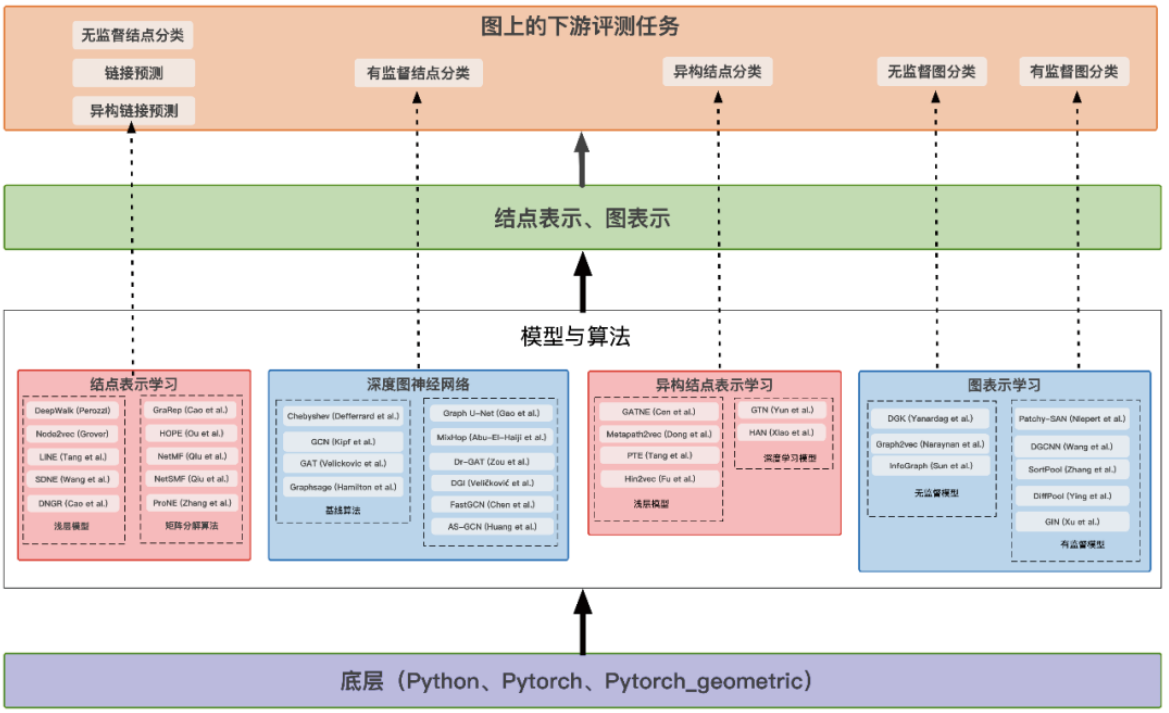
CogDL 工具包
项目页面：http://keg.cs.tsinghua.edu.cn/cogdl
GitHub 链接：https://github.com/THUDM/cogdl
中文介绍：https://github.com/THUDM/cogdl/blob/master/README_CN.md
智源链接：http://open.baai.ac.cn/cogdl-toolkit
网站（中文）：http://keg.cs.tsinghua.edu.cn/cogdl/cn/
文档：https://cogdl.readthedocs.io/

CogDL 允许研究人员和开发人员轻松地针对图数据进行表示学习、对比基线算法，支持节点分类、链接预测、图分类等任务。CogDL 的一个特点是整合了当下流行的图表示学习算法，包括：非图神经网络算法如 Deepwalk、LINE、Node2vec、NetMF、ProNE、methpath2vec、PTE、graph2vec、DGK 等；以及图神经网络算法如 GCN、GAT、GraphSAGE、GTN、HAN、GIN、DiffPool 等。工具包重现了所有算法，可以一键实现基线算法的对比。

- 无监督结点分类: ProNE (Zhang et al, IJCAI'19), NetMF (Qiu et al, WSDM'18), Node2vec (Grover et al, KDD'16), NetSMF (Qiu et at, WWW'19), DeepWalk (Perozzi et al, KDD'14), LINE (Tang et al, WWW'15), Hope (Ou et al, KDD'16), SDNE (Wang et al, KDD'16), GraRep (Cao et al, CIKM'15), DNGR (Cao et al, AAAI'16).
- 半监督结点分类: Graph U-Net (Gao et al., 2019), MixHop (Abu-El-Haija et al., ICML'19), DR-GAT (Zou et al., 2019), GAT (Veličković et al., ICLR'18), DGI (Veličković et al., ICLR'19), GCN (Kipf et al., ICLR'17), GraphSAGE (Hamilton et al., NeurIPS'17), Chebyshev (Defferrard et al., NeurIPS'16).
- 异构结点分类: GTN (Yun et al, NeurIPS'19), HAN (Xiao et al, WWW'19), PTE (Tang et al, KDD'15), Metapath2vec (Dong et al, KDD'17), Hin2vec (Fu et al, CIKM'17).
- 链接预测: ProNE (Zhang et al, IJCAI'19), NetMF (Qiu et al, WSDM'18), Node2vec (Grover et al, KDD'16), DeepWalk (Perozzi et al, KDD'14), LINE (Tang et al, WWW'15), Hope (Ou et al, KDD'16), NetSMF (Qiu et at, WWW'19), SDNE (Wang et al, KDD'16).
- 多重边链接预测: GATNE (Cen et al, KDD'19), NetMF (Qiu et al, WSDM'18), ProNE (Zhang et al, IJCAI'19), Node2vec (Grover et al, KDD'16), DeepWalk (Perozzi et al, KDD'14), LINE (Tang et al, WWW'15), Hope (Ou et al, KDD'16), GraRep (Cao et al, CIKM'15).
- 无监督图分类: Infograph (Sun et al, ICLR'20), Graph2Vec (Narayanan et al, CoRR'17), DGK (Yanardag et al, KDD'15).
- 有监督图分类: GIN (Xu et al, ICLR'19), DiffPool (Ying et al, NeuIPS'18), SortPool (Zhang et al, AAAI'18), PATCH_SAN (Niepert et al, ICML'16), DGCNN (Wang et al, ACM Transactions on Graphics'17).

CogDL 还提供了更多 benchmark 数据集来对不同模型进行更加全面的评测，提供更加客观的排行榜。与其他图表示学习工具包相比，CogDL 的特性包括:

- 任务导向: CogDL 以图上的任务为主，提供了相关的模型、数据集以及我们得到的排行榜。
- 一键运行: CogDL 支持用户使用多个 GPU 同时运行同一个任务下多个模型在多个数据集上的多组实验。
- 多类任务: CogDL 支持同构 / 异构网络中的节点分类和链接预测任务以及图分类任务。
- 可扩展性: 用户可以基于 CogDL 已有的框架来实现和提交新的数据集、模型和任务。



项目页面：http://keg.cs.tsinghua.edu.cn/cogdl
GitHub 链接：https://github.com/THUDM/cogdl

下面简单介绍一下 CogDL 当前在各个图任务上不同算法的对比情况（排行榜），包括节点分类（分为是否具有节点属性），链接预测（分为同构和异构），图分类（分有监督和无监督）。

一、节点分类

1.1 无监督多标签节点分类

这是一个根据无监督的多标签节点分类设置而构建的排行榜，研究团队在几个真实的数据集上运行 CogDL 上的无监督表示学习算法，并将输出的表示和 90% 的节点标签作为经 L2 归一化的逻辑回归中的训练数据，使用剩余 10% 的标签作为测试数据，计算并按照 Micro–F1 的大小进行排序。

Rank	Method	PPI	Blogcatalog	Wikipedia
1	ProNE (Zhang et al, IJCAI'19)	**26.32**	**43.63**	57.64
2	NetMF (Qiu et al, WSDM'18)	24.86	43.49	**58.46**
3	Node2vec (Grover et al, KDD'16)	23.86	42.51	53.68
4	NetSMF (Qiu et at, WWW'19)	24.39	43.21	51.42
5	DeepWalk (Perozzi et al, KDD'14)	22.72	42.26	50.42
6	LINE (Tang et al, WWW'15)	23.15	39.29	49.83
7	Hope (Ou et al, KDD'16)	23.24	35.52	52.96
8	SDNE (Wang et al, KDD'16)	20.14	40.32	48.24
9	GraRep (Cao et al, CIKM'15)	20.96	34.35	51.84
10	DNGR (Cao et al, AAAI'16)	16.45	28.54	48.57

1.2 半监督有属性的节点分类

下面是几种常见的图神经网络算法在半监督节点分类任务上构建的排行榜。研究团队在经典的三个数据集 Cora, Citeseer 和 Pubmed 进行了实验，以 Accuracy 指标来评价模型的效果。

Rank	Method	Cora	Citeseer	Pubmed
1	Graph U-Net (Gao et al., 2019)	**84.4 ± 0.6**	**73.2 ± 0.5**	79.6 ± 0.2
2	MixHop (Abu-El-Haija et al., ICML'19)	81.9 ± 0.4	71.4 ± 0.8	**80.8 ± 0.6**
3	DR-GAT (Zou et al., 2019)	83.6 ± 0.5	72.8 ± 0.8	79.1 ± 0.3
4	GAT (Veličković et al., ICLR'18)	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3
5	DGI (Veličković et al., ICLR'19)	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.6
6	GCN (Kipf et al., ICLR'17)	81.4 ± 0.5	70.9 ± 0.5	79.0 ± 0.3
7	GraphSAGE (Hamilton et al., NeurIPS'17)	80.1 ± 0.2	66.2 ± 0.4	76.9 ± 0.7
8	Chebyshev (Defferrard et al., NeurIPS'16)	79.2 ± 1.4	69.3 ± 1.3	68.5 ± 1.2

1.3 异构节点分类

对于异构的节点分类任务，研究团队使用 Macro F1 来评价模型的效果，在 GTN 算法的实验设置和数据集下对所有算法进行评估。

Rank	Method	DBLP	ACM	IMDB
1	GTN (Yun et al, NeurIPS'19)	**92.03**	**90.85**	**59.24**
2	HAN (Xiao et al, WWW'19)	91.21	87.25	53.94
3	PTE (Tang et al, KDD'15)	78.65	87.44	48.91
4	Metapath2vec (Dong et al, KDD'17)	75.18	88.79	43.10
5	Hin2vec (Fu et al, CIKM'17)	74.31	84.66	44.04

二、链接预测

2.1 链接预测

对于链接预测任务，我们通过隐去数据集中 10% 的边，然后对隐去的边进行预测，使用 ROC-AUC 指标来评估模型的性能。ROC-AUC 指标代表了一条随机未观察到的边对应的两个节点比一条随机不存在的边对应的两个节点更相似的概率。

Rank	Method	PPI	Wikipedia
1	ProNE (Zhang et al, IJCAI'19)	79.93	82.74
2	NetMF (Qiu et al, WSDM'18)	79.04	73.24
3	Hope (Ou et al, KDD'16)	80.21	68.89
4	LINE (Tang et al, WWW'15)	73.75	66.51
5	Node2vec (Grover et al, KDD'16)	70.19	66.60
6	NetSMF (Qiu et at, WWW'19)	68.64	67.52
7	DeepWalk (Perozzi et al, KDD'14)	69.65	65.93
8	SDNE (Wang et al, KDD'16)	54.87	60.72

2.2 异构链接预测

对于异构链接预测任务，我们会对数据集中的某些视图下的链接进行预测，然后取 Macro ROC-AUC 作为评价

指标。我们提出的 GATNE 模型是专门针对这种多视图的异构网络，而这里列举的其他方法只能处理同构网络，因此我们向这些方法分别输入不同视图下的网络，并为每种视图下的网络分别获得节点表示用于链接预测，最后同样采用 Macro ROC–AUC 作为评测指标。

Rank	Method	Amazon	YouTube	Twitter
1	GATNE (Cen et al, KDD'19)	97.44	**84.61**	**92.30**
2	NetMF (Qiu et al, WSDM'18)	**97.72**	82.53	73.75
3	ProNE (Zhang et al, IJCAI'19)	96.51	78.96	81.32
4	Node2vec (Grover et al, KDD'16)	86.86	74.01	78.30
5	DeepWalk (Perozzi et al, KDD'14)	92.54	74.31	60.29
6	LINE (Tang et al, WWW'15)	92.56	73.40	60.36
7	Hope (Ou et al, KDD'16)	94.39	74.66	70.61
8	GraRep (Cao et al, CIKM'15)	83.88	71.37	49.64

三、图分类

CogDL 统一对有监督和无监督的图分类算法在相同的若干个真实的数据集上运行和评测。有监督图分类算法使用 kfold 对算法进行调参、训练和评测；无监督图分类算法学习到图的表示之后，将其作为输入并利用 90% 的图的标签作为 SVM 的训练数据，使用剩余 10% 的标签作为测试数据。两者均计算并按照 Accuracy 的大小进行排序。

Rank	Method	MUTAG	IMDB-B	IMDB-M	PROTEINS	COLLAB
1	Infograph (Sun et al, ICLR'20)	88.95	74.50	51.33	73.93	78.14
2	GIN (Xu et al, ICLR'19)	88.33	76.70	50.80	72.86	79.52
3	DiffPool (Ying et al, NeuIPS'18)	85.18	74.30	50.73	75.30	77.20
4	SortPool (Zhang et al, AAAI'18)	85.61	75.20	51.07	74.11	79.98
5	Graph2Vec (Narayanan et al, CoRR'17)	83.68	73.90	52.27	73.30	85.58
6	PATCH_SAN (Niepert et al, ICML'16)	85.12	76.00	46.20	75.50	75.42
7	DGCNN (Wang et al, ACM Transactions on Graphics'17)	83.33	69.50	46.33	66.67	77.45
8	DGK (Yanardag et al, KDD'15)	83.68	55.00	40.40	72.59	/

四、CogDL 怎么用?

开发者在 GitHub 项目中介绍了 CogDL 的详细使用方法。

CogDL 安装请按照这里的说明来安装 PyTorch 和其他依赖项:

https://github.com/pytorch/pytorch#installation
https://github.com/rusty1s/pytorch_geometric/#installation
pip install –e .

基本用法可以使用 python train.py –-task example_task –-dataset example_dataset –-model example_method 来在 example_data 上运行 example_method 并使用 example_task 来评测结果。

- –-task，运行的任务名称，像 node_classification，unsupervised_node_classification，link_prediction 这样来评测表示质量的下游任务;
- –-dataset，运行的数据集名称，可以是以空格分隔开的数据集名称的列表，现在支持的数据集包括 cora，citeseer，pumbed，PPI，wikipedia，blogcatalog，dblp，flickr 等;
- –-model，运行的模型名称 , 可以是个列表，支持的模型包括 gcn，gat，deepwalk，node2vec，hope，grarep，netmf，netsmf，prone 等。

如果你想在 Cora 数据集上运行 GCN 模型，并用 node classification 评测，可以使用如下指令:

python train.py –-task node_classification –-dataset cora –-model gcn

五、自定义数据集或模型

提交你的“牛”算法: 如果你有一个性能优异的算法并愿意发布出来，你可以在我们的代码仓库里提出一个 issue。在验证该算法的原创性，创造性和效果后，我们将该算法的效果添加到我们的排行榜上。

添加你自己的数据集: 如果你有一个独特，有研究价值的数据集并且愿意发布出来，你可以在代码仓库里提出一个 issue，我们将把所以适合的模型在您的数据集上运行并更新我们的排行榜。

实现你自己的模型: 如果您有一个性能优秀的算法，并愿意在工具包中实现它，以帮助更多的人，您可以创建一个 pull request。

Pinterest 首席科学家 Jure Leskovec：图神经网络的最新研究进展

整理：智源社区 任黎明

在第二届北京智源大会“知识智能”专题论坛中，斯坦福大学计算机系副教授，Pinterest 首席科学家 Jure Leskovec 做了题为《Recent Advancements in Graph Neural Networks: Simulation, Q2B, and OGB》的主题演讲。Jure Leskovec 是图表示学习方法 node2vec 和 GraphSAGE 作者之一，他的主要研究为大规模社交和信息网络的挖掘和建模，具体研究可查阅 https://ogb.stanford.edu。

在本次演讲中，Jure 介绍了图神经网络的最新研究进展及如何通过深度学习网络提高图神经网络的表达能力，随后他深入讲述了图神经网络如何应用于知识图谱谱、物理建模等行业领域。以下为演讲全文。

一、研究背景

近年来，机器学习领域的研究极为火热，伴随而来产生了一系列深度学习工具箱，这些工具箱大大加速了机器学习在各行各业中的应用。但是回顾这些研究进展，我们会发现其输入信息的表示形式多为简单的序列（文本 / 语音）或矩阵（图像）。而事实上，并非所有事物的数据都可以视为序列或矩阵，有许多数据是以图（graph）的形式呈现的。因此，如何将深度学习的研究推广到这种图数据上，是一个非常有意思的问题。

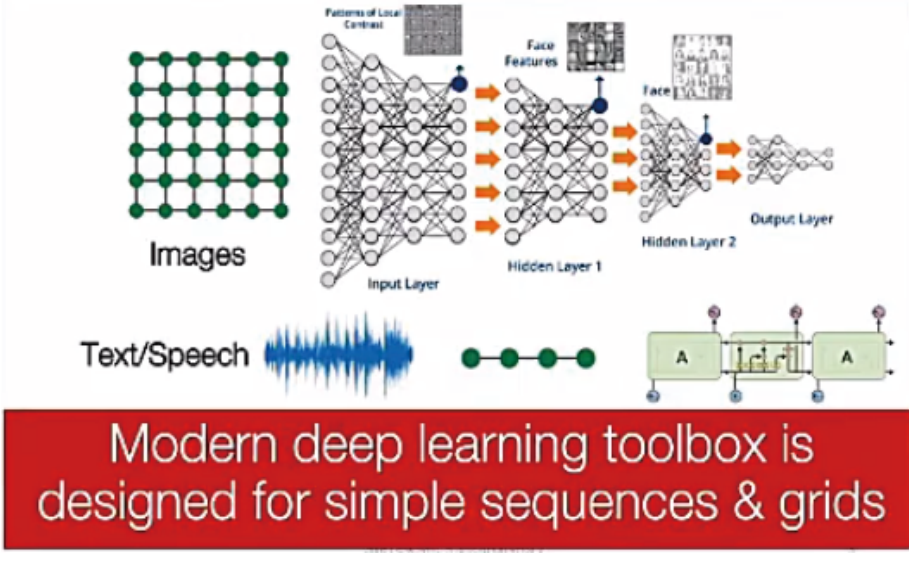


图 1：现有机器学习（深度学习）工具箱

针对这一问题，图神经网络是目前比较热门的研究方法。Jure 的研究包括对图神经网络的每个节点进行图卷积、正则化等，从而为每个节点提供表示特征。基本思路是，通过这些表示特征对图中的未知节点（node）训练一个神经网络来产生 embedding，然后利用这些 embedding 作为图神经网络的输入进行节点标签预测、链接预测及图分类等任务。

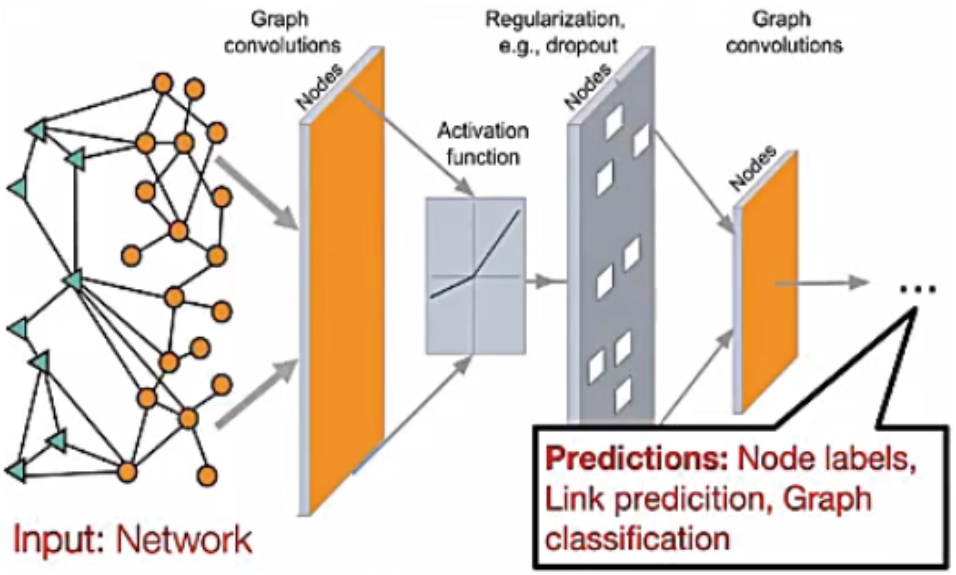


图 2：图表示学习

由于图神经网络的 size 具有任意性、结构复杂性、无序性、动态及多模态的特征（图 3），这些复杂多变的特征给图神经网络的研究带来巨大的挑战。

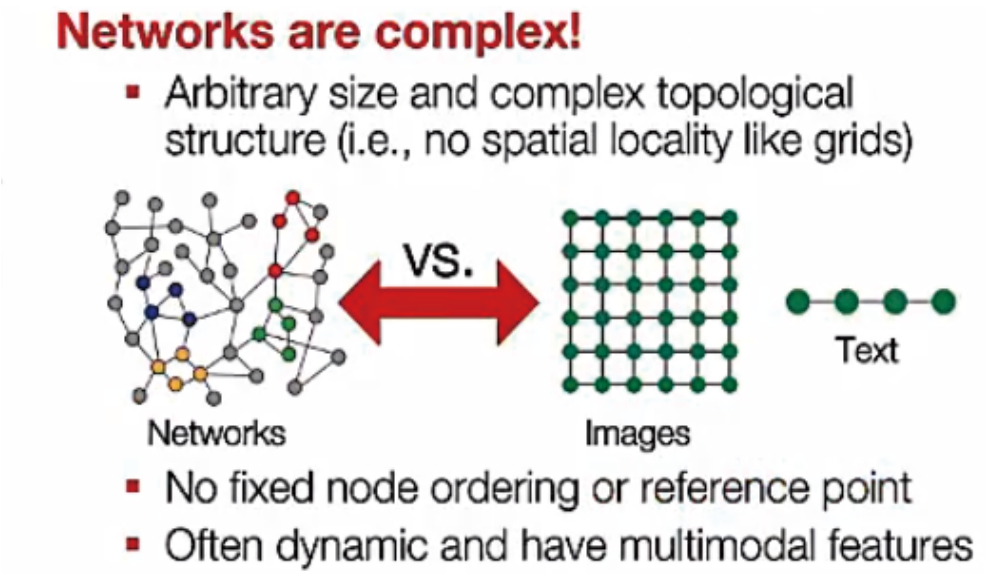


图 3：图神经网络研究的困难

二、图神经网络

2.1 GraphSAGE– 图神经网络 [2]

针对上述挑战，Jure 在 NIPS 2017 的一篇文章 [3] 中提出一种在大图上进行归纳表示学习方法。该方法通过卷积网络对图中所有节点定义一个计算图，计算图中的每一条边都是一个转换 / 聚合函数。在计算图中，目标节

点通过聚合函数从相邻节点进行采样和收集目标节点的特征，从而产生一种节点的表示方法，通过这个表示方法可以进行未知节点的分类和链接预测。举例来说，如果要预测节点 A，可以从邻近的节点（B，D，C）等收集信息（图 4），然后通过聚合函数学习邻近节点的信息，来更新节点 A 所在的计算图的神经网络以准确预测节点 A（图 5，其中 $h_A^{(k)}$ 表示节点 A 的 XA 属性，σ（·）表示 sigmod 激活函数）。

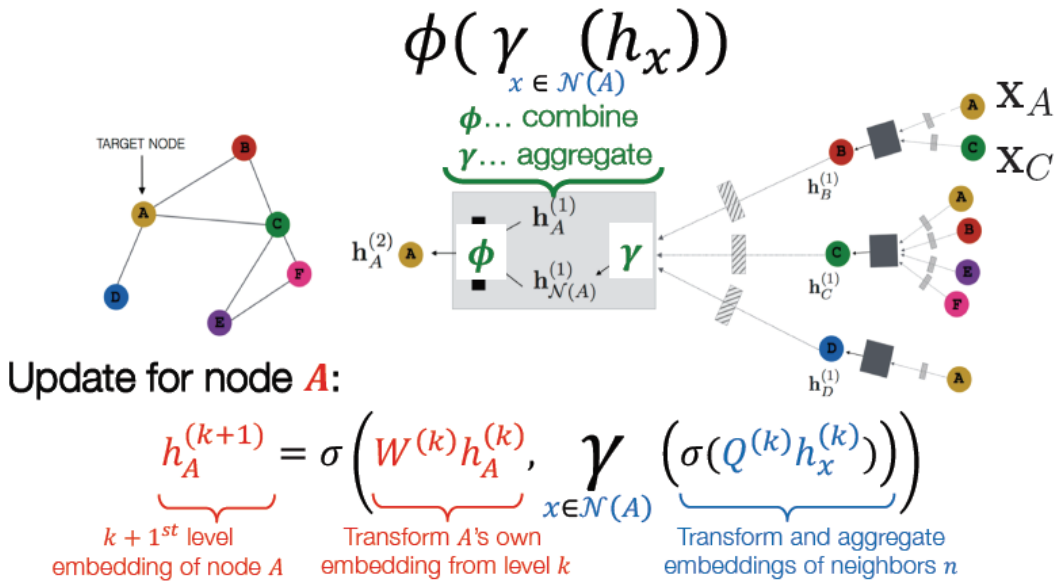


图 4： 计算图网络

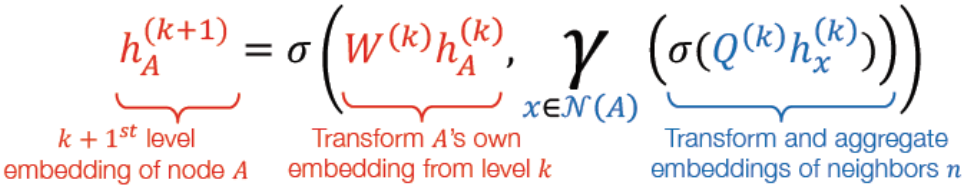


图 5：节点 A 的更新函数

与传统的图表示学习相比，计算图方法对给定节点的预测依赖于它的邻近节点的属性（图 6）；此外，该方法训练一组聚合函数，一个节点通过聚合函数从不同 hops 或者不同搜索深度的邻近节点蕴含的特征信息，可以进行未知节点预测任务。因此，计算图方法可以看作是学习图的低通和高通算子的线性组合。

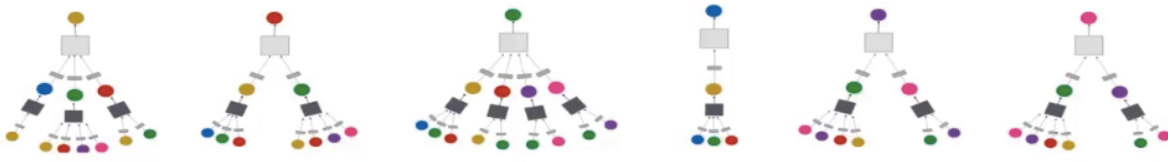


图 6：聚合邻近节点

2.2 层级图表示学习

在 NeurIPS 2018 的文章 [4] 中，Jure 等人提出了一种基于可微池化的层次图表示学习算法（Hierarchical graph representation learning with differentiable pooling）。在这项研究中，他们注意到，大多数图（网络）是具有层次结构的。而 GNN 的框架是基于 CNN 实现的，但是 CNN 的结构是扁平化的。此外，现有 GNN 仅通过图的边进行网络信息传播，无法通过层级化的方式对网络进行推断。因此，他们基于图的分层结构，提出了一种可微池化的层次图表示学习算法 DIFFPOOL，算法的框架如下（图 7）：

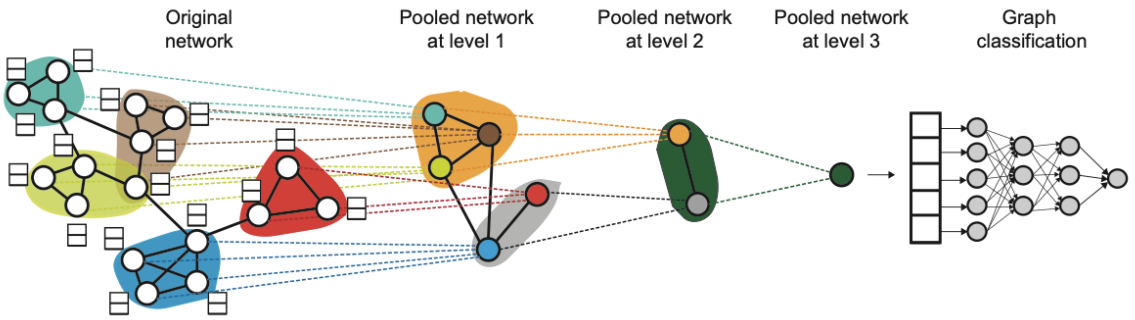


图 7：DIFFPOOL 算法框架图

从图 7 中可以看出，DIFFPOOL 算法是一种分层 pooling 的 GNN 算法，在每一层上都运行一个 GNN 模型来获得节点的嵌入；然后利用学习到的嵌入信息将节点聚类在一起，并基于这个聚类后的粗化图，再次运行 GNN 模型；直至最后一层。这个 DIFFPOOL 算法最终会得到整个图的向量表示，利用这个向量可以进行图的分类任务。

DIFFPOOL 算法的核心思想是通过提供一个能够区分图中分层节点的 pooling 操作来得到一个更深、更多层结构的 GNN 模型。同时 DIFFPOOL 算法能够和多种 GNN 模型进行融合，也说明该算法具有一定的泛化性 [9]。

2.3 GNN 的表达能力

图神经网络的研究群体日渐庞大，相继出现一系列的神经网络模型。但 GNN 模型的设计主要来自于经验，极度缺少理论性的解释。因此一个巨大疑惑自然而出：GNN 为什么有效？

在 ICLR 2019 上，Jure 等人发表的《How Powerful are Graph Neural Networks?》[6] 一文，回答了这一问题。他们利用一个理论框架分析主流图神经网络，1）描述了 GNNs 的性能上限，证明了 GNN 最多只和 Weisfeiler-Lehman（WL）test 一样有效；2）建立了一个简单的神经结构——图同构网络（GIN），并证明了它的判别 / 表达能力和 WL 测试一样；3）分析了 GCN 和 GraphSAGE 等主流 GNNs 在捕获图结构上的不足和特性。

在本次演讲中，Jure 认为，GNN 最强大的能力主要表现为它能够区分不同结构的根子树（rooted subtrees）。如图 8 所示，E 和 F 拥有相同的图结构，假如将图中所有节点信息去掉，那么图神经网络显然无法区分不同的节点；而如果 GNN 的聚合函数具有高度表达性，并且该聚合函数能够对有重复元素的邻近节点特征向量进行单射函数建模，则 GNN 就能区分紫色和粉红色的两个根结点。

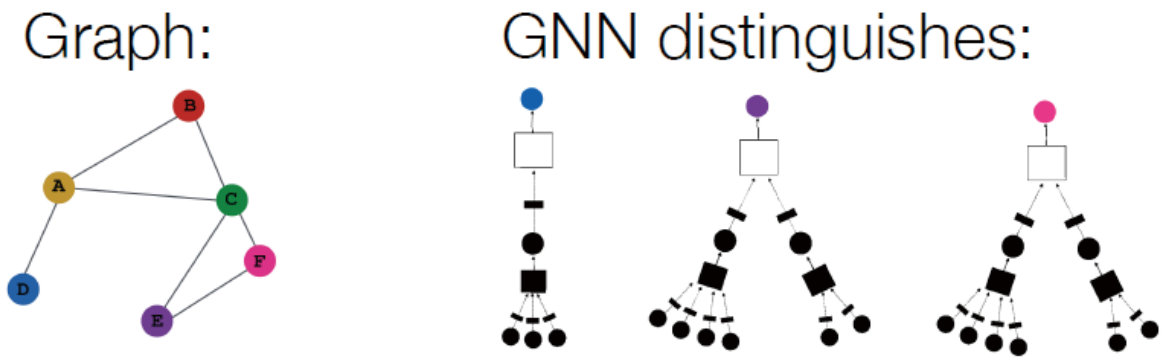


图 8： 根子树

基于以上研究，Jure 提出关于图神经网络的三个重要结论。

a. 图神经网络可以做到的两点

- 学习如何从附近节点“借用”特征信息来丰富目标节点。 如果想要对一个目标节点进行预测，那么可以通过聚合函数从邻近节点得到相关信息及特征，并学习如何以最佳的方式表示该节点的特征； 如果想提高目标节点的预测精度，也可以扩大邻近节点的范围。
- 每个节点都有不同的计算图，这个网络可以学习局部的图结构。

b. 计算图是可以选择的

- 因为聚合函数并没有要求必须所有邻近点都参与，可以对邻近点进行策略性地选择或抽样，这种策略在实际应用中已经获得了巨大的成效。

c. GNNs 在一些情况下也会出错

因为 GNN 无法区分同构节点，因此在没有节点特征的情况下，具有相同根子树的节点将会被归为同一类。

三、图神经网络的应用

3.1 图神经网络学习物理模拟器

在 ICML 2020 上，Jure 等人与 Google 、DeepMind 合作发表了一篇用图网络来学习仿真复杂物理现象（特别是复杂流体动力学和粒子运动学）的研究 [5]。用 ML 学习仿真的优点在于，通过模拟复杂物理现象的结构可以直接学习物理定律、优化效率、提高现有物理模拟器的精度，并可以用于控制和推断。

在这篇文章的研究中，Jure 等人在给定的粒子环境中，定义了一个图神经网络来学习复杂物理模拟器。他们根据规定粒子在目标粒子周围的位置特征，建立了领域图方法来预测粒子的特征，这些特征包括位置、速度、加速度等。然后通过图神经网络对整个复杂的粒子运动过程进行循环更新反馈（图 9）。通过这种方法，可以将不同类型的材料的性质转变为图神经网络中节点的特征，通过对不同特征的表示学习，从而可以了解材料的状态及性质是如何随着运动状态的改变而变化的。

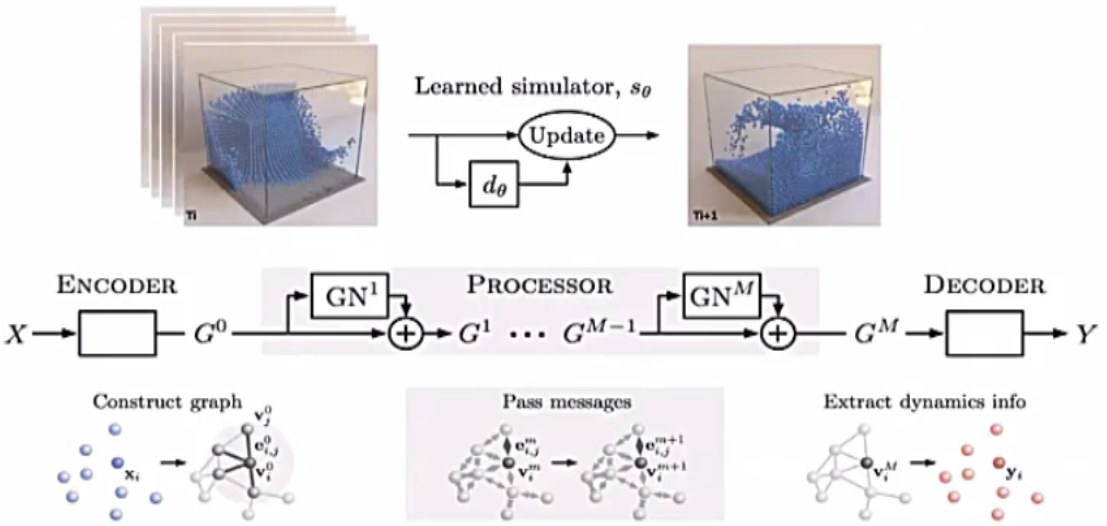


图 9 ： GNNs 在复杂物理现象模拟中的应用

3.2 知识图谱推理

图神经网络可以与知识图谱进行结合进行推理，但在推理中存在诸多问题，例如复杂异构（多种类型的实体和关系），对较大的知识图谱进行知识查询（逻辑推理）时会有许多噪声以及存在未观察到的数据等。针对这些问题，Jure 等人使用复杂的多度查询的方法对知识图谱的链接（link）进行推理预测。复杂多度的查询方法是通过图神经网络的表示学习将知识图谱（包含节点和实体的联合输入查询 q）嵌入到欧几里得空间，并在欧几里得空间中应用表示学习优化与知识图谱网络相对应的神经网络算子。在知识图谱推理过程中不同知识图谱的节点会被一个框覆盖，通过框的交叉变换对这些神经网络算子进行优化，然后通过查询嵌入空间中的邻近节点，从而预测某个节点或链接（link）是否满足知识图谱知识查询（逻辑推理）的查询目标。所以，这个方法也被称为查询框嵌入（图 10）。

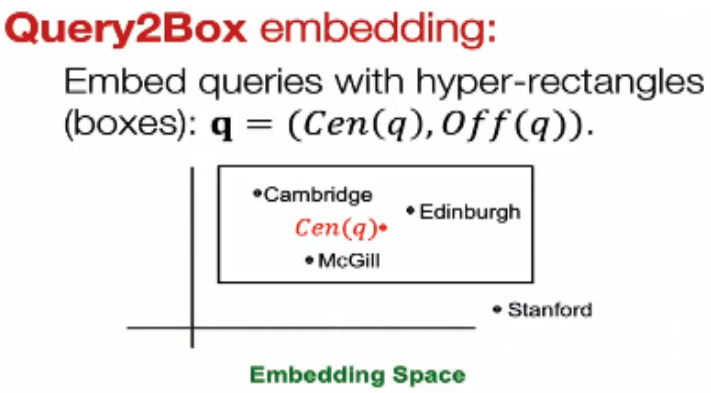


图 10 ： 查询框嵌入方法

以“所有加拿大图灵奖得主都从哪里毕业?”（Where did Canadian citizens with Turing Award graduate?）为例，用知识图谱来学习实体的嵌入。首先，从”Turing Award”节点开始，应用一个表示学习的神经网络算

子，把“Turing Award”嵌入进去，通过“Turing Award”框交叉变换可以覆盖所有图灵奖得主；然后，对于“Canadian”节点也通过神经网络算子进行转化可以覆盖所有加拿大人；最后，通过图灵奖得主框和加拿大人框的交叉点来确定加拿大的图灵奖得主的毕业院校（图 11）。通过实例表明，基于查询框的方法可以对知识图谱进行任意逻辑的推理。Jure 认为图神经网络在因果推理中具有很大的潜力，有望成为人工智能领域的新拐点。

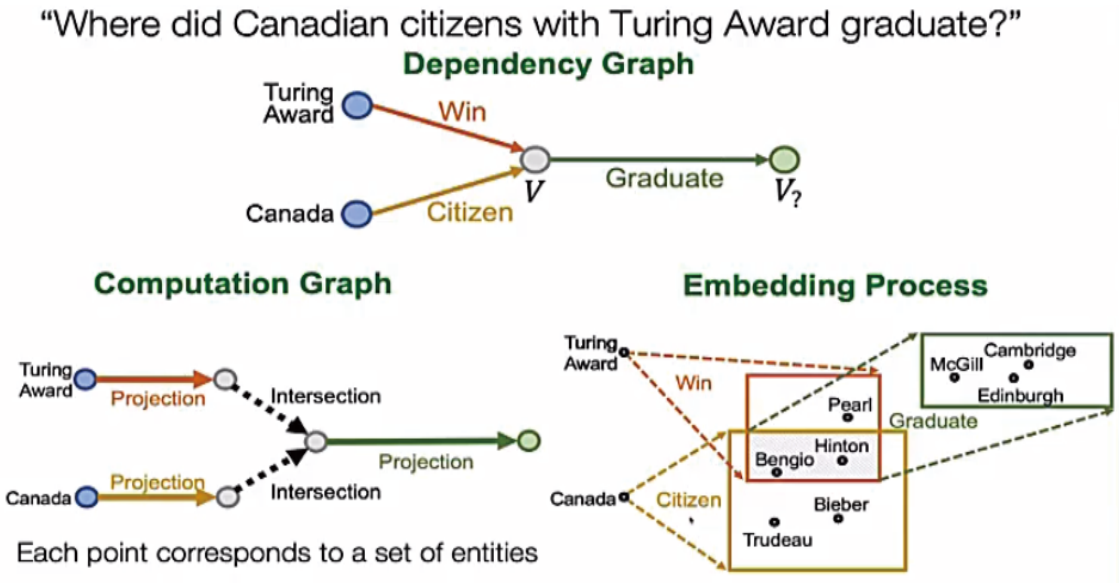


图 11 ： GNNs 在知识图谱推理中的应用

四、结语

演讲最后，Jure 介绍了其团队在机器学习社区的开放图基准（图 12）的相关研究及在斯坦福大学网站提供的 github 代码库，这个代码库与现有的深度学习框架（如 Tensorflow 和 pytorch）兼容，代码库中的模型包括数据加载、图形表示、数据分割、评估、跟踪进度及链接预测等，并且代码库中提供了从化学和生物学到社会和信息网络，及知识图谱谱的数据集。相关内容可以在 https://ogb.stanford.edu 网站获取。

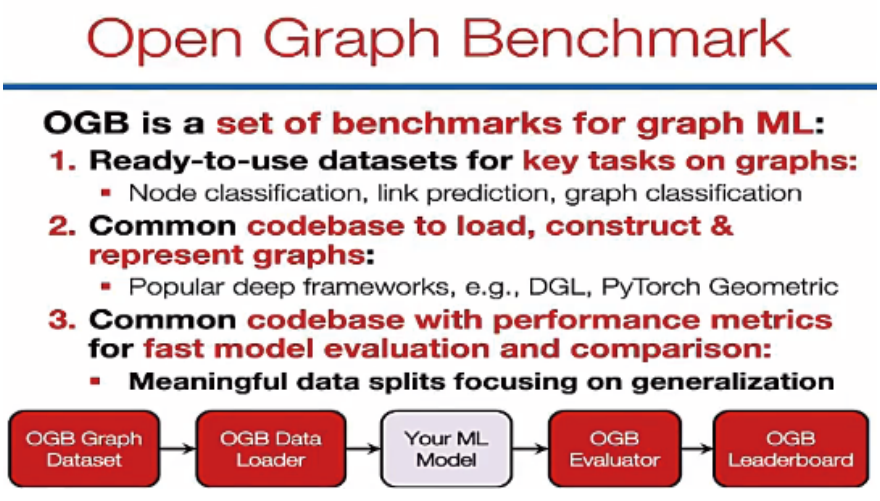


图 12 ： 开放图基准

参考文献：

[1] Scarselli et al.2005.The Graph Neural Network Model. IEEE Transactions on Neural Networks.

[2] https://blog.csdn.net/yyl424525/article/details/100532849?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2

[3] Inductive Representation Learning on Large Graphs. W.Hamilton, R.Ying, J. Leskovec. NIPS, 2017.

[4] Hierarchical Graph Representation Learning with Differentiable Pooling. R.Ying,et al. NeurIPS, 2018.

[5] https://blog.csdn.net/sinat_28978363/article/details/96478415.

[6] How Powerful Are Graph Neural Networks?. K. Xu, et,al ICLR 2019.

[7] Learning to simulate complex Physics with Graph Networks. Sanchez-Gonzalez, Godwin, Pfaff, Ying, Leskovec, Battaglia, ICML 2020.

[8] Query2box:Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. H. Ren, W.Hu, J Leskovec,ICLR 2020.

[9] Embedding Logical Queries on Knowledge Graphs. W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J.Leskovec. NeuIPS, 2018.

[10] Representation Learning on Graphs: Methods and Applications. W. Hamilton, R. Ying, J. Leskovec. IEEE Data Engineering Bulletin, 2017.

[11] Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. Vilnis, et al. ACL 2018.

[12] Open Graph Benchmark: Datasets for Machine Learning on Graphs. W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, J. Leskovec. Arxiv 2020.

[13] Graph Convolutional Neural Networks for Web-Scale Recommender Systems. R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec. KDD, 2018.