《metapath2vec: Scalable Representation Learning for Heterogeneous Networks》

《metapath2vec:异构网络的可扩展表示学习》

摘要：

1. **本文的背景：**研究者们发现基于同质图的图表征学习算法，例如 Deepwalk，Node2Vec 等并不能很好地直接应用到点和边有多个类型的异质图上。而这主要是由于这些同质图算法并不考虑节点的类型，导致它们往往有如下两方面的缺点：1）容易偏向于出现频率高的节点类型；2）容易偏向于连接相对集中的节点。因此，本文旨在解决上述网络异构所带来的特殊挑战，从而进一步扩展图表征学习的应用范围。
2. **本文的贡献：**提出了一种专门处理异质图的图嵌入学习框架——metapath2vec，以及其改进版本 metapath2vec++，有效保持了网络结构在语义上的相关性。
3. **主要创新点：**首次正式定义了 Heterogeneous Network 上的表示学习过程；提出了一种基于元路径的随机游走方法以适用于异构图网络。
4. **实验结果：**本文通过大量的实验证明了本文所提出的方法在各种异构网络挖掘任务中的有效性和可扩展性，如节点分类（相对于基准提高了 35%—319%）和节点聚类（相对于基准提高了 13%—16%）。此外，还证明了 metapath2vec 和 metapath2vec++能自动发现异构网络中不同类型节点之间的内部语义关系，而现有工作是无法发现的。

问题定义：

异构图的定义：

给定一个图$G = (V, E, T)$，对于其中每个节点v和边e都有对应的映射函数：$\phi(v): V \to T_V$，$\varphi(e): E \to T_E$。其中，T_V和T_E分别表示节点和边的类型集合，且$|T_V| + |T_E| > 2$。即，图网络的节点类型与边类型的总量大于 2。

异构图表示学习的定义：

给定一个异构图G，其任务是学习到所有节点的嵌入表示$X \in \mathbb{R}^{|V| \times d}$，其中$d \ll |V|$，而嵌入表示$X$能有效捕捉到不同类型节点的结构和语义关系。

注意：虽然异构图上的节点类型不同，但是不同类型的节点的嵌入向量会映射到同一个特征空间中，即不同类型的节点的嵌入向量的维度都是d。此外，由于网络异构性的存在，传统的基于同构网络的节点嵌入表征方法很难有效地直接应用在异构网络上。

异质 skip-gram：

传统的 skip-gram 模型：

给定一个网络$G = (V, E)$，其目标是使网络的局部结构概率最大化，即：

$$\arg max_{\theta} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\theta)$$

其中，$N(v)$代表节点v的上下文节点，而这种节点的定义方式有多种，比如$DeepWalk$中利用随机游走得到的上下文节点。而$p(c|v;\theta)$代表在已知节点v的情况下，上下文节点c存在的概率，这种概率通常用向量内积+$softmax$来实现：

$$p(c|v;\theta) = \frac{e^{X_c \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}}$$

而为了将异构网络的结构引入到上述的 skip-gram 方法中，本文提出了异质 skip-gram 的概念，即在给定节点v的情况下，最大化其异质上下文节点$N_t(v)$出现的概率，即：

$$\arg max_{\theta} \prod_{v \in V} \prod_{t \in T_V} \prod_{c_t \in N_t(v)} \log p(c_t|v;\theta)$$

其中，$N_t(v)$代表节点v的第t种类型的上下文节点；$p(c_t|v;\theta)$代表在已知节点v的情况下，上下文节点c_t存在的概率，而这种概率在本文也是用向量内积+$softmax$来实现的：

$$p(c_t|v;\theta) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}}$$

此外，与传统的 skip-gram 中一致，为了解决每次更新时都要对所有节点进行$softmax$计算的问题，异质 skip-gram 也引入了负采样策略。给定负样本数M，则优化目标可进一步表示为：

$$O(X) = \log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)} [\log \sigma(-X_{u^m} \cdot X_v)]$$

其中，$P(u)$代表负采样中样本的预定义分布；激活函数$\sigma(x)=\frac{1}{1+e^{-x}}$；$u^m$代表从负样本集中抽取的负样本。

基于元路径的随机游走：

同$DeepWalk$和$node2vec$等方法类似，$metapath2vec$也需要得到节点的游走序列，然后使用上述异质 skip-gram 模型进行优化。而为了在异构图上得到有效的游走序列，本文提出了 Meta-Path-Based Random Walks，即基于元路径的随机游走。

具体来讲，一个元路径方案$\mathcal{P}$被定义为：

$$V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots V_t \xrightarrow{R_t} V_{t+1} \dots \xrightarrow{R_{l-1}} V_l$$

注意：元路径中相邻节点的类型是不一样的，即V_t和V_{t+1}属于不同类型的节点，而R_t表示两个节点之间的连接关系。

因此，节点V_1和V_l之间的复合关系R可进一步定义为：

$$R = R_1 \circ R_2 \circ \dots \circ R_{l-1}$$

此外，与$node2vec$方法中一致，基于元路径的随机游走也给出了每一步的转移概率——给定一个异构网络$G=(V,E,T)$和一个元路径方案$\mathcal{P}: V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots V_t \xrightarrow{R_t} V_{t+1} \dots \xrightarrow{R_{l-1}} V_l$，步骤$i$处的转移概率定义如下：

$$p\left(v^{i+1}|v_t^i,\mathcal{P}\right)=\begin{cases}\dfrac{1}{|N_{t+1}(v_t^i)|} & \left(v^{i+1},v_t^i\right)\in E, \phi\left(v^{i+1}\right)=t+1\\ 0 & \left(v^{i+1},v_t^i\right)\in E, \phi\left(v^{i+1}\right)\neq t+1\\ 0 & \left(v^{i+1},v_t^i\right)\notin E\end{cases}$$

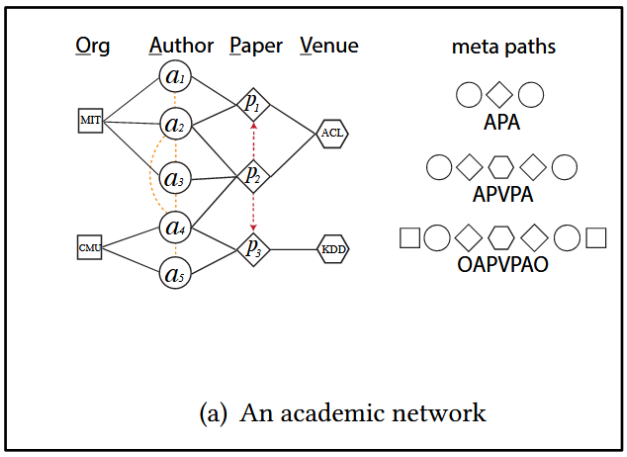
假设当前节点为v_t^i，那么下一个节点v^{i+1}的选择策略为：

1. 如果v_t^i与v^{i+1}之间存在边，且v^{i+1}的节点类型与元路径方案$\mathcal{P}$中的定义一致，那么这些节点被选中的概率是均等的，也就是$1/$可选节点个数。
2. 如果v_t^i与v^{i+1}之间存在边，但v^{i+1}的节点类型与元路径方案$\mathcal{P}$中的定义不一致，那么则不会选择该节点，即概率为 0。
3. 如果v_t^i与v^{i+1}之间不存在边，则不会选择该节点。

而除了上述三点原则外，元路径还需遵循对称原则，也就是路径中第一个节点V_1和最后一个节点V_l的类型应该一致，即：

$$p(v^{i+1}|v_t^i) = p(v^{i+1}|v_1^i), \qquad if\ t = l$$

于是，基于元路径的随机游走策略就能确保不同类型节点之间的语义关系能够被恰当地整合到 skip-gram 中。以下图为例：



(a) An academic network

假设在一个学术网络中，A表示作者节点，P表示论文，O表示作者所属组织，V表示期刊会议。而根据上述元路径的定义，可以有以下几种简单的元路径：

1. APA：一篇论文中两个作者的合著关系。
2. $APVPA$：两个作者(A)在同一期刊会议(V)上发表了论文。
3. $OAPVPAO$：两个分别所属组织O的作者(A)在同一期刊会议(V)上发表了论文。

在传统的随机游走过程中，从节点CMU过渡到节点a_4上的 walker 的下一步可以是它周围所有类型的节点，即a_2、a_3、a_5、p_2、p_3和CMU。然而，在元路径方案“$OAPVPAO$”下，下一步选择的节点则会偏向于论文节点P（严格遵循此路径的语义）。

metapath2vec++：

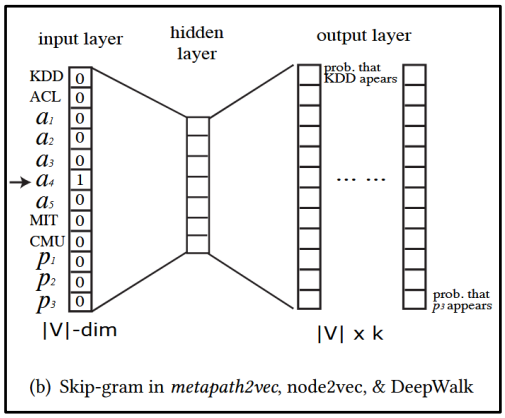
实际上，上述异质 skip-gram 模型和基于元路径的随机游走方案就构成了本文所提出的$metapath2vec$方法，而所谓的$metapath2vec++$方案则是在$metapath2vec$的基础上对优化目标做出了进一步的改进，具体细节如下：

已知，$metapath2vec$中的优化目标为：

$$\arg max_\theta \prod_{v\in V} \prod_{t\in T_V} \prod_{c_t\in N_t(v)} \log p(c_t|v;\theta)$$

其中：

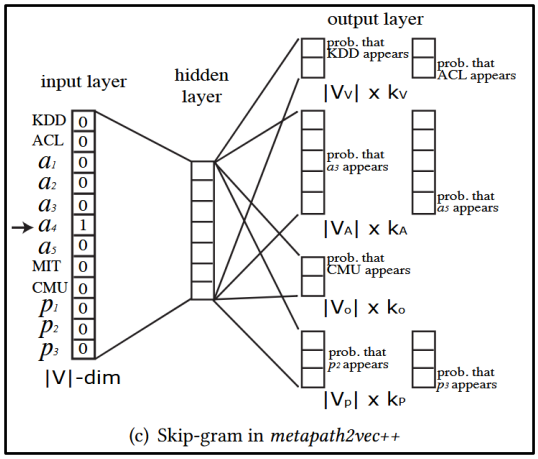
$$p(c_t|v;\theta)=\frac{e^{X_{c_t}\cdot X_v}}{\sum_{u\in V}e^{X_u\cdot X_v}}$$



(b) Skip-gram in *metapath2vec*, node2vec, & DeepWalk

根据要求，我们需要构造节点的邻居函数$N_t(v)$（上下文邻居）。而从元路径游走策略可知，$metapath2vec$中是根据元路径方案$\mathcal{P}$来确定的。然后，在进行$softmax$计算时，我们会考虑所有类型的节点，而这种方式忽略了$softmax$中节点的类型信息，如上图所示。即，在进行负采样时，$metapath2vec$不会区分节点类型，所有类型的节点都有可能被采样到，即上面式子中的$u\in V$，而作者认为这是不合理的。因此，改进后的结果如下：

$$p(c_t|v;\theta)=\frac{e^{X_{c_t}\cdot X_v}}{\sum_{u_t\in V_t}e^{X_{u_t}\cdot X_v}}$$



(c) Skip-gram in *metapath2vec++*

这样，就使得我们在计算节点向量间的内积并进行$softmax$时，会根据不同类型的节点的上下文c_t进行归一化，如上图所示。因此，$metapath2vec++$的优化目标就可以进一步定义为：

$$O(X)=\log\sigma(X_{c_t}\cdot X_v)+\sum_{m=1}^{M}\mathbb{E}_{u_t^m\sim P_t(u_t)}\left[\log\sigma\left(-X_{u_t^m}\cdot X_v\right)\right]$$

从这里我们也可以看出$metapath2vec++$被提出的根本原因：之前所有方法都是基于同质图，同质图中所有节点类型都一致，因此进行负采样时不需要额外考虑节点类型，而异质图中节点类型不一致，因此自然而然地就会想到负采样时区分节点类型。

metapath2vec++算法伪代码：

Input: The heterogeneous information network $G = (V, E, T)$, a meta-path scheme $\mathcal{P}$, #walks per node w, walk length l, embedding dimension d, neighborhood size k
Output: The latent node embeddings $\mathbf{X} \in \mathbb{R}^{|V| \times d}$

initialize $\mathbf{X}$;
for $i = 1 \rightarrow w$ **do**
  for $v \in V$ **do**
    MP = MetaPathRandomWalk($G, \mathcal{P}, v, l$) ;
    $\mathbf{X}$ = HeterogeneousSkipGram($\mathbf{X}, k, MP$) ;
  end
end
return $\mathbf{X}$;

MetaPathRandomWalk($G, \mathcal{P}, v, l$)
$MP[1] = v$;
for $i = 1 \rightarrow l-1$ **do**
  draw u according to Eq. 3 ;
  $MP[i+1] = u$;
end
return MP ;

HeterogeneousSkipGram($\mathbf{X}, k, MP$)
for $i = 1 \rightarrow l$ **do**
  $v = MP[i]$;
  for $j = max(0, i-k) \rightarrow min(i+k, l)$ & $j \neq i$ **do**
    $c_t = MP[j]$;
    $X^{new} = X^{old} - \eta \cdot \frac{\partial O(\mathbf{X})}{\partial X}$ (Eq. 7) ;
  end
end

ALGORITHM 1: The *metapath2vec++* Algorithm.

实验结果及分析：

数据集：

两个数据集——Aminer、DBIS，每个数据集中节点的种类为 3。

Baselines：

DeepWalk、node2vec、LINE、PTE、Spectral Clustering、Graph Factorization

多个任务：

节点分类（Multi-Class Classification）、节点聚类（Node Clustering）、相似性（Case Study：Similarity Search）、可视化（Case Study：Visualization）、参数实验（Scalability）

实验参数：

（1） 每个节点的随机游走次数w：1000；

（2） 每次随机游走的长度l：100；

（3） 嵌入向量维度d：128；

（4） 邻域大小k：7；

（5） 负采样大小：5；

（6） 学术网络中涉及的元路径方案有APA和$APVPA$。

节点分类实验结果：

Table 2: Multi-class venue node classification results in AMiner data.

Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.0723	0.1396	0.1905	0.2795	0.3427	0.3911	0.4424	0.4774	0.4955	0.4457
	LINE (1st+2nd)	0.2245	0.4629	0.7011	0.8473	0.8953	0.9203	0.9308	0.9466	0.9410	0.9466
	PTE	0.1702	0.3388	0.6535	0.8304	0.8936	0.9210	0.9352	0.9505	0.9525	0.9489
	metapath2vec	0.3033	0.5247	0.8033	0.8971	0.9406	0.9532	0.9529	0.9701	0.9683	0.9670
	metapath2vec++	0.3090	0.5444	0.8049	0.8995	0.9468	0.9580	0.9561	0.9675	0.9533	0.9503
Micro-F1	DeepWalk/node2vec	0.1701	0.2142	0.2486	0.3266	0.3788	0.4090	0.4630	0.4975	0.5259	0.5286
	LINE (1st+2nd)	0.3000	0.5167	0.7159	0.8457	0.8950	0.9209	0.9333	0.9500	0.9556	0.9571
	PTE	0.2512	0.4267	0.6879	0.8372	0.8950	0.9239	0.9352	0.9550	0.9667	0.9571
	metapath2vec	0.4173	0.5975	0.8327	0.9011	0.9400	0.9522	0.9537	0.9725	0.9815	0.9857
	metapath2vec++	0.4331	0.6192	0.8336	0.9032	0.9463	0.9582	0.9574	0.9700	0.9741	0.9786

Table 3: Multi-class author node classification results in AMiner data.

Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.7153	0.7222	0.7256	0.7270	0.7273	0.7274	0.7273	0.7271	0.7275	0.7275
	LINE (1st+2nd)	0.8849	0.8886	0.8911	0.8921	0.8926	0.8929	0.8934	0.8936	0.8938	0.8934
	PTE	0.8898	0.8940	0.897	0.8982	0.8987	0.8990	0.8997	0.8999	0.9002	0.9005
	metapath2vec	0.9216	0.9262	0.9292	0.9303	0.9309	0.9314	0.9315	0.9316	0.9319	0.9320
	metapath2vec++	0.9107	0.9156	0.9186	0.9199	0.9204	0.9207	0.9207	0.9208	0.9211	0.9212
Micro-F1	DeepWalk/node2vec	0.7312	0.7372	0.7402	0.7414	0.7418	0.7420	0.7419	0.7420	0.7425	0.7425
	LINE (1st+2nd)	0.8936	0.8969	0.8993	0.9002	0.9007	0.9010	0.9015	0.9016	0.9018	0.9017
	PTE	0.8986	0.9023	0.9051	0.9061	0.9066	0.9068	0.9075	0.9077	0.9079	0.9082
	metapath2vec	0.9279	0.9319	0.9346	0.9356	0.9361	0.9365	0.9365	0.9365	0.9367	0.9369
	metapath2vec++	0.9173	0.9217	0.9243	0.9254	0.9259	0.9261	0.9261	0.9262	0.9264	0.9266

如图所示，可以看到 metapath2vec 和 metapath2vec++相比基准算法表现要更好。

节点聚类实验结果：

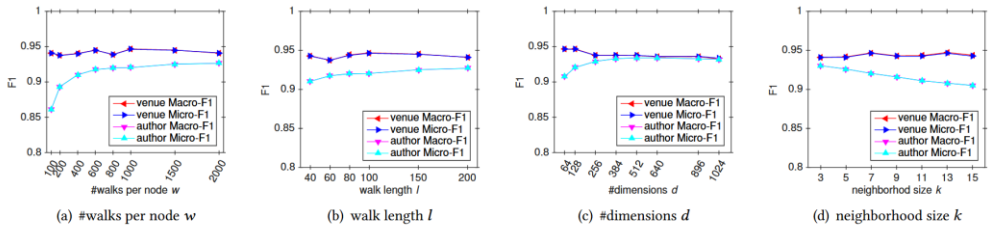
说明：采用 NMI 指标（归一化互信息）来衡量，值越大越好。而聚类结果是在得到节点表征后运行k均值算法得到的。

Table 4: Node clustering results (NMI) in AMiner data.

methods	venue	author
DeepWalk/node2vec	0.1952	0.2941
LINE (1st+2nd)	0.8967	0.6423
PTE	0.9060	0.6483
metapath2vec	0.9274	0.7470
metapath2vec++	0.9261	0.7354

如图所示，可以看到 metapath2vec 和 metapath2vec++相比基准算法表现要更好。

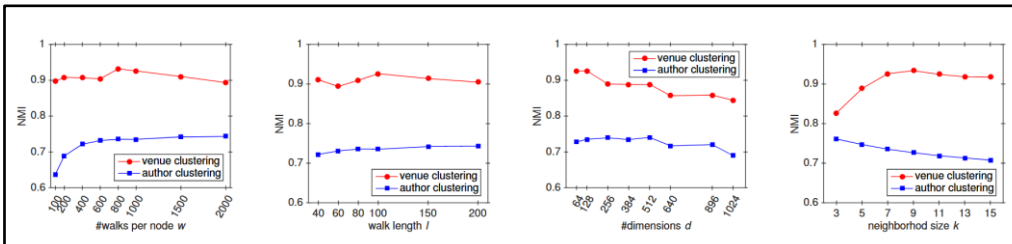
参数敏感性分析：



(a) #walks per node w (b) walk length l (c) #dimensions d (d) neighborhood size k

Figure 3: Parameter sensitivity in multi-class node classification. 50% as training data and the remaining as test data.

如图所示，可以看到在节点分类任务中，遍历次数w在 1000 时得分最高，遍历深度l在 100 时得分最高，嵌入维度d在 128 时效果最好，而邻域大小k需要视情况而定。



(a) #walks per node w (b) walk length l (c) #dimensions d (d) neighborhood size k

Figure 4: Parameter sensitivity in clustering.

如图所示，可以看到在节点聚类任务中，遍历次数w在 800 时效果最好，遍历深度l在 100 时效果最高，嵌入维度d在 128 时效果最好，而邻域大小k仍需要视情况而定。

可视化结果：

下图是会议节点$t-SNE$降维的结果，相同类型节点相同颜色。

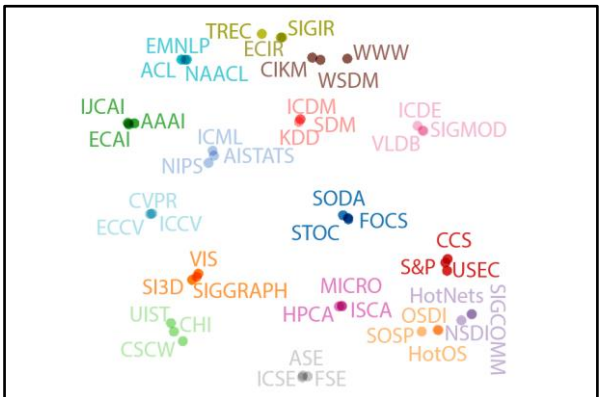


Figure 5: 2D t-SNE projections of the 128D embeddings of 48 CS venues, three each from 16 sub-fields.

会议节点相似性排名（越相关越靠上）：

Table 5: Case study of similarity search in AMiner Data

Rank	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
0	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
1	EMNLP	ICML	AAAI	ECCV	STOC	TOCS	HPCA	CCS	TOSEM	TOG	CCR	CSCW	SDM	PVLDB	ECIR	WSDM
2	NAACL	AISTATS	AI	ICCV	SICOMP	OSDI	MICRO	NDSS	FSE	SI3D	HotNets	TOCHI	TKDD	ICDE	CIKM	CIKM
3	CL	JMLR	JAIR	IJCV	SODA	HotOS	ASPLOS	USENIX S	ASE	RT	NSDI	UIST	ICDM	DE Bull	IR J	TWEB
4	CoNLL	NC	ECAI	ACCV	A-R	SIGOPS E	PACT	ACSAC	ISSTA	CGF	CoNEXT	DIS	DMKD	VLDBJ	TREC	ICWSM
5	COLING	MLJ	KR	CVIU	TALG	ATC	ICS	JCS	E SE	NPAR	IMC	HCI	KDD E	EDBT	SIGIR F	HT
6	IJCNLP	COLT	AI Mag	BMVC	ICALP	NSDI	HiPEAC	ESORICS	MSR	Vis	TON	MobileHCI	WSDM	TODS	ICTIR	SIGIR
7	NLE	UAI	ICAPS	ICPR	ECCC	OSR	PPOPP	TISS	ESEM	JGT	INFOCOM	INTERACT	CIKM	CIDR	WSDM	KDD
8	ANLP	KDD	CI	EMMCVPR	TOC	ASPLOS	ICCD	ASIACCS	A SE	VisComp	PAM	GROUP	PKDD	SIGMOD R	TOIS	TIT
9	LREC	CVPR	AIPS	T on IP	JAIG	EuroSys	CGO	RAID	ICPC	GI	MobiCom	NordiCHI	ICML	WebDB	IPM	WISE
10	EACL	ECML	UAI	WACV	ITCS	SIGCOMM	ISLPED	CSFW	WICSA	CG	IPTPS	UbiComp	PAKDD	PODS	AIRS	WebSci

可扩展性：

下图是显示当可用资源（线程）变化，模型性能指标的提升效果。而我们可以看到模型的可扩展性不错，基本呈线性。

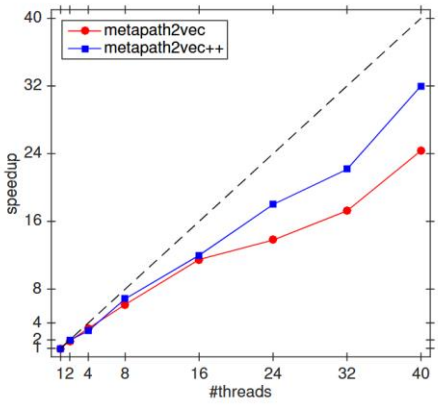


Figure 6: Scalability of *metapath2vec* and *metapath2vec++*.

结论：

本文提出了异构图嵌入模型$metapath2vec$和$metapath2vec++$。具体而言，首先定义了元路径方案及其对应的随机游走策略，该策略能够捕获不同类型节点及关系的结构和语义相关性。然后，再将随机游走序列输入到本文提出的异质skip-gram中以得到所有类型节点的嵌入表示。而需要注意的是，$metapath2vec$中skip-gram的负采样策略与同质skip-gram一致，考虑了所有节点；而$metapath2vec++$中skip-gram在进行负采样时只考虑与上下文节点同类型的节点。此外，大量实验还表明，$metapath2vec$和$metapath2vec++$学习的潜在特征表示能够改进各种异构网络挖掘任务，如相似性搜索、节点分类和聚类等。

对本文的感悟：

这篇论文首次将random walk + skip-gram的框架拓展到了异构图的研究领域，并且还有力证明了基于异构图的随机游走算法能够表达不同类型节点之间的语义和结构关联。总之，本文不仅拓展了深度学习研究图的类型，还进一步将深度学习领域的模型引入到了更广泛的图学习任务之中，也为后续类似HAN、GTN模型的提出创造了坚实的理论基础。