



Deep learning on the graphs

Лебедев Михаил

Graph autoencoders

- Техника использования представления узлов графа со снижением размерности для обучения нейросетей.

Основные методы: SAE, SDNE, DNGR, GC-MC, DRNE, Graph2Gauss(G2G), VGAE, DVNE, ARGGA, ARGVA, NetRA

Краткий обзор методов Graph autoencoders

Method	Type	Objective	Scalability	Node Features	Other Characteristics
SAE [102]	AE	L2-Reconstruction	Yes	No	-
SDNE [103]	AE	L2-Reconstruction + Laplacian Eigenmaps	Yes	No	-
DNGR [104]	AE	L2-Reconstruction	No	No	-
GC-MC [105]	AE	L2-Reconstruction	Yes	Yes	GCN Encoder
DRNE [106]	AE	Recursive Reconstruction	Yes	No	LSTM Encoder
G2G [107]	AE	KL + Ranking	Yes	Yes	Nodes as distributions
VGAE [108]	VAE	Pairwise Probability of Reconstruction	No	Yes	GCN Encoder
DVNE [109]	VAE	Wasserstein + Ranking	Yes	No	Nodes as distributions
ARGA/ARVGA [110]	AE/VAE	L2-Reconstruction + GAN	Yes	Yes	GCN Encoder
NetRA [111]	AE	Recursive Reconstruction + Laplacian Eigenmaps + GAN	Yes	No	LSTM Encoder

Sparse autoencoders (SAE)

- SAE сжимает информацию в вектор низкой размерности и изменяет признаковое описание
- Применяется метод к-средних для разбиения графа на узлы

$$\min_{\Theta} \mathcal{L}_2 = \sum_{i=1}^N \left\| \mathbf{P}(i, :) - \hat{\mathbf{P}}(i, :) \right\|_2$$

$$\hat{\mathbf{P}}(i, :) = \mathcal{G}(\mathbf{h}_i), \mathbf{h}_i = \mathcal{F}(\mathbf{P}(i, :)),$$



Variational Autoencoders

- Основное отличие от AutoEncoders - является комбинацией метода сокращения размерности вектора и генеративной моделью.

Reinforcement Learning

- Обучение с подкреплением
- Модель при обучении взаимодействует со средой.
- Основные методы: GCPN, MoIGAN, GTPM, GAN, Deeppath, MINERVA

Краткий обзор методов RL

Method	Task	Actions	Rewards	Scalability
GCPN [121]	Graph generation	Link prediction	GAN + domain knowledge	No
MolGAN [122]	Graph generation	Generate the whole graph	GAN + domain knowledge	No
GTPN [123]	Chemical reaction prediction	Predict node pairs and new bonds	Prediction results	No
GAM [124]	Graph classification	Predict graph labels and select the next node	Classification results	Yes
DeepPath [125]	Knowledge graph reasoning	Predict the next node of the reasoning path	Reasoning results + diversity	Yes
MINERVA [126]	Knowledge graph reasoning	Predict the next node of the reasoning path	Reasoning results	Yes

Graph adversarial methods

- Методы разделяются на два типа
- Методы Adversarial training, пример:
Regularization (for GAE), Regularization for network embedding
- Методы Adversarial attack, пример:
Targeted attacks of graph structures and node attributes, Non-targeted attacks of graph structures

Adversarial training

- Основная идея – обучение с построением двух связанных моделей: Дескриминатора и генератора. Генератор подаёт на вход дескриминатору элементы. Задача дескриминатора – отличать реальные данные графа от сгенерированных.

Построение итоговой модели происходит с помощью минимакса.

- $$\min_{\mathcal{G}} \max_{\mathcal{D}} \sum_{i=1}^N \left(\mathbb{E}_{v \sim p_{graph}(\cdot | v_i)} [\log \mathcal{D}(v, v_i)] + \mathbb{E}_{v \sim \mathcal{G}(\cdot | v_i)} [\log (1 - \mathcal{D}(v, v_i))] \right)$$

Graph adversarial attacks

Nettack

При работе с моделью модифицируется граф и признаковое описание

$$\begin{aligned} & \operatorname{argmax}_{(\mathbf{A}', \mathbf{F}^{V'}) \in \mathcal{P}} \max_{c \neq c_{true}} \log \mathbf{Z}_{v_0, c}^* - \log \mathbf{Z}_{v_0, c_{true}}^* \\ & s.t. \mathbf{Z}^* = \mathcal{F}_{\theta^*}(\mathbf{A}', \mathbf{F}^{V'}), \theta^* = \operatorname{argmin}_{\theta} \mathcal{L}_{\mathcal{F}}(\mathbf{A}', \mathbf{F}^{V'}) \end{aligned}$$



Источники

- <https://arxiv.org/pdf/1812.04202.pdf>