Solving QAP using k-order graph networks

Waïss Azizian & Romain Laurent

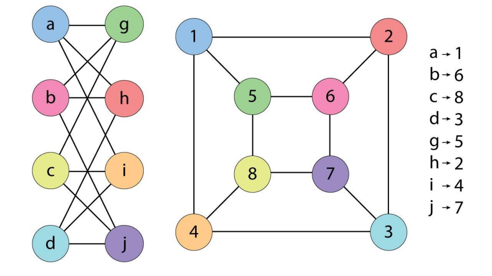
24 janvier 2020

Quadratic assignment problem

Graph isomorphism

$G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a bijection $V_1 \longrightarrow V_2$ which preserves edges.

It captures an informal notion of structure and has many applications in graph analysis.



Relaxation of graph isomorphism : QAP (Nowak et al., 2018)

Intuitively, the quadratic assignment problem is a 'noisy' version of graph isomorphism.

Formally, if A_1 & A_2 are adjacency matrices, it consists in solving :

$$\min \|P^{-1}A_1P - A_2\| \quad \text{s.t. } P \text{ is a permutation matrix}$$

$P^{-1}A_1P$ is the adjacency matrix of the permuted graph.

This is a NP-hard problem which also happens to be hard to approximate.

Synthetic datasets for QAP (Nowak et al., 2018)

We can easily generate synthetic datasets for the standard QAP. It makes both the training & the testing of neural networks much easier.

For W an adjacency matrix and N a binary symmetric random matrix whose edges are drawn i.i.d. from a Bernoulli distribution with parameter p (Erdős–Rényi noise), we have :

$$W_{noise} = W * (\mathbb{1} - N) + (\mathbb{1} - W) * N$$

Intuitively, it flips edges of W uniformly randomly with probability p.

In this example, the permutation matrix to predict is Id.

Siamese encoder and embeddings (Nowak et al., 2018)

$$
\begin{tikzcd}
G_1 \in \mathbb{R}^{n^2 \times a} \arrow[r, "\text{GNN } \mathcal{N}"] & E_1 \in \mathbb{R}^{n \times b} \arrow[dr] & \\
& & E_1 E_2^T \in \mathbb{R}^{n^2} \\
G_2 \in \mathbb{R}^{n^2 \times a} \arrow[r, "\text{GNN } \mathcal{N}"] & E_2 \in \mathbb{R}^{n \times b} \arrow[ur] &
\end{tikzcd}
$$

We use a unique GNN $\mathcal{N}$ to produce our embeddings.

What we want : if we permute the nodes of G_i, E_i must be permuted too.

k-order graph neural networks

Equivariant neural networks (Maron et al., 2019a)

We want a network $F : \mathbb{R}^{n^k \times a} \to \mathbb{R}^{n^{k'} \times b}$ which is equivariant :

$$\forall \sigma \in \mathfrak{S}_n, \quad \forall X \in \mathbb{R}^{n^k \times a} \quad F(\sigma \cdot X) = \sigma \cdot F(X)$$

where

$$(\sigma \cdot X)_{i_1,\ldots,i_k,j} = X_{\sigma^{-1}(i_1),\ldots,\sigma^{-1}(i_k),j}$$

Equivariant neural networks (Maron et al., 2019a)

We want a network $F : \mathbb{R}^{n^k \times a} \to \mathbb{R}^{n^{k'} \times b}$ which is equivariant :

$$\forall \sigma \in \mathfrak{S}_n, \quad \forall X \in \mathbb{R}^{n^k \times a} \quad F(\sigma \cdot X) = \sigma \cdot F(X)$$

where

$$(\sigma \cdot X)_{i_1,\ldots,i_k,j} = X_{\sigma^{-1}(i_1),\ldots,\sigma^{-1}(i_k),j}$$

To achieve this : composition of equivariant layers.

Blocks (Maron et al., 2019a)

$$
\begin{tikzcd}
& \mathbb{R}^{n^2 \times b} \arrow[dr] & & \\
X \in \mathbb{R}^{n^2 \times a} \arrow[ur, "m_1"] \arrow[r, "m_2"] \arrow[dr, "m_3"] & \mathbb{R}^{n^2 \times b} \arrow[r] & \text{Multiply} : \mathbb{R}^{n^2 \times b} \arrow[r] & \text{Concat} : \mathbb{R}^{n^2 \times (b+c)} \\
& \mathbb{R}^{n^2 \times c} \arrow[urr, bend right] & &
\end{tikzcd}
$$

- MLPs $m : \mathbb{R}^a \longrightarrow \mathbb{R}^b$ applied edge-wise :

$$m(X)_{i,j,:} = m(X_{i,j})_:$$

- Matrix multiplication feature-wise :

$$(m_1(X) \times m_2(X))_{:,:,k} = m_1(X)_{:,:,k} \times m_2(X)_{:,:,k}$$

$$\begin{tikzcd}
 & \mathbb{R}^{n^2\times b} \arrow[dr] & & \\
X \in \mathbb{R}^{n^2\times a} \arrow[ur, "m_1"] \arrow[r, "m_2"] \arrow[dr, "m_3"'] & \mathbb{R}^{n^2\times b} \arrow[r] & \text{Multiply} : \mathbb{R}^{n^2\times b} \arrow[r] & \text{Concat} : \mathbb{R}^{n^2\times(b+c)} \\
 & \mathbb{R}^{n^2\times c} \arrow[urr, bend right] & &
\end{tikzcd}$$

- MLPs $m : \mathbb{R}^a \longrightarrow \mathbb{R}^b$ applied edge-wise :

$$m(X)_{i,j,:} = m(X_{i,j})_:$$

- Matrix multiplication feature-wise :

$$(m_1(X) \times m_2(X))_{:,:,k} = m_1(X)_{:,:,k} \times m_2(X)_{:,:,k}$$

$\rightarrow$ Equivariant

What about message-passing networks ? (Xu et al., 2019)

$$h_v^t = \mathrm{COMBINE}\left(h_v^{t-1}, \mathrm{AGGREGATE}\left(h_u^{t-1} : u \sim v\right)\right)$$

$\rightarrow$ Strictly less discriminative than a tensor network based on blocks. (Maron et al., 2019a)

Our experiments

Architecture

$$G \in \mathbb{R}^{n^2 \times 2} \xrightarrow{\text{Block}_1} \mathbb{R}^{n^2 \times a} \xrightarrow{\text{Block}_2} \mathbb{R}^{n^2 \times a} \xrightarrow{\text{Suffix}} \mathbb{R}^{n \times b}$$

- Blocks : 3 MLPs with 3 layers each
- Features : $16 \leq a \leq 64$
- ~ 1k parameters
- Suffix : equivariant layer with no parameters based on max pooling. (using insight from (Maron et al., 2019b))
- Encoding a graph as $\mathbb{R}^{n^2 \times 2}$:

  $G_{:,:,0}$: adjacency matrix of the graph
  $G_{:,:,1}$: only the degrees of the nodes on the diagonal

Graphs distributions

We tested on three graphes distributions :

Erdős–Rényi : each edge added with some probability p

d**-Regular :** each node has d neighbours
$\rightarrow$ considered as hard examples

Barabási–Albert : models real-world graphs such as social networks

Results

	SDP	Msg. Passing	Ours
Erdős–Rényi (noise = 0.03)	1	0.93	0.99
Regular (noise = 0.05)	0.03	0.52	0.92
Barabási-Albert (noise = 0.05)			0.93

SDP from (Peng et al., 2010), message passing from (Nowak et al., 2018).

- Graphs : $n = 50$, density = 0.2
- Training set : 20000 samples
- Validation : grid search on 2000 samples

Conclusion

- Tensor networks : perform much better on QAP
- We tried using this for pretraining, but for now it does not work...

Références

H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32 : Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 2153–2164, 2019a. URL `http://papers.nips.cc/paper/8488-provably-powerful-graph-networks`.

H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman. Invariant and equivariant graph networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL `https://openreview.net/forum?id=Syx72jC9tm`.

A. Nowak, S. Villar, A. S. Bandeira, and J. Bruna. Revised note on learning quadratic assignment with graph neural networks. In *2018 IEEE Data Science Workshop, DSW 2018, Lausanne, Switzerland, June 4-6, 2018*, pages 229–233. IEEE, 2018. doi : 10.1109/DSW.2018.8439919. URL `https://doi.org/10.1109/DSW.2018.8439919`.

J. Peng, H. D. Mittelmann, and X. Li. A new relaxation framework for quadratic assignment problems based on matrix splitting. *Math. Program. Comput.*, 2