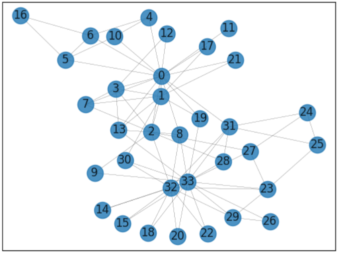
Properties captured in graph embeddings

Supervised by Rémy Cazabet and Christine Largeron

Rémi Vaudaine

Definitions

Goal of a graph embedding



5	0.613927942469497	-0.337499610753705
6	0.765099657130611	-0.325043834286658
7	0.720790603013097	-0.439167457122584
8	0.045578316656147	-0.359810996383625
9	-0.063747652435071	0.052527845139578
10	-0.385133828225982	-0.111182970967994
11	0.671119611819668	-0.192656205097888
12	0.493916367714484	-0.595658004387579
13	0.328366239448059	-0.521437327828054
14	0.061241787431519	-0.06910576801863
15	-0.543298334342232	0.399074626697218
16	-0.644806173269842	0.341737091063604
17	1	-0.48299402379058
18	0.406128251905145	-0.132931147263988
19	-0.622915054429429	0.235992439397208
20	-0.077043524232568	-0.163995169677262
21	-0.660854344172839	0.136531996328351
22	0.175053377101955	-0.513238500329956
23	-0.587621299455263	0.042550222382883
24	-0.241301442508169	0.480424196313792
25	0.128237985293679	0.53987278195411
26	-0.003951050195865	0.608790097893964
27	-0.53194916257256	0.576173378792525

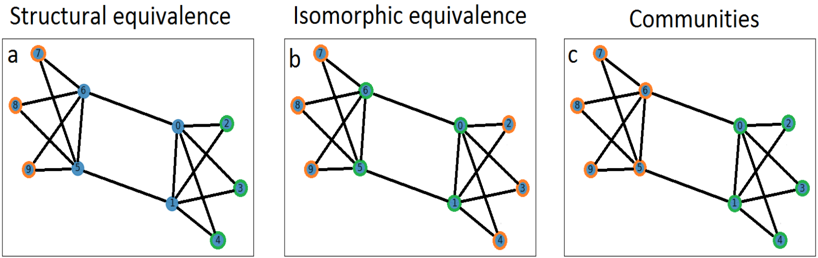
Goal of a graph embedding

Apply machine learning tools:

- link prediction
- node classification
- clustering
- visualization

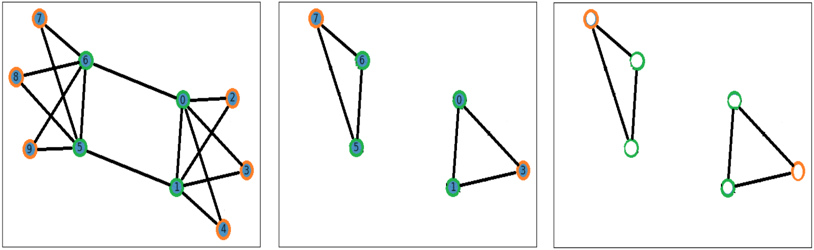
How?

Encoding some properties of the graph in the embedding: graph similarity



How?

Role of nodes - isomorphic equivalence



How?

Decoding from the embedding: embedding similarity

- L1-norm
- L2-norm (euclidean distance)
- Dot-product (cosine distance)

Embeddings methods

- Based on matrix decomposition

Laplacian Eigenmaps

- 2000, Belkin and Niyogi
- $$f(Y) = tr(Y^T LY) = \sum_{i,j} |Y_i - Y_j|^2 A_{ij}$$
- Graph similarity: neighborhood
- Embedding similarity: euclidean distance

Locally linear embedding

- 2001, Roweis and Saul
- $$f(Y) = \sum_i |Y_i - \sum_j A_{ij} Y_j|^2$$
- Graph similarity: neighborhood
- Embedding similarity: euclidean distance

HOPE

- 2016, Ou et al.
- $$f(Y) = ||S - YY^T||_F^2$$
- Graph similarity: similarity matrix S (Katz-Index)
- Embedding similarity: cosine distance

SVD

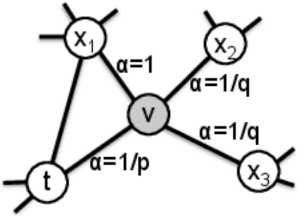
- Custom embedding
- Graph similarity: second-order proximity
- Embedding similarity: euclidean distance

Embeddings methods

- Based on random walks

Node2vec

- 2016, Leskovec & Grover
- $$f(Y) = \sum_{u \in V} \left[log \left(\sum_{i \in N(u)} exp(Y_i Y_u^T) \right) - log\left(Z_u\right) \right]$$
- Graph similarity: co-occurence in a random walk
- Embedding similarity: dot-product



Verse

- 2018, Tsitsulin et al.
- $$L = -\sum_{v \in V} sim_G(v,.) log(sim_E(v,.))$$
- Graph similarity: $sim_G(v,.)$ (Personalized Page Rank)
- Embedding similarity: dot-product

Embeddings methods

- Miscellaneous

Spring layout

- 1991, Fruchterman & Reingold
- N body simulation
- Minimize global energy of the system
- Embedding similarity: euclidean distance

SDNE

- 2016, Wang et al.
- $$f(Y) = ||(\hat{X} - X) \odot B||_F^2 + 2\alpha tr(Y^T LY)$$
  $$+\nu \frac{1}{2} \sum_{k\_layers} (||W^{(k)}||_F^2 + ||\hat{W}^{(k)}||_F^2)$$
- Graph similarity: 1st and 2nd order proximities
- Embedding similarity: euclidean distance

Multi-dimensional scaling

- Standard dimensionality reduction tool
- $$Stress_D\left(y_0, y_1, \ldots, y_{n-1}\right) = \left(\sum_{i \neq j = 0, \ldots, n-1} \left(d_{ij} - ||y_i - y_j||\right)^2\right)^{1/2}$$
- Graph similarity: distance between two nodes
- Embedding similarity: euclidean distance

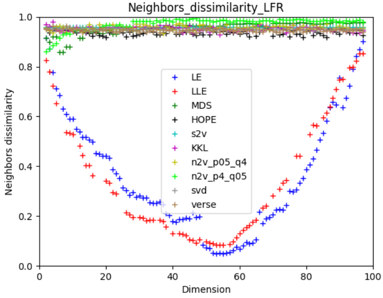
Measures

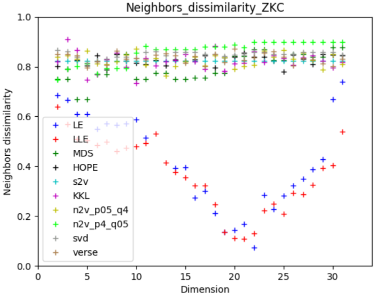
- Neighbors dissimilarity

Neighbors dissimilarity

- Capture the ability of an embedding to recover neighbors
- $$\forall v \in V, D(v) = \frac{|N(v) \bigcap N_E(v)|}{|N(v)|}$$

Neighborhood





Measures

- Structural equivalence

Structural equivalence

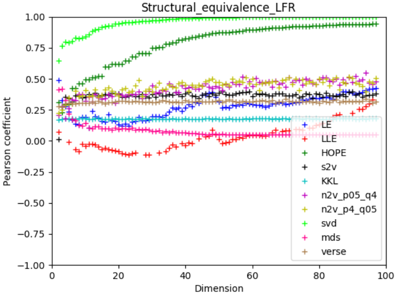
- Capture the ability of an embedding to recover the second-order proximity
- $$\forall u \in V, \forall v \in V,$$

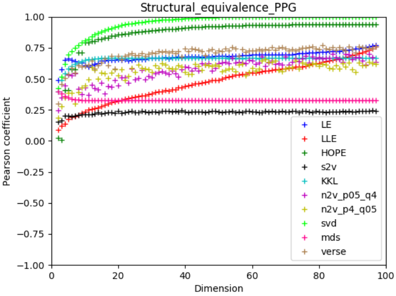
$$L_A(u,v) = dist(A_u, A_v), \qquad L_E(u,v) = dist(Y_u, Y_v)$$

- pearson correlation coefficient

$$Struct\_eq = pearson(L_A, L_E)$$

Structural equivalence





Measures

- Isomorphic equivalence

Isomorphic equivalence

- Capture the ability of an embedding to recover the roles of the nodes
- $$\forall u \in V, \forall v \in V,$$

$$L_{Egonet}(u, v) = dist(EN_u, EN_v), \qquad L_E(u, v) = dist(Y_u, Y_v)$$

- pearson correlation coefficient

$$Isom\_eq = pearson(L_{Egonet}, L_E)$$

Isomorphic equivalence

- How to compute the distance between two graphs?

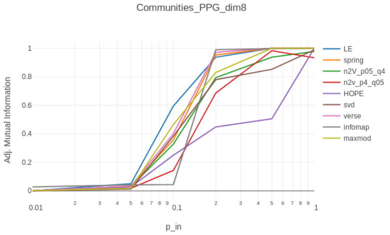
Measures

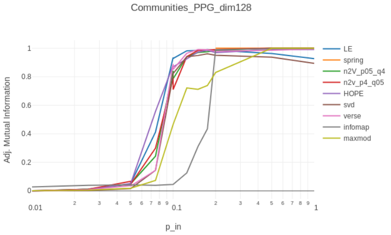
- Community detection

Community detection

- Capture the ability of an embedding to capture community structure
- $$Score = AMI(L_{Community}, L_{Clusters})$$

Community detection





Conclusion

- Different usage: clustering, role detection, structural equivalence
- Link prediction

Future:

- Dynamic graphs
- Attributed graphs
- Multitask embeddings

Graphs