1.1 Graph

A collection of objects (i.e., nodes) with a set of interactions (i.e., edges)

- Proteins and biological interactions.
- Relationships between club members.

simple graphs:

- At most one edge between each pair of nodes
- No edges between a node and itself
- Edges are all undirected

1.1.1 Multi-relational graphs

- Undirected, directed, weighted edges
- Edges are of different types

1. $\mathcal{R}$ is the set of relations
2. One $\mathbf{A}_\tau$ per edge type $\tau \in \mathcal{R}$, $(u, \tau, v) \in \mathcal{E}$
3. Adjacency tensor $\mathcal{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}| \times |\mathcal{V}|}$

- e.g., *Heterogeneous* and *multiplex* graphs

1.1.1 Heterogeneous graphs

Nodes are in disjoint sets $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \ldots \mathcal{V}_k$ where $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$, $\forall i \neq j$.

- Certain edges only connect nodes of certain types, (i.e., $(u, \tau_i, v) \in \mathcal{E} \rightarrow u \in \mathcal{V}_j, v \in \mathcal{V}_k$).
- Edges representing "treatments" only occur between drug nodes and disease nodes.
- Special case: *multipartite* graphs, (i.e., $(u, \tau_i, v) \in \mathcal{E} \rightarrow u \in \mathcal{V}_j, v \in \mathcal{V}_k \wedge j \neq k$).

1.1.1 Multiplex graphs

- Graph is decomposed in a set of k layers, (e.g., transportation network).
- Every node belong to every layer (city).
- Each layer corresponds to a unique relation (different mode of transportation).
- Inter-layer edges connect the same node across layers (possibility of switching modes of transportation within a city).

1.1.2 Feature information

Often node-level attributes: $\mathbf{X} \in \mathbb{R}^{|V| \times m}$
(e.g., a profile picture associated with a user in a social network)

Graph or network?

The difference is a historical one.

- "Graph": machine learning community
- "Network": data mining and network science communities

In this book

- *Graph* for abstract data structure
- *Network* for specific, real-world instantiations of this data structure

1.2.1 Node classification

Nodes in a graph are not i.i.d.

- *Homophily*: adjacent nodes share attributes
- *Structural equivalence*: nodes with similar local neighborhood structures have similar labels
- *Heterophily*: nodes tend to be connected to nodes with different labels

Supervised or semi-supervised?

Semi-supervised:

- Have access to the full graph during training including unlabeled test nodes
- Can use neighborhood information about test nodes to improve model during training

Supervised:

- Unlabeled nodes are completely unobserved during training

1.2.2 Relation prediction

> Given nodes $\mathcal{V}$ and an incomplete set of edges $\mathcal{E}_{\text{train}} \in \mathcal{E}$, infer missing edges.

1. Also known as link prediction, graph completion, and relational inference
2. Real-world application: recommendation system, predicting drug side-effects ...

1.2.3 Community detection

The graph analogue of unsupervised clustering

1.2.4 Graph classification, regression, and clustering

Graph are an i.i.d. data with labels.

- Challenge: define useful features upon relational structure
- E.g., predict molecule's toxicity or solubility given their structure

2.1 Graph statistics and kernel methods

Traditional machine learning: use node-level features based on heuristic functions or domain knowledge as input to a classifier.

- Popular prior to the advent of deep learning

2.1.1 Node-level statistics and features

- Node degree: $d_u = \sum_{v \in \mathcal{V}} \mathbf{A}[u, v]$
- Node centrality

> **Eigenvector centrality** for $\forall u \in \mathcal{V}$
> $e_u = \frac{1}{\lambda} \sum_{v \in \mathcal{V}} A[u, v] e_v$, constant λ.

Then $\mathbf{e}$ is the eigenvector of $\mathbf{A}$, since this equation in vector notation is $\lambda \mathbf{e} = \mathbf{A}\mathbf{e}$

Assume that centrality values are positive, by Perron-Frobenius Theorem, such $\mathbf{e}$ corresponds to the largest eigenvalue of $\mathbf{A}$.

Convergence of power iteration

For $\mathbf{A} \in \mathbb{R}^{n \times n}$ with a dominant eigenvalue, there exists a nonzero vector $\mathbf{x}$ such that the sequence $\mathbf{Ax}$, $\mathbf{A}^2\mathbf{x}$, ..., $\mathbf{A}^k\mathbf{x}$, ... approach a multiple of $\mathbf{e}$.

$\mathbf{A}$ has n linearly independent eigenvectors $\mathbf{x}_1$, ..., $\mathbf{x}_n$ with $\lambda_1 \leq \ldots \leq \lambda_n$. Define a $\mathbf{x}$ with $c_n \neq 0$ as $\mathbf{x} = \sum_{i=1}^{n} c_i x_i$. $\mathbf{A}^k\mathbf{x} = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{x}_i = \lambda_n^k \sum_{i=1}^{n} c_i \left(\frac{\lambda_i}{\lambda_n}\right)^k \mathbf{x}_i$

Since $\frac{\lambda_i}{\lambda_n} < 1$ for $i = 1 : n-1$, $\lim_{k \to \infty} \mathbf{A}^k\mathbf{x} = \lambda_n^k c_n \mathbf{x}_n$.

- Start off with $[1, \ldots, 1]^\intercal$ to compute $\mathbf{e}$

2.1.1 Node-level statistics and features

Degree centrality

- All nodes are equivalent

Eigenvector centrality

- A node is important if it is linked to by other important nodes
- Low degree node also becomes important by linking to a hub node

High degree $\neq$ High eigenvector centrality

2.1.1 Node-level statistics and features

- Clustering coefficient: how tightly clustered a node neighborhood is

local variant for $\forall u \in \mathcal{V}$
$c_u = |(v_1, v_2) \in \mathcal{E} : v_1, v_2 \in \mathcal{N}(u)|/d_u^2$

$c_u = 1$: all of u's neighbors are also neighbors of each other

2.1.2 Graph-level features

- Bag of nodes
- The Weisfieler-Lehman kernel
- Count the occurrence of different small subgraph structures (graphlets)
- Path-based methods: running walks to count occurrence of different degree sequences
  - e.g., random walk

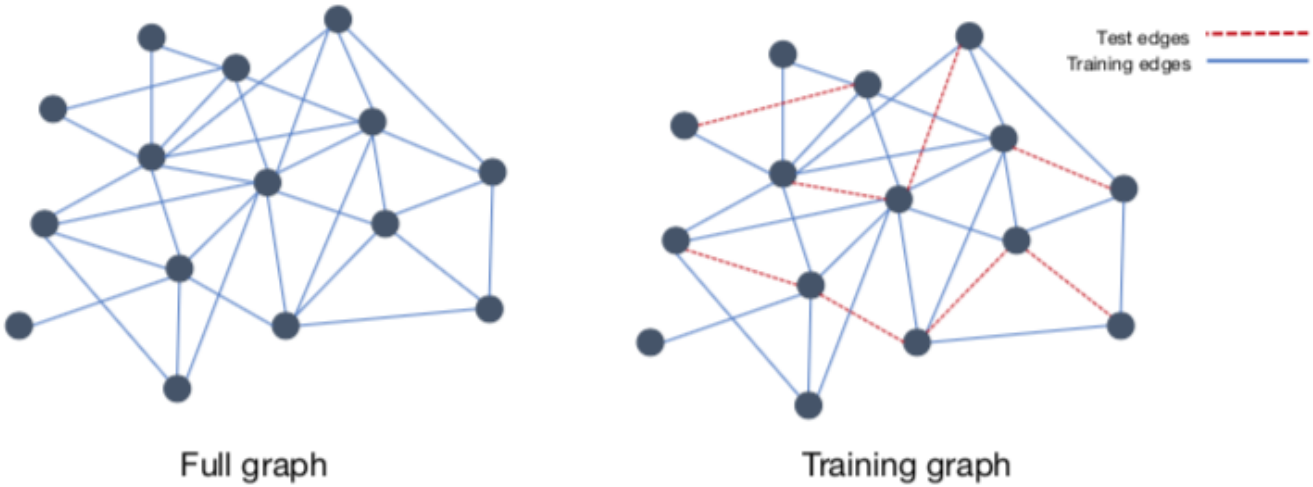
2.2 Neighborhood overlap detection

Quantify the extent to which a pair of nodes are related, e.g., the simplest neighborhood overlap measure

The similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$
$\mathbf{S}[u, v] = |\mathcal{N}(u) \cap \mathcal{N}(v)|$
$P(\mathbf{A}[u.v] = 1) \propto \mathbf{S}[u, v]$

- Relation prediction: set a threshold

2.2.1 Local overlap measures



Design functions of $|\mathcal{N}(u) \cap \mathcal{N}(v)|$ to avoid bias towards predicting edges for nodes with large degrees

- $\mathbf{S}_{\text{Sorenson}}[u, v] = 2|\mathcal{N}(u) \cap \mathcal{N}(v)|/(d_u + d_v)$
- $\mathbf{S}_{\text{Jaccard}}[u, v] = |\mathcal{N}(u) \cap \mathcal{N}(v)|/|\mathcal{N}(u) \cup \mathcal{N}(v)|$

2.2.1 Local overlap measures

Consider the importance of common neighbors

- $\mathbf{S}_{\text{Resource Allocation}}[v_1, v_2] = \sum_{u \in \mathcal{N}(v_1) \cap \mathcal{N}(v_2)} \frac{1}{d_u}$
- $\mathbf{S}_{\text{Adamic-Adar}}[v_1, v_2] = \sum_{u \in \mathcal{N}(v_1) \cap \mathcal{N}(v_2)} \frac{1}{\log(d_u)}$

- A shared low-degree neighbor is more informative than a shared high-degree one.

2.2.2 Global overlap measures

Two nodes could have no local overlap in their neighborhoods but still be members of the same community.

Katz index $\mathbf{S}_{\text{Katz}}[u, v] = \sum_{i=1}^{\infty} \beta^i \mathbf{A}^i[u, v], \beta \in \mathbb{R}^+$

- small $\beta < 1$ down-weight the importance of long paths.
- If $\lambda_{\max}$ of $\mathbf{A}$ is $< \frac{1}{\beta}$ and $(\mathbf{I} - \beta\mathbf{A})$ is non-singular, $\mathbf{S}_{\text{Katz}} = (\mathbf{I} - \beta\mathbf{A})^{-1} - \mathbf{I}$

2.2.2 Global overlap measures

- High-degree nodes are in more paths
- Katz index is biased by node degree
- $\Rightarrow$ LHN similarity considers $\frac{\mathbf{A}^i}{E\{\mathbf{A}^i\}}$

Configuration model
Draw a random graph with the same set of degrees as the given graph

- There are d_u edges leaving u
- Each has a $\frac{d_v}{2|\mathcal{E}|}$ chance of ending at v
- $E\{\mathbf{A}[u,v]\} = \frac{d_u d_v}{2|\mathcal{E}|}$.

2.2.2 Global overlap measures

For $E\{\mathbf{A}^2[v_1, v_2]\}$, consider path of length 2 with any intermediate vertex u

- $E\{\mathbf{A}[v_1, u]\} = \frac{d_{v_1} d_u}{2|\mathcal{E}|}$, $E\{\mathbf{A}[u, v_2]\} = \frac{d_{v_2}(d_u - 1)}{2|\mathcal{E}|}$
- $\Rightarrow E\{\mathbf{A}^2[v_1, v_2]\} = \frac{d_{v_1} d_{v_2}}{(2|\mathcal{E}|)^2} \sum_{u \in \mathcal{V}} (d_u - 1) d_u$

The analytical computation of expected node path becomes intractable as paths length > 3.

- Use $\lambda_{\max}$ to approximate the growth in the number of paths

The expected number of paths of length k from i to j is the j^{th} element of p_k where $p_k = \mathbf{A}^k e_i$. As $k \to \infty$, $p_{k+1} = \lambda_{\max} p_k$.

The number of paths from i to j increase by a factor of $\lambda_{\max}$ each time we add one extra step to the path length when k is large.

2.2.2 Global overlap measures

With approximation $E\{\mathbf{A}^i[u,v]\} = \frac{d_u d_v \lambda_{\max}^{i-1}}{2|\mathcal{E}|}$, the LNH index is defined as

$$\mathbf{S}_{\text{LNH}}[u,v] = \mathbf{I}[u,v] + \frac{2|\mathcal{E}|}{d_u d_v} \sum_{i=1}^{\infty} \beta^i \lambda_{\max}^{1-i} \mathbf{A}^i[u,v], \quad \beta \in \mathbb{R}^+$$

- small $\beta < 1$ also down-weight long paths
- it only gives a high similarity score if two nodes occur on more paths than we expect

2.2.2 Global overlap measures

Random walk methods: considers random walks rather than exact counts of paths.

The walker starting from node u will move to a random neighbor with probability c and return to itself with probability $1-c$.

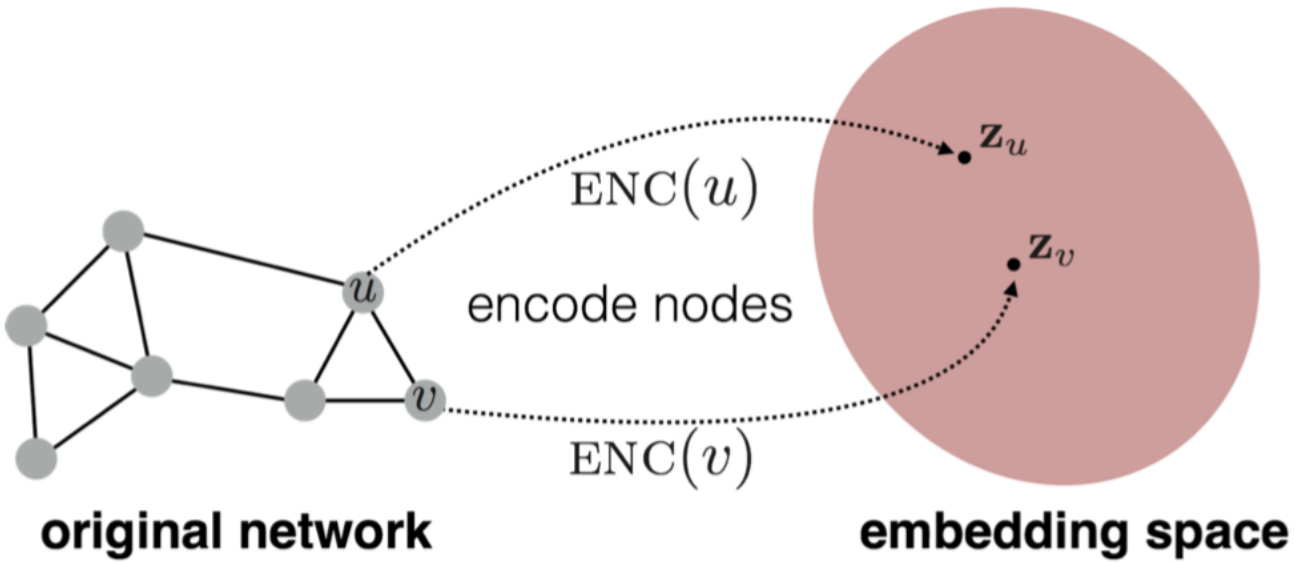
Denote q_{uv} the probability this random walker locates at node v in the steady state, we have $\mathbf{q_u} = c\mathbf{P}\mathbf{q_u} + (1-c)\mathbf{e_u}$, and $\mathbf{q_u} = (1-c)(\mathbf{I} - c\mathbf{P})^{-1}\mathbf{e_u}$

Random walk similarity index
$\mathbf{S}_{\mathrm{RW}}[u,v] = \mathbf{q_u}[v] + \mathbf{q_v}[u]$

Introduction

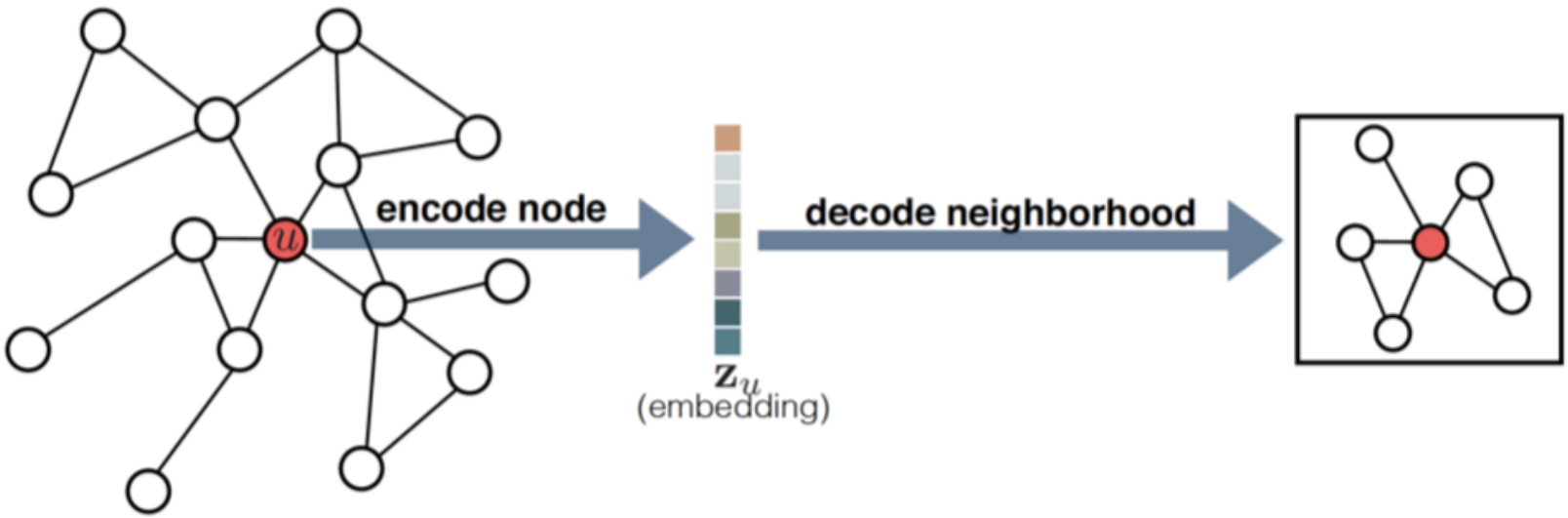
Node embedding

- Project nodes into a latent space
- Geometric relations in this latent space correspond to edges
- No node feature or extra information is used



3.1 An encoder-decoder perspective

- **Encoder** model: maps each node into a low-dimensional embeddings
- **Decoder** model: use low-dimensional embeddings to reconstruct information about node's neighborhood



3.1.1 The encoder (ENC)

Encoder $\text{ENC} : \mathcal{V} \to \mathbb{R}^d$
$\mathbf{z}_v \in \mathbb{R}^d$ is the embedding for node v

Simplest encoding approach: encoder is just an embedding-lookup

Shallow encoding
$\text{ENC}(v) = \mathbf{Z}[v], \quad \mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times d}$

3.1.2 The decoder (DEC)

Pairwise decoder $\mathrm{DEC}: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$
predict similarity of pairs of nodes

Reconstruction $\mathrm{DEC}(\mathrm{ENC}(u), \mathrm{ENC}(v)) = \mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v)$

Similarity measure $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, e.g., $\mathbf{S} = \mathbf{A}$, neighborhood overlap

Goal: minimize reconstruction loss so that $\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) \approx \mathbf{S}[u, v]$

3.1.3 Optimizing the model

Given ENC, DEC and $\mathbf{S}$, the empirical reconstruction loss is defined as:

$$\mathcal{L} = \sum_{(u,v)\in\mathcal{D}} \ell(\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v), \mathbf{S}[u,v])$$

where $\mathcal{D}$ is a set of training node pairs, and

$$\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

e.g., mean-squared error, classification loss

3.2 Factorization-based approaches

Laplacian eigenmaps

- $\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) = \|\mathbf{z}_u - \mathbf{z}_v\|_2^2$
- $\mathcal{L} = \sum_{(u,v)\in\mathcal{D}} \mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) \cdot \mathbf{S}[u, v]$

Intuition

- Penalize the model when very similar nodes have embeddings that are far apart

3.2 Factorization-based approaches

Inner-product methods

- $\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) = \mathbf{z}_u^\intercal \mathbf{z}_v$
- $\mathcal{L} = \sum_{(u,v)\in\mathcal{D}} \|\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) - \mathbf{S}[u,v]\|_2^2$ by stacking embedding into $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}|\times d}$

Intuition

- Often define $\mathbf{S}$ as some polynomial function of the adjacency matrix
- Embeddings are optimized with $\mathbf{z}_u^\intercal \mathbf{z}_v \approx \mathbf{S}[u,v]$

3.3 Random walk embeddings

Idea: Node embeddings are optimized so that two nodes with similar embeddings if they tend to co-occur on short random walks over the graph

3.3 Random walk embeddings

DeepWalk and node2vec:

- We aim to find a shallow embedding, where $\mathbf{z}_u$ is the embedding of node u
- $N_R(u)$, the neighborhood of node u is obtained by some random walk
- Denote $P(v|\mathbf{z}_u)$ as the predicted probability of visiting node v on the random walk starting from node u
- $\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) = P(v|\mathbf{z}_u) = \exp(\mathbf{z}_u^\intercal \mathbf{z}_v) / \sum_{k \in \mathcal{V}} \exp(\mathbf{z}_u^\intercal \mathbf{z}_k)$
- $\mathcal{L} = -\sum_{u \in \mathcal{V}} \sum_{v \in N_R(u)} \log(P(v|\mathbf{z}_u))$
- Find embeddings that minimize $\mathcal{L}$

3.3 Random walk embeddings

Large-scale information network embeddings:

- $\mathrm{DEC}(\mathbf{z}_u, \mathbf{z}_v) = 1/(1 + \exp(-\mathbf{z}_u^\intercal \mathbf{z}_v))$
- Use adjacency matrix to measure similarity

3.4 Limitations of shallow embeddings

- No shared parameters between nodes in the encoder
- Do not leverage node features in the encoder
- Can only generate embeddings for nodes that were present during training – need additional optimizations for new nodes

Knowledge graph completion

Knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Edge (u, τ, v) indicates the presence of a particular relation τ between two nodes.

- E.g., (u, TREATS, v) in a biomedical knowledge graph.
- Relation prediction.

4.1 Reconstructing multi-relational data

Given $\mathbf{z}_u$ and $\mathbf{z}_v$

- Predict the existence of an edge between node u and v
- Determine the type of the predicted edge

Multi-relational decoder

- $\text{DEC} : \mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \to \mathbb{R}^+$
- interpret $\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v)$ as the likelihood that the edge (u, τ, v) exists

Example: RESCAL

- $\mathbf{R}_\tau \in \mathbb{R}^{d \times d}$ is specific to relation $\tau \in \mathcal{R}$
- $\mathbf{R}_\tau$ is learnable
- $\mathrm{DEC}(u, \tau, v) = \mathbf{z}_u^\intercal \mathbf{R}_\tau \mathbf{z}_v$
- $\mathcal{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}| \times |\mathcal{V}|}$
- $\mathcal{L} = \sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{V}} \sum_{\tau \in \mathcal{R}} \left\| \mathbf{z}_u^\intercal \mathbf{R}_\tau \mathbf{z}_v - \mathcal{A}[u, \tau, v] \right\|^2$

Loss functions, decoders, and similarity functions

Relation prediction reconstruct immediate neighbors

- Have various decoders and loss functions
- Define the similarity measure directly based on the adjacency tensor

4.2 Loss functions

1. Ideally, the computation of the loss function is $O(|\mathcal{E}|)$ since many multi-relational graphs are sparse ($|\mathcal{E}| << |\mathcal{V}|^2|\mathcal{R}|$).
2. Use a loss for classification on edges to decode the adjacency tensor from the node embeddings.

Cross-entropy with negative sampling

$$\mathcal{L} = \sum_{(u,\tau,v)\in\mathcal{E}} -\log(\sigma(\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v))) - \gamma \mathbb{E}_{v_n \sim P_{n,u}(\mathcal{V})} \left[\log\left(\sigma\left(-\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_{v_n})\right)\right)\right]$$

- σ denotes the logistic function
- $P_{n,u}(\mathcal{V})$ is a "negative sampling" distribution over the set of node $\mathcal{V}$
- $P_{n,u}(\mathcal{V})$ might depend on u
- $\gamma > 0$ is a hyperparameter
- $\log(\sigma(\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v)))$ is the log-likelihood for existence of edge (u, τ, v)
- $E_{v_n \sim P_{n,u}(\mathcal{V})}[\log(\sigma(\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_{v_n})))]$ is the expected log-likelihood for non-existence of an edge
- Usually, we consider $v_n \in \mathcal{P}_{n,v}$, where $\mathcal{P}_{n,v}$ is a small set sampled from $P_{n,u}(\mathcal{V})$

Max-margin loss

$$\mathcal{L} = \sum_{(u,\tau,v)\in\mathcal{E}} \sum_{v_n\in\mathcal{P}_{n,u}} \max(0, -\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) + \text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_{v_n}) + \Delta).$$

Contrastive estimation: compare the decoded score for a true pair with a negative sample

The loss will equal 0 if the difference in scores $\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v)$ and $\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_{v_n})$ is at least Δ

4.3 Multi-relational decoders

The decoder for RESCAL $\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) = \mathbf{z}_u^\intercal \mathbf{R}_\tau \mathbf{z}_v$ has $O(d^2)$ parameters for each relation type.

- Ideally use $O(d)$ parameters

Translational decoders

- $\mathbf{r}_\tau \in \mathbb{R}^d$

TransE

$$\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) = -\|\mathbf{z}_u + \mathbf{r}_\tau - \mathbf{z}_v\|$$

- Represents relations as translations in the embedding space

TransX

$$\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) = -\|g_{1,\tau}(\mathbf{z}_u) + \mathbf{r}_\tau - g_{2,\tau}(\mathbf{z}_v)\|$$

TransH

$$\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) = -\|(\mathbf{z}_u - \mathbf{w}_r^\intercal \mathbf{z}_u \mathbf{w}_r) + \mathbf{r}_\tau - (\mathbf{z}_v - \mathbf{w}_r^\intercal \mathbf{z}_v \mathbf{w}_r)$$

Multi-linear dot products

DistMult

$\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) =< \mathbf{z}_u, \tau, \mathbf{z}_v >= \sum_{i=1}^{d} \mathbf{z}_u[i] \times \mathbf{r}_\tau[i] \times \mathbf{z}_v[i]$

- Take a straightforward generalization of the dot product
- Can only encode symmetric relations

$$\begin{aligned}\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) &=< \mathbf{z}_u, \mathbf{r}_\tau, \mathbf{z}_v > \\ &=< \mathbf{z}_v, \mathbf{r}_\tau, \mathbf{z}_u > \\ &= \text{DEC}(\mathbf{z}_v, \tau, \mathbf{z}_u)\end{aligned}$$

Complex decoders

Many relation types in multi-relational graphs are directed and asymmetric

ComplEx

$$\begin{aligned}\text{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) &= Re(< \mathbf{z}_u, \tau, \bar{\mathbf{z}}_v >) \\ &= Re(\sum_{i=1}^{d} \mathbf{z}_u[i] \times \mathbf{r}_\tau[i] \times \bar{\mathbf{z}}_v[i])\end{aligned}$$

- $\mathbf{z}_u, \mathbf{z}_v, \mathbf{r}_\tau \in \mathbb{C}^d$ are complex valued embedding
- Re takes the real component
- $\bar{\mathbf{z}}_v$ is the complex conjugate

4.3.1 Representational abilities

Symmetry $(u, \tau, v) \in \mathcal{E} \Leftrightarrow (v, \tau, u) \in \mathcal{E}$

Anti-symmetry $(u, \tau, v) \in \mathcal{E} \Rightarrow (v, \tau, u) \notin \mathcal{E}$

Inversion $(u, \tau_1, v) \in \mathcal{E} \Leftrightarrow (v, \tau_2, u) \in \mathcal{E}$

Compositionality
$(u, \tau_1, y) \in \mathcal{E} \wedge (y, \tau_2, v) \in \mathcal{E} \Rightarrow (u, \tau_3, v) \in \mathcal{E}$