FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Aprendizaje Automático para Datos en Grafos

Graph representation learning

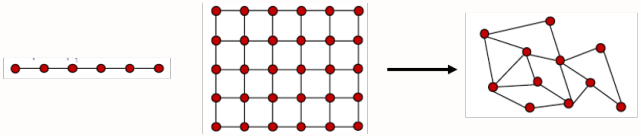
Marcelo Fiori
Muy basado en transparencias de **Gonzalo Mateos**

mfiori@fing.edu.uy
http://www.fing.edu.uy/~mfiori/

10 de noviembre, 2022

Aprendizaje automático en grafos: Motivación

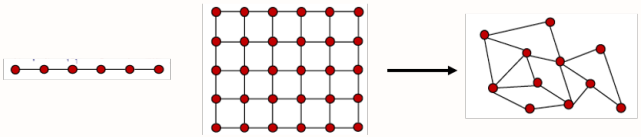
- Modelos muy exitosos para representación (y) aprendizaje en datos estructurados
  - Secuencias (e.g., texto, audio, videos) via recurrent neural networks (RNNs)
  - Clasificación de imágenes via convolutional neural networks (CNNs)



- Pero los datos no siempre son regulares $\Rightarrow$ Estructuras relacionales complejas
  - Grafos en redes sociales, química computacional, biología, ...

Aprendizaje automático en grafos: Motivación

- Modelos muy exitosos para representación (y) aprendizaje en datos estructurados
  - Secuencias (e.g., texto, audio, videos) via recurrent neural networks (RNNs)
  - Clasificación de imágenes via convolutional neural networks (CNNs)



- Pero los datos no siempre son regulares $\Rightarrow$ Estructuras relacionales complejas
  - Grafos en redes sociales, química computacional, biología, ...
- Desafíos: aplicar modelos diseñados para datos regulares, en grafos
  - La estructura de los grafos puede ser arbitraria y variar en diferentes escenarios
  - Las convoluciones no generalizan a dominios irregulares

M. Bronstein et al, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, 2017

¿De qué vamos a hablar?

Graph representation learning (GRL)

- Aprender vectores en baja dimensión (embeddings) para datos en grafos
- Tipos de aprendizaje:
  - Supervisado: aprender representaciones para clasificación de nodos o grafos
  - No-supervisado: aprender representaciones que preserven estructura de grafos
- Dominio subyacente:
  - Transductivo: estructura del grafo fija (e.g., una red social muy grande)
  - Inductivo: los grafos de entrada pueden variar (e.g., multiples moléculas)
- Información en los nodos:
  - Featureless: no tenemos información adicional (i.e., graph signals)
  - With features: los nodos tienen atributos o características

Roadmap

1. The network embedding problem
2. Una taxonomía de modelos de graph embedding
3. Unsupervised graph embedding

Embeddings de grafos

- Aprender un mapa de un grafo discreto a un dominio continuo
- Dado $G(\mathcal{V}, \mathcal{E})$ con matriz de adyacencia (con pesos) $\mathbf{W} \in \mathbb{R}^{N_v \times N_v}$
- **Objetivo**: aprender una representación en vectores d-dimensionales $\{\mathbf{z}_i\}_{i \in \mathcal{V}}$

  $\Rightarrow$ Criterio es preservar propiedades locles y globales del grafo

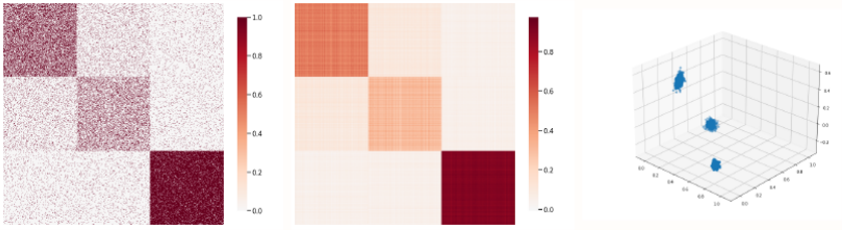
Embeddings de grafos

- Aprender un mapa de un grafo discreto a un dominio continuo
- Dado $G(\mathcal{V}, \mathcal{E})$ con matriz de adyacencia (con pesos) $\mathbf{W} \in \mathbb{R}^{N_v \times N_v}$
- **Objetivo**: aprender una representación en vectores d-dimensionales $\{\mathbf{z}_i\}_{i \in \mathcal{V}}$
  $\Rightarrow$ Criterio es preservar propiedades locles y globales del grafo
- Salida es una matriz de embeddings de nodos $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{N_v}]^\top \in \mathbb{R}^{N_v \times d}$
  $\Rightarrow$ Elegir $d \ll N_v$ para escalabilidad
  $\Rightarrow$ Hay reducción de dimensionalidad
- Es posible extender esto a un embedding del grafo entero via $\mathbf{z} \in \mathbb{R}^d$

Adjacency spectral embedding (revisitado)

- Ex: SBM con $N_v = 1500$, $Q = 3$ y parámetros:

$$\boldsymbol{\alpha} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \quad \boldsymbol{\Pi} = \begin{bmatrix} 0{,}5 & 0{,}1 & 0{,}05 \\ 0{,}1 & 0{,}3 & 0{,}05 \\ 0{,}05 & 0{,}05 & 0{,}9 \end{bmatrix}$$



- Adyacencia muestreada (izq.), $\mathbf{Z}\mathbf{Z}^\top$ (centro), filas de $\mathbf{Z}$ (der.)
- Embeddings para poder usar métodos geométricos de análisis

El rol de las señales en grafos

- Señales en grafos (a.k.a. atributos en nodos o características) $\mathbf{X} \in \mathbb{R}^{N_v \times F}$

  - Ex: Edad, género (redes sociales), señales fMRI, ratings de productos

- Los embeddings captura información estructural y semántica del grafo

$$\{\mathbf{W}, \mathbf{X}\} \mapsto \mathbf{Z}$$

El rol de las señales en grafos

- Señales en grafos (a.k.a. atributos en nodos o características) $\mathbf{X} \in \mathbb{R}^{N_v \times F}$

  - Ex: Edad, género (redes sociales), señales fMRI, ratings de productos

- Los embeddings captura información estructural y semántica del grafo

$$\{\mathbf{W}, \mathbf{X}\} \mapsto \mathbf{Z}$$

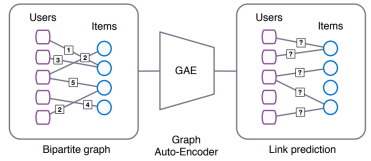
- Sin $\mathbf{X}$, el embedding $\{\mathbf{W}\} \mapsto \mathbf{Z}$ se dice featureless

  $\Rightarrow$ El mapeo solo preserva información estructural

Embeddings transductivos e inductivos

Transductive network embedding

- Embedding de nodos de un grafo fijo (en general grande)
  - Ex: Recomendación de productos o amigos via predicción de enlaces
  - Ex: Clasificación de nodos en aprendizaje semi-supervisado

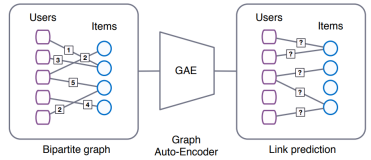


- Dados nuevos nodos, hay que actualizar o re-entrenar el modelo

Embeddings transductivos e inductivos

Transductive network embedding

- Embedding de nodos de un grafo fijo (en general grande)
  - Ex: Recomendación de productos o amigos via predicción de enlaces
  - Ex: Clasificación de nodos en aprendizaje semi-supervisado



- Dados nuevos nodos, hay que actualizar o re-entrenar el modelo

Inductive network embedding

- Aprender mapas a representaciones que generalizan a grafos no vistos
  - Ex: Embedding de grafos de cerebros para clasificación de sujetos
  - Ex: Embeding de grafos dinámicos para clustering temporal
- Típicamente se necesitan señales $\mathbf{X}$ para hacer embedding inductivo

Embeddings supervisados y no-supervisados

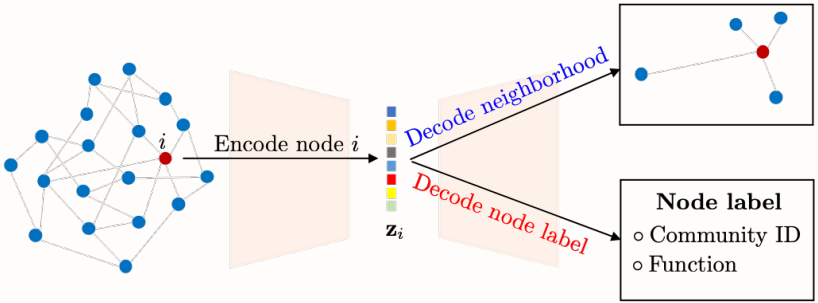
Unsupervised network embedding

- Solamente tenemos la topología del grafo $\mathbf{W}$ (y eventualmente $\mathbf{X}$)
  - Preservar estructuras del grafo optimizando una función de pérdida/reconstrucción
  - Decodificar los embedding $\mathbf{Z}$ para aproximar bien $\mathbf{W}$
- Ex: compresión, visualización, clustering, predicción de enlaces

Supervised network embedding

- Además de $\mathbf{W}$ (y $\mathbf{X}$), tenemos disponibles etiquetas de nodos o grafos $\mathbf{y}^S$
  - Optimizar embeddings para tareas aguas abajo
  - Combina reconstrucción y función de pérdida de la tarea específica
- Ex: clasificación de nodos, clasificación de grafos

Una perspectiva encoder-decoder



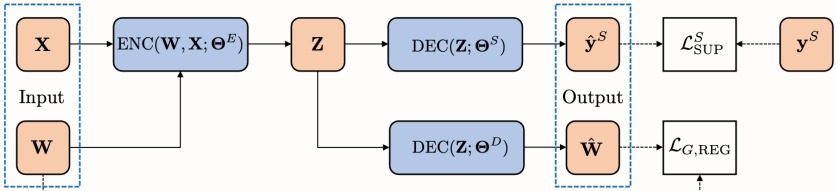
W. L. Hamilton et al, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, 2018

Roadmap

1. The network embedding problem
2. Una taxonomía de modelos de graph embedding
3. Unsupervised graph embedding

Un modelo abarcativo para graph embedding

- Graph Encoder Decoder Model (GraphEDM)
  - ⇒ Marco unificador para revisar y comparar métodos de GRL
  - ⇒ Biblioteca open-source con métodos y aplicaciones



I. Chami et al, "Machine learning on graphs: A model and comprehensive taxonomy," *arXiv:2005.03675 [cs.LG]*, 2020

- **Q:** ¿Cuáles son los componentes constitutivos del modelo?

https://github.com/google/gcnn-survey-paper

Input

- Grafo no dirigido $G(\mathcal{V}, \mathcal{E})$, con $|\mathcal{V}| = N_v$ y $|\mathcal{E}| = N_e$

  $\Rightarrow$ Matriz de adyacencia (con pesos) $\mathbf{W} \in \mathbb{R}^{N_v \times N_v}$

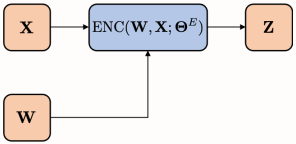
Input

- Grafo no dirigido $G(\mathcal{V}, \mathcal{E})$, con $|\mathcal{V}| = N_v$ y $|\mathcal{E}| = N_e$
  $\Rightarrow$ Matriz de adyacencia (con pesos) $\mathbf{W} \in \mathbb{R}^{N_v \times N_v}$
- Opcionalmente señales en el grafo (características de nodos) $\mathbf{X} \in \mathbb{R}^{N_v \times F}$

Input

- Grafo no dirigido $G(\mathcal{V}, \mathcal{E})$, con $|\mathcal{V}| = N_v$ y $|\mathcal{E}| = N_e$
  $\Rightarrow$ Matriz de adyacencia (con pesos) $\mathbf{W} \in \mathbb{R}^{N_v \times N_v}$
- Opcionalmente señales en el grafo (características de nodos) $\mathbf{X} \in \mathbb{R}^{N_v \times F}$
- Para aprendizaje (semi)-supervisado, también necesitamos etiquetas de:
  - Nodos (N), para clasificación de nodos y clustering
  - Aristas (E), para predicción de enlaces o clasificación de relaciones
  - Grafos (G), para clustering y clasificación de grafos
- Las etiquetas de supervisión las denotamos $\mathbf{y}^S$, donde $S \in \{N, E, G\}$

Encoder



- Graph encoder network

$$\text{ENC}_{\boldsymbol{\Theta}^E} : \mathbb{R}^{N_v \times N_v} \times \mathbb{R}^{N_v \times F} \mapsto \mathbb{R}^{N_v \times d}$$

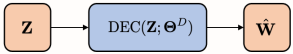
  $\Rightarrow$ Parámetros entrenables $\boldsymbol{\Theta}^E$

- Combina la estructura del grafo con señales para producir un embedding

$$\mathbf{Z} = \text{ENC}(\mathbf{W}, \mathbf{X}; \boldsymbol{\Theta}^E)$$

  $\Rightarrow$ Captura diferentes propiedades del grafo en base al tipo de supervisión

Decoder (I)



- Graph decoder network

$$\mathrm{DEC}_{\mathbf{\Theta}^D} : \mathbb{R}^{N_v \times d} \mapsto \mathbb{R}^{N_v \times N_v}$$

  $\Rightarrow$ Parámetros entrenables $\mathbf{\Theta}^D$

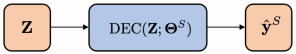
- Usa $\mathbf{Z}$ para producir scores de (di)similitud $\hat{W}_{ij}$ para cada $\{i,j\} \in \mathcal{V}^{(2)}$

$$\hat{\mathbf{W}} = \mathrm{DEC}(\mathbf{Z};\mathbf{\Theta}^D)$$

  $\Rightarrow$ Reconstrucción no supervisada del grafo

  $\Rightarrow$ Aproxima $\mathbf{W}$, o más en general una matriz de (di)similitud $s(\mathbf{W})$

Decoder (II)



- Classification network

$$\mathrm{DEC}_{\mathbf{\Theta}^S} : \mathbb{R}^{N_v \times d} \mapsto \mathbb{R}^{N_v \times |\mathcal{Y}|}$$

$\Rightarrow$ Parámetros entrenables $\mathbf{\Theta}^S$, espacio de etiquetas $\mathcal{Y}$

- Usa $\mathbf{Z}$ para producir distribuciones de etiquetas para cada nodo

$$\hat{\mathbf{y}}^S = \mathrm{DEC}(\mathbf{Z};\mathbf{\Theta}^S)$$

$\Rightarrow$ Aprendizaje (semi)-supervisado para clasificación de nodos/grafos

Output

- Matriz de similitud reconstruida $\hat{\mathbf{W}} \in \mathbb{R}^{N_v \times N_v}$

  $\Rightarrow$ Usado para entrenar algoritmos de embedding no supervisados

Output

- Matriz de similitud reconstruida $\hat{\mathbf{W}} \in \mathbb{R}^{N_v \times N_v}$

  $\Rightarrow$ Usado para entrenar algoritmos de embedding no supervisados

- Para aprendizaje (semi)-supervisado, las salidas son las etiquetas que se predicen $\hat{\mathbf{y}}^S$
  - El espacio de salida de etiquetas varía dependiendo del tipo de supervisión
- Node-level: $\hat{\mathbf{y}}^N \in \mathcal{Y}^{N_v}$ or $\hat{\mathbf{Y}}^N \in [0,1]^{N_v \times |\mathcal{Y}|}$

  $\Rightarrow$ Cuando $|\mathcal{Y}| = d$, se puede usar activación softmax en las filas de $\mathbf{Z}$
- Edge-level: $\hat{\mathbf{Y}}^E \in \mathcal{Y}^{N_v \times N_v}$, where typically $\mathcal{Y} = \{0,1\}^{\#\text{relation types}}$

  $\Rightarrow$ Cuando #relation types $= 1$ (i.e., predicción de enlaces), salida $\hat{\mathbf{W}}$
- Graph-level: $\hat{y}^G \in \mathcal{Y}$

  $\Rightarrow$ Usando $\mathbf{W}$, convertir $\mathbf{Z}$ a $\hat{y}^G$ via graph pooling

Funciones de pérdida

- Supervised loss

  $\Rightarrow \mathcal{L}^S_{\text{SUP}}$ compara las etiquetas que se predicen $\hat{\mathbf{y}}^S$ al ground truth $\mathbf{y}^S$

  Ex: clasificación semi-supervisada de nodos ($S = N$, $\mathcal{V} = \mathcal{V}_{obs} \cup \mathcal{V}_{miss}$)

$$\mathcal{L}^N_{\text{SUP}}(\mathbf{y}^N, \hat{\mathbf{y}}^N; \boldsymbol{\Theta}) = \sum_{i \in \mathcal{V}_{obs}} \ell(y_i^N, \hat{y}_i^N; \boldsymbol{\Theta})$$

Funciones de pérdida

- Supervised loss

  $\Rightarrow \mathcal{L}^S_{\text{SUP}}$ compara las etiquetas que se predicen $\hat{\mathbf{y}}^S$ al ground truth $\mathbf{y}^S$

  Ex: clasificación semi-supervisada de nodos ($S = N$, $\mathcal{V} = \mathcal{V}_{obs} \cup \mathcal{V}_{miss}$)

$$\mathcal{L}^N_{\text{SUP}}(\mathbf{y}^N, \hat{\mathbf{y}}^N; \boldsymbol{\Theta}) = \sum_{i \in \mathcal{V}_{obs}} \ell(y_i^N, \hat{y}_i^N; \boldsymbol{\Theta})$$

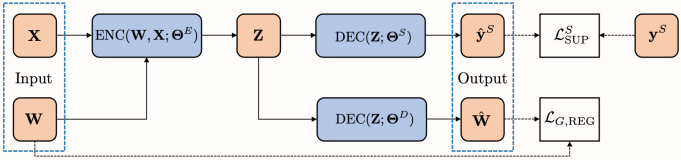
- Graph regularization loss

  $\Rightarrow \mathcal{L}_{G,\text{REG}}$ compara $\hat{\mathbf{W}}$ con matriz de (di)similitud objetivo $s(\mathbf{W})$

$$\mathcal{L}_{G,\text{REG}}(\mathbf{W}, \hat{\mathbf{W}}; \boldsymbol{\Theta}) = d_1(s(\mathbf{W}), \hat{\mathbf{W}})$$

  - $d_1(\cdot, \cdot)$: función distancia o de disimilitud
  - Aprovechr G via $s(\mathbf{W})$ para regularizar los parámetros del modelo $\boldsymbol{\Theta}$

Función objetivo

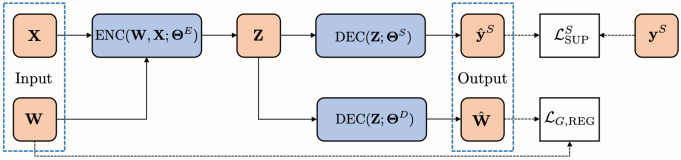


- Weight regularization loss

  $\Rightarrow \mathcal{L}_{\text{REG}}$ regulariza los parámetros entrenables $\boldsymbol{\Theta}$ para reducir overfitting

$$\mathcal{L}_{\text{REG}}(\boldsymbol{\Theta}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \|\boldsymbol{\theta}\|_2^2$$

Función objetivo



- Weight regularization loss

  $\Rightarrow \mathcal{L}_{\text{REG}}$ regulariza los parámetros entrenables $\boldsymbol{\Theta}$ para reducir overfitting

$$\mathcal{L}_{\text{REG}}(\boldsymbol{\Theta}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \|\boldsymbol{\theta}\|_2^2$$

- Función objetivo GraphEDM total

$$\mathcal{L}(\boldsymbol{\Theta}) = \alpha \mathcal{L}^S_{\text{SUP}}(\mathbf{y}^S, \hat{\mathbf{y}}^S; \boldsymbol{\Theta}) + \beta \mathcal{L}_{G,\text{REG}}(\mathbf{W}, \hat{\mathbf{W}}; \boldsymbol{\Theta}) + \mathcal{L}_{\text{REG}}(\boldsymbol{\Theta})$$

  $\Rightarrow$ Entrenamiento supervisado ($\alpha \neq 0$) o no supervisado ($\alpha = 0$)

- Q: ¿Aprendizaje supervisado end-to-end o en dos etapas?

Taxonomía de graph embedding

- Categorizamos los métodos de GRL en base al encoder y la función de pérdida
- Shallow embedding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{\Theta}^E) = \mathbf{\Theta}^E$
  - Un simple embedding lookup

Taxonomía de graph embedding

- Categorizamos los métodos de GRL en base al encoder y la función de pérdida
- Shallow embedding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{\Theta}^E) = \mathbf{\Theta}^E$
  - Un simple embedding lookup
- Graph auto-encoding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{W}; \mathbf{\Theta}^E)$
  - Transductivos como los shallow embeddings, no hay señal $\mathbf{X}$, así que funciona para grafo fijo G

Taxonomía de graph embedding

- Categorizamos los métodos de GRL en base al encoder y la función de pérdida
- Shallow embedding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{\Theta}^E) = \mathbf{\Theta}^E$
  - Un simple embedding lookup
- Graph auto-encoding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{W}; \mathbf{\Theta}^E)$
  - Transductivos como los shallow embeddings, no hay señal $\mathbf{X}$, así que funciona para grafo fijo G
- Graph regularization methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{X}; \mathbf{\Theta}^E)$
  - Usamos $\mathbf{W}$ via $\mathcal{L}_{G,\mathrm{REG}}$ para regularizar los embeddings

Taxonomía de graph embedding

- Categorizamos los métodos de GRL en base al encoder y la función de pérdida
- Shallow embedding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{\Theta}^E) = \mathbf{\Theta}^E$
  - Un simple embedding lookup
- Graph auto-encoding methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{W}; \mathbf{\Theta}^E)$
  - Transductivos como los shallow embeddings, no hay señal $\mathbf{X}$, así que funciona para grafo fijo G
- Graph regularization methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{X}; \mathbf{\Theta}^E)$
  - Usamos $\mathbf{W}$ via $\mathcal{L}_{G,\mathrm{REG}}$ para regularizar los embeddings
- Neighborhood aggregation methods $\mathbf{Z} = \mathrm{ENC}(\mathbf{W}, \mathbf{X}; \mathbf{\Theta}^E)$
  - Usamos $\mathbf{W}$ para propagar información entre nodos y aprender $\mathbf{Z}$

Roadmap

1. The network embedding problem
2. Una taxonomía de modelos de graph embedding
3. Unsupervised graph embedding

Unsupervised graph embedding

- **Objetivo:** Aprender embeddings de nodos que preserven estructura del grafo
- Optimizamos para reconstruir cierta matriz de (di)similitud entre nodos $s(\mathbf{W})$

$$\mathcal{L}(\mathbf{\Theta}) = \alpha \mathcal{L}^S_{\text{SUP}}(\mathbf{y}^S, \mathbf{y}^S; \mathbf{\Theta}) + \beta \mathcal{L}_{G,\text{REG}}(\mathbf{W}, \hat{\mathbf{W}}; \mathbf{\Theta}) + \mathcal{L}_{\text{REG}}(\mathbf{\Theta})$$

$\alpha = 0$

  - La salida del decoder es $\hat{\mathbf{W}}$, con $\hat{W}_{ij} = d_2(\mathbf{z}_i, \mathbf{z}_j)$
  - Regularización $\mathcal{L}_{G,\text{REG}} = d_1(s(\mathbf{W}), \hat{\mathbf{W}})$
  - Optimizamos sobre conjunto de entrenamiento $\{i, j\} \in \mathcal{V}^{(2)}_{obs}$, usando SGD o métodos espectrales

Unsupervised graph embedding

- **Objetivo:** Aprender embeddings de nodos que preserven estructura del grafo
- Optimizamos para reconstruir cierta matriz de (di)similitud entre nodos $s(\mathbf{W})$

$$\mathcal{L}(\mathbf{\Theta}) = \alpha \mathcal{L}^S_{\text{SUP}}(\mathbf{y}^S, \mathbf{y}^S; \mathbf{\Theta}) + \beta \mathcal{L}_{G,\text{REG}}(\mathbf{W}, \hat{\mathbf{W}}; \mathbf{\Theta}) + \mathcal{L}_{\text{REG}}(\mathbf{\Theta}) \qquad \alpha = 0$$

  - La salida del decoder es $\hat{\mathbf{W}}$, con $\hat{W}_{ij} = d_2(\mathbf{z}_i, \mathbf{z}_j)$
  - Regularización $\mathcal{L}_{G,\text{REG}} = d_1(s(\mathbf{W}), \hat{\mathbf{W}})$
  - Optimizamos sobre conjunto de entrenamiento $\{i, j\} \in \mathcal{V}^{(2)}_{obs}$, usando SGD o métodos espectrales
- La matriz de similitud objetivo $s(\mathbf{W})$ puede tomar varias formas

  Ex: Reconstruir proximidad de primer orden via $[s(\mathbf{W})]_{ij} = W_{ij}$
  Ex: Proximidad de alto orden $[s(\mathbf{W})]_{ij} = |\mathcal{N}_i \cap \mathcal{N}_j|$, Jaccard, Adamic-Adar
  Ex: Prob. $[s(\mathbf{W})]_{ij} = \text{P}\left(v_j \,\middle|\, v_i\right)$ que $i, j \in \mathcal{V}$ co-ocurran en paseos al azar (random walks)