**ECE598: Representation Learning: Algorithms and Models** Fall 2017

**Lecture 18: Graph2Vec**

*Lecturer: Pramod Viswanath* *Scribe: Qiang Ning, Nov 21, 2017*

## 18.1 Johnson-Lindenstrauss Lemma

Graph is an important structure to represent entities and the pairwise interactions between them. A graph $G$ can be written compactly as $G = (V, E)$, where $V = \{0, 1, \ldots, n-1\}$ is a set of $n$ vertices and $E \subseteq V \times V$ is the edge set such that if $(v_1, v_2) \in E$, we say vertex $v_1$ is connected with $v_2$.

Graph representations on Euclidean spaces is an old topic. A standard method is to closely mirror the edge capacities, governed by the Johnson-Lindenstrauss Lemma (JL lemma) proposed by Johnson and Lindenstrauss in [JL84].

The JL lemma was also geometrically described by the authors as "given $n$ points in Euclidean space, what is the smallest $k = k(n)$ so that these points can be moved into $k$-dimensional Euclidean space via a transformation which expands or contracts all pairwise distances by a factor of at most $1 + \varepsilon$?", whereas a more common form of the lemma now is

**Theorem 18.1** (JL Lemma)**.** *Let $\epsilon \in (0, \frac{1}{2})$. Let $Q \subset \mathbb{R}^d$ be a set of $n$ points and $k = \frac{20 \log n}{\epsilon^2}$. There exists a Lipschitz mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in Q$:*

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2. \tag{18.1}$$

There are various ways to prove the JL lemma (a brief overview can be found in [DG03]). Although not the tightest, the one I refer to uses the following lemma:

**Lemma 18.1** (Norm preservation)**.** *Let $x \in \mathbb{R}^d$. Assume that the entries in $A \subset \mathbb{R}^{k \times d}$ are sampled independently from $N(0, 1)$. Then*

$$P\{(1-\epsilon)\|x\|^2 \leq \|\frac{1}{\sqrt{k}} Ax\|^2 \leq (1+\epsilon)\|x\|^2\} \geq 1 - 2e^{-(\epsilon^2 - \epsilon^3)k/4}. \tag{18.2}$$

Given Lemma 8.1, we can readily prove the JL lemma.

*The proof of Theorem 8.1.* Choose the mapping $f$ in such a way that $f = \frac{1}{\sqrt{k}} Ax$, where $A$ is a $k \times d$ matrix the entry of which is sampled i.i.d. from a Gaussian $N(0, 1)$. We have

$$\begin{aligned} & P\{\exists u, v, \text{s.t. the mapping fails to satisfy Eq. (8.1)}\} \\ \leq & \sum_{u,v \in Q} P\{\text{s.t. the mapping fails to satisfy Eq. (8.1)}\} \\ \leq & \; 2n^2 e^{-(\epsilon^2 - \epsilon^3)k/4}. \end{aligned} \tag{18.3}$$

If we choose $k(\epsilon)$ properly (e.g., $k \geq \frac{20 \ln n}{\epsilon^2}$), the probability of (8.3) can be strictly smaller than 1, which means such a map that satisfies (8.1) always exists. □

Note the bound for $k$ does not guarantee that $k$ is smaller than $d$ (dimension reduction). It can also be seen that the requirement for $A$ to be sampled from $N(0, 1)$ can be further relaxed.

## 18.2 Discrete Fourier Transform

A time series is a special graph. Due to this special structure, we can use the discrete Fourier transform (DFT) to represent it. Let $\mathbf{x} = [x_0, x_1, \ldots, x_n - 1]^T \in \mathbb{R}^n$ be a time series. We can think of it as a graph where each $x_i$, $i = 0, \ldots, n-1$ is a node, and there is an edge between each pair of $(x_i - 1, x_i)$, $i = 1, \ldots, n-1$. Then the DFT of $\mathbf{x}$ is a vector $\mathbf{y}$ with its $i$-th entry defined as follows.

$$y_i = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j e^{-\frac{2\pi i j}{n}}.$$

DFT can also be represented compactly as

$$\mathbf{y} = F\mathbf{x},$$

where $F \in \mathbb{R}^{n \times n}$ and $F_{ij} = \frac{1}{\sqrt{n}} e^{-\frac{2\pi i j}{n}}$. Note that the matrix $F$ is an orthogonal matrix, so it is distance-preserving, i.e., $\|F\mathbf{x}_1 - F\mathbf{x}_2\|_2 = \|\mathbf{x}_1 - \mathbf{x}_2\|_2$

**Note**: The Fourier transform of a function can be viewed as an expansion of the function in terms of the eigenfunctions of the Laplacian operator (which is the second derivative operator in $\mathbb{R}$, or the second order finite difference operator if we discretize it). Generalizing this notion to graph, we can perform graph Fourier transform as we define the Laplacian matrix of a graph in the next.

## 18.3 Laplacian Matrix and Graph Embedding

**Definition 18.1** (Adjacency Matrix)**.** The adjacency matrix $A$ of a graph $G = (V, E)$ is a $|V| \times |V|$ matrix such that $A_{ij} = 1$ if $(v_i, v_j) \in E$ and $A_{ij} = 0$ otherwise.

**Definition 18.2** (Degree Matrix)**.** The degree matrix $D$ of a graph $G = (V, E)$ is a $|V| \times |V|$, diagonal matrix such that $D_{ii}$ is the degree of $v_i$, where the *degree* of a vertex is the number of edges that terminate at that vertex.

**Definition 18.3** (Laplacian Matrix)**.** The Laplacian matrix $L$ of a graph $G = (V, E)$ is defined as $D - A$, where $D$ is the degree matix and $A$ is the adjacency matrix.

**Definition 18.4** (Incidence Matrix)**.** Let each edge in the graph have an arbitrary but fixed orientation. The incidence matrix $\nabla$ is a $|E| \times |V|$ matrix defined as

$$\nabla_{ev} = \begin{cases} -1 & \text{if } e.start = v \\ 1 & \text{if } e.end = v \\ 0 & \text{otherwise} \end{cases}$$

**Proposition 18.1.** $L = \nabla^T \nabla$.

Given the above proposition, we further have:

**Proposition 18.2.** *For an undirected graph, $L$ is symmetric and positive semi-definite.*

**Proposition 18.3.** *The smallest eigenvalue of $L$ is 0. For a graph with only 1 connected component, the all-one vector is the only eigenvector associated with eigenvalue 0. In general, if a graph has $k$ connected components, then the multiplicity of eigenvalue 0 is $k$.*

To get a better understanding of the spectrum of $L$, we define

**Definition 18.5** (Fiedler Vector)**.** The smallest non-zero eigenvalue of $L$ is called the Fiedler value of a graph. The corresponding eigenvector is called the Fiedler vector.

The Fiedler value is the algebraic connectivity of a graph: the further from 0, the more connected.

**Proposition 18.4.** *The multiplicity of the Fiedler value is always 1.*

Therefore, the Fiedler vector $u$ is unique for each graph. And the mapping from each vertex $v_i \in V$ to the Fiedler vector $u_i$ is a one-dimensional embedding for the graph. Instead of mapping to a single line, if $k$ eigenvectors of more than one smallest eigenvalues of $L$ are used, we get a $k$ dimensional embedding for the graph.

## 18.4 Graph Convolutional Neural Networks

In computer vision, convolutional neural networks (CNN) leverages long-distance dependencies via multi-scaling filtering. Perhaps motivated by its success, there has been work focused on generalizing the conventional CNN structure from images (2D or 3D grid graphs) to graphs, hence the graph convolutional neural networks (GCNN) [Ano18].

Like in CNN, where distant pixels are made closer via pooling, we can use the adjacency matrix $A$ to traverse through the graph and make distant nodes closer, which leads to the *spatial GCNN*. Specifically, for input feature $\mathbf{x}_v$ at each node $v \in V$ of a graph, a typical spatial filtering operation is:

$$\mathbf{y}_v = \sigma\left(\sum_{k=0}^{K-1} W_k \left(\sum_{u \in V} (A^k)_{v,u} \mathbf{x}_u\right) + \mathbf{b}_0\right), \ \forall v \in V,$$

where $\sigma$ is the activation function (e.g., ReLU), $K \geq 0$ is a hyper-parameter that controls the filtering operation within $K$-step hops from each node $v \in V$, $W_k$, $k = 1, 2, \ldots, K$ and $\mathbf{b}_0$ are parameters to learn, and $A$ is the adjacency matrix (either normalized or not). Obviously, $A^k$ selects the nodes of a distance of $k$ hops from $v$.

Replacing the adjacency matrix by the Laplacian matrix, we get the *spectral GCNN*. Specifically, its filter is of the form:

$$\mathbf{y}_v = \sigma\left(\sum_{k=0}^{K-1} W'_k \left(\sum_{u \in V} (L^k)_{v,u} \mathbf{x}_u\right) + \mathbf{b}'_0\right), \ \forall v \in V,$$

where $W'_k$, $k = 1, 2, \ldots, K$ and $\mathbf{b}'_0$ are parameters to learn, and $L$ is the Laplacian matrix (either normalized or not).

**Theorem 18.2** (Spectral and spatial GCNN are equivalent)**.** *Let the spectral and spatial GCNN have the same activation function $\sigma$ and $K$ and both the adjacency matrix and Laplacian matrix are normalized (i.e., $\tilde{A}$ and $\tilde{L}$). Then for a same graph $G = (V, E)$, for any choice of $W_k$, $k = 1, \ldots, K$ and $\mathbf{b}_0$, there exist $W'_k$, $k = 1, \ldots, K$ and $\mathbf{b}'_0$ such that both GCNNs have the same transformation.*

*Proof.* Let $V = \{1, 2, \ldots, n\}$. Let $\mathbf{x}_i$ and $\mathbf{y}_i$ be the $d$-dimensional state and output at vertex $i \in V$. Let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ be the matrices representing them. Then the transformation of spatial GCNN can be written as

$$Y = \sigma\left(\sum_{k=0}^{K-1} W_k X \tilde{A}^k + \mathbf{b}_0 \mathbf{1}^T\right).$$

Similarly, the transformation of spectral GCNN is now

$$Y = \sigma\left(\sum_{k=0}^{K-1} W'_k X \tilde{L}^k + \mathbf{b}'_0 \mathbf{1}^T\right). \tag{18.4}$$

Considering $\tilde{A} = D^{-\frac{1}{2}} A D^{\frac{1}{2}}$, where $D$ is the degree matrix defined above and $\tilde{L} = I - \tilde{A}$, we can write Eq. (8.4) as

$$\begin{aligned}
Y &= \sigma\left(\sum_{k=0}^{K-1} W'_k X (I - \tilde{A})^k + \mathbf{b}'_0 \mathbf{1}^T\right) \\
&= \sigma\left(\sum_{k=0}^{K-1} W'_k X \sum_{i=0}^{k} \binom{k}{i} I^{k-i} (-1)^i \tilde{A}^i + \mathbf{b}'_0 \mathbf{1}^T\right) \\
&= \sigma\left(\sum_{k=0}^{K-1} \left(\sum_{i=k}^{K-1} W'_i \binom{i}{k} (-1)^{i-k}\right) X \tilde{A}^k + \mathbf{b}'_0 \mathbf{1}^T\right).
\end{aligned}$$

So apparently we have the following relations between the parameters

$$\begin{aligned}
\mathbf{b}_0 &= \mathbf{b}'_0 \\
W_{K-1} &= W'_{K-1} \\
W_{K-2} &= W'_{K-1} - W'_{K-1}(K-1) \\
&\cdots \\
W_k &= \sum_{i=k}^{K-1} W'_i \binom{i}{k} (-1)^{i-k}, \; k = K-3, K-4, \ldots, 0
\end{aligned}$$

□

# Bibliography

[Ano18] Anonymous. Graph2Seq: Scalable learning dynamics for graphs. *International Conference on Learning Representations*, 2018.

[DG03] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.