Supplemental Material for "DepthLap: A General Solution to Out-of-Sample Nodes for Network Embedding"

Appendix

Proof of Theorem 1

Lemma 1. *Let $\sigma(\cdot)$ be a nonconstant, bounded, and monotonically-increasing continuous function. For any $\epsilon > 0$ and any function $f \in C(\mathbb{R}^s)$, there exist $N \in \mathbb{N}$, $\pi_i, b_i \in \mathbb{R}$, and $\mathbf{w}_i \in \mathbb{R}^s$ ($i = 1, \cdots, N$), such that $g(\mathbf{x}) \triangleq \sum_{i=1}^{N} \pi_i \sigma\left(\mathbf{w}_i^T \mathbf{x} + b_i\right)$ satisfies $|g(\mathbf{x}) - f(\mathbf{x})| < \epsilon$ for all $\mathbf{x} \in \mathbb{R}^s$.*

Remark. It is the universal approximation theorem (Cybenko 1989; Hornik 1991). It basically suggests that a feed-forward neural network has the ability to approximate any continuous function on $\mathbb{R}^s$

Lemma 2. *For any $\epsilon > 0$ and any $f_i \in C(\mathbb{R}^s)$ ($i = 1, \ldots, d$), let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_d(\mathbf{x})]^\top$, there exists a feed-forward neural network $\mathbf{g} : \mathbb{R}^s \to \mathbb{R}^d$ that satisfies $\|\mathbf{g}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|_{l^2} < \epsilon$ for all $\mathbf{x} \in \mathbb{R}^s$.*

Remark. An obvious result of Lemma 1.

Theorem 1. *For any $\epsilon > 0$, any undirected graph $G = (V, E)$ (containing no self-loop, no parallel edge) whose connected components all have ≥ 3 nodes, and any $\mathbf{f} : \mathcal{V} \to \mathbb{R}^d$, there exists a parameter setting for DepthLap, such that: For any $v^* \in V$, after deleting all information (except G) related with v^*, DepthLap can still recover $\mathbf{f}(v^*)$'s value with error less than ϵ in terms of l^2-norm, by treating v^* as a new node and using Algorithm 1 on G.*

Proof. It is a constructive proof. Remember that $\mathbf{h}(\cdot) = [h_1(\cdot), \ldots, h_s(\cdot)]^\top$, We start by letting s to be the number of edges in G. Let the kth ($1 \leq k \leq s$) edge be (v_x, v_y). Now let us specify values of $h_k(v), a_v^{(k)}$ for all $v \in V$:

- $h_k(v_x) = 1, h_k(v_y) = -1, h_k(v) = 0, v \in V \setminus \{v_x, v_y\}$.
- $a_{v_x}^{(k)} = a_{v_y}^{(k)} = 1$.
- For all $v \in V \setminus \{v_x, v_y\}$, let $a_v^{(k)} = 0$ if v is a direct neighbor of either v_x or v_y, otherwise let $a_v^{(k)} = 1$.

Now let us define $H(v) \triangleq \{\mathbf{z} \in \mathbb{R}^s : |h_k(v)| - \gamma < |z_k| < |h_k(v)| + \gamma, k = 1, \ldots, s\}$, where $\gamma > 0$ is a very small real constant ($\gamma \ll \frac{1}{2}$). It can then be verified that $H(v) \cap H(v') = \emptyset$ if $v \neq v'$, as long as all connected components have ≥ 3 nodes. Hence $H(v)$ can be seen as a unique "representation" of node v.

Let us move on to specify the neural network, i.e. $\mathbf{g}(\cdot)$. First, notice that it is feasible to construct a function $\tilde{\mathbf{f}} : \mathbb{R}^s \to \mathbb{R}^d$ such that $\tilde{\mathbf{f}}(\mathbf{z}) = [\tilde{f}_1(\mathbf{z}), \ldots, \tilde{f}_d(\mathbf{z})]$, $\tilde{f}_i \in C(\mathbb{R}^s)(i = 1, \ldots, d)$, and $\tilde{\mathbf{f}}(\mathbf{z}) = \mathbf{f}(v)$ for all $\mathbf{z} \in H(v)$ and all $v \in V$. By Lemma 2, there exists a configuration of $\mathbf{g}(\cdot)$ such that $\|\mathbf{g}(\mathbf{z}) - \tilde{\mathbf{f}}(\mathbf{z})\|_{l^2} < \epsilon$ for all $\mathbf{z} \in \mathbb{R}^s$, and hence $\max_{\mathbf{z} \in H(v)} \|\mathbf{g}(\mathbf{z}) - \mathbf{f}(v)\|_{l^2} < \epsilon$ for all $v \in V$.

To prove the theorem, we now only need to ensure that: For any $v^* \in V$, after removing $\mathbf{f}(v^*)$, $h_k(v^*)$ and $a_{v^*}^{(k)}$ ($k = 1, \ldots, s$), Algorithm 1's prediction [1] of $\mathbf{h}(v^*)$ still lies in $H(v^*)$. Then $\mathbf{g}(\cdot)$ will be able to recover $\mathbf{f}(v^*)$ with error less than ϵ. To ensure this, we set $\eta_k = w_k \zeta_k$ and $\zeta_k > \frac{1}{w_k^2}(\frac{1}{\gamma} - 1)$, where w_k is the weight of the kth edge.

Now let us verify that Algorithm 1's prediction of $\mathbf{h}(v^*)$ lies in $H(v^*)$. Let the kth edge be (v_x, v_y), and Algorithm 1's prediction of $h_k(v^*)$ be z_k^*. Define $dist(u, v)$ to be the minimum distance (ignoring edge weights) between node u and node v. It can be verified case by case that (case 2 and case 3 might be a bit trickier than others, while case 4 is essentially the same case as case 3):

- If $v^* = v_x$ or $v^* = v_y$, then $1 - \gamma < |z_k^*| < 1$.
- If $dist(v^*, v_x) = dist(v^*, v_y) = 1$, then $|z_k^*| = 0$.
- If $dist(v^*, v_x) = 1$ and $dist(v^*, v_y) = 2$, then $|z_k^*| = 0$.
- If $dist(v^*, v_x) = 2$ and $dist(v^*, v_y) = 1$, then $|z_k^*| = 0$.
- If $dist(v^*, v_x) \geq 2$ and $dist(v^*, v_y) \geq 2$, then $|z_k^*| = 0$.

Also note that if $v^* = v_x$ or $v^* = v_y$, then $|h_k(v^*)| = 1$. Otherwise, $|h_k(v^*)| = 0$. So $|h_k(v^*)| - \gamma < |z_k^*| < |h_k(v^*)| + \gamma$ for $k = 1, \ldots, s$. As a result, Algorithm 1's prediction of $\mathbf{h}(v^*)$ lies in $H(v^*)$. □

Proof of Theorem 2

Theorem 2. *Theorem 1 will not hold if DepthLap does not model second-order proximity. In other words, there will exist $G = (V, E)$ and $\mathbf{f} : \mathcal{V} \to \mathbb{R}^d$ that DepthLap cannot model, if ζ_k is fixed to zero.*

Proof. It is proved with a counterexample (see Figure 1). Let us consider a simple graph $G = (V, E)$ with $V =$

[1] Note that during prediction, 1 is used in place of $a_{v^*}^{(k)}$, since $a_{v^*}^{(k)}$ is deleted and v^* is treated as a new node.

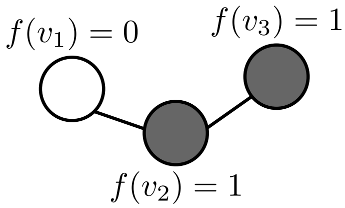

Figure 1: A simple counterexample. Each node decides its own $f(\cdot)$ by XOR-ing other nodes that are ≤ 2 steps away.

$\{v_1, v_2, v_3\}$ and $E = \{(v_1, v_2), (v_2, v_3)\}$, and $f : \mathcal{V} \to \mathbb{R}$ ($d = 1$ in this counterexample) that satisfies $f(v_1) = 0, f(v_2) = f(v_3) = 1$. If second-order proximity is not modeled, i.e. ζ_k is fixed to zero ($k = 1, \ldots, s$), then it is obvious that Algorithm 1's prediction of $\mathbf{h}(v_1)$ and $\mathbf{h}(v_3)$ will inevitably be the same. As a result, Algorithm 1 will predict $f(v_1)$ and $f(v_3)$ to be the same, which is incorrect. $\square$

References

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*.

Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*.