Learning Latent Representations Using Evolving Sets

Jainul N. Vaghasia
Department of Computer Science
University of Washington
Seattle, WA 98105
jnv3@cs.washington.edu

Abstract

In this project, we exploit volume-biased evolving set processes to embed vertices into a continuous vector space based on local communities of any given vertex. Such embedding can be naturally feed into various statistical models to perform learning on graphs. We demonstrate the method on the task of multi-label classification of users in the Blog-Catalog and p2p-Gnutella social graph. The approach is highly scalable compared to spectral clustering and performs about the same on the given task.

The code is published online and is available here: https://www.github.com/JainulV/evowalk

1 Problem Statement

We begin by introducing the following problem of classifying members of a graph into categories as posed by Perozzi et al. [2014]. Let $G(V, E)$ be the graph with V being the set of vertices and E being the set of edges. Given a partially labelled $G_L(V, E, X, Y)$ where $X \in \mathbb{R}^{|V| \times s}$ are attributes in a feature space of s dimensions and $Y \in \mathbb{R}^{|V| \times |\mathcal{Y}|}$ where $\mathcal{Y}$ is the set of labels. We aim to learn a mapping f from X to $\mathcal{Y}$. This is known as the relational classification problem and is prevalent in the literature in the form of inference problem on undirected Markov networks.

Our goal is to utilize the network topology to learn the latent feature representation of vertices $X_E \in \mathbb{R}^{|V| \times d}$ for a preferably small d. These structural features can then be used to augment the attributes X to facilitate classification decision f using any traditional classification algorithm.

2 Local Communities based Representation

Perozzi et al. [2014] outline the following criteria for their representations based on truncated random walks:

- Adaptability: We require to perform minimal learning to adapt the representation to changes in the network.
- Continuity: Having a continuous representation allows the classification algorithm to generate a robust classification boundary.
- Community aware: The metric defined on latent representation should be symbolic of the similarity of vertices in the network.
- Low dimensional: Low dimensional representations generalize well while providing obvious speed ups in performance.

Final Project Option 3 for CSE 521: Design and Analysis of Algorithms, Autumn 2018.

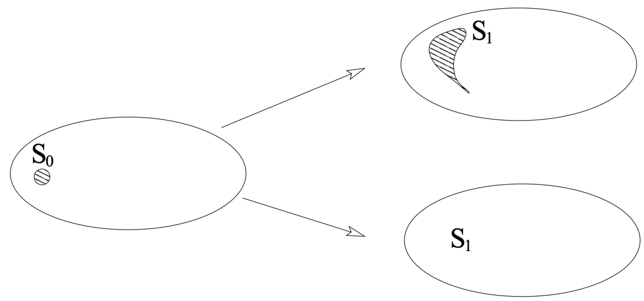

Figure 1: When Z is small, the set grows. When Z is large, the set shrinks. (Montenegro)

Our method uses the volume-biased Evolving Set Processes (Andersen et al. [2016]) in place of truncated random walks and combines them with language modeling to offer representations that satisfy all of the above criteria. See figure 2 for an illustration of the method.

ESPs are defined as Markov chain on subsets of the vertex set V. At the current state S, the next state S' is chosen by picking a threshold Z uniformly from $[0, 1]$ and letting $S' = \{u : p(u, S) \geq Z\}$ where $p(u, S)$ is the transition probability of transitioning from u to some vertex in S (See figure 1 for an example).

Let the transition kernel of ESP be denoted by $\mathbf{K}(S, S')$. Then the volume-biased ESPs are Markov chains with the following transition kernel:

$$\widehat{\mathbf{K}}(S, S') = \frac{\text{vol}(S')}{\text{vol}(S)} \cdot \mathbf{K}(S, S').$$

As Andersen et al. [2016] show, volume-biased ESPs can be used to generate a local cluster of vertices around a given seed vertex in time sublinear in the size of the input graph.

Now we would like to estimate the likelihood of sequence of vertices generated by the volume-biased ESP. Since the goal is to learn a latent representation, we introduce a mapping $\Phi : v \in V \mapsto \mathbb{R}^{|V| \times d}$ representing the social representation associated with each vertex. Then we wish to maximize the likelihood

$$\mathbb{P}\left[v_i | \Phi(v_1), \ldots, \Phi(v_{i-1})\right].$$

However, this is often infeasible to compute, so we turn to the language modeling relaxation idea of Perozzi et al. [2014]. In terms of vertex representation modeling, this yields us, for some window size w, the following optimization problem:

$$\max_{\Phi} \log \mathbb{P}[\{v_{i-w}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+w}\} | \Phi(v_i)].$$

Here, instead of using the context to estimate probability of any given vertex, we compute the probability of context given the vertex. Furthermore, it removes the ordering constraint on the context vertices which allows us to capture the notion of "nearness" that is provided by the ESPs. This relaxation idea comes from language modeling literature: consider the analogous situation where the sequence of vertices can be considered as a sentence and we would like to maximize the likelihood of a given word over the entire training corpus.

As claimed, the method produces a low dimensional, continuous mapping Φ that exploits the local graph clustering offered by the ESPs. As a result of the relaxation, it also adapts fairly well to changes in the network. In fact, Perozzi et al. [2014] provide a streaming algorithm for their truncated random walks which can easily be modified to fit out purpose.

In the next section, we offer three main algorithms: GenerateSample that simulates the volume-biased ESP (Andersen et al. [2016]), EvoWalk that generates the near vertices and SkipGram that performs the language modeling optimization suggested above (Mikolov et al. [2013]).

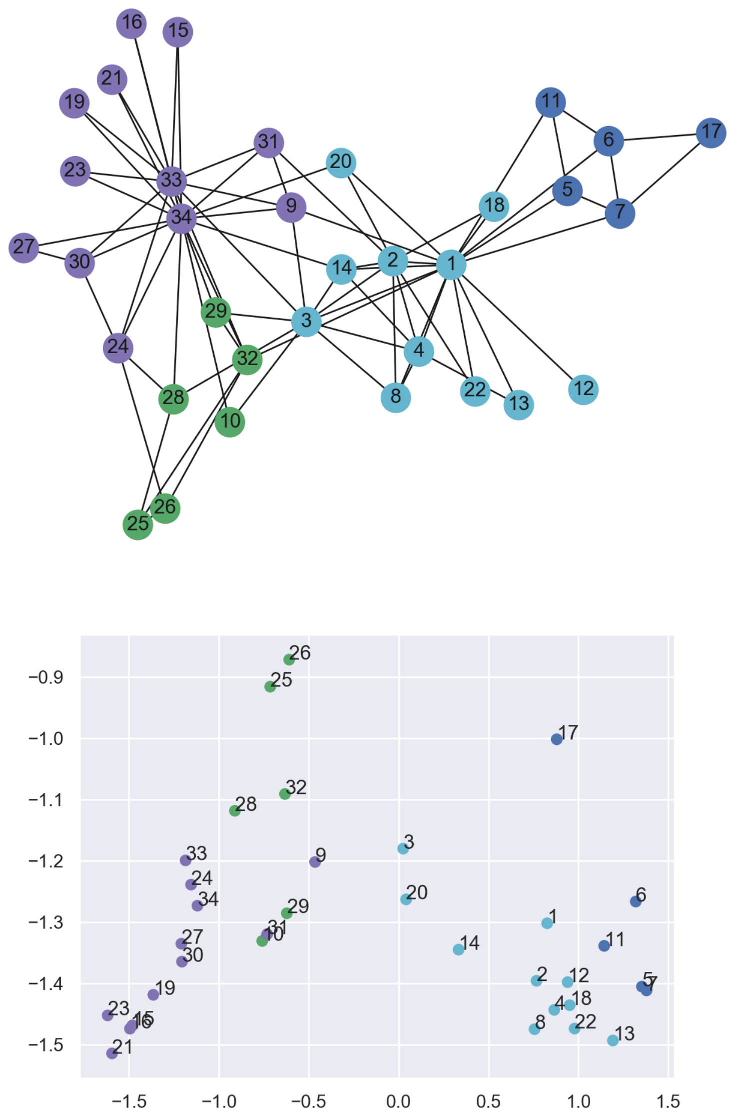

Figure 2: The proposed method learns a continuous embedding of vertices in $\mathbb{R}^d$ that preserves the community structure of vertices in the input graph. Top: Zachary [1976]'s karate graph where the color of vertices represents clustering. Bottom: An embedding of the input graph in $\mathbb{R}^2$. Note the strong communal preservation in the embedding.

3 Algorithms

generate sample ...

Algorithm 1: EvoWalk(G, w, d, γ, t)

```
Input  : graph G(V, E), window size w, embedding dimension d,
         ESPs per vertex γ, ESP stopping time t.
Output : A matric of vertex representations Φ ∈ ℝ^{|V|×d}
Initialization: Sample Φ from U^{|V|×d}
for j = 0 → γ do
    O ← Shuffle(V)
    for v_i ∈ O do
        for φ ∈ {0.1, 0.2, ..., 1} do
            k ← log Vol(V)
            W_{v_i} ← GenerateSample(G, v_i, t, ∞, φ, k)
            ;                       // Use EvoPar for result with high probability
            if W_{v_i} ≠ ∅ then
                break
            end
        end
        SkipGram(Φ, W_{v_i}, w)
    end
end
```

EvoWalk generates γ ESPs for each vertex with minimal conductance to ensure homophily and relatively low volume to ensure locality. In order to generate the minimal conductance set, we call GenerateSample with an epsilon net on the values of $\phi \in \{0.1, 0.2, \ldots, 1\}$ and break on first instance. Then we perform the relaxed optimization suggested in the previous section for each cluster found by ESP in EvoWalk. SkipGram is a prevalent algorithm in language modeling literature for this procedure:

Algorithm 2: SkipGram($\Phi, \mathcal{W}_{v_i}, w$)

```
for v_j ∈ W_{v_i} do
    for u_k ∈ W_{v_i}[j − w : j + w] do
        J(Φ) ← − log ℙ[u_k|Φ(v_j)] ;              // Negative log-likelihood
        Φ ← Φ − α · ∂_Φ J ;                // Gradient descent; learning rate α
    end
end
```

SkipGram (Mikolov et al. [2013]) iterates over vertices u_k in window w around all vertices v_j in the given local cluster $\mathcal{W}_{v_i}$. For all v_j, we maximize $\mathbb{P}[u_k|\Phi(v_j)]$ by performing a gradient descent on its negative log-likelihood. The following theorem provides an efficient way to calculate $\mathbb{P}[u_k|\Phi(v_j)]$.

Theorem 3.1. *(Hierarchical Softmax) There exists an algorithm that computes $\mathbb{P}[u_k|\Phi(v_j)]$ in time $O(\log|V|)$.*

Proof. Assign the vertices to the leaves of a binary tree. Then if $b_0, b_1, \ldots, b_{\lceil \log|V| \rceil}$ is the path from root to u_k, we have

$$\mathbb{P}[u_k|\Phi(v_j)] = \prod_{l=1}^{\lceil \log|V| \rceil} \mathbb{P}[b_l|\Phi(v_j)],$$

where $\mathbb{P}[b_l|\Phi(v_j)]$ can be modeled by a binary classifier assigned to the parent of b_l. The cost is, therefore, equal to traversal from root to a leaf in a balanced binary tree, and hence, the theorem. $\square$

4 Empirical Study

4.1 Datasets

4.2 Results

5 Conclusion

References

Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *J. ACM*, 63(2):15:1–15:31, May 2016. ISSN 0004-5411. doi: 10.1145/2856030. URL `http://doi.acm.org/10.1145/2856030`.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

Ravi Montenegro. Evolving sets and distances. URL `http://faculty.uml.edu/rmontenegro/8843/notes/lecture16.pdf`.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, 2014. doi: 10.1145/2623330.2623732. URL `http://dx.doi.org/10.1145/2623330.2623732`.

Wayne Zachary. An information flow model for conflict and fission in small groups1. *Journal of anthropological research*, 33, 11 1976. doi: 10.1086/jar.33.4.3629752.