University of
Cambridge

Mathematics Tripos

Part III Essay

Walking Deeper on Dynamic Graphs

November 19, 2019

Written by
Joshua Snyder

Contents

1 Introduction

Essay Descriptor

Walking Deeper on Dynamic Graphs: Learning Latent Representations with Random Walks for Image Classification
In the era of big data, graph representation is a natural and powerful tool for representing big data in real-world problems [1],[2],[4]; some examples include data coming from medical records, social networks, recommendation systems and transport systems. A challenging question when using graph representation is how to learn latent representations on multi-label networks for several classification tasks, and a seminal algorithm for this is the DeepWalk technique using random walks [1].
We propose two questions for investigation in this essay. Firstly, we hope that students will develop a rigorous mathematical underpinning for the DeepWalk algorithm, in the spirit of convergence guarantees.
Secondly, we seek to investigate the connection of DeepWalk to dynamic graphs. Many realworld events are dynamic - for example, in a social network new users are constantly added- while most of the body of literature is based on the unrealistic assumption that the graph is static. From the learning point of view, this assumption has a negative impact in the computations, as the graph has to be re-learned each time that an instance changes. We also hope that students will also discuss some open questions that they find interesting.
Relevant Courses
Useful: Background knowledge in Machine Learning and Statistics is helpful, as is probability to the level of Part II Applied Probability. Some content from Part III Mixing Times of Markov Chains, on the long-time behaviour of random walks on graphs, may also be useful.
References

B. Perozzi, R. Al-Rfou and S. Skiena. Deepwalk: Online learning of social representations. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 701- 710, 2014.

A.I Aviles-Rivero, N. Papadakis, R. Li, SM. Alsaleh, R. T Tan and C-B Schonlieb. Beyond Supervised Classification: Extreme Minimal Supervision with the Graph 1-Laplacian. arXiv preprint:1906.08635

HP Sajjad, A Docherty and Y. Tyshetskiy, Y. Efficient representation learning using random walks for dynamic graphs. arXiv preprint arXiv:1901.01346.

M Valko. Lecture Notes on Graphs in Machine Learning, http://researchers.lille.inria.fr/ valko/hp/mva-ml-graphs.php, 2019.

1.1 Overview

1.2 Notation

Let $\mathbb{N}$ denote $\{1, 2, 3, ...\}$ and $\mathbb{N}_0$ denote $\mathbb{N} \cup \{0\}$ with $[n] = \{1, 2, ..., n\}$ and let $[0] = \emptyset$.

Definition (string)**.** Given a set S, a *string* is a function $u : [n] \to S$ for some $n \in \mathbb{N}$. n represents the length of u and is denoted by $|u|$. The empty string is denoted by $\emptyset$.

2 WQOs

> **Definition** (quasi-ordering (QO)). Given a set A, a binary relation $\preccurlyeq$ is a *quasi-ordering* if and only if it is both reflexive and transitive.

A quasi-order that is also antisymmetric represents a partial order. It is total if every two elements are related. If two elements are incomparable, we denote this as $x|y$. We let $\prec$, $\succ$ and $\succcurlyeq$ take their usual interpretations.

> **Definition.** Given a QO $\preccurlyeq$ over a set A, an infinite sequence $(x_i)_{i\geq 1}$ is an infinite descending chain if $x_1 \succ x_2 \succ x_3 \succ ...$
> It is an infinite antichain if instead $x_i|x_j$ for all $1 \leq i < j$
> We say $\preccurlyeq$ is well-founded iff there exist no infinite descending chains with respect to $\succ$.

What naturally follows from this definition is the concept of a 'nice' quasi-ordering. This is defined as a well quasi-ordering:

> **Definition** (well quasi-ordering (WQO)). Given a set A and a corresponding quasi-order $\preccurlyeq$, an infinite sequence $(x_i)_{i\geq 1}$ of elements in A is said to be *good* iff there exists positive integers i, j such that $i < j$ and $x_i \preccurlyeq x_j$. if a sequence is not said to be good, naturally it called is bad.
> A QO is a *well quasi-ordering (WQO)* iff every infinite sequence is good.

Note that by definition every QO on a finite set is a WQO. The above is one way to define a WQO and there are other useful ways to define one. Importantly, the definitions are equivalent.

> **Lemma 2.1** (Equivalence of WQO definitions). *Given a quasi-order $\preccurlyeq$ of a set A, the following are equivalent:*
>
> 1. *Every infinite sequence is good*
> 2. *There are no infinite decreasing chains or infinite antichains*
> 3. *Every quasi-order extending $\preccurlyeq$ (including itself) is well-founded.*

Proof. Check here whether the third definition is indeed useful, if so replicate the proof from Gallier, else just prove the equivalence of definitions 1 and 2. $\square$

It is interesting to note that the property of being well-founded defines a WQO, though well-foundedness is a much weaker statement than a WQO.

2.1 Fundamental WQO Theory

This section will cover the basics of WQO theory and will give the reader the machinery with which to tackle Kruskal's Tree Theorem and the consequences of which this essay is about. It is standard and mostly displayed as in a paper by H. Gallier [**?**].

The first lemma to address and one that plays an important part in the result of Kruskal's Tree Theorem (addressed in Chapter XX) comes from a paper by Nash-Williams [**?**].

Lemma 2.2. *Given a quasi-order $\preccurlyeq$ on a set A, the following are equivalent:*

1. *$\preccurlyeq$ is a WQO on A*
2. *Every infinite sequence $s = (s_i)_{i\geq 1}$ over A contains an infinite subsequence $s' = (s_{f(i)})_{i\geq 1}$ such that $s_{f(i)} \preccurlyeq s_{f(i+1)}$ for all $i > 0$*

Proof. It is clear that (2) $\implies$ (1), we shall show the converse. Assume that $\preccurlyeq$ is a WQO. We say that a member s_i of a sequence is terminal if there is no $j > i$ such that $s_i \preccurlyeq s_j$. We claim that the number of terminal elements in the sequence s is finite. Suppose not, then this infinite sequence of terminals in s form a bad sequence, contradicting the fact that $\preccurlyeq$ is a WQO.
Thus there exists $N > 0$ such that s_i is not terminal for all $i \geq N$. Define a strictly monotonic function f as follows:

$$\begin{aligned} f(1) &= N \\ f(n+1) &= \inf\{m : s_{f(n)} \preccurlyeq s_m \text{ and } f(n) < m\} \end{aligned}$$

Note that f is monotonic by definition and by the definition of N, the recursive definition is indeed valid, since for every $i > N$ there is a $j > i$ with $s_i \preccurlyeq s_j$. □

A corollary of Theorem 2.2 is that we can stitch together two WQOs $\langle \preccurlyeq_1, A_1 \rangle$ and $\langle \preccurlyeq_2, A_2 \rangle$ to get a WQO of $A_1 \times A_2$ as follows.

Lemma 2.3. *Let $\langle \preccurlyeq_1, A_1 \rangle$ and $\langle \preccurlyeq_2, A_2 \rangle$ be WQOs, then the quasi-order $\preccurlyeq$ on $A_1 \times A_2$ defined by:*

$$(a_1, a_2) \preccurlyeq (a_1', a_2') \textit{ iff } a_1 \preccurlyeq_1 a_1' \textit{ and } a_2 \preccurlyeq_2 a_2'$$

is a WQO

Proof. Any infinite sequence s in $A_1 \times A_2$ defines an infinite sequence of pairs $(x_i, y_i) \in A_1 \times A_2$. The $(x_i)_{i\geq 1}$ form an infinite sequence in A_1 so by Lemma XX, since $\preccurlyeq_1$ is a WQO, there is some infinite subsequence $t = (x_{f(i)})_{i\geq 1}$ of (x_i) such that $t_{f(i)} \preccurlyeq_1 t_{f(i+1)}$ for all $i > 0$.
Since $\preccurlyeq_2$ is a WQO over A_2, $t' = (y_{f(i)})_{i\geq 1}$ is an infinite sequence over A_2. Thus, since there are no bad sequences in A_2, there is some i, j with $f(i) < f(j)$ and $y_{f(i)} \preccurlyeq_2 y_{f(j)}$. Therefore we have that $(x_{f(i)}, y_{f(i)}) \preccurlyeq (x_{f(j)}, y_{f(j)})$ which shows that $\preccurlyeq$ is a WQO. □

3 Kruskal's Tree Theorem

In this section we will display a proof of Kruskal's Theorem...

3.1 Higman's Theorem

First we will address a key theorem in the proof, one by Higman...

> **Definition** ($A^{<\omega}$)**.** Given a quasi-order (A,), define $A^{<\omega}$to be the set of all finite strings $A*$ ordered by $<<$ such that for any strings $u = u_1...u_n$, $v = v_1...v_n$ with $1 \leq m \leq n$ $u << viff \exists 1 \leq$

> **Theorem 3.1** (Kruskal's Tree Theorem)**.** */indexKruskal's Tree Theorem*

4 Friedmann's Finitization

5 References

Index