Motif-Aware Graph Embedding

Hoang Nguyen
Tokyo Institute of Technology
hoangnt@ai.cs.titech.ac.jp

Nukui Shun
Tokyo Institute of Technology
nukui.s@net.c.titech.ac.jp

Tsuyoshi Murata
Tokyo Insitute of Techonology
murata@c.titech.ac.jp

ABSTRACT

Given a large complex graph, how can we learn a lower dimension vector representation of each vertex that preserves structural information? Recent advancements in graph embedding have used word embedding techniques and deep architectures to propose a feasible answer to this question. However, most of these work considers the notion of "neighborhood" by node adjacency only. In this paper, we propose a novel graph embedding algorithm that employs motif structures into the latent vector representation learning process. By contrasting between sets of nodes created by random walks and sets of nodes created by biased *motif walk*, we show that embedding results of our algorithm are more accurate in various benchmark graph mining tasks compared to existing algorithms. The source code and results of our algorthms is available online at https://github.com/anonsyuushi/mage.git.

CCS Concepts

•**Computing methodologies** → **Learning latent representations;** *Neural networks;* •**Mathematics of computing** → *Graph theory;*

Keywords

Distributed representation; Graph embedding; motif; autoencoder; word2vec; MAGE

1. INTRODUCTION

Meaningful distributed representation of a high dimensional sparse dataset has proven to be useful for various machine learning tasks. For example, the **dense vector representation** of words in word2vec framework [2] has enabled machine learning researchers to **put applications and citation of word2vec here**.

Recently, starting from 2014 with the *DeepWalk* algorithm [4], there has been many proposed algorithms to encode a network's component such as vertices or edges into a high dimensional real vector. The motivation behind these algorithms is to learn a dense representation of the network anologous to learning a dense representation of a word [2]. By encoding the whole network into vectors, graph embedding algorithms have enabled network researcher to use the power of neural network techniques on network data [**?**]. Basically this solve the sparsity problem of the network.

Our algorithm outperfroms every other algorithms on their test. We conducted experiment throughly and carefully. Each experiment is ran with the same condition 10 times and take average, we have all the deviation stuff that others don't have. Really, it's really a very good paper. Please accept it so I can go to UK for a vacation. Please.

The good news is, with only a handful of manual settings[1], the LaTeX document class file handles all of this for you.

The remainder of this document is concerned with showing, in the context of an "actual" document, the LaTeX commands specifically available for denoting the structure of a proceedings paper, rather than with giving rigorous descriptions or explanations of such commands.

2. RELATED WORK

In the context of topological graph theory, an embedding refers to a representation of a graph G on a surface Σ. Graph embedding can also be viewed as dimension reduction when the dimensionality of the surface Σ is less than the dimensionality of the graph. Meaningful graph embedding, in practice, is low dimensionality vector representations of vertices preserving some analytical properties of the graph. Generally, there are two main approaches to obtain high-quality graph embeddings: affinity matrix factorization and machine learning with neural networks.

2.1 Matrix factorization

== NUKUI ==

2.2 Skipgram model

Our work in this paper is directly related to the *word2vec* model [**?**]. More specifically, the

The literature of graph embedding start with matrix decomposition techniques [**?**]. Super professor et. al. perform amazing works and achieved many success. Traditionally, graph clustering procedure where similarity between nodes or edges is defined and group together by some similarity

[1]Two of these, the `\numberofauthors` and `\alignauthor` commands, you have already used; another, `\balancecolumns`, will be used in your very last run of LaTeX to ensure balanced column heights on the last page.

Web Search and Data Mining '17 Cambridge, USA

DOI: 10.475/123_4

metric. This leads to the notation of closeness and hence community. However, one limitation of these classical graph embedding technique is that it depends on matrix decompositon which is very computational expensive.

Recently, with the emerging of deep neural network models such as autoencoder [**?**], restricted Bolzman machine [**?**], and some other that I will fill here later. Inspired by these advancement of natural language processing neural network models, Peperozi et. al. [4] proposed a graph embedding framework named DeepWalk. By taking advantage of the network structure and create an artificial "node corpus" by performing random walk, DeepWalk has achieved good performance and inspired any following researches. Since 2014, many graph embedding model based on random walk has been proposed such as LINE, GraRep, etc. These embedding algorithm proved its efficiency and usefullness in various graph mining tasks and classification problem in large graph.

Out of the aforementioned embedding algorithms, LINE has taken our notice. LINE model improved DeepWalk by incooporating the notation of second order proximity to the learning task. This leads to its outstanding performance without other auxilariy information. There are some research that has good result in embedding too, but out of all research that only use graph structure, LINE is the best. We think that the key to LINE's success was the second order proximity added to the original embedding scheme. However, LINE perform extremely well in network with citation-like structure. In other word, LINE is very good for network with wedge motif. From this observation, we think that if we can incooporate the most popular substructure of the graph to the embedding scheme, we can learn better vector representation.

2.3 Motif in Graph

Motif is defined as a small subgraph that has some funny characteristic [**?**], maybe cite some of Jure's work here.

Definition 1. A graph is represented as $G = (V, E)$, where V is the set of vertices and E is the set of edges between vertices. Each edges $e \in E$ is an ordered pair $e = (u, v)$ where $u, v \in V$. A graph is called *undirected* when $(u, v) \equiv (v, u)$, and *directed* when $(u, v) \not\equiv (v, u)$.

3. MOTIF-AWARE GRAPH EMBEDDING

As mentioned in the previous sessions, our work aims to improve embedding quality by manipulating the graph data generation process. We have two data generation processes in our framework. The first process uses conventional random walk and a "skip window" to generate positive samples, while negative samples are picked from a noise distribution $p_n(x)$. The second uses a pre-defined *motif walk* to generate positive samples, while negative samples is generated from a contrasting distribution:

$$p_{\mathrm{mc}}(x) = \boldsymbol{f}\left(p^t_{\mathrm{m}}(x) \;||\; p^t_{\mathrm{r}}(x)\right), \tag{1}$$

where $\boldsymbol{f}(.)$ is chosen to be a distance function that yields high probability for node x when x is more likely to appear in random walk starting from node t than in motif walk starting from node t. $p^t_m(x)$ and $p^t_r(x)$ are distribution of nodes count in motif walk and random walk starting from vertex t.

Our model is a variant of the skipgram model [2] using the softmax function, in which we have the conditional probability of a "class" vertex (context) given the "target" vertex (word) as follow:

$$p\left(v_{\mathrm{class}} \mid v_{\mathrm{target}}\right) = \frac{\exp(\omega_c^\top \cdot \omega_t)}{\sum_{i=1}^{|V|} \exp(\omega_i^\top \cdot \omega_t)} \tag{2}$$

Despite the efficiency and simplicity of this model, the task of computing the normalization factor requires summing all over the graph's vertices is intractable for large graph. To solve this normalization problem, estimation techiques such as hierachial softmax [3], noise contrastive estimation [1], and negative sampling [2] are employed in the recent graph embedding models [4, 5, 6]. We define the loss function with negative sampling for our model as follow:

$$\mathcal{L} = \log p\left(v_{\mathrm{class}} \mid v_{\mathrm{target}}\right) = \alpha\mathcal{L}_{\mathrm{r}} + (1-\alpha)\mathcal{L}_{\mathrm{mc}} \tag{3}$$

$$\begin{aligned}
\mathcal{L}_{\mathrm{r}} &= \log\sigma\left(\omega_c^\top \cdot \omega_t\right) + \sum_{i=1}^{k} \mathbb{E}_{\omega_s \sim p_{\mathrm{n}}(\omega)}\left[\sigma(-\omega_s^\top \cdot \omega_t)\right] \\
\mathcal{L}_{\mathrm{mc}} &= \log\sigma\left(\omega_c^\top \cdot \omega_t\right) + \sum_{i=1}^{k} \mathbb{E}_{\omega_s \sim p_{\mathrm{mc}}(\omega)}\left[\sigma(-\omega_s^\top \cdot \omega_t)\right]
\end{aligned} \tag{4}$$

3.1 Graph sampling process

The first data generation process is similar to that of the Deepwalk model. The difference in our model is negative sampling from the vertex degree distribution is used to estimate the valid distribution. Algorithm 1 describes the positive and negative samples generating from random walk process.

Algorithm 1: `gen_rand`: sample by random walk

```
Data: Graph G = (V, E)
Input: walkLength, skipWindow, numSkip, numNeg,
       distort, random_walk
Output: (targets, classes, labels)
begin
    targets ← [ ]; classes ← [ ]; labels ← [ ];
    id_list ← Shuffle(V);
    for i ∈ idList do
        walk ← randomWalk(start=i,
         length=walkLength); for j ∈ walk do
            for j ∈ range(numSkip) do
                targets.append(j);
                classes.append(random.choice(walk[j-
                 skipWindow:j+skipWindow]);
                labels.append(1.0);
            for j ∈ range(numNeg) do
                targets.append(j);
                classes.append(random.choice(V,
                 distort));
                labels.append(0.0);
    return (targets, classes, labels);
```

The second data generation process is the core of our method. Positive data samples are generated using a biased random walk, which is called `motif_walk`. For the negative

samples generation, we perform unbiased random walk and select vertices that appear frequenty in the random walk, but less frequenty in the postive motif walk. The intuition for this technique comes from our hypothesis that vertices in the same motif cluster are more likely to be related than vertices in the random walk. As discussed by Gutmann and Hyvärinen in [1] for the choice of noise distribution in practice, the more similar the noise distribution to the true distribution leads to better learning result.

Algorithm 2: `gen_motif`: sample by motif walk

Data: Graph $G = (V, E)$
Input: walkLength, skipWindow, numSkip, numNeg, contrastIter, `motif_walk`
Output: (targets, classes, labels)
begin
  targets ← [ ]; classes ← [ ]; labels ← [ ];
  idList ← Shuffle(V);
  for $i \in$ *idList* **do**
    posSet ← { };
    for $j \in$ *range(contrastIter)* **do**
    walk ← `motif_walk`(start=i, length=walkLength); **for** $j \in$ *walk* **do**
      for $k \in$ *range(numSkip)* **do**
        targets.`append`(j)
        classes.`append`(`random.choice`(walk[j-skipWindow:j+skipWindow])
        labels.`append`(1.0)
      for $k \in$ *range(numNeg)* **do**
        targets.`append`(j)
        classes.`append`(`random.choice`(V))
        labels.`append`(0.0)
  return *(targets, classes, labels)*

Definition 2.

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.[2] LaTeX handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

Because the entire article is contained in the **document** environment, you can indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph.

3.2 Neural network optimization

3.3 Training procedure

[2]This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

4. EXPERIMENTS

4.1 Type Changes and *Special* Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[3] typeface, but that is handled by the document class file. Take care with the use of[4] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

You can use whatever symbols, accented characters, or non-English characters you need anywhere in your document; you can find a complete list of what is available in the *LaTeX User's Guide*[**?**].

4.2 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

4.2.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin. . .\end` construction or with the short form `$. . .$`. You can use any of the symbols and structures, from α to ω, available in LaTeX[**?**]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

4.2.2 Display Equations

A numbered display equation – one set off by vertical space from the text and centered horizontally – is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{5}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{6}$$

just to demonstrate LaTeX's able handling of numbering.

[3]A third footnote, here. Let's make this a rather short one to see how it looks.
[4]A fourth, and last, footnote.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

4.3 Citations

Citations to articles [**?**, **?**, **?**, **?**], conference proceedings [**?**] or books [**?**, **?**] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [**?**]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[**?**].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed or supported.

4.4 Tables

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material is found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed dvi output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed dvi output of this document.

4.5 Figures

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of **.eps** files to be displayable with LaTeX. If you work with pdfLaTeX, use files in the **.pdf** format. Note that most modern TeX system

Figure 1: A sample black and white graphic.

Figure 2: A sample black and white graphic that has been resized with the `includegraphics` command.

will convert **.eps** to **.pdf** for you on the fly. More details on each of these is found in the *Author's Guide*.

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. and don't forget to end the environment with figure*, not figure!

4.6 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

This uses the **theorem** environment, created by the `\newtheorem` command:

THEOREM 1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)dt = G(b) - G(a).$$

The other uses the **definition** environment, created by the `\newdef` command:

Definition 3. If z is irrational, then by e^z we mean the unique number which has logarithm z:

$$\log e^z = z$$

Two lists of constructs that use one of these forms is given in the *Author's Guidelines*.

There is one other similar construct environment, which is already set up for you; i.e. you must *not* use a `\newdef` command to create it: the **proof** environment. Here is a example of its use:

PROOF. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)}\right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

Table 2: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

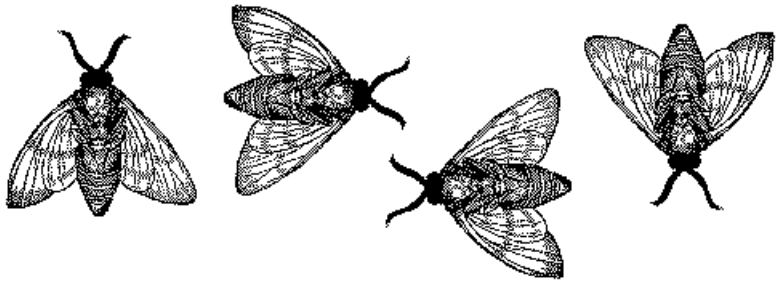

Figure 3: A sample black and white graphic that needs to span two columns of text.

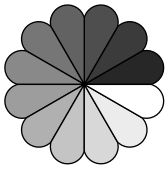

Figure 4: A sample black and white graphic that has been resized with the `includegraphics` command.

Complete rules about using these environments and using the two different creation commands are in the *Author's Guide*; please consult it for more detailed instructions. If you need to use another construct, not listed therein, which you want to have the same formatting as the Theorem or the Definition[**?**] shown above, use the `\newtheorem` or the `\newdef` command, respectively, to create it.

A *Caveat* for the TEX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TEX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LATEX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

5. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LATEX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

6. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

7. REFERENCES

[1] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, page 6, 2010.

[2] T. Mikolov and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.

[3] F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.

[6] Z. Yang, W. W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*. ICML, 2016.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used

to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations.

Display Equations.

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\addition-alauthors` command at the start of the document.

A.6 References

B. MORE HELP FOR THE HARDY

The sig-alternate.cls file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.