EPJ manuscript No.
(will be inserted by the editor)

Learning Network Representations

A review with applications to complex networks

Luis G. Moyano[1,a]

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) and Facultad de Ciencias Exactas y Naturales, Universidad Nacional de Cuyo, Mendoza, Argentina

Abstract. In this review I present several representation learning methods, and discuss the latest advancements with emphasis in applications to network science. Representation learning is a set of techniques that has the goal of efficiently mapping data structures into convenient latent spaces. Either for dimensionality reduction or for gaining semantic content, this type of feature embeddings has demonstrated to be useful, for example, for node classification or link prediction tasks, among many other relevant applications to networks. I provide a description of the state-of-the-art of network representation learning as well as a detailed account of the connections with other fields of study such as continuous word embeddings and deep learning architectures. Finally, I provide a broad view of several applications of these techniques to networks in various domains.

1 Introduction

Networks describe sets of relationships among entities, generally encoded in the form of a graph, i.e. entities are represented by nodes, and relationships by links connecting those nodes. Networks are central in many areas of research and have been the focus of interest in a vast number of disciplines, from social sciences and biology, to communication engineering, among many others. Their usefulness resides in that most groups of interconnected items or data structures describing any sort of relationship are susceptible—often conveniently—of being represented as a graph.

Real network data usually contain rich information about the systems that produce them, but many times any analysis requires vast resources due to the volume of the datasets, or the high frequency at which they are generated. Many algorithmic tools for network analysis depend heavily on aspects such as size and linearity of variables of interest. For instance, many real network data (e.g. those digitally-generated) are so vast that outrun traditional analysis algorithms. Additionally, sparse connectivity, i.e. if the network present only a small fraction of all the possible relationships among its nodes, makes analyses through simple approximations more difficult.

Even though many dimensionality reduction methods, such as Principal Component Analysis or Factor Analysis, have been studied for many decades, in recent years there have been many efforts in the literature to develop better ways of capturing non-trivial aspects of network structure in efficient ways. This is particularly the case for

[a] e-mail: lgmoyano@mendoza-conicet.gob.ar

the communities of network science, statistical physics, and computer science, that usually aim at improving and extending these methods, tailoring them to specific challenges, e.g. speech and image recognition [11,12], social network classification tasks [22], among several others.

Many systems can be described as driven by variables not directly observable or quantifiable, commonly called *hidden* or *latent* variables. There has been much interest in the assumption that, for some systems, the number of latent variables may be much smaller, sometimes orders of magnitude less, than the number of degrees of freedom of the system. For such cases, it could be very useful to gain insight from methods that infer on hidden variables. In this direction, the sustained attention by the research community on probabilistic, stochastic and statistical methods have resulted in a broad class of algorithms capable of improving the understanding of complex data such as social or technological networks. Usually these models imply finding a specific transformation or mapping of the data to some low-dimensional continuous vector space, more suitable for further modeling and analysis efforts.

In this review I present some general ideas regarding the representation of high-dimensional data onto low-dimensional vector spaces, with emphasis in models and techniques for network data, as well as some very recent progress obtained by theory and simulations. Additionally, I put forward some of the latest applications of these ideas, some of which involve empirical data from real networks. This review does not intend to be a thorough summary of the various representation learning models and algorithms in the literature, but aims to visit some key concepts around the idea of representation learning, motivate the exploration of the different models and describe some recent methods specifically around network representation learning. Finally, another important aspect in this report is to provide an update of concepts and results for the benefit of all the research communities involved: the computer science and machine learning areas, as well as the statistical physics and network science communities.

The present review is organised as follows: section 1 gives an overview of representation learning and revisits some classical methods for dimensionality reduction, as well as some general comments on classical representation learning methods in contrast to new *deep learning* algorithms. In section 3 we will review efforts around learning text representations, a subject that has been at the center of the representation learning field and that has served as incentive for network representations models. In section 4 we cover different models and algorithms for learning network representations, from geometrical methods to probabilistic and methods based on multilayer neural networks. We briefly illustrate some applications in section 5 and present some conclusions and prospects in section 6.

2 Representation learning

As we will see throughout this review, the idea of representing high-dimensional data through expressive vectors living in low-dimensional spaces has been exploited in the past in several areas of science. Here we are interested in techniques and methodologies to find effective *representations* of data, i.e. to find a set of transformations to be applied to the data in order to arrive at a more convenient structure, usually with lower dimensionality, which will be more suitable for further analysis or processing, e.g., node classification, link prediction, among others.

Traditionally, researchers have used insight from prior expert knowledge of the system to define which attributes in the data were the most convenient to take into account for any given task, e.g. the degree of a node would be a basic and general variable to quantify the network in order to describe its structure, and would be used to model things such as network dynamics, communities, and other aspects of the data. Such data attributes, also called *features*, are contrasted and validated through the performance of the learning tasks, such as classification or clustering. This *feature engineering* may be quite difficult depending on the analysis at hand, especially when the raw data are extremely high-dimensional and non-linear. Thus, it is advantageous to be able to find a convenient low-dimensional manifold that reflects the main explanatory factors for the observed variations in the data. Moreover, the possibility to find this embedding automatically, i.e. to *learn* the data representation, is at the center of the efforts driving the research forward in this area.

An essential assumption of representation learning is the *manifold hypothesis* [12], an implicit assumption that high-dimensional real-world data has the tendency to group in a a manifold $\mathcal{M}$ of lower dimension $d_{\mathcal{M}}$, i.e. a neighborhood embedded in input space $\mathbb{R}^d_x$. This manifold would provide a natural coordinate system to the representation being learned.

Representation learning is a rapidly-growing area, much encouraged by a steady stream of success in areas such as speech recognition, natural language processing, signal processing, image recognition, among others. As we will see in section 3.1, very efficient and semantically meaningful latent representations have been developed in the area of Natural Language Processing for text, based on distributed *word vectors*. These word vectors usually have real-valued dimensions, in opposition to *one-hot* representations canonically used in feature engineering, i.e. vectors with only one non-zero dimension. Such distributed word vectors typically allow for a better reflection of semantics and language relationships in machine learning classification tasks.

Typically, a good representation needs to have some key ingredients to correctly rescribe the intricacies of real-word data [93,12]. The learning model needs to be non-linear in order to adapt to the inherent non-linearity in the data. Additionally, a good representation needs also to preserve data structure, in the sense that similar data points should stay relatively close to each other in representation space, a non-trivial task as many times it is not entirely clear how to quantify these similarities. Finally, and as has been mentioned above but especially relevant for networks, a good representation needs to be able to deal with data sparsity, which usually strains classical algorithms due to a combination of high-dimensionality and insensitivity of features to small variations in input data.

2.1 Dimensionality reduction: Linear and non-linear methods

Most areas of science dealing with empirical data analysis, especially with current trends such as *big data*, are faced with the problem of high-dimensionality in data and how to reduce it to a lower-dimensional equivalent structure in order to carry out meaningful analyses.

There are classical subspace learning techniques used for dimensionality reduction to explain data variability and similarity such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Projections (LPP), Multidimensional Scaling (MDS) and Factor Analysis (FA), and their extensions. Most of these techniques can be generalised under the framework of matrix factorization. PCA is the most common unsupervised method, which uses linear transformations to find an embedding of the original data in low-dimensional space such that it maximmises the original variance of the data. LDA [29] is a supervised learning method, where data is projected into a low-dimensional space maximizing the ratio of between-class and within-class distances. MDS also produces an embedding through a linear transformation, which tends to preserve the distance between the data coordinates, and it is equivalent to PCA with Euclidean distance. LPP [34] is a linearization alternative of the non-linear Laplacian Eigenmaps method. Finally, FA also attempts to explain data variability through a smaller number of variables (factors) and models them as random variables. These techniques are linear, easy to implement and compute, and work well if the data is lying on an approximately linear subspace within the original high-dimensional space. However, when the dataset is associated to a non-linear structure, these algorithms do not provide meaningful embeddings capable of preserving any of the original associations [87]. Moreover, when the datasets are large is the case of most real networks of interest, these algorithms quickly become expensive and impractical because of the associated time complexity due to the common eigendecomposition step they all need to perform.

Several non-linear models have been proposed to overcome the limitations intrinsic in linear models mentioned above, as is the case of Isometric Feature Mapping (ISOMAP) [87], Local Linear Embedding (LLE) [74], Local Spline Embedding [96], as well as manifold learning techniques [10]. For instance, Belkin and Nigoyi [9] propose a model called Laplacian Eigenmaps (LEs), a geometrically-inspired method for representing a low-dimensional data structure contained in a high-dimensional space. Their method is capable of preserving locality, which makes it particularly stable in front of outliers and noise. The model by Belkin and Nigoyi was in turn the basis of a method proposed by Lobato et al. [4], where they use the Laplacian of a graph to embed complex networks in hyperbolic spaces, with remarkable efficiency.

2.2 Deep Learning

However, in the last decade, a set of learning models stemming mainly from the machine learning community, now generically known as *Deep Learning* models, has had a substantial impact in the representation learning literature for its significant improvements in accuracy and efficiency over previous efforts [50,12,11]. Even though deep learning models have had many applications, they can be regarded intrinsically as representation-learning methods. A deep learning architecture is typically composed by some type of an Artificial Neural Network (ANN), i.e. a stack of layers each composed by several (possibly non-linear) modules formally called neurons. Most ANNs are composed by an input layer, and output layer and one or more intermediate *hidden* layers. Each of these modules represents a non-linear mapping from input to output data, and aim at increasing the selectivity and invariance of the features learned by each layer. These layers provide multiple stages of data transformation from raw data (be it image, audio, text, etc.) to increasingly abstract levels of representations, in fact building a hierarchy of features. Such transformations can efficiently filter out irrelevant features of the input data and keep those aspects that contribute for the discriminative power of the model. Importantly, features in the intermediate layers are not design nor engineered by hand, but learned from the input data, effectively *trained* by examples.

In 2006, Hinton et al. [36] introduced a greedy unsupervised learning strategy that could compute multiple layers of feature detectors, by pre-training one layer at a time using unsupervised learning for Restricted Boltzmann Machines or RBMs (more on RBMs below). Their proposal was tested on the MNIST database of handwritten digits, which contains 60,000 training images and 10,000 test images and achieved remarkable accuracy, reviving the attention of the machine learning community in deep feedforward networks [12,50]. Some strong aspects of these deep architectures are the possibility of feature re-use and concept abstraction. The depth of a neural network refers to the amount of hidden layers in the model. More intermediate layers increase its power to re-use features, as there are multiple path combinations, which in fact grow exponentially with their number. Additionally, intermediate feature representations become more abstract from layer to layer, and this in turn allows features to be shared across learning tasks [11].

As mentioned above, deep architectures are typically trained with data, and there is a general agreement that this training can be computed by simple Stochastic Gradient Descent (where local minima normally do not impose difficulties when networks are large [50]) If the layers are composed by sufficiently smooth functions of their inputs and internal weights, the computation of gradients may be done by the Backpropagation procedure [12] (which can be thought of as a practical application of the chain rule of derivatives). Training is an integral and challenging aspect of deep learning research, as it is a process which is often quite consuming (both data and resources) if there is to achieve accurate results. Indeed, new architectures and both training and pre-training strategies are constantly proposed, examining the role of labeled data availability, depth and processing power [12].

There are several variants of deep architectures, some of the most common being Feedforward Neural Networks (FNNs), Recurrent Neural Networks (RNNs), Restricted Boltzmann Machines (RBMs), Convolucional Neural Networks (CNNs) and Deep Autoencoders (there are, of course, several others, see [12] and references there in). FNNs are a variant of an artificial neural network where there are no cycles formed by its connections. The multilayer version of FNNs is one of the most common forms of deep learning architectures. RNNs are neural network models that share parameters at each layer and process input sequences one at a time. Unlike FNNs, they do form (directed) cycles, and have an dynamical internal state that can maintain information about the history of the sequence. They are hard to train but have become powerful as generative models or in specific tasks such as machine translation [50]. RBMs are a stochastic type of ANNs, where the restriction is that modules within each layer may not have connections, which allows for training strategies that make them particularly useful and makes RBMs inference readily tractable. RBMs have had much impact in the representation learning community [36]. CNNs are a form of feed-forward neural networks, specially suited to process data in array form, as is the case of the visual cortex. CNNs typically have local connectivity between neurons and layers, each neuron connected to a small region of the input. They also use shared weights which allows for translational symmetry as well as pooling (for downsampling) layers. CNNs are generally tractable for simple backpropagating gradient computation. Finally, autoencoders are unsupervised neural networks used for learning efficient representations of data, trained to reconstruct the same data being used as input, with the goal of reducing dimensionality. *Deep autoencoders* (also known as *Deep Belief Networks*, DBNs [12]) are autoencoders with a large number of hidden layers where each pair of neighboring layers are pre-trained to approximate the solution before performing the backpropagation procedure. Typically, DBNs are stochastic generative models composed by stacked layers of RBMs.

Deep learning is making major progress in several areas of machine learning and artificial intelligence, with stunning results, for instance, for automatically generating captions from images [92]. There have also been many other remarkable results in areas such as Natural Language Processing (e.g., topic classification, sentiment analysis, question and answering), speech and image recognition, among several others [50], making deep learning a promising framework for cross-fertilization of ideas with other fields.

3 Learning Text Representations

Word embeddings are language techniques designed to find effective mappings from words to low-dimensional vectors in continuous space, in such a way that related or similar words are relatively close together. These text representations seek to improve Natural Language Processing tasks such as sentiment analysis, named entity recognition, part-of-speech tagging, among others. Applications of feature representations to text are at the root of many research efforts and have been the source of many of the ideas and developments used to approach to network representations. In this section we explore two great families of models for learning text representations [70]: global matrix factorization models and shallow window-based methods. We will highlight specific concepts that will be instrumental for understanding network representation learning models.

Several kind of global matrix factorization models have been proposed for the task of estimating (continuous) representations of words, e.g. Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), Latent Semantic Indexing (LSI) among others. LSA deals with term-document matrices, and represent a well-known analysis method developed for language models for finding a low-dimensional representation of words and documents in terms of latent class variables. These methods intend to map a query to its relevant documents at the semantic level where keyword-based matching often fails. LSI [28] eigendecomposes, by using singular-value decomposition, a bag-of-words feature space (i.e. an approximation where word ordering is irrelevant), where positions in this space serve as a kind of semantic indexing. LDA [13] is another statistical method devised for topic modeling that associates related words into sets of topics, which act as latent variables, representing documents in the form of distributions over these topics. Over large data sets, LDA is known to become computationally quite expensive.

On the other hand, local context window-based methods have been proposed to overcome general scalability issues present in matrix factorization methods. The question of whether distributed word representations are better learned by count-based methods or by prediction-based methods remains open. For instance, Baroni et al. (2014) argue in favor of prediction models [7], while others, e.g. Pennington [70], point out that each method has its strengths and weaknesses.

3.1 Shallow word embeddings: word2vec

A commonly used model for efficient text representation is the WORD2VEC model. In [62], Mikolov et al. present a predictive algorithm which extends their previous work [61], where they introduce two shallow neural network architectures, the Continuous Bag-of-Words (CBOW) model and the Skip-gram model (SG). The Skip-gram model is a neural probabilistic language model for constructing word vectors from large datasets (billions of words, with vocabularies of millions of words). It is regarded as very efficient as it does not need to perform dense matrix operations. The architecture has an input layer, a projection layer and an output layer in order to predict context words. If a corpus is composed by a sequence of words $w_1, w_2, \ldots, w_N$, then word vectors are trained to maximize the log probability of neighboring words in the corpus

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{j\in nb(j)} \log p(w_j|w_i) \tag{1}$$

where $nb(j)$ are the words in the neighborhood of word w_i and $p(w_j|w_i)$ is the conditional probability, usually computed by means of the Hierarchical Soft-max model [61].

The architecture of the Skip-gram model is designed to predict, given a target word, a number of context words associated to it. It works in opposition to the

CBOW model, in which given a number of context words, the model predicts the target word (see Fig. 1).

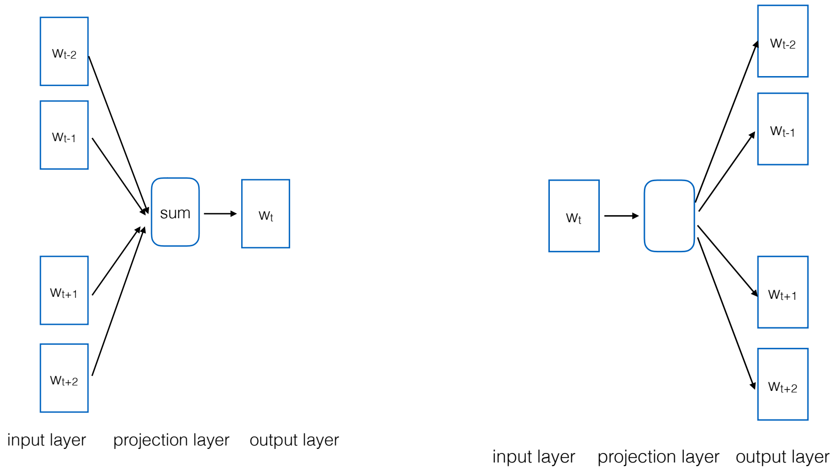


Fig. 1: Different learning methods for the word2vec model. Left: Continuous Bag-Of-Words training. The input layer corresponds to source words within a context fixed-length window, for computing the probability of the target word. Right: Skip-gram training model. The input layer contains the source word, to compute a prediction for context words in a fixed-length window.

In [62], Mikolov et al. improve their algorithm into the word2vec model which provides better quality word vectors and higher training speed, by subsampling the most frequent words and by presenting a simple training method over improbable examples (Negative Sampling). This allows for more accurate representations, especially for the case of more frequent words. Resulting word vectors have the desired property of placing semantically similar words close to each other in representation space, e.g. “strong” may lie close to “powerful”. The same effect is captured even for whole phrase representations [12]. An intriguing property of the word2vec algorithm is the possibility of linearly combining some of the word vectors produced to obtain semantically meaningful results. Thus, it is possible to find word analogies by means of vector arithmetic, e.g. “biggest” - “big” $+$ “small” $=$ “smallest” [61], which captures the idea of multi-clustered distributed representations [70,11]. This property also translates into equivalences of vector structures larger than simple words or even between languages. In [49], Le and Mikolov propose doc2vec, a model that extends word2vec to tackle larger blocks of text, i.e. sentences, paragraphs and even entire documents. This framework has been used for instance for sentiment analysis in social media for business [75].

The word2vec model has a prominent place in the representation learning literature [101,66,53,7], including some recent network representation models [71,32], as we will see in Section 4.2.

3.2 Log-bilinear models

Pennington and colleagues explore in [70] the origins of the arithmetic properties described in Section 3. They propose another shallow neural network model called

GloVe, a log-bilinear regression model that combines global matrix factorization and local context window methods, leveraging the most advantageous properties of count-based and prediction-based models. By training only nonzero elements in a word-word co-occurrence symmetrical matrix, it avoids processing the complete sparse matrix, or context word windows in extended corpora. The authors report good results for word analogy tasks, as well as similarity and named entity recognition tasks.

Log-bilinear models have also been explored by Kiros et al. [41], where they propose a multiplicative neural network model for learning distributed representations of words and text-based attributes, e.g. the language of the text, or meta-data associated with it, author information, etc. In this way, they are able to produce representations with conditional word similarity, e.g. the word "joy" may appear near the word "god" if the author is associated with the attribute "religion", but could appear near the word "comfort" is the author attribute is "science". The authors report efficiency improvements for tasks such as sentiment analysis and cross-lingual document classification.

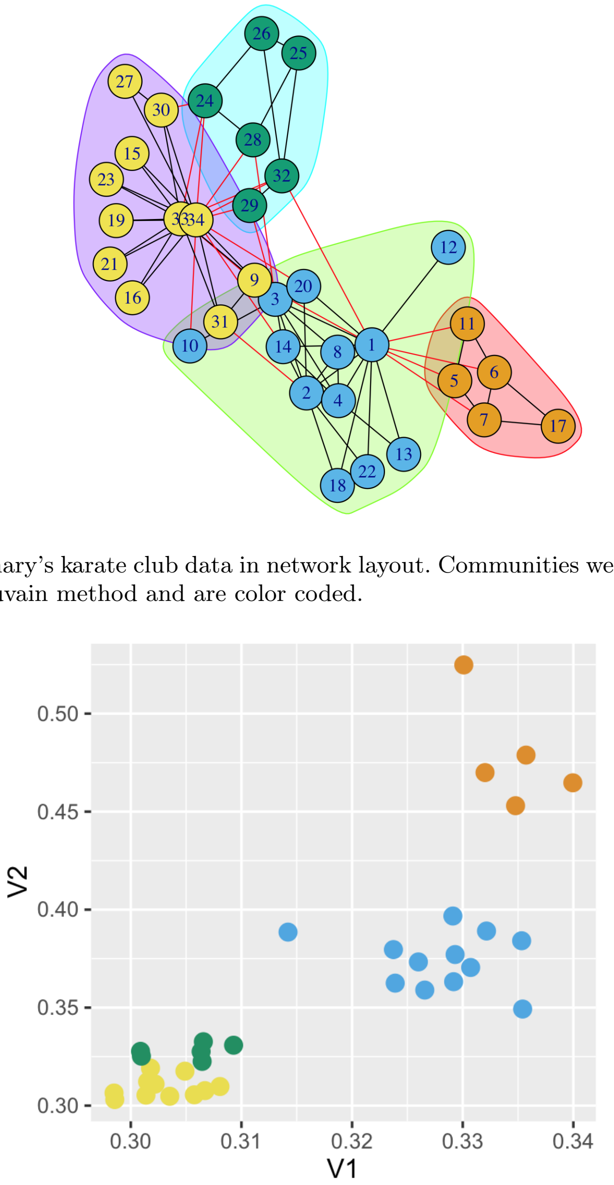
4 Learning Network Representations

Latent text representation research has gained an unprecedented level of attention, spurred by useful applications in the digital information domain and the high levels of accuracy and efficiency of recent models such as word2vec. These ideas have permeated other areas of research and has had an influence in network science through models inspired in these results. Much of the same challenges for accuracy and computational efficiency found in the case of text representations are also present in problems related to networks, also relating to high volume of data, non-linearity of the structures of interest as well as network sparsity. For instance, the intense technological advancements in the last years have given place to unprecedented ways of quantifying networks in detail, many times yielding enormous quantities of information, generating very sparse and non-trivial network structures.

To address these issues, Network Representation Learning (NRL) aims at the possibility of encoding node information in a unified continuous space. As in the case of text representations, the intention is to capture some original network property (topological, functional, etc.) into a space with lower dimensionality. As an illustration, consider Fig. 2. In Fig. 2a (above) we show the well-known Zachary's Karate club dataset where we have computed communities [1] presented in color code. In Fig. 2b we see the nodes embedded in a latent 2-dimensional space, where nodes belonging to the same community are close to each other, and thus communities appear clusterised. Different areas of study from network science, communications engineering and computer science, to name a few, have approached this problem. Next we will examine some of these concepts and some representative models found in the literature: models from a graph-theoretical and probabilistic point of view, models based on the geometrical nature of the embedding space and, finally, we comment on models based on deep neural networks.

Given a network $G = (V, E)$, a graph embedding (or network representation) consists in finding an appropriate mapping (or embedding), $\mathcal{M} : V \to X$, where X is a set of points $x_1, x_2, \ldots, x_{|V|}$ with $x_i \in \mathbb{R}^d$, every node in G is mapped into coordinates in a d-dimensional space, where possibly $d \ll |V|$. This ubiquitous form of representing information has been studied in many disciplines. For instance, in connection with statistical learning, it draws much input from Relational Learning, which aims at capturing the correlation between connected objects, especially in the presence of uncertainty [31,60,76]. Additionally, many ideas and concepts involving

[1] Communities were computed with the Louvain method [14]



(a) The Zachary's karate club data in network layout. Communities were computed with the Louvain method and are color coded.

(b) Embedded vectors for the Zachary's karate club data in a 2-dimensional latent space. Colors represent communities. Euclidean distance represents node similarity from a community point of view and same community nodes appear as clusters.

Fig. 2: The Zachary's karate club network, with communities in color code (above). Embedded vectors in representation space show that similar nodes (i.e. belonging to the same community) are close to each other in representation space.

network latent spaces come from the graph theory literature. For instance, in [25], Cohen discusses the case of three-dimensional graph drawing, which refers to the possibility of embedding an arbitrary graph in a three-dimensional space without any edge crossing. From a topological perspective, Aste and coauthors [6] point out that any network can be embedded in a surface with sufficiently high genus, making their results quite general.

Another aspect of NRL research refers to the nature of the hidden space where data is thought to cluster in. The authors in [81] present a probabilistic generative latent class model for graphs, where the probability of edges between two nodes depend on a set of latent classes. Handcock and colleagues propose in [33] a cluster model for social networks, with the aim to learn a latent Euclidean social space for prediction of social ties. In this model, the probability of a tie depends on the distance in this latent social space. Tang et al. [85] also extract latent social dimensions for use during training and improve performance of classification tasks. In what follows we will review recent results on three important and very active lines of research, embeddings based on geometric properties of the subjacent space, embeddings produced by stochastic and probabilistic methods such as random walks, and finally we will summarise methods based on different variants of neural networks, including deep learning models.

4.1 Geometric Embeddings

The idea that similar nodes (in a topological sense or from a functional perspective) may be due to closeness in a hidden metric space has also produced models grounded in more geometrical justifications. Aside from networks actually embedded in Euclidean (not hidden) space [8,27], such as transportation networks, early applications of hidden geometric embeddings appear, for instance, in the area of information networking. With the specific interest set in technological networks such as peer-to-peer networks and the internet, Shavitt and coauthors propose a method to efficiently embed graphs first in Euclidean spaces [78] and shortly afterwards in *hyperbolic* spaces [79,80], aiming at an accurate model for the internet (i.e., autonomous systems topologies).

4.1.1 Hyperbolic embeddings in complex networks

In [77], Serrano and colleagues explore the idea that topological properties of networks may be defined or influenced by geometrical properties of (hidden) subjacent metric space. This concept was introduced to explain self-similarity properties of some small-world networks. Nodes are embedded in the hidden metric space, each pair at a distance d, and there is an integrable function p governing the probability of being connected, which relates the network topology to the underlying metric space. This probability depends on the metric distance d as $p = d/d_c$, where d_c is the characteristic distance scale. The authors in [77] relate the presence of clustering to the existence of distances in this hidden metric space satisfying the triangle inequality.

The concept of an underlying metric space for networks had an immediate interest to help understand routing processes within networks [42,94,63]. These ideas were used for examine, for instance, an efficient greedy routing model for small-worlds [15], which was then expanded by the authors to also show that, for complex networks in general, there is no need to access to the complete topology in order to efficiently navigate this class of topologies [16].

Aste et al. [5] examine the idea of embedding complex networks in hyperbolic spaces and show that network properties are tightly linked to properties of the embedding hyperbolic space. In [43], Kleinberg shows that every connected finite graph has a greedy embedding in a hyperbolic space, i.e. a continuous space with constant negative curvature. Indeed, the hierarchical structure of complex networks may be approximately represented by treelike structures (small groups belonging to larger groups and so on), and hyperbolic space can be regarded as a continuous version of trees. A detailed analysis of why hyperbolic space may be consistent with complex networks was developed in [46], where Krioukov et al. argue that the scale-free property associated to some heterogeneous complex networks are associated with such hyperbolic spaces (see [79] for an earlier model involving a hyperbolic embedding specific for the internet). Krioukov and coauthors put forward a geometric model that considers nodes in a hyperbolic space and a connection probability function parametrised by a temperature T, in analogy with a grand canonical Fermi-Dirac distribution. In this model, the curvature of the metric space controls the power-law exponent in the degree distribution (i.e., the heterogeneity of the network), and the clustering is a function of the temperature T. Their framework is tested with empirical traceroute-based internet topology data, suggesting that is consistent with measurements.

The combination of greedy forwarding strategies for networks modeled as nodes embedded in hyperbolic spaces paved the way for a more thorough study on the efficient of routing [44,69] as well as in techniques to map empirical data to these types of spaces [17]. Hyperbolic embeddings have helped understand the role of core congestion in networks [65], the trade-off between similarity and popularity in network growth [68] as well as more theoretical approaches on the degree of hyperbolicity in networks [24].

Other authors have also exploited analogies with statistical mechanics. For instance Aste et al. [6] explore maximally embedded networks in surfaces and define simple energy functions from which they develop a statistical mechanics framework. Hyperbolic embeddings have been analysed as well in the context of minimizing distortion [91], and Zhao and colleagues show in [104] an application of these ideas for improving scalability in massive social networks. Lastly, hyperbolic embeddings of networks have demonstrated useful in applications, for instance, in Protein Interactions Networks (PIN), as tools for high throughput detection of novel protein interactions [1], generating best candidates for laboratory detection.

4.2 Stochastic and Probabilistic Embeddings

A number of authors, especially from the computer science community, have recently proposed representation learning models and techniques based on stochastic methods such as random walks over sequences of nodes over a network, as well as models based in optimization of appropriately defined objective functions. In this section we discuss three of the most prominent recent proposals.

4.2.1 DeepWalk

In [71], the authors introduce DeepWalk, an algorithm inspired in language modeling, more precisely in the word2vec algorithm (Section 3) with the goal of learning a social representation of the nodes in a network. The authors aim at producing a representation which is adaptable (new nodes and links should not need the representation be generated again), it should be community aware, in the sense of capturing neighborhood similarity and community membership (homophily translating in closeness), should be low dimensional for improved generalization and should be embedded in continuous space.

In analogy to word2vec, which processes short sequences of words to produce a representation or embedding of a text corpus in continuous vector space, the DeepWalk algorithm generates *sequences of nodes* from a stream of truncated random walks on the network, effectively mapping local information into features in a lower dimensional embedding. Extending the analogy of sequences of nodes as sentences composed by words, the idea in [71] is to estimate the likelihood of observing vertex v_i given the set of visited nodes in the random walk, i.e. $\Pr\big(v_i \mid (v_1, v_2, \cdots, v_{i-1})\big)$. This quantity is expected to encode local community structure and capture the diffusion process in the neighborhood of each vertex in the graph. The authors argue that, as the walk grows in length, a direct computation of this probability function is unfeasible, so they propose to follow the same strategy as the WORD2VEC algorithm, relaxing the constraints imposed by this probability function by ignoring the order of the vertices. After generating a number of random walks starting at each vertex, they perform an additional update step which makes use of Skip-gram and Hierarchical Soft-max (Section 3) as approximation procedures for the probability distribution. The resulting latent representation is then used for multi-label classification tasks on the nodes, and results are compared with other algorithms such as Spectral Clustering and Edge Clustering, outperforming them in most use cases. DEEPWALK drew much interest in the machine learning community as it carried useful ideas from the WORD2VEC algorithm to the realm of networks, spurring extensions and fruitful discussion.

Two authors of the DEEPWALK algorithm extended their idea in [72], proposing another random walk model that exploits a sampling mechanism, termed WALKLETS, over edges from powers of the adjacency matrix A. In this way, an edge sampled from A^k represents a path of length k in the original graph. The WALKLETS model has the explicit aim to capture the multiscale relationship between nodes thus generating efficient *multiscale representations* for multi-label classification tasks. Their algorithm intends to improve on limitations of DEEPWALK, where multiscale representations are not explicitly captured (as it has a strong bias towards representations that preserve mostly the lowest powers of the adjacency matrix [93,72]). Additionally, with DEEPWALK different scales of representations are not accessible independently.

Yanardag et al. [98] describe a unified framework to learn latent representations for graphs called Deep Graph Kernels. They use the WORD2VEC CBOW/Skip-gram framework making an analogy for words much like DEEPWALK, but instead of nodes they use *graphlets* (non-isomorphic sub-graphs of size k used for decomposing graphs).

Importantly, Yang, Liu and coauthors [99,100], as well as Levy and Golberg [51] and Li et al. [53], argue that the DEEPWALK's procedure for generating a representation is actually equivalent to factorizing a matrix M where each entry M_{ij} is the logarithm of the average probability that a random walk visiting vertex v_i afterwards visiting vertex v_j. Yang and colleagues propose a representation model [100] based in matrix factorization that incorporates text features of vertices, which is particularly efficient for noisy data or for cases where only a limited amount of training data is available. These sets of works strengthen the role of matrix factorization as a general framework from which to understand several latent representation algorithms. Even though DEEPWALK is a rather recent proposal, it has been widely used for benchmark comparison as well as a starting point in several posterior works [64,84,20,57,100,89,95,26].

4.2.2 NODE2VEC

Other works have explored the use of random walks over networks as a way to learn representations. In [32], Grover and Leskovec present NODE2VEC, an extension of the DEEPWALK algorithm. Their model is based on the design of a *biased random walk* mechanism controlled by two parameters p and q. These two parameters allow to tune the nature of the random walk, from exploring only neighborhood nodes to being able to visit node sequences ever farther from the root node. The two types of random walks for exploring the network add flexibility on the visited sequences, which will determine the final network representation. The authors compare their algorithm with LINE (see next section) and DeepWalk on several empirical network data, obtaining good results on a multi-label classification task. Grover et al. additionally explore the possibility of defining a binary operator acting on the space of pairs of feature vectors $f(u)$ and $f(v)$, generating a binary representation $g(u, v)$ which may be used for edge tasks such as link prediction.

4.2.3 LINE

Another recent network embedding model is LINE [84], by Tang and colleagues, which attempts to tackle the scalability of representation learning algorithms, proposing a method based on the optimization of an objective function designed explicitly for networks. The authors put forward two objective functions, the first one modeling neighboring nodes *first order proximity* [54] (local pairwise similarity), and *second order proximity*, which models the presence of neighbors of neighbors. To overcome the computationally expensive task of computing this second function over all pair of edges in the network, the authors make use of Negative Sampling [62] (Section 3). Additionally, they provide their method with an edge sampling mechanism devised to efficiently perform stochastic gradient descent in weighted networks. LINE performs well compared to other models (DeepWalk and graph factorization) in tasks such as multi-label classification or in large empirical social network data, as well as in word analogy in language networks. The LINE algorithm has had interesting applications in the area of Natural Language Processing, e.g. in entity typing models where knowledge graphs representations need to be generated [37]. LINE has also been used for link prediction tasks [90]. Finally, the authors extended LINE in [83] to deal with heterogeneous networks in which more than one type of nodes and edges are allowed to exist.

4.3 Neural Network Embeddings

As stated in Section 2.2, the field of deep learning is at the core of representation learning research and has had remarkable results, stimulating the exploration of applications of its ideas in many fields. As its natural, there is an array of research efforts to test deep learning frameworks for network feature learning.

Li and coauthors [52] were among the first to propose a “stacked” neural network architecture for latent feature learning in linked data. Inspired in existing models of graph factorization, they proposed LRBM, a binary and conditional Restricted Boltzmann Machine model for weighted networks. Their model propose latent variables (sender and receiver behaviours) aiming at capturing an effective representation for both node attributes and neighbor structure. They test their model for link prediction and node classification tasks with good results compared to baseline models, including matrix factorization .

In [93], Wang et al. propose a semi-supervised deep model named Structural Deep Network Embedding (SDNE). Much in the same way as [84], in order to preserve both local and global accurate descriptions of the network's structure, Wang and colleagues propose to optimize both first and second-order proximities [54] (i.e., similarity with neighbors and with neighbors of neighbors, respectively). The model feeds the adjacency matrix to a deep autoencoder (Section 2) and then optimizes the reconstruction error by minimizing a mixed loss function $\mathcal{L}_{mix}$ with respect to the set of parameters θ (i.e., weights and biases). The authors compare their semi-supervised deep model with other algorithms in respect to network reconstruction, multi-label classification, link prediction and visualization tasks, using several types of empirical network data, with remarkable results.

Cao, Lu and Xu [21] propose a stack of denoising autoencoders for extracting feature representations of graphs, encoding vertices into low dimensional vectors. They adopt the use of a random surfing model for producing a probabilistic co-occurrence matrix and compare their model for clustering and word-similarity tasks over several empirical networks, improving state-of-the-art results.

Several other works apply deep learning architectures to particular machine learning tasks for networks. In [97] the authors propose a conditional temporal Restricted Boltzmann Machine (ctRBM) generative model for dynamic link prediction. Authors in [55,56] also study the performance of deep belief networks for link prediction, in the specific case of signed social networks. Other models approach link prediction tasks with autoencoders for sparse graphs [103], which are also used for clustering tasks [88]. Deep architectures are also used for embeddings of heterogeneous networks [23].

5 Applications of Network Representation Learning

The possibility of learning network embeddings has multiple applications. Relational information is very general and there is a myriad of examples which can be framed in a network framework. On the other hand, the dimensionality problem and the semantic interpretation are widespread issues in applied research.

As it is natural, social networks and social media have been among the first examples to be examined under this light. For instance, Tang and Liu [85,86] extract latent social dimensions from network structure using modularity and spectral clustering techniques to improve affiliation classification tasks in social media. Jacob and coauthors [39,40] also propose a latent social space model based on loss function optimization for classification in heterogeneous (i.e., networks with different types of nodes) social networks. Nozza et al. [67] also argue that these techniques help in classifying heterogeneous networks. In their model, they optimize a classification loss function to construct a latent social space and infer the polarity of users and posts in social networks, particularly in the case of microblogs such as Twitter. Lai et al. also combine text and social information into a shared representation for improving social prediction tasks [48].

A more specific type of social relations are bibliographic co-authorship networks, i.e. nodes representing authors, connected whenever there have published together (i.e., they are actually the projection of bipartite networks formed by considering two types of nodes, authors and papers). Ganesh and colleagues [30] consider an unsupervised neural network model, based partly in paragraph2vec [49], to produce continuous author vector embeddings for this type of co-authorship networks.

We have seen before that much of these ideas have grown from and have applicability to text representations. There are several efforts in the context of distributed representations of text also involving networks. The authors of LINE propose in [83] another model for heterogeneous text network embeddings, particularly efficient for classifying long documents where labeled data are abundant. In a similar note, Ren and coauthors [73] propose a model for label noise reduction in entity typing (i.e. automatically recognizing text mentions to people, locations, organizations and other similar *entities*). They construct networks of mentions and the corresponding entity types and propose a general framework to integrate entity mentions, text features and entity types into a common low-dimensional latent space with the goal of minimizing noise in label assignment.

Applications of text representations exist for semantic knowledge networks and (web-based) entity networks, which interestingly is a topic that lies in-between text and network representation. Yang et al. [102] ask the question of learning social knowledge graphs. They propose a multi-modal Bayesian embedding framework to simultaneously capture network information (through DeepWalk) and text-based concepts (through Skip-gram [62]) in order to compute a distributed topic latent space. Luo et al. [59] exploit knowledge graphs connectivity patterns to capture context to produce more efficient embeddings. In the context of the Web, Heck et al. present in [38] a deep learning architecture for computing semantic models for web search using click-through data, i.e. a list of queries and their clicked documents. A similar deep architecture is presented by Heck and Huang [35] for embedding concepts (pages) from Wikipedia for later use in semantic parsing of Twitter dialogs. Dallman also aims at extracting semantic knowledge from Wikipedia [26] using tools such as DeepWalk.

Several other interesting applications of network embeddings have been proposed. For instance, related to information spreading and diffusion in networks [47,18], link prediction [55,56], traffic sign image recognition [58], video diffusion patterns in online social networks [57] and visualization of large-scale and high-dimensional data [82]. Network embeddings have also been applied in genetics and network medicine, e.g. to enhance topological prediction of protein interaction networks [19], analyzing nonlinear patterns in population genetics datasets [2], and in genetic interaction networks [3], among other examples.

6 Conclusions

We have reviewed different aspects of network representation learning, as well as connections with text embeddings. We have also seen some recent advances in this field and applications. Representation learning is a powerful and general set of techniques, which draws from different areas such as graph theory and heavily from the artificial intelligence and machine learning communities. Nevertheless, much of the ideas from these areas have seen fertile ground in network theory and important advances in the area of complex networks have been developed, as we have seen in the hyperbolic latent space formulation from Boguña, Krioukov and colleagues [16,45]. Moreover, we have seen that theoretical and technological breakthroughs have strongly influenced the methodologies and models proposed for constructing and understanding network latent spaces, as is the case for deep learning architectures. Indeed, the number of interesting applications in social media, biotechnology, semantic networks, image recognition, among many others, generated in connection with this body of research has had a sustained effect in the development of network research. It remains to be seen how the many research threads from the different areas of study will influence the advancement of network representation learning, but it is clear that there are as many challenges as opportunities.

I warmly thank Prof. Alberto Robledo and colleagues for their kind invitation for writing the present contribution. This work was supported by CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas.

References

1. Gregorio Alanis-Lobato. Mining protein interactomes to improve their reliability and support the advancement of network medicine. *Frontiers in genetics*, 6, 2015.

2. Gregorio Alanis-Lobato, Carlo Vittorio Cannistraci, Anders Eriksson, Andrea Manica, and Timothy Ravasi. Highlighting nonlinear patterns in population genetics datasets. *Scientific reports*, 5:8140, 2015.
3. Gregorio Alanis-Lobato, Carlo Vittorio Cannistraci, and Timothy Ravasi. Exploitation of genetic interaction network topology for the prediction of epistatic behavior. *Genomics*, 102(4):202–208, 2013.
4. Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. Efficient embedding of complex networks to hyperbolic space via their laplacian. *Scientific Reports*, 6, 2016.
5. Tomaso Aste, Tiziana Di Matteo, and ST Hyde. Complex networks on hyperbolic surfaces. *Physica A: Statistical Mechanics and its Applications*, 346(1):20–26, 2005.
6. Tomaso Aste, Ruggero Gramatica, and Tiziana Di Matteo. Exploring complex networks via topological embedding on surfaces. *Physical Review E*, 86(3):036109, 2012.
7. Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of Association for Computational Linguistics (ACL)*, pages 238–247, 2014.
8. Marc Barthélemy. Spatial networks. *Physics Reports*, 499:1–101, 2011.
9. Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.
10. Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
11. Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.
12. Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
13. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
14. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: theory and experiment*, 2008(10):P10008, 2008.
15. Marián Boguñá and Dmitri Krioukov. Navigating ultrasmall worlds in ultrashort time. *Physical review letters*, 102(5):058701, 2009.
16. Marián Boguñá, Dmitri Krioukov, and Kimberly C Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2009.
17. Marián Boguñá, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature communications*, 1:62, 2010.
18. Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 393–402. ACM, 2014.
19. Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics*, 29(13):i199–i209, 2013.
20. Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.
21. Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. AAAI Press, 2016.
22. Paulo R. Cavalin, Luis G. Moyano, and Pedro P. Miranda. A multiple classifier system for classifying life events on social media. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1332–1335. IEEE, 2015.
23. Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of*

the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 119–128. ACM, 2015.

24. Wei Chen, Wenjie Fang, Guangda Hu, and Michael W Mahoney. On the hyperbolicity of small-world and treelike random graphs. *Internet Mathematics*, 9(4):434–491, 2013.
25. Robert F. Cohen, Peter Eades, Tao Lin, and Frank Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17(2):199–208, 1997.
26. Alexander Dallmann, Thomas Niebler, Florian Lemmerich, and Andreas Hotho. Extracting semantics from random walks on wikipedia: Comparing learning and counting methods. In *Tenth International AAAI Conference on Web and Social Media*, 2016.
27. Li Daqing, Kosmas Kosmidis, Armin Bunde, and Shlomo Havlin. Dimension of spatially embedded networks. *Nature Physics*, 7(6):481–484, 2011.
28. Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.
29. Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.
30. J. Ganesh, Soumyajit Ganguly, Manish Gupta, Vasudeva Varma, and Vikram Pudi. Author2vec: Learning author representations by combining content and link information. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 49–50. International World Wide Web Conferences Steering Committee, 2016.
31. Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press Cambridge, 2007.
32. Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
33. Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.
34. Xiaofei He and Partha Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153. MIT, 2004.
35. Larry Heck. Deep learning of knowledge graph embeddings for semantic parsing of twitter dialogs. In *The 2nd IEEE Global Conference on Signal and Information Processing*. IEEE, IEEE, 2014.
36. Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
37. Lifu Huang, Jonathan May, Xiaoman Pan, and Heng Ji. Building a fine-grained entity typing system overnight for a new x (x= language, domain, genre). *arXiv preprint arXiv:1603.03112*, 2016.
38. Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2333–2338, New York, NY, USA, 2013. ACM.
39. Yann Jacob, Ludovic Denoyer, and Patrick Gallinari. Classification dans les graphes hétérogènes basée sur une représentation latente des nœuds. In *CORIA 2013*, pages 85–100, 2013.
40. Yann Jacob, Ludovic Denoyer, and Patrick Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 373–382. ACM, 2014.
41. Ryan Kiros, Richard Zemel, and Ruslan R Salakhutdinov. A multiplicative model for learning distributed text-based attribute representations. In *Advances in Neural Information Processing Systems*, pages 2348–2356, 2014.
42. Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845–845, 2000.
43. Robert Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1902–1909. IEEE, 2007.

44. Dmitri Krioukov, Fragkiskos Papadopoulos, Marián Boguñá, and Amin Vahdat. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *ACM SIGMETRICS Performance Evaluation Review*, 37(2):15–17, 2009.
45. Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.
46. Dmitri Krioukov, Fragkiskos Papadopoulos, Amin Vahdat, and Marián Boguñá. Curvature and temperature of complex networks. *Physical Review E*, 80(3):035101, 2009.
47. Cédric Lagnier, Simon Bourigault, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. Learning information spread in content networks. arXiv preprint arXiv:1312.6169, 2014.
48. Yi-Yu Lai, Chang Li, Dan Goldwasser, and Jennifer Neville. Better together: Combining language and social interactions into a shared representation. In *Proceedings of TextGraphs-10: the Workshop on Graph-based Methods for Natural Language Processing*, pages 29–33, San Diego, CA, USA, 2016. Association for Computational Linguistics.
49. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, volume 14, pages 1188–1196. ICML, 2014.
50. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
51. Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.
52. Kang Li, Jing Gao, Suxin Guo, Nan Du, Xiaoyi Li, and Aidong Zhang. Lrbm: A restricted boltzmann machine based approach for representation learning on linked data. In *2014 IEEE International Conference on Data Mining*, pages 300–309. IEEE, 2014.
53. Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pages 25–31, 2015.
54. David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.
55. Feng Liu, Bingquan Liu, Chengjie Sun, Ming Liu, and Xiaolong Wang. Deep learning approaches for link prediction in social network services. In *International Conference on Neural Information Processing*, pages 425–432. Springer, 2013.
56. Feng Liu, Bingquan Liu, Chengjie Sun, Ming Liu, and Xiaolong Wang. Deep belief network-based approaches for link prediction in signed social networks. *Entropy*, 17(4):2140–2169, 2015.
57. Yi and Long. Characterizing video diffusion patterns in online social networks. *HKU Theses Online (HKUTO)*, 2015.
58. Ke Lu, Zhengming Ding, and Sam Ge. Sparse-representation-based graph embedding for traffic sign recognition. *IEEE transactions on intelligent transportation systems*, 13(4):1515–1524, 2012.
59. Yuanfei Luo, Quan Wang, Bin Wang, and Li Guo. Context-dependent knowledge graph embedding. In EMNLP, editor, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1656–1661. ACL, 2015.
60. Sofus A Macskassy and Foster Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8(May):935–983, 2007.
61. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Workshop paper at International Conference on Learning Representations (ICLR 2013)*, 2013.
62. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

63. Luis G. Moyano, Juan P. Cárdenas, Jorge Salcedo, Mary Luz Mouronte, and Rosa M. Benito. Information transfer dynamics in fixed-pathways networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(1):013126, 2011.
64. Sharad Nandanwar and MN Murty. Structural neighborhood based classification of nodes in a network. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
65. Onuttom Narayan and Iraj Saniee. Large-scale curvature of networks. *Physical Review E*, 84(6):066108, 2011.
66. Liqiang Niu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. A unified framework for jointly learning distributed representations of word and attributes. In *Proceedings of The 7th Asian Conference on Machine Learning*, pages 143–156, 2015.
67. Debora Nozza, Daniele Maccagnola, Vincent Guigue, Enza Messina, and Patrick Gallinari. A latent representation model for sentiment analysis in heterogeneous social networks. In *International Conference on Software Engineering and Formal Methods*, pages 201–213. Springer, 2014.
68. Fragkiskos Papadopoulos, Maksim Kitsak, M Ángeles Serrano, Marián Boguná, and Dmitri Krioukov. Popularity versus similarity in growing networks. *Nature*, 489(7417):537–540, 2012.
69. Fragkiskos Papadopoulos, Dmitri Krioukov, Marian Boguna, and Amin Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.
70. Manning CD. Pennington J, Socher R. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language*, 2014.
71. Bryan Perozzi, Rami Al-Rfou, and Steven 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.
72. Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. Walklets: Multiscale graph embeddings for interpretable network classification. *arXiv preprint arXiv:1605.02115*, 2016.
73. Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
74. Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
75. Parinya Sanguansat. Paragraph2vec-based sentiment analysis on social media for business in thailand. In *2016 8th International Conference on Knowledge and Smart Technology (KST)*, pages 175–178. IEEE, 2016.
76. Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.
77. M Angeles Serrano, Dmitri Krioukov, and Marián Boguná. Self-similarity of complex networks and hidden metric spaces. *Physical review letters*, 100(7):078701, 2008.
78. Yuval Shavitt and Tomer Tankel. Big-bang simulation for embedding network distances in euclidean space. *IEEE/ACM Transactions on Networking (TON)*, 12(6):993–1006, 2004.
79. Yuval Shavitt and Tomer Tankel. On the curvature of the internet and its usage for overlay construction and distance estimation. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1. IEEE, 2004.
80. Yuval Shavitt and Tomer Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Transactions on Networking (TON)*, 16(1):25–36, 2008.
81. Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

82. Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.
83. Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM, 2015.
84. Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.
85. Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.
86. Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.
87. Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
88. Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *AAAI*, pages 1293–1299, 2014.
89. Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In *Proceedings of the 25th International Conference on Artificial Intelligence*. AAAI Press, 2016.
90. Vipul Venkataraman and Pramod Srinivasan. Graph embedding aided relationship prediction in heterogeneous networks. CS 512 Project Report, 2016.
91. Kevin Verbeek and Subhash Suri. Metric embedding, hyperbolic space, and social networks. In *Proceedings of the thirtieth annual symposium on Computational geometry*, page 501. ACM, 2014.
92. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.
93. Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
94. Douglas R White and Michael Houseman. The navigability of strong ties: Small worlds, tie strength, and network topology. *Complexity*, 8(1):72–81, 2002.
95. Fei Wu, Xinyan Lu, Jun Song, Shuicheng Yan, Zhongfei Mark Zhang, Yong Rui, and Yueting Zhuang. Learning of multimodal representations with random walks on the click graph. *IEEE Transactions on Image Processing*, 25(2):630–642, 2016.
96. Shiming Xiang, Feiping Nie, Changshui Zhang, and Chunxia Zhang. Nonlinear dimensionality reduction with local spline embedding. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1285–1298, 2009.
97. Li Xiaoyi, Li Hui Du Nan, et al. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining. Philadelphia, PA, USA: SIAM*, 2013.
98. Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.
99. Cheng Yang and Zhiyuan Liu. Comprehend deepwalk as matrix factorization. *arXiv preprint arXiv:1501.00358*, 2015.
100. Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina*, pages 2111–2117, 2015.
101. Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML 2016*, 2016.

102. Zhilin Yang, Jie Tang, and William Cohen. Multi-modal bayesian embeddings for learning social knowledge graphs. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 2016.

103. Shuangfei Zhai and Zhongfei (Mark) Zhang. *Dropout Training of Matrix Factorization and Autoencoder for Link Prediction in Sparse Graphs*, chapter 51, pages 451–459. SIAM, 2015.

104. Xiaohan Zhao, Alessandra Sala, Haitao Zheng, and Ben Y Zhao. Efficient shortest paths on massive social graphs. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*, pages 77–86. IEEE, 2011.