Capsules for Hierarchical Embedding of Attributed Complex Networks to a Hyperbolic Feature Space

David McDonald dxm237@cs.bham.ac.uk Shan He s.he@cs.bham.ac.uk

***Abstract*—TODO**

I. Introduction

Throughout our world, we observe complex systems – groups of *elements* that connect to each other by *relations* in a non-uniform way. Through these relations, these elements are able to work together and function as a coherent whole that is greater than the sum of its parts. We see this in the simple relationships amongst people that form an entire society; in the iterations between genes, proteins and metabolites that form a living organism; and in the links between pages that make up the internet. Within these systems, interactions are not controlled globally, but emerge locally based on some local organisation that gives rise to new levels of organisation. In this way, we see that the organisation of complex systems is *hierarchical*: elements belong to many different systems on many different scales, with all the levels affecting each other [1]. In addition to the hierarchical organisation of elements, we observe that entities can be richly annotated with features, that are themselves organised hierarchically. For example, a paper within a citation network may be annotated with the presence of particular key words and the presence of these words may give rise to the presence or absence of higher order (or more abstract) features such as semantics or topic.

The success of machine learning algorithms often depends upon data representation [2]. Representation learning – where we learn alternative representations of data – has become common for processing information on non-Euclidean domains, such as the domain of nodes and edges that comprise these complex systems. Prediction over nodes and edges, for example, requires careful feature engineering [3] and representation learning leads to the extraction of features from a graph that a most useful for downstream tasks, without careful design or a-priori knowledge. In particular, research has shown compelling evidence that an underlying metric space underpins the emergence of behaviour in the network – for example, two elements that appear close together in this metric space are more likely to interact [3], [4], [5] and furthermore, that the shape of this metric space is, in fact, hyperbolic. Indeed, we can interpret a hyperbolic space is a continuous representation of a discrete tree structure that captures the hierarchical organisation of elements within a complex system [6].

A. Present Work: Hierarchical Decomposition of Attributed Networks in to Hyperbolic Feature Space

Here, we examine the usefulness of Graph Signal Processing (GSP), and emergent field in the literature that considers the expressions of elements within the system to be a "signal" that structured by the relationships that we observe [7], [8], [9] – for representing the elements of a system as a set of points in n-dimensional hyperbolic space. Often, our access to these systems is incomplete, or the system is evolving over time, and so we consider this task in the *inductive* setting, where we learn representations of entities based on sampling of local features, rather than knowledge of the entire network topology. Furthermore, we leverage the advantages of "capsules" – a recently proposed variant of the standard neural-network neuron that outputs a vector that abstractly represents a feature and its "pose", to uncover the parse tree of hierarchical features in the system [10], [11], to perform a layer-wise embedding of a complex system into hyperbolic space, where representations in higher levels of the network correspond to the presence of increasingly abstract features.

II. Related Work

A. Representation Learning on Graphs

Several models in the literature assume the existence of an underlying metric space that controls the topology (and possibly dynamics) of the network. They suppose that elements that are closer together in this space are more 'similar' and have a higher probability of being connected. These models aim to infer the geometry of these spaces and the positions of nodes within the space, such that the probability of reconstructing the observed network is maximised, for purposed of better understanding of the system and visualisation. This is network embedding, and is the cornerstone of the field of *network geometry* ([6]).

Network embedding is closely related to the field of manifold learning. Indeed, many classical non-linear manifold learning techniques, such as Isomap ([12]) and Laplacian Eigenmaps ([13]), must first construct nearest neighbour graphs based on dissimilarities between samples before dimensionality reduction takes place. Many of these techniques are directly applicable to embedding of (single-layer, unweighted) complex networks by simply omitting the graph construction step.

An interesting and popular embedding paradigm in the literature comes from natural language processing (NLP). In particular, the Skipgram model and the Word2Vec algorithm that aims to vectorise words and phrases in a Euclidean 'semantic' space such that similar words are mapped close together ([14], [15]). The principle idea is, given a corpus of words and a particular sentence, generate a 'context' for each input word with the aim of maximising the likelihood of observing context words in the embedding space, given the input word. Similarities are measured by dot products and accordingly, observation

probabilities are computed using a multilayer perception with a linear hidden layer and Softmax output. Through the use of sub-sampling and negative sampling (replacing Softmax with sigmoid), training can be made very efficient and the resulting embeddings can be obtained from the activation of the hidden units. This idea naturally extends to networks, where sentences are replaced by 'neighbourhood graphs' generated from random walks. Furthermore, the shallow architecture of the Skipgram model has been replaced with multiple non-linear layers to learn the highly non-linear relationships between nodes by adopting a deep learning framework ([16], [17]). By introducing additional parameters into the random walk to control a breadth vs. depth first neighbourhood search, Grover and Leskovec [3] were able to identify neighbourhoods of nodes with high *homophily* and high structural similarity with node2vec. The use of these parameters to control the random walk, therefore controlled the definition of community and offered great flexibility to the practitioner to customised the search based on exactly what they are looking for. Node2vec, however, was not designed with attributed networks in mind.

An emerging popular belief in the literature is that the underlying metric space of most complex networks is in fact hyperbolic. Nodes in real world networks often form a *taxonomy* – where nodes are grouped hierarchically into groups in an approximate tree structure ([18]). Hyperbolic spaces can be viewed as continuous representations of this tree structure and so models that embed networks into hyperbolic space have proven to be increasingly popular in the literature ([19], [6]). In fact, this assumption has already had proven success in the task of greedy forwarding of information packets where nodes use only the hyperbolic coordinates of their neighbours to ensure packets reach their intended destination ([20]).

The most popular of all these models is the Popularity-Similarity (or PS) model ([18]). This model extends the "popularity is attractive" aphorism of preferential attachment ([1]) to include node similarity as a further dimension of attachment. Nodes like to connect to popular nodes but also nodes that 'so the same thing'. The PS model sustains that the clustering and hierarchy observed in real world networks is the result of this principle ([4]), and this trade-off is abstractly represented by distance in hyperbolic space. Maximum likelihood (ML) was used in [18] to search the space of all PS models with similar structural properties as the observed network, to find the one that fit it best. This was extended by the authors in [21], [22]. Due to the computationally demanding task of maximum likelihood estimation, often heuristic methods are used. For example, [4] used Laplacian Eigenmaps to efficiently estimate the angular coordinates of nodes in the PS model. The authors then combined both approaches to leverage the performance of ML estimation against the efficiency of heuristic search with a user controlled parameter in [5]. Additionally, [23] propose the use of classical manifold learning techniques in the PS model setting with a framework that they call *coalescent embedding*.

Beyond the two-dimensional hyperbolic disk of the PS model, we see that embedding to an n-dimensional Poincaré ball can give more degrees of freedom to the embedding and capture further dimensions of attractiveness than just "popularity" and "similarity" [24].

B. Signal Processing on Irregular Domains

Convolutional neural networks (CNNs) has enjoyed immense popularity in recent years thanks, primarily, to their great successes across a wide variety of computer vision tasks. In fact, the discovery of an efficient training algorithm for them as well as the performance of AlexNet in on the challenging and high dimensional Imagenet dataset in 2012 [25] brought the entire field of deep learning into the mainstream. We can view images as data with an inherent regular structure – a lattice of pixels with connections between neighbouring pixels that enforce a similarity between their outputs. In this way, we can view learning on images as learning on two-dimensional signals and convolution operator as a local filter applied in the spacial domain that extracts features from the input signal [7]. A further example of data with a regular structure is time-series data as we can interpret this as a chain graph, with nodes representing time-points and connections between only neighbouring nodes. The convolution operation is well defined because of the order of pixels in an image – we can move from left to right, top to bottom. However, generalising this concept to data structure represented by irregular graphs is challenging as their is no intrinsic order to nodes in a general graph.

Graph *canonicalization* was used by Patchy-San to give order to a node's neighbours within a graph, and this allowed for an efficient extraction of local graph kernels [26]. Defferrard et *al*. [7] proposed performing the convolution in the spectral domain of the graph and approximate the filter by a K dimensional Chebyshev polynomial of the graph's Laplacian matrix. This both localises the convolution operation in the spacial domain and reduces the number of parameters to learn from $\mathcal{O}(N)$ (the number of nodes in the network) to $\mathcal{O}(K)$ (the filter size) which is the same as traditional CNNs. Filtering signals on graphs can be further accelerated by using the more involved Lanczos method that, in practice, outperforms the Chebyshev polynomial approximation in approximation error [27]. Additionally, Defferrard et *al*. use a graph coarsening approach to pool graph features as a form of the one-dimensional signal pooling from regular signal processing. Kipf and Welling [8] argue that multiple convolutional layers with a small filter size stacked on top of each other, will outperform a single layer with a larger filter size, citing that this approach has improved model capacity in other domains [28]. They show that their method GCN, a deep layer-wise linear model, can out-perform state of the art embedding algorithms in the semi-supervised setting of node classification.

Due to reliance on computing the graph Laplacian, all of these approaches are *transductive* in that the entire graph structure must be known at training time. Furthermore, they require a batch training approach, that limits scalability to datasets that fit entirely into memory. Hamilton et *al*. [9] attempt to overcome these drawbacks by proposing GraphSAGE, a general *inductive* framework that learns representations based on local features only and that can be applied to very large or evolving graphs. They investigate a number of aggregation methods and, interestingly, show that the layer-wise linear filter of GCN is simply a rescaled version of element-wise mean pooling over the representations of the neighbours of a node.

C. Capsules: Learning the Parse Tree of Features

The pooling operation in CNNs is used as dimensionality reduction and to introduce some *translational invariance* to the model [25]. However, even a 2×2 pool will discard 75% of the information in the previous layer of the model, as only the largest activation within a pooling neurons receptive field is passed on the next layer in the network. Hinton argues that the fact that this works well is unfortunate and, in fact, that the use of the pooling operation at all in deep learning is a "big mistake". Instead, he proposes "capsules" that, rather than outputting a scalar as traditional neurons do, output a vector that represent the "pose" parameters of a particular feature [10]. Pose parameters such as scale, localized skew and translation, for example. Capsules in different layers do not connect to each other in the traditional way – low level capsules "select" the high level capsules to send their output to based on how well they can predict the output of that capsule [11].

III. Capsules On Complex Networks

A. Problem Setting

We consider the problem of learning a representation of a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ of N samples of features of dimension D. Sample inter-relations are given by the graph $\mathbb{G} = (V, E)$, where V is the set of vertices such that $|V| = N$ and E is the set of edges representing the relations between vertices.

B. GraphCap Layer

[9]:

$$\mathbf{h}_v^k = \sigma(\mathbf{W} \cdot \mathrm{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}))$$

squash function [11]:

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

dynamic routing algorithm: [11] TODO

C. Unsupervised Loss Function

Batch-wise loss function

$$L(\Theta) = -\frac{1}{|D|} \sum_{u \in D} \log \left[\frac{\exp(-d(\mathbf{h}_u, \mathbf{h}_v))}{\sum_{v' \in \mathcal{S}(u)} \exp(-d(\mathbf{h}_u, \mathbf{h}_{v'}))} \right]$$

where $\mathcal{S}(u)$ is the set of positive and negative samples of node u, and D is the set of nodes in the batch.

Hyperbolic distance computed as

$$d(\mathbf{h}_u, \mathbf{h}_v) = \operatorname{arccosh}\left[1 + 2\frac{||\mathbf{h}_u - \mathbf{h}_v||^2}{(1 - ||\mathbf{h}_u||^2)(1 - ||\mathbf{h}_v||^2)}\right]$$

where $||\cdot||$ is the usual Euclidean norm [24].

IV. Results

V. Discussion

References

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[2] 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.

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[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] Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. Manifold learning and maximum likelihood estimation for hyperbolic network embedding. *Applied Network Science*, 1(1):10, 2016.

[6] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

[10] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[11] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

[12] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[13] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[14] 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.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] 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.

[17] 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. International World Wide Web Conferences Steering Committee, 2015.

[18] Fragkiskos Papadopoulos, Maksim Kitsak, M Serrano, Marián Boguná, and Dmitri Krioukov. Popularity versus similarity in growing networks. *arXiv preprint arXiv:1106.0286*, 2011.

[19] Dmitri Krioukov, Fragkiskos Papadopoulos, Amin Vahdat, and Marián Boguñá. Curvature and temperature of complex networks. *Physical Review E*, 80(3):035101, 2009.

[20] Fragkiskos Papadopoulos, Dmitri Krioukov, Marián Boguñá, 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.

[21] Fragkiskos Papadopoulos, Constantinos Psomas, and Dmitri Krioukov. Network mapping by replaying hyperbolic growth. *IEEE/ACM Transactions on Networking (TON)*, 23(1):198–211, 2015.

[22] Fragkiskos Papadopoulos, Rodrigo Aldecoa, and Dmitri Krioukov. Network geometry inference using common neighbors. *Physical Review E*, 92(2):022807, 2015.

[23] Josephine Maria Thomas, Alessandro Muscoloni, Sara Ciucci, Ginestra Bianconi, and Carlo Vittorio Cannistraci. Machine learning meets network science: dimensionality reduction for fast and efficient embedding of networks in the hyperbolic space. *arXiv preprint arXiv:1602.06522*, 2016.

[24] Maximilian Nickel and Douwe Kiela. Poincar\'e embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*, 2017.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[26] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.

[27] Ana Susnjara, Nathanaël Perraudin, Daniel Kressner, and Pierre Vandergheynst. Accelerated filtering on graphs using lanczos method. *arXiv preprint arXiv:1509.04537*, 2015.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.