Learning Graph Representations for Use in Cancer Metastasis Prediction

Wei Xin Chan
Department of Computer Science
School of Computing
National University of Singapore
COM1, 13 Computing Drive
Singapore 117417
weixin@u.nus.edu

Abstract

Graphs are useful data structures that are able to succintly encode and express relationships between different entities. However, a central challenge in using graphs for machine learning tasks lies in finding a meaningful representation of these graphs in Euclidean space to be used as input features. Earlier approaches for using graphs as inputs for machine learning tasks have relied heavily on hand-engineered features. Recently, several graph representation learning methods have been proposed that are able to learn graph representations in an unsupervised manner. I review several of these methods and assess their compatibility in integrating biological pathway and gene expression information for use in a classification task - to predict relapse in treated breast cancer patients. I use the graph autoencoder to construct node and graph embeddings and evaluate the performance of classifiers that use these graph representations against classifiers that use gene-level and pathway-level features as inputs.

1 Introduction

The human body is composed of trillions of cells that perform basic cellular functions in order to survive. These cellular functions (e.g. metabolism, cell signalling) are executed by natural biological pathways, which are composed of thousands of proteins working together in an interconnected, interdependent system. A change in quantity of one protein in the pathway would result in a complicated series of downstream effects.

Studies have shown that dysregulated biological pathways are better indicators of cancer than driver gene mutations [12]. This has led to increased focus on network-based analysis of gene expression data. Network-based methods that attempt to identify differentially expressed subnetworks in cancer patients have been shown to achieve higher precision and reproducibility [15], as compared to common statistical methods that seek to identify individual genes that are differentially expressed.

Earlier works on cancer phenotype prediction used individual gene expression values as features for a supervised classifier [19]. Subsequent network-based methods devised methods to identify relevant subnetworks within biological pathways, and scored each subnetwork according to its constituent gene expression values [3, 13]. Individual subnetwork scores were used as features for the classifer. The main drawback of these methods is that they do not leverage information regarding interactions between different genes in biological pathways for their classification task.

The structure of biological pathways can be intuitively encoded as graphs with genes or gene products as nodes and the interactions between them as edges. Graphs are able to capture information regarding the relationships between different genes or gene products in biological pathways effectively. In

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

this paper, I seek to explore whether biological pathway information that are encoded in the form of graphs can be incorporated with gene expression data effectively for use in a standard machine learning classification task. However, a long-standing issue with machine learning on graphs is that it is difficult to find meaningful representations of graphs to be used as input features for machine learning models.

Earlier attempts have focused on hand-engineering features for specific machine learning tasks. An example is the construction of statistics that measure the similarity between a pair of nodes in a single graph (e.g. shortest path, proportion of common neighbours) for use as features in missing link prediction [14]. Recently, there has been a paradigm shift from hand-engineering graph features using domain knowledge to learning graph representations using machine learning algorithms. In section 2, I review several graph representation learning methods and assess their compatibility for use in generating graph representations that incorporate gene expression data. I select the graph autoencoder (GAE) proposed by Kipf and Welling [11] as my tool of choice in constructing graph representations that encode both biological pathway and gene expression information. These graph representations are subsequently used as input features in a classification task to predict whether breast cancer patients will relapse and develop tumour metastasis over the next five years following breast cancer surgery.

2 Related work

2.1 Graph representation learning algorithms

The general goal of graph representation learning algorithms is to learn to encode graph information that is non-Euclidean in nature into a low-dimensional feature vector. Majority of graph representation learning algorithms learn node embeddings, where individual nodes in a graph are represented as feature vectors. These algorithms are optimised to produce node embeddings that are able to preserve the structural information of a graph. In some cases where individual nodes in a graph are assigned features or attributes, the node embeddings are also judged based on their ability to preserve node attributes. These node embeddings can be used to achieve good performance in common network analysis tasks, such as semi-supervised node classification and missing link prediction. It is important to note that these tasks are very different from the classification task set out in this paper. In this section, we describe several *unsupervised* graph representation learning algorithms that are commonly used to generate embeddings for use in node classification and missing link prediction tasks.

The first group of methods are based on matrix-factorisation, with a popular example being the Laplacian Eigenmaps algorithm [1]. In its simplest form, the algorithm constructs a Laplacian matrix that represents a simple graph by subtracting the adjacency matrix of the graph from its diagonal degree matrix, $L = D - A$. Subsequently, eigendecomposition is performed on the graph Laplacian matrix and the top m eigenvectors aside from the first eigenvector corresponding to the eigenvalue 0 are used to embed the nodes in a m-dimensional Euclidean space. The m-dimensional node embeddings produced not only serve as graph representations, but are also widely used in conjunction with k-means clustering methods in spectral clustering.

The second group of methods are based on the random walk approach, with two prominent examples being DeepWalk [17] and node2vec [6]. These methods rely on sampling a large number of fixed-length random walks starting from each node in the graph to capture information about the graph structure in terms of a stochastic measure of node similarity. For example, nodes i and j are considered to be similar if the probability of encountering node j is high when on a random walk starting from node i. Node embeddings are optimised such that the dot product between the two embeddings z_i and z_j are proportional to the stochastic similarity measure mentioned. However, a key difference betwen node2vec and DeepWalk is that node2vec allows for the use of two hyperparameters p and q to bias the random walk. To illustrate the use of these hyperparameters, we consider a random walk that has just travelled from node i to j. Hyperparameter p controls the probability of immediately revisiting node i, while q controls the probability of travelling to a node in the one-hop neighbourhood of node i from node j. These hyperparameters are used together to control whether the random walks resemble a breadth-first search or depth-first search.

The third group of methods rely on neighbourhood aggregation to generate node embeddings that are able to take into account attributes of nodes that are in their local neighbourhood. This approach can also be termed as graph convolutions due to their similarity to convolutional filters in convolutional

neural networks (CNNs). Unlike the two groups of graph representaion learning methods mentioned above, these methods are able to incorporate information that is present in node attributes. Two of these methods are the graph convolutional network (GCN) [10] and graphSAGE [7]. I discuss in further detail the graph auto-encoder (GAE) by Kipf and Welling [11], which incorporates their work on the GCN as well. The GAE is an auto-encoder which uses the GCN as its encoder and a pairwise decoder model that reconstructs the adjacency matrix by taking the inner product of pairs of embeddings z_i and z_j. The GCN encoder works by aggregating the node attributes of itself and all the one-hop neighbours of each node through a weighted sum (similar to a convolutional kernel) and applying an activation function to obtain the hidden layer activations. Increasing the number of graph convolutional layers iterates this process, and incorporates more information of attributes belonging to node neighbours that are further away into each node. This is akin to an increase in receptive field of neurons when an increasing number of convolutional layers are used before. The weights of the graph convolutions are shared across all nodes, and as a result node embeddings can also be generated from nodes that were not present during training. GraphSAGE works by the same principle, but differs in the way it aggregates the one-hop neighbours of each node, and how it combines that information with the attributes of the node itself.

The above methods learn node embeddings of graphs, and produce node embedding matrices of a graph which are presented as a $N \times D$ matrix where N is the number of nodes and D is the dimension of node embeddings. Each row in the matrix would correspond to the embedding of a single node in the graph. In comparison, graph embedding methods aim to generate graph or subgraph embeddings where the entire graph is represented by a single D-dimensional vector. These approaches mainly differ in the way they aggregate node embeddings of a graph in order to obtain its graph embedding. A simple approach of generating graph embeddings is to perform an element-wise sum of node embeddings of all nodes present in a graph or subgraph [5]. Another approach is to employ a graph coarsening layer, where graph clustering methods are used to group nodes into clusters before aggregating the node embeddings of these clusters by element-wise max-pooling [2, 4]. This is followed by a graph convolutional layer to form a two-layer stack. This two-layer stack is repeated until a single graph embedding vector is produced.

3 Experiments

3.1 Gene expression data set

I perform my prediction task on gene expression data from the breast cancer data set collected by Wang et al. [18]. Primary tumour samples from 286 breast cancer patients were analysed using Affymetrix HG-U133A Gene Chips in order to obtain individual gene expression profiles. Patients are classified as "relapse" if metastasis tumours are detected in follow-up visits within 5 years of breast cancer surgery, and "no relapse" if no tumour was detected in the 5 year period. There were a total of 179 "no relapse" patients (negative class) and 107 "relapse" patients (positive class). The data set is publicly available in the Gene Expression Omnibus (GEO) repository and can be assessed through its accession number *GSE2034*.

I pre-processed the gene expression data for use in experiments. Probesets in the data are mapped to their respective Entrez Gene IDs, in order to match the IDs of biological pathway information. Ambiguous and unannotated probesets were removed, and the expression level of genes with multiple probeset matches were determined by the probeset with the highest value.

3.2 Biological pathway information

The Kyoto Encyclopedia of Genes and Genomes (KEGG) database [8] is a manually curated database that contains biological pathway information. A total of 330 human pathways were downloaded from the database in XML format. These pathways were pre-processed to remove duplicate edges and loops in the graphs. Also, nodes corresponding to genes that are not present in the gene expression data were also removed from the pathways. The largest connected subgraph of each pathway is used to represent the pathway. 158 pathways that have more than 30 edges were selected for use in generating graph representations. Graph information of the pathways are passed as edge lists to the graph representation learning algorithm GAE [11].

3.3 Data features

Gene-level features I evaluated the use of different numbers of genes (20, 100, 1000, 1983, 8654) as features for the classification task, and observed that using the top 100 ranked genes as features gave the best results. For the selection of the top 20, 100 and 1000 ranked genes, genes were ranked by their p-value in a two sample t-test between the positive and negative class. The 1983 genes used were present in the 158 pathways selected for constructing graph representations, and were not ranked by statistical significance. I also used the total number of 8654 genes present in the gene expression data as features.

Pathway-level features Pathway scores are calculated by averaging the expression values of every gene in the pathway. For an individual, all of the 158 selected pathways were scored and used as features for the classification task.

Graph representations The graph autoencoder (GAE) by Kipf and Welling [11] was used to learn representations of the graphs that are encoded by the 158 pathways in an unsupervised setting. The encoded graphs are simple, undirected and unweighted graphs, and hence have adjacency matrices which are symmetric binary matrices. All the individual nodes in each pathway correspond to a gene that is present in the gene expression data. For each individual, every node in all the 158 pathways is assigned an expression value according to his or her gene expression data. The graph information of each pathway is converted from an edge list to an adjacency matrix before being fed into the GAE along with the expression values of each node (provided in a separate node feature matrix). A total of 45188 representations were learned, one for each of the 158 pathways belonging to each of the 268 patients.

A GAE with a GCN [10] encoder consisting of a 32-dim and 16-dim hidden layer was used to learn graph representations. To construct each graph representation, the GAE was trained for 250 epochs using the Adam optimiser [9]. After training, the 16-dim hidden layer activations were taken as individual *node embeddings*. The node embeddings are presented as an $N \times D$ matrix where N is the number of nodes and D is the dimension of node embeddings. We obtain 158 $N_i \times 16$ matrices for each individual, where N_i depends on the number of nodes in the respective pathway. Subsequently, *graph embeddings* were generated using a sum-based approach proposed by Duvenaud et al. [5], where individual node embeddings are summed up element-wise. This results in a 1×16 graph embedding vector for each pathway. The 158 node embedding matrices of each individual are concatenated row-wise before being used as inputs for the classifier. The graph embedding vectors are also processed in the same way.

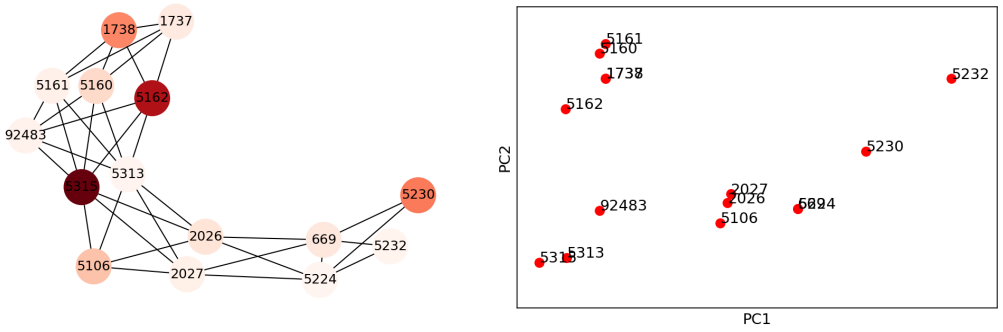
3.4 Classification algorithms

Different classification algorithms were used to demonstrate that the differences in performance are due to the difference in data features and happen regardless of the type of classifier. One-dimensional data features were fed into low capacity classifiers such as logistic regression and support vector machines (SVM), as well as classifiers with higher capacities such as the multi-layer perceptron and random forest classifier. The hyperparameters for each classification algorithm was optimised to suit the dimensions of each particular data feature. The two-dimensional graph representations were fed into convolutional neural networks (CNNs) for classification. The architecture of the CNNs consists of two convolutional layers with 7×3 filters, followed by two fully connected layers of size 64 before the output layer. Batch normalisation and dropout layers were added between every convolutional and fully connected layer.

4 Results

4.1 Node embeddings

The node embeddings learned by GAE can be visualised by means of a principal component analysis (PCA). Figure 1a shows the network graph of the *hsa00010* human pathway of a "no relapse" patient, while Figure 1b is a visualisation of its node embeddings. The aim of graph representation learning algorithms is to generate node embeddings where the geometric distance between nodes reflect their relationships in the graph very closely. We can expect good node embeddings to have adjacent



(a) Network graph. Nodes are coloured according to their gene expression values, with darker red colours indicating higher expression levels.

(b) PCA plot of node embeddings

Figure 1: Largest connected subgraph of the *hsa00010* human pathway. Individual nodes and points are labelled with their respective Entrez Gene IDs.

nodes that are close together in the embedding space. This can be observed in Figure 1b, where the Euclidean distances between the nodes reflect their relations in the graph. However, a few discrepancies can be noted where adjacent nodes are not close together in the embedding space (e.g. nodes 92483-5315, 92483-5162). In addition, nodes 2026 and 2027 that are close together in the embedding space are not adjacent to each other. A possible reason for nodes 2026 and 2027 being close together is because of the graph convolutions that GAE performs, which aggregate the attributes of one-hop neighbours of each node. Nodes 2026 and 2027 can be observed to have a Jaccard's coefficient of 100% between their neighbours (i.e. they share all of their adjacent neighbours).

4.2 Prediction of breast cancer metastasis

In this paper, the binary classification task at hand is to predict whether breast cancer patients will relapse and develop tumour metastasis over the next five years following breast cancer surgery. I evaluate the performance of several machine learning classifiers on the binary classification task when different data features are fed as inputs into the classifiers. The three main groups of data features can be categorised as gene-level features, pathway-level features and graph representations. I report only the most significant results in Table 1, as the main aim of this paper is not to assess the performance of different classifiers. In general, classifiers that received gene-level features achieved the best performance in the binary classification task, followed by those that received pathway-level features. In particular, the logistic regression classifier that received top 100 ranked genes as its features achieved the highest accuracy of 77.78%. The classifiers that received graph representations as inputs performed poorly on the test set.

The CNNs that were trained on both the GAE node embedding and graph embedding matrices were able to achieve an accuracy of $\sim$93% on the training set. However, they were unable to generalise to the test set. This indicates that the CNNs were overfitting to the training set, and failed to learn a hypothesis that could associate the gene expression and biological pathway information of an individual with the correct output class. This is likely caused by the problem of having a small data set, which is exacerbated by the high dimensionality of the graph representations. In order to investigate whether the node embeddings provide meaningful information for the classification task, I performed a PCA visualisation of node embeddings that belong to different individuals for each pathway separately. Figure 2 shows a plot of one of the pathways, and it can be observed that there is a large amount of noise in the node embeddings between the individuals. Also, no clear distinction between node embeddings that belong to different classes can be seen.

5 Conclusion

The viability of using graph representation learning algorithms such as GAE to integrate biological pathway and gene expression information for classification still needs to be evaluated more thoroughly,

Table 1: Classification results on the breast metastasis data set [18]. Recall and precision scores are calculated with "relapse" patients as true positives and "no relapse" patients as true negatives.

Data features	Classifier	Accuracy (%)	Recall (%)	Precision (%)
Top 100 genes	Logistic regression	77.78	64.28	75.00
	SVM (linear)	76.39	60.71	73.91
	Random forest	73.61	39.29	84.62
1983 genes that are present in the 158 selected pathways	Logistic regression	68.06	60.71	58.62
	SVM (linear)	69.44	60.71	60.71
	Multi-layer perceptron	62.50	96.43	50.94
Scores of 158 selected pathways	Logistic regression	68.06	60.71	58.62
	SVM (linear)	68.06	53.57	60.00
	SVM (polynomial)	66.66	67.86	55.88
GAE node embeddings	2D CNN	52.77	28.57	57.14
GAE graph embeddings	2D CNN	51.39	10.71	23.08

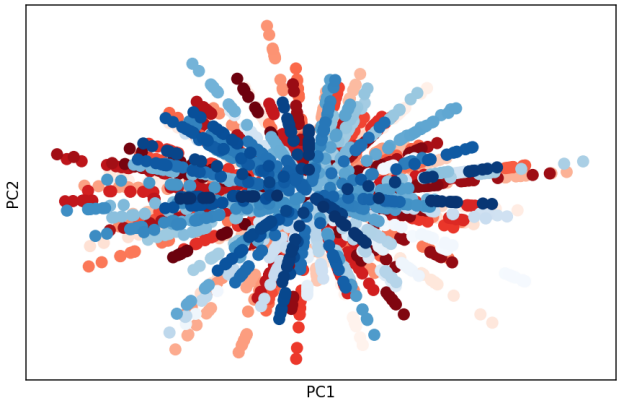


Figure 2: PCA plot of the node embeddings of the *hsa00010* human pathway for all 286 patients. Red and blue nodes represent "no relapse" and "relapse" patients respectively. Different shades of red and blue indicate nodes belonging to different patients.

with an example being to find out whether the graph representations preserve more information about the structure of the graph or the individual node attributes. Also, it should be noted that the contrasting working principles behind different graph representation learning algorithms result in different graph representations that may be more or less suited to a specific machine learning task. Performing feature selection before proceeding with the graph representation learning may help to reduce overfitting and reduce noise in the data. For example, selecting pathways based on the statistical significance of their genes in differentiating between the two classes may help to preserve more relevant pathways. A current limitation of graph representation learning methods is that they are mostly limited to accepting undirected graphs as inputs. The few methods that are able to do so are unable to learn graph representations that are able to effectively represent the directed relationships [16, 20]. In conclusion, the use of graph representations to provide a purposeful encoding of biological pathway and gene expression information has to be further investigated before it can be used effectively.

References

[1] 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.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[3] Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3(1):140, 2007.

[4] 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.

[5] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[6] 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.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[8] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[11] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[12] Nevan J Krogan, Scott Lippman, David A Agard, Alan Ashworth, and Trey Ideker. The cancer cell map initiative: defining the hallmark networks of cancer. *Molecular cell*, 58(4):690–698, 2015.

[13] Eunjung Lee, Han-Yu Chuang, Jong-Won Kim, Trey Ideker, and Doheon Lee. Inferring pathway activity toward precise disease classification. *PLoS computational biology*, 4(11):e1000217, 2008.

[14] 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.

[15] Kevin Lim, Zhenhua Li, Kwok Pui Choi, and Limsoon Wong. A quantum leap in the reproducibility, precision, and sensitivity of gene expression profile analysis even when sample size is extremely small. *Journal of bioinformatics and computational biology*, 13(04):1550018, 2015.

[16] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

[17] 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.

[18] Yixin Wang, Jan GM Klijn, Yi Zhang, Anieta M Sieuwerts, Maxime P Look, Fei Yang, Dmitri Talantov, Mieke Timmermans, Marion E Meijer-van Gelder, Jack Yu, et al. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, 365(9460):671–679, 2005.

[19] Eng-Juh Yeoh, Mary E Ross, Sheila A Shurtleff, W Kent Williams, Divyen Patel, Rami Mahfouz, Fred G Behm, Susana C Raimondi, Mary V Relling, Anami Patel, et al. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer cell*, 1(2):133–143, 2002.

[20] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.