Drug Similarity Integration Through Attentive Multi-view Graph Auto-Encoders

Tengfei Ma[1,2]**, Cao Xiao**[1,2]**, Jiayu Zhou**[3]**, Fei Wang**[4]

[1] IBM Research
[2] MIT-IBM Watson AI Lab
[3] Computer Science and Engineering, Michigan State University
[4] Weill Cornell Medical School, Cornell University

tengfei.ma1@ibm.com, cxiao@us.ibm.com, jiayuz@msu.edu, few2001@med.cornell.edu

Abstract

Drug similarity has been studied to support downstream clinical tasks such as inferring novel properties of drugs (e.g. side effects, indications, interactions) from known properties. The growing availability of new types of drug features brings the opportunity of learning a more comprehensive and accurate drug similarity that represents the full spectrum of underlying drug relations. However, it is challenging to integrate these heterogeneous, noisy, nonlinear-related information to learn accurate similarity measures especially when labels are scarce. Moreover, there is a trade-off between accuracy and interpretability. In this paper, we propose to learn accurate and interpretable similarity measures from multiple types of drug features. In particular, we model the integration using multi-view graph auto-encoders, and add attentive mechanism to determine the weights for each view with respect to corresponding tasks and features for better interpretability. Our model has flexible design for both semi-supervised and unsupervised settings. Experimental results demonstrated significant predictive accuracy improvement. Case studies also showed better model capacity (e.g. embed node features) and interpretability.

1 Introduction

The rapidly evolving technologies have made it easier to collect multiple types of drug data and thus opened new opportunities for computational drug discovery research and drug safety studies. The study of drug similarity paves the foundation for these research since similar structural, molecular and biological properties often relate to similar drug indications or adverse effects [Vilar *et al.*,]. In literature, drug similarity has been computed using molecular structure data [Jin *et al.*, 2017], interaction profile data [Fokoue *et al.*, 2016], as well as side-effect information [Cheng and Zhao, 2014; Liu *et al.*, 2012].

Recently, there has been a growing interest in learning improved drug similarity from multiple types of drug features. For example, [Li *et al.*, 2015] proposed an inductive matrix completion method to combine multiple data sources and

help predict the unknown side effects. [Zhang *et al.*, 2015] proposed an integrative label propagation algorithm to infer clinical side effects from multiple sources with considering high-order similarity. Results from these pilot studies show that combined similarity measures are usually more informative and robust to noise. These methods could be summarized into four major categories: the nearest neighbor method, the random walk based approaches, the unsupervised, and the multiple kernel learning methods. Section 2 provides more details of the related literature.

Despite potential benefits, when learning from multiple biomedical data sources, significant challenges arise from the simultaneous handling of the following issues: 1) different types of features have different levels of associations with targeting outcomes. For example, drugs' structural similarity could have more influence on their interaction profiles than drugs' indication similarity do; 2) the underlying relations of biomedical events (e.g., two drugs interact to cause a side effect) are often nonlinear and complex over all types of features [Peter *et al.*, 2006]; 3) data quality (e.g. lack of label, noise in the data) also creates challenges for similarity learning, and 4) a model that captures complex drug relations is often be very complex and lacking interpretability.

To address the aforementioned challenges, we consider each type of drug feature as a view and learn integrated drug similarity using multi-view graph autoencoders (GAE). In particular, we model each drug as a node in the drug association network and extend the graph convolutional networks (GraphCNN) [Kipf and Welling, 2016a] to embed multi-view node features and edges. Across views, we use attentive view selection scheme to enable nonlinear multi-view fusion and make the learning more interpretable and adaptive to data. By such embedding, we learn drug similarity and use them to predict outcomes (e.g., drug-drug interactions). In addition, for the setting where we integrate multiple drug similarity graph without knowing any features, we propose an alternative transductive learning method based on treating labels as latent variables. The proposed models not only improve prediction performance, but also have the following benefits.

- *Intepretable and adaptive multiview fusion*: To model the heterogeneous relevance among different views with targeting tasks, in our similarity integration, we use attentive model to fuse multiple views. The attentive view selection scheme generates task-wise feature relevance, by

which we could learn interpretable similarity measures. Also the learned similarity would be more adaptive to the underlying data, thus is more accurate.

- *Transductive prediction using unlabeled data*: Labels are expensive to acquire, and often very scarce for new drugs. By developing an auto-encoder structure, whose reconstruction loss could be seen as a regularization term that explicitly models the information of graph structure, we efficiently leverage the unlabeled data for accurate predictions.

- *Robust-to-noise*: The proposed methods inherit the advantage of autoencoders and can extract representations that are relatively stable and robust to the noise in the data, e.g. in the drug-drug interaction prediction case, sometimes unseen interactions might not indicate no interaction. The proposed methods effectively reduce the negative impacts caused by these "positive unlabeled" samples.

2 Related Work

Our work addresses the problem of multiview similarity integration. To our best knowledge, current approaches mainly could be summarized as below.

The nearest neighbor methods that make predictions based on majority cases among neighbors. To name a few, [Zhang *et al.*, 2016], [Zhang *et al.*, 2017], and [Zhang *et al.*, 2015]. However, as pointed out by [Zhang *et al.*, 2015], most of these existing methods only utilize first-order similarity to construct neighborhood and do not consider transitivity of similarities. **The random walk methods** (e.g., label propagation in [Zhang *et al.*, 2015] and [Wang *et al.*, 2010]) that leverage the assumption that data points occupying the same manifold are very likely to share the same semantic label, and then aim to propagate labeling information from labeled data points to unlabeled ones according to the intrinsic data manifold structures collectively revealed by a large number of data points. These methods can handle nonlinear relations and perform transductive learning with scarce labeled data. However, these models have fixed loss functions, hence lack of flexibility in modeling various problem settings. **The unsupervised methods** For example, in [Wang *et al.*, 2014] and [Angione *et al.*, 2016], the authors construct an integrative network to fuse multiple similarity networks via an iterative scaling approach. In [Xu *et al.*, 2016],the authors integrated feature ranking and feature variation as feature weights for weighted similarity fusion. These unsupervised methods have good flexibility, however without any supervision, unreliable results could be generated. **The multiple kernel learning (MKL) methods** such as [Zhuang *et al.*, 2011]. MKL were further extended to integrate heterogeneous data in [McFee and Lanckriet, 2011], however, most existing methods are often limited to convex integration.

3 Background

Over the past few years, several graph-based convolutional network models emerged for inducing informative latent fea-

ture representations of nodes and links. For example, [Kipf and Welling, 2016a] proposed a new graph convolutional network (GraphCNN) that learns node embeddings based on node features and their connections, which could be used in node classification. Specifically, given an undirected graph with nodes X and adjacency matrix A, a multi-layer neural network is constructed on the graph with the following layer-wise propagation rule:

$$\boldsymbol{H}^{(l+1)} = f\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right)$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ is the adjacency matrix with added self-connections, $\boldsymbol{D}$ is a diagonal matrix such that $\boldsymbol{D}_{ii} = \sum_j \tilde{\boldsymbol{A}}_{ij}$, $\boldsymbol{W}^{(l)}$ is a layer-specific parameter matrix, H^l is the node representation in the lth layer, and f is an activation function (e.g. ReLU or sigmoid). Later, [Schlichtkrull *et al.*, 2017] and [Kipf and Welling, 2016b] extended GraphCNN and proposed a graph auto-encoder (GAE) using GraphCNN for both node classification and link prediction tasks. However, their model only reconstructs the edges, and cannot work on unseen data. In the following, we will make further extension based on [Kipf and Welling, 2016b] and [Schlichtkrull *et al.*, 2017] in terms of reconstructing both links and node embeddings and allowing for inductive prediction.

4 Method

In this paper, we consider each type of drug feature as a view. For view $u \in \{1, \cdots, T\}$, we construct a graph by modeling each drug as a node and the similarity between two nodes as an edge. We denote node feature embeddings as $\boldsymbol{Z}^{(u)}$ and use similarity matrix $\boldsymbol{A}^{(u)}$ to represent the pair-wise similarity between drugs on that view. Given T different views, the task of multi-view similarity integration is to derive an integrated node embedding $\boldsymbol{Z}$ and similarity matrix $\boldsymbol{A} \in \mathbb{R}^{n*n}$ across all views.

4.1 Similarity Integration with Attentive Multi-view Graph Autoencoders

Basic GraphCNN Structure with Multiple Views For each view u, we set $\tilde{\boldsymbol{A}}^{(u)} = \boldsymbol{A}^{(u)} + \boldsymbol{I}_N$ and diagonal matrix $\boldsymbol{D}^{(u)}$ where $\boldsymbol{D}_{ii}^{(u)} = \sum_j \tilde{\boldsymbol{A}}_{ij}^{(u)}$, then we use a two-layer GraphCNN to get the node embeddings $\boldsymbol{Z}^{(u)}$ using Eq. (1).

$$
\begin{aligned}
\boldsymbol{Z}^{(u)} &= f(\boldsymbol{X}^{(u)}, \boldsymbol{A}^{(u)}; \boldsymbol{W}_1^{(u)}) \\
&= \text{Softmax}\left(\hat{\boldsymbol{A}}^u \text{ReLU}(\hat{\boldsymbol{A}}^u \boldsymbol{X}^{(u)} \boldsymbol{W}_0^{(u)}) \boldsymbol{W}_1^{(u)}\right),
\end{aligned} \tag{1}
$$

where $\hat{\boldsymbol{A}}^u = \boldsymbol{D}^{(u)-1/2}\tilde{\boldsymbol{A}}^{(u)}\boldsymbol{D}^{(u)-1/2}$, $\boldsymbol{W}_0^{(u)}$ and $\boldsymbol{W}_1^{(u)}$ are weight matrices. Given the node embeddings $\boldsymbol{Z}^{(u)}$ on each view u, we concatenate the embedding from each view $\boldsymbol{Z}^{(u)}$ to get a new representation of the node $\boldsymbol{Z}$. The prediction between two nodes x_i and x_j could be done by a sigmoid function $y(x_i, x_j) = \sigma(\boldsymbol{z}_i^T \boldsymbol{W} \boldsymbol{z}_j)$ with a matrix parameter $\boldsymbol{W}$. The structure of this method is shown in Figure 1(a).
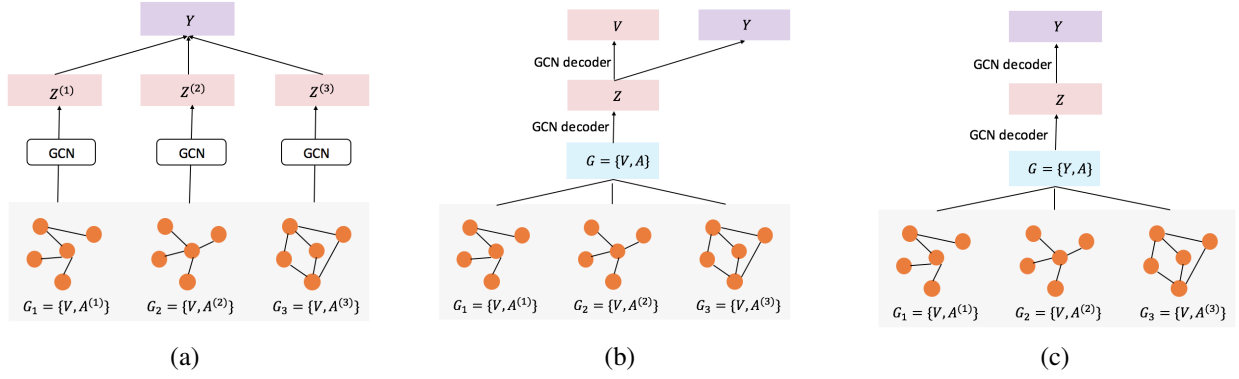
Figure 1: The illustration of GraphCNN for link prediction: (a) The basic GraphCNN structure with multiple similarity matrices. (b) Semi-supervised graph auto-encoder based on GraphCNN. (c) Transductive graph auto-encoder.

Similarity Matrix Fusion Instead of concatenating node embeddings in different views, we can also first get an integrated similarity matrix and construct only one graph for all views. In this single graph, the nodes features are fixed for all views. And the similarity fusion could be simply done as follows: considering the complexity of normalization, to fuse similarity we first normalize all similarity matrices $\boldsymbol{A}^u$ to get $\hat{\boldsymbol{A}}^u$, and then aggregate all similarity matrices to get a comprehensive one as the adjacency matrix of the graph: $\hat{\boldsymbol{A}} = \frac{1}{T}\sum \alpha_u \hat{\boldsymbol{A}}^u$, where $\{\alpha_u\}$ are mixing weights for different similarity matrices. Following the structure in [Kipf and Welling, 2016a], we use a one-layer GraphCNN to encode the nodes in our graph:

$$\boldsymbol{Z} = f(\boldsymbol{X}, \hat{\boldsymbol{A}}) = \text{Softmax}(\hat{\boldsymbol{A}}X\boldsymbol{W}_0) \qquad (2)$$

After that, we decode the embedding back to the original feature space

$$\boldsymbol{X}' = f'(\boldsymbol{Z}, \hat{\boldsymbol{A}}) = \text{Sigmoid}(\hat{\boldsymbol{A}}\boldsymbol{Z}\boldsymbol{W}_1) \qquad (3)$$

If we do not have any labels for the nodes, the objective function is the loss of the auto-encoder in Eq. 4.

$$L_{\text{ed}} = \sum ||\boldsymbol{X} - \boldsymbol{X}'||^2 \qquad (4)$$

In this case, our framework could be regarded as an unsupervised multi-graph fusion and embedding method. The derived similarity matrix $\hat{\boldsymbol{A}}$ can be used for other tasks as well, such as node clustering.

Attentive View Selection In practice, the fusion of each view could be nonlinear, while the weights of features in each view need to be decided by both the data and the targeting tasks. To allow for such a flexibility, in this section we extend the mixing scheme by fusing features from different views with attention mechanism, where weights of features in each view are determined by corresponding inputs. The attentive view selection scheme is illustrated in Fig. 2.

Assume we have adjacency matrix $\boldsymbol{A}^u$ for view u, we assign attention weights $\boldsymbol{g}^u \in \mathbb{R}^{N*N}$ to the graph edges, such that the integrated adjacency matrix becomes $\sum_u \boldsymbol{g}^u \odot \boldsymbol{A}^u$, where $\odot$ is the element wise multiplication. For each view, we first project the original adjacency matrix to an unnormalized

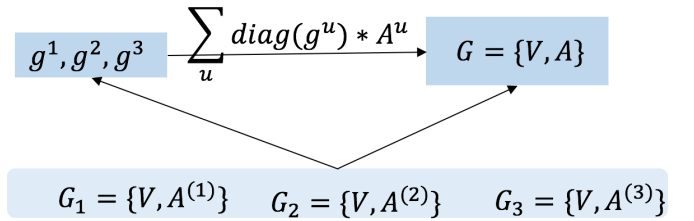Figure 2: The illustration of attention view selection scheme.

matrix $\boldsymbol{g}'^u = \boldsymbol{W}^u \boldsymbol{A}^u + \boldsymbol{b}^u$, and then normalize them over different views to get the attention weights $\boldsymbol{g}^u$. In practice, the graph is often large, thus there will be too many parameters for the attention calculation if we use a fully connected $N*N$ attention matrix. To reduce the complexity, we alternatively employ a diagonal attention matrix. To be specific, we limit $\boldsymbol{g}^u$ to be a vector, and form the weighted similarity matrix by $diag(\boldsymbol{g}^u) * \boldsymbol{A}^u$. In this way, the size of parameters (i.e. $\boldsymbol{W}^u$ and $\boldsymbol{b}^u$) is reduced from $N*N$ to N. And the attentive similarity matrix is generated as follows: $\boldsymbol{g}'^u = \boldsymbol{W}^u \boldsymbol{A}^u + \boldsymbol{b}^u$, where $\boldsymbol{W}^u, \boldsymbol{b}^u \in \mathbb{R}^{1*N}$.

Then we normalize them to get the attention weights for each position i: $\boldsymbol{g_i} = (g_i^1, ..., g_i^T) = \text{Softmax}(g_i'^1, ..., g_i'^T)$, and $\boldsymbol{g^u} = (g_1^u, ...g_N^u)$ is then used to induce the final similarity matrix $\boldsymbol{A} = \sum_u \text{diag}(\boldsymbol{g}^u) * A^u$, where $*$ is the matrix multiplication, $\text{diag}(\boldsymbol{g}^u)$ is a diagonal matrix of g^u as its diagonal value. After we get the new attention based similarity matrix $\boldsymbol{A}$, we can use the same framework as 4.2 and 4.3.

4.2 A Semi-supervised Extension Given Partial Labeled Data

The graph auto-encoder (GAE) structure could be further extended to a semi-supervised setting when we have labels for some of the nodes in the graph (in our case drugs). We could keep the auto-encoder framework unchanged, and predict the labels on training data using a network h: $h(\boldsymbol{Z}) = \text{Softmax}(\boldsymbol{W}_h \boldsymbol{Z} + \boldsymbol{b}_h)$. The prediction loss of h is formulated by Eq. (5).

$$L_{\text{train}} = L(\boldsymbol{Y}_{\text{train}}, h(\boldsymbol{Z}_{\text{train}})) = \sum_{y \in \boldsymbol{Y}_{\text{train}}} \sum_{y' \in h(\boldsymbol{Z}_{\text{train}})} -y \ln y' \qquad (5)$$

This new model then integrates the two loss functions as its objective function

$$L = L_{\text{train}} + \lambda L_{\text{ed}} \tag{6}$$

Compared to a generic neural network, which generally contains only the L_{train}, Eq. (6) can be seen as adding an auto-encoder loss as the regularization term L_{ed}. In a graph based semi-supervised learning framework, the graph Laplacian regularization is often used as the regularization to capture the graph structure. $L = L_{\text{train}} + \lambda L_{\text{lap}}$, where $L_{lap} = \sum_{i,j} \boldsymbol{A}_{i,j} ||h(x_i) - h(x_j)||^2$. Our objective function replaces the second term with the reconstruction loss of the GAE, which also explicitly models the graph structure information.

4.3 Transductive Learning using Test Labels as Variables

Sometimes when we only have the graph structure of the similarity matrix but no node features, although we could model them using one-hot representation as in [Kipf and Welling, 2016a] (for details, see Appendix A.1 in [Kipf and Welling, 2016a]), such embedding is typically not efficient. More importantly decoding the embedding vectors to the one-hot vectors cannot gain much information. This motivates us to develop another scheme to extend the previous introduced graph auto-encoder to improve learning given no node feature.

Instead of using the original node features or one-hot node vectors in GAE, we consider an alternative way: we use the training labels (i.e. DDI links of each node) as inputs and reconstruct them using the same GAE structure as in Figure 1(c). So the graph auto-encoder would output the predicted links $\boldsymbol{Y}' = f'(f(\boldsymbol{Y}, \hat{\boldsymbol{A}}), \hat{\boldsymbol{A}})$.

Moreover, if we consider similarities as graph edges, the labels of the test nodes would also impact the decoding of the training nodes. So we employ a transductive method to use the test labels as additional latent variables. The predicted labels are formulated as follows:

$$\{\boldsymbol{Y}'_{\text{train}}, \boldsymbol{Y}'_{\text{test}}\} = f'(f(\{\boldsymbol{Y}_{\text{train}}, \boldsymbol{Y}_{\text{test}}\}, \hat{\boldsymbol{A}}), \hat{\boldsymbol{A}}) \tag{7}$$

i.e. $\boldsymbol{Y}'_{\text{test}}$ is a function of $\boldsymbol{Y}_{\text{test}}$ when $\boldsymbol{Y}'_{\text{train}}$, $\boldsymbol{Y}_{\text{train}}$ and $\hat{\boldsymbol{A}}$ are known. The objective function of this model is then given by:

$$\min_{Y_{test}} L = ||\boldsymbol{Y}'_{\text{train}} - \boldsymbol{Y}_{\text{train}}||^2 + ||\boldsymbol{Y}'_{\text{test}} - \boldsymbol{Y}_{\text{test}}||^2 + \mu ||\boldsymbol{Y}_{\text{test}}||^2,$$

where $\mu ||\boldsymbol{Y}_{\text{test}}||^2$ is a regularization term which enforces stability of the solutions. Thus after inference from the training data, we can get the optimal neural network parameters as well as the latent variables Y_{test}.

5 Experiment

Detecting adverse drug-drug interaction (DDI), a modification of the effect of a drug when administered with another drug, is one of the clinically important applications as DDIs result in large amounts of fatalities per year and incur huge morbidity and mortality related cost of $\sim \$177$ billion annually [Giacomini *et al.*, 2007]. Making use of multiple drug characterizations in similarity computation is critical since drugs could have heterogeneous similarity in different feature dimensions, e.g. drugs that have similar chemical structures could have very different therapeutic target and thus result in different DDI mechanism.

5.1 Data Sources

Binary Prediction of the Occurrence of DDIs: For the first data set, we will integrate multiple similarity graph (without node feature) to predict whether there will be interaction between a new pair of drugs. In the data, we have the following views: 1) DDI: The known labels of DDIs are extracted from the Twosides database [Tatonetti *et al.*, 2012], including 645 drugs and 1318 DDI events, in total 63473 distinct pairs of drugs associated with DDI reports. 2) Label Side Effect: Drugs' side effects extracted from SIDER database [Kuhn *et al.*, 2015] are considered one type of features, including 996 drugs and 4192 side effects. We call this view as "Label Side Effect" by the convention in [Zhang *et al.*, 2015]. 3) Off-Label Side Effect: Drugs confounder-controlled side effects from OFFSIDES dataset are considered another type of features, including $1,332$ drugs and $10,093$ side effects. 4) Chemical Structure: Drug structure features (i.e. chemical fingerprints) are structural descriptors of drugs. In our study, we generate drug structure features with the extended-connectivity fingerprints with diameter 6 (ECFP6) using the R package "rcdk [**?**]". The features are hashed binary vectors of 1,024-bit length, of which each bit encodes the presence or absence of a substructure in a drug molecule. We used Jaccard index to compute similarities between all the fingerprints.

Multilabel Prediction of Specific DDI Types: For the second data, we integrate multiple type of drug views to predict specific interaction types among 1301 candidate types for new drug pairs. In the data, we have 222 drugs and the following views: 1) Drug Indication: The drug indication data of dimension 1702 is downloaded from SIDER [Kuhn *et al.*, 2015]. It is originally generated from MedDRA database, which is a widely used clinically-validated international medical terminology. 2) Drug chemical protein interactome (CPI): The CPI data from [Wishart *et al.*, 2008] provides an important measure about how much power a drug needs to bind with its protein target. Its dimension is 611. The similarity of CPI is calculated using the RBF kernel. 3) Protein and nucleic acid targets (TTD): For each drug, we associate its multiple protein and nucleic acid targets information and generate features of dimension 207. These entries are extracted from the Therapeutic Target Database (TTD) [Chen *et al.*, 2002]. 4) Chemical Structure: The chemical structure features are extracted in the same way as in dataset 1, except that we chose "pubchem" fingerprint instead, whose feature dimension is 582.

5.2 Implementation and Evaluation Strategy

Proposed Model: We implement the proposed model with Tensorflow 1.0 [Abadi *et al.*, 2015] and trained using Adam with learning rate 0.01 and early stopping with window size 30. We optimized the hyperparameter for SemiGAE on validation data and then fixed for all GAE models: 0.5 (dropout rate), 5e-4 (L2 regularization) and 64 (# of hidden units). For GCN models, we have the second layer and the

number of the hidden units in the second layer is set as 32.

Baseline: In addition, we implemented the following four baselines for comparison:

- Nearest Neighbor (NN): We implemented the NN method in [Vilar *et al.*, 2012]. It identifies novel DDIs by using the nearest neighbor similarity to drugs involved in established DDIs.

- Label Propagation (LP): We considered the LP model in [Zhang *et al.*, 2015] as a baseline. The LP method propagates the existing DDI information in the network to predict new DDIs, and could also integrate multiple similarity matrices in the network.

- GraphCNN: For single view, we use the same structure as the nonprobabilistic GAE model in [Kipf and Welling, 2016b]. We consider the DDI links as edges and form the adjacency matrix. For multiple views, we linearly integrate all similarity matrices as well as the training DDI links.

- Multiple Kernel Learning (MKL): For MKL, we used the python "Mklaren" library [Strazar and Curk, 2016]. We set $\sigma = 30$ for RBF and degree $p = 3$ for polynomial kernel. We only applied MKL on Data 2 since Data 1 does not have features for all views.

For all models, we use Tanimoto coefficient (TC) to calculate similarity except for CPI. For CPI, we measure drug similarity using RBF kernel. For all methods (except NN which already has the similar procedure in its method), following the procedures in [Zhang *et al.*, 2016], after getting the predicted labels using our model, we calculate the probability of that drug i interacts with drug j by $score_{ji} = score_{ij} = Y_{ij} + Y_{ji}$.

Evaluation: In evaluation, we adopted strategies in [Zhang *et al.*, 2015] and randomly selected a fixed percentage (i.e., 25% and 50%) of drugs, and moved all DDIs associated with these drugs for testing. For the data not in testing, we train on 90% and perform validation and model selection on 10% of the drugs. For testing data, we repeated the hold-out validation experiment 50 times with different random divisions of the data, and reported the mean and the standard deviation of the area under the receiver operating characteristic curve (ROC-AUC) as well as the area under the precision-recall curve (PR-AUC) over the 50 repetitions. In the ROC and PR analytics, we utilized DDI interactions from TWOSIDES as reference positives, and the complement set as reference negatives.

5.3 Results

Table 1 and 2 compare the performance of the proposed models against baselines on both datasets. From the tables we can see for both single view and multi-view, the proposed models significantly outperform baselines. Also, the multi-view models generally outperform corresponding single view models since our integrations provide more comprehensive measures of drug similarity. With adding attention mechanism, the relevant types of features receive more weights in similarity integration.

In addition, we observed that the attentive semi-supervised GAE (AttSemiGAE, the model of Section 4.2) often achieves the best ROC-AUC, which is due to the embedding of node features. This advantage is more obvious on Dataset 2 than Dataset 1, since for Dataset 2, we have node features on all views, while most views in Dataset 1 have no node feature. For Dataset 1, due to the lack of node feature in most views, the attentive transductive GAE (attTransGAE, the model of Section 4.3) gains better PR-AUC thanks to transductive learning from tests labels and adaptive weight learning.

5.4 Case Studies

Understanding the Major Source of Similarity When two drugs cause similar DDIs, such a similarity could be induced by various mechanisms. For example, drugs that prolong the QT interval, drugs that are CYP3A4 inhibitors, or drugs that alter another drug's metabolism via cytochrome P450 interactions or changes in protein binding, etc [Ansari, 2010]. Better understanding the major DDI mechanism would benefit us from developing actionable insights to identify proper ways to prevent DDIs. In this paper, adding attention mechanism enhances the interpretability of the models and could potentially provide understanding of the underlying DDI mechanism.

Table. 3 reports several selected DDIs and the weights of each views as predicted using AttSemiGAE. For example, the DDI "chest pain" has good prediction AUC, and the views "CPI" and "indication" both have more impact on the predictions than other views. We consult domain expert, and find it in line with domain knowledge. Many DDI cases of chest pain are due to particular drug overdose, such as Venlafaxine and Mirtazapine [Nachimuthu *et al.*, 2012], which could be prescribed together to treat depression. However, the co-use of them could cause overdose thus prolong the QT interval via chemical protein interactome (CPI), and eventually cause chest pain. For another DDI "insomnia", one major mechanism is the interaction between cytochrome P450 (CYP) inducers (e.g. Rifampicin) and Hypnosedatives. Insomnia happens when the cytochrome P450 (CYP) inducers significantly induce the metabolism of the newer hypnosedatives and decreased their sedative effects [Hesse *et al.*, 2003]. Such a process was caused by the bindings of chemical structures with proteins. In the results, the weights for "pubchem" and "CPI (compound-protein binding)" are much higher than the rest, in line with knowledge.

Importance of Multiview Feature Integration: We also examined how feature integrations across multiples views of features could help provide more accurate measures of drug similarity.

For example, Acyclovir (Pubchem ID 2022) and Ganciclovir (Pubchem ID 3454), having medium level similarity in indication and TTDS, since Acyclovir is used for treating herpes simplex virus infections and shingles but Ganciclovir is mainly used in more severe Cytomegalovirus diseases and AIDS. However, they both are analogues of 2'-deoxyguanosine and have very high structural similarity (0.961 measured using "Pubchem" fingerprint). The high structural similarity lead to many common DDIs shared by the two drugs according to the groundtruth. Our proposed model adaptively gives more weight to the structural similarity and computes an integrated similarity score at 0.682, but label propagation

Table 1: Predicting Binary DDI Outcomes on Dataset 1.

| | | Using Single View | | | |
| | Methods | Test Split (25%) | | Test Split (50%) | |
		ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Baselines	NN	0.693 ± 0.025	0.504 ± 0.043	0.681 ± 0.027	0.498 ± 0.042
	LP	0.769 ± 0.013	$\mathbf{0.648 \pm 0.028}$	0.752 ± 0.018	0.632 ± 0.035
	GraphCNN	0.723 ± 0.028	0.536 ± 0.050	0.717 ± 0.039	0.528 ± 0.076
Proposed	SemiGAE	$\mathbf{0.788 \pm 0.014}$	0.640 ± 0.036	$\mathbf{0.785 \pm 0.012}$	$\mathbf{0.637 \pm 0.035}$
	TransGAE	0.786 ± 0.013	0.644 ± 0.033	0.776 ± 0.013	0.630 ± 0.039
		Using Multiple Views			
Baselines	LP	0.768 ± 0.013	0.644 ± 0.028	0.751 ± 0.018	0.629 ± 0.036
	GraphCNN	0.696 ± 0.061	0.515 ± 0.066	0.636 ± 0.047	0.457 ± 0.064
Proposed	AttSemiGAE	$\mathbf{0.798 \pm 0.013}$	0.655 ± 0.032	$\mathbf{0.791 \pm 0.017}$	0.642 ± 0.037
	AttTransGAE	0.785 ± 0.012	$\mathbf{0.687 \pm 0.023}$	0.780 ± 0.016	$\mathbf{0.678 \pm 0.031}$

Table 2: Predicting Specific DDI Types (Multiple Outcomes) on Dataset 2.

| | | Using Single View | | | |
| | Methods | Test Split (25%) | | Test Split (50%) | |
		ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Baselines	NN	0.627 ± 0.043	0.594 ± 0.078	0.594 ± 0.033	0.554 ± 0.061
	LP	0.773 ± 0.025	$\mathbf{0.670 \pm 0.052}$	0.747 ± 0.028	$\mathbf{0.650 \pm 0.053}$
	GraphCNN	0.738 ± 0.047	0.594 ± 0.080	0.698 ± 0.090	0.583 ± 0.102
Proposed	SemiGAE	$\mathbf{0.798 \pm 0.029}$	0.661 ± 0.059	$\mathbf{0.784 \pm 0.028}$	0.649 ± 0.059
	TransGAE	0.790 ± 0.028	0.661 ± 0.068	0.770 ± 0.031	0.633 ± 0.080
		Using Multiple Views			
Baselines	LP	0.774 ± 0.025	0.672 ± 0.052	0.748 ± 0.028	0.653 ± 0.055
	GraphCNN	0.601 ± 0.067	0.526 ± 0.120	0.578 ± 0.067	0.526 ± 0.108
	MKL	0.766 ± 0.030	0.650 ± 0.061	0.724 ± 0.026	0.586 ± 0.066
Proposed	AttSemiGAE	$\mathbf{0.802 \pm 0.029}$	$\mathbf{0.678 \pm 0.060}$	$\mathbf{0.786 \pm 0.030}$	$\mathbf{0.662 \pm 0.064}$
	AttTransGAE	0.782 ± 0.026	0.670 ± 0.058	0.764 ± 0.025	0.652 ± 0.061

Table 3: Attention Weights for Selected DDIs

DDI Type	AUC	chem.	indi.	TTDS	CPI
Chest Pain	0.772	0.151	0.303	0.144	0.402
Insomnia	0.755	0.380	0.261	0.078	0.291
Aching Muscles	0.774	0.117	0.301	0.283	0.299

(LP) fails to capture such heterogeneous influences and yields an integrated score at only 0.551, which is an underestimate comparing with the groundtruth.

Similar examples include the similarity between Alprazolam (Pubchem ID 2118) and Estazolam (Pubchem ID 3261) as well as the similarity between Alprazolam (Pubchem ID 2118) and Triazolam (Pubchem ID 5556). The two pairs of drugs have quite low indication similarity, however, they all interact when in combined use with CYP3A4 inhibitors such as Cimetidine, Erythromycin, Norfluoxetine, Fluvoxamine, Itraconazole, Ketoconazole, Nefazodone, Propoxyphene, and Ritonavir. The combined uses will delay the hepatic clearance of Alprazolam, Estazolam or Triazolam, which then cause accumulation and increased severity of side effects from these drugs. The proposed model account for the feature heterogeneity and weight more on the chemical structural feature and CPI feature, leading to integrated similarity at $0.682 \sim 0.720$,

while other methods considered each views homogeneously and the resulting similarity is often low at $0.551 \sim 0.630$.

6 Conclusion

In this paper, we proposed a set of Graph Auto-Encoder based models that perform multi-view drug similarity integration with attention model to perform view selection. The nonlinear and adaptive integration not only offers superior predictive performance but also interpretable results. We extended these GAE models to semi-supervised/transductive settings and predict the unknown DDIs. Experimental results on two real-world drug datasets demonstrated the performance and efficacy of our methods. Future works include expansion along the line of data or model. Data-wise, we could try on a larger drug database to fully exploit of the power of deep learning without overfitting. Model-wise, we will pursuit directly computing integrated similarity across multiple views without the need for calculation similarity for each view first.

Acknowledgments

References

[Abadi *et al.*, 2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[Angione *et al.*, 2016] Claudio Angione, Max Conway, and Pietro Lió. Multiplex methods provide effective integration of multi-omic data in genome-scale models. *BMC Bioinformatics*, 17(4):83, Mar 2016.

[Ansari, 2010] J. Ansari. Drug interaction and pharmacist. *Journal of Young Pharmacists.*, 2(3), 2010.

[Chen *et al.*, 2002] X. Chen, ZL. Ji, and YZ. Chen. Ttd: Therapeutic target database. *Nucleic Acids Res.*, 30(1), 2002.

[Cheng and Zhao, 2014] F. Cheng and Z. Zhao. Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *JAMIA*, 21, 2014.

[Fokoue *et al.*, 2016] A. Fokoue, O. Hassanzadeh, M. Sadoghi, and P. Zhang. Predicting drug-drug interactions through similarity-based link prediction over web data. WWW '16 Companion, pages 175–178, 2016.

[Giacomini *et al.*, 2007] K. Giacomini, R. Krauss, D. Roden, M. Eichelbaum, and M. Hayden. When good drugs go bad. *Nature*, 446:975–977, 2007.

[Hesse *et al.*, 2003] LM. Hesse, LL. von Moltke, and DJ. Greenblatt. Clinically important drug interactions with zopiclone, zolpidem and zaleplon. *CNS Drugs.*, 17(7), 2003.

[Jin *et al.*, 2017] B. Jin, H. Yang, C. Xiao, P. Zhang, X. Wei, and F. Wang. Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction. In *AAAI*, 2017.

[Kipf and Welling, 2016a] TN. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. 2016.

[Kipf and Welling, 2016b] TN. Kipf and M. Welling. Variational graph auto-encoders. 2016.

[Kuhn *et al.*, 2015] M. Kuhn, I. Letunic, LJ. Jensen, and P. Bork. The sider database of drugs and side effects. *Nucleic Acids Res.*, 2015.

[Li *et al.*, 2015] R. Li, Y. Dong, Q. Kuang, Y. Wu, Y. Li, M. Zhu, and M. Li. Inductive matrix completion for predicting adverse drug reactions (adrs) integrating drug–target interactions. *Chemometrics and Intelligent Laboratory Systems*, 144:71–79, 2015.

[Liu *et al.*, 2012] M. Liu, Y. Wu, Y. Chen, J. Sun, Z. Zhao, X. Chen, M. Matheny, and H. Xu. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *JAMIA*, 19(e1):e28–e35, 2012.

[McFee and Lanckriet, 2011] B. McFee and G. Lanckriet. Learning multi-modal similarity. *J. Mach. Learn. Res.*, 12, 2011.

[Nachimuthu *et al.*, 2012] S. Nachimuthu, MD. Assar, and JM. Schussler. Drug-induced qt interval prolongation: mechanisms and clinical management. *Therapeutic Advances in Drug Safety.*, 3(5), 2012.

[Peter *et al.*, 2006] I. Peter, S. Christian, and M. Achim. Drugs, their targets and the nature and number of drug targets. *Nature Reviews Drug Discovery*, 5:821– 834, 2006.

[Schlichtkrull *et al.*, 2017] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling Relational Data with Graph Convolutional Networks. *ArXiv e-prints*, 2017.

[Strazar and Curk, 2016] M. Strazar and T. Curk. Learning the kernel matrix via predictive low-rank approximations. *CoRR*, abs/1601.04366, 2016.

[Tatonetti *et al.*, 2012] NP. Tatonetti, P. Patrick, R. Daneshjou, and RB. Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

[Vilar *et al.*,] S. Vilar, R. Harpaz, E. Uriarte, L. Santana, R. Rabadan, and C. Friedman. Drug drug interaction through molecular structure similarity analysis. *JAMIA*, (6):1066–1074.

[Vilar *et al.*, 2012] S. Vilar, R. Harpaz, E. Uriarte, L. Santana, R. Rabadan, and C. Friedman. Drug—drug interaction through molecular structure similarity analysis. *JAMIA*, 19(6):1066–1074, 2012.

[Wang *et al.*, 2010] F. Wang, P. Li, and AC. Konig. Learning a bi-stochastic data similarity matrix. In *ICDM*, pages 551–560, 2010.

[Wang *et al.*, 2014] B. Wang, A. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, and A. Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11:333–337, 2014.

[Wishart *et al.*, 2008] DS. Wishart, C. Knox, and AC. Guo. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, 36, 2008.

[Xu *et al.*, 2016] Taosheng Xu, Thuc Duy Le, Lin Liu, Rujing Wang, Bingyu Sun, and Jiuyong Li. Identifying cancer subtypes from mirna-tf-mrna regulatory networks and expression data. *PLOS ONE*, 11(4):1–20, 04 2016.

[Zhang *et al.*, 2015] P. Zhang, F. Wang, J. Hu, and R. Sorrentino. Label propagation prediction of drug-drug interactions based on clinical side effects. *Scientific reports*, 5, 2015.

[Zhang *et al.*, 2016] W. Zhang, H. Zou, L. Luo, Q. Liu, W. Wu, and W. Xiao. Predicting potential side effects of drugs by recommender methods and ensemble learning. *Neurocomputing.*, 173, 2016.

[Zhang *et al.*, 2017] W. Zhang, Y. Chen, F. Liu, F. Luo, G. Tian, and X. Li. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *BMC Bioinformatics.*, 2017.

[Zhuang *et al.*, 2011] J. Zhuang, IW. Tsang, and S. Hoi. Two-layer multiple kernel learning. In *AISTATS*, volume 15, pages 909–917, 2011.