A Novel Domain-Aware Graph Convolutional Model for Domain Pricing Prediction

Your Name
Your Institution
your.email@example.com

March 12, 2025

Abstract

In the domain name marketplace, accurate prediction of domain prices is challenging due to wide variations in scale and non-linear relationships between user engagement (e.g., watchlists) and price. We propose a novel Domain-Aware Graph Convolutional Network (DA-GCN) that integrates a log-transformation of features with a learned combination of structural and similarity-based graph information. Our model constructs a combined adjacency matrix as a convex combination of the normalized k-nearest neighbor graph and a feature-based similarity matrix, and then applies a two-layer graph convolution. In this paper, we present the theoretical formulation, key equations, and pseudocode of our approach.

1 Introduction

The prediction of domain prices from online marketplace data is critical for investors and domain brokers. Traditional graph convolutional networks (GCNs) aggregate information based solely on graph structure. However, in our setting, the watchlists feature is extremely heterogeneous, with values spanning several orders of magnitude. In addition, a simple element-wise product of the normalized adjacency and similarity matrices can overly dampen useful signals.

We address these issues by applying a logarithmic transformation to compress the scale of features, and by forming a *combined adjacency matrix* as a weighted convex combination of the normalized kNN graph and a similarity matrix based on watchlist differences.

2 Methodology

Our approach consists of the following key steps:

1. **Data Preprocessing:** We extract two key features: the number of watchlists, and the domain price. Since these values span a large numerical range, we apply a logarithmic

transformation to both features and targets:

$$x' = \log(1 + x), \quad y' = \log(1 + y).$$

2. **Graph Construction:** From the raw watchlists values (prior to log-transformation), we construct a binary k-nearest neighbor (kNN) graph. Let A be the binary adjacency matrix of the graph. We add self-loops and compute the normalized adjacency:

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}},$$

where D is the degree matrix.

3. **Similarity Scaling:** A similarity matrix S is computed from the raw watchlists using an exponential kernel:

$$S_{ij} = \exp\left(-\frac{|x_i - x_j|}{\tau}\right),$$

where τ is a scaling parameter (set to the mean absolute difference between node features).

4. **Combined Adjacency Matrix:** Instead of using the elementwise product $\hat{A} \odot S$, we form a convex combination:

$$A' = \beta\,\hat{A} + (1 - \beta)\,S,$$

and then row-normalize:

$$\tilde{A} = \frac{A'}{\sum_j A'_{ij}}.$$

Here, $\beta \in [0, 1]$ controls the balance between the structural graph signal and the feature-based similarity.

5. **Graph Convolution:** Our two-layer Domain-Aware GCN applies the combined adjacency matrix twice:

$$Z = \tilde{A}\,X'\,W_1, \tag{1}$$

$$H = \mathrm{ReLU}(Z), \tag{2}$$

$$Y_{\text{pred}} = \tilde{A}\,H\,W_2, \tag{3}$$

where X' is the log-transformed feature matrix (of dimension $n \times 1$), and W_1, W_2 are learnable weight matrices.

3 Optimization

We use the Mean Squared Error (MSE) loss:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_{\text{pred},i} - Y'_i\right)^2,$$

and a custom metric, the Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_{\text{pred},i} - Y_i'|}{Y_i' + \epsilon}.$$

Gradient descent is used to update the weights:

$$W_1 \leftarrow W_1 - \eta\,\nabla_{W_1}\mathcal{L}, \quad W_2 \leftarrow W_2 - \eta\,\nabla_{W_2}\mathcal{L},$$

where η is the learning rate.

4 Pseudocode

Algorithm 1 summarizes the training process of our model.

Algorithm 1 Training Domain-Aware GCN

Input: Feature matrix X (raw watchlists), target Y (prices)
Compute log-transformed features: $X' \leftarrow \log(1+X)$, $Y' \leftarrow \log(1+Y)$
Construct binary kNN graph A from X
Compute normalized adjacency: $\hat{A} \leftarrow D^{-1/2}(A+I)D^{-1/2}$
Compute similarity matrix: $S_{ij} \leftarrow \exp\left(-\frac{|x_i - x_j|}{\tau}\right)$
Combine: $A' \leftarrow \beta\,\hat{A} + (1-\beta)\,S$
Row-normalize A' to obtain $\tilde{A}$
Initialize weights W_1, W_2
for $t = 1$ to T **do**
    Forward pass:

$$Z \leftarrow \tilde{A}X'W_1, \quad H \leftarrow \text{ReLU}(Z), \quad Y_{\text{pred}} \leftarrow \tilde{A}HW_2$$

    Compute loss:

$$\mathcal{L} \leftarrow \frac{1}{n}\sum_{i=1}^{n}(Y_{\text{pred},i} - Y_i')^2$$

    Compute gradients ∇_{W_1} and ∇_{W_2} (via backpropagation)
    Update weights:

$$W_1 \leftarrow W_1 - \eta\nabla_{W_1}, \quad W_2 \leftarrow W_2 - \eta\nabla_{W_2}$$

    Compute MAPE for monitoring.
end for
Output: Trained weights W_1, W_2

5 Experimental Setup and Results

We evaluated our model on a dataset of domain listings obtained from an online marketplace. The main steps were:

- **Data Collection and Preprocessing:** Price and watchlists are parsed and normalized using a log-transformation.
- **Graph Construction:** A kNN graph (with $k = 5$) is built using raw watchlists, and the normalized adjacency is combined with the similarity matrix using a convex combination.
- **Model Training:** Our model is trained for T epochs with a learning rate η. The training loss (MSE) and MAPE are monitored.
- **Visualization:** An interactive graph is generated using the PyVis framework, and scatter plots comparing predicted versus actual log-transformed prices are produced.

The experimental results demonstrate that the revised model achieves a lower loss and better error metrics than the baseline. The use of the combined adjacency matrix and log-transformation yields a model that converges more stably.

6 Conclusion

We have presented a novel Domain-Aware Graph Convolutional Network for domain pricing prediction. By applying a logarithmic transformation and constructing a combined adjacency matrix

$$\tilde{A} = \mathrm{RowNorm}\big(\beta\,\hat{A} + (1-\beta)\,S\big),$$

our model better captures both structural and feature-based similarities. Our experimental results show that these modifications reduce the loss and improve performance as measured by MAPE. Future work may extend this approach to incorporate additional features and more complex graph architectures.

Acknowledgments

We thank our colleagues for valuable feedback and the funding agencies for their support.

References