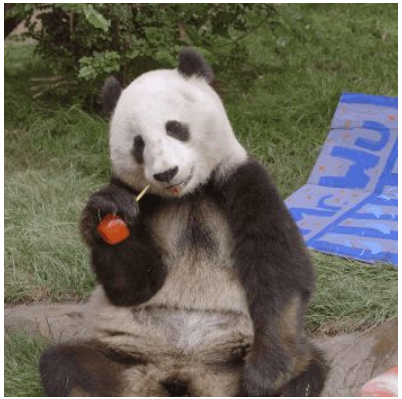
Paper Classification in Directed Citation Network

Team: Panda
Karen Quadros, Purit Punyawiwat, Srishti Majumdar



Introduction

Paper classification is the process of inferring class labels over nodes in a citation network.

In this project, we explore and understand the various methodologies used in this task.

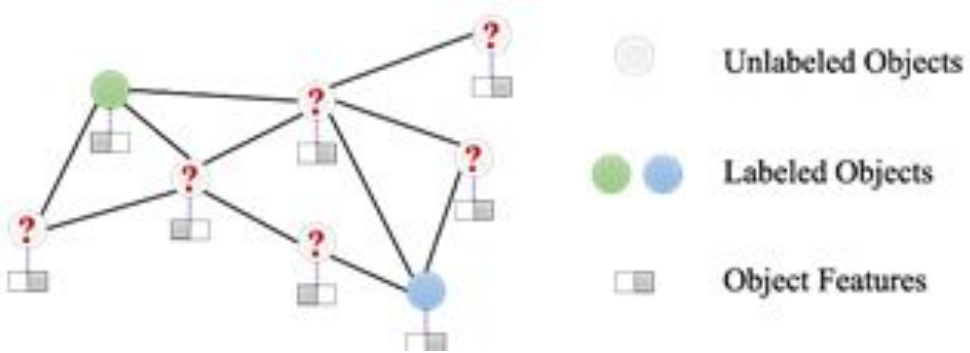
Why is this important?

Data

- **Cora Dataset**
  - 2708 Papers
  - 5429 Links
  - 1432 Unique Words
- **Labels:** Cased Based, Genetic Algorithms. Neural Networks, Probabilistic Methods, Reinforcement Learning, Rule Learning, Theory
- **First Table:** Which paper cites which paper
- **Second Table:** Which word is in each paper
- **Third Table:** The labels of each paper

GMNN

- We used GMNN as a semi-supervised learning approach.
- Reasons why we want to use GMNN
  - Model dependency between objects (Statistical Relational Learning)
  - Learn effective object representations (Graph Neural Network)



GMNN

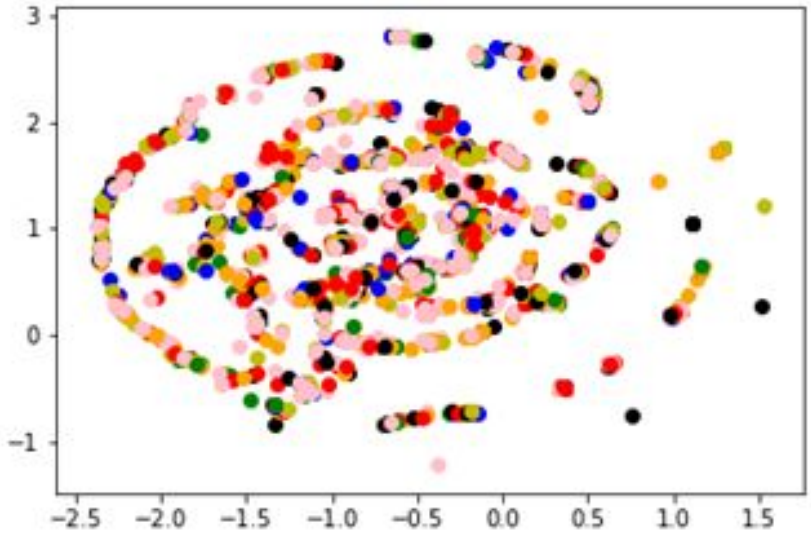
- Learning using EM algorithm
  - E-step: Learns the object representation
  - M-step: Model Local dependency
- Training: 140 papers, Validation: 500 papers, Test: 1000 papers
- Test Accuracy: 84%

GNN Feature Selection

- The method uses GNN to select feature which can also captures the link information which most feature selector doesn’t.
- This is based on Gumbel Softmax in learning instead of softmax.
  - Gumbel softmax is just a way to sample one-hot vector from a categorical distribution. To understand this, it is just a way to make one-hot encoding differentiable which in normal softmax, this wouldn’t be the case.
- Dimension of the paper word: 2708x1432
- Gumbel Matrix: 1432x225
- Final Matrix: 2708x225

GNN Feature Selection

- TSNE of the results of the words.



CRF-Suite

Sklearn-CRFSuite provides a fast implementation of CRF in Python. For training and testing purposes, the dataset needed to be modified.

Approach 1:

- Hiding dependencies
- Accuracy ~92%

Approach 2:

- Hiding Paper Information
- Accuracy ~84%

SPK CRF

Goal: Maximize conditional log likelihood p(y|x).

Model: Sequences of length 3 (including start node), features of length 1432 for each node (excluding the start node (which has no feature transition entries))

Parameters: T (transition) and E(emission) matrices of sizes 8x7 and 1432x7

Problem: Learning and inference under two different cases:

1. Fully labelled sequence (both classes known)
2. One class hidden sequence

Note: When classes of both papers are hidden we do not learn anything.. So we just ignored such citation links

Formulae to be Implemented (from class notes)

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) \quad p(\mathbf{y}, \mathbf{w}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{w}_c, \mathbf{y}_c; \theta_p)$$

$$\Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) = \exp\left\{ \sum_{k=1}^{K(p)} \lambda_{pk} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) \right\} \qquad p(\mathbf{w}|\mathbf{y}, \mathbf{x}) = \frac{1}{Z(\mathbf{y}, \mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{w}_c, \mathbf{y}_c; \theta_p)$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) \quad Z(\mathbf{y}, \mathbf{x}) = \sum_{\mathbf{w}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{w}_c, \mathbf{y}_c; \theta_p)$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \sum_{\mathbf{w}} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\mathbf{x}_c, \mathbf{w}_c, \mathbf{y}_c; \theta_p) = \frac{Z(\mathbf{y}, \mathbf{x})}{Z(\mathbf{x})}$$

Likelihood and Partial Derivative for Learning

$$\ell(\theta) = \sum_{C_p \in \mathcal{C}} \sum_{\Psi_c \in C_p} \sum_{k=1}^{K(p)} \lambda_{pk} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) - \log Z(\mathbf{x})$$

$$\frac{\partial \ell}{\partial \lambda_{pk}} = \sum_{\Psi_c \in C_p} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) - \sum_{\Psi_c \in C_p} \sum_{\mathbf{y}'_c} f_{pk}(\mathbf{x}_c, \mathbf{y}'_c) p(\mathbf{y}'_c | \mathbf{x})$$

$$\ell(\theta) = \log p(\mathbf{y}|\mathbf{x}) = \log \sum_{\mathbf{w}} p(\mathbf{y}, \mathbf{w}|\mathbf{x})$$

$$\frac{\partial \ell}{\partial \lambda_{pk}} = \sum_{\Psi_c \in C_p} \sum_{\mathbf{w}'_c} p(\mathbf{w}'_c | \mathbf{y}, \mathbf{x}) f_k(\mathbf{y}_c, \mathbf{x}_c, \mathbf{w}'_c) - \sum_{\Psi_c \in C_p} \sum_{\mathbf{w}'_c, \mathbf{y}'_c} p(\mathbf{w}'_c, \mathbf{y}'_c | \mathbf{x}_c) f_k(\mathbf{y}'_c, \mathbf{x}_c, \mathbf{w}'_c)$$

Inference

For both cases described below, we maintain a dictionary of [paper_id, array of predictions]. Every time an inference on its class is made (this occurs when we encounter a sequence containing the paper) we add the label to the list. For accuracy calculation a majority vote is taken of predicted labels.

Case 1: Both papers in citation sequence are non_hidden

Calculate p(w,y|x) for each of the seven classes and take max. Hide both papers in alternation.

Case 2: One paper in citation sequence is hidden

Calculate p(w|y,x) for each of the seven classes and take max.

Results (SPK CRF with 1432 features)

% hidden labels	Non-hidden class prediction	Hidden class prediction	Log likelihood
5	85.63%	89.23%	-17535
10	85.29%	85.44%	-16807
15	85.32%	85.38%	-16443
20	86.03%	85.63%	-15920
25	83.82%	86.32%	-15075

Results (SPK CRF with 225 features)

% hidden labels	Non-hidden class prediction	Hidden class prediction	Log Likelihood
5	86.95%	83.20%	-17588
10	86.13%	83.58%	-16959
15	85.17%	86.78	-16268
20	84.83%	86.30%	-15558
25	84.62%	86.72%	-14851

Results (NN, GMNN, CRF-Suite)

Model	Test Accuracy
NN (without dropout and batch normalization)	72.7%
NN (with dropout, without batch normalization)	75.6%
NN (with dropout, with batch normalization)	73.6%
GMNN (semi-supervised learning)	84%
CRF-Suite (supervised learning)	92%
CRF-Suite (removing paper and link completely)	84%

Conclusion

We have successfully explored Graphical Markov Neural Network and Conditional Random Fields which both provide very good results. We have also explored the use of embedding for featurization.

We find that combining statistical relational learning with neural networks, indeed, provides the best of both worlds, and provides robust modelling of graphical data.

We also find that CRF is a very good way to model the paper citation network. Even with a sequence length of 2, by majority vote we are able to decide the label of an unknown paper by local information from its neighbours.

References

[1] Cora Dataset, Relational Dataset Repository, URL: https://relational.fit.cvut.cz/dataset/CORA

[2] Meng Qu and Yoshua Bengio and Jian Tang. “GMNN: Graph Markov Neural Networks.” arXiv, 2019. doi: 1905.06214.

[3] Nocedal, Jorge. "Updating Quasi-Newton Matrices with Limited Storage." Mathematics of Computation 35, no. 151 (1980): 773-82. Accessed March 10, 2020. doi:10.2307/2006193.

[4] DeepGraphLearning. “DeepGraphLearning/GMNN.” *GitHub*, github.com/DeepGraphLearning/GMNN.

[5] Bhaskar, Deepak, et al. “Feature Selection and Extraction for Graph Neural Networks.” *ArXiv.org*, 25 Oct. 2019, arxiv.org/abs/1910.10682.

[6] Jang, Eric, et al. *Categorical Reparameterization with Gumbel-Softmax*. 5 Aug. 2017, arxiv.org/pdf/1611.01144.pd