Statistical Machine Learning Project1 Report

Hongyu Chen 1062897

15/04/20

1. Problem

The main problem in this project is finding edge-relationship in an academic network. Each node in this network represents an author with some information about his/her academic career, and we need judge the existence of a given edge from an initial network.

This issue is practical because there are many similar situations in daily life. For instance, if we want to arrive on city A from city B, the best choice is driving directly if there is a highway between them. However, what if we don't know whether this highway exists, then the algorithm implemented in this project will give us one credible answer. As illustrated in Fig1.

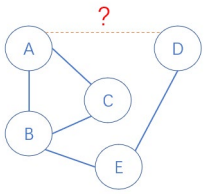

Fig1. Above is an incomplete network, the blue lines are known real edges. And we need to decide the red dash line whether exists.

2. Final Solution

The most important factor to solve this problem is the feature of each node, if we can dig out efficient features among those nodes, it will be convenient to give the right answer.

In my project, I utilized all the given features: For each author(node), the time of publication of their first and last paper, the number of their total papers, the keywords contained in their papers and the venues they have been to. Additionally, I defined one concept: 'Depth-x co-author', a useful feature, to improve the accuracy of my algorithm. The description of 'Depth-x co-author' is demonstrated in Fig2.

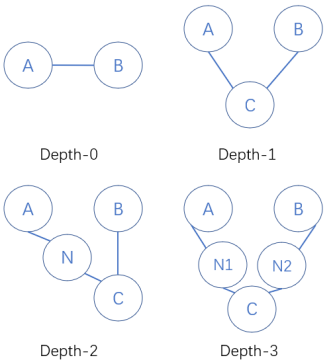

Fig2. When A and B connected directly, they are 'Depth-0 coauthor' with each other. In other sub figures, C is 'Depth-1/2/3 coauthor' with A or B respectively.

The details of my algorithm are shown below:

Rule 1:

If two nodes A and B have 'Depth-x coauthor', the smaller x is and the number of

this kind coauthor is greater, the more probability A and B have connection. In light of the complexity to compute 'Depth-x', we only take 'depth_0/1/2' into consideration.

$$Prob\_1 = \sum_{k=0}^{2} w_k * num\_of\_depth\_k$$

Rule 2:

For feature 'first publication', 'last publication', 'number of papers', we can establish 'active interval' (time between first and last) and 'average publication per year'. For given two nodes, the more overlapping between their 'active interval' and closer their 'average publication per year', the more probability A and B have connection.

$$Prob\_2 = w_{Interval} * overlapping(Interval\_A, Interval\_B) + w_{PPY} * Sim(PPY\_A, PPY\_B)$$

Where Interval_A, Interval_B are 'active inteval' of A and B respectively, PPY_A and PPY_B are 'average publication per year' of A and B.

Rule 3:

For the 'Keyword' and 'Venue' two features, they can be encoded as one-hot coding. I recorded the number of same digits and different digits. The more same digits and less different digits between two nodes, the more probability they have connection.

$$Prob\_3 = w_{Keyword} * (same\_digits - diff\_digits) + w_{Venue} * (same\_digits - diff\_digits)$$

Final decision:

$$Prob = F(\sum_{k=1}^{3} w_k * Prob\_k)$$

The AUC can reach 0.9 with these filtration rules.

3. Other Approaches

- Naïve Prediction
  We can find 366 test edges indeed exist in raw graph, also we know there are 1000 real edges and 1000 fake edges. Hence if we judge the 366 edges as real and remaining as fake, the accuracy in theory is around 0.68 (1366/2000).
- Softmax
  This problem can also be abstracted as a classification problem, softmax layer is convenient to give the probability of existence of given edge. However, since the training data almost all are positive samples, the neural network will easily be overfitted without adding negative samples manually. After several tuning processes (adding negative samples), the AUC increases from 0.58 to 0.83.

4. Conclusion

The final solution to solve this problem is finding efficient features and applying proper filtration rules. With this approach, the AUC on public testing set can reach 0.9, far better than the Naïve Prediction and simple Softmax neural network.