COMS 572: Lab #1

September 28, 2018 by 17:00pm

Professor Jin Tian

Le Zhang

Problem 1

The sample intranets 1, 5, and 7 have been randomly created, with the following propensities (from weakest to strongest):

- the more QUERY words on a page, the more likely the links on that page lead to the goal node
- the more QUERY words in the hypertext associated with a hyperlink, the more likely that hyperlink leads to the goal node
- the more consecutive and in numerical order QUERY words there are in a hyperlink, the more likely that hyperlink leads to the goal node (eg, seeing QUERY1 QUERY2 QUERY3 is a very good indicator)

QUESTIONS:

2a. Use the above information to devise a heuristic function for use in best-first search. Describe your motivation for your heuristic. Note that unlike the standard approach where the heuristic is applied to the next state, here we want to use our heuristic to decide which hyperlink to 'click on' (fetch) next. This means that your heuristic function scores each arc (hyperlink) coming out of the current node and not each child node. Is your heuristic admissible? Explain why or why not. (You're not required to write an admissible heuristic.)

Answer:

As described in the question, we have 3 levels of propensities from weakest to strongest. Thus, we want to assign scores to those links based on those propensities and then rank them by those scores. Those propensities have different importance so we need to put different weight on them. In the mean while, we don't want to mix scores from different levels. By doing some experiments, I found that the number of QUERY words on a single page is no more than 2 digits and number of QUERY words in the hypertext should be 1 digit only. Here I use x_1, x_2, and x_3 to reperesent the number of QUERY words on a page, number of QUERY words in hypertext, and length of consecutive QUERY words in numerical order, respectively. Therefore, for a node A my heuristic function is like:

$$h(A) = x_1 + 100x_2 + 1000x_3$$

In my opinion, my heuristic function is admissible. Because the strongest propensity takes control anytime before the weaker ones. And different levels of scores never mix with each other so links with higher probability of finding the goal will always be explored first. As a result, the heuristic function never overestimate the cost to reach the goal in this problem. Therefore, I'd say it is admissible.

2c. How well did your heuristic work on the sample intranets

Answer:

Here is the summary of results I got from my program:

intranet#	BFS	DFS	BEST	BEAM	⇐ My results	BFS	DFS	BEST
1	91/4	58/15	18/7	18/7		91/4	58/15	19/7
5	88/8	42/10	25/9	25/9		88/8	42/10	29/9
7	56/6	12/9	27/6	27/6	Sample Results ⇒	56/6	12/9	27/8

As we can see from the results, my heuristic function (BEST search algorithm) outperforms other algorithms. Compared with sample results given in the dataset, my heuristic algorithm outperforms the sample results as well (Intranet#1: [18/7 vs. 19/7]; Intranet#5: [25/9 vs. 29/9]; Intranet#7: [27/6 vs. 27/8]). The reason BEAM algorithm does not improve from BEST algorithm is that the intranets given are a little small and the beam width (20) is kind of large for this dataset. If we have a larger dataset or a smaller beam width, BEAM will outperform BEST to some degree.

Problem 1 continued on next page. . .

Solution Paths Found:

```
***** Intranet_1 *****
========BFS========
result  [’page1.html’, ’page18.html’, ’page29.html’, ’page99.html’, ’page50.html’]
========DFS========
result  [’page1.html’, ’page23.html’, ’page60.html’, ’page39.html’, ’page78.html’,
’page25.html’, ’page42.html’, ’page84.html’, ’page30.html’, ’page68.html’, ’page93.html’,
’page87.html’, ’page79.html’, ’page2.html’, ’page83.html’, ’page50.html’]
========BEST========
result  [’page1.html’, ’page14.html’, ’page69.html’, ’page87.html’, ’page79.html’,
’page2.html’, ’page83.html’, ’page50.html’]
========BEAM========
result  [’page1.html’, ’page14.html’, ’page69.html’, ’page87.html’, ’page79.html’,
’page2.html’, ’page83.html’, ’page50.html’]

***** Intranet_5 *****
========BFS========
result  [’page1.html’, ’page40.html’, ’page99.html’, ’page89.html’, ’page87.html’,
’page96.html’, ’page95.html’, ’page72.html’, ’page62.html’]
========DFS========
result  [’page1.html’, ’page40.html’, ’page99.html’, ’page5.html’, ’page97.html’,
’page68.html’, ’page48.html’, ’page7.html’, ’page95.html’, ’page72.html’, ’page62.html’]
========BEST========
result  [’page1.html’, ’page40.html’, ’page99.html’, ’page88.html’, ’page19.html’,
’page42.html’, ’page35.html’, ’page95.html’, ’page72.html’, ’page62.html’]
========BEAM========
result  [’page1.html’, ’page40.html’, ’page99.html’, ’page88.html’, ’page19.html’,
’page42.html’, ’page35.html’, ’page95.html’, ’page72.html’, ’page62.html’]

***** Intranet_7 *****
========BFS========
result  [’page1.html’, ’page48.html’, ’page71.html’, ’page57.html’, ’page62.html’,
’page61.html’, ’page86.html’]
========DFS========
result  [’page1.html’, ’page48.html’, ’page71.html’, ’page57.html’, ’page90.html’,
’page39.html’, ’page60.html’, ’page11.html’, ’page78.html’, ’page86.html’]
========BEST========
result  [’page1.html’, ’page48.html’, ’page71.html’, ’page57.html’, ’page62.html’,
’page61.html’, ’page86.html’]
========BEAM========
result  [’page1.html’, ’page48.html’, ’page71.html’, ’page57.html’, ’page62.html’,
’page61.html’, ’page86.html’]
```

Problem 2

Nodes visited and path found with description of query words and the domain if you did WWW adventure.
Answer:

- Query words: ‘Paolo Cesare Maldini’
- Website domain: ‘https://en.wikipedia.org’
- Starting page: ‘/wiki/Sports’
- Searching algorithm: Beam Search with beam_width = 20
- Heuristic function: if any of these words appears in the hypertext of a link, take a sum of the corresponding numbers as the score of that link: {’soccer’: 3, ’football’: 3, ’paolo’: 5, ’maldini’: 5, ’serie a’: 5, ’italy’: 4, ’milan’: 5, ’world cup’:5, ’association’: 3}
- Nodes visited: 9
- Path length: 5
- Path found:

```
['/wiki/Sports',
'/wiki/Association_football',
'/wiki/Category:Laws_of_association_football',
'/wiki/Penalty_shoot-out_(association_football)',
'/wiki/Italy_national_football_team',
'/wiki/Paolo_Maldini']
```

Discussions:

I tried to use BFS and DFS for the WWW searching in the first place but they didn’t work well. It was because there are thousands of links on each of the Wikipedia webpages. As a result, it may take hours even days before BFS and DFS can find the target page.

Best-first search seems to be a good way to do it, however, thousands of links eats the memory rapidly. Therefore, eventually, I picked beam search to finish this task.

The fact was, when I was using BFS, it visited more than 20,000 pages and the goal page is still not reached. On the other hand, with beam search, only 9 pages were visited before we get to the goal. It is effective and efficient as long as you have the correct heuristic function.

Just for fun, I also searched for query words ‘Cristiano Ronaldo dos Santos Aveiro’, started from page ‘/wiki/Main_page’, with similar heuristic function but different values to calculate scores:

```
{'soccer': 3, 'football': 3, 'cristiano': 5, 'madrid': 5, 'la liga': 5,
                 'portugal': 4, 'ronaldo': 5, 'world cup':5, 'association': 3}
```

At this time, because it started from a more neutral page “main page”, it took longer to get to the goal. It visited 35 pages and the length of solution path was 25. It is still a good result which means the heuristic function is very effective and beam search works perfectly with real world cases.

In summary, the beam search that I used worked fine with real world web searching. There is enough evidence to prove that the heuristic function I applied to my algorithm is effective in practical use.