Report of Softskills Seminar

Article : K-Nearest Neighbors in uncertain graph

Student : ZHAO Mengzi

The uncertain graph consists of nodes and edges with probability, it has $2^{number\ of\ edges}$ possible worlds because the presence of edges in the graph depends on their probability. The application of uncertain graphs exists everywhere in our life, for example, in the mobile network, connections between mobile phones are uncertain, so we can present mobiles by nodes of the uncertain graph and connections by edges with probability, the fundamental problem is to find the K-Nearest Neighbors in uncertain graph, this article presents some methods to resolve this problem, in practice, we can use these methods to find the five mobile phones which are most easiest to connect around us, our ten best friends and so on.

Before studying those methods, we need to know how to calculate some values.
The first one is how to calculate the probability of a possible world, we need to use the product of the probability of presence of edges which are present in the possible world multiplied by the product of the probability of absence of edges which are absent in the graph.
The second one is reliability, it presents the probability that one of possible paths between two nodes exist, so we just need to use 1 minus the product of the probability of the inexistence of paths.
The MostProbPath is a function to obtain the length of the path by executing the Dijkstra shortest path after creating another graph with the same nodes and edges with weights (weight can be calculated by the probability) of the original graph, but this function has some disadvantages, the length of the path can be arbitrary small, we cannot also make sure that the distance with the largest probability is the shortest distance. To overcome these limitations, we use the third value that we need to know, the distribution of the shortest path between any two nodes in the graph. It is in terms of pairs <d, p>, d is the distance of the path between 2 nodes, p is the probability which is the sum of the probabilities of all possible worlds in which the shortest path distance between any 2 nodes of the graph is exactly d. By using this distribution, we have 3 methods to calculate the probabilistic graph distance. The median distance is to take the distance whose probability is approximate to 1/2 in the distribution, the majority distance is to take the distance whose probability is largest, the expected reliable distance is to find the distance whose probability of the path between 2 nodes (if the path exists in the graph) is calculated by using a special formula. We can notice that the complexity to compute the median distance is very high, because we need to execute the algorithm from point to point in every world and then obtain the median, so this article introduces a way “sampling” to overcome this problem, we need just sample several possible graphs

from the main graph by using the probability and then calculate the median distance.

After knowing all these values, we can study the algorithm of KNN pruning, this article uses the median distance in this algorithm, but using the other ways to calculate the distance is the same principle. We have a graph, we sample some possible graphs from the original graphs, we have a distance D that we want to go in every round and it is incremented in every round by the increment distance, we have also a source and we will find the K nearest neighbors of this source. We execute the Dijkstra algorithm for every possible graph until we reach the distance D. For every node visited, we instantiate or update their distribution, we calculate the median distance between the node visited and the source by using the distribution that we compute previously, if this distance is smaller than the distance D, we add this node in a list which contains the K-Nearest Neighbors, so if the length of the list does not arrive to k, we continue finding the neighbors for the source. Excepting the difference of computing the distance between the method of the median distance and the method of the majority distance, there is just a difference, the result in the list which contains the K nearest neighbor is the final result for the algorithm using median distance, but for the algorithm using majority distance, nodes in that list can be discarded by the nodes who are on the path with larger probability. According to the experiments analysis, authors find methods mentioned in this article are much better than their competitors.

After the presentation, my classmates asked many questions, there are some questions which made me impressive. For example, “For the median distance, we take the distance when the probability is approximate to 1/2, but when the probability is larger than 1/2, how should we do ? So in this case the formula of median distance is incorrect ? ” At that moment, I did not know how to response this question, because when we read a paper, in general, we try to understand the content and we do not doubt the correctness of the content in the article and I think articles provided by our professors should not be wrong, so this thing made me realize that we should not only understand the knowledge but also consider whether it can be applied in other cases, and if it cannot be used in other cases, we need to think how we should modify the algorithm so that it can be applied in a wider domain. For this question, when the probability is more than 1/2, we should not use the method of median distance in this case and we should use the majority distance or expected reliable distance, we cannot say the method of median distance is incorrect, because it can be applied in some cases, when we want to use the median distance, we need just add a condition in the algorithm to limit the case when we cannot use the median distance.

The softskill seminar course gives us a good chance to study knowledge ourselves by reading paper, in general, reading a paper needs to use much time to understand all ideas or algorithms of authors, it is just like when we need to read other people’s code and understand their logic, sometimes it is difficult to understand everything in the paper, this is also a good way to study other people’s thinking logic and enrich our knowledge base.